2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A proposal for the hierarchical segmentation of time series. Application to trend-based linguistic description.

R. Castillo-Ortega, N. Marín, *Member, IEEE*, C. Martínez-Cruz, and D. Sánchez, *Senior Member, IEEE*

***Abstract*—In this paper we propose methods for obtaining hierarchical segmentations of time series on the basis of the Iterative End-Point Fit Algorithm. We discuss on the utility of the methods for different cases. We illustrate the usefulness of the hierarchical segmentations with an application in linguistic description of trends in time series. A linguistic description based on a segmentation of the time series that do not necessarily corresponds to a level of the hierarchy is obtained by describing segments in different levels that form a segmentation satisfying a quality model.**

I. Introduction

SEGMENTATION of time series data is a very important starting point in different analysis tasks, like time series data mining for instance, with many proposals available in the literature [6]. The objective is to obtain a so-called *Piecewise Linear Representation (PLR)* of the series, consisting in an approximation of a time series of lenght n using K straight lines (segments) with $K << n$. The lower the value of K, the most efficient is to store and work with the approximation.

The existing methods are designed to provide a single segmentation of the series, using different criteria to solve the tradeoff between quality of the approximation and the value K (the lower the value K, the worse the approximation). The segmentation is usually performed before the mining process, so algorithms performing the latter assume the segmentation to be given.

An example of such situation is mining linguistic descriptions of time series data. In the approach put forward by the authors [2], [1], a linguistic description is a collection of natural language sentences describing relevant features of the time series. Each sentence describes a certain time period. When the feature employed are trends describing the increasing/decreasing behavior of the time series, it is convenient to work on a segmentation, since each segment defines a certain time period, and the slope of the segment informs us of the trend in that period.

However, the quality of the results of the mining algorithms may depend on the segmentation, so it would be interesting to consider different segmentations of the time series, and to explore different approximations *during* the mining process. This is the case for example in the abovementioned trend-based linguistic description of time series. The quality of the description is assessed through a multidimensional model, in which the quality improves as i) the lenght of the description (number of sentences, and hence number of segments in the segmentation) is lower, and ii) the approximation is better (among other quality dimensions [4]), which are conflicting objectives.

We consider that a suitable approach to explore different segmentations during any mining process is to consider a hierarchical segmentation of the series, i.e., a hierarchical structure in which each level is a segmentation, and levels are ordered so that the number of segments in each level increases and, at the same time, the approximation is better. Our approach to linguistic description is based on this kind of structure, performing an exploration of the different levels of the hierarchy during the mining process, and obtaining a final segmentation that i) yields a description with high quality, and ii) may not correspond to any level in the hierarchy, but is built by taking suitable segments at different levels. We believe that other mining processes may well benefit from this approach.

In this paper we propose methods for obtaining hierarchical segmentations of time series on the basis of one of the most employed segmentation algorithms, the Iterative End-Point Fit Algorithm, known in cartography as the Douglas-Peucker algorithm [5], and as Ramer's algorithm in image processing [7]. We illustrate the usefulness of the hierarchical segmentations with an application in linguistic description of trends in time series.

The paper is organized as follows: section II is devoted to present our approaches to the hierarchical segmentation of the time series, while section III concerns to the application of the segmentation proposal to trend-based linguistic descriptions of time series. Some conclusions end the paper.

II. New Hierarchical Segmentation of Time Series

In order to (hierarchically) segment the series, we have to locate the relevant time instants that define the segments, according to a certain segmentation criterion.

Many techniques for segmenting time series are based on the Iterative End-Point Fit (IEPF) algorithm [7], [5]. This algorithm is a well known curve approximation method that receives as input data a curve composed of line segments and produces as output a similar curve with a reduced number of

R. Castillo-Ortega, N. Marín, and D. Sánchez are with the Department of Computer Science and Artificial Intelligence, University of Granada, Spain. email: {rita, nicm, daniel}@decsai.ugr.es. C. Martínez-Cruz is with the Computing Department, University of Jaén, Spain. email: cmcruz@ujaen.es. Daniel Sanchez is with the European Centre for Soft Computing, Mieres, Spain.

Part of this research was supported by the Andalusian Government (Junta de Andalucía, Consejería de Innovación, Ciencia y Empresa) under project P07-TIC-03175 *Representación y Manipulación de Objetos Imperfectos en Problemas de Integración de Datos: Una Aplicación a los Almacenes de Objetos de Aprendizaje*

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

points (a subset of the points that defined the initial curve). The algorithm (see algorithm 1) uses a distance threshold to determine when a single line is a good aproximation for a segment of a curve.

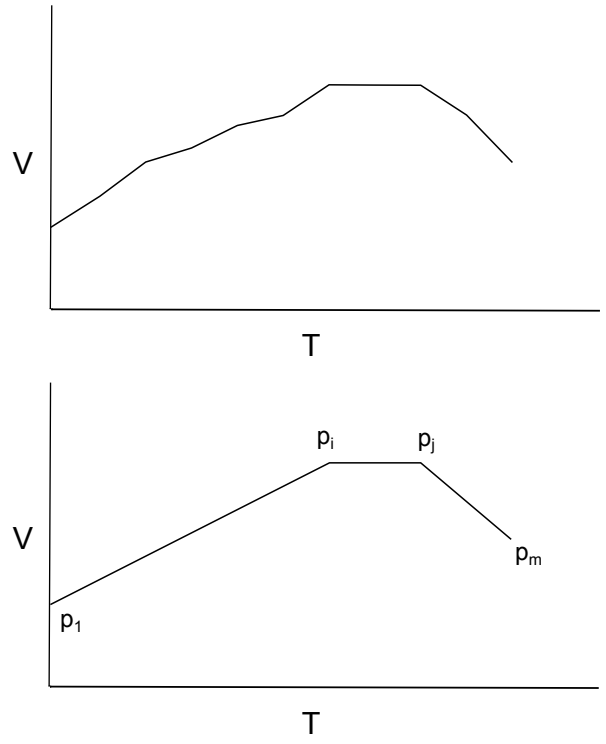

Fig. 1. Approximation of curves with the IEPF algorithm

See, for example, the naive series of the upper part of figure 1. If we execute the algorithm with a distance threshold equal to zero, we obtain an approximation that *perfectly* fits the initial curve. However, as we increase the distance threshold, *less strict* approximations are obtained, composed by fewer and fewer points (like the one showed in the lower part of the figure).

The subset of points obtained by the Iterative End-Point Fit Algorithm can be easily translated into a segmentation by simply considering the segments defined by each (ordered) pair of consecutive points selected by the algorithm. In the example, a segmentation composed by three segments (namely, $\overleftrightarrow{p_1p_i}$, $\overleftrightarrow{p_ip_j}$, $\overleftrightarrow{p_jp_m}$) is obtained.

In general, the threshold is fixed by the user as a tradeoff between the number of segments and the precision of the approximation produced by the segmentation.

A. Hierarchical segmentation based on the IEPF algorithm

The use of the IEPF algorithm described in the previous paragraphs produces a *plain* segmentation of the series. However, in many occasions, it may results interesting to have a *hierarchical* segmentation of the series. This is the case of the proposed approach for the description of trends we focus on later in this research.

In this paper, we consider two alternative ways of using the IEPF algorithm to produce a hierarchical segmentation

Algorithm 1 Algorithm IEPF

Input

TS_{in}[1..m], A time series of m points.
τ, A distance threshold.

Output

$CP_\tau(TS)$, a subset of points of TS_{in}.

Algorithm

```
1: CP_τ(TS)={TS_in[1], TS_in[m]}
2: pos_list= SelectMaxDistancePoints(TS_in, τ)
3: if SIZE(pos_list)>0 then
4:     CP_τ(TS)=CP_τ(TS) ∪ IEPF(TS_in[1..pos_list[1]],τ)
5:     for i=2;SIZE(pos_list) do
6:         CP_τ(TS)=CP_τ(TS) ∪ IEPF(TS_in[pos_list[i-1]..pos_list[i]],τ))
7:     end for
8:     CP_τ(TS)=CP_τ(TS) ∪ IEPF(TS_in[pos_list[SIZE(pos_list)]..m],τ)
9: end if
10: return CP_τ(TS)
```

of the series:

- Recursion depth based hierarchical segmentation.
- Threshold based hierarchical segmentation.

1) Recursion depth based hierarchical segmentation: This approach to building the hierarchy is based on the order the points are found during the execution of the algorithm (see algorithm 1). The algorithm works as follows:

- Firts, it considers the extreme points of the series and adds them to the solution.
- Then, it searches for the middle points of the series which are at maximum distance (above the threshold) to the line defined by the initial and end points of the series. This task is performed by function SelectMaxDistancePoints(TS_{in}, τ); it delivers an empty list when no point is selected.
- With the positions found in the previous step, the algorithm makes recursion for each defined segment.

Figure 2 depicts the first steps of the algorithm on an example series: the algorithm first locates point p_i, and makes recursion on $TS[p_1, p_i]$ and $TS[p_i, p_m]$. Recursion on $TS[p_1, p_i]$ will split in p_j and recursion on $TS[p_i, p_m]$ will split in p_k, and so on.

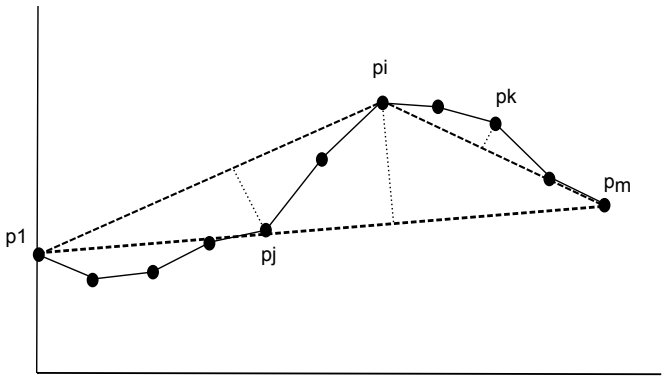

Fig. 2. Initial steps of the Iterative End-Point Fit Algorithm

That is, given a certain value of τ, each point of $CP_\tau(TS)$ is *first added* in a given recursion depth. This depth can be used to build the hierarchy of segments we are looking for.

Definition 2.1: Let $TS = \{< t_1, v_1 >, ..., < t_m, v_m >\}$ be a time series and $CP_\tau(TS)$ be the subset of points obtained by the IEPF algorithm when executed with threshold τ on TS. For each $t_i \in CP_\tau(TS)$, we call recursion depth of t_i, $RD_{TS,\tau}(t_i)$, to the level of recursion in which point t_i is first added to $CP_\tau(TS)$ during the execution of IEPF algorithm.

Without loss of generality, we consider that the initial call to the algorithm has recursion depth equal to 0.

Definition 2.2: Let $TS = \{< t_1, v_1 >, ..., < t_m, v_m >\}$ be a time series and $CP_\tau(TS)$ be the subset of points obtained by the IEPF algorithm when executed with threshold τ on TS. The set of points with recursion depth at most α, $CP_\tau(TS)^\alpha$, is defined as follows:

$$CP_\tau(TS)^\alpha = \{t_i, RD_{TS,\tau}(t_i) \leq \alpha\}$$

According to this, we can define the *Recursion depth based hierarchical segmentation* of a series as follows:

Definition 2.3: Let $TS = \{< t_1, v_1 >, ..., < t_m, v_m >\}$ be a time series and d be the maximum depth reached by the IEPF algorithm when executed on TS with threshold 0. The recursion based hierarchical segmentation of TS is a hierarchy of segments with levels L_0, L_1 .. L_d, where L_i is composed by the segments defined by each (ordered) pair of consecutive points of $CP_0(TS)^i$.

2) Threshold based hierarchical segmentation: As we have discussed before, the IEPF algorithm uses a threshold to determine when to stop in the approximation process. The lower the threshold, the *more precise* the approximation.

Let $CP_\tau(TS)$ be the subset of points obtained by the IEPF algorithm when executed on the series TS with threshold τ[1]. It can be easily proved that,

$$CP_{\tau_1}(TS) \subseteq CP_{\tau_2}(TS), \text{iff } \tau_2 \leq \tau_1$$

.

That is, as we progressively decrease the threshold used to execute the IEPF algorithm on the series, we obtain more points to approximate the series.

The values of thresholds that introduce *new* points in the approximation are of special interest for the segmentation procedure. We call *relevant thresholds* to these thresholds, according to the following definition:

Definition 2.4: We say that a threshold τ_i is a relevant threshold, iff:

$$\forall \tau_j > \tau_i, |CP_{\tau_j}| < |CP_{\tau_i}|$$

According to this, we can finally define the *Threshold based hierarchical segmentation* as follows:

Definition 2.5: Let $TS = \{< t_1, v_1 >, ..., < t_m, v_m >\}$ be a time series and $RT_{TS} = \{\tau_1, ..., \tau_h\}$ the decreasing ordered list of relevant thresholds of the series. The threshold based hierarchical segmentation of TS is a hierarchy of segments with levels L_0, L_1 .. L_h, where $L_0 = \overleftrightarrow{t_1 t_m}$, and $L_i (i > 0)$ is composed by the segments defined by each (ordered) pair of points of $CP_{\tau_i}(TS)$.

[1]Without loss of generality, we will use the time component t_i to refer to the point $< t_i, v_i >$ of the series

B. An example

Figure 3 depicts an example series with 100 points that we are going to use in order to illustrate the performance of the proposed method.

Figure 4 shows the results obtained when applying the Recursion depth based hierarchical segmentation.

Alternatively, figure 5 shows the results obtained when applying the Threshold based hierarchical segmentation. In order to work with threshold values that are independent from the value domain of the time series, we consider a normalized version of TS with all values in $[0, 1]$.

For the sake of space, figure 5 only depicts 12 selected levels from a total number of 65 levels corresponding to the relevant thresholds for this example. Concretely, levels 0, 1, 2, 3, 4, 5, 6, 7, 10, 20, 25, and 64.

C. Discussion

As can be seen, the first approach produces a more compact hierarchy with less and more dense (in terms of number of segments) levels. On the contrary, with the second one we obtain hierarchies with more levels and with small variations between consecutive levels.

While the second one produces a sequence of segmentations that perform more and more precise approximations of the original series in terms of distance, the first one could be considered as a compact representation of the relevant segments of the series we can go through in order to find suitable segmentations.

As mentioned in the introduction, the objective of the hierarchical segmentation is to provide information to the mining algorithms so that they can search for the best segmentation according to a specific criterion which is significant for the concrete mining task. In this sense, the second approach is appropriate for algorithms that consider as potential segmentations only those that fully correspond to a certain level in the hierarchy. In contrast, the first one is more convenient for algorithms that try to obtain a collection of segments which is relevant for their mining purpose regardless of the level they may appear. In this case, the hierarchy is used as a compact representation of the different segments that can be considered in the series, where segments are organized in terms of refinement through the parent-child relationship.

An example of the last kind of algorithms, for which the first approach is convenient, is described in the next section.

III. Linguistic Description of Trends in Time Series

Once we have presented our approach for the hierarchical segmentation of the series, we illustrate its usefulness for the description of time series in terms of trends, based on linguistic variables and protoforms.

A. Time series and linguistic description

Linguistic description of time series is a data mining task that attempts to provide the final user with novel, non-trivial, and potentially useful knowledge about data organized in a time series [2], [1]. Let $TS = \{< t_1, v_1 >, ..., < t_m, v_m >\}$

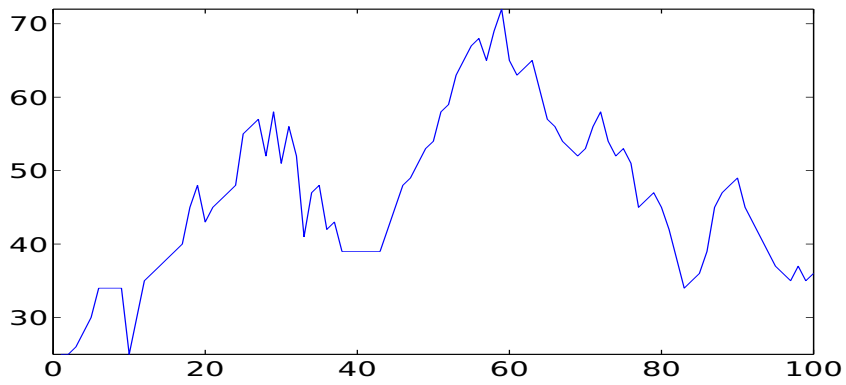

Fig. 3. Example series

be a time series, where every v_i is a value of the basic domain of the variable V and the time dimension is described in its finest grained level of granularity by members $T = \{t_1, ..., t_m\}$. In our approach [2], [1], a linguistic description is a collection of sentences in natural language, each one describing some relevant feature of a given time period. Both features and time periods are assumed to be fuzzy in general, particularly because linguistic terms use to correspond to fuzzy subsets of some reference set. For instance, a time period in a year described linguistically as *Cold weather* corresponds to a fuzzy subset of days. In the same way, the feature described by *High values* corresponds to a fuzzy subset of the domain of V, etc.

In order to have sentences whose accomplishment with the data can be calculated, we have employed different kinds of protoforms following the Computational Theory of Perceptions [8]. In our previous works, we have mainly focused on the value of the series, using protoforms based on quantified sentences of the form *"Q of D are A"* where *Q* is a linguistic quantifier, *D* is a time period and *A* is a description of the value of the series in *D*. An example is *Most of days with Cold weather, the series has High values*. The whole description is a collection of quantified sentences like this, where each sentence describes a *segment* of the series.

The process of determining a suitable description is very complex. There are many different time periods, many possible protoforms, and consequently the space of possible linguistic descriptions (sets of protoforms) is huge. In order to determine a suitable description, we have proposed a quality framework [4] and several algorithms [2], [1], [3].

B. Trends

In order to describe trends, we use the protoform *"In D, the trend is A (B)"* where *D* is a time period, *A* is a description of the trend and *B* is a number that expresses the variation observed in the series. For example, *A* can take values from the linguistic variable shown in Figure 6, defined on the slope domain.

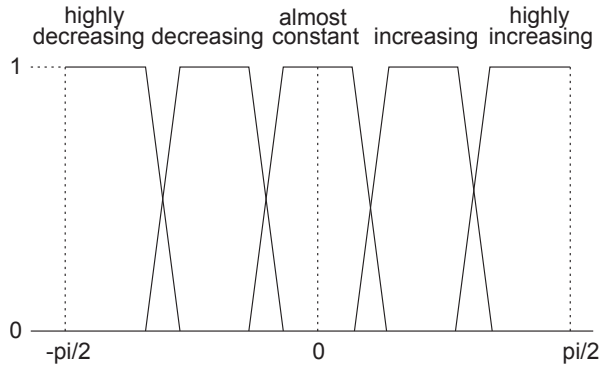

Fig. 6. Linguistic variable for trends description.

For the sake of brevity [4], the set of time periods of the sentences in the description must be a segmentation of the series, having a single sentence per time period. Ideally, this segmentation should be defined by time points where the trend changes, satisfying that the segments are a good approximation of the series and using the lesser possible amount of segments.

C. Using Hierarchical Segmentation for the Description of Trends

We propose a greedy algorithm similar to those proposed in our previous works. The objective is to obtain a summary covering the whole time domain using the minimum possible number of *good* sentences. A sentence "In D, the trend is A (B)' is considered to be *good* when in the interval D, the Mean Square Error (MSE) between the values of TS and the segment defined by the extreme points of D in TS is less or equal than a user-defined threshold γ, and A is the label with best compatibility with the slope in the considered segment (i.e. the one to which the slope pertains with the highest membership degree). In practice, as we have previously mentioned, and in order to work with values of γ that are independent from the value domain of the time series,

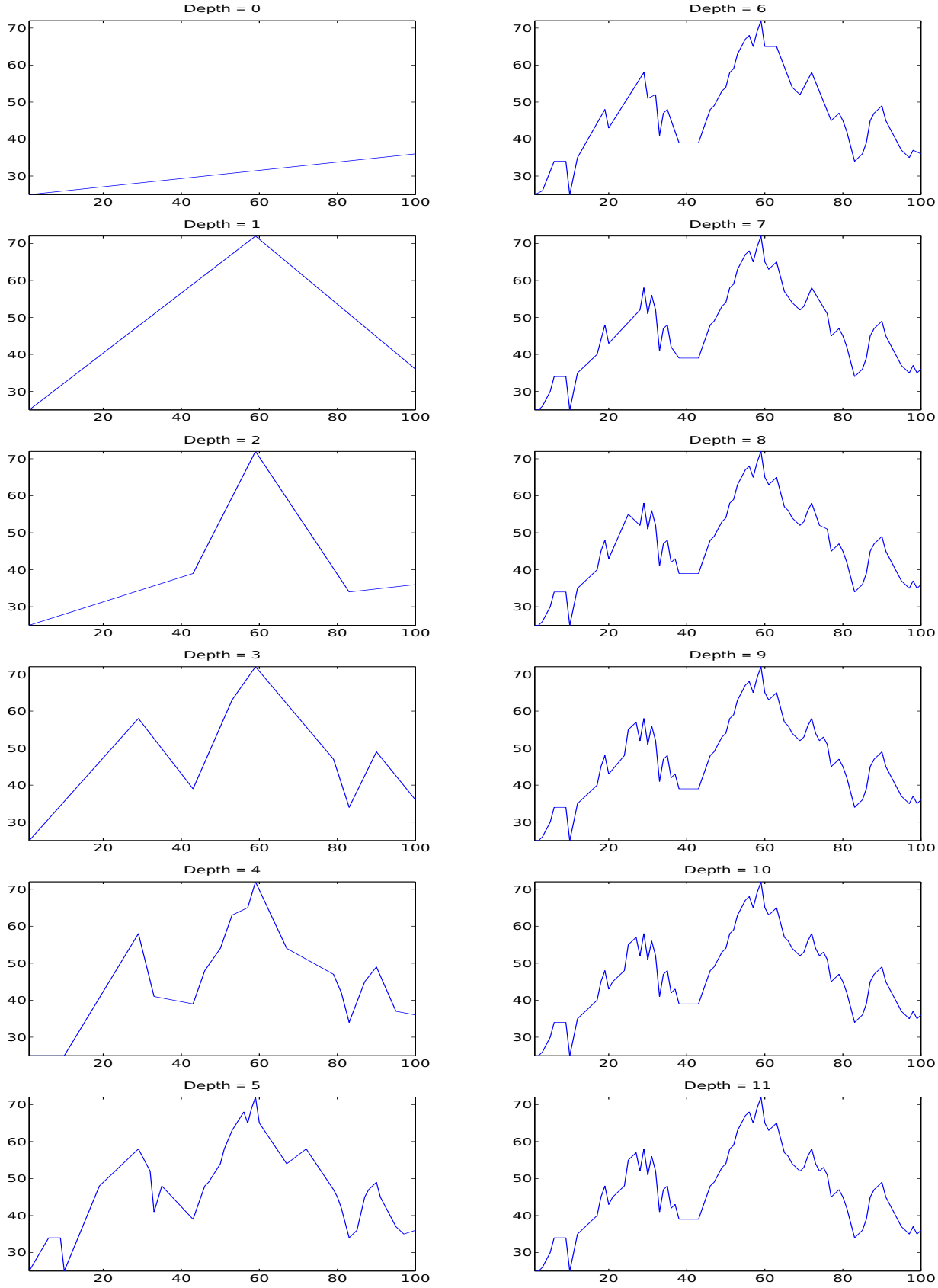

Fig. 4. Recursion depth based hierarchical segmentation

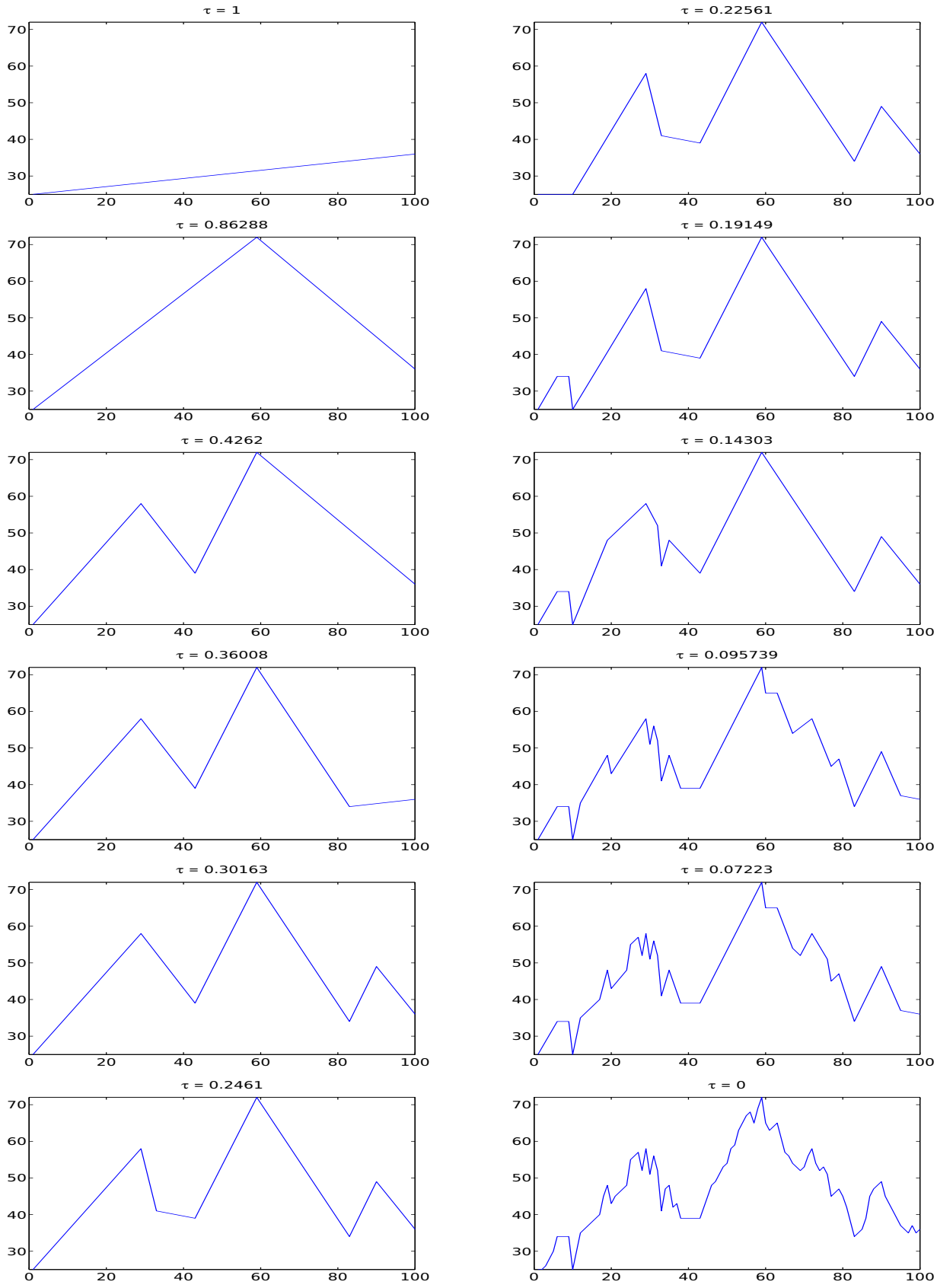

Fig. 5. Threshold based hierarchical segmentation

we will calculate the MSE as a value in $[0,1]$ by considering a normalized version of TS with all values in $[0,1]$, noted TS^*. Hence we shall consider $\gamma \in [0,1]$. Notice that TS^* will be used for calculating the MSE only, whilst for the description of slopes and approximate difference between points we will use the original time series TS.

Let $D_{i,j}$ be a time interval delimited by the change points $m_{i,j}, M_{i,j}$ with $m_{i,j} < M_{i,j}$ and let $vm_{i,j}$ and $vM_{i,j}$ be the corresponding values of TS at those time instants. Let $segment(TS, D_{i,j})$ be the segment between points $< m_{i,j}, vm_{i,j} >$ and $< M_{i,j}, vM_{i,j} >$. Let $slope(TS, D_{i,j})$ be the slope of $segment(TS, D_{i,j})$. Let TS^* be the normalization of time series TS so that all values are in $[0,1]$, and let $vm^*_{i,j}$ and $vM^*_{i,j}$ be the values of the time series TS^* at time instants $m_{i,j}$ and $M_{i,j}$, respectively. Let $MSE(TS^*, segment(TS^*, D_{i,j}))$ be the Mean Square Error between values of the time series TS^* and values of $segment(TS^*, D_{i,j})$. Algorithm 2 is our greedy algorithm for trend-based linguistic description of time series.

Algorithm 2 Algorithm to obtain trend-based linguistic description of time series.

Input

A time series TS
A hierarchical partition of the time dimension D with n levels.
A linguistic variable E for trend slope.
A threshold γ as maximum error.

Output

A Summary of TS comprised of a set of sentences "In D, the trend is A (B)".

Algorithm

1: $ToSummarize \leftarrow L_n$;
2: $Summary \leftarrow \emptyset$; $Summarized \leftarrow \emptyset$;
3: **while** $ToSummarize \neq \emptyset$ **do**
4: Take $D_{i,j} \in ToSummarize$
5: $ToSummarize \leftarrow ToSummarize \backslash \{D_{i,j}\}$;
6: $covered \leftarrow false$;
7: **if** $MSE(TS^*, segment(TS^*, D_{i,j})) \leq \gamma$ **then**
8: Let $A \leftarrow argmax *_{S \in E} S(slope(TS, D_{i,j}))$;
9: $Summary \leftarrow Summary \cup$ {In $D_{i,j}$, the trend is A ($|vm_{i,j} - vM_{i,j}|$) };
10: $Summarized \leftarrow Summarized \cup (D_{i,j})$;
11: $covered \leftarrow true$;
12: **end if**
13: **if** *not covered* and $i > 1$ **then**
14: $ToSummarize \leftarrow ToSummarize \cup ch(D_{i,j})$;
15: **end if**
16: **end while**

In order to look for *brevity*, we start from the time periods at the top level of the Recursion depth based hierarchical segmentation obtained by following the procedure described in a previous section. The set $ToSummarize$ is the collection of time periods for which a description has not been provided yet. The algorithm takes time periods from $ToSummarize$ until it is empty. If it is possible to obtain a Mean Square Error lesser than or equal to γ for a taken period, the procedure obtains a sentence for that period, which is added to $Summary$. Otherwise, the algorithm add to $ToSummarize$ the *corresponding children* in the next level of the time domain (lines 13-14). For a period $D_{i,j}$, the corresponding set of children $ch(D_{i,j})$ is defined as follows: $ch(D_{1,j}) = \emptyset$ for all j, otherwise $ch(D_{i,j}) = \{D_{i-1,k}, k \in \{1..p_{i-1}\} | D_{i-1,k} \cap D_{i,j} \neq \emptyset$ and $\neg \exists D \in ToSummarize \cup Summarized, (D_{i-1,k} \cap D_{i,j}) \subseteq D\}$. The final set of linguistically quantified sentences comprising the summary is $Summary$.

D. Example

Figure 7 shows the result obtained when applying this method to our example time series with Algorithm 2 executed with $\gamma = 0,0275$. As can be seen, the obtained segmentation does not correspond to any of the levels of neither the Recursion depth nor the Threshold based hierarchical segmentations, as it has been obtained by a mining algorithm guided by a different criterion (MSE).

With this segmentation, the linguistic description offered to the user is the following one:

- In period from 1 to 10, the trend is almost constant (0 units).
- In period from 10 to 29, the trend is highly increasing (33 units).
- In period from 29 to 33 the trend is highly decreasing (17 units).
- In period from 33 to 43, the trend is almost constant (2 units).
- In period from 43 to 59, the trend is highly increasing (33 units).
- In period from 59 to 100, the trend is decreasing (36 units).

IV. Conclusions

In this work, we have presented two hierarchical segmentation methods of time series, both based on the Iterative End-Point Fit algorithm. The first one is based on the recursion depth every point is first considered in. The second one is based on the distance threshold used to execute the algorithm. The objective of the hierarchical segmentation is to provide information to the mining algorithms so that they can search for the best segmentation according to a specific criterion which is significant for the concrete mining task. Our proposals constitute two alternatives approaches to be used according to the mining intention: the first one is more convenient when the intention is to build segmentations by taking relevant segments that may appear in different levels, whilst the second is suitable for algorithms that intend to consider as potential segmentations only levels in the hierarchy. As an application of the hierarchical segmentation in a mining task, we have also shown how it can be used in the trend-based linguistic description of time series.

References

[1] R. Castillo-Ortega, N. Marín, and D. Sánchez. A fuzzy approach to the linguisitic summarization of time series. *Special Topic Issue on Soft Computing Techniques in Data Mining in Journal of Multiple-Valued Logic and Soft Computing (JMVLSC)*, 17(2,3):157–182, 2011.

[2] R. Castillo-Ortega, N. Marín, and D. Sánchez. Linguistic query answering on data cubes with time dimension. *Special Topic Issue on Advances in Fuzzy Querying: Theory and Applications in International Journal of Intelligent Systems (IJIS)*, 26(10):1002–1021, 2011.

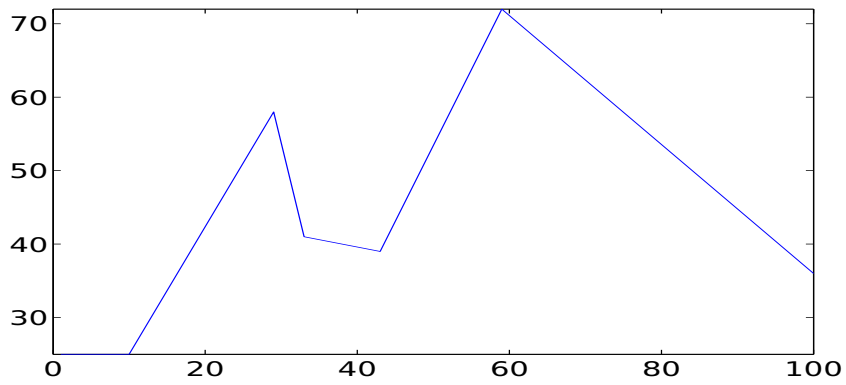

Fig. 7. Resulting segmentation for the linguistic description.

[3] R. Castillo-Ortega, N. Marín, D. Sánchez, and A.G.B. Tettamanzi. A multi-objective memetic algorithm for the linguistic summarization of time series. In *GECCO, Genetic and Evolutionary Computation Conference 2011*, pages 171–172, 2011.

[4] R. Castillo-Ortega, N. Marín, D. Sánchez, and A.G.B. Tettamanzi. Quality assessment in linguistic summaries of data. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based System. IPMU 2012*, pages 285–294, 2012.

[5] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, October 1973.

[6] E. Keogh, S. Chu, D. Hart, and M. Pazzani. *Data Mining in Time Series Databases*, chapter Segmenting Time Series: A Survey and Novel Approach, pages 1–22. World Scientific Publishing, 2004.

[7] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244 – 256, 1972.

[8] L. A. Zadeh. Computing with words and perceptions - a paradigm shift. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2010, Las Vegas, Nevada, USA, July 12-15, 2010, 2 Volumes*, pages 3–5, 2010.