Rough–Fuzzy Collaborative Clustering

Sushmita Mitra, *Senior Member, IEEE*, Haider Banka, and Witold Pedrycz, *Fellow, IEEE*

***Abstract*—In this study, we introduce a novel clustering architecture, in which several subsets of patterns can be processed together with an objective of finding a common structure. The structure revealed at the global level is determined by exchanging prototypes of the subsets of data and by moving prototypes of the corresponding clusters toward each other. Thereby, the required communication links are established at the level of cluster prototypes and partition matrices, without hampering the security concerns. A detailed clustering algorithm is developed by integrating the advantages of both fuzzy sets and rough sets, and a measure of quantitative analysis of the experimental results is provided for synthetic and real-world data.**

***Index Terms*—Cluster validity, collaborative clustering, fuzzy membership, objective function-based clustering, rough sets.**

I. Introduction

A CLUSTER is a collection of data objects which are similar to one another within the same cluster but dissimilar to the objects in other clusters. The problem is to group N patterns into c possible clusters with high intraclass similarity and low interclass similarity by optimizing an objective function. In objective function-based clustering algorithms, the goal is to find a partition for a given value of c. The c-means algorithm [1] represents each cluster by its center of gravity.

Collaborative clustering deals with revealing a structure that is common or similar to a number of subsets [2]. For example, let us consider a population of data about client information, distributed over multiple databases. An intelligent approach to mine such large volume of information would be to analyze each individual database of subpopulation locally and subsequently combine (or collaborate on) the results at a globally abstract level. This also satisfies certain security (or privacy) concerns of clients in not allowing the sharing of individual data (or samples). In such situations, one may proceed by clustering each subpopulation locally as a module, considering small random samples, thereby enabling faster convergence of clustering [3]. Subsequently, there is collaboration between these modules by intercommunicating the individual cluster centroids. These representatives from the other subpopulations serve to globally influence and refine the clustering result of each module. Eventually, since the subpopulations are derived from the same large population, we converge to a stable global clustering after effective collaboration between the modules.

The use of soft computing in clustering has been reported in literature [4], [5]. Fuzzy sets and rough sets have been incorporated in the c-means framework to develop the fuzzy c-means (FCM) [6] and rough c-means (RCM) [7] algorithms. While membership in FCM enables efficient handling of overlapping partitions, the rough sets [8] deal with uncertainty, vagueness and incompleteness in data. Image segmentation has also been done using rough sets [9].

Rough sets are used to model clusters in terms of upper and lower approximations, which are weighted by a pair of parameters while computing cluster prototypes. We observe that RCM assigns objects into two distinct regions, viz., lower and upper approximations, such that objects in lower approximation indicate definite inclusion in the cluster while those in the upper approximation correspond to possible inclusion in it. Since there is no concept of membership involved, therefore any measure of closeness of patterns to the clusters cannot be determined.

Collaborative clustering was first investigated by Pedrycz [2], using standard FCM algorithm. This concept can be further extended to the rough domain using collaborative rough or collaborative rough–fuzzy clustering.

In this paper, we present a novel collaborative clustering through the use of rough–fuzzy sets. The use of rough sets help in automatically controlling the effect of uncertainty among patterns lying between the upper and lower approximations, during collaboration between the modules. Thereby, patterns within the lower approximation play a more pivotal role during clustering. Incorporation of membership, in the RCM framework, is seen to enhance the robustness of clustering as well as collaboration. The Davies–Bouldin (DB) and Dunn (D) indexes are extended to the rough and rough–fuzzy framework, and helps determine the optimal number of clusters during collaboration. Quantitative evaluation of the level of collaboration is developed in terms of membership grades.

The paper is organized into six sections. Section II provides the basic description of the c-means, FCM and RCM clustering algorithms. The rough–fuzzy c-means (RFCM) clustering algorithm along with the modified DB and D indexes are designed in Section III. Collaboration in the RCM and RFCM frameworks are also developed here. The quantitative evaluation of the collaboration is provided in Section IV. Experimental results are presented in Section V on synthetic and real data. Finally, Section VI concludes the paper. In this study, we use a standard notation as follows:

N	number of samples;
c	number of clusters;
U_i	ith cluster in partition U;
$\|c_i\|$	cardinality of cluster U_i;

Manuscript received June 5, 2005. This work was supported in part by the Council of Scientific and Industrial Research under Grant 22/0346/02/Exclusive Marketing Right (EMR)-II. This paper was recommended by Associate Editor A. F. Gomez Skarmeta.

S. Mitra and H. Banka are with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India (e-mail: sushmita@isical.ac.in; hbanka_r@isical.ac.in).

W. Pedrycz is with Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2G7, Canada (e-mail: pedrycz@ece.ualberta.ca).

Digital Object Identifier 10.1109/TSMCB.2005.863371

$\mathbf{v}_i$	ith prototype;
m	fuzzifier;
$\mathbf{x}_k$	kth sample or pattern;
d_{ik}	distance between $\mathbf{x}_k$ and $\mathbf{v}_i$;
u_{ik}	membership of $\mathbf{x}_k$ in U_i;
$\underline{\mathrm{B}}U_i$, $\overline{\mathrm{B}}U_i$	lower and upper approximations of U_i;
w_{low}, w_{up}	importance or weight of lower and upper approximations;
$S(U_i)$	within-cluster distance of U_i;
$d(U_i, U_j)$	between-cluster separation among U_i and U_j;
DB	Davies–Bouldin index;
P	total number of modules or partitions.

II. Clustering Algorithms

In this section, we describe the different partitive algorithms used for clustering, like c-means, fuzzy c-means, and rough c-means. Our objective is to contrast these algorithms while underlining the commonalities existing between them.

A. c-Means Clustering: Brief Overview

The algorithm proceeds by partitioning N objects into c nonempty subsets. During each iteration of clustering algorithm, the centroids or means of the clusters are computed. The main steps of the c-means algorithm [1] are as follows.

1) Assign initial means $\mathbf{v}_i$ (also called centroids).
2) Assign each data object (pattern) $\mathbf{x}_k$ to the cluster U_i for the closest mean.
3) Compute new mean for each cluster using

$$\mathbf{v}_i = \frac{\sum_{\mathbf{x}_k \in U_i} \mathbf{x}_k}{|c_i|} \tag{1}$$

where $|c_i|$ is the number of objects in cluster U_i.
4) Iterate Steps 2) and 3) **until** criterion function converges, i.e., there are no more new assignments of objects.

B. FCM

This is a fuzzification of the c-means algorithm, proposed by Bezdek [6]. It partitions a set of N patterns $\{\mathbf{x}_k\}$ into c clusters by minimizing the objective function $J = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m \|\mathbf{x}_k - \mathbf{v}_i\|^2$, where $1 \leq m < \infty$ is the fuzzifier, $\mathbf{v}_i$ is the ith cluster center, $u_{ik} \in [0, 1]$ is the membership of the kth pattern to it, and $\|.\|$ is the distance, such that

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} (u_{ik})^m \mathbf{x}_k}{\sum_{k=1}^{N} (u_{ik})^m} \tag{2}$$

and

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}} \tag{3}$$

$\forall i$, with $d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\|^2$, subject to $\sum_{i=1}^{c} u_{ik} = 1$, $\forall k$, and $0 < \sum_{k=1}^{N} u_{ik} < N$, $\forall i$. The algorithm proceeds as in c - means, along with the incorporation of membership.

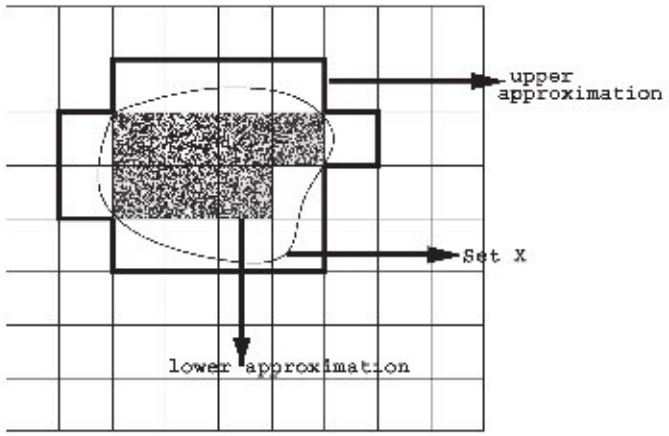

Fig. 1. Lower and upper approximations of a rough set.

C. RCM

The theory of rough sets [8] has recently emerged as another major mathematical tool for managing uncertainty that arises from granularity in the domain of discourse—that is, from the indiscernibility between objects in a set. The intention is to approximate a rough (imprecise) concept in the domain of discourse by a pair of exact concepts, called the lower and upper approximations. These exact concepts are determined by an indiscernibility relation in the domain, which, in turn, may be induced by a given set of attributes ascribed to the objects of the domain. The lower approximation is the set of objects definitely belonging to the vague concept, whereas the upper approximation is the set of objects possibly belonging to the same. Fig. 1 provides a schematic diagram of a rough set X within the upper and lower approximations, consisting of granules from the rectangular grid.

In the rough c-means algorithm, the concept of c-means is extended by viewing each cluster as an interval or rough set [7]. A rough set X is characterized by its lower and upper approximations $\underline{\mathrm{B}}X$ and $\overline{\mathrm{B}}X$, respectively, with the following properties.

1) An object $\mathbf{x}_k$ can be part of at most *one* lower approximation.
2) If $\mathbf{x}_k \in \underline{\mathrm{B}}X$ of cluster X, then simultaneously $\mathbf{x}_k \in \overline{\mathrm{B}}X$.
3) If $\mathbf{x}_k$ is not a part of any lower approximation, then it belongs to two or more upper approximations.

This permits overlaps between clusters. It is to be noted that these characteristics are not necessarily independent or complete. However, this restricted enumeration is helpful in understanding the rough set adaptation of RCM algorithm. Incorporating rough sets into c-means clustering requires the addition of the concept of lower and upper bounds.

Computation of the cluster prototypes is modified in the rough framework, by incorporating the concepts of upper and lower approximations. The right-hand side of (1) is split into two parts. Since the patterns lying in the lower approximation definitely belong to a rough cluster, they are assigned a higher weight that is controlled by parameter w_{low}. The patterns lying in the upper approximation are assigned a relatively lower weight, controlled by parameter w_{up} during computation. The centroid $\mathbf{v}_i$ of cluster U_i is evaluated as (4), shown at

the bottom of the page, where the parameters w_{low} and w_{up} correspond to the relative importance of the lower and upper approximations, respectively. Here, $|\underline{B}U_i|$ indicates the number of patterns in the lower approximation of cluster U_i, while $|\overline{B}U_i - \underline{B}U_i|$ is the number of patterns in the rough boundary lying between the two approximations.

RCM is found to generate three types of clusters, such as those having objects:

1) in both the lower and upper approximations;
2) only in lower approximation;
3) only in upper approximation.

Thereby, the three cases of (4) need to be considered while computing the cluster prototypes. When a cluster contains objects in both its lower and upper approximations, these are weighted by w_{low} and w_{up} (such that $w_{\text{low}} + w_{\text{up}} = 1$) depending on their importance during clustering. For example, w_{low} is high (or low) before (or during) collaboration. When a cluster contains objects only in its lower or in its upper approximation, the cluster prototype is computed in the classical manner without scaling down by w_{low} or w_{up}. This prohibits drifting of prototypes from their desired location. This explains the formulation of the prototype by RCM in the equation. Note that the computation of the new cluster prototype is weighted by w_{low} and w_{up} only when both its approximations are nonempty.

We now explain the condition under which an object may belong to the lower or upper bound of a cluster. Let $\mathbf{x}_k$ be an object at distance d_{ik} from centroid $\mathbf{v}_i$ of cluster U_i. The difference in distance $d_{ik} - d_{jk}$, $i \neq j$, can be used to determine whether $\mathbf{x}_k$ should belong to the lower or upper approximations of the clusters. The actual algorithm is outlined as follows.

1) Assign initial means $\mathbf{v}_i$ for the c clusters.
2) Assign each data object (pattern) $\mathbf{x}_k$ to the lower approximation $\underline{B}U_i$ or upper approximation $\overline{B}\mathrm{U}_i$, $\overline{B}\mathrm{U}_j$ of cluster pairs U_i and U_j by computing the difference in its distance $d_{ik} - d_{jk}$ from the cluster centroid pairs $\mathbf{v}_i$ and $\mathbf{v}_j$.
3) Let d_{ik} be minimum and d_{jk} be the next to minimum.
   If $d_{jk} - d_{ik}$ is less than some threshold,
   then $\mathbf{x}_k \in \overline{B}U_i$ and $\mathbf{x}_k \in \overline{B}U_j$ and $\mathbf{x}_k$ cannot be a member of any lower approximation [Property 3)],
   else $\mathbf{x}_k \in \underline{B}U_i$ such that distance d_{ik} is minimum over the c clusters [Property 2)].
4) Compute new mean for each cluster U_i using (4).
5) **Repeat** Steps 2)–4) **until** convergence, i.e., there are no more new assignments of objects.

The expression in (4) boils down to (1) when the lower approximation is equal to the upper approximation, implying an empty boundary region. It is observed that the performance of the algorithm is dependent on the choice of w_{low}, w_{up}, and threshold. We allowed $w_{\text{up}} = 1 - w_{\text{low}}$, $0.5 < w_{\text{low}} < 1$, and $0 <$ threshold < 0.5.

It is to be noted that the parameter threshold measures the relative distance of an object $\mathbf{x}_k$ from a pair of clusters having centroids $\mathbf{v}_i$ and $\mathbf{v}_j$. The larger the value of threshold, the more likely is $\mathbf{x}_k$ to lie within the rough boundary (between upper and lower approximations) of a cluster. This implies that only those points that definitely belong to a cluster (i.e., lie close to the centroid) occur within the lower approximation. A small value of threshold implies that more patterns are allowed to belong to any of the lower approximations.

The parameter w_{low} controls the importance of the objects lying within the lower approximation of a cluster in determining its centroid. A lower w_{low} implies a higher w_{up}, and hence an increased importance of patterns located in the rough boundary of a cluster towards the positioning of its centroid.

An optimal selection of these parameters is an issue of reasonable interest. Typically, experimental investigation is done for different combinations. Genetic algorithms (GAs) have been used for tuning the parameters threshold and w_{low}, while minimizing a fitness function based on clustering validity index, for generating an optimal number of clusters [10].

D. Clustering Validity Indexes

The clustering algorithms described in Sections II-A–C are partitive, requiring prespecification of the number of clusters. The results are dependent on the choice of c. There exist validity indexes to evaluate the goodness of clustering, corresponding to a given value of c. In this paper, we compute the optimal number of clusters c_0 in terms of the DB and D cluster validity indexes [11].

The DB is a function of the ratio of the sum of within-cluster distance to between-cluster separation. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_{|c_k|}\}$ be a set of patterns lying in a cluster U_k. Then, the average distance between objects within the cluster U_k is expressed as

$$S(U_k) = \frac{\sum_{i,i'} \|\mathbf{x}_i - \mathbf{x}_{i'}\|}{|c_k|\,(|c_k| - 1)} \tag{5}$$

where $\mathbf{x}_i, \mathbf{x}_{i'} \in U_k$, and $i \neq i'$. The between-cluster separation is defined as

$$d(U_k, U_l) = \frac{\sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|}{|c_k||c_l|} \tag{6}$$

$$\mathbf{v}_i = \begin{cases} w_{\text{low}} \dfrac{\sum_{\mathbf{x}_k \in \underline{B}U_i} \mathbf{x}_k}{|\underline{B}U_i|} + w_{\text{up}} \dfrac{\sum_{\mathbf{x}_k \in (\overline{B}U_i - \underline{B}U_i)} \mathbf{x}_k}{|\overline{B}U_i - \underline{B}U_i|}, & \text{if } \underline{B}U_i \neq \emptyset \wedge \overline{B}U_i - \underline{B}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in (\overline{B}U_i - \underline{B}U_i)} \mathbf{x}_k}{|\overline{B}U_i - \underline{B}U_i|}, & \text{if } \underline{B}U_i = \emptyset \wedge \overline{B}U_i - \underline{B}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in \underline{B}U_i} \mathbf{x}_k}{|\underline{B}U_i|}, & \text{otherwise} \end{cases} \tag{4}$$

where $\mathbf{x}_i \in U_k$, $\mathbf{x}_j \in U_l$, such that $k \neq l$. The optimal clustering, for $c = c_0$, minimizes

$$\text{DB} = \frac{1}{c}\sum_{i=1}^{c} \max_{j \neq i} \left\{ \frac{S(U_i) + S(U_j)}{d(U_i, U_j)} \right\} \tag{7}$$

for $1 \leq i, j \leq c$. Thereby, the within-cluster distance $S(U_i)$ is minimized while the between-cluster separation $d(U_i, U_j)$ gets maximized.

Like DB, the D [11] is designed to identify sets of clusters that are compact and separated. Here, we maximize

$$D = \min_{j} \left\{ \min_{i \neq j} \left\{ \frac{d(U_i, U_j)}{\max_k S(U_k)} \right\} \right\} \tag{8}$$

for $1 \leq i,\ j \leq c$. The intercluster separation is maximized, while minimizing intracluster distances. Note that the denominator of DB is analogous to the numerator of D.

III. Collaborative Clustering

In this section, we introduce the rough–fuzzy c-means algorithm. A collaborative rough c-means clustering is proposed, by incorporating collaboration between different partitions or subpopulations. This is then extended to the formulation of the collaborative rough–fuzzy c-means clustering. Rough and rough–fuzzy versions of the DB and D clustering validity indexes are developed for the purpose.

A. RFCM

A new rough–fuzzy c-means algorithm is proposed. This allows one to incorporate fuzzy membership value u_{ik} of a sample $\mathbf{x}_k$ to a cluster mean $\mathbf{v}_i$, relative to all other means $\mathbf{v}_j\ \forall j \neq i$, instead of the absolute individual distance d_{ik} from the centroid. This sort of relativistic measure, in terms of (2) and (3), enhances the robustness of the clustering with respect to different choices of parameters. The major steps of the algorithm are provided below.

1) Assign initial means $\mathbf{v}_i$ for the c clusters.
2) Compute u_{ik} by (3) for c clusters and N data objects.
3) Assign each data object (pattern) $\mathbf{x}_k$ to the lower approximation $\underline{\text{B}}U_i$ or upper approximation $\overline{\text{B}}U_i$, $\overline{\text{B}}U_j$ of cluster pairs U_i and U_j by computing the difference in its membership $u_{ik} - u_{jk}$ to cluster centroid pairs $\mathbf{v}_i$ and $\mathbf{v}_j$.
4) Let u_{ik} be maximum and u_{jk} be the next to maximum.
   If $u_{ik} - u_{jk}$ is less than some threshold,
   then $\mathbf{x}_k \in \overline{\text{B}}U_i$ and $\mathbf{x}_k \in \overline{\text{B}}U_j$ and $\mathbf{x}_k$ cannot be a member of any lower approximation,
   else $\mathbf{x}_k \in \underline{\text{B}}U_i$ such that membership u_{ik} is maximum over the c clusters.
5) Compute new mean for each cluster U_i, incorporating (2) and (3) into (4), as in (9), shown at the bottom of the page.
6) **Repeat** Steps 2)–5) **until** convergence, i.e., there are no more new assignments.

As indicated earlier, we use $w_{\text{up}} = 1 - w_{\text{low}}$, $0.5 < w_{\text{low}} < 1$, $m = 2$, and $0 < \text{threshold} < 0.5$.

B. DB and D Indexes

The validity indexes DB and D of (7) and (8) involve within-cluster distance $S(U_i)$ (distance of cluster prototype $\mathbf{v}_i$ from patterns $\mathbf{x}_k$ in the cluster) and between-cluster separation $d(U_i, U_j)$ (distance between prototypes $\mathbf{v}_i$ and $\mathbf{v}_j$) considering cluster pairs U_i and U_j.

1) Rough Version: The rough within-cluster distance is formulated as in (10), shown at the bottom of the page, using (4). Rough DB now becomes

$$\text{DB}_{\text{r}} = \frac{1}{c}\sum_{i=1}^{c} \max_{j \neq i} \left\{ \frac{S_{\text{r}}(U_i) + S_{\text{r}}(U_j)}{d(U_i, U_j)} \right\}. \tag{11}$$

$$\mathbf{v}_i = \begin{cases} w_{\text{low}} \dfrac{\sum_{\mathbf{x}_k \in \underline{\text{B}}U_i} u_{ik}^m \mathbf{x}_k}{\sum_{\mathbf{x}_k \in \underline{\text{B}}U_i} u_{ik}^m} + w_{\text{up}} \dfrac{\sum_{\mathbf{x}_k \in (\overline{\text{B}}U_i - \underline{\text{B}}U_i)} u_{ik}^m \mathbf{x}_k}{\sum_{\mathbf{x}_k \in (\overline{\text{B}}U_i - \underline{\text{B}}U_i)} u_{ik}^m}, & \text{if } \underline{\text{B}}U_i \neq \emptyset \wedge \overline{\text{B}}U_i - \underline{\text{B}}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in (\overline{\text{B}}U_i - \underline{\text{B}}U_i)} u_{ik}^m \mathbf{x}_k}{\sum_{\mathbf{x}_k \in (\overline{\text{B}}U_i - \underline{\text{B}}U_i)} u_{ik}^m}, & \text{if } \underline{\text{B}}U_i = \emptyset \wedge \overline{\text{B}}U_i - \underline{\text{B}}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in \underline{\text{B}}U_i} u_{ik}^m \mathbf{x}_k}{\sum_{\mathbf{x}_k \in \underline{\text{B}}U_i} u_{ik}^m}, & \text{otherwise} \end{cases} \tag{9}$$

$$S_{\text{r}}(U_i) = \begin{cases} w_{\text{low}} \dfrac{\sum_{\mathbf{x}_k \in \underline{\text{B}}U_i} \|\mathbf{x}_k - \mathbf{v}_i\|^2}{|\underline{\text{B}}U_i|} + w_{\text{up}} \dfrac{\sum_{\mathbf{x}_k \in (\overline{\text{B}}U_i - \underline{\text{B}}U_i)} \|\mathbf{x}_k - \mathbf{v}_i\|^2}{|\overline{\text{B}}U_i - \underline{\text{B}}U_i|}, & \text{if } \underline{\text{B}}U_i \neq \emptyset \wedge \overline{\text{B}}U_i - \underline{\text{B}}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in (\overline{\text{B}}U_i - \underline{\text{B}}U_i)} \|\mathbf{x}_k - \mathbf{v}_i\|^2}{|\overline{\text{B}}U_i - \underline{\text{B}}U_i|}, & \text{if } \underline{\text{B}}U_i = \emptyset \wedge \overline{\text{B}}U_i - \underline{\text{B}}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in \underline{\text{B}}U_i} \|\mathbf{x}_k - \mathbf{v}_i\|^2}{|\underline{\text{B}}U_i|}, & \text{otherwise} \end{cases} \tag{10}$$

Analogously, rough D can be expressed as

$$D_{\mathrm{r}} = \min_{j} \left\{ \min_{i \neq j} \left\{ \frac{d(U_i, U_j)}{\max_{k} S_{\mathrm{r}}(U_k)} \right\} \right\}. \tag{12}$$

2) Rough–Fuzzy Version: The rough–fuzzy within-cluster distance becomes (13), as shown at the bottom of the page, by employing (9). Rough–fuzzy DB is now expressed as

$$\mathrm{DB}_{\mathrm{rf}} = \frac{1}{c} \sum_{i=1}^{c} \max_{j \neq i} \left\{ \frac{S_{\mathrm{rf}}(U_i) + S_{\mathrm{rf}}(U_j)}{d(U_i, U_j)} \right\}. \tag{14}$$

The rough–fuzzy D becomes

$$D_{\mathrm{rf}} = \min_{j} \left\{ \min_{i \neq j} \left\{ \frac{d(U_i, U_j)}{\max_{k} S_{\mathrm{rf}}(U_k)} \right\} \right\}. \tag{15}$$

C. Collaborative RCM and RFCM

Let a dataset be divided into P subpopulations or modules. Each subpopulation is independently clustered to reveal its structure. Collaboration is incorporated by exchanging information between the modules regarding the local partitions, in terms of the collection of prototypes computed within the individual modules. This sort of divide-and-conquer strategy enables efficient mining of large databases. The required communication links are hence at a higher level of abstraction, thereby representing information granules (rough or rough–fuzzy clusters) in terms of their prototypes.

The higher the value of the threshold, the larger is the number of samples in the boundary regions of the rough–fuzzy clusters. This leads to a stronger collaboration between the prototypes of different modules, resulting in the movement of the prototypes of corresponding clusters (from different modules) towards each other. Often this is eventually followed by a merger of the corresponding prototypes, and hence clusters. This implies that the cluster prototypes from different modules influence and approach each other, due to the collaboration existing mainly in the overlapping (or boundary) regions of the corresponding clusters. The impact of the collaboration on the ensemble of modules is expressed in terms of the changes occurring in the prototypes of the individual clusters. Since the modules correspond to partitions from the same large dataset, this sort of collaborative clustering stabilizes the ensemble towards efficient determination of a globally existent structure.

There exist two phases in the algorithm.

1) Generation of RCM or RFCM clusters within the modules, without collaboration. Here, we employ $0.5 < w_{\mathrm{low}} < 1$, thereby providing more importance to samples lying within the lower approximation of clusters while computing their prototypes locally.
2) Collaborative RCM or RFCM between the clusters, computed locally for each module of the large dataset. Now, we use $0 < w_{\mathrm{low}} < 0.5$ (we chose $w_{\mathrm{low}} = 1 - w_{\mathrm{low}}$), with a lower value providing higher precedence to samples lying in the boundary region of the overlapping clusters.
   a) In collaborative RCM, a cluster U_i may be merged with an overlapping cluster U_j

   $$\textbf{if} \quad |\underline{\mathrm{B}}U_i| \leq |\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i| \tag{16}$$

   and $\mathbf{v}_i$ is closest to $\mathbf{v}_j$ in the feature space with $(|\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i| - |\underline{\mathrm{B}}U_i|)$ being the maximum among all overlapping clusters.
   b) In case of collaborative RFCM, U_i can be considered for merging with U_j

   $$\textbf{if} \quad \sum_{\mathbf{x}_k \in \underline{\mathrm{B}}U_i} u_{ik} \leq \sum_{\mathbf{x}_k \in (\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i)} u_{ik} \tag{17}$$

   and $\mathbf{v}_i$ is closest to $\mathbf{v}_j$ in the feature space with $(\sum_{\mathbf{x}_k \in (\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i)} u_{ik} - \sum_{\mathbf{x}_k \in \underline{\mathrm{B}}U_i} u_{ik})$ being the maximum among all overlapping clusters.

   Collaboration is done by exchanging cluster prototypes between modules, leading to a global determination of the overall structure within the data.

Let there be c_1 and c_2 clusters, generated by RCM or RFCM, in a pair of modules ($P = 2$) under consideration. During collaboration, we begin with $c_1 + c_2$ cluster prototypes and merge using (16) or (17), respectively.

The entire algorithm is summarized below.

1) Split the large dataset into P modules.
2) **For** each module $p = 1, \ldots, P$ **do**
   Generate c RCM or RFCM clusters (Sections II-C or III-A) with $0.5 < w_{\mathrm{low}} < 1$.
3) **For** each module p **do** collaboration.
   a) Accept $c * (P - 1)$ cluster prototypes from remaining $(P - 1)$ modules.

$$S_{\mathrm{rf}}(U_i) = \begin{cases} w_{\mathrm{low}} \dfrac{\sum_{\mathbf{x}_k \in \underline{\mathrm{B}}U_i} u_{ik}^m \|\mathbf{x}_k - \mathbf{v}_i\|^2}{\sum_{\mathbf{x}_k \in \underline{\mathrm{B}}U_i} u_{ik}^m} + w_{\mathrm{up}} \dfrac{\sum_{\mathbf{x}_k \in (\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i)} u_{ik}^m \|\mathbf{x}_k - \mathbf{v}_i\|^2}{\sum_{\mathbf{x}_k \in (\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i)} u_{ik}^m}, & \text{if } \underline{\mathrm{B}}U_i \neq \emptyset \wedge \overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in (\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i)} u_{ik}^m \|\mathbf{x}_k - \mathbf{v}_i\|^2}{\sum_{\mathbf{x}_k \in (\overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i)} u_{ik}^m}, & \text{if } \underline{\mathrm{B}}U_i = \emptyset \wedge \overline{\mathrm{B}}U_i - \underline{\mathrm{B}}U_i \neq \emptyset \\ \dfrac{\sum_{\mathbf{x}_k \in \underline{\mathrm{B}}U_i} u_{ik}^m \|\mathbf{x}_k - \mathbf{v}_i\|^2}{\sum_{\mathbf{x}_k \in \underline{\mathrm{B}}U_i} u_{ik}^m}, & \text{otherwise} \end{cases} \tag{13}$$

b) Assign each pattern $\mathbf{x}_k$ to lower or upper approximation of the $C(=c*P)$ collaborative RCM or RFCM clusters, with $0 < w_{\text{low}} < 0.5$.
c) Merge overlapping cluster pairs **while** (16) or (17) hold.
   i) Compute new prototype for merged clusters U_i and U_j as the mean of $\mathbf{v}_i$ and $\mathbf{v}_j$.
   ii) Reduce number of clusters C by one.
   iii) Reassign each pattern $\mathbf{x}_k$ to lower or upper approximation of the C collaborative RCM or RFCM clusters.
   iv) Calculate DB and D indexes, using (11), (12), (14), and (15).

IV. Quantification of Collaboration

The effect of the collaboration between clusters (or information granules) can be quantitatively evaluated in terms of a pair of measures δ and Δ [2]. In this section, we develop these measures to suit our rough–fuzzy framework. It is to be noted that here, unlike in [2], the number of clusters in a module need not remain fixed but is generalized to be different before and after collaboration.

Memberships of data objects are computed both before and after collaboration, with respect to the cluster prototypes. Since the collaboration here allows merging of overlapping clusters, the final cardinality of partitions within different modules are often different. Analogously, the number of partitions before and after collaboration, within a particular module p ($p = 1, \ldots, P$), can also be nonidentical. This lead us to use the maximum membership value $\max u_{ik}(p)$ of a data point $\mathbf{x}_k$ of module p, to one of the clusters U_i, during our computation of quantitative measures.

Moreover, a data sample may belong to different approximations (lower or upper) of a cluster, before and after collaboration both within as well as between different modules. Irrespective of whether a data object falls in the upper or lower approximation of a cluster at any stage, a higher membership is always indicative of a stronger belongingness to a cluster (likely in lower approximation) as compared to a lower membership value (likely in upper approximation). Hence, the concept of maximum membership elegantly subsumes all such complex possibilities.

The measure δ expresses how close the modules are upon collaboration. This is computed as

$$\delta = \frac{2}{NP(P-1)} \sum_{k=1}^{N} \left[\sum_{p_l, p_{l'}} \left| \max_{i|\mathbf{x}_k \in U_i} u_{ik}(p_l) - \max_{j|\mathbf{x}_k \in U_j} u_{jk}(p_{l'}) \right| \right]. \tag{18}$$

Here, U_i and U_j correspond to those clusters in modules p_l and $p_{l'}$, respectively, in which $\mathbf{x}_k$ has maximum membership. We have $l = 1, \ldots, P-1$ and $l' = l+1, \ldots, P$. Note that clusters U_i and U_j, coming from different modules, may or may not be identical to each other.

The larger the value of w_{low} before collaboration, the higher the value of w_{up} during collaboration (a lower w_{low} during collaboration, as explained in Section III-C). This leads to a stronger collaboration between modules. Similar is the case with larger values of the parameter threshold that encourages stronger collaboration. This results in a lower value of δ. Hence, a plot of δ with respect to this pair of variables provides useful information on the resultant collaboration within a dataset. For example, it quantifies how much a subset (or cluster) is susceptible to the collaborative impact coming from the other subsets of patterns.

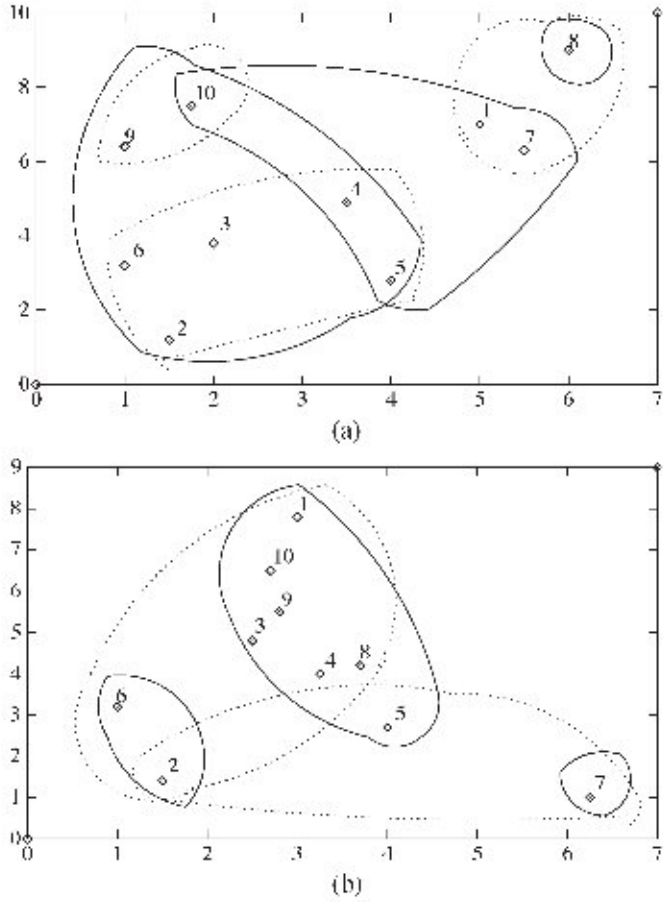

Fig. 2. Collaborative clustering on synthetic data 1 for (a) Module A and (b) Module B, with RFCM.

The second measure Δ considers the partitioning before collaboration as a reference point in the computations. Thereby, it quantifies the effect of collaboration upon the clustering. Hence, for module p, we have

$$\Delta_p = \frac{1}{N} \sum_{k=1}^{N} \left| \max_{i|\mathbf{x}_k \in U_i} u_{ik}(p_{\text{after\_collab}}) - \max_{j|\mathbf{x}_k \in U_j} u_{jk}(p_{\text{before\_collab}}) \right|. \tag{19}$$

Note that the clusters U_i and U_j, coming after and before collaboration for the same module, may or may not be the same. Finally

$$\Delta = \sum_{p=1}^{P} \Delta_p. \tag{20}$$

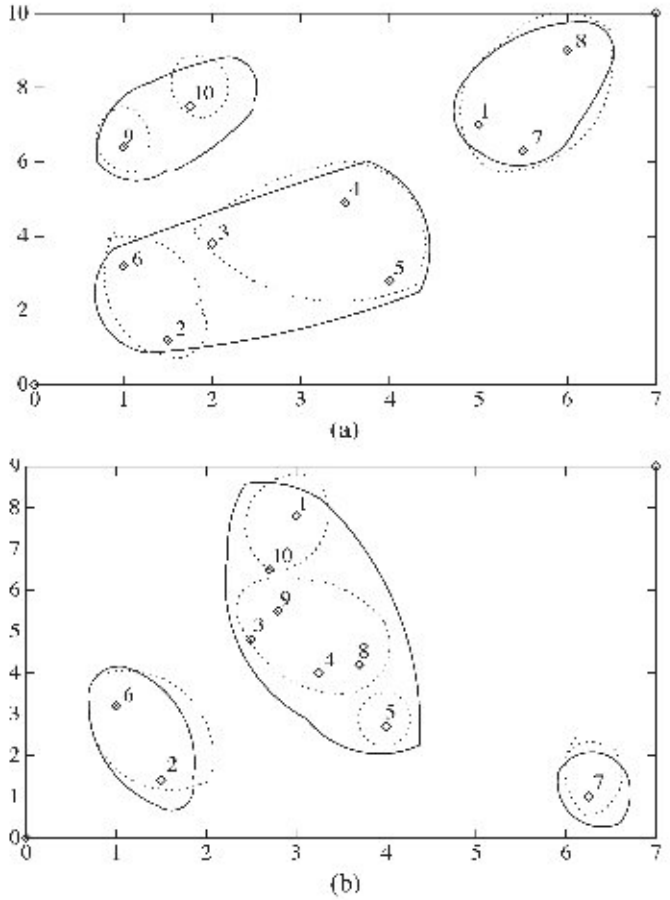

Fig. 3. Collaborative clustering on synthetic data 1 for (a) Module A and (b) Module B, with RCM.

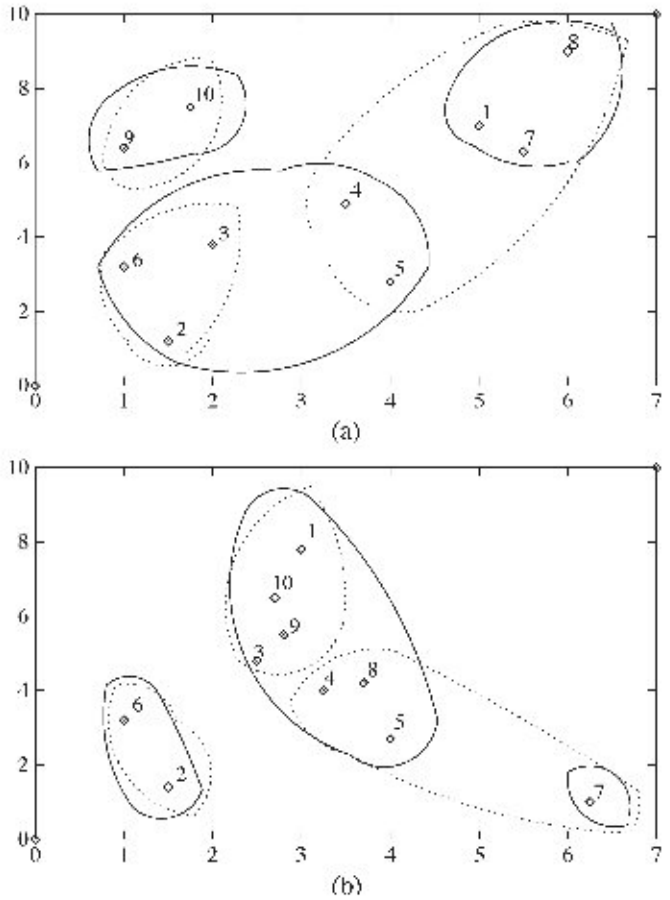

Fig. 4. Collaborative clustering on synthetic data 1 for (a) Module A and (b) Module B, with FCM.

This sum provides an overall indicator of the departure, upon collaboration, from the referential structure before collaboration. A smooth surface of the Δ-plot, in the threshold–w_{low} space, also implies a balanced collaborative effect on the patterns of the clusters.

V. Results

Results are provided on a small synthetic dataset, followed by the benchmark Iris data, and a highly nonseparable two-class dataset. The distance is typically represented in terms of the traditional Euclidean metric for numeric features.

A. Synthetic Data 1

The two-dimensional synthetic dataset [2] is shown in Figs. 2–4. There are two modules (A and B) corresponding to ten samples each, partitioned into three clusters. Sample results using threshold $= 0.2$, $w_{\text{low}} = 0.9$ for RFCM and RCM, and $w_{\text{low}} = 1 - 0.9 = 0.1$ for collaborative RFCM and RCM, are provided in Table I.

Fig. 2 indicates the clustering before (solid line) and after (dotted line) collaboration, using modules A and B, respectively, with RFCM. Analogous results are depicted for RCM and FCM, in Figs. 3 and 4, respectively. Note that there exists no membership in case of RCM, such that $u_{ik} \in \{0, 1\}$.

The quantitative measures δ and Δ (of Section IV) are depicted in Fig. 5 after collaboration using RFCM. The graphs are plotted for different values of threshold and w_{low} ($0 < w_{\text{low}} < 0.5$, during collaboration). This demonstrates how the collaborating modules influence each other. On the other hand, a lower value of Δ pertains to a decreased adaptation within a cluster, in a module, during collaboration. This implies reduced impact of collaboration from outside on a data subset.

B. Iris Data

Iris data is a typical benchmark, consisting of 150 samples of three categories of the iris flower. There are 50 samples in each class, which are expressed in terms of the four features viz., sepal length, sepal width, petal length, and petal width. We partition the dataset into two modules A and B of 75 samples each, to demonstrate collaborative clustering. Table II lists sample results for collaborative clustering, using threshold $= 0.1$ and $w_{\text{low}} = 0.9$. Results are reported for both clustering validity indexes, viz., DB and D.

It is observed that minimum values for the DB of (14) is generated for both the modules in case of collaborative RFCM, indicating an optimal clustering upon collaboration. The number of patterns being too large to be individually indicated in the table, we provide counts of those lying in the lower approximation and boundary. Note that the value of DB_{r} and DB_{rf}

TABLE I
SAMPLE COMPARATIVE PERFORMANCE OF COLLABORATIVE CLUSTERING ON SYNTHETIC DATA 1

Algorithm/ Module	Before collaboration			After collaboration		
	Prototypes	Samples (membership)		Prototypes	Samples (membership)	
		lower	boundary		lower	boundary
RFCM/A	(5.07, 6.45) (6.0, 9.0) (1.62, 3.28)	1(.85),7(1.0) 8(1.0) 2(.87),3(.9),6(1.0), 9(.55)	4(.52),5(.34),10(.39) 4(.34),5(.54),10(.32)	(1.5, 7.14) (5.73, 8.17) (2.1, 3.14)	9(.69),10(.98) 1(.55),7(.51),8(1.0) 2(.76),3(.98),4(.59), 5(.69),6(.93)	–
RFCM/B	(6.25, 1.0) (3.22, 4.75) (1.16,2.61)	7(1.0) 1(.56),3(.7),4(.95), 5(.65),8(1.0),9(.73), 10(.62) 2(.7),6(1.0)	–	(1.83, 1.47) (1.64, 1.76)	5(.96),7(.73) 1(.67),3(.85),4(.99), 6(.63),8(.96),9(.81) 10(.73)	2(.55) 2(.45)
RCM/A	(3.35, 6.42) (6.0, 9.0) (2.13, 2.75)	1, 4, 7, 9, 10 8 2, 3, 5, 6	–	(1.75, 7.5) (5.5, 7.43) (3.17, 3.83) (1.0, 6.4) (1.25, 2.2)	10 1, 7, 8 3, 4, 5 9 2, 6	– –
RCM/B	(6.25, 1.0) (3.14,6.07) (1.25, 2.3)	7 1, 3, 4, 5, 8, 9, 10 2, 6	–	(2.85, 7.15) (4.0, 2.7) (6.25, 1.0) (3.06, 4.63) (1.25, 2.3)	1, 10 5 7 3, 4, 8, 9 2, 6	– –

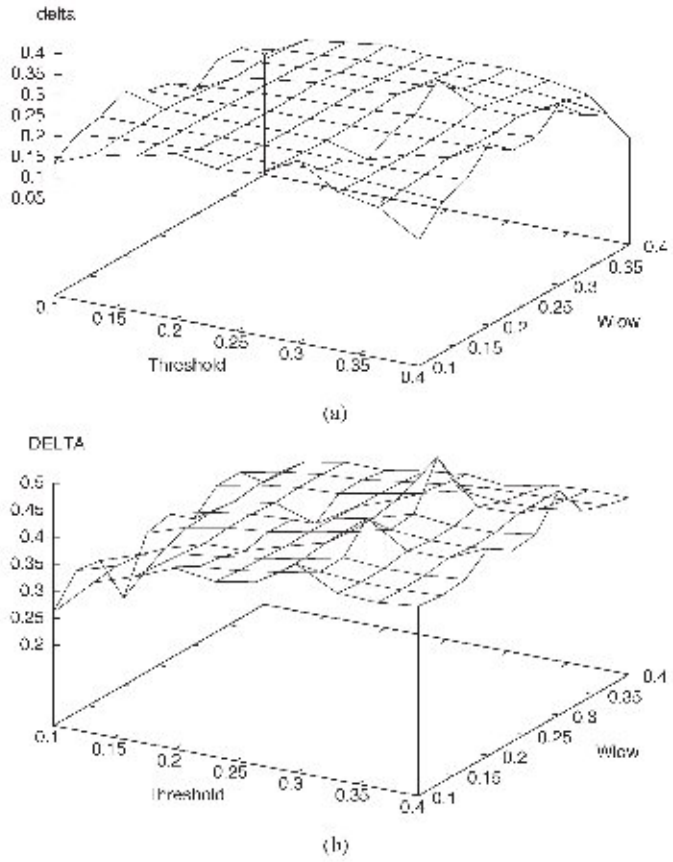

Fig. 5. Quantification of collaboration for two modules, using RFCM on synthetic dataset 1 with (a) δ (delta) and (b) Δ (DELTA).

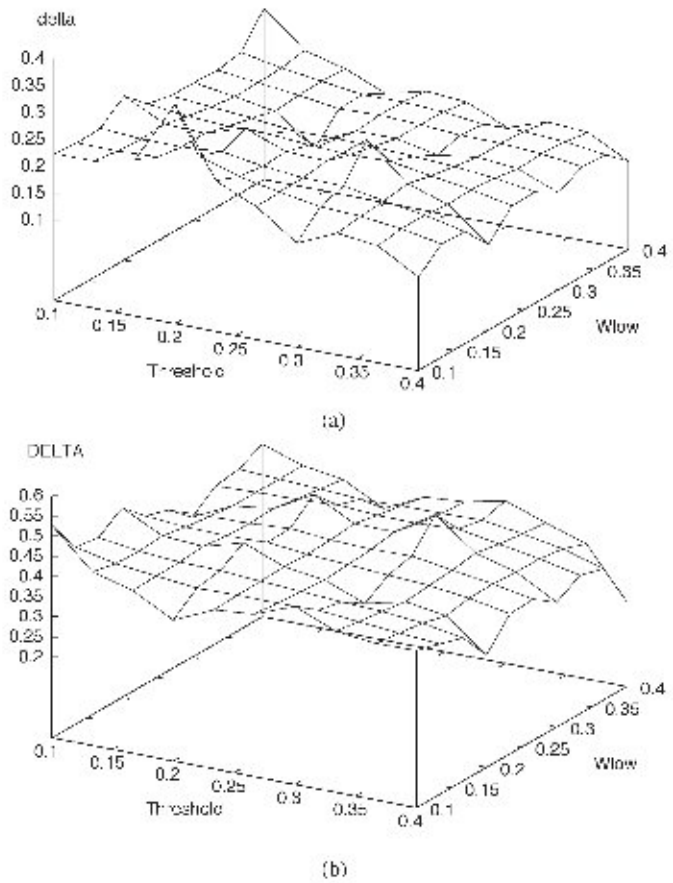

Fig. 6. Quantification of collaboration for two modules, using RFCM on Iris with (a) δ (delta) and (b) Δ (DELTA).

are also found to decrease for both modules after collaboration. This implies a resultant minimization in roughness of the partitions. Although module B contained a number of points in the cluster boundaries before collaboration, ultimately it is found to converge to a more definite (less ambiguous) partitioning with zero samples in the cluster boundaries. The absence of the membership concept leads to poorer results for RCM.

Fig. 6 illustrates the quantitative measures δ and Δ after collaboration using RFCM. It is observed that δ decreases with increasing values of threshold. This is indicative of increased collaboration between clusters. Simultaneously, a lower value of w_{low} (during collaboration) is associated with higher collaboration. The slope of the δ-plot in Fig. 6(a) ascertains this. Analogously, a higher value of Δ implies stronger

TABLE II
SAMPLE COMPARATIVE PERFORMANCE OF COLLABORATIVE CLUSTERING ON IRIS DATA

Algorithm/ Module	Before collaboration				After collaboration			
	Prototypes	No. of samples		DB (D)	Prototypes	No. of samples		DB (D)
		lower	boundary			lower	boundary	
	(7.04, 3.22, 5.99, 2.12)	19	0		(6.40, 2.90, 5.13, 1.73)	54	0	
RFCM/A	(5.01, 3.40, 1.64, 0.33)	20	1	0.63	(5.00, 3.48, 1.51, 0.26)	21	0	0.36
	(5.93, 2.68, 4.39, 1.42)	35	1	(2.63)	–	–	–	(0.36)
	(5.48, 3.36, 1.91, 0.412)	2	48		(5.24, 2.48, 3.38, 1.00)	2	0	
RFCM/B	(4.77, 3.18, 1.47, 0.24)	23	2	1.97	(6.35, 2.98, 4.99, 1.73)	42	0	0.53
	(5.21, 3.72, 2.18, 0.52)	2	46	(0.83)	(4.87, 3.28, 1.53, 0.25)	31	0	(0.22)
	(6.98, 3.22, 5.95, 2.13)	19	0		(5.00, 3.50, 1.48, 0.25)	20	0	
RCM/A	(5.01, 3.40, 1.63, 0.33)	20	1	0.68	(6.02, 2.70, 4.57, 1.46)	36	0	0.64
	(5.95, 2.68, 4.45, 1.44)	35	1	(2.39)	(6.98, 3.22, 5.95, 2.13)	19	0	(2.51)
	(6.15, 2.90, 4.74, 1.61)	43	1		(6.10, 2.84, 4.60, 1.51)	13	12	
RCM/B	(4.50, 3.44, 1.46, 0.33)	5	1	0.67	(6.23, 2.88, 4.75, 1.58)	19	12	6.57
	(4.95, 3.42, 1.57, 0.26)	26	0	(0.99)	(4.87, 3.26, 1.52, 0.26)	31	0	(0.15)

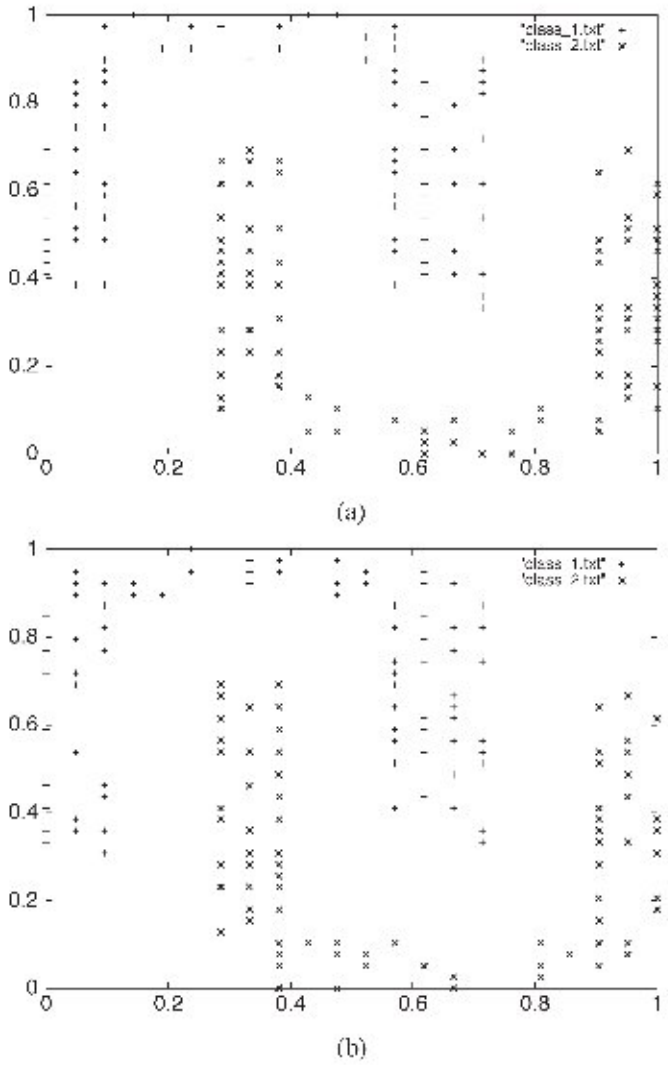

Fig. 7. Dataset Pat for (a) Module A and (b) Module B.

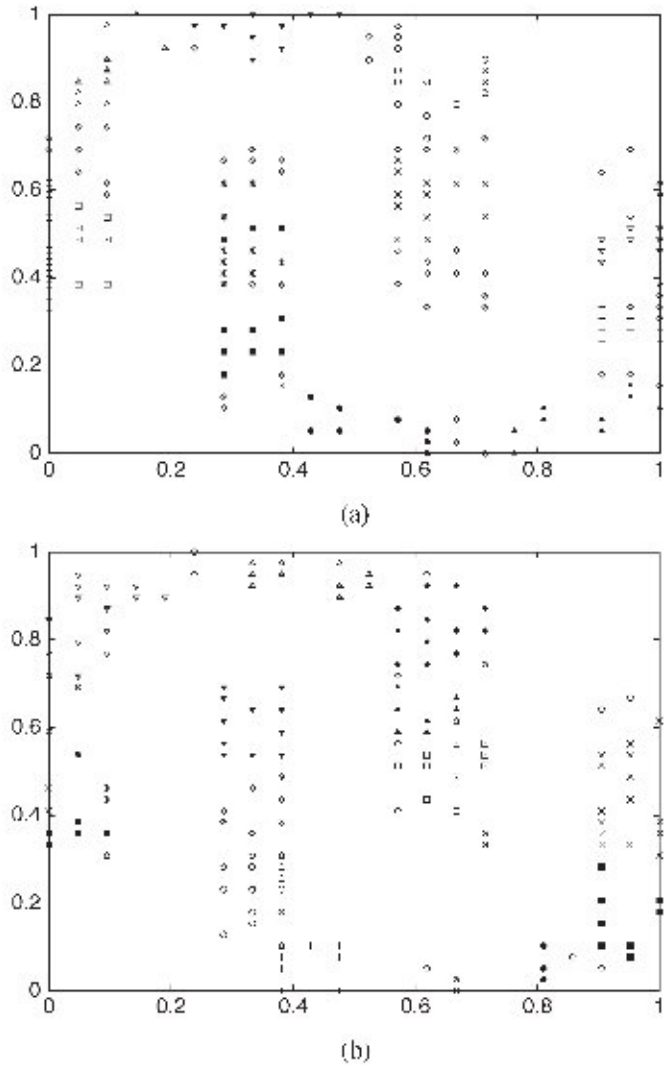

Fig. 8. Dataset Pat after collaborative RFCM clustering on (a) Module A and (b) Module B.

collaboration, and is data dependent. The slope of the Δ-plot in Fig. 6(b) validates this.

C. Synthetic Data 2

The synthetic data Pat consists of 420 patterns in the two-dimensional space. It consists of two linearly nonseparable pattern classes (C_1) and (C_2), represented in the form of an interleaved pair of horseshoes.

First, we partition the dataset into two modules with 210 samples each. This is depicted in Fig. 7, for modules A and B, respectively. Since the Pat data is highly nonseparable, we choose to partition it into ten initial clusters for both modules. Upon collaboration with 20 cluster prototypes, accompanied by

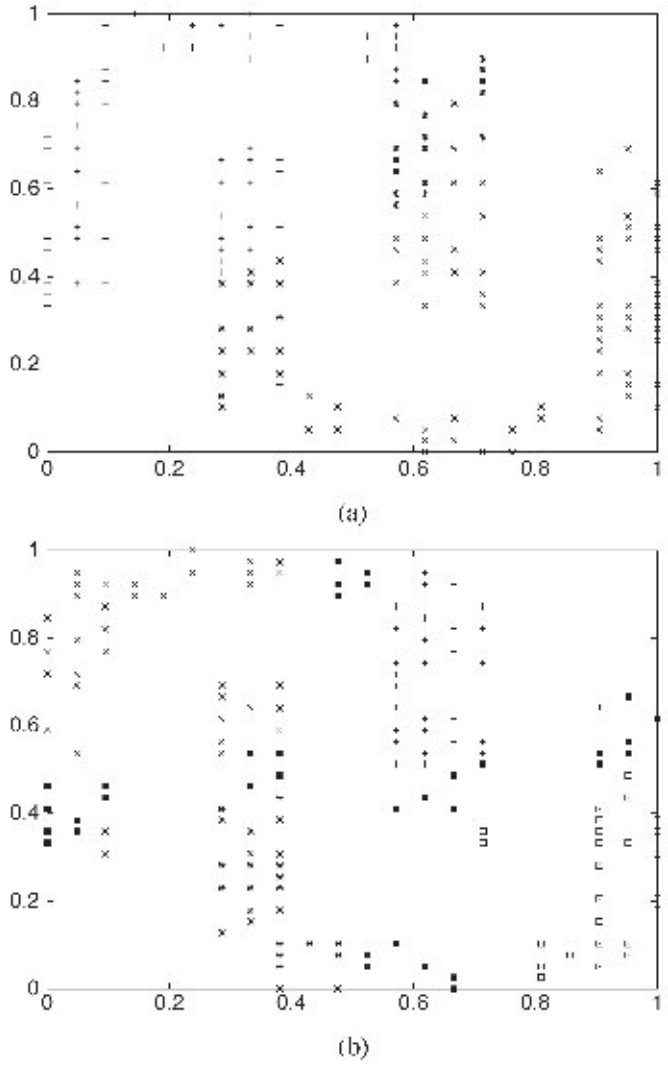

Fig. 9. Dataset Pat after collaborative RCM clustering on (a) Module A and (b) Module B.

merging, the resultant number of partitions reduced to 11 and 13 for RFCM, and 2 and 4 for RCM. Figs. 8 and 9 depict sample clustering outputs generated by the collaborative versions of RFCM and RCM. Different clusters are indicated in the figure using different symbols.

Next, the number of modules was increased to three, with three clusters each. Collaboration was done with nine initial cluster prototypes. The quantitative measures δ and Δ are depicted in Fig. 10, after collaboration using RFCM with $P = 3$. It is observed that both plots indicate a smoother surface over here. This is because a larger number of modules are able to introduce greater collaboration, and hence larger uniformity among partitions. However, this is at the expense of increased computational complexity for evaluating δ.

VI. Conclusion

Collaborative clustering is a promising approach towards modeling agent-based systems. A multiagent system is one in which a number of agents cooperate and interact with each other in a complex and distributed environment, thereby

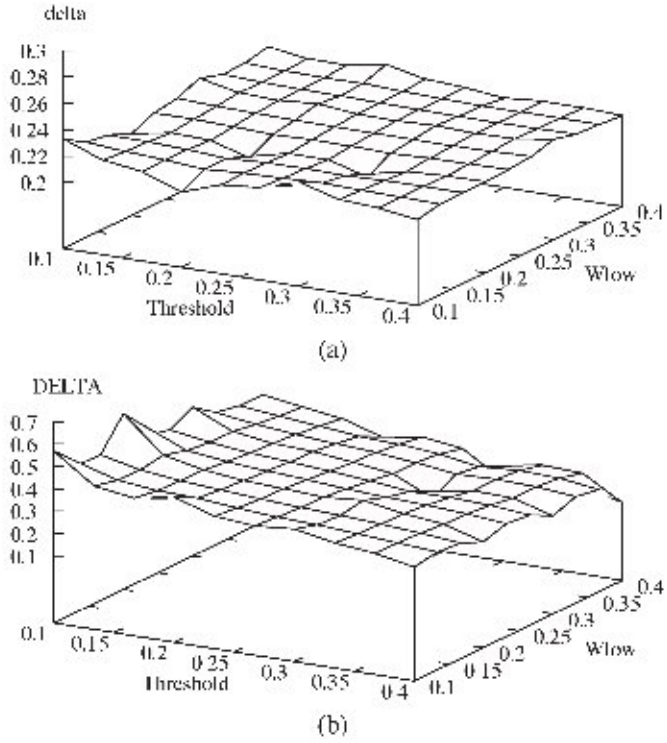

Fig. 10. Quantification of collaboration, for three modules, using RFCM on Pat with (a) δ (delta) and (b) Δ (DELTA).

achieving a global objective based on distributed data and control [12]. While handling large data in this framework, each intelligent agent may concentrate on information discovery (or clustering) within a module. Subsequently, these agents can communicate with each other at the cluster interface, using appropriate protocol, their cluster profiles represented in terms of the centroids. Thereby, an agent can refine the partitioning within its own module by collaborating with the other agents.

We have presented an RFCM clustering algorithm by incorporating membership in the RCM framework. Novel collaborative clustering is developed using the RCM and RFCM algorithms. The DB clustering validity index is suitably modified, by incorporating rough concepts, to determine optimal clustering during collaboration. Quantitative evaluation of the level of collaboration has been completed in terms of rough–fuzzy membership function.

Particle swarn optimization (PSO) is a biologically motivated scheme, which holds ample promise in modeling distributed agents. We are currently investigating the use of PSO in collaborating clustering. Moreover, finding suitable applications of this novel approach is also challenging and requires further research; specially for handling heterogeneous data and/or cardinality of the data subsets under consideration.

References

[1] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. London, U.K.: Addison-Wesley, 1974.

[2] W. Pedrycz, "Collaborative fuzzy clustering," *Pattern Recognit. Lett.*, vol. 23, no. 14, pp. 1675–1686, Dec. 2002.

[3] T. W. Cheng, D. B. Goldgof, and L. O. Hall, "Fast fuzzy clustering," *Fuzzy Sets Syst.*, vol. 93, no. 1, pp. 49–56, Jan. 1998.

[4] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Commun. ACM*, vol. 37, no. 3, pp. 77–84, Mar. 1994.

[5] S. Mitra and T. Acharya, *Data Mining: Multimedia, Soft Computing, and Bioinformatics*. New York: Wiley, 2003.
[6] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
[7] P. Lingras and C. West, "Interval set clustering of web users with rough k-means," Dept. Math.Comput. Sci., St. Mary's Univ., Halifax, NS, Canada, Tech. Rep. No. 2002-002, 2002.
[8] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning About Data*. Dordrecht, The Netherlands: Kluwer, 1991.
[9] S. K. Pal and P. Mitra, "Multispectral image segmentation using the rough-set-initialized EM algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2495–2501, Nov. 2002.
[10] S. Mitra, "An evolutionary rough partitive clustering," *Pattern Recognit. Lett.*, vol. 25, no. 12, pp. 1439–1449, Sep. 2004.
[11] J. C. Bezdek and N. R. Pal, "Some new indexes for cluster validity," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 3, pp. 301–315, Jun. 1998.
[12] J. Ferber, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. New York: Addison-Wesley, 1999.

Sushmita Mitra (M'99–SM'01) received the B.Tech. and M.Tech. degrees in computer science from the University of Calcutta, Kolkata, India, and the Ph.D. degree in computer science from the Indian Statistical Institute, Kolkata.

She is currently a Professor at the Machine Intelligence Unit, Indian Statistical Institute. From 1992 to 1994, she was in the Rheinisch-Westfällischen Technischen Hochschule (RWTH), Aachen, Germany as a Deutscher Akademischer Austauschdienst (DAAD) Fellow. She was a Visiting Professor in the Computer Science Departments of the University of Alberta, Edmonton, AB, Canada in 2004, Meiji University, Japan in 1999, 2004, and 2005, and in Aalborg University Esbjerg, Denmark in 2002 and 2003. She is the author of the books *Neuro–Fuzzy Pattern Recognition: Methods in Soft Computing* (Wiley, 1999) and *Data Mining: Multimedia, Soft Computing, and Bioinformatics* (Wiley, 2003). She has guest edited special issues of journals, and is an Associate Editor of *Neurocomputing*. She has more than 100 research publications in referred international journals. According to the Science Citation Index (SCI), two of her papers have been ranked 3rd and 15th in the list of top-cited papers in engineering science from India during 1992–2001. Her current research interests include data mining, pattern recognition, soft computing, image processing, and bioinformatics.

Dr. Mitra served in the capacity of Program Chair, Tutorial Chair, and as member of program committees of many international conferences. She received the National Talent Search Scholarship in 1978–1983 from the National Council of Educational Research and Training, India, the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award in 1994 for her pioneering work in neuro-fuzzy computing, and the CIMPA–INRIA–UNESCO Fellowship in 1996.

Haider Banka received the M.Sc. and M.Tech. degrees in computer science from the University of Calcutta, Kolkata, India, in 2001 and 2003, respectively.

During 2003 to 2004, he was a Lecturer in the Engineering College, Durgapur, India. Since 2004, he has been a Senior Research Fellow at the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. He has a number of publications in international journals and conferences. He has also served as a reviewer of several international journals. His current research interests include pattern recognition, data mining, soft computing, combinatorial optimization, machine learning, and bioinformatics.

Witold Pedrycz (M'88–SM'94–F'99) received the M.Sc., Ph.D., and D.Sci. degrees from the Silesian University of Technology, Gliwice, Poland.

He is currently a Professor and a Canada Research Chair (CRC) in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is also with the Systems Research Institute of the Polish Academy of Sciences. He is actively pursuing research in computational intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control including fuzzy controllers, pattern recognition, knowledge-based neural networks, relational computation, bioinformatics, and software engineering. He has published numerous papers in this area. He is also an author of nine research monographs covering various aspects of computational intelligence and software engineering.

Dr. Pedrycz has been a member of numerous program committees of conferences in the area of fuzzy sets and neurocomputing. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, the IEEE TRANSACTIONS ON NEURAL NETWORKS, and the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He is an Editor-in-Chief of Information Sciences, President of International Fuzzy Systems Association (IFSA) and North American Fuzzy Information Processing Society (NAFIPS).