Expert Systems with Applications 41 (2014) 6098–6105

Contents lists available at ScienceDirect

Expert Systems with Applications

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

ESPSA: A prediction-based algorithm for streaming time series segmentation

Guiling Li [a], Zhihua Cai [a], Xiaojun Kang [a], Zongda Wu [b,*], Yuanzhen Wang [c]

[a] School of Computer Science, China University of Geosciences, Wuhan 430074, China
[b] Oujiang College, Wenzhou University, Wenzhou 325035, China
[c] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Keywords:
Streaming time series
Segmentation
Exponential smoothing
Sliding window

ABSTRACT

Streaming time series segmentation is one of the major problems in streaming time series mining, which can create the high-level representation of streaming time series, and thus can provide important supports for many time series mining tasks, such as indexing, clustering, classification, and discord discovery. However, the data elements in streaming time series, which usually arrive online, are fast-changing and unbounded in size, consequently, leading to a higher requirement for the computing efficiency of time series segmentation. Thus, it is a challenging task how to segment streaming time series accurately under the constraint of computing efficiency. In this paper, we propose exponential smoothing prediction-based segmentation algorithm (*ESPSA*). The proposed algorithm is developed based on a sliding window model, and uses the typical exponential smoothing method to calculate the smoothing value of arrived data element of streaming time series as the prediction value of the future data. Besides, to determine whether a data element is a segmenting key point, we study the statistical characteristics of the prediction error and then deduce the relationship between the prediction error and the compression rate. The extensive experiments on both synthetic and real datasets demonstrate that the proposed algorithm can segment streaming time series effectively and efficiently. More importantly, compared with candidate algorithms, the proposed algorithm can reduce the computing time by orders of magnitude.

1. Introduction

In recent years, streaming time series, a new form of time series, has attracted many researchers' attention, due to its wide applications in many domains such as financial data analysis (Wu, Salzberg, & Zhang, 2004; Wu, Ke, & Yu, 2010), sensor network monitoring (Keogh, Lin, & Fu, 2005), medical diagnosis (Vullings, Verhaegen, & Verbruggen, 1997), moving object tracing (Bu, Chen, & Fu, 2009), and biologic sequence analysis (Zhang, Kao, & Cheung, 2007). As pointed out in Babcock, Bahu, and Datar (2002), compared with traditional archived data (e.g., static time series), streaming time series has the following new characteristics:

- The data elements in streaming time series arrive online.
- The data elements are fast changing and the system cannot control the order.
- The size of a streaming time series is potentially unbounded.
- Once a data element from a streaming time series has processed, it is discarded thus cannot be retrieved easily.

Therefore, in the context of streaming time series, where the data elements are continuously generated, potentially forever, the problem of approximating and mining time series becomes more interesting and challenging.

It has been widely recognized that time series segmentation is the key to time series representation and time series mining (Babcock et al., 2002; Keogh, Selina, & David, 2001). As mentioned in Keogh et al. (2001), time series segmentation can not only detect the change of system model, but also detect when the change happens once the potential system model changes. Besides, more importantly, time series segmentation can create the high-level representation of time series, such that its outcome can be used to support many tasks of time series mining, e.g., indexing, clustering, classification,discord discovery and anomaly detection. Without time series segmentation of good quality, it is an impossible task for time series mining. It has been pointed out in Liu, Lin, and Wang (2008) that a good time series segmentation should

* Corresponding author.
E-mail addresses: guiling@cug.edu.cn (G. Li), zhcai@cug.edu.cn (Z. Cai), xj_kang@126.com (X. Kang), zongda1983@163.com (Z. Wu), wangyz2005@163.com (Y. Wang).

http://dx.doi.org/10.1016/j.eswa.2014.03.043
0957-4174/

meet two basic requirements, i.e., representation quality and computing efficiency:

- The first requirement of good segmentation quality means that a segmentation algorithm should approximate the original time series using fewer segments along with a lower residual error.
- The second requirement of good computing efficiency means that a segmentation algorithm should be efficient enough to adapt for an online real-time computing environment (e.g., a scenario of streaming time series).

As mentioned previously, in the environment of streaming time series, the data elements are continuously online generated and potentially unbounded. Such unique characteristics of streaming time series brings forward a higher requirement for computing efficiency, i.e., the segmentation should be accomplished by single scan over streaming time series, so as to obtain the real-time response. This results in that most existing segmentation algorithms for static time series (Gionis & Mannila, 2005; Keogh et al., 2001), which generally have not adequately considered the problem of computing efficiency, are no longer suitable for streaming time series. Therefore, how to segment streaming time series as accurately as possible under the constraint of real-time computing efficiency, has become the first issue to be solved for approximating and mining streaming time series.

To segment streaming time series, several segmentation algorithms have been suggested, typically, such as Sliding Windows algorithm (*SW*), and Sliding Windows and Bottom-up algorithm (*SWAB*) (Keogh et al., 2001). However, in general, these algorithms have the following shortages, thereby, degrading their representation quality: (1) They just care about how to lower representation error as much as possible, without taking into account the number of segments; (2) They do not make full use of statistical characteristics of arrived data elements of a time series to predict future data elements, so as to detect out segmenting key points and thus improve the accuracy of new generated segments. In real applications such as financial transactions, it is very important to detect emergency events and particular patterns.

Motivated by these, in this paper we focus on the time series segmentation problem under streaming scenarios, and take full account of the requirements of representation quality and computing efficiency. Based on the exponential smoothing prediction, we propose an efficient algorithm for segmenting streaming time series, which has a linear complexity of computing time. The algorithm determines the segmenting key points by analyzing the statistical characteristics of prediction errors, so as to improve the accuracy of new generated segments. The main contributions of this paper are highlighted as follows:

- We propose our new prediction error criterion, which can be used to help to detect out the segmenting key points accurately, and thus provides the theoretical basis for processing the segmentation of streaming time series.
- We propose exponential smoothing prediction-based segmentation algorithm, called *ESPSA*. The algorithm is very efficient, which only needs to perform single scan on the whole time series, and thus is suitable for streaming scenarios.
- We have demonstrated the utility of our proposed algorithm through experimental evaluations. The experimental evaluations were performed over various datasets, whose results have shown that our proposed algorithm performs better in terms of effectiveness and efficiency, compared with candidate algorithms.

The rest of this paper is organized as follows. We review the related work in Section 2. In Section 3, we propose the prediction error criterion, and the prediction-based segmentation algorithm for streaming time series. In Section 4, we conduct extensive experiments to evaluate our proposed algorithm and make discussions. Finally, we conclude this paper and point out the future work in Section 5.

2. Related work

Time series segmentation has been widely studied in the last decades. Keogh et al. (2001) and Gionis and Mannila (2005) have given surveys on different segmentation algorithms. Generally speaking, segmentation algorithms can be classified into two categories: off-line segmentation and on-line segmentation. Off-line methods mainly aim to handle static time series, while on-line methods are suitable for streaming time series. In static time series, all data are archived and stored with a global view. Research on segmenting static time series has been relatively mature; however, most methods for static time series can not be applied directly into streaming time series due to its different characteristics. As data elements arrive on-line, streaming time series only has a local view and requires real-time processing.

Although described with slightly different implementation details, most segmenting methods on streaming time series can be classified as Sliding Windows algorithm. In the classical Sliding Windows algorithm (*SW*) (Keogh et al., 2001), it works by filling data elements continuously into the segment until the constraint condition is not met. Concretely, when new data element arrives, it recalculates the residual error of the current segment and compares the value with predefined error threshold *max_error*, so as to determine whether new data element is put into the current segment. If the residual error is less than *max_error*, it adds new data element into the current segment and then reads next data element for detection. Otherwise, it generates a divided segment, sets anchor by pointing to the current data element and starts segmenting the next segment.

Some researchers have improved and optimized the classical Sliding Windows algorithm. Koski, Juhola, and Meriste (1995) considered the characteristics of ECG data and increased the step forward by $k(k > 1)$ instead 1, thus improved the segmentation speed. Shatkay and Zdonik (1996) set three different segmentation conditions to determine different segmentation positions. Park, Kim, and Chu (2001) made full use of the monotonicity in Sliding Windows algorithm to realize effective segmentation.

Keogh et al. (2001) noticed that Sliding Windows algorithm lacked the ability to look ahead, thus proposed Sliding Windows and Bottom-up algorithm (*SWAB*). It combined Sliding Windows algorithm and Bottom Up algorithm, thus improved the ability of global view. *SWAB* algorithm set a small buffer, applied Bottom Up algorithm for static time series in the buffer and reported the leftmost segment. As long as data arrived, it read the next best fitting subsequence into the buffer and continued processing.

Liu et al. (2008) first proposed a new criterion called feasible space to improve computing efficiency. Based on the criterion, they proposed two segmentation methods, i.e., the feasible space window method and the stepwise feasible space window method, respectively. Fuchs, Gruber, and Nitschke (2010) presented a novel segmentation approach *SwiftSeg*, which was based on a least-squares approximation of time series in sliding windows and utilized a basis of orthogonal polynomials. The key of Sliding Windows algorithm and some variants is to set segmentation condition by error calculation and step of anchor forward, so as to determine the positions of segmentation points. Experiments on existing work of Sliding Windows algorithm have shown that the approximation effect is not as expected.

Some researchers applied prediction theory into processing streaming time series. Mori, Nejigane, and Shimosaka (2005) and