Adaptive Bi-Weighting Toward Automatic Initialization and Model Selection for HMM-Based Hybrid Meta-Clustering Ensembles

Yun Yang and Jianmin Jiang

***Abstract*—Temporal data clustering can provide underpinning techniques for the discovery of intrinsic structures, which proved important in condensing or summarizing information demanded in various fields of information sciences, ranging from time series analysis to sequential data understanding. In this paper, we propose a novel hidden Markov model (HMM)-based hybrid meta-clustering ensemble with bi-weighting scheme to solve the problems of initialization and model selection associated with temporal data clustering. To improve the performance of the ensemble techniques, the proposed bi-weighting scheme adaptively examines the partition process and hence optimizes the fusion of consensus functions. Specifically, three consensus functions are used to combine the input partitions, generated by HMM-based *K*-models under different initializations, into a robust consensus partition. An optimal consensus partition is then selected from the three candidates by a normalized mutual information-based objective function. Finally, the optimal consensus partition is further refined by the HMM-based agglomerative clustering algorithm in association with dendrogram-based similarity partitioning algorithm, leading to the advantage that the number of clusters can be automatically and adaptively determined. Extensive experiments on synthetic data, time series, and real-world motion trajectory datasets illustrate that our proposed approach outperforms all the selected benchmarks and hence providing promising potentials for developing improved clustering tools for information analysis and management.**

***Index Terms*—Data clustering, ensemble learning, hidden Markov model (HMM), model selection.**

I. Introduction

TEMPORAL data clustering has been recognized as an important research field of data mining, it aims to divide an unlabeled temporal dataset into groups or clusters, where coherent or homogeneous information [1] can be revealed. As the rapid growth of temporal data, various temporal data clustering algorithms have been developed from different perspectives [2]. Some algorithms [3]–[9] intend to partition raw temporal data, or the feature vectors extracted from them in form of static data, and hence called the proximity-based approach and feature-based approach. Other algorithms [10]–[21] are proposed to model the generation of temporal data, and identify the cluster structures of temporal data via determining the model structures and parameters. Such algorithms are often called model-based approach.

In model-based approach, each cluster can be mathematically represented by a parametric model, such as Gaussian smodel [21], hidden Markov model (HMM) [10], [11], autoregressive moving-average model (ARMA) [12], [13], mixture of Markov chain [14]–[16], fuzzy-based estimation model [17]. Among them, HMM-based clustering approaches have been widely studied for the last decade. Its earlier work [22], [23] focused on speech recognition. Recently it has been successfully expanded into general temporal data clustering applications [10], [11], [18]–[20] due to its superiority in capturing dynamic behaviors of temporal data. However, many HMM-based clustering algorithms still suffer from the critical problem of model selection, i.e., detecting the intrinsic number of clusters and initialization sensitivity.

Existing research on model selection remains active, and many algorithms have been reported in [24]–[29], though no universal model selection has been reported and well accepted for general clustering tasks. This is due to the fact that clustering performances are highly dependent on the matching between cluster structures of the target dataset and the selection of clustering techniques. Regarding the way of determining the number of clusters, existing approaches can be classified into two categories: 1) external determination and 2) internal determination. In the first category, the number of clusters is externally determined by optimizing the predefined criterion, such as Akaike information criterion [24], Bayesian information criterion (BIC) [25], and minimizing description length [16]. Recent empirical studies on model selection [30], [31], however, reveal that most of the existing criteria has limitations, leading to the problem that the cluster number is either over estimated or under estimated. In the second category, clustering algorithm

Manuscript received March 4, 2016; revised September 14, 2017, January 31, 2018, and February 12, 2018; accepted February 21, 2018. Date of publication March 27, 2018; date of current version March 5, 2019. This work was supported in part by the Natural Science Foundation China under Grant 61620106008, Grant 61402397, and Grant 61663046, in part by the Shenzhen Commission for Scientific Research and Innovations under Grant JCYJ20160226191842793, in part by the Yunnan Applied Fundamental Research Project under Grant 2016FB104, in part by the Yunnan Provincial Young Academic and Technical Leaders Reserve Talents under Grant 2017HB005, and in part by the Yunnan Provincial Innovation Team project under Grant 2017HC012. This paper was recommended by Associate Editor Y. Jin. *(Corresponding author: Jianmin Jiang.)*

Y. Yang is with the National Pilot School of Software, Yunnan University, Kunming 650000, China (e-mail: yangyun@ynu.edu.cn).

J. Jiang is with the Research Institute for Future Media Computing, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: jianmin.jiang@szu.edu.cn).

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/TCYB.2018.2809562

itself is able to gradually update the structure of clusters during an iterative learning process, and the number of clusters can be internally determined until a stop criterion is reached. Typical examples include DBSCAN [26], adaptive K-means [27], fuzzy adaptive clustering [28], and adaptive fuzzy C-means clustering [29]. Although such approaches have shown promising results to an extent, most of them still suffer from the problem of initialization sensitivity.

In fact, ensemble learning techniques [32]–[39] were originally developed for classification tasks. It is recently applied to the clustering tasks [10], [40]–[43], and attempt to improve the robustness of clustering by combining multiple clustering into a single consensus solution, which normally achieves better results in terms of average performance among input single clustering solutions, leading to a potential solution for the initialization problem. Although such techniques have been intensively studied, it is still a serious challenge to harmonically combine the various clustering solutions into an optimal ensemble clustering without supervised information.

Motivated by our early studies [2], [5], [6], [44], we propose in this paper an HMM-based hybrid meta-clustering ensemble model with a bi-weighting scheme. In our approach, the ensemble technique is used to tackle the initialization problem, which is caused by HMM-based K-models during initial clustering analysis. Our proposed ensemble model achieves an optimal reconciliation of input partitions via the proposed bi-weighting scheme, where the weights to the partitions and clusters are, respectively, assigned in accordance with their level of importance. Further, the HMM-based agglomerative clustering is applied to improve the consensus solution with automatic model selection by introducing the concept of dendrogram-based similarity partitioning algorithm (DSPA) [5]. In summary, our contributions reported in this paper can be highlighted as follows.

1) We propose a novel HMM-based hybrid meta-clustering ensemble approach to solve both the initialization problem and the model selection problem, in order to keep the benefits of using hybrid approaches for temporal data clustering.
2) We propose a novel bi-weighting scheme to optimally reconcile the input partitions into a single consolidated clustering solution, where both the partition and the cluster weights are intrinsically derived from the objective function of the clustering ensemble without any prior information.

The rest of this paper is organized as follows. Section II presents the most common model-based clustering methods, and overview of HMM-based clustering. Section III describes the motivation and our approach, together with the details of major techniques developed. Section IV reports the experimental results on various temporal datasets. Section V discusses the issues related to our approach, and finally, the conclusions are drawn in Section VI.

II. Related Work

In this section, we review the most common model-based clustering algorithms, and give an overview of the HMM-based clustering.

A. Model-Based Clustering

For model-based clustering algorithm, it is very important to select an appropriate model for target clustering tasks, such as Gaussian model, HMM, ARMA, mixture of Markov chain. However, the model type is always specified as prior information, and need to be predefined as a use-input for model-based algorithms.

Gaussian mixture models are popular among model-based approaches for a so-called speaker verification. The fuzzy c-means clustering-based normalization method [21] is one example in finding a better score to be compared with a given threshold for accepting or rejecting a claimed speaker. It overcomes the drawback of assuming equal weight for all the likelihood values of the background speakers in current normalization methods.

HMM-based clustering have been widely studied from two categories of approaches: 1) partitioning approach, such as HMM-based K-models [11] and 2) hierarchical approach, such as HMM-based agglomerative clustering [18] and HMM-based divisive clustering [19]. In addition to such standard approaches, HMM-based hybrid partitioning-hierarchical clustering and its variants [20] have also been reported to exploit the strengths of partitioning and hierarchical approaches.

ARIMA model [45] is originally designed as a combination of three types of temporal processes, including autoregressive, integrated, and moving average processes. While a stationary ARIMA model with autoregressive and moving average order is also known as ARMA model. In this paper [12], a mixture of ARMA models is commonly used for clustering time series. It is assumed that the time series are generated by k different ARMA models, where each component model corresponds to one cluster of interest, and an EM algorithm is derived for learning the mixing coefficients as well as the parameters of the component models.

Finite mixtures of Markov chains [14], [16] have also been proposed for clustering time series. The number of clusters can be determined by comparing different choices based on some scoring scheme. One possibility, used by Cadez *et al.* [16], is related to minimizing the description length. Another approach [15] to the clustering of time series modeled by Markov chains is called Bayesian clustering by dynamics. In this approach, each time series is initially mapped into a Markov chain, with its dynamics represented simply by a transition probability matrix. It then goes through an agglomerative procedure by trying to merge the two closest Markov chains at each step, using the Kullback–Leibler (KL) distance measure [46] between transition probability matrices. Based on a greedy heuristic search approach, this procedure continues until the resulting model is found to be less probable than the model before merging. Thus, the number of clusters can be determined automatically.

B. HMM-Based Clustering

HMM has outstanding ability to model the dynamic behaviors of temporal data. It is defined as an unobservable stochastic process consisting of a finite number of states, each of which is related to another stochastic process

that emits observations. As a production process of HMM-generated data, an observation o_t is initially emitted with an emission probability b_{it} at the state i, which is selected according to the initial probability π_i. The next state j is decided by the state transition probability a_{ij}, and an observation o_k is also generated based on an emission probability b_{jk} at the state j. The process repeats until a finite number of observations are generated. Essentially, the entire process produces a sequence of observations instead of the states, from which the name "hidden" is drawn. The complete set of HMM model parameters is described by a triplet $\lambda = \{\pi, A, B\}$, where $\pi = \{\pi_i\}, A = \{a_{ij}\}, B = \{b_i\}$ representing the initial probability, the state transition probability, and the emission probability. For continuously valued temporal datasets, such as time series, the emission probability of each state can be defined by a multivariate Gaussian distribution. Without losing generality, we define the emission distribution function of continuously valued temporal data as a single Gaussian distribution $b_i = \{\mu_i, \sigma_i^2\}$ in order to reduce the computational cost and prevent the risk of over-fitting. As a result, the temporal datasets can be modeled as a set of K HMMs $\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ with S states based on such single Gaussian distributed observations. Each component consists of the following parameters.

1) A S-dimensional initial state probability vector π.
2) A $s \times s$ state transition matrix A.
3) Mean vector $\{\mu_1, \mu_2, \ldots, \mu_S\}$.
4) Variance vector $\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_S^2\}$.

In HMM-based K-models [2], the entire dataset $X = \{x_n\}_{n=1}^N$ is represented as a set of K HMMs $\{\lambda_k\}_{k=1}^K$. Initially the parameters of K HMMs are estimated on K data points, which are randomly selected from X without any replacement. Then, forward and backward algorithm [47], [48] is applied to calculate the log-likelihood of each data point under K HMMs, and assign data points to the HMM with maximum value of log-likelihood. After that, the parameters of K HMMs are re-estimated on the corresponding group of data points by EM algorithm [49]. The entire process is repeated until the cluster memberships no longer change.

In HMM-based agglomerative clustering [18], N HMMs $\{\lambda_n\}_{n=1}^N$ are initially trained on the entire dataset $X = \{x_n\}_{n=1}^N$. The closest pair of clusters is then iteratively merged into a new cluster, and the parameters of HMM representing the new cluster are re-estimated by EM algorithm until a stop criterion is reached. In this approach, a symmetric version of KL distance [46] is defined as

$$\begin{aligned} D_S(\lambda_i, \lambda_j) &= \frac{1}{2}\left[D_{\mathrm{KL}}(\lambda_i, \lambda_j) + D_{\mathrm{KL}}(\lambda_j, \lambda_i)\right] \\ &= \frac{1}{2}\sum_x p(\lambda_i)\left[\log p(x|\lambda_i) - \log p(x|\lambda_j)\right. \\ &\qquad \left. + \log\ p(x|\lambda_j) - \log\ p(x|\lambda_i)\right]. \quad (1) \end{aligned}$$

HMM-based hybrid clustering [20] is proposed by combining partitioning and hierarchical approaches. In this approach, the whole dataset is initially partitioned into K clusters (K is normally greater than the intrinsic number of clusters K^*) by HMM-based K-models. The initial K clusters are then treated as the input of HMM-based agglomerative clustering, where the close clusters are iteratively merged until a stopping criterion is reached. In association with HMM-based K-models, HMM-based agglomerative clustering significantly reduces its computational cost during its merging process.

HMM-based hybrid meta-clustering is designed to make further improvements on HMM-based hybrid clustering. It treats initial clusters produced by HMM-based K-models as meta-data, and iteratively merges them into meta-clusters represented by composite models via HMM-based agglomerative clustering. During the merging process, parameters of composite models are simply obtained by combining the parameters of their child models without re-estimation. As a result, the computational cost of the agglomerative clustering can be further reduced. In this approach, the distance between two composite models I and J is defined as

$$D(\lambda_I, \lambda_J) = \frac{1}{|\lambda_I| \times |\lambda_J|} \sum_{\lambda_i \in \lambda_I} \sum_{\lambda_j \in \lambda_J} D_S(\lambda_i, \lambda_j) \quad (2)$$

where λ_I and λ_J represent composite model I and J, while λ_i and λ_j are the child model of the composite model I and J, respectively.

III. Proposed Hybrid Meta-Clustering Ensemble With BI-Weighting Scheme

In this section, we first describe the motivation of proposing our approach, and then present an HMM-based hybrid meta-clustering ensemble with a bi-weighting scheme. We then systematically describe the bi-weighted clustering ensemble model as the major component of our approach, supported by a consensus function and normalized mutual-information (NMI) based objective function. Finally, a refinement function with model selection is described in detail to finish the design of our proposed approach.

A. Motivation

As a matter of fact, each of the aforementioned HMM-based algorithms has different limitations for temporal data clustering. HMM-based K-models suffers from both of the initialization and model selection problems, while HMM-based agglomerative clustering becomes infeasible in practical applications for large temporal dataset due to its higher computational cost. Although HMM-based hybrid clustering [20] achieves better performance in comparison with both of HMM-based K-models and agglomerative approaches, it still cannot avoid the initialization problem during the initial partitioning process, leaving the major problem of model selection unresolved. As an improved version of HMM-based hybrid clustering, HMM-based hybrid meta-clustering gains benefits from adopting composite models. Such composite models are not only good at capturing the complex structures of clusters, but also require no re-estimations for the model parameters. However, the model selection and initialization problems are still unsolvable in such an approach.

Empirical studies [50]–[54] support that, generally, ensemble techniques outperform the single clustering algorithm, and

provide a potential solution for the initialization problem. In our earlier work [5], the essence of the clustering ensemble is explored in depth as such that a "mean" partition of the input partitions can be taken as the consensus if all possible partitions of the target dataset are known. In practice, only subsets of partitions are given, and hence the weighted mean of input partitions is appropriate for a consensus solution, where the weights of input partitions are normally determined by clustering quality, which is often referred to as the amount of contributions to the consensus solution. Without providing the priori labeling information about the target dataset, however, it is not possible to precisely evaluate the clustering quality of input partitions.

In this paper, we try to solve both of the initialization problem and the model selection problem by proposing an HMM-based meta-clustering ensemble model with a bi-weighting scheme. To solve the initialization problem, an optimal reconciliation of input partitions is achieved by using the proposed bi-weighting scheme. Essentially, such a weighting scheme not only assigns weights to input partitions based on the clustering quality, but also assign the weights to clusters based on their dominance in the corresponding partition. Given a formal analysis described in Section III-C1, both weights can be intrinsically derived from the objective function of clustering ensemble. To automatically detect the number of clusters and hence strengthen the solution for the model selection problem, a refinement function is further introduced, where the optimal consensus partition is refined by the HMM agglomerative clustering algorithm in association with DSPA. Due to the use of hybrid approach, the computational cost of DSPA is significantly reduced to $O(K^{o^2})$ in comparison with our previous work [5], where DSPA is directly applied to a $N \times N$ similarity matrix, and result in a computation cost of $O(N^2)$. Here, N is the number of the data points, and K^o is the number of clusters resulted in the optimal consensus partition, $N >> K^o$.

B. Description of Our Proposed Approach

As illustrated in Fig. 1, our proposed approach consists of three modules, including initial clustering analysis, bi-weighted clustering ensemble, and final refinement function with model selection, details of which are highlighted as follows.

1) In initial clustering analysis module, diverse partitions of target dataset are generated by HMM-based K-models clustering algorithm with different initializations, where the cluster number is randomly selected from a preset range for each partition.
2) In bi-weighted clustering ensemble module, these initially generated partitions are, respectively, combined into consensus partitions by three consensus functions [cluster-based similarity partitioning algorithm (CSPA), hypergraph-partitioning algorithm (HGPA) and meta-clustering algorithm (MCLA)] [51] with bi-weighting scheme, then an NMI-based objective function is adopted to select the optimal one from three consensus partitions.

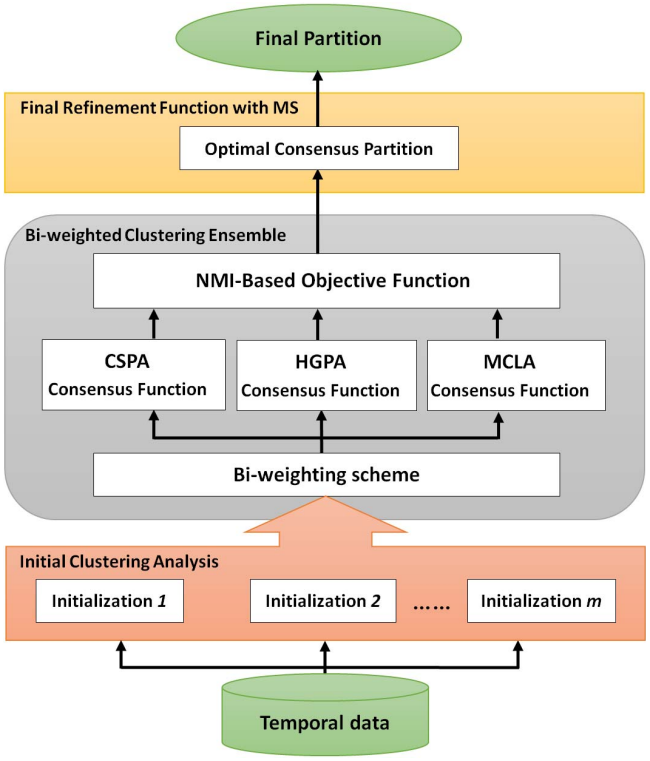


Fig. 1. Overview of our proposed approach.

3) In final refinement function with model selection module, a final partition is then obtained by further refining optimal consensus partition in a hierarchical structure, where the closest pair of clusters is merged as a composite model based on the distance defined in (2). In association with DSPA, such merging process is repeated until the final partition with the intrinsic structure of clusters is obtained.

C. Bi-Weighted Clustering Ensemble

To ensure that both clusters and their partitions make constructive contributions to the consensus solution according to their level of importance, we introduce a bi-weighting scheme to optimize the integration of input partitions. In this way, not only the weights are globally assigned to the input partitions based on their clustering quality, but also the local assignment of weights can be made adaptive to their dominance of clusters that produces the corresponding partition.

1) Bi-Weighting Scheme: Given a distance D, a consensus function of the weighted clustering ensemble is essentially to find a consensus partition P^r close to multiple input partitions $\{P^m\}_{m=1}^M$, which are obtained from the target dataset $\{x_n\}_{n=1}^N$. Therefore, the consensus function can be formulated as minimizing the following loss function [55]:

$$L = \sum_m w_m D\left(P^m, P^r\right) \tag{3}$$

where w_m refers to the weight of partition P^m, $\sum_m w_m = 1$.

In model-based clustering, each input partition P^m can be mathematically represented as a mixture of probability distributions $p_m(x_n) = \sum_{k_m} p(k_m)p(x_n|\lambda_{k_m}^m)$, where $\{\lambda_{k_m}^m\}_{k_m=1}^{K_m}$ are the mixture model parameters, K_m is the number of clusters $c_{k_m}^m(x) = p(x|\lambda_{k_m}^m)$ resulting in each of the input partitions, and $p(k_m)$ is the prior probability. Based on the KL distance [46], the loss function given in (3) can be further derived as

$$\begin{aligned}
L &= \sum_m w_m D_{\mathrm{KL}}(p_m, p_r) \\
&= \sum_m w_m \sum_n p_m(x_n) \log \frac{p_m(x_n)}{p_r(x_n)} \\
&= \sum_m w_m \sum_n \sum_{k_m} p(k_m) p\big(x_n|\lambda_{k_m}^m\big) \log \frac{p_m(x_n)}{p_r(x_n)} \\
&= \sum_m w_m \sum_{k_m} p(k_m) \Bigg[\sum_n p\big(x_n|\lambda_{k_m}^m\big) \log p_m(x_n) \\
&\qquad - \sum_n p\big(x_n|\lambda_{k_m}^m\big) \log p_r(x_n) \Bigg] \\
&= \sum_m w_m \sum_{k_m} p(k_m) \big[-H\big(c_{k_m}^m, p_m\big) + H\big(c_{k_m}^m, p_r\big) \big] \\
&= \sum_m w_m \sum_{k_m} p(k_m) \big[-H\big(c_{k_m}^m\big) - D_{\mathrm{KI}}\big(c_{k_m}^m, p_m\big) \\
&\qquad + H\big(c_{k_m}^m\big) + D_{\mathrm{KI}}\big(c_{k_m}^m, p_r\big) \big] \\
&= \sum_m w_m \sum_{k_m} p(k_m) \big[-D_{\mathrm{KI}}\big(c_{k_m}^m, p_m\big) + D_{\mathrm{KI}}\big(c_{k_m}^m, p_r\big) \big] \\
&= \underbrace{\sum_m w_m \sum_{k_m} p(k_m) \big[-D_{\mathrm{KI}}\big(c_{k_m}^m, p_m\big) \big]}_{L_1} + \underbrace{\sum_m w_m \sum_{k_m} p(k_m) \big[D_{\mathrm{KI}}\big(c_{k_m}^m, p_r\big) \big]}_{L_2}
\end{aligned} \tag{4}$$

where $H(p) = H(X) = -\sum_n p(x_n) \log p(x_n)$ is the Shannon's entropy [56] for $X = \{x_n\}_{n=1}^N$. According to the information theory, it measures uncertainty of a system, the bigger the value of the entropy, the less similarity between the system members. $H(p, q) = -\sum_n p(x_n) \log q(x_n)$ is the cross entropy between two probability distributions p and q. The KL distance between p and q can also be defined as $D_{\mathrm{KL}}(p, q) = H(p, q) - H(p)$.

The loss function in (4) can be decomposed into two separate loss functions L_1 and L_2, which manifest that the performance of a clustering ensemble depends on both the quality of input partitions and the clustering ensemble. As the first term L_1 corresponds to the quality of input partitions, a smaller value of L_1 indicates a better quality of input partitions. Indeed, the objective of clustering is to separate the dataset into different groups or clusters as such that the data points inside the same cluster should be less dissimilar, where the dissimilarity is determined by an intracluster distance, and the dissimilarity across the different clusters should be larger, determined by an intercluster distance. By taking a close look at $D_{\mathrm{KI}}(c_{k_m}^m, p_m) = H(p_m) - H(c_{k_m}^m)$ shown in (4), we realize that $H(p_m)$ just refers to the dissimilarity of clusters resulting in partition P^m corresponding to intercluster distance, while $H(c_{k_m}^m)$ refers to the dissimilarity of data points inside the cluster $C_{k_m}^m$ corresponding to intracluster distance. As a result, the clustering quality of input partition P^m can be quantified as

$$Q_m = \sum_{k_m} p_r(k_m) \Big[-D_{\mathrm{KI}}\Big(c_{k_m}^m, p_m\Big) \Big]. \tag{5}$$

The smaller value of Q_m indicates a better quality of input partition, where the intracluster distance should be small and the intercluster distance should be large.

Intuitively, the partition weights could be determined by minimizing L_1, where larger weights should be assigned to the better quality partitions as determined by smaller value of Q_m. However, such simple scheme eventually allocates a single maximum weight to the input partition with the smallest value of Q_m, and all other weights are set to zero. Under this circumstance, the consensus function is turned into a selection function. To prevent such situation and make all input partitions contribute to the consensus solution, we introduce a regularization term $w_m \log w_m$ [57], which represents the negative entropy of partition weights, into L_1 to form a regularized loss function

$$\begin{aligned}
L_3 &= \sum_m \Bigg[w_m \sum_{k_m} p_r(k_m) \big[-D_{\mathrm{KI}}\big(c_{k_m}, p_m\big) \big] + \alpha w_m \log w_m \Bigg] \\
&= \sum_m \big[w_m Q_m + \alpha w_m \log w_m \big]
\end{aligned} \tag{6}$$

where $\alpha \geq 0$ is a coefficient that controls the strength of the added regularization term, and increasing its value will whip the enthusiasm of input partitions for clustering ensemble. In our experiments, we set $\alpha = 0.5$.

Consequently, the appropriate partition weights can be determined by minimizing L_3 [57]

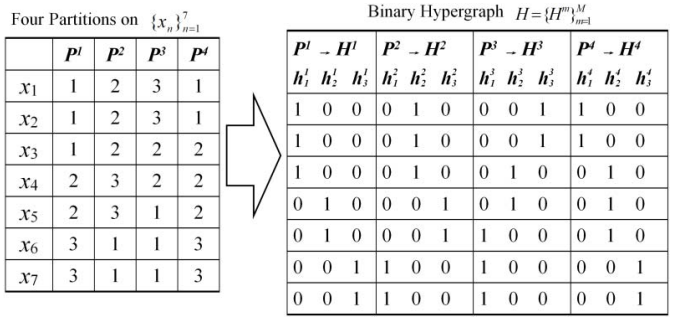
$$w_m = \frac{\exp(-Q_m/\alpha)}{\sum_m \exp(-Q_m/\alpha)}. \tag{7}$$

Once the input partitions and their corresponding weights are determined, the first term L_1 of L is fixed, and hence the performance of clustering ensemble is primarily controlled by L_2. Therefore, minimizing L is equivalent to minimizing the value of L_2. To optimize the process, we introduce a double layer-weighting scheme to determine the consensus partition that is close to all clusters. Inside the loss function: $L_2 = \sum_m w_m \sum_{k_m} p(k_m)[D_{\mathrm{KI}}(c_{k_m}^m, p_r)]$, the first layer weights are the partition weights obtained in (7), and the second layer weights are obviously the weights of clusters, which are defined as

$$w_{k_m}^m = p(k_m) = \frac{N_{k_m}^m}{N} \tag{8}$$

where $N_{k_m}^m$ is the number of data points in the cluster $C_{k_m}^m$ resulted in partition P^m, and N is the total number of data points.

2) Consensus Functions With Bi-Weighting Scheme: Existing work on cluster ensemble [51] applies three hypergraph-based consensus functions to produce the consensus partition, and multiple input partitions need to be initially mapped onto a hypergraph H by concatenating all binary membership indicators $H = \{H^m\}_{m=1}^M$. Such indicators are obtained by mapping each input partition P^m on $\{x_n\}_{n=1}^N$ into



	P^1	P^2	P^3	P^4
x_1	1	2	3	1
x_2	1	2	3	1
x_3	1	2	2	2
x_4	2	3	2	2
x_5	2	3	1	2
x_6	3	1	1	3
x_7	3	1	1	3

$P^1 \to H^1$			$P^2 \to H^2$			$P^3 \to H^3$			$P^4 \to H^4$		
h_1^1	h_2^1	h_3^1	h_1^2	h_2^2	h_3^2	h_1^3	h_2^3	h_3^3	h_1^4	h_2^4	h_3^4
1	0	0	0	1	0	0	0	1	1	0	0
1	0	0	0	1	0	0	0	1	1	0	0
1	0	0	0	1	0	0	1	0	0	1	0
0	1	0	0	0	1	0	1	0	0	1	0
0	1	0	0	0	1	1	0	0	0	1	0
0	0	1	1	0	0	1	0	0	0	0	1
0	0	1	1	0	0	1	0	0	0	0	1

Fig. 2. Example of hypergraph.

an adjacency matrix $H^m = \{h_{k_m}^m\}_{k_{m=1}}^{K_m}$. In such hypergraph, the vertices refer to a dataset of N objects $\{x_n\}_{n=1}^N$, whilst hyper-edge $h_{k_m}^m$ connecting a set of vertices indicates which objects belonging to the clusters k_m in partition P^m. One hypergraph is illustrated by a simple example shown in Fig. 2. To further improve the hypergraph-based consensus partition via our proposed bi-weighting scheme, we develop a weighted hypergraph as

$$G = \left\{ \sqrt{w_m} G^m \middle| G^m = \left\{ \sqrt{w_{k_m}^m} h_{k_m}^m \right\}_{k_{m=1}}^{K_m} \right\}_{m=1}^{M}. \tag{9}$$

In our approach, we apply three existing hypergraph-based consensus functions [51] on the weighted hypergraph to generate consensus partitions. These include CSPA, HGPA, and the MCLA [51]. The characteristics of these three consensus partitions can be briefly summarized as follows.

1) CSPA is a straightforward consensus function, where a similarity matrix S for input partitions encoded in a weighted hypergraph is derived from $G : S = GG^T$, and then the similarity matrix S is simply partitioned by a graph-based clustering algorithm (METIS) [58] to yield a consensus partition.
2) HGPA offers an alternative consensus function by casting the clustering ensemble problem on how to partition the weighted hypergraph by cutting minimal weighted hyper-edges. In this approach, hypergraph partitioning package (HMETIS) [59] is used to segment hypergraph G to obtain a consensus partition. Unlike the CSPA that takes the local piecewise similarity into account, HGPA considers a relatively global relationship among target dataset across multiple input partitions.
3) MCLA reaches a consensus solution by applying meta-clustering on weighted hypergraph G. In this approach, all the clusters represented by hyper-edges of hypergraph G are grouped into meta-clusters, and then these meta-clusters are further collapsed by assigning each data point to the collapsed hyper-edges, where its participation remains the strongest.

3) Objective Functions: Without the prior information, it is impossible to select a proper function in advance to form a clustering ensemble. By using the existing solutions, we adopt the NMI-based objective function [51] to determine the optimal consensus partition. Among the three candidates, the optimal consensus partition is selected as the one that possesses the maximum average mutual information with all M input partitions obtained from the initial clustering analysis module. Such objective function is defined as

$$\begin{aligned} P^o &= \arg \max_{1 \le r \le R} \sum_{m=1}^{M} \mathrm{NMI}\left(P^r, P^m\right) \\ &= \arg \max_{1 \le r \le R} \sum_{m=1}^{M} \frac{\sum_{i=1}^{K_r} \sum_{j=1}^{K_m} N_{ij}^{rm} \log\left(\frac{N N_{ij}^{rm}}{N_i^r N_j^m}\right)}{\sum_{i=1}^{K_r} N_i^r \log\left(\frac{N_i^r}{N}\right) + \sum_{j=1}^{K_m} N_j^m \log\left(\frac{N_j^m}{N}\right)} \end{aligned} \tag{10}$$

where P^m is the mth partition obtained from the initial clustering analysis, P^r is a consensus partition generated by the rth consensus function, and P^o is the optimal consensus partition. K_r and K_m represent the number of clusters resulted in P^r and P^m on a dataset of N data points, respectively. N_{ij}^{rm} is the number of shared data points between the cluster $C_i^r \in P^r$ and the cluster $C_j^m \in P^m$, N_i^r and N_j^m are the number of data points in C_i^r and C_j^m, respectively.

D. Final Refinement Function With Model Selection

Unlike the model-based approaches, three consensus functions (CSPA, HGPA, MCLA) can only obtain the labeling information for consensus partitions, and these consensus partitions are produced with predefined number of clusters, such as maximum number of clusters resulted in the input partitions. Following that, the optimal consensus solution is selected from three consensus candidates. In order to optimize such consensus solution with intrinsic number of clusters via HMM-based agglomerative clustering, we re-estimate the parameters of HMMs representing the clusters of optimal consensus partition P^o, and then refine it into a final partition P^*, where similar clusters in P^o are merged into a meta-cluster represented by a composite model [20], which is proved to be better in capturing the complex structure of clusters in P^*. Such merging process could be achieved by applying HMM-based agglomerative clustering in association with DSPA. This achieves the major advantage that the number of clusters in the final partition can be automatically determined. The entire process of final refinement function can be described as the following steps.

1) Train the component HMMs on the clusters of optimal consensus partition P^o.
2) Iteratively merge the closest clusters obtained in partition P^o into a meta-cluster represented by a composite model. Such process produces a hierarchical structure of clusters named as dendrogram. Inside the dendrogram, the horizontal axis indexes the clusters of optimal consensus partition P^o, and the vertical axis indicates the distance between the meta-clusters, which is illustrated in Fig. 5.
3) Obtain the final partition P^* with the intrinsic number of clusters K^* by cutting the dendrogram at the largest range of dissimilarity between successive merged clusters.

IV. Experiments

In this section, we conduct our experiments on three sets of datasets, including HMM-generated dataset, benchmarking

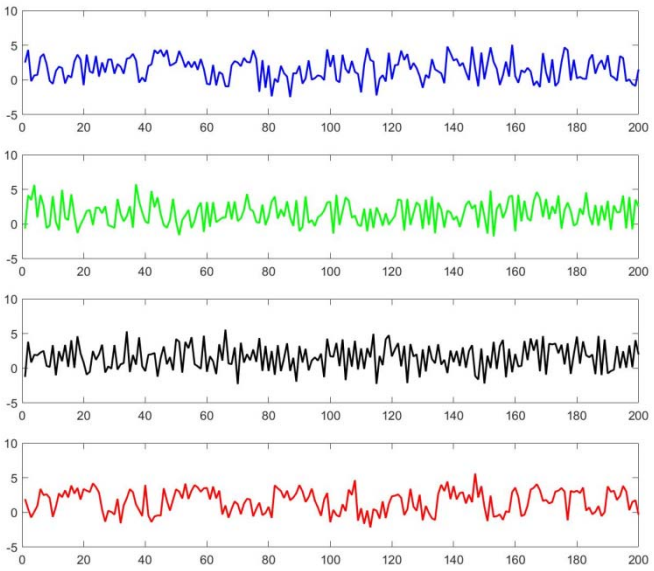


Fig. 3. HMM-generated dataset.

time series, and CAVIAR database of motion trajectories to evaluate the effectiveness of our proposed approach from different perspectives. Initially we carry out an experiment with HMM-generated dataset to test the fundamental idea of the proposed approach, and then we evaluate the performance of our approach on the benchmarking time series dataset to illustrate that our approach works for general clustering, and finally our approach is validated on CAVIAR database for real-world applications. Due to the fact of that classification accuracy [2] is commonly adopted by many existing algorithms in the published literature, we also evaluate performance of the tested algorithms by using this criterion in our experiments. The source code of implementing our approach is available upon request.

A. Experimental Settings and Datasets

The first dataset for our experiments is an HMM-generated dataset, which consists of 200 sequences generated by a mixture of four HMM models with two hidden states, and each component generates 50 sequences with an identical length of 200 as illustrated in Fig. 3. The parameters of four components are set as follows.

1) The initial state probabilities π_k are randomly generated from uniform distribution.
2) Transition matrices A_k are defined as

$$A_1 = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix} A_2 = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix} A_3 = \begin{bmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{bmatrix}$$
$$A_4 = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}.$$

3) Emission distribution corresponding to each state is characterized as a single Gaussian with mean and variance $\{\mu_k, \sigma_k^2\}$: $\{3, 1\}, \{3, 1\}, \{3, 1\}, \{3, 1\}$ for state 1, and $\{0, 1\}, \{0, 1\}, \{0, 1\}, \{0, 1\}$ for state 2.

With this synthetic dataset, we evaluate the model selection ability of our approach in comparison with the existing methods based on BIC. Further, we also compare the proposed approach with five HMM-based clustering algorithms, including HMM-based K-models, HMM-based agglomerative, HMM-based hybrid clustering, HMM-based hybrid meta-clustering, and HMM-based hybrid meta-clustering ensemble [2] from perspectives of classification accuracy and computational efficiency. In this part of experiments, the compared HMM-based approaches are parameterized with the number of states $S = 2$ and the number of clusters $K = 4$. To quantify the contribution of our proposed ideas on individual basis, we also implemented a prototype of the HMM-based hybrid meta clustering ensemble without bi-weighting, and its final partition is produced with a predefined number of clusters $K = 4$. All tested algorithms are run ten times, and an average classification accuracy is listed in Table II in the form of mean $\pm$ std.

Further, we examine our approach for general temporal data clustering tasks by using a collection of 16 time series benchmarking datasets [60]. The information of these datasets is detailed in Table I, including the number of classes, size of dataset, and the length of time series in each dataset. With these benchmarking time series, we initially compare our approach with five baseline algorithms including K-means, dynamic time warping (DTW) based K-means [3], HMM-based K-model, HMM hybrid clustering and HMM hybrid meta-clustering. In order to validate the effectiveness of our proposed bi-weighting scheme, we also compare our approach with its prototype that is HMM hybrid meta-clustering ensemble. Due to the fact that all algorithms compared here do not have the capability of automated model selection, we have to manually select the appropriate state number of HMMs from a range of candidates for each time series in the collection by using forward and backward algorithms [47], [48], and the correct numbers of clusters are also provided for the first five baseline algorithms for verification purposes. We run each algorithm ten times with optimal parameter setting and the best results of the tested algorithms are reported in Table III.

To evaluate our proposed approach in real-world applications, finally, we take CAVIAR database [61] as another benchmarking dataset to run the experiments, which consists of 222 high-quality motion trajectories, examples of which are summarized in Fig. 4. As seen, all the motion trajectories in the database are manually annotated from video sequences of pedestrians, and represented as 2-D temporal data. For configuration of HMMs, we use forward and backward algorithms again to determine the appropriate state number within a preset range, which has maximum log-likelihood for the estimated HMMs.

In our experiments, our approach runs the HMM K-models ten times as the initial clustering analysis by choosing a K value from a preset range ($8 \leq K \leq 18$ is defined in the experiment with CAVIAR database, and $K^* - 1 \leq K \leq K^* + 3$ in the rest of the experiments) under different initializations. After that, ten partitions of initial clustering analysis are fed to the bi-weighted clustering ensemble model to yield the optimal consensus partition, and such optimal consensus partition is further refined into a final partition with the intrinsic structures of clusters.

TABLE I
INFORMATION OF TIME SERIES BENCHMARKING DATASETS

Dataset	Number of Class K^*	Size of Dataset (Training+Testing)	Length
Syn 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

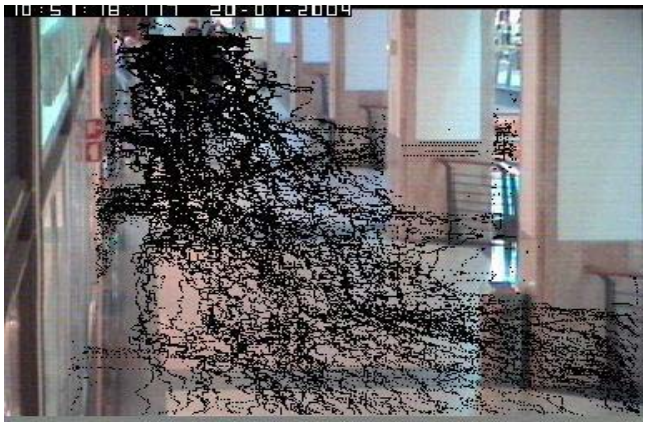


Fig. 4. CAVIAR database with 222 manually annotated trajectories.

B. Experimental Results and Analysis

1) Fundamental Performance Analysis: Fig. 5 illustrates the dendrogram obtained by applying our approach to HMM-generated dataset, where the correct number of clusters ($K^* = 4$) can be determined by cutting the dendrogram at a threshold value corresponding to the largest range of dissimilarity between successive merged clusters during the final refinement process. For comparison purposes, we also apply our approach to the target dataset by fixing the cluster size in the range of $2 \leq K \leq 10$ instead of using DSPA, and then calculate the BIC values on different numbers of clusters resulted in the final partition. As shown in Fig. 6, the optimal number of clusters is selected as six with a minimum value of BIC, indicating a failure of the BIC-based model selection method.

Table II shows the results of six HMM-based approaches on HMM-generated dataset in term of classification accuracy and computational efficiency. From the perspective of classification accuracy, it can be seen that the top four baseline algorithms without model selection produce higher standard deviation for the classification accuracy, indicating that the performances of such algorithms are un-stabilized due to the model initialization problem. This is especially true with the HMM-based K-model, due to the fact that it is quite

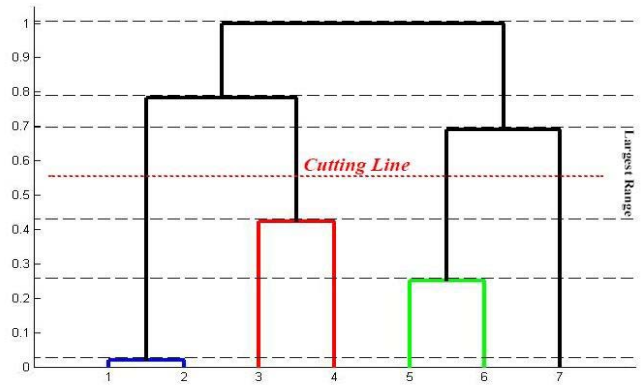


Fig. 5. Dendrogram of the final partition on HMM-generated dataset.

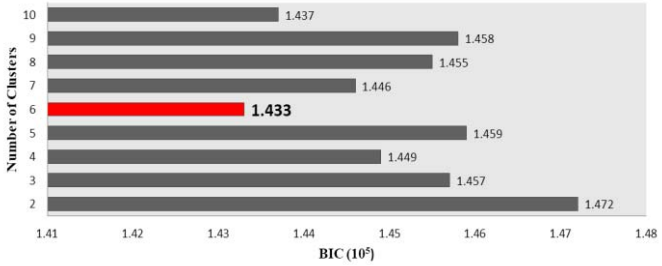


Fig. 6. BIC values of different cluster numbers on HMM-generated dataset.

TABLE II
RESULTS ON HMM-GENERATED DATASET

HMM-based clustering algorithms	Average Classification Accuracy (%) (*mean* ±*std*)	CPU time in seconds (*mean* ±*std*)
HMM K-Model	73.2 ± 5.5	**102.9 ± 0.4**
HMM agglomerative	71.1 ± 2.1	1502.5 ± 3.3
HMM hybrid clustering	73.8 ± 3.9	193.1 ± 0.4
HMM hybrid meta-clustering	74.2 ± 2.6	106.9 ± 0.6
HMM hybrid meta-clustering EN	83.1 ± 1.8	1034.2 ± 0.9
Our Approach	**88.5 ± 1.8**	1041.5 ± 1.8

sensitive to model initialization, which is clearly demonstrated by its largest value of standard deviation. In contrast, HMM-based hybrid meta-clustering ensemble makes improvement in term of the classification accuracy and initialization sensitivity. This is demonstrated by its higher average of classification accuracy and smaller standard deviations. By introducing the bi-weighting scheme and the refinement function with the ability of automatic model selection, it can be seen from the results that our proposed approach achieves the best average of classification accuracy with the smallest standard deviation among all the tested algorithms. From the perspective of computational efficiency, Table II also reports the CPU time of implementing these HMM-based approaches. While the ensemble approaches achieve higher classification accuracies, as seen, they are generally slower than single algorithms due to the fact of that generation of input partitions consumes a large amount of computing resource. Compared with the HMM-based agglomerative clustering algorithm, however, our proposed still requires less computing resources, yet

TABLE III
RESULTS ON TIME SERIES BENCHMARKING DATASETS—CLASSIFICATION ACCURACY (%)

Dataset	K-means	DTW based K-means	HMM K-Model	HMM hybrid clustering	HMM hybrid meta-clustering	HMM hybrid meta-clustering ensemble	Our Approach with BIC model selection	Our Approach
Syn Control	67.9	69.8	69.1	69.8	71.1	73.2*	**75.2***	**75.2***
Gun-Point	50.0	65.6	43.8	51.8	50.0	65.2*	**69.1***	**69.1***
CBF	62.6	**80.9**	60.1	63.2	65.2	64.3*	70.0*	70.0*
Face (all)	36.0	**49.4**	37.8	36.4	39.2	42.8	32.6	38.6
OSU Leaf	37.8	35.1	44.2	40.8	45.1	47.2	35.0	**49.5***
Swedish Leaf	40.6	48.1	38.6	47.6	**49.2**	42.5*	39.3	46.7*
50Words	42.0	37.2	40.8	38.9	41.0	**46.2***	39.1	45.9*
Trace	48.5	63.4	50.9	56.3	59.8	63.9*	**66.2***	**66.2***
Two Patterns	32.2	**56.3**	33.1	35.2	38.1	50.6*	31.6	52.1*
Wafer	62.5	47.5	63.9	**65.1**	62.9	58.1	54.2*	52.1
Face (four)	66.9	70.7	69.1	64.2	61.9	61.4	**70.9***	63.8
Lightning-2	61.1	62.1	57.7	63.2	66.8	67.6*	51.6	**71.2***
Lightning-7	48.4	50.5	51.2	47.3	45.3	50.0*	49.5	**53.7***
ECG	69.8	62.8	**70.3**	61.6	63.3	67.9	55.5	68.8
Adiac	38.4	39.6	38.9	42.0	40.2	43.2*	**45.4***	**45.4***
Yoga	51.7	56.3	48.5	44.3	47.1	63.8*	45.8	**66.9***

significantly improves the classification accuracy with both the model selection and initialization problems resolved.

2) Benchmarking Analysis: Table III shows the experiment results on the benchmarking datasets of time series, from which it can be seen that our approach wins the first place by achieving the best performance on 8 out of 16 datasets. While DTW-based *K*-means achieves the best results for three datasets, and all others only win on one dataset, respectively. It is worth mentioning that these baseline algorithms implemented by providing the intrinsic number of clusters actually take the advantage of our approach without such prior information. In order to examine the model selection capability of our approach in different versions, we mark the classification accuracies in Table III with a * symbol if their clustering results are achieved with the correct cluster number determined. As seen, our approach is able to find the correct cluster number on 12 out of 16 datasets, yet the BIC-based method can only manage seven datasets.

In fact, all these approaches compared in Table III try to solve the temporal data clustering problems from different perspectives. *K*-means simply uses Euclidean distance to measure the similarity between time series based on local comparisons, where the time series are aligned point by point. As shown in Table III, such baseline algorithm fails to achieve satisfactory results, especially when the observations of time series are shifted, such as *Gun-Point, CBF*, and *Two Patterns*. In order to overcome such limitations, a DTW distance [4] is developed to determine a warping distance out of the best alignment between two time series. From the results shown in Table III, it can be seen that DTW-based *K*-means outperforms standard *K*-means on 12 out of 16 datasets. For high dimensional time series, such as *OSU Leaf*, *Lightning-2*, and *Yoga*, however, the results achieved by DTW-based *K*-means show little improvement, yet take a considerably longer time than other algorithms. While HMM-based approaches manage to capture the cluster structures by considering the temporal information of the time series, the improvement achieved are still limited. Comparative studies on the results listed in Table III indicate that our approach achieves the best performance in term of both model selection and classification accuracies. It typically works well for high dimensional time series, and achieves the best results for the longest time series, including *OSU Leaf*, *Lighting-2*, *Lighting-7*, and *Yoga*. Further, Table III also shows that our approach outperforms HMM hybrid meta-clustering ensemble as its prototype by winning 13 out of 16 datasets, which clearly justifies the effectiveness of the proposed bi-weighting scheme.

3) Real-World Application Analysis: Fig. 7 shows the clustering analysis of all moving trajectories in the CAVIAR database achieved by our approach. It actually provides a good potential for developing video content analysis algorithms based on unsupervised learning. As seen, our approach divides 222 motion trajectories into ten groups, where trajectories with similar motion behaviors are properly grouped within the same cluster while dissimilar ones are separated into different clusters. From the viewpoint of front camera, Fig. 7(i) illustrates the group of trajectories that corresponds to the activity "pass in front of cameras," Fig. 7(a) and (d) illustrate groups of trajectories having "walk and watch" movements at different locations, and Fig. 7(c), (f), and (h) illustrate groups of trajectories having the activities of "enter and exit the store" with three motion paths. While the trajectories corresponding to "wandering in the hall" are represented within a single cluster as shown in Fig. 7(j), and trajectories corresponding to "walk through the hall" with same direction are grouped into a single cluster as shown in Fig. 7(e). Meanwhile, the trajectories corresponding to "move horizontally" movements at different locations are grouped into two clusters as shown in Fig. 7(b) and (g).

In application of video surveillances, moving objects tracking is often interfered by external noises or obstacles, consequently the collection of motion trajectories are always corrupted via tracking algorithms. In order to evaluate the robustness of our approach in such scenarios, we carry out one more experiment of clustering-based classifications on corrupted trajectories. In this simulation, the clean version of trajectories are corrupted by: 1) adding different amounts of Gaussian noise $N(0, \sigma)$ and 2) removing segments of trajectories measured by a percentage of the trajectory length

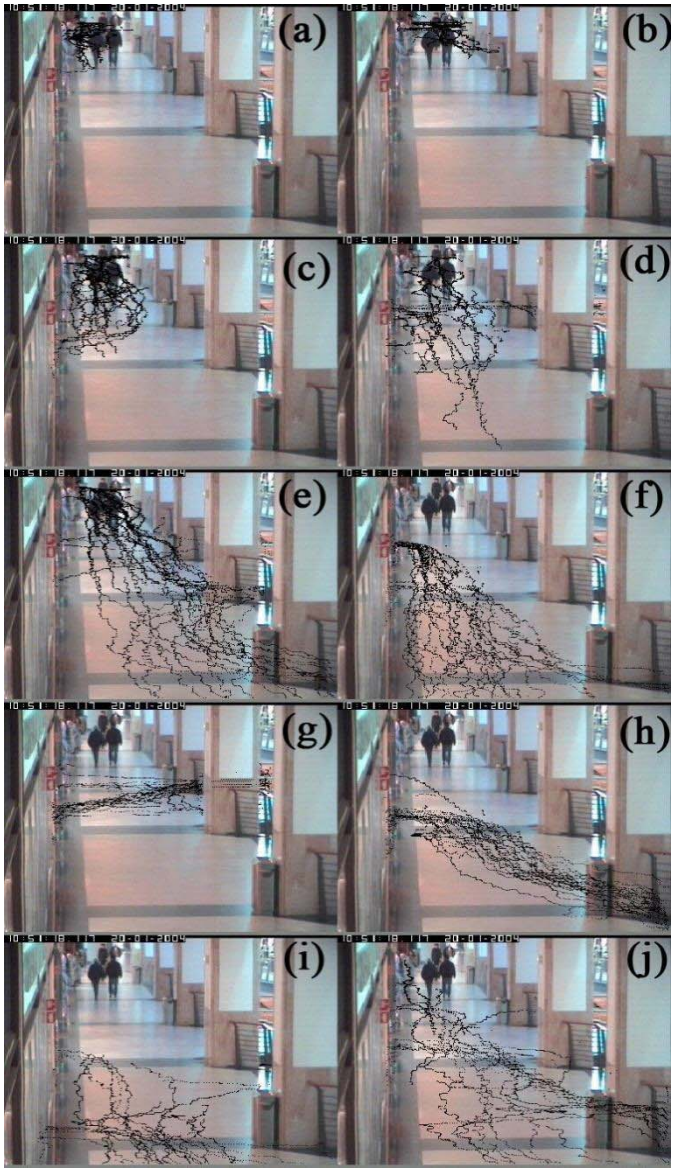


Fig. 7. Clustering analysis on CAVIAR database achieved by our approach. (a) and (d) Activities of "walk and watch." (b) and (g) "Move horizontally." (c), (f), and (h) "Enter and exit the store." (e) "Walk through the hall." (i) "Pass in front of cameras." (j) "Wandering in the hall."

at random locations. The corrupted trajectories are then classified by clustering analysis of the clean version of trajectories as illustrated in Fig. 7. As seen, the label of corrupted trajectory is assigned by the cluster, whose corresponding HMM model has the maximum log-likelihood to generate this corrupted trajectory. Finally, the classification accuracies are obtained by using the clustering label of clean version as the ground-truth. Table IV shows the experimental results with different level of corruptions. It can be seen that our approach still achieves satisfactory classification accuracy (62.6%) for the worst corrupted trajectories, which are created by adding Gaussian noise $N(0, 0.4)$ to, and removing 90% information from clean version trajectories. While the last column and row of table report the average classification accuracies with standard deviations at a range of missing data and noise, respectively, these results further demonstrate the robustness and effectiveness of our approach in the real-world applications.

TABLE IV
TESTING RESULTS ON CORRUPTED TRAJECTORIES—CLASSIFICATION ACCURACY (%)

Percentages of missing data	Added noise with $N(0,0.1)$	Added noise with $N(0,0.2)$	Added noise with $N(0,0.3)$	Added noise with $N(0,0.4)$	***mean* ± *std***
10%	87.7	85.7	83.3	79.1	84.0 ± 3.7
20%	88.4	84.3	79.8	76.6	82.3 ± 5.2
30%	88.2	82.2	77.2	75.2	80.7 ± 5.8
40%	85.9	81.6	76.1	72.5	79.0 ± 5.9
50%	84.3	81.2	76.9	71.2	78.4 ± 5.7
60%	83.6	80.8	75.8	70.9	77.8 ± 5.6
70%	79.1	79.0	74.3	67.9	75.1 ± 5.3
80%	78.0	75.8	70.0	66.0	72.5 ± 4.2
90%	75.1	72.2	68.9	62.6	69.7 ± 5.4
mean* ± *std	83.4 ± 4.9	80.1 ± 4.2	75.8 ± 4.2	71.3 ± 5.0	

V. DISCUSSION

Existing work on weighted clustering ensemble algorithms [5], [57], [62], [63] can be generally categorized into either cluster weighting or partition weighting. A cluster weighting scheme [57], [62] associates the clusters in a partition with a weighting vector and embeds it in a subspace spanned by adaptive combination of feature dimensions. A partition weighting scheme [5], [58], [63] assigns a weight vector to the partitions to be combined. In our approach, we efficiently combine both partition weighting and cluster weighting into a single bi-weighting scheme. Such weighting scheme provides an optimal reconciliation among all input partitions to produce a consolidated consensus partition. Our proposed HMM-based meta-clustering ensemble model with a bi-weighting scheme has been empirically and theoretically justified for temporal data clustering.

From experimental analysis, experiment results show that our approach always achieves the best performance in comparison with similar algorithms. The results illustrated in Table II indicate that our approach is much more robust to the model initialization than existing HMM-based clustering algorithms. Further, the extensive experiment results reported in Table III also demonstrate that the proposed final refinement function with model selection is superior in determining the number of clusters automatically. Furthermore, the effectiveness and practicability of our approach have also been demonstrated on a real-world application as illustrated in Table IV and Fig. 7.

From theoretical analysis, the loss function L of clustering ensemble derived in (4) suggests that the best reconciliation of input partitions should be achieved by considering together the importance of partitions and the corresponding clusters. In order to quantify such importance in our ensemble model, a bi-weighting scheme is developed from the two terms in (4). According to the first term L_1, a good partition manifests the cluster structure with a small intracluster distance and a large intercluster distance, which should make more contribution to the consensus partition in order to minimize the value of L_1. However, such optimization can only produce the best partition selected as a consensus solution. In order to encourage contributions from all input partitions for clustering ensemble, L_3 is defined by adding a regularization term into L_1. By minimizing L_3, the weights of partitions are obtained in (7). As long as the input partitions are given, the first term L_1

is often fixed, and the performance of a clustering ensemble is controlled by the second term L_2. It actually suggests that a consensus partition should be close to all the clusters resulted in the input partitions based on a weighted distance. Since the weights of partitions are solved by optimizing L_3, the weights of clusters based on cluster size can be naturally defined in (8) by minimizing the value of L_2.

In addition, existing techniques on automatic identification of cluster numbers often involve complicated procedures and hence computing intensive. A representative example [64] can be highlighted with three steps of operations.

1) Generate a set of partitions obtained by a clustering algorithm with a range of cluster number initializations.
2) Map them into an adjacency matrix, and then iteratively apply a selected graph partitioning algorithm on the adjacency matrix with decreased resolution in order to produce a list of partitioning candidates.
3) Identify a long-life structure of the clusters out of all such partition candidates and hence complete the determination of the cluster number. In contrast, our approach can simplify such iterative decomposition of adjacency matrix and graph partitioning by simply cutting the dendrogram obtained from the consensus partition at a range of threshold values, and these threshold values can be easily determined corresponding to the longest range of dissimilarity between successive merged clusters. In this way, our proposed approach can reduce the computational complexity significantly without compromise on the effectiveness of the clustering performances.

Future research can be considered to address the model configuration of HMM due to the fact of that determination of emission distribution and state number are always critical issues for HMM-based approaches. Meanwhile, our proposed ensemble approach provides a promising solution for both initialization problem and model selection problem. The weakness of our approach, however, is that it is time-consuming in generating input partitions during initial clustering analysis, which could be a problem for data mining of large datasets. Therefore, it will be an interesting and challenge research topic to reach a compromise between computational efficiency and classification accuracy for ensemble techniques.

VI. Conclusion

In this paper, we have presented a novel HMM-based hybrid meta-clustering ensemble approach with bi-weighting derived from a formal analysis of the objective function of clustering ensemble. The extensive experiments on various temporal datasets demonstrate that our approach achieves the promising performance for temporal data clustering analysis and is suitable for applications in an unknown environment. In the conclusion, four major advantages can be highlighted for our proposed approach, which include: 1) the initialization problem can be solved by adopting ensemble technique; 2) the correct cluster number can be automatically determined on the final partition via HMM-based agglomerative clustering in association with DSPA; 3) A bi-weighting scheme is developed to obtain an improved clustering ensemble solution based on optimal synergy between partitions and clusters; and 4) the complex structure of clusters can be intrinsically captured by a composite model in the final refinement.

Acknowledgment

The authors would like to thank E. Keogh for providing the Benchmark time series dataset for evaluating our proposed model and also would like to thank A. Strehl, who published his Cluster Ensemble code online in helping us to complete the comparative studies.

References

[1] Y. Liu *et al.*, "Understanding and enhancement of internal clustering validation measures," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 982–994, Jun. 2013.
[2] Y. Yang, *Temporal Data Mining via Unsupervised Ensemble Learning*. San Diego, CA, USA: Elsevier, 2016.
[3] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial–temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007.
[4] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Min. Knowl. Disc.*, vol. 7, no. 4, pp. 349–371, 2003.
[5] Y. Yang and K. Chen, "Temporal data clustering via weighted clustering ensemble with different representations," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 2, pp. 307–320, Feb. 2011.
[6] Y. Yang and K. Chen, "Time series clustering via RPCL network ensemble with different representations," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 2, pp. 190–199, Mar. 2011.
[7] A. Bagnall, C. A. Ratanamahatana, E. Keogh, S. Lonardi, and G. Janacek, "A bit level representation for time series data mining with shape based similarity," *Data Min. Knowl. Disc.*, vol. 13, no. 1, pp. 11–40, 2006.
[8] C. Cheong, W. Lee, and N. Yahaya, "Wavelet-based temporal clustering analysis on stock time series," in *Proc. Int. Conf. Quant. Sci. Appl.*, 2005.
[9] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulos, "Iterative incremental clustering of time series," in *Proc. Adv. Database Technol. (EDBT)*, Heraklion, Greece, 2004, pp. 106–122.
[10] N. Asadi, A. Mirzaei, and E. Haghshenas, "Creating discriminative models for time series classification and clustering by HMM ensembles," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2899–2910, Dec. 2016.
[11] A. Panuccio, M. Bicego, and V. Murino, "A hidden Markov model-based approach to sequential data clustering," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR)*, Windsor, ON, Canada, 2002, pp. 734–743.
[12] Y. Xiong and D.-Y. Yeung, "Mixtures of ARMA models for model-based time series clustering," in *Proc. IEEE Int. Conf. Data Min.*, Maebashi, Japan, 2002, pp. 717–720.
[13] A. J. Bagnall and G. J. Janacek, "Clustering time series from ARMA models with clipped data," in *Proc. Int. Conf. Knowl. Disc. Data Min.*, Seattle, WA, USA, 2004, pp. 49–58.
[14] P. Smyth, "Probabilistic model-based clustering of multivariate and sequential data," in *Proc. 7th Int. Workshop Artif. Intell. Stat.*, vol. 99, 1999, pp. 299–304.
[15] M. Ramoni, P. Sebastiani, and P. Cohen, "Bayesian clustering by dynamics," *Mach. Learn.*, vol. 47, no. 1, pp. 91–121, 2002.
[16] I. Cadez, D. Heckerman, C. Meek, P. Smyth, and S. White, "Visualization of navigation patterns on a Web site using model-based clustering," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Boston, MA, USA, 2000, pp. 280–284.
[17] S. Policker and A. B. Geva, "Nonstationary time series analysis by temporal clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 2, pp. 339–343, Apr. 2000.
[18] P. Smyth, "Clustering sequences with hidden Markov models," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 648–654.
[19] C. Li and G. Biswas, "Applying the hidden Markov model methodology for unsupervised learning of temporal data," *Int. J. Knowl. Based Intell. Eng. Syst.*, vol. 6, no. 3, pp. 152–160, 2002.
[20] S. Zhong and J. Ghosh, "A unified framework for model-based clustering," *J. Mach. Learn. Res.*, vol. 4, pp. 1001–1037, Jan. 2003.
[21] D. Tran and M. Wagner, "Fuzzy C-means clustering-based speaker verification," in *Proc. Adv. Soft Comput. (AFSS)*, Kolkata, India, 2002, pp. 318–324.

[22] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[23] J. Picone, "Continuous speech recognition using hidden Markov models," *IEEE ASSP Mag.*, vol. 7, no. 3, pp. 26–41, Jul. 1990.
[24] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 1974.
[25] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.
[26] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Disc. Data Min.*, 1996, pp. 226–231.
[27] S. K. Bhatia, "Adaptive k-means clustering," in *Proc. Int. Florida Artif. Intell. Res. Soc. Conf.*, 2004, pp. 695–699.
[28] F. Ensan, M. H. Yaghmaee, and E. Bagheri, "FACT: A new fuzzy adaptive clustering technique," presented at the 11th IEEE Symp. Comput. Commun., 2006, pp. 442–447.
[29] J. Beringer and E. Hullermeier, "Adaptive optimization of the number of clusters in fuzzy clustering," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Los Alamitos, CA, USA, 2007, pp. 1–6.
[30] W. Zucchini, "An introduction to model selection," *J. Math. Psychol.*, vol. 44, no. 1, pp. 41–61, 2000.
[31] X. Hu and L. Xu, "A comparative study of several cluster number selection criteria," in *Proc. 4th Int. Conf. Intell. Data Eng. Autom. Learn.*, 2003, pp. 195–202.
[32] P. Yang *et al.*, "Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 445–455, Mar. 2014.
[33] R. Diao, F. Chao, T. Peng, N. Snooke, and Q. Shen, "Feature selection inspired classifier ensemble reduction," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1259–1268, Aug. 2014.
[34] W. Kim, J. Park, J. Yoo, H. J. Kim, and C. G. Park, "Target localization using ensemble support vector regression in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 4, pp. 1189–1198, Aug. 2013.
[35] X. Zhu, P. Zhang, X. Lin, and Y. Shi, "Active learning from stream data using optimal weight classifier ensemble," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1607–1621, Dec. 2010.
[36] S. Pan, J. Wu, X. Zhu, and C. Zhang, "Graph ensemble boosting for imbalanced noisy graph stream classification," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 954–968, May 2015.
[37] V. Soto, S. García-Moratilla, G. Martinez-Munoz, D. Hernández-Lobato, and A. Suárez, "A double pruning scheme for boosting ensembles," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2682–2695, Dec. 2014.
[38] Z. Yu, L. Li, J. Liu, and G. Han, "Hybrid adaptive classifier ensemble," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 177–190, Feb. 2015.
[39] L. Zhang and P. N. Suganthan, "Oblique decision tree ensemble via multisurface proximal support vector machine," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2165–2176, Oct. 2015.
[40] T. Wang, "CA-tree: A hierarchical structure for efficient and scalable coassociation-based cluster ensembles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 686–698, Jun. 2011.
[41] Y. Hong, S. Kwong, Y. Chang, and Q. Ren, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," *Pattern Recognit.*, vol. 41, no. 9, pp. 2742–2756, 2008.
[42] B. Fischer and J. M. Buhmann, "Bagging for path-based clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 11, pp. 1411–1415, Nov. 2003.
[43] W. Zhuang, Y. Ye, Y. Chen, and T. Li, "Ensemble clustering for Internet security applications," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1784–1796, Nov. 2012.
[44] Y. Yang and J. Jiang, "Hybrid sampling-based clustering ensemble with global and local constitutions," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 952–965, May 2016.
[45] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Holden-Day, 1976.
[46] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.
[47] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bull. Amer. Math. Soc.*, vol. 73, no. 3, pp. 360–363, 1967.
[48] L. E. Baum and G. R. Sell, "Growth transformations for functions on manifolds," *Pac. J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.
[49] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.
[50] N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information: Ranking and clustering," *J. ACM*, vol. 55, no. 5, pp. 1–27, 2008.
[51] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.
[52] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Disc. Data*, vol. 1, no. 1, p. 4, 2007.
[53] A. L. N. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 835–850, Jun. 2005.
[54] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data," *Mach. Learn.*, vol. 52, nos. 1–2, pp. 91–118, 2003.
[55] A. P. Topchy, M. H. C. Law, A. K. Jain, and A. L. Fred, "Analysis of consensus partition in cluster ensemble," in *Proc. IEEE Int. Conf. Data Min.*, Brighton, U.K., 2004, pp. 225–232.
[56] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 4, pp. 379–423, Oct. 1948.
[57] M. Al-Razgan and C. Domeniconi, "Weighted clustering ensembles," presented at the SIAM Int. Conf. Data Min., 2006.
[58] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, 1998.
[59] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Applications in VLSI domain," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 1, pp. 69–79, Mar. 1999.
[60] E. Keogh. *Temporal Data Mining Benchmarks*. Accessed: May 2014. [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_series_data
[61] CAVIAR. *Context Aware Vision Using Image-Based Active Recognition*. Accessed: Jun. 2015. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR
[62] H. Cheng, K. A. Hua, and K. Vu, "Constrained locally weighted clustering," in *Proc. AMC Int. Conf. Very Large Data Bases*, vol. 1, 2008, pp. 90–101.
[63] T. Li and C. Ding, "Weighted consensus clustering," in *Proc. SIAM Int. Conf. Data Min.*, 2008, pp. 798–809.
[64] P. Y. Mok, H. Q. Huang, Y. L. Kwok, and J. S. Au, "A robust adaptive clustering analysis method for automatic identification of clusters," *Pattern Recognit.*, vol. 45, no. 8, pp. 3017–3033, 2012.
[65] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

Yun Yang received the B.Sc. (Hons.) degree in information technology and telecommunication from Lancaster University, Lancaster, U.K., in 2004, the M.Sc. degree in advanced computing from Bristol University, Bristol, U.K., in 2005, and the M.Phil. degree in informatics and the Ph.D. degree in computer science from the University of Manchester, Manchester, U.K., in 2006 and 2011, respectively.

He was a Research Fellow with the University of Surrey, Surrey, U.K., from 2012 to 2013. He is currently with the National Pilot School of Software, Yunnan University, Kunming, China, as a Full Professor of machine learning. His current research interests include machine learning, data mining, pattern recognition, and temporal data process and analysis.

Jianmin Jiang received the Ph.D. degree from the University of Nottingham, Nottingham, U.K., in 1994.

From 1997 to 2001, he was a Full Professor of computing with the University of Glamorgan, Wales, U.K. In 2002, he joined the University of Bradford, Bradford, U.K., as a Chair Professor of Digital Media, and the Director of Digital Media and Systems Research Institute. From 2010 to 2014, he was with the University of Surrey, Surrey, U.K., as a Professor of media computing. He is currently a Chinese National 1000-plan Distinguished Professor and the Director of the Research Institute for Future Media Computing, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He has published around 400 refereed research papers. His current research interests include machine learning, image/video processing in compressed domain, digital video coding, multimedia informatics, medical imaging, computer graphics, and pattern recognitions.

Dr. Jiang is a Chartered Engineer, a fellow of IEE and RSA, a member of EPSRC College, and EU FP-6/7 evaluator.