

Combining Fuzzy Partitions Using Fuzzy Majority Vote and KNN

Chun sheng Li¹ Yaonan Wang² Haidong Yang³

¹ Department of Mathematics and Computational Science, Guang Dong University of Business Studies, Guangzhou, China, 510320, Email: lcs5812084@sina.com

² College of Electrical and Information Engineering, Hunan University, Changsha, China, 410083

³ College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, 510635

Abstract—this paper firstly generalizes majority vote to fuzzy majority vote, then proposes a cluster matching algorithm that is able to establish correspondence among fuzzy clusters from different fuzzy partitions over a common data set. Finally a new combination model of fuzzy partitions is build on the basis of the proposed cluster matching algorithm and fuzzy majority vote. Comparative results show that the proposed combination model is able to foster strengths and circumvent weaknesses of component fuzzy partitions and to combine the component fuzzy partitions into a better fuzzy partition than any of component fuzzy partitions and those resulted from two current combination models of fuzzy partitions.

Index Terms—fuzzy vote, fuzzy majority vote, combination of fuzzy partitions, evaluation of fuzzy partition

I. INTRODUCTION

Fuzzy clustering has been proved preferable to crisp clustering and a number of fuzzy clustering algorithms [1-4] have been proposed. However, different fuzzy clustering algorithms may produce different fuzzy partitions over the common data set, and none of them are universal enough to perform equally well in any cases. For example, FCM [1] performs well on noiseless dataset with hyper-spherical shape, G-k [2] algorithm on noiseless dataset with hyper-ellipsoidal shape and both AFCM[3] and PFCM[4] are similar to FCM except that they are robust to noises. For a life dataset it may be of different shapes, therefore no single fuzzy clustering algorithm can accurately discover its structure and it makes some errors. However the errors made by different fuzzy clustering algorithms would not necessarily overlap. This suggests that different clusterings potentially offer complementary information about the patterns to be partitioned, which could be harnessed to improve the performance of pattern recognition systems. Therefore, A promising direction for accurate discovery of the data structure may be to combine diverse fuzzy partitions into a consolidate one, which is expected to merge advantages of multiple candidate fuzzy clusterings into one whole. Similar problems associated with crisp clusterings have been studied extensively and there is an extensive body of work on combining multiple crisp clusterings [5-8]. However, the topic of combining fuzzy clusterings has not received the same attention. Evgenia Dimitriadou[9]

proposes a combination scheme for fuzzy clusterings that aims to find a consensus fuzzy partition which optimally represents the set of component fuzzy clusterings over the same data set. A.D. Gordon [10] also presents a combination model that aims to identify a consensus fuzzy partition which closely fits the set of component fuzzy partitions over the same data set. However, no theory guarantees that a consensus fuzzy partition representing or fitting a set of fuzzy partitions can represent or fit the real structure of the data set. The current paper also addresses the problem of combining fuzzy partitions with the same number of clusters over the same data set.

There are two difficult problems in combining multiple fuzzy partitions. One is to establish the correspondences among clusters of the component fuzzy partitions so that the first cluster of one partition means the same as that of another one, so is the second cluster and so on, the other problem is to design the rule of combining multiple fuzzy partitions. To solve the first problem, Evgenia Dimitriadou [9] first builds up the confusion matrix between the consensus fuzzy partition that is initialized by one of the component fuzzy partitions and the component fuzzy partition, then the first two clusters associated with the first maximum element of the confusion matrix correspond to each other, so do the second two clusters associated with the second maximum element of the confusion matrix, and so on. Since the initial consensus fuzzy partition is randomly selected from the set of component fuzzy partitions and then updated by each of the other component fuzzy partitions step by step, the resultant consensus fuzzy partition suffers from both the initial consensus fuzzy partition and the order of the component fuzzy partition to take part in updating the consensus fuzzy partition. Unlike Evgenia Dimitriadou[9], A.D. Gordon[10] first builds up the dissimilarity matrix between the consensus fuzzy partition that is initialized randomly and each of the component fuzzy partitions, then treats the problem of cluster correspondence as the problem of assignment and solves it by Hungarian method[11]. The resultant consensus fuzzy partition suffers from the initialization of the consensus fuzzy partition.

To overcome the sensitivity of the above approaches to the initial consensus fuzzy partition in matching clusters from different fuzzy partitions, we transform the

problem of establishing the correspondence among the clusters of component fuzzy partitions into the problem of partitioning their cluster centers so that the cluster centers in the same cluster correspond to each other. The second problem is solved by generalizing the majority voting rule for ensemble of crisp partitions to the fuzzy majority voting rule for ensemble of fuzzy partitions. Based on this, a new combination model of fuzzy partitions is build, and its performance are studied intensively by simulation experiments.

The rest of this paper is organized as follows. In section II the related work is reviewed. The traditional majority voting rule is generalized to the fuzzy majority voting rule in section III. An algorithm for matching clusters of different fuzzy partitions is proposed in section IV. A new combination model of fuzzy partitions is proposed in section V. Numerical experiments and conclusions are given in section VI and VII, respectively.

II. Related Work

A The Voting Algorithm [9]

The main idea of literature [9] is to find partition P of a given data set $X=\{x_1, x_1, \dots, x_N\}$ with g clusters which optimally represents a given set of M partitions of X . Each of these M partitions is represented by an $N \times g$ membership matrix U_h ($h=1, 2, \dots, M$). The final partition P_M is encoded as an $N \times g$ matrix. The element $u_{ij}^{(h)}$ of U_h is the degree of membership of x_i to the j -th class of the h -th partition. We denote the i -th row of U_h as $u_i^{(h)}$, that is $u_i^{(h)}$ is the membership vector of the pattern x_i for the partition U_h . The final partition P is encoded as a $N \times g$ matrix with elements p_{ij} and rows p_i . The task of finding an optimal partition is given by the minimization problem:

$$\min_P l(U_1, U_2, \dots, U_M, P) = \min_{P_1, \dots, P_N} \min_{\Pi_1, \dots, \Pi_M} \left(\frac{1}{M} \sum_{h=1}^M \frac{1}{N} \sum_{i=1}^N \|\Pi_h(u_i^{(h)}) - p_i\|^2 \right) \tag{1}$$

Where $\Pi_h(U_h)$ is any permutation of the columns of U_h .

This minimization problem is solved by the voting algorithm[9], which is described in Table I.

Table 1 Voting Algorithm [9]

Step1 set $P^{(1)}=U_1$ and $\hat{\Pi}_1 = id$ (id means identical permutation);
Step2 for $m=2$ to M
(a) compute the solution $\hat{\Pi}_m$ of
$\max_{\Pi_m} tr \left(\left(\sum_{l=1}^{m-1} \hat{\Pi}_l(U_l) \right) \Pi_m(U_m) \right) = \max_{\Pi_m} tr \left((P^{(m-1)}) \Pi_m(U_m) \right)$
by the following approximation algorithm:
(1) build up the confusion matrix between P and U;
(2) find the maximum element in this confusion matrix;
(3) associate the two clusters corresponding to the maximum element;
(4) remove these two clusters;
(5) with the reduced confusion matrix go to (2);
(b) compute the voting result $P^{(m)}$ after m runs as
$P^{(m)} = \frac{m-1}{m} P^{(m-1)} + \frac{1}{m} \hat{\Pi}_m(U_m)$

$P^{(m)}$ denotes the voting result after the first m steps.

B A Model for Fitting a Fuzzy Consensus Partition to a Set of Membership Functions [10]

This model identifies the ‘‘closest’’ consensus fuzzy partition P_M fitting its membership function matrix U_M to the membership function matrices $\{U_h$ ($h=1,2,\dots,M$)}, that have been permuted to ‘‘best’’ match g classes of P_h with g classes of P_M . The ‘‘closest’’ consensus fuzzy partition P_M of $\{P_h$ ($h=1,2,\dots,M$)} can be obtained by solving the following problem in the integer variables $Y_h=[y_{hpl}]$ and nonnegative membership functions $U_M = (u_{ij}^{(M)})$:

$$[P1] \min F(Y_1, Y_2, \dots, Y_r, U_M) = \sum_{h=1}^r w_h \|U_h Y_h - U_M\|^2 \tag{2}$$

$$= \sum_{h=1}^M \sum_{i=1}^N \sum_{p=1}^g \sum_{l=1}^g w_h (u_{ip}^{(h)} - u_{il}^{(M)})^2 y_{hpl}$$

Subject to the constraints

$$\sum_{p=1}^g y_{hpl} = 1 (l=1,2,\dots,g; h=1,2,\dots,M) \tag{3}$$

$$\sum_{l=1}^g y_{hpl} = 1 (p=1,2,\dots,g; h=1,2,\dots,M) \tag{4}$$

$$y_{hpl} \in \{0,1\} (p,l=1,2,\dots,g; h=1,2,\dots,M) \tag{5}$$

$$m_{il} \geq 0 (i=1,2,\dots,N; l=1,2,\dots,g) \tag{6}$$

$$\sum_{l=1}^g m_{il} = 1 (i=1,2,\dots,N) \tag{7}$$

The constrained problem [P1] can be minimized by means of the alternating least-square algorithm (ALS) described in Table II, that alternates between minimizing $F(Y_1, Y_2, \dots, Y_M, U_M)$ with respect to $\{Y_h$ ($h=1,2,\dots,M$)} given the current estimate of the median membership function matrix U_M ; and minimizing $F(Y_1, Y_2, \dots, Y_M, U_M)$ with respect to U_M given the current matching of classes between Y_h and U_M ($h=1, 2, \dots, M$).

Table 2 The Alternating Least-square Algorithm (ALS)

Step1 Given the estimates of the median membership function matrix U_M , new least-squares estimates of the elements of Y_h ($h=1, 2, \dots, M$) can be determined by solving M independent matching problems:
$[P1a] \min \Delta_l(U_h, U_M, Y_h) = \sum_{i=1}^N \sum_{p=1}^g \sum_{l=1}^g (u_{ip}^{(h)} - u_{il}^{(M)})^2 y_{pl}$ ($h=1,2,\dots,M$)
Subject to constraints (3), (4) and (5).
([P1a] can be efficiently solved using the well-known Hungarian method [11] in $O(g^3)$ time complexity).
Step2 treating the elements of Y_h ($h=1, 2, \dots, M$) as constraints, it is necessary to solve:
$[P1b]: \min F(Y_1, Y_2, \dots, Y_M, U_M)$
Subject to constraints (6) and (7).
The solution is given by
$u_{il}^{(M)} = \frac{\sum_{h=1}^M \sum_{p=1}^g w_h u_{ip}^{(h)} y_{hpl}}{\sum_{h=1}^M w_h} (i=1,2,\dots,N; l=1,2,\dots,g)$

C Majority Voting Rule (MAJ)

This rule does not require the *a posteriori* outputs for each class, and each classification gives only one crisp class output as a vote for that class. Then, the ensemble output is assigned to the class with the maximum number of votes among all classes. For any sample $x \in X$, for a group of M classification in a g -class problem, we denote the decision of label outputs for x from classification $f(i)$ is $c(i)$, $1 \leq c(i) \leq g$. Several terminologies are defined in the following.

Definition 1 For a sample $x \in X$, the crisp vote $d_{i,l}(x)$ for class l given by classification $f(i)$ is defined as

$$d_{il}(x) = \begin{cases} 1 & c(i) = l \\ 0 & c(i) \neq l \end{cases} \quad (8)$$

$d_{il}=1$ means that classification $f(i)$ votes for class l , and $d_{il}=0$ means that classification $f(i)$ votes against class l .

Definition 2 For a sample $x \in X$, the discriminating function $g(l|x)$ for class $l (1 \leq l \leq g)$ is defined as

$$g(l|x) = \sum_{i=1}^M d_{il}(x) \quad (9)$$

The discriminating function represents the total number of votes given to a class by all classifications. The higher value of the discriminating function $g(l|x)$ indicates more supports for class l . Consequently, the output of an ensemble of classifications is the class label with the maximum value of the discriminating function.

$$k = \arg \max_{1 \leq l \leq g} g(l|x) \quad (10)$$

III Fuzzy Majority Voting Rule

The definition of crisp vote $d_{il}(x)$ indicates that a crisp partition $f(i)$ either supports or denies utterly that pattern x belongs to class l . Since a fuzzy partition considers that any patterns belong to all clusters with different membership degrees, the crisp vote has to be revised before it is applied to fuzzy partitions.

Since a crisp clustering is a special case of fuzzy clustering, Definition 1 indicates that the crisp vote $d_{il}(x)$ is actually the membership degree of pattern x belonging to class l . From this point of view the natural way of generalizing a crisp vote to a fuzzy vote is to define the fuzzy vote $\tilde{d}_{il}(x)$ given to pattern x by the fuzzy clustering $\tilde{f}(i)$ as the fuzzy membership degree $u_i^{(l)}(x)$ of pattern x belonging to class l derived from the fuzzy clustering $\tilde{f}(i)$. This yields that the consensus fuzzy partitions is the mean of all the component fuzzy partitions, which is the optimal representation of all the component fuzzy partitions, just as Evgenia Dimitriadou et al stated in the literature[9]. The reason can be found in remarks at the end of this section. However experiments in section VI show that the mean of all the component fuzzy partitions is not sure to represent the real structure of the data set. Considering this, we do not simply define the fuzzy vote $\tilde{d}_{il}(x)$ as the fuzzy membership degree $u_i^{(l)}(x)$, but treat the classes differently, that is, we directly define $\tilde{d}_{il}(x) = u_i^{(l)}(x)$ for class $l = \arg \max_{1 \leq l \leq g} u_i^{(l)}(x)$, but for other classes, we define

$$\tilde{d}_{ik}(x) = u_k^{(i)}(x) \left(1 - \max_{1 \leq l \leq g} u_l^{(i)}(x) \right), \quad k \in \{1, 2, \dots, g\} \text{ and } k \neq l.$$

Experiments in section VI show that this definition of fuzzy vote yields relatively good consensus fuzzy partition. The following definitions are the fuzzy counterparts of definitions 1-2.

Definition 3 for a pattern $x \in X$, the fuzzy vote given to class l by the fuzzy partition $U^{(i)} = (u_l^{(i)}(x))_{N \times g}$ is defined as

$$\tilde{d}_{il}(x) = \begin{cases} u_l^{(i)}(x) & l = \arg \max_{1 \leq l \leq g} u_l^{(i)}(x) \\ u_l^{(i)}(x) \left(1 - \max_{1 \leq l \leq g} u_l^{(i)}(x) \right) & \text{otherwise} \end{cases} \quad (11)$$

Where N is the number of patterns, g the number of clusters and $u_l^{(i)}(x)$ the membership degree of pattern x belonging to cluster l . Contrary to the crisp vote, the fuzzy vote indicates that a fuzzy partition neither supports nor denies utterly that a pattern belongs to a cluster, but supports it belongs to all clusters to different extents. It is obvious that the crisp vote is the special case of the fuzzy vote.

Definition 4 for a pattern $x \in X$, the fuzzy discriminating function for class $l (1 \leq l \leq g)$ is defined as

$$\tilde{g}(l|x) = \sum_{i=1}^M \tilde{d}_{il}(x) \quad (12)$$

Where M is the number of component fuzzy partitions. Like the discriminating function defined by formula (9), the fuzzy discriminating function of a class also represents the amount of supports given to it by all fuzzy partitions. Higher value of the fuzzy discriminating function $\tilde{g}(l|x)$ means more supports for pattern x belonging to class l . Unlike simple majority voting rule, instead of assigning a class label to pattern x , we calculate the membership degree $cu_l(x)$ of x belonging to each class $l, 1 \leq l \leq g$, determined by M fuzzy partitions jointly as follows

$$cu_l(x) = \tilde{g}(l|x) / \sum_{l=1}^g \tilde{g}(l|x), 1 \leq l \leq g \quad (13)$$

Formula (13) indicates that the combination of fuzzy partitions is still a fuzzy partition. This is different from the consensus crisp partition. If formula (13) is replaced with $k = \arg \max_{1 \leq l \leq g} \tilde{g}(l|x)$, it is obvious that the majority voting rule is the special case of the fuzzy majority voting rule.

In the following we exemplify the fuzzy majority voting rule. Supposing that there are three component fuzzy partitions over the same data set, each of which has three clusters and is denoted by its fuzzy partition matrix $U^{(i)} (i=1, 2, 3)$. In the case that the correspondence among clusters from the component fuzzy partitions is established, i.e., the first column of the fuzzy component partitions represents the same class, so are the second and third column. Given a pattern x , the membership degree of x belonging to each cluster derived from $U^{(i)} (i=1, 2, 3)$ is

$$\begin{aligned} (u_1^{(1)}(x), u_2^{(1)}(x), u_3^{(1)}(x)) &= (0.1894, 0.6894, 0.1212) \\ (u_1^{(2)}(x), u_2^{(2)}(x), u_3^{(2)}(x)) &= (0.4187, 0.2761, 0.3052) \\ (u_1^{(3)}(x), u_2^{(3)}(x), u_3^{(3)}(x)) &= (0.2527, 0.4743, 0.2730) \end{aligned}$$

Definition 3 yields

$$(\tilde{d}_{11}(x), \tilde{d}_{12}(x), \tilde{d}_{13}(x)) = (0.0588, 0.6894, 0.0376).$$

$$\begin{aligned} (\tilde{d}_{21}(x), \tilde{d}_{22}(x), \tilde{d}_{23}(x)) &= (0.4187, 0.1605, 0.1774) \\ (\tilde{d}_{31}(x), \tilde{d}_{32}(x), \tilde{d}_{33}(x)) &= (0.1328, 0.4743, 0.1435) \end{aligned}$$

Formular (12) gives

$$(\tilde{g}(1|x), \tilde{g}(2|x), \tilde{g}(3|x)) = (0.6103, 1.3242, 0.3585)$$

Formula (13) results in

$$(cu_1(x), cu_2(x), cu_3(x)) = (0.2662, 0.5775, 0.1563).$$

Remarks: if $\tilde{d}_{ij}(x) = u_{ij}(x)$, then

$$\begin{aligned} cu_i(x) &= \frac{\tilde{g}(i|x)}{\sum_{t=1}^g \tilde{g}(t|x)} \\ &= \frac{\sum_{j=1}^M u_j^{(i)}(x)}{\sum_{t=1}^g \sum_{j=1}^M u_t^{(i)}(x)} \\ &= \frac{\sum_{j=1}^M u_j^{(i)}(x)}{\sum_{i=1}^M \sum_{t=1}^g u_t^{(i)}(x)} \\ &= \frac{1}{M} \sum_{i=1}^M u_i^{(i)}(x) \end{aligned}$$

IV A Cluster Matching Algorithm Based on KNN

When no a priori class information for the patterns is available, a direct application of fuzzy majority voting rule to combining fuzzy partitions is not possible, for it is not immediately clear which cluster from a specific partition corresponds to what in another. Therefore, it is necessary to establish the correspondence among clusters of all component fuzzy partitions so that the same column of $U^{(i)} = (u_j^{(i)}(x))_{N \times g}, i=1,2,\dots,M$ defines the same cluster. Both Evgenia Dimitriadou[9] and A.D. GORDON[10] establish this kind of correspondence by corresponding the clusters of each component fuzzy partition with those of the consensus fuzzy partition, which is firstly initialized randomly or with one of the component fuzzy partitions, then updated adaptively. They suffer from the initialization of the consensus fuzzy partition. The underlying idea of establishing the correspondence between clusters of one fuzzy partition and those of another is that the similar clusters correspond to each other so that the sum of dissimilarities between two fuzzy partitions is minimized, as shown in the literatures [9, 10]. Inspired by this idea, we transfer the problem of pairing clusters from different fuzzy partitions into the problem of partitioning the set of cluster centers. Supposing that there are M fuzzy partitions $U^{(1)}, U^{(2)}, \dots, U^{(M)}$, each of which has g clusters. Each cluster is represented by its center. This yields a set of cluster centers, $V = \{v_1^{(1)}, \dots, v_g^{(1)}, v_1^{(2)}, \dots, v_g^{(2)}, \dots, v_1^{(M)}, \dots, v_g^{(M)}\}$, where $v_j^{(i)}$ is the j -th cluster center of the i -th fuzzy partition. We define the similarity between two clusters as the Euclidian distance between their centers. Consequently, establishing the correspondence among the clusters of M fuzzy partitions is transferred into partitioning the set V of center vectors into g clusters, each of which contains M center vectors from different fuzzy partitions. The center vectors belonging to the same cluster correspond

to each other, i.e., if $v_{j_1}^{(1)}, v_{j_2}^{(2)}, \dots, v_{j_M}^{(M)}$ belong to the same cluster, then the j_1 -th cluster of $U^{(1)}$, the j_2 -th cluster of $U^{(2)}$, ..., the j_M -th cluster of $U^{(M)}$ define the same cluster, where $s \in \{1, 2, \dots, M\}, j_s \in \{1, 2, \dots, g\}$. To assure the center vectors in the same cluster are from different fuzzy partitions, we define the dissimilarity between two cluster centers as

$$d(v_j^{(i)}, v_t^{(s)}) = \begin{cases} \|v_j^{(i)} - v_t^{(s)}\|_2^2, & i \neq s \\ +\infty, & i = s, j \neq t \end{cases} \quad (14)$$

The K nearest neighbors method (KNN) is employed to partition the data set V into g clusters, each of which contains M center vectors. Consequently, an approach to establishing the correspondence among the clusters from different fuzzy partitions is developed, which is described by the pseudo code in Table 3.

Table 3 The Cluster-matching Algorithm Based on KNN

- 1 Compute the centre vectors of each fuzzy partition by $v_k^{(i)} = \sum_{x \in X} u_k^{(i)}(x) \cdot x / \sum_{x \in X} u_k^{(i)}(x), k=1, \dots, g; i=1, \dots, M$
- 2 Compute dissimilarity $d(v_j^{(i)}, v_t^{(s)}), j, t=1, \dots, g, i, s=1, \dots, M$ using (14);
- 3 Find M nearest neighbours for each centre vector $v_t^{(i)}, t=1, \dots, g; i=1, \dots, M$. They form a M -nearest neighbourhood denoted by $\{v_t^{(i)}\}$;
- 4 Compute the compactness $comp(\cdot)$ of each M -nearest neighbourhood by $comp(\{v_t^{(i)}\}) = \sum_{v_1, v_2 \in \{v_t^{(i)}\}} d(v_1, v_2)$;
- 5 Select g most compact and disjoint M -nearest neighbourhoods. The centre vectors belonging to the same M -nearest neighbourhood represent the same cluster and the columns corresponding to them in $U^{(i)} (i=1, 2, \dots, M)$ are labeled as the same class label.

In the following we exemplify the proposed cluster-matching algorithm. Supposing that there are three component fuzzy partitions, each of which has three clusters. Their cluster centers are listed in Table 4. The dissimilarity between any pair of cluster centers is derived from formula (14). The M -nearest neighbourhood of each center vector and its compactness are listed in Table 4, where $M=3$. The set of nine cluster center vectors is partitioned into three disjoint clusters: $\{v_1^{(1)}, v_2^{(2)}, v_2^{(3)}\}, \{v_2^{(1)}, v_1^{(2)}, v_1^{(3)}\}$ and $\{v_3^{(1)}, v_3^{(2)}, v_3^{(3)}\}$. The cluster $\{v_1^{(1)}, v_2^{(2)}, v_2^{(3)}\}$ means that the first cluster of the first fuzzy partition, the second cluster of the second partition and the second cluster of the third fuzzy partition define the same cluster, so do $\{v_2^{(1)}, v_1^{(2)}, v_1^{(3)}\}$ and $\{v_3^{(1)}, v_3^{(2)}, v_3^{(3)}\}$.

V A Combination Scheme for Fuzzy Partitions Using Fuzzy Majority Voting Rule and KNN

Supposing that there are M fuzzy partitions over the data set X , each of which is denoted by a fuzzy partition matrix $U^{(i)}, i=1, 2, \dots, M$. They are matched by the cluster matching algorithm in Table 3, then the well

matched M fuzzy partitions are combined into a consensus fuzzy partition using fuzzy majority voting

rule. The pseudo description of the proposed combination model is given in Table 5.

Table 3 The Cluster Centers of Three Component Fuzzy Partitions

$v_1^{(1)}$	$v_2^{(1)}$	$v_3^{(1)}$	$v_1^{(2)}$	$v_2^{(2)}$	$v_3^{(2)}$	$v_1^{(3)}$	$v_2^{(3)}$	$v_3^{(3)}$
6.6294	5.0626	5.9525	5.0626	6.6294	5.9525	5.0626	6.6294	5.9525
3.0091	3.3618	2.7947	3.3618	3.0091	2.7947	3.3618	3.0091	2.7947
5.4126	1.6670	4.4202	1.6670	5.4126	4.4202	1.6670	5.4126	4.4202
1.9359	0.3271	1.4351	0.3271	1.9359	1.4351	0.3271	1.9359	1.4351

Table 4 the 3-nearest neighbourhood of each cluster center and its compactness

	3-nearest neighbourhood	compactness
$v_1^{(1)}$	$\{v_1^{(1)}, v_2^{(2)}, v_2^{(3)}\}$	0.2287×10^{-5}
$v_2^{(1)}$	$\{v_2^{(1)}, v_1^{(2)}, v_1^{(3)}\}$	0.0408×10^{-5}
$v_3^{(1)}$	$\{v_3^{(1)}, v_3^{(2)}, v_3^{(3)}\}$	0.2593×10^{-5}
$v_1^{(2)}$	$\{v_1^{(2)}, v_1^{(3)}, v_2^{(1)}\}$	0.0408×10^{-5}
$v_2^{(2)}$	$\{v_2^{(2)}, v_2^{(3)}, v_1^{(1)}\}$	0.2287×10^{-5}
$v_3^{(2)}$	$\{v_3^{(2)}, v_3^{(3)}, v_3^{(1)}\}$	0.2593×10^{-5}
$v_1^{(3)}$	$\{v_1^{(3)}, v_1^{(2)}, v_2^{(1)}\}$	0.0408×10^{-5}
$v_2^{(3)}$	$\{v_2^{(3)}, v_2^{(2)}, v_1^{(1)}\}$	0.2287×10^{-5}
$v_3^{(3)}$	$\{v_3^{(3)}, v_3^{(2)}, v_3^{(1)}\}$	0.2593×10^{-5}

Table 5 A Combination Model of Fuzzy Partitions Based on Fuzzy Majority Voting Rule and KNN

Step1 input the fuzzy partition matrices $U^{(i)}$ ($i=1, 2, \dots, M$) and pattern samples X ;
 Step2 establish the correspondence among clusters from M fuzzy partitions using the cluster-matching algorithm in Table 3;
 Step3 combine M fuzzy partitions using fuzzy majority voting rule described in section III;

V Experiments

An important consideration in the combination of partitions is that much better results can be achieved if diverse partitions, rather than similar partitions, are combined. To create diverse fuzzy partitions we employ three fuzzy clustering algorithms—FCM[1], PFCM[2] and AFCM[3], each of which (except PFCM[2] that is initialized by the output of FCM) is initialized by three centre initialization methods—CCIA[12], kd-tree[13] and MST[14], respectively. Therefore, there are totally nine fuzzy partitions denoted by FCM-CCIA, FCM-MST, FCM-kd-tree, PFCM-CCIA, PFCM-MST, PFCM-kd-tree, AFCM-CCIA, AFCM-MST, AFCM-kd-tree, respectively. They are combined into a consensus fuzzy partition, denoted by FMV, by the combination model in Table 5, in the way depicted in Fig. 1.

To test the performance of the proposed combination model, we compare it with two combination methods *voting* [9] and *ALS*[10] on four real data sets, which are described in Table 6.

For all fuzzy clustering algorithms we use the following *Computational Protocols*; convergence term $\epsilon=0.0001$, maximum number of iterations=100, the

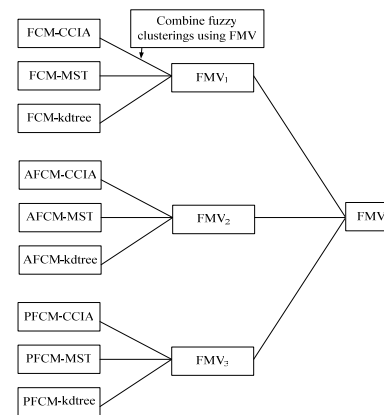


Fig.1 the flowchart of combining fuzzy partitions (FMV_{*i*}, ($i=1, 2, 3$) means the combination results of the first time and FMV the final result of combination.)

Table 6 The brief description of data sets

	Number of instances	Number of attributes	Number of clusters
Pima-Indians-diabete [15]	768	8	2
svmguide3[16]	1243	22	2
ionosphere[16]	351	34	2
Sat.image[16]	4435	36	6

fuzzifier $m=2$. The parameters of PFCM are initialized as follows: $m=2, \eta=1.5, a=1, b=3$. *ALS* suffers from the initialization of the consensus fuzzy partition. We initialize it with each of nine component fuzzy partitions respectively. The *voting* [9] algorithm suffers from the sequence of the component fuzzy partitions to take part in the combination of fuzzy partitions. We place nine component fuzzy partitions in the order of FCM-CCIA, FCM-MST, FCM-kd-tree, AFCM-CCIA, AFCM-MST, AFCM-kd-tree, PFCM-CCIA, PFCM-MST, PFCM-kd-tree, then initialize the consensus fuzzy partition $P^{(1)}$ with each of the above nine fuzzy partitions, respectively and fix others in their places.

We evaluate the fuzzy partition using pattern recognition rate PR that is a standard evaluation index, partition coefficient PC[18] that measures the fuzzy degree of fuzzy partitions, fuzzy Rand index ω_R and related indexes — fuzzy Jaccard coefficient ω_{JC} , fuzzy Fowlkes-Mallows index ω_{FM} , fuzzy Minkowski measure ω_M and fuzzy Γ statistic ω_Γ [17], which are objective criteria for the evaluation of fuzzy partitions, as R. J. G. B. Campello stated [17]. The big values of the indexes $\omega_R, \omega_{JC}, \omega_{FM}$ and ω_Γ indicate the good

closeness between the reference partition and the fuzzy partition to be evaluated, while the low value of ω_M reveals good closeness between them.

To test the performance of the proposed combination model of fuzzy partitions, we compare it with all component fuzzy partitions and two consensus fuzzy partitions, *voting* [9] and *ALS* [10], over a small size and a large size data set, ionosphere [15] and sat.image [16]. We also compare it with only two consensus fuzzy partitions, *voting* [9] and *ALS* [10], over two middle size data sets, *diabetes* and *svmguide3*. The comparative results listed in Table 7-9 indicate that, in terms of all evaluation indexes for fuzzy partitions, the proposed combination model FMV outperforms two consensus fuzzy partitions, *voting* [9] and *ALS* [10]. Table 7 and 9 show that, in terms of pattern recognition rate, the proposed combination model FMV is comparable to the best component fuzzy partitions FCM-CCIA, FCM-MST and FCM-kdtree, while in terms of other indexes, FMV is preferable to them over the data sets ionosphere and sat. image, respectively. Table 10 indicates that, in terms of CPU time, *voting* [9] is the cheapest, and FMV is a little more expensive computation than *ALS* [10]. In a word, the proposed combination method FMV is able to foster strengths and circumvent weaknesses of component fuzzy partitions and outperforms *voting* [9] and *ALS* [10] at the cost of a little extra computation in our experiments. It is important for the consensus fuzzy partition to be comparable, but uncertain to be preferable, to the best component fuzzy partition in any cases, for no fuzzy clustering algorithm can generate good partitions in all cases and we do not know which fuzzy clustering algorithm may produce a good clustering in advance. Furthermore, we do not know how to accurately assess a fuzzy partition, much less select the best individual fuzzy partition when no information about the data set is available. In this sense, the consensus fuzzy partition obtained from the proposed combination model is more stable and reliable than any component fuzzy partition,

for it can combine the advantages of all component fuzzy partitions and pool them into a consensus fuzzy partition that is uncertain to be better than any component fuzzy partition, but sure to be superior to an overwhelming majority of the component fuzzy partitions in any cases.

To compare FMV with *voting* [9], *ALS* [10] and component fuzzy partitions, we test them over the data set ionosphere. Table 7 shows that all evaluation indexes agree with our consensus fuzzy partition FMV outperforms *ALS* and *voting* a little. Compared to nine individual fuzzy partitions, in terms of pattern recognition rate, the consensus fuzzy partition generated by the proposed combination model is as good as the best component fuzzy partition FCM-CCIA, FCM-MST and FCM-kdtree, better than other six component fuzzy partitions. Table 7 also shows that other evaluation indexes agree with that FMV outperforms all the component fuzzy partitions. In general, the proposed consensus fuzzy partition FMV is able to combine the advantages of all component fuzzy partitions and at least comparable to the best component fuzzy partition. Contrary to FMV, in terms of pattern recognition rate, *ALS* and *voting* are worse than the best component fuzzy partition. Furthermore, *ALS* and *voting* are worse than part of the component fuzzy partitions in terms of other evaluation indexes. In a word, *ALS* and *voting* fail to combine the advantages of the component fuzzy partitions and are inferior to the best component fuzzy partitions. It is also revealed by Table 7 that when the pattern recognition rates of FMV, FCM-MST and FCM-kdtree are equal, other indexes for fuzzy partitions still can distinguish the three fuzzy partitions. This indicates that it is not sufficient to evaluate fuzzy partitions only by pattern recognition rate and other evaluation indexes for fuzzy partitions are also powerful tools for assessing fuzzy partitions.

To further compare the performances of FMV, *ALS* and *voting*, we test them over two middle size data sets, *diabetes* and *svmguide3*. Table 8 shows that FMV

Table 7 the comparative results among consensus fuzzy partitions and individual fuzzy partitions over ionosphere

Fuzzy partitions		PR(%)	PC	ω_R	ω_{JC}	ω_{FM}	ω_{Γ}	ω_M
FMV		70.9402	0.8151	0.5602	0.4056	0.5775	0.1205	0.9038
<i>ALS</i>	Mean	70.3704	0.5303	0.5161	0.3671	0.5373	0.0312	0.9480
	std	0	0	0.0000	0	0	0	0
<i>voting</i>	Mean	70.3704	0.5303	0.5161	0.3671	0.5373	0.0312	0.9480
	std	0	0.0000	0	0	0	0.0000	0
FCM-CCIA		70.9402	0.6487	0.5364	0.3899	0.5612	0.0703	0.9279
FCM-MST		70.9402	0.6487	0.5364	0.3899	0.5612	0.0703	0.9279
FCM-kdtree		70.9402	0.6487	0.5364	0.3899	0.5612	0.0703	0.9279
AFCM-CCIA		69.5157	0.5000	0.5000	0.3500	0.5189	0.0000	0.9636
AFCM-MST		69.5157	0.5000	0.5000	0.3500	0.5189	0.0000	0.9636
AFCM-kdtree		69.5157	0.5000	0.5000	0.3500	0.5189	0.0000	0.9636
PFCM-CCIA		69.2308	0.6423	0.5338	0.3847	0.5558	0.0662	0.9305
PFCM-MST		53.8462	0.5000	0.5000	0.3500	0.5189	0.0000	0.9636
PFCM-kdtree		60.3989	0.5031	0.5015	0.3518	0.5208	0.0028	0.9622

The bold number means the optimal value of evaluation index

outperforms *ALS* and *voting* by huge margins in terms of pattern recognition. The partition coefficients PC listed in Table 8 show that *ALS* and *voting* average conflicting memberships towards $1/c$ ($c=2$) over the data set *diabetes*. The fuzzy rand index and related

indexes listed in Table 8 agree with that FMV is a little better than *ALS* and *voting*. Table 8 also shows that *ALS* is as bad as *voting* in terms of all evaluation indexes.

Table 8 the comparative results among consensus fuzzy partitions over two small size data sets with two clusters

Data set	Consensus fuzzy partitions		PR(%)	PC	ω_R	ω_{JC}	ω_{FM}	ω_Γ	ω_M
<i>diabetes</i>	FMV		65.8854	0.7766	0.5252	0.4141	0.5863	0.0330	0.9334
	<i>ALS</i>	Mean	61.9792	0.5990	0.5118	0.3771	0.5477	0.0174	0.9465
		std	0	0	0	0	0	0	0
	<i>voting</i>	Mean	61.9792	0.5990	0.5118	0.3771	0.5477	0.0174	0.9465
std		0	0.0000	0.0000	0.0000	0	0.0000	0.0000	
<i>svmguide3</i>	FMV		57.2808	0.7696	0.5076	0.3980	0.5731	0.0121	0.8793
	<i>ALS</i>	Mean	50.6034	0.5406	0.5061	0.3975	0.5725	0.0077	0.8806
		std	0	0	0	0.0000	0	0	0.0000
	<i>voting</i>	Mean	50.6034	0.5406	0.5061	0.3975	0.5725	0.0077	0.8806
std		0	0.0000	0	0.0000	0	0	0.0000	

To further investigate the performance of FMV, we compare it with two consensus fuzzy partitions, *voting* [9] and *ALS* [10], and all the component fuzzy partitions over a large size data set with six clusters. Table 9 shows that three consensus fuzzy partitions, FMV, *voting* [9] and *ALS* [10], and three individual fuzzy partitions, FCM-CCIA, FCM-MST and FCM-kdtree, are very close to each other and FMV is a little better than the best individual fuzzy partition in terms of pattern recognition rate. In terms of other evaluation indexes, *voting* [9] and

ALS [10] are inferior to FCM-CCIA, FCM-MST, FCM-kdtree, PFCM-CCIA and PFCM-MST, while FMV is superior to all component individual fuzzy partitions, as shown by Table 9. This experiment once more confirms that FMV is able to combine the advantages of component individual fuzzy partitions and at least comparable to the best component fuzzy partition in the case of large size data sets with multiple clusters, while *voting* [9] and *ALS* [10] can not.

Table 9 the comparative results among consensus fuzzy partitions and individual fuzzy partitions over Sat. image

Fuzzy partitions		PR(%)	PC	ω_R	ω_{JC}	ω_{FM}	ω_Γ	ω_M
FMV		70.9583	0.6067	0.7883	0.3959	0.5737	0.4895	1.0090
<i>ALS</i>	Mean	70.5850	0.3700	0.6955	0.2974	0.4732	0.3814	1.1995
	std	0.4152	0.0003	0.0010	0.0013	0.0014	0.0019	0.0032
<i>Voting</i>	Mean	69.8409	0.3657	0.6950	0.2959	0.4713	0.3793	1.2009
	std	0.8031	0.0034	0.0027	0.0028	0.0034	0.0044	0.0038
FCM-CCIA		70.3495	0.4799	0.7407	0.3262	0.4995	0.4034	1.1090
FCM-MST		70.3495	0.4799	0.7407	0.3262	0.4995	0.4034	1.1090
FCM-kdtree		70.3495	0.4799	0.7407	0.3262	0.4995	0.4034	1.1090
AFCM-CCIA		59.6843	0.3570	0.6661	0.2552	0.4174	0.3198	1.2386
AFCM-MST		63.7880	0.3675	0.6761	0.3056	0.4934	0.4082	1.2467
AFCM-kdtree		60.2029	0.3340	0.6382	0.2389	0.3985	0.2988	1.2816
PFCM-CCIA		62.6607	0.4223	0.7083	0.3253	0.5027	0.4159	1.1388
PFCM-MST		66.8997	0.4383	0.7198	0.3238	0.5031	0.4165	1.1538
PFCM-kdtree		53.7317	0.3685	0.6543	0.2781	0.4515	0.3542	1.2412

The performance of an algorithm is one important aspect and the computational complexity is another important aspect of the algorithm. So the computational complexities of the proposed combination model FMV, Voting[9] and ALS[10] are also compared in terms of CPU times. The comparative results listed in Table 10 show that Voting[9] is of the cheapest computation and FMV is a little more expensive computation than ALS[10].

Table 10 Computation Complexity of Three Combination Methods

Data set	CPU time (second)		
	FMV	ALS	<i>voting</i>
<i>ionosphere</i>	0.110129	0.082110	0.026875
<i>svmguide3</i>	0.299652	0.113652	0.067559
<i>diabetes</i>	0.167901	0.137785	0.045441
Sat. image	3.086964	2.117051	0.790977

The computer system is of Genuine Intel® CPU 2140, 1.60GHz and 1.60GHz, 1GB memory. CPU time is only composed of the running time of FMV, ALS and *voting*, but not that of any individual fuzzy clustering algorithm.

VII CONCLUSIONS

This paper generalizes the traditional majority voting rule to the fuzzy majority voting rule and proposes a cluster matching algorithm, based on which a combination model of fuzzy partitions is developed. We

compare our combination method with other two combination methods – *voting* [9] and *ALS* [10] and individual fuzzy partitions. Comparative results show that our combination method outperforms *voting* [9] and *ALS* [10] in terms of all evaluation indexes used in this paper. The reason may be that both *voting* [9] and *ALS* [10] aim to find the consensus fuzzy partition that optimally represents and closely fits the set of component fuzzy partitions, and the optimal representation and fitting of a set of fuzzy partitions do not equal the optimal representation of the real structure of the data set, that is, if the consensus fuzzy partition optimally represents or fits a collection of fuzzy partitions, it does not guarantee to represent the real structure of the data set. We also find that *voting* [9] and *ALS* [10] are a little worse than some of the component individual fuzzy partitions, while FMV is at least comparable to the best one of the component individual fuzzy partitions in all cases. This confirms that FMV is able to foster strengths and circumvent weaknesses of component fuzzy partitions, while *voting* [9] and *ALS* [10] can not. In a word, FMV is not only superior to *voting* [9] and *ALS* [10], but also more stable and reliable than any individual fuzzy partition in some cases.

It is still important for the consensus fuzzy partition to be comparable to, but not sure to outperform, the best component fuzzy partition in any cases, for no fuzzy clustering algorithm can generate good partitions in all cases and we do not know which fuzzy clustering algorithm may produce a good clustering over a given data set in advance. Furthermore, when no information about the data set is available, it is hard to us accurately evaluate the fuzzy partition, much less pick out the best individual fuzzy partition. In this sense, the consensus fuzzy partition is more stable and reliable than any component individual fuzzy partition, for it is able to combine multiple fuzzy partitions into a consolidate one that is at least comparable to the best component fuzzy partitions in any cases.

Acknowledgements

The project is supported by the National Natural Science Foundation under Grant No. 60835004 and the Scientific Research Foundation for Doctor Grant No. 8451064101000630.

REFERENCES

- [1] C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press, 1981.
- [2] D. E. Gustafson, W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conf. Decision Contr.*, San Diego, CA, pp. 761–766, 1979.
- [3] Kuo-Lung Wu, Miin-Shen Yang, "Alternative C-means clustering algorithm," *Pattern Recognition*, vol.35, , pp.2267-2278, 2002.
- [4] Nikhil R. Pal, Kuhu Pal, James M. Keller, James C. Bezdek, "a possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. On Fuzzy systems*, vol.13(4), pp517-530, 2005.
- [5] Ana L.N. Fred, Anil K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation," *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 27(6), pp835 – 850, 2005.
- [6] Alexander Topchy, Anil K. Jain and William Punch, "Clustering ensembles: models of consensus and weak partitions," *IEEE Transactions on pattern analysis and machine intelligence*, VOL. 27(12), pp1866–1881, 2005.
- [7] Tianming Hu, Ying Yu, Jinzhi Xiong and Sam Yuan Sung, "Maximum likelihood combination of multiple clusterings," *Pattern Recognition Letters* 27, pp.1457–1464, 2006.
- [8] Alexander Topchy, Anil K. Jain, and William Punch, "Combining Multiple Weak Clusterings," *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*.
- [9] Evgenia Dimitriadou, Andreas Weingessel, and Kurt Hornik, "A combination scheme for fuzzy clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol.16(7), pp.901-912, 2002.
- [10] A.D. GORDON, M. VICHI, "Fuzzy partition models for fitting a set of partitions," *Psychometrika*, VOL. 66(2), pp.229—248, 2002.
- [11] G. Carpento, S. Maxtello, E. Toth, "Algorithms and codes for the assignment problem," In B. Simeone, E Toth, G. Gallo, E Maffioli, & S. Pallottino (Eds.), *Fortan codes for network optimization*. *Annals of Operations Research*, 13, pp.193-224, 1988.
- [12] S. Shehroz Khan, Amir Ahmad, "cluster center initialization algorithm for K-means clustering," *Pattern Recognition Letters* 25, pp.1293-1302, 2004.
- [13] S.J.,Redmond, C.,Heneghan, "A method for initializing the K-means clustering algorithm using kd-trees," *Pattern Recognition Letters*, vol.28, pp 965-973, 2007.
- [14] Yao-nan Wang, Chun-sheng Li, Yi Zuo, "A Selection Model for Optimal Fuzzy Clustering Algorithm and Number of Clusters Based on Competitive Comprehensive Fuzzy Evaluation," *IEEE Transaction On Fuzzy Systems*, vol 17(3), pp. 568-577, June 2009.
- [15] The UCI Machine Learning Repository, 1993, <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [16] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [17] R.J.G.B Campello, "A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment," *Pattern Recognition Letters* 28, pp.833-841, 2007.
- [18] E.Trauwert, , "On the meaning of Dunn's partition coefficient for fuzzy clusters," *Fuzzy Sets Systems* 25, pp.217–242, 1988.

Li Chun-sheng received the B.S and the Master degree in Mathematic from Jiang Xi Normal University in 1992 and the Central South University in 1999 respectively, China. He now is a Ph.D. at the College of Electrical and Information Engineering, Hunan University. His research interests are computational intelligence, intelligent control and intelligent information processing.