Piecewise segmentation for financial data

Paolo Antonini Davide Azzalini Fabio Azzalini

Abstract

Time series data is characterised as large in data size, high dimensionality and update continuously. Moreover, the time series data is always considered as a whole instead of individual numerical fields. As a consequence, in order to analyse and mine time series data, segmentation and dimensionality reduction are essential. In particular, in the following pages we are going to collect and study some segmentation methods, applied in particular to stock market data. Stock time series has its own characteristics over other time series.

1 Introduction

The tasks of segmentation and dimensionality reduction, as well as identification of trends, are fundamental to allow a number of time series analysis and mining tasks. As a matter of fact, the fields of application of such procedures are numerous (ECG, exchange rates, sensor detections of any kind, . . .) and methods differ from application to application, due to the characteristics of data. As a consequence, literature regarding these issues is vast.

In particular, our focus is on stock market data, which is inherently large in size, noisy and continuously updated. In this paper we are going first to list some of the main research papers where dimensionality reduction and piecewise segmentation are studied, in section 2. Then, in section 3 we are going to present some implementations of the methods.

2 Methods

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

2.1 [1]. *A Pattern Distance-Based Evolutionary Approach to Time Series Segmentation*

Author Yu, Yin, Zhou, *et al.*

Title *A Pattern Distance-Based Evolutionary Approach to Time Series Segmentation*

Year 2006

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

3 Implementations

Turning points Due to its simplicity, the apparently good results and openness to further improvements, we decided to implement the *turning points* method presented in [2]. So, in the following sections 3.1 and 3.2 we are presenting our implementation both in Matlab and in Python of the method.

Basically, the algorithm is divided into two phases. First, during a preprocessing phase all points which are neither local maxima nor minima are discarded; then some patterns are simplified, since immaterial. However, we are not going to discuss the theoretical foundations and the steps of the algorithm in detail, as the aforementioned article if sufficient.

Other implementations In order to provide a better view on the implementations, we are presenting other methods in section 3.3. Most notably, ...

3.1 Turning points in Matlab

Implementation The method is developed and tested over `CSV` files downloaded from Yahoo! Finance website[1]. Should another source be used, basic adaptations may be necessary, mostly in the handling of temporal data[2] (i.e., dates), which is performed in `TP_prepareData()` procedure.

After importing the aforementioned `CSV` file (we suggest to use the graphical interface provided by Matlab itself), the user should issue the following command, in order to compute and plot the results:

```
y = TurningPoints(time, values, n);
```

Here, `time` and `values` represent the time series as imported from the `CSV`; `n` instead lets the user specify the number of times the algorithm shall be performed (preprocessing is excluded from this count). The results are both displayed in a plot and stored in `y` variable.

`TurningPoints()` function is implemented as shown in listing 1, with the support of some side functions.

[1] `http://finance.yahoo.com/market-overview/`

[2] Due to our limited knowledge of Matlab language, we may have dealt with the source data in a naïve way. Nevertheless, the procedure seems to work well.

```matlab
%% Actual TP function
function tp = TurningPoints(time, value, n)

x = TP_prepareData(time, value);
tp = TP_preprocess(x);

while n > 0

    n = n-1;
    y = tp;
    clear tp;

    i = 1;
    while i < (length(y)-3)

        p0 = y(i+0,2);
        p1 = y(i+1,2);
        p2 = y(i+2,2);
        p3 = y(i+3,2);

        condUT = p0 < p1 && p0 < p2 && p1 < p3 && p2 < p3 ...  % uptrend
            && abs(p1 - p2) < abs(p0 - p2) + abs(p1 - p3);

        condDT = p0 > p1 && p0 > p2 && p1 > p3 && p2 > p3 ...  % downtrend
            && abs(p2 - p1) < abs(p0 - p2) + abs(p1 - p3);

        eps = 0.05 * mean([p0 p1 p2 p3]);
        condST = abs(p0 - p2) < eps && abs(p1 - p3) < eps;  % same trend

        if condUT || condDT || condST
            tp(i,:) = y(i,:);
            tp(i+3,:) = y(i+3,:);
            i=i+3;
        else
            tp(i,:) = y(i,:);
            i=i+1;
        end  % end if

    end  % end while i < (length(y)-3)

    tp(length(y),:) = y(length(y),:);
    tp = TP_cleaning(tp);
    TP_output(y,tp)

end  % end while n > 0

plot( datetime ( x(:,1), 'ConvertFrom', 'datenum'), x(:,2), ...
    datetime ( tp(:,1), 'ConvertFrom', 'datenum'), tp(:,2));

end  % end TurningPoints()
```

Listing 1: `TurningPoints()` function.

The main supporting function is `TP_preprocess()`, which implements the preprocessing phase, and is presented in listing 2. The other supporting functions are of secondary importance, so we are not presenting them here. Basically we have:

- `TP_prepareData()` takes in input raw Yahoo! Finance data and prepares them to the processing;
- `TP_cleaning()` cleans data matrix after processing;
- `TP_output()` shows information about the processing (i.e., the number of deleted elements).

```
%% Data preprocessing
function y = TP_preprocess(x)

% Boundary elements
y(1,:) = x(1,:);
y(length(x),:) = x(length(x),:);

% Core
for i=2:(length(x)-1)

    prec = x(i-1,2);
    curr = x(i,2);
    succ = x(i+1,2);

    condMIN = curr < prec && curr < succ;  % curr: local minimum
    condMAX = curr > prec && curr > succ;  % curr: local maximum
    if condMIN || condMAX
        y(i,:) = x(i,:);
    end  % end if

end  % end for

y = TP_cleaning(y);
TP_output(x,y)

end  % end TP_preprocess()
```

Listing 2: `TP_preprocess()` supporting function.

Test To conclude our discussion, we are presenting some tests we performed. Weekly stock market data from A2A (`A2A.MI`) over the whole 2015 were used as source. In particular, we plotted the weekly `Close` time series. In fig. 1 we show the original data in blue, the preprocessed data in orange and the data after one full run of the algorithm in yellow.

Original data contains 53 samples; preprocessing reduces them to 27, and finally, after the actual processing, only 12 samples are left.

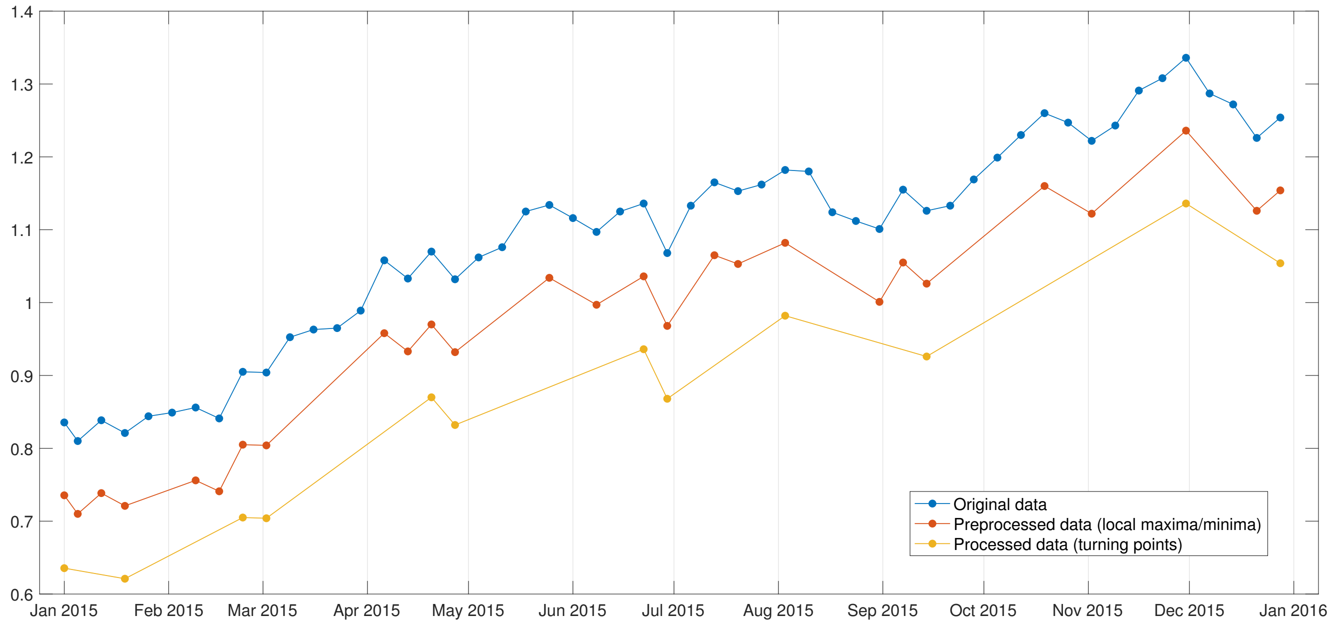

Figure 1: `A2A.MI` weekly 2015. Original data is in the correct position. The other two series are shifted down by 0.1 each.

3.2 Turning points in Python

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

3.3 Other implementations

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Bibliography

DOI (Digital Object Identifier), when defined, serves as URL to retrieve the document. Documents may be accessible only on institutional log in (e.g., academic credentials).

[1] J. Yu, J. Yin, D. Zhou, and J. Zhang, *A pattern distance-based evolutionary approach to time series segmentation*, 2006. doi: `10.1007/978-3-540-37256-1_99`.

[2] J. Yin, Y.-W. Si, and Z. Gong, "Financial time series segmentation based on turning points", in *System Science and Engineering (ICSSE), 2011 International Conference on*, Jun. 2011, pp. 394–399. doi: `10.1109/ICSSE.2011.5961935`.

[3] F.-L. Chung, T.-C. Fu, V. Ng, and R. W. Luk, "An evolutionary approach to pattern-based time series segmentation", *Trans. Evol. Comp*, vol. 8, no. 5, pp. 471–489, Oct. 2004. doi: `10.1109/TEVC.2004.832863`.

[4] T.-c. Fu, F.-l. Chung, and N. Chak-man, "Financial time series segmentation based on specialized binary tree representation", in *Int. Conf. on Data Mining*, 2006, pp. 3–9.

[5] T.-c. Fu, F.-l. Chung, R. Luk, and C.-m. Ng, "Stock time series pattern matching: Template-based vs. rule-based approaches", *Engineering Applications of Artificial Intelligence*, vol. 20, no. 3, pp. 347–364, 2007. doi: `http://dx.doi.org/10.1016/j.engappai.2006.07.003`.

[6] ——, "Representing financial time series based on data point importance", *Engineering Applications of Artificial Intelligence*, vol. 21, no. 2, pp. 277–300, 2008. doi: `http://dx.doi.org/10.1016/j.engappai.2007.04.009`.

[7] C. Phetking, M. N. M. Sap, and A. Selamat, "Identifying zigzag based perceptually important points for indexing financial time series", in *Cognitive Informatics, 2009. ICCI '09. 8th IEEE International Conference on*, Jun. 2009, pp. 295–301. doi: `10.1109/COGINF.2009.5250725`.

[8] T.-c. Fu, "A review on time series data mining", *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011. doi: `http://dx.doi.org/10.1016/j.engappai.2010.09.007`.

[9] T.-c. Fu, F.-l. Chung, K.-y. Kwok, and C.-m. Ng, "Stock time series visualization based on data point importance", *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1217–1232, 2008. doi: `http://dx.doi.org/10.1016/j.engappai.2008.01.005`.