This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2017.2729501, IEEE Transactions on Fuzzy Systems

Ensemble Fuzzy Clustering using Cumulative Aggregation on Random Projections

Punit Rathore, *Member, IEEE*, James C. Bezdek, *Life Fellow, IEEE,* Sarah M. Erfani, Sutharshan Rajasegarar and Marimuthu Palaniswami, *Fellow, IEEE*

***Abstract*—Random projection is a popular method for dimensionality reduction due to its simplicity and efficiency. In the past few years, random projection and fuzzy c-means based cluster ensemble approaches have been developed for high dimensional data clustering. However, they require large amounts of space for storing a big affinity matrix, and incur large computation time while clustering in this affinity matrix. In this paper, we propose a new random projection, fuzzy c-means based cluster ensemble framework for high dimensional data. Our framework uses cumulative agreement to aggregate fuzzy partitions. Fuzzy partitions of random projections are ranked using external and internal cluster validity indices. The best partition in the ranked queue is the core (or base) partition. Remaining partitions then provide cumulative inputs to the core, thus arriving at a consensus best overall partition built from the ensemble. Experimental results with Gaussian mixture datasets and a variety of real datasets demonstrate that our approach outperforms three state-of-the-art methods in terms of accuracy and space-time complexity. Our algorithm runs one to two orders of magnitude faster than other state-of-the-arts algorithms.**

***Index Terms*—High Dimensional Data, Fuzzy Clustering, Random Projection, Ensemble Clustering, Cumulative Agreement.**

I. Introduction

Clustering is an essential method of exploratory data analysis in which data are partitioned into several subsets such that objects in each subset are similar to each other, and dissimilar to members of other subsets. Clustering is an underlying tool for knowledge discovery [1], outlier/anomaly detection [2]–[5], indexing [6], and compression [7]. With the rapid advancement of *Internet of Things* (IoT) technologies, mobile computing, smart mobile devices, and social network services, data are growing at very fast rates. Many biomedical applications such as physiological monitoring, imaging, and sequencing [8] produce large amounts of high-dimensional data [9]. This article is about clustering algorithms that can be used for such large, high dimensional datasets.

High dimensional feature vector data, i.e., data described by a large number of attributes, poses two challenges for clustering. First, the so-called 'curse of dimensionality', which is caused by the lack of a sufficient number of samples in most high dimensional data, makes it difficult to find statistically meaningful structures in the data [10]. Second, noisy and irrelevant attributes in the data can worsen the performance of a clustering algorithm. One possible solution to improve the utility of clustering algorithms for high dimensional data is to perform dimensionality reduction [11]. Feature subset selection [12] and feature transformations to lower dimensional spaces are two well known methods for dimensionality reduction. Popular algorithms for feature extraction, such as *Principal Component Analysis* (PCA) [13] and *Singular Value Decomposition* (SVD) [14], use well-defined criteria to optimize the projection in lower dimensional space. Unlike these algorithms, random projection [15]–[17] is a relatively simple, computationally efficient linear transformation method which does not use any special criteria to find "optimal" lower dimensional projections. Two key properties, namely low computational complexity and (approximate) distance preservation in lower dimension subspaces, make random projection [16] an attractive choice for dimensionality reduction.

Over the past few years, ensemble clustering has drawn significant attention in addressing the clustering problem. Random projection based ensemble frameworks [18]–[21] have been proposed for high-dimensional clustering using fuzzy or probabilistic clustering algorithms. These approaches use random projection to generate multiple subsets into a lower dimension from the original dataset, and then some method of integration is used across the soft clustering results obtained on all projected datasets. Among these random projection based fuzzy clustering approaches, the most recent approaches [20], [21] require less memory and run faster than earlier approaches [18], [19]. However, the ensemble algorithms developed in [20], [21] still require very large amounts of space for storing a big affinity matrix; moreover, they take a lot of time to cluster the affinity matrix.

Generating and combining multiple output partitions from clustering has been done in several ways [22]–[28]. However, most of the existing merging algorithms suffer from time and/or space complexity problems. Among these approaches, agreement (voting) based merging [25]–[28], is the most popular and relatively computationally efficient approach. To the best of our knowledge, none of the algorithms based on merging cluster ensembles using the agreement approach have been studied for large and high-dimensional datasets.

In this paper, we propose a new, simple and efficient random projection based ensemble framework using a cumulative agreement scheme to aggregate multiple fuzzy membership

Punit Rathore, and Marimuthu Palaniswami are with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria, Australia.
E-mail: {prathore.student, palani}@unimelb.edu.au.
James C. Bezdek, and Sarah M. Erfani are with the School of Computing and Information Systems, The University of Melbourne, Victoria, Australia.
E-mail: {jbezdek, sarah.erfani}@unimelb.edu.au.
Sutharshan Rajasegarar is with the School of Information Technology, Deakin University, Geelong, Victoria, Australia.
E-mail: sutharshan.rajasegarar@deakin.edu.au.

matrices based on their quality. *Cluster Validity Indices* (CVIs) are used to determine the quality of consensus partitions. This framework eliminates the need of a final time-consuming clustering step such as the ones reported in [19]–[21] to obtain output partitions. Our aggregation method employs an agreement based approach [27], [28], which, to our knowledge, has been previously studied for only crisp partitions. Our algorithm extends this idea to the soft case for effective aggregation of fuzzy partitions, which are obtained using the *Fuzzy c-Means* (FCM) clustering algorithm [29] on randomly projected datasets. The ensemble approach used in our framework combines fuzzy partitions in a sequential manner, thus avoiding the complexity required by simultaneous aggregation of the suite of fuzzy partitions produced by clustering many random projections of the high dimensional data. Our method, which we call *Cumulative Agreement* FCM (CAFCM), scales linearly in the number of data points and the number of repetitions, making our random projection based ensemble approach feasible for large and high dimensional datasets. We evaluate the performance of our proposed framework on two synthetic and six real high dimensional datasets to demonstrate its superiority and robustness over three state-of-the-art approaches.

Here is an outline of the rest of this article. Section II presents preliminaries on fuzzy and crisp partitions and random projection methods. Section III presents a review of related work. Our agreement based aggregation model is discussed in Section IV. Section V describes the use of CVIs in our framework to achieve the best performance. Section VI presents the proposed framework for *Cumulative Agreement Fuzzy c-Means* (CAFCM) for ensemble fuzzy clustering which uses random projection and cumulative agreement. Section VII discusses the numerical experiments and results, followed by the conclusions and discussion in Section VIII.

II. Preliminaries

In this section, we introduce our notation for crisp and soft partitions and present the random projection method.

A. Matrix Representation for Fuzzy and Crisp Partitions

Consider a set of n objects $O = \{o_1, o_2, ..., o_n\}$, where each object is defined by a set of features in the form of $X = \{x_1, x_2, ..., x_n\} \subset \mathbb{R}^p$. The non-degenerate (no zero rows corresponding to empty clusters), soft (fuzzy/probabilistic) and crisp c-partitions of n objects are matrices, denoted as:

$$\begin{aligned} M_{fcn} = \{U \in \mathbb{R}^{c\times n} | \forall\ i \in \{1,c\},\ j \in \{1,n\} : u_{ij} \in [0,1]; \\ \sum_{i=1}^{c} u_{ij} = 1 \forall j; \sum_{j=1}^{n} u_{ij} > 0. \forall i.\}; \end{aligned} \tag{1a}$$

$$M_{hcn} = \{U \in M_{fcn} | u_{ij} \in \{0,1\} \forall i,j\}, \tag{1b}$$

where u_{ij} represents the membership of data point j in cluster i for fuzzy clustering. If the clustering is probabilistic, the value $u_{ij} = p_{ij}$ of data point j is the posterior probability that, given point j, it came from class i. Soft partitions are more flexible than crisp partitions in that each object can have membership in more than one cluster. In this paper, FCM is used to generate soft partitions in random projections of X. However, our ensemble approach for high dimensional data clustering is equally applicable to probabilistic clustering algorithms such as the *Gaussian Mixture Model* (GMM) [30], implemented with the *Expectation-Maximization* (EM) [31] algorithm.

B. Random Projection

A *random projection* (RP) is a linear transformation from $\mathbb{R}^p$ to $\mathbb{R}^q$, represented by a matrix T. Let $X = \{x_1, x_2, ..., x_n\} \subset \mathbb{R}^p$ be a set of n points in p dimensions, denoted as the "upspace". X can be mapped to a reduced dimension dataset $Y = \{y_1, y_2, ..., y_n\} \subset \mathbb{R}^q, q \ll p$, denoted as the "downspace", by the linear transformation of X with T. Most random projection methods are based on the *Johnson-Lindenstrauss* (JL) lemma [32]. It is not clear from [15]–[17] which random projection function T is best for clustering, so we will use a variant of the JL lemma proposed by Achlioptas in [16]. The theorem proved by Achlioptas is as follows:

Theorem 1: Let matrix $X \subset \mathbb{R}^{n\times p}$ be a dataset of n points and p attributes. Given $\varepsilon > 0$, and $\beta > 0$, for any integer q

$$q \geq q_0 = \frac{(4+2\beta)log(n)}{\varepsilon^2/2 - \varepsilon^3/3}. \tag{2}$$

The parameter ε controls the accuracy in distance preservation, while β controls the probability that distance preservation to within $1 \pm \varepsilon$ is achieved. Let T be a $p \times q$ random matrix, in which each element $t_{i,j}$ is drawn from one of the following independently identically distributed distributions:

$$t_{i,j} = \begin{cases} +1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases} \tag{3}$$

$$t_{i,j} = \begin{cases} +\sqrt{3} & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -\sqrt{3} & \text{with probability } 1/6. \end{cases} \tag{4}$$

Let $Y = \frac{1}{\sqrt{q}} XT$ be the projection matrix of the n points in $\mathbb{R}^q$. Let $f : \mathbb{R}^p \to \mathbb{R}^q$ map the i^{th} row of X to the i^{th} row of Y. Then for any $u, v \in X$ with probability at least $1 - n^{-\beta}$, we have

$$(1-\varepsilon)||u-v||^2 \leq |f(u) - f(v)||^2 \leq (1+\varepsilon)||u-v||^2.$$

According to Theorem 1, if the reduced (downspace) dimension q is equal or bigger than the JL lower bound q_0, then pairwise Euclidean distance squares are preserved within a multiplicative factor of $1 \pm \varepsilon$, and we say that Y has *JL certificate*. An older version of this projection operator is based on randomly choosing each element of T from a Gaussian distribution with zero mean and unit variance which carries a similar guarantee [16], [33]. However, the authors in [34] assert that the JL bound often holds for $q \ll q_0$. They called such projections "rogue random projections". We will study the use of rogue random projections in our ensemble clustering approach.

III. Related Work

In this section, we review existing random projection based cluster ensemble methods for high dimensional data clustering and agreement based combination schemes.

A. Random Projection Based Ensemble Approaches

Several ensemble approaches have been proposed for high dimensional data clustering, which are based on random projection and fuzzy c-means. The main idea of the existing approaches is as follows; First, multiple downspace datasets $\{Y_r\}_{r=1}^{N}$ are generated in a fixed lower dimension $\mathbb{R}^q$ using RP, where N is the number of RPs. Then, FCM clustering is performed on each downspace copy to obtain N fuzzy partitions, e.g., $U_r = \text{FCM}\ (Y_r)$, where $U_r \in M_{fcn}$. These output partitions $\{U_r\}_{r=1}^{N}$ are aggregated using an ensemble scheme. The final output partition is typically obtained by performing soft clustering on the rows of an aggregated matrix.

Apparently, the first cluster ensemble approach that used random projection was proposed in [19], in which GMM/EM clustering was used to obtain probabilistic partitions $P \in M_{fcn}$, where $p(c|i,\theta)$ is the probability of point i being in cluster c under a model θ. Subsequently, a similarity matrix M_i was computed between two joint probability distributions for each downspace dataset. The final similarity matrix M was obtained by averaging the M_is, and then the final clustering output was obtained by applying a hierarchical clustering algorithm, called *complete linkage* (CL), on the aggregated similarity matrix M.

A similar approach using FCM for fuzzy clustering (EFCM) was used in [18] to find the significant genes in DNA micro-array data. Random projection was used to reduce the data dimensionality. Then, FCM clustering algorithm was employed on each downspace dataset to generate membership matrices $U_r \in M_{fcn}$. Then for each r, a similarity matrix M_r was computed as $M_r = U_r^T U_r \subset \mathbb{R}^{n\times n}$. Then, an aggregated similarity matrix (M) was calculated by averaging the N M_rs across multiple projection runs. The distance matrix $D = 1 - M$ was computed, and then FCM was performed on the rows of $D \subset \mathbb{R}^{n\times n}$ to obtain a final membership matrix.

Both of the above approaches have space complexity $O(n^2)$ for storing the similarity matrix (M). There is a time complexity of $O(n^2 log(n))$ in applying complete linkage (GMM/EM based approach) and $O(dlnc^2)$ in applying FCM (the EFCM approach) on $D \subset \mathbb{R}^{n\times n}$, where n is number of data points, d is the dimensions of the matrix on which clustering is applied (for EFCM approach, $d = n$), c is the number of clusters, and l is the number of iterations used by FCM. There is an additional time complexity of $O(cNn^2)$ in the EFCM approach due to computing the product of the N partition matrices and their transposes. Therefore, both of these algorithms are limited to applications for which the number of objects n is small (e.g., some thousands of samples), and the original dimension p of the upspace data is large (e.g., more than tens of thousands). As n increases, the EFCM approach becomes intractable for big data.

To address the limitations of these two approaches for big data clustering, Popescu *et al.* [20] proposed a new method, RPFCM-A, that began with FCM clustering of random projections of the data. The resultant membership matrices $\{U_r\}_{r=1}^{N}$ were concatenated as $U_{con} = [U_1^T|U_2^T|....|U_N^T]$, and the final membership partition was obtained by applying FCM to the rows of the aggregated matrix $U_{con} \subset \mathbb{R}^{n\times cN}$. Concatenating N partitions of $n \times c$ dimension by stacking them along the element dimension results in an $n \times cN$ matrix which is significantly smaller than M_r. This approach eliminates the time complexity spent computing products of the membership matrices and their transposes. Thus, it seems more suitable than the EFCM based approach. However, it still requires the multiplication of the concatenated matrix with its transpose when a crisp output partition is desired. Moreover, this scheme has time complexity of $O(dlnc^2)$ when applying FCM to the concatenated matrix $U_{con} \subset \mathbb{R}^{n\times cN}$, where $d = cN$. If the number of clusters c in the data and the number of downspace datasets N are such that $cN > p$; it means the dimension of the agreement matrix becomes higher than the original dimension of dataset, which makes this approach unsuitable for high dimensional data clustering.

Mao *et al.* [21] proposed a modified approach, RPFCM-B, based on spectral graph partitioning. Instead of considering the full agreement matrix U_{con}, they performed the clustering on the first c left singular vectors of $\hat{U}_{con}$, where $\hat{U}_{con} = SVD(U_{con}) \subset \mathbb{R}^{n\times c}$, which reduces the computational time as compared to RPFCM-A approach. However, there is space complexity of $O(cnN)$, and computational complexity of $O(n(cN)^2)$ for SVD and $O(dlnc^2)$ for the FCM clustering, where $d = c$.

B. Agreement Based Combination Schemes

Among existing ensemble approaches, agreement based merging algorithms are popular due to their simplicity and computational efficiency. The idea of the agreement based combination scheme for fuzzy clustering was first introduced by Dimitriadou *et al.* [27], which is based on minimizing the average squared distance between ensemble membership partitions and an output optimal partition. This algorithm computes an approximate solution in a sequential manner, in which, the best cluster label permutation is obtained for each ensemble partition with respect to a reference partition, followed by updating the reference partition through averaging. However, the determination of the best cluster label for each cluster in a partition for large values of c is a time consuming task due to the computation of squared distances between partitions across each possible permutation of cluster labels. The labelling correspondence problem is solved in [26] using a maximum-likelihood estimate found with the Hungarian method [35], and then plurality voting is applied to obtain an optimal partition. The Hungarian algorithm can be costly because it is $O(c^3)$. The most recent work on consensus clustering employs a voting based mechanism [28], where the cluster label assignment problem is addressed using a contingency matrix which requires less computation time than that required by previous methods. The study in [28] was limited to crisp partitions. This scheme may not enjoy the same performance for soft partitions, which are obtained from projected datasets

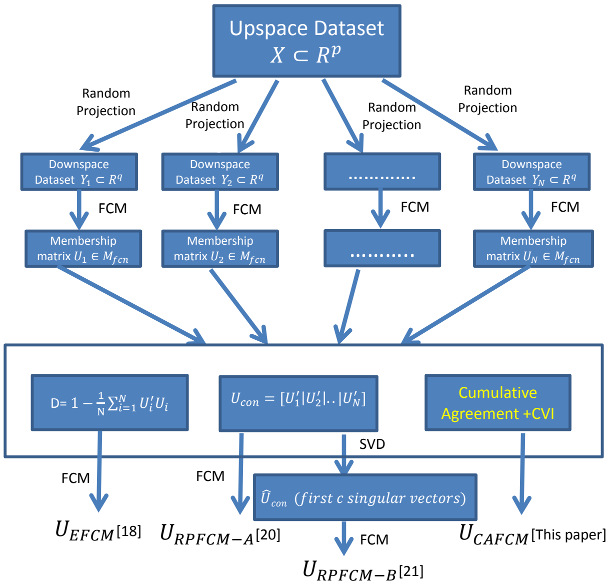

Upspace Dataset
$X \subset R^p$
Random Projection
Random Projection
Random Projection
Random Projection
Downspace Dataset $Y_1 \subset R^q$
Downspace Dataset $Y_2 \subset R^q$
....
Downspace Dataset $Y_N \subset R^q$
FCM
FCM
FCM
FCM
Membership matrix $U_1 \in M_{fcn}$
Membership matrix $U_2 \in M_{fcn}$
....
Membership matrix $U_N \in M_{fcn}$
D= $1 - \frac{1}{N}\sum_{i=1}^{N} U_i' U_i$
$U_{con} = [U_1'|U_2'|..|U_N']$
Cumulative Agreement +CVI
SVD
FCM
FCM
$\hat{U}_{con}$ (first c singular vectors)
U_{EFCM}[18]
$U_{RPFCM-A}$[20]
FCM
U_{CAFCM}[This paper]
$U_{RPFCM-B}$[21]

Fig. 1: Four methods of ensemble FCM clustering using random projection

using random projection. This is because random projection produces highly unstable and radically different outputs [19], [33].

Although a fair amount of work has been done on agreement based aggregation schemes, only a few schemes are applicable to soft clustering. In our work, we eliminate the use of FCM clustering on the aggregated matrix to get a final output partition, using an agreement based aggregation scheme which is computationally efficient and easy to implement. Fig. 1 compares the three FCM based schemes in [18], [20] and [21] to our proposed CAFCM method.

In the next section, we discuss our agreement based scheme for aggregating the fuzzy partitions $\{U_r\}_{r=1}^{N}$, obtained from FCM clustering on N randomly projected datasets.

IV. Agreement based Aggregation Model

The objective of an aggregation model is to find a partition U_f, which represents a set of N fuzzy partitions $\{U_r\}_{r=1}^{N}$, the representation being optimal in some well-defined sense. We assume that U_f and the U_r are all the same size $(c \times n)$. Let $u_i^{(r)}$ and $u_i^{(f)}$ be the label vectors of data point x_i for the partitions U_r and U_f, respectively. That is, $u_i^{(r)}$ is the i-th column of U_r, and similarly for $u_i^{(f)}$. The average dissimilarity function $h(U_r, U_f)$ is chosen as an optimality criteria, and can be expressed as the average squared distance between the N columns of U_r and U_f, as [27]

$$h(U_r, U_f) = \frac{1}{n}\sum_{i=1}^{n} ||(u_i^{(r)} - u_i^{(f)})||^2. \tag{5}$$

The computation in equation (5) measures the similarity between U_r and (the unknown solution) U_f on the assumption that the c clusters in U_r and U_f are "aligned", i.e., the rows of U_r and U_f represent the clusters in the same order. This is the so-called "registration problem" in clustering, and care must be taken to ensure that all of the partitions being aggregated are aligned in this sense. This problem is exacerbated when the partitions are fuzzy. We want to relabel the N U_r's so that they are aligned. This ensures that they will be aligned with the unknown U_f.

One way to approach this problem is to let $\Pi_b(U_r)$ represent the mapping of partition U_r to an optimally relabelled partition $U_{r,b}$ with respect to some base (or core) partition U_b. Then, an optimal partition can be obtained as the solution to [27],

$$U_f = \underset{U_b \in M_{fcn}}{\arg\min} \left(\frac{1}{N}\sum_{r=1}^{N} h(\Pi_b(U_r), U_b) \right). \tag{6}$$

The solution of this minimization problem in [27] gives $u_i^{(f)}$ as the arithmetic mean of $u_i^{(r)}$ over all partitions. In order to obtain the best cluster label permutation for each ensemble partition, the squared distance (minimization) between the ensemble and base partitions was chosen as mapping $\Pi_b(U_r)$. A contingency weight matrix based mapping scheme was proposed in [28] as a solution of (6). These solutions are not effective in combining multiple fuzzy partitions which are obtained using random projections. Our experiments with this method did not show very promising results. So, we turned to another approach, which effectively combines fuzzy partitions, obtained using RPs, based on their quality, as measured by cluster validity indices.

The concept behind agreement based ensemble approach is that pairs of points that stick together (appear in the same cluster) in most or all of the individual partitions should also stick together in the final ensemble partition. Suppose the number of clusters c_r for individual partitions U_r is randomly selected within some range $\{c_{min}, c_{max}\}$. The intuition underlying our approach is that the pairs of points that are members of a cluster for higher values of c should be considered to be more strongly associated to each other than pairs of point which are together in a cluster at a smaller value of c.

The N partitions obtained by applying FCM clustering to N random projections will have different information content (quality). The best quality partition, which has maximum information content about the cluster labels distribution, is chosen as the base partition, U_b, in the first step of the aggregation. Assuming that we do not have any prior knowledge for the selection of the base partition and the "true" number of clusters, we use an internal cluster validity index (CVI) to choose the base partition (discussed in the next section).

The remaining $N-1$ partitions are ranked in decreasing order of quality based on their relationship to the base partition, and are combined sequentially based on their rank. The objective of this scheme is to secure the strongest agreement between the highest ranked partitions in the queue with the base partition. In this way, low-quality partitions will have minimal effect on the quality of the overall output partition. Minor variations in ranking are not expected to impact the performance of this scheme, because using an ordered sequence based on decreasing quality effectively integrates the good and bad fuzzy partitions, and decreases the effects of bad partitions on the overall output. If the base partition is of poor quality or there is major variation in ranking (for example, a few poor-quality partitions are in the top five partitions in the CVI queue), then we expect performance to deteriorate. At the other extreme, if all N partitions are of roughly the same quality, then the selection of the base partition and ranking of the remaining partitions will not have a significant effect on the output partition.

In the next section, we discuss the use of CVIs to achieve the best performance for CAFCM.

V. Quality of Consensus Partitions

The projected datasets can be drastically different from each other due to the random mapping from upspace to downspace. Consequently, clustering on these different downspace datasets with any algorithm may result in output partitions of different quality. To determine the quality of partitions, we use a *cluster validity index* (CVI). A CVI is a measure of cluster quality that can be used to identify the "best" member amongst a set of multiple partitions (where best means, with respect to the CVI in use). External CVIs require ground truth information, whereas internal CVIs use only data and/or algorithmic outputs. See [36]–[40] and Table X for a detailed analysis and discussion on various internal and external CVIs.

The quality of the output partition U_f constructed by CAFCM depends on the quality of the base partition U_b, which is chosen in the initialization phase. The fuzzy partition from the set $\{U_r\}_{r=1}^N$, which best preserves the structure of the ground truth partition of labeled data will be taken as the base partition. The intuition behind using the best member from the set of ensemble partitions as the base partition is that the output partition U_f should contain the maximum amount of information about structure in the data that is present in the best quality partition amongst all ensemble partitions. Most importantly, this will eventually lead us to a method for identifying U_b for the unlabeled data case.

The quality of individual fuzzy partitions compared to a ground truth (labeled data) partition can be determined using a soft external CVI. Let the quality of any partition U_r with respect to the ground truth partition U_{gt}, using an external soft CVI $\mathscr{V}_{ext_s}$, be denoted as $\mathscr{V}_{ext_s}(U_r|U_{gt})$, where subsubscript "$s$" means soft. Based on the optimality of $\mathscr{V}_{ext_s}(U_r|U_{gt})$, the N ensemble partitions can be ranked in descending order of quality such that

$$\mathscr{V}_{ext_s}(U_{(1)}|U_{gt}) \geq \mathscr{V}_{ext_s}(U_{(2)}|U_{gt}) \geq ... \geq \mathscr{V}_{ext_s}(U_{(N)}|U_{gt}), \quad (7)$$

where parenthetical subscripts indicate the permutation of the original indices that results in the ordering shown in (7), and we assume without loss of generality that the CVI is max-optimal (best is maximum). This gives a set of sorted partitions $\mathbb{U}_{sorted}^{(ext_s)}$ based on their quality with respect to the external CVI $\mathscr{V}_{ext_s}$. In real-world applications, the data is unlabeled so the ground truth information, which is required to evaluate partition quality based on (7), is not available. In this case, a question that must be answered is: can internal CVIs ($\mathscr{V}_{int_s}$) be used to achieve similar rankings for a set of partitions $\mathbb{U}_{sorted}^{(int_s)}$? Internal/external (I/E) matching analysis is discussed in Section VII to determine whether the same base partition and similar ranking of the sorted partitions, suggested by an external CVI, can be obtained using internal CVIs.

Assuming that similar sets of partitions $\mathbb{U}_{sorted}^{(int_s)} = \mathbb{U}_{sorted}^{(ext_s)}$ can be obtained using an internal CVI, the best quality partition for unlabeled data, $U_{(1)}$ from $\mathbb{U}_{sorted}^{(int_s)}$, can be chosen as the base partition U_b. Using the base partition in Algorithm 1, chosen by this criterion, results in an output partition U_f, which is an aggregation of the ensemble of inputs that is optimal with respect to the chosen CVI. This minimizes the average dissimilarity between ensemble matrices and the best quality partition, which best preserves apparent cluster structure or information about X. Next, we discuss the proposed framework, CAFCM.

VI. Cumulative Agreement FCM (CAFCM) Algorithm

Suppose we have a set of ensemble partitions $\mathbb{U}_{sorted} = \{U_{(r)}\}_{r=1}^N$, each partition having c_r clusters, ranked according to (7) in decreasing order of their quality with respect to a specified CVI. Let the best (first) partition $U_{(1)}$ in $\mathbb{U}_{sorted}$ have c clusters and take $U_{(1)} = U_b$. The partitions $\{U_{(r)}\}_{r=2}^N$ are designated as voting partitions with respect to U_b. The entries of each column vector of stochastic matrix $U_{(r)} \in M_{fc_rn}$ represent the degree of membership of that object in each cluster (rows), and sum to 1, whereas, in the Moore-Penrose pseudoinverse $U_{(r)}^{-1} \in M_{fnc_r}$, each column vector turns into the row (cluster) vector $\{c_i\}_{i=1}^{c_r}$ whose entries sum to 1 [41]. These values can be interpreted as the weight of each data point (rows) in cluster (columns) vector c_i. Multiplying the pseudoinverse of $U_{(r)}$ with base partition U_b gives the weight matrix $W_{r,b} \subset \mathbb{R}^{c \times c_r}$,

$$W_{r,b} = U_b U_{(r)}^{-1}. \quad (8)$$

Due to the pseudoinverse $U_{(r)}^{-1}$ in the weight matrix calculation, the entries in $W_{r,b}$ do not lie in the range [0,1]. The relabelling of partition $U_{(r)}$ against the base partition U_b is achieved by multiplying $U_{(r)}$ with this weight matrix $W_{r,b}$, which gives the transformed partition $U_{r,b}$ as

$$U_{r,b} = W_{r,b} U_{(r)}. \quad (9)$$

The degrees of membership in the transformed partition $U_{r,b}$ correspond to degrees of memberships in $U_{(r)}$, which are scaled by the entries of $W_{r,b}$. This accomplishes the vote by $U_{(r)}$ to the base partition U_b. The ensemble approach in [28], that computes the weight matrix W [1] as

$$W = U_b U_{(r)}^T, \quad (10)$$

is a special case of approach (8) (suitable for fuzzy partitions).

Both approaches are demonstrated in Example 1 with a base partition U_b and an ensemble partition $U_{(r)}$. The mutual information between the transformed and the base partition is measured using the soft *Normalized mutual information* index (NMI) $\mathscr{V}_{NMI_s}$ [37] . It can be inferred from the NMI values in Example 1 that $U_{r,b}$ contains more mutual information with respect to the base partition U_b, than U (obtained using (10) and (9)).

[1] The columns of weight matrix, W, are normalized in [28] such that $w_{ij} \in [0,1]$, and $\sum_{j=1}^{c_r} w_{ij} = 1$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2017.2729501, IEEE Transactions on Fuzzy Systems

***Example** 1:* Consider a fuzzy base partition U_b of size 3×4 and an ensemble fuzzy partition $U_{(r)}$ of size 2×4, as given below:

$$U_b = \begin{bmatrix} 0.8 & 0.9 & 0.0 & 0.1 \\ 0.1 & 0.1 & 0.9 & 0.1 \\ 0.1 & 0.0 & 0.1 & 0.8 \end{bmatrix}, U_{(r)} = \begin{bmatrix} 0.6 & 0.7 & 0.1 & 0.1 \\ 0.4 & 0.3 & 0.9 & 0.9 \end{bmatrix}$$

The weight matrix $W_{r,b}$, computed using (8), and the matrix W, computed with (10), are as follows:

$$W_{r,b} = \begin{bmatrix} 1.35 & -0.09 \\ -0.15 & 0.57 \\ -0.20 & 0.52 \end{bmatrix}, W = \begin{bmatrix} 0.74 & 0.27 \\ 0.15 & 0.39 \\ 0.11 & 0.34 \end{bmatrix},$$

which gives the corresponding transformed partitions $U_{r,b}$ and U, using (9), as:

$$U_{r,b} = \begin{bmatrix} .78 & .92 & .05 & .05 \\ .14 & .06 & .50 & .50 \\ .08 & .02 & .45 & .45 \end{bmatrix}, U = \begin{bmatrix} .56 & .60 & .32 & .32 \\ .24 & .22 & .36 & .36 \\ .20 & .18 & .32 & .32 \end{bmatrix},$$

$$\mathscr{V}_{NMI_s}(U_{r,b}|U_b)\text{=}0.2178, \qquad \mathscr{V}_{NMI_s}(U|U_b)\text{=}0.0217.$$

When multiplying the partition $U_{(r)}$ with weight matrix $W_{r,b}$, each row vector $\{c_i\}_{i=1}^{c_r}$ of $U_{(r)}$ votes for each of the clusters $\{c_j\}_{j=1}^{c}$ of U_b, with weights w_{ij} from the cumulative vote weight matrix $W_{r,b}$. In the general case, each partition $U_{(r)}$ from $\mathbb{U}_{sorted}$, casts its vote with U_b this way in decreasing order of their quality in a sequential manner. Following [27], the base partition $U_b^{(i)}$ at iteration i is calculated by averaging the last base partition $U_b^{(i-1)}$ with transformed partition $U_{r,b}^{(i)}$.

It is evident from (8) and (9) that $U_{r,b}$, and in turn U_f, will have the same number of clusters as the base partition U_b. If the number of clusters c_r for each ensemble partition is chosen randomly from c_{min} to c_{max}, the criterion of selecting the base partition based on the CVI ranking (refer to Section V) does not always capture the most 'meaningful' information i.e., true number of clusters in the base partition. The problem of finding the true or best number of clusters, using CVIs, is well addressed in the literature. In our work, each ensemble partition having the best number of clusters c_r is obtained using a chosen CVI. For each downspace dataset, FCM clustering is performed with the number of clusters varying from c_{min} to c_{max}. Depending on the evaluation of the CVI, the ensemble partition U_r having the CVI-best number of clusters, c_r is obtained for each downspace dataset.

TABLE I: Time and space complexity of four FCM-based ensemble approaches

Ensemble Methods	Time Complexity	Space Complexity
EFCM [18]	$O(dlnc^2) + O(cNn^2)$, $d = n$	$O(n^2)$
RPFCM-A [20]	$O(dlnc^2) + O(cNn^2)$, $d = cN$	$O(n^2)$
RPFCM-B [21]	$O(dlnc^2) + O(n(cN)^2)$, $d = c$	$O(cnN)$
CAFCM (Proposed)	$O(nNc^2)$	$O(cn)$

l is the number of iterations to termination, d is the dimensions of the matrix on which clustering is applied, c is the number of clusters, n is the number of data points, and N is the number of random projections.

Algorithm 1 CAFCM: Cluster Ensemble for FCM Clustering with Random Projection

```
Input: Dataset $X \subset \mathbb{R}^{n \times p}$, cluster range = $\{c_{min}, c_{max}\}$, downspace dimension $q$, number of random projections $N$.
Output: Fuzzy partition $U_f$.
Step 1: Dataset generation in downspace.
  for $r$ = 1 to $N$ do
    Generate downspace datasets $Y_r \subset \mathbb{R}^{n \times q}$ using $Y = \frac{1}{\sqrt{q}}XT$, where $T \subset \mathbb{R}^{p \times q}$ is the random matrix built using (3).
  end for
Step 2: Run FCM on each $Y_r$, obtaining $U_r \in M_{fcn}$: $c = c_{min}$ to $c_{max}$.
Step 3: Get partitions $\{U_r\}_{r=1}^{N} \in M_{fc_rn}$, each partition having a CVI-best $c_r$ number of clusters, choosing each $c_r$ with an internal cluster validity index, $\mathscr{V}_{int_s}$.
Step 4: Get a set $\mathbb{U}$ of sorted partitions $\{U_{(r)}\}_{r=1}^{N} \in M_{fc_rn}$, as given in (7), using the cluster validity index, $\mathscr{V}_{int_s}$.
Step 5: Assign the best partition $U_{(1)}$ (from Step 4) as the base partition, i.e., $U_b^{(1)} = U_{(1)}$.
  for $i$ = 2 to $N$ do
    $W_{i,b} = U_b^{(i-1)} U_{(i)}^{-1}$
    $U_{i,b} = W_{i,b} U_{(i)}$
    $U_b^{(i)} = \frac{i-1}{i} U_b^{(i-1)} + \frac{1}{i} U_{i,b}$
  end for
$U_f = U_b$.
```

Our CAFCM algorithm for high dimensional data clustering using random projection and cumulative agreement based aggregation with FCM clustering is presented in Algorithm 1. In Step 1 of the Algorithm 1, multiple downspace datasets $\{Y_r\}$ are generated in a fixed lower dimensions; downspace $\mathbb{R}^q$ using random projection, as discussed in Section II. In Step 2, FCM clustering is applied to each downspace dataset Y_r, with the number of clusters varying from c_{min} to c_{max}. In Step 3, the partition U_r with the best number of clusters c_r is obtained for each downspace dataset, using a chosen CVI. This step gives N fuzzy partitions, each having a CVI-best number of clusters c_r. In Step 4, these N fuzzy partitions are ranked based on their quality as in (7). In our experiments, the *Normalized Partition Entropy* (PEB) $\mathscr{V}_{PEB_s}$ [38] was chosen as an internal index in Steps 3 and 4. Step 5 corresponds to the cumulative agreement based aggregation approach, as discussed in this Section. While FCM is part of the title of our algorithm, we point out that this scheme applies without change when the ensemble of soft partitions is generated by ANY fuzzy or probabilistic clustering algorithm.

The time and space complexity of the proposed aggregation approach and the three state-of-the-art ensemble approaches that are used for comparison is shown in Table I. Our aggregation approach has time complexity of $O(nNc^2)$ for matrix multiplication and computation of pseudoinverse of the rectangular matrix [42]. The fast Moore Penrose inverse method [42] was used to compute the pseudo inverse of ensemble partition $U_{(r)}$. Therefore, the proposed aggregation method has linear computational complexity in the number (n) of input samples. The CAFCM approach has the minimal space complexity, $O(cn)$, which is required to store the base partition that is updated sequentially in each iteration.

VII. Experiments

We performed five sets of experiments. In the first experiment, we explored the effect of using downspace datasets generated by different RP distributions (3) and (4) on the output partition. In the second experiment, an internal CVI validation test was performed among all internal CVIs to choose the best 'c_r' corresponding to each RP, and subsequently a best internal CVI is chosen. In the third experiment, an Internal/External (I/E) agreement test was performed to determine whether the partitions ranking, achieved by a soft external CVI, can also be obtained using a soft internal CVI. Based on the agreement performance of each internal CVI against the soft external CVI, we choose one best internal CVI to get sorted partitions for each dataset in our ensemble approach. In the fourth experiment, we demonstrate the effect of altering the ordering sequence of ensemble partitions on the output partition for CAFCM. In the last experiment, we compare different cluster ensemble approaches for high dimensional data clustering. To facilitate the comparison of these different approaches, we denote the approaches of [18] as EFCM, of [20] as RPFCM-A, of [21] as RPFCM-B, and our cumulative agreement based approach (Algorithm 1) as CAFCM. The experiments were performed in the MATLAB environment on a normal PC with the following configurations; OS: Windows 7 (64 bit); processor: Intel(R) Core(TM) i7-4770 @3.40GHz; RAM: 16GB.

A. Datasets and Parameter Settings

We performed our experiments on the following datasets.

1) Synthetic datasets: Two synthetic datasets, each having $n = 10000$ data points in $p = 1000$ dimensions, were constructed by drawing labeled samples from a mixture of three Gaussian distributions. GM1 is a well separated Gaussian mixture, while GM2 presumably has overlapping Gaussian clusters because its means are closer than those in GM1. The properties of these synthetic datasets are given in Table II.

TABLE II: Properties of two synthetic datasets GM1 and GM2

Component	**1**	**2**	**3**
Means			
GM1	$(-6,-6,...,-6)_{1000}$	$(0,0,...,0)_{1000}$	$(6,6,...,6)_{1000}$
GM2	$(-2,-2,...,-2)_{1000}$	$(0,0,...,0)_{1000}$	$(2,2,...,2)_{1000}$
Standard deviations in all directions			
GM1	$(1,1,...,1)_{1000}$	$(2,2,...,2)_{1000}$	$(3,3,...,3)_{1000}$
GM2	$(1,1,...,1)_{1000}$	$(2,2,...,2)_{1000}$	$(3,3,...,3)_{1000}$

2) Real datasets: Six publicly available real high-dimensional labeled datasets were chosen to demonstrate the applicability of our approach. The details are as follows:

KDD CUP 99 [43]: We used a sample of KDD CUP 99, which contains a wide variety of internet attacks simulated in a military environment. It consists of 494021 instances of 41 dimensional vectors, and each vector is labeled to specify the attack type. We normalized all 41 features to the interval $[0,1]$ by subtracting the minimum and then dividing by the subsequent maximum so that they all had same scale. This dataset contains 22 types of simulated attacks which fall into one of four main categories [43].

ACT [44]: This is a time-series dataset which contains data representing 19 activities such as sitting, walking, jumping etc., captured by 45 motion sensors over a 5 minute window sampled at 25Hz. Each activity is performed by 8 different subjects. The 5-min signals are divided into 5-sec segments so that 480 $(= 60 \times 8)$ signal segments are obtained for each activity. In each segment, there are a total of 125 $(=$5sec $\times$25Hz$)$ rows and 45 columns. We concatenated each segment data to obtain 9120 $(= 480 \times 19)$ instances in 5625 dimensions. All features were normalized to [0,1] using the method discussed earlier.

Forest Covertype [45]: These data consist of 54 cartographic features obtained by the U.S. Geological Survey and U.S. Forest Services, collected from a total of 581012 $(30m \times 30m)$ cells, which were then categorized into 7 forest cover types. This is a challenging dataset for any clustering algorithm as it contains ten continuous features, and 44 binary features (four wilderness types and 40 soil types). Because of the different nature of 54 features, we started developing our own distance metric using Euclidean and Hamming distance with normalized continuous feature (within [0,1]) that accounts for these differences to give similar weight to all the features. But the clustering results were slightly worse than using Euclidean distance alone. After several experiments, we discovered that the binary features do not add too much value in discriminating the forest Cover type. Using the Euclidean distance with scaled continuous features, with all binary features, yielded the best results in our experiments, therefore, we used Euclidean distance model for Forest dataset. We normalized the continuous features to the interval [0,1].

MNIST [46]: This dataset is a subset of a large set of handwritten images from the *National Institute of Standards and Technology* (NIST). It contains a total of 70000 784 $(= 28 \times 28)$ dimensional binary images of the digits 0 to 9. The main problem with handwritten images is that a single character can be written in many often quite different ways. This causes overlapping clusters in the data and makes it challenging for clustering.

HAR [47]: This time-series dataset contains 10299 instances of 6 daily activities performed by 30 subjects, while carrying a waist-mounted smart phone with embedded inertial sensors. It is a preprocessed dataset which has 561 features with time and frequency domain variables.

CIFAR 10 [48]: This dataset contains 60000 32*x*32 color images in 10 classes, with 6000 images per class. The classes are mutually exclusive. We concatenated each image into a $3072 = (32 \times 32 \times 3)$ dimensional feature vector.

3) Parameters: The model and error norms were both Euclidean for FCM except for the two time-series datasets. The

Cosine distance was used as model norm for HAR and ACT, based on its performances in previous studies [20]. This was done by replacing the Euclidean norm by the Cosine distance in the FCM function. In this case, the resultant algorithm is not alternating optimization since the FCM objective function has been abandoned. So this is an instance of alternating cluster estimation. The number of random projection (RPs), N is chosen as 30, unless stated otherwise. The weighting exponent $m = 2$, termination threshold $\varepsilon = 0.000001$, and the number of maximum iterations is chosen as 100 for the MATLAB implementation of FCM. Termination occurs when the absolute value of the difference between successive values of the FCM objective function using either distance is less than ε.

B. Evaluation Criteria

Adjusted Rand Index: The soft version [36] of the *adjusted rand index*, ARI_s (Hubert and Arabie [49]) is used as an external soft CVI. This index $\mathscr{V}_{ARI_s}(U|U_{gt})$ measures the degree to which a fuzzy partition U matches a crisp U_{gt}. Higher values indicate a better match, so $\mathscr{V}_{ARI_s}$ is a max-optimal CVI. This index maximizes at 1 when $U = U_{gt}$, and it's minimum may be negative when its expected value is not zero.

The *Normalized Partition Entropy* (PEB) $\mathscr{V}_{PEB_s}$ [38], *Partition Index* (SC) $\mathscr{V}_{SC_s}$ [50], *Normalized Partition Coefficient* (PCR) $\mathscr{V}_{PCR_s}$ [51], and *Xie-Beni index* (XB) $\mathscr{V}_{XB_s}$ [52], are used for internal CVI comparisons. Based on the min or max-optimality of internal CVIs, a set $\mathbb{U}$ of partitions, ordered in decreasing quality as in (7), is obtained for each internal CVI $\mathscr{V}_{int_s}$. The performance of each internal CVI $\mathscr{V}_{int_s}$ against the external CVI $\mathscr{V}_{ARI_s}$ is evaluated using two metrics:

Kendall's rank correlation coefficient [53]: Let E_{ext_s} and E_{int_s} be position vectors of $\mathscr{V}_{ext_s}$ and $\mathscr{V}_{int_s}$ respectively, which contain the ranking of sorted (descending order of quality) partitions. Kendall's coefficient τ measures the similarity between orderings in E_{ext_s} and E_{int_s}, which is given as [53]:

$$\tau = \frac{\begin{array}{c}\text{Number of}\\ \text{concordant pairs}\end{array} - \begin{array}{c}\text{Number of}\\ \text{discordant pairs}\end{array}}{N(N-1)/2}. \tag{11}$$

Kendall's τ is valued in $[-1,1]$, where 1 is for perfect agreement between two rankings, and -1, for perfect disagreement.

Position of the base partition: The selection of the best quality partition to be the base partition is important in our approach. Let the position of the best partition $U_{(1)}$ (first in E_{ext_s}) in E_{int_s} be denoted as $e_{U_{(1)}}$, then a position metric V_{U_b} is used to evaluate how accurately an internal CVI determines the position of the base partition in E_{int_s}, thus

$$V_{U_b} = 1 - \frac{e_{U_{(1)}} - 1}{N-1}. \in [0,1] \tag{12}$$

The integer $e_{U_{(1)}}$ is the position of the partition in the internal ranking E_{int_s} whose partition matches $U_{(1)} = U_b$, so $e_{U_{(1)}}$ can take any value from 1 to N. Suppose $e_{U_{(1)}} = 1$, so that $U_{(1)}$ is the best partition in both rankings E_{ext_s} and E_{int_s}, then $V_{U_b} = 1$. On the other hand, suppose $e_{U_{(1)}} = N$, then $V_{U_b} = 0$, So the range of V_{U_b} is $[0,1]$, maximum at 1 when the best external and best internal partition are the same; and minimum at 0 when the best external partition is the worst internal partition. The higher the value of V_{U_b}, the higher the ranking of the best partition $U_{(1)}$ in E_{int_s}.

TABLE III: The average $\mathscr{V}_{ARI_s}$ and downspace data generation time for distribution (3) and (4)

Random Matrix\Datasets	GM1		GM2	
	$\mathscr{V}_{ARI_s}$	Time (s)	$\mathscr{V}_{ARI_s}$	Time (s)
Distribution (3)	1.00	0.0266	0.90	0.0267
Distribution (4)	1.00	0.0265	0.90	0.0265

The evaluation criteria to compare the performances of different ensemble approaches are:

Accuracy: The similarity of the final clustering solution U_f with respect to ground truth partition U_{gt} is measured using $\mathscr{V}_{ARI_s}(U_f|U_{gt})$, for all four fuzzy ensemble approaches.

Run-Time: Running time is also an important criteria for comparison, which is related to the scalability of an algorithm. For each dataset, we pre-generated the downspace datasets using random projection, and used the same projection matrices for all algorithms. We keep the number of RPs N, and other parameters fixed for all approaches. We also compare the four fuzzy ensemble approaches based on the aggregation time T_{agg}, required to get a final output partition U_f from the N ensemble partitions.

C. Selection of Random Matrix T for Downspace Data (Y) Generation

We conducted an experiment to demonstrate that we can use either of equations (3) or (4) as the basis for random projection. Using datasets GM1 and GM2 with distributions (3) and (4), we generated downspace datasets $\{Y_r\}$ ($q = 100$) and used them in our framework for ensemble clustering. The average (10 trials) execution times for downspace data generation and the corresponding soft adjusted rand indices $\mathscr{V}_{ARI_s}$ for output partitions are shown in Table III. These values confirm that there is very little difference between the projections based on equations (3) and (4). As also shown in [16], both (3) and (4) are very simple probability distributions and all mathematical operations required to compute $Y = \frac{1}{\sqrt{q}}XT$ are very efficient and easy to implement. Subsequently, we used distribution (3) to generate downspace datasets in all the remaining experiments.

D. Internal CVIs Validation for Best 'c_r'

The base partition should ideally contain the nominally "true" target value for the number of clusters c_{gt}, that are identified by U_{gt}. In this regard, the best-c validation test [40] was performed using the four soft internal CVIs to estimate c_{gt} in all datasets. The downspace dimension q was chosen as 20. For the choices of $\varepsilon = \beta = 0.25$, and $n = 10000$, $q_o = 1591$, so q is well below the JL bound q_o. In this experiment, FCM was performed on each downspace dataset by partitioning the data at each value of c between $\{c_{min}, c_{max}\}$. The lower (c_{min}) and the upper (c_{max}) limits were chosen such that they under- and

TABLE IV: The average (20 trials) of the best 'c's from all internal CVIs ($\mathcal{V}_{int_s}$)

<Internal CVI>	**c_{gt}**	$<\mathcal{V}_{PEB_s}>$	$<\mathcal{V}_{SC_s}>$	$<\mathcal{V}_{XB_s}>$	$<\mathcal{V}_{PCR_s}>$
Synthetic Datasets					
GM1	3	3.0	3.0	2.1	3.0
GM2	3	3.0	3.0	2	2.9
Real Datasets					
MNIST	10	10.83	11.98	6	10.12
CIFAR	10	7.1	9.6	6.2	6.4
HAR	6	5.3	6.5	3	4.9
FOREST	7	4.8	6.7	4.2	4.4
ACT	19	18.8	21.5	17.1	18.2
KDD CUP	23	19.3	20.7	19.5	18.8
Root Mean Square Error		5.30	**4.00**	8.04	6.26

TABLE V: Average Values (5 trials) of Kendall's τ and (V_{U_b}) of internal CVIs against $\mathcal{V}_{ARI_s}$.

<Internal CVI>	$<\mathcal{V}_{PEB_s}>$	$<\mathcal{V}_{SC_s}>$	$<\mathcal{V}_{XB_s}>$	$<\mathcal{V}_{PCR_s}>$
Synthetic Datasets				
GM1	1.00 (1.00)	0.99 (1.00)	0.05 (0.96)	1.00 (1.00)
GM2	0.89 (1.00)	0.99 (1.00)	0.01 (0.41)	0.89 (1.00)
Real Datasets				
MNIST	0.36 (0.98)	0.11 (0.95)	0.01 (0.66)	0.23 (0.97)
CIFAR 10	0.25 (0.98)	0.42 (0.98)	-0.06(0.55)	0.28 (0.98)
HAR	0.68 (1.00)	0.26 (0.98)	0.06 (0.96)	0.58 (0.99)
FOREST	0.17 (0.98)	0.11 (0.98)	0.10 (0.86)	0.15 (0.96)
ACT	0.65 (1.00)	0.36 (1.00)	0.17 (0.94)	0.64 (1.00)
KDD CUP	0.19 (0.93)	0.09 (0.28)	0.10 (0.96)	0.18 (0.93)
Column Average	**0.52 (0.98)**	0.41 (0.89)	0.04 (0.78)	0.49 (0.98)

over-estimated the possible number of clusters in the data. The best quality partition, U_r, having c_r clusters, was chosen using each CVI based on its min/max optimality. This procedure was performed for each downspace projection, and *the (round) average of the 'best c's was used as an estimate of the true number of clusters in the upspace data*. In this test, randomly chosen subsets of each upspace dataset were used for the big datasets.

Table IV shows the estimated number of clusters in each dataset for each of the internal CVIs. The value of the apparent[2] true number of clusters c_{gt} is shown in the second column of Table IV. The values in the last row of Table IV show the square root of the sum of squared errors (RMSE) between c_{gt} and the estimated values for each internal CVI. In this exercise, $\mathcal{V}_{SC_s}$ produces slightly more reliable estimates of c_{gt} than the other three CVIs, whilst $\mathcal{V}_{PEB_s}$ produces the second best estimates of c_{gt}. We remark that these conclusions are not generally applicable. You could test many different CVIs and get different best results. Or you could change datasets and discover that $\mathcal{V}_{SC_s}$ and $\mathcal{V}_{PEB_s}$ performed badly. And so on, ad infinitum. It can also be observed from Table IV that $\mathcal{V}_{PCR_s}$ works best for MNIST, $\mathcal{V}_{PEB_s}$ for ACT, while $\mathcal{V}_{SC_s}$ is best for rest of the datasets. We tested the performance of the CAFCM algorithm using both $\mathcal{V}_{SC_s}$ and $\mathcal{V}_{PEB_s}$ in Step 3, and the final results were very similar. Therefore, we chose $\mathcal{V}_{PEB_s}$ as the best internal CVI based on this and the I/E agreement test (next) for use in Steps 3 and 4 of CAFCM Algorithm.

E. The Internal/External (I/E) Agreement Test

In this experiment, we performed the Internal/External (I/E) agreement test, in which the performance of an internal CVI is compared with the performance of an external CVI to assess whether they both yield similar base partition and similar partition rankings or not [29], [54]. We compared the partition rankings and the base partition obtained using the external CVI ($\mathcal{V}_{ARI_s}$), with the partition rankings and base partition obtained using each of the four internal CVIs. Among these four internal CVIs, the CVI which determines the most similar partition ranking and base partition, obtained using the external CVI, is chosen for use in our framework. Using this best internal CVI, we hope to achieve the desired partition rankings and base partition in the best possible way when ground truth data are not available (the unlabeled case).

1) Partition rankings comparison: Step 3 of the CAFCM algorithm produces the N ensemble partitions having best 'c_r' number of clusters. The ranking of each ensemble of fuzzy partitions is established using the external CVI $\mathcal{V}_{ARI_s}$, and the four soft internal CVIs $\mathcal{V}_{PEB_s}$, $\mathcal{V}_{SC_s}$, $\mathcal{V}_{XB_s}$, and $\mathcal{V}_{PCR_s}$, based on the partition quality. The partitions ranking, E_{int_s}, of each of the four soft internal CVIs was compared with the partitions ranking, E_{ext_s}, of soft external CVI, $\mathcal{V}_{ARI_s}$, for each dataset using the Kendall rank correlation coefficient.

2) Base partition comparison: Besides the partition rankings, the selection of the base partition, U_b, is also important in our framework. In this experiment, the position $e_{U_{(1)}}$ of the base partition U_b, the best external CVI partition (first in E_{ext_s}), in each internal CVI partition ranking E_{int_s} was used to compute the position metric V_{U_b} for each internal CVI and for each dataset.

The values of τ and V_{U_b} were computed between rankings $E_{ext_s} = \{E_{ARI_s}\}$ and each ranking of $E_{int_s} = \{E_{PEB_s}, E_{SC_s}, E_{XB_s}, E_{PCR_s}\}$, using (11) and (12). This procedure was repeated 5 times for each dataset.

Table V shows the averaged values of τ and V_{U_b} (in parentheses) corresponding to the order of the N fuzzy partitions established by each internal CVI for each dataset. The notation <CVI> in the first row of the table indicates the basis of the τ and V_{U_b} values that are displayed in each column, not to be confused with the value of the CVIs, which are NOT shown. The values in each column are formatted with just enough resolution so that the optimal values can be seen.

Apparently all of the CVIs except $\mathcal{V}_{XB_s}$ perform well for the two synthetic datasets, which means three internal CVIs are able to achieve almost the same ranking of partitions as obtained by the external CVI $\mathcal{V}_{ARI_s}$. The τ value of all four CVIs degrades for the real datasets. However, the (V_{U_b}) values of $\mathcal{V}_{PCR_s}$ and $\mathcal{V}_{PEB_s}$ are high for all real datasets, which means they reliably choose the best quality partition from the N ensemble partitions. The last row of Table V contains

[2] We say apparent because it is well known that labeled data which contain $c1$ physically labeled subsets often possess $c2 \neq c1$ "best clusters" with respect to a given model and algorithm [29].

TABLE VI: The effects of ordered versus random aggregation of ensemble partitions (tabulated values are the 10 trial average of $\mathscr{V}_{ARI_s}$).

Sequence Ordering of Partitions	**GM1** ($q = 30$)	**GM2** ($q = 100$)
Decreasing order of quality	1.00	0.90
Arbitrary order	0.98	0.85

column averages, and it shows that overall $\mathscr{V}_{PCR_s}$ and $\mathscr{V}_{PEB_s}$ perform well (with a very slight advantage to $\mathscr{V}_{PEB_s}$), while $\mathscr{V}_{XB_s}$ performs worst.

Based on this overall performance of four internal CVIs in determining partition rankings and the base partition, the performance of $\mathscr{V}_{PEB_s}$ (internal CVI) agrees best with the performance of the soft external index $\mathscr{V}_{ARI_s}$. Therefore, we chose $\mathscr{V}_{PEB_s}$ to determine the base partition and a set of sorted partitions, required in Step 4 of CAFCM Algorithm. The CVI $\mathscr{V}_{PEB_s}$ is also used in Step 3 of Algorithm 1 to obtain the ensemble partitions, having the best 'c_r' number of clusters.

F. Effect of Ordering Sequence of Partitions on Output Partition

To demonstrate the effect of altering the ordering of the ranked queue, as shown in (7), on the output partition, we performed an experiment using datasets GM1 and GM2 considering two cases viz., where sequence of ensemble partitions is (i) ordered and (ii) arbitrary. First, we obtained a base partition for each dataset in the manner described. Table VI compares the $\mathscr{V}_{ARI_s}$ values of the output partition obtained when the ensemble partitions are combined in a sequential manner based on their CVI quality as in (7) to the $\mathscr{V}_{ARI_s}$ values of the output partition obtained when the $N-1$ remaining partitions are combined with the base partition in an arbitrary order. The average $\mathscr{V}_{ARI_s}$ values (10 trials) in Table VI make it clear that combining the remainder partitions according to their CVI rank yields better $\mathscr{V}_{ARI_s}$ values (and hence, a better output partition) than arbitrary combination.

G. Comparison of Different Cluster Ensemble Methods

In this experiment, we compare the performance of our approach with three existing ensemble approaches for high dimensional data clustering using random projection with FCM. We discuss the performance of all four cluster ensemble approaches in 5 data groups (**G1-G5**), based on the different attributes of datasets.

*Synthetic datasets of different downspace dimensions q (**G1**):* For synthetic datasets GM1, GM2, experiments were performed for downspace dimension $q = 10, 20, 30, 50, 100$. These q values are corresponding to rogue random projections, which are chosen irrespective of ε and β (below the JL bound) as mentioned in Section VII-D. The average $\mathscr{V}_{ARI_s}$ values and ensemble time T_{agg} of all approaches over 5 trials for GM1 and GM2 are shown in Table VII. The best performance approach for each downspace dimension is highlighted in bold. It is evident from the values in Table VII that even with $q = 10$, all the ensemble approaches achieve very good clustering results ($\mathscr{V}_{ARI_s} > 0.9$) for the GM1 dataset. This is because the clusters in this dataset are (probably) well separated from each other. EFCM and RPFCM-B get perfect results ($\mathscr{V}_{ARI_s} = 1$) for $q = 10$ and 20. The CAFCM approach performs reasonably well ($\mathscr{V}_{ARI_s} > 0.9$) in significantly less computation time, and achieves perfect results for $q = 30$. It can be concluded from Table VII that the CAFCM approach is $10-100$ times faster than the other three approaches. All four approaches get perfect results for $q = 30$ and above, so we do not compare them for higher downspace dimensions.

TABLE VIII: Average $\mathscr{V}_{ARI_s}$ values and ensemble time T_{agg} (s) for different number of RPs (N) on the GM2 dataset.

	EFCM		***RPFCM-A***		***RPFCM-B***		***CAFCM***	
N	ARI	T_{agg}	ARI	T_{agg}	ARI	T_{agg}	ARI	T_{agg}
5	**0.56**	75	0.43	0.12	0.45	0.12	0.52	**0.00**
10	**0.69**	70	0.44	0.17	0.60	0.12	0.65	**0.00**
20	0.66	88	0.43	0.40	0.70	0.11	**0.74**	**0.01**
30	0.62	98	0.58	0.66	0.71	0.13	**0.79**	**0.02**
40	0.63	97	0.41	0.85	0.74	0.16	**0.85**	**0.03**
50	0.80	126	0.62	1.08	0.82	0.18	**0.89**	**0.03**

For the GM2 dataset, CAFCM performs significantly better than the other three approaches for all downspace dimensions except $q = 10$. The weak performance of CAFCM for $q = 10$ may be because, the distribution of points among clusters changes in each consensus partition, which in turn, causes the weak agreements of points for any cluster across all consensus partitions. Whereas for $q > 10$, more features make stronger agreement of each data point for any cluster. The CAFCM algorithm performs aggregation in negligible time compared to the other three approaches, for both synthetic datasets. This is because, unlike other ensemble approaches, CAFCM does not use FCM on a final aggregation matrix to get the final membership matrix.

In order to compare the performance of all four ensemble methods with respect to stability, the standard deviation (rounded off) of $\mathscr{V}_{ARI_s}$ values with average values are shown in Table VII. We can see that CAFCM seems to be the least variable among all the approaches. This might be due to the smoothing effect from sequential averaging of the transformed partitions and base partition (refer to Algorithm 1). The EFCM algorithm seems to be the most stable of the other three approaches.

*Synthetic dataset GM2 for different number of RPs, N (**G2**):* We conducted another experiment for GM2 dataset for different numbers of RPs, N (ensemble size). For datasets having high diversity (overlapping clusters) like GM2, increasing N may be beneficial because there will probably be much more diversity in the random projections due to the mixed clusters in the upspace. Table VIII shows the average $\mathscr{V}_{ARI_s}$ values and ensemble time (5 trials) of all approaches for a fixed value of $q(= 40)$. It can be noted that CAFCM gives the best performance for all Ns except $N = 5$ and 10. As expected, the adjusted rand index ($\mathscr{V}_{ARI_s}$) increases for all approaches as N increases. Unlike existing approaches, increasing the ensemble size has a negligible effect on the computational time of CAFCM. The maximum speedup is CAFCM:EFCM is $4200:1$ at $N = 50$, and the minimum speedup is CAFCM:RPFCM-B

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TFUZZ.2017.2729501, IEEE Transactions on Fuzzy Systems

TABLE VII: Average $\mathscr{V}_{ARI_s}$ values and ensemble time T_{agg} (in s) for all approaches on the GM1 and GM2 datasets.

	EFCM		***RPFCM-A***		***RPFCM-B***		***CAFCM***	
q	$\mathscr{V}_{ARI_s}$	T_{agg}	$\mathscr{V}_{ARI_s}$	T_{agg}	$\mathscr{V}_{ARI_s}$	T_{agg}	$\mathscr{V}_{ARI_s}$	T_{agg}
GM1 Dataset $c_r \in \{2,8\}$								
10	1.00± 0.0	68.9	0.97± 0.0	0.36	**1.00± 0.0**	0.15	0.94± 0.0	**0.01**
20	1.00± 0.0	70.9	0.99± 0.0	0.39	**1.00± 0.0**	0.13	0.99± 0.0	**0.01**
30	1.00± 0.0	71.9	1.00± 0.0	0.40	1.00± 0.0	0.16	**1.00± 0.0**	**0.02**
GM2 Dataset $c_r \in \{2,8\}$								
10	**0.76± 0.02**	89.6	0.40± 0.02	0.18	0.75± 0.12	0.15	0.61± 0.01	**0.00**
20	0.60± 0.10	83.2	0.43± 0.15	0.54	0.45± 0.26	11.7	**0.68± 0.02**	**0.01**
30	0.79± 0.18	82.4	0.47± 0.03	0.52	0.30± 0.01	11.03	**0.83± 0.01**	**0.02**
50	0.90± 0.02	71.1	0.55± 0.16	0.54	0.70± 0.22	0.12	**0.90± 0.01**	**0.02**
100	0.85± 0.19	73.1	0.75± 0.23	0.47	0.63± 0.29	0.11	**0.90± 0.02**	**0.02**

$= 11 : 1$ at $N = 20$.

*High dimensional real datasets (ACT, HAR, MNIST and CIFAR) for different q (**G3**):* In this group, we discuss the performance on the real datasets ACT, HAR, MNIST and CIFAR, which have relatively high dimensions (in hundreds and thousands) as compared to the KDD CUP and FOREST datasets, which have smaller upspace dimensions. For G3 datasets, the downspace dimensions $q = 10, 20, 30, 50, 100$ were chosen. Line-plots are used to present the $\mathscr{V}_{ARI_s}$ values of all ensemble approaches for different downspace dimensions, which are shown in the left columns of Figs. 2 and 3, whereas, the right columns in Figs. 2 and 3 shows the time performance (on logarithmic scale) of all ensemble approaches for different numbers of downspace dimensions. We did not apply EFCM to MNIST, CIFAR (as $n > 50000$) to avoid an out of memory error, and its associated computational load. Therefore, the time performance for these datasets is shown on a non-logarithmic scale. The minimum and maximum number of clusters in consensus partitions is shown in the title of the figure for each dataset.

Figs. 2(a) and (b) show that CAFCM outperforms all other ensemble methods for the two time-series datasets (HAR and ACT). For the image datasets (MNIST and CIFAR), the performance of CAFCM is comparable to RPFCM-B, and outperforms RPFCM-A. The aggregation time for CAFCM is quite small compared to the other three approaches, which agrees with our time complexity analysis as discussed in Section III.

*KDD CUP and FOREST Covertype (**G4**):* The upspace dimensions for FOREST and KDD CUP are 41 and 54, respectively, so we chose the downspace dimensions to be $q = 10, 20, 30, 40$. For each of these datasets, the experiments were performed on a subset of $n = 100,000$ instances. Consequently, the EFCM algorithm was not applied on these datasets to avoid the associated computational load. The performance of all ensemble approaches for these two datasets, is shown in Figs. 3 (a) and (b) respectively. The CAFCM approach performs better than the other three ensemble methods for almost all of the downspace dimensions. The CAFCM algorithm achieves near to best accuracy even with $q = 10$ (25%) dimensions for these two datasets. The time performance in Fig. 3 (b) shows that even for the large datasets, CAFCM takes negligible time for aggregation compared to the other approaches.

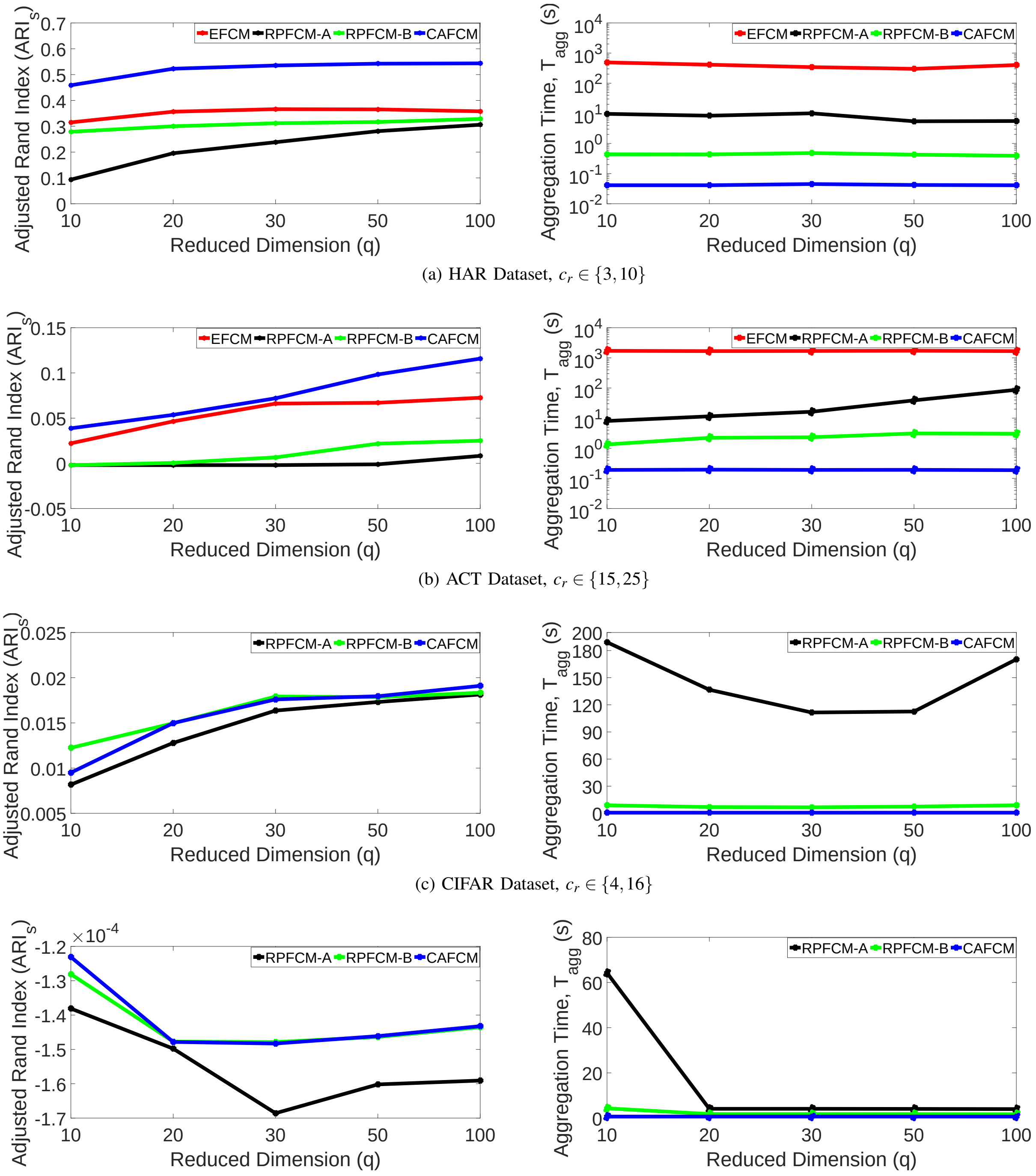
*Performance of all ensemble approaches for different number of samples (n) (**G5**):* In order to demonstrate the applicability of our algorithm for big data, the time performance of each ensemble approach for different number of samples of the KDD CUP dataset is presented in Fig. 4 (on logarithmic scale). EFCM tests were limited to $n = 20,000$ input samples to avoid the large computational burden. We see that CAFCM takes just a few seconds for even $n = 100,000$ samples. The maximum computational time (for $100,000$ samples) of CAFCM is no more than the minimum time (for $10,000$ samples) taken by the other approaches.

VIII. Conclusions and Discussion

This paper introduces a simple and computationally efficient framework called CAFCM for high dimensional data clustering, which employs FCM clustering an ensemble of random projections. Three other state-of-the-art ensemble approaches that also use FCM clustering are discussed in this paper. These approaches require large amounts of space for storing a big affinity matrix. In addition, they also require FCM clustering on a large affinity matrix to get the final partition, so they incur much larger computation time than CAFCM does.

The CAFCM algorithm eliminates the complexity involved in dealing with a final affinity matrix using a cumulative agreement based fuzzy partition aggregation approach. The final CAFCM partition is achieved with cumulative agreement based relabelling and averaging of the ensemble of fuzzy partitions. Each partition is taken sequentially from a ranked queue established per equation (7). The ranks are computed with a cluster validity index. The highest ranking partition becomes the core partition U_b, and this partition drives the agreement procedure.

We experimented with different internal CVIs to assess the quality of ensemble partitions having known target (true) numbers of labeled subsets. The performance of four internal CVIs were correlated with the assessments made by the soft external ARI, $\mathscr{V}_{ARI_s}$. The normalized soft partition entropy ($\mathscr{V}_{PEB_s}$) index led to the best final partitions in the experiments presented here. Once the CVIs for steps 3 and 4 in Algorithm 1 are chosen, our approach does not require any prior knowledge of the number of clusters that might be present in the dataset, which makes it attractive for real clustering problems. We demonstrated the superiority of our CAFCM approach by comparing it with three existing approaches on two Gaussian



(a) HAR Dataset, $c_r \in \{3,10\}$

(b) ACT Dataset, $c_r \in \{15,25\}$

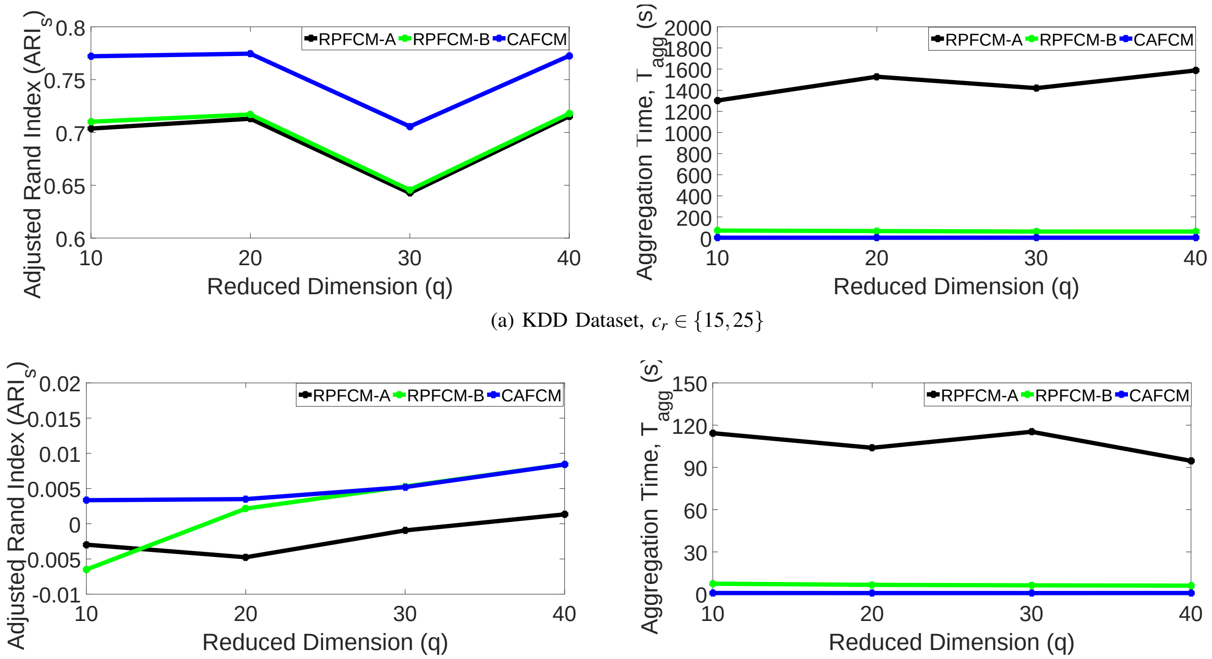
(c) CIFAR Dataset, $c_r \in \{4,16\}$

(d) MNIST Dataset,, $c_r \in \{4,16\}$

Fig. 2: $\mathscr{V}_{ARI_s}$ values (in left column) and Aggregation time T_{agg} (in right column) for different downspace dimensions

mixture datasets and six real datasets. Our experimental results show that CAFCM outperforms the other three approaches in terms of accuracy, stability, space, and time complexity. Experimental results reveal that on average our algorithm runs one to two orders of magnitude $(10-100$ times) faster than other state-of-the-arts algorithms, and at best, can achieve speedups in on the order of $4000:1$.

We also showed that CAFCM can produce reasonable performance even for downspace dimensions well below the JL bound (rogue random projections). This is very important when the dataset has many features. For example, even with $q=10$, the CAFCM approach produced good results on the ACT data. The proposed CAFCM algorithm has linear $O(n)$ time complexity in the number (n) of data points. We also showed empirically that our algorithm scales linearly in the number of samples (n) for a big dataset (KDD CUP). The CAFCM ensemble time for $n=100{,}000$ samples was less than the minimum ensemble time for the other approaches for any number of samples. The CAFCM algorithm may take hundreds of seconds for very large $(n \sim 10^9)$ datasets.



(a) KDD Dataset, $c_r \in \{15, 25\}$

(b) FOREST Dataset, $c_r \in \{3, 15\}$

Fig. 3: $\mathscr{V}_{ARI_s}$ values (in left column) and Aggregation time T_{agg} (in right column) for different downspace dimensions

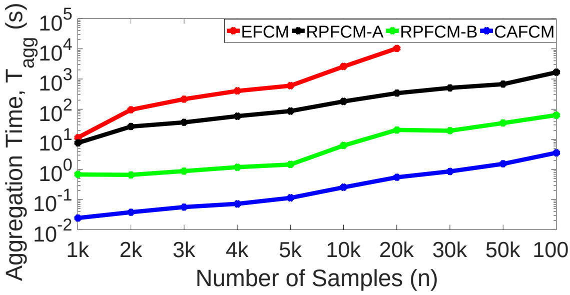


Fig. 4: KDD CUP Dataset: Aggregation time T_{agg} for different number of samples

However, our aggregation approach takes only about a second for $n = 100,000$ samples, and we estimate that it will take only a few seconds for a $n = 10^6$ data points.

Acknowledgment

We acknowledge the support from Australian Research Council (ARC) Linkage Project grant (LP120100529), the ARC Linkage Infrastructure, Equipment and Facilities scheme (LIEF) grant (LF120100129).

References

[1] S. K. Halgamuge and L. Wang, *Classification and clustering for knowledge discovery*. Springer Science & Business Media, 2005, vol. 4.

[2] M. Moshtaghi, S. Rajasegarar, C. Leckie, and S. Karunasekera, "Anomaly detection by clustering ellipsoids in wireless sensor networks," in *5th International Conference on Intelligent Sensors, Sensor Networks and Informations Processing (ISSNIP)*, 2009, pp. 331–336.

[3] J. C. Bezdek, T. C. Havens, J. M. Keller, C. Leckie, L. Park, M. Palaniswami, and S. Rajasegarar, "Clustering elliptical anomalies in sensor networks," in *IEEE International Conference on Fuzzy systems (FUZZ)*, 2010, pp. 1–8.

[4] S. M. Erfani, M. Baktashmotlagh, S. Rajasegarar, S. Karunasekera, and C. Leckie, "R1SVM: a Randomised Nonlinear Approach to Large-Scale Anomaly Detection," in *Proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*, 2015, pp. 432–438.

[5] S. M. Erfani, M. Baktashmotlagh, S. Rajasegarad, V. Nguyen, C. Leckie, J. Bailey, and K. Ramamohanarao, "R1stm: One-class support tensor machine with randomised kernel," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 198–206.

[6] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," *ACM SIGMOD Record*, vol. 30, no. 2, pp. 151–162, 2001.

[7] Q. Du and J. E. Fowler, "Hyperspectral image compression using jpeg2000 and principal component analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 201–205, 2007.

[8] E. P. Xing, M. I. Jordan, R. M. Karp *et al.*, "Feature selection for high-dimensional genomic microarray data," in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 1, 2001, pp. 601–608.

[9] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 245–250.

[10] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New directions in Statistical Physics*. Springer, 2004, pp. 273–309.

[11] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[12] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *SIGMOD Rec.*, vol. 27, pp. 94–105, 1998.

[13] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of Educational Psychology*, vol. 24, no. 6, p. 417, 1933.

[14] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, pp. 403–420, 1970.

[15] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," in *Proceedings of International Joint Conference on Neural Networks*, vol. 1, 1998, pp. 413–418.

[16] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2001, pp. 274–281.

[17] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, "Latent

semantic indexing: A probabilistic analysis," in *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 1998, pp. 159–168.

[18] R. Avogadri and G. Valentini, "Fuzzy ensemble clustering based on random projections for dna microarray data analysis," *Artificial Intelligence in Medicine*, vol. 45, no. 2, pp. 173–183, 2009.

[19] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 3, 2003, pp. 186–193.

[20] M. Popescu, J. Keller, J. Bezdek, and A. Zare, "Random projections fuzzy c-means (rpfcm) for big data clustering," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2015, pp. 1–6.

[21] M. Ye, W. Liu, J. Wei, and X. Hu, "Fuzzy-means and cluster ensemble with random projection for big data clustering," *Mathematical Problems in Engineering*, 2016.

[22] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.

[23] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *Proceedings of the twenty-first ACM Twenty-first International Conference on Machine Learning*, 2004, p. 36.

[24] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[25] A. L. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 4, 2002, pp. 276–280.

[26] S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure," *Bioinformatics*, vol. 19, no. 9, pp. 1090–1099, 2003.

[27] E. Dimitriadou, A. Weingessel, and K. Hornik, "A combination scheme for fuzzy clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 07, pp. 901–912, 2002.

[28] H. G. Ayad and M. S. Kamel, "Cumulative voting consensus method for partitions with variable number of clusters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 160–173, 2008.

[29] J. C. Bezdek, *Primer on Cluster Analysis: Four Basic Methods that (Usually) Work*. First Edition Design Publishing, 2017, vol. 1.

[30] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*. Marcel Dekker, 1988, vol. 84.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (methodological)*, pp. 1–38, 1977.

[32] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary Mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[33] S. Dasgupta, "Experiments with random projection," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, 2000, pp. 143–151.

[34] J. C. Bezdek, X. Ye, M. Popescu, J. Keller, and A. Zare, "Random projection below the JL limit," in *Proceedings of International Joint Conference on Neural Network (IJCNN)*, 2016, pp. 2414–2423.

[35] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[36] D. T. Anderson, J. C. Bezdek, M. Popescu, and J. M. Keller, "Comparing fuzzy, probabilistic, and possibilistic partitions," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 5, pp. 906–918, 2010.

[37] Y. Lei, J. C. Bezdek, J. Chan, N. X. Vinh, S. Romano, and J. Bailey, "Generalized information theoretic cluster validity indices for soft clusterings," in *IEEE Symposium on Proceedings of the Eighth International Conference on Numerical Taxonomy*, 2014, pp. 24–31.

[38] J. C. Bezdek, "Mathematical models for systematics and taxonomy," in *Proceedings of the Eighth International Conference on Numerical Taxonomy*, 1975, pp. 143–66.

[39] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 370–379, 1995.

[40] J. C. Bezdek, M. Moshtaghi, T. Runkler, and C. Leckie, "The generalized C index for internal fuzzy cluster validity," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1500–1512, 2016.

[41] J. Wall, "Generalized inverses of stochastic matrices," *Linear Algebra and its Applications*, vol. 10, no. 2, pp. 147–154, 1975.

[42] P. Courrieu, "Fast computation of moore-penrose inverse matrices," *CoRR*, vol. abs/0804.4809, 2008. [Online]. Available: http://arxiv.org/abs/0804.4809

[43] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications*, 2009.

[44] K. Altun, B. Barshan, and O. Tunçel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, 2010.

[45] J. A. Blackard and D. J. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," *Computers and Electronics in Agriculture*, vol. 24, no. 3, pp. 131–151, 1999.

[46] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist dataset of handwritten digits," *URL http://yann. lecun. com/exdb/mnist*, 1998.

[47] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *ESANN*, 2013.

[48] A. Krizhevsky and G. Hinton, *Learning multiple layers of features from tiny images*. Citeseer, 2009.

[49] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[50] N. Zahid, M. Limouri, and A. Essaid, "A new cluster-validity for fuzzy clustering," *Pattern Recognition*, vol. 32, no. 7, pp. 1089–1097, 1999.

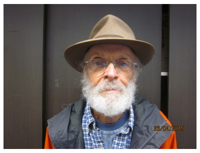
[51] M. Roubens, "Pattern classification problems and fuzzy sets," *Fuzzy Sets and Systems*, vol. 1, no. 4, pp. 239–253, 1978.

[52] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 841–847, 1991.

[53] M. G. Kendall, *Rank correlation methods*. Griffin, 1948.

[54] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, 2013.

Punit Rathore received the Master of Technology (M.Tech) in Electrical Engineering (Instrumentation) from the Indian Institute of Technology, Kharagpur, India in 2011. He has worked as Researcher in TATA Steel Limited, India for three and half years (2011-14). He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, Australia. His research interests include big data clustering, incremental clustering, spatio-temporal analytics, Internet of Things, machine learning, pattern recognition, and signal processing.



James C. Bezdek (LF'10) received the PhD in Applied Math, Cornell University, 1973. Jim is past president of NAFIPS (North American Fuzzy Information Processing Society), IFSA (International Fuzzy Systems Association) and the IEEE CIS (Computational Intelligence Society as the NNC): founding editor the Int'l. Jo.Approximate Reasoning and the IEEE Transactions on Fuzzy Systems: Life fellow of the IEEE and IFSA; recipient of the IEEE 3rd Millennium, IEEE CIS Fuzzy Systems Pioneer, IEEE Frank Rosenblatt TFA and the Kempe de Feret IPMU awards. He retired in 2007. His research interests include optimization, pattern recognition, clustering in very large data, coclustering, and visual clustering.

Sarah M. Erfani is a lecturer in the School of Computing and Information Systems at The University of Melbourne. Her research interests include machine learning, large-scale data mining, cyber security, and data privacy.

Sutharshan Rajasegarar received the B.Sc. Engineering degree in electronic and telecommunication engineering (First Class Hons.) from the University of Moratuwa, Moratuwa, Sri Lanka, in 2002, and the Ph.D. degree from the University of Melbourne, Melbourne, VIC, Australia, in 2009.

He is currently a Research Fellow with the Department of Electrical and Electronic Engineering, University of Melbourne. His current research interests include wireless sensor networks, anomaly/outlier detection, spatio-temporal estimations, Internet of Things, machine learning, pattern recognition, signal processing, and wireless communication.

Marimuthu Palaniswami (F'12) received the M.E. degree in electrical, electronic and control engineering from the Indian Institute of Science, Bengaluru, India, the M.Eng.Sc. degree in electrical, electronic and control engineering from the University of Melbourne, Melbourne, VIC, Australia, and the Ph.D. degree from the University of Newcastle, N.S.W., Australia.

He is currently a Professor with the University of Melbourne. He is representing Australia as a core partner in EU FP7 projects such as SENSEI, SmartSantander, Internet of Things Initiative, and SocIoTal. He has been funded by several Australian Research Council (ARC) and industry grants (over 40 million) to conduct research in sensor network, Internet of Things (IoT), health, environmental, machine learning, and control areas. He has published over 400 refereed research papers, and leads one of the largest funded ARC Research Network on Intelligent Sensors, Sensor Networks and Information Processing Programme. His current research interests include SVMs, sensors and sensor networks, IoT, machine learning, neural network, pattern recognition, and signal processing and control.

TABLE IX: The Contingency Table A to compare partition U and V

		Partition V v_j = row j of V	
	Class	v_1 v_2 ... v_r	Sums
Partition U u_i = row i of U	u_1 u_2 u_3 . u_c	$A = \begin{bmatrix} n_{11} & n_{12} & \dots & n_{1r} \\ n_{21} & n_{22} & \dots & n_{2r} \\ n_{31} & n_{32} & \dots & n_{3r} \\ \dots & \dots & \dots & \dots \\ n_{c1} & n_{c2} & \dots & n_{cr} \end{bmatrix} = UV^T$	$n_{1\bullet}$ $n_{2\bullet}$ $n_{3\bullet}$. $n_{c\bullet}$
	Sums	$n_{\bullet 1}$ $n_{\bullet 2}$ ... $n_{\bullet r}$	$n_{\bullet\bullet} = n$

TABLE X: Cluster Validity Indices used in this paper

CVI	**Formula**	**Description**	**Optimality/Range**
Soft External CVI			
Adjusted Rand Index (ARI_s) [36]	$\dfrac{a - \frac{(a+c)(a+b)}{(a+b+c+d)}}{\frac{(a+c)+(a+b)}{2} - \frac{(a+c)(a+b)}{(a+b+c+d)}}$	The parameter a,b,c, and d (refer to [49]) are derived from generalized contingency matrix $A^* = \phi UV^T$ (Table IX), where $\phi = \frac{n}{\sum_{i=1}^{c} n_{i\bullet}}$	Max-optimal, maximum=1, minimum can be negative if index is less than expected value.
Normalized Mutual Information (NMI_s) [37]	$MI(U,V)/max(H(U),H(V))$	$MI = \sum_{i=1}^{c}\sum_{j=1}^{r}(n_{ij}/n)log(\frac{n_{ij}/n}{n_{i\bullet}n_{j\bullet}/n^2})$, $H(U) = -\sum_{i=1}^{c}(n_{i\bullet}/n)log(n_{i\bullet}/n)$, where $n_{i\bullet}$ are derived from contingency matrix.	Max-optimal, and ranges in [0,1]
Soft Internal CVI			
Normalized Partition Entropy (PEB) [38]	$\left(-\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}log(u_{ij})\right)/ln_a c$	u_{ij} is the fuzzy membership degree of object x_j to i-th cluster. The c is the number of clusters. This validity index requires only membership values.	Min-optimal, and ranges in [0,1]
Normalized Partition Coefficient (PCR) [51]	$(c\frac{\|\|U\|\|_2^2}{n} - 1)/(c-1)$	$\|\|U\|\|_2^2 = \sum_{i=1}^{c}\sum_{j=1}^{n}(u_{ij})^2$.This validity index requires only membership values.	Max-optimal, and ranges in [0,1]
Partition Index (SC) [50]	$\sum_{i=1}^{c}\left(\frac{\sum_{j=1}^{n}(u_{ij})\|\|x_j - V_i\|\|^{m/2}}{\sum_{j=1}^{n}(u_{ij})^2}\right)$	$V_i(1 \leq i \leq c)$ is center for each cluster, and m is the weighting exponent. This validity index requires the membership values and the dataset both.	Max-optimal
Xie Beni (XB) [52]	$\frac{\sum_{i=1}^{c}\sum_{j=1}^{n}[u_{ij}^m\|\|x_j - V_i\|\|^2]}{n\ \min_{i\neq j}(\|\|V_i - V_j\|\|)}$	This validity index requires the membership values and the dataset both.	Max-optimal