Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition 38 (2005) 1857–1874

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Clustering of time series data—a survey

T. Warren Liao*

Industrial & Manufacturing Systems Engineering Department, Louisiana State University, 3128 CEBA, Baton Rouge, LA 70803, USA

Received 16 September 2003; received in revised form 21 June 2004; accepted 7 January 2005

Abstract

Time series clustering has been shown effective in providing useful information in various domains. There seems to be an increased interest in time series clustering as part of the effort in temporal data mining research. To provide an overview, this paper surveys and summarizes previous works that investigated the clustering of time series data in various application domains. The basics of time series clustering are presented, including general-purpose clustering algorithms commonly used in time series clustering studies, the criteria for evaluating the performance of the clustering results, and the measures to determine the similarity/dissimilarity between two time series being compared, either in the forms of raw data, extracted features, or some model parameters. The past researchs are organized into three groups depending upon whether they work directly with the raw data either in the time or frequency domain, indirectly with features extracted from the raw data, or indirectly with models built from the raw data. The uniqueness and limitation of previous research are discussed and several possible topics for future research are identified. Moreover, the areas that time series clustering have been applied to are also summarized, including the sources of data used. It is hoped that this review will serve as the steppingstone for those interested in advancing this area of research.

Keywords: Time series data; Clustering; Distance measure; Data mining

1. Introduction

The goal of clustering is to identify structure in an unlabeled data set by objectively organizing data into homogeneous groups where the within-group-object similarity is minimized and the between-group-object dissimilarity is maximized. Clustering is necessary when no labeled data are available regardless of whether the data are binary, categorical, numerical, interval, ordinal, relational, textual, spatial, temporal, spatio-temporal, image, multimedia, or mixtures of the above data types. Data are called static if all their feature values do not change with time, or change negligibly. The bulk of clustering analyses has been performed on static data. Most, if not all, clustering programs developed as an independent program or as part of a large suite of data analysis or data mining software to date work only with static data. Han and Kamber [1] classified clustering methods developed for handing various static data into five major categories: partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. A brief description of each category of methods follows.

Given a set of n unlabeled data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster containing at least one object and $k \leqslant n$. The partition is crisp if each object belongs to exactly one cluster, or fuzzy if one object is allowed to be in more than one cluster to a different degree. Two renowned heuristic methods for crisp partitions are the *k-means* algorithm [2], where each cluster is represented by the mean value of the objects in the cluster and the *k-medoids*

* Tel.: +1 225 578 5365; fax: +1 225 578 5109.
E-mail address: ieliao@lsu.edu.

0031-3203/$30.00 © 2005 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2005.01.025

algorithm [3], where each cluster is represented by the most centrally located object in a cluster. Two counterparts for fuzzy partitions are the *fuzzy c-means* algorithm [4] and the *fuzzy c-medoids* algorithm [5]. These heuristic algorithms work well for finding spherical-shaped clusters and small to medium data sets. To find clusters with non-spherical or other complex shapes, specially designed algorithms such as Gustafson–Kessel and adaptive fuzzy clustering algorithms [6] or density-based methods to be introduced in the sequel are needed. Most genetic clustering methods implement the spirit of partitioning methods, especially the *k-means* algorithm [7,8], the *k-medoids* algorithm [9], and the *fuzzy c-means* algorithm [10].

A hierarchical clustering method works by grouping data objects into a tree of clusters. There are generally two types of hierarchical clustering methods: agglomerative and divisive. Agglomerative methods start by placing each object in its own cluster and then merge clusters into larger and larger clusters, until all objects are in a single cluster or until certain termination conditions such as the desired number of clusters are satisfied. Divisive methods do just the opposite. A pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed. For improving the clustering quality of hierarchical methods, there is a trend to integrate hierarchical clustering with other clustering techniques. Both Chameleon [11] and CURE [12] perform careful analysis of object "linkages" at each hierarchical partitioning whereas BIRCH [13] uses iterative relocation to refine the results obtained by hierarchical agglomeration.

The general idea of density-based methods such as DBSCAN [14] is to continue growing a cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold. Rather than producing a clustering explicitly, OPTICS [15] computes an augmented cluster ordering for automatic and interactive cluster analysis. The ordering contains information that is equivalent to density-based clustering obtained from a wide range of parameter settings, thus overcoming the difficulty of selecting parameter values.

Grid-based methods quantize the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. A typical example of the grid-based approach is STING [16], which uses several levels of rectangular cells corresponding to different levels of resolution. Statistical information regarding the attributes in each cell are pre-computed and stored. A query process usually starts at a relatively high level of the hierarchical structure. For each cell in the current layer, the confidence interval is computed reflecting the cell's relevance to the given query. Irrelevant cells are removed from further consideration. The query process continues to the next lower level for the relevant cells until the bottom layer is reached.

Model-based methods assume a model for each of the clusters and attempt to best fit the data to the assumed model. There are two major approaches of model-based methods: statistical approach and neural network approach. An example of statistical approach is AutoClass [17], which uses Bayesian statistical analysis to estimate the number of clusters. Two prominent methods of the neural network approach to clustering are competitive learning, including ART [18] and self-organizing feature maps [19].

Unlike static data, the time series of a feature comprise values changed with time. Time series data are of interest because of its pervasiveness in various areas ranging from science, engineering, business, finance, economic, health care, to government. Given a set of unlabeled time series, it is often desirable to determine groups of similar time series. These unlabeled time series could be monitoring data collected during different periods from a particular process or from more than one process. The process could be natural, biological, business, or engineered. Works devoting to the cluster analysis of time series are relatively scant compared with those focusing on static data. However, there seems to be a trend of increased activity.

This paper intends to introduce the basics of time series clustering and to provide an overview of time series clustering works been done so far. In the next section, the basics of time series clustering are presented. Details of three major components required to perform time series clustering are given in three subsections: clustering algorithms in Section 2.1, data similarity/distance measurement in Section 2.2, and performance evaluation criterion in Section 2.3. Section 3 categories and surveys time series clustering works that have been published in the open literature. Several possible topics for future research are discussed in Section 4 and finally the paper is concluded. In Appendix A, the application areas reported are summarized with pointers to openly available time series data.

2. Basics of time series clustering

Just like static data clustering, time series clustering requires a clustering algorithm or procedure to form clusters given a set of unlabeled data objects and the choice of clustering algorithm depends both on the type of data available and on the particular purpose and application. As far as time series data are concerned, distinctions can be made as to whether the data are discrete-valued or real-valued, uniformly or non-uniformly sampled, univariate or multivariate, and whether data series are of equal or unequal length. Non-uniformly sampled data must be converted into uniformed data before clustering operations can be performed. This can be achieved by a wide range of methods, from simple down sampling based on the roughest sampling interval to a sophisticated modeling and estimation approach.

Various algorithms have been developed to cluster different types of time series data. Putting their differences aside, it is far to say that in spirit they all try to modify the existing algorithms for clustering static data in such a way that

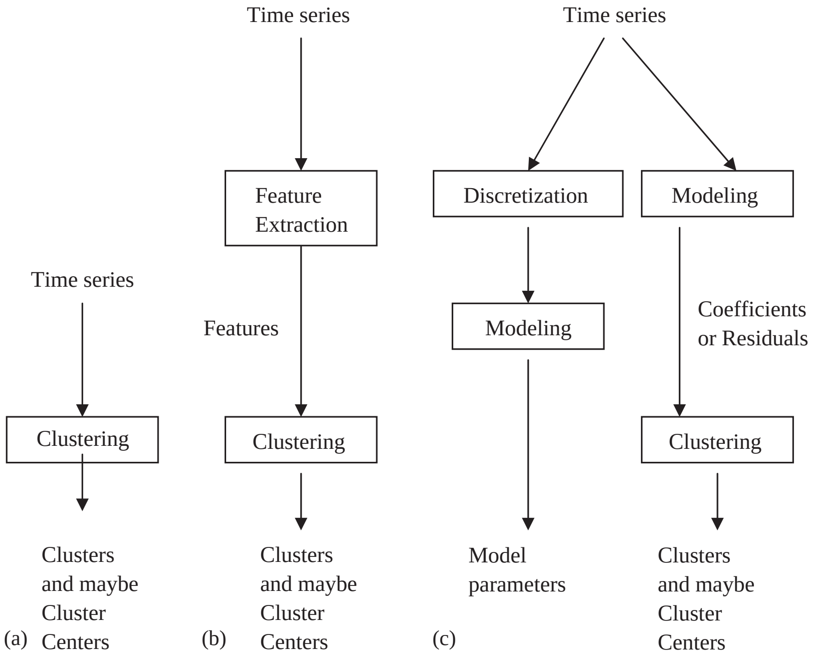

Fig. 1. Three time series clustering approaches: (a) raw-data-based, (b) feature-based, (c) model-based.

time series data can be handled or to convert time series data into the form of static data so that the existing algorithms for clustering static data can be directly used. The former approach usually works directly with raw time series data, thus called raw-data-based approach, and the major modification lies in replacing the distance/similarity measure for static data with an appropriate one for time series. The latter approach first converts a raw time series data either into a feature vector of lower dimension or a number of model parameters, and then applies a conventional clustering algorithm to the extracted feature vectors or model parameters, thus called feature- and model-based approach, respectively. Fig. 1 outlines the three different approaches: raw-data-based, feature-based, and model-based. Note that the left branch of model-based approach trained the model and used the model parameters for clustering without the need for another clustering algorithm.

Three of the five major categories of clustering methods for static data as reviewed in the Introduction, specifically partitioning methods, hierarchical methods, and model-based methods, have been utilized directly or modified for time series clustering. Several commonly used algorithms/procedures are reviewed in more details in Section 2.1. Almost without exception each of the clustering algorithms/procedures reviewed in Section 2.1 requires a measure to compute the distance or similarity between two time series being compared. Depending upon whether the data are discrete-valued or real-valued and whether time series are of equal or unequal length, a particular measure might be more appropriate than another. Several commonly used distance/similarity measures are reviewed in more detail in Section 2.2. Most clustering algorithms/procedures are iterative in nature. Such algorithms/procedures rely on a criterion to determine when a good clustering is obtained in order to stop the iterative process. Several commonly used evaluation criteria are reviewed in more detail in Section 2.3.

2.1. Clustering algorithms/procedures

In this subsection, we briefly describe some general-purpose clustering algorithms/procedures that have been employed in the previous time series clustering studies. Interested readers should refer to the original papers for the details of specially tailored time series clustering algorithms/procedures.

2.1.1. Relocation clustering

The relocation clustering procedure has the following three steps:

Step 1: Start with an initial clustering, denoted by C, having the prescribed k number of clusters.

Step 2: For each time point compute the dissimilarity matrix and store all resultant matrices computed for all time points for the calculation of trajectory similarity.

Step 3: Find a clustering C', such that C' is better than C in terms of the *generalized Ward criterion function*. The clustering C' is obtained from C by relocating one member for C_p to C_q or by swapping two members between C_p and C_q, where $C_p, C_q \in C$, $p, q = 1, 2, \ldots, k$, and $p \neq q$. If no such clustering exists, then stop; else replace C by C' and repeat Step 3.

This procedure works only with time series with equal length because the distance between two time series at some cross sections (time points where one series does not have value) is ill defined.

2.1.2. Agglomerative hierarchical clustering

A hierarchical clustering method works by grouping data objects (time series here) into a tree of clusters. Two types of hierarchical clustering methods are often distinguished: agglomerative and divisive depending upon whether a bottom-up or top-down strategy is followed. The agglomerative hierarchical clustering method is more popular than the divisive method. It starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all the objects are in a single cluster or until certain termination conditions are satisfied. The single (complete) linkage algorithm measures the similarity between two clusters as the similarity of the closest (farthest) pair of data points belonging to different clusters, merges the two clusters having the minimum distance, repeats the merging process until all the objects are eventually merged to form one cluster. The *Ward's minimum variance algorithm* merges the two clusters that will result in the smallest increase in the value of the sum-of-squares variance. At each clustering step, all possible mergers of two clusters are tried. The sum-of-squares variance is computed for each and the one with the smallest value is selected.

The performance of an agglomerative hierarchical clustering method often suffers from its inability to adjust once a merge decision has been executed. The same is true for divisive hierarchical clustering methods. Hierarchical clustering is not restricted to cluster time series with equal length. It is applicable to series of unequal length as well if an appropriate distance measure such as dynamic time warping is used to compute the distance/similarity.

2.1.3. k-Means and fuzzy c-means

The k-means (interchangeably called c-means in this study) was first developed more than three decades ago [2]. The main idea behind it is the minimization of an objective function, which is normally chosen to be the total distance between all patterns from their respective cluster centers. Its solution relies on an iterative scheme, which starts with arbitrarily chosen initial cluster memberships or centers. The distribution of objects among clusters and the updating of cluster centers are the two main steps of the c-means algorithm. The algorithm alternates between these two steps until the value of the objective function cannot be reduced anymore.

Given n patterns $\{x_k | k = 1, \ldots, n\}$, c-means determine c cluster centers $\{v_i | i = 1, \ldots, c\}$, by minimizing the objective function given as

$$\text{Min } J_1(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik} \|x_k - v_i\|^2 \tag{1}$$

s.t. (1) $u_{ik} \in \{0, 1\} \forall i, k$, (2) $\sum_{i=1,c} u_{ik} = 1, \ \forall k$. $\| \cdot \|$ in the above equation is normally the Euclidean distance measure. However, other distance measures could also be used. The iterative solution procedure generally has the following steps:

(1) Choose $c (2 \leqslant c \leqslant n)$ and ε (a small number for stopping the iterative procedure). Set the counter $l = 0$ and the initial cluster centers, $V^{(0)}$, arbitrarily.
(2) Distribute $x_k, \forall k$ to determine $U^{(l)}$ such that J_1 is minimized. This is achieved normally by reassigning x_k to a new cluster that is closest to it.
(3) Revise the cluster centers $V^{(l)}$.
(4) Stop if the change in V is smaller than ε; otherwise, increment l and repeat Steps 2 and 3.

Dunn [20] first extended the c-means algorithm to allow for fuzzy partition, rather than hard partition, by using the objective function given in Eq. (2) below:

$$\text{Min } J_2(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^2 \|x_k - v_i\|^2. \tag{2}$$

Note that $U = [\mu_{ik}]$ in this and the following equations denotes the matrix of a fuzzy c-partition. The fuzzy c-partition constraints are (1) $\mu_{ik} \in [0, 1] \forall i, k$, (2) $\sum_{i=1,c} \mu_{ik} = 1, \ \forall k$, and (3) $0 < \sum_{k=1,n} \mu_{ik} < n, \ \forall i$. In other words, each x_k could belong to more than one cluster with each belongingness taking a fractional value between 0 and 1. Bezdek [4] generalized $J_2(U, V)$ to an infinite number of objective functions, i.e., $J_m(U, V)$, where $1 \leqslant m \leqslant \infty$. The new objective function subject to the same fuzzy c-partition constraints is

$$\text{Min } J_m(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\mu_{ik})^m \|x_k - v_i\|^2. \tag{3}$$

By differentiating the objective function with respect to v_i (for fixed U) and to μ_{ik} (for fixed V) subject to the conditions, one obtains the following two equations:

$$v_i = \frac{\sum_{k=1}^{n} (\mu_{ik})^m x_k}{\sum_{k=1}^{n} (\mu_{ik})^m}, \quad i = 1, \ldots, c. \tag{4}$$

$$\mu_{ik} = \frac{(1/\|x_k - v_i\|^2)^{1/(m-1)}}{\sum_{j=1}^{c} (1/\|x_k - v_j\|^2)^{1/(m-1)}}, \quad i = 1, \ldots, c; \quad k = 1, \ldots, n. \tag{5}$$

To solve the fuzzy c-means model, an iterative alternative optimization procedure is required. To run the procedure the number of clusters, c, and the weighting coefficient, m, must be specified. The FCM algorithm has the following steps:

(1) Choose $c(2 \leqslant c \leqslant n)$, $m(1 < m < \infty)$, and ε (a small number for stopping the iterative procedure). Set the counter $l = 0$ and initialize the membership matrix, $U^{(l)}$.
(2) Calculate the cluster center, $v_i^{(l)}$ by using Eq. (4).
(3) Update the membership matrix $U^{(l+1)}$ by using Eq. (5) if $x_k \neq v_i^{(l)}$ Otherwise, set $\mu_{jk} = 1$ (0) if $j = (\neq) i$.
(4) Compute $\Delta = \|U^{(l+1)} - U^{(l)}\|$. If $\Delta > \varepsilon$, increment l and go to Step 2. If $\Delta \leqslant \varepsilon$, stop.

This group of algorithms works better with time series of equal length because the concept of cluster centers becomes unclear when the same cluster contains time series of unequal length.

2.1.4. Self-organizing maps

Self-organizing maps developed by Kohonen [19] are a class of neural networks with neurons arranged in a low-dimensional (often two-dimensional) structure and trained by an iterative unsupervised or self-organizing procedure. The training process is initialized by assigning small random values to the weight vectors w of the neurons in the network. Each training-iteration consists of three steps: the presentation of a randomly chosen input vector from the input space, the evaluation of the network, and an update of the weight vectors. After the presentation of a pattern, the Euclidean distance between the input pattern and the weight vector is computed for all neurons in the network. The neuron with the smallest distance is marked as t. Depending upon whether a neuron i is within a certain spatial neighborhood $N_t(l)$ around t, its weight is updated according to the following updating rule:

$$w_i(l+1) = \begin{cases} w_i(l) + \alpha(l)[x(l) - w_i(l)] & \text{if } i \in N_t(l), \\ w_i(l) & \text{if } i \notin N_t(l). \end{cases} \tag{6}$$

Both the size of the neighborhood N_t and the step size of weight adaptation α shrink monotonically with the iteration. Since the neighboring neurons are updated at each step, there is a tendency that neighboring neurons in the network represent neighboring locations in the feature space. In other words, the topology of the data in the input space is preserved during the mapping. Like the group of k-means and fuzzy c-means algorithms, SOM does not work well with time series of unequal length due to the difficulty involved in defining the dimension of weight vectors.

2.2. Similarity/distance measures

One key component in clustering is the function used to measure the similarity between two data being compared. These data could be in various forms including raw values of equal or unequal length, vectors of feature-value pairs, transition matrices, and so on.

2.2.1. Euclidean distance, root mean square distance, and Mikowski distance

Let x_i and v_j each be a P-dimensional vector. The Euclidean distance is computed as

$$d_E = \sqrt{\sum_{k=1}^{P} (x_{ik} - v_{jk})^2}. \tag{7}$$

The root mean square distance (or average geometric distance) is simply

$$d_{rms} = d_E / n. \tag{8}$$

Mikowski distance is a generalization of Euclidean distance, which is defined as

$$d_M = \sqrt[q]{\sum_{k=1}^{P} (x_{ik} - v_{jk})^q}. \tag{9}$$

In the above equation, q is a positive integer. A normalized version can be defined if the measured values are normalized via division by the maximum value in the sequence.

2.2.2. Pearson's correlation coefficient and related distances

Let x_i and v_j each be a P-dimensional vector. Pearson's correlation factor between x_i and v_j, cc, is defined as

$$cc = \frac{\sum_{k=1}^{P} (x_{ik} - \mu_{x_{ik}})(v_{jk} - \mu_{v_{jk}})}{S_{x_i} S_{v_j}}, \tag{10}$$

where μ_{Xi} and S_{Xi} are, respectively, the mean and scatter of x_i, computed as below:

$$\mu_{xi} = \frac{1}{P} \sum_{k=1}^{P} x_{ik} \quad \text{and} \quad S_{xi} = \left[\sum_{k=1}^{P} (x_{ik} - \mu_{xi}) \right]^{0.5}. \tag{11}$$

Two cross-correlation-based distances used by Golay et al. [21] in the fuzzy c-means algorithm are

$$d_{cc}^1 = \left(\frac{1 - cc}{1 + cc} \right)^{\beta} \tag{12}$$

and

$$d_{cc}^2 = 2(1 - cc). \tag{13}$$

In Eq. (12), β has a similar function as m in the fuzzy c-means algorithm and take a value greater than zero.

2.2.3. Short time series distance

Considering each time series as a piecewise linear function, Möller-Levet et al. [22] proposed the STS distance as the sum of the squared differences of the slopes in two time series being compared. Mathematically, the STS distance between two time series x_i and v_j is defined as

$$d_{STS}=\sqrt{\sum_{k=1}^{P}\left(\frac{v_{j(k+1)}-v_{jk}}{t_{(k+1)}-t_k}-\frac{x_{i(k+1)}-x_{ik}}{t_{(k+1)}-t_k}\right)^2}, \qquad (14)$$

where t_k is the time point for data point x_{ik} and v_{jk}. To remove the effect of scale, z standardization of the series is recommended.

2.2.4. Dynamic time warping distance

Dynamic time warping (DTW) is a generalization of classical algorithms for comparing discrete sequences to sequences of continuous values. Given two time series, $Q=q_1, q_2, \ldots, q_i, \ldots, q_n$ and $R=r_1, r_2, \ldots, r_j, \ldots, r_m$, DTW aligns the two series so that their difference is minimized. To this end, an $n \times m$ matrix where the (i, j) element of the matrix contains the distance $d(q_i, r_j)$ between two points q_i, and r_j. The Euclidean distance is normally used. A warping path, $W=w_1, w_2, \ldots, w_k, \ldots, w_K$ where $\max(m, n) \leqslant K \leqslant m+n-1$, is a set of matrix elements that satisfies three constraints: boundary condition, continuity, and monotonicity. The boundary condition constraint requires the warping path to start and finish in diagonally opposite corner cells of the matrix. That is $w_1=(1,1)$ and $w_K=(m,n)$. The continuity constraint restricts the allowable steps to adjacent cells. The monotonicity constraint forces the points in the warping path to be monotonically spaced in time. The warping path that has the minimum distance between the two series is of interest. Mathematically,

$$d_{DTW}=\min\frac{\sum_{k=1}^{K} w_k}{K}. \qquad (15)$$

Dynamic programming can be used to effectively find this path by evaluating the following recurrence, which defines the cumulative distance as the sum of the distance of the current element and the minimum of the cumulative distances of the adjacent elements:

$$d_{cum}(i,j)=d(q_i,r_j)+\min\{d_{cum}(i-1,j-1), d_{cum}(i-1,j), d_{cum}(i,j-1)\}. \qquad (16)$$

2.2.5. Probability-based distance function for data with errors

This function was originally developed by Kumar et al. [23] in their study of clustering seasonality patterns. They defined the similarity/distance between two seasonalities, A_i and A_j, as the probability of accepting/rejecting the null hypothesis $\mathrm{H}_0 : A_i \sim A_j$. Assuming A_i and A_j, each comprised T independent samples drawn from Gaussian distributions with means x_{it} and x_{jt} and standard deviations σ_{it} and σ_{jt}, respectively, the statistic $\sum_{t=1,T}, (x_{it}-x_{jt})^2/(\sigma_{it}^2+\sigma_{jt}^2)$ follows the chi-square distribution with $T-1$ degrees of freedom.Consequently,

$$d_{ij}=\chi_{T-1}^2\left(\sum_{t=1}^{T}\frac{(x_{it}-x_{jt})^2}{\sigma_{it}^2+\sigma_{jt}^2}\right). \qquad (17)$$

The null hypothesis $A_i \sim A_j$ denotes $\mu_{it}=\mu_{jt}$ for $t=1,\ldots,T$.

2.2.6. Kullback–Liebler distance

Let P_1 and P_2 be matrices of transition probabilities of two Markov chains (MCs) with s probability distributions each and $p_{1_{ij}}$ and $p_{2_{ij}}$ be the $i->j$ transition probability in P_1 and P_2. The asymmetric Kullback–Liebler distance of two probability distributions is

$$d(p_{1_i}, p_{2_i})=\sum_{j=1}^{s} p_{1_{ij}} \log(p_{1_{ij}}/p_{2_{ij}}). \qquad (18)$$

The symmetric version of Kullback–Liebler distance of two probability distributions is

$$D(p_{1_i}, p_{2_i})=[d(p_{1_i}, p_{2_i})+d(p_{2_i}, p_{1_i})]/2. \qquad (19)$$

The average distance between P_1 and P_2 is then $D(P_1, P_2)=\sum_{i=1,s} D(p_{1i}, p_{2i})/s$.

2.2.7. J divergence and symmetric Chernoff information divergence

Let $f_T(\lambda_s)$ and $g_T(\lambda_s)$ be two spectral matrix estimators for two different stationary vector series with p dimensions and T number of time points, where $\lambda_s=2\pi s/T$, $s=1,2,\ldots,T$. The J divergence and symmetric Chernoff information divergence are computed as [24]

$$J(f_T; g_T)=\frac{1}{2}T^{-1}\sum_{s}(tr\{f_T g_T^{-1}\} + tr\{g_T f_T^{-1}\}-2p) \qquad (20)$$

and

$$JB_\alpha(f_T; g_T)=\frac{1}{2}T^{-1}\sum_{s}\left(\log\frac{|\alpha f_T+(1-\alpha)g_T|}{|g_T|} + \log\frac{|\alpha g_T+(1-\alpha)f_T|}{|f_T|}\right), \qquad (21)$$

where $0<\alpha<1$ and p is the size of spectral matrices. Both divergences are quasi-distance measures because they do not satisfy the triangle inequality property.

There is a locally stationary version of J divergence for measuring the discrepancy between two non-stationary time series. The details can be found in Refs. [25,26].

2.2.8. Dissimilarity index based on the cross-correlation function between two time series

Let $\rho_{i,j}^2(\tau)$ denote the cross-correlation between two time series x_i and v_j with lag τ. One dissimilarity index based on the cross-correlation function is defined as

$$d_{i,j} = \sqrt{(1-\rho^2_{i,j}(0)) \Big/ \sum_{\tau=1}^{\max} \rho^2_{i,j}(\tau)}, \tag{22}$$

where max is the maximum lag. The similarity counterpart of the above index can be defined as

$$s_{i,j} = \exp(-d_{i,j}). \tag{23}$$

2.2.9. Dissimilarity between two spoken words

Let x_i be a pattern representing a replication of one specific spoken word. Each pattern has an inherent duration (e.g., x_i is n_i frames long) and each frame is represented by a vector of LPC coefficients. A symmetric distance between patterns x_i and x_j, $d_{SW}(x_i, x_j)$ is defined as

$$d_{sw}(x_i, x_j) = \frac{\delta(x_i, x_j) + \delta(x_j, x_i)}{2} \tag{24}$$

and

$$\delta(x_i, x_j) = \frac{1}{n_i} \sum \log \left[\frac{(a^j_{w(k)})' R^i_k (a^j_{w(k)})}{(a^i_k)' R^i_k (a^i_k)} \right], \tag{25}$$

where a^i_k is the vector of LPC coefficients of the kth frame of pattern i, R^i_k is the matrix of autocorrelation coefficients of the kth frame of pattern i, and $'$ denotes vector transpose. The function $w(k)$ is the warping function obtained from a dynamic time warp match of pattern j to pattern i which minimizes their distance over a constrained set of possible $w(k)$.

2.3. Clustering results evaluation criteria

The performance of a time series clustering method must be evaluated with some criteria. Two different categories of evaluation criteria can be distinguished: known ground truth and unknown ground truth. The number of clusters is usually known for the former and not known for the latter. We will review only some general criteria below. Readers should refer to the original paper for each criterion specific to a particular clustering method.

2.3.1. Criteria based on known ground truth

Let G and C be the set of k ground truth clusters and those obtained by a clustering method under evaluation, respectively. The cluster similarity measure is defined as

$$Sim(G, C) = \frac{1}{k} \sum_{i=1}^{k} \max_{1 \leqslant j \leqslant k} Sim(G_i, C_j), \tag{26}$$

where

$$Sim(G_i, C_j) = \frac{2|G_i \cap C_j|}{|G_i| + |C_j|}. \tag{27}$$

$|\cdot|$ in the above equation denotes the cardinality of the elements in the set.

2.3.2. Criteria based on unknown ground truth

Two cases can be further distinguished: one assuming that the number of clusters is known a priori and the other not. The relocation clustering, k-means, and fuzzy c-means algorithms all require the number of clusters to be specified. A number of validation indices has been proposed in the past [27]. Maulik and Bandyopadhyay [28] evaluated four cluster validity indices. No time series clustering studies used any one of the validity indices to determine the appropriate number of clusters for their application.

Let P_k denote the set of all clusterings that partition a set of multivariate time series into a pre-specified k numbers of clusters. Košmelj and Batagelj [29] determined the best among all possible clusterings by the following criterion function:

$$P(C^*) = \min_{C_j \in C \in P_k} \sum_{j=1}^{k} p(C_j), \tag{28}$$

where

$$p(C_j) = \sum_{t=1}^{T} \alpha_t(C_j) p_t(C_j) \tag{29}$$

and

$$p_t(C_j) = \frac{1}{2w(C_j)} \sum_{X,Y \in C_j} w(X) w(Y) d_t(X, Y). \tag{30}$$

In the above equation, $w(X)$ represents the weight of X, $w(C_j) = \sum_{X \in C_j} w(X)$ represents the weight of cluster C_j, and $d_t(X, Y)$ the dissimilarity between X and Y at time t. By varying k, the most appropriate number of clusters is the one with minimum $P(C^*)$.

To determine the number of clusters g, Baragona [30] maximizes the following function:

$$\sum_{\omega=1}^{g} \sum_{i,j \in C\omega, i \neq j} s_{i,j}, \tag{31}$$

where $s_{i,j}$ is a similarity index as defined in Eq. (23). Information criteria such as AIC [31], BIC [32], and ICL [33] can be used if the data come from an underlying mixture of Gaussian distributions with equal isotropic covariance matrices. The optimal number of clusters is the one that yields the highest value of the information criterion.

3. Major time series clustering approaches

This paper groups previously developed time series clustering methods into three major categories depending upon whether they work directly with raw data, indirectly with features extracted from the raw data, or indirectly with models built from the raw data. The essence of each study is summarized in this section. Studies using clustering algorithms, similarity/dissimilarity measures, and evaluation criteria reviewed in Section 2.1, 2.2, and 2.3, respectively, are as italicized.

3.1. Raw-data-based approaches

Methods that work with raw data, either in the time or frequency domain, are placed into this category. The two time series being compared are normally sampled at the same interval, but their length (or number of time points) might or might not be the same.

For clustering multivariate time varying data, Košmelj and Batagelj [29] modified the *relocation clustering procedure* that was originally developed for static data. For measuring the dissimilarity between trajectories as required by the procedure, they first introduced a cross-sectional approach-based general model that incorporated the time dimension, and then developed a specific model based on the compound interest idea to determine the time-dependent linear weights. The proposed cross-sectional procedure ignores the correlations between the variables over time and works only with time series of equal length. To form a specified number of clusters, the best clustering among all the possible clusterings is the one with the minimum *generalized Ward criterion function*. Also taking the cross-sectional approach, Liao et al. [34] applied several clustering algorithms including *K-means*, fuzzy *c-means*, and genetic clustering to multivariate battle simulation time series data of unequal length with the objective to form a discrete number of battle states. The original time series data were not evenly sampled and made uniform by using the simple linear interpolation method.

Golay et al. [21] applied the *fuzzy c-means* algorithm to functional MRI data (univariate time series of equal length) in order to provide the functional maps of human brain activity on the application of a stimulus. All three different distances: the *Euclidean distance* and two *cross-correlation-based distances* were alternatively used in the algorithm. One of the two cross-correlation-based distances, d_{cc}^{1}, was found to be the best. Several data preprocessing approaches were evaluated, and the effect of number of clusters was also discussed. However, they proposed no procedure to determine the optimal number of clusters. Instead, they recommended using a large number of clusters as an initial guess, reserving the possibility of reducing this large number to obtain a clear description of the clusters without redundancy or acquisition of insignificant cluster centers.

van Wijk and van Selow [35] performed an *agglomerative hierarchical clustering* of daily power consumption data based on the *root mean square distance*. How the clusters distributed over the week and over the year were also explored with calendar-based visualization.

Kumar et al. [23] proposed a *distance function based on the assumed independent Gaussian models of data errors* and used a *hierarchical clustering* method to group seasonality sequences into a desirable number of clusters. The experimental results based on simulated data and retail data showed that the new method outperformed both k-means and Ward's method that do not consider data errors in terms of (arithmetic) average estimation error. They assumed that data used have been preprocessed to remove the effects of non-seasonal factors and normalized to enable comparison of sales of different items on the same scale.

For the analysis of dynamic biomedical image time series data, Wismüller et al. [36] showed that deterministic annealing by the minimal free energy vector quantization (VQ) could be effective. It realizes a hierarchical unsupervised learning procedure to unveil the structure of the data set with gradually increasing clustering resolution. In particular, the method was used (i) to identify activated brain regions in functional MRI studies of visual stimulation experiments, (ii) to unveil regional abnormalities of brain perfusion characterized by differences of signal magnitude and dynamics in contrast-enhanced cerebral perfusion MRI, and (iii) for the analysis of suspicious lesions in patients with breast cancer in dynamic MRI mammography data.

In their study of DNA microarray data, Möller-Levet et al. [22] proposed *short time series* (STS) *distance* to measure the similarity in shape formed by the relative change of amplitude and the corresponding temporal information of uneven sampling intervals. All series are considered sampled at the same time points. By incorporating the STS distance into the standard fuzzy c-means algorithm, they revised the equations for computing the membership matrix and the prototypes (or cluster centers), thus developed a fuzzy time series clustering algorithm.

To group multivariate vector series of earthquakes and mining explosions, Kakizawa et al. [24] applied hierarchical clustering as well as *k-means clustering*. They measured the disparity between spectral matrices corresponding to the $p \times p$ matrices of autocovariance functions of two zero-mean vector stationary time series with two quasi-distances: the *J divergence* and *symmetric Chernoff information divergence*.

Shumway [26] investigated the clustering of non-stationary time series by applying locally stationary versions of *Kullback–Leibler discrimination information measures* that give optimal time–frequency statistics for measuring the discrepancy between two non-stationary time series. To distinguish earthquakes from explosions, an *agglomerative hierarchical cluster analysis* was performed until a final set of two clusters was obtained.

Policker and Geva [37] modeled non-stationary time series with a time varying mixture of stationary sources, comparable to the continuous hidden Markov model. The fuzzy clustering procedure developed by Gath and Geva [38] was applied to a series of P data points as a set of unordered observations to compute the membership matrix for a specified number of clusters. After performing the clustering, the series is divided into a set of P/K segments with each including K data points. The temporal value of each segment belonging to each cluster is computed as the average membership values of its data points. The optimal number of clusters is determined by a temporal cluster validation criterion. If the *symmetric Kullback–Leibler distance* between all the probability function pairs is bigger than a given small threshold, then the number of clusters being tested is set as the optimal one; otherwise, retain the old optimal value. The resultant membership matrix associated with the determined number of clusters was given an interpretation as the weights in a time varying, mixture probability distribution function.

Liao [39] developed a two-step procedure for clustering multivariate time series of equal or unequal length. The first step applies the *k-means* or *fuzzy c-means* clustering algorithm to time stripped data in order to convert multivariate real-valued time series into univariate discrete-valued time series. The converted variable is interpreted as state variable process. The second step employs the *k*-means or FCM algorithm again to group the converted univariate time series, expressed as transition probability matrices, into a number of clusters. The traditional Euclidean distance is used in the first step, whereas various distance measures including the symmetric version of Kullback–Liebler distance are employed in the second step.

Table 1 summarizes the major components used in each raw-data-based clustering algorithm and the type of time series data the algorithm is for.

3.2. Feature-based approaches

Clustering based on raw data implies working with high-dimensional space—especially for data collected at fast sampling rates. It is also not desirable to work directly with the raw data that are highly noisy. Several feature-based clustering methods have been proposed to address these concerns. Though most feature extraction methods are generic in nature, the extracted features are usually application dependent. That is, one set of features that work well on one application might not be relevant to another. Some studies even take another feature selection step to further reduce the number of feature dimensions after feature extraction.

With the objectives to develop an automatic clustering algorithm, which could be implemented for any user with a minimal amount of knowledge about clustering procedures, and to provide the template sets as accurate as those created by other clustering algorithms, Wilpon and Rabiner [40] modified the standard *k*-means clustering algorithm for the recognition of isolated words. The modifications address problems such as how to obtain cluster centers, how to split clusters to increase the number of clusters, and how to create the final cluster representations. Each pattern representing a replication of one specific spoken word has an inherent duration (e.g., n_i frames long), and each frame is a vector of linear predictive coding (LPC) coefficients. To measure the distance between two spoken word patterns, *a symmetric distance measure* was defined based on the Itakura distance for measuring the distance between two frames. The proposed modified *k*-means (MKM) clustering algorithm was shown to outperform the well-established unsupervised without averaging (UWA) clustering algorithm at that time.

Shaw and King [41] clustered time series indirectly by applying two hierarchical clustering algorithms, the *Ward's minimum variance algorithm* and the *single linkage algorithm*, to normalized spectra (normalized by the amplitude of the largest peak). The spectra were constructed from the original time series with the means adjusted to zero. The principal component analysis (PCA) filtered spectra were also clustered; it was found that using 14 most significant eigenvectors could achieve comparable results. The Euclidean distance was used.

Goutte et al. [42] clustered fMRI time series (P slices of images) in groups of voxels with similar activations using two algorithms: *k-means* and *Ward's hierarchical clustering*. The *cross-correlation function* between the fMRI activation and the paradigm (or stimulus) was used as the feature space, instead of the raw fMRI time series. For each voxel *j* in the image, y_j denotes the measured fMRI time series and *p* is the activation stimulus (assumed a square wave but not limited to), common to all *j*. The cross-correlation function is defined as

$$x_j(t) = \frac{1}{P} \sum_{u=1}^{P} y_j(u) p(u-t), \quad -T < t < T, \tag{32}$$

where $p(i) = 0$ for $i < 0$ or $i > P$ and T is of the order of the stimulus period. In a subsequent paper Goutte et al. [43] further illustrated the potential of the feature-based clustering method. First, they used only two features, namely the delay and strength of activation measured on a voxel-by-voxel basis to show that one could identify the regions with significantly different delays and activations. Using the *k*-means algorithm, they investigated the performance of three information criteria including AIC [31], BIC [32], and ICL [33] for determining the optimal number of clusters. It was found that ICL was most parsimonious and AIC tended to overestimate. Then, they showed that feature-based clustering could be used as a meta-analysis tool in evaluating the similarities and differences of the results obtained by several individual voxel analyses. In this case, features are results of previous analyses performed on the data.

Fu et al. [44] described the use of self-organizing maps for grouping data sequences segmented from the numerical time series using a continuous sliding window with the aim to discover similar temporal patterns dispersed along the time series. They introduced the perceptually important point (PIP) identification algorithm to reduce the dimension of the input data sequence *D* in accordance with the query sequence *Q*. The distance measure between the PIPs found in *D* and *Q* was defined as the sum of the mean squared distance along the vertical scale (the magnitude) and that along the horizontal scale (time dimension). To process multiresolution patterns, training patterns from different resolutions are grouped into a set of training samples to which the SOM clustering process is applied only once. Two enhancements

Table 1
Summary of raw-data-based time series clustering algorithms

Paper	Variable	Length	Distance measure	Clustering algorithm	Evaluation criterion	Application
Golay et al. [21]	Single	Equal	Euclidean and two cross-correlation-based	Fuzzy c-means	Within cluster variance	Functional MRI brain activity mapping
Kakizawa et al. [24]	Multiple	Equal	J divergence and symmetric Chernoff information divergence	Agglomerative hierarchical	N/A	Earthquakes and mining explosions
Košmelj and Batagelj [29]	Multiple	Equal	Euclidean	Modified relocation clustering procedure	Generalized Ward criterion function	Commercial energy consumption
Kumar et al. [23]	Single	Equal	Based on the assumed independent Gaussian models of data errors	Agglomerative hierarchical	N/A	Seasonality pattern in retails
Liao [39]	Multiple	Equal & unequal	Euclidean and symmetric version of Kullback–Liebler distance	k-Means and fuzzy c-Means	Within cluster variance	Battle simulations
Liao et al. [34]	Single	Equal & unequal	DTW	k-Medoids-based genetic clustering	Several different fitness functions	Battle simulations
Möller-Levet et al. [22]	Single	Equal	Short time series (STS) distance	Modified fuzzy c-means	Within cluster variance	DNA microarray
Policker and Geva [37]	Single	Equal	Euclidean	Fuzzy clustering by Gath and Geva	Symmetric Kullback–Leibler distance between probability function pairs	Sleep EEG signals
Shumway [26]	Multiple	Equal	Kullback–Leibler discrimination information measures	Agglomerative hierarchical	N/A	Earthquakes and mining explosions
Van Wijk and van Selow [35]	Single	Equal	Root mean square	Agglomerative hierarchical	N/A	Daily power consumption
Wismüller et al. [36]	Single	Equal	N/A	Neural network clustering performed by a batch EM version of minimal free energy vector quantization	Within cluster variance	Functional MRI brain activity mapping

were made to the SOM: filtering out those nodes (patterns) in the output layer that did not participate in the recall process and consolidating the discovered patterns with a relatively more general pattern by a redundancy removal step.

An algorithm called sequence cluster refinement algorithm (SCRA) was developed by Owsley et al. [45] to group machine tool monitoring data into clusters represented as discrete hidden Markov models (HMM), with each reflecting a kind of health status of the tool. The developed algorithm differs from the generalized Lloyd algorithm, a vector quantization algorithm, in representing the clusters as HMMs' instead of template vectors. Instead of processing the entire raw data, the transient events in the bulk data signal are first detected by template matching. A high-resolution time–frequency representation of the transient region is then formed. To reduce dimension, they modified the self-organizing feature map algorithm in order to improve its generalization abilities.

Vlachos et al. [46] presented an approach to perform incremental clustering of time series at various resolutions using the Haar wavelet transform. First, the Haar wavelet decomposition is computed for all time series. Then, the k-means clustering algorithm is applied, starting at the coarse level and gradually progressing to finer levels. The final centers at the end of each resolution are reused as the initial centers for the next level of resolution. Since the length of the data reconstructed from the Haar decomposition doubles as we progress to the next level, each coordinate of the centers at the end of level i is doubled to match the dimensionality of the points on level $i+1$. The clustering error is computed at the end of each level as the sum of number of incorrectly clustered objects for each cluster divided by the cardinality of the dataset.

Table 2 summarizes major components used in each feature-based clustering algorithm. They all can handle series with unequal length because the feature extraction operation takes care of the issue. For a multivariate time series, features extracted can simply be put together or go through some fusion operation to reduce the dimension and improve the quality of the clustering results, as in classification studies.

3.3. Model-based approaches

This class of approaches considers that each time series is generated by some kind of model or by a mixture of underlying probability distributions. Time series are considered similar when the models characterizing individual series or the remaining residuals after fitting the model are similar.

For clustering or choosing from a set of dynamic structures (specifically the class of ARIMA invertible models), Piccolo [47] introduced the Euclidean distance between their corresponding autoregressive expansions as the metric. The metric satisfies the classical properties of a distance, i.e., non-negativity, symmetry, and triangularity. In addition, six properties of the metric were discussed. The distance matrix between pairs of time series models was then processed by a *complete linkage clustering method* to construct the dendrogram.

Baragona [30] evaluated three meta-heuristic methods for partitioning a set of time series into clusters in such a way that (i) the cross-correlation maximum absolute value between each pair of time series that belong to the same cluster is greater than some given threshold, and (ii) the k-min cluster criterion is minimized with a specified number of clusters. The cross-correlations are computed from the residuals of the models of the original time series. Among all methods evaluated, Tabu search was found to perform better than single linkage, pure random search, simulation annealing and genetic algorithms based on a simulation experiment on ten sets of artificial time series generated from low-order univariate and vector ARMA models.

Motivated by questions raised in the context of musical performance theory, Beran and Mazzola [48] defined hierarchical smoothing models (or HISMOOTH models) to understand the relationship between the symbolic structure of a music score and its performance, with each represented by a time series. The models are characterized by a hierarchy of bandwidths and a vector of coefficients. Given n performances and a common explanatory time series, the estimated bandwidth values are then used in clustering using the S-plus functions *plclust* and *hclust* that plots the clustering tree structure produced by agglomerative hierarchical clustering.

Maharaj [49] developed an agglomerative hierarchical clustering procedure that is based on the p-value of a test of hypothesis applied to every pair of given stationary time series. Assuming that each stationary time series can be fitted by a linear AR(k) model denoted by a vector of parameters $\pi' = [\pi_1, \pi_2, \ldots, \pi_k]$, a chi-square distributed test statistic was derived to test the null hypothesis that there is no difference between the generating processes of two stationary time series or H_0: $\pi_x = \pi_y$. Two series are grouped together if the associated p-value is greater than the pre-specified significance level. The clustering result is evaluated with a measure of discrepancy, which is defined as the difference between the actual number of clusters and the number of exactly correct clusters generated.

Ramoni et al. [50] presented BCD: a Bayesian algorithm for clustering by dynamics. Given a set S of n numbers of univariate discrete-valued time series, BCD transforms each series into a Markov chain (MC) and then clusters similar MCs to discover the most probable set of generating processes. BCD is basically an unsupervised agglomerative clustering method. Considering a partition as a hidden discrete variable C, each state C_k of C represents a cluster of time series, and hence determines a transition matrix. The task of clustering is regarded as a Bayesian model selection problem with the objective to select the model with the maximum posterior probability. Since the same data are used to compare all models and all models are equally likely, the comparison can be based on the marginal

Table 2
Summary of feature-based time series clustering algorithms

Paper	Variable	Features	Feature selection	Distance measure	Clustering algorithm	Evaluation criterion	Application
Fu et al. [44]	Single	Perceptually important points	No	Sum of the mean squared distance along the vertical and horizontal scales	Modified SOM	Expected squared error	Hong Kong stock market
Goutte et al. [42]	Single	Cross-correlation function	No	Euclidean	Agglomerative hierarchical and k-means	N/A and Within cluster variance	Functional MRI brain activity mapping
Owsley et al. [45]	Single	Time-frequency representation of the transient region	Modified SOM	Euclidean	Modified k-means (Sequence cluster refinement)	Within cluster variance	Tool condition monitoring
Shaw and King [41]	Single	Normalized spectra	PCA	Euclidean	Agglomerative hierarchical	N/A	Flow velocity in a wind tunnel
Vlachos et al. [46]	Single	Haar wavelet transform	No	Euclidean	Modified k-means (called I-k-means)	Within cluster variance	Non-specific
Wilpon and Rabiner [40]	Single	LPC coefficients	No	A symmetric measure based on the Itakura distance	Modified k-means	Within cluster variance	Isolated word recognition

likelihood $p(S|MC)$, which is a measure of how likely the data are if the model MC is true. The similarity between two estimated transition matrices is measured as an average of the *symmetrized Kullback–Liebler distance* between corresponding rows in the matrices. The clustering result is evaluated mainly by a measure of the loss of data information induced by clustering, which is specific to the proposed clustering method. They also presented a Bayesian clustering algorithm for multivariate time series [51]. The algorithm searches for the most probable set of clusters given the data using a similarity-based heuristic search method. The measure of similarity is an average of the Kullback–Liebler distances between comparable transition probability tables. The similarity measure is used as a heuristic guide for the search process rather than a grouping criterion. Both the grouping and stopping criteria are based on the posterior probability of the obtained clustering. The objective is to find a maximum posterior probability partition of set of MCs.

Kalpakis et al. [52] studied the clustering of ARIMA time-series, by using the Euclidean distance between the Linear Predictive Coding cepstra of two time-series as their dissimilarity measure. The cepstral coefficients for an AR(p) time series are derived from the auto-regression coefficients. The partition around medoids method [3] that is a k-medoids algorithm was chosen, with the clustering results evaluated with the cluster similarity measure and Silhouette width. Based on a test of four data sets, they showed that the LPC cepstrum provides higher discriminatory power to tell one time series from another and superior clusterings than other widely used methods such as the Euclidean distance between (the first 10 coefficients of) the DFT, DWT, PCA, and DFT of the auto-correlation function of two time series.

Xiong and Yeung [53] proposed a model-based method for clustering univariate ARIMA series. They assumed that the time series are generated by k different ARMA models, with each model corresponds to one cluster of interest. An expectation-maximization (EM) algorithm was used to learn the mixing coefficients as well as the parameters of the component models that maximize the expectation of the complete-data log-likelihood. In addition, the EM algorithm was improved so that the number of clusters could be determined automatically. The evaluation criterion used is the *cluster similarity measure* detailed in Section 5.1.1. The proposed method was compared with that of Kalpakis et al. using the same four datasets.

Assuming the Gaussian mixture model for speaker verification, Tran and Wagner [54] proposed a fuzzy c-means clustering-based normalization method to find 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 of all the likelihood values of the background speakers in current normalization methods. Let λ_0 be the claimed speaker model and λ_i, $i = 1, \ldots, B$, be a model representing another possible speaker model and B is the total number of "background" speaker models. $P(X|\lambda_0)$ and $P(X|\lambda_i)$ are the likelihood functions of the claimed speaker and an impostor, respectively, for a given input utterance X. The FCM membership score is defined as follows:

$$S(X) = \left\{ \sum_{i=1}^{B} [\log P(X|\lambda_0) / \log P(X|\lambda_i)]^{1/(m-1)} \right\}^{-1}. \tag{33}$$

Biernacki et al. [33] proposed an integrated completed likelihood (ICL) criterion for choosing a Gaussian mixture model and a relevant number of clusters. The ICL criterion is essentially the ordinary BIC penalized by the subtraction of the estimated mean entropy. Numerical experiments with simulated and real data showed that the ICL criterion seems to overcome the practical possible tendency of Bayesian information criterion (BIC) to overestimate the number of clusters.

Considering that a set of multivariate, real-valued time series is generated according to hidden Markov models, Oates et al. [55] presented a hybrid clustering method for automatically determining the k number of generating HMMs, and for learning the parameters of those HMMs. A standard hierarchical, agglomerative clustering algorithm was first applied to obtain an initial estimate of k and to form the initial clusters using dynamic time warping to assess the similarity. These initial clusters serve as the input to a process that trains one HMM on each cluster and iteratively moves time series between clusters based on their likelihoods given the various HMMs.

Li and Biswas [56] described a clustering methodology for temporal data using the hidden Markov model representation. The temporal data are assumed to have Markov property, and may be viewed as the result of a probabilistic walk along a fixed set of (not directly observable) states. The proposed continuous HMM clustering method can be summarized in terms of four levels of nested searches. From the outer most to the inner most levels, they are the search for (1) the number of clusters in a partition based on the partition mutual information (PMI) measure, (2) the structure for a given partition size according to the k-means or depth-first binary divisive clustering, (3) the HMM structure for each cluster that gives the highest marginal likelihood based on the BIC and the Cheeseman–Stutz approximation, and (4) the parameters for each HMM structure according to the segmental k-means procedure. For the second search level, the sequence-to-model likelihood distance measure was chosen for object-to-cluster assignments. The HMM refinement procedure for the third-level search starts with an initial model configuration and incrementally grows or shrinks the model through HMM state splitting and merging operations. They generated an artificial data set from three random generative models: one with three states, one with four states, and one with five states, and showed that their method could reconstruct the HMM with the correct model size and near perfect model parameter values. Li et al. [57] presented a Bayesian HMM clustering algorithm that uses BIC as the model selection criterion in levels 1 and 3 and exploits the monotonic characteristics of the BIC function to develop a sequential search strategy. The strategy starts with the simplest model, gradually increases the model size, and stops when the BIC score of the current model is less than that of the previous model. Experimental results using both artificially generated data and ecology data showed the effectiveness of the clustering methodology.

A framework was presented by Wang et al. [58] for tool wear monitoring in a machining process using discrete hidden Markov models. The feature vectors are extracted from the vibration signals measured during turning operations by wavelet analysis. The extracted feature vectors are then converted into a symbol sequence by vector quantization, which in turn is used as input for training the hidden Markov model by the expectation maximization approach.

Table 3 summarizes the major components used in each model-based clustering algorithm. Like feature-based methods, model-based methods are capable of handling series with unequal length as well through the modeling operation. For those methods that use log-likelihood as the distance measure, the model with the highest likelihood is concluded to be the cluster for the data being tested.

4. Discussion

Among all the papers surveyed the studies of Ramoni et al. [50,51] are the only two that assumed discrete-valued time series data. The work of Kumar et al. [23] is the only one that takes data error into account. Most studies address evenly sampled data while Möller-Levet et al. [22] are the only ones who consider unevenly sampled data. Note that some studies such as Maharaj [49] and Baragona [30] are restricted to stationary time series only whereas most others are not. None of the papers included in this survey handle multivariate time series data with different length for each variable.

Several studies including Košmelj and Batagelj [29] and Kumar et al. [23] made the assumption that the T samples of a time series are independent (come from independent distribution), ignoring the correlations in consecutive sample values in time. Modeling a time series by a (first order) Markov chain as done by Ramoni et al. [50,51] assumes that the probability of a variable at time t is dependent upon only the variable values at time $t-1$ and independent of the variable values observed prior to time $t-1$. The hidden Markov models provide a richer representation of time series, especially for systems where their real states are not observable and the observation is a probability function of the state. Note that both studies of using HMM models for the multidimensional case assumed that temporal features are independent [55,57]. In the case that time series data has a longer memory, higher orders of Markov chain or hidden Markov models should be considered.

Table 3
Summary of model-based time series clustering algorithms

Paper	Variable	Model	Model output of interest	Distance measure	Clustering algorithm	Evaluation Criterion	Application
Baragona [30]	Single and multiple	ARMA	Residuals	Cross-correlation based	Tabu search, GA, and simulated annealing	Specially designed	Non-specific
Beran and Mazzola [48]	Single	Hierarchical smoothing models	Coefficients	Unknown (most likely Euclidean)	Agglomerative hierarchical	N/A	Music performance
Biernacki et al. [33]	Multiple	Gaussian mixture	Parameters	Log-likelihood	EM algorithm	Log-likelihood	Non-specific
Kalpakis et al. [52]	Single	AR	LPC coefficients of AR coefficients	Euclidean	Partition around medoids	Cluster similarity metric and Silhouette width	Public data
Li and Biswas [56]	Multiple	Continuous HMM	HMM parameters	Log-likelihood	Four nested levels of search	Partition mutual information	Non-specific
Li et al. [57]	Multiple	Continuous HMM	HMM parameters	Log-likelihood	Four nested levels of search	Partition mutual information	Ecology
Maharaj [49]	Single	ARCoefficients	P-value of hypothesis testing	Agglomerative hierarchical	N/A	Number of dwelling units financed	
Oates et al. [55]	Multiple	Discrete HMM (discretized by SOM)	HMM parameters	Log-likelihood	Initialized by DTW followed by a fixed point operation	Log-likelihood	Robot sensor data
Piccolo [47]	Single	AR(∞)	Coefficients	Euclidean	Agglomerative hierarchical	N/A	Industrial production indices
Ramoni et al. (2001)	Single discrete-valued	Markov Chain	Transition probabilities	Symmetrized Kullback–Liebler distance	Agglomerative clustering	N/A	Robot sensor data
Ramoni et al. [51]	Multiple discrete-valued	Markov Chain	Transition probabilities	Symmetrized Kullback–Liebler to guide search and Posterior probability as a grouping criterion	Agglomerative clustering	Marginal likelihood of a partition	Robot sensor data
Tran and Wagner [54]	Single	Gaussian mixture	Cepstral coefficients	Log-likelihood	Modified fuzzy *c*-means	Within cluster variance	Speaker verification
Wang et al. [58]	Single (discretized by vector quantization from wavelet coefficients)	Discrete HMM	HMM parameters	Log-likelihood	EM learning	Log-likelihood	Tool conditionmonitoring
Xiong and Yeung [53]	Single	ARMA mixture	Coefficients	Log-likehood	EM learning	Cluster similarity metric	Public data

The majority of time series clustering studies are restricted to univariate time series. Among all the univariate time series clustering studies, most work only with the observed data with a few exceptions. One exception assumes that for all the observed time series there is a common explanatory time series [48] while some others assume that the observed time series are generated based on a known stimulus [21,36,42]. The studies that addressed multivariate (or vector) time series include Košmelj and Batagelj [29], Kakizawa et al. [24], Ramoni et al. [51], Oates et al. [55], Li et al. [57], etc. Some multivariate time series clustering studies assume that there is no cross-correlation between variables. Those based on the probability framework simplify the overall joint distribution by assuming conditional independence between variables [51].

In some cases, time series clustering could be complemented with a change-point detection algorithm in order to automatically and correctly identify the start times (or the origins) of the time series before they are matched and compared. This point was brought up by Shumway [26] in their clustering study of earthquakes and mining explosions at regional distances.

All in all, clustering of time series data differs from clustering of static feature data mainly in how to compute the similarity between two data objects. Depending upon the types and characteristics of time series data, different clustering studies provide different ways to compute the similarity/dissimilarity between two time series being compared. Once the similarity/dissimilarity of data objects is determined, many general-purpose clustering algorithms can be used to partition the objects, as reviewed in Section 2.1. Therefore, for any given time series clustering application, the key is to understand the unique characteristics of the subject data and then to design an appropriate similarity/dissimilarity measure accordingly.

We note that to date very few time series clustering studies made use of more recently developed clustering algorithms such as genetic algorithms (the work of Baragona [30] is the only exception). It might be interesting to investigate the performance of GA in time series clustering. There seems to be either no or insufficient justification why a particular approach was taken in the previous studies. We also note that there is a lack of studies to compare different time series clustering approaches. In the case where more than one approach is possible, it is desirable to know which approach is better.

It has been shown beneficial to integrate an unsupervised clustering algorithm with a supervised classification algorithm for static feature data [59]. There is no reason why the same cannot true for time series data. An investigation on such integration is thus warranted. It also has been reported that an ensemble of classifiers could be more effective than an individual classifier [60,61]. Will an ensemble of clustering algorithms be as effective? It might be interesting to find out as well. A problem that an ensemble of classifiers does not have is the random labeling of the clustering results, which most likely would create problems in attempting to compile the clustering results obtained by each clustering algorithm together and a solution must be found.

It should be noted that most scaled-up clustering algorithms developed to handle larger data sets consider only static data. Examples are clustering large applications (CLARA) proposed by Kaufman and Rousseeuw [3], clustering large applications based on randomized search (CLARANS) developed by Ng and Han [62], and pattern count-tree based clustering (PCBClu) described by Ananthanarayana et al. [63]. It is definitely desirable to develop scaled-up time series clustering algorithms as well in order to handle large data sets. In this regard, most efforts have been devoted to data representation using so called segmentation algorithms [64]. The work of Fu et al. [44] for finding the PIPs is a segmentation algorithm.

Finally, time series clustering is part of a recent survey of temporal knowledge discovery by Roddick et al. [65]. In that survey, they discussed only a few research works related to time series clustering in less than two pages. Our survey intends to provide a more detailed review of this expanding research area.

5. Concluding remarks

In this paper we surveyed most recent studies on the subject of time series clustering. These studies are organized into three major categories depending upon whether they work directly with the original data (either in the time or frequency domain), indirectly with features extracted from the raw data, or indirectly with models built from the raw data. The basics of time series clustering, including the three key components of time series clustering studies are highlighted in this survey: the clustering algorithm, the similarity/dissimilarity measure, and the evaluation criterion. The application areas are summarized with a brief description of the data used. The uniqueness and limitation of past studies, and some potential topics for future study are also discussed.

Appendix A. Previous applications and data used

The studies focusing on the development of new methods usually do not have a particular application in mind. For testing a new method and for comparing existing methods, the researchers normally either generate simulated data or rely on public-accessible time series data depositories such as the UCR time series data mining archive [http://www.cs.ucr.edu/∼eamonn/TSDMA/index.html].

Other studies set out to investigate issues directly related to a particular application. As can be seen in the below summary, clustering of time series data is necessary in widely different applications.

A.1. Business and socio-economics

- Clustering seasonality patterns of retail data [22]. The sales data from several departments of a major retail chain were used in the study.
- Cluster analysis of country's energy consumption [29]. The data used are the energy consumption of 23 European countries in the years 1976–1982.
- Discovering consumer power consumption patterns for the segmentation of markets [35]. The data used are daily power demands at a research facility.
- Discovery patterns from stock time series [44]. The Hong Kong stock exchange data were used in their study.
- To cluster the number of dwelling units financed from January 1978 to March 1998 for all states and territories in Australia [49].
- Clustering population data [52]. The data used were the population estimates from 1900–1999 in 20 states of the US (http://www.census.gov/population/www/estimates/st_stts.html).
- Clustering personal income data [52]. They used a collection of time series representing the per capita personal income from 1929 to 1999 in 25 states of the USA (http://www.bea.gov/bea/regional/spi).

A.2. Engineering

- Unsupervised learning of different activities by mobile robots [51].
- Identifying similar velocity flows [42]. They used 42 512-point time series collected in a special wind tunnel setup.
- Speech recognition [40,54]. Wilpon and Rabiner used a set of speech data consisting of 4786 isolated patterns form a digits vocabulary. Tran and Wagner performed experiments on the T146 and the ANDOSL speech corpora.
- Space Shuttle maneuver monitoring [46]. They extracted time series of length 512, at random starting points of each sequence from the original data collected from various inertial sensors from Space Shuttle mission STS-57.
- Two health monitoring [45,58]. Wang et al. [58] performed cutting tests to collect vibration signals for detecting two tool states: sharp and worn.

A.3. Science

- Clustering temperature data [52]. They used 30 time series of the daily temperature in the year 2000 in various places in Florida, Tennessee, and Cuba taken from the National Climatic Data Center (http://www.ncdc.noaa.gov/rcsg/datasets.html).
- Grouping genes [22]. The authors used a subset of microarray data available from http://cmgm.standford.edu/pbrown/sporulation
- Discriminating seismic data from different sources [26].
- Cluster models of ecological dynamics [57]. They used data collected from 30 sites on a salt marsh area south of Brisbane, Australia. At each site, four measurements of species performance and 11 environmental measurements were collected.

A.4. Medicine

- Clustering fMRI time series for identifying regions with similar patterns of activation [21,42,43]. The data used in their studies were experimentally acquired with an MR scanner.
- Clustering ECG data [52]. The data were taken from the ECG database at PhysioNet (http://www.physionet.org/physiobank/database).

A.5. Art and entertainment

- Clustering musical performances with a similar “style” [48]. They analyzed 28 tempo curves from performances of Schumann's Träumerei op. 15/7.

References

[1] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 2001 pp. 346–389.

[2] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: L.M. LeCam, J. Neyman (Eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297.

[3] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, New York, 1990.

[4] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York and London, 1987.

[5] R. Krishnapuram, A. Joshi, O. Nasraoui, L. Yi, Low-complexity fuzzy relational clustering algorithms for web mining, IEEE Trans. Fuzzy Systems 9 (4) (2001) 595–607.

[6] R. Krishnapuram, J. Kim, A note on the Gustafson–Kessel and adaptive fuzzy clustering algorithms, IEEE Trans. Fuzzy Systems 7 (4) (1999) 453–461.

[7] K. Krishna, M.N. Murty, Genetic k-means algorithms, IEEE Trans. Syst. Man Cybernet.-B: Cybernet. 29 (3) (1999) 433–439.

[8] L. Meng, Q.H. Wu, Z.Z. Yong, A genetic hard c-means clustering algorithm, Dyn. Continuous Discrete Impulsive Syst. Ser. B: Appl. Algorithms 9 (2002) 421–438.

[9] V. Estivill-Castro, A.T. Murray, Spatial clustering for data mining with genetic algorithms, http://citeseer.nj.nec.com/estivill-castro97spatial.html.

[10] L.O. Hall, B. Özyurt, J.C. Bezdek, Clustering with a genetically optimized approach, IEEE Trans. Evolutionary Computat. 3 (2) (1999) 103–112.

[11] G. Karypis, E.-H. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer August (1999) 68–75.

[12] S. Guha, R. Rastogi, K. Shim, CURE: an efficient clustering algorithm for large databases, Proceedings of the 1998 ACM-SIGMOD International Conference on Management of Data, Seattle, WA, June 1998, pp. 73–84.

[13] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: an efficient data clustering method for very large databases, Proceedings of the 1996 ACM-SIGMOD International Conference on Management of Data, Montreal, Canada, June 1996, pp. 103–114.

[14] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases, Proceedings of the 1996 International Conference on Knowledge Discovery and Data Mining (KDD'96), Portland, OR, 1996, pp. 226–231.

[15] M. Ankerst, M. Breunig, H.-P. Kriegel, J. Sander, OPTICS: ordering points to identify the clustering structure, Proceedings of the 1999 ACM-SIGMOD International Conference on Management of Data, Philadelphia, PA, June 1999, pp. 49–60.

[16] W. Wang, J. Yang, R. Muntz, R., STING: a statistical information grid approach to spatial data mining, Proceedings of the 1997 International Conference on Very Large Data Base (VLDB'97), Athens, Greek, 1997, pp. 186–195.

[17] P. Cheeseman, J. Stutz, Bayesian classification (AutoClass): theory and results, in: U.M. Fayyard, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Cambridge, MA, 1996.

[18] G.A. Carpenter, S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, Comput. Vision Graphics Image Process. 37 (1987) 54–115.

[19] T. Kohonen, The self organizing maps, Proc. IEEE 78 (9) (1990) 1464–1480.

[20] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, J. Cybernet. 3 (1974) 32–57.

[21] X. Golay, S. Kollias, G. Stoll, D. Meier, A. Valavanis, P. Boesiger, A new correlation-based fuzzy logic clustering algorithm for fMRI, Mag. Resonance Med. 40 (1998) 249–260.

[22] C.S. Möller-Levet, F. Klawonn, K.-H. Cho, O. Wolkenhauer, Fuzzy clustering of short time series and unevenly distributed sampling points, Proceedings of the 5th International Symposium on Intelligent Data Analysis, Berlin, Germany, August 28–30, 2003.

[23] M. Kumar, N.R. Patel, J. Woo, Clustering seasonality patterns in the presence of errors, Proceedings of KDD '02, Edmonton, Alberta, Canada.

[24] Y. Kakizawa, R.H. Shumway, N. Taniguchi, Discrimination and clustering for multivariate time series, J. Amer. Stat. Assoc. 93 (441) (1998) 328–340.

[25] R. Dahlhaus, On the Kullback–Leibler information divergence of locally stationary processes, Stochastic Process. Appl. 62 (1996) 139–168.

[26] R.H. Shumway, Time–frequency clustering and discriminant analysis, Stat. Probab. Lett. 63 (2003) 307–314.

[27] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybernet. B: Cybernet. 28 (3) (1998) 301–315.

[28] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1650–1654.

[29] K. Košmelj, V. Batagelj, Cross-sectional approach for clustering time varying data, J. Classification 7 (1990) 99–109.

[30] R. Baragona, A simulation study on clustering time series with meta-heuristic methods, Quad. Stat. 3 (2001) 1–26.

[31] H. Akaike, A new look at the statistical model identification, IEEE Trans. Automat. Control 19 (1974) 716–723.

[32] G. Schwartz, Estimating the dimension of a model, Ann. Stat. 6 (1978) 461–464.

[33] C. Biernacki, G. Celeux, G. Govaert, Assessing a mixture model for clustering with the integrated completed likelihood, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 719–725.

[34] T.W. Liao, B. Bolt, J. Forester, E. Hailman, C. Hansen, R.C. Kaste, J. O'May, Understanding and projecting the battle state, 23rd Army Science Conference, Orlando, FL, December 2–5, 2002.

[35] J.J. van Wijk, E.R. van Selow, Cluster and calendar based visualization of time series data, Proceedings of IEEE Symposium on Information Visualization, San Francisco, CA, October 25–26, 1999.

[36] A. Wismüller, O. Lange, D.R. Dersch, G.L. Leinsinger, K. Hahn, B. Pütz, D. Auer, Cluster analysis of biomedical image time series, Int. J. Comput. Vision 46 (2) (2002) 103–128.

[37] S. Policker, A.B. Geva, Nonstationary time series analysis by temporal clustering, IEEE Trans. Syst. Man Cybernet.-B: Cybernet. 30 (2) (2000) 339–343.

[38] I. Gath, A.B. Geva, Unsupervised optimal fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 7 (1989) 773–781.

[39] T.W. Liao, Mining of vector time series by clustering, Working paper, 2005.

[40] J.G. Wilpon, L.R. Rabiner, Modified *k*-means clustering algorithm for use in isolated word recognition, IEEE Trans. Acoust. Speech Signal Process. 33 (3) (1985) 587–594.

[41] C.T. Shaw, G.P. King, Using cluster analysis to classify time series, Physica D 58 (1992) 288–298.

[42] C. Goutte, P. Toft, E. Rostrup, On clustering fMRI time series, Neuroimage 9 (3) (1999) 298–310.

[43] C. Goutte, L.K. Hansen, M.G. Liptrot, E. Rostrup, Feature-space clustering for fMRI meta-analysis, Hum. Brain Mapping 13 (2001) 165–183.

[44] T.-C. Fu, F.-L. Chung, V. Ng, R. Luk, Pattern discovery from stock time series using self-organizing maps, KDD 2001 Workshop on Temporal Data Mining, August 26–29, San Francisco, 2001, pp. 27–37.

[45] L.M.D. Owsley, L.E. Atlas, G.D. Bernard, Self-organizing feature maps and hidden Markov models for machine-tool monitoring, IEEE Trans. Signal Process. 45 (11) (1997) 2787–2798.

[46] M. Vlachos, J. Lin, E. Keogh, D. Gunopulos, A wavelet-based anytime algorithm for *k*-means clustering of time series, Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, May 1–3, 2003.

[47] D. Piccolo, A distance measure for classifying ARMA models, J. Time Ser. Anal. 11 (2) (1990) 153–163.

[48] J. Beran, G. Mazzola, Visualizing the relationship between time series by hierarchical smoothing models, J. Comput. Graph. Stat. 8 (2) (1999) 213–238.

[49] E.A. Maharaj, Clusters of time series, J. Classification 17 (2000) 297–314.

[50] M. Ramoni, P. Sebastiani, P. Cohen, Bayesian clustering by dynamics, Mach. Learning 47 (1) (2002) 91–121.

[51] M. Ramoni, P. Sebastiani, P. Cohen, Multivariate clustering by dynamics, Proceedings of the 2000 National Conference on Artificial Intelligence (AAAI-2000), San Francisco, CA, 2000, pp. 633–638.

[52] K. Kalpakis, D. Gada, V. Puttagunta, Distance measures for effective clustering of ARIMA time-series, Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, November 29–December 2, 2001, pp. 273–280.

[53] Y. Xiong, D.-Y. Yeung, Mixtures of ARMA models for model-based time series clustering, Proceedings of the IEEE International Conference on Data Mining, Maebaghi City, Japan, 9–12 December, 2002.

[54] D. Tran, M. Wagner, Fuzzy c-means clustering-based speaker verification, in: N.R. Pal, M. Sugeno (Eds.), AFSS 2002, Lecture Notes in Artificial Intelligence, 2275, 2002, pp. 318–324.

[55] T. Oates, L. Firoiu, P.R. Cohen, Clustering time series with hidden Markov models and dynamic time warping, Proceedings of the IJCAI-99 Workshop on Neural, Symbolic, and Reinforcement Learning Methods for Sequence Learning.

[56] C. Li, G. Biswas, Temporal pattern generation using hidden Markov model based unsupervised classification, in: D.J. Hand, J.N. Kok, M.R. Berthold (Eds.), Lecture Notes in Computer Science, vol. 164, IDA '99, Springer, Berlin, 1999, pp. 245–256.

[57] C. Li, G. Biswas, M. Dale, P. Dale, Building models of ecological dynamics using HMM based temporal data clustering—a preliminary study, in: F. Hoffmann et al. (Eds.), IDA 2001, Lecture Notes in Computer Science, vol. 2189, 2001, pp. 53–62.

[58] L. Wang, M.G. Mehrabi, E. Kannatey-Asibu Jr., Hidden Markov model-based wear monitoring in turning, J. Manufacturing Sci. Eng. 124 (2002) 651–658.

[59] K. Josien, T.W. Liao, Simultaneous grouping of parts and machines with an integrated fuzzy clustering method, Fuzzy Sets Syst. 126 (1) (2002) 1–21.

[60] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.

[61] G.S. Ng, H. Singh, Democracy in pattern classifications: combinations of votes from various pattern classifiers, Artif. Intell. Eng. 12 (1998) 189–204.

[62] R. Ng, J. Han, Efficient and effective clustering method for spatial data mining, Proceedings of the 1994 International Conference on Very Large Data Bases (VLDB'94), Santiago, Chile, September 1994, pp. 144–155.

[63] V.S. Ananthanarayana, M.N. Murty, D.K. Subramanian, Efficient clustering of large data sets, Pattern Recognition 34 (2001) 2561–2563.

[64] E. Keogh, S. Chu, D. Hart, M. Pazzani, Segmenting time series: a survey and novel approach, in: M. Last, A. Kandel, H. Bunke (Eds.), Data Mining in Time Series Databases, World Scientific, Singapore, 2004.

[65] J.F. Roddick, M. Spiliopoulou, A survey of temporal knowledge discovery paradigms and methods, IEEE Trans. Knowledge Data Eng. 14 (4) (2002) 750–767.

About the Author—T. WARREN LIAO received his Ph.D. in Industrial Engineering from Lehigh University in 1990 and is currently a Professor with the Industrial Engineering Department, Louisiana State University. His research interests include soft computing, pattern recognition, data mining, and their applications in manufacturing. He has more than 50 refereed journal publications and was the guest editor for several journals including *Journal of Intelligent Manufacturing*, *Computers and Industrial Engineering*, *Applied Soft Computing*, and *International Journal of Industrial Engineering.*