CHAPTER

Unsupervised Learning via an Iteratively Constructed Clustering Ensemble

6

CHAPTER OUTLINE

6.1 Introduction 75
6.2 Iteratively Constructed Clustering Ensemble 76
6.2.1 Motivation 77
6.2.2 Model Description 80
6.3 Simulation 83
6.3.1 Cylinder-Bell-Funnel Data Set 83
6.3.2 Time Series Benchmarks 85
6.3.3 Motion Trajectory 88
6.4 Summary 91

6.1 INTRODUCTION

Although clustering ensembles have been widely recognized as providing a effective method for improving robustness, stability, and accuracy in unsupervised classification or clustering tasks, they always incur high computational costs in terms of time and memory consumption. These may be trivial issues for clustering tasks based on small data sets, but for temporal data clustering, they become critical, for example, our proposed Hiddern Markov Model-based clustering ensemble model presented in the previous chapter suffer a major problem of time consumption. Therefore, a trade-off solution between clustering performance and computational efficiency must be sought. A clustering ensemble in asswith a subsampling technique can bring improvements to clustering performance in terms of both accuracy and computational efficiency.

Essentially, ensemble approaches can be summarized in two categories. In the first (Strehl and Ghosh, 2003; Fred and Jain, 2005; Viswanath and Jayasurya, 2006), all independent ensemble members (partitions) on the training set are generated in parallel and the optimal solution is obtained by combining multiple partitions into a single consensus partition. However, the major disadvantage of this approach is the lack of interaction among individual partitions. For example, the generated multiple partitions are highly biased toward similar clustering structures, which may not be useful enough to improve the clustering performance.

Temporal Data Mining via Unsupervised Ensemble Learning. http://dx.doi.org/10.1016/B978-0-12-811654-8.00006-3

Copyright © 2017 Elsevier Inc. All rights reserved.

Therefore, variances such as bagging (Dudoit and Fridlyand, 2003; Fischer and Buhmann, 2003) are introduced in order to increase the diversity of multiple input partitions during the procedure of initial clustering analysis by bootstrapping the training set. In contrast, the second category of ensemble approaches, such as boosting (Frossyniotis et al., 2004; Pavlovic, 2004; Liu et al., 2007; Saffari and Bischof, 2007), provides a new iterative approach creating a meaningful combination of sequentially constructed partitions which deals with the hard-clustered data points in the previous iteration through the use of a "smart" weighting scheme.

Although much of the literature (Dudoit and Fridlyand, 2003; Fischer and Buhmann, 2003; Monti et al., 2003; Strehl and Ghosh, 2003; Weingessel et al., 2003; Frossyniotis et al., 2004; Fred and Jain, 2005; Viswanath and Jayasurya, 2006; Gionis et al., 2007; Liu et al., 2007; Saffari and Bischof, 2007; Singh et al., 2007) shows improved performance of those ensemble approaches in terms of robustness and quality of clustering tasks, each still has its own limitations. For instance, bagging introduces randomness in the production of diverse multiple input partitions, but without a proper objective, it improves clustering performance purely by chance. On the other hand, boosting adjusts the sample weights by using clustering error, which is determined by an objective function based on certain clustering quality measurements, in spite of the fact that clustering quality measurements are often unilateral and biased toward certain cluster structure.

In this chapter, we present a novel ensemble approach to iteratively construct an ensemble of partitions generated on subsets sampled from the original data set through a hybrid sampling scheme. The approach is inspired by both boosting and bagging techniques originally proposed for supervised learning tasks. Our approach combines the strengths of both boosting and bagging approaches while attempting to avoid their drawbacks. It has been applied on a set of synthetic and real-world data sets. Simulation results indicate that such approach yields favorable performance in general clustering tasks.

The rest of this chapter is organized as follows: Firstly, we discuss the most common clustering ensemble approaches related to this work, including both bagging and boosting, and then describe the motivation behind the proposed model. Subsequently, simulation results for a variety of clustering tasks are reported. A summary concludes this chapter.

6.2 ITERATIVELY CONSTRUCTED CLUSTERING ENSEMBLE

In this section, we provide a brief analysis of boosting and bagging, as they relate to proposed clustering ensemble approach, and then describe the motivation for proposing the model. Finally, the algorithm description presents the proposed clustering ensemble model.

6.2.1 MOTIVATION

For both of boosting and bagging, we can observe that the major difference between the approaches is their respective sampling schemes determining their own behaviors, which in turn creates both advantages and disadvantages in dealing with various structured data for clustering tasks.

The randomness introduced by the sampling scheme in bagging creates no guarantees for the performance of a clustering task. Furthermore, if there is a large data set, it is impossible to see the whole training set due to the computational cost. In such instance, a sampling scheme based on the bootstrap with replicates of the training set becomes unfeasible. Instead, we can apply the subsampling approach to generate the training sample, although this alternative often cancels out any gains achieved by a bagging ensemble.

Although the outstanding performance of boosting in classification tasks gives strong motivation for extending it as an unsupervised learning approach. it is not easy to develop a boosting algorithm for clustering as there is no ground truth for guiding learning procedure. There also remain some critical issues that have to be addressed. First, instance weights driven from clustering quality measures such as the objective function always emphasize the unilateral aspects of clustering quality, which differs from classification tasks that associate with ground truth as objective criteria. Thus, the instances weighting scheme may not be trustworthy. Second, for multiple partitions, the ordering of cluster labels is arbitrary, and it is not meaningful to compare such cluster labels resulted in multiple partitions. This is referred to as a cluster-correspondence problem. Finally, boosting is sensitive to noise and unbalanced clustered structure, which can cause some "important" instances to lose their opportunity to be trained.

In response, we believe that a hybrid sampling scheme derived from these ensemble approaches, where part of the subset is selected by a weighting scheme based on boosting with the rest of the set randomly picked up with a bagging approach, may combine each of their strengths while compensating for each of their weaknesses. In order to achieve a visual inspection on the demonstration, this idea is illustrated by the following example.

As illustrated in Fig. 6.1A, there is a two-dimensional synthetic data set which is subjected to a mixture of Gaussian distribution where there are five intrinsic clusters of heterogeneous structures. The ground-truth partition is given for evaluation and marked by green triangle (cluster 1), red circle (cluster 2), cyan square (cluster 3), red cross (cluster 4), and blue dot (cluster 5). A visual interpretation of the structure of the data set shown in Fig. 6.1 suggests that while clusters 1 and 2 are relatively separate, cluster 3 spreads widely and clusters 4 and 5 of different populations overlap each other.

Without any sampling approaches, we initially apply the single K-means algorithm on the whole data set and a simple clustering ensemble that combines 10 partitions obtained by running K-means with different restarts on the whole data set into a single partition based on the linear combination, while other ensemble algorithms with corresponding sampling approaches are also applied by using K-means

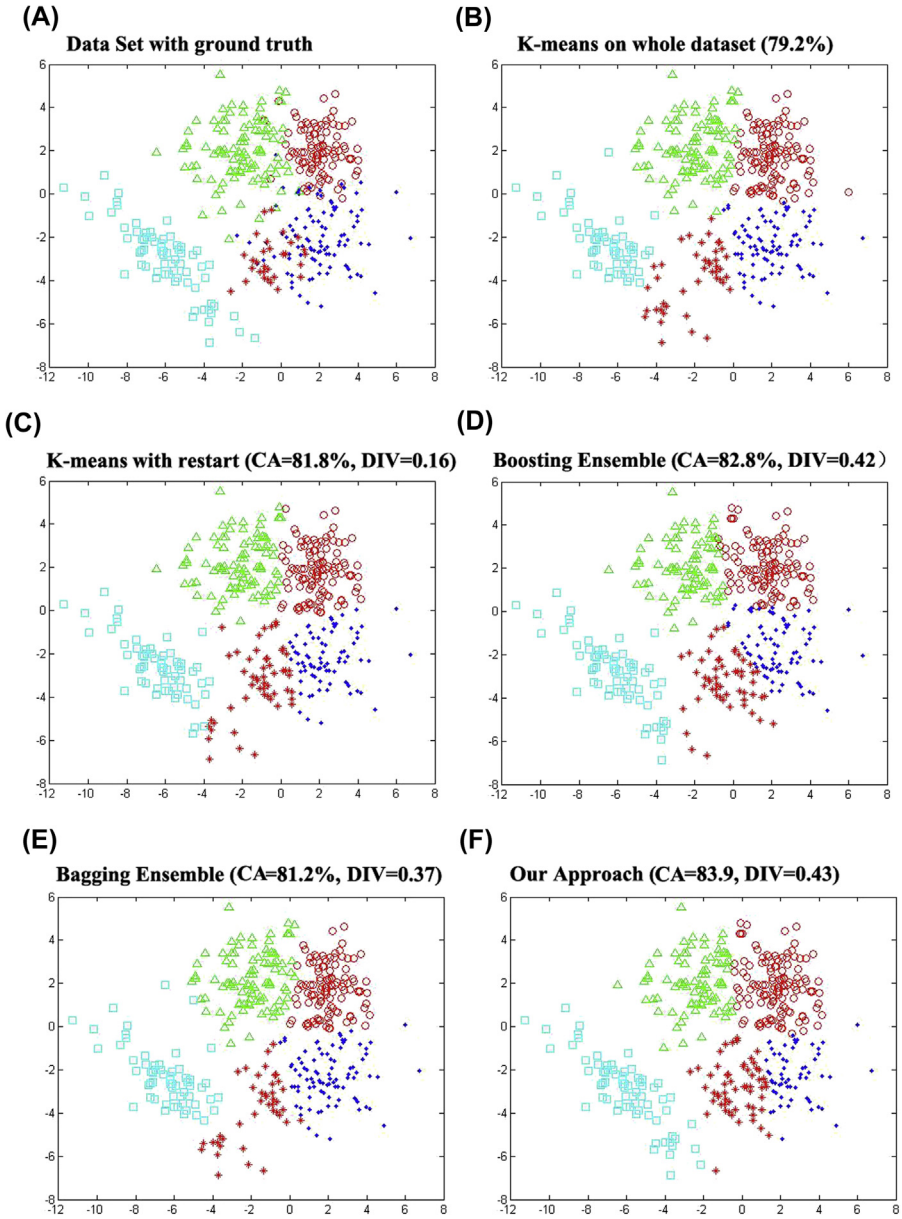

FIGURE 6.1

Results of various clustering approaches on synthetic data set with classification accuracy.

as base learner, where 10 partitions obtained from the subsampled training set (sampling rate, $\gamma = 0.1$) are combined into a single partition based on the linear combination.

By visual inspection, we evaluate the clustering quality of each method. All methods can well identify clusters 1 and 2 illustrated in Fig. 6.1B–F. With regard to the whole data set, Fig. 6.1B and C shows that the K-means and clustering ensemble based on restart K-means still fails to fully separate clusters 3–5, with a classification accuracy of 79.2% and 81.8% separately. Boosting attempts to focus on the hard-clustered instances to recycle the subtraining set sampled from this region by a weighting scheme. This produces a slightly better result in comparison with single K-means and clustering ensemble based on restart K-means, which gives a classification accuracy of 82.8%, as shown in Fig. 6.1D. However, the weighting scheme derived from a clustering quality measurement based only on cluster compactness and separability does not adapt to a complex cluster structure. In this structure, clusters 4 and 5 (of different populations) overlap each other and, in a few instances, belong to the manifold shape in cluster 3 at the bottom of Fig. 6.1D, which is set apart from the dense region of data points. As illustrated in Fig. 6.1D, part of cluster 4, cluster 5, and a few of the outliers in cluster 3 are improperly grouped together due to overtraining on this region by boosting. Bagging gives each instance an equal opportunity to be trained by a random sampling approach and combines the parallel generated partitions based on majority voting. This is more likely to result in clusters of a balanced structure. As illustrated in Fig. 6.1E, bagging is able to separate clusters 4 and 5 but fails to group cluster 3 where part of cluster 3 is merged into cluster 4, achieving a classification accuracy of 81.2%. In contrast, our proposed clustering ensemble partly adds random sampling based on bagging to weight-based sampling based on boosting, where part of the subtraining set ($\eta = 0.5$) is obtained by boosting sampling schema and the rest of the subtraining set ($1 - \eta$) is constructed using random sampling from the input space based on bagging ensemble approach, thereby building up a hybrid sampling scheme. This significantly reduces biases caused by a weighting scheme based on the unilateral clustering quality measurement and also maintains the gains of bagging, which is insensitive to the noise data. Fig. 6.1F demonstrates an outstanding performance of our proposed approach in comparison to other methods.

Studies (Brown et al., 2002; Kuncheva and Whitaker, 2003; Kuncheva and Hadjitodorov, 2004) have indicated that more diversity offers larger improvement on the clustering ensemble than less diversity. In other words, higher diversity among ensemble members more likely produces higher clustering performance gain in the final consensus partition, and the diversity of clustering ensemble can be obtained by several manners. In this case, we mainly focus on combining different clustering results by sampling data to produce a final single result. Actually there are a number of different ways to measure the diversity of ensemble members including Rand Index, Jaccard Index, Adjusted Rand Index, Normalized Mutual Information (NMI) (Hadjitodorov et al., 2006). Most of them are based on label matching between two partitions, where two partitions are to be diverse if the labels

of one partition do not match well with the labels of the other. One diversity measure (DIV) commonly used in the literature is the NMI, which is formulated as following:

$$\mathrm{NMI}(P_a, P_b) = \frac{\sum_{i=1}^{K_a}\sum_{j=1}^{K_b} N_{ij}^{ab} \log\left(\frac{NN_{ij}^{ab}}{N_i^a N_j^b}\right)}{\sum_{i=1}^{K_a} N_i^a \log\left(\frac{N_i^a}{N}\right) + \sum_{j=1}^{K_b} N_j^b \log\left(\frac{N_j^b}{N}\right)} \tag{6.1}$$

$$\mathrm{DIV} = 1 - \frac{2}{T(T-1)} \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \mathrm{NMI}(P_i, P_j) \tag{6.2}$$

where a set of T input partitions $P = \{P_t\}_{t=1}^{T}$ obtained from a target data set, P_i, $P_j \in P$. DIV based on NMI is valued from 0 to 1, the higher value represents a collection of more diverse ensemble members obtained from the target data set. In order to explore the relation between diversity of clustering ensemble and performance based on classification accuracy, we further apply NMI-based DIV among the ensemble members obtained from different clustering algorithms. As illustrated in Fig. 6.1, the sampling-based clustering ensemble algorithms including boosting, bagging, and our approach produces much more diverse partitions with DIV value (0.42, 0.37, 0.43) shown in Fig. 6.1D–F. In contrast, the clustering ensemble produces less diversity among the input partitions obtained by applying K-means with restart on the whole training data set, which gives a DIV value of 0.16 shown in Fig. 6.1C. It is obvious that the restart K-means-based clustering ensemble results in a higher classification accuracy with lower DIV value than bagging ensemble, which produces an inconsistent relation between diversity and performance for clustering ensemble. Therefore, we believe that the diversity among ensemble members still cannot be solely used as main factor to judge the performance of clustering ensemble, which has been mentioned in Section 4.4. However, the partition yielded by our proposed model is very close to the ground truth in Fig. 6.1A with the highest classification accuracy of 83.9% and DIV value of 0.43, strongly encouraging the development of an iteratively constructed clustering ensemble with a novel sampling scheme.

6.2.2 MODEL DESCRIPTION

Based on the motivation described previously, we propose a sampling-based clustering ensemble approach; an iteratively constructed clustering ensemble. This approach iteratively constructs the multiple partitions on the subset of whole input instances selected by a hybrid of boosting and bagging sampling schemes as illustrated in Fig. 6.2. At each iteration, weights over instances are updated, part of the subset defined by the fraction parameter η is chosen according to the weights over instances as a selection probability, and the rest of the subset $(1 - \eta)$ is constructed using random sampling from the input data set. Then, a clustering algorithm

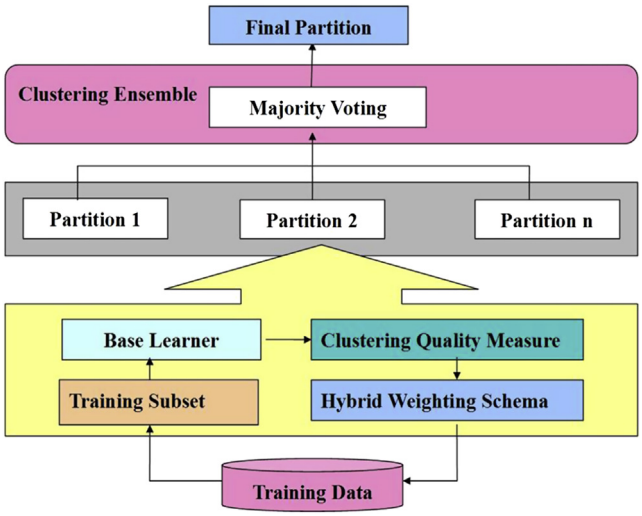

FIGURE 6.2

Iteratively constructed clustering ensemble (Yang and Chen, 2010).

such as K-means is applied to partition the subsets in order to generate several partitions, then the partitions are combined into a final result using a majority voting, where the weights of the partition measure the clustering quality. The detailed algorithm description is given as follows:

1. Set all instance weights equal, $W^t = \{w_i^t\}_{i=1}^N = 1/N$ and $t = 1$.
2. Produce a training set consisting of two parts from the original data set. One part of training set with higher weights are selected by boosting sampling schema, the rest of training set are randomly selected by bagging sampling schema.
3. Given a predefined clustering number K*, K-means is applied as a base learner on the training set in order to produce a partition P_t.
4. Get the cluster hypothesis, $H_i^t = \left(h_{i,1}^t, h_{i,2}^t, \ldots, h_{i,k*}^t\right)$ for each of instance, where $h_{i,j}^t$ is a membership degree of instance x_i to cluster $C_j^t \in P_t$. It is defined as $h_{i,j}^t = \dfrac{1}{\sum_{k=1}^{K^*} \dfrac{d(x_i, \mu_j)}{d(x_i, \mu_k)}}$, where $d(x_i,\mu_j)$ and $d(x_i,\mu_k)$ are Euclidean distance between instance x_i and centers (μ_j, μ_k) of cluster $(C_j^t \in P_t, C_k^t \in P_t)$, respectively.
5. Form an $N \times K^*$ cluster hypothesis, $H^t = \left[h_{i,j}^t\right]_{i=1,\ldots,N;\ j=1,\ldots,K^*}$ for whole data set. If $t > 1$, renumber the cluster indexes of H^t using the aggregate hypothesis, H_{ag}^{t-1}, in the previous iteration for cluster-correspondence problem by Hungarian algorithm (Winston and Goldberg, 1994). Otherwise, go directly to Step 6.

6. Compute the pseudo loss $\varepsilon_t = \frac{1}{2}\sum_{i=1}^{N} w_i^t Q_i^t$ and set current partition weight $\beta_t = \frac{1-\varepsilon_t}{\varepsilon_t}$, where Q_i^t is clustering quality measurement to evaluate how well an instance x_i is clustered in current partition P_t and defined as $Q_i^t = 1 - \max\left(h_i^t\right) - \min\left(h_i^t\right)$.
7. Update weights of instances $w_i^{t+1} = \frac{w_i^t \beta_t^{Q_i^t}}{Z_t}$, where Z_t is a normalization constant
8. Aggregate cluster hypothesis $H_{ag}^t = \sum_{\tau=1}^{t}\left[\frac{\log(\beta_\tau)}{\sum_{j=1}^{t}\log\left(\beta_j\right)}H^\tau\right]$ and $t = t+1$. If stopping criteria ($t > T$) is satisfied go to Step 9, otherwise go to Step 2.
9. Output the final cluster hypothesis H_{ag}^T and final partition $H_{ag}^T(x_i) = \arg\max_{k=1,\ldots,K^*}\sum_{t=1}^{T}\left[\frac{\log(\beta_t)}{\sum_{j=1}^{T}\log\left(\beta_j\right)}h_{i,k}^t\right]$, where $H_{ag}^T(x_i)$ is the cluster label of instance x_i in the final partition. A pseudocode is also given

Input:

- *a data set* $X = \{x_1, x_2, \ldots, x_N\}$.
- *an integer* K^* *(intrinsic cluster number)*
- *an integer T (number of iterations)*
- *an integer* γ *(sampling rate)*
- *an integer*η *(fraction rate)*
- *the K-means clustering algorithm KM*
- *the Euclidean distance function d*
- *the Hungarian algorithm HUNGARIAN*

Initialize $W^t = \left\{w_i^t\right\}_{i=1}^{N} = 1/N$, *set* $t = 1$
while $t \le T$
  Produce a training set consisting of two parts from the original dataset:
  $|X_{training}| = \gamma|X|$
  $X_{training} = \{X_{boosting}, X_{bagging}\}$
  $X_{boosting}$ *is selected according to instance weights* W^t *(Boosting sampling)*
  $X_{bagging}$ *is randomly selected (Bagging sampling)*
  $|X_{training}| = \eta|X_{boosting}| + (1-\eta)|X_{bagging}|$
  $P_t = Kmeans(X_{training}, K^*)$
  for i = 1 to N
    for j = 1 to K^*
    $h_{i,j}^t = \frac{1}{\sum_{k=1}^{K^*}\frac{d(x_i,\mu_j)}{d(x_i,\mu_k)}}$; //$\mu_j = \overline{X}_{C_j^t}$ *and* $\mu_k = \overline{X}_{C_k^t}$ *(cluster j and k in partition* P_t, $C_j^t \in P_t$, $C_k^t \in P_t$*)*
    end for
  end for
  $H^t = [h_{i,j}^t]_{N\times K^*}$;
  if $t > 1$

$H^t = HUNGARIAN\left(H^t, H_{ag}^{t-1}\right)$;
end if
$Q_i^t = 1 - \max\left(h_i^t\right) - \min\left(h_i^t\right)$; $\varepsilon_t = \frac{1}{2}\sum_{i=1}^{N} w_i^t Q_i^t$; $\beta_t = \frac{1-\varepsilon_t}{\varepsilon_t}$;
$w_i^{t+1} = \frac{w_i^t \beta_t^{-Q_i^t}}{Z_t}$;// Z_t *is a normalization constant*
$H_{ag}^t = \sum_{\tau=1}^{t}\left[\frac{\log(\beta_\tau)}{\sum_{j=1}^{t}\log\left(\beta_j\right)} H^\tau\right]$; $t = t+1$
end while
$H_{ag}^T = H_{ag}^t$
Output: *the final partition* $H_{ag}^T(x_i) = \arg\max_{k=1,\ldots,K^*} \sum_{t=1}^{T}\left[\frac{\log(\beta_t)}{\sum_{j=1}^{T}\log\left(\beta_j\right)} h_{i,k}^t\right]$, *where* $H_{ag}^T(x_i)$ *is the cluster label of instance* x_i *in the final partition.*

6.3 SIMULATION

In this section, we present our experimental methodology and simulation results. Although the outcome of clustering analysis can be used for miscellaneous tasks, we focus only on clustering-based classification tasks in simulations. We apply the proposed ensemble approach to a synthetic time series—Cylinder-Bell-Funnel, a collection of 16 synthetic or real-world time series data sets (Keogh, 2003), and motion trajectories database (CAVIAR, 2002). In the following section, we report the performance of our model in three benchmark clustering tasks.

6.3.1 CYLINDER-BELL-FUNNEL DATA SET

This data set consists of three classes of data, cylinder (c), bell (b), or funnel (f). They are already used for evaluating the proposed HMM-based hybrid meta-clustering ensemble model presented in the previous chapter. The details of the data set have been described in Section 5.3.2. For comparison, we employ five algorithms on this synthetic data. First, we apply K-means on the whole data set as a baseline, as well as a clustering ensemble algorithm achieved by applying restart K-means on the whole data set with the correct number of clusters ($K = 3$). Then, both boosting and bagging clustering ensemble approaches are implemented while K-means remains as the base learner. Given the fact that both ensemble approaches were developed without addressing model selection, we also use the correct number of clusters ($K = 3$) in K-means, while the best parameter setup ($\gamma = 0.1$, $\eta = 0.5$) are given for our approach. Thus, 10 partitions generated are combined by different ensemble approaches. For our approach, we use exactly the same procedure for K-means to produce partitions on the subset with the hybrid sampling scheme described earlier. After visual inspection, we represent the best clustering results obtained by these algorithms in the original time domain illustrated in Fig. 6.3B–F. It is observed from Fig. 6.3B that the whole data set can be seen, and K-means still presents significant challenges in directly performing clustering analysis on high-dimensional data such as time series, achieving a classification accuracy of 61.1%. On the other hand, by using the same K-means algorithms as base learners, restarting K-means-based clustering ensemble without sampling on the whole

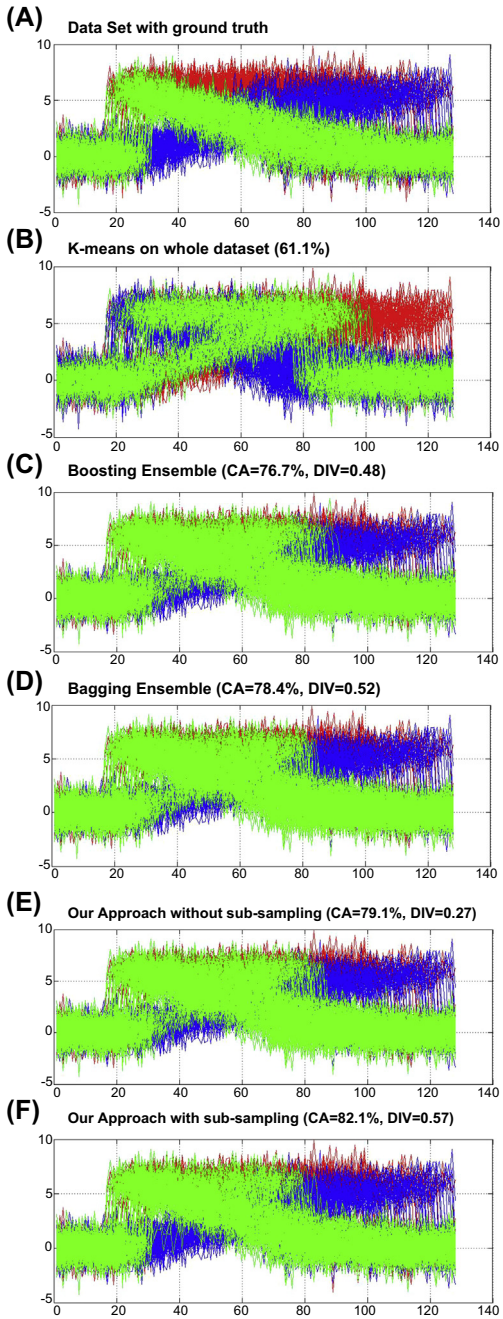

FIGURE 6.3

Results of various clustering approaches on CBF data set. *CBF*, Cylinder-Bell-Funnel; *DIV*, diversity measure.

dataset and both boosting and bagging ensemble approaches with the corresponding sampling techniques obtain better clustering results (79.1, 76.7%, and 78.4%). However, due to the sampling techniques, the boosting and bagging clustering ensemble algorithms produces much more diversity among the ensemble members, which is evaluated by an NMI-based DIV (0.48, 0.52) shown in Fig. 6.1D and E than clustering ensemble achieved by applying K-means with restart on the whole data set, which results DIV = 0.27 shown in Fig. 6.1B. As illustrated in Fig. 6.3F, the proposed approach with a hybrid sampling on the whole data set produces much more diverse input partitions (DIV = 0.53), which gives more opportunity that input partitions capture a significant amount of different features based on the intrinsic cluster structure from different perspectives. As a result, it achieves the best performance based on classification accuracy of 82.1% where the final partition yielded by our proposed model is very close to the ground truth in Fig. 6.3A in comparison with the other four algorithms.

Furthermore, we examine the performance of our approach using different values of η. As illustrated in Fig. 6.4, the average classification accuracy of our approach varies with the input parameter without any regularity. In fact, additional experiments, not reported here, have been performed in order to explore the nature of subtraining set fraction parameter λ on different data sets. Based on the results, we realize that is extremely difficult to establish a general framework to determine the optimal value of this input parameter on various data sets, which would be an open question for future research.

6.3.2 TIME SERIES BENCHMARKS

For investigating the performance of ensemble learning, we further apply the proposed iteratively constructed clustering ensemble associated various static clustering algorithms (Hierarchical clustering, K-NN, and K-means) as base learner and relative clustering ensemble algorithms (boosting and bagging) on time series benchmarks of 16

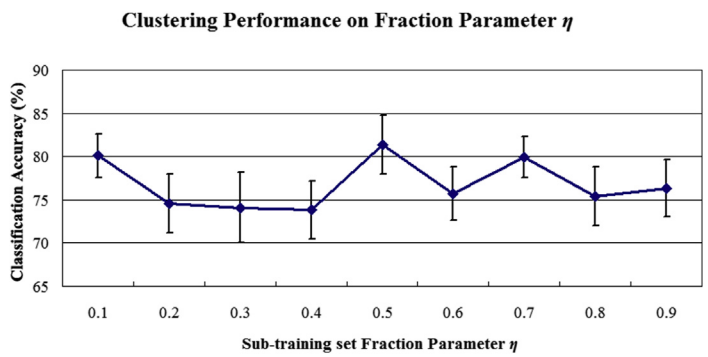

FIGURE 6.4

Performance of iteratively constructed clustering ensemble model on subtraining set fraction parameter η (Yang and Chen, 2010).

synthetic or real-world time series data sets (Keogh, 2003), previously used in Chapter 5 for evaluating our proposed HMM-based clustering ensemble model. The information on all 16 data sets has been presented in Section 5.3.3.

In this experiment, we also use a whole data set by merging training and testing subsets for the evaluation of clustering algorithms. Given the fact that these ensemble approaches were developed without addressing model selection, the correct cluster number is given for the initial clustering analysis and the optimal parameter setup invoked in our approach is manually determined by an exhausted search, which is shown in Table 6.1 for each of time series benchmarks.

Table 6.2 lists all the results achieved in this experiment. For each algorithm, we run the experiment 10 times with best parameter setup, and the best result is reported in the table, where the performance of the K-means algorithm is provided by benchmark collectors. It is observed from Table 6.2 that our approach associated with three different basic clustering algorithms as base learner, all achieves the promising results, as it has the best performance on 14 of 16 data sets in comparison with other clustering algorithms, where the K-means, boosting ensemble clustering, and bagging ensemble clustering algorithms only win on one, zero, and one data sets, respectively. For comparison between different clustering algorithms as base learner invoked in our proposed iteratively constructed clustering ensemble, the results reported in the last three columns of Table 6.2 are quite even, where there is not any algorithm that had had outstanding performance than others. Comparably, K-means-based approach outperforms other methods, as it has the best performance on 6 out

Table 6.1 Optimal Parameter Setup of Our Approach on Time Series Benchmarks

Data Set	Number of Clusters (K*)	Sampling Rate (γ)	Sub-Training Set Fraction Rate (λ)
Syn control	6	0.50	0.50
Gun-point	2	0.20	0.25
CBF	3	0.50	0.50
Face (all)	14	0.50	0.50
OSU leaf	6	0.33	0.33
Swedish leaf	15	0.50	0.50
50 words	50	0.50	0.50
Trace	4	0.50	0.50
Two patterns	4	0.50	0.50
Wafer	2	0.50	0.50
Face (four)	4	0.50	0.50
Lightning-2	2	0.20	0.50
Lightning-7	7	0.50	0.83
ECG	2	0.33	0.50
Adiac	37	0.83	0.25
Yoga	2	0.17	0.17

Table 6.2 Classification Accuracy (%) of Clustering Algorithms on Time Series Benchmarks

Data Set	K-means	Bagging	Boosting	Our Approach (HIE)	Our Approach (K-NN)	Our Approach (K-means)
Syn control	67.9	70.7	73.0	60.8	71.0	**74.3**
Gun-point	50.0	50.0	65.1	**69.2**	57.3	56.7
CBF	62.6	64.2	64.3	66.2	**69.4**	64.8
Face (all)	36.0	39.2	38.9	34.0	40.1	**40.4**
OSU leaf	37.8	39.1	39.3	38.6	**43.3**	41.8
Swedish Leaf	40.6	43.5	41.6	32.0	**45.7**	44.6
50 words	42.0	42.4	39.0	**43.2**	39.4	40.9
Trace	48.5	52.5	52.9	50.6	**53.9**	53.5
Two patterns	32.2	33.2	32.4	**34.4**	33.3	33.2
Wafer	62.5	62.5	62.7	62.7	**63.6**	62.5
Face (four)	**66.9**	63.4	66.1	61.2	62.8	64.0
Lightning-2	61.1	62.9	64.8	63.1	64.6	**68.6**
Lightning-7	48.4	46.5	43.2	**50.9**	45.3	50.0
ECG	69.8	69.8	**76.5**	72.7	72.4	72.0
Adiac	38.4	40.8	37.2	24.6	43.8	**44.4**
Yoga	51.7	50.5	50.9	50.1	51.4	**51.9**

of 16 data sets. Hierarchical clustering and K-NN as base learner achieves best results for five data sets. As a result, we believe that our approach provides a more general framework for ensemble learning, where most of conventional clustering algorithms can be used as a base learner.

6.3.3 MOTION TRAJECTORY

In order to explore a potential application, we apply our approach to the CAVIAR database for trajectory clustering analysis. The CAVIAR database (CAVIAR, 2002) was previously used in Chapter 5 for evaluating the proposed HMM-based clustering ensemble model.

Motion trajectories tend to have various lengths, and therefore, a normalization technique would be required to unify the various lengths of motion trajectories. In this experiment, the motion trajectory is resampled with a prespecified number of sample points (1500) by a polynomial interpolation algorithm. After resampling, all trajectories are normalized to a Gaussian distribution of zero mean and unit variance in x and y directions. It is different from previous experiments presented in Section 5.3.4, the motion trajectories as 2D time series data of the notation $\{(x(t), y(t))\}_{t=1}^{T}$, where $(x(t),y(t))$ are the coordinates of an object tracked at frame t, are represented by concatenating two time series corresponding to its x and y projection into single dimensional time series in this simulation which are shown in Fig. 6.5.

In our simulation, the proposed approach is applied to all trajectories for clustering analysis. Since information on the "right" number of clusters is unavailable, we run the K-means algorithm 10 times on a subset of the training set by manually choosing a K value of 14 generating 10 partitions that are fed to the proposed ensemble model in order to yield a final partition with user-defined input parameter ($\eta = 0.35$), as shown in Fig. 6.6. Without the ground truth, human visual inspection has to be applied for evaluating the results, as suggested (Khalid and Naftel, 2005).

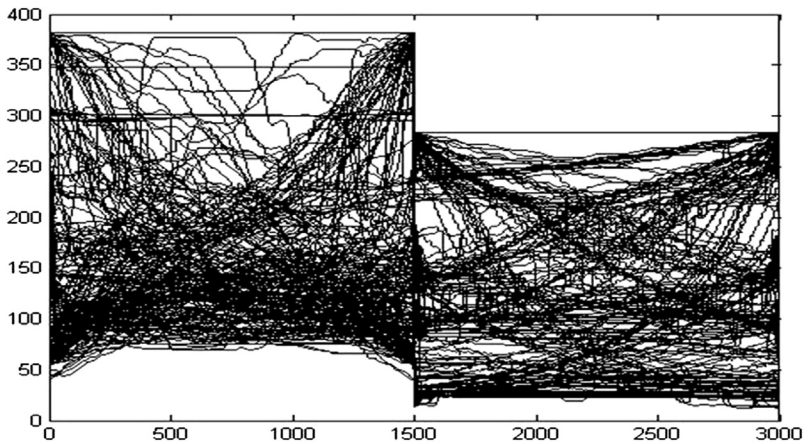

FIGURE 6.5

Preprocessed trajectories in the CAVIAR database.

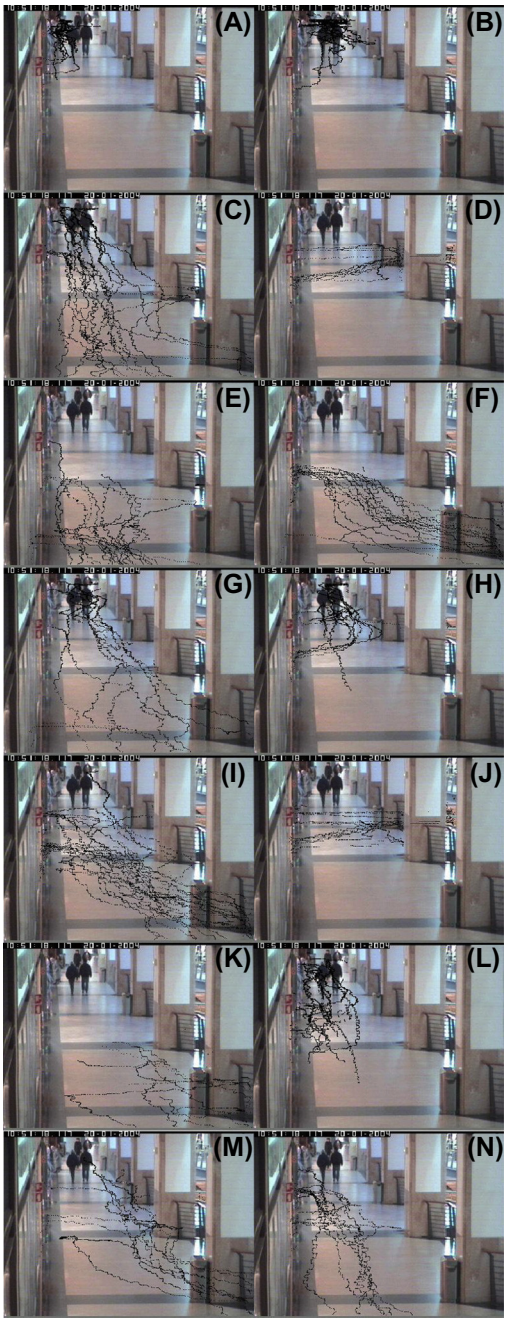

FIGURE 6.6

A clustering analysis of all moving trajectories on the CAVIAR database made by iteratively constructed clustering ensemble model. Plots in A—N correspond to 14 clusters of moving trajectories in the final partition (Yang and Chen, 2010).

Using common visual experience, behaviors of pedestrians across the shopping mall are roughly divided into five categories from the camera viewpoint: "move up," "move down," "stop," "move left," and "move right" (CAVIAR, 2002). Ideally, trajectories of similar behaviors are grouped together along a motion direction, and then results of clustering analysis are used to infer different activities at a semantic level, for example, "enter the store," "exit from the store," "pass in front," and "stop to watch."

As observable in Fig. 6.6, coherent motion trajectories have been properly grouped together while dissimilar ones are distributed into different clusters. For example, the trajectories corresponding to the activity of "stop to watch" are accurately grouped in the cluster shown in Fig. 6.6A, while the "walk and watch" trajectories are grouped into the cluster shown in Fig. 6.6M. Trajectories corresponding to movement from left-to-right and right-to-left are properly grouped into two separate clusters, as shown in Fig. 6.6D and J. Trajectories corresponding to "move up" and "move down" are grouped very well into two clusters as shown in Fig. 6.6C and G. Fig. 6.6B, F, H, I, L, and N indicate that trajectories corresponding to most activities of "enter the store" and "exit from the store" are properly grouped together via multiple clusters in light of various starting positions, locations, moving directions, and so on. Finally, Fig. 6.6E and K illustrate two clusters roughly corresponding to the activity "pass in front."

As mentioned in the Chapter 5, the former algorithm (HMM-based clustering ensemble) revealed a major problem of computational cost, which becomes much more critical for real-world application such as performing clustering analysis on CAVIAR database of motion trajectories (execution time, 4623.44s). In order to improve the algorithm efficiency, we intuitively proposed a new approach as the iteratively constructed clustering ensemble associated with a hybrid sampling technique presented in this chapter. For comparison of computational cost resulting in both proposed ensemble approaches, we also monitor the duration of clustering analysis on the CAVIAR database by applying the ensemble model presented in this chapter. Surprisingly, it is round 377 times faster than HMM-based clustering ensemble approach and results an execution time of only 12.27 seconds, which sufficiently demonstrates that the reduction of computational costs can be achieved by several manners, for example, implementing a simple clustering algorithm such as K-means as base learner, a simple consensus function such as majority voting, and reducing the training set by subsampling on the whole data set for ensemble learning approach.

To simulate trajectories of corrupted data in the real-world environment, we have performed one more additional experiment in classification by adding different amounts of Gaussian noise $N(0,\sigma)$, to the range of coordinates and removed five segments of trajectory of the identical length at random locations. Missing segments of various lengths are used for testing where a decision is made by finding a cluster whose center is closest to the tested trajectory in terms of the Euclidean distance to see if its clean version belongs to this cluster. Apparently, the classification accuracy depends largely on the quality of clustering analysis. Fig. 6.7 shows

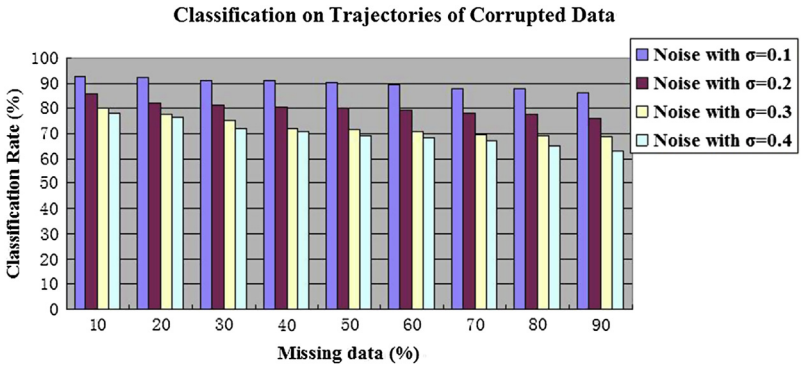

FIGURE 6.7

Performance of the iteratively constructed clustering ensemble model on CAVIAR with corrupted data appears (Yang and Chen, 2010).

performance evolution in the presence of missing data measured by a percentage of the trajectory length added by different amounts of noise. It is evident that our approach performs well in real-world situations.

In summary, all the previously mentioned simulation results suggest that our model leads to robust clustering analysis. Therefore, the results may be used for higher level video-content analyses.

6.4 SUMMARY

In this chapter, we have presented an unsupervised learning model for clustering tasks by using an ensemble of iteratively constructed partitions on the subtraining set obtained by a hybrid sampling scheme. Simulation results on different types of data sets demonstrate that our model yields favorable results. The experiments based on a set of synthetic and real data sets have justified the legitimacy of the fundamental concept behind our proposed approach. For real-world applications, the robustness and feasibility have been further evaluated using the motion video trajectory data set.

A similar ensemble algorithm (Kotsiantis and Pintelas, 2004) has also been developed for supervised learning. It combines two subensembles obtained by bagging and boosting on the same training set into a single classifier with sum rule voting in a parallel procedure. Although this algorithm uses similar ideas to those used in the construction of our algorithm, the classifiers are individually generated by bagging and boosting resulting in a lack of interaction between these classifiers. Furthermore, this algorithm would be unfeasible for large data sets due to high computational cost. In contrast, our algorithm develops a hybrid sampling scheme maximizing the synergy between bagging and boosting for clustering tasks,

and the iteratively constructed partitions on the subtraining set significantly improve the interaction of multiple clustering solutions. This, in turn, leads to much faster computation. It is noteworthy that the efficiency issue is critical for some real-world applications, for example, temporal data and large data set clustering.

There are several issues to be studied in the ongoing research. First, the selection of the optimal number of clusters continues to pose a well-known model-selection problem for this approach. Without being given this a priori information, a rather poor performance is always produced. Next, although the clustering ensemble approach is able to make significant improvements on the single clustering algorithm and obtains a trade-off solution for arbitrary-structured data sets, the appropriate clustering algorithm as a base learner would further optimize clustering performance. Therefore, selection of a proper clustering algorithm for certain characterized data could be a further research topic. Finally, we released the proposed clustering ensemble model that has the major difficulties of dealing with the data sets with various lengths, which requires the data sets to be of uniform length, and combining the input partitions with different number of clusters due to the limitation of majority voting combination.