Stoch Environ Res Risk Assess (2012) 26:139–155
DOI 10.1007/s00477-011-0542-0

ORIGINAL PAPER

Improved Gath–Geva clustering for fuzzy segmentation of hydrometeorological time series

Nini Wang · Xiaodong Liu · Jianchuan Yin

Published online: 26 November 2011

Abstract In this paper, an improved Gath–Geva clustering algorithm is proposed for automatic fuzzy segmentation of univariate and multivariate hydrometeorological time series. The algorithm considers time series segmentation problem as Gath–Geva clustering with the minimum message length criterion as segmentation order selection criterion. One characteristic of the improved Gath–Geva clustering algorithm is its unsupervised nature which can automatically determine the optimal segmentation order. Another characteristic is the application of the modified component-wise expectation maximization algorithm in Gath–Geva clustering which can avoid the drawbacks of the classical expectation maximization algorithm: the sensitivity to initialization and the need to avoid the boundary of the parameter space. The other characteristic is the improvement of numerical stability by integrating segmentation order selection into model parameter estimation procedure. The proposed algorithm has been experimentally tested on artificial and hydrometeorological time series. The obtained experimental results show the effectiveness of our proposed algorithm.

Keywords Gath–Geva (GG) clustering · Minimum message length (MML) criterion · Time series segmentation · Expectation maximization (EM) algorithm · Segmentation order

1 Introduction

Time series analysis is the art of the statistical processing (Warren Liao 2005). A related term, *time series segmentation*, refers to the process of segmenting a given time series to detect homogeneous segments with similar statistical characteristics (Abonyi et al. 2005).

Hydrometeorological time series (streamflow, precipitation, temperature records, etc.) contain potentially valuable information. For example, a sudden shift of river cross-section after a catastrophic flood may cause permanent change in the streamflow record of this given cross-section (Gedikli et al. 2010). Another example, a gradual shift of global temperature records during the past centuries can be used to validate the hypothesis that the "greenhouse effect" is ongoing (Gedikli et al. 2008; Aksoy et al. 2008a). It would be helpful if the sudden or gradual shifts of multivariate time series could be detected by statistical characteristics. These segmentation results can then be deeper studied to provide more insight into natural or manmade influence to the hydrology, meteorology and environmetrics. The information gained from the segmentation results can be very valuable for engineering practice such as design and construction of new dams and hydroelectric power plants.

N. Wang (✉) · X. Liu
Research Center of Information and Control,
Dalian University of Technology, Dalian 116024, China
e-mail: wangnini2008@gmail.com

N. Wang · X. Liu
Department of Mathematics, Dalian Maritime University,
Dalian 116026, China

J. Yin
Navigation College, Dalian Maritime University,
Dalian 116026, China

J. Yin
School of Naval Architecture, Ocean and Civil Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China

Research on the area of hydrometeorological time series segmentation has been growing concern in hydrometeorological literature. Motivated by the pioneering work of Hubert (2000), Kehagias (2004), Gedikli et al. (2008) and Aksoy et al. (2007) worked on the segmentation of hydrometeorological time series with a continuous effort (Kehagias et al. 2005; Kehagias and Fortin 2006; Kehagias et al. 2007; Gedikli et al. 2010; Aksoy et al. 2008a, b).

Kehagias (2004) used hidden Markov model (HMM) to divide temperature and river discharge time series into segments. In analyses of HMM's application in hydrological segmentation problem, Kehagias and Fortin (2006) presented the shifting means model (SMM) (a special type of HMM's) to lessen the computational complications needed for the HMM's to converge to a stationary state.

Kehagias et al. (2005) divided univariate hydrological and environmental time series into segments using dynamic programming (DP) solution. Gedikli et al. (2008) used "branch and bound" type algorithm (denoted as AUG) to segment long hydrometeorological and geophysical time series. This algorithm is similar to the approach of Hubert (2000) with difference of the upper bound of a branch (segment). Aksoy et al. (2008a) presented offline segmentation algorithms by using DP and AUG approaches. Gedikli et al. (2010) offline divided long hydrometeological time series into segments by modified DP (mDP) algorithm that combinies DP with the remaining cost concept of Aksoy et al. (2008a) qand Gedikli et al. (2008).

However, we have noted that the above literature tends to segment time series on particular time point. While the changing behavior of the hydrometeorological time series is usually vague. Fuzzy clustering algorithms have demonstrated a high degree of suitability to process this kind of problem (Chatzis and Varvarigou 2008; Kehagias and Fortin 2006). Abonyi et al. (2003); Abonyi et al. (2005) have specifically developed the modified Gath–Geva (GG) algorithm to divide time-varying multivariate data into segments by using fuzzy sets to represent the segments in time.

Automatically partitioning a time series into optimal number of homogeneous segments is an important data-mining problem. This problem has been researched extensively and has not yet been solved satisfactorily (Abonyi et al. 2003; Abonyi et al. 2005; Liu et al. 2008; Keogh and Kasetty 2003; Fuchs et al. 2009, 2010; Fisch et al. 2011; Kehagias et al. 2005; Seghouane and Amari 2007). Kehagias et al. (2005) presented a DP algorithm that uses Schwarz's Bayesian information criterion (BIC) as the segmentation order selection criterion (Seghouane and Amari 2007) to find the globally optimal segmentations for every value of segmentation order. Abonyi et al. (2005) proposed a modified GG clustering algorithm that utilizes recursive cluster-merging technique to evaluate the compatibility of the adjacent clusters and merges the clusters that are found to be compatible. One major disadvantage of previous approaches is that the segmentation order selection and model parameter estimation are performed independently, which inevitably increases the complexity of computation. It is therefore desirable to combine model parameter estimation with the segmentation order selection for fast implementations.

Figueiredo and Jain (2002) proposed an unsupervised algorithm for learning finite mixture models which seamlessly integrates model parameter estimation and model order selection. Minimum message length (MML) is directly implemented in the modified expectation maximization (EM) algorithm to automatically select the number of components. The resulting technique is very fast and can avoid well known drawbacks of EM for mixture fitting: (i) it requires careful initialization of the parameters; (ii) it may converge to the boundary of the parameter space. Another excellence of the method is that it can be applied to any type of parametric mixture model trained by EM algorithm.

In this paper, an improved Gath–Geva (GG) clustering algorithm is proposed to automatically segment univariate and multivariate hydrometeorological time series with optimal segmentation order. DP, mDP and AUG algorithms can segment univariate time series. The modified GG algorithm can segment multivariate time series. Our proposed algorithm can segment univariate and multivariate time series. The changes of hydrometeorological time series are usually vague and do not suddenly happen on any particular time point. Therefore, the current study is based on the method of Abonyi et al. (2003, 2005), which considers hydrometeorological time series segmentation as a fuzzy clustering problem and the fuzzy sets are used to represent the segments in time. The method of Figueiredo and Jain (2002) is used to select the optimum segmentation order of hydrometeorological time series. In our proposed algorithm, MML criterion (Figueiredo and Jain 2002) is directly implemented by the modified *component-wise* EM (CEM^2) algorithm (Celeux et al. 1999) in GG clustering for time series segmentation (Gath and Geva 1989; Abonyi et al. 2003, 2005). Due to the combination of model parameter estimation and segmentation order selection, the aforementioned drawbacks of the classical EM algorithm in GG clustering can be avoided and the speed as well as the stability can be improved.

To show the appropriateness and effectiveness of our proposed algorithm, univariate and multivariate time series are segmented by our proposed algorithm without a specified number of segments in advance. Experimental results show that the proposed method can be applied to extract useful information from hydrometeorological time series fastly.

The remainder of the paper is organized as follows: In Sect. 2, we give a review on GG clustering-based time series segmentation algorithm. In Sect. 3, we present the improved GG clustering algorithm , in which MML criterion is directly implemented by the modified CEM^2 algorithm in GG clustering. The performance of the proposed algorithm on both artificial and real world hydrometeorological time series are examined in Sect. 4. Concluding remarks are presented in Sect. 5.

2 GG clustering-based time series segmentation

2.1 GG clustering algorithm

The close relationships between fuzzy clustering techniques and probability models indicate that it is possible to adopt probability models to modify fuzzy approaches (Povinelli et al. 2004). The GG clustering algorithm (Gath and Geva 1989) can be categorized into the group of the objective function based clustering algorithms (Vernieuwe et al. 2006):

$$J_{GG} = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m\|\mathbf{x}_k - \boldsymbol{\eta}_i^x\|^2 = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m D^2(\mathbf{x}_k, \boldsymbol{\eta}_i^x), \quad (1)$$

where $\mathbf{x}_k$ is a q-dimensional column vector, $\boldsymbol{\eta}_i^x$ is the ith cluster prototypes, $D^2(\mathbf{x}_k, \boldsymbol{\eta}_i^x)$ represents the distance between the data point $\mathbf{x}_k$ and the ith cluster prototypes $\boldsymbol{\eta}_i^x$, $\mu_{i,k}$ represents the membership degree of data point $\mathbf{x}_k$ to the ith cluster ($i = 1, \ldots, c$), which is given by

$$\mu_{i,k} = \frac{1}{\sum_{j=1}^{c}(D(\mathbf{x}_k, \boldsymbol{\eta}_i^x)/D(\mathbf{x}_k, \boldsymbol{\eta}_j^x))^{2/(m-1)}} \in [0,1], \quad (2)$$

and $m \in (1, \infty)$ represents the weighting exponent that determines the fuzziness of the resulting clusters. Bezdek and Dunn (1975) pointed out that when $m = 1$ the resulting algorithm is equivalent to the "nonfuzzy". A common choice of the weighting exponent is $m = 2$ and this value will be used throughout this paper. The membership degree $\mu_{i,k}$ is subject to the following constraints

$$\mu_{i,k} \in [0,1], \forall i,k; 0 < \sum_{k=1}^{n}\mu_{i,k} < n, \forall i; \sum_{i=1}^{c}\mu_{i,k} = 1, \forall k. \quad (3)$$

In order to minimize the objective function (Eq. 1), Lagrange multipliers for the constraints (Eq. 3) was introduced so that the objective function of GG clustering can be written as

$$\overline{J}_{GG} = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m D^2(\mathbf{x}_k, \boldsymbol{\eta}_i^x) + \sum_{k=1}^{n}\lambda_k\left(\sum_{i=1}^{c}\mu_{i,k} - 1\right). \quad (4)$$

where λ_k is Lagrange multiplier. A small distance means a high probability and a large distance means a low probability of a vector belongs to a cluster. So the distance measurement $D^2(\mathbf{x}_k, \boldsymbol{\eta}_i^x)$ is chosen indirectly proportional to the posteriori probability $p(\mathbf{x}_k|\boldsymbol{\eta}_i^x)$ as follows

$$D^2(\mathbf{x}_k, \boldsymbol{\eta}_i^x) = \frac{1}{P_i p(\mathbf{x}_k|\boldsymbol{\eta}_i^x)}, \quad (5)$$

where $p(\mathbf{x}_k|\boldsymbol{\eta}_i^x)$ is the probability that $\mathbf{x}_k$ belongs to the ith cluster, which is represented by Gaussian function $G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x)$ as follows

$$p(\mathbf{x}_k|\boldsymbol{\eta}_i^x) = G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x) = \frac{1}{(2\pi)^{q/2}\sqrt{det\mathbf{F}_i^x}} \cdot \exp\left(-\frac{1}{2}(\mathbf{x}_k - \mathbf{v}_i^x)^T(\mathbf{F}_i^x)^{-1}(\mathbf{x}_k - \mathbf{v}_i^x)\right), \quad (6)$$

where $\mathbf{v}_i^x$ and $\mathbf{F}_i^x$ are centers and covariances of Gaussian function $G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x)$, and P_i is the coefficient designed for eliminating the sensitivity of the algorithm, which is given by

$$P_i = \frac{1}{n}\sum_{k=1}^{n}\mu_{i,k}. \quad (7)$$

Assuming that $\mathbf{x}_k$ is effectively modelled as a mixture of Gaussian distribution, the GG clustering(Gath and Geva 1989) is equivalent to the EM identification of the Gaussian mixture $\boldsymbol{\eta}_x = \{\boldsymbol{\eta}_1^x, \ldots, \boldsymbol{\eta}_c^x, P_1, \ldots, P_c\}$ to minimize the objective function (Eq. 4), i.e.

$$\widehat{\boldsymbol{\eta}}_x = \arg\min_{\boldsymbol{\eta}_x}\overline{J}_{GG}. \quad (8)$$

The pseudocode of GG clustering algorithm is shown in the Appendix A.

However, the EM algorithm for GG clustering suffers several major drawbacks. First, the number of clusters has to be a priori specified to hit the right balance between the accuracy and the complexity. Also, because EM is a local greedy method, the GG clustering algorithm becomes more sensitive to initialization and it may converge to the boundary of the parameter space with increasing complexity (Figueiredo and Jain 2002).

2.2 GG clustering-based time series segmentation

A time series $\mathcal{X} = \{\mathbf{x}_k | k = 1, \ldots, n\}$ is a finite set of n samples labelled by time-coordinate $\mathcal{T} = \{t_k | k = 1, \ldots, n\}$, where $\mathbf{x}_k = [x_{1,k}, x_{2,k}, \ldots, x_{q,k}]^T, 1 \le k \le n$. A crisp segmentation is to seek a segmentation $\{t_{n_i} | i = 1, \ldots, c\}$ of $\mathcal{T}$, which

satisfy $1 = t_{n_0} < t_{n_1} < \cdots < t_{n_c} = t_n$. The intervals $[t_{n_0}, t_{n_1}], [t_{n_1}+1, t_{n_2}], \ldots, [t_{n_{c-1}}+1, t_{n_c}]$ are called segments, the time points $t_{n_0}, t_{n_1}, \ldots, t_{n_c}$ are called segment boundaries and the number of segments c is called the segmentation order. In such cases, the segmentation problem can be formulated as an optimization problem. Based on the notation in Kehagias et al. (2005) and Abonyi et al. (2005), the crisp segmentation cost is defined as follows:

$$cost_{crisp}(\mathbf{t}) = \sum_{k=1}^{N}\sum_{i=1}^{c}\beta_i(t_k)D^2(\mathbf{x}_k, \mathbf{v}_i^x), \tag{9}$$

where $\beta_i(t_k) \in \{0, 1\}$ is the crisp membership of the kth data point in the ith segment, and

$$\beta_i(t_k) = \begin{cases} 1, & \text{if } t_{n_{i-1}} < t_k \le t_{n_i}; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Instead of defining crisp bounds of the segments, Abonyi et al. (2003, 2005) developed an algorithm for dividing time series into fuzzy segments, which considers time series segmentation as GG clustering with time-coordinate as an additional variable. Assuming that the data point, $\mathbf{z}_k = [t_k, \mathbf{x}_k^T]^T$, can be effectively modeled as a mixture of multivariate Gaussian distribution, so the objective function can be written as the sum of the weighted squared distances between $\mathbf{z}_k$ and cluster prototypes $\boldsymbol{\eta}_i$ (Abonyi et al. 2003, 2005), i.e.,

$$\begin{aligned} J_{GGTS} &= \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m \|\mathbf{z}_k - \boldsymbol{\eta}_i\|^2 \\ &= \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m D^2(\mathbf{z}_k, \boldsymbol{\eta}_i). \end{aligned} \tag{11}$$

Since t_k is independent from $\mathbf{x}_k$, the distance measurement of Eq. 11, $D^2(\mathbf{z}_k, \boldsymbol{\eta}_i)$, is defined as (Abonyi et al. 2003, 2005)

$$D^2(\mathbf{z}_k, \boldsymbol{\eta}_i) = \frac{1}{\alpha_i p(\mathbf{z}_k|\boldsymbol{\eta}_i)} = \frac{1}{\alpha_i p(t_k|\boldsymbol{\eta}_i^t) p(\mathbf{x}_k|\boldsymbol{\eta}_i^x)}. \tag{12}$$

In Eq. 12, probability density function $p(t_k|\boldsymbol{\eta}_i^t)$ is given by Gaussian function $G(t_k; v_i^t, \sigma_{i,t}^2)$ to represent the probability of t_k belonging to the ith cluster in time-coordinate (Abonyi et al. 2003, 2005),

$$p(t_k|\boldsymbol{\eta}_i^t) = G(t_k; v_i^t, \sigma_{i,t}^2) = \frac{1}{\sqrt{2\pi\sigma_{i,t}^2}} \exp\left(-\frac{1}{2}\frac{(t_k - v_i^t)^2}{\sigma_{i,t}^2}\right), \tag{13}$$

where v_i^t and $\sigma_{i,t}^2$ are separately centers and variances of Gaussian function $G(t_k; v_i^t, \sigma_{i,t}^2)$. Probability density function of Eq. 12, $p(\mathbf{x}_k|\boldsymbol{\eta}_i^x)$, is given by Gaussian function $G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x)$ to represent the probability of $\mathbf{x}_k$ belonging to the ith cluster in time series data (Abonyi et al. 2003, 2005),

$$\begin{aligned} p(\mathbf{x}_k|\boldsymbol{\eta}_i^x) &= G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x) \\ &= \frac{1}{(2\pi)^{\frac{r}{2}}\sqrt{\det \mathbf{F}_i^x}} \\ &\quad \cdot \exp\left(-\frac{1}{2}(\mathbf{x}_k - \mathbf{v}_i^x)^T (\mathbf{F}_i^x)^{-1}(\mathbf{x}_k - \mathbf{v}_i^x)\right), \end{aligned} \tag{14}$$

where $\mathbf{v}_i^x$ and $\mathbf{F}_i^x$ are separately centers and covariances of Gaussian function $G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x)$, and r is the rank of $\mathbf{F}_i^x$. In Eq.12, α_i is the mixing coefficient satisfying

$$\sum_{i=1}^{c}\alpha_i = 1, \quad \alpha_i \ge 0, i = 1, \ldots, c. \tag{15}$$

Adjoining the constraints Eq. 3 to J_{GGTS} by means of Lagrange multipliers, the objective function of GG clustering-based time series segmentation (Abonyi et al. 2003, 2005) can be written as

$$\overline{J}_{GGTS} = \sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m D^2(\mathbf{z}_k, \boldsymbol{\eta}_i) + \sum_{k=1}^{n}\lambda_k\left(\sum_{i=1}^{c}\mu_{i,k} - 1\right). \tag{16}$$

The optimization of the vector of the parameters $\boldsymbol{\eta} = \{\boldsymbol{\eta}_i^t, \boldsymbol{\eta}_i^x, \alpha_i | 1 \le i \le c\} = \{v_i^t, \sigma_{i,t}^2, \mathbf{v}_i^x, \mathbf{F}_i^x, \alpha_i | 1 \le i \le c\}$ is considered as the EM identification of mixtures of Gaussian functions (Eqs. 13, 14). The interested reader can find more details in Abonyi et al. (2003, 2005). Then, the fuzzy segmentation of a time series, $\beta_i(t_k)$, is defined as (Abonyi et al. 2003, 2005)

$$\beta_i(t_k) = \frac{A_i(t_k)}{\sum_{j=1}^{c} A_j(t_k)} \in [0, 1], \tag{17}$$

where $A_i(t_k)$ is Gaussian membership function, which is give by

$$A_i(t_k) = \exp\left(-\frac{1}{2}\frac{(t_k - v_i^t)^2}{\sigma_{i,t}^2}\right). \tag{18}$$

3 Improved GG clustering-based time series segmentation

In this section, an improvement of the GG clustering (Gath and Geva 1989) is performed for the automatic fuzzy segmentation of hydrometeorological time series.

3.1 MML criterion for GG clustering-based time series segmentation algorithm

Segmentation order selection of a time series is often facilitated by the use of the model selection criterion (Kehagias et al. 2005). The underlying idea of model

selection criteria is a tradeoff between data fitting and complexity.

Based on information theoretic arguments, MML criteria is a powerful model selection criteria that has not yet been applied to time series segmentation problems. The rationale of MML criterion is: the shorter the encoding message length for data is, the better the corresponding data generation model is. For detailed accounts on MML criterion (see Figueiredo and Jain 2002; Lanterman 2001; Nascimento et al. 2010; Hanlon and Forbes 2002).

MML criteria for the vector of parameters,

$$\boldsymbol{\eta} = \{\boldsymbol{\eta}_i, \alpha_i | i = 1, \ldots, c\} = \{\boldsymbol{\eta}_i^t, \boldsymbol{\eta}_i^x, \alpha_i | i = 1, \ldots, c\}, \tag{19}$$

has the following general form (Hanlon and Forbes 2002):

$$MessLen = L(\boldsymbol{\eta}) + \text{penalty term}, \tag{20}$$

where c is the number of clusters, $L(\boldsymbol{\eta})$ is negative log-likelihood of $\boldsymbol{\eta}$, and the MML penalty term has the following form:

$$\text{Penalty term} = -\log p(\boldsymbol{\eta}) + \frac{1}{2}\log(\det \mathbf{I}(\boldsymbol{\eta})) + \frac{d}{2}(1 + \log \kappa_d), \tag{21}$$

where $d = Nc + c$, N is the dimensionality of $\boldsymbol{\eta}_i$, $p(\boldsymbol{\eta})$ is the prior density over the parameters $\boldsymbol{\eta}$, $\det \mathbf{I}(\boldsymbol{\eta})$ is the determinant of the Fisher information matrix, and κ_d is the d dimensional quantizing lattice constant required for optimal encoding of the d dimensional $\boldsymbol{\eta}$. Values of quantizing lattice constant, κ_d, for the first eight dimensions can be found in Hanlon and Forbes (2002).

Figueiredo and Jain (2002) demonstrated the principles of MML criterion for mixtures. Under the assumption that the prior density over the different cluster's parameters $p(\boldsymbol{\eta}_i)$ is independent and also is independent from the mixing coefficients $p(\alpha_1 \alpha_2 \ldots \alpha_c)$, we have (Figueiredo and Jain 2002)

$$p(\boldsymbol{\eta}) = p(\alpha_1 \alpha_2 \ldots \alpha_c) \prod_{i=1}^{c} p(\boldsymbol{\eta}_i). \tag{22}$$

The standard non-informative Jeffrey's priors for $p(\boldsymbol{\eta}_i)$ and $p(\alpha_1 \alpha_2 \ldots \alpha_c)$ have the following form

$$p(\boldsymbol{\eta}_i) \propto \sqrt{|\mathbf{I}^{(1)}(\boldsymbol{\eta}_i)|}, \tag{23}$$

$$p(\alpha_1 \alpha_2 \ldots \alpha_c) \propto (\alpha_1 \alpha_2 \ldots \alpha_c)^{-1/2}, \tag{24}$$

where $\mathbf{I}^{(1)}(\boldsymbol{\eta}_i)$ is the Fisher matrix for a single observation known to have been produced by the ith cluster, $0 \leq \alpha_1, \alpha_2, \ldots, \alpha_c \leq 1$ and $\alpha_1 + \alpha_2 + \cdots + \alpha_c = 1$. Due to the above inferences, Eq. 21 becomes (Figueiredo and Jain 2002)

$$\text{Penalty term} = \frac{N}{2}\sum_{i=1}^{c}\log\frac{n\alpha_i}{12} + \frac{c}{2}\log\frac{n}{12} + \frac{c(N+1)}{2}. \tag{25}$$

The main derivation of Eq. 25 is shown in Figueiredo and Jain (2002).

The definition of distance measurement $D^2(\mathbf{z}_k, \boldsymbol{\eta}_i)$ used in our proposed methodology is heavily based on Athanasiadis et al. (2009) as follows

$$D^2(\mathbf{z}_k, \boldsymbol{\eta}_i) = -\log \alpha_i p(\mathbf{z}_k|\boldsymbol{\eta}_i) = -\log \alpha_i p(t_k|\boldsymbol{\eta}_i^t) p(\mathbf{x}_k|\boldsymbol{\eta}_i^x). \tag{26}$$

Substituting Eq. 26 into Eq. 11, the objective function of our proposed improved GG clustering-based time series segmentation algorithm is derived as follows

$$J_{IGGTS} = -\sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m \log \alpha_i p(t_k|\boldsymbol{\eta}_i^t) p(\mathbf{x}_k|\boldsymbol{\eta}_i^x). \tag{27}$$

To reflect the size of the message required to describe the data, $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\}$, by our proposed algorithm, the negative log-likelihood of Eq. 20 is then given by

$$\begin{aligned} L(\boldsymbol{\eta}) &= J_{IGGTS} \\ &= -\sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m \log \alpha_i p(t_k|\boldsymbol{\eta}_i^t) p(\mathbf{x}_k|\boldsymbol{\eta}_i^x). \end{aligned} \tag{28}$$

From the work of Figueiredo and Jain (2002) on the MML penalty term (Eq. 25) and the negative log-likelihood (Eq. 28), MML criterion for our proposed improved GG clustering-based time series segmentation algorithm takes the following form

$$\begin{aligned} MessLen_{IGGTS}(\boldsymbol{\eta}, \mathcal{Z}) = & \\ & -\sum_{k=1}^{n}\sum_{i=1}^{c}(\mu_{i,k})^m \log \alpha_i p(t_k|\boldsymbol{\eta}_i^t) p(\mathbf{x}_k|\boldsymbol{\eta}_i^x) \\ & + \frac{N}{2}\sum_{i=1}^{c}\log\frac{n\alpha_i}{12} + \frac{c}{2}\log\frac{n}{12} + \frac{c(N+1)}{2}. \end{aligned} \tag{29}$$

Figueiredo and Jain (2002) stated that if any of the α_is is zero, MML penalty term in Eq. 25 does not make sense. This difficulty vanishes by deleting those clusters whose probability is zero and only coding the parameters of those clusters whose probability is nonzero (Figueiredo and Jain 2002). Following Figueiredo and Jain (2002), MML criterion for our proposed improved GG clustering-based time series segmentation algorithm is derived from Eq. 29 as follows

$$MessLen_{IGGTS}(\boldsymbol{\eta}_{curr}, \mathcal{Z}) = -\sum_{k=1}^{n}\sum_{i=1}^{c_{nz}}(\mu_{i,k})^m \log \alpha_i p(t_k|\boldsymbol{\eta}_i^t)p(\mathbf{x}_k|\boldsymbol{\eta}_i^x) + \frac{N}{2}\sum_{i=1}^{c_{nz}}\log\frac{n\alpha_i}{12} + \frac{c_{nz}}{2}\log\frac{n}{12} + \frac{c_{nz}(N+1)}{2}, \quad (30)$$

where c_{nz} denotes the number of non-zero-probability clusters, $\boldsymbol{\eta}_{curr} = \{\boldsymbol{\eta}_i, \alpha_i | i = 1, \ldots, c_{nz}\} = \{\boldsymbol{\eta}_i^t, \boldsymbol{\eta}_i^x, \alpha_i | i = 1, \ldots, c_{nz}\}$ represents the vector of the current parameters.

Minimizing Eq. 30, the vector of the optimal parameters is obtained as follows

$$\boldsymbol{\eta}_{opt} = \arg\min_{\boldsymbol{\eta}_{curr}} MessLen_{IGGTS}(\boldsymbol{\eta}_{curr}, \mathcal{Z}). \quad (31)$$

By means of Lagrange multipliers to adjoin the constraints Eqs. 3 to 30, the objective function of our proposed improved GG clustering-based time series segmentation algorithm can be written as follows

$$\overline{J}_{IGGTS} = MessLen_{IGGTS}(\boldsymbol{\eta}_{curr}, \mathcal{Z}) + \sum_{k=1}^{n}\lambda_k\left(\sum_{i=1}^{c_{nz}}\mu_{i,k} - 1\right). \quad (32)$$

3.2 Algorithm and implementation

To couple segmentation order selection and model parameter estimation in a unified framework, a reformulation of CEM^2 algorithm (Figueiredo and Jain 2002) for GG clustering-based time series segmentation is employed. The algorithm starts with a large number of clusters all over the space, which makes the algorithm robust with respect to initialization. And all that has to be done is to select the necessary clusters in a top-down manner. Let

$$\boldsymbol{\eta}(l) = \{\boldsymbol{\eta}_i(l), \alpha_i(l) | i = 1, \ldots, c_{nz}\} = \{\boldsymbol{\eta}_i^t(l), \boldsymbol{\eta}_i^x(l), \alpha_i(l) | i = 1, \ldots, c_{nz}\} = \{v_i^t(l), \sigma_{i,t}^2(l), \mathbf{v}_i^x(l), \mathbf{F}_i^x(l), \alpha_i(l) | i = 1, \ldots, c_{nz}\}$$

be the vector of parameters at the current iteration. The *E-step* updates the fuzzy partition matrix $\mathbf{U}(l) = \left[\mu_{i,k}^{(l)}\right]_{c_{nz}\times n}$, where $\mu_{i,k}^{(l)}$ is the membership degree of the kth data point to the ith cluster:

$$\mu_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^{c_{nz}}\left(D(\mathbf{z}_k, \boldsymbol{\eta}_i(l))/D(\mathbf{z}_k, \boldsymbol{\eta}_j(l))\right)^{2/m-1}} = \frac{\left(\alpha_i(l)p(t_k|\boldsymbol{\eta}_i^t(l))p(\mathbf{x}_k|\boldsymbol{\eta}_i^x(l))\right)^{m-1}}{\sum_{j=1}^{c_{nz}}\left(\alpha_j(l)p(t_k|\boldsymbol{\eta}_j^t(l))p(\mathbf{x}_k|\boldsymbol{\eta}_j^x(l))\right)^{m-1}}. \quad (33)$$

Then, the *M-step* involves simultaneously updating the vector of the parameters $\boldsymbol{\eta}(l)$ and annihilating clusters with vanishing mixing coefficients to reduce the segmentation order. For segmentation order reduction, Dirichlet-type priors are enforced on the mixing coefficients (Figueiredo and Jain 2002; Fu et al. 2010):

$$p(\alpha_1, \ldots, \alpha_{c_{nz}}) \propto \left\{-\frac{N}{2}\sum_{i=1}^{c_{nz}}\log\alpha_i\right\}. \quad (34)$$

There are two main advantages to adopt Dirichlet-type prior (Fu et al. 2010). Firstly, its negative exponents in Eq. 34 promotes the competition among clusters and penalises complexity. Secondly, it is the conjugate prior of multinomial random variables.

Therefore, for the ith cluster, the mixing coefficient $\alpha_i(l+1)$ can be calculated by (Figueiredo and Jain 2002; Fu et al. 2010):

$$\alpha_i(l+1) = \frac{\max\left\{0, \sum_{k=1}^{n}\mu_{i,k}^{(l)} - \frac{N}{2}\right\}}{\sum_{j=1}^{c_{nz}}\max\left\{0, \sum_{k=1}^{n}\mu_{j,k}^{(l)} - \frac{N}{2}\right\}}. \quad (35)$$

Recalling the constraints in Eq. 15, the normalized mixing coefficient vector $\mathcal{W}(l+1)$ is defined as

$$\mathcal{W}(l+1) = \left\{\alpha_1(l+1), \ldots, \alpha_{c_{nz}}(l+1)\right\} = \frac{\left\{\alpha_1(l+1), \ldots, \alpha_{c_{nz}}(l+1)\right\}}{\sum_{j=1}^{c_{nz}}\alpha_j(l+1)}. \quad (36)$$

Then, the ith cluster corresponding to $\alpha_i(l+1) = 0$ is annihilated, else the parameters of the ith cluster $\boldsymbol{\eta}_i(l+1)$ is updated by

$$v_i^t(l+1) = \left(\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m\right)^{-1}\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m t_k,$$

$$\mathbf{v}_i^x(l+1) = \left(\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m\right)^{-1}\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m \mathbf{x}_k,$$

$$\sigma_{i,t}^2(l+1) = \frac{\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m\left(t_k - v_i^t(l+1)\right)\left(t_k - v_i^t(l+1)\right)^T}{\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m},$$

$$\mathbf{F}_i^x(l+1) = \frac{\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m\left(\mathbf{x}_k - \mathbf{v}_i^x(l+1)\right)\left(\mathbf{x}_k - \mathbf{v}_i^x(l+1)\right)^T}{\sum_{k=1}^{n}\left(\mu_{i,k}^{(l)}\right)^m}. \quad (37)$$

It is to be noted that if the initial number of clusters c_{nz} is too large, it may happen that no cluster has enough initial support ($\sum_{k=1}^{n}\mu_{i,k}^{(l)} < \frac{N}{2}$, for $i = 1, \ldots, c_{nz}$) (Figueiredo and Jain 2002). To sidestep this difficulty, CEM^2 with the *M-step* is adopted to sequentially update parameters $\boldsymbol{\eta}(l)$ at the lth iteration: update $\alpha_1(l+1)$ by Eq. 35 and $\boldsymbol{\eta}_1(l+1)$ by Eq. 37, recompute $\mathcal{W}(l+1)$ by Eq. 36, update $\alpha_2(l+1)$ and $\boldsymbol{\eta}_2(l+1)$, recompute $\mathcal{W}(l+1)$, and so on.

Each CEM^2 iteration decreases the objective function given by Eq. 32. After convergence, i.e., when the relative decrease between the fuzzy partition matrices of the previous

and the current iteration $\|\mathbf{U}(l) - \mathbf{U}(l-1)\|$ falls below a threshold ε, there is no guarantee that we have found a minimum of MML criterion given by Eq. 30. Following the method of Figueiredo and Jain (2002), we check if smaller values of MML criterion are achieved by annihilating the least probable cluster (with smallest α_i). Algorithm 1 contains a detailed pseudocode description of our proposed algorithm.

Input: Data set $\{\mathbf{z}_k | 1 \leqslant k \leqslant n\}$; Termination tolerance ε; Initial segmentation order c_{max}; Minimum segmentation order c_{min};Current segmentation order $c_{nz} = c_{max}$
Output: Optimal segmentation order c_{opt}; Optimal cluster parameters $\boldsymbol{\eta}_{opt}$

Initialization: Initialize parameters by Eqs. 38-44.
while $c_{nz} \geqslant c_{min}$ **do**
  repeat for $l = 1, 2, \ldots$
    for $i = 1$ **to** c_{nz} **do**
      E-step: calculate $\mathbf{U}(l-1) = \left[\mu_{i,k}^{(l-1)}\right]_{c_{nz} \times n}$ via Eq. 33.
      M-step: update $\alpha_i(l)$ via Eqs. 35, 36;
      if $\alpha_i(l) > 0$ **then**
        update $\boldsymbol{\eta}_i(l)$ via Eq. 37.
        update $\mathbf{U}(l) = \left[\mu_{i,k}^{(l)}\right]_{c_{nz} \times n}$.
      else
        reduce the segmentation order $c_{nz} = c_{nz} - 1$.
        delete $\boldsymbol{\eta}_i(l)$ and $\alpha_i(l)$ from $\boldsymbol{\eta}_{curr}(l)$.
        update $\mathbf{U}(l) = \left[\mu_{i,k}^{(l)}\right]_{c_{nz} \times n}$.
    compute MML criterion $MessLen_{IGGTS}(\boldsymbol{\eta}_{curr}, \mathcal{Z})$ via Eq. 30.
  until $\|U(l) - U(l-1)\| < \varepsilon$;
  if $MessLen_{IGGTS}(\boldsymbol{\eta}_{curr}, \mathcal{Z}) \leqslant MML_{min}$ **then**
    $MML_{min} = MessLen_{IGGTS}(\boldsymbol{\eta}_{curr}, \mathcal{Z})$;
    $\boldsymbol{\eta}_{opt} = \boldsymbol{\eta}_{curr}$;
  $i^* = \arg\min_i\{\alpha_i > 0\}$,
  delete $\boldsymbol{\eta}_{i^*}$ and α_{i^*} from $\boldsymbol{\eta}_{curr}$,
  $c_{nz} = c_{nz} - 1$.

Algorithm 1 Improved GG clustering-based time series segmentation algorithm

4 Experiments

In this section, the effectiveness of our developed algorithm will be illustrated using several artificial time series and real-world hydrometeorological time series. All the simulations for our proposed algorithm are carried out in MATLAB 7.4 environment running at 2.40 GHz (CPU) and 1.92 GB memory (RAM).

4.1 Parameter initialization

Unless otherwise stated, the initial segmentation order c_{max} is set to 20, the minimum segmentation order c_{min} is set to 2, and the termination tolerance ε is set to 0.0001.

The initial centers $\mathbf{v}_i^x(0)$ and covariances $\mathbf{F}_i^x(0)$ are primarily based on the approach of Figueiredo and Jain (2002) as follows: the initial centers $\mathbf{v}_i^x(0)$ are randomly chose from time series data, i.e.,

$$\mathbf{v}_i^x(0) \in \mathcal{X}, 1 \leq i \leq c_{max}; \tag{38}$$

the initial covariances are made proportional to the identity matrix, i.e.,

$$\mathbf{F}_{i,x}(0) = \sigma^2 \mathbf{I}_{c_{max} \times c_{max}}, \tag{39}$$

where σ is the fraction of the mean of the variances along each dimension of time series data $\mathcal{X}$,

$$\sigma^2 = \frac{1}{10} trace\left(\frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T\right), \tag{40}$$

where $\mathbf{m} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_k$ is the global data mean.

The initial centers $v_i^t(0)$ and variances $\sigma_{i,t}^2(0)$ are calculated as follows

$$v_i^t(0) = \frac{\sum_{k=1}^{n}\left(\mu_{i,k}^{(0)}\right)^m t_k}{\sum_{k=1}^{n}\left(\mu_{i,k}^{(0)}\right)^m}, \tag{41}$$

$$\sigma_{i,t}^2(0) = \frac{\sum_{k=1}^{n}\left(\mu_{i,k}^{(0)}\right)^m \left(t_k - v_i^t(0)\right)\left(t_k - v_i^t(0)\right)^T}{\sum_{k=1}^{n}\left(\mu_{i,k}^{(0)}\right)^m}, \tag{42}$$

where initial membership degrees is given by

$$\mu_{i,k}^{(0)} = \begin{cases} 1, & \text{if } (i-1) \times S < k \leq i \times S; \\ 0, & \text{otherwise}, \end{cases} \tag{43}$$

where $1 \leq i \leq c_{max}$, $S = int(n/c_{max})$. In Eq. 43, for $i = c_{max}$, we have $i \times S = n$.

The initial mixing coefficient α_i ($1 \leq i \leq c_{max}$) are set proportional to c_{max}, i.e.,

$$\alpha_i = 1/c_{max}, 1 \leq i \leq c_{max}. \tag{44}$$

4.2 Evaluation metric for time series segmentation

The advantage of using artificial data sets is that the correct segmentations of time series is known and hence an evaluation can be made on the segmentation accuracy and the usefulness of the proposed algorithm.

Beeferman's segmentation metric P_k, proposed by Beeferman et al. (1999), is becoming the standard measure for assessing segmentation accuracy of time series (Kehagias et al. 2005; Kehagias and Fortin 2006). Following the method of (Kehagias et al. 2005; Kehagias and Fortin 2006), in our paper segmentation accuracy is measured by Beeferman's segmentation metric as follow

$$P_k(\mathbf{s}, \mathbf{t}) = \frac{1}{n}\sum_{i=1}^{n-k-1} |\delta_{\mathbf{s}}(i, i+k+1) - \delta_{\mathbf{t}}(i, i+k+1)|, \tag{45}$$

where k is half the average segment length, $P_k(\mathbf{s}, \mathbf{t})$ is the error measure between a proposed segmentation $\mathbf{s} = (0, s_1, \ldots, s_{K-1}, t)$ and a "true" segmentation $\mathbf{t} = (0, t_1, \ldots, t_{L-1}, t)$; function $\delta_{\mathbf{s}}(i, j)$ is defined to be 1 when i and j are in the same segment under $\mathbf{s}$ and 0 otherwise; function $\delta_{\mathbf{s}}(i, j)$ is defined similarly. A small value of P_k indicates low segmentation error ($P_k = 0$ indicates a perfect segmentation) (Kehagias et al. 2005; For more details, please see Kehagias et al. 2005; Kehagias and Fortin 2006).

Since membership degree $\mu_{i,k}$ of our proposed fuzzy segmentation algorithm, defined by Eq. 33, just represents the degree of the kth data point to the ith segment of a time series, we cannot compute P_k values directly. So classical unsupervised fuzzy classifier strategy "winner takes all" is used to represent crisp segmentation results corresponding to fuzzy segmentation results obtained by our proposed algorithm. The corresponding crisp segmentation result is the class related to the output of membership degree $\mu_{i,k}$ that gets the highest degree:

$$i^* = \arg\max_{1 \le i \le c_{opt}} \mu_{i,k}, \quad 1 \le k \le n. \tag{46}$$

4.3 Artificial univariate time series experiments

First our proposed algorithm was applied on three artificial univariate data sets of lengths in the order of 100, 1000 and 10000. Among them, artificial time series of length 100 was previously used by Gedikli et al. (2010), which is available at http://web.itu.edu.tr/~gedikliab/Segmenter/ts1a.txt. Artificial time series of lengths 1000 and 10000, characterized by the parameters listed in Table 1, were generated by using the procedure of Gedikli et al. (2010).

In the experiments, we found that the segmentation results are better with the smaller threshold for short time series and similar for the longer time series. So threshold ε was chosen to be 0.0001 for short time series and 0.01 for the longer time series to shorten execution time. We separately conducted the segmentation 50 trials for these univariate time series. The initial fuzzy segmentation results (see Eq. 17) from one of 50 runs are plotted in Figs. 1a, 2a, 3a (along with the time series of length 100, 1000 and 10000, respectively). From the above figures, it can be seen that the initial fuzzy segments are uniformly distributed by our propose algorithm.

Table 1 The characteristics of the artificial time series: n (Length), c (number of segments), μ_i (average value in ith segment, σ (standard deviation of segment (same for all segments))

Time series	n	c	μ_i	σ
1	1000	5	(3,5,2,5,4)	1.5
2	10000	10	(2,4,1,2,4,6,4,2,5,3)	1.5

The evolution of MML criterion for segmentation order is shown in Figs. 4a–c, (for the time series of length 100, 1000 and 10000, respectively). From Figs. 4a–c, it can be seen that when the segmentation order is respectively 9, 20 and 20 for these time series of length 100, 1000 and 10000, the relative decrease between the fuzzy partition matrices of the previous and the current iteration $\|\mathbf{U}(l) - \mathbf{U}(l-1)\|$ (see Line 15 of Algorithm 1) falls below predefined threshold ε. When the segmentation order is respectively 4, 5 and 10 for the time series of length 100, 1000 and 10000, MML criterion has the lowest value. So the optimal

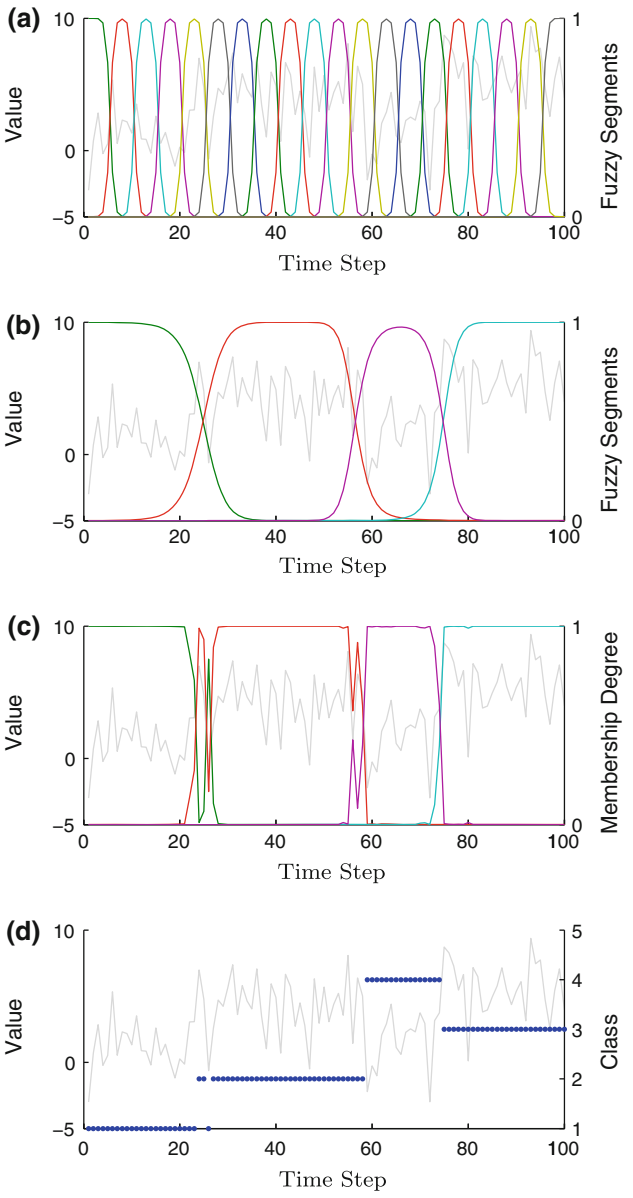

Fig. 1 Fuzzy segmentation results of the time series of length 100 with $c_{opt} = 4$: **a** initial fuzzy segments, **b** optimal fuzzy segments, **c** optimal membership degrees, and **d** corresponding optimal crisp segmentation results

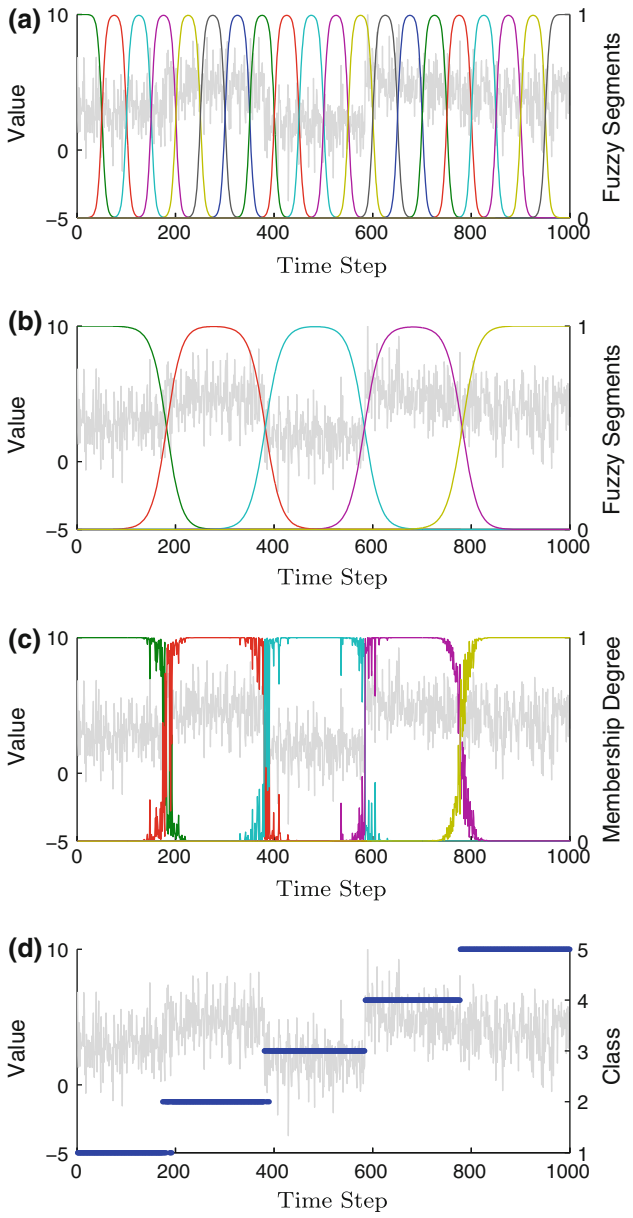

Fig. 2 Fuzzy segmentation results of the time series of length 1000 with $c_{opt} = 5$: **a** initial fuzzy segments, **b** optimal fuzzy segments, **c** optimal membership degrees, and **d** corresponding optimal crisp segmentation results

segmentation order is selected with 4,5,10 for the time series of length 100, 1000 and 10000, respectively.

Figure 5a–c separately show the histograms of the three univariate time series data together with the optimal mixture densities $\sum_{i=1}^{c_{nz}} \alpha_i p(x_k|\boldsymbol{\eta}_i^x)$ obtained by our algorithm, where each optimal mixture component $p(x_k|\boldsymbol{\eta}_i^x)$ is a Gaussian function $G(x_k; v_i^x, F_i^x)$ denoted by Eq. 14 and the bin height of the histograms represents the fraction of the frequency distribution within each bin denoted by $Q/n/\Delta$, where Q is frequency counts, n is total data number and Δ is bin width given by

$$\Delta = \frac{\max x_k - \min x_k}{nbins},$$

where *nbins* represents the bin number. In this paper, the bin number is chosen to be 40.

The optimal fuzzy segmentation results from one of 50 runs are plotted in Figs. 1b, 2b, and 3b (along with the time series of length 100, 1000 and 10000, respectively). From Figs. 1b, 2b, and 3b, it can be seen that our proposed algorithm is able to detect meaningful temporal shapes from short or long univariate time series. Then, the corresponding optimal membership degrees (see Eq. 33) along with these time series of length 100, 1000 and 10000 are plotted in Figs. 1c, 2c, 3c. By using Eq. 46, the corresponding crisp segmentation results are obtained, which are plotted in Figs. 1d, 2d and 3d. So we can measure segmentation accuracy by Beefermans segmentation metric P_k. The average mean and standard deviation (std) of the P_k values over the 50 repetitions are shown in Table 2 (for the time series of length 100, 1000 and 10000, respectively).

It can been seen from Table 2 that the average std of the P_k values is 0 for all the artificial univariate time series of lengths 100, 1000 and 10000. These results indicate that initial parameters of our algorithm have no influence on segmentation accuracy for the three artificial univariate time series. The P_k values for the three artificial time series listed in Table 2 are very close to 0, which indicates the high segmentation accuracy of our proposed algorithm achieves. The average execution time over the 50 trials are also shown in Table 2 (for the time series of length 100, 1000 and 10000, respectively).

We compare our proposed algorithm with existing time series segmentation methods. Executable versions of DP, mDP and AUG algorithms (in combination with the BIC test) are provided by Gedikli et al. at http://web.itu.edu.tr/~gedikliab/Segmenter/. Since this software doesn't allow the time series of length 10000 to conduct, one trial has been conducted for the DP, mDP and AUG algorithms (in combination with the BIC test) on the time series of length 100 and 1000 on a Windows PC running at 2.40 GHz (CPU) and 1.92 GB memory (RAM). The segmentation results and the execution time obtained by these algorithm are also listed in Table 2.

Based on the experimental results in Table 2, it can be seen that all the algorithms (DP, mDP, AUG and our proposed algorithm) achieve the correct segmentation order for the time series of length 100 and 1000. DP, mDP and AUG algorithms with BIC test give the same segmentation results. The value of P_k in Table 2 shows all the algorithms achieve the same value of $P_k = 0.1$ for the short time series of length 100, which indicates that all the algorithms perform the same segmentation accuracy for short time series of length 100. For the time series of length

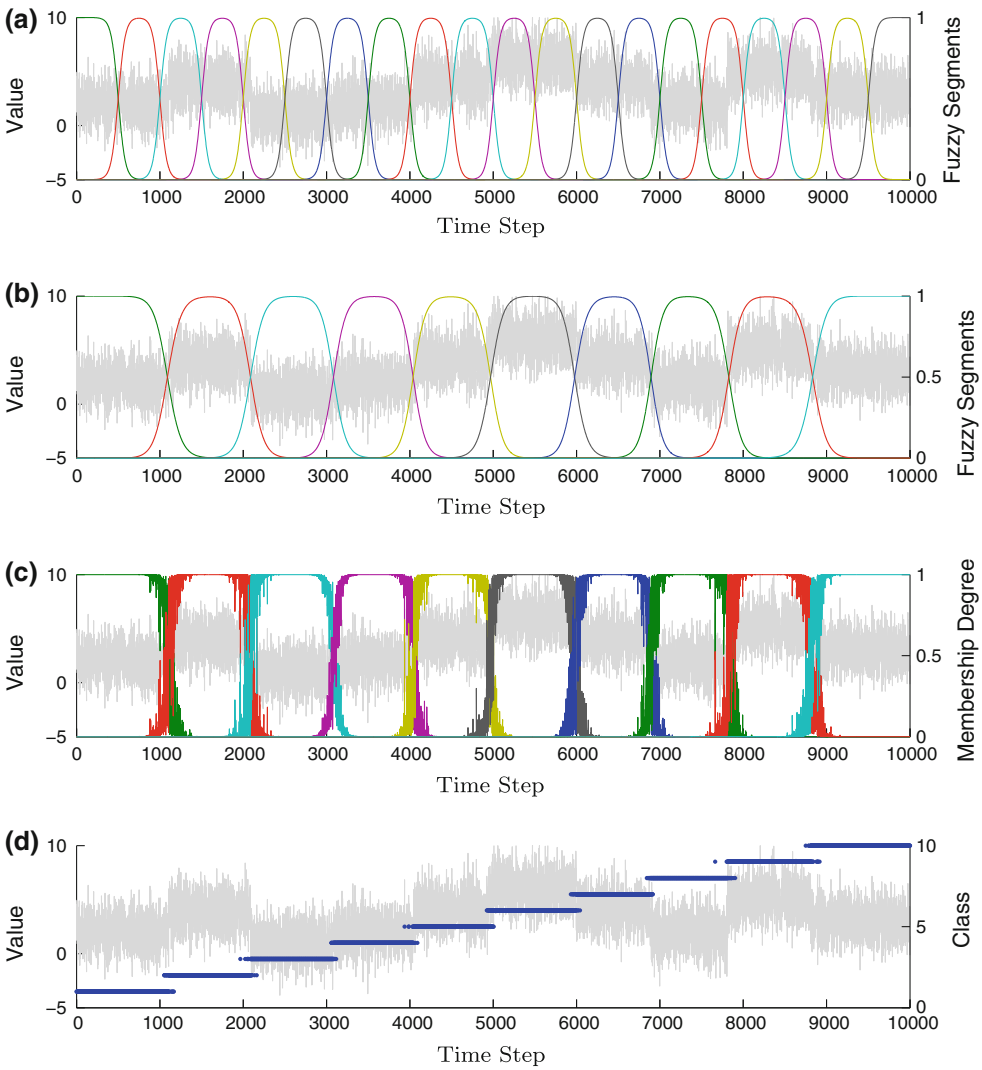

Fig. 3 Fuzzy segmentation results of the time series of length 10000 with $c_{opt} = 10$: **a** initial fuzzy segments, **b** optimal fuzzy segments, **c** optimal membership degrees, and **d** corresponding optimal crisp segmentation results

1000, the average P_k value achieved by our proposed algorithm is $P_k = 0.018$. While DP, mDP and AUG algorithms with BIC test can reach $P_k = 0$, which is lower than the results obtained by our proposed algorithm. These results indicate that DP, mDP and AUG algorithms with BIC test achieve better segmentation accuracy than our proposed algorithm for the time series of length 1000.

It should be noted that in order to compare our proposed fuzzy segmentation algorithm with the existing crisp segmentation methods such as DP, mDP and AUG algorithms, all the segment boundaries of the artificial univariate time series in Sect. 4.3 correspond to abrupt changes and evaluation metric P_k (see Eq. 45) is a non-fuzzy performance criterion. After fuzzy segmentation results obtained by our proposed algorithm were converted into the corresponding crisp ones by Eq. 46, the advantages of our proposed algorithm could not be shown exactly. So seen from the P_k value in Table 2, our proposed algorithm can therefore be considered as an alternative to the existing crisp segmentation algorithms for hydrometeorological time series.

Based on the experimental results in Table 2, we compare the execution time of the four algorithms for the time series of length 100 and 1000. Seen from the execution time of Table 2, DP, mDP and AUG algorithms with BIC test run around 4 times faster than our proposed algorithm for the short time series of length 100. For the time series of length 1000, execution time is 4,434 s for DP, 2.156 s for mDP, 75.750 for AUG and 95.969 for our proposed algorithm. The simulations for DP, mDP and AUG algorithms are carried out using compiled C++ coded segmentation package, while the simulations for our proposed algorithm is carried out in MATLAB 7.4 environment. Considering these algorithms executed in different executable environment, we can't further compare the execution time.

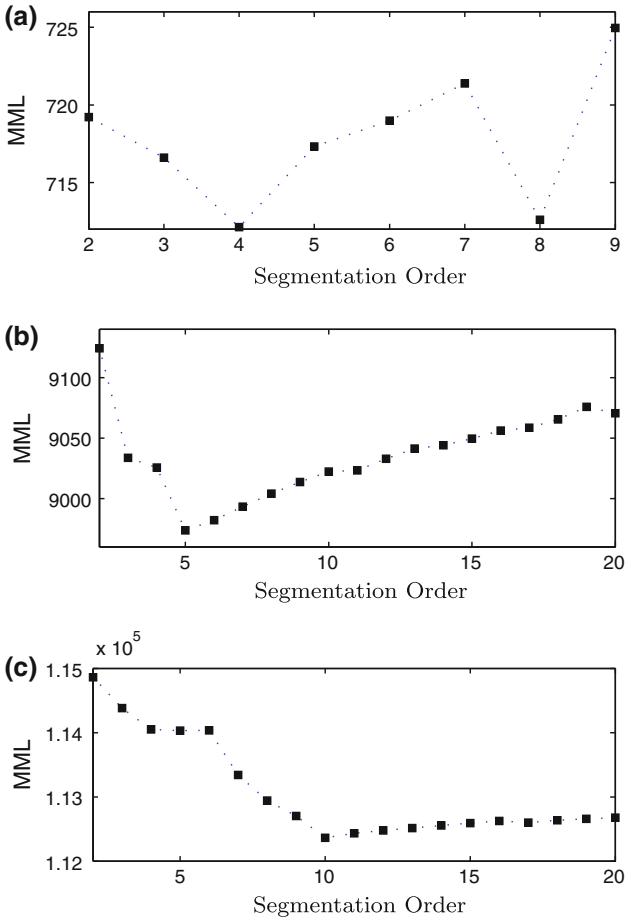

Fig. 4 Evolution of MML criterion of **a** the time series of length 100 with $c_{opt} = 4$, **b** the time series of length 1000 with $c_{opt} = 5$, and **c** the time series of length 10000 with $c_{opt} = 10$

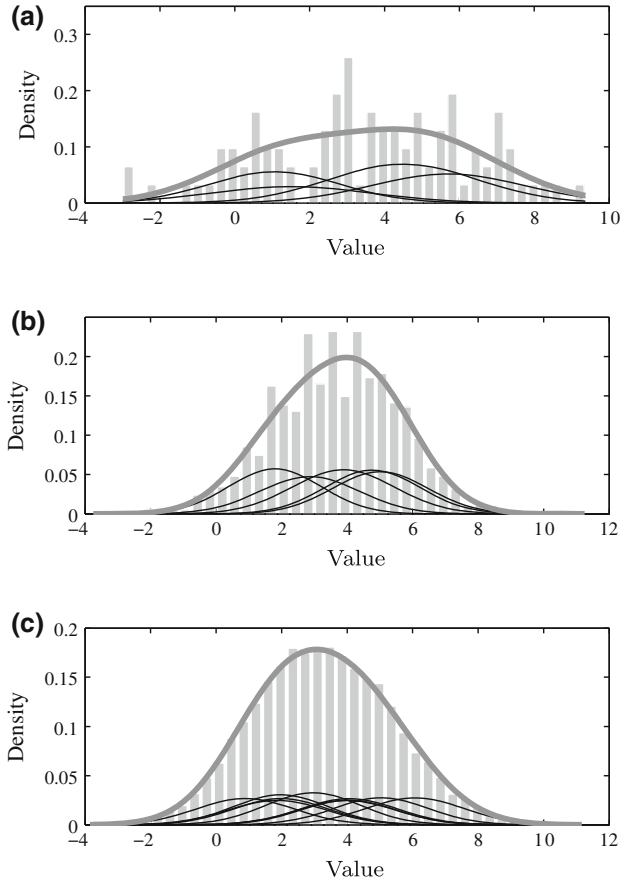

Fig. 5 Density estimates with 40 equally spaced binned time series data for **a** length 100 with $c_{opt} = 4$, **b** length 1000 with $c_{opt} = 5$, and **c** length 10000 with $c_{opt} = 10$. The bold lines signal optimal mixture densities; the solid lines indicate optimal mixture components; the bars represent the fraction of the frequency distribution within each bin

4.4 Artificial multivariate time series experiments

Next our proposed algorithm was applied on artificial multivariate data sets. Since artificial multivariate time series are very rare and expensive to construct, three-dimensional artificial time series were generated by the latent variables plotted in Fig. 6 according to the method of Abonyi et al. (2005). The MATLAB code for generating multivariate time series is available through internet at http://www.fmt.vein.hu/softcomp/segment/datagen.html. The correlation among the latent variables changes at the quarter of the time period, and the mean of the latent variables changes at the half of the time period.

As described in Abonyi et al. (2005), the goal is to illustrate how our proposed algorithm detects the changes of the latent process behind the multivariate time series. However, the correct segmentations of this multivariate time series are not available. Hence, segmentation results for the multivariate data sets were compared with the results of the modified GG algorithm for time series segmentation algorithm and the bottom-up segmentation

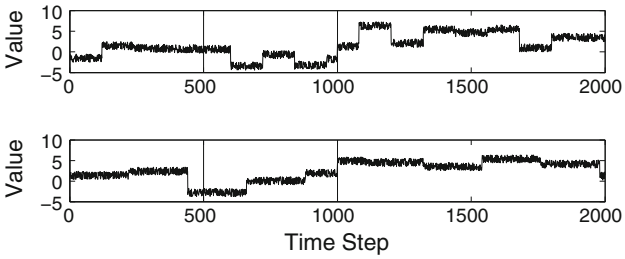

Fig. 6 Latent variables

method based on the Hotelling T^2 (top) and the reconstruction error Q (bottom). The pseudocode of the bottom-up segmentation method based on the Hotelling T^2 (top) and the reconstruction error Q (bottom) is shown in the Appendix B. Matlab codes of the modified GG and bottom-up algorithms are available at http://www.fmt.vein.hu/softcomp/segment. In this paper, the parameters of the modified GG algorithm are defined as follows: the initial number of the segments was 15, the principal components

Table 2 Performance comparison of the DP, mDP, AUG algorithms and our proposed algorithms (IGGTS) for the artificial time series (TS) of length 100, 1000 and 10000

Experiment	Method	Segmentation order	Change points	P_k		Execution time (s)
				Mean	Std	
Artificial TS of length 100	DP	4	0,21,58,73,100	0.1		0.281
	mDP	4	0,21,58,73,100	0.1		0.213
	AUG	4	0,21,58,73,100	0.1		0.235
	IGGTS	4		0.1	0	0.859
True segmentation		4	0,22,55,74,100			
Artificial TS of length 1000	DP	5	0,178,379,584,777,1000	0		4.434
	mDP	5	0,178,379,584,777,1000	0		2.156
	AUG	5	0,178,379,584,777,1000	0		75.750
	IGGTS	5		0.018	0	95.969
True segmentation		5	0,178,379,584,777,1000			
Artificial TS of length 10000	DP					
	mDP					
	AUG					
	IGGTS	10		0.0286	0	793.2969
True segmentation		10	0,1106,2086,3093,4043, 4943,5993,6883,7814, 8807,10000			

were set to $q = 2$, the fuzziness parameter was chosen to $m = 2$, the termination tolerance was chosen to $\varepsilon = 10^{-4}$, and the threshold γ for the compatibility matrix was chosen to be 0.75. For more details about the compatibility matrix, please see (Abonyi et al. 2005).

50 trials have been conducted on this artificial time series by our proposed algorithm with $c_{max} = 15$. Each time this three-dimensional time series was segmented into 14 segments. The average execution time over the 50 trials is 29.843 s. Segmentation results along with this three-dimensional time series from one of 50 runs are plotted in Fig. 7. The corresponding optimal Gaussian mixture components $G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x)(i = 1, \ldots, c_{opt})$ are shown in Fig. 8. Matlab code PlotGM.m for plotting Gaussian mixture components is available at http://www.lx.it.pt/~mtf/mixturecode.zip. From Fig. 7, it can be seen that our proposed algorithm can detect meaningful changes. But our proposed algorithm only found the changes in the mean of the data and it doesn't detect the changes in the correlation structure at the quarter of the time period (see Fig. 6). These results were compared with the results of the classical bottom-up method based on the Hotelling T^2 (top) and the reconstruction error Q (bottom) with 14 segments in Figs. 9a, b. It can be seen from Figs. 9a, b that the results are very different and these two algorithms can find the change in the correlation structure.

The modified GG algorithm was also applied on this three-dimensional artificial time series. The average execution time of 10 trial is 96.8750 s. According to the threshold $\gamma = 0.75$ for the compatibility matrix, the three-dimensional artificial time series was fuzzy segmented into 12 segments at each time and the segmentation results from one of 10 runs are shown in Fig. 10. These results were compared with our proposed algorithm with $c_{nz} = 12$. Fuzzy segmentation results by our proposed algorithm are shown in Fig. 12. As can be seen from Figs. 10 and 12, the results by the modified GG algorithm and our proposed algorithm are similar. These two algorithms can simultaneously detect the change of the latent process and the change of the mean of the variables (Fig. 11).

4.5 Univariate hydrometeorological time series experiments

Then our proposed algorithm was applied on annual temperature data of the northern hemisphere for the years 1000–1980 that was previously used by Gedikli et al. (2008). This data set is available at http://www.ncdc.noaa.gov/paleo/ei/ei_data/nhem-recon.dat.

The advantage of using real-world hydrometeorological time series is to show how our proposed algorithm detects meaningful changes from historical hydrometeorological data. Since the "true" segmentation is not available, P_k values cannot be computed. For comparison purposes, we list the optimal segmentation results by DP, mDP and AUG algorithms with BIC test to evaluate our algorithm performance on realistic hydrometeorological problems.

Our proposed algorithm was run 50 times and every time it identified 18 segments. The average execution time is 70.750 s. The original time series together with optimal fuzzy segments obtained by our proposed algorithm are plotted in Fig. 12a. From Fig. 12a, it can be seen that our

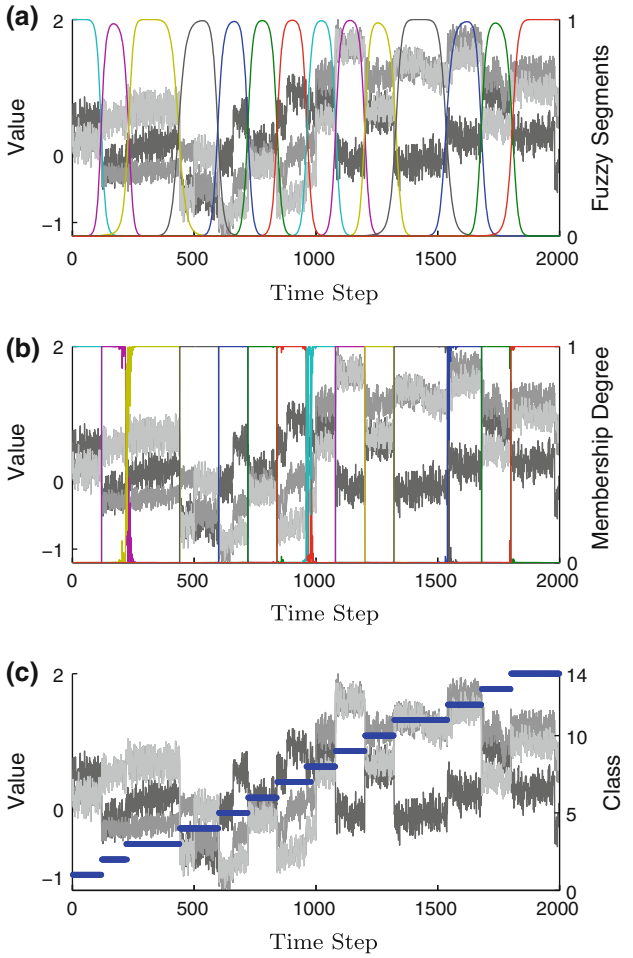

Fig. 7 Segmentation results of the three-dimensional artificial time series obtained by our proposed algorithm with $c_{opt} = 14$: (**a**) optimal fuzzy segments, (**b**) optimal membership degrees, (**c**) corresponding optimal crisp segmentation results

proposed algorithm can detect meaningful temporal shapes. The optimal membership degrees are plotted in Fig. 12b. The corresponding crisp segmentation results obtained by Eq. 46 are plotted in Fig. 12c. One trial on this time series has been conducted by DP, mDP and AUG algorithms with BIC test. The segmentation results and the execution time obtained by BIC are listed in Table 3. Execution time is 4.381 s for DP, 2.187 s for mDP and 58.474 s for AUG. DP, mDP and AUG algorithms with BIC test segmented this time series into 19 segments. These change points are marked by vertical dotted lines on Fig. 12c. It can be seen from Fig. 12c that our algorithm highly matches 13 change points with DP, mDP and AUG algorithms with BIC test. Since our proposed algorithm is a different approach to DP, mDP and AUG algorithms, our proposed segmentation algorithm can be considered as an alternative to the existing segmentation algorithms for hydrometeorological time series.

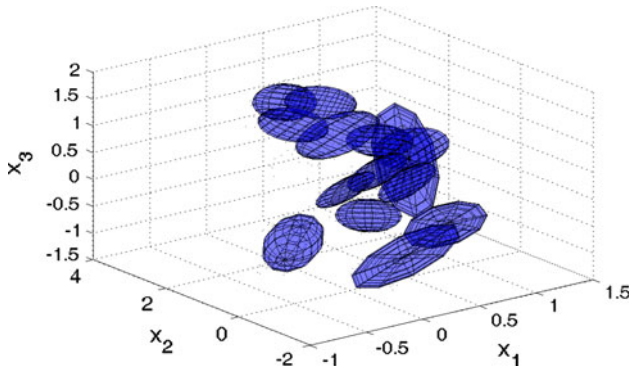

Fig. 8 The three-dimensional artificial time series together with optimal Gaussian mixture components $G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x)(i = 1, \ldots, c_{opt})$ obtained by our proposed algorithm with $c_{opt} = 14$. Three-dimensional shaded surfaces represent Gaussian mixture components $G(\mathbf{x}_k; \mathbf{v}_i^x, \mathbf{F}_i^x)(i = 1, \ldots, c_{opt})$.

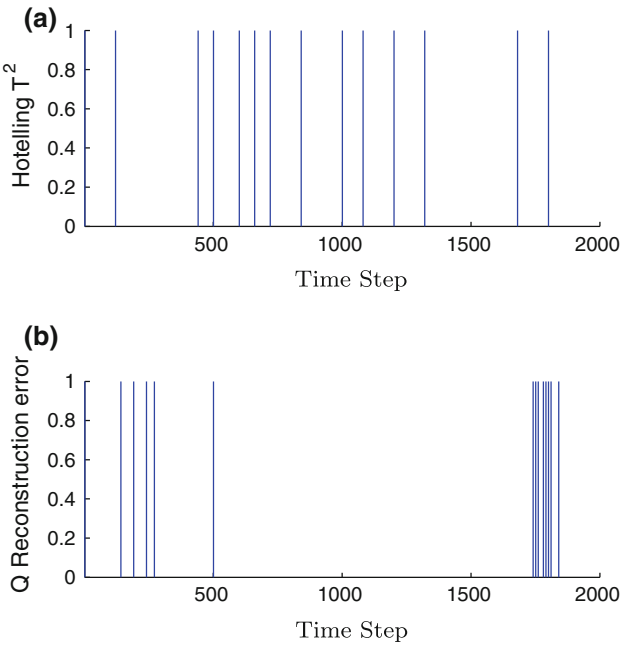

Fig. 9 Segmentation results of the three-dimensional artificial time series obtained by the bottom-up algorithm with 14 segments in case of two principal components based on: (**a**) the Hotelling T^2 (top), (**b**) the reconstruction error Q (bottom)

4.6 Multivariate hydrometeorological time series experiments

Finally our proposed algorithm was applied on multivariate hydrometeorological time series. Wind speed, wind gusts and wind direction courtesy of Arecibo, PR from 2011/03/01 to 2011/03/03 shown in Fig. 13 are designed to illustrate how our proposed algorithm segments multivariate hydrometeorological time series. The time series are available every six minutes at http://co-ops.nos.noaa.govfromNOAA/NOS/CO-OPS. Using $c_{max} = 15$, our

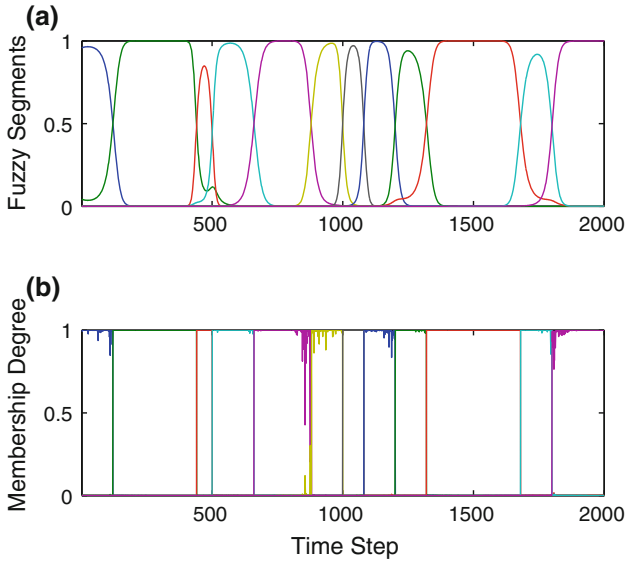

Fig. 10 Segmentation results of the three-dimensional artificial time series by the modified GG algorithm with 12 segments

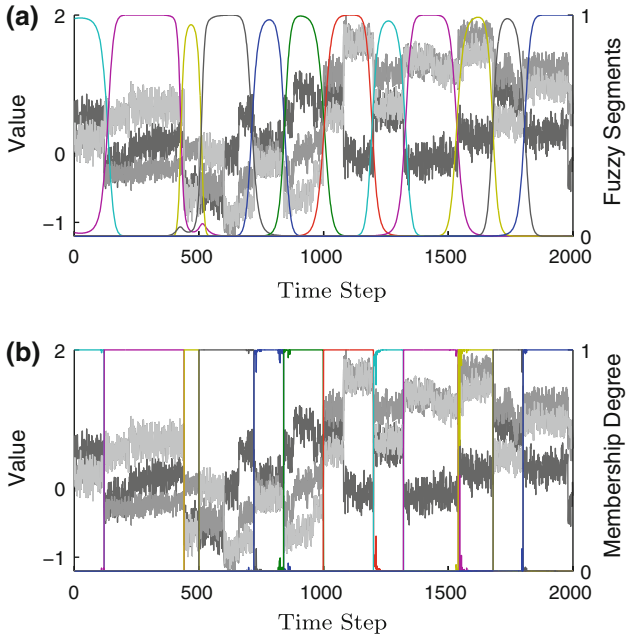

Fig. 11 Fuzzy segmentation results of the three-dimensional artificial time series obtained by our proposed algorithm with $c_{nz} = 12$: **a** fuzzy segments and **b** membership degrees

proposed algorithm was run 50 times, and every time this three-dimensional hydrometeorological time series was segmented into 13 segments. The average execution time is 11.703 s. The optimal fuzzy segments and membership degrees from one of 50 runs are separately shown in Figs. 14 and 15. These results were compared with the segmentation results of the modified GG algorithm. 10 trials have been conducted for the modified GG algorithm and every time this three-dimensional hydrometeorological time series was also segmented into 13 segments. The average execution time is 39.906 s and the optimal fuzzy

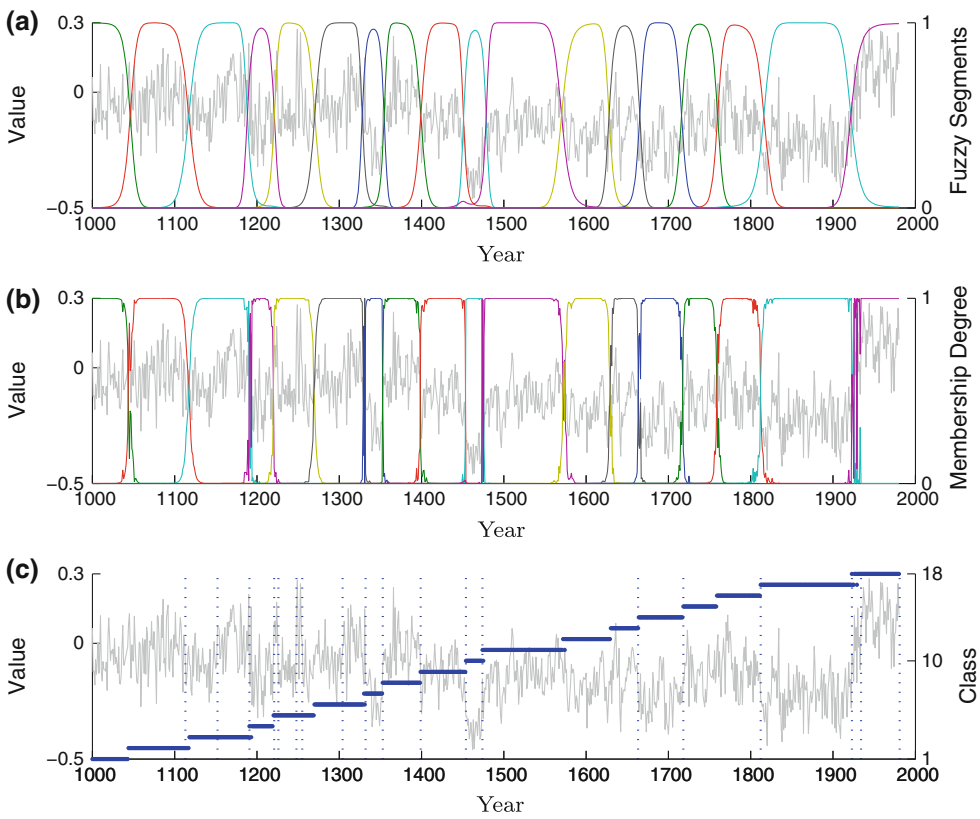

Fig. 12 Fuzzy segmentation results of annual temperature data of the northern hemisphere (1000–1980) obtained by our proposed algorithm with $c_{opt} = 18$: **a** optimal fuzzy segments, **b** optimal membership degrees, **c** corresponding optimal crisp segmentation results. The vertical dotted lines on **d** represent the change points obtained by DP, mDP and AUG algorithms

Table 3 Segmentation results and the execution time required for the DP, mDP, AUG and our proposed algorithm

Method	Order	Change points	Execution time (s)
DP	19	0,1000,1113,1152,1191,1221, 1226,1248,1255,1304,1332, 1353,1399,1454,1474,1663,1718, 1812, 1923,1934,1981	4.381
mDP	19	0,1000,1113,1152,1191,1221,1226, 1248,1255,1304,1332, 1353,1399,1454,1474,1663, 1718,1812, 1923,1934,1981	2.187
AUG	19	0,1000,1113,1152,1191,1221,1226, 1248,1255,1304,1332, 1353,1399,1454,1474,1663,1718, 1812, 1923,1934,1981	58.474
IGGTS	18		70.750

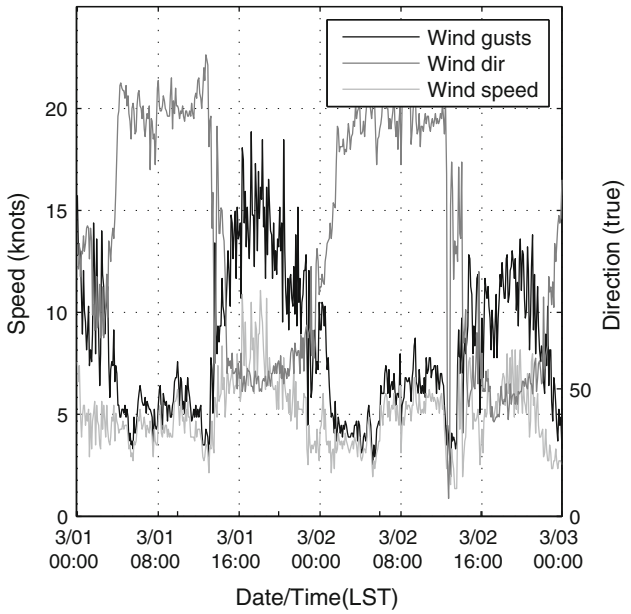

Fig. 13 Time series of wind gusts/dir/speed in the region of Arecibo, PR from 2011/03/01 to 2011/03/03

segmentation results from one of 10 runs are shown in Fig. 16. As it can be seen from Figs.14, 15 and 16, our proposed algorithm and the modified GG algorithm gave similar results. This example shows the applicability of our proposed fuzzy segmentation algorithm for multivariate hydrometeorological time series.

5 Conclusion

This paper proposed an improved Gath–Geva clustering-based segmentation algorithm for fuzzy segmentation of

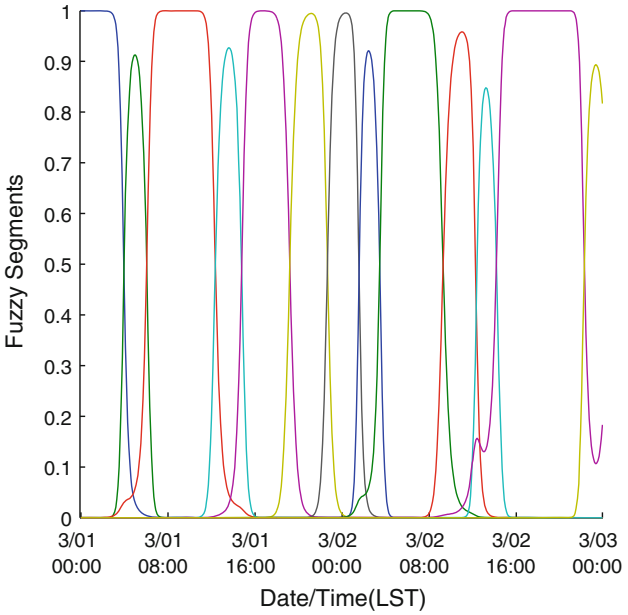

Fig. 14 Optimal fuzzy segments of the time series in Fig. 13 obtained by our proposed algorithm with $c_{opt} = 13$

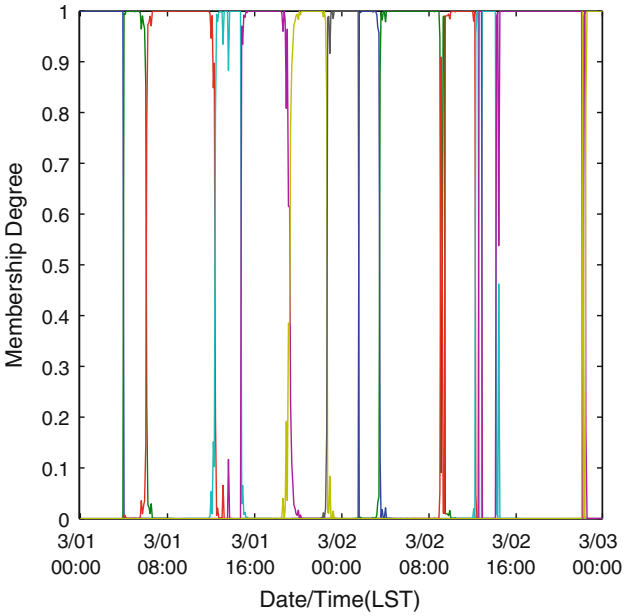

Fig. 15 Optimal membership degrees of the time series in Fig. 13 obtained by our proposed algorithm with $c_{opt} = 13$

univariate and multivariate hydrometeorological time series, which has several interesting and significant features:

1. Our proposed algorithm can segment a variety of time series (such as univariate vs. multivariate, short vs. long).
2. The changes of hydrometeorological time series are usually vague and do not suddenly happen on any

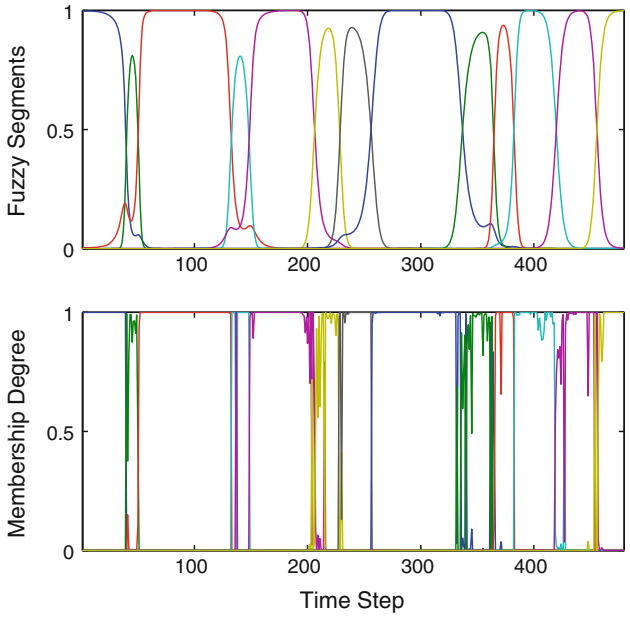

Fig. 16 Segmentation results of the time series in Fig. 13 obtained by the modified GG algorithm with 13 segments

particular time point. So our proposed algorithm is based on the identification of fuzzy sets that represent the fuzzy segments in time.

3. In our proposed algorithm, the MML criterion is directly implemented in the modified CEM^2 algorithm to automatically determine segmentation order. Thus, our proposed algorithm tends to have the better stability and performance.

Our proposed algorithm can be considered as an alternative to the existing hydrometeorological time series segmentation algorithms. Further, It should be noted that our proposed algorithm could be applied in a wider range of time series and not only to hydrometereological time series.

Acknowledgements The authors sincerely thank Professor Victor Leiva (Associate editor), Professor George Christakos (Editor), and the anonymous referees for their kind advice and comments. Their suggestions have led to a major improvement of the paper. This work is supported by the National Natural Science Foundation of China under Grants (No. 61175041) and the Fundamental Research Funds for the Central Universities (No. 2011QN147).

Appendix 1: Gath–Geva clustering algorithm

Inputs: data set $\mathcal{X} = \{\mathbf{x}_k | 1 \le k \le n\}$, number of clusters $1 < c < n$, weighting exponent $m > 1$, termination tolerance $\varepsilon > 0$, initial parameters. $\widehat{\boldsymbol{\theta}}(0) = \{\widehat{\boldsymbol{\theta}}_1, \ldots, \widehat{\boldsymbol{\theta}}_c, \widehat{P}_1, \ldots, \widehat{P}_c\}$

Output: optimal parameters $\widehat{\boldsymbol{\theta}}_{opt}$.

Initialize: partition matrix $\mathbf{U}(0) = [\mu_{i,k}^{(0)}]_{c \times n}$ such that Eq. 3 holds.

Repeat for $l = 1, 2, \ldots$

Calculate parameters $\widehat{\boldsymbol{\theta}}(l)$.

$$\begin{aligned}
\mathbf{v}_i(l) &= \frac{\sum_{k=1}^{n} \left(\mu_{i,k}^{(l-1)}\right)^m \mathbf{x}_k}{\sum_{k=1}^{n} \left(\mu_{i,k}^{(l-1)}\right)^m}, \\
\mathbf{F}_i(l) &= \frac{\sum_{k=1}^{n} \left(\mu_{i,k}^{(l-1)}\right)^m (\mathbf{x}_k - \mathbf{v}_i(l))(\mathbf{x}_k - \mathbf{v}_i(l))^T}{\sum_{k=1}^{n} \left(\mu_{i,k}^{(l-1)}\right)^m}, \\
P_i(l) &= \frac{1}{n} \sum_{k=1}^{n} \mu_{i,k}^{(l-1)}, \quad 1 \le i \le c.
\end{aligned} \tag{47}$$

Compute the distance measurement $D(\mathbf{x}_k,\mathbf{v}_i)^2$.

$$\begin{aligned}
D(\mathbf{x}_k, \mathbf{v}_i)^2 &= \frac{1}{P_i(l) G(\mathbf{x}_k; \mathbf{v}_i(l), \mathbf{F}_i(l))} = \frac{(2\pi)^{q/2} \sqrt{det(\mathbf{F}_i(l))}}{P_i(l)} \\
&\quad \cdot \exp\left(\frac{1}{2} (\mathbf{x}_k - \mathbf{v}_i(l))^T (\mathbf{F}_i(l))^{-1} (\mathbf{x}_k - \mathbf{v}_i(l))\right).
\end{aligned} \tag{48}$$

Update the partition matrix $\mathbf{U}(l) = \left[\mu_{i,k}^{(l)}\right]_{c \times n}$.

$$\begin{aligned}
\mu_{i,k}^{(l)} &= \frac{1}{\sum_{j=1}^{c} \left(D(\mathbf{x}_k, \mathbf{v}_i) / D(\mathbf{x}_k, \mathbf{v}_j)\right)^{2/(m-1)}}, \\
&\quad 1 \le i \le c, 1 \le k \le n.
\end{aligned} \tag{49}$$

until $\|\mathbf{U}(l) - \mathbf{U}(l-1)\| < \varepsilon$.

Appendix 2: Bottom-up segmentation method

Create initial fine approximation by segment boundaries $0 = t_{n_0} < t_{n_1} < \cdots < t_{n_c} = t_n$.

Find the cost of merging for each pair of segments: $mergecost(i) = cost(t_{n_i} + 1, t_{n_{i+2}})$

while $\min(mergecost) < maxerror$

Find the cheapest pair to merge:

$i = \arg\min_i(mergecost(i))$.

Merge the two segments, update the $(t_{n_i}, t_{n_{i+1}})$ **boundary indices, and recalculate the merge costs.**

$mergecost(i) = cost(t_{n_i} + 1, t_{n_{i+2}})$,

$mergecost(i-1) = cost(t_{n_{i-1}} + 1, t_{n_{i+1}})$.

end

Let covariance matrix $\mathbf{F}_i^x$ decompose to the matrix $\mathbf{\Lambda}_i$ that includes the eigenvalues of $\mathbf{F}_i^x$ in its diagonal in decreasing order, and to the matrix $\mathbf{U}_i$ that includes the eigenvectors corresponding to the eigenvalues in its columns, i.e., $\mathbf{F}_i^x = \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i^T$. The segmentation cost can be equal to the reconstruction error of this segment

$$cost(t_{n_i}+1, t_{n_{i+1}}) = \frac{1}{t_{n_{i+1}} - t_{n_i} + 1} \sum_{k=t_{n_i}+1}^{t_{n_{i+1}}} Q_{i,k},$$

where $Q_{i,k} = \mathbf{x}_k^T(\mathbf{I} - \mathbf{U}_{i,p}\mathbf{U}_{i,p}^T)\mathbf{x}_k$, and $\mathbf{U}_{i,p}$ is the eigenvectors corresponding to the first few p nonzero eigenvalues. The segmentation cost can also be equal to the Hotelling T^2 measure of this segment

$$cost(t_{n_i}+1, t_{n_i+1}) = \frac{1}{t_{n_{i+1}} - t_{n_i} + 1} \sum_{i=t_{n_i}+1}^{t_{n_{i+1}}} T_{i,k}^2,$$

where $T_{i,k}^2 = \mathbf{y}_{i,k}^T\mathbf{y}_{i,k}, \mathbf{y}_{i,k} = \boldsymbol{\Lambda}_{\boldsymbol{i,p}}^{-\frac{1}{2}}\mathbf{U}_{\boldsymbol{i,p}}^{\boldsymbol{T}}\mathbf{x}_{\boldsymbol{k}}$. The interested reader can find more details about the bottom-up method in Abonyi et al. (2005).

References

Abonyi J, Feil B, Nemeth S, Arva P (2003) Fuzzy clustering based segmentation of time-series. In: Lecture notes in computer science, pp 275–286

Abonyi J, Feil B, Nemeth S, Arva P (2005) Modified Gath–Geva clustering for fuzzy segmentation of multivariate time-series. Fuzzy Sets Syst 149:39–56

Aksoy H, Unal NE, Gedikli A (2007) Letter to the editor. Stoch Environ Res Risk Assess 21:447–449

Aksoy H, Gedikli A, Unal NE, Kehagias A (2008) Fast segmentation algorithms for long hydrometeorological time series. Hydrol Process 22:4600–4608

Aksoy H, Unal NE, Pektas AO (2008) Smoothed minima baseflow separation tool for perennial and intermittent streams. Hydrol Process 22:4467–4476

Athanasiadis EI, Cavouras DA, Spyridonos PP, Glotsos DT, Kalatzis IK, Nikiforidis GC (2009) Complementary DNA microarray image processing based on the fuzzy gaussian mixture model. IEEE Trans Inf Technol Biomed 13(4):419–425

Beeferman D, Berger A, Lafferty J (1999) Statistical models for text segmentation. Mach Learn 34:177–210

Bezdek JC, Dunn JC (1975) Optimal fuzzy partitions: a heuristic for estimating the parameters in a mixture of normal distributions. IEEE Trans Comput 835–838

Celeux G, Chretien S, Forbes F, Mkhadri A (1999) A component-wise EM algorithm for mixtures. Technical report 3746, INRIA, France

Chatzis S, Varvarigou T (2008) Robust fuzzy clustering using mixtures of student's-t distributions. Pattern Recognit Lett 29:1901–1905

Figueiredo M, Jain AK (2002) Unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell 24(3):381–396

Fisch D, Gruber T, Sick B (2011) Swiftrule: Mining comprehensible classification rules for time series analysis. IEEE Trans Knowl Data Eng 23(5):774–787

Fu Z, Robles-Kelly A, Zhou J (2010) Mixing linear SVMs for nonlinear classification. IEEE Trans Neural Netw 21:1963–1975

Fuchs E, Gruber T, Nitschke J, Sick B (2009) On-line motif detection in time series with swiftmotif. Pattern Recognit 42:3015–3031

Fuchs E, Gruber T, Nitschke J, Sick B (2010) Online segmentation of time series based on polynomial least-squares approximations. IEEE Trans Pattern Anal Mach Intell 32(12):2232–2245

Gath I, Geva AB (1989) Unsupervised optimal fuzzy clustering. IEEE Trans Pattern Anal Mach Intell 7:773–780

Gedikli A, Aksoy H, Unal NE (2008) Segmentation algorithm for long time series analysis. Stoch Environ Res Risk Assess 22(3): 291–302

Gedikli A, Aksoy H, Unal NE, Kehagias A (2010) Modified dynamic programming approach for offline segmentation of long hydrometeorological time series. Stoch Environ Res Risk Assess 24: 547–557

Hanlon B, Forbes C (2002) Model selection criteria for segmented time series from a bayesian approach to information compression. Working paper, Department of Econometrics and Statistics, Monash University, Melbourne, Australia

Hubert P (2000) The segmentation procedure as a tool for discrete modeling of hydrometeorological regimes. Stoch Environ Res Risk Assess 14:297–304

Kehagias A (2004) A hidden markov model segmentation procedure for hydrological and environmental time series. Stoch Environ Res Risk Assess 18:117–130

Kehagias A, Fortin V (2006) Time series segmentation with shifting means hidden markov models. Nonlinear Process Geophys 13: 339–352

Kehagias A, Nidelkou E, Petridis V (2005) A dynamic programming segmentation procedure for hydrological and environmental time series. Stoch Environ Res Risk Assess 20:77–94

Kehagias A, Petridis V, Nidelkou E (2007) Reply by the authors to the letter by Aksoy et al. Stoch Environ Res Risk Assess 21: 451–455

Keogh E, Kasetty S (2003) On the need for time series data mining benchmarks: a survey and empirical demonstration. Data Mining Knowl Discov 7(4):349–371

Lanterman AD (2001) Schwarz, Wallace, and Rissanen intertwining themes in theories of model order estimation. Int Stat Rev 69(2): 185–212

Liu X, Lin Z, Wang H (2008) Novel online methods for time series segmentation. IEEE Trans Knowl Data Eng 20:1616–1626

Nascimento JC, Figueiredo M, Marques JS (2010) Trajectory classification using switched dynamical hidden Markov models. IEEE Trans Image Process 19(5):1338–1348

Povinelli R, Johnson M, Lindgren A, Ye J (2004) Time series classification using Gaussian mixture models of reconstructed phase spaces. IEEE Trans Knowl Data Eng 16(6):779–783

Seghouane A, Amari S (2007) The AIC criterion and symmetrizing the Kullback-Leibler divergence. IEEE Trans Neural Netw pp 97–106

Vernieuwe H, De Baets B, Verhoest NEC (2006) Comparison of clustering algorithms in the identification of Takagi-Sugeno models: A hydrological case study. Fuzzy Sets Syst 157: 2876–2896

Warren Liao T (2005) Clustering of time series data-a survey. Pattern Recognit 38:1857–1874