Practical Course: A (very) simple piecewise linear segmentation algorithm

Sergio Peignier

Abstract

In this practical course we will program a simple algorithm of linear piecewise segmentation algorithm for time series data.

1 Introduction

A time series can be defined informally as series of data points indexed in time order. Usually these data points are sampled successively and regularly at a given time rate (discrete events). More formally, let $X = (x_0, x_1, \ldots, x_n)$ such that $x_i \in \mathbb{R}, \forall x_i \in X$ denote a time series of n real valued data points and let $t_i \in \mathbb{R}^+$ denote its time stamp. The analysis of time series has become a very important task in the Data Mining community and it is often referred as Time Series Data Mining. The study of time series involves a large range of applications and tasks, (e.g., forecasting, anomaly detection, classification). An interesting task that has received a lot of attention is the piecewise segmentation of time series. Informally, this task aims at approximating a time series X with a list of l straight lines. The advantage of such a method relies mainly in its easy implementation and the intuitive understanding of its output. We refer the reader to [DYKL10] and [KCHP04] for some examples of algorithms proposed in the literature for time series segmentation, and we refer the reader to [LMS14] for an overview of different available techniques.

In this practical course we will implement a simple linear piecewise segmentation algorithm, and we will apply to the study of EEG signals.

2 Algorithm

2.1 Overview

In a nutshell, the algorithm estimates first the empirical distribution of the derivatives of the time series at each point, then it computes the quantiles of the given distribution (the number of quantiles is a parameter), subsequently the algorithm maps each data object to its corresponding derivative quantile, generating thus a list of derivative quantiles. Large subsequences of repeated derivative quantiles denote (nearly) linear pieces in the time series, whoever small ones are likely to correspond to noise. Hence, the algorithm proceeds to iterate over the list of derivatives to filter the subsequences which length is smaller than a given threshold (parameter of the algorithm) assigning their points to contiguous large subsequences. In practice the algorithm filters subsequences with increasing sizes (first subsequences of unitary length, then subsequences with length 2 and so on). Finally it is possible to compute the slope and intercept of each segment or use the segment directly.

2.2 Moving average

In order to deal with noisy time series, a useful preprocessing step consists in applying a moving average to smooth the time series. This preprocessing step is simple and relies on a single parameter w, which is the width of a sliding window that will be used to compute the moving average along the time series. For each data point x_i, the algorithm considers neighborhood of the given point $(x_i - (w-1)/2, \ldots, x_i + (w-1)/2)$ and computes the mean value $\tilde{x}_i = \frac{1}{w} \sum_{x=x_i-(w-1)/2}^{x_i+(w-1)/2} x$. For the sake of simplicity we consider here that w is an odd number. Notice that the definition does not hold for $i < (w-1)/2$ nor $(w+1)/2$, hence you should modify the previous definition in such cases.

Implement the moving average algorithm using numpy. Does numpy or pandas already have a ready to use moving average function?

2.3 Compute the distribution of the derivatives

Compute the derivatives Compute the derivate at point x_i as $\dot{x}_i = \frac{x_{i+1}-x_i}{t_{i+1}-t_i}$. When is this estimation valid? If the time stamps are not available we compute directly $\dot{x}_i \simeq x_{i+1} - x_i$. Implement a function to compute the derivatives. Does numpy have a better function to estimate the derivatives?

Compute the distribution Compute the derivatives using the previous method for various time series obtained for the same system and concatenate the measures. Plot the histogram of the derivatives (observations?).

Computing the quantiles Sort the list of derivatives and find the required quantiles. Implement a function that returns the requested quantiles of a given list of values. Does numpy have a ready to use function that computes quantiles?

2.4 Compute piecewise linear segments

Map a time series to a derivative quantiles sequence Implement a function that computes the derivatives of a time series received as input, and maps each value to the corresponding quantile (list of quantiles given as a parameter of the function). Does numpy have a ready to use function that does this?

Filter the derivative quantiles sequence As aforementioned, large subsequences of repeated derivative quantiles denote (nearly) linear pieces in the time series, whoever small ones are likely to correspond to noise. Hence, implement a function that proceeds to iterate over the list of derivatives to filter the subsequences which length is smaller than a given threshold (parameter of the algorithm) assigning their points to contiguous large subsequences. In practice the algorithm filters subsequences with increasing sizes (first subsequences of unitary length, then subsequences with length 2 and so on).

Wrap everything Implement a function that received as inputs a sequence, a list of quantiles, a threshold segment size and outputs a piecewise linear segmentation of the time series.

2.5 Application

Download the UCI EEG dataset at https://archive.ics.uci.edu/ml/datasets/EEG+Eye+State. Consider each dimension of the dataset (i.e., each channel) as an independent time series in order to compute the distribution of derivatives. Try different values for the window size of the moving average function, what do you notice? Try different values for the number of quantiles kept as well as for the subsequence threshold size, what do you notice?

References

[DYKL10] Yongwei Ding, Xiaohu Yang, Alexsander J Kavs, and Juefeng Li. A novel piecewise linear segmentation for time series. In *Computer and Automation Engineering (ICCAE), 2010 the 2nd international conference on*, volume 4, pages 52–55. IEEE, 2010.

[KCHP04] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–22, 2004.

[LMS14] Miodrag Lovrić, Marina Milanović, and Milan Stamenković. Algoritmic methods for segmentation of time series: An overview. *Journal of Contemporary Economic and Business Issues*, 1(1):31–53, 2014.