Int J Software Informatics, Vol.1, No.1, December 2007, pp. 67–84 E-mail: ijsi@iscas.ac.cn
International Journal of Software and Informatics ISSN 1673-7288 http://www.ijsi.org
©2007 by Institute of Software, Chinese Academy of Sciences. All rights reserved. Tel: +86-10-62661040

Entropy-Inspired Competitive Clustering Algorithms*

Daoqiang Zhang[1], Songcan Chen[1], Zhi-Hua Zhou[2]

[1] (Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

[2] (National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract In this paper, the well-known competitive clustering algorithm (CA) is revisited and reformulated from a point of view of entropy minimization. That is, the second term of the objective function in CA can be seen as quadratic or second-order entropy. Along this novel explanation, two generalized competitive clustering algorithms inspired by Renyi entropy and Shannon entropy, i.e. RECA and SECA, are respectively proposed in this paper. Simulation results show that CA requires a large number of initial clusters to obtain the right number of clusters, while RECA and SECA require small and moderate number of initial clusters respectively. Also the iteration steps in RECA and SECA are less than that of CA. Further CA and RECA are generalized to CA-p and RECA-p by using the p-order entropy and Renyi's p-order entropy in CA and RECA respectively. Simulation results show that the value of $p$has a great impact on the performance of CA-p, whereas it has little influence on that of RECA-p.

Key words: competitive clustering; fuzzy c-means; optimal number of clusters; cluster validity; entropy minimization

Zhang DQ, Chen SC, Zhou ZH. Entropy-Inspired competitive clustering algorithms. *Int J Software Informatics*, **2007, 1(1): 67–84. http://www.ijsi.org/1673-7288/1/67.pdf**

1 Introduction

Clustering is the process of grouping data points into classes or clusters so that members of the same cluster are more similar to each other in some sense than to members of other clusters[1,13]. Clustering is useful for exploring the underlying structure of a given data set and has been widely used in many scientific and engineering fields such as pattern recognition, image processing, data mining, etc. Generally, clustering methods can be divided into the following categories: partitional methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods[13]. In this paper, we focus on competitive clustering algorithm (CA), which

* This work was supported by the National Science Foundation of China under Grant Nos. 60496320, 60505004, 60721002, the Jiangsu Science Foundation under Grant No BK2006521, and the Foundation for the Author of National Excellent Doctoral Dissertation of China under Grant No. 200343.
Corresponding author: Daoqiang Zhang, dqzhang@nuaa.edu.cn
Manuscript received 12 Jul. 2007; revised 17 Oct. 2007; accepted 8 Dec. 2007; published online 29 Dec. 2007.

belongs to the partitional clustering algorithms. Other types of clustering methods can be found in an excellent recent survey[13].

The CA algorithm is an extension of the well-known fuzzy c-means (FCM) algorithm. The FCM has the shortcoming that it needs the number of clusters to be predefined. This problem, i.e. determining the optimal number of clusters C, is often addressed by using the general technique of cluster validity[21,23]. A typical solution for cluster validity is to run a given clustering algorithm within a range of C values, and then evaluate the validity of the resulting partition for each value. The partition exhibiting the optimal validity is chosen as the true partition. Another solution for the third problem is the well-known subtractive clustering methods proposed by Yager and Filev[22] and Chiu[6]. Recently, Yu and Cheng[23] give a detailed and theoretical analysis on the upper bound of the optimal number of clusters in fuzzy clustering.

In order to embed the idea of cluster validity into FCM, Frigui and Krishnapuram first introduced an extra term to the original objective function of FCM and derived the CA algorithm[8−9]. In the CA algorithm, there exists a process of competitive agglomerating between clusters, where the clusters with small cardinalities are discarded. That is very similar to classical hierarchical agglomerative clustering algorithm[10]. Boujemaa regarded the added extra term of the CA objective function as a regularization term and gave a novel explanation for the algorithm based on regularization theory, and then suggested a new clustering scheme by using various cluster validity criteria as corresponding regularization terms to replace the extra term[3]. However, there exists a problem of Boujemaa's explanation, that is, the difficulty of choosing the regularization coefficient α. In[3], there is no explicit description on how to determine the value of α. Recently, Medasani and Krishnapuram proposed a similar algorithm as CA called robust agglomerative gaussian mixture decomposition (RAGMD), which introduces an entropy regularization term into classical Gaussian mixture[16−17]. It was reported that the entropy term in RAGMD prevented premature convergence and accelerated convergence as one neared the right number of clusters[16]. However, in[16] only the Shannon entropy is considered.

In this paper, the CA algorithm is revisited and reformulated from a point of view of entropy minimization[4,7,14−15,18−20]. Firstly, we show that the second (extra) term of the objective function in CA can be viewed as quadratic or second-order entropy. We then propose a general entropy-inspired competitive clustering framework, from which two distinct competitive clustering algorithms are derived based on Renyi entropy-like function (RECA) and Shannon entropy-like function (SECA) respectively. Furthermore, the relationship between CA, RECA and SECA are indicated. Finally, we discuss further extensions on CA and RECA through introducing high-order entropy functions. Simulation results on some synthetic and real data sets show that CA requires a large number of initial clusters to obtain the right number of clusters, while RECA and SECA require small and moderate number of initial clusters respectively. Since more initial clusters means more computing time, the computational load of RECA and SECA are less than that of CA.

The rest of this paper is organized as follows. Section 2 briefly reviews the original CA algorithm and gives it an entropy explanation. In Section 3, two entropy-inspired competitive clustering algorithms, i.e. Renyi entropy-like function based competitive clustering (RECA) and Shannon entropy-like function based competitive clustering (SECA), are presented. Section 4 generalizes RECA and SECA to CA-p and RECA-p respectively. Simulation results are reported in Section 5. Finally, Section 6 concludes and discusses on future works.

2 Reformulation of CA

The objective function of the original CA algorithm is defined as (1)[8],

$$J_{CA} = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2} - \alpha \sum_{i=1}^{C}\left[\sum_{k=1}^{N} u_{ik}\right]^{2} \tag{1}$$

where u_{ik} is the membership degree of the k-th data point x_k in the i-th cluster represented by v_i, and it satisfies (2),

$$\sum_{i=1}^{C} u_{ik} = 1, \forall 1 \leqslant k \leqslant N \text{ and } u_{ik} \in [0,1] \tag{2}$$

And d_{ik} is the distance between the k-th data point x_k and the i-th cluster represented by v_i, N is the total number of data points and C is the true number of clusters to be found. The parameter m is a weighting exponent which adjusts the fuzziness and is usually set to 2.

Although CA is proven effective in finding the optimal clusters on some data sets, its second term in the objective function lacks intuitive explanations. In the rest of this section, a novel explanation will be given from the view of entropy minimization.

First, define the average membership of an arbitrary data point in cluster i (also called the *relative fuzzy cardinality* of cluster i) as follows:

$$p_i = \frac{1}{N}\sum_{k=1}^{N} u_{ik}, \forall 1 \leqslant i \leqslant C \tag{3}$$

From (2), u_{ik} take value between 0 and 1, so the value of p_i is also between 0 and 1 and the sum of all p_i is 1. It is worth noting that in some sense, p_i in (3) can be regarded as the probability that a random data point belongs to cluster i. p_i approaching to 1 means that cluster i dominates the whole data set while the other clusters are nearly empty. On the other hand, every p_i equal $1/C$ means that all clusters have the same importance and none of them can be removed arbitrarily. Except for those extreme cases, usually some p_is are notably larger than the others, i.e. some clusters with large p_i values dominate the data set, and thus one can remove the clusters with negligible p_i values.

According to (3), we reformulate the objective function of CA as

$$J_{CA} = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2} + \alpha\left(-\sum_{i=1}^{C} p_i^{2}\right) \tag{4}$$

Now it is obvious that the second term $-\sum_{i=1}^{C} p_i^2$ can be viewed as derived from the quadratic entropy $1-\sum_{i=1}^{C} p_i^2$ [4]. Note that the only difference between these two terms is a constant 1, so they achieve minimum at the same p_i values. The effect of minimizing the second term in (4) is to make the number of clusters as small as possible. Whereas the first term in (4), also called the sum of fuzzy within-class distances, is minimized when the number of clusters is equal to that of data points, i.e. $C = N$. In other words, the effect of minimizing the first term in (4) is equivalent making the number of clusters as large as possible. The parameter α controls the balance between the two terms to find an optimal C value.

3 Generalized CA

The reformulation of the CA algorithm from the view of entropy minimization in the last section inspires a general framework for entropy based competitive clustering. That is, if we replace the quadratic entropy in (4) with any other entropy, different algorithm will be derived. In this section, we investigate using Renyi-entropy and Shannon entropy for competitive clustering.

3.1 Renyi-Entropy Inspired Competitive Clustering (*RECA*)

The objective function in RECA is defined as

$$J_{RECA} = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^m d_{ik}^2 + \alpha\left(-\ln\left(\sum_{i=1}^{C}(p_i+1)^2\right)\right) \tag{5}$$

Here the second term $-\ln\left(\sum\limits_{i=1}^{C}(p_i+1)^2\right)$ is from Renyi's second-order entropy $-\ln\left(\sum\limits_{i=1}^{C}p_i^2\right)$ [18,14]. The function of the constant 1 is to assure $\ln\left(\sum\limits_{i=1}^{C}(p_i+1)^2\right) > 0$, and thus guarantee $\alpha > 0$, as shown in the following (7). Optimizing (5) with respect to u_{ik} under the constraint of (2) results in (6):

$$u_{ik} = \frac{\frac{1}{d_{ik}^{2/(m-1)}}}{\sum\limits_{t=1}^{C}\frac{1}{d_{tk}^{2/(m-1)}}} + \frac{\alpha}{Nd_{ik}^2\sum\limits_{i=1}^{C}(p_i+1)^2}\left(p_i - \frac{\sum\limits_{t=1}^{C}\frac{p_t}{d_{tk}^2}}{\sum\limits_{t=1}^{C}\frac{1}{d_{tk}^2}}\right) \tag{6}$$

Here the coefficient α at iteration l is updated using the following equation

$$\alpha = \eta_0 e^{-l/\tau}\frac{\sum\limits_{i=1}^{C}\sum\limits_{k=1}^{N}u_{ik}^m d_{ik}^2}{\ln\left(\sum\limits_{i=1}^{C}(p_i+1)^2\right)} \tag{7}$$

As in [8], η_0 and τ are set to 1 and 10, respectively. The same settings will be used in the rest of this paper. On the other hand, the update of prototype v_i is the same as that in FCM and is dependent on the distance measure adopted. In this paper, only the Euclidean distance *d(x, y)*=||*x*-*y*||^2 is considered. Thus, the updating equation for prototype v_i is

$$v_i = \frac{\sum\limits_{k=1}^{N} u_{ik}^m x_k}{\sum\limits_{k=1}^{N} u_{ik}^m} \tag{8}$$

Here, x_k.denotes the k-th data point. Based on (6) and (8), an iterative procedure for the RECA algorithm is as follows.

Algorithm 1: The RECA algorithm

```
    Step 1: Fix the maximum number of clusters $C = C_{\max}$; Initialize the membership $u_{ik}$, and set the iteration step $k$=0. Set $\varepsilon_1$ and $\varepsilon_2$ with small positive numbers.
    Step 2: For $i$ =1 to $C$, do:
    Compute the relative fuzzy cardinality $p_i$ using (3);
    If $p_i <\varepsilon_2$, then
    Discard the $i$-th cluster;
    $C = C$–1;
endif
endfor
    Step 3: Update the prototype $v_i$ using (8).
    Step 4: Update the coefficient $\alpha$ using (10).
    Step 5: Update the membership $u_{ik}$ using (9).
    Repeat Steps 2-5 until the maximum change in $u_{ik}$ is less than $\varepsilon_1$.
```

3.2 Shannon-Entropy Inspired Competitive Clustering (*SECA*)

The objective function in SECA is defined as

$$J_{SECA} = \sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^m d_{ik}^2 + \alpha\left(-\sum_{i=1}^{C}(1+p_i)\ln(1+p_i)\right) \tag{9}$$

Here the second term $-\sum\limits_{i=1}^{C}(1+p_i)\ln(1+p_i)$ originates from the Shannon entropy $-\sum\limits_{i=1}^{C} p_i \ln p_i$ [3,17]. Like in RECA, the constant 1 is added for each p_i in the logarithm function to assure $\ln(1+p_i) > 0$, and thus guarantee $\alpha > 0$, as shown in the following (11). Optimizing (9) with respect to u_{ik} under the constraint of (2) results in the following updating equation

$$u_{ik} = \frac{\frac{1}{d_{ik}^{2/(m-1)}}}{\sum\limits_{t=1}^{C}\frac{1}{d_{tk}^{2/(m-1)}}} + \frac{\alpha}{2Nd_{ik}^2}\left(\ln(p_i+1) - \frac{\sum\limits_{t=1}^{C}\frac{\ln(p_t+1)}{d_{tk}^2}}{\sum\limits_{t=1}^{C}\frac{1}{d_{tk}^2}}\right) \tag{10}$$

where the coefficient α is updated using

$$\alpha = \eta_0 e^{-l/\tau}\frac{\sum\limits_{i=1}^{C}\sum\limits_{k=1}^{N} u_{ik}^m d_{ik}^2}{\sum\limits_{i=1}^{C} p_i \ln(1+p_i)} \tag{11}$$

It is worth noting that the method in[3] also used the Shannon entropy in generalized CA algorithm, but the algorithm was derived from the regularization theory. Moreover, there are no explicit descriptions on how to choose the value of the regularization coefficient α. In contrast to it, the SECA algorithm is derived from the principle of entropy minimization and an explicit equation for choosing the parameter α can be given, as shown in (11).

The description of the SECA algorithm is as follows.

```
Algorithm 2: The SECA algorithm
Step 1-3: The same as that in the RECA algorithm.
Step 4: Update the coefficient $\alpha$ using (11).
Step 5: Update the membership $u_{ik}$ using (10).
Repeat Steps 2-5 until the maximum change in $u_{ik}$ is less than $\varepsilon_1$.
```

3.3 Relations between CA, RECA and SECA

For convenience, we first derive the updating equations of the membership u_{ik} and the coefficient α for CA, based on its objective function (4), as follows:

$$u_{ik} = \frac{\frac{1}{d_{ik}^{2/(m-1)}}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^{2/(m-1)}}} + \frac{\alpha}{N d_{ik}^2} \left(p_i - \frac{\sum\limits_{t=1}^{C} \frac{p_t}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}} \right) \tag{12}$$

$$\alpha = \eta_0 e^{-l/\tau} \frac{\sum\limits_{i=1}^{C} \sum\limits_{k=1}^{N} u_{ik}^m d_{ik}^2}{\sum\limits_{i=1}^{C} p_i^2} \tag{13}$$

The equations (6), (10) and (12) all consist of two components. The first components are exactly all the same, and are identical to the membership updating equation in FCM; while the second terms in (6) and (12) are formally very similar, and the item in the bracket can be rewritten as

$$p_i - \frac{\sum\limits_{t=1}^{C} \frac{p_t}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}} = \sum_{t=1}^{C} \frac{\frac{1}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}} (p_i - p_t) \tag{14}$$

Thus (14) can be regarded as the weighted arithmetic mean of differences between the relative fuzzy cardinality of the i-th cluster of interest and the relative fuzzy cardinality of each cluster t. Similarly, the item in the bracket of the second term of (10) can be rewritten as

$$\ln(p_i+1) - \frac{\sum\limits_{t=1}^{C} \frac{\ln(p_t+1)}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}} = \sum_{t=1}^{C} \frac{\frac{1}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}} (\ln(p_i+1) - \ln(p_t+1)) = \ln \left(\prod_{t=1}^{C} \left(\frac{p_i+1}{p_t+1} \right)^{\frac{\frac{1}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}}} \right) \tag{15}$$

Now (15) is the logarithm of the weighted geometric mean of quotients between p_i+1 and p_t+1. Equivalently, a weighted arithmetic mean of differences between the logarithm of p_i+1 and the logarithm of p_t+1.

Substituting (7) into (6), (11) into (10), and (13) into (12) respectively, and ignoring the effect of u_{ik} and d_{ik}, the regularization coefficients (coefficients before the second terms of (6), (10) and (12)) in RECA, SECA and CA approximately change into:

$$R_{RECA}=\frac{\eta_0 e^{-l/\tau}}{N\left(\sum_{i=1}^{C}(p_i+1)^2\right)\ln\left(\sum_{i=1}^{C}(p_i+1)^2\right)} \tag{16a}$$

$$R_{SECA}=\frac{\eta_0 e^{-l/\tau}}{2N\sum_{i=1}^{C}p_i\ln(1+p_i)} \tag{16b}$$

$$R_{CA}=\frac{\eta_0 e^{-l/\tau}}{N\sum\limits_{i=1}^{C}(p_i)^2} \tag{16c}$$

Theorem 1. *Let $0\leqslant p_i\leqslant 1$, and $\sum\limits_i p_i=1$. As in (16), the regularization coefficients in (6), (10) and (12) are denoted as R_{RECA}, R_{SECA}* and *R_{CA} respectively, then $R_{RECA}\leqslant R_{SECA}\leqslant R_{CA}$.*

Theorem 1 indicates the relationship between RECA, SECA and CA. The detailed proof for Theorem 1 can be found in Appendix A. Theorem 1 states that at each updating step of u_{ik}, CA is more affected by the second term than SECA, and SECA is more affected by the second term than RECA. Recall that the second terms in (6), (10) and (12) reflect the relative value between the relative fuzzy cardinality of the i-th cluster of interest and the averaged relative fuzzy cardinalities of other clusters (see (14) and (15)). If the value is bigger than 0, it means that the i-th cluster is predominant in the competition with other clusters; if the value is smaller than 0, it implies that the i-th cluster is in inferior position when competing with other clusters. So, Theorem 1 also implies that CA advocates competition more than SECA, and SECA more than RECA.

4 Further Generalization on CA and RECA

In the objective functions of CA and RECA, the second-order entropy-like function and Renyi's second-order entropy-like function are adopted respectively, as shown in (4) and (5). In fact, more general entropy functions with higher-order can also be used. This section will derive the competitive clustering algorithms based on$p$-order entropy-like function (CA-p) and Renyi's p-order entropy-like function (RECA-p) respectively.

The objective function of the CA-p algorithm is defined as

$$J_{CA-p}=\sum_{i=1}^{C}\sum_{k=1}^{N}u_{ik}^m d_{ik}^2+\alpha\left(-\sum_{i=1}^{C}p_i^p\right) \tag{17}$$

Here the superscript p denotes the order of the entropy-like function but is not to be confused with the relative fuzzy cardinality p_i of the i-th cluster in terms of context. Optimizing (9) with respect to u_{ik} under the constraint of (2) leads to the following updating equation for u_{ik}

$$u_{ik} = \frac{\frac{1}{d_{ik}^{2/(m-1)}}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^{2/(m-1)}}} + \frac{\alpha}{N d_{ik}^2} \left(p_i^{p-1} - \frac{\sum\limits_{t=1}^{C} \frac{p_i^{p-1}}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}} \right) \tag{18}$$

The coefficient α is updated again using

$$\alpha = \eta_0 e^{-l/\tau} \frac{\sum\limits_{i=1}^{C} \sum\limits_{k=1}^{N} u_{ik}^m d_{ik}^2}{\sum\limits_{i=1}^{C} p_i^p} \tag{19}$$

Similarly, define the objective function of RECA-p as follows

$$J_{RECA-p} = \sum_{i=1}^{C} \sum_{k=1}^{N} u_{ik}^m d_{ik}^2 + \alpha \left(-\ln \left(\sum_{i=1}^{C} (p_i + 1)^p \right) \right) \tag{20}$$

Optimizing (20) with respect to u_{ik} under the constraint of (2) leads to

$$u_{ik} = \frac{\frac{1}{d_{ik}^{2/(m-1)}}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^{2/(m-1)}}} + \frac{p\alpha}{2N d_{ik}^2 \sum\limits_{i=1}^{C} (p_i + 1)^p} \left(p_i^{p-1} - \frac{\sum\limits_{t=1}^{C} \frac{p_t^{p-1}}{d_{tk}^2}}{\sum\limits_{t=1}^{C} \frac{1}{d_{tk}^2}} \right) \tag{21}$$

Here the coefficient α is updated using the following equation

$$\alpha = \eta_0 e^{-l/\tau} \frac{\sum\limits_{i=1}^{C} \sum\limits_{k=1}^{N} u_{ik}^m d_{ik}^2}{\ln \left(\sum\limits_{i=1}^{C} (p_i + 1)^p \right)} \tag{22}$$

When p is set to 2, CA-p degenerates to CA and RECA-p degenerates to RECA, respectively. The descriptions of CA-p and RECA-p are similar to those of CA and RECA.

5 Experiments

5.1 Data Sets

In this section, the performances of CA, RECA, SECA, CA-p and RECA-p are compared six real data sets from the UCI machine learning repository[2]. Table 1 lists the information of the experimental data sets. Here *Breast* denotes the 'breast-cancer-wisconsin' data set whose first attribute has been discarded, and *Pid* denotes the 'pima-indians-diabetes' data set.

Table 1 Data sets used in the experiments

Data sets information	Bupa	Pid	Breast	Ionosphere	Wine	Soybean
True number of clusters	2	2	2	2	3	4
Number of data points	345	768	683	351	178	47
Data dimension	6	8	9	34	13	35

In all the experiments, the parameter ε_1 is set to 0.001 and the exponent m is set to 2. For all the five algorithms, experiments are performed for a range of values of $C_{\max}$ and the parameter ε_2 is set to $1/C_{\max}$. A standard FCM randomly initialized under the constraint of (2) is run for five iterations, and then the memberships after five iterations are used as the initial membership u_{ik} for CA, RECA, SECA, CA-p and RECA-p. Euclidian distance measure is used in the objective functions of the above algorithms.

5.2 *Results on CA, RECA and SECA*

In this subsection, the performances of CA, RECA and SECA are compared. For each value of $C_{\max}$, 100 independent runs of CA, RECA and SECA are performed with different initializations, and the percent of matches are counted, where the word 'match' implies that the numbers of clusters found respectively by CA, RECA and SECA are equal to the true number of clusters of the data set. That is, the percent of repeated runs that came up with the true number of cluster in the actual data is computed. At the same time, the iteration steps used by the algorithms and their resulting cluster validity are also considered. The well-known Xie-Beni index is used to measure the cluster validity, which is defined as[21]

$$V_{XB} = \frac{\sum_{i=1}^{C}\sum_{k=1}^{N} u_{ik}^2 \left\|x_k - v_i\right\|^2}{N\left(\min_{j\neq i}\left\|v_j - v_i\right\|^2\right)} \tag{23}$$

Here the numerator component denotes the fuzzy within-cluster distances and the denominator component represents the between-cluster distances. The equation (23) reflects an intuitive idea that when a data set is well partitioned its within-cluster distances values should be as small as possible, whereas its between-cluster distances should be as large as possible. Thus for a good partition V_{XB} should be as small as possible.

Figure 1 gives the percent of finding the true number of clusters by CA, RECA and SECA, respectively, under a series of $C_{\max}$ values. In order to further compare the performances of the three algorithms when they all find the true number of clusters, we compute the averaged iteration steps and the cluster validity index V_{XB} under different $C_{\max}$ values when the algorithms find the true number of clusters. Table 2 shows the corresponding results.

Figure 1 indicates that almost for every data set, the distribution of the appropriate $C_{\max}$ values for the three algorithms are different. Generally, RECA, SECA and CA find the true number of clusters with high possibility when using small, moderate and large initial $C_{\max}$ values, respectively. This phenomenon is more apparent on *Bupa* and *Pid*, where there exist clear boundaries between the regions of the appropriate $C_{\max}$ values for the three algorithms. Within their own regions, the percent of repeated runs that came up with the true number of cluster in the actual data is nearly one, whereas outside their own regions, the ratio drops to zero rapidly. Recall that larger $C_{\max}$ values naturally lead to more computing time in the early iterations, the computational cost of CA is bigger than that of RECA and SECA, which is verified by the left part of Table 2. The right part of Table 2 shows that the cluster validity indices are very close.

Table 2 Averaged iteration steps and the cluster validity index V_{XB} of CA, RECA and SECA

Data sets	Iteration steps			Cluster validation (V_{XB})		
	CA	RECA	SECA	CA	RECA	SECA
Bupa	69	42	61	0.12	0.12	0.12
Pid	67	43	59	0.12	0.12	0.12
Breast	52	27	43	0.11	0.11	0.11
Ionosphere	63	37	54	0.71	0.71	0.71
Wine	66	25	49	0.13	0.13	0.12
Soybean	54	25	45	1. 78	1.78	1.79

Why does this interesting phenomenon happen? Why can different $C_{\max}$ values bring so great differences for CA, RECA and SECA? The reason may be partially ascribed to the different entropy-like functions used in CA, RECA and SECA. In Section 3.3, it has been shown that CA advocates competition more than SECA, and SECA more than RECA. When $C_{\max}$ is apparently larger than the true number of clusters, the active competition between clusters in CA can rapidly remove most clusters in inferior position in the early stage, and only support the remaining few clusters for final competition in later stage, which can avoid getting into local minimum to some degree. But the under-competition in RECA can easily lead to getting into local minimum, i.e. finding more clusters than the true number of clusters. On the other hand, when $C_{\max}$ is only a little bigger than the true number of clusters, the serious competition in CA may lead to over-competition, i.e. real clusters are also discarded. Whereas RECA can effectively deal with this case because the competition in the early stage in RECA is not as strong as that in CA. Therefore, it seems that SECA achieves a tradeoff between CA and RECA.

To verify the above conjecture, CA, RECA and SECA are run on *Bupa* under $C_{\max}$ =30, 17, and 8 respectively, and the numbers of clusters found by the three algorithms at each iteration are correspondingly plotted in Fig. 2 (a), (c) and (e). It can be seen that CA, SECA, and RECA correctly find the true number of clusters with $C_{\max}$ =30, 17, and 8 respectively. One can also infer this result from Fig. 1 (a), where the percent of finding the true number of clusters of CA, SECA and RECA under 30, 17 and 8, respectively, are nearly one. Fig.2(a), (c) and (e) also show that when finding the true number of clusters, RECA and SECA need less iteration than CA before ceasing, i.e. the maximum change in u_{ik} is less than ε_1, which coincides with the left part of Table 2. Fig.2 (b), (d) and (f) show values of the regularization coefficients (see (16)) in CA, RECA and SECA at each iteration when $C_{\max}$=30, 17 and 8 respectively. Fig.2 clearly shows that when $C_{\max}$ is small, the over-competition makes CA produce clustering result with only one cluster rapidly. Whereas for large values of $C_{\max}$, the under-competition makes RECA generate clustering result with more clusters than the ground truth. The performance of SECA is between RECA and CA, which tries to seek a tradeoff.

Figure 1 has shown that CA, RECA and SECA have their unique regions for $C_{\max}$ values where they can find the right number of clusters with the highest possibility. Here these regions are called as *working regions*, which means that the algorithms work well within those regions while may fail outside the regions. As can be seen

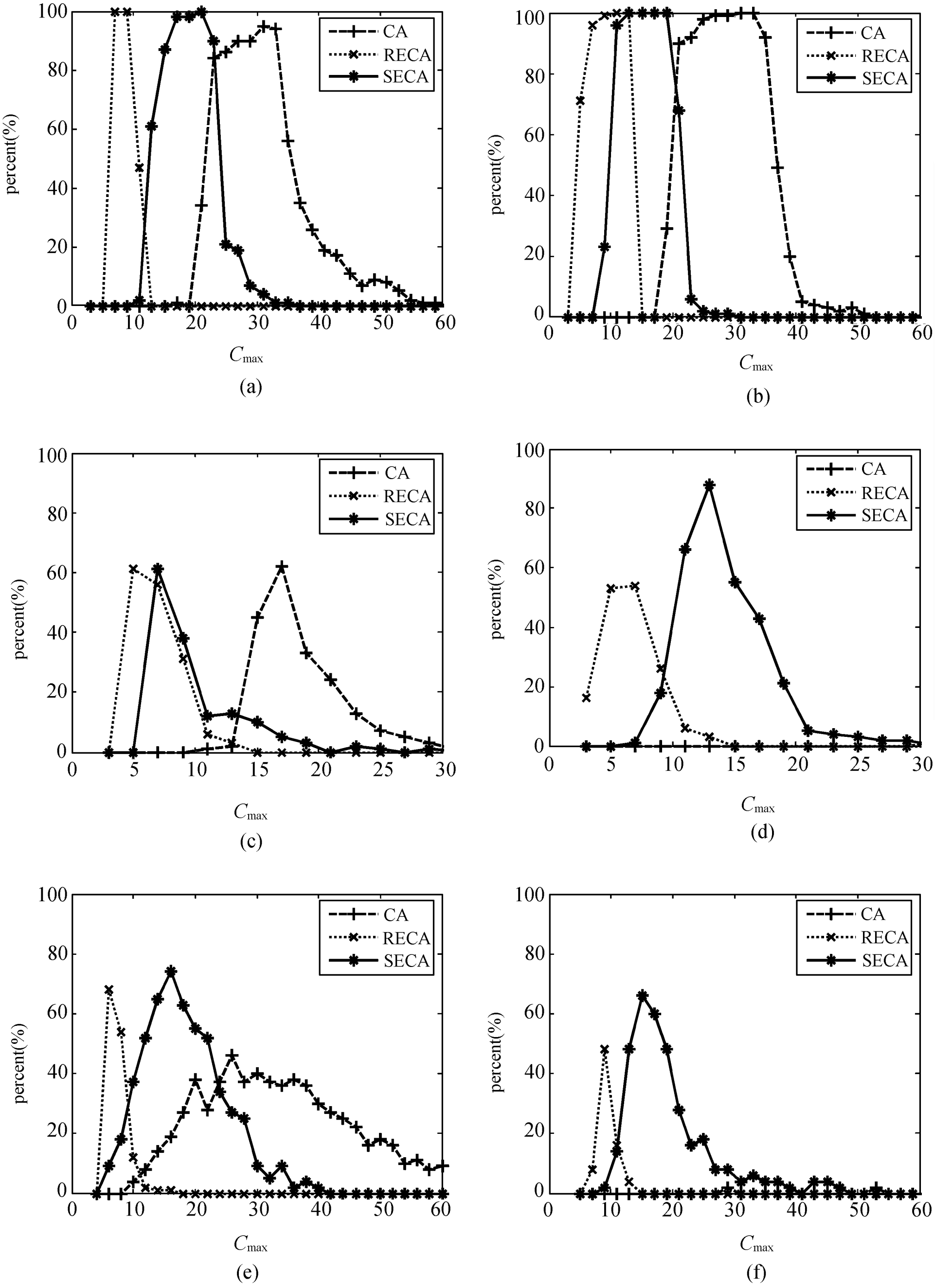
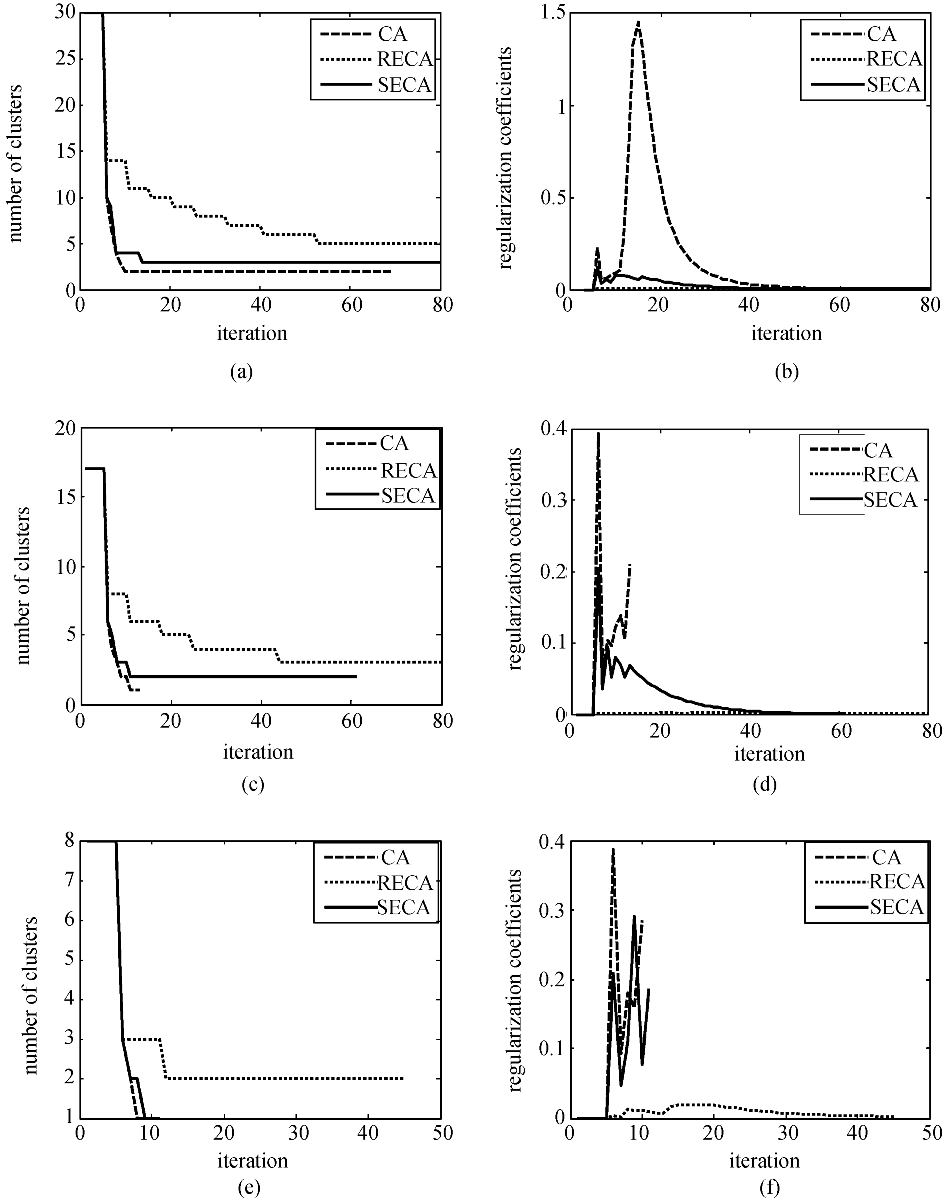


Figure 1. Percent of repeated runs that came up with the true number of cluster in the actual data for a range of $C_{\max}$ values. (a) On *Bupa*. (b) On *Pid*. (c) On *Breast*. (d) On *Ionosphere*. (e) On *Wine*. (f) On *Soybean*

from Fig.1 that, as $C_{\max}$ increases, RECA, SECA and CA come into their working regions in succession. When $C_{\max}$ approaches some large value, none of the three algorithms can find their right number of clusters any more. Fig.1 also indicates that the working region of RECA is narrower than those of both SECA and CA, and thus RECA is more sensitive to the choice of $C_{\max}$. Table 3 computes the areas of each curve (CA, RECA and SECA) in Fig.1. The areas of curves reflect the whole likelihood of finding the true number of clusters by the three methods for a range of



(a) (b)

(c) (d)

(e) (f)

Figure 2. Results on *Bupa* vs. iteration. (a), (c) and (e) are number of clusters found by CA, RECA and SECA at each iteration when $C_{\max}$=30, 17 and 8 respectively. (b), (d) and (f) are values of regularization coefficients in CA, RECA and SECA at each iteration when $C_{\max}$=30, 17 and 8 respectively

$C_{\max}$ values. It can be seen from Table 3 that the averaged likelihood of SECA is similar as that of CA and both are much higher than that of RECA.

Table 3 Are a under the curve of Fig.2

Data sets	CA	RECA	SECA
Bupa	7.71	2.47	5.89
Pid	8.87	4.66	5.97
Breast	1.96	1.57	1.46
Ionosphere	0.01	1.58	3.08
Wine	6.38	1.38	5.42
Soybean	0.04	0.76	3.46
Averaged	4.16	2.07	4.21

Finally, Fig.3 plots the Xie-Beni index of RECA, SECA and CA for a range of $C_{\max}$ values on *Bupa* and *Pid*. Compared with Fig.1, the regions corresponding to the lowest part of the curves of Fig.3 are very close to the corresponding working regions of RECA, SECA and CA in Fig.1 (a) and (b). On the other hand, Fig.4 plots the curves of the most frequent number of clusters obtained by the three algorithms for a range of $C_{\max}$ values on *bupa* and *Pid*. Fig.4 confirms that the competitive ability among clusters of SECA is weaker than that of CA and better than that of RECA. It is worthy noting that both Fig.3 and Fig.4 are generated without knowing the true number of clusters of data sets.

Both Fig.3 and Fig.4 can be used to estimate the working regions of CA, RECA and SECA. From Fig.3, one can find the lowest and widest stable regions. Here the stable region means that when $C_{\max}$ take values from that region, the cluster validation indexes have little changes. For example, the lowest and widest stable regions of Fig.3(a) and (b) are 5-33 and 5-40 respectively. Once that region is found, a natural method to combine the results of RECA, SECA and CA is as follows: 1) when $C_{\max}$ is between that region (e.g. 5-33 for Fig. 3 (a)), we first find the stable sub-regions for RECA, SECA and CA (e.g. the stable sub-regions in Fig. 3(a) are 5-9, 10-22 and 22-33 respectively), and then use results of RECA, SECA and CA in their respective stable sub-regions as the final results; 2) when $C_{\max}$ takes values

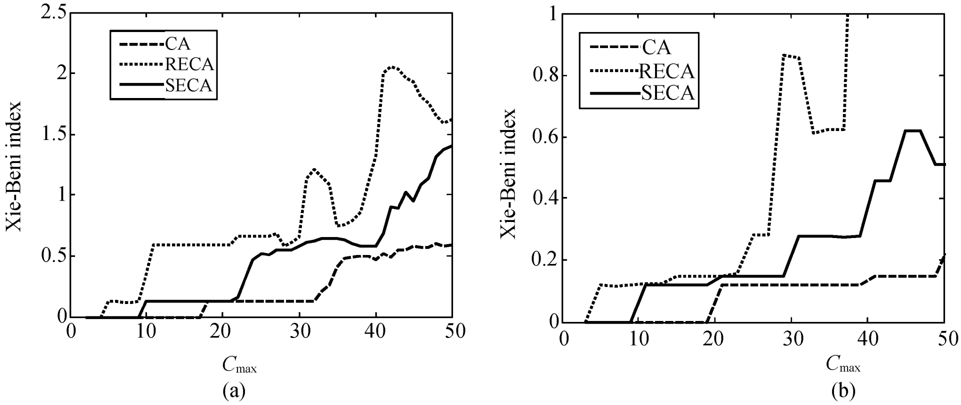


Figure 3. Cluster validation index of CA, RECA and SECA. (a) On *Bupa*. (b) On *Pid*

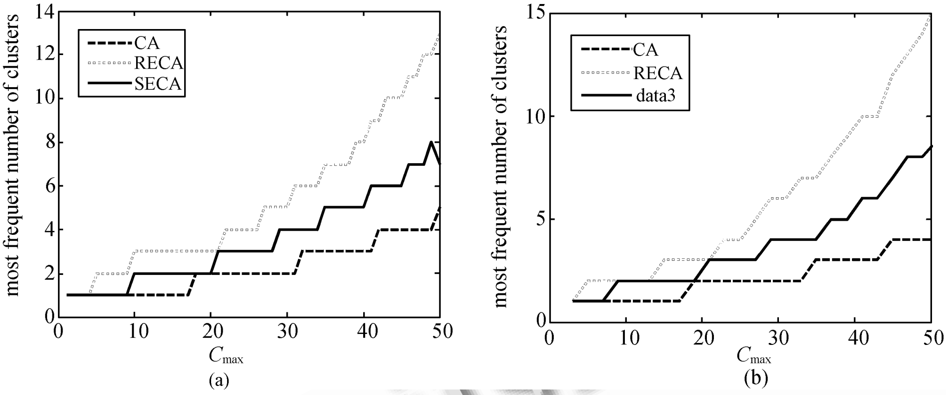


Figure 4. Most frequent number of clusters obtained by CA, RECA and SECA. (a) On *Bupa*. (b) On *Pid*

before that region (e.g. <5 for Fig. 3(a)), we use the results of RECA as the final results; 3) when $C_{\max}$ takes values after that region (e.g. >33 for Fig. 3(a)), we use the results of CA as the final results. Fig.5 shows the combining results on *Bupa* and *Pid*. It can be seen from the figure that the combining method effectively use their respective advantages of RECA, SECA and CA.

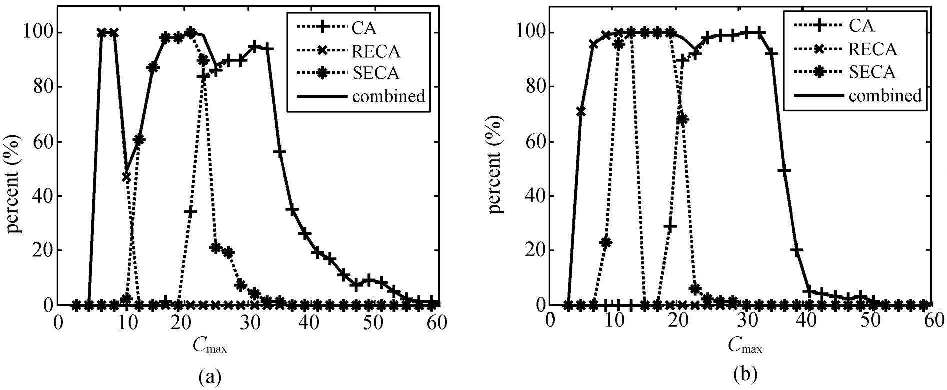


Figure 5. Results of combining CA, RECA and SECA. (a) On *Bupa*. (b) On *Pid*

5.3 Results on CA-p and RECA-p

In this subsection, the performances of CA-p and RECA-p are tested. For both algorithms, the parameter p is varied from 1.1 to 2.5 with the step size of 0.1. For each data set, experiments are performed under a series of different values of $C_{\max}$ and then the results are averaged. Concretely, the $C_{\max}$ values are from 5 to 14 for *Bupa*, *Breast*, *Pid* and *Ionosphere*.,7 to 16 for W*ine*, and 10 to 35 for *Soybean*.

Figure 6 plots the averaged percent of finding the true number of clusters of CA-p and RECA-p vs. the parameter p. It is interesting that the curve of RECA-p in nearly all the sub-figures of Fig. 6 is almost a horizontal line, while the curve of CA-p is irregular and fluctuates greatly. In other words, the value of $p$has a great impact

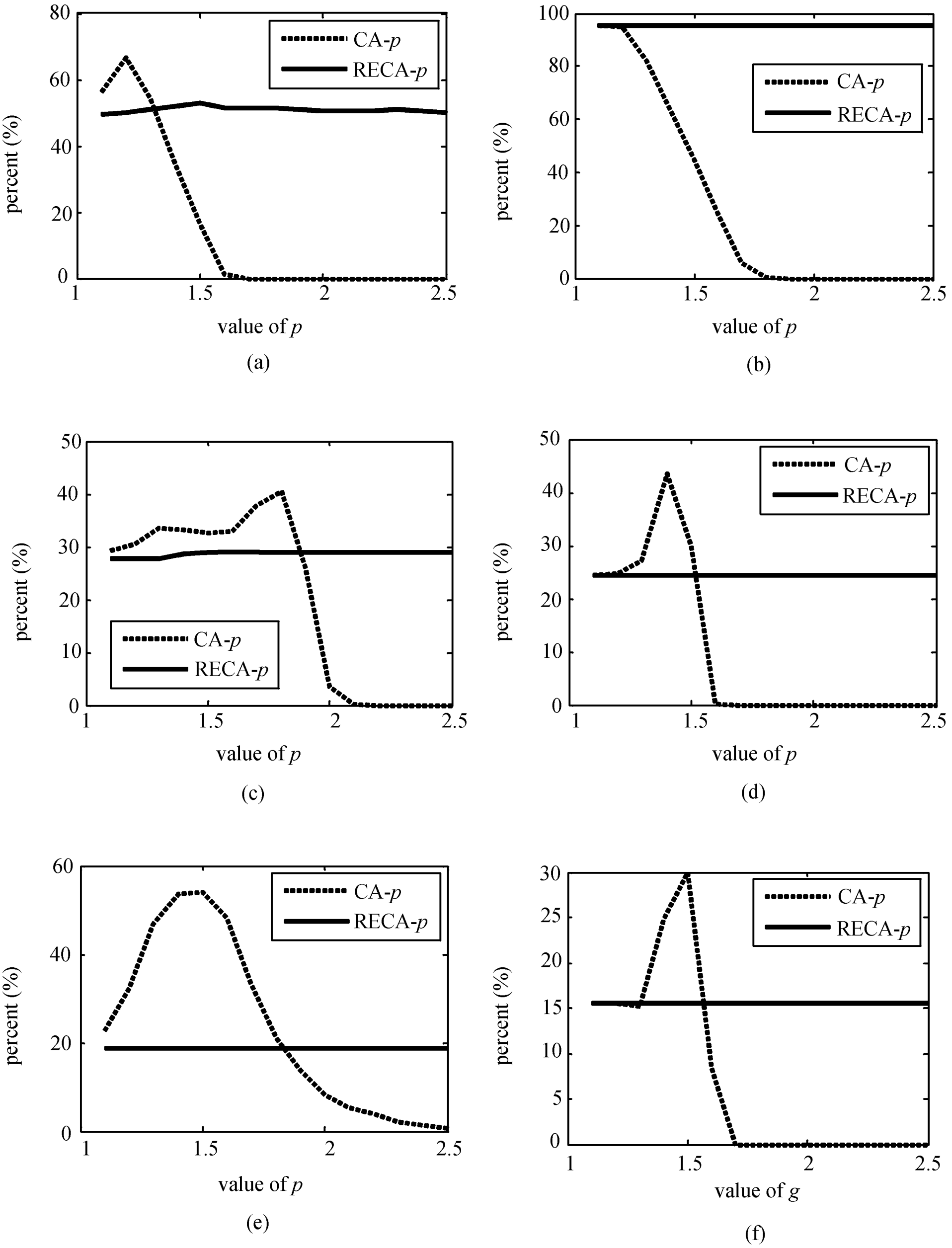


Figure 6. Averaged percent of repeated runs that came up with the true number of cluster in the actual data for a range of p values. (a) On *Bupa*. (b) On *Pid*. (c) On *Breast*. (d) On *Ionosphere*. (e) On *Wine*. (f) On *Soybean*

on the performance of CA-p, whereas it has little influence on that of RECA-p. To explain that interesting phenomenon, it is helpful to revisit the p-order entropy-like function $-\sum_{i=1}^{C} p_i^p$ in (17) and Renyi's p-order entropy-like function $-\ln\left(\sum_{i=1}^{C}(p_i+1)^p\right)$ in (20). Note that except for the extreme one-cluster case, where p_i=1 for some i and

p_j=0, for all $j \neq i$, in most cases 0< p_i <1 for all i. Then

$$-\ln\left(\sum_{i=1}^{C}(p_i+1)^p\right) \approx -\ln\left(\sum_{i=1}^{C}\left(1+pp_i+\mathrm{O}(p_i^2)\right)\right) = -\ln\left(C+p+\sum_{i=1}^{C}\mathrm{O}(p_i^2)\right) \tag{24}$$

C is usually bigger than p. According to (24), changes of the value of p cannot have dominant effect on the value of the logarithm in (24), unlike the case in the p-order entropy-like function in (17). Therefore, RECA-p is less affected by the value of p than CA-p.

Figure 6 indicates that for all data sets, using p value smaller than 2 in CA-p outperforms using p=2, which reflects that p=2 in conventional CA algorithm is not a good choice. From Fig. 6, a satisfactory value for the parameter p should be around 1.5. Thus it suggests using p=1.5 in CA-p for practical use. Table 4 shows that the averaged iteration steps of RECA-p are much less than that of CA-p and the cluster validity indices of both algorithms are very close.

Table 4 Averaged iteration steps and the cluster validity index V_{XB} of CA-p and RECA-p

Data sets	Iteration steps		Cluster validation (V_{XB})	
	CA-p	RECA-p	CA-p	RECA-p
Bupa	52	38	0.12	0.12
Pid	51	42	0.12	0.12
Breast	41	24	0.11	0.11
Ionosphere	46	34	0.71	0.71
Wine	52	25	0.12	0.13
Soybean	39	33	1.79	1.79

6 Conclusion and Future Work

In this paper, the competitive clustering (CA) algorithm is generalized based on entropy-like functions. Four generalized competitive clustering algorithms called RECA, SECA, CA-p and RECA-p are proposed, which are based on Renyi's second-order entropy-like function, Shannon entropy-like function, p-order entropy-like function and Renyi's p-order entropy-like function, respectively. Simulation results show that in order to obtain the true number of clusters with the highest possibility, CA requires a large number of initial clusters, while RECA and SECA require small and moderate number of initial clusters respectively. Thus, the iteration steps in RECA and SECA are less than that of CA. Furthermore, experimental results on both CA-p and RECA-p show that the value of $p$has a great impact on the performance of CA-p, but little influence on that of RECA-p.

In the objective functions of CA, RECA, SECA, CA-p and RECA-p, only the Euclidean distance measure is considered in this paper. However, there exist two disadvantages of Euclidean distance when it is used in the objective functions of FCM-like algorithms. First, the algorithm adopting Euclidean distance can only detect spherically-distributed clusters. To detect complicated clusters, other distance measures should be used, as[8] did. Second, Euclidean distance is not very robust[12]. In [9], weight functions were used to robustify the Euclidean distance.

Since kernel-induced distance measures have been proven effective in avoiding the above two problems[5,24−25], it can be expected that the use of kernel-induced distance measures is also effective in CA and generalized CA algorithms, which will be investigated in future work.

The estimation of initial $C_{\max}$ value is essential not only to RECA and SECA but also to CA. In[8−9], the authors empirically set $C_{\max}$ to $N/(n)$ for CA, where N is the total number of data points and n is a flexible parameter which is usually chosen a little larger than 10 for large-size data sets and less than 10 for moderate-size data sets. Accordingly, we may empirically set $C_{\max}$ to $N/(2n)$ and $N/(4n)$ for SECA and RECA respectively based on our observations in experiments.

Furthermore, instead of estimating the exact $C_{\max}$ values, in this paper we find a region for it based on cluster validation index. This method requires clustering on the whole data set for a range of $C_{\max}$ values, and hence is very time-consuming on large data sets. To deal with that problem, a feasible technique is to first perform a subsampling on the whole data sets and then estimate the region of $C_{\max}$ on the reduced or sub-sampled data set. Another direction for future research is to investigate efficient methods for integrating RECA, SECA and CA. In this paper, we combine results of the three methods simply based on cluster validation index. Can we integrate them more directly? For example, we are currently investigating integrating RECA, SECA and CA into a unified objective function and directly optimize them. Both these issues will be further investigated in the future.

Acknowledgements

The authors are grateful to the anonymous reviewers for their comments and suggestions which greatly improve the presentation of this paper. This work was supported by the National Science Foundation of China under Grant Nos. 60496320, 60505004, 60721002, the Jiangsu Science Foundation under Grant No BK2006521, and the Foundation for the Author of National Excellent Doctoral Dissertation of China under Grant No. 200343.

Appendix A: Proof of Theorem 1.

Proof: Since $0 \leqslant p_i \leqslant 1$, $1 \leqslant p_i + 1 \leqslant 2$ can be derived, then

$$\begin{aligned}\left(\sum_{i=1}^{C}(p_i+1)^2\right)\ln\left(\sum_{i=1}^{C}(p_i+1)^2\right) &= \sum_{i=1}^{C}\left[(p_i+1)^2\ln\left(\sum_{i=1}^{C}(p_i+1)^2\right)\right] \\ &\geqslant \sum_{i=1}^{C}(2p_i)\ln(1+p_i)\end{aligned} \tag{25}$$

Let $f(p_i) = 2\ln(1+p_i) - p_i$, then its derivative is

$$f'(p_i) = \frac{2}{1+p_i} - 1 \geqslant 0 \tag{26}$$

Thus, $f(p_i)$ is a monotonously increasing function. Since $0 \leqslant p_i \leqslant 1$, $f(0) = 0$, and $f(1) = 2\ln 2 > 0$, having $f(p_i) \geqslant 0$ for $0 \leqslant p_i \leqslant 1$. So

$$2\sum_{i=1}^{C} p_i \ln(1+p_i) \geqslant \sum_{i=1}^{C} (p_i)^2 \tag{27}$$

As $\eta_0 e^{-l/\tau} \geqslant 0$, according to (A.1) and (A.2), the inequality (A.1) is proved.

References

[1] Bezdek J. Pattern recognition with fuzzy objective function algorithms. New York: Plenum Press, 1981.

[2] Blake C, Keogh E, Merz CJ. UCI repository of machine learning databases, [http://www.ics.uci.edu/~mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA, 1998.

[3] Boujemaa N. Generalized competitive clustering for image segmentation. Proceedings of 19th International Meeting of the North American Fuzzy Information Processing Society, Atlanta, 2000, pp.13–15.

[4] Brand M. Pattern discovery via entropy minimization. In: Heckerman D, Whittaker C, eds. Artificial Intelligence and Statistics, 7, Morgan Kaufmann, 1999.

[5] Chen SC, Zhang DQ. Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics, 2004, 34(4): 1907–1916.

[6] Chiu SL. Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems, 1994, (2): 267–278.

[7] Dhillon I, Mallela S, Kumar R. A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research, 2003, (3): 1265–1287.

[8] Frigui H, Krishnapuram R. Clustering by competitive agglomeration. Pattern Recognition, 1997, 30(7): 1223–1232.

[9] Frigui H, Krishnapuram R. A robust competitive clustering algorithm with applications in computer vision. IEEE Trans. on Pattern Anal. Machine Intell., 1999, 21(5): 450–465.

[10] Guedalia ID, London M, Werman M. An on-line agglomerative clustering method for nonstationary data. Neural Computation, 1999, 11: 521–540.

[11] Hofmann T, Buhmann J. Pairwise data clustering by deterministic annealing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19: 1–14.

[12] Huber PJ. Robust Statistics. New York: Wiley, 1981.

[13] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Computing Surveys, 1999, 31(3): 264–323.

[14] Jenssen R, Hild KE, Erdogmus D, Principe JC, Eltoft T. Clustering using Renyi's entropy. In: Proceedings of the International Joint Conference on Neural Networks, 2003, pp.523–528.

[15] Jenssen R, Eltoft T, Principe JC. Information theoretic spectral clustering. Proceedings of Int'l Joint Conference on Neural Networks (IJCNN2004), Budapest, Hungary, 2004, pp.111–116.

[16] Medasani S, Krishnapuram R. Image categorization for efficient retrieval using robust mixture decomposition. Computer Vision and Image Understanding, 2001, 83: 216–235.

[17] Medasani S, Krishnapuram R. Robust gaussian mixture decomposition by maximizing trimmed log-likelihood and entropy. Proceedings of the North American Fuzzy Information Processing Society Conference (NAFIPS-98), Florida, 1998.

[18] Renyi A. On measures of entropy and information. Proceedings of the Fourth Berkeley Symposium on Math. Statist. Prob., 1960, University of California Press, Berkeley, 1961, pp. 547–561.

[19] Roberts S, Everson R, Rezek I. Maximum certainty data partitioning. Pattern Recognition, 2000, 33: 833–839.

[20] Tishby N, Pereira F, Bialek W. The information bottleneck method. Proceedings of 37th Allerton Conference on Communication and Computation, 1999.

[21] Xie XL, Beni G. A validity measure for fuzzy clustering. IEEE Trans. Pattern Anal. Machine Intell., 1991, 13: 841–847.

[22] Yager RR, Filev DP. Generation of fuzzy rules by mountain clustering. Journal of Intelligent and Fuzzy Systems, 1994, 2: 209–219.

[23] Yu J, Cheng QS. The upper bound of the optimal number of clusters in fuzzy clustering. Science in China (Series F), 2001, 44(2): 119–125.

[24] Zhang DQ, Chen SC. Clustering incomplete data using kernel-based fuzzy c-means algorithm. Neural Processing Letters, 2003, 18(3): 155–162.

[25] Zhang DQ, Chen SC. A novel kernelised fuzzy c-means algorithm with application in medical image segmentation. Artificial Intelligence in Medicine, 2004, 32(1): 37–50.