JOURNAL OF APPLIED QUANTITATIVE METHODS

International Symposium on Stochastic Models in Reliability Engineering, Life Sciences and Operations Management (SMRLO'10)

ONLINE HOEFFDING BOUND ALGORITHM FOR SEGMENTING TIME SERIES STREAM DATA[1]

Dima ALBERG
PhD Candidate, Department of Information Systems Engineering,
Ben-Gurion University of the Negev,
Beer-Sheva, Israel

E-mail: alberg@bgu.ac.il

Avner BEN-YAIR
PhD, Center for Reliability and Risk Management,
SCE - Shamoon College of Engineering,
Beer-Sheva, Israel

E-mail: avner2740@gmail.com

Abstract: *In this paper we introduce the ISW (Interval Sliding Window) algorithm, which is applicable to numerical time series data streams and uses as input the combined Hoeffding bound confidence level parameter rather than the maximum error threshold. The proposed algorithm has two advantages: first, it allows performance comparisons across different time series data streams without changing the algorithm settings, and second, it does not require preprocessing the original time series data stream in order to determine heuristically the reasonable error value. The proposed algorithm was implemented in two modes: off line and online. Finally, an empirical evaluation was performed on two types of time series data: stationary (normally distributed data) and non stationary (financial data).*

Key words: *data stream; time series; linear approximation; segmentation; Hoeffding bound; SWAB (Sliding Window Bottom Up) algorithm; ISW (Interval Sliding Window) algorithm*

1. Introduction

Time series data streams are ubiquitous in finance, meteorology and engineering. They are an application area of growing importance in the data stream mining research. For example, sensors generate one million samples every three minutes [13], therefore one of the primary purposes of data stream research boils down to fast and reliable time series data streams segmentation or dimensionality reduction techniques. These techniques are used in many areas of data stream mining as: frequent patterns finding, structural changes and

concept drifts detection [4], time series classification and prediction [13], time series similarities searching [8], [15], etc. The main principle of segmentation algorithms concludes in reducing the time series dimensionality by dividing the time axis into intervals behaving approximately according to a simple model. A good time series data stream segmentation algorithm has on-line, fast, accurate and comparable with other algorithms structure. For example the Sliding Window algorithm [8] on the one hand is online, very fast and relatively simple for using in online segmentation applications but on the other hand, it sometimes gives poor accuracy and does not allow to perform online multivariate segmentation.

The segmentation problem can be defined in following way: first, given a time series data stream to produce the best representation such that the maximum error for any segment does not exceed some user specified confidence level error threshold. It is important to add, that using a combined relative parameter such as Hoeffding bound [5] confidence level will allow to evaluate an online multivariate segmentation and second, to construct a user friendly segmentation application which will evaluate and compare the proposed online segmentation algorithms in real time. As we shall see in later sections, the state-of-the-art segmentation algorithms do not meet all these requirements.

The rest of the paper is organized as follows. In Section 2, we provide a literature review of three state-of-the-art online piecewise linear segmentation algorithms. In Section 3, we provide a methodology for improving the existing state-of-the-art online segmentation algorithms. The proposed methodology based on Hoeffding bound error estimation, which uses a relative probability parameter instead of maximum error nominal parameter and allows performing online multivariate segmentation. Section 4 briefly demonstrates a real-time segmentation application. Finally, in Section 5 and 6 we provide brief and meaningful empirical comparison of the proposed algorithms and suggest final conclusions.

2. Related Studies

Several high level representations of time series have been proposed in the research literature, including Fourier Transforms [8], Wavelets [1], Symbolic Mappings [3], [17] and Piecewise Linear Approximation (PLA) [20], [1], [4], [6], [5], [7], [10], [11], [14], [16], [18], [19], [21]. In this work, our attention will confine to PLA, perhaps the most frequently used representation in continuous time series data streams. Obviously, all piecewise linear segmentation algorithms can also be classified as batch or online [20]. The problem discussed by Shatkay and Zdonnik, [17], Keogh et al., [9], Biffet and Kirkby [1] is actually how to build online, adaptive, fast and accurate algorithm for piecewise linear segmentation of time series data stream, because on the one hand, the main problem of online Sliding Window algorithm [2], [7] concerns in its poor accuracy [18], [21] and its inability to look ahead. On the other hand the offline accurate Bottom Up [9] algorithm is impractical or may even be unfeasible in a data mining context, where the data are in the order of terabytes or arrive in continuous streams. This problem is very important because for scalability purposes the proposed piecewise linear segmentation algorithm needs to capture the online nature of sliding windows and yet retain the superiority of Bottom Up algorithm.

Keogh et al. [9] introduced new online Sliding Window Bottom Up (SWAB) algorithm which scales linearly with the size of the dataset, requires only constant space, produces high quality approximations of the initial time series data, and can be seen as operating on a continuum between the two extremes of Sliding Windows and Bottom-Up. The authors have shown that the most popular Sliding Window approach generally produces

very poor results, and that while the second most popular approach, Top-Down, can produce reasonable results, it does not scale well with massive time series stream data. The main problem with the Sliding Windows algorithm is its inability to look ahead, lacking the global view of its offline (batch) counterparts. The Bottom-Up and the Top-Down [7] approaches produce better results, but are offline and require the scanning of the entire data set. For example, the SWAB [9] algorithm has three nominal input parameters, which need to be defined carefully by the user in order to obtain an accurate segmentation model. Often the user obstructs to determine for the value of the maximal error threshold, because the data has very noisy non-stationary behavior. Therefore, in aim to produce an accurate segmentation model, the user needs to perform the preprocessing of the obtained data or to perform a time consuming experiment design. Second, the inner loop of the SWAB algorithm simply invokes the Bottom-Up algorithm each time. This results in some computation redundancy and increases the computational complexity of algorithm. Second, the performance of the Sliding Window and SWAB algorithms depends on the value of maximal error. As maximal error goes to zero the Sliding Window and SWAB algorithms have the same performance, since they would produce multiple short segments with no error. At the opposite end, as the maximal error becomes very large, the algorithms once again will all have the same performance, since they will simply approximate a data stream with a single best-fit line.

However, most works along these research lines that we know of [1], [19] and [9] recommend to test the relative performance for some "reasonable value" of maximal error, a value that achieves a good tradeoff between compression and fidelity. Because this "reasonable value" is subjective and dependent on the data mining application and the data itself, they did the following. First, they chose a "reasonable value" of maximal error for each dataset and then bracketed it with 6 values separated by powers of two. The lowest of these values tends to produce an over-fragmented approximation, and the highest tends to produce a very coarse approximation. Second, they chose performance in the mid-range of the 6 values which by their opinion should be considered most important. Obviously, the maximal error calculation routine proposed by and [8], [9] and [15] are heuristic, requires multi pass computational efforts and have no rigorous guarantees of performance.

We therefore, introduce an improved algorithm combining the online nature of the sliding window algorithm and time series data stream, decreasing the number of input parameters and decreasing computational redundancy and complexity. We call the proposed algorithm *ISW (Interval Sliding Window) algorithm*.

3. Methodology

3.1. The ISW Segmentation Algorithm

The proposed ISW algorithm derives the maximal error by defining appropriate confidence level (e.g. 95%) and using Hoeffding bound one pass calculation. In fact, a similar approach was used in the VFDT decision tree induction algorithm introduced in Hulten and Domingos [13]. Suppose we have segment A with range R_A and n_A observations, and segment B with range R_B and n_B observations, which belong to a sliding window S of time series T. Assume that $\bar{x}_A$ and $\bar{x}_B$ are the sample means of segments A and B, respectively. The new merged segment AB has range R_{AB}, $(n_A + n_B)$

observations and sample mean $\bar{x}_{AB}$ equals $c_A \bar{x}_A + c_B \bar{x}_B$, where c_A and c_B equal to $\frac{n_A}{n_A + n_B}$ and $\frac{n_B}{n_A + n_B}$, respectively.

Proposition 1: The Hoeffding bound states that with confidence level of δ the true mean of the merged segment AB lies in the interval $\bar{x}_{AB} \pm \varepsilon_{AB}$ where:

$$\varepsilon_{AB} = R_{AB} \sqrt{\frac{\ln\left(\frac{1}{\delta}\right)}{2(n_A + n_B)}} \tag{1}$$

Proof of Proposition 1: Let $X_1, X_2, ..., X_n$ be independent random variables. Assume that each x_i is bounded, that is $P(X_i \in R = [a_i, b_i]) = 1$. Let $\bar{x}_{AB} = \frac{1}{n_A + n_B} \sum_{i=1}^{n_A + n_B} x_i$, with expected value $E(X_{AB})$. Then according to Hoeffding inequality theorem [5] for any $\varepsilon > 0$

$$P\left[\bar{x}_{AB} - E[X_{AB}] > \varepsilon\right] \le e^{-2 \cdot \frac{(n_A + n_B)^2 \varepsilon^2}{R_{AB}^2}} < \delta .$$

From this theorem we can derive absolute error ε_{AB} with confidence level of δ.

According to Motwani and Raghavan [12] the Hoeffding bound ε_{AB} represented in (1) is independent of the distribution generating the examples. This bound is applicable to all situations where observations are independent and generated by a stationary distribution. Important to note, that Hoeffding bound is addictive, its error is absolute and it does not require calculation of expected means of two merged segments. It is easy to show, that when the confidence level $\delta = 1$, the Hoeffding error value is equal to zero meaning that the observed and the segmented time series data streams are the same.

The DASWI algorithm works in the following way: each time a new observation arrives the algorithm calculates the Hoeffding bound using (1) and a user defined confidence level δ then in case the new calculated error is greater than the previously calculated Hoeffding bound, the algorithm starts a new sliding window, otherwise it continues with the current sliding window. This incremental technique on the one hand is more sensitive to data stream concept drift changes and on the other hand allows to create relatively large segments when the data stream is stable and therefore to decrease significantly the running time of the proposed algorithm. The pseudocode for the ISW algorithm is shown in Figure 1.

Input:	Data stream, DS Confidence Level, δ Sliding Window size, SW
Output:	ISW Segmented data stream.

```
anchor = 1;
While not finished segmenting time series
    i = 2;
--- Bound and Error Calculation
While Model_error < Hoeffding_Bound
   Model_Error(Segment[anchor: anchor + i ])
```

```
Hoeffding_Bound(Segment[anchor], δ)
 i ++;
 New_Segment = Create_Segment(T[anchor: anchor + (i-1)]
 Segment = Merge(Segment, New_Segment)
 anchor = anchor + i;
```

Figure 1. **The ISW algorithm pseudocode**

Example 1. The following numerical example briefly explains main calculation procedure of the ISW algorithm. Suppose that the current sliding window segment includes four observations: 1, 2, 3 and 4 (Table 1). Now, a new, fifth observation arrives and the ISW algorithm checks whether to start a new segment or to continue updating the previous one. The current segment range equals 0.8 (4.8-4.0), the number of observations is 4 and with the user specified confidence level of 95% the value of Hoeffding bound equals 0.06. Now, with the aid of the new, fifth observation we will recalculate the linear interpolation model error. The new model error equals 0.025 and it is lower than the calculated Hoeffding bound therefore the algorithm increases the current segment.

Table 1. Observations for ISW numerical example

n	1	2	3	4	5 (new)
Obs. Value	4.1	4	4.4	4.8	4.7
Segment	1	1	1	1	2

4. Experimental Results

In aim to compare accuracies of the proposed algorithms to the traditional algorithms SW [8] and SWAB [9] the following validation experiment was performed. First, three time series data streams used in [9] were selected (ECG, Space Shuttle and Radio Waves), after that the mean square error accuracies of algorithms SW and SWAB were evaluated with zero error threshold value. Second, on the basis of evaluated accuracies the appropriate confidence levels of the proposed algorithms were retrieved. Finally, Table 2 organizes the obtained results and clearly demonstrates that our proposed methods don't inferior to traditional SW and SWAB algorithms reported accuracies.

Table 2. Comparison between SW and SWAB algorithms

Dataset	**ECG**	**Space Shuttle**	**Radio Waves**
ISW	95%	90%	93%
ISWAB	95%	92.5%	95%

Our experimental study is aimed at estimating the accuracy and comparing the performance of the proposed algorithms. The first part was focused on stationary time series data streams (TSDS) and the second one was focused on non stationary data streams. The stationary data stream was generated from two synthetically distributed normally distributed time series ND25 and ND100 whereas the non-stationary data streams was obtained from two Israel's daily financial indexes TA25 and TA100 [20] . These finance indexes behavior strongly depends on time and therefore they demonstrate non stationary behavior. The few descriptive statistics for the four selected time series is shown in Table 3.

Table 3. Descriptive statistics

TSDS	N	Max	Min	Avg.	St.Dev.
ND25	1,228	1,693.82	-29.21	748.06	244.32
ND100	1,228	1,426.14	-22.79	756.70	225.00
TA25	1,228	1,237.13	333.90	748.06	244.32
TA100	1,228	1,189.04	341.04	756.70	225.00

The ND25 time series is similar to TA25 because their averages, standard deviations and lengths are equal. Same thing is right regarding to TA100. Figure 2 demonstrates the ISW algorithm evaluation on the four collected time series. The blue columns point out the non stationary data as financial indexes and red stationary data e.g. normal processes ND25 and ND100. The most obvious result is that ISW produces more accurate results[2] on stationary data (red columns) when the user specified confidence level is greater than 80%. In case of non stationary data streams the ISW algorithm produces stable but less accurate results. This stable quality pattern results from the ISW algorithm ability to detect mean concept drifts in time series data stream behavior. The remaining error will be caused by other non stationary elements, e.g. non stationary variance or/and non-stationary auto covariance.

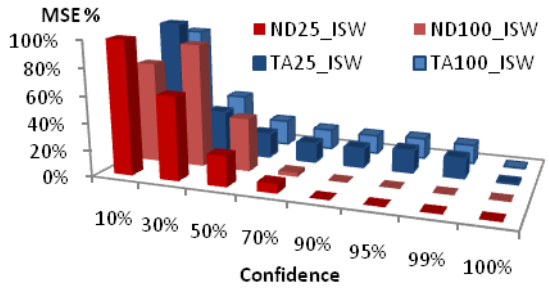

Figure 2. The ISW algorithm accuracy

Figure 2 demonstrates the performance results of ISW algorithm. Actually, this figure Y axis shows the number of created segments when it is obvious that a large number of segments increases the evaluation time of algorithm and vice versa. As previously mentioned the ISW algorithm produces good accuracy results for stationary data, i.e. red columns accuracies range from 0% to 10% when confidence level is greater than 80%.

5. Conclusions

This study has highlighted a number of limitations in existing state-of-the-art online piecewise linear segmentation approaches: Sliding Window (SW) and SWAB. First, the new relative parameter of confidence level was used instead of nominal input parameter of maximal error threshold. This parameter has two advantages: first is that the user does not need to preprocess the original time series data stream in aim to determine the reasonable maximum error value and second is that the proposed technique allows to perform cross comparisons between different time series data streams. Also, the implementation of new real time application was performed. Finally, we have performed an empirical comparison of the proposed time series segmentation algorithms on two types of time series data: stationary (normally distributed data) and non-stationary (financial data).

References

1. Biffet, A. and Kirkby, R. **Data stream mining - A practical approach**, COSI, available at http://www.cs.waikato.ac.nz /~abifet /MOA, 2009, pp 127-141
2. Chan, K. and Fu, W. **Efficient time series matching by wavelets**, proceedings of the 15th IEEE International Conference on Data Engineering, 1999
3. Das, G., Lin, K., Mannila, H., Renganathan, G. and Smyth, P. **Rule discovery from time series**, proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining, 1998, pp. 16-22
4. Ge, X. and Smyth P. **Segmental Semi-Markov models for endpoint detection in plasma etching**, IEEE Transactions on Semiconductor Engineering, 2001.
5. Hoeffding, W. **Probability inequalities for sums of bounded random variables**, Journal of the American Statistical Association, 58(301), 1963, pp. 13-30
6. Hunter, J. and McIntosh, N. **Knowledge-based event detection in complex time series data**, in: Artificial Intelligence in Medicine, Springer, 1999, pp. 271-280.
7. Junker, H., Amft, O., Lukowicz, P. and Tröster, G. **Gesture spotting with body-worn inertial sensors to detect user activities**, in: Source Pattern Recognition, Elsevier, 2008, pp. 2010-2024
8. Keogh, E., Chakrabarti, K., Pazzani, M. and Mehrotra. **Dimensionality reduction for fast similarity search in large time series databases**, Journal of Knowledge and Information Systems, Proceedings of the 22th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 2000
9. Keogh, E., Chu, S., Hart, D. and Pazzani, M. **Segmenting time series: A survey and novel approach**, in: Data Mining in Time Series Databases, World Scientific Publishing Company, 2004, pp. 1-21
10. Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., and Allan, J. **Mining of concurrent text and time series**, Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, 2000, pp. 37-44
11. Li, C., Yu, P. and Castelli, V. **MALM: A framework for mining sequence database at multiple abstraction levels**, Proceedings of the 9th International Conference on Information and Knowledge Management, 1998, pp. 267-272
12. Motwani, R. and Raghavan, P. **Randomized algorithms**, ACM Computing Surveys, 1996, pp. 33-37
13. Hulten, P. and Domingos, G. **A general framework for mining massive data streams**, Journal of Computational and Graphical Statistics, 12(4), 2003, pp. 945-949
14. Osaki, R., Shimada, M. and Uehara, K. **Extraction of primitive motion for human motion recognition**, The 2nd International Conference on Discovery Science, 1999, pp. 351-352
15. Park, S., Kim, S.W. and Chu, W.W. **Segment-based approach for subsequence searches in sequence databases**, Proceedings of the 16th ACM Symposium on Applied Computing, 2001
16. Park, S., Lee, D. and Chu, W.W. **Fast retrieval of similar subsequences in long sequence databases**, Proceedings of the 3rd IEEE Knowledge and Data Engineering Exchange Workshop, 1999
17. Perng, C., Wang, H., Zhang, S. and Parker, S. **Landmarks: A new model for similarity-based pattern querying in time series databases**, Proceedings of 16th International Conference on Data Engineering, 2000
18. Qu, Y., Wang, C. and Wang, S. **Supporting fast search in time series for movement patterns in multiples scales**, Proceedings of the 7th International Conference on Information and Knowledge Management, 1998
19. Shatkay, H. and Zdonik, S. **Approximate queries and representations for large data**

sequences, Proceedings of the 12th IEEE International Conference on Data Engineering, 1996, pp. 546-553

20. Vullings, H.J.L.M., Verhaegen, M.H.G. and Verbruggen, H.B. **ECG segmentation using time-warping**, Proceedings of the 2nd International Symposium on Intelligent Data Analysis, 1997
21. Wang, C. and Wang, S. **Supporting content-based searches on time series via approximation**, Proceedings of the 12th International Conference on Scientific and Statistical Database Management, 2000
22. www.finance.yahoo.com

[1] **Reproducible Results Statement:**
In the interests of competitive scientific inquiry, all datasets and code used in this work are available, together with a spreadsheet detailing the original results, by emailing the first author.

[2] Each experimental accuracy result (i.e. a chart column) is defined by aid of mean square error measure and after that normalized by dividing by the accuracy of the worst algorithm on that experiment.