Pattern Recognition 45 (2012) 1076–1091

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Classification trees for time series

Ahlame Douzal-Chouakria *, Cécile Amblard

LIG University of Joseph Fourier Grenoble 1, 38041 Grenoble cedex 9, France

ARTICLE INFO

Article history:
Received 17 June 2010
Received in revised form
20 July 2011
Accepted 12 August 2011
Available online 23 August 2011

Keywords:
Time series proximity measures
Supervised classification
Classification trees
Learning metric

ABSTRACT

This paper proposes an extension of classification trees to time series input variables. A new split criterion based on time series proximities is introduced. First, the criterion relies on an adaptive (i.e., parameterized) time series metric to cover both behaviors and values proximities. The metrics parameters may change from one internal node to another to achieve the best bisection of the set of time series. Second, the criterion involves the automatic extraction of the most discriminating subsequences. The proposed time series classification tree is applied to a wide range of datasets: public and new, real and synthetic, univariate and multivariate data. We show, through the experiments performed in this study, that the proposed tree outperforms temporal trees using standard time series distances and performs well compared to other competitive time series classifiers.

1. Introduction

Time series classification has been the subject of extensive research in the last several years. A first category of proposals consists of mapping time series to a new description space where conventional classifiers can be applied. Signal processing or statistical tools are commonly used to project time series into a given functional basis space. For instance, such projection can be performed by a Fourier or wavelet transform, a polynomial or an ARIMA approximation. Standard classifiers are subsequently applied on the fitted basis coefficients [1–5]. A second class of works proposes new heuristics, generally starting with the time series segmentation to extract prototypes that best characterize the time series classes. The prototypes, defined by such factors as a set of subsequences or regions of values, are subsequently described by a set of numerical features where standard classifiers can be applied [6–11]. A third category may be distinguished that consists of the hidden Markov models [12], which is frequently used for speech recognition and signal processing.

This paper focuses on a distance-based approach to extending classification trees to temporal data. We propose a new time series split criterion characterized by, on the one hand, the use of an adaptive metric to cover both behaviors and values proximities. This metric may change from one node to another according to the set of time series to be divided. On the other hand, the proposed split involves an automatic extraction of the most discriminating subsequences (i.e., segments of time series). We show, through the experiments performed, that the proposed tree outperforms temporal trees using standard time series distances and performs well compared to other competitive time series classifiers.

The rest of the paper is organized as follows. In the next section, we discuss two distance-based temporal trees proposed by Yamada et al. [13] and by Balakrishnan and Madigan [14]. In Section 3, the major metrics for time series are presented in a novel unified formalism. Section 4 presents the new time series classification tree, provides the main algorithms and discusses their complexity. In Section 5, the proposed classification tree is performed on six public and three new simulated datasets. The induced trees are compared to temporal trees using standard distances and are compared to other competitive time series classifiers.

2. Related works

In this section, we describe two temporal classification trees proposed by Yamada et al. in 2003 and by Balakrishnan and Madigan in 2006. Both works build binary classification trees in which internal nodes are labeled by one or two time series. Proposed classifiers are mainly based on new split tests to bisect the set of time series within internal nodes most effectively.

Yamada et al. [13] proposes two split tests. The first test, called the *standard-example* split test, uses an exhaustive search to select one existing time series (called the standard time series), leading to division with a maximum purity gain ratio. The first child node is composed of time series with a distance to the standard time series that is less than a given threshold, while the second child node contains the remaining time series. If more than one standard time series provides the largest value of the purity gain

* Corresponding author. Tel.: +33 4 56 52 00 68; fax: +33 4 56 52 00 22.
E-mail addresses: Ahlame.Douzal@imag.fr (A. Douzal-Chouakria), Cecile.Amblard@imag.fr (C. Amblard).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.018

ratio, a class isolation criterion is used to select the split that exhibits the most dissimilar child nodes.

The second proposed split test, which is called the *cluster-example* split test, performs an exhaustive search for two standard time series. The bisection is constructed by assigning each time series to the nearest standard time series. Similarly, the purity gain ratio and the class isolation criterion are used to select the best split test. For both split tests, the dynamic time warping is used as the time series proximity measure.

Balakrishnan and Madigan [14] look for a pair of reference time series that best bisects the set of time series according to a clustering-goodness criterion. For this purpose, a *k*-means algorithm is used. This algorithm ensures a partitioning that optimizes clustering criteria, namely, the compactness and isolation of the clusters but not their purity. To alleviate this problem, the authors repeat the *k*-means clustering several times and select the partition that gives the highest Gini index. The centers of the clusters define the pair of reference time series for the split test. For the time series proximities, both the Euclidean distance and the dynamic time warping are used to compare the efficiency of the obtained classification trees.

In summary, the *cluster-example* split test of Yamada et al. [13] and the one proposed by Balakrishnan and Madigan [14] are highly similar. The former first looks for a set of time series bisections with the highest purity clusters (i.e., the highest Gini index) and picks the one optimizing some clustering criteria (i.e., maximizing the separability of the clusters), whereas the latter first looks for a set of splits that optimize clustering criteria (i.e., *k*-means criteria) and accordingly selects the one exhibiting the highest purity clusters (i.e., maximizing the Gini index). When giving priority to a clustering criterion instead of the purity of the clusters, the split test may fail to select bisections of lower clustering criteria but of higher purity.

Let us make some remarks about the above proposed split tests. First, as for many distance-based approaches, the Euclidean distance and the dynamic time warping are considered for the time series proximities. These standard measures are values-based metrics and ignore the behaviors of the time series as discussed in Section 3. Second, the proposed splits use the same metric to divide all the nodes, but the peculiarities of the time series may change from one node to another. Finally, the time series distances are calculated using the whole time series values, even though the discrimination is determined by particular subsequences.

3. Time series metrics

We present, in a unified formalism, three categories of time series metrics. The first category relies on two standard values-based metrics: the dynamic time warping and the Euclidean distance. In the second category, we recall the definition of the correlation coefficient and the temporal correlation coefficient, which are used as behavior-based metrics. In the third category, we present a model to cover both behaviors and values components of time series. In particular, extensions of the Euclidean distance and of the dynamic time warping are provided to cover both behaviors and values proximities.

Let $S_1=(u_1,\ldots,u_p)$ and $S_2=(v_1,\ldots,v_q)$ be two time series of p and q values observed at the time instants $(t_1,\ldots,t_p)$ and $(t'_1,\ldots,t'_q)$, respectively. A mapping r between S_1 and S_2 is defined as a sequence of m pairs of observations $((u_{a_1},v_{b_1}),(u_{a_2},v_{b_2}),\ldots,(u_{a_m},v_{b_m}))$, with $a_i\in\{1,\ldots,p\}$, $b_i\in\{1,\ldots,q\}$, and $i\in\{1,\ldots,m-1\}$ obeying the order constraints:

$a_1=1,\quad a_m=p,\quad a_{i+1}=a_i$ or a_i+1 and,

$b_1=1,\quad b_m=q,\quad b_{i+1}=b_i$ or b_i+1.

with $m\in[max(p,q),p+q-1]$. Let R be a subset of such mappings, possibly satisfying some additional constraints, and let $c(r)$ $(r\in R)$ be the mapping cost function measuring the distance between the coupled values in r. A unified formalism of the time series proximity measures, denoted *dUnif*, may be presented as an optimization problem minimizing the cost function $c(r)$ on the search space R:

$$dUnif_{(c,R)}(S_1,S_2)=\min_{r\in R}c(r). \tag{1}$$

3.1. Values-based metrics

For the cost function definition $c(r)=\sum_{i=1}^{m}|u_{a_i}-v_{b_i}|$, $dUnif_{(c,R)}$ (Eq. (1)) leads to the standard dynamic time warping [15]:

$$d_{Dtw}(S_1,S_2)=\min_{r\in R}\left(\sum_{i=1}^{m}|u_{a_i}-v_{b_i}|\right) \tag{2}$$

In the case of times series of the same length ($m=p=q$), and for the cost function definition $c(r)=(\sum_{i=1}^{m}(u_i-v_i)^2)^{1/2}$ minimized on $R=\{r_0\}$, $dUnif_{(c,R)}$ gives the Euclidean distance, with:

$$r_0=((u_1,v_1),(u_2,v_2),\ldots,(u_m,v_m)) \tag{3}$$

$$d_E(S_1,S_2)=c(r_0)=\left(\sum_{i=1}^{m}(u_i-v_i)^2\right)^{1/2} \tag{4}$$

The above cost functions $c(r)$ involve the differences between the aligned values, without allowance for the values neighborhoods. This characteristic can be illustrated by the following example. Let $S_i=(0,1,-3,-2)$, $S_j=(4,8,5,8)$, and $S_k=(2,-2,-1,-3)$ be the three time series given in Fig. 1. Note that S_i and S_j are close in behaviors (i.e., they increase or decrease simultaneously) and far apart in values, whereas S_i and S_k are close in values and opposite in behaviors (i.e., S_k increases when S_i decreases and vice-versa). Both the Euclidean distance and the dynamic time warping give S_i closer to S_k than to S_j with $d_E(S_i,S_k)=4.24<d_E(S_i,S_j)=15.13<d_E\ (S_j,S_k)=16.15$, and $d_{Dtw}(S_i,S_k)=6<d_{Dtw}(S_i,S_j)=29\le d_{Dtw}(S_j,S_k)=29$.

3.2. Behavior-based metrics

Let us define two time series S_1 and S_2 to be similar in behavior if, during any observed period $[t_i,t_{i+1}]$, they increase or decrease simultaneously with the same growth rate. In contrast, they are considered to be opposite in behavior if, during any observed period $[t_i,t_{i+1}]$ in which S_1 increases, S_2 decreases and (vice-versa) with the same growth rate (in absolute value).

Until recently, many applications in different domains (e.g., speech recognition, system design control, functional MRI, microarrays and gene expression analysis) have used the Pearson correlation coefficient as a behavior proximity measure between signals [16–20]. Let us consider an equivalent formula for the

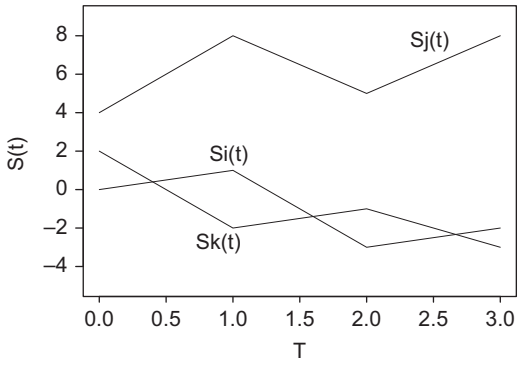

Fig. 1. Close on values and far on behavior vs. close on behavior and far on values.

correlation coefficient relying on pairwise values differences:

$$Cor(S_1,S_2)=\frac{\sum_{i,i'}(u_i-u_{i'})(v_i-v_{i'})}{\sqrt{\sum_{i,i'}(u_i-u_{i'})^2}\sqrt{\sum_{i,i'}(v_i-v_{i'})^2}} \quad (5)$$

It can be seen that the correlation coefficient assumes the independence of data as based on the differences between all of the pairs of values observed at $[t_i,t_{i'}]$; in contrast, the behavior proximity needs only to capture how time series behave at $[t_i,t_{i+1}]$. Thus, the correlation coefficient is biased by all of the remaining pairs of values observed at $[t_i,t_{i'}]$ with $|i-i'|>1$. For instance, for the time series in Fig. 1, the correlation coefficient fails by placing S_i closer to S_k than to S_j with $cor(S_j,S_k)=-0.89 < cor(S_j,S_i)=0.18 < cor(S_i,S_k)=0.25$.

To cope with temporal data, a variant of the Pearson correlation involving first-order differences is used:

$$Cort(S_1,S_2)=\frac{\sum_i(u_{i+1}-u_i)(v_{i+1}-v_i)}{\sqrt{\sum_i(u_{i+1}-u_i)^2}\sqrt{\sum_i(v_{i+1}-v_i)^2}} \quad (6)$$

with $Cort(S_1,S_2)$ belonging to $[-1,1]$. The value $Cort(S_1,S_2)=1$ indicates that S_1 and S_2 exhibit similar behavior. The value $Cort(S_1,S_2)=-1$ indicates that S_1 and S_2 exhibit opposite behavior. Finally, $Cort(S_1,S_2)=0$ expresses that the growth rates S_1 and S_2 are stochastically, linearly independent, thereby identifying time series of different behaviors, namely, that they are neither similar nor opposite. Similarly, let us consider the time series in Fig. 1 to show the success of the temporal correlation in determining that S_i is closer to S_j than to S_k with $cor(S_j,S_k)=-0.93 < cor(S_k,S_i)=-0.51 < cor(S_i,S_j)=0.77$. In the following discussion, we denote *Cor* and *Cort* as the total and temporal correlation coefficients, respectively.

Both the total (5) and temporal (6) correlation coefficients assume a mapping r_0 (Eq. (3)) between time series of the same length m. For a given mapping $r=((u_{a_1},v_{b_1}),(u_{a_2},v_{b_2}),\ldots,(u_{a_m},v_{b_m}))$ in R, these definitions may be generalized as follows:

$$Cor(S_1,S_2)=\frac{\sum_{i,i'}(u_{a_i}-u_{a_{i'}})(v_{b_i}-v_{b_{i'}})}{\sqrt{\sum_{i,i'}(u_{a_i}-u_{a_{i'}})^2}\sqrt{\sum_{i,i'}(v_{b_i}-v_{b_{i'}})^2}} \quad (7)$$

$$Cort(S_1,S_2)=\frac{\sum_i(u_{a_i}-u_{a_{i+1}})(v_{b_i}-v_{b_{i+1}})}{\sqrt{\sum_i(u_{a_i}-u_{a_{i+1}})^2}\sqrt{\sum_i(v_{b_i}-v_{b_{i+1}})^2}} \quad (8)$$

These coefficients $Cor(S_1,S_2)$ and $Cort(S_1,S_2)$ are simply denoted $Cor(r)$ and $Cort(r)$ in the following discussion.

3.3. Values and behavior based metrics

To define a proximity measure covering both the behaviors and values components, we consider the cost function $c_k(r)$ introduced in Douzal-Chouakria et al. [21], modulating the values-based proximity according to the behavior-based proximity:

$$c_k(r)=\frac{2}{1+\exp(k\,Co(r))}.c(r), \quad k\geq 0 \quad (9)$$

where $c(r)$ and $Co(r)$ define, respectively, values-based (Eqs. (2), (4)) and behavior-based (Eqs. (7), (8)) cost functions. Fig. 2 shows the modulating effect for several values of the parameter $k\geq 0$. The modulating function increases when the temporal correlation decreases from 0 to -1 (i.e., behaviors are increasingly opposite); thus, $c_k(r)$ becomes increasingly a values-based cost function. The modulating function decreases when the temporal correlation increases from 0 to 1 (i.e., the behaviors are increasingly similar), and $c_k(r)$ becomes increasingly a behavior-based cost function. Finally, this function leads to the proximity of values when the temporal correlation is zero (i.e., for different behaviors).

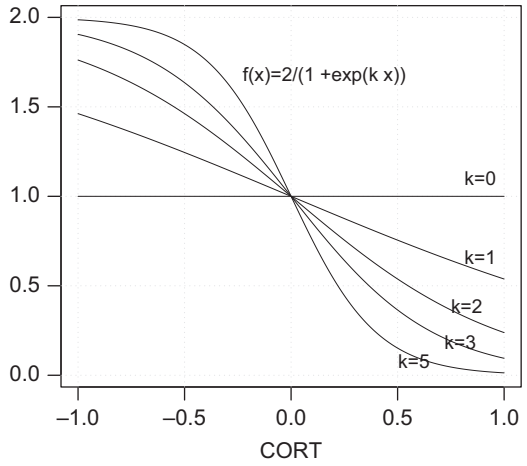

Fig. 2. The tuning effect of the parameter k.

An exponential form of $c_k(r)$ is preferred over, for instance, a simple linear combining function to provide a lower slope around the extreme values ($cort=-1,1$) and to ensure a nearly equal modulating effect for these extreme values and their nearest neighbors. The parameter k defines the relative contributions of the behavior and values components to $c_k(r)$. For a mapping cost function $c_k(r)$ covering only the values proximity component (i.e., ignoring the behavior component), k is fixed to 0 and $c_{k=0}(r)=c(r)$. On the other hand, for $k\geq 6$, $c_{k=6}(r)$ completely includes the behavior proximity component. Hence, if $Co(r)=1$, then $c_{k=6}(r)\approx 0$, which means that if two time series are of similar behavior, the cost function is reduced to zero regardless of the value of $c(r)$. If $Co(r)=-1$, then $c_{k=6}(r)\approx 2c(r)$; this statement corresponds, in the case of time series of opposite behaviors, to a penalty of a factor of 2 to $c(r)$. Finally, if $Co(r)=0$, then $c_{k=6}(r)\approx c(r)$, which means that in the case of time series of different behaviors, the mapping cost $c_{k=6}(r)$ is determined by the only available information $c(r)$.

Based on the cost function $c_k(r)$, the definition of the adaptive dissimilarity covering both values and behavior proximities is as follows:

$$D_k(S_1,S_2)=\min_{r\in R}\left(\frac{2}{1+\exp(k\,Co(r))}c(r)\right)$$

In particular, for $R=\{r_0\}$, $Co(r)=Cort(r)$, and $c(r)=(\sum_{i=1}^{m}(u_i-v_i)^2)^{\frac{1}{2}}$, D_k defines an extension of the Euclidean distance, denoted DE_k^{cort}, to cover both values and behavior proximities:

$$\mathrm{DE}_{\mathrm{k}}^{cort}(\mathrm{S}_1,\mathrm{S}_2)=\frac{2}{1+\exp(\mathrm{k}\,Cort(\mathrm{r}_0))}\left(\sum_{\mathrm{i}=1}^{\mathrm{m}}(\mathrm{u_i}-\mathrm{v_i})^2\right)^{1/2}$$

For $Co(r)=Cort(r)$ and $c(r)=\sum_{i=1}^{m}|u_{a_i}-v_{b_i}|$, D_k gives an extension of the dynamic time warping, denoted DTW_k^{cort}, to cover both values and behavior components:

$$\mathrm{DTW}_{\mathrm{k}}^{cort}(\mathrm{S}_1,\mathrm{S}_2)=\min_{\mathrm{r}\in\mathrm{R}}\left(\frac{2}{1+\exp(\mathrm{k}\,Cort(\mathrm{r}))}\sum_{\mathrm{i}=1}^{\mathrm{m}}|\mathrm{u_{a_i}}-\mathrm{v_{b_i}}|\right)$$

Finally, all the metrics presented above can be generalized to multivariate time series by summing over the n observed variables. For instance, the multidimensional form of d_{Dtw} measure is

$$d_{Dtw}(S_1,S_2)=\min_{r\in R}\left(\sum_{i=1}^{m}\sum_{j=1}^{n}|u_{ai}-v_{bi}|\right)$$

where $u_i=(u_{i1},\ldots,u_{in})$ and $v_i=(v_{i1},\ldots,v_{in})$ are the n-dimensional vectors observed at time t_i for the multivariate time series S_1, and

Table 1
A unified formalism for time series metrics.

Type	R	$c(r)$	$Co(r)$	Metric
Values	$R \subset M$	$\sum_{i=1}^{m} \lvert u_{a_i} - v_{b_i} \rvert$	–	$d_{Dtw} = \min_{r \in R} \left(\sum_{i=1}^{m} \lvert u_{a_i} - v_{b_i} \rvert \right)$
	$R = \{r_0\}$	$\left(\sum_{i=1}^{m} (u_i - v_i)^2 \right)^{1/2}$	–	$d_E = c(r_0) = \left(\sum_{i=1}^{m} (u_i - v_i)^2 \right)^{1/2}$
Behavior	$R = \{r_0\}$	–	$Cor(r)$	$d_{Cor} = 1 - Cor(r_0)$
	$R = \{r_0\}$	–	$Cort(r)$	$d_{Cort} = 1 - Cort(r_0)$
	$R \subset M$	–	$Cor(r)$	$dtw_{Cor} = \min_{r \in R}(1 - Cor(r))$
	$R \subset M$	–	$Cort(r)$	$dtw_{Cort} = \min_{r \in R}(1 - Cort(r))$
Val. &	$R = \{r_0\}$	$\left(\sum_{i=1}^{m} (u_i - v_i)^2 \right)^{1/2}$	$Cor(r)$	$DE_k^{Cor} = \frac{2}{1 + \exp(k\ Cor(r_0))} \left(\sum_{i=1}^{m} (u_i - v_i)^2 \right)^{1/2}$
	$R = \{r_0\}$	$\left(\sum_{i=1}^{m} (u_i - v_i)^2 \right)^{1/2}$	$Cort(r)$	$DE_k^{Cort} = \frac{2}{1 + \exp(k\ Cort(r_0))} \left(\sum_{i=1}^{m} (u_i - v_i)^2 \right)^{1/2}$
Beh.	$R \subset M$	$\sum_{i=1}^{m} \lvert u_{a_i} - v_{b_i} \rvert$	$Cor(r)$	$DTW_k^{Cor} = \min_{r \in R} \left(\frac{2}{1 + \exp(k\ Cor(r))} \sum_{i=1}^{m} \lvert u_{a_i} - v_{b_i} \rvert \right)$
	$R \subset M$	$\sum_{i=1}^{m} \lvert u_{a_i} - v_{b_i} \rvert$	$Cort(r)$	$DTW_k^{Cort} = \min_{r \in R} \left(\frac{2}{1 + \exp(k\ Cort(r))} \sum_{i=1}^{m} \lvert u_{a_i} - v_{b_i} \rvert \right)$

S_2, respectively. Table 1 summarizes in a unified formalism the proximity measures defined above.

4. Time series classification trees

In this section, we present a new split test for multivariate time series classification trees, which are characterized by two additive values. The first is the use of an adaptive metric that may change from one internal node to another to bisect the set of time series most effectively. The second is the involvement of the automatic extraction of the most discriminating subsequences. Given a time series $s = (u_1, \ldots, u_p)$, a subsequence $(u_i, \ldots, u_{i+q-1})$ $(1 \le i \le p-q+1)$ of s is a sampling of length $q < p$ of contiguous position from s.

Let $\{s_1, \ldots, s_N\}$ be a set of N multivariate time series partitioned into C classes, and let $I_1, \ldots, I_N$ $(I_i = [1, T_i])$ be their respective observation intervals. Before building the classification tree, time series are preprocessed to make them of equal length $I = [1, T]$, and the pairwise time series dissimilarities are computed.

4.1. Time series length normalization

To make the time series of the same length, two cases have to be considered. For data allowing time delays, time series are simply resampled by a linear interpolation to make them of equal length $I = [1, T]$. In the case of data that do not allow time delays, the smallest observation period $I = \min(I_1, \ldots, I_N)$ is considered; and linear interpolations may be used to resample the data within I.

4.2. Time series split (TSSplit) test algorithms

To split a given node S composed of a set of time series, the procedure *TSSplit* (S, I, α) is used with the following as input parameters: the set $S = \{s_1, \ldots, s_N\}$ of time series to bisect, the observation interval $I = [1, T]$, and a rate α needed for the discriminant subsequences search. In *TSSplit* (S, I, α) (Algorithm 1), a first call to *AdaptSplit*(S, I) is performed to determine the best split of S involving the adaptive metric D_k evaluated on I.

The main idea behind the procedure *AdaptSplit*(S, I) (Algorithm 2) is that, given a value of the parameter $k \in [0, 6]$ and two time series (l, r) from $S \times S$, a bisection of S, denoted $\sigma(l, r, k, I)$, is obtained by assigning each time series $ts \in S$ to the left node if it is closer to the time series l than to r, specifically, if $D_k(ts, l) \le D_k(ts, r)$, and to the right node otherwise. To determine the best split, several values of the triplet (l, r, k) are explored to find the bisection that exhibits the minimum impurity Gini index. As output, *AdaptSplit*(S, I) returns the best split $\sigma(l_*^I, r_*^I, k_*^I, I)$ and its impurity Gini index $GI(\sigma(l_*^I, r_*^I, k_*^I, I))$.

The best split $\sigma(l_*^I, r_*^I, k_*^I, I)$ is obtained by comparing the time series proximities according to their observations within I. In the case in which time series differentiation is induced by some subsequences instead of implicating all of the observations of I, the split $\sigma(l_*^I, r_*^I, k_*^I, I)$ may fail to reach higher purity classes. To alleviate this limitation, *DichoSplit* allows us to determine subsequences of I entailing a bisection of lower impurity than $\sigma(l_*^I, r_*^I, k_*^I, I)$ through a dichotomy search within the left and right subsequences of I.

The *DichoSplit* $(S, \sigma(l_*^I, r_*^I, k_*, I), e_I, \alpha)$ (Algorithm 3) is used with the following as input parameters: the set of time series S, the best split $\sigma(l_*^I, r_*^I, k_*^I, I)$ of S obtained by comparing time series over I, its corresponding impurity Gini index error named e_I, and the rate α needed to define the boundaries of the left I_L and right I_R sub-intervals of I. Two calls to *AdaptSplit* are performed to split S based on the observations of I_L and I_R, respectively.

If the impurity Gini index is not improved $(e_I \le \min(e_{I_L}, e_{I_R}))$, all of the observations within I are needed to best discriminate the time series. *DichoSplit* stops and returns the split $\sigma(l_*^I, r_*^I, k_*^I, I)$. However, if the impurity Gini index is improved by at least one of the splits based on I_L or I_R, a call to *DichoSplit* is pursued with the most discriminative sub-interval.

4.3. Algorithm specifications

In *AdaptSplit*, the explored splits $\sigma(l, r, k, I)$ (Algorithm 2, line 3) rely on the adaptive metric D_k, for seven values of $k \in [0, 6]$. In *DichoSplit*, a value of $\alpha = 0.6$ is taken to divide I into the left and right covering intervals, which makes it possible to cover discriminating subsequences in the central region of I. Of course, other values of α may be taken to divide I into either disjoint $(\alpha = 0.5)$ or more covering $(\alpha > 0.6)$ left and right sub-intervals.

After the two calls to *AdaptSplit* based on I_L and I_R (lines 4 and 5), some options are available when faced with the equality cases. First, in the case of $e_I = e_{I_L} = e_{I_R}$, *DichoSplit* (Algorithm 3, line 6) stops and returns the best split based on I. A more costly variant may continue by exploring the splits based on I_L and I_R and stop

only if $e_I < \min(e_{I_L}, e_{I_R})$. Second, if $e_{I_L} = e_{I_R} < e_I$ (line 8), the split continues with only the left sub-interval.

Once the two calls to *AdaptSplit* based on I_L and I_R achieved (lines 4 and 5), some options are taken when faced with the equality cases. First, in the case of $e_I = e_{I_L} = e_{I_R}$, *DichoSplit* (Algorithm 3, line 6) stops and returns the best split based on I. Whereas a more costly variant may continue by exploring the splits based on I_L and I_R and stops only if $e_I < \min(e_{I_L}, e_{I_R})$. Second, if $e_{I_L} = e_{I_R} < e_I$ (line 8); the split continues with only the left sub-interval.

The three algorithms *TSSplit*, *AdaptSplit*, and *DichoSplit* were implemented in C and integrated with the CART algorithm of the R package tree proposed by B. Ripley (http://cran.r-project.org/web/packages/tree/). For its default parameters, the time series decision tree is induced without pruning and with a minimum size of leaves of 2 time series.

4.4. Time series classification

Each node of the induced time series classification tree (*TSTree*) is characterized by a split test $\sigma(l_*, r_*, k_*, I_*)$ described by the two representative time series (l_*, r_*), the optimal value k_* of the learned metric D_{k_*} and the most discriminating subsequence I_*. A new time series *ts* is assigned to the left sub-node if it is closer to the left time series l_* than to r_* with $D_{k_*}(ts, l_*) \le D_{k_*}(ts, r_*)$; otherwise, it is assigned to the right sub-node. The time series proximities D_{k_*} are evaluated over the discriminant period I_*. As in conventional classification trees, *ts* is assigned to the class of the leaf in which it falls.

Algorithm 1. *TSSplit*(S, I, α).

1: $(\sigma(l^I_*, r^I_*, k^I_*, I), e_I) = AdaptSplit(S, I)$
2: $\sigma(l_*, r_*, k_*, I_*), e_{I_*}) = DichoSplit(S, \sigma(l^I_*, r^I_*, k^I_*, I), e_I, \alpha)$
3: **return** $(\sigma(l_*, r_*, k_*, I_*), e_{I_*})$

Algorithm 2. *AdaptSplit*(S,I).

1: $e_* = \infty$
2: **for** k in [0;6]
3: $(l_k, r_k) = \arg\min_{(l,r)}(GI(\sigma(l, r, k, I)))$
4: **if** $GI(\sigma(l_k, r_k, k, I)) < e_*$
5: $e_* = GI(\sigma(l_k, r_k, k, I))$
6: $l^I_* = l_k$, $r^I_* = r_k$, $k^I_* = k$
7: **end if**
8: **end for**
9: **return** $(\sigma(l^I_*, r^I_*, k^I_*, I), e_*)$

Algorithm 3. *DichoSplit*($S, \sigma(l^I_*, r^I_*, k^I_*, I), e_I, \alpha$).

1: $[a, b] = I$
2: $I_L = [a, a + \alpha(b-a)]$
3: $I_R = [b - \alpha(b-a), b]$
4: $(\sigma(l^{I_L}_*, r^{I_L}_*, k^{I_L}_*, I_L), e_{I_L}) = AdaptSplit(S, I_L)$
5: $(\sigma(l^{I_R}_*, r^{I_R}_*, k^{I_R}_*, I_R), e_{I_R}) = AdaptSplit(S, I_R)$
6: **if** $e_I \le \min(e_{I_L}, e_{I_R})$ **then**
7: **return** $(\sigma(l^I_*, r^I_*, k^I_*, I), e_I)$
8: **else if** $e_{I_L} \le e_{I_R}$ **then**
9: *DichoSplit* $(S, \sigma(l^{I_L}_*, r^{I_L}_*, k^{I_L}_*, I_L), e_{I_L}, \alpha)$
10: **else**
11: *DichoSplit*$(S, \sigma(l^{I_R}_*, r^{I_R}_*, k^{I_R}_*, I_R), e_{I_R}, \alpha)$
12: **end if**

4.5. TSSplit complexity

Let N be the number of time series, K the number of explored values of the parameter k, T the initial time series length, and α the cover rate for the dichotomous search. On the one hand, the core of the complexity of *AdaptSplit* is determined by the exhaustive search of the triple (l,r,k) which is $O(KN^2)$. For each explored triple, the Gini index (complexity of $O(N)$) is evaluated, leading to a total complexity for *AdaptSplit* of $O(KN^3)$.

The complexity of *AdaptSplit* can be reduced by limiting the exhaustive search to the pairs of time series of different classes. On the other hand, *DichoSplit* performs two calls to *AdaptSplit* and a recursive call to *DichoSplit* if either I_L or I_R provides a better purity Gini index than the interval I. The maximum number of recursive calls is $\mathrm{Log}_{\frac{1}{\alpha}}(T)$, corresponding to the number of dichotomous splits of $I = [0,T]$ that would produce a sub-interval of length one. Thus, in the worst case, the complexity of *DichoSplit* is $O(\mathrm{Log}_{\frac{1}{\alpha}}(T) 2\, KN^3)$.

Finally, based on the complexities of *AdaptSplit* and *DichoSplit*, the complexity of *TSSplit* is dominated by $O(\mathrm{Log}_{1/\alpha}(T) 2\, KN^3 + KN^3)$, which is, globally, approximately $O(Log_{1/\alpha}(T) KN^3)$. Note that if T is in the order of N, the complexity of distance-based time series classifiers *TSTree*, Yamada et al. [13], Balakrishnan and Madigan [14], etc.) is generally dominated by $O(N^4)$.

5. Experimental study

5.1. Frequently used datasets

The proposed time series classification tree *TSTree* is first applied to four public datasets, CBF [22], CBF-TR [9], CC [23], and TWO-PAT [9]; which are frequently used in the literature for the validation of the major competitive approaches. Figs. 3–6 show the time series class specifications.

Note that the four datasets share some similar characteristics: each class identifies a distinctive global behavior; classes are well discriminated by their global behaviors; and time series progress in relatively close domains. In real applications, time series specifications may be more complex. For instance, time series may involve time delay, have tendency or amplitude variations, share a global common profile or be characterized by certain local common events.

To complete and broaden the validation process to properties frequently encountered in temporal applications, we propose five additional datasets. On the one hand, we propose three new and synthetic time series datasets that include variations in the range of values, tendency effects, and time series discrimination based on local events rather than on global behaviors. On the other hand, to illustrate the tree induction on real and multivariate time series, we use two multivariate real datasets describing character trajectories and handwritten digits [23]. In the following section, we detail the specifications of these additional datasets.

5.2. Additional time series datasets

5.2.1. Local discrimination (LOCAL-DISC)

The aim of the LOCAL-DISC dataset is to study the efficiency of time series classification trees when faced with time series discrimination based on local events. The LOCAL-DISC dataset is composed of three time series classes, *Begin*, *Middle* and *End*. In the *Begin* class, the time series share a common local event characterized by a small bell arising at the beginning of the period; the *Middle* class consists of time series sharing similar global behavior characterized by a centered large bell; and the time series of the *End* class share a common local event corresponding to a bell arising at the end period. The *Begin*, *Middle* and *End* time series classes are illustrated in Fig. 7. First, note that the global behavior is not a discriminative criterion as time series of different classes may share similar global behaviors (e.g., a

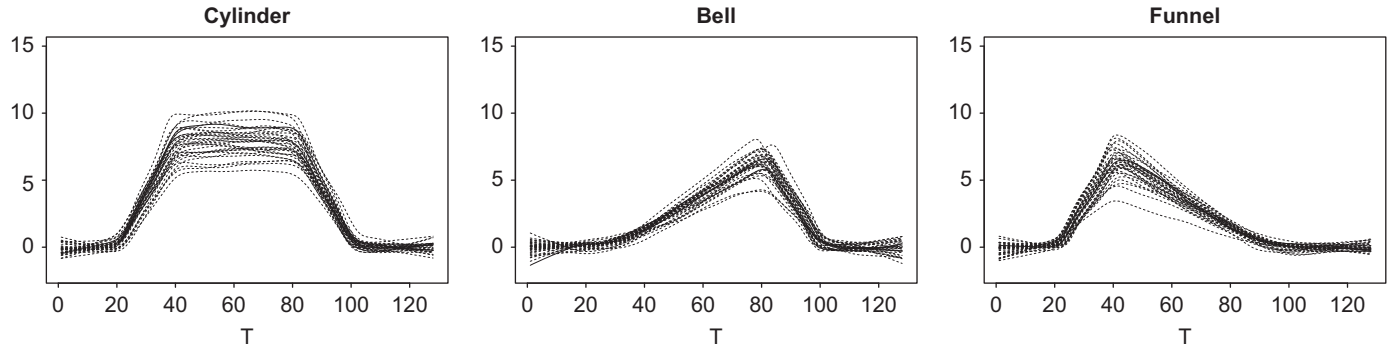

Fig. 3. CBF.

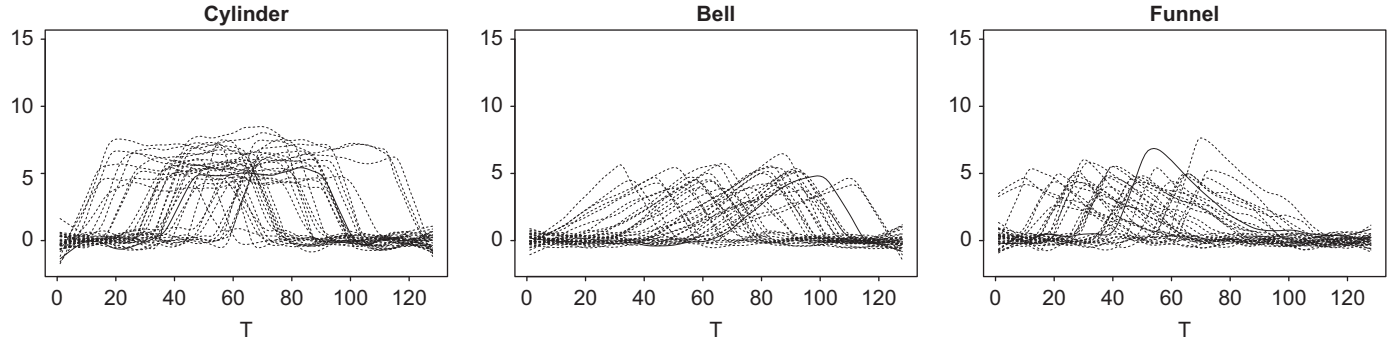

Fig. 4. CBF-TR.

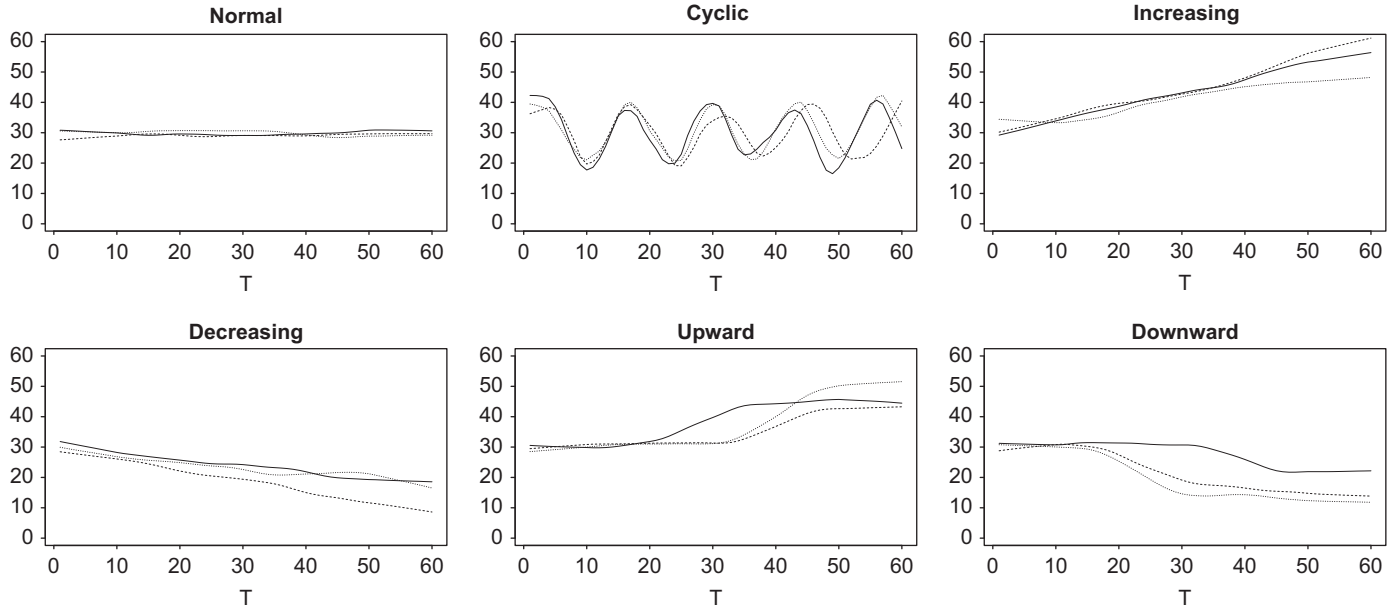

Fig. 5. CC.

cylinder shape). Second, time series of the same class may eventually have different global behaviors and progress in different ranges of values.

5.2.2. *Range variations (CBF-RANGVAR)*

The aim of the CBF-RANGVAR dataset is to study the efficiency of time series classification trees when faced with time series progressing in different ranges of values. The CBF-RANGVAR dataset is obtained by introducing random variations to the CBF-TR time series. As illustrated in Fig. 8, time series of the same class still share the same global profile, but with a progression in different ranges of values.

5.2.3. *Genes expression profiles (GENES)*

The GENES dataset describes genes expression profiles during the cell division cycle. Our aim with this dataset is to study the

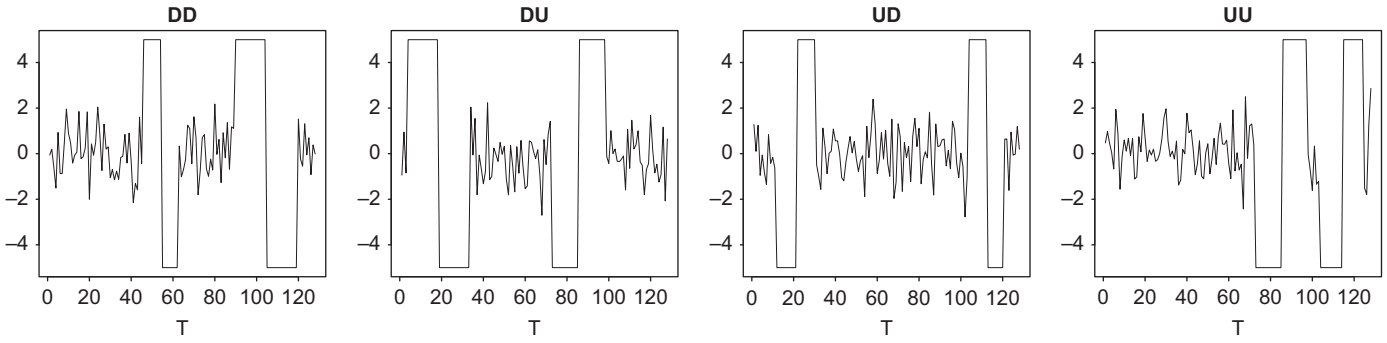

Fig. 6. TWO-PAT.

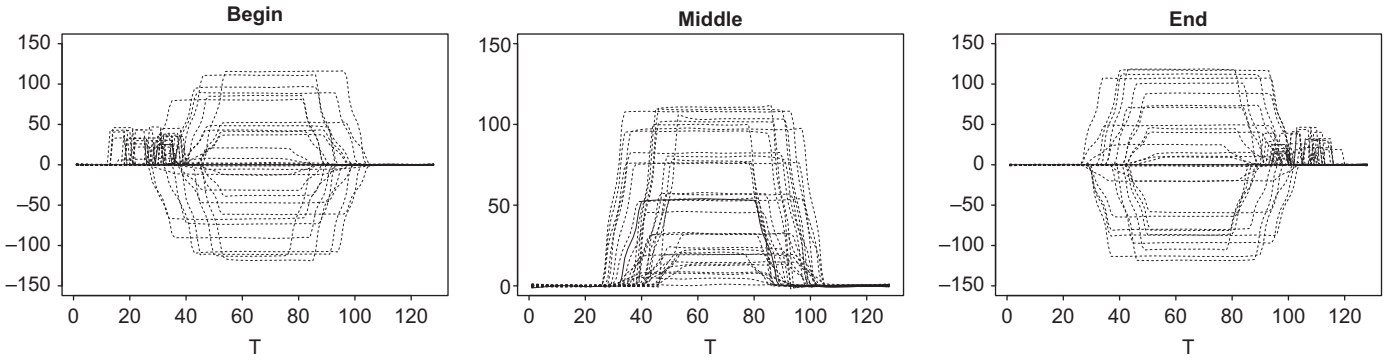

Fig. 7. LOCAL-DISC time series classes.

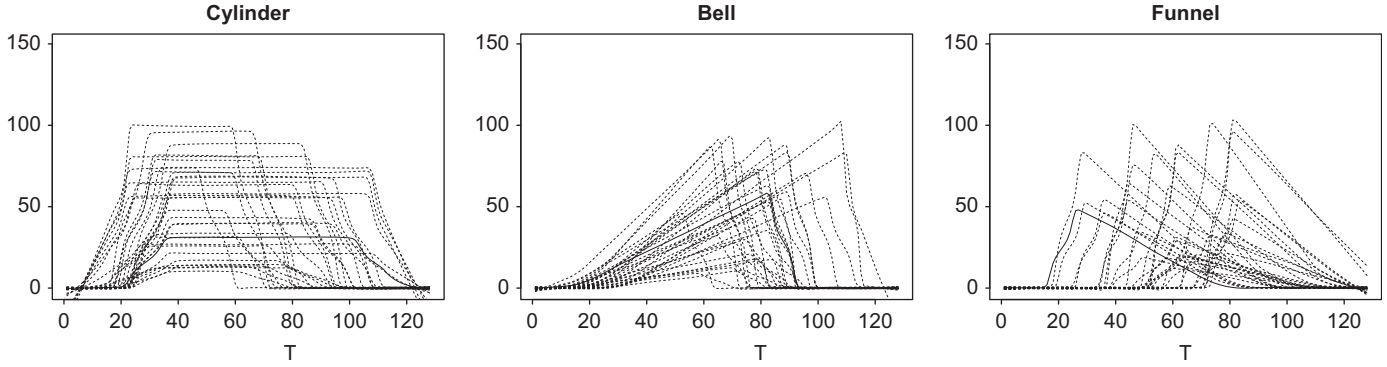

Fig. 8. CBF-RANGVAR time series classes.

efficiency of time series classification trees when faced with classes including periodicity, tendency effects and variations in the range of values; which are types of variations observed experimentally in genes expression profiles. The GENES time series classes do not involve time delays, as the profiles are periodic and the identification of active genes during the cell-cycle is mainly based on the time at which genes are highly expressed. Simulated profiles are generated using the random-periods model proposed by Liu et al. [24]. The model accounts for observed biological variations, such as attenuation in cycle amplitude, drift in the expression profiles, and variations of the initial amplitude or of the cycle duration. The sinusoid function for characterizing the expected periodic expression of a cell-cycle gene g is as follows:

$$f(t,\theta_g) = a_g + b_g t + \frac{K_g}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \cos\left(\frac{2\pi t}{T\exp(\sigma z)} + \Phi_g\right) \exp\left(-\frac{z^2}{2}\right) dz$$

where θ_g is explicitly $(K_g, T, \sigma, \Phi_g, a_g, b_g)$ specific to each gene g. The parameter Φ_g corresponds to the cell-cycle phase during which the gene undergoes its peak transcription, with $\Phi_g = 0$ corresponding to the point when cells are first released to resume cycling. For simulations, Φ_g takes the values 0, 5.190, 3.823, 3.278, or 2.459 to simulate the expression profiles of the five classes G_1/S, S, G_2, G_2/M, or M/G_1, respectively. The parameter K_g defines the initial amplitude of the periodic expression pattern, it varies randomly in [0.34, 1.33]. The parameters a_g and b_g account for any drift (intercepts and slopes, respectively) in a gene's background expression level, with $a_g \in [0,0.8]$ and $b_g \in [-0.05,0.05]$. The parameter σ governs the rate of attenuation in amplitude. If σ is zero, the duration of the cell-cycle does not vary, the cells remain synchronous through time, and the expression profile shows no attenuation in amplitude, for our simulations σ varies randomly in [0.054, 0.115]. Finally, T a parameter of the lognormal distribution of the cell cycle duration is fixed

to 15. Fig. 9 illustrates the progression of the generated genes expression profiles during five cell-cycle phases.

5.2.4. Character trajectories (CHAR-TRAJ)

The character trajectories dataset [23] consists of a set of pen tip trajectories recorded while writing individual characters. All samples are from the same writer, for the purposes of primitive extraction. Only characters with a single pen-down segment were considered. The data were captured using a WACOM tablet. Each handwritten character trajectory is a 3-dimensional time series: x, y for the pen positions and z for the pen tip force.

5.2.5. Handwritten digits (DIGITS)

The handwritten digits data are extracted from the UJI Pen Characters database [23]. Samples are collected from 11 writers, with two samples for each writer/digit pair. Only x and y coordinate information was recorded along the strokes by the acquisition program, without, for instance, pressure level values or timing information. As several handwritten prototypes may have been used by the 11 writers to generate the same digit, a class may be composed of time series of different global behaviors.

Table 2 gives the main characteristics of the above datasets, both commonly used and new: the way the data were generated for this work (Source = 1 for our own simulations performed according to the first paper specifications, 2 for simulations downloaded from the UCI machine learning repository [23] and 3 for our own simulations performed according to specifications indicated in this paper), the sample size, the number of classes (Num. classes), the number of time series per class (Num. TS/class), their lengths (TS length), dimensionality (univariate vs. multivariate), and type (real vs. synthetic).

Table 3 summarizes the time series properties, specifically, if they include time delays, progress in different ranges of values, or have tendency effects, and whether the time series discrimination is based on global or local behaviors. For instance, line 5 indicates that the LOCAL-DISC time series may involve time delays and may progress in different ranges of values and that the discrimination is based on some local events.

5.3. Validation protocol

To highlight and validate the additive value of the new temporal classification tree, several configurations of the split

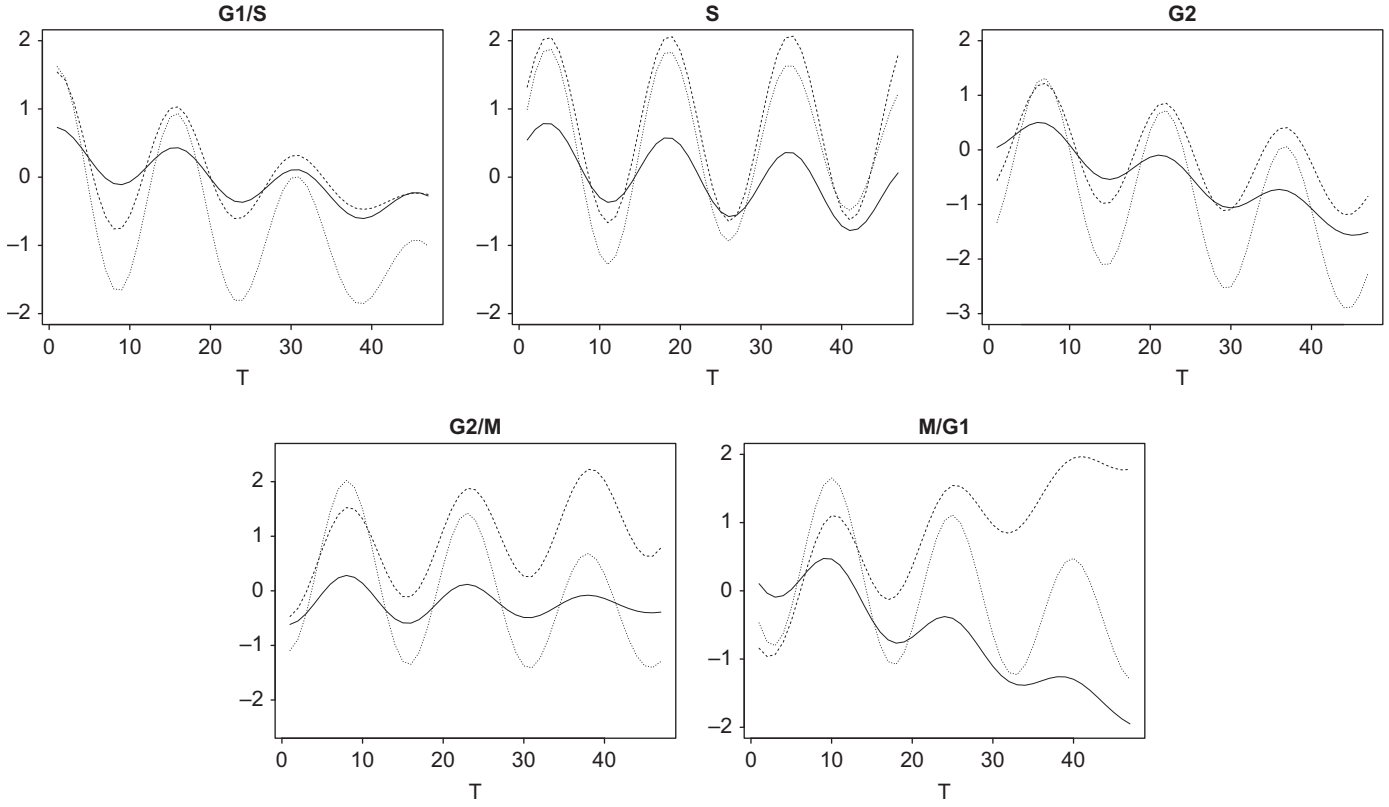

Fig. 9. GENES expression profiles during cell division cycle.

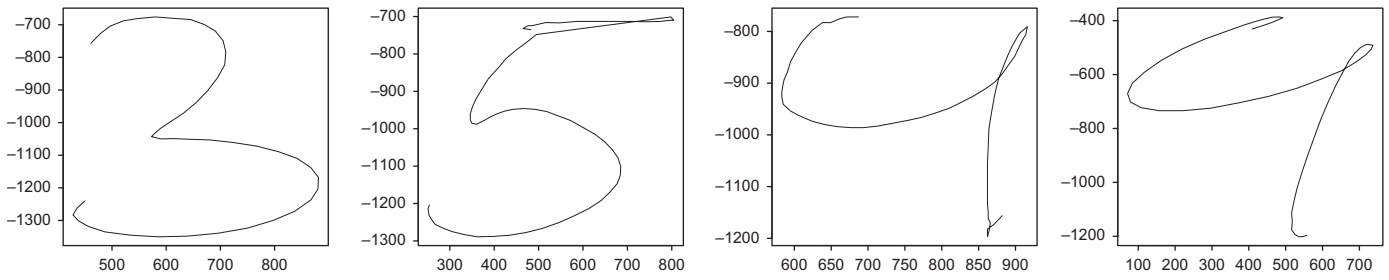

Fig. 10. The closeness of the second half trajectories of the digits 3 and 5 (resp. 4 and 9).

Table 2
Usual and additional datasets description.

Name	Source	Sample size	Num. classes	Num. TS/class	TS lengths	Multi. TS.	Real data
CBF	1	300	3	100	128	No	No
CBF − TR	1	300	3	100	128	No	No
CC	2	600	6	100	60	No	No
TWO − PAT	1	400	4	100	128	No	No
LOCAL − DISC	3	300	3	100	128	No	No
CBF − RANGVAR	3	300	3	100	128	No	No
GENES	3	250	5	50	47	No	No
CHAR − TRAJ	2	400	20	20	[100–200]	Yes	Yes
DIGITS	2	220	10	22	110	Yes	Yes

Table 3
Time series specifications.

Name	Time delay	Range vari.	Tend. effect	Local discr.
CBF	No	No	No	No
CBF-TR	Yes	No	No	No
CC	Yes	No	No	No
TWO-PAT	Yes	No	No	No
LOCAL-DISC	Yes	Yes	No	Yes
CBF-RANGVAR	Yes	Yes	No	No
GENES	No	Yes	Yes	No
CHAR-TRAJ	Yes	No	No	No
DIGITS	Yes	No	No	Yes

Table 4
The studied configurations for *TSSplit*.

Time delay	Adap. metric	Dicho. search	Behav. cost	Metric
Yes	Yes	Yes	*Cort*	DTW_k^{cort}
	Yes	Yes	*Cor*	DTW_k^{cor}
	Yes	No	*Cort*	DTW_k^{cort}
	Yes	No	*Cor*	DTW_k^{cor}
	No	No	–	d_{Dtw}
No	Yes	Yes	*Cort*	DE_k^{cort}
	Yes	Yes	*Cor*	DE_k^{cor}
	Yes	No	*Cort*	DE_k^{cort}
	Yes	No	*Cor*	DE_k^{cor}
	No	No	–	d_E

procedure are considered: an adaptive metric (i.e., a behavior and values based metric) vs. a non-adaptive metric (i.e., a classical values based metric), a dichotomous vs. a non-dichotomous approach, and a temporal correlation vs. a total correlation for the behavior cost-function. In addition, according to classes including or not including time delays (see Table 3), these configurations are modulated for several variants of the DTW, or of the Euclidean distance, respectively. Table 4 summarizes the studied configurations of *TSSplit*. A time series classification tree is induced for each dataset given in Table 2 and for each metric specified in Table 4. A misclassification error rate, based on a 10-fold stratified cross-validation, is estimated (i.e., the folds contain approximately the same proportions of labels as the original dataset.).

Furthermore, the proposed classification tree is compared to the Hidden Markov Model (HMM) classifier (based on the R package RHmm), which is used in similar conditions to *TSTree*, namely, without a great deal of a priori knowledge on data. For this purpose, an HMM is learned for each class based on a training set and using the BaumWelch algorithm. To classify a test time series, the likelihood of each model is calculated, and the assignment is performed according to the most likely model. The HMMs are performed with several numbers of states (3, 5, 7 and 9) and the following typologies: ergodic transition matrix, left-right (band matrix with a bandwidth=2, 3), and an upper triangular matrix. The minimum misclassification error rates over all the studied typologies and numbers of states are reported and discussed in Section 6. All of the computations presented in this paper were performed on the cluster HEALTHPHY (CIMENT, GRENOBLE, FRANCE).

Table 5
Times series classification trees on the usual datasets: adaptive & Dicho vs. static metrics.

Datasets	Metric	Adap.	Dicho.	Error rate	Nb. leaves
CBF	DE_k^{Cort}	Yes	Yes	**0.000**	3
	DE_k^{Cor}	Yes	Yes	**0.000**	3
	DE_k^{Cort}	Yes	No	**0.000**	3
	DE_k^{Cor}	Yes	No	**0.000**	3
	d_E	No	No	0.006	3
CBF-TR	DTW_k^{Cort}	Yes	Yes	**0.023**	3
	DTW_k^{Cor}	Yes	Yes	0.170	22
	DTW_k^{Cort}	Yes	No	**0.023**	3
	DTW_k^{Cor}	Yes	No	0.183	23
	d_{Dtw}	No	No	0.136	30
CC	DTW_k^{Cort}	Yes	Yes	**0.005**	6
	DTW_k^{Cor}	Yes	Yes	0.028	7
	DTW_k^{Cort}	Yes	No	**0.005**	6
	DTW_k^{Cor}	Yes	No	0.025	10
	d_{Dtw}	No	No	0.021	13
TWO-PAT	DTW_k^{Cort}	Yes	Yes	0.002	6
	DTW_k^{Cor}	Yes	Yes	0.002	4
	DTW_k^{Cort}	Yes	No	0.002	6
	DTW_k^{Cor}	Yes	No	0.002	4
	d_{Dtw}	No	No	**0.000**	4

5.4. Performance results

Tables 5 and 7 give, for each standard and new dataset, the misclassification error rates and the number of leaves of the induced trees. These results allow us to study the effect of each *TSSplit*'s configuration (Table 4) on the performances of the induced tree. In particular, they allow us to compare the decision trees performances when the split criterion uses an adaptive metric (Adap=Yes) vs. a static one (Adap=No) and when it involves a dichotomous search (Dicho=Yes) vs. not (Dicho=No).

The aim of Table 6 is to situate the performances of *TSTree* (using an adaptive metric based on the temporal correlation) with those of six time series classification methods achieving competitive performances on the frequently used datasets considered. Let us describe briefly the spirit of each approach. The DTP is a decision tree based on prototype extraction, proposed by Geurts [8,9]. It relies on a linear piecewise model to extract discriminative subsequences from the time series samples, and some of the extracted subsequences are subsequently selected to characterize each class and classify new time series. In the same papers,

a segmentation approach SEG is proposed that consists of dividing the set of time series into equal length segments. The time series are then described by numerical features defined by the average values observed within extracted segments where, finally, standard classifiers can be applied. The segment and combined SC approach proposed in Geurts et al. [10] relies on a generic preprocessing of selected subsequences, all of which are of the same length. A generic supervised learning method is applied to the sample of subsequences so as to derive subsequence classifiers. The FDT is the functional decision tree approach proposed by Balakrishnan and Madigan [14] and described in Section 2. The BOOSTINT approach by Rodríguez et al. [7] extends (boosts) an inductive logic programming system with the definition of time series predicates that are suited for the task of time series classification. The last approach, TCLASS, proposed in Kadous and Sammut [11], extracts parameterized events from time series. These events are clustered in the parameter space, and the resulting prototypes are used as a basis for creating classifiers.

Table 6
Performances of *TSTree* and six competitive time series classifiers.

Time series classifiers	CBF	CBF–TR	CC	TWO–PAT
TSTree	0.0000	0.0233	0.0050	0.0026
DTP [8]	0.0117	0.0233	0.0233	–
SEG [8]	0.0050	0.0166	0.0050	–
SC (Geurts et al., 2005)	0.0038	0.0163	0.0033	0.0037
FDT (Balakrishnan et al., 2006)	0,0013	–	0,0200	–
BOOSTINT [7]	0.0113	0.0664	0.0083	0.2000
TCLASS (Kadous, 2005)	0	–	–	–

Figs. 11–18 visualize the trees minimizing the error rate over the studied *TSSplit*'s configurations. Let us first introduce some interpretation elements of the built classification trees. Each node is characterized by the triplet (*Type*,I_*,*Class*), indicating, respectively,

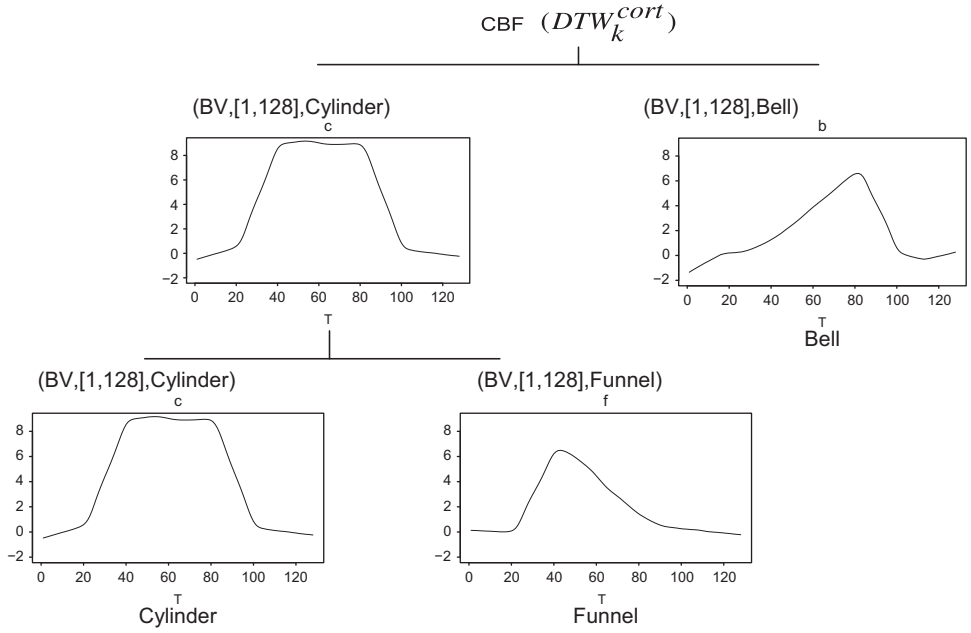

Fig. 11. Classification tree of CBF data.

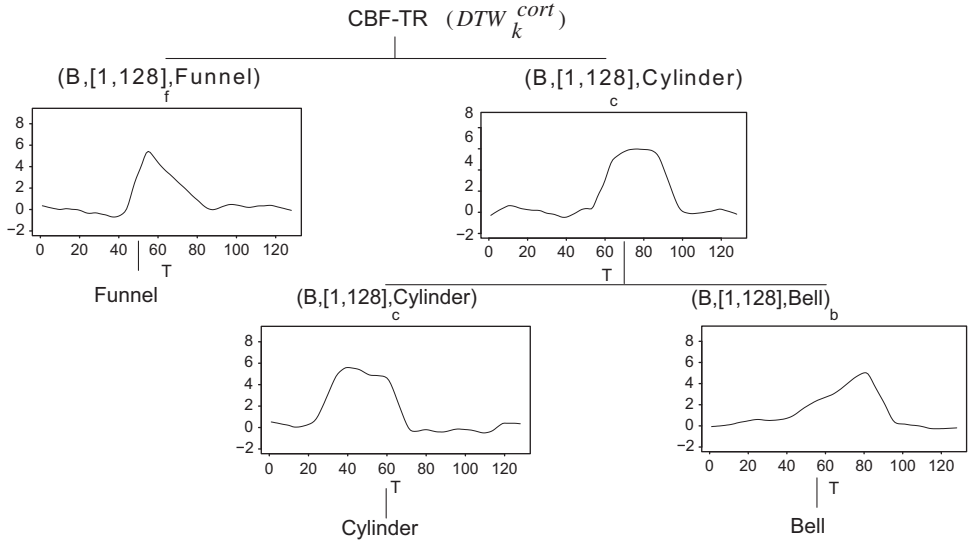

Fig. 12. Classification tree of CBF–TR data.

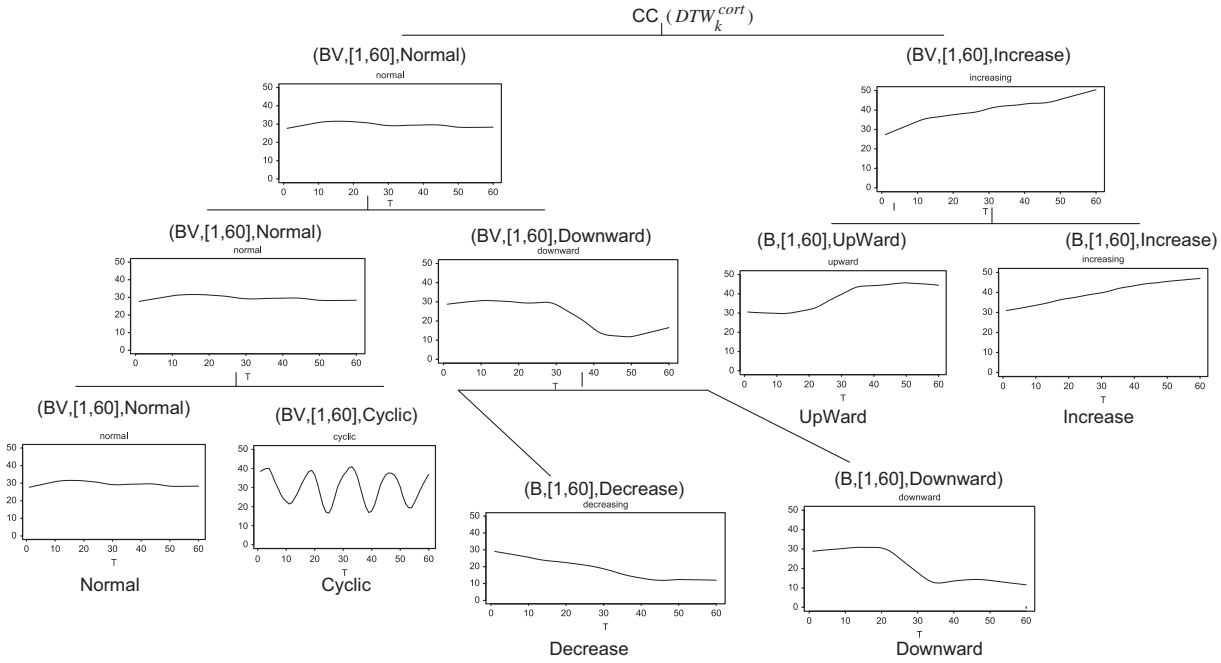

Fig. 13. Classification tree of CC data.

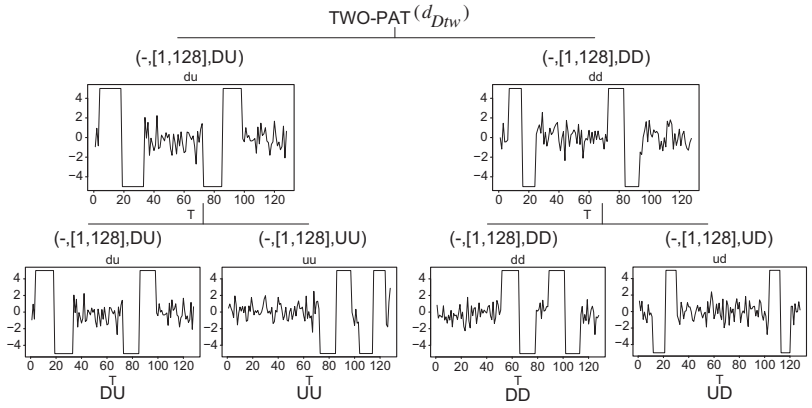

Fig. 14. Classification tree of TWOPAT data.

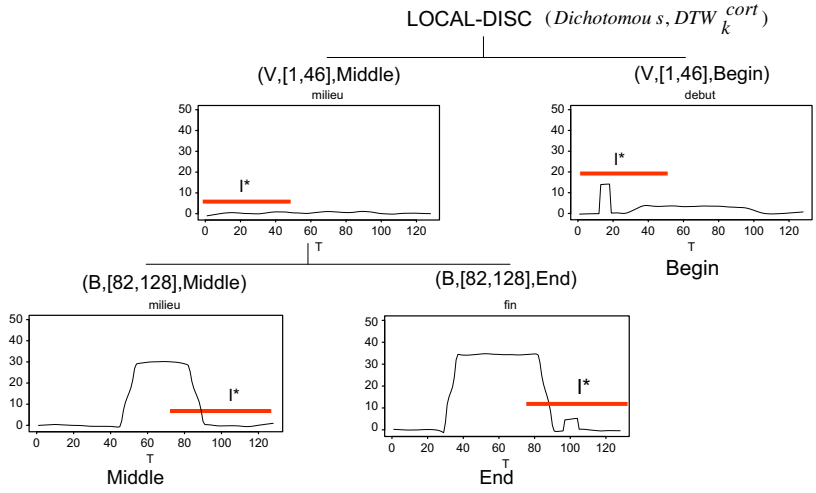

Fig. 15. Classification tree of LOCAL-DISC data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the metric's type (that is if the learned D_{k_*} is behavior-based labeled "B" for k_* greater than 3, values-based labeled "V" for k_* lower than 3, or equally behavior and values-based labeled "BV" for $k=3$), the most discriminating interval or sub-interval I_* on which D_{k_*} will be evaluated, and the class label of the representative time series.

For instance, for the CBF classification tree (Fig. 11), the first split selects two representative time series $l_{cylinder}$ and r_{Bell} from the *Cylinder* and *Bell* classes. A new time series *ts* is assigned to the left sub-tree if it is closer to $l_{Cylinder}$ than to l_{Bell}, that is, $DTW^{cort}_{k_*}(ts,l_{Cylinder}) \leq DTW^{cort}_{k_*}(ts,r_{Bell})$; otherwise, it is assigned to the right sub-tree. The dissimilarities $DTW^{cort}_{k_*}$ are evaluated on the whole observation interval $I_* = [1,128]$, and they involve equally the behaviors and values components.

Let us now discuss some elements of the LOCAL-DISC tree involving a dichotomous search (Fig. 15). The first split is characterized by two reference time series from the *Middle* and *Begin* classes, and a learned dissimilarity $DTW^{cort}_{k_*}$, to be evaluated on the localized discriminating subsequence [1, 46] (red underlined). During that period, the left representative subsequence describes a line, and the right representative subsequence describes a bell. Thus, the left

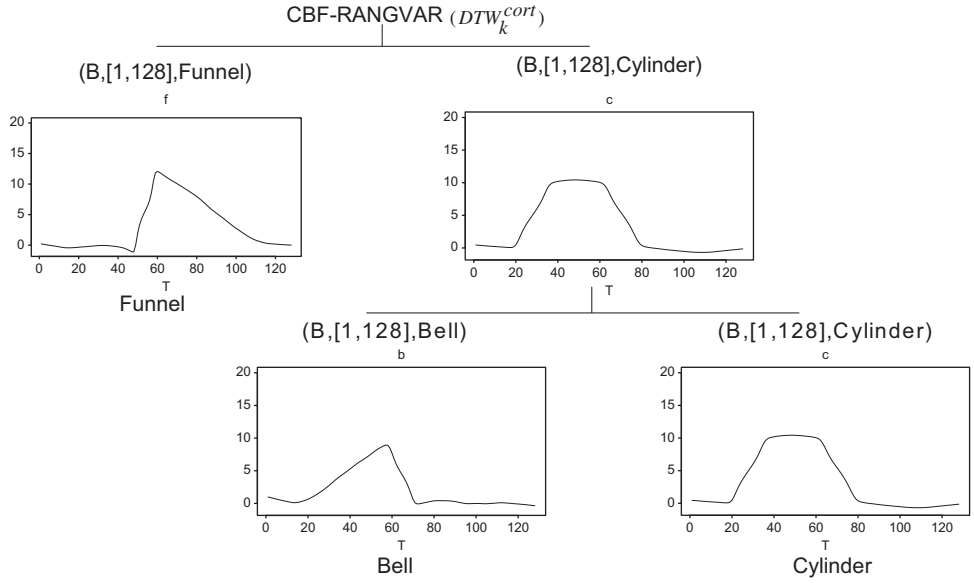

Fig. 16. Classification tree of CBF-RANGVAR data.

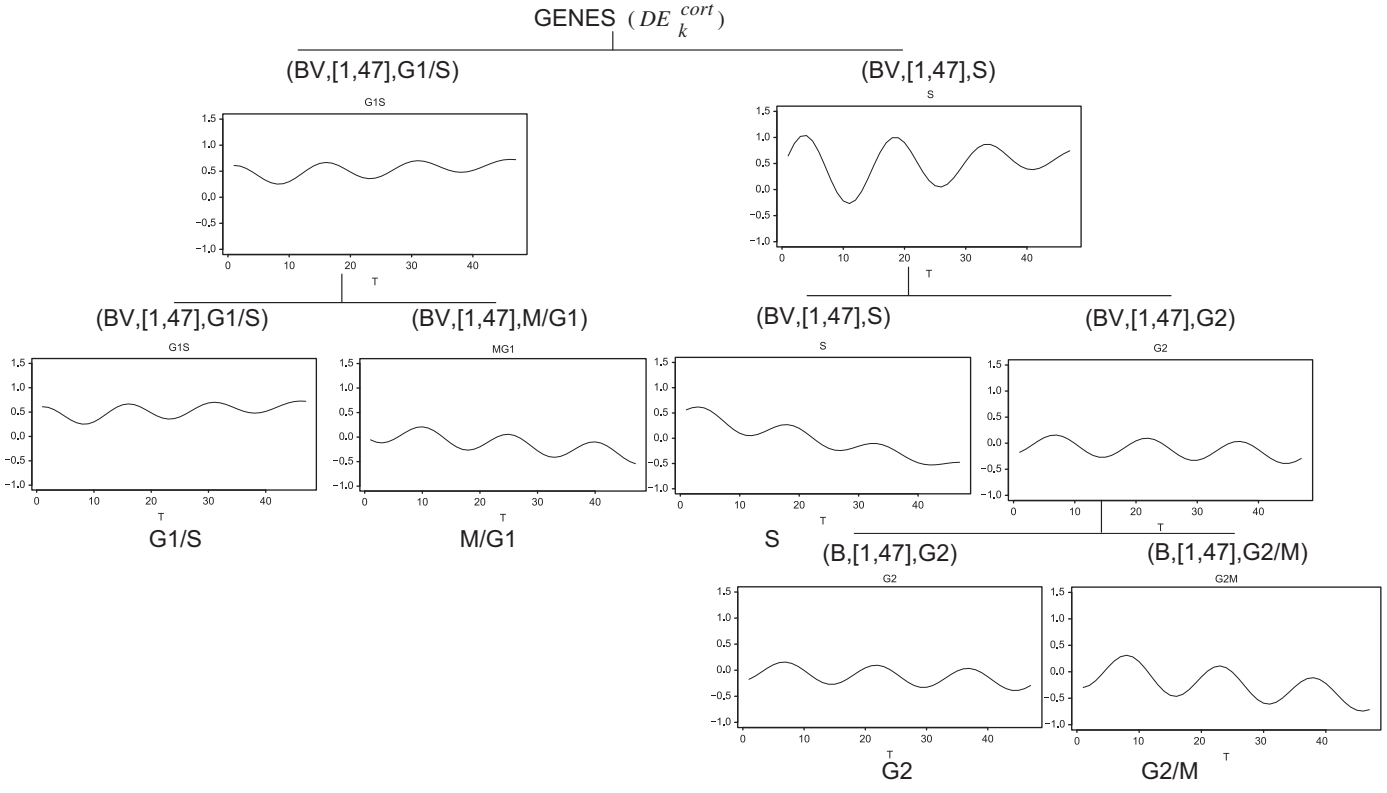

Fig. 17. Classification tree of GENES data.

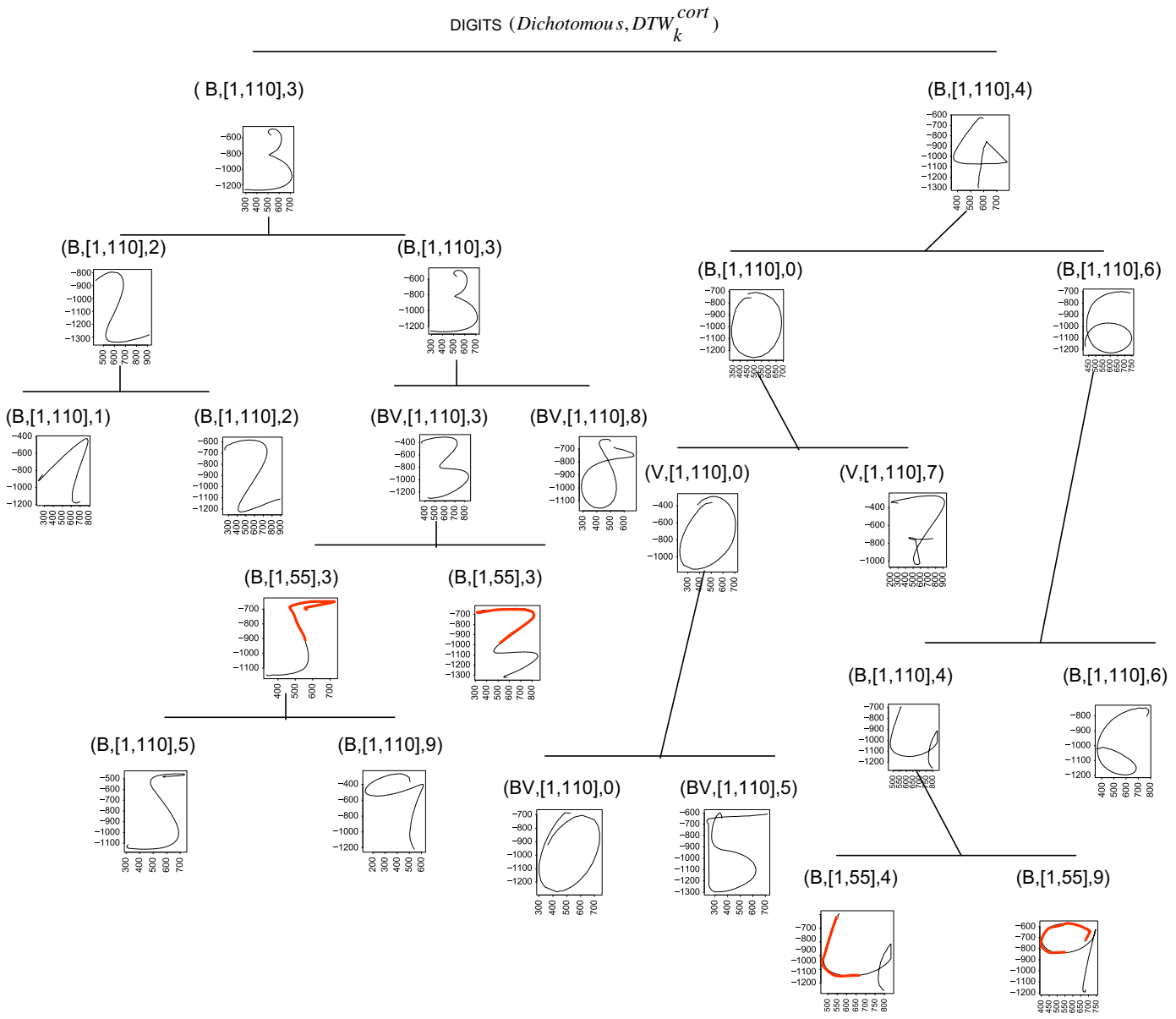

Fig. 18. Classification tree of DIGITS data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

subtree will contain time series of the *End* or *Middle* classes as characterized by a line during the first observation period, whereas the right subtree will contain time series of the *Begin* class, which is characterized by a small bell appearing at the first observation period. For the second split, to separate the time series of the *End* and *Middle* classes, the dichotomous search localizes the subsequence [82, 128] as the most discriminating. The left subsequence describes a line, and the right subsequence describes a small bell. Thus, the left subtree will contain time series of the *Middle* classes as characterized by a line during the last observation period, whereas the right subtree will contain time series of the *End* class, which are characterized by a small bell arising in the last period.

Finally, for all induced trees, the interpretation rules indicate, for each split, the typical profiles of the time series belonging to each child node, whether the profiles cover all the observation periods or reference subsequences, and whether the assignation rules are values or behavior based. Although the best specification for *TSSplit* involves an adaptive metric, the temporal correlation and a dichotomous search, studying the other configurations may be informative: it may reveal whether the time series involve amplitude variations and drifts, whether the discrimination is due to local events or to the global behavior, and whether it is values or behavior based. Fig. 19 summarizes the main interpretation rules revealing the temporal data specifications.

6. Discussion

From Table 5 we can see that for all datasets (except for the noisy TOW-PAT), a *TSTree* based on an adaptive metric (the fourth first configurations) outperforms a tree based on static metrics (d_E or d_{DTW}). The performances of *TSTree* remain the same whether it involves or does not involve the dichotomous search, revealing that each class is characterized by one distinctive global behavior and a discriminatory power that is mainly based on the global behaviors of time series. Finally, for all of the datasets, the performances of *TSTree* are improved when it involves the temporal correlation instead of the total correlation.

Let us discuss the performances of Table 6 (empty cells correspond to unpublished results). First, we can note that *TSTree*

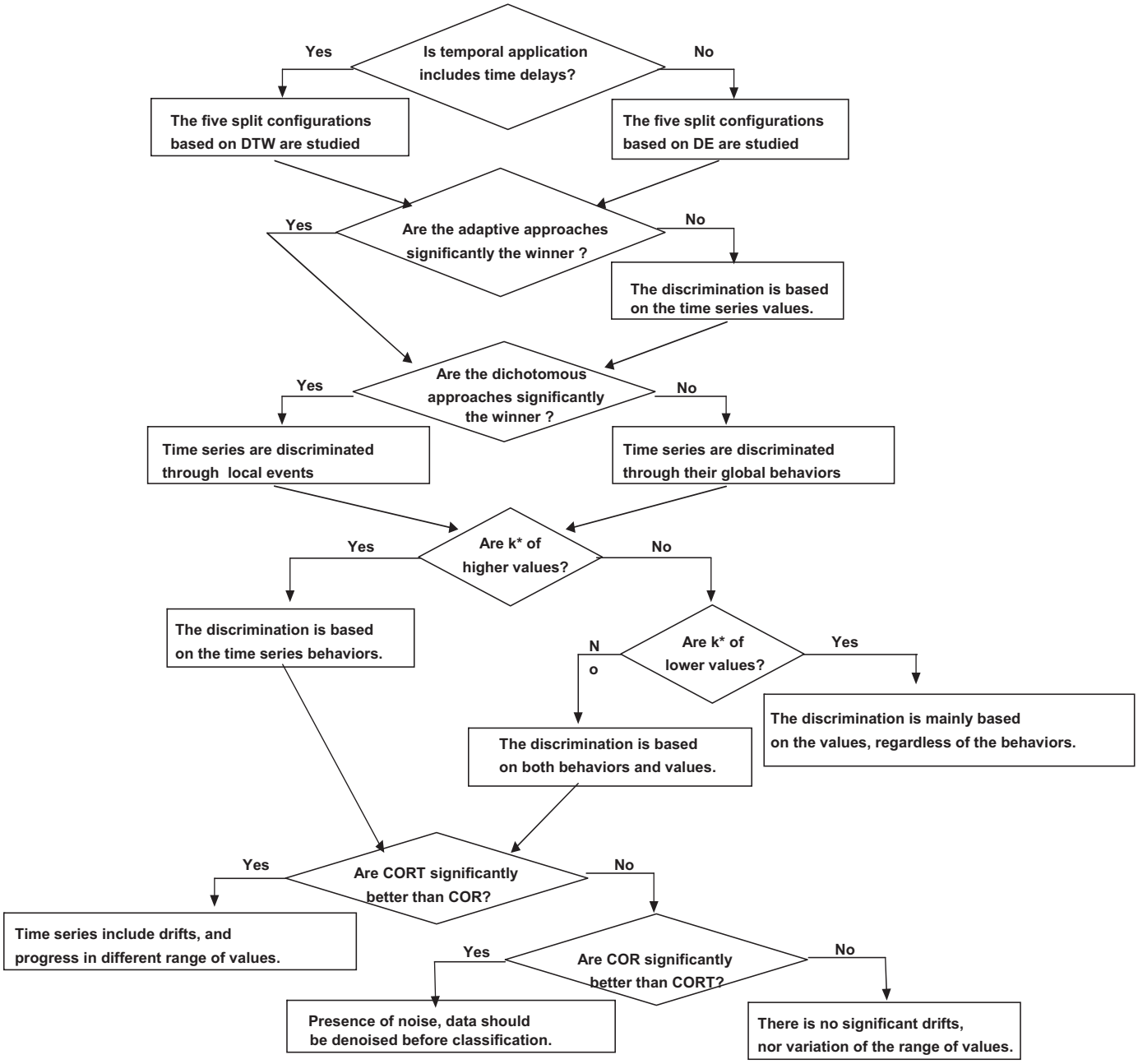

Fig. 19. Interpretation rules of the split configurations results.

and the six considered approaches globally succeed in classifying the frequently used datasets with low error rates. These performances mainly reveal, as pointed in paragraph 5.1, a good separability of the time series classes and too simplistic temporal peculiarities of the frequently used datasets. Although these results appear comparable, it is important to show that *TSTree* is competitive with better performances 11 times out of the 17 published results, the same performances three times, and lower performances only three times. The performances shown in Table 6 reinforce the idea that one needs to validate time series classification approaches on more realistic temporal specifications.

From Table 7, we can see that for all additional datasets introducing several complex temporal peculiarities, the *TSTree* based on an adaptive metric outperforms a tree based on static metrics (non adaptive). These performances are always improved with use of the temporal correlation instead of the total correlation.

The dichotomous search significantly improves the results for LOCAL-DISC and DIGITS, as classes may be composed of time series of different global behaviors. In fact, for DIGITS the 11 writers may follow different trajectories to write the same digit. From the tree given in Fig. 18, we can see that the dichotomous search plays a part at two nodes: in separating the digits 3 and 5, then in separating 4 and 9. In fact, time series of the digits 3 and 5 (resp. 4 and 9) provided by different writers may be very close on the second half of the trajectories as shown in Fig. 10. Thus, the dichotomous search selects the first half of the trajectories (underlined in red in Fig. 18) as best discriminating these digits. In other words, to best separate the digits 3 and 5 (or 4 and 9) the dissimilarities between those digits are evaluated based on the first half of their trajectories.

In addition we compare *TSTree* and HMM classifier performances based on similar conditions, that is, without a great deal of prior knowledge of the data (see specifications in paragraph 5.3).

Table 7
Times series classification trees on the additional datasets: adaptive & Dicho vs. static metrics.

Datasets	Metric	Adap.	Dicho.	Error rate	Nb. leaves
LOCAL-DISC	DTW_k^{cort}	Yes	Yes	**0.020**	3
	DTW_k^{cor}	Yes	Yes	**0.020**	5
	DTW_k^{cort}	Yes	No	0.073	13
	DTW_k^{cor}	Yes	No	0.096	22
	d_{Dtw}	No	No	0.096	30
CBF-RANGVAR	DTW_k^{cort}	Yes	Yes	**0.006**	3
	DTW_k^{cor}	Yes	Yes	0.053	10
	DTW_k^{cort}	Yes	No	**0.006**	3
	DTW_k^{cor}	Yes	No	0.070	15
	d_{Dtw}	No	No	0.060	21
GENES	DE_k^{cort}	Yes	Yes	**0.004**	5
	DE_k^{cor}	Yes	Yes	**0.004**	5
	DE_k^{cort}	Yes	No	**0.004**	5
	DE_k^{cor}	Yes	No	**0.004**	5
	d_E	No	No	0.036	8
CHAR-TRAJ	DTW_k^{cort}	Yes	Yes	**0.075**	20
	DTW_k^{cor}	Yes	Yes	0.082	20
	DTW_k^{cort}	Yes	No	**0.075**	24
	DTW_k^{cor}	Yes	No	0.095	24
	d_{Dtw}	No	No	0.080	24
DIGITS	DTW_k^{cort}	Yes	Yes	**0.065**	12
	DTW_k^{cor}	Yes	Yes	0.141	11
	DTW_k^{cort}	Yes	No	0.141	13
	DTW_k^{cor}	Yes	No	0.161	12
	d_{Dtw}	No	No	0.247	16

The obtained HMM's performances on the standard and additional datasets are: CBF (0.013), CBF-TR (0.140), CC (0.070), TWO-PAT (0.730), LOCAL-DISC (0.533), CBF-RANGVAR (0.320), GENES (0.610), CHAR-TRAJ (0.500), and DIGITS (0.450). These weak performances mainly rely on four factors: (1) For a classification task, an HMM is generally learned for each cluster regardless of the discriminative elements between the classes. Thus, even though the HMMS succeed in modeling each class, they may fail to achieve good classification accuracy. In particular, if the time series of different clusters are of nearly similar global behaviors with few differences between classes, as is the case for LOCAL-DISC and CHAR-TRAJ, the learned HMM of a given class may generate adequately time series of an other class. (2) An HMM assumes the sequences as piecewise stationary processes, a property violated particularly by the GENES dataset, which includes high tendency effects. (3) It is known that higher amplitude variations, combined with heteroscedasticity (the variance is time dependent), appearing for nearly all additional datasets, decrease the performances and the accuracy of HMMS. (4) The considered model based on one learned HMM per class is not efficient if faced with classes composed of several prototypes as for LOCAL-DISC and the multi-writer DIGITS. We are convinced that HMMS may be able to succeed in classifying these complex data, but would need great and costly effort to fine-tune a huge number of parameters based on an a priori deep analysis of the data. That analysis may include, for instance, appropriate data preprocessing, estimating the number of states and typologies for each learned HMM, using discriminative models, and estimating several HMMS per class (if several prototypes appear within classes). A good survey on the prerequisites and limitations of HMMS is provided in Bilmes [25].

In conclusion, the performances obtained for *TSTree* illustrate the two *TSSplit* additive values. First, knowing that temporal peculiarities may change from one internal node to another when building the decision tree, using an adaptive metric that allows us to modulate the metric from one internal node to another to best bisect the time series, leads to better performances than using a static metric. Second, when faced with classes composed of time series of different global behaviors, the decision trees are more efficient when using a dichotomous search to localize the discriminant subsequences. Time series are subsequently compared on those discriminating subsequences instead of their different global behaviors.

Our purpose in future work is to make the proposed time series split test more robust to the over-fitting problem. To this end, our aim is to introduce more variability in the proposed split test in the following ways: first, by enlarging the selection of the pair of representative time series to the selection of left and right sub-sets of representative time series; second, by keeping a set of partitions providing an equivalent purity Gini index instead of selecting one partition maximizing the Gini index; and finally, by selecting the best partition using a new isolation criterion based on the distribution of the misclassified time series within classes.

7. Conclusion

The proposed time series classification tree is based on a new time series split procedure involving two main functionalities. First, an adaptive (i.e., parameterized) time series metric is used to cover both behavior and values proximities. The metric's parameters may change from one internal node to another to partition the time series most effectively. Second, the method includes the automatic extraction of the most discriminating subsequences. The experiments performed suggest considering a split test based on an adaptive metric, including a dichotomous search, and based on a temporal correlation coefficient for a behavior proximity measure. This paper reveals *TSTree* as a promising time series classifier that outperforms temporal trees using standard distances and leads to good performances compared to other competitive time series classification methods.

Acknowledgments

The authors thank Eric Gaussier for the useful discussion on the algorithm complexity, and the anonymous reviewers for their valuable comments.

References

[1] L.A. Garcia-Escudero, A. Gordaliza, A proposal for robust curve clustering, Journal of Classification 22 (2005) 185–201.
[2] N. Serban, L. Wasserman, CATS: cluster after transformation and smoothing, Journal of the American Statistical Association 100 (2004) 990–999.
[3] J. Caiado, N. Crato, D. Pena, A periodogram-based metric for time series classification, Computational Statistics and Data Analysis 50 (2006) 2668–2684.
[4] Y. Kakizawa, R.H. Shumway, N. Taniguchi, Discrimination and clustering for multivariate time series, Journal of the American Statistical Association 93 (1998) 328–340.
[5] E.A. Maharaj, Cluster of time series, Journal of Classification 17 (2000) 297–314.
[6] M. Kudo, J. Toyama, M. Shimbo, Multidimensional curve classification using passing-through regions, Pattern Recognition Letters 20 (11–13) (1999) 1103–1111.
[7] J.J. Rodríguez, C.J. Alonso, H. Bostrom, Boosting interval-based literals, Intelligent Data Analysis 5 (3) (2001) 245–262.
[8] P. Geurts, Pattern extraction for time series classification, LNCS Principles of Data Mining and Knowledge Discovery (2001) 115–127.
[9] P. Geurts, Contributions to Decision Tree Induction: Bias/Variance Tradeoff and Time Series Classification, Ph.D. Thesis, Department of Electrical Engineering, University of Liege, Belgium, 2002.
[10] P. Geurts, L. Wehenkel, Segment and combine approach for non-parametric time-series classification, in: 9th Conference on Principles and Practice of Knowledge Discovery in Database, Springer, 2005.
[11] M.W. Kadous, C. Sammut, Classification of multivariate time series and structured data using constructive induction, Machine Learning Journal 58 (2005) 179–216.
[12] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[13] Y. Yamada, E. Suzuki, H. Yokoi, K. Takabayashi, Decision-tree induction from time-series data based on standard-example split test, in: International Conference on Machine Learning, 2003.
[14] S. Balakrishnan, D. Madigan, Decision trees for functional variables, in: ICDM International Conference on Data Mining, 2006, pp. 798–802.
[15] J.B. Kruskall, M. Liberman, The symmetric time warping algorithm: from continuous to discrete, in: Time Warps String Edits and Macromolecules, Addison-Wesley, 1983.
[16] B.D. MacArthur, A. Lachmann, I.R. Lemischka, A. Ma'ayan, GATE: software for the analysis and visualization of high-dimensional time series expression data, Bioinformatics 26 (1) (2010) 628 143–144.
[17] J. Ernst, G.J. Nau, Z. Bar-Joseph, Clustering short time series gene expression data, Bioinformatics 21 (2005) 159–168.
[18] Z. Abraham, P. Tan, An integrated framework for simultaneous classification and regression of time-series Data, SIAM International Conference on Data Mining, 2010, pp. 653-664.
[19] F. Cabestaing, T.M. Vaughan, D.J. McFarland, J.R. Wolpaw, Classification of evoked potentials by Pearson's correlation in a brain–computer interface, Modelling C Automatic Control (Theory and Applications) 67 (2007) 156–166.
[20] J. Rydell, M. Borga, H. Knutsson, Robust correlation analysis with an application to functional MRI, in: IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, USA, 2008, pp. 453–456.
[21] A. Douzal-Chouakria, A. Diallo, F. Giroud, Adaptive clustering for time series: application for identifying cell cycle expressed genes, Computational Statistics and Data Analysis 53 (4) (2009) 1414–1426 Elsevier.
[22] N. Saito, Local Feature Extraction and Its Application Using a Library of Bases, Ph.D. Thesis, Department of Mathematics, Yale University, 1994.
[23] A. Asuncion, D.J. Newman, UCI Machine Learning Repository [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. University of California, School of Information and Computer Science, Irvine, CA, 2007.
[24] D. Liu, D.M. Umbach, S.D. Peddada, L. Li, P.W. Crockett, C.R. Weinberg, A random-periods model for expression of cell-cycle genes, Proceedings of the National Academy of Sciences of the United States of America 101 (2004) 7240–7245.
[25] J.A. Bilmes, What HMMS can do, IEICE Transactions on Information and Systems E89-D (3) (2006) 869–891.