Pattern Recognition 42 (2009) 668 – 675

Contents lists available at ScienceDirect

Pattern Recognition

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

Clustering aggregation by probability accumulation

Xi Wang, Chunyu Yang, Jie Zhou*

Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 11 May 2007
Received in revised form 15 September 2008
Accepted 21 September 2008

Keywords:
Clustering aggregation
Evidence accumulation
Probability accumulation

ABSTRACT

Since a large number of clustering algorithms exist, aggregating different clustered partitions into a single consolidated one to obtain better results has become an important problem. In Fred and Jain's *evidence accumulation* algorithm, they construct a co-association matrix on original partition labels, and then apply minimum spanning tree to this matrix for the combined clustering. In this paper, we will propose a novel clustering aggregation scheme, *probability accumulation*. In this algorithm, the construction of correlation matrices takes the cluster sizes of original clusterings into consideration. An alternate improved algorithm with additional pre- and post-processing is also proposed. Experimental results on both synthetic and real data-sets show that the proposed algorithms perform better than *evidence accumulation*, as well as some other methods.

1. Introduction

As an important approach of unsupervised learning, clustering has been playing a more and more indispensable role in knowledge discovery, with application to numerous fields. These include data mining [1], information retrieval [2,3], image segmentation [4], machine learning, and so on. Informally, clustering can be defined as the problem of partitioning samples into different clusters, so that the samples in the same cluster are more similar than those in different clusters.

A large number of clustering algorithms exist [5–7], yet no single one can handle all types of cluster shapes and structures. Additionally, there are still some problems with conventional algorithms: on one hand, results depend much on the parameters and initializations, such as the times of iterations in K-means and the threshold in hierarchical clustering; on the other hand, most algorithms cannot automatically determine how many clusters the data-set should be partitioned into. For these reasons, partitions given by different algorithms might be greatly different. Determining which one of them should be trusted becomes a difficult problem.

Inspired by sensor fusion and classifier combination [8], and for the sake of improving the robustness and quality of clustering solutions, recent research is increasingly focusing on combining multiple partitions into a single one, i.e., clustering aggregation. This process can be simply described as given a data-set with multiple partitions, the manipulation will combine the inputs to get a final partition.

Some work has been done in this new field: Fred and Jain introduced the so-called *evidence accumulation* (EA), which is based on a co-association matrix [9–11]. Strehl and Ghosh's *hypergraph partitioning* proposed a new hypergraph cuts approach [12], and Fern and Brodley developed this algorithm further, designing their *hybrid bipartite graph formulation* (HBGF) [13]. *Mutual information algorithm* (QMI) [14] and *voting approach* [15] have also been put forward recently to solve the problem of clustering aggregation. Among these algorithms, EA has proven to be most effective [9].

In EA, each given partition is mapped into a $n \times n$ 0–1 symmetric matrix, where n is the number of samples in the data-set. These matrices can be regarded as component matrices. In these matrices, the value 1 denotes that the corresponding data pair are partitioned into the same cluster, while 0 denotes that they are divided into different clusters. The mean of all the component matrices is defined as the co-association matrix. Then, minimum spanning tree (MST) [16] with single link (SL) or average link (AL) [5,6] is applied to the co-association matrix to obtain the combined final partition. Moreover, Fred and Jain proposed a *highest lifetime* criterion to determine the number of clusters. As a result, this algorithm can determine a sole combined clustering out of the original partitions labels.

The co-association matrix provides the pairwise correlations by simple counting. This kind of matrix refers only to the partition labels, and it identically define the correlation of every pair from the same cluster as 1. Thus it does not take clusters' other characteristics into account, such as their sizes.

In this paper, we propose a novel clustering aggregation algorithm: *probability accumulation* (PA). For each given partition, PA takes the size of each cluster, as well as the dimension of samples into consideration, so that elements in the component matrix are

* Corresponding author. Tel.: +86 10 6278 2447; fax: +86 10 6278 6911.
E-mail address: jzhou@tsinghua.edu.cn (J. Zhou).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.09.013

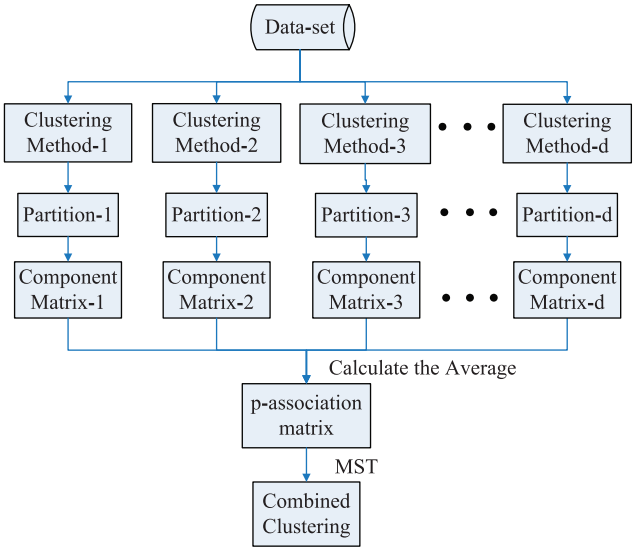

Fig. 1. The overall procedure of clustering aggregation.

continuous values rather than binary values. As a result, the proposed algorithm can measure the pairwise correlations in higher resolution. The overall procedure of our algorithm is shown in Fig. 1, in which we name the mean of all new component matrices as p-association matrix instead of co-association matrix in EA. Both theoretical analysis and experimental results illustrate that our algorithm outperforms (EA).

The rest of this paper is structured as follows: In Section 2 we will introduce the proposed PA. Starting from an ideal formula, the algorithm is simplified on two different levels to deduce its practical form. Moreover, we also propose a pre/post-processing step to deal with data-sets that do not satisfy the assumption. In Section 3, experiments on synthetic and real data-sets are presented and analyzed. Finally, in Section 4, a conclusion is drawn.

2. Probability accumulation

We begin our discussion about PA by introducing our notations: let $X=\{x_1, x_2, \ldots, x_n\}$ be a data-set, with $|X|=n$, and x_i is an $m\times 1$ vector. $C^{(p)}=\{c^{(1)}(x_1), c^{(2)}(x_2), \ldots, c^{(p)}(x_n)\}$ denotes the partition label from the $\boldsymbol{p}$-th clustering method, where $p=1, 2, \ldots, d$. There, $c^{(p)}(x_i)$ means that in the $\boldsymbol{p}$-th clustering x_i is partitioned into the $c^{(p)}(x_i)$-th cluster. $X_k^{(p)}$ denotes samples which are partitioned into the $\boldsymbol{k}$-th cluster by $C^{(p)}$. In addition, $A^{(p)}=(a_{ij})_{n\times n}$ represents the p-component matrix corresponding to partition $C^{(p)}$, where a_{ij} denotes the correlation of the couple (x_i, x_j), which will be explained in details in the rest of this section.

The process of clustering aggregation can be described as follows: given a set of partitions $\mathbb{C}=\{C^{(1)}(X), C^{(2)}(X), \ldots, C^{(d)}(X)\}$, the combined partition of the data-set, $P=\{p_1, p_2, \ldots, p_n\}$, is determined and output.

In this section, the deduction of PA is structured as follows: (1) We firstly assume the pairwise distances are available, and give out a general framework of PA which takes all pairwise distances of the data-set into consideration. (2) Then we relax such strict assumption, and use the average pairwise distance of a cluster to represent the distance of all pairs belong to that cluster. Here, we introduce the cluster conditional probability densities to calculate this average distance. (3) Furthermore, when only partition labels are available, we make the uniform distribution assumption on basis of maximum entropy theory and the spherical distribution assumption regarding the characters of K-means. Then we get the practical form of PA algorithm. Aside from the theoretic deduction, the pre/post-processing is introduced at the end of this section, which could help to deal with the data-sets which do not satisfy the assumptions.

2.1. A new measurement of correlation

We mark the sample feature space as Ω. Since all the samples have been mapped into Ω, we can calculate any pair's distance, and use $D(x_i, x_j)$ to denote the distance between x_i and x_j.

The basic idea of our construction of p-component and p-association matrix is that the larger two samples' distance is, the weaker their correlation usually becomes. As a result, we define the correlation of any couple (x_i, x_j) as

$$A(x_i, x_j)=\frac{\lambda}{\lambda+D(x_i, x_j)}, \tag{1}$$

where λ is a coefficient to be determined. This kind of correlation lies in the range $[0, 1]$. On one extreme, when two samples are too far away from each other, we set their distance as infinite, and then their correlation becomes extremely weak, essentially 0. On the other extreme, the correlation between any individual sample and itself is the strongest, where $D(x_i, x_i)$ becomes 0, and $A(x_i, x_i)$ becomes 1.

This kind of p-component matrix has given out a general framework for the description of pairwise correlations. When the details of samples or the distance matrix are available, we can apply this definition directly. If not, according to different prior information, we can simplify the formula for the p-component matrix.

2.2. Simplified form on probability densities

Since neither samples' details nor the distance matrix are usually available, we have to relax our algorithm's requirement. In this part, our work is based on the hypothesis that in each original partition, the cluster conditional probability densities of samples in all clusters can be obtained.

Firstly, we consider the case that two samples are partitioned into the same cluster k by partition $C^{(p)}$, or rather $x_i, x_j \in X_k^{(p)}$. We do not know the accurate value of each $D(x_i, x_j)$. However, the average distance of all the pairs can be used instead approximately. Given the sub-data-set's probability density function is given, this average distance can be calculated easily. And then, according to Eq. (1), we can get the corresponding average correlation of $X_k^{(p)}$. We use this value to represent the correlation between each of the two samples that both belong to $X_k^{(p)}$.

There, we let $\Omega_k^{(p)}$ be the sub-region of Ω supporting $X_k^{(p)}$. Q is set to denote a point in Ω. The cluster conditional probability density function of $X_k^{(p)}$ is denoted as $f_k^{(p)}(Q)=p(x|x\in\Omega_k^{(p)})$. Assume that Q_1 and Q_2 are two points in $\Omega_k^{(p)}$ and $D(Q_1, Q_2)$ denotes their actual distance. Additionally, let $\overline{D_k^{(p)}}$ be the average distance of all pairs in $X_k^{(p)}$. Since we have assumed that $f_k^{(p)}(Q)$ is already known, the average distance can be calculated as below:

$$\overline{D_k^{(p)}}=\iint_{Q_1, Q_2\in\Omega_k^p} f_k^{(p)}(Q_1) f_k^{(p)}(Q_2) D(Q_1, Q_2)\,\mathrm{d}Q_1\,\mathrm{d}Q_2. \tag{2}$$

We replace $D(x_i, x_j)$ by $\overline{D_k^{(p)}}$ in Eq. (1), which can then be rewritten as

$$A^{(p)}(x_i, x_j)=\frac{\lambda}{\lambda+\overline{D_k^{(p)}}}. \tag{3}$$

Considering that the average distance can be any continuous value, the correlation $A^{(p)}(x_i, x_j)$ would also be a continuous value.

This definition of correlation is also applicable to other two situations: (1) As to a couple belong to two different clusters, since we have no information of their real distance, we set it to be infinite. As a result, their correlation becomes 0 now. (2) As to the correlation between any sample and itself, or a cluster composed of a singular sample, the distance becomes 0 and thus the correlation becomes 1. This describes the strongest correlation in our p-component matrix. In consequence, all situations have been taken into consideration and can be unanimously measured by Eq. (3). For clearer description about our entire construction of PAs p-component matrix, we give it out here according to different conditions

$$A^{(p)}(x_i, x_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j \text{ and } c^{(p)}(x_i) \neq c^{(p)}(x_j), \\ \dfrac{\lambda}{\lambda + \overline{D_k^{(p)}}}, & i \neq j \text{ and } c^{(p)}(x_i) = c^{(p)}(x_j) = k. \end{cases} \tag{4}$$

When the probability distributions for all the clusters in partition $C^{(p)}$ are given, we can calculate their corresponding p-component matrix according to Eq. (4), and we get a simplified form of PA. All these correlations are continuous values in the area [0, 1]. Compared with the 0–1 component matrices in EA, our method can introduce more information, and therefore better performance can be achieved.

Each partition determines the sole p-component matrix for PA. Subsequently, we calculate the mean of all these matrices, similarly to EA. In EA, this average matrix is called co-association matrix. In our approach, we name it after p-association matrix. Subsequently, we apply MST to this p-association matrix with the *highest lifetime* criterion as Fred and Jain do in Ref. [9], and the combined partition can be obtained.

2.3. Simplified form on partition labels

By introducing the cluster probability density functions instead of samples' details, we have simplified our PA algorithm. Yet in most situations, only the partition labels are available, just like the assumption in Ref. [9]. In order to address this circumstance, we deduce a further simplified form of PA. Since we cannot get the real distributions, we have to estimate them. Maximum entropy theory helps us to draw the uniform distribution, and subsequently the more simplified and practical form of PA. Moreover, under a special circumstance, when each cluster is of the same size, our PA algorithm would degenerate to EA. This relationship is also discussed theoretically at the end of this part.

Uniform distribution means that the probability density throughout each sub-area $\Omega_k^{(p)}$ is constant. In another word, the probability of every sample x of $X_k^{(p)}$ in $\Omega_k^{(p)}$ is the same. This hypothesis is ensured by maximum entropy theory.

Maximum entropy theory aims at the problem of estimating a system's model out of insufficient limitations on the system [17]. It is based on the concept of information entropy, which is defined as

$$h(Y) = -\int p(y)\log(p(y))\,\mathrm{d}y, \tag{5}$$

where Y is a random variable and $p(y)$ denotes its probability density function. According to maximum entropy theory, the probability density function of the random variable Y we seek is the one to maximize the value of function (5).

This theory describes an approach to estimate a variable's distribution when prior knowledge is not sufficient to determine it completely. Since we hope for different clusters' distributions in the m-dimensional space while we have not any prior information rather than partition labels, we resort to maximum entropy theory. After simple calculation and deduction, we can easily optimize the solution under such circumstance: $\hat{p}(y) = \mathrm{e}^{-\lambda_0}$. As a result, we can get the conclusion: when no information about the probability density is provided, the best estimation should be the uniform distribution. In another word, the probability each sample locates at any point in the area is identical.

Under the premise we propose above, all the probability density functions have thus become constant, or rather $f_k^{(p)}(Q) \equiv 1/V_k^{(p)}$, where $V_k^{(p)}$ denotes the volume of the area $\Omega_k^{(p)}$.

After that, another factor to be determined for a cluster's average distance is the shape of $\Omega_k^{(p)}$. There we make the hypothesis that every $\Omega_k^{(p)}$ is a sphere in the space. The reason lies in the following two aspects: On one hand, when carrying out clustering aggregation, we have not any prior knowledge about each cluster's details. Under this circumstance, the sphere distribution is the most common hypothesis. On the other hand, K-means is one of the simplest and most commonly used clustering algorithm. It is oriented to minimize the squared error and therefore tend to divide samples into a series of spheres in the corresponding space [5]. In Ref. [9], Fred and Jain use K-means to generate original partitions. Thus, we also decide to use K-means to produce partition labels in this study. With respect to these two reasons, we assume each $\Omega_k^{(p)}$ to be a sphere, and set $R_k^{(p)}$ to denote its radium.

In spheres, the distance $D(Q_1, Q_2)$ is in proportion to $R_k^{(p)}$. According to Eq. (2), the average distance $\overline{D_k^{(p)}}$ is actually a weighed average of all the $D(Q_1, Q_2)$ with their corresponding weights and therefore in proportion to $R_k^{(p)}$ as well. Other factors that would affect $\overline{D_k^{(p)}}$ is the dimension m, which determines the proportion and this coefficient is marked as $\alpha(m)$. Thus, we have

$$\overline{D_k^{(p)}} = \alpha(m) R_k^{(p)}. \tag{6}$$

In addition, we introduce the concept of sample proportional density to describe the density of samples in each sub-area $\Omega_k^{(p)}$. It is defined as

$$\rho_k^{(p)} = \frac{|X_k^{(p)}|}{n V_k^{(p)}}. \tag{7}$$

There, $|X_k^{(p)}|$ denotes the number of samples that $X_k^{(p)}$ contains. And this density describes how many percents of all the samples that unit volume of each sub-area contains on average. Since all the $\Omega_k^{(p)}$ are spheres in m-dimensional space, according to sphere's volume formula, we can get the correlation between $V_k^{(p)}$ and $R_k^{(p)}$:

$$V(R, m) = \frac{\pi^{m/2} R^m}{\Gamma(1 + m/2)}, \tag{8}$$

where $\Gamma(\cdot)$ denotes the gamma function. Moreover, Eq. (7) has described the proportional correlation between $R_k^{(p)}$ and $\overline{D_k^{(p)}}$. Consequently, we can deduce the connection between $|X_k^{(p)}|$ and $\overline{D_k^{(p)}}$.

Finally, by combining Eqs. (3), (7), and (8), we can get the expression for calculating the corresponding correlation value as

$$A^{(p)}(x_i, x_j) = \frac{\lambda}{\lambda + \overline{D_k^{(p)}}} = \frac{\lambda \sqrt[m]{n\mu(m)\rho_k^{(p)}}}{\lambda \sqrt[m]{n\mu(m)\rho_k^{(p)}} + \sqrt[m]{|X_k^{(p)}|}}. \tag{9}$$

There, $\mu(m) = \pi^{m/2}/\Gamma(1 + m/2)$ denotes the coefficient completely determined by the dimension m. From this formula we can easily

see that, once the data-set is given, the coefficient n, m, and $\mu(m)$ are determined and therefore can be regarded as constant. λ is constant, too. Thus, aside from these factors above, this correlation value is only subject to the number of samples, i.e., $|X_k^p|$ and the sample proportional density $\rho_k^{(p)}$. The former shows the contents of different sub-data-sets while the latter shows their respective densities. That means, when we consider two clusters partitioned by a clustering, if they have the same density, the one containing more samples would construct a bigger sphere in the space. Thus, the average correlation of any two samples would be weaker. Similarly, if they both contained the same number of samples and constructed two spheres of different volumes, in the smaller sphere, samples' average correlation would be stronger. The result of Eq. (9) shows the same tendency.

The $|X_k^{(p)}|$ can be directly extracted from the partition label $C^{(p)}$, but $\rho_k^{(p)}$ must be given previously. When the probability densities or $\rho_k^{(p)}$ are available, we can apply Eq. (9) directly. However, when they are unknown, we have to make the hypothesis that all the clusters' uniform distributions have the same probability densities, and it means that all the $\rho_k^{(p)}$ are of the same value $\rho_0^{(p)}$. Additionally, for simplification, we set the coefficient λ by $\lambda = (n\mu(m)\rho_0^{(p)})^{-1/m}$, which makes the numerator of Eq. (9) equal 1. As a result, $|X_k^p|$ and m become the sole factors that would affect the correlation $A^{(p)}(x_i, x_j)$, and we get the simplified form of PA algorithm when only partition labels and basic knowledge about the data-set (dimension m) are available.

To sum up, we combine all the three situations and therefore get the entire formula to construct p-component matrix for PA:

$$A^{(p)}(x_i,x_j) = \begin{cases} 1, & i=j, \\ 0, & i\neq j \text{ and } c^{(p)}(x_i)\neq c^{(p)}(x_j), \\ \dfrac{1}{1+\sqrt[m]{|X_k^{(p)}|}}, & i\neq j \text{ and } c^{(p)}(x_i)=c^{(p)}(x_j)=k. \end{cases} \quad (10)$$

There, $x_i, x_j \in X, i, j = 1, 2, \ldots, n$. And the matrix $A^{(p)}(a_{ij})_{n\times n}$ is p-component matrix we calculate from the partition label $C^{(p)}$. As introduced previously, this is the most important difference between EA and PA.

One special situation we want to talk about here is when a clustering partitions whose samples are divided into clusters of the same size. Under such circumstance, all the $|X_k^p|$ become the same. When two samples are in the same cluster, their correlation is $(1+|X_k^p|)^{-1}$; otherwise, it is 0. Now, our p-association matrix has only three different values: 0, 1, and $(1+|X_k^p|)^{-1}$. Since the diagonal of the matrix would not affect the result of **MST**, our p-association matrix actually becomes a binary one. It means that, in this special situation, our PA algorithm degenerates to EA. Our experimental results in the next section also verify this relationship.

2.4. The overall procedure

After calculating all the p-component matrices from the given partition labels, we get p-association matrix by calculating their mean, as EA does to obtain the co-association matrix. Apply MST to this p-association matrix with Fred and Jain's *highest lifetime* criterion [9], we can get the combined clustering in the end. The overall procedure of PA summarized in Table 1.

This procedure has represented the basic framework of our algorithm. Both EA and PA demand for partition labels and try to deduce the combined result based only on them. Compared with the former, our algorithm needs only one more input, the data-set's dimension. This factor is easy to obtain in real applications. Moreover, when constructing the p-component matrices, what we can exploit from the labels is not only whether any two samples are in the same cluster, but also the number of samples in each cluster and the whole data-set. However, EA only makes use of the former information. Since PA has utilized more information, better performance can be expected. And the experimental results in next section have also shown its capability.

Table 1
The Procedure of Algorithm

`Probability Accumulation`
Inputs: n—number of samples
m—dimension of samples
$\{C^{(p)}\}_{p=1}^{d}$—d partition labels
1. **Calculate all the component matrices**
repeat $\boldsymbol{d}$ times: $C^{(p)} \to A^{(p)}$, $p = 1, 2, \ldots, d$

$$A^{(p)}(x_i,x_j) = \begin{cases} 1, & i=j \\ 0, & i\neq j \text{ and } c^{(p)}(x_i)\neq c^{(p)}(x_j) \\ \dfrac{1}{1+\sqrt[m]{|X_k^{(p)}|}}, & i\neq j \text{ and } c^{(p)}(x_i)=c^{(p)}(x_j)=k \end{cases}$$

2. **Calculate the p_association matrix:**

$$p\_\text{association} = \sum_{p=1}^{d} A^{(p)}$$

3. **Apply MST with** *highest lifetime* **to p_association**
Output: P—the combined clustering

2.5. Improved scheme with pre/post-processing strategy

As introduced previously, our PA algorithm is based on the hypothesis uniform distribution supported by maximum entropy theory, and the densities of all the clusters are identical, when no other information is available. As a result, this approach is more applicable to the data-sets which satisfy, or mostly satisfy this distribution. For those which do not, our algorithm may not perform well enough. To make the algorithm more effective, an improved scheme with a pre/post-processing to original data set is proposed as below:

(1) Pre-process: we define a reflection $\Gamma(X, \varepsilon) = X'$ here, which acts on the original data-set X. This reflection firstly divides the high-dimensional space into grids of the same size. Samples in the same grid would then be reflected to the same point in the grid's space. There, we use one of the grid's vertexes. The threshold ε determine the size of each grid. And all these reflections compose the new data-set X', which now satisfy the uniform distribution.
(2) Run K-means X' and apply PA to the original partition labels to get the combined clustering P', just the same as the procedure described in Table 1.
(3) Post-process: we give the partition label back to the original data-set for the real combined clustering P. It means that each sample in X get the same clustering index as its reflection in X', i.e., $P(x_i) = P'(\Gamma(x_i, \varepsilon))$.

Considering that the threshold ε depends much on the size of the original data-set, we also suggest normalizing the data-set to zero mean and unitary variance before the pre-processing step. Thus, a rational threshold would be versatile to all data-sets. The procedure for all these extra processes is summarized in Table 2.

We can find that the proposed pre-processing step serves to convert the original data-set into one, i.e., X', that satisfies PAs assumption. Thus, PA would be more applicable and effective. Since PA here is actually carried out on X' rather X, the post-processing step is designed to get the combined clustering for X on basis of that for X'. Experimental results in next section reveal that these extra processes do improve the performance of our PA algorithm, especially on the data-sets which do not satisfy our hypothesis. Considering we

rarely know data-sets' distributions beforehand, and this extra processes do not increase much computation expenditure, we suggest implementing them with PA all the time.

3. Experimental results

We have conducted extensive experiments to test the quality of PA on both synthetic and real data-sets. Parts of these data-sets are used by Fred and Jain in Ref. [9], while others are downloaded from the UCI Machine Learning Repository. In this section, we firstly introduce the data-sets we have run our experiments on. Subsequently, we test the performance of EA, Meta-Clustering Algorithm (MCLA) [12], HBGF [13], and PA on these data-sets, and compare their capability for clustering aggregation. Finally, some additional experiments are carried out: we do the pre-process and construct data-sets with constant probability density from the original ones. Clustering aggregation on these data-sets helps to show PAs capability when our hypothesis of uniform distribution exists.

3.1. Description of data-sets

Fred and Jain used nine data-sets altogether in Ref. [9] to test their EAs capability on. Some of them are synthetic 2D data-sets, while other real data-sets are mainly downloaded from UCI Machine Learning Repository. These data-sets include:

(1) *Half-rings*—two clusters (see Fig. 2a);
(2) *Three-rings*—three clusters (see Fig. 2b);
(3) *Cigar data*—four clusters (see Fig. 2c);
(4) *Iris data*—three clusters, from UCI;
(5) *Wisconsin breast-cancer*—two clusters, from UCI;
(6) *Optical-digits*—10 clusters, from UCI. Following the experiments in Ref. [9], we only use a subset composed of its first 100 samples;
(7) *Log yeast*, and
(8) *Std yeast* consist of the logarithm and the standardized version (normalization to zero mean and unitary variance), respectively, of gene expression levels of 384 genes over two cell cycles of yeast cell data.

The first three 2D data-sets are shown in Fig. 2.

Aside from these data-sets above, for further comparison, we additionally use other five data-sets, which are all widely used in the research of data mining and machine learning, and downloaded from the UCI Machine Learning Repository. These include:

(1) *Wine data*—178 samples in three clusters;
(2) *Glass data*—216 samples in six clusters;
(3) *Waveform data*—three clusters. There, we use a subset composed of its first 300 samples;
(4) *Waveform with noise*—three clusters with 40 dimensions. Actually this data-set is the original form of "waveform". Compared with the 21 attributes of *Waveform*, the 19 addition attributes are actually noises. We also carry out our experiment on its first 300 samples;

Table 2
The procedure of pre and post-processing steps

Pre/post Processing
Inputs: X—original data-set
ε—threshold for pre-process
1. **Normalizing the data-set** X **to** $\overline{X}$
satisfying: $E(\overline{X}=0)$ and $D(\overline{X})=1$
2. **Run pre-process on** $\overline{X}$
get the reflected data-set: $\Gamma(X,\varepsilon)=\overline{X}'$
3. **Run K-means on** $\overline{\mathbf{X}}$
get partition labels: $\{C^{(p)}\}_{p=1}^{d}$
4. **Apply PA to** $\overline{X}$ **and** $\{C^{(p)}\}_{p=1}^{d}$
get combined clustering: $\overline{P}$
5. **Run post-process on** $\overline{P}$
get $P=\overline{P}: P(x_i)=\overline{P}(x_i)=P'(\Gamma(x_i,\varepsilon))$
Output: P—the combined clustering

Table 3
Overview of data-sets

Data-set	Description of data-set		
	#Sample	#Attribute	#Class
Half-rings	500	2	2
Three-rings	500	2	3
Cigar	190	2	4
Iris	150	3	3
Optical-digit	100	64	10
Breast-cancer	683	9	2
Log yeast	384	76	5
Std yeast	384	76	5
Wine	178	13	3
Glass	214	9	6
Waveform	300	21	3
Wave + noise	300	40	3
Credit	652	15	2

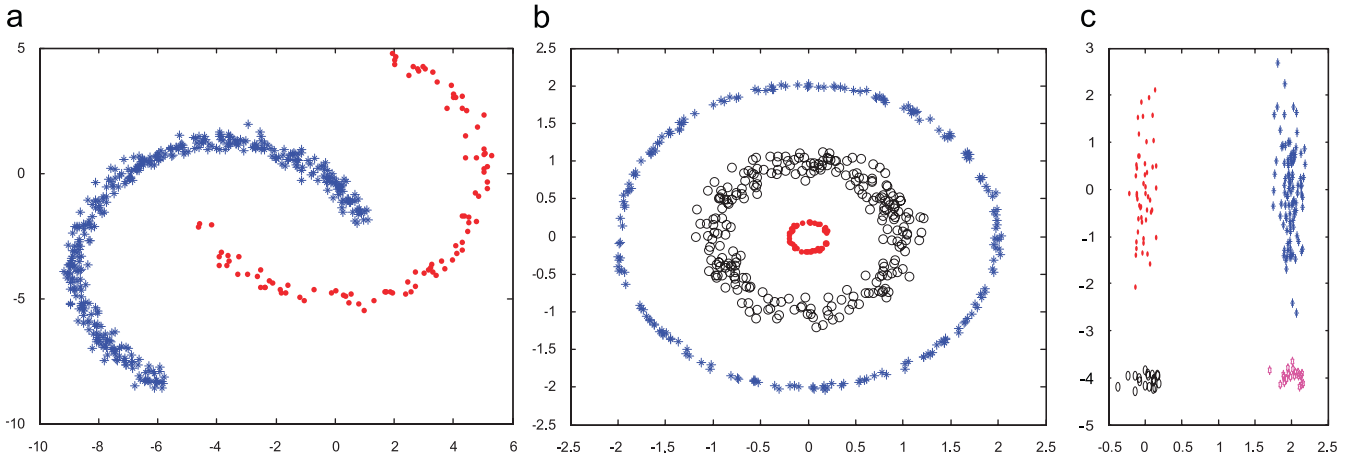

Fig. 2. Test data-sets. (a) Half-rings shape clusters. Left cluster has 100 patterns and the right cluster has 400 patterns. (b) Three-rings data-set. Number of patterns in three rings are, respectively, 50, 250, and 200. (c) Cigar data-set. The top-left cluster has 50 patterns, and the top-right cluster has 100 patterns. Those two bottom clusters have 20 patterns, respectively.

Table 4
Average error rates (in percentage) of combining 10 clusterings

Data-set	Without pre/post-process				With pre/post-process			
	EA	MCLA	HBGF	PA	P-EA	P-MCLA	P-HBGF	P-PA
Half-rings	2.88	28.10	28.21	3.07	1.64	19.09	20.89	1.25
Three-rings	30.63	45.31	47.07	22.40	28.39	44.93	45.90	14.43
Cigar	9.07	27.57	28.26	13.38	6.00	24.85	25.35	5.52
Iris	33.23	33.25	33.04	33.23	29.69	34.20	34.51	21.36
Optical-digit	68.90	64.48	64.44	69.12	68.10	58.42	58.88	1.04
Breast-cancer	15.50	36.41	43.65	15.17	2.81	53.95	59.73	2.89
Log yeast	45.44	49.13	51.01	44.34	42.56	50.34	53.30	41.58
Std yeast	54.25	49.68	52.17	50.90	48.25	50.09	51.08	43.94
Wine	47.08	36.94	39.26	40.27	34.80	31.36	34.19	29.61
Glass	56.54	55.19	55.94	53.13	53.44	55.83	58.74	48.32
Wave	35.68	46.80	46.98	34.38	30.57	52.84	52.79	28.00
Wave + noise	32.62	57.40	58.30	33.12	19.47	57.80	57.83	17.16
Credit	58.80	63.91	66.10	35.44	8.98	45.56	47.81	9.01
Average	37.74	45.71	47.26	34.46	28.82	44.56	46.23	24.93

Table 5
Minimum error rates (in percentage) of combining 10 clusterings

Data-set	Without pre/post-process				With pre/post-process			
	EA	MCLA	HBGF	PA	P-EA	P-MCLA	P-HBGF	P-PA
Half-rings	0.00	20.80	22.90	0.00	0.00	0.00	13.80	0.00
Three-rings	0.00	24.30	22.30	0.00	0.00	25.00	27.00	0.00
Cigar	0.00	15.79	16.84	0.00	0.00	17.63	16.32	0.00
Iris	30.67	33.33	33.33	30.67	0.67	4.67	5.33	0.67
Optical-digit	53.00	29.00	33.00	17.00	48.00	19.00	25.00	17.00
Breast-cancer	0.29	2.20	10.25	0.29	0.15	16.84	20.06	0.15
Log yeast	1.56	38.02	37.50	1.56	1.30	38.93	40.10	1.30
Std yeast	1.82	37.76	40.63	1.30	1.04	41.28	44.01	1.04
Wine	27.53	24.72	25.28	27.53	2.25	5.62	5.62	2.25
Glass	20.09	38.32	41.12	12.62	4.67	41.59	38.32	4.67
Wave	0.60	28.60	23.60	0.60	0.40	31.40	27.40	0.40
Wave + noise	0.60	38.40	29.60	0.60	0.40	41.80	38.20	0.40
Credit	0.31	31.14	32.06	0.31	0.15	30.37	24.85	0.15
Average	10.50	27.87	28.34	7.11	4.54	24.16	25.08	2.16

(5) *Credit-screening*—the original data-set consists of 690 samples. After removing those with missing attribute values, we have 652 complete samples.

The characters of all these data-sets are summarized in Table 3.

3.2. Performance of algorithms

For each data-set, we run K-means d times with given numbers of clusters, K, as initialization, and therefore we obtain d partition labels for clustering aggregation. We then transport these original clustering results to all four approaches, respectively, to get the combined partition. In addition, we also run these algorithms with the pre/post-process, following the steps described in Table 2, to test its capability. Since we have all the real labels of these data-sets, we can calculate error rates under the criterion used by Fred and Jain in Ref. [9]. For EA and PA, we apply SL to the step of MST. In addition, considering that MCLA and HBGF cannot automatically determine the right number of clusters for their following re-partition by METIS, we input the cluster numbers PA detected for them to use.

This experiment has two parameters to be designed previously: d, the number of clusterings to combine, and K, the number of clusters for K-means. For the parameter d, we set it as 10 and 50, and conduct experiments, respectively. For the parameter K, Fred and Jain suggested that it could be randomly generated for each K-means, but can neither be too large nor too small [9]. Small values of K are not able to capture the complexity of the data, while large values may produce an over fragmentation of the data. Besides, they also suggested that the initial K could be set as the square root of the data-set's sample numbers [10]. In another word, if the data-set contains N samples, K should be set as $\sqrt{N}$. In our experiments, of all the data-sets, the maximum number of clusters is 10, and therefore we set $K_{\min}$ as 10. On the other hand, of all the data-sets in our experiment, the maximum number of samples is 683, while the minimal number is only 100. Besides, the average number is 348.8, and its square root is 18.7. Thus, we set 20 to be the medium value of the range. As a result, we decide to set the range [10, 30] for random initialization of K.

Similar to Ref. [9], for each data-set, we repeat the experiment for 50 times and get 50 error rates, respectively, for each algorithm. We record the minimum one of all these 50 error rates, and also calculate the mean of them. All these minimum error rates and average error rates for $d = 10$ and 50 are listed in Tables 4–7. Additionally, we also calculate each algorithm's average error rates on all data-sets, and list them at the bottom of the tables. There, algorithms with pre/post-process are, respectively, referred to as P-EA, P-MCLA, P-HBGF, and P-PA for short.

As listed in Table 4, PA has the best performance of all these methods. Its minimum error rates and average error rates of 50

Table 6
Average error rates (in percentage) of combining 50 clusterings

Data-set	Without pre/post-process				With pre/post-process			
	EA	MCLA	HBGF	PA	P-EA	P-MCLA	P-HBGF	P-PA
Half-rings	0.00	25.42	25.20	0.00	0.00	19.98	20.81	0.00
Three-rings	0.80	48.85	49.33	0.00	0.00	49.53	49.51	0.00
Cigar	5.75	24.39	24.87	8.61	4.09	20.98	21.46	3.27
Iris	33.33	33.57	33.80	33.33	18.25	33.43	32.97	17.85
Optical-digit	64.86	59.88	60.22	64.66	61.46	55.62	56.30	60.30
Breast-cancer	30.05	28.16	31.74	25.08	0.42	30.08	32.30	0.41
Log yeast	59.79	45.65	47.42	56.45	55.22	45.22	47.89	51.71
Std yeast	46.10	48.76	52.49	42.47	45.03	49.70	52.58	41.92
Wine	36.38	37.38	37.90	36.61	31.00	30.08	32.61	30.20
Glass	45.28	51.00	51.11	44.55	41.39	53.55	54.39	39.01
Wave	46.01	48.27	45.51	30.60	18.63	48.10	46.87	16.04
Wave + noise	44.93	47.21	45.11	30.77	31.94	50.96	52.20	17.60
Credit	35.05	63.87	64.66	31.79	3.56	41.61	44.66	2.69
Average	34.49	43.26	43.80	31.15	23.92	40.68	41.89	21.62

Table 7
Minimum error rates (in percentage) of combining 50 clusterings

Data-set	Without pre/post-process				With pre/post-process			
	EA	MCLA	HBGF	PA	P-EA	P-MCLA	P-HBGF	P-PA
Half-rings	0.00	23.80	23.80	0.00	0.00	12.80	14.40	0.00
Three-rings	0.00	28.50	29.60	0.00	0.00	26.70	25.90	0.00
Cigar	0.00	17.37	18.42	0.00	0.00	16.32	17.89	0.00
Iris	33.33	33.33	33.33	33.33	2.00	5.33	3.33	2.00
Optical-digit	54.00	42.00	36.00	54.00	20.00	32.00	34.00	20.00
Breast-cancer	0.29	2.49	3.95	0.29	0.15	12.88	16.98	0.15
Log yeast	1.30	35.42	36.46	1.56	1.04	34.64	36.72	1.04
Std yeast	1.82	40.76	39.32	1.30	1.56	38.93	44.40	1.56
Wine	27.53	26.97	26.40	27.53	2.25	6.74	5.06	2.25
Glass	21.96	35.05	36.45	21.96	5.14	41.12	35.51	4.67
Wave	0.60	32.20	24.20	0.60	0.40	34.20	27.60	0.40
Wave + noise	0.60	38.60	34.20	0.60	0.40	41.00	40.40	0.40
Credit	0.46	50.31	50.61	0.31	0.61	37.27	41.72	0.61
Average	10.92	31.29	30.21	10.88	2.58	26.15	26.46	2.54

runs are mostly lower than those of its three peers. As illustrated by the average performance on all data-sets, using PA instead of EA can help reduce error rates by more than 3%. It can reduce error rates by around 10% compared with using MCLA and HBGF under the same circumstance. It means that our algorithm can produce more accurate and robust consolidated partitions, on basis of the same original clustering ensemble. In addition, we can easily find the proposed pre/post-process can improve the performance of both EA and PA evidently, by reducing the average error rates by around 10%. Aside from that, it can also enhance the performance of MCLA and HBGF. This result verifies the effectiveness and efficiency of pre/post-process.

As to the Half-rings and Cigar data-sets, PA performed slightly worse than EA when carried out directly. This is due to that samples in these data-sets do not obey a uniform distribution, which can be easily found from Fig. 2. In Fig. 2a, two clusters are, respectively, composed of 400 and 100 samples. However, the areas they expanded are of a similar size. It means that they have much different densities, and therefore PA is not applicable to this situation. Cigar also suffers from this problem. Although PA performs worse than EA on these two data-sets, P-PA has lower error rates than both EA and P-EA. It means that the pre/post-process can strengthen both these two algorithms, especially for PA. The data-sets of Half-rings, Three-rings and Cigar after the pre-processing step of normalization are shown in Fig. 3, in which we can easily find that these samples distribute more uniformly.

It is also interesting to analyze the experimental results on Iris. From results in Table 3, we can see that results of EA and PA are same. This phenomenon is due to the fact that Iris contains three clusters, and each of them consists of 50 samples. As described in last section, when data-sets are composed of clusters of the same size, the *p*-association matrix would degenerate to co-association matrix, and therefore PA and EA would produce the same combined clustering.

Statistic results of combining 50 clusterings in Tables 6 and 7 also reveal the same tendency as addressed above, and can further validate that the proposed PA algorithm outperforms other three algorithms, i.e., EA MCLA and HBGF, in clustering aggregation, and the pre/post-processing can help to further improve capability of all these four methods, especially on EA and PA.

4. Conclusion

In this paper, we have proposed a new approach for clustering aggregation called *probability accumulation*, which has provided a general framework for the aggregation of partitions. Based on it, we deduce two simplified form of our algorithm: one is for the situation that probability densities can be obtained, while another is applicable to situations when only partition labels are provided. Moreover, we propose the pre-processing and post-processing methods to deal with data-sets that do not satisfy the uniform distribution. Finally, we compare our algorithms' behaviors with three other

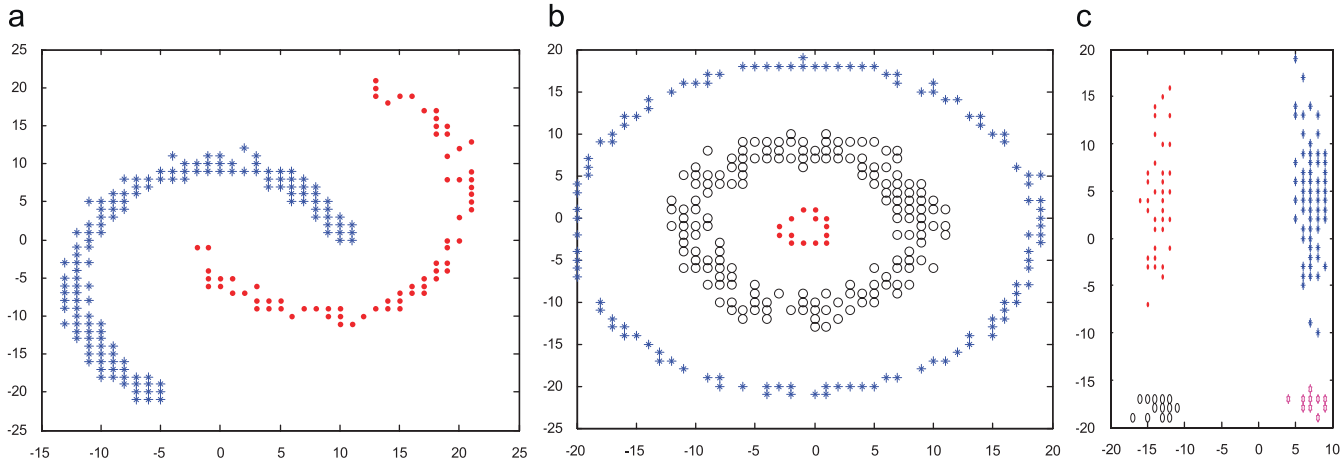

Fig. 3. Data-sets after the pre-processing step of normalization. (a) Half-rings, (b) Three-rings, and (c) Cigar.

state-of-the-art approaches for clustering aggregation, and the experimental results illustrated our algorithm's effectiveness.

Acknowledgments

This work is under the Services Science joint research project between Tsinghua University and IBM China Research Lab. It is also supported by Natural Science Foundation of China under Grants 60573062, 60673106 and 60721003, and the Specialized Research Fund for the Doctoral Program of Higher Education.

References

[1] P.M.D. Judd, A. Jain, Large-scale parallel data clustering, IEEE Trans. Pattern Anal. Mach. Intell. 19 (2) (1997) 153–158.
[2] S. Bhatia, J. Deogun, Conceptual clustering in information retrieval, IEEE Trans. Syst. Man Cybern. 28 (3) (1998) 427–536.
[3] C. Carpineto, G. Romano, A lattice conceptual clustering system and its application to browsing retrieval, Mach. Learn. 24 (2) (1996) 95–122.
[4] H. Frigui, R. Krishnapuram, A robust competitive clustering algorithm with applications in computer vision, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 450–466.
[5] A.K. Jain, M.N. Murty, P. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[6] R.O. Duda, P.E. Hart, D.G. Stock, Pattern Classification, second ed., Wiley, New York, USA, 2001.
[7] Z. Bian, X. Zhang, Pattern Recognition, second ed., Tsinghua University Press, Beijing, China, 2000.
[8] R.D.J. Kittler, M. Hatef, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[9] A. Fred, A. Jain, Combining multiple clusterings using evidence accumulation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 835–850.
[10] A. Fred, A. Jain, Data clustering using evidence accumulation, in: Proceedings of the 16th International Conference on Pattern Recognition (ICPR), Quebec City, 2002, pp. 276–280.
[11] A. Fred, A. Jain, Evidence accumulation clustering based on the k-means algorithm, in: Proceedings of the International Workshops on Structural and Syntactic Pattern Recognition (SSPR), 2002, pp. 442–451.
[12] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.
[13] X.Z. Fern, C.E. Brodley, Solving cluster ensemble problems by bipartite graph partitioning, in: Proceedings of the 21st International Conference on Machine Learning (ICML), Banff, Canada, 2004.
[14] A. Topchy, A. Jain, W. Punch, A mixture model of clustering ensembles, in: Proceedings of the SIAM International Conference on Data Mining, 2004.
[15] S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure, Bioinformatics 19 (9) (2003) 1090–1099.
[16] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ.
[17] X. Zhu, Fundamentals of Applied Information Theory, first ed., Tsinghua University Press, Beijing, China, 2001.

About the Author—XI WANG was born in 1983. He received the B.S. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2006, where he is currently pursuing the master degree. His research interests are in pattern recognition, machine learning, data mining and intelligent information processing.

About the Author—CHUNYU YANG was born in 1982. He received the B.S. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2004, where he is currently pursuing the Ph.D. degree. His research interests are in pattern recognition, machine learning, data mining and intelligent information processing.

About the Author—JIE ZHOU was born in 1968. He received the B.S. and M.S. degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the Ph.D. degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From 1995 to 1997, he was a Postdoctoral Fellow with the Department of Automation, Tsinghua University, Beijing, China. Currently, he is a Full Professor and Assistant Chair with the Department of Automation, Tsinghua University. His research area includes pattern recognition, image processing, computer vision, and information fusion. Recently, he has authored more than 10 papers in international journals and more than 40 papers in international conferences. He is an associate editor for the International Journal of Robotics and Automation. Dr. Zhou received the Best Doctoral Thesis Award from HUST, the First Class Science and Technology Progress Award from the Ministry of Education, China, and the Excellent Young Faculty Award from Tsinghua University in 1995, 1998, and 2003, respectively.