Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)

From Comparing Clusterings to Combining Clusterings

Zhiwu Lu and **Yuxin Peng*** and **Jianguo Xiao**
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
{luzhiwu,pengyuxin,xjg}@icst.pku.edu.cn

Abstract

This paper presents a fast simulated annealing framework for combining multiple clusterings (i.e. clustering ensemble) based on some measures of agreement between partitions, which are originally used to compare two clusterings (the obtained clustering vs. a ground truth clustering) for the evaluation of a clustering algorithm. Though we can follow a greedy strategy to optimize these measures as objective functions of clustering ensemble, some local optima may be obtained and simultaneously the computational cost is too large. To avoid the local optima, we then consider a simulated annealing optimization scheme that operates through single label changes. Moreover, for these measures between partitions based on the relationship (joined or separated) of pairs of objects such as Rand index, we can update them incrementally for each label change, which makes sure the simulated annealing optimization scheme is computationally feasible. The simulation and real-life experiments then demonstrate that the proposed framework can achieve superior results.

Introduction

Comparing clusterings plays an important role in the evaluation of clustering algorithms. A number of criteria have been proposed to measure how close the obtained clustering is to a ground truth clustering, such as mutual information (MI) (Strehl and Ghosh 2002), Rand index (Rand 1971; Hubert and Arabie 1985), Jaccard index (Denoeud and Guénoche 2006), and Wallace index (Wallace 1983). One important application of these measures is to make objective evaluation of image segmentation algorithms (Unnikrishnan, Pantofaru, and Hebert 2007), since image segmentation can be considered as a clustering problem.

Since the major difficulty of clustering combination is just in finding a consensus partition from the ensemble of partitions, these measures for comparing clusterings can further be used as the objective functions of clustering ensemble. Here, it is only different in that the consensus partition has to be compared to multiple partitions. Such consensus functions have been developed in (Strehl and Ghosh 2002) based on MI. Though a greedy strategy can be used to maximize normalized MI via single label change, the computational cost is too large. Hence, we resort to those measures between partitions based on the relationship (joined or separated) of pairs of objects such as Rand index, Jaccard index, and Wallace index, which can be updated incrementally for each single label change. Moreover, to resolve the local convergence problem, we follow a simulated annealing optimization scheme, which is computationally feasible due to the incremental update of objective function.

We have actually proposed a fast simulated annealing framework for clustering ensemble based on measures for comparing clusterings. There are three main advantages to the proposed framework: 1) developing a series of consensus functions for clustering ensemble, not just one; 2) avoiding the local optima problem; 3) low computational complexity of our consensus functions - $O(nkr)$ for n objects, k clusters in the target partition, and r clusterings in the ensemble. Our framework is readily applicable to large data sets, as opposed to other consensus functions which are based on the co-association of objects in clusters from an ensemble with quadratic complexity $O(n^2kr)$. Moreover, unlike those algorithms that search for a consensus partition via re-labeling and subsequent voting, this framework can operate with arbitrary partitions with varying numbers of clusters, not constrained to a predetermined number of clusters in the ensemble partitions.

The rest of this paper is organized as follows. Section 2 describes relevant research on clustering combination. In section 3, we briefly introduce some measures for comparing clusterings and especially give three of them in detail. Section 4 then presents the simulated annealing framework for clustering ensemble based on the three measures. The experimental results on several data sets are presented in section 5, followed by the conclusions in section 6.

Motivation and Related Work

Approaches to combination of clusterings differ in two main respects, namely the way in which the contributing component clusterings are obtained and the method by which they are combined. One important consensus function is proposed by (Fred and Jain 2005) to summarize various clustering results in a co-association matrix. Co-association values represent the strength of association between objects by an-

*Corresponding author.

alyzing how often each pair of objects appears in the same cluster. Then the co-association matrix serves as a similarity matrix for the data items. The final clustering is formed from the co-association matrix by linking the objects whose co-association value exceeds a certain threshold. One drawback of the co-association consensus function is its quadratic computational complexity in the number of objects $O(n^2)$. Moreover, experiments in (Topchy, Jain, and Punch 2005) show co-association methods are usually unreliable with the number of clusterings $r < 50$.

Some hypergraph-based consensus functions have also been developed in (Strehl and Ghosh 2002). All the clusters in the ensemble partitions can be represented as hyperedges on a graph with n vertices. Each hyperedge describes a set of objects belonging to the same cluster. A consensus function can be formulated as a solution to k-way min-cut hypergraph partitioning problem. One hypergraph-based method is the meta-clustering algorithm (MCLA), which also uses hyperedge collapsing operations to determine soft cluster membership values for each object. Hypergraph methods seem to work best for nearly balanced clusters.

A different consensus function has been developed in (Topchy, Jain, and Punch 2003) based on information-theoretic principles. An elegant solution can be obtained from a generalized definition of MI, namely Quadratic MI (QMI), which can be effectively maximized by the k-means algorithm in the space of specially transformed cluster labels of the given ensemble. However, it is sensitive to initialization due to the local optimization scheme of k-means.

In (Dudoit and Fridlyand 2003; Fischer and Buhmann 2003), a combination of partitions by re-labeling and voting is implemented. Their works pursue direct re-labeling approaches to the correspondence problem. A re-labeling can be done optimally between two clusterings using the Hungarian algorithm. After an overall consistent re-labeling, voting can be applied to determine cluster membership for each object. However, this voting method needs a very large number of clusterings to obtain a reliable result.

A probabilistic model of consensus is offered by (Topchy, Jain, and Punch 2005) using a finite mixture of multinomial distributions in the space of cluster labels. A combined partition is found as a solution to the corresponding maximum likelihood problem using the EM algorithm. Since the EM consensus function needs to estimate too many parameters, accuracy degradation will inevitably occur with increasing number of partitions when sample size is fixed.

To summarize, existing consensus functions suffer from a number of drawbacks that include complexity, heuristic character of objective function, and uncertain statistical status of the consensus solution. This paper just aims to overcome these drawbacks through developing a fast simulated annealing framework for combining multiple clusterings based on those measures for comparing clusterings.

Measures for Comparing Clusterings

This section first presents the basic notations for comparing two clusterings, and then introduces three measures of agreement between partitions which will be used for combining multiple clusterings in the rest of the paper.

Notations and Problem Statement

Let λ^a and λ^b be two clusterings of the sample data set $X = \{x_t\}_{t=1}^n$, with k_a and k_b groups respectively. To compare these two clusterings, we have to first give a quantitative measure of agreement between them. In the case of evaluating a clustering algorithm, it means that we have to show how close the obtained clustering is to a ground truth clustering. Since these measures will further be used as objective functions of clustering ensemble, it's important that we can update them incrementally for single label change. The computation of the new objective function in this way can lead to much less computational cost. Hence, we focus on these measures which can be specified as:

$$S(\lambda^a, \lambda^b) = f(\{n_i^a\}_{i=1}^{k_a}, \{n_j^b\}_{j=1}^{k_b}, \{n_{ij}\}_{ij}), \quad (1)$$

where n_i^a is the number of objects in cluster C_i according to λ^a, n_j^b is the number of objects in cluster C_j according to λ^b, and n_{ij} denotes the number of objects that are in cluster C_i according to λ^a as well as in group C_j according to λ^b.

When an object (which is in C_j according to λ^b) moves from cluster C_i to cluster $C_{i'}$ according to λ^a, only the following updates arise for this single label change:

$$\hat{n}_i^a = n_i^a - 1,\ \hat{n}_{i'}^a = n_{i'}^a + 1, \quad (2)$$

$$\hat{n}_{ij} = n_{ij} - 1,\ \hat{n}_{i'j} = n_{i'j} + 1. \quad (3)$$

According to (1), $S(\lambda^a, \lambda^b)$ may then be updated incrementally. Though many measures for comparing clusterings can be represented as (1), we will focus on one special type of measures based on the relationship (joined or separated) of pairs of objects such as Rand index, Jaccard index, and Wallace index in the following.

The comparison of partitions for this type of measures is just based on the pairs of objects of X. Two partitions λ^a and λ^b agree on a pair of objects x_1 and x_2 if these objects are simultaneously joined or separated in them. On the other hand, there is a disagreement if x_1 and x_2 are joined in one of them and separated in the other. Let n_A be the number of pairs simultaneously joined together, n_B the number of pairs joined in λ^a and separated in λ^b, n_C the number of pairs separated in λ^a and joined in λ^b, and n_D the number of pairs simultaneously separated. According to (Hubert and Arabie 1985), we have $n_A = \sum_{i,j} \binom{n_{ij}}{2}$, $n_B = \sum_i \binom{n_i^a}{2} - n_A$, and $n_C = \sum_j \binom{n_j^b}{2} - n_A$. Moreover, we can easily obtain $n_D = \binom{n}{2} - n_A - n_B - n_C$.

Rand Index

Rand index is a popular nonparametric measure in statistics literature and works by counting pairs of objects that have compatible label relationships in the two clusterings to be compared. More formally, the Rand index (Rand 1971) can be computed as the ratio of the number of pairs of objects having the same label relationship in λ^a and λ^b as:

$$R(\lambda^a, \lambda^b) = (n_A + n_D) / \binom{n}{2}, \quad (4)$$

where $n_A + n_D = \binom{n}{2} + 2\sum_{i,j} \binom{n_{ij}}{2} - \sum_i \binom{n_i^a}{2} - \sum_j \binom{n_j^b}{2}$.

A problem with the Rand index is that the expected value of the Rand index of two random partitions does not take a constant value. The corrected Rand index proposed by (Hubert and Arabie 1985) assumes the generalized hypergeometric distribution as the model of randomness, i.e., the two partitions λ^a and λ^b are picked at random such that the number of objects in the clusters are fixed. Under this model, the corrected Rand index can be given as:

$$CR(\lambda^a, \lambda^b) = \frac{\sum\limits_{i,j} \binom{n_{ij}}{2} - h^a h^b / \binom{n}{2}}{\frac{1}{2}(h^a + h^b) - h^a h^b / \binom{n}{2}}, \tag{5}$$

where $h^a = \sum\limits_i \binom{n_i^a}{2}$ and $h^b = \sum\limits_j \binom{n_j^b}{2}$. In the following, we actually use this version of Rand index for combining multiple clusterings.

Jaccard Index

In the Rand index, the pairs simultaneously joined or separated are counted in the same way. However, partitions are often interpreted as classes of joined objects, the separations being the consequences of this clustering. We then use the Jaccard index (Denoeud and Guénoche 2006), noted J, which does not consider the n_D simultaneous separations:

$$J(\lambda^a, \lambda^b) = \frac{n_A}{\binom{n}{2} - n_D} = \frac{\sum\limits_{i,j} \binom{n_{ij}}{2}}{h^a + h^b - \sum\limits_{i,j} \binom{n_{ij}}{2}}, \tag{6}$$

where $\binom{n}{2} - n_D = n_A + n_B + n_C = h^a + h^b - n_A$.

Wallace Index

This index is very natural, and it's the number of joined pairs common to two partitions λ^a and λ^b divided by the number of possible pairs (Wallace 1983):

$$W(\lambda^a, \lambda^b) = \frac{n_A}{\sqrt{h^a h^b}} = \frac{\sum\limits_{i,j} \binom{n_{ij}}{2}}{\sqrt{h^a h^b}}. \tag{7}$$

This last quantity depends on the partition of reference and, if we do not want to favor neither λ^a nor λ^b, the geometrical average is used.

The Proposed Framework

The above measures of agreement between partitions for comparing clusterings are further used as objective functions of clustering ensemble. In this section, we first give details about the clustering ensemble problem, and then present a fast simulated annealing framework for combining multiple clusterings that operates through single label changes to optimize these measure-based objective functions.

The Clustering Ensemble Problem

Given a set of r partitions $\Lambda = \{\lambda^q | q = 1, ..., r\}$, with the q-th partition λ^q having k^q clusters, the consensus function Γ for combining multiple clusterings can be defined just as (Strehl and Ghosh 2002):

$$\Gamma : \Lambda \rightarrow \lambda, \; N^{n \times r} \rightarrow N^n, \tag{8}$$

which maps a set of clusterings to an integrated clustering. If there is no prior information about the relative importance of the individual groupings, then a reasonable goal for the consensus answer is to seek a clustering that shares the most information with the original clusterings.

More precisely, based on the measure of agreement (i.e. shared information) between partitions, we can now define a measure between a set of r partitions Λ and a single partition λ as the average shared information:

$$S(\lambda, \Lambda) = \frac{1}{r} \sum_{q=1}^{r} S(\lambda, \lambda^q). \tag{9}$$

Hence, the problem of clustering ensemble is just to find a consensus partition λ^* of the data set X that maximizes the objective function $S(\lambda, \Lambda)$ from the gathered partitions Λ:

$$\lambda^* = \arg\max_{\lambda} \frac{1}{r} \sum_{q=1}^{r} S(\lambda, \lambda^q). \tag{10}$$

The desired number of clusters k^* in the consensus clustering λ^* deserves a separate discussion that is beyond the scope of this paper. Here, we simply assume that the target number of clusters is predetermined for the consensus clustering. More details about this model selection problem can be found in (Figueiredo and Jain 2002).

To update the objective function of clustering ensemble incrementally, we have to consider those measures which take the form of (1). Though many measures for comparing clusterings can be represented as (1), we will focus on one special type of measures based on the relationship (joined or separated) of pairs of objects in the following. Actually, only three measures, i.e. the Rand index, Jaccard index, and Wallace index, are used as the objection functions of clustering ensemble. Moreover, to resolve the local convergence problem of the greedy optimization strategy, we further take into account the simulated annealing scheme.

Note that our clustering ensemble algorithms developed in the following can be modified slightly when other types of measures specified as (1) are used as objective functions. Hence, we have actually presented a simulated annealing framework for combining multiple clusterings.

Clustering Ensemble via Simulated Annealing

Given a set of r partitions $\Lambda = \{\lambda^q | q = 1, ..., r\}$, the objective function of clustering ensemble can just be set as the measure between a single partition λ and Λ in (9). The measure $S(\lambda, \lambda^q)$ between λ and λ^q can be Rand index, Jaccard index, or Wallace index. According to (5)–(7), we can set $S(\lambda, \lambda^q)$ as any of the following three measures:

$$S(\lambda, \lambda^q) = \frac{h_0^q - h_1 h_2^q / \binom{n}{2}}{\frac{1}{2}(h_1 + h_2^q) - h_1 h_2^q / \binom{n}{2}}, \tag{11}$$

$$S(\lambda, \lambda^q) = h_0^q / (h_1 + h_2^q - h_0^q), \tag{12}$$

$$S(\lambda, \lambda^q) = h_0^q / \sqrt{h_1 h_2^q}, \tag{13}$$

where $h_0^q = \sum\limits_{i,j} \binom{n_{ij}^q}{2}$, $h_1 = \sum\limits_i \binom{n_i}{2}$, and $h_2^q = \sum\limits_j \binom{n_j^q}{2}$. Here, the frequency counts are denoted a little differently from (1): n_i is the number of objects in cluster C_i according to λ, n_j^q is the number of objects in cluster C_j according to λ^q, and n_{ij}^q is the number of objects that are in cluster C_i according to λ and in cluster C_j according to λ^q. Note that the corresponding algorithms based on these three measures which follow the simulated annealing optimization scheme are denoted as SA-RI, SA-JI, and SA-WI, respectively.

To find the consensus partition from the multiple clusterings Λ, we can maximize the objective function $S(\lambda, \Lambda)$ by single label change. That is, we randomly select an object x_t from the data set $X = \{x_t\}_{t=1}^n$, and then change the label of it $\lambda(x_t) = i$ to another randomly selected label $i' \neq i$ according to λ, i.e., move it from the current cluster C_i to another cluster $C_{i'}$. Such single label change only leads to the following updates:

$$\hat{n}_i = n_i - 1,\ \hat{n}_{i'} = n_{i'} + 1, \quad (14)$$
$$\hat{n}_{ij}^q = n_{ij}^q - 1,\ \hat{n}_{i'j}^q = n_{i'j}^q + 1, \quad (15)$$

where $j = \lambda^q(x_t)$ $(q = 1, ..., r)$. For each $\lambda^q \in \Lambda$, to update $S(\lambda, \lambda^q)$, we can first calculate h_1 and h_0^q incrementally:

$$\hat{h}_1 = h_1 + n_{i'} - n_i + 1, \quad (16)$$
$$\hat{h}_0^q = h_0^q + n_{i'j}^q - n_{ij}^q + 1. \quad (17)$$

Note that h_2^q keeps fixed for each label change. Hence, we can obtain the new $\hat{S}(\lambda, \lambda^q)$ according to (11)–(13), and the new objective function $\hat{S}(\lambda, \Lambda)$ is just the mean of $\{\hat{S}(\lambda, \lambda^q)\}_{q=1}^r$. Here, it is worth pointing out that the update of the objective function has only linear time complexity $O(r)$ for single label change, which makes sure that the simulated annealing scheme is computationally feasible for the maximum of $S(\lambda, \Lambda)$.

We further take into account a simplified simulated annealing scheme to determine whether to select the single label change $\lambda(x_t) : i \to i'$. At a temperature T, the probability of selecting the single label change $\lambda(x_t) : i \to i'$ can be calculated as follows:

$$P(\lambda(x_t) : i \to i') = \begin{cases} 1 & \text{if } \Delta S > 0 \\ e^{\frac{\Delta S}{T}} & \text{otherwise} \end{cases}, \quad (18)$$

where $\Delta S = \hat{S}(\lambda, \Lambda) - S(\lambda, \Lambda)$. We actually select the single label change if $P(\lambda(x_t) : i \to i')$ is higher than a threshold P_0 $(0 < P_0 < 1)$; otherwise, we will discard it and begin to try the next single label change.

The complete description of our simulated annealing framework for clustering ensemble is finally summarized in Table 1. The time complexity is $O(nk^*r)$.

Table 1: Clustering Ensemble via Simulated Annealing

Input:

1. A set of r partitions $\Lambda = \{\lambda^q | q = 1, ..., r\}$
2. The desired number of clusters k^*
3. The threshold for selecting label change P_0
4. The cooling ratio c $(0 < c < 1)$

Output:

The consensus clustering λ^*

Process:

1. Select a candidate clustering λ by some combination methods, and set the temperature $T = T_0$.
2. Start a loop with all objects set unvisited ($v(t) = 0, t = 1, ..., n$). Randomly select an unvisited object x_t from X, and change the label $\lambda(x_t)$ to the other $k^* - 1$ labels. If a label change is selected according to (18), we immediately set $v(t) = 1$ and try a new unvisited object. If there is no label change for x_t, we also set $v(t) = 1$ and go to a new object. The loop is stopped until all objects are visited.
3. Set $T = c \cdot T$, and go to step 2. If there is no label change during two successive loops, stop the algorithm and output $\lambda^* = \lambda$.

Experimental Results

The experiments are conducted with artificial and real-life data sets, where true natural clusters are known, to validate both accuracy and robustness of consensus via our simulated annealing framework. We also explore the data sets using seven different consensus functions.

Data Sets

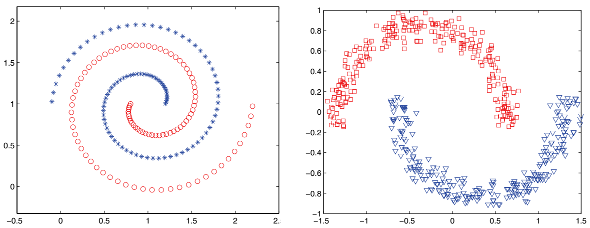
The details of the four data sets used in the experiments are summarized in Table 2. Two artificial data sets, 2-spirals and half-rings, are shown in Figure 1, which are difficult for any centroid based clustering algorithms. We also use two real-life data sets, iris and wine data, from UCI benchmark repository. Since the last feature of wine data is far larger than the others, we first regularize them into an interval of $[0, 10]$. Note that the other three data sets keep unchanged.

Table 2: Details of the four data sets. The average clustering error is obtained by the k-means algorithm.

Data sets	#features	k^*	n	Avg. error (%)
2-spirals	2	2	190	41.5
half-rings	2	2	500	26.4
iris	4	3	150	21.7
wine	13	3	178	8.4

The average clustering errors by the k-means algorithm for 20 independent runs on the four data sets are listed in Table 2, which are considered as baselines for those consensus functions. As for the regularization of wine data, the average error by the k-means algorithm can be decreased from 36.3% to 8.4% for 20 independent runs.

Here, we evaluate the performance of a clustering algorithm by matching the detected and the known partitions of the data sets just as (Topchy, Jain, and Punch 2005). The best possible matching of clusters provides a measure of perfor-



(a) (b)

Figure 1: Two artificial data sets difficult for any centroid based clustering algorithms: (a) 2-spirals; (b) half-rings.

mance expressed as the misassignment rate. To determine the clustering error, one needs to solve the correspondence problem between the labels of known and derived clusters. The optimal correspondence can be obtained using the Hungarian method for minimal weight bipartite matching problem with $O(k^3)$ complexity for k clusters.

Selection of Parameters and Algorithms

To implement our simulated annealing framework for clustering ensemble, we have to select two important parameters, i.e., the threshold P_0 for selecting label change and the cooling ratio c $(0 < c < 1)$. When the cooling ratio c takes a larger value, we may obtain a better solution but the algorithm may converge slower. Meanwhile, when the threshold P_0 is larger, the algorithm may converge faster but the local optima may be avoided at a lower probability. To achieve a tradeoff between the clustering accuracy and speed, we simply set $P_0 = 0.85$ and $c = 0.99$ in all the experiments. Moreover, the temperature T is initialized by $T = 0.1S_0$ where S_0 is the initial value of objective function.

Our three simulated annealing methods (i.e. SA-RI, SA-JI, and SA-WI) for clustering combination are also compared to four other consensus functions:

1. k-modes algorithm for consensus clustering in this paper, which is originally developed to make categorical clustering (Huang 1998).
2. EM algorithm for consensus clustering via the mixture model (Topchy, Jain, and Punch 2005).
3. QMI approach described in (Topchy, Jain, and Punch 2003), which is actually implemented by the k-means algorithm in the space of specially transformed cluster labels of the given ensemble.
4. MCLA[1] which is a hypergraph method introduced in (Strehl and Ghosh 2002).

Note that our methods are initialized by k-modes just because this algorithm runs very fast, and other consensus functions can be used as initializations similarly. Since the co-association methods have $O(n^2)$ complexity and may lead to severe computational limitations, our methods are not compared to these algorithms. The performance of the co-association methods has been already analyzed in (Topchy, Jain, and Punch 2003).

[1]The code is available at http://www.strehl.com

Table 3: Average error rate (%) on the 2-spirals data set. The k-means algorithm randomly selects $k \in [4, 7]$ to generate r partitions for different combination methods.

r	SA-RI	SA-JI	SA-WI	k-modes	EM	QMI	MCLA
10	**37.5**	39.6	38.7	45.2	45.2	46.8	39.3
20	**35.9**	37.8	37.3	43.8	44.4	47.8	37.6
30	**36.0**	37.0	39.3	41.2	43.6	47.3	40.1
40	**37.6**	39.7	**37.6**	40.8	42.2	46.9	38.4
50	36.2	39.1	**36.1**	42.8	43.9	44.4	36.4

Table 4: Average error rate (%) on the half-rings data set. The k-means algorithm randomly selects $k \in [3, 5]$ to generate r partitions for different combination methods.

r	SA-RI	SA-JI	SA-WI	k-modes	EM	QMI	MCLA
10	20.4	21.4	**20.3**	26.9	26.4	25.7	24.6
20	**18.5**	22.5	23.5	27.7	24.4	25.3	19.9
30	**18.2**	20.4	19.0	25.1	26.9	24.6	24.9
40	**17.6**	17.7	19.1	28.5	27.5	25.9	23.5
50	**18.3**	19.4	20.0	29.3	28.5	26.6	21.7

The k-means algorithm is used as a method of generating the partitions for the combination. Diversity of the partitions is ensured by: (1) initializing the algorithm randomly; (2) selecting the number of clusters k randomly. In the experiments, we actually give k a random value around the number of true natural clusters k^* $(k \geq k^*)$. We have found that this method of generating partitions leads to better results than that only by random initialization. Moreover, we vary the number of combined clusterings r in the range $[10, 50]$.

Comparison with Other Consensus Functions

Only main results for each of the four data sets are presented in Tables 3–6 due to space limitations. Actually, we have initialized our simulated annealing methods by other consensus functions besides k-modes, and some similar results can be obtained. Here, the tables report the average error rate (%) of clustering combination from 20 independent runs.

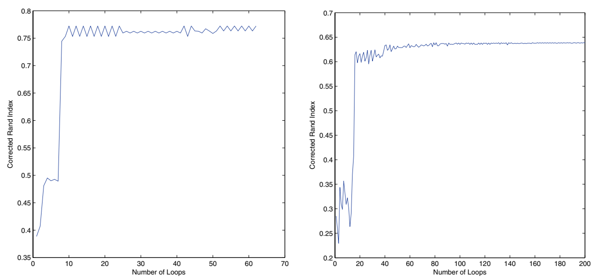
First observation is that our simulated annealing methods (especially SA-RI) perform generally better than other consensus functions. Since our methods only lead to slightly higher clustering errors in a few cases as compared with MCLA, we can think our methods preferred by overall eval-

Table 5: Average error rate (%) on the iris data set. The k-means algorithm randomly selects $k \in [3, 5]$ to generate r partitions for different combination methods.

r	SA-RI	SA-JI	SA-WI	k-modes	EM	QMI	MCLA
10	10.7	10.7	10.6	23.4	12.3	14.3	**10.4**
20	**10.6**	10.9	10.8	22.9	17.5	14.8	**10.6**
30	10.7	10.7	10.9	23.2	18.1	12.3	**10.5**
40	**10.7**	11.8	**10.7**	22.6	16.6	13.9	**10.7**
50	**10.7**	**10.7**	**10.7**	19.9	26.9	12.6	**10.7**

Table 6: Average error rate (%) on the wine data set. The k-means algorithm randomly selects $k \in [4, 6]$ to generate r partitions for different combination methods.

r	SA-RI	SA-JI	SA-WI	k-modes	EM	QMI	MCLA
10	**6.5**	6.7	**6.5**	12.3	17.1	8.8	7.6
20	6.5	6.5	**6.3**	11.4	17.9	10.4	8.5
30	6.4	**6.3**	**6.3**	12.4	13.1	7.5	7.4
40	6.3	6.3	**6.2**	10.2	17.2	7.4	7.5
50	6.3	**6.2**	**6.2**	8.1	21.1	7.3	7.8



(a) (b)

Figure 2: The ascent of corrected Rand index on two real-life data sets (only SA-RI considered): (a) iris; (b) wine.

uation. Among our three methods, SA-RI performs the best generally. All co-association methods are usually unreliable with $r < 50$ and this is where our methods are positioned. The k-modes, EM, and QMI consensus functions all have the local convergence problem. Since our methods are just initialized by k-modes, we can find that local optima are successfully avoided due to the simulated annealing optimization scheme. Figure 2 further shows the ascent of corrected Rand index on two real-life data sets (only SA-RI with $r = 30$ considered) during optimization.

Moreover, it is also interesting to note that, as expected, the average error of consensus clustering by our simulated annealing methods is lower than average error of the k-means clusterings in the ensemble (Table 2) when k is chosen to be equal to the true number of clusters k^*.

Finally, the average time taken by our three methods (Matlab code) is less than 30 seconds per run on a 1 GHz PC in all cases. As reported in (Strehl and Ghosh 2002), experiments with $n = 400$, $k = 10$, $r = 8$ average one hour using the greedy algorithm based on normalized MI (similar to our methods). However, our methods only take about 10 seconds in this case, i.e., our methods are computationally feasible in spite of the costly annealing procedure.

Conclusions

We have proposed a fast simulated annealing framework for combining multiple clusterings based on some measures for comparing clusterings. When the objective functions of clustering ensemble are specified as those measures based on the relationship of pairs of objects in the data set, we can then update them incrementally for each single label change, which makes sure that the proposed simulated annealing optimization scheme is computationally feasible. The simulation and real-life experiments then demonstrate that the proposed framework can achieve superior results. Since clustering ensemble is actually equivalent to categorical clustering, our methods will further be evaluated in this application in the future work.

Acknowledgements

This work was fully supported by the National Natural Science Foundation of China under Grant No. 60503062, the Beijing Natural Science Foundation of China under Grant No. 4082015, and the Program for New Century Excellent Talents in University under Grant No. NCET-06-0009.

References

Denoeud, L., and Guénoche, A. 2006. Comparison of distance indices between partitions. In *Proceedings of the IFCS'2006: Data Science and Classification*, 21–28.

Dudoit, S., and Fridlyand, J. 2003. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* 19(9):1090–1099.

Figueiredo, M. A. T., and Jain, A. K. 2002. Unsupervised learning of finite mixture models. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 24(3):381–396.

Fischer, R. B., and Buhmann, J. M. 2003. Path-based clustering for grouping of smooth curves and texture segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(4):513–518.

Fred, A. L. N., and Jain, A. K. 2005. Combining multiple clusterings using evidence accumulation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27(6):835–850.

Huang, Z. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery* 2:283–304.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of Classification* 2:193–218.

Rand, W. M. 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* 66:846–850.

Strehl, A., and Ghosh, J. 2002. Cluster ensembles - a knowledge reuse framework for combining partitionings. In *Proceedings of Conference on Artificial Intelligence (AAAI)*, 93–99.

Topchy, A.; Jain, A. K.; and Punch, W. 2003. Combining multiple weak clusterings. In *Proceedings of IEEE International Conference on Data Mining*, 331–338.

Topchy, A.; Jain, A. K.; and Punch, W. 2005. Clustering ensembles: models of consensus and weak partitions. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 27(12):1866–1881.

Unnikrishnan, R.; Pantofaru, C.; and Hebert, M. 2007. Toward objective evaluation of image segmentation algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 29(6):929–944.

Wallace, D. L. 1983. Comment on a method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* 78:569–576.