Time Series Clustering Via RPCL Network Ensemble With Different Representations

Yun Yang and Ke Chen, *Senior Member, IEEE*

***Abstract*—Time series clustering provides underpinning techniques for discovering the intrinsic structure and condensing/summarizing information conveyed in time series, which is demanded in various fields ranging from bioinformatics to video content understanding. In this paper, we present an unsupervised ensemble learning approach to time series clustering by combining rival-penalized competitive learning (RPCL) networks with different representations of time series. In our approach, the RPCL network ensemble is employed for clustering analyses based on different representations of time series whenever available, and an optimal selection function is applied to find out a final consensus partition from multiple partition candidates yielded by applying various consensus functions for the combination of competitive learning results. As a result, our approach first exploits its capability of the RPCL rule in clustering analysis of automatic model selection on individual representations and subsequently applies ensemble learning for the synergy of reconciling diverse partitions resulted from the use of different representations and augmenting RPCL networks in automatic model selection and overcoming its inherent limitation. Our approach has been evaluated on 16 benchmark time series data mining tasks with comparison to state-of-the-art time series clustering techniques. Simulation results demonstrate that our approach yields favorite results in clustering analysis of automatic model selection.**

***Index Terms*—Automatic model selection, different representations, rival-penalized competitive learning (RPCL), time series clustering, unsupervised ensemble learning.**

I. Introduction

TIME series data are ubiquitous in the real world, and there are many application areas ranging from biological information processing to temporal data mining. In general, essential temporal series processing techniques are categorized as modeling, clustering, classification, and prediction. Unlike static data, there is a high amount of dependency among time series and the proper treatment of data dependency or correlation becomes critical in time series processing.

Time series clustering analysis provides an effective way to discover the intrinsic structure and to condense/summarize information conveyed in temporal data by exploring dynamic behaviors hidden underlying time series in an unsupervised learning paradigm. The ultimate objective of time series clustering analysis is to partition a set of unlabeled time series into groups or clusters where all the sequences 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 way that estimates the intrinsic number of clusters underlying a temporal dataset, while the latter demands a proper grouping rule to gather coherent sequences 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]. In particular, recent empirical studies in time series data mining reveal that most of the existing clustering algorithms do not work well due to their complexity of underlying structure and data dependency [2], which poses a real challenge in clustering time series of a high dimensionality, complicated temporal correlation, and a substantial amount of noise. In the context of the treatment of data dependency, existing time series clustering algorithms can be categorized as *temporal-proximity-based, model-based*, and *representation-based* clustering methodologies summarized in Table I. Note that our taxonomy in Table I mainly serves for facilitating our presentation on time series clustering addressed in this paper and might not exhaustively cover all types of temporal data clustering algorithms by any means.

Temporal-proximity and model-based clustering algorithms directly working on time series where temporal correlation is dealt with directly during clustering analysis by means of temporal similarity measures [2]–[5], e.g., dynamic time warping, or dynamic models [6]–[10], e.g., hidden Markov model (HMM). In contrast, a representation-based algorithm converts time series into lower dimensionality of feature space, where any static-data-clustering algorithm is applicable to time series clustering, which is especially efficient in computation. Various representations have been proposed for time series [5], [11]–[19]. 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 time series. Due to the high complexity and varieties of time series, there is no universal representation that perfectly characterizes different types of time series [5]. Therefore, a representation is merely applicable to a class of time series 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. To tackle the information loss problem, a multiscale representation based on wavelet analysis was proposed [20] so that a clustering algorithm like K-Mean

Manuscript received June 24, 2009; revised December 3, 2009 and March 8, 2010; accepted June 5, 2010. Date of publication July 8, 2010; date of current version February 16, 2011. This paper was recommended by Associate Editor G. I. Papadimitriou.

The authors are with the School of Computer Science, The University of Manchester, Manchester, M13 9 PL, U.K. (e-mail: yun.yang@postgrad.manchester.ac.uk; chen@cs.manchester.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2010.2052608

TABLE I
TAXONOMY ON TIME SERIES CLUSTERING ALGORITHMS

Methodology	Working Mechanism	Advantage	Disadvantage	Example
Temporal-proximity based clustering	● Works directly on temporal data ● Similarity measure considering temporal relations	● Prevent from losing any information ● A direct way to capture the dynamic behaviors ● Flexible means to deal with variable length of temporal data	● Sensitive to initialization ● Model selection problem ● High computational complexity	Dynamic Time Warping (DTW) [2], Temporal K-mean/Hierarchical clustering [3], [4].
Model based clustering	● Clusters of temporal data are specified by a mixture of dynamic models. ● Identify the data dependency and regularity behind the dynamic behaviors of temporal data	● Generally suitable for coping with data dependency among temporal data ● Temporal data characterized with generative models	● Model selection problem ● High computational complexity	Hidden Markov Model (HMM) [6], Dynamic Bayesian Networks [7], Auto Regressive Moving Average Model (ARMA) [8].
Representation based clustering	● A set of features are extracted from raw temporal data ● All existing clustering algorithms can be applied directly on the feature space.	● Significantly reduce the computational cost ● Compatible with existing static data clustering algorithms	● Loss of useful information conveyed in the original temporal data ● Model selection problem	Polynomial/Spline Curve Fitting [9],[10], Adaptive Piecewise Constant Approximation [14], Curvature-based PCA segments [15], Multi-scaled i-kmean [20]

can work on different scales in a sequential way in order to capture useful information. In general, the aforementioned model selection problem is still unavoidable for any representation-based algorithms.

In this paper, we propose a novel yet practical approach to time series clustering via an ensemble of rival-penalized competitive learning (RPCL) networks [21] with different representations, which addresses both grouping and model selection problems in clustering analysis as a whole. Unlike the previous method [20], our approach is motivated by our previous success in the use of different representations to construct an ensemble model for dealing with difficult supervised [22]–[24] and semisupervised learning tasks [25], where the use of different representations better exploits the information conveyed in the raw data and therefore leads to the better performance. For each individual representation, we first employ an RPCL network for clustering analysis of automatic model selection, and the nature of an RPCL network often leads to quick clustering analysis. Recent researches in clustering ensembles [26], [27] provide feasible techniques to enable us to construct an ensemble of RPCL networks trained on different representations for robust clustering analysis. By modifying the clustering ensemble technique [26], our RPCL network ensemble copes with the diversity of groupings generated by RPCL networks on different representations by reconciling them in an optimal way. As a result, our RPCL network ensemble considerably reduces ambiguities resulting from the use of different initialization, learning rates, and termination conditions in an individual RPCL network, and further augments their automatic model selection capability on different representations. In order to demonstrate its usefulness, we have applied our approach to 16 time series data mining benchmarks where the ground truth is available for performance evaluation. Simulation results suggest that our approach yields the favorite performance. In particular, our study reveals that the use of clustering ensemble techniques overcomes the weakness of representation-based clustering and improves the automatic model selection performance of individual RPCL.

The rest of this paper is organized as follows. Section II addresses time series representation issues. Section III presents our approach and overviews component techniques including a variant of the RPCL network and clustering ensemble techniques used in our simulations. Section IV reports simulation results in the time series data mining benchmarks. Section V discusses issues related to our approach, and the last section draws a conclusion.

II. TIME SERIES REPRESENTATION

In general, time series representations are divided into two categories: piecewise verses holistic representations. A piecewise representation is generated by partitioning time series into segments at critical points based on a criterion, and then each segment will be modeled with a concise representation. As a result, all segment representations are lumped up together to constitute a piecewise representation collectively, e.g., adaptive piecewise constant approximation [11] and curvature-based principal component analysis segments [12]. In contrast, a holistic representation is derived by modeling time series via a set of basis functions and therefore coefficients of basis functions forms a holistic representation that can reconstruct time series approximately. Typical holistic representations include polynomial/spline curve fitting [11], [12], discrete Fourier transforms (DFT) [13], discrete wavelet transforms [14], [15], [28], [29], and so on.

Here, we review two piecewise representations, piecewise local statistics (PLS) and piecewise discrete wavelet transform (PDWT), and two holistic representations, polynomial curve fitting (PCF), and DFT, which will be used in our simulations described in Section IV.

A. PLS Representation

Motivated by the short-term analysis in speech signal processing and discovery of time series motifs [30], we adopt a window-based statistic analysis for time series. As a preprocessing, we use a window of the fixed size to block time series into a set of segments.

For each segment, we use both the first- and second-order statistics to be features for characterizing this segment. For the

nth segment, its local statistics μ_n and σ_n are estimated by

$$\mu_n = \frac{1}{|W|} \sum_{t=1+(n-1)|W|}^{n|W|} x(t)$$

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

where $|W|$ is the size of the window. The PLS representation would be viewed as an extension of the representation proposed in [18] where the first-order statistics only is used.

B. PDWT Representation

Discrete wavelet transform turns out to be an effective multiscale analysis tool. Like the preprocessing in the PLS representation, time series $\{x(t)\}_{t=1}^{T}$ is blocked into a set of segments with a window of size $|W|$. We apply the DWT to each segment for a multiscale analysis in order to capture local details in a more accurate way, e.g., abrupt changes, that often fail to be characterized accurately by local statistics in our PLS representation described earlier.

The DWT decomposes time series via the successive use of low-pass and high-pass filtering at appropriate levels. At level j, $|W|2^{-j}$ coefficients of high-pass filters Ψ_H^j encode the detailed information, while those of low-pass filters Ψ_L^j characterize coarse information. For the nth segment with a multiscale analysis of J levels, the application of the DWT leads to a piecewise representation with all coefficients collectively.

$$\{x(t)\}_{t=(n-1)|W|}^{n|W|} \Rightarrow \left\{\Psi_L^J, \{\Psi_H^j\}_{j=1}^J\right\}. \tag{2}$$

For the piecewise representation in (2), however, its dimensionality is the same as the length of each segment, i.e., the window size. Therefore, a dimensionality reduction technique must be applied to generate a parsimonious representation. The Sammon mapping technique [31] nonlinearly maps a high-dimensional point to a low-dimensional space by minimizing an error function E.

$$E = \frac{1}{\sum_{i=1}^{M-1}\sum_{j=i+1}^{M} \hat{d}_{ij}} \sum_{i=1}^{M-1}\sum_{j=i+1}^{M} \frac{(\hat{d}_{ij} - d_{ij})^2}{\hat{d}_{ij}} \tag{3}$$

where M is the number of data in a given dataset, and d_{ij} and $\hat{d}_{ij}$ are the distance between two points in a high-dimensional space and in the low-dimensional space after projection, respectively. Since the Sammon mapping technique has a capability of generating arbitrary nonlinear mappings, we always map wavelet coefficients achieved in (2) to a prespecified low-dimensional space to form our PDWT representation. It should be mentioned that the Sammon mapping technique was previously applied to the DWT to produce a holistic representation of time series [17]. Unlike their work [17], here we apply such a technique to generate a piecewise representation of time series.

C. PCF Representation

In [10], time series is modeled by fitting it to a parametric polynomial function

$$x(t) = \alpha_M t^M + \alpha_{M-1} t^{M-1} + \cdots + \alpha_1 t + \alpha_0. \tag{4}$$

Here, $\alpha_m (m = 0, 1, \ldots, M)$ is the polynomial coefficient of the mth-order. The fitting is carried out by minimizing a least square error function by considering all temporal points of time series and the polynomial model of a given order with respect to $\alpha_m (m = 0, 1, \ldots, M)$. All $M + 1$ coefficients obtained via the optimization constitute a PCF representation, a temporal point location-dependent global representation of time series.

D. DFT Representation

Discrete Fourier transforms have been applied to derive a global representation of time series in frequency domain [13]. The DFT of time series $\{x(t)\}_{t=1}^{T}$ yields a set of Fourier coefficients for $k = 0, 1, \ldots, T - 1$.

$$a_k = \frac{1}{T} \sum_{t=1}^{T} x(t) \exp\left(\frac{-j2\pi kt}{T}\right). \tag{5}$$

In order to form a robust representation in presence of noise, only few top $k (k \ll T)$ coefficients, i.e., real and imaginary parts, corresponding to low frequencies collectively form a Fourier descriptor, a temporal point location-independent global representation of time series.

III. RPCL Network Ensemble on Different Representations

In this section, we present our RPCL ensemble clustering model on different representations for time series clustering and review the underpinning techniques used in our simulations, including the rival penalization controlled competitive learning (RPCCL) network [32], an improved version of RPCL network [21], and a clustering ensemble method [26] with an additional consensus function proposed for overcoming the limitation of a RPCL network in automatic model selection.

A. Model Description

As pointed out in Section I, there is no universal representation that perfectly characterizes different types of time series. In general, a holistic representation is often good at characterizing global features by smoothing out those local or fine details, while a piecewise representation characterizes the local features very well but may fail to highlight the global-level characteristics. Furthermore, different representations in the same category (either holistic or piecewise) could also emphasize different aspects. Thus, the nature of holistic and piecewise representations suggests that RPCL networks trained on different representations would result in the diversity of clustering. With such diversity, it is likely to reach the synergy to capture the intrinsic structure of a time series dataset. Inspired by our previous work in supervised and semisupervised ensemble learning [22]–[25],

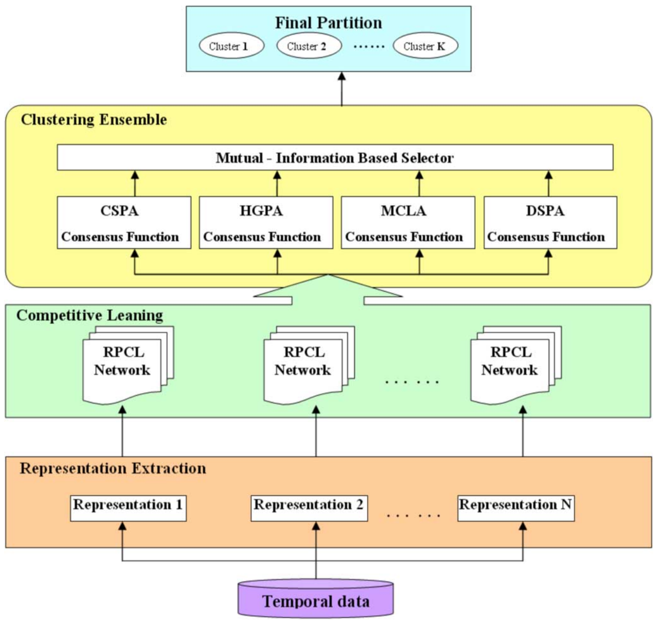

Fig. 1. RPCL network ensemble on different representations.

we come up with a model by the use of different representations for time series clustering.

Fig. 1 illustrates the architecture of our RPCL ensemble model on different representations where our model consists of three modules, i.e., representation extraction, RPCL competitive learning, and clustering ensemble. In the representation extraction module, various representations of the complementary nature are demanded, as exemplified by four methods described in Section II. Thus, time series are transformed into different representations to be the input of RPCL networks. In the competitive learning module, an RPCL network on an individual representation would be trained with its learning rules for clustering analysis. Since the performance of an RPCL network is often sensitive to initialization, learning rates, and termination conditions, we perform RPCL clustering analysis for the same representations under different conditions for several times, which yields multiple yet different partitions in the same representation space. Furthermore, RPCL networks on different representations leads to a collection of multiple partitions in different representation space. In the clustering ensemble module, a reconciliation mechanism regardless of representations is employed to make a consensus on diversified partitions produced by RPCL networks. In addition, our empirical studies in time series clustering also indicate that its performance is sensitive to the learning rate, initialization, and termination conditions [33]. To our knowledge, there is no systematic way to choose the aforementioned parameters. Thus, our model cannot solely rely on RPCL networks for robust clustering analysis, and hence other advanced techniques need to be sought to ensure the robustness, which further justifies the necessity of using clustering ensemble techniques in our model. In this paper, we extend the cluster ensemble technique [26], which presents a knowledge reuse framework for combining multiple partitions without access to representations used to yield different partitions, for integrating multiple partitions.

B. RPCCL Network

Like the original RPCL network [21], an RPCCL network consists of K binary units arranged in a layer. We use u_i and w_i to denote the output of unit i and its weight. All weights are initialized randomly at the beginning of learning. For a dataset of N objects,$\{\mathbf{x}_n\}_{n=1}^N$, generating from K^* unknown intrinsic groups, the RPCCL algorithm [32] tends to find out a set of proper weights adaptively so that

$$u_i(\mathbf{x}_n) = \begin{cases} 1, & \text{if } i = \arg\min_{1\le j\le M} \|\mathbf{x}_n - \mathbf{w}_j\|^2 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

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

In order to obtain (6), the RPCCL algorithm [32] adopts the same scheme of the original RPCL algorithm [21], which consists of the following two steps:

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

$$u_i(\mathbf{x}_n) = \begin{cases} 1, & \text{if } i = c, c = \arg\min_{1\le j\le M} \rho_j \|\mathbf{x}_n - \mathbf{w}_j\|^2 \\ -1, & \text{if } i = r, r = \arg\min_{1\le j\le M, j\ne c} \rho_j \|\mathbf{x}_n - \mathbf{w}_j\|^2 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

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

$$\Delta \mathbf{w}_i = \begin{cases} \eta_c(\mathbf{x}_n - \mathbf{w}_i), & \text{if } u_i(\mathbf{x}_n) = 1 \\ -\eta_r(\mathbf{x}_n)(\mathbf{x}_n - \mathbf{w}_i), & \text{if } u_i(\mathbf{x}_n) = -1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

and increment N_j only if $u_j(\mathbf{x}_n) = 1$; i.e., only if unit j is the winner.

In step 2, η_c, and $\eta_r(\mathbf{x}_n)$ are the learning and the delearning rates. The learning rate needs to be set prior to competitive learning like the original RPCL network [21]. Unlike the original RPCL network [21], where the delearning rate needs to be set prior to competitive learning as well, a data-driven delearning rate is automatically generated by an additional RPCCL rule [32] as follows:

$$\eta_r(\mathbf{x}_n) = -\eta_c \frac{\min\{\|\mathbf{w}_r - \mathbf{w}_c\|^2, \|\mathbf{x}_n - \mathbf{w}_c\|^2\}}{\|\mathbf{w}_r - \mathbf{w}_c\|^2}. \tag{9}$$

For competitive learning, the algorithm repeats steps 1 and 2 until a preset termination condition is reached.

C. Clustering Ensemble Technique

The basic idea underlying clustering ensemble techniques is combining multiple partitions of a dataset by a reconciliation mechanism to yield a final partition that is more likely to reflect the intrinsic structure of the dataset.

A clustering ensemble method named *cluster ensemble* [26] has been recently proposed for combining multiple partitions of data. In [26], three consensus functions have been proposed as independent reconciliation mechanisms from different perspectives. An objective function based on mutual information is further defined to evaluate the performance of candidate consensus partitions yielded by different consensus functions to reduce biases and find out a final consensus in an optimal way. However, all three consensus functions suffer from a weakness, i.e., the number of clusters in a consensus partition is determined manually in advance or simply set to be the maximal number of clusters appearing in multiple partitions to be combined for model selection. As pointed out previously, the RPCL network is subject to the limitation in automatic model selection, i.e., multiple partitions resulting from RPCL networks in the competitive learning module may not have the same number of clusters. Thus, a reconciliation mechanism that can determine the number of clusters is required to augment the capability of RPCL networks in term of model selection. Motivated by the evidence accumulation idea [27], we introduce an alternative consensus function that can automatically determine the number of clusters in a consensus partition. Thus, the mutual-information-based objective function will be applied to three consensus functions in [26] and ours to find out a final consensus partition, as illustrated in Fig. 1. Here, we briefly review the cluster ensemble algorithm [26] and present our consensus function in the context of the RPCL network ensemble on different representations.

1) Consensus Functions: In the Cluster Ensemble [26], multiple partitions generated by RPCL networks are first mapped onto a hypergraph where its hyperedges are allowed to connect any set of vertices. In the hypergraph, one vertex corresponds to one time series and one cluster forms a hyperedge linked to all time series in the cluster. For partition q, a binary membership indicator matrix $\mathbf{H}_q$ where a row corresponds to one time series and a column refers to a binary encoding vector of one cluster in partition q. As a result, concatenating all $\mathbf{H}_q$ of multiple partitions leads to an adjacency matrix $\mathbf{H}$ by all time series in the dataset versus all the partitions resulting from RPCL networks on different representations under various conditions, i.e., initialization, learning rates, and termination. Based on such a hypergraph representation, three consensus functions were developed.

The cluster-based similarity-partitioning algorithm (CSPA) is a straightforward consensus function [26]. The hypergraph encodes the piecewise similarity between any two sequences in their representation space, i.e., a similarity value is one if two time series are grouped into the same cluster and a similarity value is zero otherwise. Thus, a similarity matrix $\mathbf{S}$ for all the partitions encoded in a hypergraph is derived from the adjacency matrix H: $\mathbf{S} = \frac{1}{|P|}\mathbf{H}\mathbf{H}^T$, where $|P|$ is the number of the overall partitions yielded by RPCL networks. The average of similarities yielded from multiple partitions can be used to recluster all the sequences to yield a final consensus.

The hypergraph-partitioning algorithm (HGPA) [26] offers an alternative consensus function by casting the clustering ensemble problem into how to partition the hypergraph by cutting a minimal number of hyperedges. Such a graph-partitioning problem has been well studied in graph theory, and hence several existing algorithms are available to solve such a problem. In our simulation, we used the hypergraph-partitioning package, Hypergraph and Circuit Partitioning Software Package (HMETIS), recommended in [26]. Unlike the CSPA that takes the local piecewise similarity into account, the HGPA considers a relatively global relationship among sequences across different partitions resulting from RPCL networks. In addition, the HGPA tends to yield a consensus partition where all clusters contain approximately equal number of sequences.

The meta-clustering algorithm (MCLA) [26] results in another consensus function by grouping clusters from previous multiple partitions resulting from RPCL networks. Its basic idea is reclustering hyperedges to reduce its number from the overall number of clusters by multiple partitions to a smaller number specified by a user as the final partition. The MCLA is composed of four steps: constructing metagraph, cluster hyperedges, collapse metaclusters, and compete for time series. The constructing metagraph step converts the hypergraph into a metagraph representation based on a suitable similarity measure. The cluster hyperedges step finds out matching labels by partitioning the metagraph into several balanced metaclusters. The collapse metaclusters step collapses the hyperedges into a single metaedges that indicates the weakest or strongest association with the corresponding metacluster. Finally, they compete to assign each time series to its most associated metacluster to form a consensus partition.

Unlike the earlier three consensus functions derived from a hypergraph, we propose a dendrogram-based similarity-partitioning algorithm that automatically determines the number of clusters in a consensus partition. First of all, we construct a coassociate matrix used to reflect the relationship among all the sequences in multiple partitions. In the coassociate matrix, an element at location (*i, j*) describes the similarity defined as the number of occurrences as two sequences i and j are grouped into the same cluster. The coassociate matrix actually tends to accumulate evidence and allows us to apply any clustering algorithm over this new similarity matrix for finding out a consensus partition. Motivated by the work in [27], we adopt a classical hierarchical clustering (HC) method: the average link, which first converts the coassociation matrix into a dendrogram representation [33]. In the dendrogram, its horizontal axis indexes time series in a given data sets and its vertical axis indicates the lifetimes of clusters. The lifetime of cluster in dendrogram is defined as an interval from the time the cluster is established to the time the cluster disappears by merging with other clusters. Therefore, the number of clusters in a consensus partition can be determined automatically by cutting the dendrogram at a range of thresholds values corresponding to the longest cluster's lifetime [33].

2) Mutual-Information-Based Objective Function: Although four consensus functions can be used independently to yield a consensus partition from multiple partitions generated by RPCL networks, their performance could be different as applied to datasets of various distributions. Without the prior information, it seems impossible to select a proper function in advance to form a clustering ensemble. As a result, a normalized mutual-information (NMI)-based objective function [26] has been proposed to measure the consistency between any two partitions.

$$\mathrm{NMI}(P^a, P^b) = \frac{\sum_{i=1}^{K_a} \sum_{j=1}^{K_b} N_{ij}^{ab} \log\left(\frac{N N_{ij}^{ab}}{N_i^a N_j^b}\right)}{\sum_{i=1}^{K_a} N_i^a \log(\frac{N_i^a}{N}) + \sum_{j=1}^{K_b} N_j^b \log\left(\frac{N_j^b}{N}\right)}. \tag{10}$$

Here P^a and P^b are labelings for two partitions that divide a dataset 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 (10), the optimal consensus partition as the final consensus partition is determined by a search for the one that possesses the maximal average mutual information with all $|P|$ partitions resulting from RPCL networks prior to the clustering ensemble. Thus, finding the proper one from R various consensus functions (in our case, R is equal to four) can be done via the following optimization procedure.

$$\hat{P}^* = \arg \max_{1 \le r \le R} \sum_{p=1}^{|P|} \mathrm{NMI}(\hat{P}^r, P^p) \tag{11}$$

Here, P^p is the pth one of P partitions generated from RPCL networks and $\hat{P}^r$ is a consensus partition yielded by the rth one of R consensus functions. In other words, the consensus function yielding the partition $\hat{P}^*$ would be chosen as the final consensus partition for the given time series dataset.

D. Algorithm Complexity

In our approach, we employ the RPCL network of a low computational complexity $O(KNd)$, where $K (K \ll N)$ is the number of neurons, N is the number of data objects, and d is the dimension of representation space, for the initial clustering analysis of automatic model selection. Thus, multiple partitions are generated in parallel by independently running RPCL networks with different representations on different conditions, e.g., initialization, learning rates, and stopping rules. According to [26], the clustering ensemble has a computational complexity $O(|P|N^2)$, where $|P| (|P| \ll N)$ is the number of partitions to be combined. Note that the candidate partition generation with different consensus functions is independent and hence can be done in parallel.

Any representation-based clustering methods inevitably encounter the model selection problem. In an alternative way, a statistical model selection method, e.g., Akaike information criterion (AIC) [34] or Bayesian information criterion (BIC) [35], is often applied, which incurs an exhausted search process in a large parameter space so that a clustering algorithm needs to run many times with different parameter settings to obtain a set of optimal parameters. As most of effective clustering algorithms, e.g., HC, have a computational complexity of $O(dN^2)$ in general, the trial-and-test process results in a computational complexity of $O(dDN^2)$, where D is the dimension of all parameter space. When some parameters take continuous values, D could be infinitely large. When one uses a composite representation where different representations are lumped up together, i.e., d is the sum of all component representation dimensions, the situation becomes even worse. Therefore, traditional clustering algorithms along with model selection criteria [34], [35] are often not efficient and, in particular, sensitive to noisy data.

TABLE II
INFORMATION OF THE BENCHMARK TIME SERIES DATASETS [36]

Data Set	Number of Class	Size of Data set (training+testing)	Length
Synthetic Control	6	300+300	60
Gun-Point	2	50+150	150
CBF	3	30+900	128
Face (all)	14	560+1,690	131
OSU Leaf	6	200+242	427
Swedish Leaf	15	500+625	128
50Words	50	450+455	270
Trace	4	100+100	275
Two Patterns	4	1,000+4000	128
Wafer	2	1,000+6,174	152
Face (four)	4	24+88	350
Lightning-2	2	60+61	637
Lightning-7	7	70+73	319
ECG	2	100+100	96
Adiac	37	390+391	176
Yoga	2	300+3,000	426

It is also worth mentioning that the use of different representations to construct a clustering ensemble as suggested in this paper does not suffer from a higher computational burden than other ensemble learning strategies, e.g., combining multiple partitions produced by different clustering algorithms on a single representation [26] or by running a single-clustering algorithm multiple times with different parameters and initialization conditions [27]. Although alternative ensemble learning strategies roughly have the same complexity as ours, the earlier empirical study [33] indicates that our approach often outperforms these strategies used to form a clustering ensemble in both effectiveness and efficiency [26], [27].

IV. SIMULATIONS

In this section, we describe our experimental setting and report simulation results based on a collection of benchmark datasets for time series data mining [36]. Information on these data sets is tabulated in Table II that lists the number of classes, the number of time series in the predivided training and testing subsets and the length of time series in a data-set. Our simulations consist of clustering analysis and clustering-based classification. For each benchmark, clustering analysis works on the whole datasets while clustering-based classification first performs clustering analysis on the training subset and then uses a prototype-based supervised learning method to fulfill test on the testing subset. For clustering analysis, we first apply various types of time series clustering algorithms on the benchmark collection to be the baseline performance. Those time series clustering algorithms are typical ones chosen from three methodologies summarized in Table I. Then, we evaluate the performance of clustering ensemble approach that combines input partitions yielded with various clustering algorithms on different representations during initial clustering analysis in comparison to the baseline performance.

By the given ground truth, *clustering success rate* is defined as the ratio of the number of time series of the same class label

TABLE III
CLUSTERING SUCCESS RATE (%) OF SINGLE MODELS

Data Set	Temporal-Proximity based			Model based	Single Representation based on K-mean			
	K-mean	HC	RPCCL	K-HMM	PCF	DFT	PLS	PDWT
Synthetic Control	67.9	59.5	67.3*	**69.1**	59.3	60.3	66.3	64.7
Gun-Point	50.0	41.9*	46.1	43.8	43.4	45.4	48.5	49.4
CBF	62.6	50.9	**63.2***	60.1	57.6	60.0	60.6	60.5
Face (all)	36.0	31.9	33.6	37.8	30.5	29.5	34.1	33.1
OSU Leaf	37.8	39.1*	32.2	**44.2**	30.6	26.4	33.5	31.7
Swedish Leaf	40.6	35.6	**41.0**	38.6	35.9	34.4	38.4	39.4
50Words	**42.0**	39.5	39.7	40.8	35.4	37.4	39.8	39.0
Trace	48.5	40.2	**53.1***	50.9	40.5	42.0	42.3	45.4
Two Patterns	32.2	29.8	**33.9**	33.1	27.7	26.5	28.6	30.4
Wafer	62.5	53.2	**64.8***	63.9	57.3	59.6	60.5	61.7
Face (four)	66.9	62.7*	65.3*	69.1	50.7	58.6	59.6	60.4
Lightning-2	61.1	**62.0***	56.3	57.7	53.3	53.0	55.8	57.1
Lightning-7	48.4	39.4	**53.0***	51.2	42.5	43.5	49.9	49.1
ECG	69.8	59.4	67.4*	70.3	59.3	60.3	60.4	64.0
Adiac	38.4	30.2	36.3	**38.9**	30.3	31.2	31.9	32.7
Yoga	**51.7**	44.2	50.8	48.5	45.6	48.6	47.6	48.5

TABLE IV
CLUSTERING SUCCESS RATE (MEAN ± STD)% WITH ENSEMBLE LEARNING

Data Set	Ensemble of single representation based on K-mean				Ensemble of multiple representations		
	PCF	DFT	PLS	PDWT	K-mean	RPCCL (*K*=*K**)	RPCCL (*K*>*K**)
Synthetic Control	61.5 ± 0.1	62.3 ± 0.3	67.5 ± 0.2	66.8 ± 0.3	68.8 ± 0.1	69.1 ± 0.1	70.2 ± 0.2
Gun-Point	45.2 ± 0.2	46.8 ± 0.2	49.2 ± 0.2	48.8 ± 0.2	51.8 ± 0.2	56.8 ± 0.2	57.1 ± 0.1
CBF	59.2 ± 0.1	60.3 ± 0.3	62.5 ± 0.2	62.8 ± 0.3	63.8 ± 0.1	66.1 ± 0.2	64.5 ± 0.2
Face (all)	32.1 ± 0.4	30.3 ± 0.2	34.3 ± 0.3	33.9 ± 0.1	35.1 ± 0.2	35.8 ± 0.1	34.9 ± 0.1
OSU Leaf	30.1 ± 0.4	29.3 ± 0.2	32.3 ± 0.3	33.9 ± 0.1	35.2 ± 0.1	34.9 ± 0.2	35.7 ± 0.1
Swedish Leaf	39.2 ± 0.4	38.2 ± 0.2	40.3 ± 0.3	40.9 ± 0.1	41.2 ± 0.2	42.7 ± 0.1	41.9 ± 0.2
50Words	38.7 ± 0.2	39.1 ± 0.2	40.1 ± 0.3	40.7 ± 0.1	41.6 ± 0.1	42.2 ± 0.1	41.2 ± 0.2
Trace	42.1 ± 0.2	42.9 ± 0.2	45.3 ± 0.3	47.9 ± 0.1	50.5 ± 0.2	53.5 ± 0.3	50.8 ± 0.5
Two Patterns	29.1 ± 0.4	30.1 ± 0.2	31.2 ± 0.3	33.1 ± 0.1	36.1 ± 0.3	37.4 ± 0.2	35.2 ± 0.1
Wafer	60.7 ± 0.1	60.1 ± 0.2	62.1 ± 0.2	61.9 ± 0.2	65.1 ± 0.1	67.1 ± 0.2	65.8 ± 0.1
Face (four)	55.0 ± 0.1	60.8 ± 0.2	61.5 ± 0.2	62.8 ± 0.1	65.2 ± 0.2	68.7 ± 0.2	67.1 ± 0.1
Lightning-2	56.1 ± 0.2	58.6 ± 0.1	58.2 ± 0.2	57.9 ± 0.1	60.1 ± 0.2	64.7 ± 0.2	62.1 ± 0.1
Lightning-7	44.1 ± 0.1	46.1 ± 0.1	51.5 ± 0.2	50.8 ± 0.1	53.1 ± 0.1	61.9 ± 0.2	59.2 ± 0.1
ECG	61.8 ± 0.3	63.1 ± 0.2	63.6 ± 0.2	66.9 ± 0.1	69.2 ± 0.1	73.3 ± 0.2	71.1 ± 0.2
Adiac	31.2 ± 0.1	32.6 ± 0.2	32.3 ± 0.1	32.9 ± 0.1	36.2 ± 0.3	38.1 ± 0.1	37.1 ± 0.2
Yoga	50.4 ± 0.4	49.1 ± 0.2	51.2 ± 0.3	50.9 ± 0.1	52.5 ± 0.1	54.5 ± 0.3	52.1 ± 0.2

grouped together into the same cluster to the overall number of time series of the dataset. For clustering-based classification, data in the training subset are used as prototypes after clustering analysis and the classification accuracy on the testing subset is used as a performance index.

In our experiments, all parameters used in four representation, i.e., $|W_{\mathrm{PLS}}| = 8$, $|W_{\mathrm{DWT}}| = 8$, $M = 4$, and $k = 16$ are fixed. For the DWT, we set the decomposition level to three and use only the third-level approximation and first-level detail coefficients to form a feature vector for each segment of time series.

A. Experiment on Clustering Analysis

To achieve the baseline performance, we employ three temporal-proximity-based algorithms, *K*-mean, HC [2]–[4], and RPCCL [32] with the DTW distance [5], and a model-based algorithm, K-mean based on HMM (K-HMM) [6], directly working on time series for clustering analysis. In addition, we apply K-mean to four representations, respectively, to achieve the representation-based baseline performance. It is worth mentioning that HC is capable of automatic model selection for a dataset of arbitrary shapes, while K-HMM is one of the best techniques for time series clustering. Hence, such results reflect the state-of-the-art baseline performance. In the experiments, the genuine class number K^* is used in the K-mean and K-HMM algorithm. For RPCCL, we simply set an initial number of neurons to K, which is greater than K^*, for automatic model selection.

Table III lists clustering results yielded by typical time series clustering algorithms, where results of the temporal-proximity-based K-mean are provided by the benchmark collectors [36]. It is observed from Table III that there is no algorithm that always outperforms others, which indicates the challenge of model selection and proper grouping in clustering time series of high dimensions. In general, RPCCL yields the satisfactory performance as it achieves the best performance on 6 out of 16 datasets, as marked with bold font. Furthermore, it is observed that the representation-based *K*-mean clustering performance is inferior to that of its counterpart directly working on original time series in general. Here, our experimental results demonstrate the essential weakness of traditional representation-based time series clustering and the difficulty in selecting a representation.

For investigating the performance of ensemble learning on multiple representations, we apply the ensemble technique described in Section III-C to representation-based clustering algorithms including K-mean and RPCCL for comparison. Table IV tabulates results of K-mean clustering ensemble on individual and different representations, RPCCL clustering ensemble on different representations.

We first assess the performance without considering model selection. It is evident from Tables III and IV that by using the prior knowledge K^*, K-mean clustering ensembles on single representations always outperform single K-mean models on same representations. Furthermore, the K-mean clustering ensemble on different representations leads to a further improvement. In contrast, our RPCL network ensemble ($K = K^*$) achieves the best performance on 15 out of 16 datasets by comparing with all other clustering ensembles. For further evaluation in term of model selection, we randomly set an initial number of neurons to K on the condition $K > K^*$ with the exactly same setting used in temporal-proximity-based RPCCL in our previous experiment. We observe from Table IV that the RPCL network ensemble performs better on 13 out of 16 datasets.

In comparison with the baseline performance shown in Table III, our RPCL ensemble networks of automatic model selection on different representations achieves better results on 12, 15, 15, and 11 data-sets than K-mean, HC, temporal-proximity-based RPCCL, and K-HMM, respectively. Note that we compare the averaged results obtained by our RPCL ensemble networks with the best results obtained by various temporal clustering approaches.

From Tables III and IV, however, it is also observed that the clustering success rates on *face (all), OSU leaf, Swedish leaf, 50 words, two patterns* and *Adiac* are below 50% by all algorithms used in our simulations. To understand their nature, we have applied various visualization techniques to the aforementioned datasets. Our visualization analysis on five datasets reveals that there are extremely complex cluster structures, i.e., the datasets contain a large number of classes, unbalanced cluster membership, low intracluster dissimilarity, high intercluster similarity, high dimensionality, and arbitrary cluster shapes, which results

TABLE V
CLASSIFICATION ACCURACY (MEAN ± STD)% OF DIFFERENT APPROACHES ON TRAINING AND TESTING SUBSETS

Data Set	RPCL Ensemble Networks on Different Representations (*K*>*K**)				RPCL Ensemble Networks on Different Representations (*K*=*K**)				K-HMM (*K*=*K**)			
	Training	testing1 (mean)	testing2 (1NN)	testing3 (3NN)	Training	testing1 (mean)	testing2 (1NN)	testing3 (3NN)	Training	testing1 (mean)	testing2 (1NN)	testing3 (3NN)
Synthetic Control	69.1 ± 0.2	66.3 ± 0.1	65.1 ± 0.1	67.4 ± 0.1	71.2 ± 0.1	67.9 ± 0.2	66.2 ± 0.2	66.9 ± 0.3	68.8 ± 0.1	65.3 ± 0.3	64.9 ± 0.4	66.0 ± 0.3
Gun-Point	56.4 ± 0.1	51.5 ± 0.1	50.2 ± 0.1	52.1 ± 0.1	58.1 ± 0.2	53.2 ± 0.3	51.7 ± 0.2	52.9 ± 0.1	46.2 ± 0.4	43.3 ± 0.3	42.2 ± 0.2	41.4 ± 0.4
CBF	61.3 ± 0.2	57.9 ± 0.1	54.2 ± 0.2	56.1 ± 0.1	65.1 ± 0.3	62.1 ± 0.1	63.2 ± 0.2	59.1 ± 0.2	59.1 ± 0.3	57.2 ± 0.5	58.9 ± 0.4	54.8 ± 0.2
Face (all)	35.1 ± 0.1	29.5 ± 0.4	31.3 ± 0.2	30.5 ± 0.1	35.4 ± 0.1	31.1 ± 0.4	32.3 ± 0.2	31.3 ± 0.2	38.2 ± 0.3	35.2 ± 0.2	32.9 ± 0.4	33.2 ± 0.3
OSU Leaf	33.8 ± 0.1	30.8 ± 0.2	29.3 ± 0.1	30.2 ± 0.2	35.1 ± 01	31.2 ± 0.4	30.1 ± 0.2	32.2 ± 0.2	46.7 ± 0.4	40.2 ± 0.4	39.5 ± 0.3	36.9 ± 0.4
Swedish Leaf	42.3 ± 0.1	38.1 ± 0.1	39.3 ± 0.1	37.7 ± 0.2	43.8 ± 0.3	39.9 ± 0.2	40.8 ± 0.1	38.5 ± 0.1	40.2 ± 0.2	37.6 ± 0.3	38.6 ± 0.3	36.9 ± 0.1
50Words	41.9 ± 0.2	37.5 ± 0.1	38.9 ± 0.2	36.7 ± 0.1	41.1 ± 0.2	37.3 ± 0.2	37.3 ± 0.1	36.5 ± 0.1	39.3 ± 0.3	35.6 ± 0.4	36.8 ± 0.2	34.7 ± 0.1
Trace	49.2 ± 0.3	47.5 ± 0.3	46.8 ± 0.3	46.1 ± 0.1	52.8 ± 0.3	50.1 ± 0.4	51.2 ± 0.2	49.5 ± 0.3	50.5 ± 0.2	46.6 ± 0.4	47.1 ± 0.2	46.4 ± 0.3
Two Patterns	34.8 ± 0.1	28.4 ± 0.2	29.3 ± 0.2	27.2 ± 0.3	36.1 ± 0.2	31.1 ± 0.2	30.3 ± 0.2	28.8 ± 0.3	32.6 ± 0.2	26.9 ± 0.3	28.1 ± 0.2	25.3 ± 0.4
Wafer	63.2 ± 0.1	58.1 ± 0.1	56.8 ± 0.2	55.2 ± 0.1	68.2 ± 0.1	62.9 ± 0.1	61.1 ± 0.1	59.4 ± 0.2	61.8 ± 0.2	57.3 ± 0.2	56.2 ± 0.4	54.4 ± 0.3
Face (four)	67.9 ± 0.2	63.2 ± 0.2	61.9 ± 0.1	60.2 ± 0.3	69.1 ± 0.1	64.1 ± 0.1	62.2 ± 0.2	61.9 ± 0.3	71.8 ± 0.3	66.9 ± 0.2	64.2 ± 0.2	63.7 ± 0.5
Lightning-2	58.5 ± 0.1	53.9 ± 0.1	52.8 ± 0.2	54.2 ± 0.1	63.1 ± 0.2	60.7 ± 0.2	58.1 ± 0.3	59.6 ± 0.2	56.6 ± 0.2	52.5 ± 0.2	52.1 ± 0.3	50.6 ± 0.3
Lightning-7	58.1 ± 0.1	52.2 ± 0.1	53.2 ± 0.3	50.2 ± 0.2	60.1 ± 0.2	58.1 ± 0.2	57.1 ± 0.2	57.6 ± 0.2	52.8 ± 0.3	50.0 ± 0.2	51.1 ± 0.3	49.2 ± 0.4
ECG	67.5 ± 0.2	62.9 ± 0.2	61.1 ± 0.2	60.8 ± 0.1	70.4 ± 0.2	65.8 ± 0.2	63.1 ± 0.2	60.9 ± 0.1	71.3 ± 0.2	65.2 ± 0.3	63.9 ± 0.3	61.6 ± 0.2
Adiac	35.6 ± 0.1	29.7 ± 0.2	30.2 ± 0.1	30.6 ± 0.2	37.0 ± 0.1	30.9 ± 0.1	32.1 ± 0.2	29.9 ± 0.2	40.3 ± 0.3	35.3 ± 0.4	36.2 ± 0.3	36.9 ± 0.4
Yoga	50.9 ± 0.2	49.1 ± 0.2	48.1 ± 0.1	45.9 ± 0.2	53.2 ± 0.3	50.2 ± 0.3	51.1 ± 0.3	48.2 ± 0.2	47.2 ± 0.4	46.2 ± 0.2	45.2 ± 0.2	44.6 ± 0.3

in failures for existing clustering algorithms including ours. Moreover, our empirical studies along with others [21], [32], [33], [37]–[40] indicate that our RPCL network ensemble on four representations described in Section II performs well for only the data with balanced and a small number of clusters structures, which is caused by the limitation of the coarse representations and RPCL network.

B. Experiments on Clustering-Based Classification

To evaluate the generalization capability of the RPCL ensemble network, we also conduct experiments for clustering-based classification. For comparison, the K-HMM clustering ensemble is also applied on the same condition. After clustering analysis, all the time series in the training subset are labeled as follows: time series grouped into a cluster are assigned the same label. The labeled time series in the training subset are used as prototypes for instance learning. In our simulations, we use two prototype-based supervised learning methods for classification, i.e., centroid-based and K nearest neighbor (KNN)-based classification. Table V lists clustering results on training subset and all testing results in terms of clustering-based classification.

As observed from Table V, our RPCL network ensemble of automatic model selection yields the satisfactory clustering performance. Results on the training subset of each dataset is consistent with those reported in Table IV where the whole dataset are used for clustering analysis, which manifests the scalability of our approach. In terms of generalization, results on the testing subset of each dataset are comparable with the K-mean baseline outcome [36] and the results reported in Table III, although they are the best results achieved based on the whole dataset and some of them also use the genuine class number information. In particular, the results on the testing subset are even better than the baseline performance on the Gun Point and the Lightning-7 datasets. In general, testing results on most of datasets with different instance learning criteria are consistent with clustering analysis on the whole dataset. The favorite results manifest the quality and the robustness of our proposed approach by using a clustering ensemble on different representations despite the fact that results by the KNN test are slightly different from those by the centroid-based measure.

Without considering automatic model selection, we also conducted an experiment by using the genuine class number of each dataset in RPCCL networks on different representations. As shown in Table V, the performance in both clustering analysis on training subsets and clustering-based classification on testing subsets has been improved, which suggests our model performs better as prior knowledge is incorporated.

Furthermore, we make a comparison between the RPCL network ensemble and K-HMM, notably one of the best time series clustering algorithms [6], [41] designed by reaching a synergy between K-mean and HMM for handling temporal correlation effectively during clustering. In our experiment, the K-HMM is trained with the prior knowledge $K = K^*$, and the numbers of states in HMMs are selected with the exhausted search and the best results are reported in Table V. From Table V, our RPCL network ensemble with $K = K^*$ wins on 12 out of 16 datasets by using centroid based, 11 out of 16 datasets with 1-NN and 3-NN tests. Moreover, our RPCL network ensemble with model selection ($K > K^*$) wins on 11 out of 16 datasets by using centroid based, 9 out of 16 datasets with 1-NN and 3-NN tests. The comparative results further suggest that our proposed approach achieves the synergy between RPCCL networks on different representations and the cluster ensemble technique in both model selection and proper grouping. Thus, our approach is suitable for being applied to time series clustering in an unknown environment.

V. DISCUSSION

As one of the most important factors for success, the selection of appropriate different representations for a given dataset would affect the performance of our proposed approach. In our paper, we use four common yet simple time series representations for the demonstration purpose and achieve the favorite results on 16 benchmark time series data mining datasets [36]. While four representations are applicable to time series clustering without prior knowledge, low performance on several datasets, e.g., *Adiac* suggests that exploration of effective yet complementary representations would be a prominent topic to be investigated in our ongoing research.

While the RPCL network offers clear strengths for clustering analysis in terms of efficiency and automatic model selection, there are some inherent weaknesses, e.g., its implicit assumption on the shape of clusters and the limitation in coping with a dataset of many clusters [21], [32]. Apparently, such weaknesses inevitably limit the performance of our proposed RPCL network ensemble on a dataset violating the assumption despite the fact that our ensemble scheme tends to alleviate such adverse effects. Nevertheless, other time series clustering algorithms, more or less, suffer from their own inherent limitation as well. For instance, K-HMM [6], [41], a state-of-the-art time series clustering algorithm, used in our simulations for comparison has an implicit assumption that there exists only the first-order temporal correlation underlying time series and its own model selection problem that needs to specify a HMM architecture with an appropriate number of states and proper connections in advance. From our simulations, it is evident that K-HMM does not yield better performance despite the fact that we use the exhausted search for a reasonable parameter subspace to find the best architecture. In particular, the K-HMM training is rather time consuming and takes a significantly longer time than training the RPCL network ensemble. As a result, how to overcome the limitation of time series clustering analysis is still an open problem to be studied.

On the other hand, our empirical studies further show that the use of the RPCL network results in twofold 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 competitive learning rules seems to hinder generating truly diverse partitions for time series of complicated structures. Although the use of different learning rates, initialization, and the termination conditions leads to different partitions, the correlation among them is often quite high. To our knowledge, there has been no theoretic analysis available so far regarding combining the highly correlated partitions for clustering analysis. Nevertheless, the theoretic analysis in supervised ensemble learning suggests that combining highly correlated classifiers is unlikely to yield the considerate improvement in classification [42]. In our ongoing research, we would investigate these issues to overcome the limitation of RPCL networks for constructing effective clustering ensembles.

As pointed out previously, clustering ensemble ideas provide an underpinning yet enabling techniques for overcoming the weakness of representation-based time series clustering. Although the cluster ensemble technique [26] yields the relatively satisfactory performance in combining RPCL networks working on different representations, the further improvement can be made. A fundamental weakness in the cluster ensemble [26] is that different partitions are treated equally during reconciliation, while different partitions contain various amount of information. When RPCL networks produce highly correlated yet inaccurate partitions, the final partition achieved by the cluster ensemble is biased to those most correlated partitions. In our ongoing work, we have been developing a novel clustering ensemble algorithm to overcome this fundamental weakness.

VI. Conclusion

In this paper, we have presented an unsupervised ensemble-learning model for time series clustering by combining RPCL networks on different representations. Without the use of prior information on a given dataset, our model yields favorite results on the benchmark time series benchmark datasets [36]. Here, we emphasize that the synergy of three compulsory components results in the satisfactory performance of our proposed approach. In particular, the joint use of different representations in our approach leads to the significant improvement. Our approach neither uses the prior knowledge on time series nor needs a tedious parameter-tuning process. As a consequence, our proposed approach provides a practical yet effective way for time series clustering analysis.

Acknowledgment

The authors would like to thank the anonymous reviewers for their comments that improve the presentation of this paper. They would also like to thank Y. M. Cheung for providing his rival penalization controlled competitive learning [32] code used in our simulations. The source code implementing the cluster ensemble algorithm [26] was downloaded from authors' Web site, and a modified version was used in our experiments.

References

[1] J. Kleinberg, "An impossible theorem for clustering," in *Proc. Advances in Neural Information Processing Systems*, vol. 15, 2002, pp. 446–453.

[2] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical," *Knowl. Data Discov.*, vol. 6, pp. 102–111, 2002.

[3] A. Jain, M. Murthy, and P. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, pp. 264–323, 1999.

[4] R. Xu and D. Wunsch II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[5] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time Series data: experimental comparison of representations and distance measures," in *Proc. VLDB*, 2008, pp. 1542–1552.

[6] P. Smyth, "Probabilistic model-based clustering of multivariate and sequential data," in *Proc. Int. Workshop Artif. Intell. Statist.*, 1999, pp. 299–304.

[7] K. Murphy, "Dynamic Bayesian networks: Representation, inference and learning," Ph.D. dissertation, Dept. Comp. Sci., Univ. Calif., Berkeley, CA, 2002.

[8] Y. Xiong and D. Yeung, "Mixtures of ARMA models for model-based time series clustering," in *Proc. IEEE Int. Conf. Data Ming*, 2002, pp. 717–720.

[9] H. Liu and D. E. Brown, "A new point process transition density model for space-time event prediction," *IEEE Trans. Syst., Man, Cybern. (Part C: Rev. Appl.)*, vol. 34, no. 3, pp. 310–324, Aug. 2004.

[10] S. Policker and A. B. Geva, "Nonstationary time series analysis by temporal clustering," *IEEE Trans. Syst., Man, Cybern. (Part B: Cybern.)*, vol. 30, no. 2, pp. 339–343, Apr. 2000.

[11] N. Dimitova and F. Golshani, "Motion recovery for video content classification," *ACM Trans. Inf. Syst.*, vol. 13, pp. 408–439, 1995.

[12] W. Chen and S. Chang, "Motion trajectory matching of video objects," in *Proc. SPIE/IS&T Conf. Storage Retrieval Media Database*, 2000, pp. 544–553.

[13] C. Faloutas, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Proc. ACM SIGMOD Conf.*, 1994, pp. 419–429.

[14] E. Sahouria and A. Zakhor, "Motion indexing of video," in *Proc. IEEE Int. Conf. Image Process.*, 1997, vol. 2, pp. 526–529.

[15] C. Cheong, W. Lee, and N. Yahaya, "Wavelet-based temporal clustering analysis on stock time series," in *Proc. ICOQSIA*, 2005, pp. 113–121.

[16] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrota, "Locally adaptive dimensionality reduction for indexing large scale time series databases," in *Proc. ACM SIGMOD Conf.*, 2001, pp. 151–162.

[17] F. Bashir, "MotionSearch: Object motion trajectory-based video database system—Index, retrieval, classification and recognition," Ph.D. dissertation, Dept. Elect. Eng., Univ. Illinois, Chicago, IL, 2005.

[18] E. Keogh and M. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2001, pp. 122–133.

[19] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. SIGKDD*, 2009, pp. 947–956.

[20] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos, "Iterative incremental clustering of time series," presented at the IX Conf. Extending Database Technology, Crete, Greece, 2004.

[21] L. Xu, A. Krzyzak, and E. Oja, "Rival penalized competitive learning for clustering analysis, RBF net, and curve detection," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 636–648, Jun. 1993.

[22] K. Chen, L. Wang, and H. Chi, "Methods of combining multiple classifiers with different feature sets and their applications to text-independent speaker identification," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 11, pp. 417–445, 1997.

[23] K. Chen and H. Chi, "A method of combining multiple probabilistic classifiers through soft competition on different feature sets," *Neurocomputing*, vol. 20, pp. 227–252, 1998.

[24] K. Chen, "On the use of different speech representations for speaker modeling," *IEEE Trans. Syst., Man, Cybern. (Part C: Rev. Appl.)*, vol. 35, no. 3, pp. 301–314, Aug. 2005.

[25] S. Wang and K. Chen, "Ensemble learning with active data selection for semi-supervised pattern classification," in *Proc. Int. J. Conf. Neural Netw.*, 2007, pp. 355–360.

[26] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learning Res.*, vol. 3, pp. 583–617, 2002.

[27] A. Fred and A. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.

[28] P. C. Chang and C. Y. Fan, "A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting," *IEEE Trans. Syst., Man, Cybern. C*, vol. 38, no. 6, pp. 802–815, Nov. 2008.

[29] O. Renaud, J. Starck, and F. Murtagh, "Wavelet-based combined signal filtering and prediction," *IEEE Trans. Syst., Man, Cybern. (Part B: Cybern.)*, vol. 35, no. 6, pp. 1241–1251, Dec. 2005.

[30] J. Lin, E. Keogh, S. Lonardi, and P. Patel, "Finding motifs in time series," in *Proc. Workshop Temporal Data Mining (KDD2002)*, pp. 53–68.

[31] J. Sammon, Jr., "A nonlinear mapping for data structure analysis," *IEEE Trans. Comput.*, vol. C-18, no. 5, pp. 401–409, May 1969.

[32] Y. Cheung, "On rival penalization controlled competitive learning for clustering with automatic number," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1583–1588, Nov. 2005.

[33] Y. Yang and K. Chen, "Combining competitive learning networks of various representations for temporal data clustering," in *Trends in Neural Computation*. Berlin, Germany: Springer-Verlag, 2007, ch. 13, pp. 315–336.

[34] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 2005.

[35] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 641–646, 1978.

[36] E. Keogh, Temporal data mining benchmarks. (2005). [Online]. Available: http://www.cs.ucr.edu/∼eamonn/time_series_data

[37] X. Hu and L. Xu, "A comparative study of several cluster number selection," in *Proc. IDEAL2003*, pp. 195–202.

[38] Y. Cheung, "Maximum weighted likelihood via rival penalized EM for density mixture clustering and with automatic model selection," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 750–761, Jun. 2005.

[39] J. Ma and T. Wang, "A cost-function approach to rival penalized competitive learning," *IEEE Trans. Syst., Man, Cybern. (Part B: Cybern.)*, vol. 36, no. 4, pp. 722–737, Aug. 2006.

[40] Q. Lu and X. Yao, "Clustering and learning Gaussian distribution for continuous optimization," *IEEE Trans. Syst., Man, Cybern. (Part C: Rev. Appl.)*, vol. 35, no. 2, pp. 195–204, May 2005.

[41] A. Panuccio, M. Bicego, and V. Murino, "A hidden Markov model-based approach to sequential data clustering," in *Structural, Syntactic and Statistical Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2002, pp. 734–742.

[42] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers," *Connect. Sci.*, vol. 8, pp. 385–404, 1996.

[43] A. Panuccio, M. Bicego, and V. Murino, "A hidden Markov model-based approach to sequential data clustering," in *Structural, Syntactic and Statistical Pattern Recognition* (Lecture Notes in Computer Science 2396), T. Caelli, A. Amin, R. Duin, M. Kamel, and D. de Ridder, Eds. New York: Springer-Verlag, 2002, pp. 734–742.

[44] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers," *Connect. Sci.*, vol. 8, pp. 385–404, 1996.

Yun Yang received the B.Sc. degree (First Class Honors) from Lancaster University, Lancaster, U.K., in 2004, the M.Sc. degree from Bristol University, Bristol, U.K., in 2005, and the MPhil degree from the School of Computer Science, The University of Manchester, Manchester, U.K., in 2006. He is currently working toward the Ph.D. degree at The University of Manchester.

His current research interests include pattern recognition and machine learning.

Ke Chen (M'97–SM'00) received the B.Sc., M.Sc., and Ph.D. degrees in 1984, 1987 and 1990, respectively, all in computer science.

Since 2003, he has been with the School of Computer Science, The University of Manchester, Manchester, U.K. He was with The University of Birmingham, Peking University, The Ohio State University, Kyushu Institute of Technology, and Tsinghua University. In 2000, he was a Visiting Professor at Microsoft Research Asia. In 2001, he was with Hong Kong Polytechnic University. He is the author or coauthor of more than 100 academic papers published in refereed journals and conferences. His current research interests include pattern recognition, machine learning, machine perception, and computational cognitive systems.

Dr. Chen has been on the Editorial Board of several academic international journals including the IEEE TRANSACTIONS ON NEURAL NETWORKS 2005–2010 and serves as the Category Editor of the *Machine Learning and Pattern Recognition in Scholarpedia*. He has served as a Technical Program Cochair of several international conferences, including the International Joint Conference on Neural Networks (IJCNN) 2012 and the International conference on Natural Computation 2005, and a member of the Technical Program Committee of numerous international conferences such as the Cognitive Science Society and the IJCNN. During 2007–2009, he chaired the Intelligent Systems Applications Technical Committee and the University Curricula Subcommittee, the IEEE Computational Intelligence Society (IEEE CIS). He also served as the Task Force Chair and a member of the Neural Networks Technical Committee, the Emergent Technologies Technical Committee, and the Data Mining Technical Committee in the IEEE CIS. He is the recipient of several academic awards including the National Natural Science Foundation of China Distinguished Principal Young Investigator Award and the Japan Society for the Promotion of Science Research Award.