

Cascaded cluster ensembles

Li Zhang · Xing-Hong Ling · Ji-Wen Yang ·
Xiao-Qian Wang · Fan-Zhang Li

Received: 18 July 2011 / Accepted: 15 December 2011 / Published online: 31 December 2011
© Springer-Verlag 2011

Abstract Combining multiple clusterers is emerged as a powerful method for improving both the robustness and the stability of unsupervised classification solutions. In this paper, a framework for cascaded cluster ensembles is proposed, in which there are two layers of clustering. The first layer is considering about the diversity of clustering, and generating different partitions. In doing so, the samples in input space are mapped into labeled samples in a label attribute space whose dimensionality equals the ensemble size. In the second layer clustering, we choose a clustering algorithm as the consensus function. In other words, a combined partition is given by using the clustering algorithm on these labeled samples instead of input samples. In the second layer, we use the reduced k -means, or the reduced spectral, or the reduced hierarchical linkage algorithms as the clustering algorithm. For comparison, nine consensus functions, four of which belong to cascaded cluster ensembles are used in our experiments. Promising results are obtained for toy data as well as UCI data sets.

Keywords Unsupervised learning · Clustering ensemble · k -means clustering

Electronic supplementary material The online version of this article (doi:10.1007/s13042-011-0065-5) contains supplementary material, which is available to authorized users.

L. Zhang (✉) · X.-H. Ling · J.-W. Yang · X.-Q. Wang ·
F.-Z. Li
Research Center of Machine Learning and Data Analysis,
School of Computer Science and Technology,
Soochow University, Suzhou 215006, Jiangsu, China
e-mail: zhangliml@suda.edu.cn

1 Introduction

Clustering analysis is an important tool for image processing, remote sensing, data mining, biology, and pattern recognition [1, 2]. The idea of ensemble can be found in many fields [3–5]. Cluster ensembles have been introduced as a more accurate alternative to individual clustering algorithms [6].

Two major themes in ensembles are combination rules of the ensemble votes and the diversity of clusterers. Firstly, we consider how to build diverse yet accurate individual clusterers, or to select clustering algorithms for the ensemble. Various methods have been proposed for getting the diversity of clusterers, such as random initialization of the clustering algorithm, resampling the data [7, 8], resampling the features of the data [8], etc. Finally, we need to know how to combine the outputs of multiple clusterers, or to construct a consensus function. There are several approaches for constructing consensus functions, such as the relabeling method [7–10], the feature-based approach [11, 12], the hyper-graph approach [8] and others.

In [12], a consensus function based on quadratic mutual information (QMI) was presented and reduced to the k -means clustering in the space of specially transformed cluster labels. Here we focus on these label attributes instead of some transformed partition attributes.

In this paper, a framework for cascaded cluster ensembles is proposed, in which there are two layers of clustering. The first layer deals with the diversity of clustering, and generates different partitions. In doing so, the samples in input space are mapped into labeled samples in a label attribute space whose dimensionality equals the ensemble size. In the second layer clustering, we choose a clustering algorithm as the consensus function. In other words,

a combined partition is given by using the clustering algorithm on these labeled samples instead of input samples. We also give some reduced clustering algorithms, e.g., k -means, spectral, and hierarchical single-linkage, for the second layer clustering. We have two main contributions. One is that a framework of cascaded cluster ensembles is presented. The other is that some traditional clustering algorithms are reduced and adapted for re-clustering.

The rest of this paper is organized as follows. In Sect. 2, we review the related work on consensus functions, such as relabeling methods and hypergraph methods. Section 3 presents the framework for cascaded cluster ensembles. We also discuss reduced clustering algorithms, such as k -means, spectral, hierarchical single-linkage algorithms, for the second layer clustering. We apply cascaded cluster ensembles to toy and UCI data sets in Sect. 4 and conclude our paper in Sect. 5.

2 Related work

Notation. Let the set of N unlabeled samples be $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, where $\mathcal{X} \in \mathbb{R}^d$, and d is the dimensionality of sample space. Assume that there are L partitions $\mathcal{P} = \{P_1, \dots, P_L\}$ for \mathcal{X} , where L is the ensemble size or the number of clusterers, and $P_i \in \mathbb{R}^N$ is the label vector of \mathcal{X} obtained by the i -th individual clusterer. Labels take value from 1 to k , where k is the number of clusters. L clusterers give a set of labels for each sample $\mathbf{x}_i, i = 1, \dots, N$

$$\mathbf{x}_i \rightarrow \mathbf{y}_i = \{P_1(\mathbf{x}_i), \dots, P_L(\mathbf{x}_i)\} \quad (1)$$

where $P_j(\mathbf{x}_i)$ denotes a label assigned to \mathbf{x}_i by the j -th clusterer, and \mathbf{y}_i called labeled samples corresponding to \mathbf{x}_i is all L labels assigned to \mathbf{x}_i by all individual clusterers. Denote $\mathcal{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ be a set of labeled samples in a label attribute space \mathbb{Y} .

We want to give \mathbf{x}_i an optimal label y_i^* or $P^*(\mathbf{x}_i)$. The goal is to find a combined partition P^* by using a consensus function. There are many consensus functions available, which are briefly described as follows.

2.1 Relabeling methods

These methods are also called direct approaches or voting approaches, such as two bagged clustering procedures in [10], path based clustering in [9], bagging-based selective cluster ensemble (BBSCE) in [7].

Assume that the set of L partitions \mathcal{P} is generated. Since no assumption is made about the correspondence between the labels produced by individual clusterers, the label $P_j(\mathbf{x}_i)$ maybe differs from $P_l(\mathbf{x}_i)$. Relabeling method is

firstly to relabel partitions according to a fixed reference partition which can be selected from these partitions. The complexity of relabeling two partitions is $O(k^3)$ if the Hungarian method is employed. The total complexity of relabeling process should be $O((L-1)k^3)$. Finally relabeling method is to find the best partition P^* by some combination rules such as the voting rule.

In [7], the selective weighted voting rule is presented for getting the best partition. The mutual information between the reference partition and other partitions is taken as the weights of these partitions. If the weights are larger than the average weight, the corresponding partitions are used to determine a single consensus clustering.

2.2 Hypergraph methods

These methods are to construct a hypergraph representing partitions from the clusterers and cutting the redundant edges. A hypergraph contains of vertices and hyperedges. Vertices denote samples in \mathcal{X} and each hyperedge describes a set of samples belonging to the same cluster. Here, the cluster ensemble problem is casted into an optimization problem of finding the k -way minimum-cut of a hypergraph.

Strehl and Ghosh [8] proposed three different consensus functions for ensembles. The cluster-based similarity partitioning algorithm (CSPA) generates a graph from a similarity matrix and reclusters it using the similarity-based clustering algorithms. The hypergraph partitioning algorithm (HGPA) represents each cluster by a hyperedge in a graph and uses minimal cut algorithms to find good hypergraph partitions. The meta-clustering algorithm (MCLA) is to group and collapse related hyperedges and assign each sample to the collapsed hyperedge in which it shares most strongly. By doing so, MCLA can determine soft cluster-membership values for each sample. By using CSPA and MCLA as consensus functions, a new selective clustering ensemble algorithm was proposed in [13].

The computational complexity of CSPA, HGPA and MCLA are $O(kN^2L)$, $O(kNL)$ and $O(k^2NL^2)$, respectively.

2.3 Feature-based methods

A unified representation of multiple clusterings was given in [11] and [12]. The outputs from the multiple clusterers are treated as L new features. The next problem is how to get a clustering by using these new features. A probabilistic model consensus using a finite mixture of multinomial distribution was proposed in [11] and [12]. The consensus problem is casted into the maximum-likelihood problem which is solved by the EM algorithm. A consensus function

based on quadratic mutual information (QMI) was also presented and reduced to the k -means clustering in the space of specially transformed cluster labels.

2.4 Pairwise methods

In pairwise methods, consensus functions operate on the co-association matrix which can be taken as a similarity matrix for the data points. A voting- k -means algorithm was proposed in [14] where the combination of partitions is performed by transforming data partitions into a co-association matrix. The data points should be linked into a clusters if their corresponding co-association values exceed a given threshold. Fred and Jain [15] adopted a hierarchical clustering to the co-association matrix instead of a fixed threshold.

3 Cascaded cluster ensembles

In this section, we discuss cascaded cluster ensembles. Here are two main contributions. One is to give a framework of cascaded cluster ensembles. The other is to propose reduced versions for some traditional clustering algorithms and apply them to re-clustering.

Figure 1 gives the framework of cascaded cluster ensembles. There are L individual clusterers in the first layer clustering. The goal of the first layer is to generate different partitions, or P_1, P_2, \dots, P_L . In the second layer clustering, different partitions are combined to an optimal partition by using some clustering algorithm, such as k -means.

3.1 First layer clustering

In this layer, we consider about the diversity of clustering, or generating different partitions. There are several methods available for getting diverse partitions:

1. Using different clustering algorithms such as k -means, fuzzy k -means, mixture of Gaussian, graph partitioning based, statistical mechanics based, etc., [8].
2. Using randomness or different parameters of some algorithms, e.g., initialization and various values of k for k -means clustering algorithm [6, 15, 16].
3. Using different data subsets, such as bootstrap samples [7, 8], and feature subsets [8, 17, 18].

The only constraint in our frame is that k is fixed to a constant. $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_L$, the inputs of the first layer can be different data subsets if the same type of individual clusterers are used, and be the same data set if the different types of individual clusterers are exploited. The outputs of the first layer are different partitions P_1, P_2, \dots, P_L .

3.2 Second layer clustering

From Fig. 1, we can see that the inputs of the second layer clustering are P_1, P_2, \dots, P_L which are called partition attributes. For a sample \mathbf{x}_i , its original attributes is itself \mathbf{x}_i , and its label attributes \mathbf{y}_i can be expressed as (1). Note that the label attributes are all represented by integers, say $1, \dots, k$. An optimal partition P^* can be obtained by using any clustering algorithm on these labeled samples. Here the consensus function is a clusterer.

In the label attribute space, we can use different clustering algorithm to handle the second layer clustering problem, including k -means, fuzzy k -means, spectral, hierarchical clustering, and so on.

Firstly, we illustrate why we consider a clustering algorithm as the consensus function by using a simple example. Suppose that there are seven points belonging to three classes. Three individual clusterers are used to generate three different partitions shown in the left hand of Table 1. Inspection of these label attributes for seven samples reveals that the label attributes of $\mathbf{x}_1, \mathbf{x}_2$ and \mathbf{x}_3 are identical, \mathbf{x}_4 and \mathbf{x}_5 are also identical, and \mathbf{x}_6 and \mathbf{x}_7 are some different. Given that centers of three classes be $(2, 3, 1)^T, (1, 2, 3)^T$ and $(3, 1, 2)^T$, respectively, the right hand of Table 1 shows the

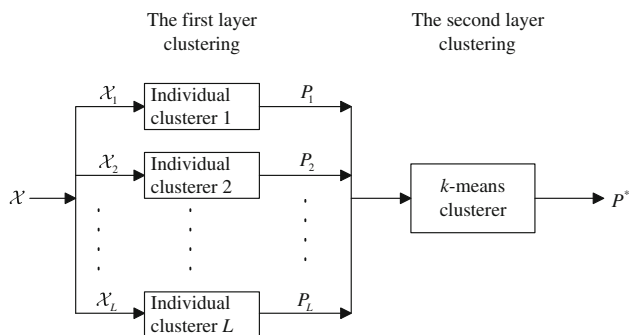


Fig. 1 Framework of cascaded ensembles

Table 1 Illustrative example problem with $k = 3, l = 7, L = 3$

Partitions	\mathbf{x}_1	\mathbf{x}_2	\mathbf{x}_3	\mathbf{x}_4	\mathbf{x}_5	\mathbf{x}_6	\mathbf{x}_7
P_1	2	2	2	1	1	3	3
P_2	3	3	3	2	2	1	3
P_3	1	1	1	3	3	2	2
Label attributes	\mathbf{y}_1	\mathbf{y}_2	\mathbf{y}_3	\mathbf{y}_4	\mathbf{y}_5	\mathbf{y}_6	\mathbf{y}_7
Centers	\mathbf{x}_1	\mathbf{x}_2	\mathbf{x}_3	\mathbf{x}_4	\mathbf{x}_5	\mathbf{x}_6	\mathbf{x}_7
$(2, 3, 1)^T$	0	0	0	3	3	3	2
$(1, 2, 3)^T$	3	3	3	0	0	3	3
$(3, 1, 2)^T$	3	3	3	3	3	0	1

Hamming distance between samples to these centers. In information theory, the Hamming distance between two strings (or vector) of equal length is the number of positions for which the corresponding symbols (or figures) are different. By simply observing these Hamming distances, we can sure that $\mathbf{x}_1, \mathbf{x}_2$ and \mathbf{x}_3 belong to the same cluster, \mathbf{x}_4 and \mathbf{x}_5 belong to another cluster, and \mathbf{x}_6 and \mathbf{x}_7 are in the same cluster.

From the example, we can see that there still exist similarities in points belonging to the same cluster even if P_1, P_2, \dots, P_L are different partitions generated by L individual clusterers. Therefore, we can directly adopt this information to combine L partitions using a clusterer. Although [12] uses the k -means clustering algorithm, the data clustered locate in a transformed spaces instead of the label attribute space.

Next we discuss how to use these clustering algorithms in the second layer clustering. Note that most of label attributes are identical, such as $\mathbf{y}_1, \mathbf{y}_2$ and \mathbf{y}_3 in our example. Hence, we do not need process all labeled samples. So the first step that we need do is to select all distinct labeled samples from the set \mathcal{Y} . Let $\bar{\mathcal{Y}} = \{\bar{\mathbf{y}}_j\}_{j=1}^n$ be the set of distinct labeled samples, where $n < N$. The Hamming distance between $\bar{\mathbf{y}}_i$ and $\bar{\mathbf{y}}_j$ is at least 1. In the following, some detail clustering algorithms are discussed.

3.2.1 Reduced k -means algorithm

In the label space, k -means clustering algorithm with the Hamming distance metric is one simple choice. It is well known, the k -means clustering algorithm is sensitive to initialization, which could be used in the first layer clustering for increasing diversity but should be avoided in the second layer clustering. Thus it is necessary to consider the selection of initial centers for k -means algorithm. Now $\bar{\mathcal{Y}}$ is the candidate set for selecting the initial centers $\{\mathbf{c}_i\}_{i=1}^k$. After we get these centers, we can directly compute the Hamming distance between a labeled sample \mathbf{y}_j to those centers. Once we find that the minimal Hamming distance is that between \mathbf{c}_m and \mathbf{y}_j , the corresponding label m should be assigned to the sample \mathbf{x}_j . The reduced k -means clustering used in the second layer clustering is shown in Algorithm 1.

Algorithm 1 Reduced k -means clustering

Input: The number of clusters k , and the labeled sample set $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N, \mathbf{y}_i \in \mathbb{R}^L$.
Output: The optimal partition P^* for the sample set \mathcal{X} .

1. Select the distinct sample set $\bar{\mathcal{Y}}$ from \mathcal{Y} .
2. Select k centers (means) from the set $\bar{\mathcal{Y}}$.
3. **for** $i = 1$ to k **do**
 if $i = 1$ **then**
 Select a center randomly from $\bar{\mathcal{Y}}$, and let it be \mathbf{c}_1 .
 else
 Select the i -th center which has the farthest Hamming distance from all other centers $\mathbf{c}_1, \dots, \mathbf{c}_{i-1}$.
 end if
4. **end for**
5. Compute the Hamming distances between $\mathbf{c}_i, i = 1, \dots, k$ and $\mathbf{y}_j, j = 1, \dots, N$, or $d(\mathbf{c}_i, \mathbf{y}_j)$.
6. Assign a label m to \mathbf{x}_j or \mathbf{y}_j , if $m = \min_{i=1, \dots, k} h(\mathbf{c}_i, \mathbf{y}_j)$. Then $P_j^* = m$.

Table 2 Illustrative example problem with $k = 3, \ell = 7, L = 3$

	$(2, 3, 1)^T$	$(1, 2, 3)^T$	$(3, 1, 2)^T$	$(3, 3, 2)^T$
$(2, 3, 1)^T$	0	3	3	2
$(1, 2, 3)^T$	3	0	3	3
$(3, 1, 2)^T$	3	3	0	1
$(3, 3, 2)^T$	2	3	1	0

3.2.2 Reduced spectral clustering

Remember that all attributes of labeled samples are only expressed by positive integers. We compute the Hamming distance matrix \mathbf{H} between these distinct labeled samples. Namely,

$$\mathbf{H}_{ij} = h(\bar{\mathbf{y}}_i, \bar{\mathbf{y}}_j), \quad i, j = 1, \dots, n \tag{2}$$

where $h(\cdot, \cdot)$ denotes the Hamming distance function, and \mathbf{H} is a symmetrical matrix and can be taken as a dissimilarity matrix. Return to the above example, Table 2 shows the Hamming distances between four distinct labeled samples. We can see that the maximum Hamming distance is three which equals the ensemble size. The reduced spectral clustering used in the second layer clustering is shown in Algorithm 2.

Algorithm 2 Reduced Spectral clustering

Input: The number of clusters k , and the labeled sample set $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^N, \mathbf{y}_i \in \mathbb{R}^L$.
Output: The optimal partition P^* for the sample set \mathcal{X} .

1. Select the distinct sample set $\bar{\mathcal{Y}}$ from \mathcal{Y} .
2. Compute the Hamming distance matrix \mathbf{H} as in (2).
3. Compute the Laplacian matrix \mathbf{L} :
 $\mathbf{W}_{ij} = \exp(-\mathbf{H}_{ij})$,
 $\mathbf{D} = \sum_{j=1}^n \mathbf{W}_{ij}$,
 $\mathbf{L} = \mathbf{D} - \mathbf{W}$
4. Compute the eigenvectors $\{\mathbf{v}_p\}$ and eigenvalues $\{\lambda_p\}$ of \mathbf{L} , and take the eigenvectors $\{\mathbf{v}_{p1}, \mathbf{v}_{p2}, \dots, \mathbf{v}_{pk}\}$ corresponding to the first maximum k non-zeros eigenvalues as the projection vectors.
5. Perform k -means clustering with the Euclidean distance metric on the projection vectors, and obtain k clusters. Then the corresponding distinct labeled samples have their own labels, or $\{(\bar{\mathbf{y}}_i, m_i), i = 1, \dots, n, m_i \in \{1, \dots, k\}\}$.
6. Use the nearest neighbor algorithm to assign labels to all samples.
 Assign a label m_i to \mathbf{y}_j , if the nearest neighbor of \mathbf{y}_j is $\bar{\mathbf{y}}_i$. Namely $P_j^* = m_i$.

3.2.3 Reduced hierarchical clustering

Hierarchical clustering creates a hierarchy of clusters for which may be represented in a tree structure called a dendrogram. The root of the tree consists of a single cluster containing all samples, and the leaves correspond to individual samples. According to the definition of the distance between one cluster and another cluster, there have three different hierarchical clustering algorithms, such as single-linkage, complete-linkage and average-linkage clustering. Here we take the single-linkage clustering as an example, and show its reduced version in Algorithm 3.

Algorithm 3 Reduced hierarchical clustering

Input: The number of clusters k , and the labeled sample set $\mathcal{Y} = \{y_i\}_{i=1}^N, y_i \in \mathbb{R}^L$.
Output: The optimal partition P^* for the sample set \mathcal{X} .

1. Select the distinct sample set $\bar{\mathcal{Y}}$ from \mathcal{Y} .
2. Compute the Hamming distance matrix \mathbf{H} as in (2).
3. Assign each distinct labeled sample $\bar{y}_i, i = 1, \dots, n$ to a cluster. Thus there are n clusters firstly.
4. Find the closest pair of clusters and merge them into a single cluster. Now the number of cluster is $n - 1$.
5. Compute distances between the new cluster and each of the old clusters. (The distance between one cluster and another cluster is defined to equal the shortest Hamming distance from any member of one cluster to any member of the other cluster.)
6. Repeat steps 4) and 5) until the number of clusters is k .
7. Obtain k clusters, and have labels for the distinct labeled samples. $(\bar{y}_i, m_i), i = 1, \dots, n, m_i \in \{1, \dots, k\}$
8. Use the nearest neighbor algorithm to assign labels to all samples. Assign a label m_i to y_j , if the nearest neighbor of y_j is \bar{y}_i . Namely $P_j^* = m_i$.

4 Simulation

In order to validate the performance of cascaded cluster ensembles, we perform experiments on two toy [12] and seven UCI data sets [19] described in Table 3, where true natural clusters are known, and compare them with five other methods for cluster ensembles.

The performance criterion is the same as [12]. The performance of all methods is evaluated by matching the optimal partition P^* with the known partitions of data sets and expressed as the clustering error. Table 3 also gives the average clustering errors of these data sets, which are obtained by running 30 k -means clustering algorithms independently.

All numerical experiments were performed on the personal computer with a 1.8 GHz Pentium III and 1 G bytes of memory. This computer runs on Windows XP, with Matlab 7.1 installed.

4.1 Selection of algorithms and their parameters

The k -means clustering is used to generate the different partitions in the first layer clustering. The diversity of the partitions is obtained by using different data subsets, including bootstrap samples and feature subsets. The

Table 3 Description of data sets used in experiments

Data set	No. of features (d)	No. of class (k)	No. of samples (N)	Av. k -means error (%)
Half-rings	2	2	400	24.54
2-spirals	2	2	200	41.66
Glass	9	6	213	51.57
Liver	6	2	345	45.79
Musk	166	2	476	46.4
Soy	208	17	289	55.46
Vote	16	2	435	13.56
Wdbc	30	2	569	9.85
Wpdb	33	2	198	40.93

following parameters for the first layer are especially important.

- k : the number of clusters. We set it as the same as the true class number.
- L : the ensemble size or the number of individual clusterers. Its value varies in the set $[5, 10, 15, \dots, 50]$.
- r : the bootstrap sampling ratio (the ratio of the bootstrap sample number to the total sample number N), or the feature sampling ratio (the ratio of the feature number of subset to the whole feature number d). r takes only three values, or 25, 50 and 75%.

After we get different partitions, five other consensus functions besides clusterer consensus functions in cascaded cluster ensembles are adopted, including BBSCE, CSPA, MCLA, the EM and QMI algorithms. BBSCE in [7] is a relabeling method. The code of CSPA and MCLA in [8] is available at <http://www.strehl.co>. The code of the EM and QMI algorithms was provided by Dr. A. Topchy. In cascaded cluster ensembles, we choose the k -means, spectral, and hierarchical single-linkage (H-single) and complete-linkage (H-complete) clusterers as consensus functions, respectively. Thus there are nine consensus functions. In the second layer clustering, k takes the same value as that in the first layer clustering.

4.2 Experiments with bootstrap sampling method

In this part, the diversity of the partitions in the first layer clustering is obtained by using the bootstrap sampling method which are obtained by randomly sampling with replacement from the original data set. Here r denotes the bootstrap sampling ratio. The number of samples with whole features for individual clusterers in the first layer clustering takes value from $\{\lceil N \times 0.25 \rceil, \lceil N \times 0.5 \rceil, \lceil N \times 0.75 \rceil\}$, where $\lceil \cdot \rceil$ rounds \cdot to the nearest integers towards infinity. For each data set, we perform 30 independent runs. Only the average errors on 30 runs are reported.

4.2.1 Toy data sets

Two artificial data sets, Half-rings and two-spirals shown in Fig. 2, are traditionally difficult for any centroid-based clustering algorithm [12]. Table 4 shows the average clustering error rate (%) for the Half-rings data set with 25, 50 and 75% bootstrap sampling ratio, respectively. The average clustering error rate in bold type in each column of Table 4 is the best one of the corresponding algorithm with different bootstrap sampling ratio.

For different bootstrap sampling ratio, most methods obtain the similar results, except for BBSCE. In a nutshell about the Half-rings data set H-single gets the best performance 19.83%, followed by spectral clustering 20.65%.

Fig. 2 Two toy data sets

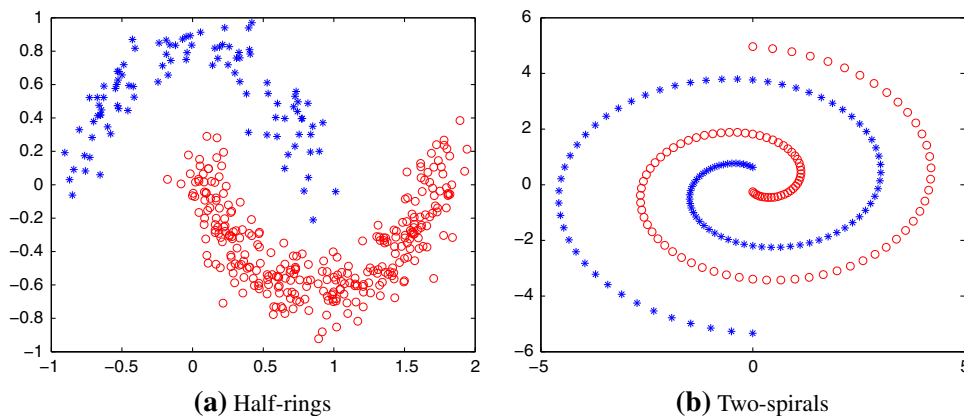


Table 4 Average clustering error rate (%) for the “Half-rings” data set with bootstrap samples

L	Type of consensus function								
	BBSCE	CSPA	MCLA	EM	QMI	Clusterer			
						k-means	Spectral	H-single	H-complete
Bootstrap sampling ratio 25%									
5	27.48	28.75	23.85	22.06	23.75	22.75	24.32	21.63	21.68
10	29.53	28.17	21.53	21.29	21.68	22.33	25.18	21.26	20.88
15	28.22	28.48	23.12	22.21	23.35	22.32	27.46	19.83	21.67
20	29.18	28.08	22.81	21.67	21.51	22.90	25.75	21.43	22.14
25	29.29	28.20	22.30	21.53	21.31	22.33	23.49	20.74	21.07
30	27.76	28.08	22.48	21.51	21.33	23.05	22.23	21.17	21.86
35	28.78	28.13	21.83	21.16	20.80	22.16	23.11	20.63	22.03
40	30.63	28.03	22.27	21.51	21.37	21.34	21.47	21.23	21.8
45	29.88	28.10	22.45	21.81	21.83	23.44	22.46	21.33	22.29
50	28.59	27.98	22.27	21.52	21.71	22.17	22.50	20.38	20.86
Bootstrap sampling ratio 50%									
5	27.18	28.16	23.05	22.44	22.33	22.14	26.39	20.96	21.68
10	27.73	28.26	22.92	23.35	24.26	22.82	26.48	21.43	23.02
15	29.03	28.34	22.62	22.04	22.64	22.28	23.57	20.67	22.03
20	30.54	28.14	22.30	21.24	22.17	22.95	23.41	21.89	21.87
25	30.92	28.23	21.85	20.86	20.84	22.31	23.61	20.61	21.57
30	30.30	28.24	21.56	21.01	21.49	22.03	22.63	21.14	21.46
35	30.08	28.28	22.37	22.43	22.00	23.83	23.01	21.42	21.88
40	29.53	28.14	22.48	20.97	20.71	22.04	20.65	21.06	21.77
45	27.80	27.97	21.93	21.20	20.98	21.52	21.58	21.07	20.96
50	29.24	27.99	22.30	21.40	21.26	21.53	22.69	20.61	21.88
Bootstrap sampling ratio 75%									
5	24.42	28.70	22.84	23.39	23.65	22.05	24.26	21.28	21.51
10	26.87	28.18	21.88	22.23	23.06	23.28	23.40	20.96	21.28
15	28.22	28.40	23.48	21.47	21.94	22.12	23.93	21.27	21.40
20	29.71	28.28	21.89	21.06	21.77	22.23	25.01	21.80	21.74
25	29.02	28.13	22.18	21.57	21.75	23.19	24.81	20.73	21.82
30	28.64	28.13	22.30	22.06	21.17	22.57	22.89	20.71	22.52
35	30.38	28.13	22.37	20.80	21.88	22.19	23.40	20.93	22.07
40	28.26	28.04	22.19	21.41	21.09	21.77	23.49	20.46	21.68
45	27.78	28.05	21.88	20.86	21.26	21.97	21.83	21.82	22.48
50	28.93	27.97	22.01	22.19	21.20	22.88	22.26	21.11	22.10

Table 5 Results for the “two-spiral” data set with bootstrap samples

Algorithm	25% sample ratio		50% sample ratio		75% sample ratio	
	Error rate (%)	<i>L</i>	Error rate (%)	<i>L</i>	Error rate (%)	<i>L</i>
BBSCE	42.35	5	41.73	5	40.33	10
CSPA	40.15	5	40.38	15	40.92	5
MCLA	39.28	25	38.98	40	38.20	5
EM	38.77	30	38.65	30	38.17	5
QMI	39.15	40	38.62	5	38.35	10
<i>k</i> -means	39.00	45	38.93	30	38.07	45
Spectral	38.35	35	39.35	30	39.37	45
H-Single	46.27	50	46.13	50	43.60	35
H-Complete	39.20	20	38.67	35	38.50	35

For the two-spirals data sets, we only report the best performance of all algorithms under different bootstrap sample ratios, and the corresponding ensemble size in Table 5 due to space limitations. The average clustering error rate in bold type in each column of Table 5 is the best one under the same bootstrap sampling ratio. Under different bootstrap sample ratios, spectral clusterer, QMI, and *k*-means get the best performance 38.35, 38.62 and 38.07%, respectively.

Thus in both toy data sets, the bootstrap sample ratio plays a little role on the performance of mostly algorithms. Moreover ensembles of very large size are less important.

4.2.2 UCI data sets

Due to space limitation, Table 6 only reports the average best results of all algorithms on UCI data sets with different bootstrap sample ratios and different ensemble sizes, where the error in bold type in each row is the best one for the corresponding data set. From this table, we can see that H-single is better than other methods. BBSCE has the best accuracy on the Wdbc data set, which is only 0.2% higher than the next highest H-Complete consensus function (see the first and ninth columns of Table 6). MCLA achieves the best accuracies on the Vote data set, but its advantage is very slight. The *k*-means consensus function has about 3% higher in accuracy than the next highest CSPA (see the second and the seventh columns of Table 6) on the Soy data set. The advantage of spectral consensus function on the Musk data set is not so obvious, only 0.3% higher than the H-Single clusterer. The H-Single consensus function performs well on the Glass, Liver and Wpbd data sets. It achieves almost 10% higher in accuracy than the next highest spectral consensus function (see the seventh and eighth columns of Table 6) on the Wpbc data set.

4.3 Experiments with feature subset method

Here, the diversity of the partitions in the first layer clustering is assured by using the feature subset method which are obtained by randomly sampling with replacement from the original data feature set. Here *r* denotes the feature

Table 6 Average clustering error rate (%) for seven UCI data sets with bootstrap samples

Data set	BBSCE	CSPA	MCLA	EM	QMI	<i>k</i> -means	Spectral	H-single	H-complete
Glass	52.15	61.42	59.52	55.90	57.02	55.56	51.25	50.16	55.20
Liver	44.21	46.82	45.50	45.53	44.59	44.68	43.82	42.17	44.71
Musk	44.52	45.46	45.46	45.60	44.85	45.18	44.29	44.59	45.28
Soy	46.47	42.47	87.49	43.19	44.23	39.77	44.98	47.32	43.36
Vote	17.72	14.19	13.29	13.30	13.32	13.31	14.32	13.33	13.30
Wdbc	6.92	15.01	7.21	7.21	7.17	7.17	11.39	7.19	7.11
Wpbd	38.79	42.21	38.60	38.72	38.13	38.35	33.23	24.90	37.78

Table 7 Average clustering error rate (%) for seven UCI data sets with feature subset

Data set	BBSCE	CSPA	MCLA	EM	QMI	<i>k</i> -means	Spectral	H-Single	H-Complete
Glass	53.15	58.83	57.15	54.22	56.11	51.95	50.55	53.27	52.38
Liver	45.24	46.34	45.40	46.21	45.27	44.83	42.53	42.12	44.48
Musk	44.89	45.20	45.45	45.57	45.17	44.47	44.52	43.36	44.45
Soy	47.43	40.47	89.08	43.07	43.81	39.53	47.14	49.10	43.49
Vote	15.13	14.11	13.21	13.38	13.39	13.57	14.26	36.93	13.40
Wdbc	7.33	14.47	7.33	7.18	7.19	7.12	7.64	27.04	7.04
Wpbd	38.75	41.70	38.45	39.24	38.82	38.08	29.53	24.21	36.11

sampling ratio. The number of samples with a part of features for individual clusterers in the first layer clustering is N , while the dimensionality of feature subset takes value from $\{\lceil d \times 0.25 \rceil, \lceil d \times 0.5 \rceil, \lceil d \times 0.75 \rceil\}$. For each data set, we also perform 30 independent runs and only report the average clustering errors on 30 runs.

Table 7 summarizes the average best results of all algorithms on UCI data sets with different feature sampling ratios and different ensemble sizes, where the error in bold type in each row is the best one for the corresponding data set. From this table, we can see that H-single is better than other methods. MCLA achieves the best accuracies on the Vote data set, but its advantage is very slight. The k -means consensus function also achieves 0.94% higher in accuracy than the next highest CSPA (see the second and the seventh columns of Table 7) on the Soy data set. On the Glass data set, the spectral consensus function gets the best accuracy which is 1.4% higher than the next highest k -mean clusterer. The H-Single consensus function performs well on the Liver, Musk, and Wpbd data sets, which is 0.41, 1.1 and 5.32% than the next corresponding highest consensus function, respectively.

From all experimental results, we have a conclusion that the framework of cascaded cluster ensembles is feasible and promising. But which clustering algorithm should we exploit in the second layer clustering? It is difficult to answer it. From the result analysis, we know the reduced hierarchical single-linkage has good performance in our experiments. As is known, however, each clustering algorithm has its own drawback and could not perform well on all types of data. In the reduced k -means clustering, the selected k centers may be not the optimal ones. The spectral clustering does not get the best performance, which does not mean that it is bad one. Here we used the standard k -means clustering in the spectral clustering, which may be one reason for explaining its results.

5 Conclusions

A framework for cascaded cluster ensembles is proposed, in which there are two layer clustering. The goal of the first layer is to generate different partitions. The first layer is a necessary step for all cluster ensembles. Various methods (consensus functions) were proposed to combine these different partitions to an optimal partition. Here we focus on taking clustering algorithms as consensus functions, which actually is the second layer clustering. In the view of mapping, the first layer could map the samples in the input space into a label attribute space spanned by labeled samples whose dimensionality equals the ensemble size. Some clustering algorithm is performed on these labeled samples instead of input samples. The clustering problem in the

second layer can be solved by existing clustering algorithms, such as k -means, fuzzy k -means, spectral, hierarchical clustering, and so on. We also give some reduced clustering algorithms, e.g., k -means, spectral, and hierarchical single-linkage, for the second layer clustering. An experimental comparison with other consensus functions shows that clusterer consensus functions have compared performance, where the reduced hierarchical single-linkage clustering methods perform well in our experiments.

In the second layer, we just use the reduced traditional clustering algorithms. There provide some new clustering algorithms, such as algorithms in [20–22]. Thus, we try to use some new stable clustering algorithm in the second layer in the future.

Acknowledgments We want to thank Dr. Alexander Topchy who graciously shares the code for comparison purposes. We would like to thank three anonymous reviewers and Editor Xi-Zhao Wang for their valuable comments and suggestions, which have significantly improved this paper. This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 60970067 and 61033013, by the Natural Science Foundation of Jiangsu Province of China under Grant No. BK2011284.

References

- Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. *ACM Comput Surv* 31(3):264–323
- Yu J (2005) General C-means clustering model. *IEEE Trans Pattern Anal Mach Intell* 27(8):1197–1211
- Chen H, Tino P, Yao X (2009) Predictive ensemble pruning by expectation propagation. *IEEE Trans Knowl Data Eng* 21(7):999–1013
- Wang XZ, Zhai JH, Lu SX (2008) Induction of multiple fuzzy decision trees based on rough set technique. *Info Sci* 178(16):3188–3202
- Zhang L, Zhou WD (2011) Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recogn* 44:97–106
- Kuncheva LI, Vetrov DP (2006) Evaluation of stability of k -means cluster ensembles with respect to random initialization. *IEEE Trans Pattern Anal Mach Intell* 28(11):1798–1808
- Zhou ZH, Tang W (2006) Clusterer ensemble. *Knowl Based Syst* 19(1):77–83
- Strehl A, Ghosh J (2002) Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *J Mach Learn Res* 3:583–617
- Fischer B, Buhmann JM (2003) Bagging for path-based clustering. *IEEE Trans Pattern Anal Mach Intell* 25(11):1411–1415
- Dudoit S, Fridlyand J (2003) Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* 19(9):1090–1099
- Topchy A, Jain A, Punch W (2004) A mixture model for clustering ensembles. In: *Proceedings of SIAM Data Mining*, pp 379–390
- Topchy A, Jain A, Punch W (2005) Clustering ensembles: models of consensus and weak partitions. *IEEE Trans Pattern Anal Mach Intell* 27(12):1866–1881
- Jia J, Xiao X, Liu B, Jiao L (2011) Bagging-based spectral clustering ensemble selection. *Pattern Recogn Lett* 32(10):1456–1467

14. Fred A (2001) Finding consistent clusters in data partitions. In: MCS '01: Proceedings of the Second International Workshop on Multiple Classifier Systems, Springer, London, pp 309–318
15. Fred A, Jain AK (2002) Data clustering using evidence accumulation. In: Proceedings of the 16th International Conference on Pattern Recognition, pp 276–280
16. Leisch F (1999) Bagged clustering, SFB “Adaptive Information Systems and Modeling in Economics and Management Science”, p 51
17. Domeniconi C, Papadopoulos D, Gunopoulos D, Ma S (2004) Subspace clustering of high dimensional data. In: Proceedings of the fourth SIAM International Conference on Data Mining, pp 517–521
18. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. SACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets 6(1):90–105
19. Murphy P, Aha D (1992) UCI machine learning repository URL: <http://www.ics.uci.edu/mllearn/MLRepository.html>
20. Liang J, Song W (2011) Clustering based on Steiner points. Int J Mach Learn Cybern. doi:10.1007/s13042-011-0047-7
21. Graaff AJ, Engelbrecht AP (2011) Clustering data in stationary environments with a local network neighborhood artificial immune system. Int J Mach Learn Cybern. doi:10.1007/s13042-011-0041-0
22. Guo G, Chen S, Chen L (2011) Soft subspace clustering with an improved feature weight self-adjustment mechanism. Int J Mach Learn Cybern. doi:10.1007/s13042-011-0038-8