CHAPTER

7 Temporal Data Clustering via a Weighted Clustering Ensemble With Different Representations

CHAPTER OUTLINE

7.1 INTRODUCTION

As presented in previous chapters, proximity and model-based clustering approaches directly work on temporal data where temporal correlation is dealt directly during clustering analysis by means of temporal similarity measures (Jain et al., 1999; Keogh and Kasetty, 2003; Xu and Wunsch, 2005; Ding et al., 2008), for example, dynamic-time warping or dynamic models (Smyth, 1999; Policker and Geva, 2000; Murphy, 2002; Xiong and Yeung, 2002; Liu and Brown, 2004), for example, hidden Markov model (HMM). As presented in Chapter 5, a model-based approach, *HMM-based meta-clustering ensemble*, has been initially proposed in order to mainly solve problems in finding the intrinsic number of clusters and model initialization problems. However, it is quite time consuming, especially for the temporal data with huge volume and high dimensionality. Therefore, a

Temporal Data Mining via Unsupervised Ensemble Learning. http://dx.doi.org/10.1016/B978-0-12-811654-8.00007-5

proximity-based approach, *Iteratively constructed clustering ensemble*, presented in Chapter 6 has been proposed in order to reduce the computational cost and provide a meaningful combination of input partitions by a hybrid sampling technique. However, it strictly requires the prior number of clusters, the data sets with uniform length, and input partitions resulting in identical number of clusters. In order to solve the problems existed in both proposed clustering ensemble models, we consider that a feature-based approach would be an alternative solution.

Basically, a feature-based approach attends to convert temporal data into a lower dimensionality in feature space. Here, temporal data clustering supports the use of any static data−clustering algorithm, creating high computational efficiency. In the last 20 years, numerous representations have been proposed for temporal data clustering (Faloutsos et al., 1994; Dimitrova and Golshani, 1995; Chen and Chang, 2000; Keogh et al., 2001; Chakrabarti et al., 2002; Bagnall and Janacek, 2004; Bashir, 2005; Cheong et al., 2005; Bagnall et al., 2006; Gionis et al., 2007; Ding et al., 2008; Ye and Keogh, 2009). Nevertheless, one representation tends to encode only those features well presented in its representation space, which inevitably causes the loss of other useful information conveyed in the original temporal data. Due to the high complexity and varieties of temporal data, to our knowledge, there is no universal representation that perfectly characterizes miscellaneous temporal data. Therefore, a representation is merely applicable to a class of temporal data where their salient features can be fully captured in the representation space, but such information is hardly available without prior knowledge and a careful analysis. Furthermore, the aforementioned model-selection problem is still unavoidable, which is independent of the use of a representation.

In Section 4.5, we described in depth the essence of the clustering ensemble: a data set is subjected to a multiple partition combination in the expectation of producing a consensus partition superior to that of the given input partitions. Growing empirical evidence supports such a concept, indicating new cluster structures that the clustering ensemble is capable of detecting. Moreover, analyses reveal that under certain conditions, a consensus solution will in fact uncover the intrinsic underlying structure of a given data set (Topchy et al., 2004). These are strong, generic techniques enabling the joint use of different representations in temporal data clustering.

In response to the previously mentioned data and building on previous successes with different representations in complex pattern classification tasks (Chen et al., 1997; Chen, 1998; Chen and Chi, 1998; Chen, 2005a,b; Wang and Chen, 2007), this chapter presents an approach which can overcome weaknesses inherent in representation-based temporal data−clustering analysis. Our approach consists of initial clustering analysis on different representations to produce multiple partitions and clustering ensemble construction to produce a final partition by combining those partitions achieved in initial clustering analysis. While initial clustering analysis can be done by any existing clustering algorithms, we propose a novel weighted clustering ensemble (WCE) algorithm of a two-stage reconciliation process. In the proposed algorithm, a weighting consensus function reconciles input partitions to candidate consensus partitions according to various clustering validation criteria.

Then, an agreement function yields the final partition by further reconciling the candidate consensus partitions.

The contributions of this work are summarized as follows. First, we develop a practical temporal data clustering model by different representations via clustering ensemble learning to overcome the fundamental weakness in the representation-based temporal data−clustering analysis. Next, we propose a novel WCE algorithm, which not only provides an enabling technique to support our model but also can be used to combine any input partitions. Finally, we demonstrate the effectiveness and the efficiency of our model for a variety of temporal data clustering tasks as well as its easy-to-use nature as all internal parameters are fixed in our simulations.

In the rest of the chapter, first we address temporal data representation issues, then motivation and the proposed approach is described, and then the proposed WCE algorithm is presented along with algorithm analysis. After this, simulation results on a variety of temporal data clustering tasks are reported. At the end of the chapter we discuss issues relevant to our approach and draws conclusions.

7.2 WEIGHTED CLUSTERING ENSEMBLE WITH DIFFERENT REPRESENTATIONS OF TEMPORAL DATA

In this work, we propose a weighted consensus function guided by clustering validation criteria to reconcile initial partitions to candidate consensus partitions from different perspectives and then introduce an agreement function to further reconcile those candidate consensus partitions to a final partition. As a result, the proposed WCE algorithm provides an effective enabling technique for the joint use of different representations, which cuts the information loss in a single representation and exploits various information sources underlying temporal data. In addition, our approach tends to capture the intrinsic structure of a data set, for example, the number of clusters.

In this section, we first describe the motivation to propose our temporal data clustering model. Then, we present such model working on different representations via clustering ensemble learning.

7.2.1 MOTIVATION

It is known that different representations encode various structural information facets of temporal data in their representation space. For illustration, we perform the principal component analysis (PCA) on four typical representations described in Section 2.2.2 of a synthetic time series data set. The data set is produced by the stochastic function $F(t) = A\sin(2\pi\alpha t + B) + \varepsilon(t)$, where A, B, and α are free parameters and $\varepsilon(t)$ is the added noise drawn from the normal distribution N(0,1). The use of four different parameter sets (A, B, α) leads to time series of four classes and 100 time series in each class.

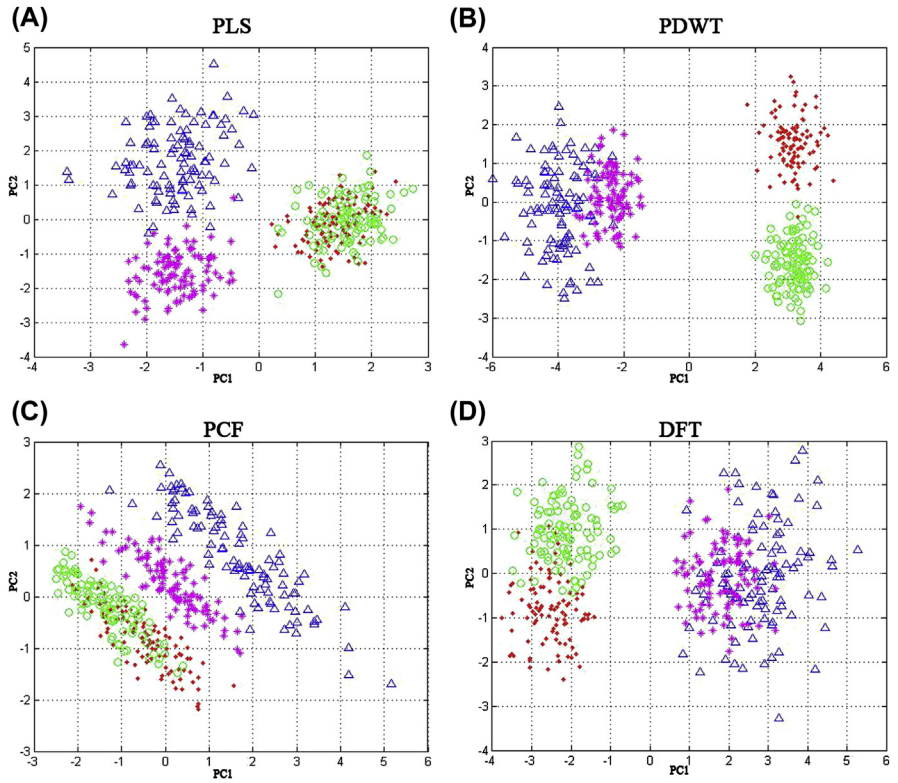

FIGURE 7.1

Distributions of the time series dataset in various principal component analysis representation manifolds formed by the first two principal components of their representations (Yang and Chen, 2011a). (A) PLS (B) PDWT (C) PCF (D) DFT.

Fig. 7.1 shows four representations of time series with various distributions in their PCA representation subspaces. Classes marked with blue triangles and magenta stars easily separate from the other two overlapped classes in Fig. 7.1A, while the same is true for the classes marked by green circles and red dots in Fig. 7.1B. Moreover, we can also gain such structural information from plots in Fig. 7.1C and D. Intuitively, our observation suggests that a single representation simply captures partial structural information, and the joint use of different representations is more likely to capture the intrinsic structure of a given temporal data set. When a clustering algorithm is applied to different representations, diverse partitions would be generated. To exploit all information sources, we need to reconcile diverse partitions to find out a consensus partition superior to any input partitions.

From Fig. 7.1, we also observe that the unlikelihood that partitions yielded by a clustering algorithm will carry an equal amount of useful information, due to their

distributions in different representation spaces. The fact that most extant clustering ensemble methods treat all partitions equally during the reconciliation process, merely brings about an averaging effect. Based on our previous work (Yang, 2006; Yang and Chen, 2006), we have discovered various adverse outcomes in this area: clustering ensemble methods often yield a worse final partition, when the partitions to be combined are radically different. Moreover, the averaging effect is particularly harmful in a majority voting mechanism, especially as many highly correlated partitions appear inconsistent with the intrinsic structure of a given data set. All of this supports our belief in applying a different treatment to the input partitions: the measurement of their contributions with a weighted consensus function.

Without the ground truth, the contribution of a partition is actually unknown in general. Fortunately, existing clustering validation criteria (Halkidi et al., 2001) measure the clustering quality of a partition from different perspectives, for example, the validation of intraclass and interclass variation of clusters. To a great extent, we can employ clustering validation criteria to estimate contributions of partitions in terms of clustering quality. However, a clustering validation criterion often measures the clustering quality from a specific viewpoint only by simply highlighting a certain aspect. In order to estimate the contribution of a partition precisely in terms of clustering quality, we need to use various clustering validation criteria jointly. This idea is empirically justified by a simple example in the following section. In the following description, we omit technical details of WCEs, which will be presented in Section 7.2.3, for illustration only.

Fig. 7.2A shows a two-dimensional synthetic data set subject to a mixture of Gaussian distribution where there are five intrinsic clusters of heterogeneous structures, and the ground-truth partition is given for evaluation and marked by red (cluster 1), blue (cluster 2), black (cluster 3), cyan (cluster 4), and green (cluster 5). The visual inspection on the structure of the data set shown in Fig. 7.2A suggests that clusters 1 and 2 are relatively separate while cluster 3 spreads widely, and clusters 4 and 5 of different populations overlap each other.

Applying the K-means algorithm on different initialization conditions, including the center of clusters and the number of clusters, to the data set yields 20 partitions. Using different clustering validation criteria (Halkidi et al., 2001), we evaluate the clustering quality of each single partition. Fig. 7.2B–D depict single partitions of maximum value in terms of different criteria. The Dunn's Validity Index (DVI) criterion always favors a partition of balanced structure. Although the partition in Fig. 7.2B meets this criterion well, it properly groups clusters 1–3 only but fails to work on clusters 4 and 5. The Modified Huber's Γ index (MHΓ) criterion generally favors partition with bigger number of clusters. The partition in Fig. 7.2C meets this criterion but fails to group clusters 1–3 properly. Similarly, the partition in Fig. 7.2D fails to separate clusters 4 and 5 but is still judged as the best partition in terms of the Normalized Mutual Information (NMI) criterion that favors the most common structures detected in all partition.

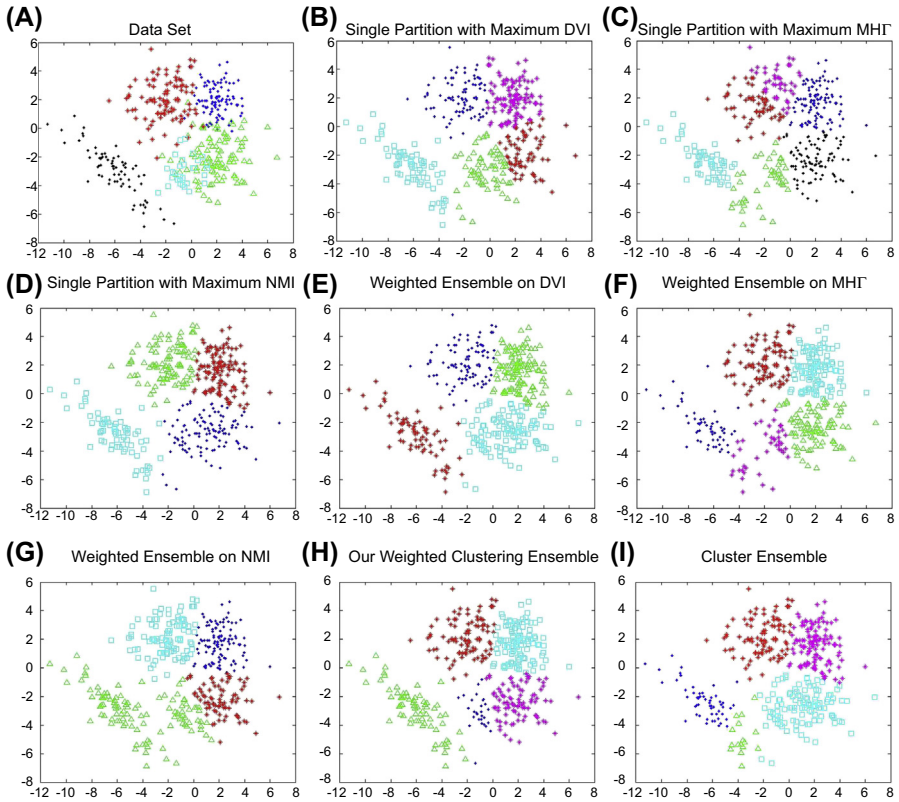

FIGURE 7.2

Results of clustering analysis and clustering ensembles (Yang and Chen, 2011a). (A) The data set of ground truth. (B) The partition of maximum DVI. (C) The partition of maximum MHΓ. (D) The partition of maximum NMI. (E) DVI WCE. (F) MHΓ WCE. (G) NMI WCE. (H) Multiple criteria WCE. (I) The Cluster Ensemble (Strehl and Ghosh 2003). *DVI*, Dunn's Validity Index; *MHΓ*, Modified Huber's Γ index; *NMI*, Normalized Mutual Information; *WCE*, weighted clustering ensemble.

By the use of a single criterion to estimate the contribution of partitions in the WCE, it inevitably leads to incorrect consensus partitions as illustrated in Fig. 7.2E–G, respectively. As three criteria reflect different yet complementary facets of clustering quality, the joint use of them to estimate the contribution of partitions become a natural choice. As illustrated in Fig. 7.2H, the consensus partition yielded by the multiple criteria–based WCE is very close to the ground truth in Fig. 7.2A. As a classic approach, cluster ensemble (CE) (Strehl and Ghosh, 2003) treats all partitions equally during reconciling input partitions. When applied to this data set, it yields a consensus partition shown in Fig. 7.2I that fails to detect the intrinsic structure underlying the data set.

In summary, the previous intuitive demonstration strongly suggests the joint use of different representations for temporal data clustering and the necessity of developing a WCE algorithm.

7.2.2 MODEL DESCRIPTION

Thus motivated, we propose a temporal data clustering model with a WCE working on different representations. As illustrated in Fig. 7.3, the model consists of three modules, that is, feature extraction, initial clustering analysis, and WCE.

1. In the representation extraction module, different representations are extracted by transforming raw temporal data to feature vectors of fixed dimensionality for initial clustering analysis. Because various representations of a complementary nature are demanded in our ensemble model, we recommend the use of both piecewise and global temporal data representations. Therefore, four representations presented in Section 2.2.2 would be applied in the simulation.
2. In the initial clustering analysis module, a clustering algorithm is applied to different representations received from the representation extraction module. As a result, a partition for a given data set is generated based on each representation. When a clustering algorithm of different parameters is used, for example, K-means, more partitions based on a representation would be produced by

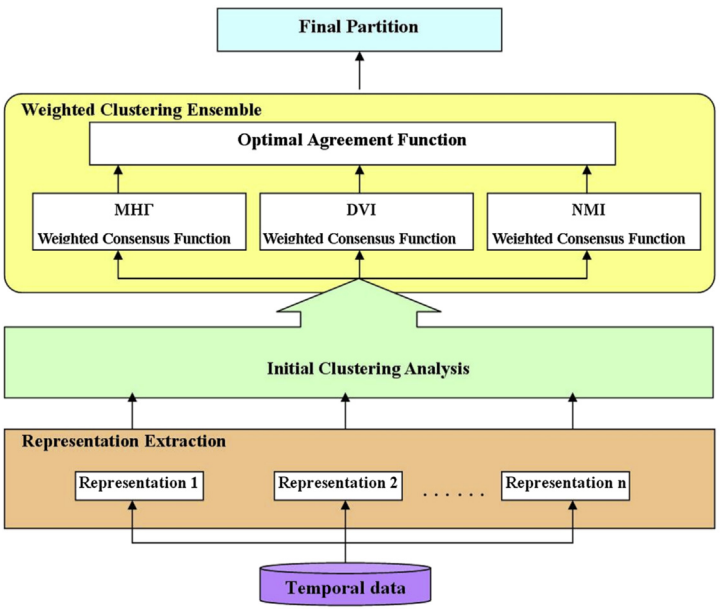

FIGURE 7.3

WCE with different representations (Yang and Chen, 2011a).

running the algorithm on various initialization conditions. Thus, the clustering analysis on different representations leads to multiple partitions for a given data set. All partitions achieved will be fed to the WCE module for the reconciliation to a final partition.

3. In the WCE module, a weighted consensus function works on three clustering validation criteria to estimate the contribution of each partition received from the initial clustering analysis module. The consensus function with single criterion−based weighting schemes yields three candidate consensus partitions, respectively, as presented in Section 7.2.3. Then, candidate consensus partitions are fed to the agreement function presented in Section 7.2.4, to form a final agreed partition where the number of clusters is automatically determined.

A pseudo code is also provided in the following section.
Input:

- *a data set X = {x1, x2,…, xN}.*
- *an integer R (number of feature representations)*
- *an integer T (number of iterations)*
- *the feature extraction function FEATURE*
- *the clustering algorithm, CLUSTERING*
- *Clustering validity index π*
- *DSPA consensus function, DSPA*

for r = 1 to R.
  fr = FEATURE$_r$(X)
  for t = 1 to T.
    $P_t^r = CLUSTERING(f_r)$;
  end for
end for

$$P = \left\{P_t^r\right\}_{t=1,\ldots,T;\ r=1,\ldots,R};\ M = T \times R \rightarrow |P|;\ P_m \in P,$$

$$w_m^{MHT} = \frac{\pi(P_m)}{\sum_{m=1}^{M}\pi(P_m)},\ \pi = MHT$$

$$w_m^{DVI} = \frac{\pi(P_m)}{\sum_{m=1}^{M}\pi(P_m)},\ \pi = DVI$$

$$w_m^{NMI} = \frac{\pi(P_m)}{\sum_{m=1}^{M}\pi(P_m)},\ \pi = NMI$$

for m = 1 to M.
  Construct a binary membership indicator matrix of P_m: $H_m = \{0,1\}^{N\times K_m}$
  Compute its similarity matrix: $S_m = H_m H_m^{Tr}$
end for

$$S^{MHT} = \sum_{m=1}^{M} w_m^{MHT} S_m;\ S^{DVI} = \sum_{m=1}^{M} w_m^{DVI} S_m;\ S^{NMI} = \sum_{m=1}^{M} w_m^{NMI} S_m;$$

$$P^{MHT} = DSPA\left(S^{MHT}\right);\ P^{DVI} = DSPA\left(S^{DVI}\right);\ P^{NMI} = DSPA\left(S^{NMI}\right);$$

Construct a binary membership indicator matrix $H = \left[H^{MHT} \middle| H^{DVI} \middle| H^{NMI}\right]$ *based on consensus partitions.* $\left\{P^{MHT}, P^{DVI}, P^{NMI}\right\}$

Compute the similarity matrix: $\overline{S} = \frac{1}{3} HH^{Tr}$

$$\overline{P} = DSPA\left(\overline{S}\right)$$

Output: the final partition $\overline{P}$.

7.2.3 WEIGHTED CONSENSUS FUNCTION

Weighted consensus function lies the essential idea of using the pairwise similarity between objects in a partition for evident accumulation. In this approach, a pairwise similarity matrix is derived from weighted partitions and weights are determined by measuring the clustering quality with different clustering validation criteria, then, a dendrogram (Jain et al., 1999) is constructed based on all similarity matrices to generate candidate consensus partitions.

Partition Weighting Scheme

Assume that $X = \{\mathbf{x}_n\}_{n=1}^{N}$ is a data set of N objects and there are M partitions $P = \{P_m\}_{m=1}^{M}$ on X, where the cluster number in M partitions could be different, as obtained by initial clustering analysis. Our partition weighting scheme assigns a weight w_m^{π} to each P_m in terms of a clustering validation criterion π, and weights for all partitions based on the criterion π collectively form a weight vector $\mathbf{w}^{\pi} = \left\{w_m^{\pi}\right\}_{m=1}^{M}$ for the partition collection P. In the partition weighting scheme, we define a weight

$$w_m^{\pi} = \frac{\pi(P_m)}{\sum_{m=1}^{M} \pi(P_m)}, \tag{7.1}$$

where $w_m^{\pi} > 0$ and $\sum_{m=1}^{M} w_m^{\pi} = 1$. $\pi(P_m)$ is the clustering validity index value in terms of the criterion π. Intuitively, the weight of a partition would express its contribution to the combination in terms of its clustering quality measured by the clustering validation criterion π.

In order to estimate the contribution of a partition, we examine as many different aspects of clustering quality as possible. After looking into all existing of clustering validation criteria, we select three criteria of complementary nature for generating weights from different perspectives, that is, MHΓ, DVI, and NMI. As has already been mentioned in Section 3.3, these criteria each measure only an aspect of clustering quality and all behave differently. For instance, MHΓ strongly favors a partition with more clusters. As with MHΓ, the DVI is insensitive to the number of clusters in a partition but is significantly less robust due to its use of a single linkage

distance and the diameter information of clusters. As a result, it is quite sensitive to noise or outliers for any cluster of a large diameter. Intuitively, a high NMI value suggests a well-accepted partition that is more likely to reflect the intrinsic structure of a given data set. This criterion shows bias toward the highly correlated partitions and favors clusters containing a similar number of objects.

Finally three weight vectors, $\mathbf{w}^{MHT}$, $\mathbf{w}^{DVI}$, and $\mathbf{w}^{NMI}$ are obtained by respectively substituting π for a specific clustering validity index and are then used to weigh the similarity matrix, respectively.

Weighted Similarity Matrix

For each partition P_m, a binary membership indicator matrix $H_m = \{0,1\}^{N\times K_m}$ is constructed where K_m is the number of clusters in the partition P_m. In the matrix H_m, a row corresponds to one datum and a column refers to a binary encoding vector for one specific cluster in the partition P_m. Entities of the column with one indicates that the corresponding objects are grouped into the same cluster and zero otherwise. Now we can use the matrix H_m to derive an $N \times N$ binary similarity matrix S_m which encodes the pairwise similarity between any two objects in a partition. For each partition P_m, its similarity matrix $S_m = \{0,1\}^{N\times N}$ is constructed by

$$S_m = H_m H_m^{Tr} \tag{7.2}$$

In Eq. (7.2), the element $(S_m)_{ij}$ is equal to the inner product between rows i and j of the matrix H_m. Therefore, objects i and j are grouped into the same cluster if the element $(S_m)_{ij} = 1$ and in different clusters otherwise.

Finally, a weighted similarity matrix S^{π} concerning all the partitions in P is constructed from a linear combination of their similarity matrix S_m and their weight w_m^{π} as

$$S^{\pi} = \sum_{m=1}^{M} w_m^{\pi} S_m. \tag{7.3}$$

In this approach, three weighted similarity matrices, S^{MHT}, S^{DVI}, and S^{NMI} are constructed, respectively.

Candidate Consensus Partition Generation

A weighted similarity matrix S^{π} is used to reflect the collective relationship between all data in terms of different partitions and a clustering validation criterion π. However, the weighted similarity matrix actually tends to accumulate evidence in terms of clustering quality and hence treats all partitions differently.

Motivated by the technique used by Fred and Jain (2005), we employ the dendrogram-based similarity partitioning algorithm (DSPA) developed in our previous work (Yang and Chen, 2007) to produce a candidate consensus partition from a weighted similarity matrix S^{π}. DSPA algorithm makes use of an average-link hierarchical clustering (HC) algorithm. This converts the weighted similarity matrix into a dendrogram (Jain et al., 1999) with all the data in a given data set indexed in its horizontal axis and the lifetime of all possible cluster formations expressed in the vertical.

The lifetime of a cluster in the dendrogram is defined as an interval from the moment that this cluster is created to the moment that it disappears by merging with other clusters. Here, we would emphasize that, due to the use of a weighted similarity matrix, the lifetime of clusters is weighted by clustering quality in terms of a specific clustering validation criterion. Therefore, the dendrogram will differ from that yielded by a similarity matrix without a weighting component (Fred and Jain, 2005).

Consequently, the number of clusters in the candidate consensus partition P^{π} can be determined automatically by cutting the dendrogram derived from S^{π} to form clusters at the longest lifetime. Using the DSPA algorithm, we achieve three candidate consensus partitions P^{π}, $\pi = \{\text{MH}\Gamma, \text{DVI}, \text{NMI}\}$, in this approach.

7.2.4 AGREEMENT FUNCTION

The proposed weighted consensus function yields three candidate consensus partitions, respectively, according to three different clustering validation criteria. In general, these partitions are not always consistent with each other (see Fig. 7.2 for example), and hence, a further reconciliation is required for a final partition as the output of the proposed clustering ensemble model.

In order to obtain a final partition, we develop an agreement function by means of the evident accumulation (Fred and Jain, 2005) again. A pairwise similarity $\overline{S}$ is constructed with three candidate consensus partitions in the same way as described in Section 7.2.3. That is, a binary membership indicator matrix H^{π} is constructed from partition P^{π} where $\pi = \{\text{MH}\Gamma, \text{DVI}, \text{NMI}\}$. Then, concatenating three H^{π} matrices leads to an adjacency matrix consisting of all the data in a given data set versus candidate consensus partitions, $H = \left[H^{MH\Gamma} \middle| H^{DVI} \middle| H^{NMI}\right]$. Thus, the pairwise similarity matrix $\overline{S}$ is achieved by

$$\overline{S} = \frac{1}{3} H H^{Tr} \tag{7.4}$$

Finally, a dendrogram is derived from $\overline{S}$ and the final partition $\overline{P}$ is achieved by using DSPA algorithm.

7.2.5 ALGORITHM ANALYSIS

Under the assumption that any partition of a given data set is a noisy version of its ground-truth partition subject to the normal distribution (Topchy et al., 2004), the clustering ensemble problem can be viewed as finding a "mean" partition of input partitions in general. If we know the ground-truth partition, P_c, and all possible partitions, P_t, of the given data set, the ground-truth partition would be the "mean" of all possible partitions (Topchy et al., 2004):

$$P_C = \arg\min_{P} \sum_{i} \Pr(P_i = P_c) d(P_i, P), \tag{7.5}$$

where $\Pr(P_i = P_c)$ is the probability that P_c, is randomly distorted to be P_t and $d(\bullet, \bullet)$ is a distance metric for any two partitions. Under the normal distribution assumption, $\Pr(P_i = P_c)$ is proportional to the similarity between P_t and P_c.

In a practical clustering ensemble problem, an initial clustering analysis process returns only a partition subset, $\mathbf{P} = \{P_m\}_{m=1}^{M}$. From Eq. (7.5), finding the "mean", P^*, of M partitions in P can be performed by minimizing the cost function:

$$\Phi(P) = \sum_{m=1}^{M} \mu_m d(P_m, P), \tag{7.6}$$

where $\mu_m \propto \Pr(P_m = P_c)$ and $\sum_{m=1}^{M} \mu_m = 1$. The optimal solution to minimizing (7.6) is the intrinsic "mean", P^* :

$$P^* = \arg\min_{P} \sum_{m=1}^{M} \mu_m d(P_m, P). \tag{7.7}$$

In this work, we use the piecewise similarity matrix to characterize a partition and hence define the distance as $d(P_m, P) = \|S_m - S\|^2$ where S is the similarity matrix of a consensus partition, P. Thus, Eq. (7.6) can be rewritten as

$$\Phi(P) = \sum_{m=1}^{M} \mu_m \|S_m - S\|^2. \tag{7.8}$$

Let S^* be the similarity matrix of the "mean", P^*. Finding P^* to minimize $\Phi(P)$ in Eq. (7.8) is analytically solvable (Cox et al., 2006), that is, $S^* = \sum_{m=1}^{M} \mu_m S_m$. By connecting this optimal "mean" to the cost function in Eq. (7.8), we have

$$\begin{aligned}
\Phi(P) &= \sum_{m=1}^{M} \mu_m \|S_m - S\|^2 \\
&= \sum_{m=1}^{M} \mu_m \|(S_m - S^*) + (S^* - S)\|^2 \\
&= \sum_{m=1}^{M} \mu_m \|(S_m - S^*)\|^2 + \sum_{m=1}^{M} \mu_m \|(S^* - S)\|^2.
\end{aligned} \tag{7.9}$$

Note that the fact $\sum_{m=1}^{M} \mu_m \|S_m - S^*\| = 0$ is applied in the last step of Eq. (7.9) due to $\sum_{m=1}^{M} \mu_m = 1$ and $S^* = \sum_{m=1}^{M} \mu_m S_m$. The actual cost of a consensus partition is now decomposed into two terms in Eq. (7.9). The first term corresponds to the quality of input partitions, for example, how close they are to the ground-truth partition, P_c, solely determined by an initial clustering analysis regardless of clustering ensemble. In other words, the first term is constant once the initial clustering analysis returns a collection of partitions, P. The second term is determined by the performance of a clustering ensemble algorithm that yields the consensus partition, P, that is, how close the consensus partition is to the weighted "mean" partition. Thus, Eq. (7.9) provides a generic measure to analyze a clustering ensemble algorithm.

In our WCE algorithm, a consensus partition is an estimated "mean" characterized in a generic form: $S = \sum_{m=1}^{M} w_m S_m$, where w_m is w_m^{π} yielded by a single clustering validation criterion π, or $\overline{w}_m$ produced by the joint use of multiple criteria.

By inserting the estimated "mean" in the previously mentioned form and the intrinsic "mean" into Eq. (7.9), the second term of Eq. (7.9) becomes

$$\sum_{m=1}^{M} \mu_m \left\| \sum_{m=1}^{M} (\mu_m - w_m) S_m \right\|^2. \tag{7.10}$$

From Eq. (7.10), it is observed that the quantities $|\mu_m - w_m|$ critically determine the performance of a WCE.

It has been assumed in clustering analysis that an underlying structure can be detected if it holds well-defined cluster properties, such as compactness, separability, and cluster stability (Jain et al., 1999; Halkidi et al., 2001; Xu and Wunsch, 2005). We expect that such properties to be measurable by clustering validation criteria so that w_m is as close to μ_m as possible. In reality, the ground-truth partition is generally not available for a given data set, and without the knowledge of μ_m, it is impossible to formally analyze how good any given clustering validation criteria are for estimating intrinsic weights. Instead, we undertake empirical studies to investigate its capacity and limitation of our WCE algorithm based on elaborately designed synthetic data sets of the ground-truth information, which is presented in Appendix.

7.3 SIMULATION

In this section, we present our experimental methodology and report simulation results. Despite the wide range of applications suitable to clustering analysis, we focus only on clustering-based classification tasks which organize objects into groups based on chosen similarity measures (Euclidean distance on time series benchmarks and CAVIAR database, correlation on Physiological Data Modeling Contest - PDMC time series data stream), with no labeling information on the target data set. In simulations, to examine the legitimacy of fundamental concepts associated with our approach, we apply our proposed WCE to several benchmark data sets for time series data mining (Keogh, 2003). An objective trajectory benchmark of the CAVIAR visual tracking database (CAVIAR, 2002) and a time series data stream of the PDMC sensor data set (PDMC, 2004) are used to evaluate the robustness and feasibility of the proposed approach.

7.3.1 TIME SERIES BENCHMARKS

Time series benchmarks of 16 synthetic or real-world time series data sets (Keogh, 2003) have been collected to evaluate a number of temporal data classification and clustering algorithms in the context of temporal data mining. In this collection, the ground truth, that is, the class label of time series in a data set, is available and each data set is further divided into the training and testing subsets in advance for the evaluation of a classification algorithm. This benchmarks has been also used for evaluating the previous proposed ensemble approaches presented in this book, the information on all 16 data sets is tabulated in 5.3, including the number of classes, the number of time series, and the length of time series in every data set.

Three sets of experiments are carried out in this simulation. First, we employ classic temporal-proximity and model-based approaches on this time series benchmark. There are Dynamic Time Warping-based K-means (DTW K-means), HC algorithm, and K-means-based HMM K-model clustering, the performance of the K-means algorithm is provided by benchmark collectors. Second, we examine clustering performance on four representations to see if the use of a single representation is sufficient for temporal data clustering. For this we employ two well-known algorithms, that is, K-means, an essential algorithm and, DBSCAN, a more sophisticated density-based algorithm that can discover clusters of arbitrary shape and offers good model selection ability. Third, we apply our WCE algorithm to combine the input partitions collectively yielded by K-means, HC, and DBSCAN with different representations during the initial clustering analysis.

For the first set of experiments, we directly apply DTW K-means, HC and HMM K-models under the same condition on the benchmarks, where allows both of DTW K-means and HMM K-model to use the correct cluster number, K^*, the results of K-means is provided by benchmark collector, and no parameter setting in HC. For second set of experiments, K^* is also pre-defined for K-means. Although DBSCAN requires no prior knowledge of a given data set, two parameters in the algorithm need to be tuned for the good performance (Ester et al., 1996). As suggested in the literature (Ester et al., 1996), we find the best parameters for each data set by an exhaustive search within a proper parameter range. Given the fact that K-means is sensitive to initial conditions even though K^* is given, we run the algorithm 10 times on each data set with different initial conditions in the two aforementioned experiments. The results are used for comparison to the clustering ensemble. Thus, only the best results are reported, in the interests of fairness.

For the third set of experiments, we use K-means as a base learner without prior knowledge, K*, to produce partitions on different representations, which is designed to test model-selection capability of the proposed ensemble model. As a result, we take the following procedure to produce a partition by K-means. With a random number generator of uniform distribution, we draw a number within the range $K^* - 2 \leq K \leq K^* + 2(K > 0)$ used in K-means to produce a partition on an initial condition. For each data set, we repeat the previously mentioned procedure to produce 10 partitions on a single representation so that there are totally 40 partitions during initial clustering analysis. As mentioned previously, there is no parameter setting in the HC and, therefore, only one partition is yielded for a given data set. Thus, we need to combine only four partitions returned by the HC on four representations for each data set. When DBSCAN is used in initial clustering analysis on different representations, we combine four best partitions of each data set achieved from single-representation experiments mentioned previously. Here, we emphasize that there is no parameter tuning in our simulations. Internal parameters of the representations for all 16 data sets are fixed, and as used by the benchmark collectors (Keogh, 2003), the classification accuracy described in Section 3.3.1 is used to evaluate the performance of compared approaches.

Table 7.1 lists all results achieved in the three sets of experiments. The first part of results shows that DTW K-means outperforms K-means and HC, HMM K-model, on 11 of the 16 data sets. Given the fact of that HC is capable of finding a cluster number, we also report the model-selection performance of HC noting that * is added behind the classification accuracy if it finds the correct cluster number. As a result, HC manages to find the correct cluster number for four data sets only, indicating the challenges inherent in clustering temporal data of high dimensions. It is necessary to mention that the HMM K-model takes a considerably longer time to perform when compared with other algorithms, including the proposed clustering ensembles.

The second part of results also shows that, with regard to clustering performance on single representations, no one can always lead to best performance on all data sets, regardless of which algorithm is used or that the winning performance is achieved across four representations. The results simply demonstrate how difficult the choice of representation is. DBSCAN is generally better than K-means algorithm on the appropriate representation space. In total, it correctly detects the right cluster number for 9 of 16 data sets with appropriate representations and the best parameter setup. This implies the difficulties in model selection in the single representation space, in other words, a sophisticated algorithm does not perform well due to lost information at the representation extraction stage.

In terms of both classification accuracy and model selection, and regardless of the algorithm used in initial clustering analysis, the third part of the table shows the significantly enhanced performance of the proposed ensemble approach. In general, the ensemble result on different representations outperforms the best result yielded by the same algorithm on single representations, and even the average result achieved by the proposed ensemble model is better than the best result from single representations when K-means is used. In model selection, the proposed ensemble model fails to detect the right cluster number for only 2 and 1 of the 16 data sets only, respectively, with K-means and HC used in initial clustering analysis. While the proposed ensemble model with a base learner of DBSCAN is successful for detecting the correct number of clusters on 10 of the 16 data sets. The bold entries in Table 7.1 shows the best results achieved. It is observed that the proposed ensemble approach achieves the best classification accuracy for 12 of the 16 data sets, in which K-means, HC, and DBSCAN are used as base learner to achieve best performance for six, two, and four data sets, respectively.

For comparison, we further employ three state-of-the-art ensemble learning approach on the benchmarks, these approaches include: CE (Strehl and Ghosh, 2003), hybrid bipartite graph formulation (HBGF) algorithm (Fern and Brodley, 2004), and semi-definite programming−based clustering ensemble (SDP-CE) (Singh et al., 2007). This simulation uses K-means for initial clustering analysis. Due to the fact of that all three ensemble approaches were developed without reference to model-selection problems, we use the correct cluster number K^* for each data set. Therefore, 10 partitions are generated with different initial conditions on

Table 7.1 Classification Accuracy (%)[a] of Different Clustering Algorithms on Time Series Benchmarks

	Time Series				Single Representation								Different Representations		
					K-means				DBSCAN				WCE		
Data Set	K-means	DTW K-means	HC	HMM K-model	PCF	DFT	PLS	PDWT	PCF	DFT	PLS	PDWT	K-means	HC	DBSCAN
Syn control	67.9	69.8	59.5	69.1	58.3	62.4	64.7	68.5	34.5	70.0*	70.4*	55.8	**86.1 ± 2.5***	73.8*	78.2*
Gun-point	50.0	65.6	41.9*	43.8	44.3	43.2	47.0	48.3	48.1*	51.5*	42.4	50.0*	54.1 ± 1.8*	**69.1***	52.0*
CBF	62.6	**80.9**	50.9	60.1	53.5	49.3	52.9	61.2	53.8*	46.7	51.7	62.9*	63.9 ± 2.1*	70.7*	63.2*
Face (all)	36.0	49.4	31.9	37.8	31.5	30.2	32.1	33.4	13.6	34.7	19.9	32.3	**51.9 ± 1.4***	50.2*	33.9
OSU leaf	37.8	35.1	39.1*	44.2	32.6	29.3	31.4	35.7	20.9	21.6	25.0	39.9*	45.5 ± 3.5*	45.9*	**46.2***
Swedish leaf	40.6	48.1	35.6	38.6	34.3	33.5	36.9	38.1	14.4	25.3	32.7	32.9	**59.8 ± 2.1***	56.4*	38.3*
50words	42.0	37.2	39.5	40.8	34.2	36.1	32.3	37.0	32.7	30.6	32.5	29.3	37.2 ± 2.7	**46.9***	36.1
Trace	48.5	**63.4**	40.2	50.9	39.6	42.3	41.8	46.3	33.3	43.7	47.5	49.5	57.2 ± 2.3*	58.0*	60.6*
Two patterns	32.2	**56.3**	29.8	33.1	26.1	27.6	29.2	32.0	24.7	27.4	21.9	21.3	37.7 ± 2.5*	38.3*	29.0
Wafer	62.5	47.5	53.2	63.9	55.3	59.8	54.8	61.8	71.9*	54.4*	59.1	38.4	71.7 ± 2.4*	69.7*	**73.3***
Face (four)	66.9	70.7	62.7*	69.1	49.3	55.3	60.3	67.2	17.3	17.2	35.1	49.1	**78.9 ± 3.0***	75.8*	51.7
Lightning-2	61.1	62.1	62.0*	57.7	52.1	53.5	54.3	56.1	62.4*	56.7	45.7	43.7	**77.9 ± 1.8***	75.1*	71.9*
Lightning-7	48.4	50.5	39.4	51.2	40.7	42.6	48.3	49.2	44.0	33.7	45.7*	33.8	**58.7 ± 3.4***	56.7*	53.9*
ECG	69.8	62.8	59.4	70.3	58.8	61.0	61.4	62.2	49.0*	50.6*	47.6*	60.9*	69.0 ± 1.7*	73.5*	**74.9***
Adiac	38.4	**39.6**	30.2	38.9	30.9	29.3	32.1	31.6	32.0	12.6	33.1	25.3	36.8 ± 2.5	33.6	34.2
Yoga	51.7	56.3	44.2	48.5	59.1	52.6	46.8	51.0	46.1	66.8*	41.6	43.3	62.6 ± 2.0*	64.3*	**67.5**

HC, *hierarchical clustering;* HMM, *hidden Markov model;* WCE, *weighted clustering ensemble.*
[a] *Notation of correct cluster number determined.*

single representation, in turn, 40 partitions are totally obtained from four different representations. In our WCE, we use the same procedure described earlier for K-means to produce partitions, where K is randomly chosen from $K^* - 2 \leq K \leq K^* + 2 (K > 0)$. We conduct 10 trials and report the average and standard deviations of classification accuracy rates.

Table 7.2 shows the performance of four clustering ensemble approaches using the same notation as in Table 7.1. It is observed that the SDP-CE, HBGF, and CE algorithms win on five, two, and one data sets, respectively, while the proposed WCE algorithm achieves best results for the remaining eight data sets. Closer observation indicates that for four of the eight data sets where other algorithms win, WCE algorithm achieves the second best results. SDP-CE suffers from the highest computational burden, while WCE incurs higher computational costs than the CE and the HBGF. Considering its capability of model selection and a trade-off between performance and computational efficiency, we believe that our WCE algorithm is especially suitable for temporal data clustering with different representations.

In order to make comparison between three proposed ensemble approaches including *HMM-based meta-clustering ensemble*, *Iteratively constructed clustering ensemble with a hybrid sampling, WCE with multiple representations* presented in this book, we further summarize the experimental results obtained by these three approaches on the time series benchmark. We employ K-means as base learner, where the K value as cluster number is selected from a preset range $K^* - 2 \leq K \leq K^* + 2 (K > 0)$ for both of *HMM-based meta-clustering ensemble* and *WCE with multiple representations*, and the correct cluster number K^* is pre-defined for *Iteratively*

Table 7.2 Classification Accuracy (%)[a] of Clustering Ensembles on Time Series Benchmarks (Yang and Chen, 2011a)

Data Set	CE	HBGF	SDP-CE	WCE
Syn control	68.8 + 2.1	74.8 ± 2.2	82.1 + 1.9	**86.1 ± 2.5***
Gun-point	51.8 + 1.4	53.8 ± 2.0	50.0 ± 0.9	**54.1 ± 1.8***
CBF	53.8 + 2.4	66.0 ± 1.9	**66.3 ± 2.1**	63.9 ± 2.1*
Face (all)	35.1 + 1.9	44.8 ± 2.5	50.5 ± 1.2	**51.9 ± 1.4***
OSU leaf	35.2 + 1.7	**48.1 ± 2.9**	46.9 ± 2.1	45.5 ± 3.5*
Swedish leaf	41.2 + 0.8	52.8 ± 2.3	**62.6 ± 1.8**	59.8 ± 2.1*
50words	**39.6 + 1.6**	39.1 ± 2.1	38.9 ± 1.9	37.2 ± 2.7
Trace	50.5 + 2.0	**45.6 ± 2.2**	55.1 ± 1.9	**57.2 ± 2.3***
Two patterns	33.1 + 1.8	**33.0 ± 1.9**	36.9 ± 2.3	**37.7 ± 2.5***
Wafer	62.1 + 1.9	**72.8 ± 2.6**	70.0 ± 2.4	71.7 ± 24*
Face (four)	**65.2 + 2.1**	**72.1 ± 3.1**	71.8 ± 3.5	**78.9 ± 3.0***
Lightning-2	**60.1 + 1.3**	**59.3 ± 2.1**	66.2 ± 1.6	**77.9 ± 1.8***
Lightning-7	**53.1 + 2.1**	**55.6 ± 3.0**	57.9 ± 2.4	**58.7 ± 3.4***
ECG	65.2 + 1.6	68.7 ± 2.0	**69.2 ± 1.7**	69.0 ± 1.7*
Adiac	36.2 + 2.3	41.4 ± 2.5	**45.9 ± 1.9**	36.8 ± 2.5
Yoga	50.6 + 2.3	60.0 ± 2.2	**68.2 ± 2.2**	62.6 ± 2.0 *

CE, cluster ensemble*; HBGF,* hybrid bipartite graph formulation*; SDP-CE,* semi-definite programming –based clustering ensemble*; WCE,* weighted clustering ensemble.

[a] *Notation of correct cluster number determined.*

constructed clustering ensemble with a hybrid sampling during their initial clustering analysis due to their own model-selection abilities. Following exactly same experimental setup described previously, we run this simulation 10 times with best parameter setup, and the best results obtained by three proposed ensemble approaches are reported in Table 7.3. As shown in this table, *WCE with multiple representations* has the best performance on 9 out of 16 data sets. *HMM-based meta-clustering ensemble* achieves best results for five data sets, *Iteratively constructed clustering ensemble with a hybrid sampling* win on two. Given the fact that both of *HMM-based meta-clustering ensemble* and *WCE with multiple representations* are capable for finding a cluster number in a given data set by using DSPA consensus function, we also report their model selection performance with the former notation (*). As a result, *WCE with multiple representations* once again achieves the best results that is able to find the correct cluster number on 14 out of 16 data sets, and *HMM-based meta clustering ensemble* also manages to find the correct cluster number for 11 data sets.

To conclude, in comparison with classical temporal data clustering, "state-of-the-art" clustering ensemble algorithms, and even two former proposed ensemble approaches, the favorable results achieved on the benchmark time series collection

Table 7.3 Classification Accuracy (%)[a] of Our Proposed Clustering Ensemble Models on Time Series Benchmarks

Data Set	HMM Hybrid Meta-Ensemble Clustering	Iteratively Constructed Clustering Ensemble With Hybrid Sampling	Weighted Clustering Ensemble With Multiple Representations
Syn control	73.2*	74.3	**88.6***
Gun-point	**65.2***	56.7	55.9*
CBF	64.3*	64.8	**66.0***
Face (all)	31.4	40.4	**53.3***
OSU leaf	38.0	41.8	**49.0***
Swedish leaf	42.5*	44.6	**61.9***
50words	**46.2***	40.9	39.9
Trace	**63.9***	53.5	59.5*
Two patterns	**50.6***	33.2	40.2*
Wafer	53.4	62.5	**74.1***
Face (four)	58.8	64.0	**81.9***
Lightning-2	67.6*	68.6	**79.7***
Lightning-7	50.0*	50.0	**62.1***
ECG	65.8	**72.0**	70.7*
Adiac	43.2*	**44.4**	39.3
Yoga	**63.8***	51.9	63.6*

HMM, hidden Markov model.
[a] *Notation of correct cluster number determined.*

clearly suggest the strengths and ease of use that our *WCE with multiple representations* provides in temporal data clustering task.

7.3.2 MOTION TRAJECTORY

To explore potential applications, the proposed WCE is also applied on the CAVIAR database for trajectory clustering analysis, previously used in Chapters 5 and 6 for evaluating both of *HMM-based meta-clustering ensemble* and *Iteratively constructed clustering ensemble model*. Note that the preprocessing described in Section 6.3.3 is also required to normalize the trajectories into uniform length before feature extraction is applied to the trajectory dataset.

Given no prior knowledge of the "right" number of clusters for this database, K-means algorithm as the base learner is run 10 times by randomly choosing a K value from an interval between 5 and 25. 10 partitions are then generated with different initial conditions on each of four feature representations. In turn, total of 40 partitions are then fed to the proposed WCE to achieve the final partition shown in Fig. 7.4. Without the ground truth, we must apply human visual observation for evaluating the results, as suggested by Khalid and Naftel (2005). In the language of common human visual experience, the behavior of pedestrians across the shopping mall can be roughly divided into five categories of motion behaviors: "move up," "move down," "stop," "move left," and "move right" (CAVIAR, 2002). Using clustering analysis, these behavioral trajectories are grouped together along a motion direction and used to infer and name different activities at a semantic level, for example, "enter the store," "exit from the store," "pass in front," and "stop to watch."

In Fig. 7.4, 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 grouped accurately in the cluster shown in Fig. 7.4E. Those corresponding to "moving from left-to-right" or "right-to-left" are properly grouped into two separate clusters, shown in Fig. 7.4C and F. The trajectories "move up" and "move down" are grouped very effectively into the two clusters shown in Fig. 7.4J and K. Fig. 7.4A,D,G–I,N,O also indicate that trajectories corresponding to the activities "enter the store" and "exit the store" are properly grouped together via multiple clusters in light of various starting positions, locations, motion direction, and so on. Finally, Fig. 7.4L and M illustrate two clusters roughly corresponding to the activity "pass in front."

As described in the Chapters 5 and 6, we have also examined former proposed approaches, which are *HMM-based meta-clustering ensemble* and *Iteratively constructed clustering ensemble with a hybrid sampling*, on the CAVIAR database. Based on experimental results obtained by these proposed ensemble models, it is obvious that *HMM-based meta clustering ensemble* is quite different from other proposed ensemble models due to its insensitive to distinguish the trajectories following a similar motion path with opposite directions. For example, the trajectories corresponding to "move left" and "move right" trajectories shown in

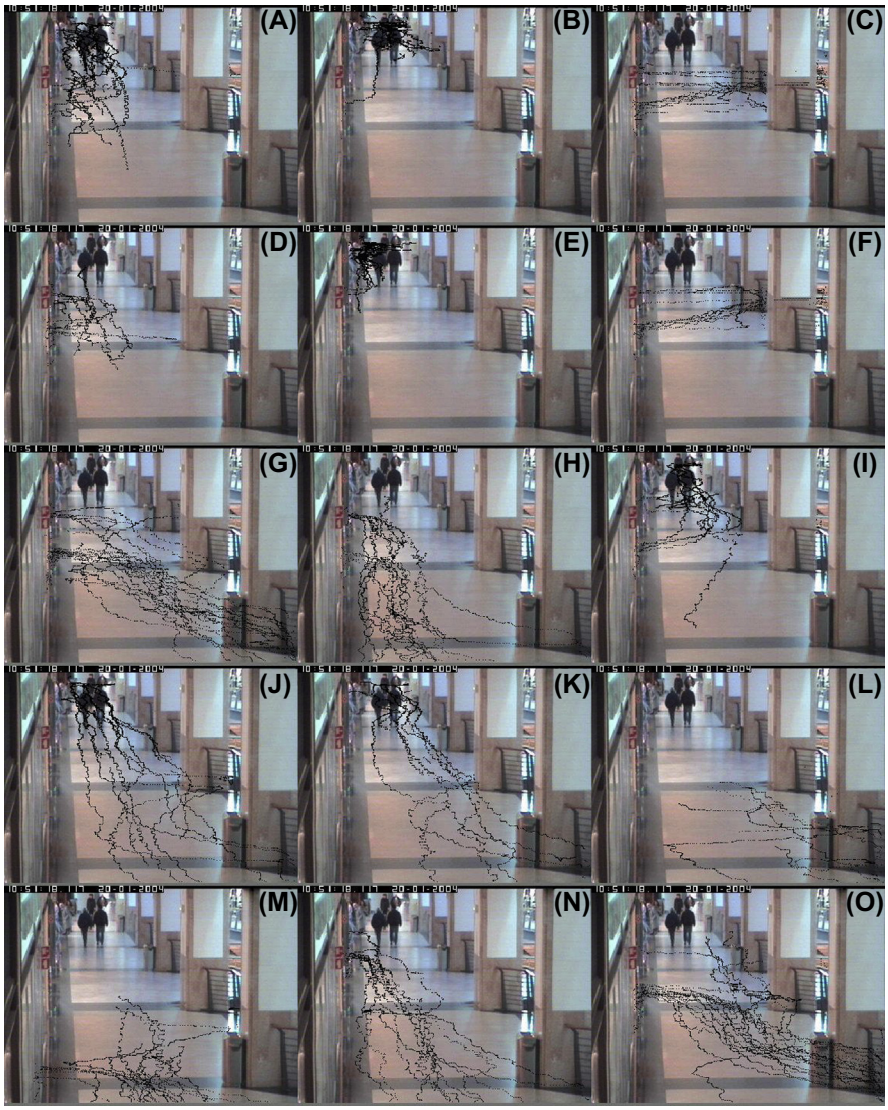

FIGURE 7.4

The final partition on the CAVIAR database by WCE with different representations; plots in (A)—(O) correspond to 15 clusters of moving trajectories (Yang and Chen, 2011a).

Fig. 6.6D and J obtained from *Iteratively constructed clustering ensemble with a hybrid sampling* or Fig. 7.4C and F obtained from *WCE with multiple representations* are grouped into a single cluster of "move vertically" shown in Fig. 5.8C obtained from *HMM-based meta-clustering ensemble*, while "move up" and "move down" trajectories shown in Fig. 6.6C and G obtained from *Iteratively constructed clustering ensemble with a hybrid sampling* or Fig. 7.4J,K and N obtained from *WCE with multiple representations* are either wholly or partially grouped into a cluster of "move vertically" shown in Fig. 5.8A. However, *HMM-based meta-clustering ensemble* is quite good at properly grouping the trajectories following the similar motion paths. For example, the trajectories corresponding to "enter store" shown in Fig. 5.8E obtained from *HMM-based meta-clustering ensemble* are separated into different clusters either shown in Fig. 6.6C and N obtained from *Iteratively constructed clustering ensemble with a hybrid sampling* or Fig. 7.4H and N obtained from *WCE with multiple representations*. On the other hand, we could identify that most of the clusters obtained from both of *Iteratively constructed clustering ensemble with a hybrid sampling* and *WCE with multiple representations*, respectively, are quite similar, and all trajectories are properly grouped into different clusters based on the similar motion behaviors, and these consistent and quality clustering results further justify the feasibility of fundamental concepts from different perspective. However, both algorithms still yield slightly different clustering results due to their own characteristics inherent in the different approaches. For *Iteratively constructed clustering ensemble with a hybrid sampling*, it significantly reduces the sensitivity to the noise data by using a subsampling schema. As illustrated in Fig. 6.6, a group of abnormal motion trajectories (noise data) shown in Fig. 6.6M are well separated from their closed "meaningful" motion trajectories shown in Fig. 6.6F. However, it improperly merges two groups of trajectories shown in Fig. 7.4H and J based on two different motion behaviors into single cluster shown in Fig. 6.6C due to similarity with the limited information in spatial domain. In contrast, *WCE with different representations*, which is presented in this chapter, is able to explore the hidden information exited in temporal data from different domains. As a result, the trajectories shown in Fig. 6.6C are well separated into two clusters shown in Fig. 7.4H and J, which is able to explore the detailed motion behaviors on the target data set. However, it fails to identify the abnormal motion trajectories in the cluster shown in Fig. 7.4O due to sensitive to the noise data. As a result, we believe that all proposed ensemble models behave differently with their own strength and weakness.

Moreover, we also record the execution time (117.98s) of *WCE with multiple representations* on CAVIAR database, which has been also applied for both of former proposed ensemble models presented in Chapters 5 and 6. By summering the implementation efficiency of three proposed ensemble models, where K is the number of clusters, N is the size of data set/number of objects, T is the number of ensemble members/input partitions, Table 7.4 shows that *Iteratively constructed clustering ensemble* has less computational complexity in comparison with others but it requires several key user−input parameters such as cluster number K. Although both of *HMM-*

Table 7.4 Computational Complexity of Our Proposed Clustering Ensemble Models on CAVIAR Database

Our Proposed Clustering Ensemble Models	Computational Complexity	Execution Time on the CAVIAR Database (s)
HMM based hybrid nieta-clustering ensemble	$O(KN^2T) - CSPA$ $O(KNT) - HGPA$ $O(K^2NT^2) - MCLA$ $O(KN^2T) - DSPA$	*4623.44*
Iteratively constructed clustering ensemble with a hybrid sampling	*O(K3)*	12.27
Weighted clustering ensemble with multiple representations	*O(KN2T)*	117.98

HMM, *hidden Markov model.*

based meta-clustering ensemble and *WCE with multiple representations* demand higher computational resource, it requires minimum user-input parameters with automatic model-selection ability vice versa. Therefore, we believe that all proposed ensemble model behave differently with their own strength and weakness in terms of performance and computational cost.

The CAVIAR database was also used in the study by Khalid and Naftel (2005) where the self-organizing map with the single Discrete Fourier Transform representation was used for clustering analysis. In their simulation, the number of clusters was determined manually, and all trajectories were simply grouped into nine clusters. Although most clusters achieved in their simulation are consistent with ours, their method failed to separate a number of trajectories corresponding to different activities by simply putting them together into a cluster called "abnormal behaviors" instead. In contrast, ours properly groups them into several clusters shown in Fig. 7.4A,B,E and I. If the clustering analysis results are employed for modeling events or activities, their merged cluster inevitably fails to provide any useful information for a higher level analysis.

Clearly implying the benefits of the joint use of different representations, Fig. 7.5 illustrates several meaningless clusters of trajectories, judged by visual inspection, yielded by the same proposed approach but on single representations, respectively. Fig. 7.5A and B show that the sole use of a Polynomial Curve Fitting representation will create improper grouping of some trajectories of line structures along the x- and y-axes. The x and y components of these trajectories perpendicular to each other have a considerable coefficient value on their linear basis but only a tiny coefficient value on any higher order basis. Consequently, this leads to a short distance between such trajectories in the PCF representation space, which is responsible for improper grouping. Fig. 7.5C and D illustrate a limitation of the DFT representation, that is,

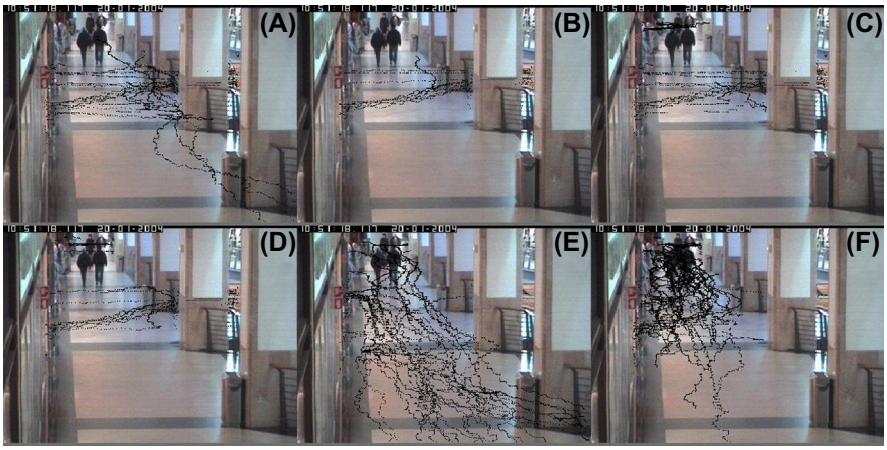

FIGURE 7.5

Meaningless clusters of trajectories on the CAVIAR database with a single representation.

trajectories with the same orientation but with different starting points are improperly grouped together since the DFT representation is in the frequency domain and therefore independent of spatial locations. Although the Piecewise Local Statistics and Piecewise Discrete Wavelet Transform representations highlight local features, global characteristics of trajectories could be neglected. The cluster based on the PLS representation shown in Fig. 7.5E improperly groups trajectories belonging to two clusters in Fig. 7.4K and N. Likewise, Fig. 7.5F shows an improper grouping based on the PDWT representation that merges three clusters in Fig. 7.4A,D and I. All the above results suggest that the joint use of different representations is capable of overcoming limitations of individual representations.

Two additional simulations take our evaluation further. The first tests the generalized performance on noisy data produced by adding different amount of Gaussian noise $N(0, \sigma)$ to the range of coordinates of the moving trajectories. The second simulates a scenario in which a tracked moving object is obstructed by other objects or by the background, leading inevitably to missing data in a trajectory. Fifty independent trials have been performed in our simulations, in the interests of securing a robust result.

Table 7.5 presents results of classifying noisy trajectories with the final partition shown in Fig. 7.4 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 highly depends on the quality of clustering analysis. It is evident from Table 7.1 that the performance is satisfactory in contrast to those of the clustering ensemble on a single representation especially as a substantial amount of noise is added, which again demonstrates the synergy between different representations.

Table 7.5 Performance on the CAVIAR Corrupted With Noise (Yang and Chen, 2011a)

Representation	WCE			
	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$
PCF	87.6 ± 2.6	81.7 ± 3.3	78.1 ± 3.7	72.9 ± 4.2
DFT	$92.0 + 3.1$	88.6 ± 3.5	85.9 ± 3.9	79.1 ± 3.6
PLS	95.2 ± 1.8	90.0 ± 2.5	89.5 ± 3.1	84.1 ± 4.0
PDWT	94.3 ± 2.8	90.1 ± 3.2	86.6 ± 2.6	84.2 ± 3.6
Multiple	**97.2 ± 1.7**	**93.1 ± 2.3**	**91.9 ± 1.9**	**85.9 ± 2.3**

WCE, *Weighted Clustering Ensemble.*

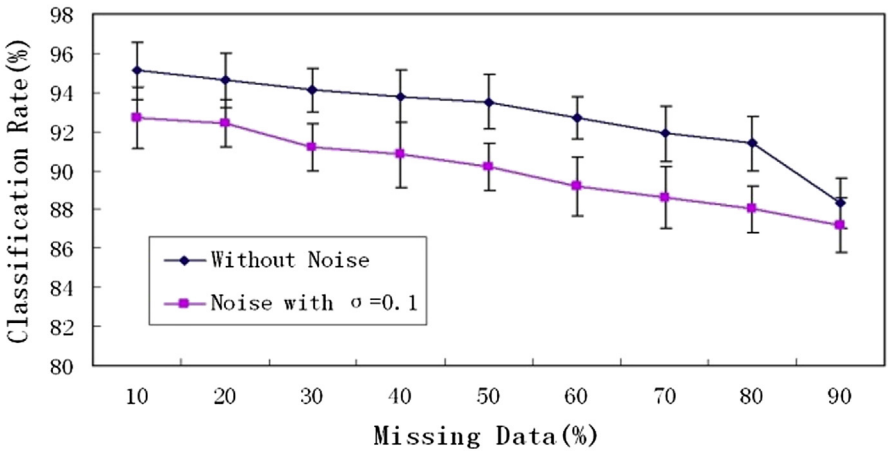

FIGURE 7.6

Classification accuracy of our WCE on the CAVIAR database and its noisy version in simulated occlusion situations.

To simulate missing data trajectories, we remove five identical segments of trajectory at random locations, while missing segments of various lengths are used for testing. We classify a trajectory of missing data using its observed data only. The same decision-making rule mentioned previously is also used. This simulation is conducted on all corrupted trajectories and their corresponding noisy versions by adding the Gaussian noise N(0, 0.1). Fig. 7.6 shows performance evolution in the presence of missing data measured by a percentage of the trajectory length. Our approach evidently performs well in simulated occlusion situations.

7.3.3 TIME-SERIES DATA STREAM

In contrast to temporal data collected prior to processing, a data stream consisting of variables comes from the continuous data flow of a given source, for example, sensor networks (PDMC, 2004), with high speeds generating examples over time. Hence, to perform clustering on a temporal data stream is to find groups of variables that behave similarly over time. Here, traditional temporal clustering algorithms meet with fresh challenges—an algorithm for temporal data stream clustering needs to effectively deal with each example in constant time and memory (Rodrigues et al., 2008).

Recent research with time-series data stream clustering algorithms, for example, the Online Divisive-Agglomerative Clustering (ODAC) algorithm (Rodrigues et al., 2008), has resulted in most such algorithms being developed to work on a stream fragment, thereby fulfilling clustering in constant time and memory. However, to fully exploit potential yet hidden information and to demonstrate the proposed approach in temporal data stream clustering, we also use the dynamic properties of a temporal data stream itself. It is known that dynamic properties of time series, *x(t)*, can be well described by its derivatives of different orders, $x^{(n)}(t), n = 1, 2, \ldots$ As a result, we would treat the derivatives of a stream fragment itself as different representations and then use them for the initial clustering analysis in our approach. Given that the estimate of the *n*th derivative requires only $n + 1$ successive points, a slightly larger constant memory is used in our approach that is, for the nth-order derivative of time series, the size of our memory is n points larger than the size of the memory used by the ODAC algorithm (Rodrigues et al., 2008).

Our simulations use the PDMC data set, collected from streaming sensor data of approximately 10,000 h of time-series data streams, containing several variables such as user ID, session ID, session time, two characteristics, annotation, gender, and nine sensors (PDMC, 2004). Following the same experimental setting as (Rodrigues et al., 2008), we use their ODAC algorithm for initial clustering analysis on three representations—time series, *x(t)*, and its first- and second-order derivatives, $x^{(1)}(t)$, and $x^{(2)}(t)$. We adopt the same criteria, MHΓ and DVI, used by (Rodrigues et al., 2008) to evaluate performance. This allows us a straightforward comparison between the proposed approach and this state-of-the-art technique,

Table 7.6 Results of the ODAC Algorithm Versus Our WCE (Yang and Chen, 2011a)

	ODAC			WCE		
Data Set	***K***	**MHΓ**	***DVI***	***K***	**MHΓ**	***DVI***
User ID = 6	3	0.377	0.891	5	0.418	1.931
User ID = 25	3	0.191	1.026	4	0.420	1.701

DVI, *Dunn's Validity Index; MHΓ, modified Huber's Γ index;* ODAC, *Online Divisive-Agglomerative Clustering;* WCE, *Weighted Clustering Ensemble.*

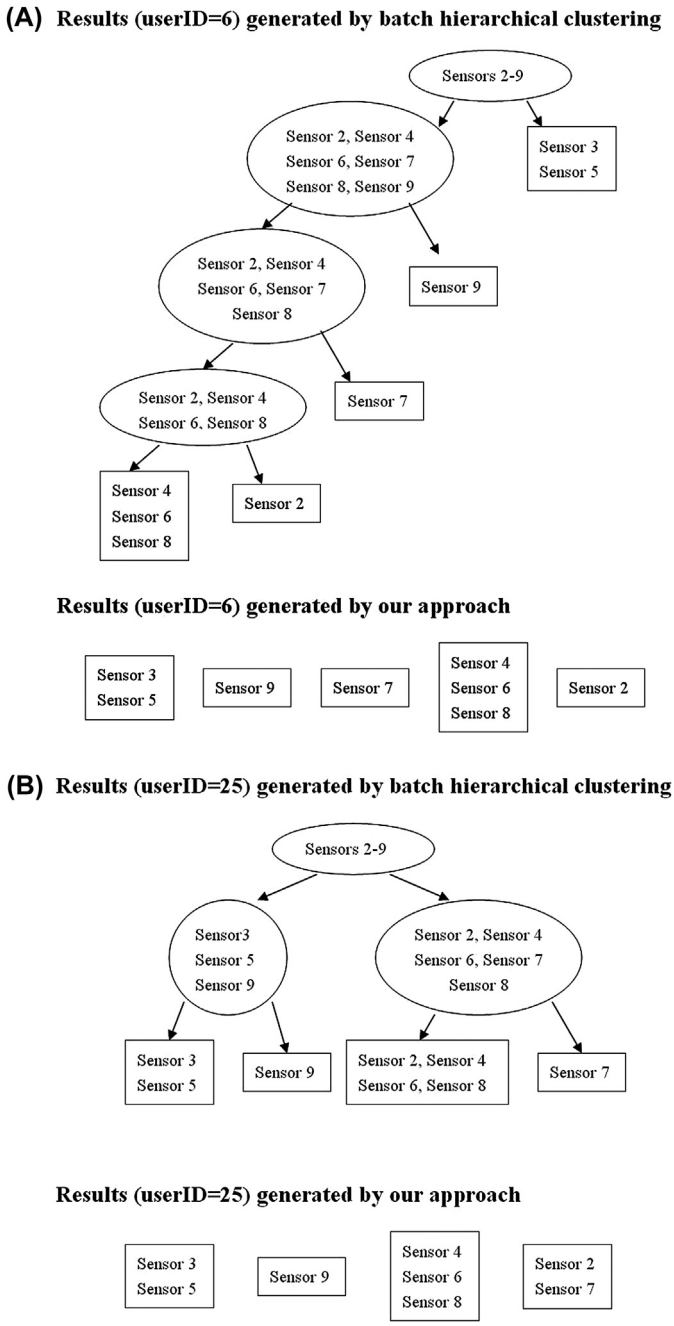

FIGURE 7.7

Results of the batch hierarchical clustering algorithm versus our WCE on two data stream collections (Yang and Chen, 2011a). (A) User ID = 6; (B) user ID = 25.

and demonstrating the performance gain by the proposed ensemble approach via the exploitation of hidden yet potential information.

Table 7.6 displays the comparative results obtained by the proposed approach and those reported by (Rodrigues et al., 2008) on two collections: user ID = 6 of 80,182 observations and user ID = 25 of 141,251 observations. The task involved finding the correct number of clusters on eight sensors from 2 to 9. The best results on two collections are reported in the study Rodrigues et al. (2008). Working on the time-series data streams, their ODAC algorithm only finds three sensor clusters for each user. Their clustering quality is evaluated by the MHΓ and the DVI. Table 7.6 shows our approach to achieve much higher MHΓ and DVI values while finding considerably different cluster structures, there are five clusters found for user ID = 6 and four clusters found for user ID = 25.

Despite outperforming the ODAC algorithm in terms of clustering quality criteria, we still would not conclude that our approach is much better than the ODAC algorithm as those criteria evaluate clustering quality from a specific perspective only. Although there is no ground truth available we believe that a whole stream of all observations should contain precise information on its intrinsic structure and be able to verify our results by comparing it with the result obtained by a batch hierarchical clustering (BHC) algorithm (Johnson, 1967). Fig. 7.7 shows the resultant similarities and contrasts between the BHC and the proposed approach. On user ID = 6, ours is completely consistent with the BHC. On user ID = 25, they are also identical, except that our group sensors 2 and 7 in a cluster while the BHC separates them and merges sensor 2 with a larger cluster. Clearly, the structures uncovered by the ODAC are quite distinct from those yielded by the BHC algorithm and although our partition on user ID = 25 is not consistent with that of the BHC, the overall results on two streams are considerably better than those of the ODAC.

In summary, the previously mentioned simulation demonstrates how, by exploiting the additional information, our approach leads to a substantial improvement in an emerging real-world application of temporal data clustering. However, a clustering ensemble with different representations naturally has a higher computational burden, requiring a slightly larger memory to perform the initial clustering analysis. Nevertheless, it is apparent from Fig. 7.3 that initial clustering analysis on different representations is a completely independent process, as is the generation of candidate consensus partitions. Thus, we hold that the highly advanced computing technology of today, for example, distributed computing, comprises adaptive capabilities which can help to overcome these weaknesses in a real-world application.

7.4 SUMMARY

If information loss during representation extraction is a fundamental weakness of representation-based temporal data clustering, the use of different temporal data representations in this proposed approach plays an important role in diminishing it.

Conceptually, temporal data representations acquired from different domains and on different scales tend to be complementary: temporal versus frequency, local versus global, and fine versus coarse. In the reported simulations, we simply use four complementary temporal data representations to demonstrate a method for cutting information loss. Our focus on representation-based clustering, however, has been somewhat to the detriment of representation-related issues per se, which includes the development of novel temporal data representations and how a synergy of representations can produce appropriate partitions for the clustering ensemble. We anticipate improvements to the proposed approach once these representation-related problems are tackled effectively.

The cost function derived in Eq. (7.9) suggests that the performance of a clustering ensemble depends both on the quality of input partitions and a sound clustering ensemble scheme. Initial clustering analysis is vital to performance. According to the first term of Eq. (7.9), good performance demands that the variance of input partitions is small and the optimal "mean" is close to the intrinsic "mean", that is, the ground-truth partition. Hence, clustering algorithms must first be appropriately matched to the nature of a given problem to produce such input partitions, aside from the use of different representations. Domain knowledge can be integrated via appropriate clustering algorithms during initial clustering analysis. Moreover, structural information underlying any given data set can be exploited, for example, by manifold clustering (Souvenir and Pless, 2005), to produce input partitions which reflect its intrinsic structure. As long as an initial clustering analysis returns input partitions which encode domain knowledge and characterize the intrinsic structural information, the "abstract" similarity (i.e., whether or not two entities are in the same cluster) used in our WCE will inherit them during the combination of input partitions. Additionally, the weighting scheme allows the integration of other useful criteria and domain knowledge, all of which suggests new ways to continue improving this proposed ensemble approach.

As shown, a clustering ensemble algorithm enables the flexible and effective use of different representations. Previous work (Chen et al., 1997; Chen, 1998; Chen and Chi, 1998; Chen, 2005a,b; Wang and Chen, 2007) demonstrates that a single learning model working on a composite representation formed by lumping different representations together is often inferior to an ensemble of multiple learning models on different representations in supervised and semisupervised learning. Moreover, our earlier empirical studies (Yang and Chen, 2006; Yang and Chen, 2007) and others not reported here, confirm our previous findings related to temporal data clustering. Thus, we believe that the proposed approach is both more effective and more efficient than a single learning model applied on the composite representation, which results in much higher dimension.

As a generic technique, our WCE algorithm is applicable to combinations of any input partitions in its own right, regardless of temporal data clustering. Therefore, we would link our algorithm to the most significant works and highlight the essential differences between them.

The CE algorithm (Strehl and Ghosh, 2003) presents three heuristic consensus functions for combining multiple partitions, which are applied to produce three candidate consensus partitions respectively. Then, the NMI criterion is employed to find a final partition by selecting, from the candidate partitions, the one of maximum NMI value. Although there is a two-stage reconciliation process in both their algorithm and ours, other factors clearly distinguish them from each other. First, ours uses only a uniform weighted consensus function, allowing various clustering validation criteria for weight generation. These criteria are used to produce multiple candidate consensus partitions (in this chapter, we use only three criteria). Then, we employ an agreement function to generate a final partition by combining all the candidate consensus partitions.

Our consensus and agreement functions are developed subject to the evidence accumulation framework (Fred and Jain, 2005). Unlike the original algorithm (Fred and Jain, 2005) where all input partitions are treated equally, we use the evidence accumulated in a selective way. When Eq. (7.10) is applied to the original algorithm (Fred and Jain, 2005) for analysis, it can be viewed as a special case of our algorithm as $w_m = 1/M$. Thus, its cost defined in Eq. (7.9) is simply a constant, independent of any combination. In other words, the algorithm (Fred and Jain, 2005) does not exploit useful information on the relationships between input partitions and works well only if all input partitions are of similar distance to the ground-truth partition in the partition space. Thus, we believe this analysis to expose the fundamental weakness of a clustering ensemble algorithm which treats all input partitions equally during the combination process.

Other WCE algorithms (Al-Razgan and Domeniconi, 2006; Cheng et al., 2008; Li and Ding, 2008) can, in general, be divided into two categories in terms of the weighting scheme: cluster or partition weighting. A cluster weighting scheme (Al-Razgan and Domeniconi, 2006; Cheng et al., 2008) associates the clusters in a partition with a weighting vector and embeds it in the subspace spanned by an adaptive combination of feature dimensions. A partition weighting scheme (Li and Ding, 2008) assigns a weight vector to the partitions to be combined. Our algorithm belongs to the latter category but adopts a different principle from that used by (Li and Ding, 2008) to generate weights.

The algorithm proposed by Li and Ding (2008) comes up with an objective function which encodes the overall weighted distance between all input partitions to be combined and the consensus partition to be found. Thus, an optimization problem has to be solved to find the optimal consensus partition. However, according to our analysis in Section 7.2.5, the optimal "mean" in terms of their objective function may be inconsistent with the optimal "mean" by minimizing the cost function defined in Eq. (7.8). Here, then, the quality of their consensus partition is not guaranteed. Although their algorithm is developed under the non-negative matrix factorization framework (Li and Ding, 2008), the iterative procedure for an optimal solution incurs a high computational complexity of O(n3). Our algorithm, by contrast, directly calculates weights with clustering validation criteria, allowing for the use of multiple criteria to measure the contribution of partitions, in turn

leading to much faster computation. Indeed, the efficiency issue remains critical for several real-world applications, such as temporal data stream clustering. In our ongoing work, we are developing a weighting scheme to maximize the synergy between cluster and partition-based weighting.

To conclude, this chapter has presented a new temporal data clustering approach via a WCE on different representations and further proposed useful measures for understanding clustering ensemble algorithms based on formal clustering ensemble analysis (Topchy et al., 2004). The simulations have shown favorable results, in terms of clustering quality and model selection, for a variety of temporal data clustering tasks. Furthermore, as a generic framework, our approach allows for the direct incorporation of other validation criteria (Halkidi et al., 2001) to generate new weighting schemes, as long as the intrinsic structure underlying a data set would be better reflected. Finally, our approach does not involve a tedious parameter tuning process or bring with it a high level of computational complexity. It is a promising yet easy-to-use technique, designed for real-world applications.