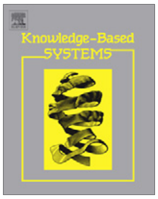
Knowledge-Based Systems 45 (2013) 83–93

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

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



A hierarchical clusterer ensemble method based on boosting theory

Elaheh Rashedi [*], Abdolreza Mirzaei

ECE Department, Isfahan University of Technology, Isfahan, Iran

ARTICLE INFO

Article history:
Received 15 May 2012
Received in revised form 12 February 2013
Accepted 13 February 2013
Available online 28 February 2013

Keywords:
Boosting
Ensemble
Hierarchical clustering
Rényi

ABSTRACT

Bagging and boosting are two well-known methods of developing classifier ensembles. It is generally agreed that the clusterer ensemble methods that utilize the boosting concept can create clusterings with quality and robustness improvements. In this paper, we introduce a new boosting based hierarchical clusterer ensemble method called *Bob-Hic*. This method is utilized to create a consensus hierarchical clustering (h-clustering) on a dataset, which is helpful to improve the clustering accuracy. *Bob-Hic* includes several boosting iterations. In each iteration, first, a weighted random sampling is performed on the original dataset. An individual h-clustering is then created on the selected samples. At the end of the iterations, the individual clusterings are combined to a final consensus h-clustering. The intermediate structures used in the combination are distance descriptor matrices which correspond to individual h-clustering results. This final integration is done through an information theoretic approach. Experiments on popular synthetic and real datasets confirm that the proposed method improves the results of simple clustering algorithms. In addition, our experimental results confirm that this method provides better consensus clustering quality compared to other available ensemble techniques.

1. Introduction

The idea of ensemble learning is to combine multiple learners' predictions. In supervised (classification) and unsupervised (clustering) learning algorithms, ensembles often lead to better results in comparison with single solutions. Classifier ensemble methods combine classifiers to achieve a classification solution of a higher predictive accuracy [1]. Similarly, clusterer ensemble methods improve clustering quality by aggregating clusterers. The most recent general ensemble methods that can be used to reduce errors in both classification and clustering cases are bagging and boosting algorithms [2–4].

The general idea of bagging, which is shortened form of the words "bootstrap aggregating", is to generate an ensemble of learners built on bootstrap replicates of the training set and also to combine learners' outputs [5]. Very similar to bagging, boosting is defined as a general method of machine learning which converts a weak learning algorithm into one with higher accuracy [5]. Also, boosting is one of the most powerful methods for creating classifier ensembles. There are also some bagging and boosting based multi clustering algorithms introduced on "partitional" clusterings [6,7]. Accordingly, there is a potential of "hierarchical" clustering (h-clustering) quality improvement through using multi hierarchical clustering methods.

In this paper a boosting based hierarchical clusterer ensemble technique, *Bob-Hic*, is proposed, which is a new method for the framework previously presented by the same authors in [8]. The new method provides solutions of better quality compared to other h-clustering fusion methods.

This paper is organized as follows: The related works on classification and partitional clusterings are demonstrated in Section 2, followed by an introduction to h-clustering combination approaches. In Section 3, a new boosting based multi hierarchical clustering algorithm is presented. The experimental results of the proposed method on both real and synthetic datasets are discussed in Sections 4 and 5. The comparative results between the proposed method and both single and ensemble h-clustering methods are also illustrated in these sections. Finally, our concluding remarks are presented in Section 6.

2. Related works

There are four approaches to build and combine classifiers: (i) methods which use different data subsets, (ii) methods which use different feature subsets, (iii) methods which apply different basic classifiers and, (iv) methods which utilize different combiners to aggregate ensembles [5]. The quality of a classifier ensemble depends on both how the classifier ensembles are *generated* and how they are *combined*.

Numerous methods have been introduced to generate and integrate classifier ensembles. An extensive body of work has been

[*] Corresponding author. Tel.: +98 9186735783.
E-mail addresses: elahehrashedi@gmail.com, e.rashedi@alumni.ut.ac.ir (E. Rashedi), mirzaei@cc.iut.ac.ir (A. Mirzaei).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.02.009

reported on bagging, boosting, and their advantages over other classifier ensemble methods [3,4,9–12]. Various hierarchical classification ensemble methods have also been proposed combining information of several labeled trees into one single representative tree [13,14]. These techniques are widely used in phylogenetics [15,16]. A brief explanation of these methods is given in Section 2.1.

In the area of clustering methods, various techniques are proposed to create clusterer ensembles [17–21], and some of them are based on bagging and boosting [3,22,23]. These methods usually use partitional (non-hierarchical) clustering algorithms to create basic clusterings [7]. A brief explanation of these methods is presented in Section 2.2.

The last point to be addressed is h-clusterer ensemble methods that can be used to combine a set of h-clusterings. A review of them has been provided in Section 2.3.

2.1. Classifier ensemble methods based on bagging and boosting

Using the best of both classifier ensemble generation method and ensemble combination method leads to a more accurate classification decision. However, this approach increases the complexity. The most popular ensemble methods utilized for reducing classification errors are bagging and boosting [5,9,24]. Bagging is a multiple learner combination method which always uses resampling. First, bootstrap replications of data points are created where the sampling is based on a random and uniform selection. Then, individual copies of the weak learner algorithm are applied on samples, and finally, the results are combined using majority voting. It can be shown that for unstable classifiers, bagging leads to performance improvement [5,9,25].

Very similar to bagging, boosting is an ensemble method used to improve the accuracy of any learning algorithm [26]. The main difference is the use of reweighting. In boosting, the weights of data points are updated based on their learning complexity while in bagging the weights remain unchanged. A popular adaptive boosting algorithm called AdaBoost is used to reduce the error and boost any weak learner's performance [3,11,24,26].

2.2. Partitional clusterer ensemble methods

There are many consensus functions introduced in partitional clusterer ensembles area, which use several mathematical and computational tools [7]. Some of the functions are as follows: relabeling and voting [27–31], fuzzy clustering [32], genetic algorithms [33], Co-association matrix [18], graph and hyper-graph (*CSPA*, *HPGA*, *MCLA*) [34], Mirkin distance [35], information theory [36], finite mixture models [19,37], locally adaptive clustering algorithm [38], Kernel [7], and non-negative matrix factorization [39].

These techniques mostly improve the result of single clustering algorithms. The main target of these methods is to find consistent clusters through clustering combination. In the combination step the information from all individual partitions of the ensemble are integrated and possible errors of single clustering methods are rectified. So the final combined clustering represents a better solution.

Similar to classification problems, constructing clusterer ensembles using bagging and boosting have a potential of clustering accuracy improvement. This method can also provide more robust results [10,27].

A clusterer ensemble method based on boosting, *boost-clustering*, is introduced in [1]. In this ensemble method, a simple partitional clusterer, e.g. *k-means*, is utilized to create multiple clustering results. The clustering results are then combined through weighted voting. Also, there are some resampling based combination solutions in which a selection of the clustering results are combined [17,22,23].

Methods which are introduced to combine h-clusterings are described in next Section 2.3 [40–44].

2.3. Hierarchical clusterer ensemble methods

In h-clustering methods, data points are organized into a nested sequence of groups instead of simple partitions. These techniques are grouped into two categories, agglomerative and divisive. Some popular agglomerative h-clustering methods are *Centroid* Linkage, *Single* Linkage, *Average* Linkage, *Complete* Linkage, *Weighted* Linkage, *Median* Linkage and *Ward* Linkage. These methods construct the hierarchy by recursively merging the two clusters with minimum defined distance values.

Hierarchies are illustrated using a tree diagram called dendrogram listing all data points and indicating at which level of similarity any two clusters are joined. Accordingly, an h-clusterer ensemble method is defined as a technique combining the basic dendrograms. The dendrograms can be represented in the form of similarity or dissimilarity matrices where these matrices are called descriptors [44]. Some descriptors are Cophenetic Difference (*CD*), Partition Membership Divergence (*PMD*), Cluster Membership Divergence (*CMD*), Sub-tree Membership Divergence (*SMD*) and Path Difference (*PD*) [45]. The descriptor used in our experiments is *CD*. In *CD* descriptors, the distance between two data points is defined as the lowest level of the hierarchy where these pairs are joined together. According to this new definition, h-clusterer combination method is defined as a technique for combining the descriptors corresponding to the basic dendrograms.

An h-clusterer combination method, *HCC*, is introduced in [44]. In *HCC*, different dissimilarity matrices are combined into one aggregated matrix using matrix summation operator. The aggregated matrix might not necessarily have an associated dendrogram. Therefore, a dendrogram recovery phase is then applied to derive the final dendrogram from the aggregated dissimilarity matrix. The recovery process employs the maximum transitive similarity property to derive the final dendrogram. The recovery phase has a great effect on the quality of the final dendrogram [41,43].

Another combination method based on fuzzy similarity relations, *MATCH*, is proposed in [41,43]. In this method, descriptors are aggregated into one transitive consensus matrix from which the final dendrogram can be directly formed with no need to the additional recovery phase. By skipping the recovery phase, the chaining effects are excluded [41,43]. Some experiments show that *MATCH* is a better solution in comparison with *HCC* in terms of quality [44]. This quality is achieved in price of a more time complexity. It should be mentioned that the time complexity of *MATCH* is $O(Ln^4)$, where L is the ensemble size and n is the number of data points.

Hierarchical ensemble clustering technique, *HEC*, is introduced in [46], as combining partitional and hierarchical clusterings into one consensus h-clustering. *HEC* and *MATCH* are dual algorithms; the former generates minimum transitive dissimilarity matrix closure, and the latter generates maximum transitive similarity matrix closure.

An information-theory-based combination method is presented in [40]. In this method, each row of the basic descriptor matrix is supposed to be a probability distribution function, *PDF*. Accordingly, each row in the consensus matrix is calculated as a *PDF* with the most similarity to all *PDF*s associated to the same row in basic descriptors. The *Rényi* divergence is utilized as a measure to calculate the *PDF* values of the consensus matrix. The *Rényi* divergence is a member of the family of functions quantifying the diversity of a system and can be used to calculate the dissimilarity of two *PDF*s.

Let P^* be the consensus matrix, which stands for the nearest *PDF* to all basic *PDF* matrices P^i. Based on *Rényi*, P^* is calculated as:

$$p^*_{m,n} = \frac{1}{r}\left(\sum_{i=1}^{L}\left(p^i_{mn}\right)^{1-\alpha}\right)^{\frac{1}{1-\alpha}} \tag{1}$$

in which, L is the number of basic clusters in the ensemble and $p^i_{m,n}$ is the probability value placed in the mth row and nth column of the ith basic *PDF* matrix. Similarly, p^*_{mn} is the probability value placed in mth row and nth column of the consensus *PDF* matrix. Finally, r is the normalization constant. Setting α to a fixed value, the consensus *PDF* matrix is calculated and is recovered to generate the final dendrogram [40]. *HCC* can be viewed as a special case of the information theory based combination method in which the *Rényi* divergence with $\alpha = 0$ is used.

In this paper, the information theory based combination method is used to aggregate multiple descriptors. The proposed method is compared to *MATCH*.

3. The boosting based hierarchical clusterer ensemble method: *Bob-Hic*

In this section we propose a new clusterer ensemble method for h-clusterings. This is a general method in which any h-clustering algorithm can be used in ensemble generation and any hierarchical combination method can be used in individual clusterings aggregation. The algorithm is given in Section 3.1, and its time complexity analysis is discussed in Section 3.2.

3.1. Bob-Hic algorithm

The steps of this algorithm are as follows: (a) at the first iteration of boosting, a new sample set is provided by random sampling (without replacement) from the original dataset; (b) the h-clustering algorithm is applied on the selected samples to create the first hierarchical clustering of data points; (c) each data point's weight is updated based on the efficacy of the previous hierarchical clusterings which are built on it, and the next basic single clustering is generated according to the new weights; and (d) the final clustering solution is produced by combining all hierarchies in the ensemble.

Here, standard h-clustering methods like *Centroid*, *Single*, *Average*, *Complete*, *Weighted*, *Median* and *Ward Linkages* are used for creating basic clusterings. The similarity descriptors of basic single clustering results are then combined into one descriptor. The combination stage is done based on the *Rényi* divergence approach. The pseudo code of the boosted combination algorithm is illustrated in Fig. 1. The main framework of the boosted h-clusterer ensemble method is based on the general boosting algorithm, arc-x4 [5]. The algorithm starts to generate a consensus h-clustering on a data set of N data points, $D = (x_1, x_2, \ldots, x_N)$.

Step 1, initializing the variables: At the beginning of the algorithm, some initializations are performed as follows: The iteration number, i, is set to 1, and the maximum number of iterations, T, is set to a predefined fixed number. Also, the size of the ensemble is denoted by L. L is equal to T, as in each iteration one individual clustering of the ensemble is created. Then, the initial weight of each data point is assigned to $1/N$. Thereafter, a basic h-clustering algorithm, *Hclusterer*, is selected for generating the basic single h-clusterings. Finally, an h-clustering combination method, *Hcombiner*, is chosen to aggregate single h-clusterings.

Step 2, performing the iterations: This step is an iterative process which is supposed to be done for T times.

Step 2.1, generating sample sets: In this step, the boosting stage is iteratively done as follows: the sample set of the ith iteration, D^i, is prepared using the weighted sampling approach, in which the samples are selected randomly according to the distribution of data points. This distribution is related to the weights assigned to each data point (see Eq. (2)). The pseudo-code of the *Sampler* algorithm used in Eq. (2) is illustrated in Fig. 2. The *Sampler* is based on the weighted roulette wheel selection algorithm [47].

Step 2.2, creating hierarchical clustering: In this step, one basic h-clustering is generated by applying basic h-clustering algorithm to D^i. The single clustering generated in the ith iteration is denoted as H^i (Eq. (3)).

Step 2.3, combining hierarchical clusterings: H^i_{agg} is the aggregated clustering obtained from the ith iteration (Eq. (4)). At the starting point, this value is equal to H^1, and in the next iterations, it is equal to the combination of H^{i-1}_{agg} and H^i. Where H^{i-1}_{agg} is the aggregated clustering obtained from the previous iterations, and H^i is the newly generated clustering.

In this step, H^i and H^{i-1}_{agg} are aggregated by applying the h-clustering combination method, *Hcombiner*, and H^i_{agg} is created.

As discussed in Section 2.3, combination methods often perform the aggregation process using descriptors instead of direct dendrograms. Hence, Eq. (4) is redefined as Eq. (11) in which f is a conversion function creating a similarity matrix M from the hierarchy H, and f^{-1} is a reverse function recovering the hierarchy from the similarity matrix.

$$H^i_{agg} = f^{-1}\left(Mcomb\left(f\left(H^{i-1}_{agg}\right), f(H^i)\right)\right) \tag{11}$$

In Eq. (11), *Mcomb* is an element wise matrix combination function based on the *Rényi* divergence approach introduced in Eq. (1) (with $1-\alpha$ equals 1). H^i_{agg} is generated by applying Eq. (11)

Step 2.4, calculating boosting values: In this step, the h-clustering quality is calculated in order to update each data points' weight. The most challenging problem in h-clusterer ensembles is finding a measurement concept that predicts clustering quality of each instance; because it is hard to detect how well a data point is clustered in a hierarchy. Here, the quality of each data point x_n is calculated by the boosting value, BV_n [5], where BV_n takes high values when the data point x_n has been well-clustered in the aggregated hierarchy H^i_{agg} and low values when the clustering quality of the data point x_n in the hierarchy is unpleasant (Eq. (5)).

In our framework, BV is measured by a comparison between the hierarchical distances of the data points (i.e. descriptors) and Euclidean distances of the original data set. Accordingly, we formulate the BV of the data point x_n in the ith iteration, BV^i_n, as a modulus correlation coefficient between these two distance sets:

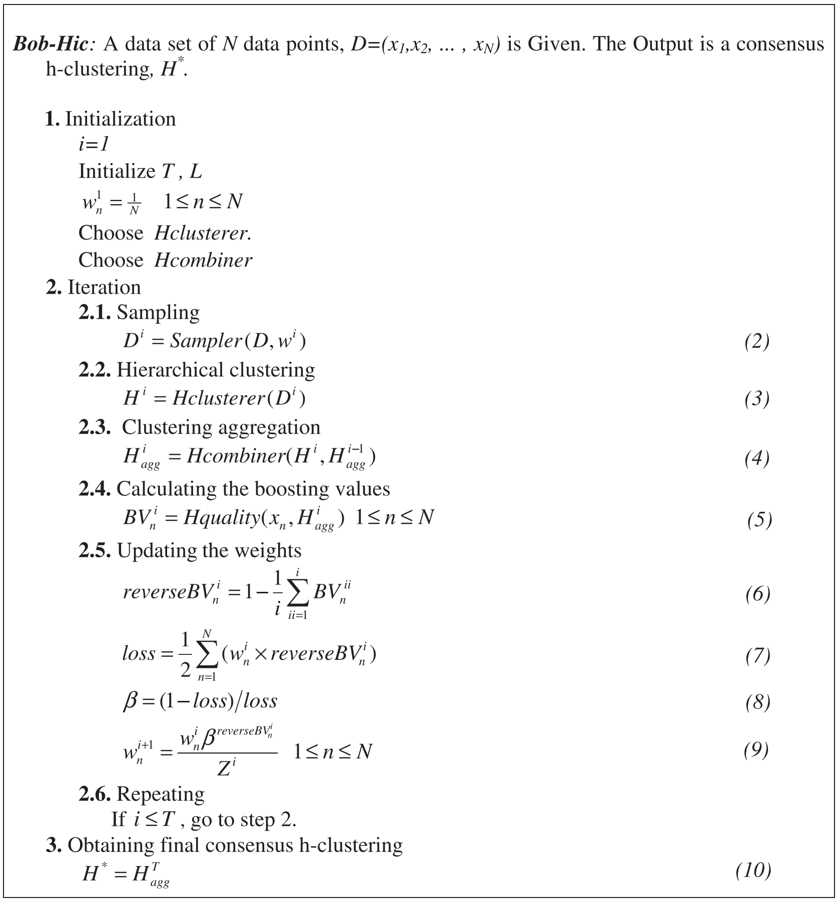
1. Hierarchical distances of the sample x_n from other data points in H^i_{agg} i.e., the nth row of the descriptor of H^i_{agg} (denoted by Hd^i_n),
2. The Euclidian distances of the sample x_n from other data points in the original data set, i.e., the nth row of Euclidean distance matrix, (denoted by Ed_n).

So the BV^i_n in Eq. (5) will be redefined as Eq. (12):

$$BV^i_n = \left|\frac{\sum_{c=1}^{N}\left(Hd^i_{n,c} - \frac{1}{N}\sum_{c=1}^{N}Hd^i_{n,c}\right)\left(Ed_{n,c} - \frac{1}{N}\sum_{c=1}^{N}Ed_{n,c}\right)}{\sqrt{\sum_{c=1}^{N}\left(Hd^i_{n,c} - \frac{1}{N}\sum_{c=1}^{N}Hd^i_{n,c}\right)^2\sum_{c=1}^{N}\left(Ed_{n,c} - \frac{1}{N}\sum_{c=1}^{N}Ed_{n,c}\right)^2}}\right| \tag{12}$$

The BV takes a value from $[0,1]$ interval. According to Eq. (12), better quality samples get higher boosting values, which are caused by high correlation between the Euclidean distance matrix and the h-clustering descriptor.

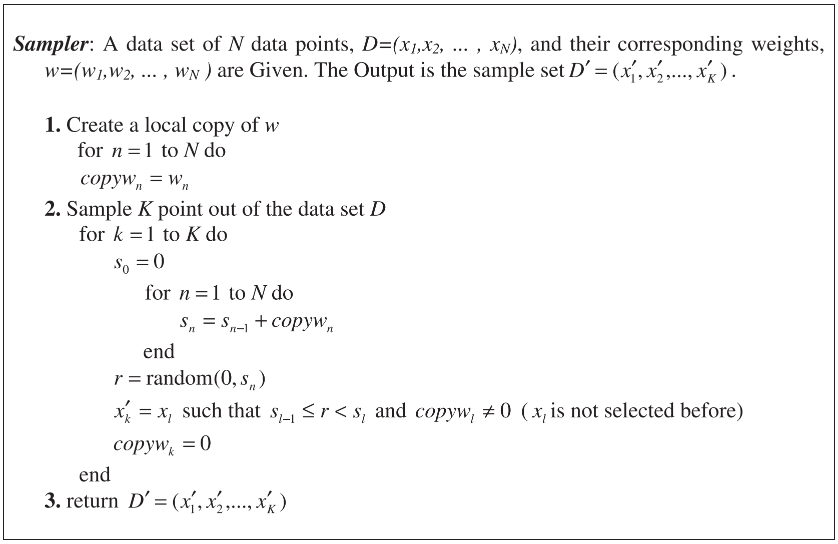
Step 2.5, updating the weights: In Eq. (12), we consider the aggregated clustering in order to calculate the clustering quality. This approach is used in arc-x4 boosting algorithm [5]. Following this approach, we update each sample's weight, where these



```
Bob-Hic: A data set of N data points, $D=(x_1,x_2, \dots , x_N)$ is Given. The Output is a consensus
    h-clustering, $H^*$.

  1. Initialization
      $i=1$
      Initialize $T$ , $L$
      $w_n^1 = \frac{1}{N} \quad 1 \le n \le N$
      Choose Hclusterer.
      Choose Hcombiner
  2. Iteration
      2.1. Sampling
          $D^i = Sampler(D, w^i)$                                              (2)
      2.2. Hierarchical clustering
          $H^i = Hclusterer(D^i)$                                              (3)
      2.3. Clustering aggregation
          $H_{agg}^i = Hcombiner(H^i, H_{agg}^{i-1})$                          (4)
      2.4. Calculating the boosting values
          $BV_n^i = Hquality(x_n, H_{agg}^i) \quad 1 \le n \le N$              (5)
      2.5. Updating the weights
          $reverseBV_n^i = 1 - \frac{1}{i}\sum_{ii=1}^{i} BV_n^{ii}$           (6)
          $loss = \frac{1}{2}\sum_{n=1}^{N}(w_n^i \times reverseBV_n^i)$       (7)
          $\beta = (1-loss)/loss$                                              (8)
          $w_n^{i+1} = \frac{w_n^i \beta^{reverseBV_n^i}}{Z^i} \quad 1 \le n \le N$   (9)
      2.6. Repeating
          If $i \le T$ , go to step 2.
  3. Obtaining final consensus h-clustering
      $H^* = H_{agg}^T$                                                        (10)
```

Fig. 1. Pseudo code of the boosted h-clustering combination algorithm.



```
Sampler: A data set of N data points, $D=(x_1,x_2, \dots , x_N)$, and their corresponding weights,
    $w=(w_1,w_2, \dots , w_N)$ are Given. The Output is the sample set $D' = (x'_1, x'_2, \dots, x'_K)$ .

  1. Create a local copy of w
      for $n = 1$ to N do
      $copyw_n = w_n$
  2. Sample K point out of the data set D
      for $k = 1$ to K do
          $s_0 = 0$
              for $n = 1$ to N do
                  $s_n = s_{n-1} + copyw_n$
              end
          $r = \text{random}(0, s_n)$
          $x'_k = x_l$ such that $s_{l-1} \le r < s_l$ and $copyw_l \ne 0$ ( $x_l$ is not selected before)
          $copyw_k = 0$
      end
  3. return $D' = (x'_1, x'_2, \dots, x'_K)$
```

Fig. 2. Pseudo code of the *Sampler* algorithm.

weights have to be changed so that the instances which are harder to cluster get higher probability of being selected in the next iterations. Similarly, the instances which are well clustered should get lower probability to be selected. In order to reach this aim, we propose *reverseBV*, as the reverse of *BV*, which takes high values when a sample $x_n$has not been well-clustered in previous iterations (Eq. (6)) [5]. Based on *reverseBV*, the loss and pseudo loss is computed using Eqs. (7) and (8) [5], and then, the distribution of the samples' weight, w_n^{i+1}, is calculated using Eq. (9). In Eq. (9), Z^{i+1} is normalization constant used to enforce $\sum_{n=1}^{N} w_n^{i+1}$ equal to 1. This formula assigns higher weights to badly clustered samples and consequently lower weights to well clustered ones [5].

Step 2.6, stopping criteria: The algorithm terminates if the iteration number, *i*, equals the maximum number of iterations, *T*. The newly assigned weights, i.e. w^{i+1}, are used in the sampling algorithm for the next iteration.

Step 3, creating the final clustering: The consensus clustering generated in the *T*th iteration, H^T_{agg}, is the final clustering which is claimed to be better in quality (Eq. (10).

3.2. Time complexity analysis

In *Bob-Hic* algorithm, three methods, the *Hclusterer*, the *Hcombiner* and the *recovery* (f^{-1}), are iteratively done for *T* times. Suppose that the time-complexities of the mentioned methods are consecutively denoted by $T_{cluster}$, $T_{combine}$ and $T_{recovery}$. So, the overall time complexity of the proposed algorithm will be $O(T \times (T_{cluster} + T_{combine} + T_{recovery}))$, or $O(L \times (T_{cluster} + T_{combine} + T_{recovery}))$.

The worst computational complexity of the *Hclusterer* method and also the *recovery* method used in this framework (i.e. *Centroid*, *Single*, *Average*, *Complete*, *Weighted*, *Median* or *Ward* linkage methods), is $O(n^2lgn)$ [48]. Also, the underused *Hcombiner* methods are three common matrix operations, *minimum*, *average* and *maximum* (further discussed in 4.1), which have the computational complexity of $O(n^2)$. Therefore, the overall complexity of *Bob-Hic* is $O(L \times (n^2lgn + n^2 + n^2lgn) = O(Ln^2logn)$.

In the next section, the details of the experimental set up and results are given.

4. Experimental set up and results

The *Bob-Hic* has been evaluated on various benchmark datasets given in Table 1. Datasets are collected from popular real dataset repositories, University of California Irvine Repository of Machine Learning Datasets [49] and Real Medical Datasets [50]. The trial datasets contain different number of data points, from 85 to 768. The experimental methodology, quality measurement method, statistical analysis and experimental results are presented in the next Sections 4.1–4.4.

4.1. Experimental methodology

At the starting point of the experiments, a basic single h-clustering algorithm is utilized to create dendrogram ensembles. The proposed method is evaluated using seven well known agglomerative h-clustering methods. We define the variable *F*1 to show the clusterer types in 7 levels as below:

$$F1 = \{Centroid, Single, Average, Complete, Weighted, Median, Ward\}$$

The clustering algorithms are applied on a sample of data. Here, the sample is set to be 40% of the original dataset. Selecting the subsamples, the next step is to apply the basic clustering algorithm on the samples to create the dendrogram. After that, we represent the dendrograms in the form of descriptors. In our experiments, the cophenetic difference (*CD)* matrices are used.

Table 1
Source and characteristics of datasets used in this experiment.

Dataset	#Col	#Instance	#Feature	Source
breast_cancer	1	263	9	[49]
contractions	2	98	27	[50]
diabetes	3	768	8	[49]
heart	4	270	13	[49]
laryngeal2	5	692	16	[50]
laryngeal3	6	353	16	[50]
liver_disorders	7	345	6	[49]
pima-indians-diabetes	8	768	8	[49]
respiratory	9	85	17	[50]
thyroid_aeberhard	10	215	5	[49]
thyroid	11	215	5	[49]
weaning	12	302	17	[50]
wpbc	13	198	32	[49]

A combination method which is based on the *Rényi* divergence approach is then applied to the descriptors. Setting $1 - \alpha$ parameter equal to $\{-\infty, 1, +\infty\}$, the combination function is converted to common operators *minimum*, *average* and *maximum*. Therefore, we define the variable *F*2 to show the combination types in 3 levels as below:

$$F2 = \{Max, Min, Average\}$$

These combination methods are applied on descriptors. The consensus matrix is then fed into a dendrogram recovery function, f^{-1}, to retrieve the final h-clustering result. The tested recovery functions are the common agglomerative h-clustering methods namely *Centroid*, *Single*, *Average*, *Complete*, *Weighted*, *Median* and *Ward*. Thereby, we define the variable *F*3 to show the recovery methods in 7 levels as below:

$$F3 = \{Centroid, Single, Average, Complete, Weighted, Median, Ward\}$$

According to *clusterer types*, *combination types* and *recovery methods*, the experiments are conducted with 7 × 3 × 7 = 147 different parameter values to evaluate the accuracy of consensus hierarchical partition results. So, the ensemble method is applied 147 times on each dataset. The number of boosting iterations, *T*, in each application is set to 100.

After performing all of the experiments, we design a statistical analysis to find the most effective parameter values among 147 different parameter values. After finding the most effective parameter values, the *Bob-Hic* is reduced to the *preferred Bob-Hic* method with a single value for each of the parameters *clusterer type*, *combination type* and *recovery method.*

Finally, the quality of final consensus h-clustering generated by the preferred method is compared to the standard h-clustering methods and also the fusion method, *MATCH*.

The quality measurement method, statistical analysis and comparative results are explained further in Sections 4.2–4.4.

4.2. Quality measurement method

In order to evaluate the proposed method, a quality measurement is needed to verify whether the consensus h-clustering structure fits the original data [6,51]. The most popular measurement is cophenetic correlation coefficient (*CPCC*). This coefficient compares the two matrices containing defined distances between pairs of data points, one of them corresponds to the original data and the other corresponds to the hierarchy [6,46,51–53]. The defined distances on original data can be Euclidian and the distances on hierarchy can be described by cophenetic difference (*CD*). The *CPCC* between two different distance matrices *X* and *Y* is defined as Eq. (13), in which $X_{i,j}$ and $Y_{i,j}$ are distances between objects *i* and *j* in matrices *X* and *Y* respectively. Also $\overline{X}$ and $\overline{Y}$ are the average distances.

$$CPCC = \left| \frac{\sum_{i<j}(X_{i,j} - \overline{X})(Y_{i,j} - \overline{Y})}{\sqrt{\sum_{i<j}(X_{i,j} - \overline{X})^2 \sum_{i<j}(Y_{i,j} - \overline{Y})^2}} \right| \tag{13}$$

The *CPCC* takes a value from the interval [0, 1], where a higher value shows a better agreement between two tested matrices. In our experiments, the *CPCC* between the Euclidian matrix of the original data and the cophenetic matrix of consensus dendrogram is measured to show the quality of the results.

The statistical analysis and experimental results are described in the following Sections 4.3 and 4.4.

4.3. Statistical analysis

In this subsection, we design a statistical analysis to find the most effective parameter values among the 147 different parameter values. Here, we perform a factorial analysis of variance (ANOVA) [54]. Factorial ANOVA is a technique which is used to prospect the variation of a continuous dependent response variable under the experimental conditions identified by independent categorization variables. The variation in the response is supposed to be caused by effects of the categorization variables, and the variation caused by random error. Similarly, factorial ANOVA is used when experimental data have multiple categorization variables and a continuous response variable. So the factorial ANOVA is believed to be appropriate in our experiments, with three multiple categorization variables (*F*1, *F*2 and *F*3) and one response variable (*CPCC*).

Here, we design a factorial ANOVA model, *M*1, on *CPCC* values of 147 experiments on all datasets. We compare *CPCC* levels for each *F*1, *F*2 and *F*3. The defined factorial model also includes the effects of up to 3-way interactions, $F1 \times F2$, $F2 \times F3$, $F1 \times F3$ and $F1 \times F2 \times F3$. In this analysis, the variation of the *CPCC* is expected to be the sum of the variations caused by multiple way interaction effects of the categorization variables (i.e. *F*1, *F*2 and *F*3), and the variation caused by random error.

Table 2 is the result of analyzing the defined *M*1 model. The resulting table includes six columns namely *Source*, *DF*, *Sum of Squares*, *Mean Square*, *F*-value, and *Pr* > *F*. The column *Source* stands for source of variation, *DF* column stands for the degrees of freedom, *F*-value column shows the division of mean square of the model by the mean square of the error, and *Pr* > *F* is the probability note providing statistical significance [54]. Here, we focus on the last column, *Pr* > *F*, which is also called *p*-value. The *p*-value resulted from our ANOVA analysis is <0.0001 (see Table 2). This value indicates that the *M*1 model is significant in explaining the variation in the *CPCC* at the $\alpha = 0.05$ level (that is, each *p*-value is "much" less than $\alpha = 0.05$ level).

The effect of the categorization variables on *M*1 model is shown in Table 3. The detailed description of each column is reported in [54]. According to calculated *p*-value shown in this table, it is observed that the effects of *F*2, *F*3 and their 2-way interaction effect i.e., $F2 \times F3$, are significant due to the fact that their corresponding *p*-values are <0.0001, which is "much" less than α level. In contrast, the effects of *F*1 and its multiple way interaction effects, i.e. $F1 \times F2$, $F1 \times F3$ and $F1 \times F2 \times F3$, are not significant due to the fact that their corresponding *p*-values are much more than α level.

According to these results, it is observed that the differences between *F*1 levels are not significant. So, the analysis is continued upon a new model, *M*2, with two multiple categorization variables *F*2 and *F*3 and the response variable *CPCC*.

Table 4 shows the result of analysis of defined *M*2 model. A small *p*-value <0.0001 indicates that the overall *M*2 model is significant in explaining the variation in the *CPCC*. According to the new calculated *p*-values shown in Table 5, it is observed that the effects of *F*2, *F*3 and $F2 \times F3$ are significant.

In the following, we perform Duncan's multiple range test (MRT) to obtain the means of *CPCC* in multiple levels of *F*2 and *F*3. Duncan MRT is a type of multiple comparison procedure which compares the sets of means and determines the significant differences between means [55]. Multiple comparisons are used in the statistical analysis that includes a number of formal comparisons, with the attention focused on the strongest differences among all these comparisons.

In the beginning of the Duncan's MRT, some sets are indicated as related to each parameter. Each set contains the results of the applied method taking one parameter's value unchanged while those remaining are varied in the relevant indicated range of values. Then, the means of sets are arranged into some groups. The grouping indication is in such a way that the means within the same group are not significantly different and, those from different groups are significantly different at an assumed $\alpha = 0.05$ level.

Table 2
standard ANOVA table as the result of the analysis of *M*1 model.

Source	DF	Sum of Squares	Mean Square	*F* value	*Pr* > *F*
Model	146	49.125	0.336	14.09	<0.0001
Error	1763	42.102	0.024		
Corrected Total	1909	91.227			

Table 3
Tests of effects of categorization variables *F*1, *F*2 and *F*3 of *M*1 model.

Source	DF	Type III SS	Mean Square	*F* value	*Pr* > *F*
F1	6	0.388	0.065	2.71	0.0130
F2	2	0.952	0.476	19.93	<0.0001
F3	6	30.850	5.142	215.31	<0.0001
F1 × F2	12	0.572	0.048	2.00	0.0218
F1 × F3	36	0.377	0.010	0.44	0.9985
F2 × F3	12	15.435	1.286	53.86	<0.0001
F1 × F2 × F3	72	0.843	0.012	0.49	0.9999

Table 4
Standard ANOVA table as the result of the analysis of *M*2 model.

Source	DF	Sum of Squares	Mean Square	*F* value	*Pr* > *F*
Model	20	46.922	2.346	101.68	<0.0001
Error	1889	43.170	0.023		
Corrected Total	1909	90.093			

Table 5
Tests of Effects of categorization variables *F*2 and *F*3 of *M*2 model.

Source	DF	Type III SS	Mean Square	*F* value	*Pr* > *F*
F2	2	0.949	0.475	20.57	<0.0001
F3	6	30.871	5.145	223.00	<0.0001
F2 × F3	12	15.435	1.286	55.75	<0.0001

Table 6
Duncan's multiple range test for *F*2 of *M*2 model.

Duncan Grouping	Mean	F2
A	0.62890	Min
B	0.58024	Average
	0.55863	Max

Table 7
Duncan's multiple range test for *F*3 of *M*2 model.

Duncan Grouping	Mean	F3
A	0.74023	Average
	0.73316	Centroid
B	0.68833	Median
	0.67200	Weighted
C	0.63398	Single
D	0.38398	Complete
E	0.25932	Ward

Tables 6 and 7 shows the results of Duncan's multiple range tests for *F*2 and *F*3 of *M*2 model.

The Duncan grouping column in Tables 6 and 7 shows the means which are significantly different. According to Table 6, we can conclude that the mean *CPCC* for *Min* combination type is higher than the means for all other combination types and differences between other means are not significant. Similarly, from Table 7 we can conclude that the mean of the *CPCC* for *Average* and *Centroid* recovery methods is higher than the means of all other recovery methods.

Accordingly, it can be said that the main effect of *Min* level of *F*2 and *Average* level of *F*3 on *CPCC* is higher than all other levels of *F*2 and *F*3. But, as the 2-way interaction effect of $F2 \times F3$ is significant, it cannot directly be concluded that the two variable levels *Min* from *F*2 and *Average* from *F*3 cause the strongest variation on *CPCC*. In order to consider the 2-way interaction effect the mean plot of *CPCC* versus *F*2 and *F*3 is used. In Fig. 3 the means plot are illustrated, where the seven curves display *CPCC* values versus combination types. Here, each curve corresponds to one of the recovery methods. The means plot in Fig. 3 indicates that the *CPCC* values are higher in the case that *F*2 is on the *Min* level and *F*3 is on the *Average* level.

According to the analysis of the model it can be concluded that in the aggregation of any of the clusterer types, *Min* combination type and *Average* recovery method commonly generate results of better quality (i.e. higher *CPCC* value). Hence, we can choose any clusterer type (here the *Centroid*), *Min* combination type and *Average* recovery method as the preferred level.

4.4. Experimental results and comparison

In order to show the quality improvement of the preferred level of *Bob-Hic* method, (i.e. *Centroid* clusterer type, *Min* combination type and *Average* recovery method) over basic h-clustering methods and *MATCH* algorithm, a comparison is carried out and the results are shown in Table 8. The maximum *CPCC* values obtained from proposed solution and basic methods are also compared.

It is indicated from the Table 8 that the preferred *Bob-Hic* performs better than basic h-clustering methods in all cases and it performs as well as *MATCH*. Comparing the time complexity of the proposed method with *MATCH*, i.e. $O(Ln^2 logn)$ versus $O(Ln^4)$, we can say that *Bob-Hic* performs better in reducing time complexity.

5. Experimental results on 2-dimensional labeled datasets

In this subsection, some 2-dimensional labeled datasets are used to better demonstrate the *Bob-Hic* performance. Datasets are illustrated in Fig. 4. Datasets named Banana, Long1, Spiral, Circle, Smile and Triangle2 include elongated clusters. These datasets are hard to be clustered with compactness based clustering algorithms. Dataset named Square1 includes four clusters which are equal in size and the level of spreading of data points. Square4 and Size5 are variations of Square1 dataset with different degree of overlapping. These datasets are hard to be clustered with connectedness based clustering algorithms. Finally, datasets named Longspiral, Longsquare and Spiralsquare include different types of clusters which are hard to be clustered with both mentioned clustering algorithms [56].

The result of applying *Single Linkage* clustering method on each dataset is illustrated in Fig. 5. It can be seen from Fig. 5 that the *Single Linkage* method acts well on datasets containing elongated clusters, but does a poor clustering on datasets containing either overlapped clusters or different clusters types. The illustration of our consensus clustering results and evaluation of the results are presented in the next Sections 5.1 and 5.2.

5.1. Illustrated clustering results of Bob-Hic

In order to evaluate the proposed method, the *Bob-Hic* clusterer ensemble algorithm is applied on the datasets shown in Fig. 4. The initialization step of the *Bob-Hic* algorithm is as follows: First, the number of iterations, *T*, is set to 100. In order to create basic single h-clusterings, one of the agglomerative h-clustering is randomly selected, since the effect of different basic standard h-clusterings is statistically proved to be not significantly different. Then, the subsample is set to be 40% of the original dataset and the distances are computed in Euclidean type. Finally, the *Min* aggregation function and the *Average* linkage method are chosen as the h-clustering combination method and the recovery method respectively. Initializing the algorithm, a consensus h-clustering is generated on each dataset. The results of the consensus clustering on each dataset are illustrated in Fig. 6.

According to Fig. 6, the *Bob-Hic* algorithm acts well on the most of datasets consisting of both elongated and overlapped clusters. And it can be seen that the proposed method creates good clusters on some datasets containing different cluster types.

5.2. Evaluation criterion of clustering quality

In order to evaluate the quality of the final resulting clusters, we use the accuracy measure, *Fscore*. This measurement evaluates the similarity of a clustering to ground truth information of classes [57]. Let *c* be the number of individual classes. Then, the total *Fscore* will be computed as the weighted sum of these classes' *Fscore* according to their size. The *Fscore* can be calculated using Eq. (14), in which X_ρ is a class with the size of n_r and $F(C_r)$ is the *Fscore* of the class X_ρ.

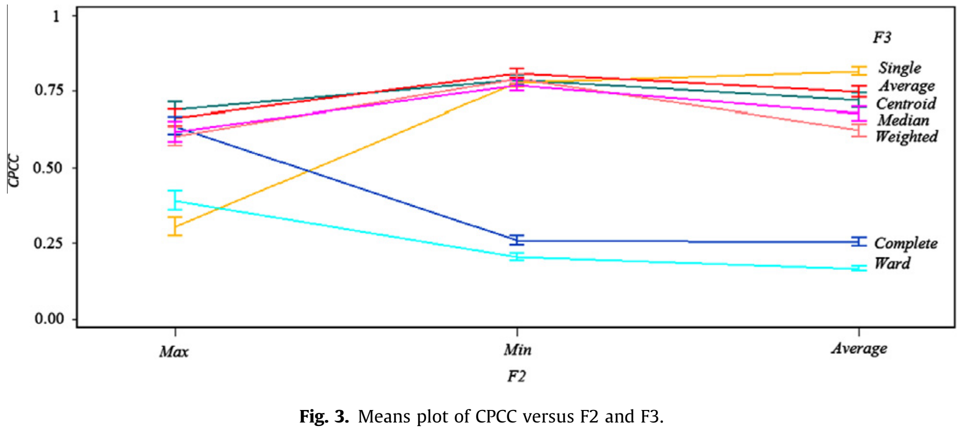


Fig. 3. Means plot of CPCC versus F2 and F3.

Table 8
Comparison of *CPCC* values between proposed solution and seven well known agglomerative h-clustering methods namely *Centroid*, *Single*, *Average*, *Complete*, *Weighted*, *Median* and *Ward* with no subsampling and also *MATCH* fusing algorithm.

Dataset		1	2	3	4	5	6	7	8	9	10	11	12	13
Bob-Hic	Preferred	0.762	0.796	0.792	0.771	0.905	0.911	0.905	0.800	0.735	0.935	0.938	0.736	0.805
	Max	0.762	0.995	0.795	0.800	0.905	0.911	0.913	0.800	0.745	0.949	0.948	0.750	0.820
Standard methods	Centroid	0.715	0.756	0.753	0.601	0.875	0.885	0.875	0.753	0.652	0.895	0.895	0.653	0.767
	Single	0.691	0.478	0.727	0.617	0.850	0.854	0.882	0.727	0.573	0.896	0.896	0.681	0.747
	Average	0.716	0.779	0.756	0.693	0.870	0.889	0.873	0.756	0.676	0.899	0.899	0.685	0.764
	Complete	0.594	0.697	0.477	0.488	0.636	0.722	0.736	0.477	0.530	0.791	0.791	0.434	0.554
	Weighted	0.644	0.720	0.594	0.547	0.724	0.658	0.703	0.594	0.622	0.819	0.819	0.384	0.443
	Median	0.631	0.669	0.554	0.529	0.786	0.731	0.666	0.554	0.594	0.594	0.594	0.546	0.683
	Ward	0.610	0.666	0.435	0.436	0.470	0.607	0.660	0.435	0.459	0.794	0.794	0.426	0.423
	Max	0.716	0.779	0.756	0.693	0.875	0.889	0.882	0.756	0.676	0.899	0.899	0.685	0.767
MATCH	–	0.762	0.624	0.790	0.881	0.921	0.895	0.954	0.780	0.641	0.869	0.900	0.752	0.812

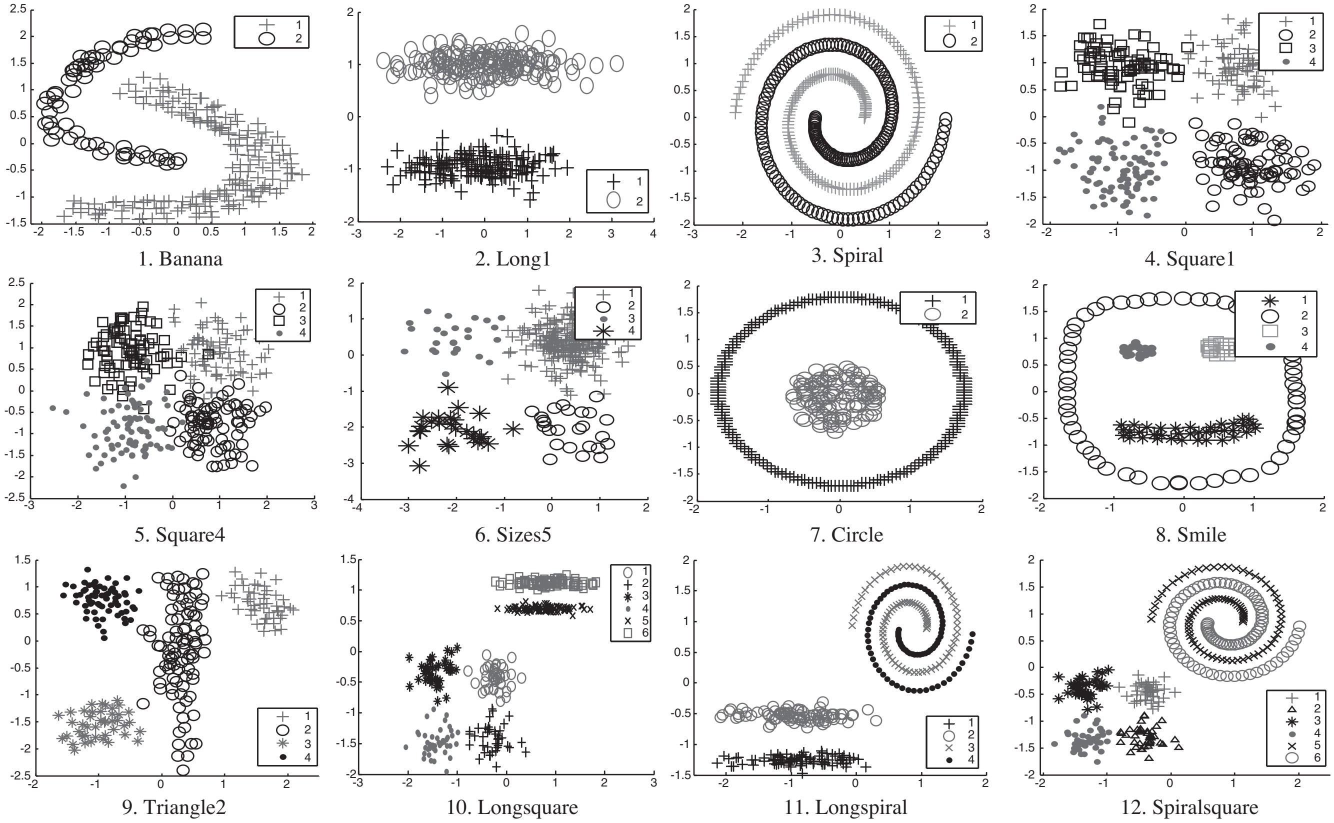

1. Banana
2. Long1
3. Spiral
4. Square1
5. Square4
6. Sizes5
7. Circle
8. Smile
9. Triangle2
10. Longsquare
11. Longspiral
12. Spiralsquare

Fig. 4. Datasets used to demonstrate the quality of the proposed h-clusterer ensemble method, *Bob-Hic*.

$$Fscore = \sum_{r=1}^{c} \frac{n_r}{N} F(C_r) \tag{14}$$

Assume that Σ_ι is the $\iota^{\tau\eta}$ cluster; For each class X_ρ, $F(C_r)$ finds a corresponding cluster S_i in hierarchy H, that agrees with X_ρ better than the other clusters. $F(C_r)$ is calculated using Eq. (15), where P_{S_i} is the precision (the number of objects in the cluster S_i belonging to the class X_ρ, divided by the number of objects in the cluster S_i) and R_{S_i} is the recall (the number of objects in the cluster S_i belonging to the class X_ρ, divided by number of objects in the class X_ρ).

$$F(C_r) = \max_{S_i \in H} \left\{ \frac{2P_{S_i}R_{S_i}}{P_{S_i} + R_{S_i}} \right\} \tag{15}$$

In the first step, we extract a predefined number of clusters for each dataset from the corresponding dendrogram, which is used as a representative data of the h-clustering result. In the next step, each data point is labeled according to the clustering result. Finally, the *Fscore* measure evaluates the similarity between the class label of every point in the dataset and the label, extracted from the ensemble result.

5.3. Comparison of the results on 2-dimensional labeled datasets

Table 9 demonstrates the *Fscore* results of applying standard clustering methods, *MATCH* and *Bob-Hic* on the datasets shown in Fig. 4. The computed *Fscore* values of each method on different datasets are averaged and shown in the last row of Table 9. This table reveals that the average *Fscore* value of *Bob-Hic* is significantly better than both non-ensemble standard methods and the *MATCH* ensemble method.

From the other point of view, it can be seen that *Bob-Hic* can get good clusters on the datasets which are hard to be clustered with

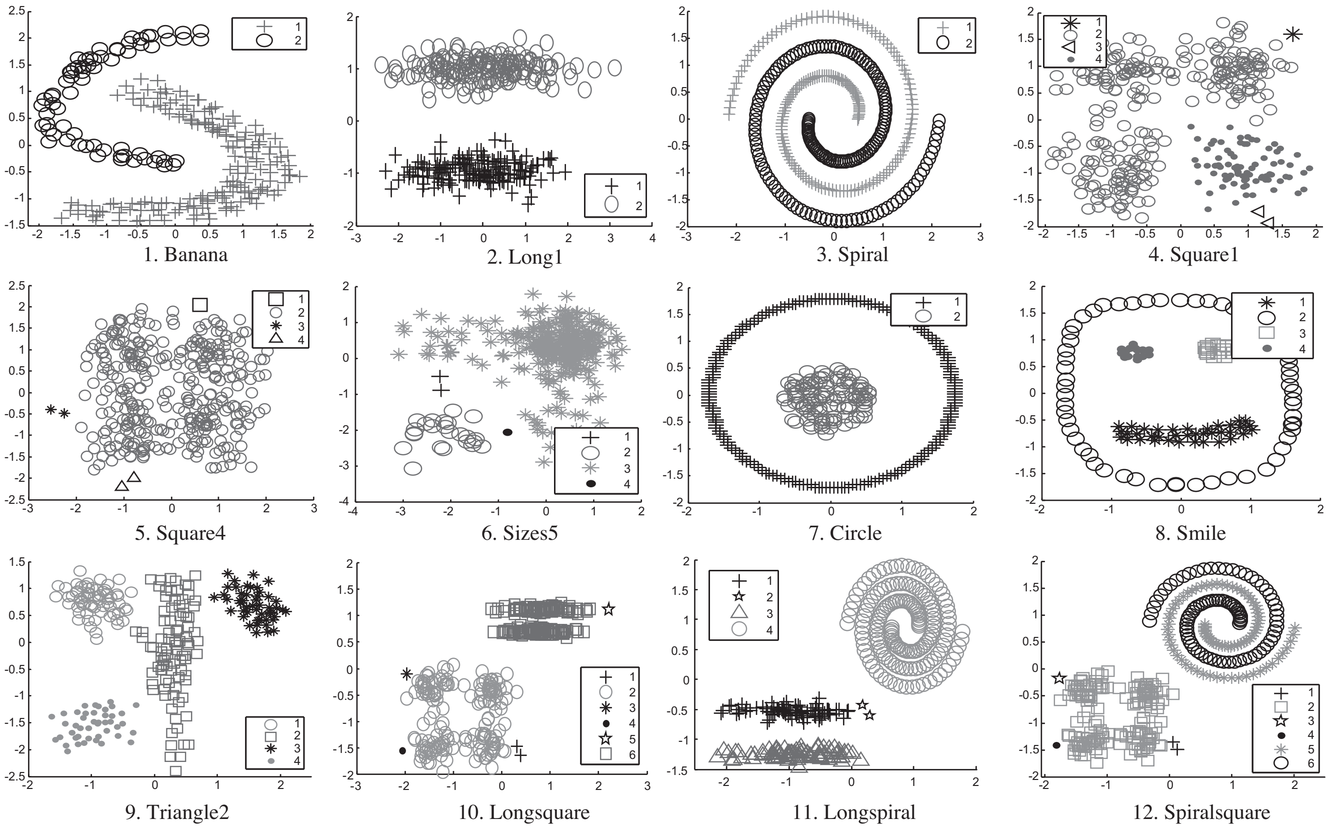

1. Banana
2. Long1
3. Spiral
4. Square1
5. Square4
6. Sizes5
7. Circle
8. Smile
9. Triangle2
10. Longsquare
11. Longspiral
12. Spiralsquare

Fig. 5. The resulting partition of experimental datasets shown in Fig. 3 by Single Linkage h-clustering method.

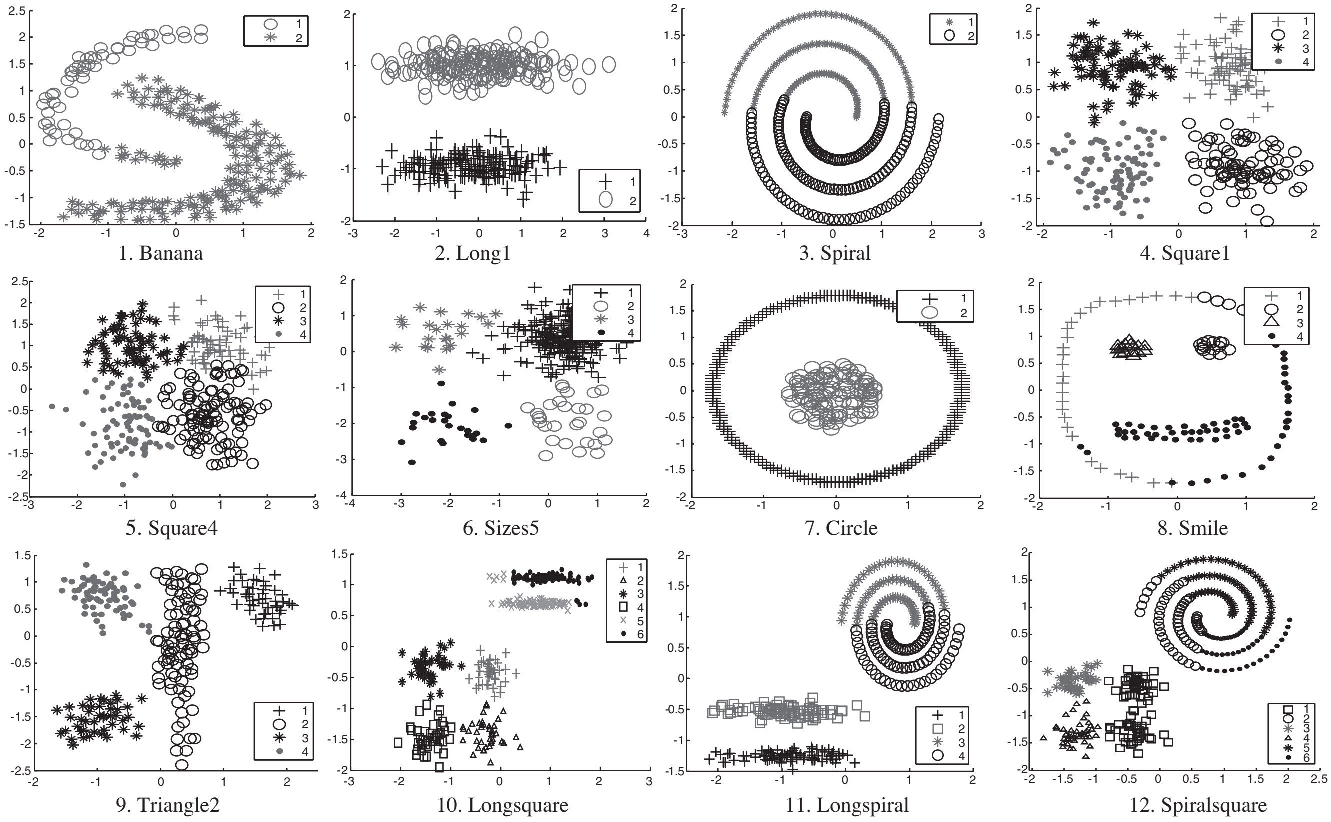

1. Banana
2. Long1
3. Spiral
4. Square1
5. Square4
6. Sizes5
7. Circle
8. Smile
9. Triangle2
10. Longsquare
11. Longspiral
12. Spiralsquare

Fig. 6. The resulting partition of experimental datasets shown in Fig. 3 by *Bob-Hic* clusterer ensemble algorithm.

Table 9
Comparison of *Fscore* values between preferred *Bob-Hic* and basic h-clustering approach and *MATCH*.

Dataset	Standard methods							*MATCH*	*Bob-Hic preferred*
	Centroid	Single	Average	Complete	Weighted	Median	Ward		
Banana	0.945	1.000	0.716	0.779	0.716	0.716	1.000	0.915	0.925
Long1	0.517	1.000	0.517	0.502	0.610	0.712	0.955	1.000	1.000
Spiral	0.680	1.000	0.576	0.568	0.552	0.680	0.640	0.570	0.626
Square1	0.984	0.203	0.981	0.982	0.863	0.760	0.982	0.972	0.982
Square4	0.907	0.204	0.900	0.788	0.782	0.687	0.853	0.844	0.916
Sizes5	0.989	0.761	0.982	0.910	0.955	0.981	0.711	0.969	0.979
Circle	0.637	1.000	0.637	0.637	0.637	0.637	0.574	1.000	1.000
Smile	0.527	1.000	0.527	0.527	0.661	0.535	0.551	0.521	0.614
Triangle2	0.946	0.995	0.955	0.800	0.930	0.942	0.897	0.945	0.955
Longsquare	0.745	0.338	0.763	0.748	0.755	0.753	0.779	0.619	0.743
Longspiral	0.516	0.501	0.515	0.549	0.511	0.609	0.730	0.722	0.813
Spiralsquare	0.590	0.600	0.538	0.526	0.538	0.824	0.661	0.654	0.719
Average *Fscore*	0.748	0.716	0.717	0.693	0.709	0.736	0.777	0.811	0.856

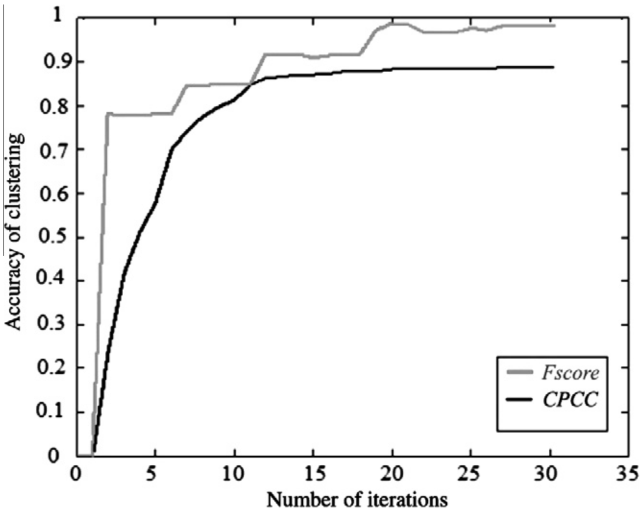


Fig. 7. Variation of two quality measures *CPCC* and *Fscore* at different iterations of *Bob-Hic* algorithm on the Size5 dataset.

connectedness based clustering algorithms (i.e. Square1, Square4 and Size5). Also, *Bob-Hic* gets good clusters on some datasets which are hard to be clustered with compactness based clustering algorithms (like Banana, Long1, Circle, Triangle2 and Longspiral). But, the results on some other datasets which include elongated clusters (like Smile, Spiral, Spiralsquare and Longsquare) are not so satisfying. This might be cause of the effect of the *Average* method which is used to recover the final dendrogram. Nevertheless, in these databases, the results are better than those algorithms which are based on compactness clustering, like the *Average* standard method.

The variation of the two quality measurements *CPCC* and *Fscore* versus iteration number is shown in Fig. 7. The result in this figure comes from applying the *Bob-Hic* algorithm on the Size5 dataset. This figure shows that the algorithm successfully improves both the *Fscore* measurement and *CPCC*.

6. Conclusions

In this paper a novel hierarchical clusterer ensemble method, based on the boosting theory, has been proposed. There are several motivations beyond this hierarchical ensemble method. To the best of our knowledge, numerous ensemble methods exists which construct a set of classifiers or partitional clusterers, whereas a few have been designed to handle the situation when a hierarchy of clusters is needed. In this study, we aim to address this issue.

Boosting is a popular classifier ensemble method that can be successful when applied to the clusterer ensembles. It is a multiple learner combination method that always uses resampling and reweighting. In order to reweight the samples, we have introduced a validation procedure to assess how well an individual data point has been clustered in the hierarchy.

Several real datasets have been used to illustrate that boosting is a good method to build hierarchical clusterer ensembles. Based on the experimental results, we have shown that the quality of the final clustering derived from *Bob-Hic* is superior to any of the clusterings performing alone. As a final note, although the proposed method, *Bob-Hic*, and the *MATCH* method, achieve the same clustering quality in many cases, we have proved that the time complexity of *Bob-Hic* is less than *MATCH*.

Acknowledgment

We thank Dr. Mehran Safayani from ECE Department, Isfahan University of Technology, and anonymous reviewers for their useful comments and language editing which have greatly improved the manuscript.

References

[1] D. Frossyniotis, A. Likas, A. Stafylopatis, A clustering method based on boosting, Pattern Recognition Letters 25 (2004) 641–654.
[2] T.G. Dietterich, Ensemble methods in machine learning, in: First International Workshop on Multiple Classifier Systems, Springer-Verlag, London, UK, 2000, pp. 1–15.
[3] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: 13th International Conference on Machine Learning, Bari, Italy, 1996, pp. 148–156.
[4] Y.M. Sun, Y. Wang, A.K.C. Wong, Boosting an associative classifier, IEEE Transactions on Knowledge and Data Engineering 18 (2006) 988–992.
[5] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[6] C. Romesburg, Cluster Analysis for Researchers, Lulu Press, North Carolina, 2004.
[7] S. Vega-pons, J. Ruiz-shulcloper, A survey of clustering ensemble algorithms, International Journal of Pattern Recognition and Artificial Intelligence 25 (2011) 333–372.
[8] E. Rashedi, A. Mirzaei, A novel multi-clustering method for hierarchical clusterings based on boosting, in: 19th Iranian Conference on Electrical Engineering, Tehran, Iran, 2011, pp. 1–4.
[9] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.
[10] S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure, Bioinformatics 19 (2003) 1090–1099.
[11] Y. Freund, R.E. Schapire, A short introduction to boosting, Journal of Japanese Society for Artificial Intelligence 14 (1999) 771–780.
[12] J.R. Quinlan, Bagging, Boosting, and C4.5, in: 13th National Conference on Artificial Intelligence, 1996, pp. 725–730.
[13] E.N. Adams, Consensus techniques and the comparison of taxonomic trees, Systematic Biology 21 (1972) 390–397.
[14] E.N. Adams, N-trees as nestings: complexity, similarity, and consensus, Journal of Classification 3 (1986) 299–317.

[15] D. Bryant, A classification of consensus methods for phylogenetics, DIMACS series in discrete mathematics and theoretical computer science 61 (2003) 163–184.
[16] M.M. Kulkarni, B.M.E. Moret, Consensus methods using phylogenetic databases, in: Computational Systems Bioinformatics Conference, IEEE, 2005, pp. 61–62.
[17] M. Al-Razgan, C. Domeniconi, Weighted clustering ensembles, in: 6th SIAM International Conference on Data Mining, 2006, pp. 258–269.
[18] A. Fred, A.K. Jain, Combining Multiple Clusterings Using Evidence Accumulation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 835–850.
[19] T.M. Hu, Y. Yu, J.Z. Xiong, S.Y. Sung, Maximum likelihood combination of multiple clusterings, Pattern Recognition Letters 27 (2006) 1457–1464.
[20] A.K. Jain, J.V. Moreau, Bootstrap technique in cluster analysis, Pattern Recognition 20 (1987) 547–568.
[21] A. Topchy, B. Minaei-Bidgoli, A.K. Jain, W.F. Punch, Adaptive clustering ensembles, in: 17th International Conference on Pattern Recognition 2004, pp. 272–275.
[22] J. Chang, D.M. Blei, Mixtures of clusterings by boosting, in: Learning Workshop, Hilton Clearwater, 2009.
[23] B. Minaei-bidgoli, E. Topchy, W.F. Punch, Ensembles of partitions via data resampling, in: International Conference on Information Technology, IEEE Computer Society, Washington, DC, USA, 2004, pp. 188–199.
[24] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1997) 119–139.
[25] M. Skurichina, R.P.W. Duin, Bagging for linear classifiers, Pattern Recognition 31 (1998) 909–930.
[26] Y. Freund, Boosting a weak learning algorithm by majority, Information and Computation 2 (1995) 256–285.
[27] B. Fischer, J.M. Buhmann, Bagging for path-based clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 1411–1415.
[28] H.G. Ayad, M.S. Kamel, On voting-based consensus of cluster ensembles, Pattern Recognition 43 (2010) 1943–1953.
[29] Z.H. Zhou, W. Tang, Clusterer ensemble, Knowledge-Based Systems 19 (2006) 77–83.
[30] A. Fred, Finding consistent clusters in data partitions, Multiple Classifier Systems (2001) 309–318.
[31] D. Frossyniotis, M. Pertselakis, A. Stafylopatis, A multi-clustering fusion algorithm, Methods and Applications of Artificial Intelligence 2308 (2002) 225–236.
[32] J.V.d. Olivera, W. Pedrycz, Advances in Fuzzy Clustering and Its Applications, Wiley, 2007.
[33] H.-S. Yoon, S.-Y. Ahn, S.-H. Lee, S.-B. Cho, J.H. Kim, Heterogeneous clustering ensemble method for combining different cluster results, BioDM 2006, LNBI, 3916 (2006) 82–92.
[34] A. Strehl, J. Ghosh, Cluster ensembles: a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.
[35] V. Filkov, S. Skiena, Integrating microarray data by consensus clustering, International Journal of Artificial Intelligence Tools 13 (2004) 863–880.
[36] A. Topchy, A.K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1866–1881.
[37] A. Topchy, A.K. Jain, W. Punch, A mixture model of clustering ensembles, SIAM International Conference of Data Mining, 2004, pp. 379–390.
[38] C. Domeniconi, M. Al-Razgan, Weighted cluster ensembles: methods and analysis, ACM Transactions on Knowledge Discovery from Data 2 (2009) 1–40.
[39] T. Li, C. Ding, M.I. Jordan, Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization, in: 7th International Conference of Data Mining (ICDM), IEEE Computer Society, Washington, DC, USA, 2007, pp. 577–582.
[40] A. Mirzaei, Combining hierarchical clusterings with emphasis on retaining the structural contents of the base clusterings, in: Computer Engineering & IT Department, Amir-kabir University of Technology, Tehran, 2009.
[41] A. Mirzaei, M. Rahmati, Combining hierarchical clusterings using min-transitive closure, in: 19th International Conference on Pattern Recognition, IEEE, Tampa, Florida, USA, 2008, pp. 1–4.
[42] A. Mirzaei, M. Rahmati, New strategies for solving cluster ensemble problems, in: 2nd Joint Conference on Fuzzy and Intelligent Systems, ISFS 2008, 2008.
[43] A. Mirzaei, M. Rahmati, A novel hierarchical-clustering-combination scheme based on fuzzy-similarity relations, IEEE Transactions on Fuzzy Systems 18 (2010) 27–39.
[44] A. Mirzaei, M. Rahmati, M. Ahmadi, A new method for hierarchical clustering combination, Intelligent Data Analysis 12 (2008) 549–571.
[45] J. Podani, Simulation of random dendrograms and comparison tests: some comments, Journal of Classification 17 (2000) 123–142.
[46] L. Zheng, T. Li, C. Ding, Hierarchical ensemble clustering, in: 10th International Conference on Data Mining, IEEE, Sydney, NSW, 2010, pp. 1199–1204.
[47] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[48] F. Murtagh, A survey of recent advances in hierarchical clustering algorithms, The Computer Journal 26 (1983) 354–359.
[49] C. Blake, C.J. Merz, University of California Irvine Repository of machine learning databases, Department of Information and Computer Science 55 (1998).
[50] L. Kuncheva, Real medical data sets: technical report, in: School of Informatics, University of Waless, Bangor, UK, 2005.
[51] R.R. Sokal, F.J. Rohlf, The comparison of dendrograms by objective methods, Taxon 11 (1962) 33–40.
[52] V.P. Lessig, Comparing cluster analyses with cophenetic correlation, Journal of Marketing Research 9 (1972) 82–84.
[53] F.J. Rohlf, D.R. Fisher, Tests for hierarchical structure in random data sets, Systematic Biology 17 (1968) 407.
[54] D.A. Freedman, Statistical Models: Theory and Practice, Cambridge University Press, 2005.
[55] D.B. Duncan, Multiple range and multiple *F* tests, Biometrics 11 (1995) 1–42.
[56] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Transactions on Evolutionary Computation 11 (2007) 56–76.
[57] Y. Zhao, G. Karypis, Evaluation of hierarchical clustering algorithms for document datasets, in: Proceeding of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA, 2002, pp. 515–524.