ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Segmented Shape-Symbolic Time Series Representation

Herbert Teun Kruitbosch[1], Ioannis Giotis[2] and Michael Biehl[1]

1- University of Groningen - Johann Bernoulli Institute
Nijenborgh 9, 9747 AG Groningen - The Netherlands

2- Target Holding B.V., Nettelbosje 1, 9747 AJ Groningen - The Netherlands

Abstract. This paper introduces a symbolic time series representation using monotonic sub-sequences and bottom up segmentation. The representation minimizes the square error between the segments and their monotonic approximations. The representation can robustly classify the direction of a segment and is scale invariant with respect to the time and value dimensions. This paper describes two experiments. The first shows how accurately the monotonic functions are able to discriminate between different segments. The second tests how well the segmentation technique recognizes segments and classifies them with correct symbols. Finally this paper illustrates the new representation on real-world data.

1 Introduction

A time series is a sequence of measurements taken at successive moments in time and often contain a large number, N, of measurements, $T = t_1, \ldots, t_N$ with $t_i \in \mathbb{R}$, possibly too many to store in memory or do calculations. Two important *reduced* time series representations are PLR [1] and SAX [2]. PLR segments a time series into pieces of possibly different sizes and approximates them with linear curves. Other higher order polynomial representations are SwiftSeg [3] and the shape space representation [4]. These representations have been used by various researchers to support clustering [1, 2, 5], classification [2, 5], association rule mining [2, 6], query by content [2] and anomaly detection [1, 7, 8] in time series data. SAX splits a time series into segments of an equal, predefined size and assigns a symbol to each piece, depending on the range in which the mean value of the segment lies. The SAX representation is not scale-invariant with respect to time, does not store information about the shape of a segment and requires a parameter to state the size of an individual segment. The value of this parameter can prevent us from detecting small or big patterns.

This paper introduces a piece-wise representation which tries to find monotonic pieces, which may differ in length. Each piece is represented by a symbol, indicating direction and curvature. Hence our representation is symbolic like SAX, but the symbols also capture shape and direction and have variant size pieces like PLR. Section 2 gives a brief overview of the used techniques, section 3 discusses the method, section 4 tests our representation on synthetic data and data from the IJkdijk project[9]. Finally, section 5 discusses the conclusions and directions for future work.

2 Background and related work

Our representation uses least square estimations (LSE) to approximate and classify the direction and shape of a segment. This section will discuss linear LSE and a relevant Bayesian consideration regarding classification. Next we discuss bottom up segmentation. Finally, we explain the GAP statistic [10] to determine the number of segments.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Linear least squares estimation In order to approximate and assign a symbol to a segment, we use LSE. We assume a segment $\mathbf{s} = s_1, \ldots, s_N$, such that $s_n = \sum_{m=1}^{M} \theta_m f_m(n) + w_n$, where $f_1, \ldots, f_M$ is a set of basis functions and $w_n \sim \mathcal{N}(0, \sigma^2)$ is white Gaussian noise, independent and identically distributed (i.i.d.). A LSE determines the unknown parameters $\theta_1, \ldots, \theta_M$, such that the *error* between s_n and $\sum_{m=1}^{M} \theta_m f_m(n)$ is minimal:

$$\hat{\theta}_{LSE} = \mathsf{argmin}_\theta \Sigma_{n=1}^{N} \left(s_n - \Sigma_{m=1}^{M} \theta_m f_m(n)\right)^2. \quad (1)$$

We only consider linear estimations of the form $\alpha + \theta f(x)$ (based on $M = 1$ monotonically increasing basis function $f = f_1$ and a constant), which is monotonic for any α and θ. We set f to have zero mean: $\sum_{n=1}^{N} f(x_n) = \langle f(\mathbf{x}), \mathbf{1} \rangle = 0$, such that the basis $f, \mathbf{1}$ is orthogonal, allowing for LSE in linear time.

Bayesian classification In order to classify a segment $\mathbf{s}$, i.e. $cl : Segment \mapsto f_1, \ldots, f_M$, we need to find the model with the highest posterior probability $P(f_i|\mathbf{s}) = P(\mathbf{s}|\alpha + \theta f_i(x))P(\alpha + \theta f_i(x))/P(\mathbf{s})$. Hence we could classify by maximizing the posterior probability: $cl(\mathbf{s}) = \mathsf{argmax}_{\alpha+\theta f_i} P(\mathbf{s}|\alpha + \theta f_i(x))P(\alpha + \theta f_i(x))$. This expression classifies, whereas (1) fits, hence the meaning of i in f_i is different. The prior probability $P(\alpha + \theta f(x))$ is hard to determine, since $P(\alpha, \theta|f_i)$ and $P(f_i)$ are often unknown. Therefore, our classification model is simplified to maximum likelihood classification:

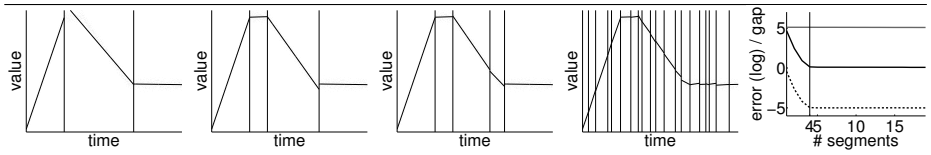
$$cl(\mathbf{s}) = \mathsf{argmax}_{\alpha+\theta f_i} P(\mathbf{s}|\alpha + \theta f_i(x)) = \mathsf{argmax}_{\alpha+\theta f_i} - \Sigma_{n=1}^{N} (\alpha + \theta f_i(x_n) - s_n)^2. \quad (2)$$

Bottom up Segmentation Bottom up segmentation [1] is one of many segmentation techniques, like sliding window and top down segmentation. None of them consider all possible segmentations and hence have to deal with avoiding local optima. In general bottom up outperforms other methods [11] and can use of the GAP statistic to determine the number of segments. The bottom up approach starts with a fine grained segmentation, with small equal sized segments. Based on the introduced error, adjacent segments are merged greedily and iteratively until some condition is met (Figure 2).

A local optimum occurs when merging 3 adjacent segments is beneficial, but merging any two of them is expensive. Assuming that merging two same-direction monotonic segments is never expensive and that merging two different-direction monotonic segments always is, such a local optimum will not occur.

GAP statistic We use the GAP statistic [10] to determine the optimal number of segments. However, we use a different error function than the original GAP statistic, since we are segmenting instead of clustering:

$$\varepsilon_{segment}(\mathbf{s}) = \|\mathbf{s} - fit(\mathbf{s})\|_2^2, \;\; \varepsilon_{segmentation}(S) = \Sigma_{\mathbf{s}\in S} \varepsilon_{segment}(\mathbf{s}),$$



(a) $k = 3$ (b) $k = 4$ (c) $k = 5$ (d) $k = 20$ (e) GAP

Legend of (e): — $\varepsilon_{segmentation}$ of $\mathbf{t}$, — $\varepsilon_{segmentation}$ of $\mathbf{u}$, -- $gap(k|\mathbf{t})$

Fig 1. Bottom up segmentation of a 4-piece piece-wise linear time series $\mathbf{t}$ into k segments. The right-most figure shows the GAP statistic as discussed in section 2.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

where fit approximates a segment $\mathbf{s}$ and S is the set of segments. The GAP statistic is based on the error of a segmentation of a time series, and the expected error of a uniform distribution in the same range. Therefore we define a reference signal $\mathbf{u} = u_1, \ldots, u_N$ where $u_i \sim \mathscr{U}(0,1)$: uniformly randomly distributed between 0 and 1. We also define a time series $\mathbf{t} = t_1, \ldots, t_N$ with values normalized between 0 and 1. The optimal number of segments is determined from:

$$gap(k|\mathbf{t}) = \log\left(\varepsilon_{segmentation}(S_{\mathbf{t},k})\right) - \log\left(\varepsilon_{segmentation}(S_{\mathbf{u},k})\right),$$

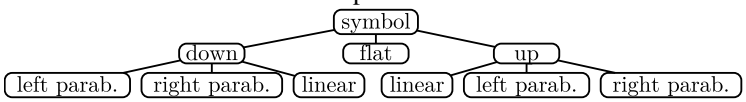
where $S_{\mathbf{t},k}$ and $S_{\mathbf{u},k}$ respectively are segmentations of $\mathbf{t}$ and $\mathbf{u}$ of k segments. We estimated $\varepsilon_{segmentation}(S_{U,k})$ by taking the mean of $Z = 10$ instances of $\mathbf{u}$, $\hat{\mathbf{u}} = \hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_Z$, and corrected it with the standard deviation.

The *elbow* of $gap(k|\mathbf{t})$ (Fig. 2e) is the point where adding more segments will only result in fitting to noise. This elbow may not be located at $\mathsf{argmin}_k gap(k|\mathbf{y})$, due to fitting to noise or numerical instability. We selected the k_{elbow} such that the second order derivative of $gap(k|\mathbf{t}) - gap(k+1|\mathbf{t})$ is maximal.

3 Method

Our representation is created by first segmenting a time series using an approximation as described in algorithm 1. The GAP statistic determines the amount of segments and each final segment is classified according to algorithm 1. The result is a symbolic string which can then be used by various machine learning algorithms.

Assigning Shapes to Segments To detect non-monotonic structures in time series, this paper focuses on finding monotonic segments. This way a concatenation of monotonic segments defines a more general non-monotonic structure. We classify a segment as one of four shapes, shown in figure 2. Except for the first, each shape is monotonically increasing. However, we also classify whether the fit of such a shape goes up or down. This results in a classification tree for shapes:



Detecting flat segments In order to classify the flat shape of figure 2, we could fit a segment $f_0(x) = 0$. However, for $\theta_i = 0$, each of f_1, f_2, f_3 will have the same likelihood as f_0. This disallows us to distinguish between f_0 and any of f_1, f_2, f_3. We could solve this by adding a prior, but like stated in section 2 this can not always be determined.

Algorithm 1 Classification and approximation of a segmented time series

Require: Segment $S = (s_1, \ldots, s_N)$ with mean 0 and std dev 1, Threshold T (Sec 3)
Ensure: Classification $C \in \{f_0, f_1, f_2, f_3\} \times \{none, up, down\}$ and approximation of S.
Set $x_1, \ldots, x_N$ equidistant on $[0,1]$
for $i = 1,2,3$ **set** $\theta_i = \Sigma_{n=1}^{N} f_i(x_n) \cdot s_n \;/\; \Sigma_{n=1}^{N} f_i^2(x_n)$ ▷ Linear LSE (for f_i see Fig. 2)
$i = \mathsf{argmax}_i = \sum_{n=1}^{N} (\theta_i f_i(x_n) - s_N)^2$ ▷ Likelihood based classification
if $|\theta_i| < T$ **then** $i = 0$; $dir = none$; $\theta_0 = 0$; $f_0(x) = 0$ ▷ Flat segment
else if $\theta_i < 0$ **then** $dir = down$ **else** $dir = up$ ▷ Down / Up
$C = (f_i, dir)$ ▷ Symbol of the segment
Set $A = (a_1, \ldots, a_N)$ such that $a_n = \theta_i f_i(x_n)$ ▷ Approximation of the segment

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

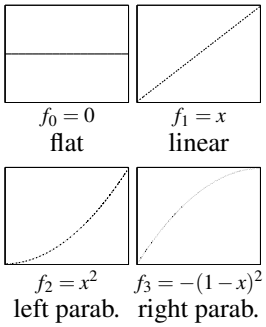
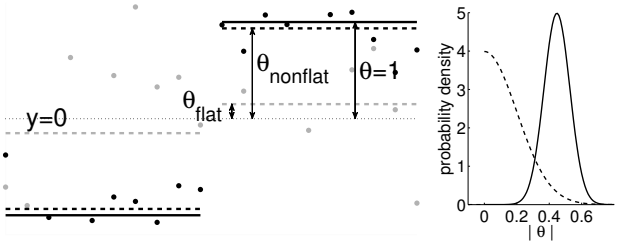


Fig. 2: Five monotonic shapes, all are defined on the domain $[0,1]$.



(a) Fit (b) $|\theta|$ distribution

Legend of (a): • / • data, — f, -- / -- fit of f, ····· $y = 0$
Legend of (b): — flat shape, -- non-flat

Fig 3. In (a) the fit of f on the data of a flat model $\overline{s}_n = \overline{w}_n$ (gray) and a non-flat model $\tilde{s}_n = \theta f(x) + \tilde{w}_n$ (black) are shown. See Sec 3. In (c) the probability distribution of θ for both models for $\theta = 1$, $\sigma = 1$ and $N = 5$ is shown.

Therefore, we introduce a heuristic based on the signal to noise ratio to classify flat segments. We consider a noisy, flat segment $\overline{s} = \overline{s}_1, \ldots, \overline{s}_N$, $\overline{s}_n = \overline{w}_n$ ($\overline{w}_n \sim \mathcal{N}(0, \sigma^2)$), shown in figure 3a, for which we want to determine whether it is flat or has some non-flat f shape. We consider a simple case where f has the first half of the points equal to -1 and the second half equal to 1, hence $Var(f) = 1$. Then we normalize $\overline{s}$ to have variance 1. Due to f's shape, estimation of θ will result in the average of the right half's average and the left half's negative average. Therefore the distribution of the LSE $|\overline{\theta}|$ for f depends on the distribution of the average value of $\overline{s}$: $P(|\overline{\theta}|) = 2\mathcal{N}(|\overline{\theta}|; 0, 1/N)$.

Now we consider a noisy, non-flat segment $\tilde{s}_n = \theta f(x_n) + \tilde{w}_n$ ($\tilde{w}_n \sim \mathcal{N}(0, \sigma^2)$) (Fig. 3a). We have that $Var(\tilde{s}_n) = \theta^2 + \sigma^2$. Hence if we normalize $\tilde{s}$ to variance 1, we get $\tilde{w}_n \sim \mathcal{N}(0, \sigma^2/\sqrt{\theta^2+\sigma^2})$, and the probability of a LSE $|\tilde{\theta}|$ for f is:

$$P(|\tilde{\theta}|) \approx \mathcal{N}(|\tilde{\theta}|; \theta/\sqrt{\theta^2+\sigma^2}, \sigma^2/(N\sqrt{\theta^2+\sigma^2})).$$

Notice that $\theta/\sqrt{\theta^2+\sigma^2}$ is a normalized θ. The variance of θ for the flat shape and the f shape depends on the number of samples by a factor $1/N$ and on the signal to noise ratio $\sigma^2/\sqrt{\theta^2+\sigma^2}$ in case of a non-flat signal. Hence more noise moves the means closer to each other; decreasing the discriminative power of $|\theta|$.

Figure 3b shows an example of both distributions and suggests a threshold $|\theta| \approx 0.26$, such that $|\theta| < 0.26$ would classify as flat and $|\theta| \geq 0.26$ as not flat. This threshold T is used in 1. We assumed a block-shaped f. Assuming other shapes of f makes it harder to determine the distribution of $|\hat{\theta}|$, because the variance $Var(f(x) + w_x)$ is not necessarily $Var(f(x)) + Var(w_x)$, the idea is however similar.

Summary In contrast with PLR, the introduced method is symbolic and allows for processing using discrete algorithms and data structures. In contrast to SAX, the proposed representation uses segmentation and can identify two similar segments with a different number of samples as the same, allowing for scale invariance. Whereas SAX symbols are based on mean values, the introduced symbols are based on shapes. In summary, the introduced method is based on a symbolic representation like SAX but retains the scale invariance of the shape based PLR.

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

4 Results

The accuracy of our symbolic representation is quantified using two measures. These two measures are used to first determine how often the classification of a synthetic segment matches the label in section 4; this tests the accuracy of algorithm 1 and illustrates how well the classifier is able to discriminate between shapes. Section 4 illustrates the accuracy of our representation including segmentation. Finally an example of real world data from the IJkdijk project is represented using our method.

The two accuracy measures are based on the *symbol tree* of section 3. The first measure quantifies the fraction of segments or samples for which the direction of the shape is correctly classified: the direction accuracy, the second quantifies the fraction for which both the direction and the shape are correct: the shape accuracy.

Classification accuracy To measure the discriminating ability of our classification described by algorithm 1, we created 100 synthetic shapes with variance 1 for each direction of the 4 template functions (Figure 2). Then we tested the accuracy as the fraction of correctly classified segments, with respect to the amount of i.i.d. white Gaussian noise, the number of samples in a segment and both measures. The results are shown in figure 4. The direction accuracy stays above 80% up to $\sigma = 0.7$ and the direction accuracy up to $\sigma = 0.4$. Note that the curvature gets lost with more noise, but the direction does not. For $\sigma \geq 1$ both accuracies drop as all segments are classified as flat.

Segmentation and classification accuracy To further evaluate the accuracy and robustness of the proposed technique we measured the fraction of correctly classified samples of a segmented time series. We created 100 time series of 1001 samples by concatenating 4 monotonic segments of random size with at least 50 samples based on the shapes in figure 2. The time series have variance 1. We label each sample as the shape it was created from. The segments are concatenated such that there are no gaps in the y direction. Finally the synthetic time series is segmented and each sample is classified by classifying the encapsulating segment using algorithm 1. The fraction of matching classified and labelled samples is shown in figure 5. The direction accuracy is robust with respect to noise; it remains above 0.8 for up to noise $\sigma = 1$, whereas the shape accuracy scores lower, showing that shape information is lost when noise is added.

IJkdijk data example Finally, some subsequences from the LiveDijk Eemshaven[9] measurements were taken and translated into symbolic strings. The results are shown in figure 6. The plots show that indeed monotonic structures can be found, the first and third figure have smaller patterns and the middle figure has larger ones. This illustrates the scale invariance of our representation. This behaviour, i.e. finding small or

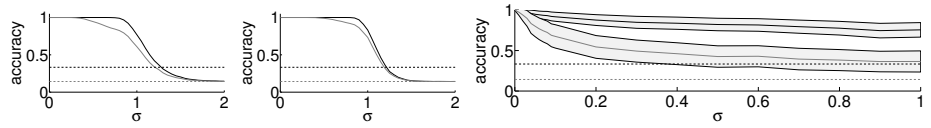


Fig. 4: Accuracy of identifying the segment of one synthetic shape for respectively 501 and 1001 samples. The MC classification accuracy is of a classifier which randomly assigns a direction and shape.

Fig. 5: Segmentation and segment classification accuracy as described in section 4. The gray area is half of the standard deviation of the simulation. In this case $T = 0.26$ and 500 simulations were run.

Accuracy legend: — direction, — shape, -- MC direction, -- MC shape

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7.
Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

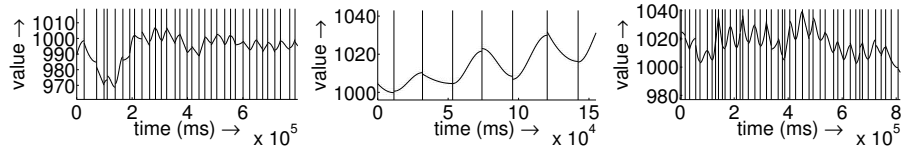


Fig. 6: All three figures show a segmentation of sub-sequences taken from measurements on the LiveDijk at different locations. The flat-threshold was set to $T = 0.26$.

large patterns is influenced by the estimation of the number of segments. Changing the trade-off between the amount of segments and the amount of error will change how the representation deals with smaller, less defined monotonic segments.

5 Conclusion and future work

We introduced a scale-invariant, symbolic representation of a time series, based on monotonic segments of possibly different sizes, which captures direction and curvature. The representation enables machine learning algorithms to efficiently find anomalies or structures in a time series based on the shape of monotonic sub-sequences. However, selecting the number of segments is still an ill-defined problem. Future work could define a hybrid representation, representing both smaller and larger, encapsulating segments, for example by finding a tree instead of a sequence of segments. Clustering patterns generated by our representation on the IJkdijk data may give insight in the general behaviour and hence help define and determine anomalous sub-sequences.

References

[1] E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Data Mining, 2001. ICDM 2001, Proc. IEEE International Conf. on*, pages 289–296. IEEE, 2001.

[2] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.

[3] E. Fuchs, T. Gruber, J. Nitschke, and B. Sick. Online segmentation of time series based on polynomial least-squares approximations. *Patt. Analysis and Machine Intell., IEEE Trans.*, 32:2232–2245, 2010.

[4] E. Fuchs, T. Gruber, H. Pree, and B. Sick. Temporal data mining using shape space representations of time series. *Neurocomputing*, 74(1):379–393, 2010.

[5] E.J. Keogh and M.J. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *KDD*, volume 98, pages 239–243, 1998.

[6] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining*, pages 37–44. Citeseer, 2000.

[7] C.-S. Perng, H. Wang, S.R. Zhang, and D.S. Parker. Landmarks: a new model for similarity-based pattern querying in time series databases. In *Data Engineering, 2000. Proc. 16th Intl. Conf. on*, pages 33–42. IEEE, 2000.

[8] C. Wang and X Sean Wang. Supporting content-based searches on time series via approximation. In *Scientific and Statistical Database Management, Proc. 12th Intl. Conf. on*, pages 69–81. IEEE, 2000.

[9] IJkdijk website http://www.ijkdijk.nl/en/ijkdijk.

[10] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *J. of the Royal Statistical Society: Series B*, 63(2):411–423, 2001.

[11] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–22, 2004.