Chapter 13

COMBINING COMPETITIVE LEARNING NETWORKS OF VARIOUS REPRESENTATIONS FOR SEQUENTIAL DATA CLUSTERING

Yun Yang and Ke Chen
School of Informatics
The University of Manchester
Manchester M60 1QD, U.K.
Yun.Yang@postgrad.manchester.ac.uk, Ke.Chen@manchester.ac.uk

Abstract Sequential data clustering provides useful techniques for condensing and summarizing information conveyed in sequential data, which is demanded in various fields ranging from time series analysis to video clip understanding. In this chapter, we propose a novel approach to sequential data clustering by combining multiple competitive learning networks incorporated by various representations of sequential data and thus the clustering will be performed in the feature space. In our approach, competitive learning networks of a rival-penalized learning mechanism are employed for clustering analyses based on different sequential data representations individually while an optimal selection function is applied to find out a final consensus partition from multiple partition candidates yielded by applying alternative consensus functions to results of competitive learning on various representations. Thanks to its capability of the rival penalized learning rules in automatic model selection and the synergy of diverse partitions on various representations resulting from diversified initialization and stopping conditions, our ensemble learning approach yields favorite results especially in model selection, i.e. no assumption on the number of clusters underlying a given data set is needed prior to clustering analysis, which has been demonstrated in synthetic time series and motion trajectory clustering analysis tasks.

Key words: Sequential data clustering, unsupervised ensemble learning, rival penalized competitive learning, local and global representations, model selection, motion trajectory analysis, time series classification

1. INTRODUCTION

Sequential data are ubiquitous in the real world and there are many application areas ranging from multimedia information processing to financial data analysis. Unlike static data, there is a high amount of dependency among sequential data and the proper treatment of data dependency or correlation becomes critical in sequential data processing.

Clustering analysis provides an effective way to condensing and summarizing information conveyed in data, which is demanded by a number of application areas for organizing or discovering structures in data. The objective of clustering analysis is to partition a set of unlabeled objects into groups or clusters where all the objects grouped in the same cluster should be coherent or homogeneous. There are two core problems in clustering analysis; i.e., model selection and proper grouping. The former is seeking a solution that estimates the intrinsic number of clusters underlying a data set, while the latter demands a rule to group coherent objects together to form a cluster. From the perspective of machine learning, clustering analysis is an extremely difficult unsupervised learning task since it is inherently an ill-posed problem and its solution often violates some common assumptions [1]. There have been many studies in clustering analysis, which leads to various clustering algorithms categorized as either hierarchical or non-hierarchical algorithms [2]. However, the recent empirical studies in sequential data analysis reveal that most of the existing clustering algorithms do not work well for sequential data due to their special structure and data dependency [3], which presents a big challenge in clustering sequential data of a high dimensionality, very high feature correlation and a substantial amount of noise.

Competitive learning has been studied in the neural network community and turned out to be a useful tool for clustering analysis [4]. Among many competitive learning algorithms, however, few of them are capable of model selection and the number of clusters needs to be specified prior to clustering analysis. A *rival penalized competitive learning* (RPCL) algorithm was proposed to tackle both model selection and grouping problems under the framework of competitive learning [5]. More recently, one variant of RPCL, named *rival penalization controlled competitive learning* (RPCCL), has been proposed to improve its performance by using a data-driven de-learning rate [6]. Although RPCL and its variants have been successfully applied in static data clustering analysis [5],[6], our empirical studies indicate that the direct use of a RPCL-style algorithm in sequential data clustering tasks often fails to yield the satisfactory performance.

It is well known that the direct sequential data clustering often suffers from the curse-of-dimensionality and difficulties in capturing the long-term

temporal dependency. Feature extraction is a process that distills the salient features from data and widely applied in pattern recognition. Unlike the direct use of sequential data, a feature extraction method often results in a parsimonious yet more effective representation of sequential data so that sequential data analysis can be fulfilled in the representation or feature space of low dimensionality. It has been proved that clustering analysis on the representation space often outperforms that on the original data space [2],[3]. A number of feature extraction methods for sequential data have been come up with from different perspectives [3], and they are roughly divided into two categories: local and global representations. A local representation tends to encode the local features precisely but is very difficult to characterize the global landscape of sequential data while a global representation models the landscape well by scarifying those local fine details in sequential data. To our knowledge, there is no universal representation that perfectly characterizes miscellaneous sequential data. As suggested by our recent work in non-verbal speech information processing [7], it is more likely that different representations need to be employed simultaneously to characterize complex sequential data entirely. Our earlier work in the real world sequential data classification, e.g. speaker recognition [8],[9], showed that the simultaneous use of different representations yields the significantly better performance than the use of an individual representation.

In this chapter, we propose a novel approach to sequential data clustering by an ensemble of RPCCL networks with different representations. We anticipate that the use of a RPCCL network, to a great extent, tackles the model selection problem without the use of prior information on the data domain while the ensemble of RPCCL networks tends to incorporate different representations to reach the synergy for clustering analysis. Recent researches in clustering ensembles [10],[11] provide feasible techniques to enable us to construct an ensemble RPCCL networks trained on different representations. We have applied our approach into time series and motion trajectory clustering tasks. Simulation results indicate that our approach yields the favorite performance in sequential data clustering. Our study reveals that the use of clustering ensemble techniques [10],[11] further improves the model selection performance, in particular, as individual RPCCL networks fail to estimate "right" cluster numbers.

The rest of this chapter is organized as follows. Sect. 2 describes three sequential data representations used in our simulations. Sec. 3 presents an ensemble competitive learning approach for sequential data clustering with the use of different representations and overviews the rival penalized competitive network. Sect. 4 reports simulation results in the time series and the motion trajectory clustering tasks. The last section draws conclusions.

2. SEQUENTIAL DATA REPRESENTATIONS

There have been a number of sequential data representations used in different application areas. In general, such representations can be classified into two categories: *global* and *piecewise* representations. A global representation is derived by modeling the sequential data via a set of basis functions and therefore coefficients in the parameter space forms a global representation that can be used to reconstruct the sequential data approximately. Some commonly used global representations are polynomial/spline curve fitting [12],[13], discrete Fourier transforms [14], discrete wavelet transforms [13]. In contrast, a piecewise representation is generated by partitioning the sequential data into segments at critical points based on a criterion then each segment will be characterized by a concise representation. As a result, all segment representations constitute an entire piecewise representation collectively, e.g. adaptive piecewise constant approximation [16] and curvature-based PCA segments [17]. As pointed out in Sect. 1, there is no universal representation that perfectly characterizes all sequential data. Thus, a global representation is often good at characterizing global features by smoothing out those local or fine details whilst a piecewise representation characterizes the local features very well but may fail to highlight the global landscape underlying sequential data.

Apparently, the complementary nature of global and piecewise representations suggests that it is likely to reach the synergy if we jointly use such representations to characterize sequential data. From the computation perspective, a representation of sequential data in a fixed dimension converts a temporal data clustering task into a static data clustering in the feature space. A global representation always leads to a feature vector of the fixed dimension regardless of the length of sequential data, while a piecewise representation often forms a feature vector of a dynamic dimension that depends on the nature of data, i.e. the number of critical points. Thus, most of the existing piecewise representations are not applicable in competitive learning due to dynamic dimensionalities of their feature vectors for different sequential data. As a result, we develop a coarse piecewise representation named *piecewise local statistics* (PLS) for our purpose. Thus we would adopt both the proposed piecewise representation and two typical global representations, i.e. polynomial curve fitting (PCF) and discrete Fourier transforms (DFT), in our simulations in order to demonstrate the benefit of using different representations for sequential data clustering analysis.

Most of sequential data can be regarded as time series of T sequential points expressed as $\{x(t)\}_{t=1}^{T}$. For instance, motion trajectory resulting from motion tracking has been employed to express video sequences. A motion

trajectory is a 2-D spatiotemporal data of the notation $\{(x(t), y(t))\}_{t=1}^{T}$, where $(x(t), y(t))$ is the coordinates of an object tracked at frame *t*, and therefore can also be treated as two separate time series $\{x(t)\}_{t=1}^{T}$ and $\{y(t)\}_{t=1}^{T}$ by considering its *x*- and *y*-projection respectively. As a result, the representation of a motion trajectory is simply a collective representation of two time series corresponding to its *x*- and *y*-projection. In the sequel, we shall present our PLS representation and briefly review two global representations only for univariate time series without a loss of generality.

2.1 Piecewise Local Statistics

Motivated by the short-term analysis in speech signal processing, we adopt a window based statistic analysis for time series. First of all, we use a window of the fixed size to block time series into a set of segments. For each segment, we estimate the 1st- and 2nd-order statistics used as features of this segment. For segment *n*, its local statistics, μ_n and σ_n, are estimated by

$$\mu_n = \frac{1}{|W|}\sum_{t=1+(n-1)|W|}^{n|W|} x(t), \quad \sigma_n = \sqrt{\frac{1}{|W|}\sum_{t=1+(n-1)|W|}^{n|W|}[x(t)-\mu_n]^2} \tag{13.1}$$

where $|W|$ is the size of the window.

For time series, a PLS representation of a fixed dimension is formed by the collective notation of local statistic features of all segments though the estimate might be affected at the end of time series where the window is delimited.

2.2 Polynomial Curve Fitting

In [13], time series is modeled by fitting it to a parametric polynomial function

$$x(t) = \alpha_P t^P + \alpha_{P-1} t^{P-1} + \cdots + \alpha_1 t + \alpha_0. \tag{13.2}$$

Here α_p $(p = 0,1,\cdots,P)$ is the polynomial coefficient of the *p*th order. The fitting is carried out by minimizing a least-square error function by considering all sequential points of time series and the polynomial model of a given order, with respect to α_p $(p = 0,1,\cdots,P)$. All coefficients obtained via optimization constitute a PCF representation, a sequential point location dependent global representation of time series.

2.3 Discrete Fourier Transforms

Discrete Fourier transforms have been applied to derive a global representation of time series in frequency domain [14]. The DFT of time series $\{x(t)\}_{t=1}^{T}$ yields a set of Fourier coefficients:

$$a_k = \frac{1}{T}\sum_{t=1}^{T} x(t)\exp(\frac{-j2\pi kt}{T}),\ k = 0,1,\cdots,T-1 \tag{13.3}$$

In order to form a robust representation in presence of noise, only few top *K (K<<T)* coefficients corresponding to low frequencies are collectively to form a Fourier descriptor, a sequential point location independent global representation of time series.

3. ENSEMBLE COMPETITIVE LEARNING MODEL

In this section, we present an unsupervised ensemble learning model for combining multiple competitive learning networks individually trained on different representations for sequential data clustering. We first review the underpinning techniques used in our model, including the component RPCCL network [6] as well as clustering ensemble methods [10], and then describe our model. Finally we demonstrate how our unsupervised ensemble learning model and its components work with two 2-D synthetic data sets.

3.1 RPCCL Network

The RPCCL network [6] is a variant of the RPCL network [5] for competitive learning. Inheriting the strength of automatic model selection from the RPCL network, the RPCCL network overcomes the difficulty in determining de-learning rate and therefore is easier to use.

A RPCCL network consists of *M* binary units arranged in a layer. We use u_i and $\mathbf{w}_i$ to denote the output of unit *i* and its weight. All weights are initialized randomly at the beginning of learning. For a data set of *N* objects, $\{\mathbf{x}_n\}_{n=1}^{N}$, generating form $\overline{M}$ unknown intrinsic groups, the RPCL learning [5] tends to find out a set of proper weights adaptively so that

$$u_i(\mathbf{x}_n) = \begin{cases} 1, \text{if } i = \arg\min_{1\le j\le \overline{M}} \| \mathbf{x}_n - \mathbf{w}_j \|^2 \\ 0, \text{otherwise,} \end{cases} \tag{13.4}$$

Here $\|\bullet\|$ is the Euclidean norm.

In order to obtain (13.4), the RPCL learning rule [5] has been developed, which consists of the following two steps:

1) Randomly choose an object $\mathbf{x}_n$ from the data set $\{\mathbf{x}_n\}_{n=1}^N$ and for $i=1,2,...,k$, set the output of units by

$$u_i(\mathbf{x}_n) = \begin{cases} 1, \text{if } i = c, c = \arg\min_{1 \le j \le M} \rho_j \| \mathbf{x}_n - \mathbf{w}_j \|^2 \\ -1, \text{if } i = r, r = \arg\min_{1 \le j \le M, j \ne c} \rho_j \| \mathbf{x}_n - \mathbf{w}_j \|^2 \\ 0, \text{otherwise} \end{cases} \tag{13.5}$$

where $\rho_j = N_j / \sum_{i=1}^M N_i$ and N_j is the total number of the winning occurrences of unit j so far.

2) Update the weights of units by

$$\Delta \mathbf{w}_i = \begin{cases} \eta_c (\mathbf{x}_n - \mathbf{w}_i), \text{if } u_i(\mathbf{x}_n) = 1 \\ -\eta_r(\mathbf{x}_n)(\mathbf{x}_n - \mathbf{w}_i), \text{if } u_i(\mathbf{x}_n) = -1 \\ 0, \text{otherwise} \end{cases} \tag{13.6}$$

and increment N_j only if $u_j(\mathbf{x}_n) = 1$ or for the winner.

In Step 2, η_c and $\eta_r(\mathbf{x}_n)$ are the learning and the de-learning rates. While the learning rate needs to be set prior to competitive learning, the data-driven de-learning rate is automatically determined by the RPCCL rule [6] as follows:

$$\eta_r(\mathbf{x}_n) = -\eta_c \frac{\min\{\| \mathbf{w}_r - \mathbf{w}_c \|^2, \| \mathbf{x}_n - \mathbf{w}_c \|^2\}}{\| \mathbf{w}_r - \mathbf{w}_c \|^2} \tag{13.7}$$

For competitive learning, the algorithm repeats Steps 1 and 2 until a pre-set stopping condition is reached.

In the RPCCL algorithm, there are several crucial factors that critically determine its performance of competitive learning. Our empirical studies in sequential data clustering indicate that its performance is sensitive to the learning rate selection as well as the initialization and the stopping conditions as exemplified in sec. 3.3. To our knowledge, there is no systematic way to choose the aforementioned parameters. Thus, our model cannot solely rely on this algorithm for robust clustering analysis; hence other advanced techniques need to be sought to ensure the robustness, which would be a reason why we employ clustering ensemble techniques in our model.

3.2 Clustering Ensemble Techniques

Clustering ensemble techniques have been recently studied to tackle several difficulties faced by an individual clustering algorithm. The basic idea underlying clustering ensemble techniques is combining multiple partitions of a data set by a consensus function to yield a final partition that more likely reflects the intrinsic structure of the data set.

A clustering ensemble method named Cluster Ensembles [10] has been recently proposed for the problem of combining multiple partitions of data without accessing the representations of data that determined these partitions, which results in a knowledge reusable framework. In [10], three consensus functions have been proposed and an objective function based on mutual information is further introduced to evaluate the performance of candidates yielded by different consensus functions optimally. However, all three consensus functions suffers from a weakness; i.e. the number of clusters in a final partition needs to be determined manually in advance or simply use the maximal number of clusters appearing in multiple partitions to be combined for model selection. Our empirical studies on different consensus functions indicate that the *cluster-based similarity partitioning algorithm* (CSPA) [10] often outperforms other two proposed in [10]. Therefore, we adopt the CSPA as one of consensus function in our ensemble learning model. Motivated by the clustering ensemble based on evidence accumulation [11], we also introduce an alternative consensus function that can automatically determine the number of clusters in the final partition. Below we briefly review the CSPA consensus function as well as the objective function for the final partition selection [10], and present the alternative consensus function.

3.2.1 Consensus functions

In the Cluster Ensemble [10], multiple partitions by an individual clustering algorithm are first mapped onto a hypergraph where its edge named hyperedge is allowed to connect any set of vertices. In the hypergraph, one vertex corresponds to one object in the data set and one cluster forms a hyperedge linked to all objects in the cluster. For partition q, a binary membership indicator matrix $\mathbf{H}_q$ where a row corresponds to one object and a column refers to a binary encoding vector of one cluster in partition q. Thus concatenating all $\mathbf{H}_q$ of multiple partitions leads to an adjacency matrix $\mathbf{H}$ by all objects in the data set versus all the available partitions.

Based on such a hypergraph representation, the CSPA specifies a consensus function as follows. The hypergraph representation encodes the

piecewise similarity between any two objects; i.e. the similarity of one indicates two objects are grouped into the same cluster and a similarity of zero otherwise. Thus a similarity matrix $\boldsymbol{S}$ for all available partitions represented in a hypergraph is derived from the adjacency matrix $\mathbf{H}$: $\mathbf{S} = \frac{1}{|P|}\mathbf{H}\mathbf{H}^T$ where $|P|$ is the number of partitions yielded by multiple-round clustering analyses. The average of similarities yielded from multiple partitions can be used to re-cluster objects to yield a final consensus.

We adapt the idea of [11] into a *dendrogram-based similarity partitioning algorithm* (DSPA) that determines the number of clusters in the final partition automatically. First of all, a co-associate matrix reflecting the relationship of all objects in multiple partitions is established where an element indicates the similarity defined by the number of occurrences as two specific objects are grouped into the same cluster. The co-associate matrix actually accumulates evidence and allows us to apply any clustering algorithm over this new similarity matrix for finding out a final partition. In our simulation, the average link method is applied to yield a dendrogram representation [2],[11]. Thus the proper number of clusters in the final partition is determined by cutting the dendrogram at the range of threshold points corresponding to the longest lifetime of clusters [2],[11].

3.2.2 Mutual-information based objective function

Although the aforementioned two consensus functions can be used individually to yield a clustering ensemble, their performance could be different as applied to data sets of various distributions. Without the prior information, it seems impossible to select a proper consensus function in advance to form a clustering ensemble. As a result, we apply a normalized mutual-information (NMI) based objective function proposed in [10] to measure the consistency between any two partitions:

$$\mathrm{NMI}(P^a, P^b) = \frac{\sum_{i=1}^{K_a}\sum_{j=1}^{K_b} N_{ij}^{ab} \log(\frac{NN_{ij}^{ab}}{N_i^a N_j^b})}{\sum_{i=1}^{K_a} N_i^a \log(\frac{N_i^a}{N}) + \sum_{j=1}^{K_b} N_j^b \log(\frac{N_j^b}{N})} \tag{13.8}$$

Here P^a and P^b are labeling for two partitions that divide a data set of N objects into K_a and K_b clusters, respectively. N_{ij}^{ab} is the number of shared objects between clusters $C_i^a \in P^a$ and $C_j^b \in P^b$, where there are N_i^a and N_j^b objects in C_i^a and C_j^b.

Based on (13.8), the optimal final partition can be determined by finding out the one that possesses maximal average mutual information with all $|P|$ partitions available from multiple-round clustering analyses prior to the

clustering ensemble [10]. Thus finding the proper one from R various consensus functions can be done by

$$P^* = \arg\max_{1 \le r \le R} \sum_{p=1}^{|P|} \mathrm{NMI}(P^r, P^p) \tag{13.9}$$

In other words, the consensus function yielding the partition P^* is the proper one for the given data set.

3.2.3 Model Description

Based on the underpinning techniques presented above, we come up with an ensemble competitive learning model for sequential data clustering.

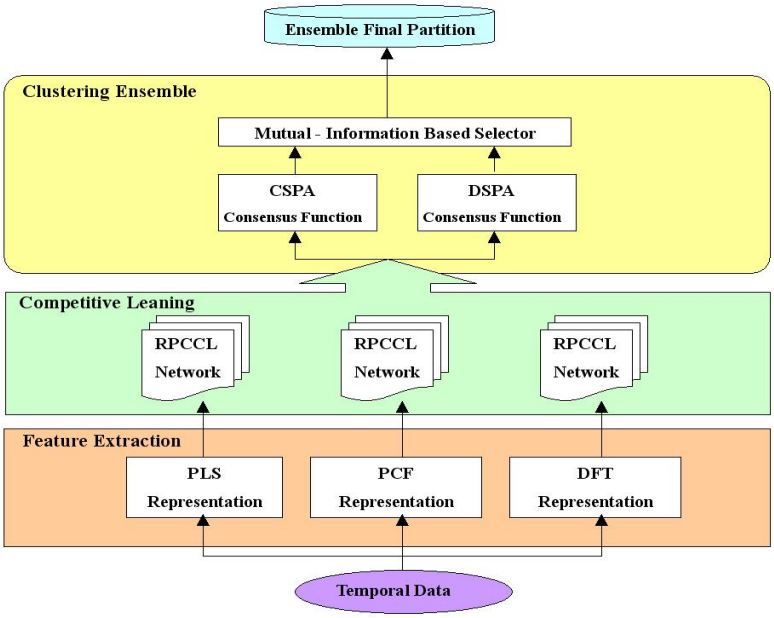

Figure 13.1. An ensemble competitive learning model with different representations.

As illustrated in Figure 13.1, our model consists of three modules; i.e. feature extraction, RPCCL competitive learning and clustering ensemble. In the feature extraction module, various representation methods of the complementary nature are demanded, as exemplified by three methods described in Sect. 2. Thus, sequential data are transformed into different representations to be the input of RPCCL networks. In the competitive learning module, a RPCCL network on an individual representation would

be trained with its learning rules given in (13.5)-(13.7). Since the performance of a RPCCL network is often sensitive to the learning rate, the initialization and the stopping conditions, a RPCCL network would be trained in different conditions for several times, which yields multiple partitions. As a consequence, RPCCL networks on different representations lead to a collection of multiple partitions that can be encoded by a hypergraph described in Sect. 3.2.1. In the clustering ensemble module, two consensus functions presented in Sect. 3.2.1 are applied, respectively, by combining the collection of multiple partitions generated from the competitive learning to form the partition candidates from different perspectives. Finally the use of the objective function in (13.8) offers a mutual-information based selector in (13.9) that determines an optimal partition, named final partition, from a group of candidates for a given sequential data set.

3.3 Demonstrable Examples

In order to demonstrate the ensemble competitive learning and investigate the performance of a RPCCL network, we use a Gaussian mixture model to produce two 2-D data sets. In our experiment, we set the learning rate as 0.001 by default for the RPCCL network as suggested in [6], and choose six seed points whose initial positions are all randomly assigned in the input data space to test its model selection performance.

We produce 1000 data points randomly from a mixture of four Gaussian distributions as follows:

$$\begin{aligned} p(\mathbf{x}) = {} & 0.26N\left[\mathbf{x} \mid \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0.15 & 0\\0 & 0.15\end{pmatrix}\right] + 0.22N\left[\mathbf{x} \mid \begin{pmatrix}1\\2.5\end{pmatrix}, \begin{pmatrix}0.15 & 0\\0 & 0.15\end{pmatrix}\right] \\ & + 0.20N\left[\mathbf{x} \mid \begin{pmatrix}2.5\\2.5\end{pmatrix}, \begin{pmatrix}0.15 & 0\\0 & 0.15\end{pmatrix}\right] + 0.32N\left[\mathbf{x} \mid \begin{pmatrix}-1\\-1\end{pmatrix}, \begin{pmatrix}0.15 & 0\\0 & 0.15\end{pmatrix}\right] \end{aligned} \tag{13.10}$$

where each component $N(\mathbf{x} \mid \mathbf{m}, \boldsymbol{\Sigma})$ denotes the Gaussian probability density function of variable $\mathbf{x}$ with the mean vector $\mathbf{m}$ and the covariance matrix $\boldsymbol{\Sigma}$. As shown in Figure 13.2(a), a set of 2-D data points form four ball-shaped clusters where we mark the four clusters with different colors to indicate their ground truth; i.e. the data points marked by the same color are produced by the same Gaussian component distribution in (13.10). As a result, it is observed from Figure 13.2(a) that three clusters are overlapped moderately at the upper-right corner while one cluster is well separated from others at the lower-left corner. In terms of the shape and location of different data clusters, we would refer to this data set as a simple data set.

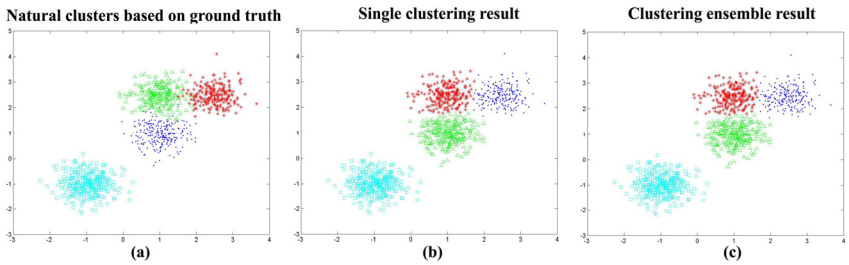

Figure 13.2. Performance on a simple data set. (a) The data set produced by (13.11). (b) The result of a RPCCL. (c) The result of a RPCCL ensemble.

Now we report results on the RPCCL on this data set for a single run. After 150 epochs, four seed points are moved into the appropriate position where is the center of each cluster, and the rest two seed points are pushed far away from the data points. The Figure 13.2(b) shows the clustering result by grouping the data points into the closed seed points. Next, we run the RPCCL with different initial positions of seed points for 20 trials, which produces 20 partitions. Then we use the ensemble technique presented in Sect. 3.2 to combine these partitions into a final single partition. As illustrated in Figure 13.2(c), the clustering result is exactly the same as the single RPCCL run.

From the results on the data set produced by (13.10), it is evident that the RPCCL network is capable of tackling the model selection problem by automatically determining the number of clusters without the use of any prior information on the data set. Moreover, the application of the ensemble learning on partitions produced by RPCCL network does not alter its correct partition and the number of clusters automatically determined previously.

By manipulating parameters of a Gaussian mixture distribution, we could re-locate clusters in the data space by altering their mean vectors and shapes as well as spread levels of clusters by altering their covariance matrixes. In order to produce a more challenging data set of 1000 data sets, we employ another mixture of four Gaussian distributions as follows:

$$\begin{aligned} p(\mathbf{x}) = 0.25N\left[\mathbf{x}\,|\begin{pmatrix}5\\0\end{pmatrix},\begin{pmatrix}0.9 & 1\\0 & 0.9\end{pmatrix}\right] + 0.25N\left[\mathbf{x}\,|\begin{pmatrix}10\\0\end{pmatrix},\begin{pmatrix}9 & 0\\0 & 9\end{pmatrix}\right] \\ + 0.25N\left[\mathbf{x}\,|\begin{pmatrix}20\\0\end{pmatrix},\begin{pmatrix}0.9 & 5\\0 & 0.9\end{pmatrix}\right] + 0.25N\left[\mathbf{x}\,|\begin{pmatrix}40\\3\end{pmatrix},\begin{pmatrix}0.9 & 5\\0 & 0.9\end{pmatrix}\right] \end{aligned} \tag{13.11}$$

As illustrated in Figure 13.3(a), the 2-D data form four clusters marked by different colors and cluster shapes become irregular other than the ball shape shown in Figure 13.2(a). On the right hand side, two stripe-shaped

clusters are well separated. On the left hand side, however, one dense cluster is wrapped by another sparse cluster.

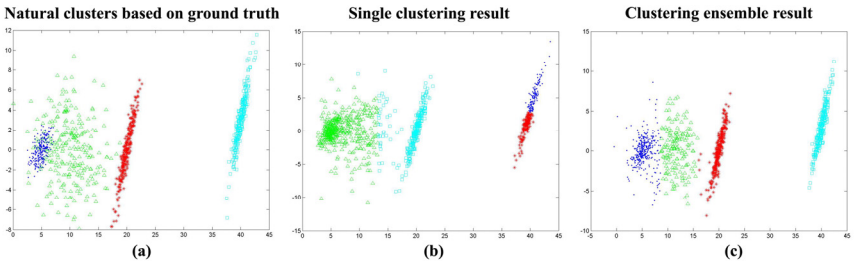

Figure 13.3. Performance on a complex data set. (a) The data set produced by (13.11). (b) The result of a RPCCL. (c) The result of a RPCCL ensemble.

As shown in the Figure 13.3(b), the single RPCCL run results in the poor grouping for the data set produced by (13.11). From Figure 13.3(b), one strip-shaped cluster is wrongly partitioned into two separate clusters, and the wrapped dense cluster fails to be separate from the adjacent sparse cluster. In contrast, the application of the ensemble technique to the RPCCL networks yields a significantly improved partition as illustrated in Figure 13.3(c) where all the data points in the stripe-shaped cluster are correctly grouped and most of data points in the wrapped dense cluster are isolated from the data points in the adjacent sparse cluster though their boundary is less precise.

From this example, we would conclude that a single RPCCL run could fail to solve the model selection problem and fulfill the proper grouping for a data set like that produced by (13.11). Here we would refer to such a data set as a complex data set. Fortunately the ensemble learning technique appears to be helpful in this circumstance; to a great extent, it takes different partitions of RPCCL networks and manages to tackle both model selection and grouping problem for a complex data set.

In summary, the above examples demonstrate the effectiveness of combining multiple RPCCL networks for clustering.

4. 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 on only the clustering-based classification

tasks in simulations. Clustering analysis is first done by partitioning a set of training data into several clusters. The training data are labeled based on clustering analysis and then used as prototypes to classify testing data unseen during training.

In our simulations, we apply an ensemble of RPCCL competitive learning networks to a synthetic time series benchmark, the Cylinder-Bell-Funnel data set [3], and an objective trajectory benchmark, the CAVIAR visual tracking database [18]. Below we report the performance of our model in two benchmark clustering tasks.

4.1 The Cylinder-Bell-Funnel Data Set

This data set has been used as a benchmark in sequential data mining [3]. The data are generated by three time series functions:

$$\begin{aligned} c(t) &= (6+\kappa)x_{[a,b]}(t) + \varepsilon(t), \\ b(t) &= (6+\kappa)x_{[a,b]}(t) + \varepsilon(t), \\ f(t) &= (6+\kappa)x_{[a,b]}(t)(t-b)/(b-a) + \varepsilon(t). \end{aligned} \tag{13.12}$$

where κ and $\varepsilon(t)$ are drawn from a Gaussian distribution $N(0,1)$, a an b are two integers randomly drawn from intervals [16, 32] and [48, 128], and $x_{[a,b]}(t)$ is defined as one if $b \le t \le a$ and zero otherwise. Three stochastic functions in (13.12) randomly generate time series of 128 frames corresponding to three classes: Cylinder, Bell and Funnel. In our simulations, we generated 200 samples for each class and then randomly divided them into two subsets of equal sizes; one for training and the other for test.

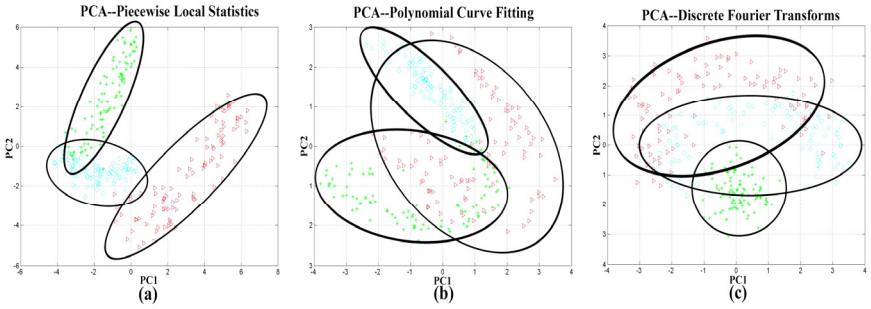

Figure 13.4. Distribution of samples in various PCA representation manifolds formed by the 1st and 2nd principal components. (a) PCL. (b) PCF. (c) DFT.

Feature extraction was first done for all training samples to generate three representations for each time series. In order to investigate the nature of different representations, we perform the principal component analysis (PCA) on three representations for the training set. As illustrated in Figure 13.4, different representations lead to diverse distributions of samples in the PCA representation subspace so that intra- and inter-class variations of samples can be various in different representation space. To some extent, this justifies that the use of multiple complementary representations paves a way to promote the diversity of ensemble clustering.

For robust competitive learning, the RPCCL networks were independently trained on individual representations for 20 times by setting various learning rates, initialization and stopping conditions so that 20 partitions can be generated for each representation. In order to evaluate the model selection performance, we deliberately choose the initial number of units in a RPCCL network as **five** although the ground truth is available, i.e. **three** classes underlying the data set. Thus the RPCCL network needs to estimate the proper number of clusters during its rival penalized learning.

Once multiple partitions are available, two consensus functions are applied to yield final partition candidates and then the mutual-information based selector to determine the optimal final partition.

Table 13.1. Classification Rate (%) of Various Methods and Parameters.

Representation Method	Parameter in Representation	Single Clustering (classification rate) training	testing	Clustering Ensemble (classification rate) training	testing
PLS	$\|W\|=3$	73.0	71.9	78.9	77.9
	$\|W\|=6$	73.1	72.4	79.4	78.1
	$\|W\|=12$	70.6	69.1	78.5	75.1
PCF	$P=3$	60.2	59.2	63.3	62.1
	$P=4$	61.1	60.2	64.1	62.3
	$P=5$	58.3	54.9	62.2	60.0
DFT	$K=8$	66.7	63.3	67.9	64.6
	$K=16$	75.7	72.9	76.9	74.7
	$K=32$	60.7	58.1	63.2	60.7
Different Representations	$\|W\|=6$, $P=4$, $K=16$	N/A		82.7	81.0

For statistical reliability, the aforementioned experiment is repeated 40 times and the averaging results are reported here. Table 13.1 shows the performance of single RPCCL networks with single representations, an ensemble of RPCCL networks with single representations and an ensemble

of RPCCL networks with different representations. From Table 13.1, it is observed that the performance of RPCCL networks changes on single representations due to different distributions demonstrated in Figure 13.4 and ensembles of RPCCL networks, in particular, with different representations significantly improve the performance on both training and testing sets. In addition, the performance is relatively insensitive to parameters of different representations. Note the classification rate is calculated by comparing the results to the ground truth. Here we emphasize that all results are achieved based on initializing the RPCCL network with the number of clusters (i.e., **five**) inconsistent with the ground truth (i.e., **three**).

Using the proper number of clusters given in the ground truth, many clustering algorithms, e.g. K-means and SOM, with an individual representation, have been applied to the data set, and typical classification rates reported are about between 75% and 80% [3],[19]. For comparison, our ensemble model has also been used on the same condition, i.e. the use of ground truth information, and the averaging classification rates on training and testing sets are 85.2% and 83.7%, respectively. In contrast to the performance of our ensemble model in Table 13.1 without the use of prior information on the data set, we conclude that our ensemble competitive learning model yields robust clustering analysis and hence suitable for being applied in an unknown environment.

4.2 The CAVIA Visual Tracking Database

The CAVIA database is a benchmark designed for video content analysis [18]. From the manually annotated video sequences of pedestrians, a set of 222 high-quality moving trajectories, as shown in Figure 13.6(a), are achieved for clustering analysis.

In our simulation, our model is first applied to all trajectories for clustering analysis. Since the information on the "right" number of clusters is unavailable, 20 units are initially adopted in the RPCCL network for competitive learning based on our belief that too many clusters make little sense for a higher level analysis. Similarly, the RPCCL network based on an individual representation yields 20 partitions by training repeatedly with different conditions and all 60 partitions are combined by different consensus functions to yield final partition candidates. The final partition is eventually chosen by the mutual-information based selector. As illustrated in Figure. 13.5, our model automatically partitions the 222 moving trajectories into 14 clusters. Human visual inspection suggests that similar motion trajectories have been grouped together properly while dissimilar ones are distributed into different clusters. For example, those trajectories

corresponding to moving from left-to-right and right-to-left are properly grouped into two separate clusters as shown in Figure 13.5(f) and (g).

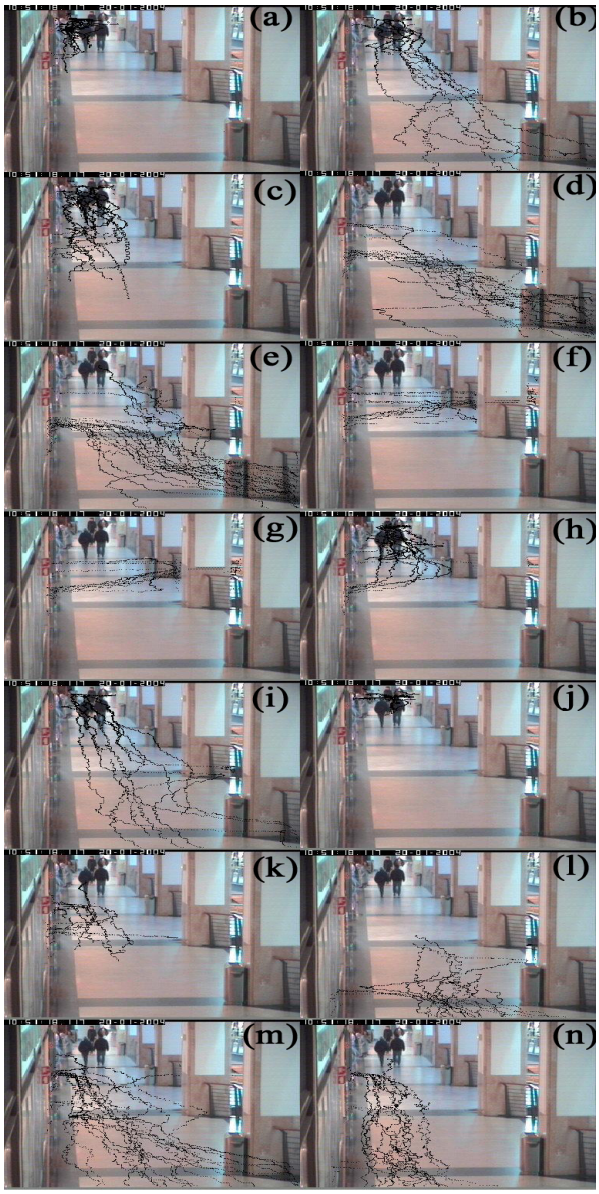

Figure 13.5. A clustering analysis of all moving trajectories in CAVIAR database by our ensemble competitive learning model. Plots in (a)-(n) correspond to 14 clusters of moving trajectories in the final partition.

The CAVIAR database has also been used in [19] to evaluate their method where the SOM with an individual representation was used. In their simulation, the number of clusters was determined manually, and all trajectories are simply grouped into nine clusters. Although most of clusters achieved in their method are consistent with ours, a couple of separate clusters achieved in our model, e.g. clusters shown in Figure 13.5(a) and (c), are merged into a single cluster in theirs. If trajectories in separate clusters express different events, such a merge would lead to a difficulty in a higher level analysis.

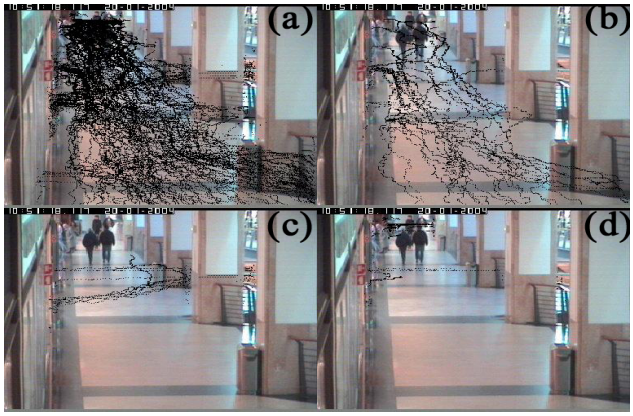

Figure 13.6. All moving trajectories with background scene in CAVIAR database and typical clustering of RPCCL network ensembles only with individual representations. (a) All moving trajectories. (b) PLS. (c) PCF. (d) DFT.

For demonstrating the benefit of jointly using different representations in our model, Figure 13.6(b)-(d) illustrate three typical clusters of moving trajectories yielded by RPCCL network ensembles only with individual representations. Visual inspection indicates inadequate clustering achieved due to their weakness of individual representations. The cluster shown in Figure 13.6(b) groups trajectories from two clusters shown in Figure 13.5(b) and 13.5(m) since the PLS representation highlights local features but relatively neglects global characteristics. In Figure 13.6(c), some "orthogonal" trajectories are improperly grouped together due to the limited representation capability of the PCF. Similarly, trajectories with the same orientation but starting from different positions are incorrectly grouped together because the DFT is a representation independent of spatial locations. In contrast to the clustering results shown in Figure 13.5, all of such less meaningful clusters no longer exist thanks to the joint use of different representations.

For further evaluation, we have done two additional experiments in classification: a) adding different amounts of Gaussian noise, $N(0,\sigma)$, to a range of coordinates in the database; b) randomly removing different parts of each moving trajectory and producing its noisy missing data version by further adding the Gaussian noise with $\sigma = 0.1$. The former tends to generate testing data sets whilst the latter simulates common scenarios that a moving object tracked is occluded by other objects or the background so that a tracking algorithm has to produce a trajectory with missing data.

Table 13.2. Classification Rate (%) of Ensemble methods on CAVIAR.

Representation Method	Parameter in Representation	(Clustering Ensemble) σ =0.1	σ =0.2	σ =0.3	σ =0.4
PLS	$\|W\|$=150	95.8	91.7	87.2	84.3
PCF	P=4	89.5	83.6	78.0	70.2
DFT	K=16	93.9	88.2	87.2	77.4
Different Representations	$\|W\|$ = 150 P = 4 K = 16	96.4	92.3	87.8	84.7

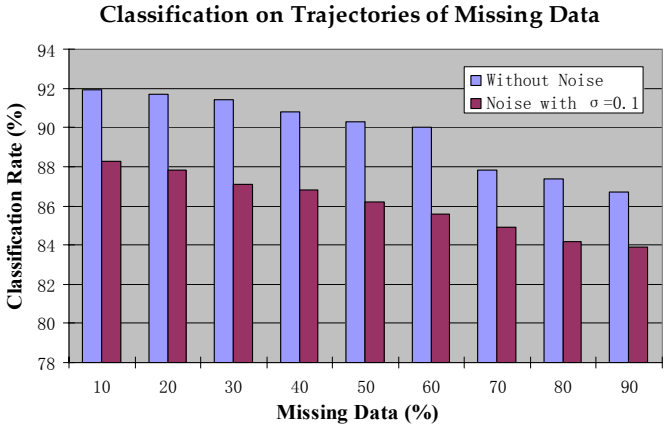

Figure 13.7. Performance of our ensemble model with different representations on CAVIAR as missing data appear.

Table 13.2 lists results for classification based on the clustering analysis in Figure 13.5. Apparently doing so highly relies on the quality of clustering analysis that was backed by the human visual inspection. The results in Table 13.2 reveal that the classification performance would be satisfactory and acceptable especially as a substantial amount of noise is added. Figure 13.7 illustrates the evolution of performance degradation caused by different amounts of missing data. It is apparent that our model is capable of dealing with trajectories of an occluded object tracked.

In summary, all above simulation results suggest that our model leads to a robust clustering analysis without the use of prior information on a given data set and, therefore, its outcome can be used for higher level video content analyses.

5. CONCLUSION

In this chapter, we have presented an unsupervised ensemble learning model for sequential data clustering by combining multiple competitive learning networks with different representations. Without the use of prior information on a given data set, simulation results on different types of sequential data demonstrate that our model yields favorite results in both model selection and grouping. In particular, the joint use of different representations leads to a significant improvement.

There are several issues to be studied in our ongoing research. First, three representations used in this chapter are coarse for representing sequential data, which is simple for a demonstration purpose. Exploration of effective yet complementary representations would be an urgent topic to be investigated in our research. Next, our simulations indicate that the use of the RPCCL network results in two-fold effects. On the one hand, its automatic model selection capability is helpful to cope with problems in an unknown environment. On the other hand, our simulation results including those not reported here due to space indicate that its learning rule seems to hinder generating truly diverse partitions; although the use of different learning rates, the initialization and the stopping conditions leads to different partitions, the correlation among them seems quite high. To our knowledge, there has been no theoretic analysis available so far regarding combining the highly correlated partitions. Nevertheless the theoretic analysis in combining classifiers suggests that combining highly correlated classifiers is unlikely to yield the considerate improvement in classification [20]. We trust that their conclusion should be more or less applicable to ensemble clustering. As a result, we need to investigate the correlation problem behind the RPCL style

clustering analysis. Finally, we are exploiting intrinsic contextual information underlying some types of sequential data, e.g. video clips, and exploring a semi-supervised learning way to incorporate such information into sequential data clustering.

ACKNOWLEDGMENT

Authors are grateful to Yiu-ming Cheung for providing his RPCCL Matlab code and A. Strehl for making their Cluster Ensembles Matlab code available on line. These programs have been used in our simulations reported in this chapter.

REFERENCES

[1] Kleinberg, J.: An impossible theorem for clustering. Proceeding of Advances in Neural Information Processing Systems, vol. 15 (2002)
[2] Jain, A., Murthy, M., and Flynn, P.: Data clustering: A review. ACM Computing Surveys. vol. 31 (1999) 264-323
[3] Keogh, E. and Kasetty, S.: On the need for time series data mining benchmarks: a survey and empirical. Knowledge and Data Discovery. vol. 6 (2002) 102-111
[4] Hertz, J., Krogh, A. and Palmer, R.: Introduction to Theory of Neural Computation. New York: Addison-Wesley (1991)
[5] Xu, L., Krzyzak, A. and Oja, E.: Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. IEEE Transaction on Neural Networks. vol. 4 (1993) 636–648
[6] Cheung,Y.: On rival penalization controlled competitive learning for clustering with automatic number. IEEE Transaction on Knowledge and Data Engineering. vol. 17 (2005) 1583–1588
[7] Chen, K.: On the use of different speech representations for speaker modeling. IEEE Transaction on Systems, man, and Cybernetics (Part C). vol. 35 (2005) 301-314
[8] Chen, K., Wang, L. and Chi, H.: Methods of combining multiple classifiers with different feature sets and their applications to text-independent speaker identification. International Journal of Pattern Recognition and Artificial Intelligence. vol. 11 (1997) 417–445
[9] Chen, K. and Chi, H.: A method of combining multiple probabilistic classifiers through soft competition on different feature sets. Neurocomputing. vol. 20 (1998) 227–252
[10] Strehl, A. and Ghosh, J.: Cluster ensembles – A knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research. vol. 3 (2002) 583–617
[11] Fred, A. and Jain, A.: Combining multiple clusterings using evidence accumulation. IEEE Transaction on Pattern Analysis and Machine Intelligence. vol. 27 (2005) 835-850
[12] Dimitova, N. and Golshani, F.: Motion recovery for video content classification. ACM Transaction on Information Systems. vol. 13 (1995) 408-439
[13] Chen, W. and Chang, S.: Motion trajectory matching of video objects. In: Proceeding of SPIE/IS&T Conference on Storage & Retrieval for Media Database (2000)

[14] Faloutas, C., Ranganathan, M. and Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: Proceeding of ACM SIGMOD Conference. (1994) 419-429
[15] Sahouria, E. and Zakhor, A.: Motion indexing of video. In: Proceeding of IEEE International Conference on Image Processing. vol.2 (1997) 526-529
[16] Keogh, E., Chakrabarti, K., Pazzani, M. and Mehrota, S.: Locally adaptive dimensionality reduction for indexing large scale time series databases. In: Proceeding of ACM SIGMOD Conference. (2001) 151-162
[17] Bashir, F.: MotionSearch: object motion trajectory-based video database system – Index, retrieval, classification and recognition. Ph.D. dissertation, Dept. Elect. Eng., Univ. of Illinois, Chicago, U.S.A. (2005)
[18] CAVIAR: Context aware vision using image-based active recognition. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR.
[19] Khalid, E. and Naftel, A.: Classifying spatiosequential object trajectories using unsupervised learning of basis function coefbficients. Proceeding of ACM International Workshop on Video Surveillance & Sensor Networks, Singapore (2005) 45-52
[20] Tumer, K. and Ghosh, J.: Error correlation and error reduction in ensemble classifiers. Connection Science. vol. 8 (1996) 385-404