

Financial Time Series Segmentation based on Specialized Binary Tree Representation

Tak-chung Fu^{1,2,†}, Fu-lai Chung¹ and Chak-man Ng²

¹*Department of Computing*

The Hong Kong Polytechnic University

Hung Hom, Kowloon, Hong Kong.

²*Department of Computing and Information Management*

Hong Kong Institute of Vocational Education (Chai Wan)

30, Shing Tai Road, Chai Wan, Hong Kong.

Abstract—Segmentation is one of the fundamental components in time series data mining. One of the uses of the time series segmentation is trend analysis - to segment the time series into primitive trends like uptrend and downtrend. In this paper, a time series segmentation method based on a specialized binary tree representation scheme is proposed; this representation scheme is customized for financial time series to cater for its unique behaviors. The proposed segmentation method is based on the concept of data point importance and the location of the cutting points is already encoded in the representation scheme. Therefore, no additional effect is needed to determine the cutting points. One may find it particularly attractive in applications like stock data analysis. The unique behavior of the proposed segmentation method is demonstrated by applying to financial time series.

I. INTRODUCTION

RECENTLY, the increasing use of temporal data has initiated various researches and development efforts in the field of data mining. Time series is an important class of temporal data objects and can be easily obtained from financial and scientific applications (e.g., daily temperature, weekly sales totals, and prices of mutual funds and stocks). They are, in fact, major sources of temporal databases. Unlike transactional databases with discrete items, time series data are characterized by their numerical and continuous nature. Shatkey and Zdonik [1] suggest dividing the sequences into meaningful subsequences and representing those subsequences using real-valued functions. Time series segmentation is considered to be one of the fundamental components in time series data mining for efficient data storage, transmission, computation, visualization and trend analysis.

In this paper, a time series segmentation method based on the specialized binary tree (i.e. SB-Tree) representation scheme is proposed. The representation scheme has first proposed in [2] and is customized for financial time series applications. Stock time series has its own characteristics over other time series data (e.g. data from scientific areas like

electrocardiogram, ECG). For example, it is typically characterized by a few critical points and multi-resolution consideration is always necessary for long-term and short-term analyses. In addition, technical analysis is usually used to identify patterns of market behavior, which have high probability to repeat themselves. These patterns are similar in the overall shape but with different amplitudes and/or durations. Based on the SB-Tree representation scheme, time series segmentation can be easily achieved by accessing the tree, no additional process is required for time series segmentation. The paper is organized into five sections. The next section contains a discussion of related works. Section 3 presents the SB-Tree representation scheme and introduces how to achieve the segmentation result under the SB-Tree representation scheme. The simulation results are reported in section 4 and the conclusion is made in the final section.

II. RELATED WORK

Time series segmentation is also considered as a discretization problem. In [3], a simple discretization method is proposed. A fixed length window is used to segment a time series into subsequences and the time series is then represented by the primitive shape patterns that are formed. This discretization process mainly depends on the choice of the window width. However, using fixed-length segmentation is an over-simplified approach to solve the problem. There are at least two identified disadvantages. First, with fixed-length subsequences, only patterns whose length does not vary will be considered in the mining process. However, meaningful patterns typically appear with different lengths throughout a time series. Second, as a result of the even segmentation of a time series, meaningful patterns may be missed if they are split across cutting points. Thus, it is better to use a dynamic approach, which identifies the cutting points in a more flexible way (i.e., using different window widths). However, this is certainly not a trivial segmentation problem. Common segmentation methods include detecting special events in the time series as the cutting points [4], minimum message length segmentation

[†] Corresponding author, e-mail: cstcfu@comp.polyu.edu.hk

[5], and segmentation by piecewise linear approximation (PLA) [6].

In [7], it suggests that the cutting points are identified at which behavior changes occur in a time series. In the statistical term, this is called the “change-point detection problem”. The standard solution involves fix the number of change-points, then identify their positions, and finally determine functions for curve fitting the intervals between successive change-points. In [4], an iterative algorithm is proposed that fits a model to a time segment and then uses a likelihood criterion to determine if the segment should be partitioned further. In [8], it suggests discovering the underlying switching process in a time series, which entails identifying the number of sub-processes and the dynamics of each sub-process. The concept of the nonlinear gated experts derived from statistical physics was proposed to perform the segmentation. In [9][10], dynamic programming is proposed to determine the total number of intervals within the data, the location of these intervals and the order of the model within each segment. In [11], the segmentation problem is considered with a tool for exploratory data analysis and data mining, called the scale-sensitive gated experts (SSGE), which can partition a complex nonlinear regression surface into a set of simpler surfaces called “features”.

The segmentation problem has also been considered from the perspective of finding cyclic periodicity for all of the segments. In [12][13], the data cube and the Apriori data mining techniques were used to mine segment-wise periodicity using a fixed length period. An off-line technique for the competitive identification of piecewise stationary time series is described in [14]. In addition to performing piecewise segmentation and identification, the proposed technique maps similar segments of a time series as neighbors on a neighborhood map.

Furthermore, a pattern-based time series segmentation method is proposed in [15] which is based on evolutionary computation. An intensive revision on the problem of time series segmentation can be found in [6]. It classifies the methods for time series segmentation to three categories: sliding windows, top-down and bottom-up approaches. It also proposes a hybrid method for online time series segmentation.

As we can see, much of the recent research on this and similar problems can be characterized procedurally in the following general manner [16]: (i) find an approximation and robust representation for a time series, for example, Fourier coefficients and piecewise linear models; (ii) define a flexible matching function that can handle various pattern variations (scaling, transformations, etc.); and (iii) provide an efficient scalable algorithm, using the adopted representation and matching function, for massive time-series data sets. In this paper, the time series segmentation method is already encoded in the time series representation scheme and therefore step (ii) and (iii) are not necessary.

III. SPECIALIZED BINARY TREE REPRESENTATION

In this section, the financial time series representation scheme, which is adopted in this paper, will be revisited. It is based on determining the data point importance in the time series. Instead of storing the time series data according to time or transforming it into other domains (e.g. frequency domain), data points of a time series are stored according to their importance.

A. Data Point Importance

In view of the importance of extreme points in stock time series, the identification of Perceptually Important Points (PIP) is first introduced in [17] and used for pattern matching of technical (analysis) patterns in financial applications. The idea was later found similar to a technique proposed about 30 years ago for reducing the number of points required to represent a line [18] (see also [19]). Similar idea can be found in independent works [20][21][22].

The frequently used stock patterns are typically characterized by a few critical points. For example, the head-and-shoulder pattern consists of a head point, two shoulder points and a pair of neck points. These points are perceptually important in the human identification process and should be considered as having higher importance. The proposed scheme follows this idea by reordering the sequence P based on the PIP identification process, where the data point identified in an earlier stage is considered as being more important than those points identified afterwards.

The distance measurement depicted in Figure 1 is the vertical distance (VD) between the test point p_3 and the line connecting the two adjacent PIPs, i.e.,

$$VD(p_3, p_c) = |y_c - y_3| = \left| \left(y_1 + (y_2 - y_1) \cdot \frac{x_c - x_1}{x_2 - x_1} \right) - y_3 \right| \quad (1)$$

where $x_c = x_3$.

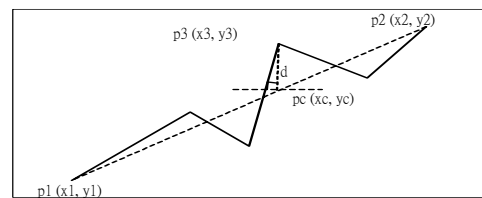


Fig. 1. Vertical distance measure: PIP-VD

To illustrate the identification process, Figure 2 shows the step-by-step results. Here, the number of data points in the sample time series P and the first 5 data point identification processes are shown.

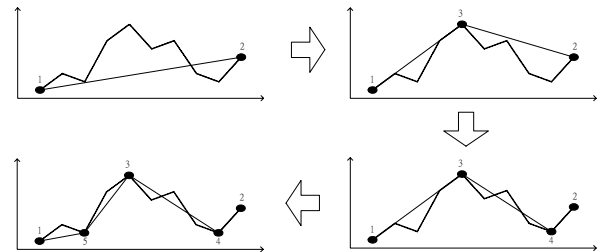


Fig. 2. Identification of the first 5 perceptually important points

B. SB-tree Data Structure

After introducing the concept of the data point importance, the time series can be reordered and a Specialized Binary Tree (SB-Tree) structure is recommended for storing the time series data. The SB-Tree is first proposed in reference [2].

To create a SB-Tree, the PIP identification process is adopted. The first and last data points in the given sequence are not necessarily added to the tree as they are fixed. The third PIP identified becomes the root node of the tree. Next, a recursive interpretation for building a tree is demonstrated. Starting from the root node ptr , the current identified PIP forms a $node$:

- If $node \rightarrow x$ is less than $ptr \rightarrow x$ then goto the left arc of ptr
 - If $ptr \rightarrow left$ is empty, add $node$ to this position
 - Else $ptr = ptr \rightarrow left$ and start the next iteration
- Else goto the right arc of ptr
 - If $ptr \rightarrow right$ is empty, add $node$ to this position
 - Else $ptr = ptr \rightarrow right$ and start the next iteration

Figure 3 shows the detailed algorithm for creating a SB-Tree. As a simple example, Figure 4 gives a sample time series with 10 data points and Figure 5 shows the steps of creating the corresponding SB-Tree. In Figure 4, the points with smaller number labels are more important (e.g. PIP3 is more important than PIP4).

Fig. 3. Pseudo code of building the SB-tree

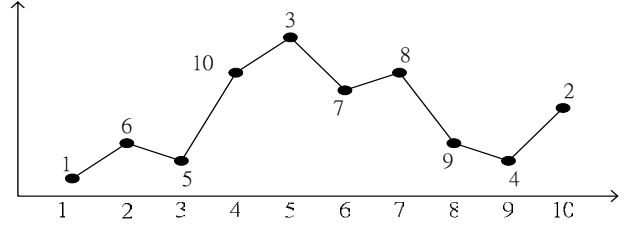


Fig. 4. Sample time series with 10 data points

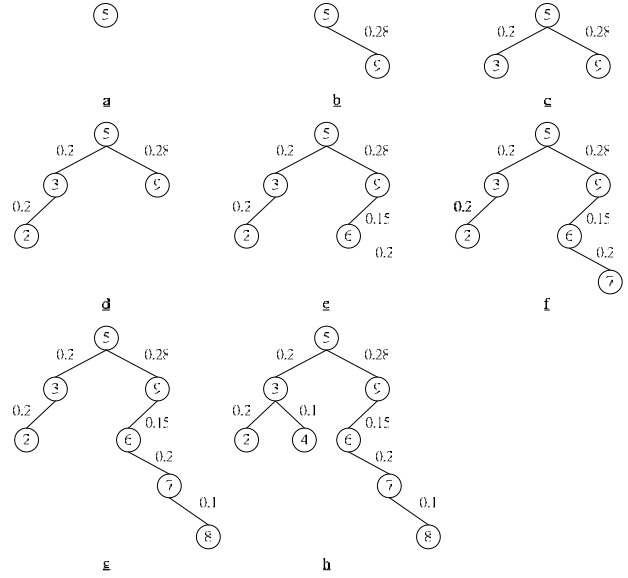


Fig. 5. The tree built from the sample time series

```

Function Create_SB_Tree (P)
Input:  sequence P[1..m]
Output: SBTree root
Begin
root = NULL
Repeat until all P[1..m] are marked
Begin
Select point P[j] which does not marked
and with maximum VD to the adjacent marked
points
Create node

If (root = NULL) Then
root=node
Else
ptr=root
Repeat until found the placing
position
Begin
If (node->x > ptr->x) Then
If (ptr->left = NULL) Then
ptr->left=node
Else
ptr=ptr->left
End
Else
If (ptr->right = NULL) Then
ptr->right=node
Else
ptr=ptr->right
End
End
End
End
Return root
End
    
```

C. Segmentation based on SB-Tree

In this section, a segmentation method which can discretize a complex time series into primitive patterns like uptrend and downtrend is proposed. After representing the time series using a SB-Tree, the information of the cutting points for time series segmentation is already encoded in the SB-Tree.

By considering the PIPs in the SB-Tree as the cutting points for time series segmentation, similar results of segmenting time series to primitive patterns can be obtained. As the PIP identification process will first identify the data point which is the most important one, in a sense that the data point has the greatest influence to the shape of the overall time series. If it is considered as a cutting point, the time series will be segmented into two major trends: a major uptrend and a major downtrend.

Therefore, supposing that an SB-Tree and the number of cutting points required are given. The SB-Tree is accessed recursively, started from the root and the time series cutting points will be retrieved according to their importance.

- The root of the SB-tree is the first cutting point. It will be marked as USED when retrieved
- The SB-tree is accessed from the root and each accessible node in each path of the tree will be reached.

An accessible node is defined as the first node in a path that is not marked as USED

- By comparing the distances among all the accessible nodes, the one with maximum distance is selected as the next cutting point. Again, this node will be marked as USED when retrieved
- This process continues until the required number of cutting points is selected (i.e. marked as USED).

Figure 6 shows the pseudo code for retrieving the cutting points from the SB-Tree. Figure 7 shows different numbers of cutting point selected from the SB-Tree for segmentation. Figure 8a shows that the sample time series in Figure 4 is segmented into a major uptrend and a major downtrend by retrieving one PIP from the SB-Tree. The proposed method can segment the time series into a series of uptrends and downtrends by retrieving a given number of PIPs from the tree and considering them as the cutting points for segmentation. Figure 8 shows the result when considering different numbers of cutting point for segmentation. In fact, it is another way to represent a time series in different resolutions in the view of the segmentation problem.

```

Function Cutpt_Retrieval(root, no_cutpt)
  Input: SBTree root
         Integer no_cutpt
  Output: List list //list of cutting pts
  Begin
    PIPNode node
    Repeat until no. of node selected = no_cutpt
    Begin
      node->dist = -1
      Find_MAX_PIP(root, node)
      Append node->x To list
      Marked node as USED
    End
    Return list
  End

Function Find_MAX_PIP
  Input: PIPNode cur_node
         PIPNode Max_node
  Begin
    If node NOT marked USED Then
      If (cur_node->dist > Max_node->dist)
      Then
        Max_node=cur_node
      End
    Else
      If (cur_node->left <> NULL) Then
        Find_MAX_PIP(pt->left, Max_node)
      End
      If (cur->node->right <> NULL) Then
        Find_MAX_PIP(pt->right, Max_node)
      End
    End
  End
End
  
```

Fig. 6. Pseudo code of retrieving the cutting points from the SB-tree

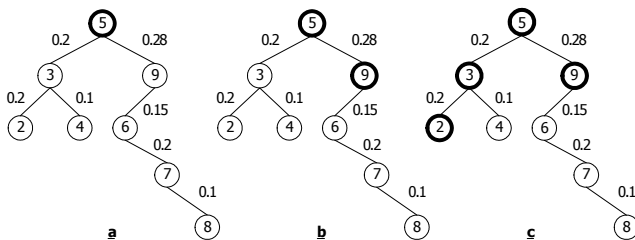


Fig. 7. Different number of cutting points (a=1, b=2 and c=4) is retrieved from the SB-tree for segmentation

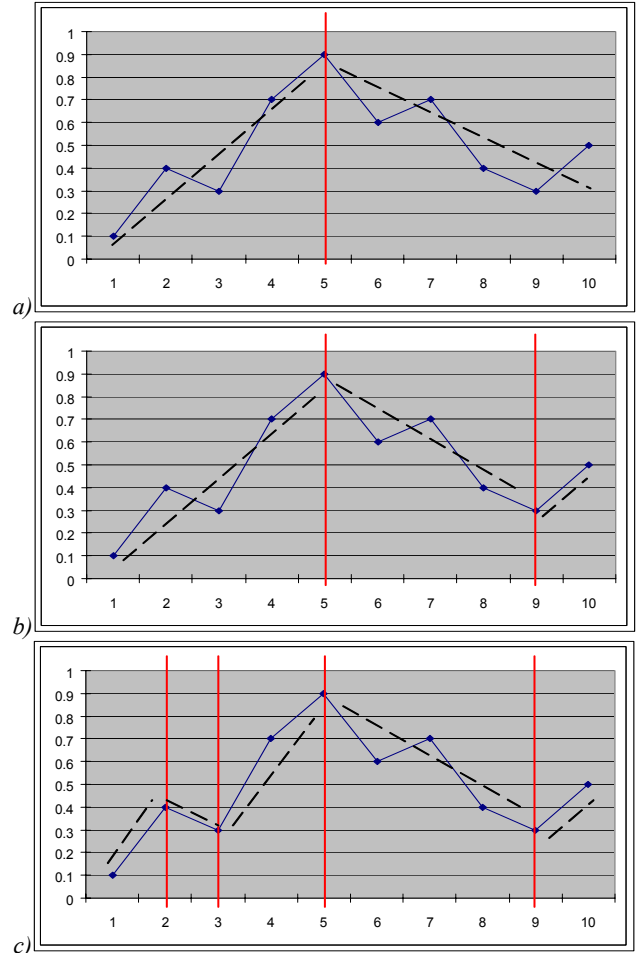


Fig. 8. Time series segmentation result using the proposed method based on different number of cutting points (a=1, b=2 and c=4)

Furthermore, besides predefining the number of time points required as the segmentation criteria, another way is to consider the error rate as the segmentation criterion, that is the information loss if the time series is represented with the trends after segmentation. Therefore, selecting more time points will decrease the information loss.

IV. EXPERIMENTAL RESULTS

In this section, the segmentation result of using the proposed method is compared with an existing technique, Piecewise Linear Approximation (PLA) [6]. Time series with 2352 data points captured from the past 10 years Hong Kong Hang Seng Index (HSI) was used for demonstration. At the end of this section, more segmentation results on two Hong Kong stock closing price series are reported. The error (distance) between the time series represented by the cutting points selected and the original time series is also evaluated. Error here is defined as the mean square distance between the original time series and the series formed by n number of cutting points. In other words, the error is calculated by the

linear interpolation between retained points (i.e. cutting points) and the original time series. Fig.9 shows the error when only 3 cutting points are used to represent the sample time series.

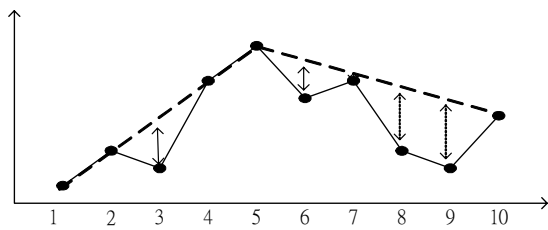


Fig. 9. Error of representing a time series with 3 cutting points compared to the original time series

First, the result of segmenting the HSI time series based on the tree representation is shown. By selecting different numbers of cutting point from the SB-Tree, different levels of segmentation results can be obtained. Figure 10a, 10b and 10c show the result from three different levels of cutting point's consideration. Primitive trends can be formed by the proposed segmentation method. The corresponding segmentation results of PLA are shown in Figure 10d, 10e and 10f. As we can see, the PLA approach preferred to segment the time series evenly while the proposed method tended to segment the time series based on the fluctuation of the time series. The nature of the representation is to capture the fluctuation of a time series based on the degree of importance of each data point. Therefore, the cutting points maybe locate in a small segment that has great fluctuation.

Figure 11 shows the error between the time series represented by the cutting points selected and the original time series. By representing the time series by a few cutting points only, the error of the proposed SB-Tree approach is smaller than the PLA approach. More importantly, by visualizing the time series by the cutting points selected, the proposed SB-Tree approach obtains a better shape reconstruction effect (Figure 12a, 12b and 12c) than the PLA approach (Figure 12d, 12e and 12f). It is an important factor especially for visualizing the stock time series.

Moreover, no additional effort is needed for the segmentation process if the SB-Tree representation is already adopted to represent a time series. There is nearly no time required for obtaining the cutting points from the SB-Tree but time consumption is an important consideration in the segmentation process when using the PLA approach.

Besides specifying the number of cutting point the users want to investigate, the users can also set this parameter with a specific error rate acceptance by representing the time series using the segments. Obviously, the more the segments, the lower the error level of the time series data compared to the original one. Figure 13 shows the relationship of the number of segment and the error on the HSI time series. For example, if the error rate acceptance set by a user is 25, about

1000 cutting points (i.e. 1001 segments) are needed to represent the time series.

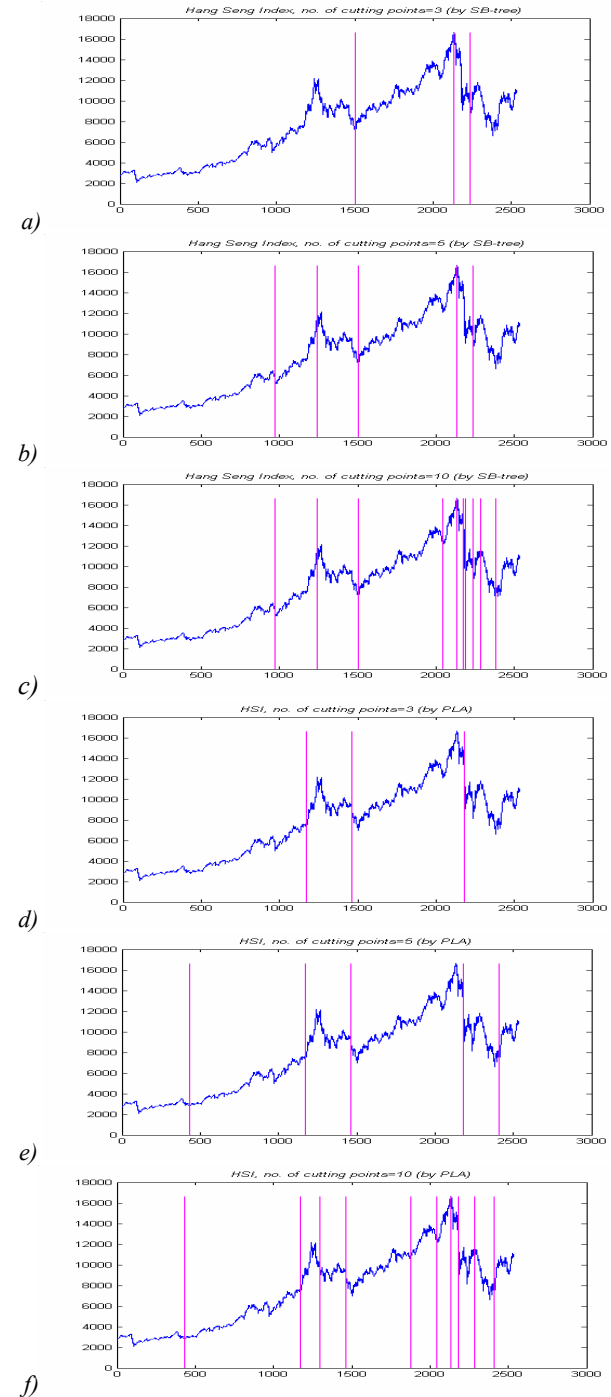


Fig. 10. Segmentation result on the Hang Seng Index with 2532 data points by the proposed segmentation approach (a, b & c) and PLA approach (d, e, & f) (different numbers of cutting point are selected: i.e. a & d = 3, b & e = 5 and c & f = 10)

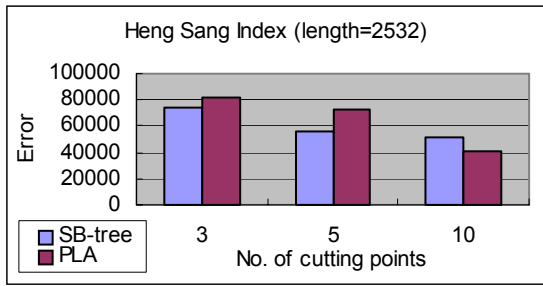


Fig. 11. Error by representing the time series with different number of cutting points by the SB-Tree and PLA approaches

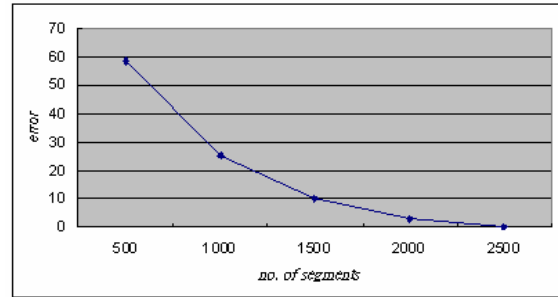


Fig. 13. Error on using different number of segments to represent the time series

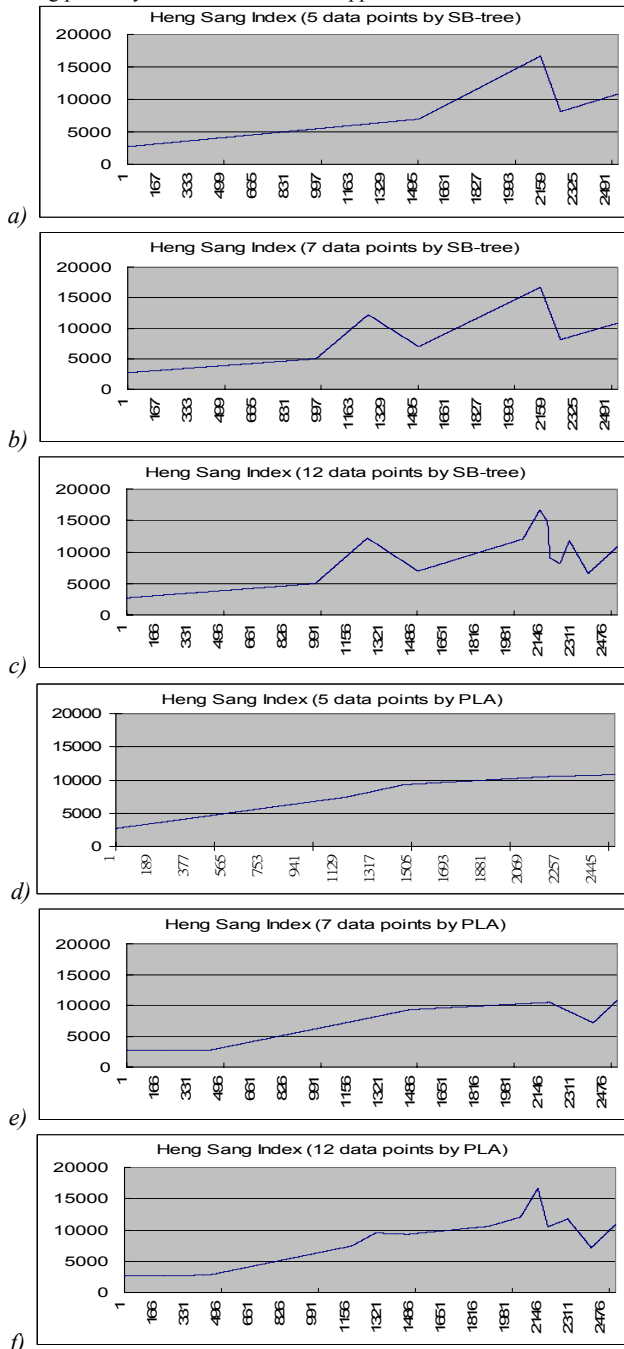


Fig. 12. Visualization results on representing the time series with different number of cutting points by the SB-Tree (a, b & c) and PLA (d, e, & f) approaches (different numbers of cutting point are used: i.e. a & d = 3, b & e = 5 and c & f = 10)

Finally, the segmentation results of two more closing price time series from the Hong Kong stock market are also reported. Figure 14 shows the error by representing the corresponding time series by the cutting points selected. Again, the error of using the proposed SB-Tree approach is smaller than the PLA approach by representing the time series using a few cutting points. A point needed to be mentioned here is that the error of using the SB-Tree approach is not guaranteed to decrease with using more cutting points to reconstruct the time series. It is because the PIP identification process depends on the global shape captured in previous PIP identification iteration but not an overall shape. Therefore, the change of such shape will affect the error of the original time series compared with the time series formed by the cutting points selected (e.g. used 5 cutting points in stock time series 0008).

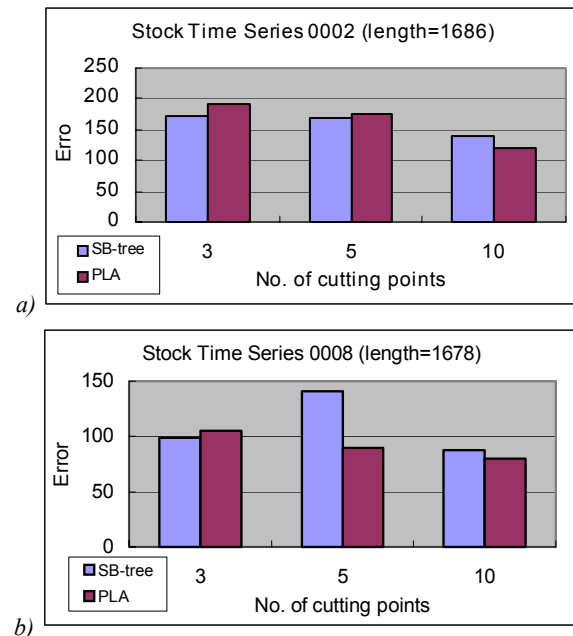


Fig. 14. Error by representing the stock 0002 (left) and 0008 (right) time series with different number of cutting points by the SB-Tree and PLA approaches

V. CONCLUSION

In this paper, the usage of the SB-Tree time series representation on the segmentation problem is demonstrated. Differing from the traditional segmentation approaches, the proposed segmentation method is customized for financial time series and based on the importance of the data points. The more important of a data point, the earlier this point will be considered as a cutting point. It can segment time series into a series of primitive trends. Moreover, the location of the cutting points has already encoded in the representation scheme. Therefore, no additional effort will be needed to determine the cutting points.

REFERENCES

- [1] H. Shatkay and S. Zdonik, "Approximate Queries and Representations for Large Data Sequences," *In Proc. of the 12th ICDE*, 1996, pp.536-545.
- [2] T.C. Fu, F.L. Chung, R. Luk and C.M. Ng, "A Specialized Binary Tree for Financial Time Series Representation," *The 10th ACM SIGKDD Workshop on Temporal Data Mining*, 2004, pp.96-103.
- [3] G. Das, K. I. Lin and H. Mannila, "Rule discovery from time series," *In Proc. of the 4th ACM SIGKDD*, 1998, pp.16-22.
- [4] V. Guralnik and J. Srivastava, "Event detection from time series data," *In Proc. of the 5th ACM SIGKDD*, 1999, pp.33-42.
- [5] J.J. Oliver, R.A. Baxter and C.S. Wallace, "Minimum message length segmentation," *In Proc. of the PAKDD*, 1998, pp.222-233.
- [6] E. Keogh, S. Chu, D. Hart and M. Pazzani, "An Online Algorithm for Segmenting Time Series," *In Proc. of the 1st IEEE ICDM*, 2001, pp.289-296.
- [7] J.J. Oliver and C.S. Forbes, "Bayesian approaches to segmenting a simple time series," *Technical Reports 97/336, Department of Computer Science*, Monash University, Melbourne, Australia, 1997, pp.1-20.
- [8] A.N. Srivastava and A.S. Weigend, "Improved Time Series Segmentation using Gated Experts with Simulated Annealing," *In Proc. of the ICNN*, 1996, pp.1883-1888.
- [9] G.F. Bryant, S.R. Duncan, "A solution to the segmentation problem based on dynamic programming," *In Proc. of the 3rd IEEE Conference on Control Applications*, Vol.2, 1994, pp.1391-1396.
- [10] S.R. Duncan and G. F. Bryant, "A new algorithm for segmenting data from time series," *In Proc. of the 35th IEEE Decision and Control Conference*, Vol.3, 1996, pp.3123-3128.
- [11] A.N. Srivastava, R. Su and A.S. Weigend, "Data mining for features using scale-sensitive gated experts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.21, No.12, pp.1268-1279, Dec. 1999.
- [12] J. Han, W. Gong and Y. Tin, "Mining segment-wise periodic patterns in time-related databases," *In Proc. of the 4th ACM SIGKDD*, 1998, pp.214-218.
- [13] J. Han, G. Dong and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database," *In Proc. of the 15th ICDE*, 1999, pp.106-115.
- [14] C.L. Fancoua, J.C. Principe, "A neighborhood map of competing one step predictors for piecewise segmentation and identification of time series," *In Proc. of the ICNN*, Vol.4, 1996, pp.1906-1911.
- [15] F.L. Chung, T.C. Fu, Vincent Ng, and Robert Luk, "An Evolutionary Approach to Pattern-based Time Series Segmentation," *IEEE Transactions on Evolutionary Computation*, Vol.8, pp.471-489, 2004.
- [16] X. Ge and P. Smyth, "Deformable Markov model templates for time-series pattern matching," *In Proc. of the 6th ACM SIGKDD*, 2000, pp.81-90.
- [17] F.L. Chung, T.C. Fu, R. Luk and V. Ng, "Flexible Time Series Pattern Matching Based on Perceptually Important Points," *IJCAI Workshop on Learning from Temporal and Spatial Data*, 2001, pp.1-7.
- [18] D. Douglas and T. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *The Canadian Cartographer*, Vol.10, No.2, 1973, pp.112-122.
- [19] J. Hershberger and J. Snoeyink, "Speeding up the Douglas-Peucker line-simplification algorithm," *In Proc. of the 5th Symposium on Data Handling*, 1992, pp.134-143.
- [20] C.S. Perng, H. Wang, R. Zhang and D. Parker, "Landmarks: A new model for similarity-based pattern querying in time series databases," *In Proc. of the 16th ICDE*, 2000, pp.33-42.
- [21] B. Pratt and E. Fink, "Search for patterns in compressed time series," *Image and Graphics*, Vol.2, No.1, 2002, pp.89-106.
- [22] E. Fink and B. Pratt, "Indexing of compressed time series," *Data Mining in Time Series Databases*, 2003, pp.51-78.