Pattern Recognition 43 (2010) 2315–2329

Contents lists available at ScienceDirect

Pattern Recognition

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

Cooperative clustering

Rasha Kashef [a,*], Mohamed S. Kamel [b]

[a] *Arab Academy for Science and Technology, Computer Science Department, Egypt*
[b] *University of Waterloo, Electrical and Computer Engineering Department, Canada*

ARTICLE INFO

Article history:
Received 15 October 2008
Received in revised form
17 December 2009
Accepted 27 December 2009

Keywords:
Cooperative clustering
Similarity histogram
Cooperative contingency graph

ABSTRACT

Data clustering plays an important role in many disciplines, including data mining, machine learning, bioinformatics, pattern recognition, and other fields, where there is a need to learn the inherent grouping structure of data in an unsupervised manner. There are many clustering approaches proposed in the literature with different quality/complexity tradeoffs. Each clustering algorithm works on its domain space with no optimum solution for all datasets of different properties, sizes, structures, and distributions. In this paper, a novel cooperative clustering (CC) model is presented. It involves cooperation among multiple clustering techniques for the goal of increasing the homogeneity of objects within the clusters. The CC model is capable of handling datasets with different properties by developing two data structures, a histogram representation of the pair-wise similarities and a cooperative contingency graph. The two data structures are designed to find the matching sub-clusters between different clusterings and to obtain the final set of clusters through a coherent merging process. The cooperative model is consistent and scalable in terms of the number of adopted clustering approaches. Experimental results show that the cooperative clustering model outperforms the individual clustering algorithms over a number of gene expression and text documents datasets.

1. Introduction

Data clustering is a data analysis technique that enables the abstraction of large amounts of data by forming meaningful groups or categories of objects, formally known as clusters, such that objects in the same cluster are similar to each other, and those in different clusters are dissimilar according to some similarity criteria. The increasing importance of data clustering in its widespread applications has led to the development of a variety of algorithms with different quality/complexity tradeoffs [1–18]. The algorithms differ in many aspects, such as the types of attributes they use to characterize the dataset, the similarity measure used, and the representation of the clusters. It is well known that no clustering method can adequately handle all types of cluster structures and properties (e.g. overlapping, shape, size and density). In fact, the cluster structure produced by a clustering method is sometimes an artifact of the method itself that is actually imposed on the data rather than that discovered about its true structure.

Combining multiple clustering is considered as an example to further broaden and stimulate new progress in the area of data clustering. Current approaches of combining clusterings include ensemble clustering and hybrid clustering [19–26]. Ensemble clustering integrates a collection of "base clusterings" to produce a more accurate partition of a dataset. Recent ensemble clustering techniques have been shown to be effective in improving the accuracy and stability of standard clustering algorithms and can provide novel, robust, and stable solutions. However, inherent drawbacks of these techniques are: (1) the computational cost of generating and combining multiple clusterings of the data, and (2) designing a proper cluster ensemble that addresses the problems associated with high dimensionality and parameter tuning. Hybrid clustering assumes a set of cascaded clustering algorithms that cooperate with the goal of refining the clustering solutions produced by a former clustering algorithm(s) or to reduce the size of the input representatives to the next level of the cascaded model. Hybrid clustering does not allow synchronous execution of the clustering algorithms; instead one or more of the clustering algorithms remain idle until another algorithm finishes its clustering.

One way to enable concurrent implementation of the multiple clustering algorithms and benefit from each other with better performance synchronously is by using cooperative clustering. The cooperative clustering model proposed in this paper is mainly based on four components (1) co-occurred sub-clusters, (2) histogram representation of the pair-wise similarities within the sub-clusters, (3) cooperative contingency graph, and (4) coherent merging of histograms. These components are developed to obtain a cooperative model that is capable of clustering data

[*] Corresponding author.
E-mail addresses: rkashef@pami.uwaterloo.ca (R. Kashef), mkamel@pami.uwaterloo.ca (M.S. Kamel).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.12.018

with better quality than that of the adopted individual techniques.

The rest of this paper is organized as follows: in Section 2, related work to data clustering is given. The proposed cooperative clustering model and its complexity are presented in Section 3. The developed overall weighted similarity ratio (*OWSR*) measure and the *Scatter F-measure* are presented in Sections 4 and 5, respectively. The scalability of the cooperative model in terms of number of clustering techniques is discussed in Section 6. Section 7 provides a formulation of some external and internal quality measures to assess the clustering quality. Experimental results are presented and discussed in Section 8. Finally, we draw some conclusions and outline future work in Section 9.

2. Related work and background

Jain and Murty [1] give a comprehensive account of clustering algorithms. Most clustering algorithms can be classified into two groups: *Hierarchical* and *Partitional* clustering. The hierarchical techniques produce a nested sequence of partitions, with a single, all-inclusive cluster at the top and single clusters of individual objects at the bottom (leaf nodes) (divisive hierarchical clustering) or a set of singleton clusters at the top and one single partition at the bottom (agglomerative hierarchical clustering). The *Partitional* clustering approaches partition a collection of objects into a set of groups, so as to maximize the quality of clustering. Some *hierarchical* and *Partitional* techniques that are employed in the experimental results for a comparison purpose are discussed in the following sub-sections.

2.1. K-means (KM) clustering

k-means (KM) clustering [5] selects k objects randomly in the dataset X as initial seeds for the cluster's centroids, and then assigns each object $\boldsymbol{x}$ to the closest centroid c_i, i=1,2,..k. The new centroids are generated for each cluster by calculating the mean of the objects set assigned to each cluster. The iterative KM minimizes an *objective function*, in this case a squared error function defined as the sum of distances of the n data points $\boldsymbol{x}_j$, j=1,..,n, from their respective cluster centers c_i, i=1,..,k.

2.2. Bisecting k-means (BKM) clustering

Bisecting k-means (BKM) [7] begins with the whole dataset as one cluster. At each step, one cluster is selected and bisected into two partitions using the basic k-means [5] algorithm. This process continues until the desired number of clusters is obtained or some other specified stopping condition is reached. The bisecting approach is very attractive in many applications such as document-retrieval/indexing problems and gene expression analysis [11].

2.3. Partitioning around medoids (PAM) clustering

Rather than calculating the mean of the objects in each cluster as in the k-means (KM) clustering, the partition around medoids (PAM) algorithm [27] chooses a representative object, or medoid, for each cluster *at each iteration*. Medoids for each cluster are calculated by finding an object m_i within the cluster that minimizes the objective function defined as the sum of distances of all objects within the clusters to the cluster medoid. PAM has the advantage of its robustness to noisy data and outliers compared to k-means. PAM works well for small datasets but not for large datasets. The authors of [27] also present clustering large applications (CLARA), which draws one or more random samples from the whole data set and runs PAM on the samples. Ng and Han [28] propose "clustering large applications" based on Randomized Search (CLARANS) as an extension to PAM. Although CLARA and CLARANS are more scalable than PAM, they are inefficient for disk-resident datasets as they require multiple scans of the entire dataset and also a good clustering of a sample does not mean good clustering for the whole dataset.

3. The cooperative clustering (CC) model

The cooperative clustering (CC) model is mainly based on a cooperative methodology utilizing multiple clustering algorithms with the goal of achieving better clustering quality than individual approaches. The cooperative model takes the dataset and a set of clustering algorithms as inputs. Each clustering algorithm generates a set of k clusters. The cooperative model employs an agreement strategy between the multiple clustering algorithms to find the set of intersections between the different clusterings informs of sub-clusters. The extracted sub-clusters are then represented by similarity histograms; each sub-cluster next becomes a node in a cooperative contingency graph (*CCG*). Edges of the *CCG* are weighted by a cohesiveness factor for merging two sub-clusters into one cluster. Finally, a coherent merging of sub-clusters is performed to obtain the expected number of clusters. Fig. 1 illustrates the different components of the cooperative clustering model.

3.1. Inputs

The inputs are the data to be clustered and c clustering algorithms A_i, i=1,..,c, with their parameters. The dataset of d-dimensional vectors represented by a $n \times d$ matrix X={$\boldsymbol{x_i}$},i=1,..n, where n is the number of objects and the row vector $\boldsymbol{x_i}$ represents the ith object. The pair-wise similarities between objects are stored in a two dimensional $n \times n$ similarity (or distance) matrix, *SM*. The similarity matrix is a symmetric matrix, so we store only ($n \times (n–1)/2$) elements. The cosine similarity (used by [29]) is adopted to calculate the similarity between objects, such that $Sim(\boldsymbol{x}, \boldsymbol{y}) \in [-1,1]$, where $\boldsymbol{x}$ and $\boldsymbol{y} \in X$.

The cooperative clustering model relies on four main components: co-occurred sub-clusters, similarity histograms, a cooperative contingency graph and a coherent merging procedure of the histograms. Each of those components is discussed in the following sub-sections.

3.2. Generation of sub-clusters

In general, let A={A_1, A_2,.., A_c} be a set of c clustering techniques in the model. Assume $\{S^{A_i}(k) = \{S_j^{A_i},\ 0 \le j \le k-1\}$ is the set of k clusters generated by clustering technique A_i. We assume that the number of clusters, k, is the same for each clustering algorithm. For each object $\boldsymbol{x} \in X$, a cluster identifier, $mem(\boldsymbol{x})|_{A_i}$is assigned by each clustering algorithm A$_i$ such that $mem(\boldsymbol{x})|_{A_i} \in \{0,1,..,k-1\}$. In order to find the co-occurrence of objects between the multiple c clusterings, a new set of disjoint sub-clusters Sb is generated. The maximum number of disjoint sub-clusters (n_{sb}) is k^c. In order to find the association of objects in the corresponding set of sub-clusters, a new sub-cluster membership value is assigned to each object. This clusterings-mapping recognizes the set of disjoint sub-clusters $Sb = \{Sb_i\}_{i=0}^{n_{sb}-1}$, generated by the intersection of the c clusterings. Thus, the underlying model indicates the agreement between the various clustering techniques on clustering the data into a set of clusters. The new cooperative sub-cluster

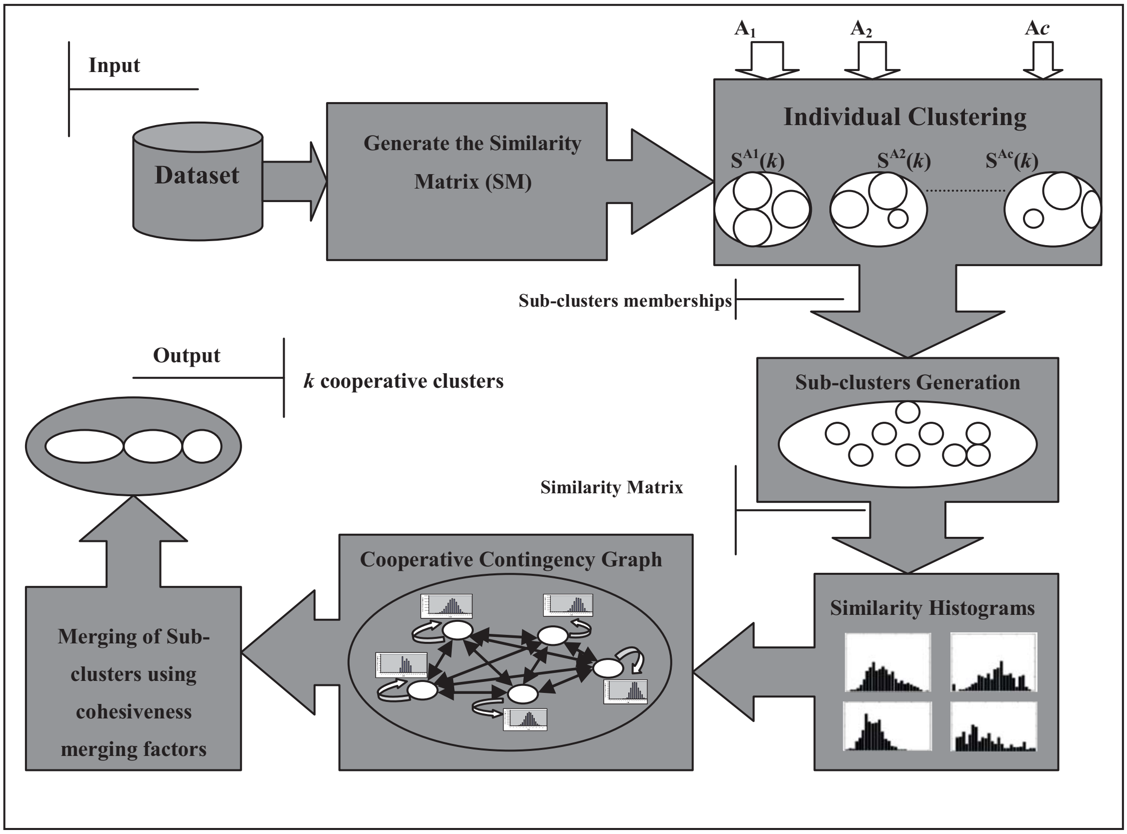

Fig. 1. The cooperative clustering model.

membership is defined as

$$mem(\boldsymbol{x})|_{A_1,A_2,\ldots,A_c} = mem(\boldsymbol{x})|_{A_1} + mem(\boldsymbol{x})|_{A_2} * k + mem(\boldsymbol{x})|_{A_3} * k^2 + \ldots + mem(\boldsymbol{x})|_{A_c} * k^{c-1} \tag{1}$$

Definition. For any two objects $\boldsymbol{x}$ and $\boldsymbol{y} \in X$, if $mem(\boldsymbol{x})$=$mem(\boldsymbol{y})$, then $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to the same cluster (or sub-cluster) where $mem(\boldsymbol{x})$ and $mem(\boldsymbol{y}) \in \{0,1,..,k-1\}$

Each sub-cluster is then represented by a concise statistical representation called *Similarity Histogram* used by [9] and [10]. A similarity histogram *H* is a concise statistical representation of the set of pair-wise similarities distribution in a collection of objects. The number of bins in the histogram corresponds to fixed similarity value intervals. Each bin contains the count of pair-wise similarities in the corresponding interval. In this paper, the similarity between any pair of objects is calculated using the widely accepted *cosine* coefficient. The similarity histogram in our model is built over the interval $[-1,1]$ with fixed size of bins, *BinSize*. The number of bins in the histogram is *NumBins* (a user input parameter), thus *BinSize* equals 2/*NumBins*. A coherent cluster has high pair-wise similarities. For a fixed bin size, *BinSize*, the $binId^{th}$ bin in the histogram contains the count of similarities that fall in the interval $[(binId-(NumBins/2))*\ BinSize, (binId-(NumBins/2)) * BinSize + BinSize]$. The first bin (i.e. bin with index=0) also contains similarities equal to -1.

3.3. Cooperative contingency graph

The cooperative clustering model is primarily based on the construction of the cooperative contingency graph (*CCG*).

Definition. The *CCG* is an undirected graph *G*={*Sb*, *E*} where the co-occurred sub-clusters are represented as vertices *Sb* of the graph. The relationships among sub-clusters are represented by the set *E*.

The quality of merging two sub-clusters is calculated by the coherency of merging the corresponding histograms. We assume the number of bins is the same in each sub-cluster's histogram. The process of merging two sub-clusters produces a new histogram. This histogram is constructed by adding the corresponding counts for each bin of the two merged histograms, and also by adding the additional count of the object pair similarities that are obtained during the merging of the two sub-clusters that were not calculated in each individual histogram. The new histogram is constructed as

$$H_{ij}(bin) = \begin{pmatrix} (H_i(bin) + H_j(bin) + |Sim(\boldsymbol{x},\boldsymbol{y})|), \forall \boldsymbol{x} \in Sb_i, \boldsymbol{y} \in Sb_j, bin = 0, 1, \ldots, NumBins-1) \\ Such\ that\{((bin-(NumBins/2)) * BinSize) < Sim(\boldsymbol{x},\boldsymbol{y}) \le ((bin-(NumBins/2)) * BinSize + BinSize)\} \end{pmatrix} \tag{2}$$

where H_{ij} is the histogram of the new cluster and $H_i(bin)$ is the count of similarities in the bin^{th} bin of the similarity histogram H_i. $|Sim(\boldsymbol{x},\boldsymbol{y})|$ refers to the number of the additional pair-wise similarities due to the merging.

A coherent will have a similarity frequency distribution that is skewed to the right while a loose cluster will have a frequency distribution that is skewed to the left. Each edge in the *CCG* graph is assigned a weighting factor. This factor represents the coherency (quality) of merging two sub-clusters into a new coherent cluster. We will refer to this factor as the *merging cohesiveness factor* (*mcf*) that is the ratio of the count of similarities weighted by the bin similarity above a certain similarity threshold δ to the total count of similarities in the new merged histogram. Let $|Sb_i|$ and $|Sb_j|$ be the number of objects in sub-clusters Sb_i, Sb_i, respectively. The number of similarities for merging the two sub-clusters together is $n_{sim}(Sb_i,Sb_j)=(|Sb_i|+|Sb_j|)*(|Sb_i|+|Sb_j|-1)/2$. The *mcf* ($Sb_i$, Sb_j) between any two sub-clusters Sb_i, Sb_j is calculated by the following formula:

$$mcf(Sb_i,Sb_j)=\frac{\sum_{bin=binThreshold}^{numBins-1}(((bin*binSize)-1+(binSize/2))*H_{ij}(bin))}{n_{Sim}(Sb_i,Sb_j)} \tag{3}$$

where *binThreshold* is the bin corresponding to the similarity threshold δ. The higher the *mcf*, the more coherent the new generated cluster. The *CCG* is illustrated in Fig. 2 and the algorithm for constructing the *CCG* graph using sub-clusters and histograms is illustrated in Fig. 3.

3.4. Coherent merging of sub-clusters

The cooperative clustering model CC(A_1,A_2,…,A_c) is comprised of two main phases *Phase* 1 and *Phase* 2. The first phase includes building the *CCG* graph and associating edges with the corresponding cohesiveness factors. The second phase includes attaining the same number of clusters *k* from the set of n_{sb} sub-clusters as the original designed clustering problem through merging of sub-clusters within the *CCG*. The best sub-clusters (most similar sub-clusters) for merging are defined as those that have maximal *mcf* value. Thus, the two most similar sub-clusters are merged first into a new cluster; i.e. cluster with better homogeneity than the two sub-clusters; and then both the vertices and edges in the *CCG* are updated based on the new added cluster. This step is repeated until the desired number of clusters *k* is reached. The multi-level cooperative model is described in Fig. 4.

Since clustering is unsupervised classification of objects, the number of clusters is not known. Therefore in the cooperative clustering model both *Phase* 1 and *Phase* 2 are repeated for different numbers of clusters $l\geq 2$. The cooperative model reveals a homogenous clustering solution at number of clusters $l=k$ with the maximum quality value (measure by relative or internal clustering quality measures). In the experimental results, we rely on the *SI* index (as internal quality measure) defined in Section 7 to assess finding the proper number of clusters (*k*). Thus the proper number of clusters is obtained as the value of *k* that corresponds to the lowest value of the separation index (where lower values of the SI index indicate better clustering quality as illustrated in Section 7). In Fig. 4, if external information about the dataset is given (i.e. class labels) then the CC model will be performed at the given number of clusters *k* such that $k^{initial}=k^{final}=k$.

3.5. Complexity analysis

All the basic operations are assumed to take the same time. Assume $T^{A_1}(l)$, $T^{A_2}(l),\ldots,T^{A_c}(l)$ are the computational time complexities of the clustering techniques A_1, A_2,…,A_c, respectively, for a given number of clusters l=2,3,…,k. The analysis of the cooperative model can be divided into two stages based on the processing of each individual phase of the cooperative model as follows:

- *Phase* 1: Complexity of constructing the contingency cooperative graph, (T^{Phase1})
  (a) Finding the set of sub-clusters takes *n* operations where *n* is the total number of objects.
  (b) Building a histogram of a sub-cluster Sb_i needs $(|Sb_i|*(|Sb_i|-1))/2$ operations. Thus $\sum_{i=0}^{n_{sb}-1}|Sb_i|*(|Sb_i|-1)/2$ operations are required to construct the n_{sb} histograms, where $|Sb_i|$ is the size of the sub-cluster Sb_i.
  (c) Calculating the *mcf* for each pair of sub-clusters Sb_i and Sb_j in the *CCG* takes (*NumBins*-δ) +$|Sb_i|*|Sb_j|$ operations.
  (d) Thus *Phase* 1 is of order O($n+|Sb_i|^2$), $\forall i=0,1,...,n_{sb}-1$

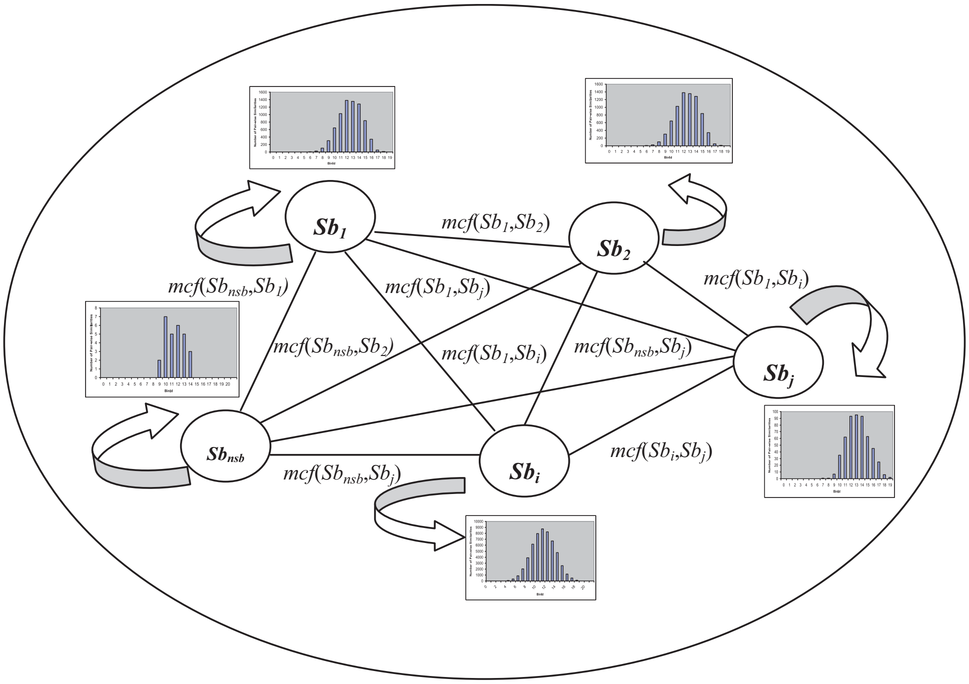

Fig. 2. The *CCG* Graph.

```
Algorithm: Build-CCG Graph ($S^{Ai}(k)=\{S_j^{Ai}, 0\le j \le k-1\}, SM, \delta, NumBins$)
Input: A set of $c$ clusterings, each clustering solution consists of $k$ clusters $S_0,S_1,..,S_{k-1}$,
       similarity Matrix $SM$, similarity threshold $\delta$, and number of bins, $NumBins$.
Output: The Cooperative Contingency Graph ($CCG$)
Initialization: $Sb=\{\}$, $n_{sb}=0$
Begin
    Step1: For all $x \in X$
      $mem(x)|_{A1, A2,.., Ac} = mem(x)|_{A1} + mem(x)|_{A2}*k + mem(x)|_{A3}*k^2 + \ldots + mem(x)|_{Ac}*k^{c-1}$
          Assign $index = mem(x)|_{A1, A2,.., Ac}$
          If $Sb_{index}$ is empty, then
              Create new sub-cluster $Sb_{index}$, insert $x$ to it, add $Sb_{index}$ to $Sb$,
              Increment $n_{sb}$ by one
          Else add $x$ directly to the sub-cluster $Sb_{index}$
          End
PS: The set $Sb$ contains $n_{sb}$ disjoint sub-clusters, $Sb=\{Sb_i, i=0,1,\ldots,n_{sb}-1\}$
    Step2: For each sub-cluster $Sb_i \in Sb$
          Build-Histogram for each sub-cluster
          End
    Step3: Create the cooperative Contingency graph $CCG=G(Sb,E)$, where
          $Sb=\{Sb_i.i=1,..,n_{sb}\}$, $E=\{e_{ij}(Sb_i,Sb_j)\}$ where each edge $e_{ij}$ is assigned a weight
          $= mcf(Sb_i,Sb_j)$(Eq. (3))
Return CCG
End
```

Fig. 3. Building the cooperative contingency graph (*CCG*).

```
Algorithm: Multi-Level Cooperative Clustering: CC($X, SM, A_1, A_2,.., A_c, \zeta, \delta, NumBins$)
Input: Dataset X, similarity matrix $SM$, $c$ clustering algorithms, $A_1,A_2,..,A_c$, input parameters
       $\zeta=\{\zeta_i\}$ for each algorithm $A_i$, similarity threshold $\delta$, and number of bins, $NumBins$.
Output: Set of Cooperative Clusters, $S^{Cooperative}(k)=\{S_0,S_1,..,S_{k-1}\}$
Initializations: Let $k^{initial}=2$
Begin
  For number of partitions $l=k^{initial}$ to $k^{final}$ (Non-cooperative Clustering Step)
    Phase 1:
      Step1: Synchronously generate the $c$ clusterings sets $S^{A1}(l)$, $S^{A2}(l)$ ,..,and $S^{Ac}(l)$
             where $S^{Ai}(l)= A_i(X, l, \zeta_i)=\{S_j^{Ai}, 0\le j \le l-1\}$
      Step2: Build_CCG($S^{Ai}(l)$, $SM$, $\delta$, $NumBins$) (Cooperation Step)
    Phase 2: Repeat (Merging Step)
          Step 1: Merge the two most similar sub-clusters into one cluster, i.e. two
                  sub-clusters with the highest $mcf$ in the graph and update the $CCG$.
          Step2: Reduce the number of sub-clusters $n_{sb}$ by one
        Until (number of sub-clusters $n_{sb}=l$)
  $S^{cooperative}(l)$= final set of the merged $l$ sub-clusters
  End
Return the final set of $k$ clusters $S^{cooperative}(k)$ with the maximum quality
End
```

Fig. 4. The multi-level cooperative clustering model.

The number of sub-clusters $n_{sb} \le l^c$, and the size of sub-clusters determine the cost of generating the *CCG* graph.

- *Phase* 2: Complexity of merging histograms, (T^{Phase2})
  (e) Finding the two most homogenous sub-clusters to be merged generate a new cluster S_i is of order O ($n_{sb}{}^2$) operations, $n_{sb} \le l^c$.
  (f) Updating the *CCG* with the new added cluster takes ($NumBins * \sum_{j=0}^{n_{sb}-3} |S_i| * |Sb_j|$) operations.
  (g) Thus *Phase* 2 is of order $O(n_{sb}^2 + |S_i| * |Sb_j|)\ \forall i,j = 1,\ldots,n_{sb} O(n^2)$.

The time complexity of the cooperative clustering (*CC*) model for *l* partitions is computed as

$$T^{CC}(l) = O(\max(T^{Ai}(l))) + T^{Phase1} + T^{Phase2} \tag{4}$$

The time complexity of the *CC* is based on the clustering algorithm with the maximum running time and the additional computational costs of both phases that is mainly based on the number of sub-clusters and the size of each sub-cluster, which is much lower than n^2.

4. Overall weighted similarity ratio (*OWSR*)

We developed a new measure called the overall weighted similarity ratio (*OWSR*) that monitors the quality of the set of sub-clusters defined by

$$SimRatio(Sb_i) = \left\{ \begin{array}{ll} \frac{\sum_{bin=BinThreshold}^{NumBins-1}(((bin * binSize) - 1 + (binSize/2)) * H_i(bin))}{|Sb_i| * (|Sb_i| - 1)/2} & \text{if } Sb_i| > 1 \\ 0 & \text{if } |Sb_i| = 1 \end{array} \right\} \tag{5}$$

where H_i is the histogram representation of the sub-cluster Sb_i, $|Sb_i|$ is the number of objects in the sub-cluster Sb_i, and *binThreshold* is the bin corresponding to the similarity threshold δ. The value of the $SimRatio(Sb_i)$ increases if objects within the sub-cluster Sb_i are of maximum similarity above the similarity threshold δ. Sub-clusters of only one object have the lowest similarity ratio. The *overall weighted similarity ratio* (*OWSR*) for a set of n_{sb} sub-clusters is calculated as the average of the similarity ratio of each sub-cluster weighted by the size of each sub-cluster.

$$Overall\ weighted\ similarity\ ratio(n_{sb}) = OWSR(n_{sb}) = \frac{\sum_{i=0}^{n_{sb}-1} SimRatio(Sb_i) * |Sb_i|}{n} \tag{6}$$

This measure is used to compare two partitions having different number of sub-clusters.

5. Scatter F-measure

The traditional *F-measure* (used by [12]) measures the difference between the original labeling of the dataset (i.e. class labels) and the resulting clustering of the data. The proposed *scatter F-measure* measures the diversity of the clustering solutions obtained from two clustering algorithms. Given two clustering algorithms A_1 and A_2, each algorithm generates a clustering set of k-clusters $S^{A1}(k) = \{S_i^{A1}, 0 \le i \le k-1\}$, and $S^{A2}(k) = \{S_j^{A2}, 0 \le j \le k-1\}$, respectively. Assume $|S_i^{A1}|$ is the number of objects in cluster $S_i^{A1}(S_i^{A1} \in S^{A1}(k))$, and $|S_j^{A2}|$ is the number of objects in cluster $S_j^{A2}(S_j^{A2} \in S^{A2}(k))$. The *F-score* of a cluster S_i^{A1} is defined as

$$F\text{-}score(S_i^{A_1}) = \max_j \frac{2 * n_{ij}}{|S_i^{A_1}| + |S_j^{A_2}|} \tag{7}$$

where n_{ij} is the number of objects of cluster S_i^{A1} that co-occurred in the cluster S_j^{A2}. With respect to cluster S_i^{A1} we consider the cluster with the highest value of *F-score* to be the cluster S_j^{A2} that is mapped to cluster S_i^{A1}, and that value becomes the score for cluster S_i^{A1}. The overall *scatter F-measure* for the clustering result of k clusters is the weighted average of the *F-score* for each cluster S_i^{A1}:

$$Scatter\ F\text{-}measure = \frac{\sum_{i=0}^{k-1}(|S_i^{A_1}| \times F\text{-}score(S_i^{A_1}))}{\sum_{i=0}^{k-1}|S_i^{A_1}|} \tag{8}$$

The higher the overall *Scatter F-measure*, the close solution both A_1 and A_2 generate due to the higher accuracy of the resulting clusters of A_2 mapping to the clusters generated by A_1. In cooperative clustering, we seek lower values of the *scatter F-measure* in order to obtain significant improvement in the clustering performance. If two clustering algorithms closely generate the same clustering solutions, then the generated set of sub-clusters will be the same as the original clusters. Thus, in turn no additional information is obtained within the set of sub-clusters. We rely on the *overall weighted similarity ratio* (*OWSR*) as an internal quality measure for evaluating the homogeneity of sub-clusters at different values of the *scatter F-measure*.

6. Scalability of the cooperative model

Let B be the clustering technique that will be added to the cooperative model (that contains c clustering algorithms). If the set of sub-clusters Sb remains the same, i.e. $\exists A_i \in \{A_1, A_2,.., A_c\}$ such that the *Scatter F-measure* between B and A_i is of maximum value, then the resulting c+1 cooperation is almost the same as the c cooperation. However, if adding B to the model generates a new set of sub-clusters with better homogeneity than the old sub-clusters then the new set of sub-clusters acts as incremental agreement between the c clustering techniques and the additional approach B. Thus adding the new technique B to the system was beneficial and it moved the clustering process into a more homogenous clustering process. The homogeneity is evaluated using the *OWSR* measure. In general, increasing the number of algorithms in the model will in turn increase the number of sub-clusters n_{sb}; the upper bound of number of sub-clusters is k^c, where k is number of clusters and c is number of clustering techniques. Thus if c is large enough then the number of the generated sub-clusters $n_{sb} \rightarrow n$, which extremely increases the computational complexity of the cooperative model. In this case, each sub-cluster will be a singleton sub-cluster with a maximum of one object and with a similarity ratio of value equals zero (Eq. (5)) then the quality of the sub-clusters will be of a minimum value. Thus after a specific value of c, c^*, the cooperative quality degrades rapidly. Then after c^*, no more techniques can be added to the model. The value of c^* was determined experimentally as will be illustrated in Section 8. Future work involves evaluating the value of c^* theoretically.

7. Clustering quality measures

The clustering results of any clustering algorithm should be evaluated using an informative quality measure(s) that reflects the "goodness" of the resulting clusters. External quality measures including *F-measure*, *entropy*, and *purity* (used by [10]) are used, based on a correct classification [30,31].

The *SI Index* [30] is used as internal quality measure, which does not require a prior classification about the objects. It is defined as the ratio of average within-cluster variance (cluster scatter) to the minimum pair-wise dissimilarity (in this paper we measured the pair-wise dissimilarity by the *cosine correlation* measure) between clusters:

$$SI(k) = \frac{\sum_{i=1}^{k}\sum_{\forall \boldsymbol{x}_j \in S_i} 1 - CosSim(\boldsymbol{x}_j, z_i)}{n * min_{r,s=1,\ldots,k, r \neq s}\{1 - CosSim(z_r, z_s)\}} \tag{9}$$

where z_i is the prototype of the cluster S_i. The smaller the *SI* the more separate the clusters. Thus, the smallest *SI* indeed indicates a valid optimal partition.

8. Experimental results

The cooperative clustering model has been evaluated by applying it to seven datasets: four gene expression datasets and three document datasets. The main measures of evaluation are the external and internal quality of the output clusters.

8.1. Datasets

Experiments were performed on gene expression and documents datasets with various characteristics, dimensions, and sizes. The gene expression datasets are *Leukemia*, *Yeast*, *Breast Cancer*, and *Serum* and the document datasets are *UW*, *SN*, and *Yahoo*.

8.1.1. Gene Expression datasets

Four gene expression datasets are used, the *Leukemia* dataset [32], *Yeast* gene expression dataset [33], *Breast Cancer* data set [34], and *Serum* dataset [35]. *Leukemia* (*Leuk*) dataset contains the expression of 999 genes along 38 samples. The *Yeast* cell cycle time series dataset contains 703 from 6218 genes using the same filtering and data normalization procedures of [37]. Based on the analysis conducted by Spellman et al. [38], five main clusters are generated. The *Breast Cancer* (*BC*) [34] contains 7129 gene expression and 49 tumors. The *Serum* dataset [35] is a time series gene expression dataset containing 12 time point expressions for about 500 genes. The classification model (i.e. class labels) for both the leukemia and the breast cancer datasets is discovered using the same approach as in [36]. More details of these datasets can be found in [10].

8.1.2. Document datasets

In this paper, we used the *Vector space model* (VSM) [39], which is the most common document representation model used in text mining; another document representations can be found in [9] and [40]. In VSM each document is represented by a feature vector $\boldsymbol{x}$ of dimensionality d, in the term space, $\boldsymbol{x}=[f_1, f_2 \ldots f_d]$. In the experimental analysis, the feature vector[1] combines the term frequency with the inverse document frequency (*TF-IDF*) as used in [9]. The document frequency df_i is the number of documents in a collection of n documents in which the term t_i occurs. A typical inverse document frequency (*idf*) factor of this type is given by $\log(n/df_i)$. Each feature f_i, $i=1,\ldots,d$, is calculated by weighting the term frequency tf_i, by the *idf*, Where tf_i, $i=1,\ldots,$ d is the term frequency in the document, or the number of occurrences of the term t_i in a document $\boldsymbol{x}$.

Three documents datasets are used; the *UW*, *SN*, and *Yahoo* document datasets. The words are tokenized in the three document datasets as in [9]. The *UW* dataset contains manually collected 314 documents from the University of Waterloo's[2] various web sites. The *UW* dataset was primarily used for the work presented in [9]. The *SN* dataset is a data set of 2371 metadata records collected from Canada's SchoolNet [3]website. The *Yahoo* dataset is a collection of Reuter's news articles from the Yahoo! News website, also was used by Boley et al. [8,41]. Table 1 summarizes the gene expression and documents datasets.

Table 1
Summary of the datasets.

Dataset	N	k	d
Leukemia (Leuk)	999	3	38
Yeast	703	5	73
Breast Cancer (BC)	7129	4	49
Serum	517	*No external classification*	12
UW	314	10	15,134
SN	2,371	17	7,167
Yahoo	2,340	20	28,298

8.2. Significance of results

To back the claim of clustering quality improvement, statistical significance testing is presented here, where by the average values of a variable taken by any two approaches are compared. This is a comparison of two-means test that enables us to calculate the confidence intervals for the difference between the two means. Assume q (e.g. *F-measure* or *Entropy* or *Purity* or *SI*) is the quality measure used for comparison between any two clustering techniques A_1 and A_2. Let q_1 and q_2 be the two samples of the quality measure q for the clustering results of both A_1 and A_2, respectively. Our null hypothesis (which we will argue to be rejected in favor of the alternate hypothesis) is that the average values of q for A_1 and A_2 are the same (i.e. no significance difference).

$$H_0 : \overline{q}_1 = \overline{q}_2 \text{ (No significant improvement in } q \text{ using } A_1) \tag{10}$$

where $\overline{q}_1$ is the average q value for A_1 clustering over n_1 samples, and $\overline{q}_2$ is the corresponding average value of q for A_2 clustering over n_2 samples. The alternative hypothesis is

$$H_1 : \overline{q}_1 \neq \overline{q}_2 \text{ (better improvement in } q \text{ using } A_1) \tag{11}$$

For directional difference, for example *F-measure*, the Null hypothesis H_0 is $\overline{F}_1 = \overline{F}_2$ and the alternative hypothesis is $\overline{F}_1 > \overline{F}_2$ where $\overline{F}_1$ is the average *F-measure* of the cooperative clustering over all runs, and $\overline{F}_2$ is the corresponding average *F-measure* obtained from the non-cooperative algorithm. Since the actual underlying means and standard deviations are not known, we are going to use a two-sample *t*-statistic, in which the population standard deviations are estimated by the calculated standard deviations sd_1 and sd_2 from the samples. The *t*-statistic is given by

$$\frac{t = (\overline{q}_1 - \overline{q}_2)}{\sqrt{\dfrac{sd_1^2}{n_1} + \dfrac{sd_2^2}{n_2}}} \tag{12}$$

where sd_1 and sd_2 are the calculated standard deviations, and n_1 and n_2 are the sample sizes from the two populations. The confidence interval of the difference between the two means at a confidence level α is given by

$$(\overline{q}_1 - \overline{q}_2) \pm t^{critical} \sqrt{\frac{sd_1^2}{n_1} + \frac{sd_2^2}{n_2}} \tag{13}$$

where $t^{critical}$ is the upper (1–α)/2 critical value for the t distribution with *df* degree of freedom which equals (n_1+n_2–2). To compute a (1–α)% confidence interval (usually 95%) for the difference between the two means:

1 First, find the $t^{critcial}$ value from the *t*-distribution table[4] at degree of freedom *df* and confidence interval α;

[1] Obviously the dimensionality of the feature vector is always very high, in the range of hundreds and sometimes thousands.

[2] http://www.uwaterloo.ca

[3] http://www.schoolnet.ca

[4] http://www.medcalc.be/manual/t-distribution.php

2 If the calculated t value $< t^{critcial}$ then the Null hypothesis H_0 is accepted otherwise the "significance difference" hypothesis H_1 is accepted and H_0 is rejected.

In our experiments, we obtained 20 runs of each algorithm. Thus at df=38 and confidence interval 95%, the critical value of t (cut off), $t^{critcial}$ equals 2.024. Thus, in each experiment we evaluate the t-test value of the evaluation measure and compare this value to the assigned critical t-value at 95% confidence interval.

8.3. Quality measures

In order to evaluate the quality of the clustering, we adopted four quality measures that are widely used in the data clustering literature, *F-measure*, *Entropy*, *Purity*, and *SI* (defined in Section 7). Basically we would like to maximize the *F-measure*, minimize the *Entropy* of clusters, maximize the *Purity* of solutions, and minimize the separation index of the obtained clusters to achieve high quality clustering. By using both external and internal measures we have confidence that our evaluation of both the cooperative and non-cooperative approaches will be justified.

8.4. Cooperative clustering performance evaluation

In this section, three well-known clustering algorithms, KM, BKM, and PAM are invoked in the cooperative model. Initially, let c=2 (i.e. binary cooperation), we will refer to the cooperation between any two algorithms A_1 and A_2, where $A_1, A_2 \in$ {KM, BKM, PAM} as CC($A1$, $A2$).

8.4.1. Clustering quality

Tables 2–7 present the average performance of 20 runs of the individual KM, BKM, and PAM algorithms as well the three cooperative models, CC(KM,BKM), CC(KM,PAM), and CC(BKM,PAM). The similarity threshold δ is in [0.1, 0.25] and [0.2, 0.3] for the gene expression datasets and the document datasets, respectively.

Assume q is any measure of the quality measures described above. For the non-cooperative KM, BKM, and PAM, each cell contains two entities:

- $\overline{q}$: The average value of the variable q over the 20 runs;
- $\pm sd$: The standard deviation of the variable q.

For any cooperative model CC(A_1, A_2), each cell can be described by the 5-tuple ($\overline{q}$, sd, t_1, t_2, $+q\%$); each element of the 5-tuple is described as

- $\overline{q}$, sd: The average and the standard deviation of the variable q;
- t_1 and t_2: the t-test values between the results of CC(A_1, A_2) and A_1, and the results of CC(A_1, A_2) and A_2, respectively;
- $+q\%$: the percentage in improvement in q using the cooperative model CC(A_1, A_2) compared to the value of q that is calculated by A_1 and A_2.

Table 2
Performance evaluation of the cooperative and non-cooperative approaches [*Leuk*].

(k=3)	*F-measure*	*Entropy*	*Purity*	*SI*
KM	0.8366 ± (0.021)	0.4086 ± (0.032)	0.8328 ± (0.036)	0.3921 ± (0.034)
BKM	0.8073 ± (0.022)	0.4738 ± (0.011)	0.8048 ± (0.028)	0.4502 ± (0.025)
PAM	0.8754 ± (0.033)	0.3971 ± (0.026)	0.8478 ± (0.020)	0.3615 ± (0.024)
CC(KM,BKM)	**0.9375** ± (0.019)	**0.3109** ± (0.026)	**0.9096** ± (0.019)	**0.2647** ± (0.018)
	t_1=15.93	t_1=10.59	t_1=8.43	t_1=15.16
	t_2=20.03	t_2=25.81	t_2=13.85	t_2=26.23
Improvement (%)	**(+12%)**	**(+24%)**	**(+9%)**	**(+32%)**
CC(KM,PAM)	**0.9485** ± (0.013)	**0.2966** ± (0.017)	**0.9381** ± (0.011)	**0.2385** ± (0.017)
	t_1=20.26	t_1=13.82	t_1=12.51	t_1=18.50
	t_2=9.21	t_2=14.46	t_2=17.69	t_2=18.70
Improvement (%)	**(+8%)**	**(+25%)**	**(+11%)**	**(+34%)**
CC(BKM,PAM)	**0.9630** ± (0.010)	**0.2673** ± (0.028)	**0.9565** ± (0.035)	**0.2071** ± (0.029)
	t_1=24.65	t_1=30.69	t_1=15.13	t_1=27.91
	t_2=11.36	t_2=15.19	t_2=12.05	t_2=18.34
Improvement (%)	**(+10%)**	**(+32%)**	**(+13%)**	**(+42%)**

Table 3
Performance evaluation of the cooperative and non-cooperative approaches [*Yeast*].

(k=5)	*F-measure*	*Entropy*	*Purity*	*SI*
KM	0.6301 ± (0.040)	0.4351 ± (0.032)	0.6715 ± (0.031)	1.6303 ± (0.206)
BKM	0.6784 ± (0.011)	0.4136 ± (0.024)	0.7496 ± (0.020)	1.2991 ± (0.185)
PAM	0.6922 ± (0.035)	0.4032 ± (0.031)	0.7667 ± (0.028)	1.0433 ± (0.176)
CC(KM,BKM)	**0.8175** ± (0.041)	**0.2748** ± (0.022)	**0.8946** ± (0.019)	**0.6162** ± (0.092)
	t_1=14.63	t_1=18.46	t_1=27.44	t_1=20.10
	t_2=14.65	t_2=19.07	t_2=23.50	t_2=15.38
Improvement (%)	**(+21%)**	**(+34%)**	**(+19%)**	**(+52%)**
CC(KM,PAM)	**0.8586** ± (0.026)	**0.2463** ± (0.020)	**0.9326** ± (0.017)	**0.5815** ± (0.066)
	t_1=21.42	t_1=22.37	t_1=33.02	t_1=21.68
	t_2=17.07	t_2=19.02	t_2=22.64	t_2=10.98
Improvement (%)	**(+24%)**	**(+39%)**	**(+22%)**	**(+44%)**
CC(BKM,PAM)	**0.7812** ± (0.012)	**0.3047** ± (0.030)	**0.8792** ± (0.011)	**0.7757** ± (0.057)
	t_1=28.24	t_1=12.67	t_1=25.39	t_1=12.09
	t_2=10.75	t_2=10.21	t_2=16.72	t_2=6.46
Improvement (%)	**(+12%)**	**(+24%)**	**(+14%)**	**(+26%)**

Table 4
Performance evaluation of the cooperative and non-cooperative approaches [*BC*].

(k=4)	*F-measure*	*Entropy*	*Purity*	*SI*
KM	0.4271 ± (0.011)	0.8031 ± (0.041)	0.5734 ± (0.018)	0.7578 ± (0.024)
BKM	0.4355 ± (0.020)	0.7948 ± (0.032)	0.5825 ± (0.016)	0.7363 ± (0.018)
PAM	0.5012 ± (0.023)	0.7114 ± (0.022)	0.6107 ± (0.017)	0.6930 ± (0.033)
CC(KM,BKM)	**0.4915** ± (0.016)	**0.7042** ± (0.015)	**0.6606** ± (0.020)	**0.6998** ± (0.015)
	t_1=24.01	t_1=11.46	t_1=14.48	t_1=9.16
	t_2=10.02	t_2=10.35	t_2=13.64	t_2=6.96
Improvement (%)	**(+12%)**	**(+11%)**	**(+13%)**	**(+5%)**
CC(KM,PAM)	**0.5935** ± (0.020)	**0.6102** ± (0.012)	**0.7294** ± (0.013)	**0.5418** ± (0.019)
	t_1=32.60	t_1=23.35	t_1=31.43	t_1=31.55
	t_2=13.54	t_2=20.19	t_2=32.55	t_2=17.75
Improvement (%)	**(+18%)**	**(+14%)**	**(+19%)**	**(+22%)**
CC(BKM,PAM)	**0.6402** ± (0.017)	**0.5188** ± (0.029)	**0.7637** ± (0.012)	**0.4943** ± (0.027)
	t_1=34.87	t_1=25.31	t_1=40.51	t_1=33.35
	t_2=21.73	t_2=23.66	t_2=32.87	t_2=20.84
Improvement (%)	**(+27%)**	**(+26%)**	**(+25%)**	**(+29%)**

Table 5
Performance evaluation of the cooperative and non-cooperative approaches [*UW*].

(k=10)	*F-measure*	*Entropy*	*Purity*	*SI*
KM	0.6988 ± (0.027)	0.2579 ± (0.022)	0.6879 ± (0.031)	1.6921 ± (0.152)
BKM	0.7520 ± (0.031)	0.2281 ± (0.024)	0.7334 ± (0.024)	1.3772 ± (0.140)
PAM	0.6463 ± (0.041)	0.3592 ± (0.013)	0.6490 ± (0.033)	3.1932 ± (0.543)
CC(KM,BKM)	**0.8387** ± (0.018)	**0.1819** ± (0.011)	**0.8420** ± (0.015)	**1.0559** ± (0.019)
	t_1=19.28	t_1=13.82	t_1=20.01	t_1=18.57
	t_2=10.81	t_2=7.82	t_2=17.16	t_2=10.17
Improvement (%)	**(+12%)**	**(+20%)**	**(+15%)**	**(+23%)**
CC(KM,PAM)	**0.8672** ± (0.013)	**0.1646** ± (0.028)	**0.8734** ± (0.029)	**0.9678** ± (0.112)
	t_1=25.13	t_1=11.71	t_1=19.54	t_1=17.32
	t_2=22.96	t_2=28.19	t_2=22.84	t_2=17.97
Improvement (%)	**(+24%)**	**(+36%)**	**(+27%)**	**(+43%)**
CC(BKM,PAM)	**0.8746** ± (0.024)	**0.1513** ± (0.035)	**0.8819** ± (0.017)	**0.8707** ± (0.185)
	t_1=13.99	t_1=8.09	t_1=22.58	t_1=9.76
	t_2=21.49	t_2=24.90	t_2=28.05	t_2=18.10
Improvement (%)	**(+16%)**	**(+33%)**	**(+20%)**	**(+37%)**

Table 6
Performance evaluation of the cooperative and non-cooperative approaches [*SN*].

(k=17)	*F-measure*	*Entropy*	*Purity*	*SI*
KM	0.4927 ± (0.029)	0.3787 ± (0.018)	0.6449 ± (0.025)	1.7441 ± (0.215)
BKM	0.5281 ± (0.037)	0.3585 ± (0.022)	0.6867 ± (0.024)	1.2876 ± (0.146)
PAM	0.3412 ± (0.034)	0.5831 ± (0.072)	0.4787 ± (0.042)	4.2001 ± (0.833)
CC(KM,BKM)	**0.5823** ± (0.013)	**0.3374** ± (0.016)	**0.7661** ± (0.023)	**1.0955** ± (0.086)
	t_1=12.61	t_1=7.67	t_1=15.95	t_1=12.52
	t_2=6.18	t_2=3.45	t_2=10.45	t_2=4.96
Improvement (%)	**(+10%)**	**(+6%)**	**(+12%)**	**(+15%)**
CC(KM,PAM)	**0.6184** ± (0.015)	**0.3244** ± (0.037)	**0.7903** ± (0.043)	**0.8531** ± (0.065)
	t_1=17.22	t_1=5.90	t_1=13.07	t_1=17.74
	t_2=33.36	t_2=14.29	t_2=23.21	t_2=17.91
Improvement (%)	**(+25%)**	**(+14%)**	**(+23%)**	**(+51%)**
CC(BKM,PAM)	**0.6436** ± (0.022)	**0.3153** ± (0.031)	**0.8457** ± (0.029)	**0.7517** ± (0.039)
	t_1=11.99	t_1=5.08	t_1=18.89	t_1=15.85
	t_2=33.39	t_2=15.28	t_2=32.15	t_2=18.49
Improvement (%)	**(+22%)**	**(+12%)**	**(+23%)**	**(+41%)**

In each table, t_1 and $t_2 > 2.024$, and therefore the Null hypothesis H_0 is rejected and the obtained results from the cooperative models are significantly different with better performance than those obtained using the adopted individual approaches. The cooperation between KM and BKM, CC(KM,BKM), achieves improvement up to 21% in *F-measure*, up to 34% in *Entropy*, up to 19% in *Purity*, and up to 52% in *SI index* for the *Yeast* dataset. The cooperative model, CC(KM,PAM) achieves improvement up to 25% in *F-measure* (*SN* dataset), up to 39% in *Entropy* (*Yeast* dataset), up to 27% in *Purity* (*UW* dataset), and up to 51% in *SI* index for the *SN* dataset. Finally, CC(BKM,PAM) achieves improvement up to 27% in *F-measure* (*Breast Cancer* dataset), up to 33% in *Entropy* (*UW* dataset), up to 25% in *Purity* (*Breast Cancer* dataset), and up to 42% in *SI* index for the *Leukemia* dataset. It can be shown that cooperative clustering produces clustering solutions with higher values for both *F-measure* and *Purity* and lower values for *Entropy* and *SI* index than those of the individual algorithms. The main reason for this improvement in the

Table 7
Performance evaluation of the cooperative and non-cooperative approaches [*Yahoo*].

(k=20)	*F-measure*	*Entropy*	*Purity*	*SI*
KM	0.4585 ± (0.011)	0.3815 ± (0.025)	0.6192 ± (0.024)	2.3246 ± (0.134)
BKM	0.5501 ± (0.031)	0.3128 ± (0.029)	0.7171 ± (0.017)	1.6641 ± (0.089)
PAM	0.4476 ± (0.016)	0.4876 ± (0.034)	0.5381 ± (0.053)	2.5138 ± (0.206)
CC(KM,BKM)	**0.6619** ± (0.023)	**0.2397** ± (0.014)	**0.8078** ± (0.011)	**1.1272** ± (0.055)
Improvement (%)	t_1=35.67	t_1=22.13	t_1=31.95	t_1=36.97
	t_2=12.95	t_2=10.15	t_2=20.03	t_2=22.95
	(+20%)	**(+23%)**	**(+13%)**	**(+32%)**
CC(KM,PAM)	**0.4794** ± (0.010)	**0.3674** ± (0.009)	**0.6474** ± (0.022)	**2.1764** ± (0.026)
	t_1=6.29	t_1=2.373	t_1=3.87	t_1=4.86
	t_2=7.54	t_2=15.28	t_2=8.51	t_2=7.27
Improvement (%)	**(+5%)**	**(+4%)**	**(+5%)**	**(+6%)**
CC(BKM,PAM)	**0.6162** ± (0.018)	**0.2687** ± (0.028)	**0.7788** ± (0.012)	**1.3695** ± (0.033)
	t_1=8.24	t_1=4.89	t_1=13.26	t_1=13.88
	t_2=31.30	t_2=22.23	t_2=19.80	t_2=16.76
	(+12%)	**(+14%)**	**(+9%)**	**(+18%)**

Table 8
Scatter F-measure and quality of clusters [*Yeast*].

k=5	CC(KM,BKM)	CC(KM,PAM)	CC(BKM,PAM)
Scatter F-measure	0.6956	0.5309	0.7363
# Sub-clusters	15	18	10
Quality of Sub-clusters (*OWSR*)↑	0.2871	0.2914	0.2755
SI↓	0.6162	0.5815	0.7757

Table 9
Scatter F-measure and quality of clusters [*Breast Cancer*].

k=4	CC(KM,BKM)	CC(KM,PAM)	CC(BKM,PAM)
Scattering-F-measure	0.8432	0.6306	0.6175
# Sub-clusters	10	14	15
Quality of Sub-clusters (*OWSR*)↑	0.7543	0.8623	0.8955
SI↓	0.6998	0.5418	0.4943

Table 10
Scatter F-measure and quality of clusters [*UW*].

k=10	CC(KM,BKM)	CC(KM,PAM)	CC(BKM,PAM)
Scattering-F-measure	0.6672	0.5655	0.4618
# Sub-clusters	28	32	44
Quality of Sub-clusters (*OWSR*)↑	0.2441	0.2657	0.2932
SI↓	1.0559	0.9678	0.8707

Table 11
Scatter F-measure and quality of clusters [*SN*].

k=17	CC(KM,BKM)	CC(KM,PAM)	CC(BKM,PAM)
Scattering-F-measure	0.6109	0.4835	0.4312
# Sub-clusters	139	208	214
Quality of Sub-clusters (*OWSR*)↑	0.6473	0.7647	0.7887
SI↓	1.0955	0.8531	0.7517

Table 12
Scatter F-measure and quality of clusters [*Yahoo*].

k=20	CC(KM,BKM)	CC(KM,PAM)	CC(BKM,PAM)
Scattering-F-measure	0.4075	0.7612	0.4563
# Sub-clusters	231	131	225
Quality of Sub-clusters (*OWSR*)↑	0.7788	0.59554	0.7601
SI↓	1.1272	2.1764	1.3695

clustering quality is that, each of the cooperative models takes the intersection of the individual clusterings and obtains new clusterings with maximum Intra cluster homogeneity and maximum Inter-cluster separation using both the notion of similarity histograms and cooperative merging.

8.4.2. *Scatter-F-measure evaluation*

In this sub-section, we use the *Scatter F-measure* (defined in Section 5) as a measure of diversity between the clustering algorithms. If there is no scattering between the clustering solutions of the individual clustering algorithms (i.e. they generate the same solution), it would lead to the same set of sub-clusters as the original set of clusters. Then the cooperative merged clusters will be the same as those of both of the original non-cooperative approaches. On the other hand, when two clustering approaches generate two different clustering solutions (i.e. lower value of the *Scatter F-measure*) then a new set of sub-clusters with better homogeneity than the original clusters is generated. Therefore greater improvement in the clustering quality is achieved. The *Scatter F-measure*, number of sub-clusters, quality of sub-clusters (measured by the *OWSR* measure (defined in Section 4)) and the corresponding values of the *SI* index of the obtained k clusters are reported in Tables 8–12.

In the *Breast Cancer* dataset, both KM and BKM are close to each in terms of their clustering solutions (measured by both internal and external quality measures) where the value of the *scatter F-measure* is 0.8432. Thus the quality of the generated clusters is almost the same as that of both of them and the percentage of improvement is only 12% for *F-measure* and 5% for *SI* Index. On the other hand, the cooperative models CC(KM,PAM) and CC(BKM,PAM) achieve an improvement in the performance of up to 27% in F-measure and 29% in SI. The main reason for this significant improvement is that either CC(BKM,PAM) or CC(KM,PAM) generates solutions that are different and that reveals different set of sub-clusters with better quality. For the *Yahoo* dataset, the least improvement in the clustering quality is provided by the CC(KM,PAM), where the cooperation results in an improvement of only 5% in *F-measure* and 6% in *SI* for large value of the *Scatter F-measure* equals 0.7612 while CC(KM,BKM) for example achieves improvement up to 20% in *F-measure* and 32% in *SI* at *scatter F-measure* equals 0.4075 for the same dataset. We can conclude that the scattering between the clustering results of the adopted clustering techniques enables the cooperative model to work with more homogenous set of sub-clusters that yield better final cooperative clustering results.

8.4.3. Performance evaluation at c=3

In this section we evaluate the performance of the cooperative model by combining the clustering solutions of the three approaches, KM, BKM, and PAM together in one solution (i.e. c=3). We will refer to the cooperation between the three algorithms as CC(KM,BKM,PAM). The values of both *F-measure* and *SI index* using the cooperative and non-cooperative approaches for *Leukemia*, *Yeast*, *Breast Cancer*, *UW*, *SN*, and *Yahoo* datasets are reported in Figs. 5 and 6, respectively.

We can see that CC(KM,BKM,PAM) achieves better clustering quality than the pair-wise cooperation measured by higher values for *F-measure* and lower values for *SI* index. This enhancement in the performance caused by the triple cooperation is mainly because a new set of sub-clusters is obtained with better homogeneity than that of the pair-wise cooperation. This set of sub-clusters acts as an agreement between the three techniques together which gives an additional confidence of the distribution of objects within clusters. This agreement directs the cooperative model in such away to group more homogenous sub-clusters than those of the pair-wise cooperation.

8.4.4. Performance evaluation at c=4 (adding FCM)

The performance of the fuzzy c-means (FCM) [6] is added to the model as shown in Figs. 7 and 8 for both the *Yeast* and *UW* datasets. The CC(KM,BKM,PAM,FCM) refers to the cooperative model that combines the clustering solutions of the four techniques.

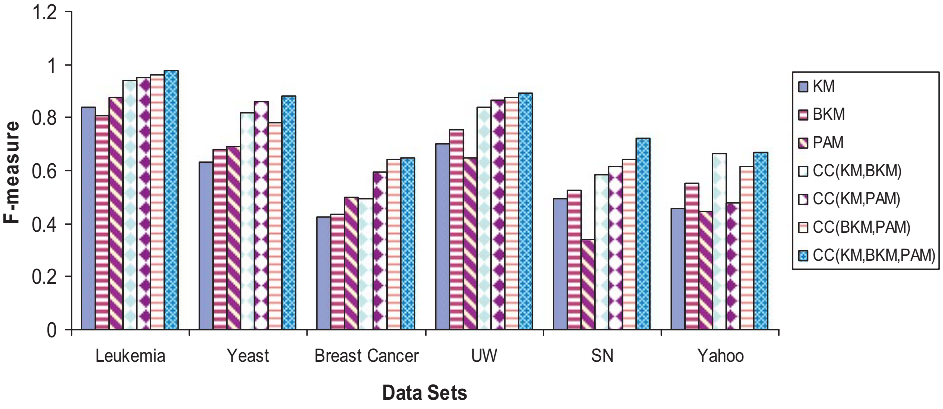

Fig. 5. Further improvement in *F-measure* using triple cooperation.

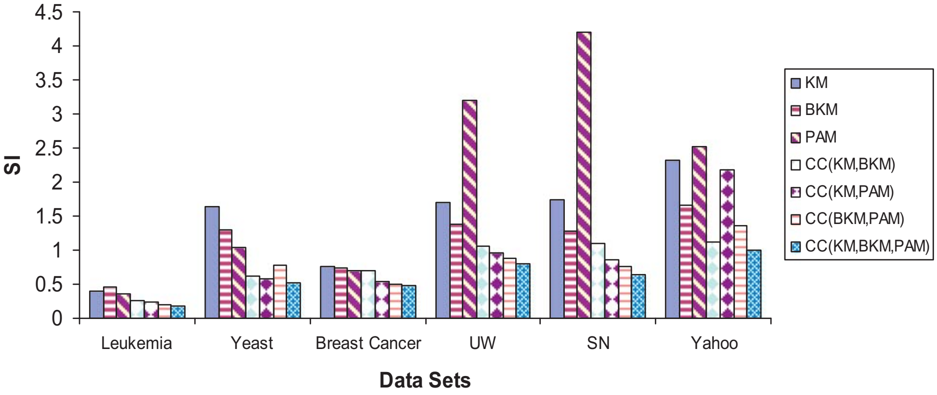

Fig. 6. Further improvement in *SI* using triple cooperation.

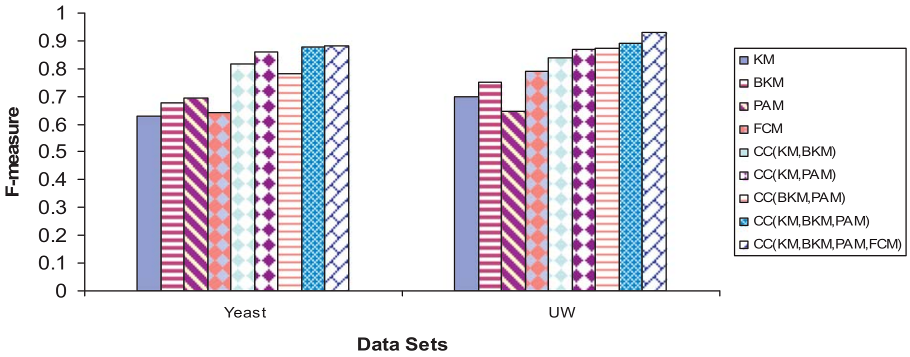

Fig. 7. Improvement in *F-measure* by adding FCM to the cooperative model (c=4).

It can be shown that adding FCM to the cooperative model maintains the same clustering quality for the *Yeast* dataset, as the performance of FCM is almost the same as KM. Thus adding FCM has no additional benefit to the cooperative model. On the other hand, for the *UW* dataset, FCM has better performance than KM. This difference in the performance with better sub-clusters homogeneity provides the cooperative model CC(KM, BKM,PAM,FCM) with more homogenous set of sub-clusters that achieves a better clustering quality. Thus adding FCM provides clustering solutions with higher values of *F-measure* and lower values for the *SI* index as illustrated in Figs. 7 and 8.

8.4.5. Scalability of the cooperative clustering (CC) model

In order to evaluate the scalability of the cooperative model in terms of number of clustering techniques, we target only combinations of KM, BKM and PAM such that c (number of clustering techniques) ranges from 2 up to 100 algorithms as shown in Figs. 9–11, for *Leukemia*, *Yeast*, and *UW* datasets, respectively.

In each table, we plot the ratio of number of singleton sub-clusters (sub-clusters with size 1) to the total number of sub-clusters, the quality of sub-clusters (measured by the *OWSR* measure), and the quality of the overall set of k clusters (measured by *F-measure*). For the *Leukemia* dataset, it can be noticed that the cooperative model achieves better clustering results using up to 39 algorithms measured by higher values of the *OWSR* of the generated set of sub-clusters as well as the higher values of the *F-measure* for the overall set of k clusters. For the *Yeast* dataset, a combination of up to 37 algorithms are used to obtain better results than the original individual approaches, and finally for the UW dataset, up to 13 algorithms are invoked to

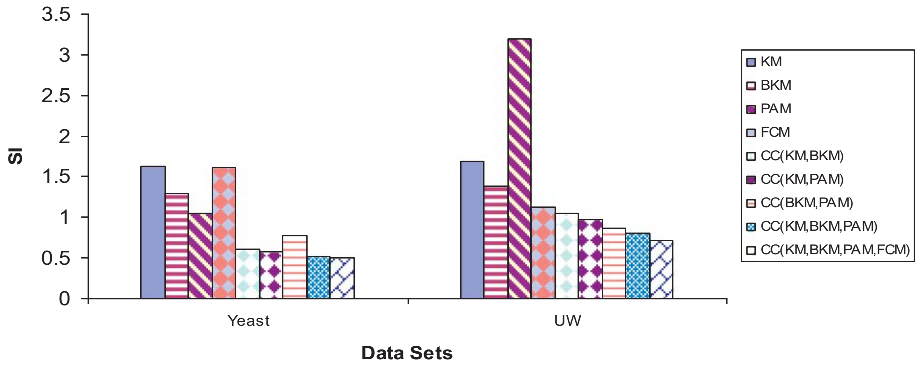

Fig. 8. Improvements in *SI* by adding FCM to the cooperative model (c=4).

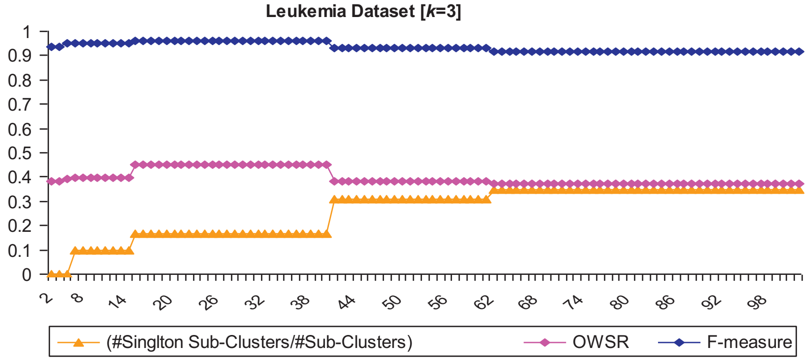

Fig. 9. Algorithm scalability of the cooperative model [*Leukemia*].

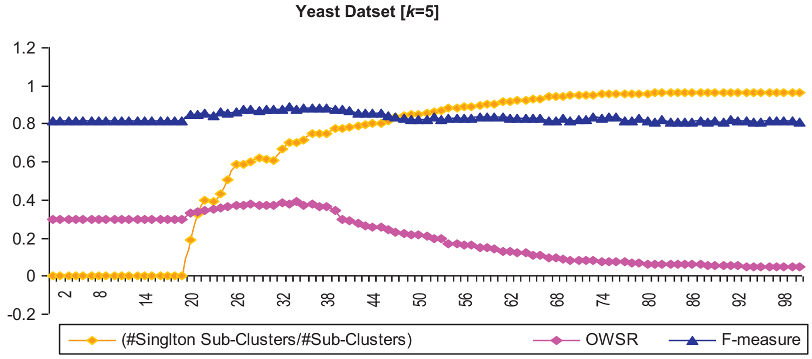

Fig. 10. Algorithm scalability of the cooperative model [*Yeast*].

obtain the best cooperative clustering results than those of the adopted non-cooperative approaches.

An interesting observation is that the cooperative clustering model after a specific value of c, c^* (e.g. c^*=39 in the *Leukemia* dataset), the cooperative clustering quality degrades rapidly. It is not surprising that this is the case, since at larger number of algorithms with different clustering solutions; the generated set of sub-clusters is expected to have larger number of singleton sub-clusters which drops the overall quality of the set of sub-clusters. The value of c^* provides a clue of the relation between the number of sub-clusters, number of singleton sub-clusters, and the overall quality of the set of sub-clusters (measured by *OWSR*), beyond which the number of algorithms should not be increased. An appropriate strategy for automatically detecting the value of c^* is to compare the values of *OWSR* before and after adding the additional clustering techniques, if a sufficiently drop in the *OWSR* is noticed then no more algorithms can be added to the cooperative clustering model.

8.4.6. Variable number of clusters

As clustering is known as unsupervised classification of data, the number of clusters is unknown as in the *Serum* dataset. In this experiment, we investigate the performance of the cooperative models as well as the individual approaches along with variable number of clusters. The proper number of clusters (i.e. natural grouping of data) is obtained based on the lowest value of the *SI* index. Fig. 12 shows the performance of the cooperative models as well as the non-cooperative algori-

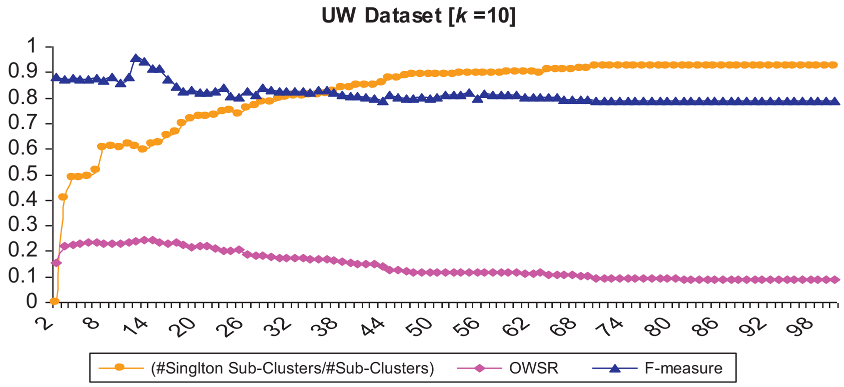

Fig. 11. Algorithm scalability of the cooperative model [*UW*].

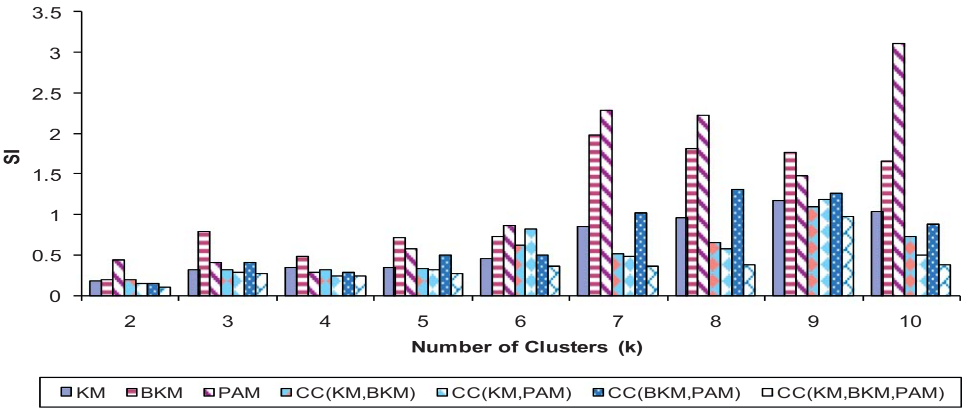

Fig. 12. Finding proper number of clusters (k is unknown) [*Serum*].

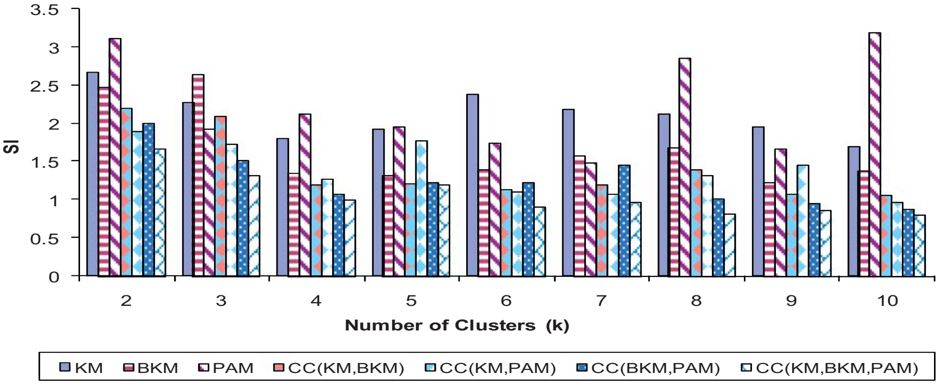

Fig. 13. Finding proper number of clusters (k is known) [*UW*].

thms for the *Serum* dataset at variables number of clusters. For the non-cooperative algorithms, it can be shown that the best performance of KM is achieved at $k=2$, BKM at $k=2$, and PAM at $k=4$. The CC(KM,BKM) has the lowest value of the *SI* index at $k=2$, the CC(KM,PAM) achieves its best performance also at $k=2$, CC(BKM,PAM) has the best results at $k=2$. Finally, for the triple cooperation model, CC(KM,BKM,PAM), the best results are obtained at $k=2$. The natural grouping of data (i.e. proper number of clusters) is determined by a majority vote between the four cooperative models as they have better clustering quality than the non-cooperative approaches. Then the natural grouping of data is obtained at $k=2$.

The *UW* dataset has external information about its true class labels where k is known apriori as 10 clusters. In Fig. 13, it can be illustrated that the four cooperative models, CC(KM,BKM), CC(KM,PAM), CC(BKM,PAM), and CC(KM,BKM,PAM) obtain the true number of clusters ($k=10$) while the non-cooperative BKM achieves its best performance at $k=9$ and PAM declares that its best results are obtained at $k=7$. KM obtains the best results at $k=10$. It can be shown that the cooperation between multiple clustering approaches is capable of finding the proper number of clusters in data.

For both the *Serum* and *UW* datasets, we can see that the cooperative models outperform the individual clustering techniques for variable number of clusters measured by lower values of the *SI* index.

9. Conclusions

In this paper, a new cooperative clustering (CC) model was presented to improve the clustering solutions over the traditional non-cooperative techniques. The CC model is primarily based on finding the intersection between the multiple clusterings in terms of a set of sub-clusters. Each sub-cluster is represented by a similarity histogram. By carefully monitoring the pair-wise similarities between objects in the sub-clusters, the CC model applies a homogeneous merging procedure on the cooperative contingency graph to attain the same number of clusters. The CC model provides clustering solutions of better quality and it is scalable in terms of number of clustering algorithms. The complexity analysis of the cooperative model was presented and analyzed. Also the notion of the *Scatter F-measure* was presented to show the scattering in clustering solutions between two clustering approaches. A new internal quality measure named, *overall weighted similarity ratio* was formally defined and proposed to assess the quality of the generated sub-clusters. Experiments were performed on actual gene expression datasets and text documents datasets representing different characteristics. Based on the experimental results, we can conclude that cooperative clustering achieves better clustering quality measured by both internal and external quality measures than the non-cooperative traditional clustering algorithms. Also a number of experiments were conducted to show the capability of the cooperative model to generate better clustering solutions with variable number of clusters. Also, undertaken experimental results show that the cooperative clustering model is scalable in terms of number of clustering techniques. Future work includes:

- Applying the same cooperative methodology if the number of the generated clusters is different from one partitioning to another and employing a new membership function to find the intersection between the c clustering solutions.
- Developing a scatter F-measure that finds the diversity in the clustering solutions between two or more approaches.
- Comparing the time and quality performances of the cooperative clustering model to those of ensemble clustering and hybrid clustering.
- Calculating the value of c^* (the maximum number of cluster techniques in the cooperative model to maintain better clustering quality) is done experimentally, proving the value of c^* theoretically will be of interest as a future work.

References

[1] Jain, M. Murty, P. Flynn, Data clustering: a review, ACM Computing Surveys 31 (1999) 264–323.
[2] R. Xu, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–678.
[3] M. Steinbach, G. Karypis, V. Kumar, A Comparison of document clustering techniques, in: Proceedings of the KDD Workshop on Text Mining, 2000, pp. 109–110.
[4] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.
[5] J. Hartigan, M. Wong, A k-means clustering algorithm, Applied Statistics 28 (1979) 100–108.
[6] J. Bezdek, R. Ehrlich, W. Full, The fuzzy C-means clustering algorithm, Computers and Geosciences 10 (1984) 191–203.
[7] S.M. Savaresi, D. Boley, On the performance of bisecting K-means and PDDP, in: Proceedings of the First SIAM International Conference on Data Mining, 2001, pp. 1–14.
[8] D. Boley, Principal direction divisive partitioning, Data Mining and Knowledge Discovery 2 (4) (1998) 325–344.
[9] K. Hammouda, M. Kamel, Collaborative document clustering, 2006 SIAM Conference on Data Mining (SDM06), 2006, pp. 453–463.
[10] R. Kashef, M. Kamel, Enhanced bisecting k-means clustering using intermediate cooperation, Pattern Recognition 42 (11) (2009) 2557–2569.
[11] Y. Zhao, G. Karypis, Criterion functions for document clustering: experiments and analysis, Technical Report, 2002.
[12] R. Kashef, M.S. Kamel, Cooperative partitional-divisive clustering and its application in gene expression analysis, in: IEEE Seventh International Conference on BioInformatics and BioEngineering (BIBE07), 2007, pp. 116–122.
[13] H. Ning, W. Xu, Y. Chi, Y. Gong, T. Huang, Incremental spectral clustering by efficiently updating the eigen-system, Pattern Recognition 43 (1) (2010) 113–127.
[14] J. Fan, M. Han, J. Wang, Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation, Pattern Recognition 42 (11) (2009) 2527–2540.
[15] J. Lai, T. Huang, Y. Liaw, A fast k-means clustering algorithm using cluster center displacement, Pattern Recognition 42 (11) (2009) 2551–2556.
[16] A. Qin, P. Suganthan, Robust growing neural gas algorithm with application in cluster analysis, Neural Networks 17 (8-9) (2004) 1135–1148.
[17] J. Kim, S. Choi, Clustering with *r*-regular graphs, Pattern Recognition 42 (9) (2009) 2020–2028.
[18] A. Qin, P. Suganthan, Enhanced neural gas network for prototype-based clustering, Pattern Recognition 38 (8) (2005) 1275–1288.
[19] A. Strehl, J. Ghosh, Cluster ensembles—knowledge reuse framework for combining partitionings, in: Conference on Artificial Intelligence, AAAI/MIT Press, 2002, pp. 93–98.
[20] Y. Qian, C. Suen, Clustering combination method, in: International Conference on Pattern Recognition, ICPR 2000, vol. 2, 2000, pp. 732–735.
[21] D. Greene, P. Cunningham, Efficient ensemble methods for document clustering, Technical Report, Computer Science Department, Trinity College Dublin, 2006.
[22] H. Ayad, M. Kamel, Cumulative Voting Consensus Method for Partitions with Variable Number of Clusters, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Computer Society Digital Library, IEEE Computer Society, 2007.
[23] C. Lin, M. Chen, Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging, IEEE Transactions on Knowledge and Data Engineering 17 (2) (2005) 145–159.
[24] Y. Eng, C. Kwoh, Z. Zhou, On the two-level hybrid clustering algorithm, AISAT04, in: International Conference on Artificial Intelligence in Science and Technology, 2004, pp. 138–142.
[25] S. Xu, J. Zhang, A hybrid parallel web document clustering algorithm and its performance study, Journal of Supercomputing 30 (2) (2004) 117–131.
[26] M. Ismail, M. Kamel, Multidimensional data clustering utilizing hybrid search strategies, Pattern Recognition 22 (1989) 75–89.
[27] L. Kaufmann, P. Rousseeuw, Finding Groups in Data, Wiley, New York, 1990.
[28] R. Ng, J. Han, Efficient and effective clustering methods for spatial data mining, in: VLDB, 1994, pp. 144–155.
[29] G. Salton, Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer, Reading Mass: Addison Wesley, 1989.
[30] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1650–1654.
[31] K. Wu, M. Yang, J. Hsieh, Robust cluster validity indexes, Pattern Recognition 42 (11) (2009) 2541–2550.
[32] S. Monti, P. Tamayo, J. Mesirov, T. Golub, Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data, Kluwer Academic Publishers, 2003.

[33] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, G. Church, Systematic determination of genetic network architecture, Nature Genetics 22 (1999) 281–285.
[34] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. Marks, J. Nevins, Predicting the clinical status of human breast cancer by using gene expression profiles, Proceedings of the National Academy of Sciences, USA 98 (2001) 11462–11467.
[35] ⟨http://www.sciencemag.org/feature/data/984559.sh⟩.
[36] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M.A. Caligiuri, C. Bloomfield, E. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[37] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E.S. Lander, T. Golub, Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, PNAS 96 (1999) 2907–2912.
[38] M. Eisen, P. Spellman, P. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proceedings of the National Academy of Sciences of the United States of America 95 (25) (1998) 14863–14868.
[39] G. Salton, A. Wong, C. Yang, A vector space model for automatic indexing, Communications of the ACM 18 (11) (1975) 613–620.
[40] D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, IEEE Transactions on Knowledge and Data Engineering 17 (12) (2005) 1624–1637.
[41] D. Boley, M. Gini, R. Gross, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, J. Moore, Partitioning-based clustering for web document categorization, Decision Support Systems 27 (1999) 329–341.

About the Author—RASHA KASHEF received the B.Sc. degree in Computer Engineering from the Faculty of Engineering, Alexandria University, Egypt, in 2001. She received the M.A.Sc. degree from the Department of Computer Engineering, Arab Academy for science and Technology, Egypt, in 2004. From 2001 to 2005, she was with the Department of Computer Engineering, Faculty of Engineering, Arab Academy for science and Technology, as a an assistant lecturer. She received her Ph.D. from the University of Waterloo, Department of Electrical and Computer Engineering in September 2008. Currently, she is hired as an assistant professor at the Arab Academy for Science and Technology. Her research interests are in data mining, especially cooperative clustering and distributed cooperative clustering, social networks analysis, grid computing, and bioinformatics.

About the Author—MOHAMED S. KAMEL received the Ph.D. degree in Computer Science from the University of Toronto, Canada. He is at present a professor and director of the Pattern Analysis and Machine Intelligence Laboratory at the Department of Systems Design Engineering, University of Waterloo, Canada. Dr. Kamel holds a Canada Research Chair in Cooperative Intelligent Systems. He has authored and coauthored more than 180 papers in journals and conference proceedings, two patents, and numerous technical and industrial project reports. Under his supervision, 44 Ph.D. and M.A.Sc. students have completed their degrees. Dr. Kamel is a member of the ACM, the AAAI, the CIPS, the APEO, and a senior member of the IEEE, and is editor inchief of the International Journal of Robotics and Automation, associate editor of four international journals, and guest editor for special issues in four journals. He is a member of the board of directors and cofounder of Virtek Vision International in Waterloo. He is a consultant to many companies including NCR, IBM, Nortel, VRP, and CSA.