Artif Intell Rev (2014) 41:27–48
DOI 10.1007/s10462-011-9295-x

Effects of resampling method and adaptation on clustering ensemble efficacy

Behrouz Minaei-Bidgoli · Hamid Parvin · Hamid Alinejad-Rokny · Hosein Alizadeh · William F. Punch

Published online: 27 December 2011

Abstract Clustering ensembles combine multiple partitions of data into a single clustering solution of better quality. Inspired by the success of supervised bagging and boosting algorithms, we propose non-adaptive and adaptive resampling schemes for the integration of multiple independent and dependent clusterings. We investigate the effectiveness of bagging techniques, comparing the efficacy of sampling with and without replacement, in conjunction with several consensus algorithms. In our adaptive approach, individual partitions in the ensemble are sequentially generated by clustering specially selected subsamples of the given dataset. The sampling probability for each data point dynamically depends on the

This work is based an earlier work of Minaei-Bidgoli et al. (2004b).

B. Minaei-Bidgoli · H. Parvin · H. Alizadeh
Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

B. Minaei-Bidgoli
e-mail: b_minaei@iust.ac.ir

H. Parvin
e-mail: parvin@iust.ac.ir

H. Alizadeh
e-mail: halizadeh@iust.ac.ir

H. Alinejad-Rokny (✉)
Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
e-mail: alinejad@ualberta.ca

W. F. Punch
Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824, USA
e-mail: punch@cse.msu.edu

H. Alinejad-Rokny
7 Tir Street, Tirkhatir Street, Kafshgarkola Street, Imam Square, Ghaemshahr, Mazandaran, 4761764467, Iran

consistency of its previous assignments in the ensemble. New subsamples are then drawn to increasingly focus on the problematic regions of the input feature space. A measure of data point clustering consistency is therefore defined to guide this adaptation. Experimental results show improved stability and accuracy for clustering structures obtained via bootstrapping, subsampling, and adaptive techniques. A meaningful consensus partition for an entire set of data points emerges from multiple clusterings of bootstraps and subsamples. Subsamples of small size can reduce computational cost and measurement complexity for many unsupervised data mining tasks with distributed sources of data. This empirical study also compares the performance of adaptive and non-adaptive clustering ensembles using different consensus functions on a number of datasets. By focusing attention on the data points with the least consistent clustering assignments, whether one can better approximate the inter-cluster boundaries or can at least create diversity in boundaries and this results in improving clustering accuracy and convergence speed as a function of the number of partitions in the ensemble. The comparison of adaptive and non-adaptive approaches is a new avenue for research, and this study helps to pave the way for the useful application of distributed data mining methods.

Keywords Clustering ensembles · Consensus functions · Distributed data mining · Bootstrap · Subsampling · Adaptive clustering

1 Introduction

While some try to better solving of the clustering (Saha and Bandyopadhyay 2009), some others turn to ensemble methods. Exploratory data analysis and, in particular, data clustering can significantly benefit from combining multiple data partitions. Clustering ensembles can offer better solutions in terms of robustness, novelty and stability (Fred and Jain 2002, 2005; Strehl and Ghosh 2003; Topchy et al. 2003). Moreover, their parallelization capabilities are a natural fit for the demands of distributed data mining. Yet, achieving stability in the combination of multiple clusterings presents difficulties.

The combination of clusterings is a more challenging task than the combination of supervised classifications (Parvin et al. 2008a,b,c,d; Mohammadi et al. 2008). In the absence of labeled training data, we face a difficult correspondence problem between cluster labels in different partitions of an ensemble. Generating diversity in base classifier/clustering techniques is the common aspect between supervised and unsupervised combining approaches. Parvin et al. (2008a) have proposed a new way for generating diversity in classifiers which is called CCHR. CCHR emphasizes on crucial and boundary points during training phase. Recent studies (Topchy et al. 2004a) have demonstrated that consensus clustering can be found outside of voting-type situations using graph-based, statistical or information-theoretic methods without explicitly solving the label correspondence problem. Other empirical consensus functions were also considered in Dudoit and Fridlyand (2003), Fischer and Buhmann (2003), Fern and Brodley (2003). However, the problem of consensus clustering is known to be NP complete (Barthelemy and Leclerc 1995). Evaluation of a clustering algorithm can be done by matching final outputs of that algorithm and external desired output according to Tan et al. (2005), Jain and Dubes (1988).

Beside the consensus function, clustering ensembles need a partition generation procedure. Several methods are known to create partitions for clustering ensembles. For example, one can use:

(1) Different clustering algorithms (Strehl and Ghosh 2003),
(2) Different initializations – parameter values or built-in randomness of a specific clustering algorithm (Fred and Jain 2002, 2005),
(3) Different subsets of features (weak clustering algorithms) (Strehl and Ghosh 2003; Topchy et al. 2003),
(4) Different subsets of the original data (data resampling) (Strehl and Ghosh 2003; Dudoit and Fridlyand 2003; Fern and Brodley 2003; Minaei-Bidgoli et al. 2004a).

The focus of this paper is the last method, namely, the combination of clusterings using random samples of the original data. Conventional data resampling generates ensemble partitions independently; the probability of obtaining the ensemble consisting of B partitions $\{\pi_1, \pi_2, \ldots, \pi_B\}$ of the given data, D, can be factorized as:

$$p(\{\pi_1, \pi_2, \ldots, \pi_B\}|D) = \prod_{t=1}^{B} p(\pi_t|D). \tag{1}$$

Hence, the increased efficacy of an ensemble is mostly attributed to the number of independent, yet identically distributed partitions, assuming that a partition of data is treated as a random variable π. Even when the clusterings are generated sequentially, it is traditionally done without considering previously produced clusterings:

$$p(\pi_t|\pi_{t-1}, \pi_{t-2}, \ldots, \pi_1; D) = p(\pi_t|D). \tag{2}$$

However, similar to the ensembles of supervised classifiers using boosting algorithms (Breiman 1998), a more accurate consensus clustering can be obtained if contributing partitions take into account the previously determined solutions. Unfortunately, it is not possible to mechanically apply the decision fusion algorithms from the supervised (classification) to the unsupervised (clustering) domain. New objective functions for guiding partition generation and the subsequent decision integration process are necessary in order to guide further refinement. Frossyniotis et al. (2004) apply the general principle of boosting to provide a consistent partitioning of a dataset. At each boosting iteration, a new training set is created and the final clustering solution is produced by aggregating the multiple clustering results through a weighted voting.

In this paper, we propose a simple adaptive approach to partition generation that makes use of clustering history. In clustering, it is assumed that the ground truth in the form of class labels is not available. Therefore, we need an alternative measure of performance for an ensemble of partitions, during clustering process. We determine clustering consistency for data points by evaluating a history of cluster assignments for each data point within the generated sequence of partitions. Clustering consistency serves for adapting the data sampling to the current state of an ensemble during partition generation. The goal of adaptation is to improve confidence in cluster assignments by concentrating sampling distribution on problematic regions of the feature space. In other words, by focusing attention on the data points with the least consistent clustering assignments, one can better approximate (indirectly) the inter-cluster boundaries.

The main contribution of this paper is three-fold:

- To present a detailed taxonomy of clustering ensemble approaches (Sect. 2),
- To contribute to the new field of adaptive partitioning ensembles proposing a simple adaptive approach for partition generation (Sect. 4),
- And finally to provide a detailed comparison of bootstrap versus subsampling ensemble generation (Sect. 5).

The remainder of the paper is devoted to different consensus functions used in our experiments (Sect. 2), different algorithms for resampling schemes (Sects. 2, 3, 4), addressing the problems of estimation of clustering consistency for finding a consensus clustering (Sect. 4). Finally, we evaluate the performance of adaptive clustering ensembles (Sect. 5) on a number of real-world and artificial datasets in comparison with non-adaptive clustering ensembles of bootstrap partitions (Dudoit and Fridlyand 2003; Fischer and Buhmann 2003; Minaei-Bidgoli et al. 2004a; Topchy et al. 2004b).

2 Taxonomy of different clustering ensemble approaches

A growing number of techniques have been applied to clustering ensembles. A co-association consensus function was introduced for finding a combined partition in Fred and Jain (2002, 2005). The authors further studied combining k-means partitions with random initializations and a random number of clusters. Topchy et al. proposed new consensus functions related to intra-class variance criteria as well as the use of weak clustering components (Topchy et al. 2003, 2004a). Strehl and Ghosh (2003) have made a number of important contributions, such as their detailed study of hypergraph-based algorithms for finding consensus partitions as well as their object-distributed and feature-distributed formulations of the problem. They also examined the combination of partitions with a deterministic overlap of points between data subsets (non-random).

Resampling methods have been traditionally used to obtain more accurate estimates of data statistics. Efron generalized the concept of so-called "pseudo-samples" to sampling *with* replacement—the *bootstrap* method (Efron 1979). Resampling methods such as bagging have been successfully applied in the context of supervised learning (Breiman 1996). Jain and Moreau employed bootstrapping in cluster analysis to estimate the number of clusters in a multi-dimensional dataset as well as for evaluating cluster tendency/validity (Jain and Moreau 1987). A measure of consistency between two clusters is defined in Levine and Domany (2001). Data resampling has been used as a tool for estimating the validity of clustering (Dudoit and Fridlyand 2003; Ben-Hur et al. 2002) and its reliability (Roth et al. 2002; Fischer and Buhmann 2002).

The taxonomy of different consensus functions for clustering combination is shown in Fig. 1. This taxonomy presents solutions for the generative procedure as well. Details of the algorithms can be found in the listed references in Table 1.

It is a long-standing goal of clustering research to design scalable and efficient algorithms for large datasets (Zhang et al. 1996). One solution to the scaling problem is the parallelization of clustering by sharing processing among different processors (Zhang et al. 2000; Dhillon and Modha 2000). Recent research in data mining has considered a fusion of the results from multiple sources of data or from data features obtained in a distributed environment (Park and Kargupta 2003). Distributed data clustering deals with the combination of partitions from many data subsets (usually disjoint). The combined final clustering can be constructed centrally either by combining explicit cluster labels of data points or, implicitly, through the fusion of cluster prototypes (e.g., centroid-based). We analyze the first approach, namely, the clustering combination via consensus functions operating on multiple labelings of the different subsamples of a dataset. This study seeks to answer the question of the optimal size and granularity of the component partitions.

Here, among the generative mechanisms based on single algorithm, the Different subsets of objects, especially the resampling method is investigated in detail. Section 3.1 describes the non-adaptive methods for resampling. Also, the proposed adaptive resampling approach

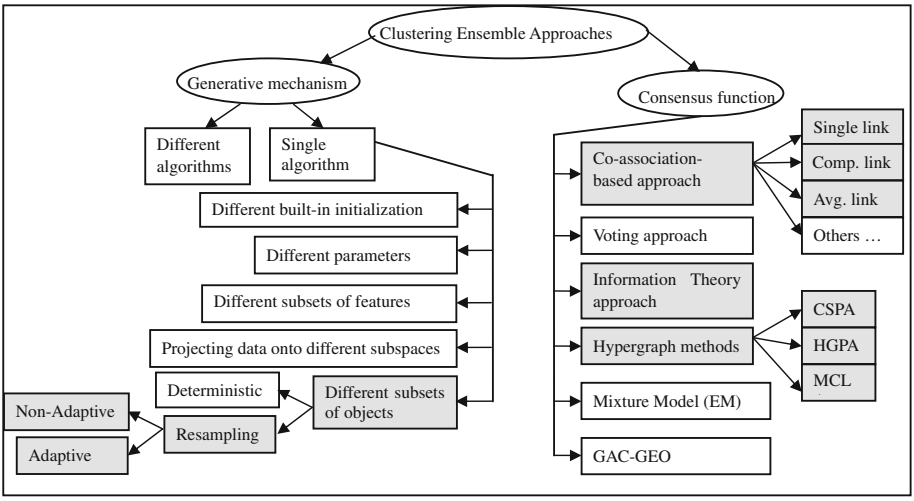

Fig. 1 Taxonomy of different approaches to clustering combination, *left side*: different approaches to obtain the diversity in clustering; *right side*: different consensus functions to find the clustering ensembles

is illustrated in Sect. 4. From the right side of Fig. 1, three kinds of consensus functions are applied in this paper, which are discussed in Sect. 3.2.

3 Non-adaptive sampling scheme

3.1 Non-adaptive resampling algorithms

Bootstrap (sampling with replacement) and subsampling (without replacement) can discern various statistics from replicate subsets of data while the samples in both cases are independent of each other. Our goal is to obtain a reliable clustering with measurable uncertainty from a set of different k-means partitions. The key idea of the approach is to integrate multiple partitions produced by clustering of pseudo-samples of a dataset.

Clustering combinations can be formalized as follows. Let D be a dataset of N data points in d-dimensional space. The input data can be represented as an $N \times d$ pattern matrix or $N \times N$ dissimilarity matrix, potentially in a non-metric space. Suppose that $X = \{X_1, \ldots, X_B\}$ is a set of B bootstrap samples of size N or subsamples of size $S < N$. A chosen clustering algorithm is run on each of the samples in X, which results in B partitions $\Pi = \{\pi_1, \pi_2, \ldots, \pi_B\}$. Each component partition in Π is a set of non-overlapping and exhaustive clusters with $\pi_i = \left\{C_1^i, C_2^i, \ldots, C_{K(i)}^i\right\}$, $X_i = C_1^i \cup \cdots \cup C_{K(i)}^i, \forall \pi_i$, where $k(i)$ is the number of clusters in the i-th partition.

The problem of combining partitions is to find a new partition $\sigma = \{C_1, \ldots, C_M\}$ of the entire dataset D given the partitions in Π, such that the data points in any cluster of σ are more similar to each other than to points in different clusters within σ. We assume that the number of clusters, M, in the consensus clustering is predefined and can be different from the number of clusters, k, in the ensemble partitions. In order to find the target partition σ, one needs a consensus function utilizing information from the partitions $\{\pi_i\}$. Several known consensus functions (Fred and Jain 2005; Strehl and Ghosh 2003; Topchy et al. 2003) can be

Table 1 Different approaches to clustering combination

Generative mechanisms (How to obtain different components?)

1. Apply various clustering algorithms (Strehl and Ghosh 2003)
2. Use a single algorithm
   - 2.1. Different built-in initialization (Topchy et al. 2003; Barthelemy and Leclerc 1995)
   - 2.2. Different parameters (Barthelemy and Leclerc 1995)
   - 2.3. Different subsets of data points
     - 2.3.1. Deterministic subsets (Strehl and Ghosh 2003)
     - 2.3.2. Resampling (Minaei-Bidgoli et al. 2004a,b; Topchy et al. 2004b; Jain and Dubes 1988)
       - 2.3.2.1. Non-Adaptive
         - 2.3.2.1.1. Bootstrap (Sampling with replacement)
         - 2.3.2.1.2. Subsampling (Sampling without replacement)
       - 2.3.2.2. Adaptive scheme (Topchy et al. 2004b; Frossyniotis et al. 2004)
   - 2.4. Projecting data onto different subspaces (Topchy et al. 2003; Fern and Brodley 2003)
   - 2.5. Different subset of features (Strehl and Ghosh 2003)

Consensus functions (How to integrate cluster ensemble?)

1. Using Co-association Matrix (Barthelemy and Leclerc 1995; Minaei-Bidgoli et al. 2004a; Jain and Dubes 1988)
   - 1.1. Single Link (SL)/ Minimum Spanning Tree (MST)
   - 1.2. Complete Link (CL)
   - 1.3. Average Link (AL)
   - 1.4. Ward, or other similarity based algorithms
2. (Hyper) Graph Partitioning (Strehl and Ghosh 2003)
   - 2.1. Hyper Graph Partition Algorithm (HGPA)
   - 2.2. Meta CLustering Algorithm (MCLA)
   - 2.3. Clustering Similarity Partition Algorithm (CSPA)
3. Information-theoretic methods, e.g. Quadratic Mutual Information (Topchy et al. 2003)
4. Voting Approach (Minaei-Bidgoli et al. 2004a)
5. Mixture Model (Breiman 1998)
6. Generic agglomerative clustering framework, GAC-GEO (Jiamthapthaksin et al. 2010)

employed to map a given set of partitions $\Pi = \{\pi_1, \pi_2, \ldots, \pi_B\}$ to the target partition, σ, in our study.

3.2 Consensus functions

A consensus function maps a given set of partitions $\Pi = \{\pi_1, \ldots, \pi_B\}$ to a target partition σ. In this paper we have employed three types of consensus functions:

3.2.1 Co-association based functions

This consensus function operates on the co-association matrix. Similarity between points (co-association values) can be estimated by the number of clusters shared by two points in all the partitions of an ensemble.

Table 2 Clustering ensemble, based on co-association matrix and using different similarity-based consensus functions

Input: D—the input dataset N points
B—number of partitions to be combined
M—number of clusters in the final partition, σ
k—number of clusters in the components of the combination
Γ—a similarity-based clustering algorithm
for $j = 1$ to B
  Draw a random pseudosample X_j
  Cluster the sample $X_j : \pi(i) \leftarrow k - \text{means}(\{X_j\})$
  Update similarity values (co-association matrix) for all patterns in X_j
end
Combine partitions via chosen $\Gamma : \sigma \leftarrow \Gamma(P)$
Validate final partition, σ (optional)
return σ *// consensus partition*

The similarity between two objects, x and y, is defined as follows:

$$sim(x, y) = \frac{1}{B} \sum_{i=1}^{B} \delta\left(\pi_i(x), \pi_i(y)\right), \quad \delta(a, b) = \begin{cases} 1, \text{ if } a = b \\ 0, \text{ if } a \neq b \end{cases} \tag{3}$$

Similarity between a pair of objects simply counts the number of clusters shared by the objects in the partitions $\{\pi_1, \ldots, \pi_B\}$. In this type, similarity-based clustering algorithms are used as the consensus function, Γ.

A numerous hierarchical agglomerative algorithms (criteria) can be applied to the co-association matrix to obtain the final partition, σ, including Single Link (SL), Average Link (AL) and Complete Link (CL) (Jain and Dubes 1988).

The general pseudocode of these algorithms is shown in Table 2.

There are three main drawbacks to this approach.

- First, it has a quadratic computational complexity in the number of patterns and features $O(kN^2d^2)$ (Duda et al. 2001), where k is the number of clusters, N is the number of data points, and d is the number of features.
- Second, there are no established guidelines for which clustering algorithm should be applied, e.g. single linkage or complete linkage.
- Third, an ensemble with a small number of partitions may not provide a reliable estimate of the co-association values (Topchy et al. 2004a).

3.2.2 Quadratic mutual information algorithm (QMI)

Assuming that the partitions are independent, a consensus function based on k-means clustering in the space of standardized features can effectively maximize a generalized definition of mutual information (Topchy et al. 2003) or even each one of other similarities measurements of partitioning (Pfitzner et al. 2009). The complexity of this consensus function is $O(kNB)$, where k is the number of clusters, N is the number of items, and B is the number of partitions. Though the QMI algorithm can be potentially trapped in a local optimum, its relatively low computational complexity allows the use of multiple restarts in order to choose a quality consensus solution with minimum intra-cluster variance.

3.2.3 Hypergraph partitioning

The clusters could be represented as hyperedges on a graph whose vertices correspond to the data points to be clustered. The problem of consensus clustering is then reduced to finding the minimum-cut of the resulting hypergraph. The minimum k-cut of this hypergraph into k components gives the required consensus partition (Strehl and Ghosh 2003). Hypergraph algorithms seem to work effectively for approximately balanced clusters. Though the hypergraph partitioning problem is NP-hard, efficient heuristics to solve the k-way min-cut partitioning problem are known, i.e. the complexity of CSPA, HGPA and MCLA is estimated in Strehl and Ghosh (2003) as $O(kN^2B)$, $O(kNB)$, and $O(k^2NB^2)$, respectively. These hypergraph algorithms are described in Strehl and Ghosh (2003) and their corresponding source codes are available at http://www.strehl.com. A drawback of hypergraph algorithms is that they seem to work the best for nearly balanced clusters (Topchy et al. 2004a).

The performance of all these consensus methods is empirically analyzed as a function of two important parameters: the type of sampling process (sample redundancy) and the granularity of each partition (number of clusters).

4 Adaptive sampling scheme

While there are many ways to construct diverse data partitions for an ensemble, not all of them easily generalize to adaptive clustering. The adaptive approach (Topchy et al. 2004b) extends the studies of ensembles whose partitions are generated via data resampling (Dudoit and Fridlyand 2003; Fischer and Buhmann 2003; Minaei-Bidgoli et al. 2004a,b). Though, intuitively, clustering ensembles generated by other methods also can be boosted. The adaptive partition generation mechanism as discussed by Breiman (1996, 1998) is aimed at reducing the variance of inter-class decision boundaries. Unlike the regular bootstrap method that draws subsamples uniformly from a given dataset, adaptive sampling favors points from regions close to the decision boundaries. At the same time, the points located far from the boundary regions are sampled less frequently. It is instructive to consider a simple example that shows the difference between ensembles of bootstrap partitions with and without the weighted sampling.

Figure 2 shows how different decision boundaries can separate two natural classes depending on the sampling probabilities. Here we assume that the k-means clustering algorithm is applied to the subsamples. In this Figure, it is inferred that the clustering boundaries can gradually inclined to learn boundary datapoints. Maybe we can't claim that this process betters performance of clustering algorithm, but we can say that this generates more diverse partitions during adaptive algorithm than previous non-adaptive algorithms. According to Frossyniotis et al. (2004), the more diverse the components existing in the ensembles, the more accurate the ensembles.

4.1 Resampling

Initially, all the data points have the same weights, namely, the sampling probability $p_i = 1/N, i \in [1, \ldots, N]$. Clearly, the main contribution to the clustering error is due to the sampling variation that causes inaccurate inter-cluster boundaries. Solution variance can be significantly reduced if sampling is increasingly concentrated only on the subset of objects at iterations $t_2 > t_1 > t_0$, as demonstrated in Fig. 2.

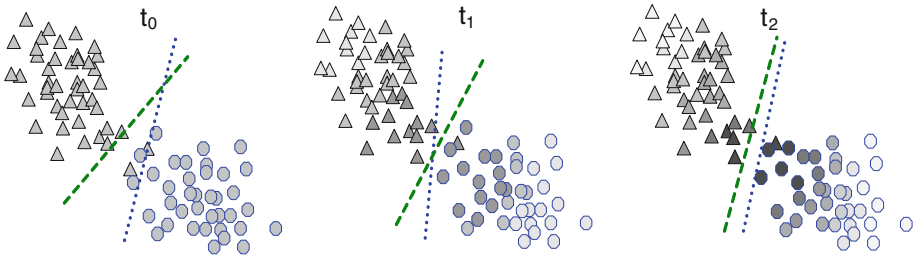

Fig. 2 Two possible decision boundaries for a 2-cluster data set. Sampling probabilities of data points are indicated by *gray* level intensity at different iterations ($t_0 < t_1 < t_2$) of the adaptive sampling. True components in the 2-class mixture are shown as *circles* and *triangles*. reduction difference between the two decision boundaries over the time in the future iterations, results in reduction of their variance

The key issue in the design of the adaptation mechanism is the updating of probabilities. We have to decide how and which data points should be sampled as we collect more and more clusterings in the ensemble. A consensus function based on the co-association values (Fred and Jain 2002, 2005) provides the necessary guidelines for adjustments of sampling probabilities. Remember that the co-association similarity between two data points, x and y, is defined as the number of clusters shared by these points in the partitions of an ensemble, Π:

A consensus clustering can be found by using an agglomerative clustering algorithm (e.g., single linkage) applied to such a co-association matrix constructed from all the points. The quality of the consensus solution depends on the accuracy of similarity values as estimated by the co-association values. The least reliable co-association values come from the points located in the problematic areas of the feature space. Therefore, our adaptive strategy is to increase the sampling probability for such points as we proceed with the generation of different partitions in the ensemble.

4.2 Relabeling

The sampling probability can be adjusted not only by analyzing the co-association matrix, which is of quadratic complexity $O(N^2)$, but also by applying the less expensive $O(BNK + BK^3)$ estimation of clustering consistency for the data points, where B is number of those partitions that need to be combined, K is number of clusters in the partitions of the ensemble and N is the size of the original data sample. Again, the motivation is that the points with the least stable cluster assignments, namely those that frequently change the cluster they are assigned to, require an increased presence in the data subsamples. In this case, a label correspondence problem must be approximately solved to obtain the same labeling of clusters throughout the ensemble's partitions. By default, the cluster labels in different partitions are arbitrary. To make the correspondence problem more tractable, one needs to re-label each partition in the ensemble using some fixed reference partition. Table 3 illustrates how four different partitions of twelve points can be re-labeled using the first partition as a reference.

At the $(t + 1)$-th iteration, when some t different clusterings are already included in the ensemble, we use the Hungarian algorithm for minimal weight bipartite matching problem in order to re-label the $(t + 1)$-th partition. Note that the order of the Hungarian algorithm for re-labeling the $(t + 1)$-th partition is $O(K^3)$, Where K is number of clusters in the $(t + 1)$-th partition of the ensemble. Besides running the Hungarian algorithm for all B partitions plus the analysis of the mapped results, results in the above time order, $O(BNK + BK^3)$.

Table 3 Consistent re-labeling of 4 partitions of 12 objects

	π_1	π_2	π_3	π_4	π'_1	π'_2	π'_3	π'_4	Consistency
$\mathbf{x}_1$	2	*B*	*X*	α	2	1	2	1	0.5
$\mathbf{x}_2$	2	*A*	*X*	α	2	2	2	1	0.75
$\mathbf{x}_3$	2	*A*	*Y*	β	2	2	1	2	0.75
$\mathbf{x}_4$	2	*B*	*X*	β	2	1	2	2	0.75
$\mathbf{x}_5$	1	*A*	*X*	β	1	2	2	2	0.75
$\mathbf{x}_6$	2	*A*	*Y*	β	2	2	1	2	0.75
$\mathbf{x}_7$	2	*B*	*X*	α	2	1	2	1	0.5
$\mathbf{x}_8$	1	*B*	*Y*	α	1	1	1	1	1
$\mathbf{x}_9$	1	*B*	*Y*	β	1	1	1	2	0.75
$\mathbf{x}_{10}$	1	*A*	*Y*	α	1	2	1	1	0.75
$\mathbf{x}_{11}$	2	*B*	*Y*	α	2	1	1	1	0.75
$\mathbf{x}_{12}$	1	*B*	*Y*	α	1	1	1	1	1

Table 4 An illustrative example of re-labeling difficulty involving five data points and four different clusterings of four bootstrap samples

	P_1	P_2	P_3	P_4
x_1	1	?	2	3
x_2	1	2	3	1
x_3	?	2	?	2
x_4	?	?	1	?
x_5	2	1	3	1

The numbers represent the labels assigned to the objects and the "?" shows the missing labels of data points in the bootstrapped samples

There are some problems in relabeling of the resampled data partitions. In all resampling methods, some of the objects are missed in drawn samples. When one uses co-association based methods, this poses no difficulty because the co-association values are only updated for existing objects. However, missing labels can cause a number of problems for other consensus functions. For example, when an object is missing in a bootstrap sample, there will be no label assigned to it after running the clustering algorithm. Thus, special consideration of the missing labels is necessary during the process of re-labeling, before running a consensus function.

We must consider how to re-label two bootstrap samples with missing values. When the number of objects in the drawn samples is too small, this problem becomes harder. For example, consider four partitions, $P_1, \ldots, P_4$ for five data points $x_1, \ldots, x_5$ as shown in Table 4.

One can re-label the above partitions in relation to some reference partition. However, the missing labels should not be considered in the re-labeling process. Therefore, if the reference partition is P_1, and we want to re-label P_2, then only the data points x_2 and x_5 participate in the re-labeling process. Similarly, if P_3 is re-labeled based on the reference P_1, then x_1, x_2 and x_5 are used in the Hungarian algorithm to find the best match. Once the best agreement

among the given labels is found then all the objects in the partition, except those with missing labels, are re-labeled.

4.3 Consistency index

As an outcome of the re-labeling procedure, we can compute the consistency index of clustering for each data point. Clustering consistency index CI at iteration t for a point x is defined as the ratio of the maximal number of times the object is assigned in a certain cluster to the total number of partitions:

$$CI(x) = \frac{1}{B} \max \left\{ \sum_{i=1}^{B} \delta(\pi_i'(x), L) \right\}_{[L \in cluster\_labels]} \tag{4}$$

The values of consistency indices are shown in Table 3 after four partitions were generated and re-labeled. We should note that clustering of subsamples of the dataset, D, does not provide the labels for the objects missing (not drawn) in some subsamples. In this situation, the summation in Eq. (5) skips the terms containing the missing labels.

The clustering consistency index of a point can be directly used to compute its sampling probability. In particular, the probability value is adjusted at each iteration as follows:

$$p_{t+1}(x) = Z(\alpha p_t(x) + 1 - CI(x)), \tag{5}$$

where α is a discount constant for the current sampling probability and Z is a normalization factor. Empirically, the discount constant was set to $\alpha = 0.3$ in our experiments.

At each iteration, the coassociation matrix is updated. The results of relations between points from all B partitions are accumulated in the final coassociation matrix. As the sampling is focused on the problematic areas of the data, more important information is stored in this matrix rather than the coassociation matrix obtained from the non-adaptive approaches. The proposed clustering ensemble algorithm is summarized in pseudocode in Table 5.

5 Experimental study and discussion

The experiments were performed on several datasets, including two challenging artificial problem, the "Halfrings" dataset, and the "2-Spiral" dataset, two datasets from UCI repository, the "Iris" and "Wine" and two other real world dataset, the "LON" and "Star/Galaxy" datasets. A summary of dataset characteristics is shown in Table 6.

5.1 Datasets

The Halfrings and 2-Spiral datasets, as shown in Fig. 3, consist of two clusters, an unbalanced clusters with 100- and 300-point patterns in the Halfrings dataset and a balanced clusters in the 2-Spiral. The k-means algorithm by itself is not able to detect the two natural clusters since it implicitly assumes them as hyperspherical clusters. 3-Gaussian is a simulated dataset which includes three unbalanced classes with 50, 100, and 150 data points. The Wine dataset described in Aeberhard et al. (1992) contains special features of the chemical composition of wines grown in the same region but derived from three different cultivars. The patterns are described by the quantities of thirteen constituents (features) found in each of the three types of wines. There are 178 samples in total.

Table 5 Algorithms for adaptive clustering ensembles

Input: D—dataset of N points
B—number of partitions to be combined
M —number of clusters in the consensus partition σ
K—number of clusters in the partitions of the ensemble
Γ—chosen consensus function operating on cluster labels
p—sampling probabilities (initialized to $1/N$ for all the points)
Reference Partition $\leftarrow$ k-means(D)
for $i = 1$ to B
  Draw a subsample X_i from D using sampling probabilities **p**
  Cluster the sample X_i : $\pi(i) \leftarrow k$-means (X_i)
  Update the coassociation matrix
  Re-label partition $\pi(i)$ using the reference partition
  Compute the consistency indices for the data points in D
  Adjust the sampling probabilities **p**
end
Apply consensus function Γ to ensemble Π to find the partition σ
Validate the target partition σ (optional)
return σ *// consensus partition*

Table 6 A summary of datasets characteristics

	No. of classes	No. of features	No. of patterns	Patterns per class
Star/Galaxy	2	14	4, 192	2082-2110
Wine	3	13	178	59-71-48
LON	2	6	227	64-163
Iris	3	4	150	50-50-50
3-Gaussian	3	2	300	50-100-150
Halfrings	2	2	400	100-300
2-Spirals	2	2	200	100-100

The LON dataset (Minaei-Bidgoli and Punch 2003) is extracted from the activity log in a web-based course using an online educational system developed at Michigan State University (MSU): the Learning Online Network with Computer-Assisted Personalized Approach (LON-CAPA[1]). The dataset includes the student and course information on an introductory physics course (PHY183), collected during the spring semester 2002. This course included 12 homework sets with a total of 184 problems, all were completed online using LON-CAPA. The dataset consists of 227 student records from one of the two groups: "Passed" for the grades above 2.0, and "Failed" otherwise. Each sample contains 6 features.

The Iris dataset contains 150 samples in 3 classes of 50 samples each, where each class refers to a type of iris plant. One class is linearly separable from the others, and each sample has four continuous-valued features. The Star/Galaxy dataset described in Odewahn et al. (1992) has a significantly larger number of samples ($N = 4, 192$) and features ($d = 14$). The

[1] See http://www.lon-capa.org.

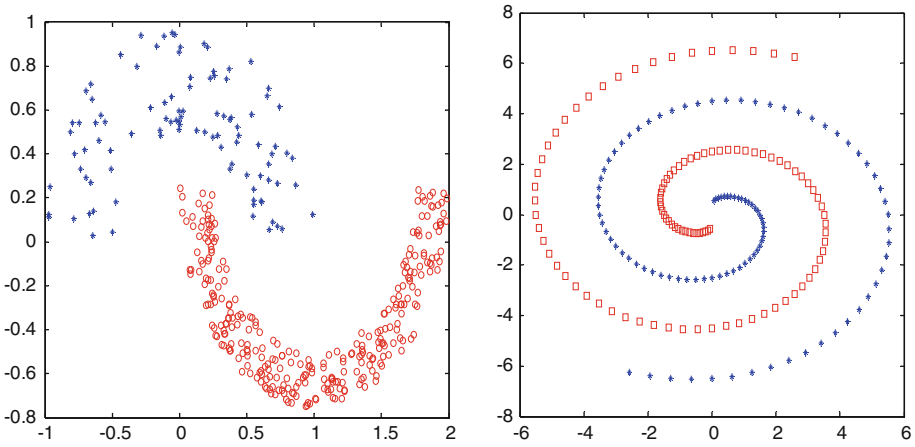

Fig. 3 "Halfrings" dataset with 400 patterns (100-300 per class), "2-Spirals" dataset with 200 patterns (100-100 per class)

task is to separate observed objects into stars or galaxies. Domain experts manually provided true labels for these objects.

For all these datasets the number of clusters, and their assignments, are known. Therefore, one can use the misassignment (error) rate of the final combined partition as a measure of performance of clustering combination quality; however, it is obvious that the true labels of data point can not be used in the clustering process. One can determine the error rate after solving the correspondence problem between the labels of derived and known clusters. The Hungarian method for solving the minimal weight bipartite matching problem can efficiently solve this label correspondence problem.

5.2 The empirical results on non-adaptive approaches

The bootstrap experiments probe the accuracy of partition combination as a function of the resolution of partitions (k value) and the number of partitions, B (number of partitions to be aggregated).

One of our goals was to determine the satisfactory minimum number of bootstrap samples, B, necessary to form high-quality combined cluster solutions. In addition, different values of k in the k-means algorithm provide different levels of resolution for the partitions in the combinations. We studied the dependency of the overall performance on the number of clusters, k. In particular, clustering on the bootstrapped samples was performed for B values in the range [5, 1,000] and the k values in the interval [2, 20].

Analogously, the size of the pseudosample, S, in subsampling experiments is another important parameter. Our experiments were performed on different subsample sizes in the interval $[N/20, 3N/4]$, where N is the size of the original data set.In the case of the Halfrings, S is taken in the range [20, 300] where the original sample size is $N=400$, while in the case of the Galaxy dataset, parameter S was varied in the range [200, 3,000] with $N=4{,}192$. Therefore, in resampling without replacement, we analyze how the clustering accuracy is influenced by these three parameters: number of clusters, k, in every clustering, number of drawn samples, B, and the sample size, S. Note that all the experiments are repeated 20 times and the average error rate for 20 independent runs is reported, except for the Star/Galaxy dataset where only 10 runs are carried out.

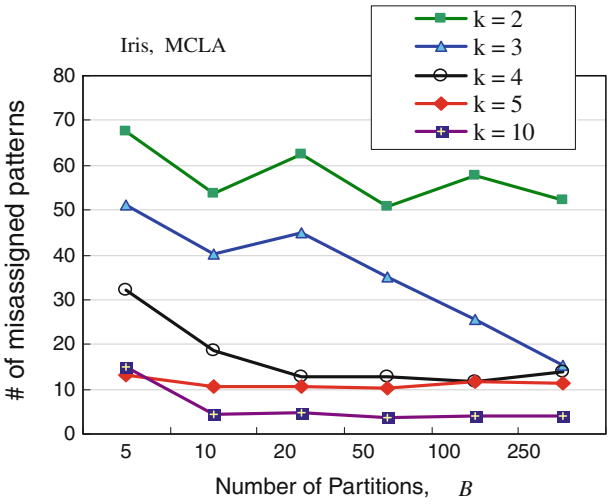

Fig. 4 "Iris" dataset. Bootstrapping for fixed consensus function MCLA, different B, and different values of k

The experiments employed eight different consensus functions: co-association based functions (single link, average link, and complete link), hypergraph algorithms (HGPA, CSPA, MCLA), as well as the QMI algorithm.

5.2.1 The role of consensus functions and algorithm's parameters (bootstrap algorithm)

Perhaps the most important single design element of the combination algorithm is the choice of a consensus function.

In the Halfrings dataset the true structure of the dataset (100% accuracy) was obtained using co-association based consensus functions (both single and average link) in when $k = 10$ and the number of partitions took part in the combination was $B \cdot 100$. None of the other six consensus methods converged to an acceptable error rate for this dataset.

For the Wine dataset an optimal accuracy of 73% was obtained with both the hypergraph-CSPA algorithm and co-association based method using average link (AL) with different parameters as shown in Table 10. For the LON dataset the optimal accuracy of 79% was achieved only by co-association-based (using the AL algorithm) consensus function. This accuracy is comparable to the results of the k-NN classifier, multilayer perceptron, naïve Bayes classifier, and some other algorithms when the "*LON*" dataset is classified in a supervised framework based on labeled patterns (Minaei-Bidgoli and Punch 2003).

For the "*Iris*" dataset, the hypergraph consensus function, HPGA algorithm led to the best results when $k \cdot 10$. The AL and the QMI algorithms also gave acceptable results, while the single link and average link did not demonstrate a reasonable convergence. Figure 4 shows that the optimal solution could not be found for the Iris dataset with k in the range [2, 5], while the optimum was reached for $k \cdot 10$ with only $B \cdot 10$ partitions.

For the Star/Galaxy dataset the CSPA function (similarity based hypergraph algorithm) could not be used due to its computational complexity because it has a quadratic complexity in the number of patterns $O(kN^2B)$.

The HGPA function did not converge at all in any condition. Independent from the parameter values, k and B, the SL function did not also converge at all, as shown in Table 7, also you

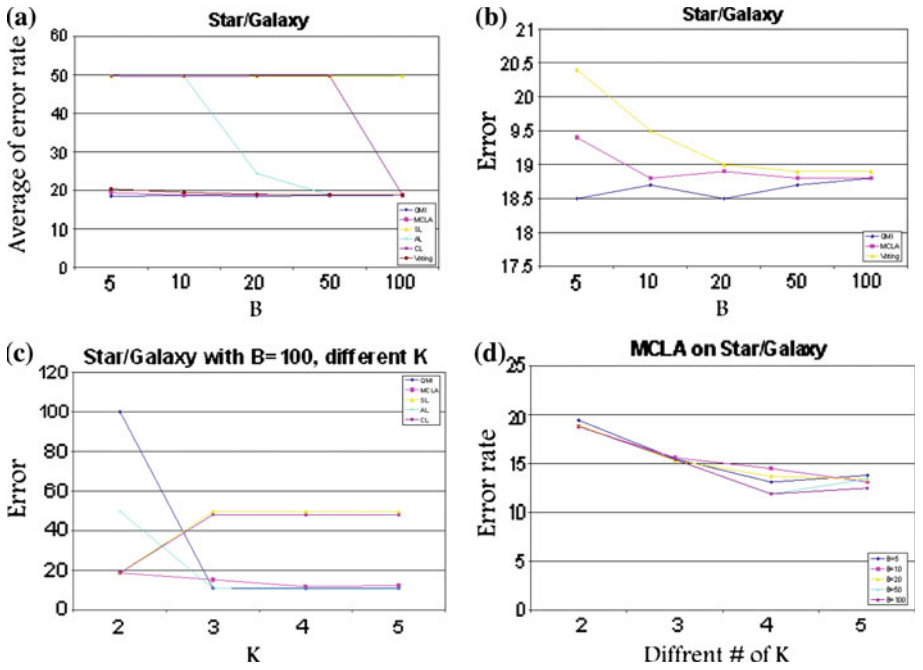

Fig. 5 The effect of different parameters over clustering performance on Star/Galaxy dataset. **a** Effect of partition number on error rate of clustering using different consensus functions while $K = 2$. **b** Effect of partition number on error rate of clustering using different consensus functions while $K = 2$. **c** Effect of cluster number on error rate of clustering using different consensus functions while $B = 100$. **d** Determining the best value of K applying different value of B

can refer to Figs. 6a, c for having a better view. Also, CL did not yield the optimal solutions. However, the MCLA, the QMI and the AL functions led to an error rate of approximately 10%, which is better than the performance of an individual k-means result (21%).

The major problem in co-association based functions is that they are computationally expensive. The complexity of these functions is very high ($O(kN^2d^2)$) and therefore, it is not effective to use the co-association based functions as a consensus function for the large datasets.

Note that the QMI algorithm did not work well when the number of partitions exceeded 200, especially when the k value was large. This might be due to the fact that the core of the QMI algorithm operates in $k \times B$-dimensional space. The performance of the k-means algorithm degrades considerably when B is large (>100) and, therefore, the QMI algorithm should be used with smaller values of B.

The question of the best parameter setting is comparatively difficult to answer. As it is inferred from Fig. 4, for higher values of k and B, we have better clustering results over Iris dataset. According to this figure, we can claim that that in order to minimize the B value we can follow the following rule of thumb: "assuming fixed value m (m is equal to two or three times of real cluster number), the higher values of B will lead to the better results. But when real cluster number isn't defined previously $B = 10$ can be a good option".

Also according to Fig. 5, we can deduce that in Star/Galaxy dataset results differ depending on employed consensus function. But like before, the greater value of B, the better results. The best value of k in Star/Galaxy is $k = 3$. And also the best value of B over this dataset is

Table 7 "Star/Galaxy" data experiments

K	*B*	QMI	MCLA	SL	AL	CL
2	5	18.5	19.4	49.7	49.7	49.7
2	10	18.7	18.8	49.6	49.6	49.6
2	20	18.5	18.9	49.6	24.4	49.7
2	50	18.7	18.8	49.6	18.8	49.7
2	100	18.8	18.8	49.7	18.8	18.8
3	5	13.4	15.5	49.7	49.7	49.7
3	10	17.8	15.6	49.6	49.6	49.6
3	20	11.5	15.3	49.7	18.8	42.9
3	50	13.3	15.4	49.7	11	35.9
3	100	11	15.4	49.7	11	48.2
4	5	15.2	13.1	49.7	49.7	49.7
4	10	11.4	14.5	49.6	49.7	49.7
4	20	14	13.7	49.6	24.3	48.7
4	50	22.2	11.9	49.7	10.7	48
4	100	11	11.9	49.7	10.7	47.9
5	5	14.9	13.8	49.7	49.7	49.7
5	10	14.9	13.1	49.7	47.9	49.6
5	20	10.7	13.4	49.6	11	49.7
5	50	11.4	13.4	49.7	10.8	48.7
5	100	11	12.5	49.7	10.9	48

Average error rate (% over 10 runs) of clustering combination using resampling algorithms with different number of components in combination *B*, resolutions of components, *k*, and types of consensus functions

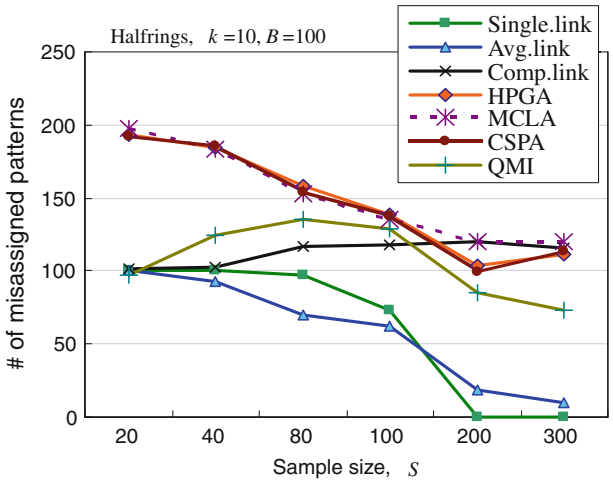

Fig. 6 "Halfrings" dataset. Experiments using subsampling with $k = 10$ and $B = 100$, different consensus function, and sample sizes S

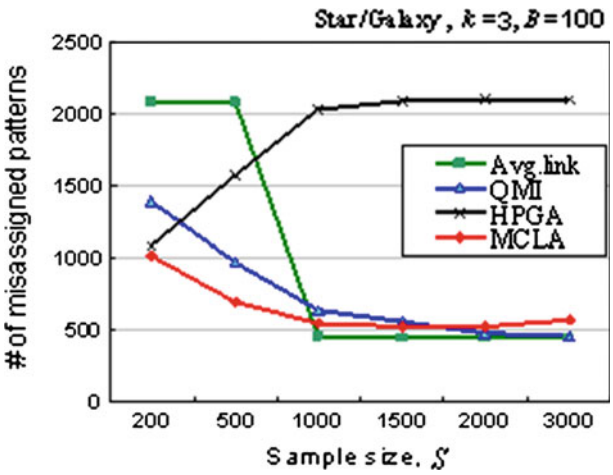

Fig. 7 "Star/Galaxy" dataset. Experiments using subsampling, with $k = 3$ and $B = 100$ and different consensus function and sample sizes S

Table 8 The average error rate (%) of classical clustering algorithms

Dataset	k-means	Single link (%)	Complete link (%)	Average link (%)
Halfrings	25	24.3	14	5.3
Iris	15.1	32	16	9.3
Wine	30.2	56.7	32.6	42
LON	27	27.3	25.6	27.3
Star/Galaxy	21	49.7	44.1	49.7

An average over 100 independent runs is reported for the k-means algorithms

$B = 100$. This matter is evaluated for four other datasets (Wine, Halfrings, 3Gaussian and StarGalaxy) in Fig. 8.

The question of the best consensus function remains open for further study. Each consensus function explores the structure of dataset in different ways, thus its efficiency greatly depends on different types of existing structure in the dataset. One can suggest having several consensus functions and then combining the consensus function results through maximizing mutual information (Strehl and Ghosh 2003), but running different consensus functions on large datasets would be computationally expensive.

5.2.2 Effect of the resampling method (bootstrap vs. subsampling)

In subsampling or bootstrap, with a smaller size of the S we face a lower complexity of the k-means clustering. Therefore decreasing the size of data to be clustered can results in much smaller complexity in the whole process of cluster ensembles. Comparing the results of the bootstrap and the subsampling methods shows that when the bootstrap technique converges to an optimal solution, the optimal result could be obtained by the subsampling as well, but subsampling method rather bootstrap method needs a smaller size of the data points, i.e. S. For example, in the Halfrings dataset the perfect clustering can be obtained using a single-link consensus function with $k = 10$, $B = 100$ and $S \cdot 200$ (50% data size) as shown

Table 9 Summary of the best results of subsampling methods

Dataset	Best consensus function (s)	Lowest error rate obtained (%)	Parameters
Halfrings	Co-association, SL	0	$k \cdot 10, B \cdot 100$
	Co-association, AL	0	$k \cdot 15, B \cdot 100$
Iris	Hypergraph-HGPA	2.7	$k \cdot 10, B \cdot 20$
Wine	Hypergraph-CSPA	26.8	$k \cdot 10, B \cdot 20$
	Co-association, AL	27.9	$k \cdot 4, B \cdot 100$
LON	Co-association, CL	21.1	$k \cdot 4, B \cdot 100$
Galaxy/ Star	Hypergraph-MCLA	9.5	$k \cdot 20, B \cdot 10$
	Co-association, AL	10	$k \cdot 10, B \cdot 100$
	Mutual Information	11	$k \cdot 3, B \cdot 20$

Table 10 Bootstrap methods: trade-off among the values of k, the number of partitions B, and the sample size, S

Dataset	Best consensus function (s)	Lowest error rate (%)	k	B	S	% of entire data
Halfrings	SL	0	10	100	300	**75**
	SL	0	10	150	200	**50**
	SL	0	10	400	80	**20**
	AL	0	15	1000	80	20
	AL	0	20	500	100	25
Iris	HGPA	2.3	10	100	50	33
	HGPA	2.1	15	50	50	**33**
Wine	AL	27.5	4	50	100	56
	HPGA	28	4	50	20	**11**
	CSPA	27.5	10	20	50	28
LON	CL	21.5	4	500	100	44
	CSPA	21.3	4	100	100	**44**
Galaxy/Star	MCLA	10.5	10	50	1500	36
	MCLA	11.7	10	100	200	**5**
	AL	11	10	100	500	12
	AL	10.9	3	100	4192	100

Last column denote the percentage of sample size regarding the entire dataset. (Bold represents most optimal)

in Fig. 6. This perfect result can be achieved by $k = 10,\ B = 150$, and $S \cdot 200$ (50% data size) or $k = 10,\ B = 100$, and $S \cdot 300$ (75% data size) according to the bootstrap results in Table 10. Thus, there is a trade off between the number of partitions B and the sample size S. This comparison shows that the subsampling method can be much faster than the bootstrap ($N = 400$) in relation to the computational complexity.

The results of subsampling for "Star/Galaxy" dataset as given in Fig. 7, shows that at the fixed resolution value of $k = 3$ and the fixed number of partitions $B = 100$, with minimun sample size $S = 1000$ (24% of the entire data size) 89% accuracy is achivable. Note

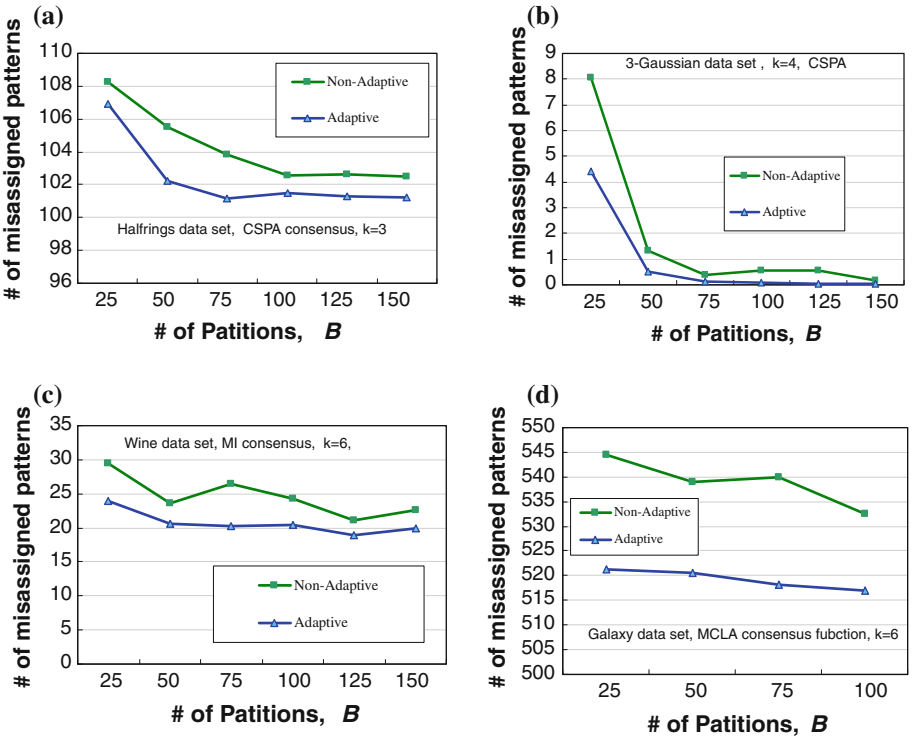

Fig. 8 Clustering accuracy for ensembles with adaptive and non-adaptive sampling mechanisms as a function of ensemble size for some datasets and selected consensus functions

that this result is also achivable by the bootstrap method, but with a greater minimum sample size S, i.e. with entire dataset (100% of the entire data size) provided that $k = 3$ and $B = 100$. It shows that for such a large dataset, a small fraction of data can be representative of the entire dataset, and computationally this would be very interesting in distributed data mining.

Note that in both the bootstrap and the subsampling algorithms all of the samples are drawn independently, and thus the resampling process can be performed in parallel. Therefore, by the B parallel processes, the computational process could be B times faster.

Table 8 shows the error rate of classical clustering algorithms, which are used in this research. The error rates for the k-means algorithm were averaged over 100 runs, with random initializations for the cluster centers, where the value of k was fixed to the true number of clusters. One can compare it to the error rate of ensemble algorithms in Table 9.

The optimal size S and granularity of the component partitions derived by subsampling are reported in Table 10. We see that the accuracy of the resampling method is very similar to that of the bootstrap algorithm, as reported in Table 8. This level of accuracy was reached with remarkably smaller sample sizes and much lower computational complexity! The trade-off between the accuracy of the overall clustering combination and computational effort for generating component partitions is shown in Table 10, where we compare accuracy of consensus partitions. The most promising result is that only a small fraction of data (i.e., 12 or 5% for the "Star/Galaxy" dataset) is required to obtain the optimal solution of clustering, both in terms of accuracy and computational time.

5.3 Empirical study on adaptive approach

The experiments were conducted on artificial and real-world datasets ("Galaxy", "half-rings", "wine", "3-gaussian", "Iris", "LON"), with known cluster labels, to validate the accuracy of consensus partition. A comparison of the proposed adaptive and previous non-adaptive (Minaei-Bidgoli et al. 2004a) ensemble is the primary goal of the experiments. We evaluated the performance of the clustering ensemble algorithms by matching the detected and the known partitions of the datasets. The best possible matching of clusters provides a measure of performance expressed as the misassignment rate. To determine the clustering error, one needs to solve the correspondence problem between the labels of known and derived clusters. Again, the Hungarian algorithm was used for this purpose. The k-means algorithm was used to generate the partitions of samples of size N drawn with replacement, similar to bootstrap, albeit with dynamic sampling probability. Each experiment was repeated 20 times and averaged numbers of misassignment patterns are shown in Fig. 8.

Consensus clustering was obtained by four different consensus functions: hypergraph-based MCLA and CSPA algorithms (Strehl and Ghosh 2003), quadratic mutual information (Topchy et al. 2003) and EM algorithm based on mixture model (Topchy et al. 2004a). Herein, we report only the key findings. The main observation is that adaptive ensembles slightly outperform the regular sampling schemes on most benchmarks. Accuracy improvement depends on the number of clusters in the ensemble partitions (k). Generally, the adaptive ensembles were superior for values of k larger than the target number of clusters, M, by 1or 2. With either too small or too large a value of k, the performance of adaptive ensembles was less robust and occasionally worse than corresponding non-adaptive algorithms. A simple inspection of probability values always confirmed the expectation that points with large clustering, uncertainty are drawn more frequently.

The most significant progress was detected when combination consisted of 25–75 partitions. Large numbers of partitions ($B > 75$) almost never led to further improvement in clustering accuracy. Moreover, for $B > 125$ most of times we observed increased error rates (except for the hypergraph-based consensus functions), due to the increase in complexity of the consensus model and in the number of model parameters which they must be estimated with trial and error method. Of course this matter is valid just for small datasets. It means the larger number of data samples, the better adaptive ensembles than non-adaptive ones. It can be due to the fact that with increasing k, in the final steps, the algorithm just selects boundary data samples and lacks its generality. For "LON" and "Iris" datasets, the adaptive method didn't work so better than non-adaptive one that it isn't reported here. The results of adaptive method are always equal or better than the non-adaptive method. Even in the case of "LON" and "Iris" datasets we face improvements, but little improvements. In all experiment, with same B, S, k values and same consensus function, the adaptive method outperforms the non-adaptive method, but in many cases this improvement is not considerable.

6 Concluding remarks

A new approach to combine partitions is proposed by resampling of original data. This study showed that meaningful consensus partitions for the entire dataset of objects emerge from clusterings of bootstrap and subsamples of small size. Empirical studies were conducted on various simulated and real datasets for different consensus functions, number of partitions in the combination and number of clusters in each component, for both bootstrap (*with* replacement) and subsampling (*without* replacement). The results demonstrate that there is a

trade-off between the number of clusters per component and the number of partitions, and the sample size of each partition needed in order to perform the combination process converges to an optimal error rate.

The bootstrap technique was recently applied in Dudoit and Fridlyand (2003), Fischer and Buhmann (2003), Monti et al. (2003) to create diversity in clusterings ensemble. However, our work extends the previous studies by using a more flexible subsampling algorithm for ensemble generation. We also provided a detailed comparative study of several consensus techniques. The challenging points of using resampling techniques for maintaining diversity of partitions were discussed in this paper. We showed that there exists a critical fraction of data such that the structure of entire dataset can be perfectly detected. Subsamples of small sizes can reduce costs and measurement complexity for many explorative data mining tasks with distributed sources of data.

We have extended clustering ensemble framework by adaptive data sampling mechanism for generation of partitions. We dynamically update sampling probability to focus on more uncertain and problematic points by on-the-fly computation of clustering consistency. Empirical results demonstrate improved clustering accuracy and faster convergence as a function of the number of partitions in the ensemble.

Further study of alternative resampling methods, such as the balanced (stratified) and re-centered bootstrap methods are critical for more generalized and effective results.

References

Aeberhard S, Coomans D, de Vel O (1992). Comparison of classifiers in high dimensional settings. Technical Report no. 92-02, Department of Computer Science and Department of Mathematics and Statistics, James Cook University of North Queensland

Ayad HG, Kamel MS (2008). Cumulative voting consensus method for partitions with a variable number of clusters. IEEE Trans Pattern Anal Mach Intell 30(1)

Barthelemy J, Leclerc B (1995) The median procedure for partition. In: Cox IJ et al (eds) Partitioning data sets. AMS DIMACS series in discrete mathematics, vol 19, pp 3–34

Ben-Hur A, Elisseeff A, Guyon I (2002). A stability based method for discovering structure in clustered data. In: Pacific symposium on biocomputing, vol 7, pp 6–17

Breiman L (1996) Bagging predictors. J Mach Learn 24(2):123–140

Breiman L (1998) Arcing classifiers. Ann Stat 26(3):801–849

Dhillon IS, Modha DS (2000) A data-clustering algorithm on distributed memory multiprocessors. In: Proceedings of large-scale parallel KDD systems workshop, ACM SIGKDD, in large-scale parallel data mining, lecture notes in artificial intelligence, vol 1759, pp 245–260

Dixon JK (1979) Pattern recognition with partly missing data. IEEE Trans Syst Man Cybern SMC 9:617–621

Duda RO, Hart PE, Stork DG (2001) Pattern classification. 2 (edn). John Wiley & Sons, New York

Dudoit S, Fridlyand J (2003) Bagging to improve the accuracy of a clustering procedure. Bioinformatics 19(9):1090–1099

Efron B (1979) Bootstrap methods: another Look at the Jackknife. Ann Stat 7:1–26

Fern X, Brodley CE (2003) Random projection for high dimensional data clustering: a cluster ensemble approach. In: Proceedings of 20th international conference on Machine Learning, ICML 2003

Fischer B, Buhmann JM (2002) Data resampling for path based clustering. In: Van Gool L (ed) Pattern recognition—-Symposium of the DAGM. Springer, LNCS, vol 2449, pp 206–214

Fischer B, Buhmann JM (2003) Path-based clustering for grouping of smooth curves and texture segmentation. IEEE Trans PAMI 25(4):513–518

Fred ALN, Jain AK (2002) Data clustering using evidence accumulation. In: Proceedings of the 16th international conference on pattern recognition, ICPR 2002, Quebec City, pp 276–280

Fred ALN, Jain AK (2005) Combining multiple clustering using evidence accumulation. IEEE Trans Pattern Anal Mach Intell 27(6)

Frossyniotis D, Likas A, Stafylopatis A (2004) A clustering method based on boosting. Pattern Recognit Lett 25(6):641–654

Hastie T, Tibshirani R, Friedman JH (2001) The elements of statistical learning. Springer, Berlin

Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Englewood Cliffs
Jain AK, Moreau JV (1987) The bootstrap approach to clustering. In: Devijver PA, Kittler J (eds) Pattern recognition theory and applications. Springer, Berlin pp 63–71
Jiamthapthaksin R, Eick CF, Lee S (2010) GAC-GEO: a generic agglomerative clustering framework for geo-referenced datasets. Knowl Inf Syst
Levine E, Domany E (2001) Resampling method for unsupervised estimation of cluster validity. Neural Comput 13:2573–2593
Minaei-Bidgoli B, Punch WF (2003) Using genetic algorithms for data mining optimization in an educational web-based system. GECCO :2252-2263
Minaei-Bidgoli B, Topchy A, Punch WF (2004a) Ensembles of partitions via data resampling. In: Proceedings of international conference on information technology, ITCC 04, Las Vegas
Minaei-Bidgoli B, Topchy A, Punch WF (2004b) A comparison of resampling methods for clustering ensembles. In: Proceedings of conference on machine learning methods technology and application, MLMTA 04, Las Vegas
Mohammadi M, Alizadeh H, Minaei-Bidgoli B (2008) Neural network ensembles using clustering ensemble and genetic algorithm. In: Proceedings of international conference on convergence and hybrid information technology, ICCIT08, 11–13 Nov 2008, published by IEEE CS, Busan, Korea
Monti S, Tamayo P, Mesirov J, Golub T (2003) Consensus clustering: a resamlping-based method for class discovery and visualization of gene expression microarray data. J Mach Learn 52(1)
Odewahn SC, Stockwell EB, Pennington RL, Humphreys RM, Zumach WA (1992) Automated star/galaxy discrimination with neural networks. Astron J 103:308–331
Park BH, Kargupta H (2003) Distributed data mining. In: Ye N (ed) The handbook of data mining. Lawrence Erlbaum Associates, Hillsdale
Parvin H, Alizadeh H, Minaei-Bidgoli B, Analoui M (2008a) CCHR: combination of classifiers using heuristic retraining. In: Proceedings of international conference on networked computing and advanced information management (NCM 2008), Korea, Sep 2008, published by IEEE CS
Parvin H, Alizadeh H, Minaei-Bidgoli B, Analoui M (2008b) An scalable method for improving the performance of classifiers in multiclass applications by pairwise classifiers and GA. In: Proceedings of international conference on networked computing and advanced information management (NCM 2008), Korea, Sep 2008, published by IEEE CS
Parvin H, Alizadeh H, Minaei-Bidgoli B (2008c) A new approach to improve the vote-based classifier selection. In: Proceedings of international conference on networked computing and advanced information management (NCM 2008), Korea, Sep 2008, published by IEEE CS
Parvin H, Alizadeh H, Moshki M, Minaei-Bidgoli B, Mozayani N (2008d) Divide & conquer classification and optimization by genetic algorithm. In: Proceedings of international conference on convergence and hybrid information technology, ICCIT08, Nov 11–13 2008, published by IEEE CS, Busan, Korea
Pfitzner D, Leibbrandt R, Powers D (2009) Characterization and evaluation of similarity measures for pairs of clusterings. Knowl Inf Syst
Roth V, Lange T, Braun M, Buhmann JM (2002) A resampling approach to cluster validation. In: Proceedings in computational statistics: 15th symposium COMPSTAT 2002. Physica-Verlag, Heidelberg, pp 123–128
Saha S, Bandyopadhyay S (2009) A new multiobjective clustering technique based on the concepts of stability and symmetry. Knowl Inf Syst
Strehl A, Ghosh J (2003) Cluster ensembles-a knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–617
Tan PN, Steinbach M, Kumar V (2005) Introduction to data mining, 1st edn. Addison-Wesly, Reading
Topchy A, Jain AK, Punch WF (2003) Combining multiple weak clusterings. In: Proceedings of 3rd IEEE international conference on data mining, pp 331–338
Topchy A, Jain AK, Punch WF (2004a) A mixture model for clustering ensembles. In: Proceedings of SIAM international conference on data mining, SDM 04, pp 379–390
Topchy A, Minaei-Bidgoli B, Jain AK, Punch WF (2004b) Adaptive clustering ensembles. In Proceedings of international conference on pattern recognition, ICPR 2004, Cambridge, UK
Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. ACM SIGMOD Record 25(2):103–114
Zhang B, Hsu M, Forman G (2000) Accurate recasting of parameter estimation algorithms using sufficient statistics for efficient parallel speed-up demonstrated for center-based data clustering algorithms. In: Proceedings of 4th European conference on principles and practice of knowledge discovery in databases, in principles of data mining and knowledge discovery