Iat-Long Lei, Phoey Lee Teh, Yain-Whar Si, Direct least squares fitting of ellipses segmentation and prioritized rules classification for curve-shaped chart patterns, Applied Soft Computing, Volume 107, 2021, 107363, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2021.107363

Direct Least Squares Fitting of Ellipses Segmentation and Prioritized Rules Classification for Curve-shaped Chart Patterns

Iat-Long Lei[#], Phoey Lee Teh[*], Yain-Whar Si[#]
Department of Computer and Information Science, University of Macau[#]
Department of Computing and Information Systems, Sunway University[*]
nico.lei@icloud.com, phoeyleet@sunway.edu.my, fstasp@umac.mo

Abstract. In financial markets, appearances of chart patterns in time series are commonly considered as potential signals for imminent change in the direction of price movement. To identify chart patterns, time series data is usually segmented before it can be processed by different classification methods. However, existing segmentation methods are less effective in classifying 16 curve-shaped chart patterns from financial time series. In this paper, we propose three novel segmentation methods for classification of curve-shaped chart patterns based on direct least squares fitting of ellipses. These methods are implemented based on the principles of sliding windows, turning points, and bottom-up piece wise linear approximation. To further enhance the efficiency of classifying chart patterns from real-time streaming data, we propose a novel algorithm called Accelerating Classification with Prioritized Rules (ACPR). Experiments based on real datasets from financial markets reveal that the proposed approaches are effective in classifying curve-shaped patterns from time series. Experiment results reveal that the proposed segmentation methods with ACPR can significantly reduce the total execution time.

Keywords: Financial time series, Segmentation, Sliding window, Chart patterns, Direct least squares fitting of ellipses.

1. Introduction

Time series is a set of sequence data points and each data point is indexed by a timestamp. One of the best known applications of time series analysis is the identification of chart patterns from stock market data. In technical analysis, traders often consider the appearance of chart patterns in price data as an imminent sign of change in future price trend. Therefore, detection and identification of chart patterns from time series become an important task in financial trading community.

According to [1], there are 53 chart patterns commonly used in financial trading community. These patterns were later classified into five categories based on their shapes [2]. Accordingly, we number the five categories from C1 to C5: C1-Variable fluctuation chart patterns; C2-Fixed fluctuation chart patterns; C3-Curve-shaped chart patterns; C4-Candlestick patterns with spikes; C5-Candlestick patterns with gaps. The patterns from C1 and C2 only contain straight lines and they could be classified by using existing time series segmentation methods such as PIP [3], TP [4] , PLA [5], and PAA [6]. The patterns from C4 and C5 are Candlestick patterns and they could also be segmented by PIP, TP, PLA, and PAA after pre-processing of the data. However, to the best of authors' knowledge, there are no existing segmentation methods which are suitable for classifying curve-shaped patterns from category C3. Therefore, in this paper, we focus on the classification of chart patterns from category C3-Curve-shaped chart patterns. Altogether, there are 16 curve-shaped patterns in C3. These patterns are depicted in Figure 1.

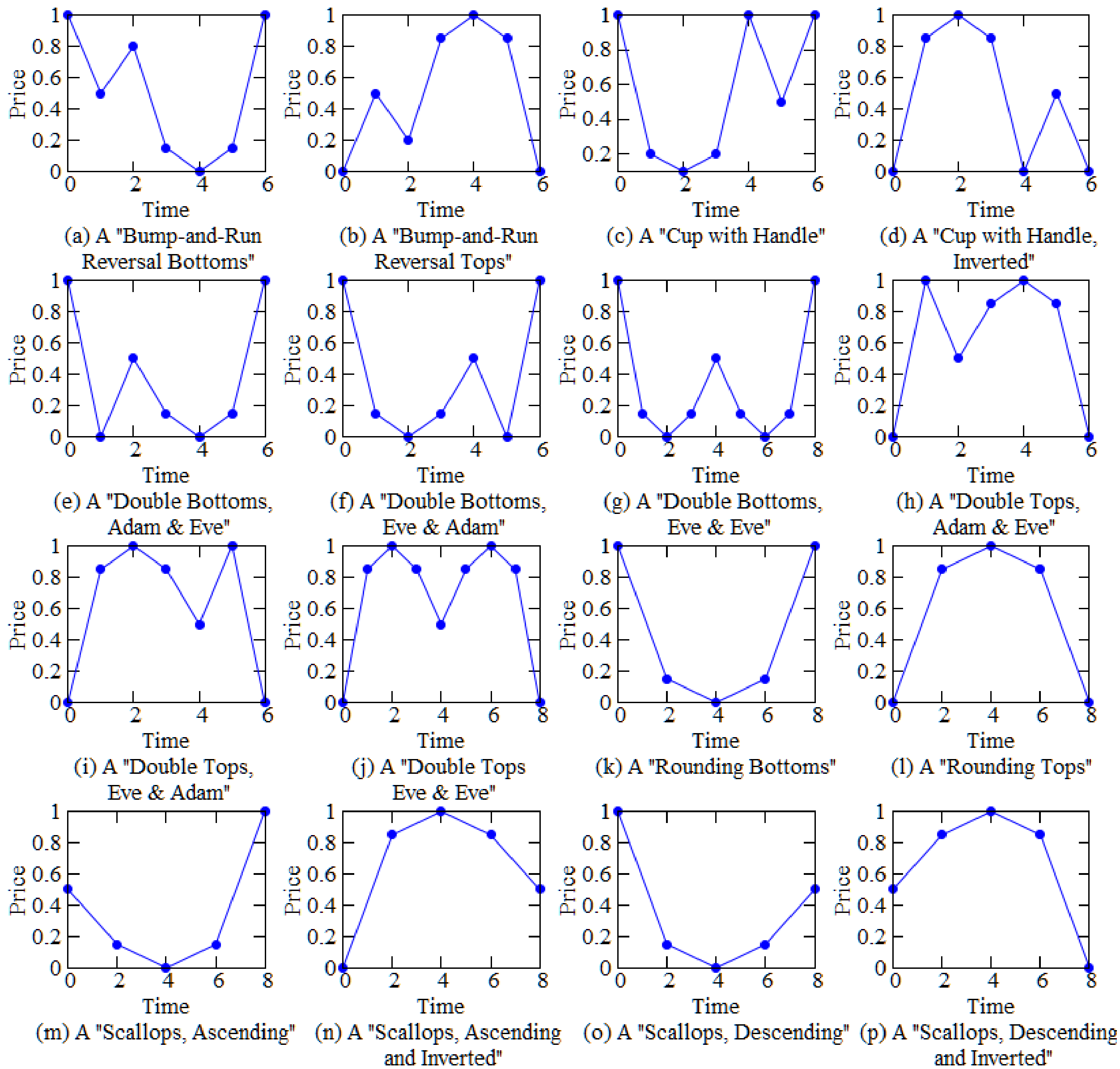

Figure 1. Examples of the curve-shaped chart patterns in C3.

Segmentation is an important pre-processing step for classification tasks in time series analysis. There are several segmentation methods proposed in recent years. One of the well-known approaches is Perceptually Important Points (PIP) [3] method. PIP extracts a set of key points which are called "PIPs" from the original subsequence. According to [7], Vertical Distance Perceptually Important Points (PIP-VD) performs best when it is compared to other variants of PIP. Other segmentation approaches are Turning Points (TP) method [4], Piecewise Aggregate Approximation (PAA) method [6], and Piecewise Linear Approximation (PLA) method [5]. PLA can be obtained through top down, bottom-up or sliding window method. However, these segmentations methods do not perform well for curve-shaped chart patterns in C3 especially in time series with high fluctuation. An example of a subsequence containing a curve is shown in Figure 2. Although this subsequence contains a curve, existing methods such as PIP-VD, PLA, and PAA cannot properly pre-process it. Specifically, simple straight-line segments commonly considered during segmentation process are insufficient in approximating curve-shaped segments. As a result, curves within the subsequence are lost or smoothed out into straight lines after the segmentation process. This situation is depicted in Figure 2 (b), (c), and (d). In these figures, the subsequences produced by PIP-VD, PLA, and PAA segmentation methods are no longer resemble to the original curve shown in Figure 2 (a).

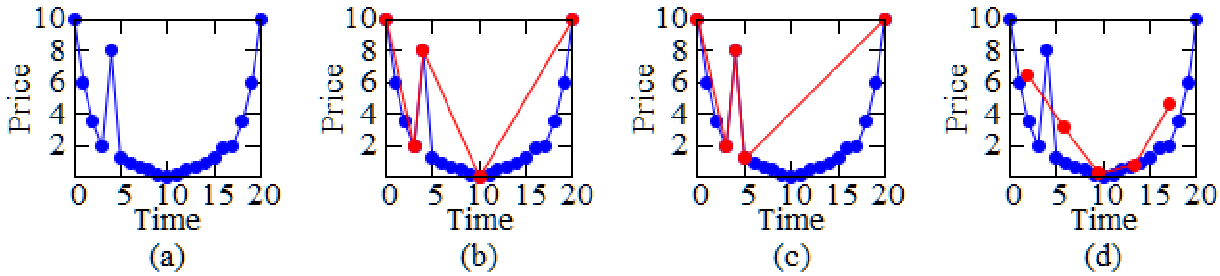

Figure 2. (a) An example of a curve-shaped subsequence. The blue line represents the input time series. The red line represents the segmentation results of (b) PIP-VD, (c) PLA, and (d) PAA.

To alleviate this problem, in this paper, we propose three novel segmentation methods to classify curve-shaped chart patterns. These methods are:

1. Segmentation method based on sliding window and direct least squares fitting of ellipses (SE),
2. Segmentation method based on turning points, sliding window, and direct least squares fitting of ellipses (TSE), and
3. Segmentation method based on PLA-bottom up, sliding window, and direct least squares fitting of ellipses (PSE).

The above segmentation methods are designed to correctly identify the existence/absence of curves in subsequences. In addition, in the subsequent sections of this paper, we further integrate these segmentation methods with rule-based pattern classification approach to further improve the accuracy in classification of curve-shaped patterns.

In financial trading, "recently appeared patterns" are commonly used for forecasting price trends in stock markets. One of the common characteristics of recently appeared patterns is that the rightmost points of these patterns are always anchored to current (today) date/time point and from that anchor point, the patterns extend to the left of the time series. The length of the recently appeared patterns may span from one week to 6 months depending on the size of the pattern the traders is trying to identify. In the context of technical analysis, "online classification of chart patterns" is a problem of locating recently appeared patterns of specific length which rightmost point is anchored to current date (or current hour in the case of intra-day trading). In order to classify recently appeared patterns, in this paper, we propose a new algorithm called Accelerating Classification with Prioritized Rules (ACPR). The main objective of ACPR is to accelerate the online classification of chart patterns in time series from real-time data streams from stock markets. In this algorithm, we prioritized the rules for classification based on their importance in identifying the key features/shapes of the patterns. With prioritized rules, we can significantly accelerate the online pattern matching process by early abandoning of fruitless comparisons.

There are 16 curve-shaped chart patterns in C3. In this paper, we further divide them into two categories.

- Patterns in category A contain only one shape which is either a U or an inverted U Shape. There are 6 patterns belong to category A. They are "Rounding Bottoms", "Rounding Tops", "Scallops, Ascending", "Scallops, Ascending and Inverted", "Scallops, Descending" and "Scallops, Descending and Inverted".
- Patterns in category B contain two segments. One of the segments is either a U or an inverted U shape. The second segment is either a U (or an inverted U shape) or a V (or an inverted V shape). There are 10 chart patterns belong to class B. They are "Bump-and-Run Reversal Bottoms", "Double Bottoms, Adam & Eve", "Bump-and-Run Reversal Tops", "Double Tops, Adam & Eve", "Cup with Handle", "Double Bottoms, Eve & Adam", "Cup with Handle, Inverted", "Double Tops, Eve & Adam", "Double Bottoms, Eve & Eve", and "Double Tops, Eve & Eve".
- In this paper, we select two representative patterns "Cup with Handle" and the "Double Bottoms, Eve & Eve" from each category for the experiments.

The rest of the paper is organized as follows. In Section 2, we briefly review the related work. In Section 3, we review the existing segmentation methods and the Direct Least Squares Fitting of Ellipses algorithm. Based on these algorithms, we propose three new segmentation methods for classification of 16 curve-shaped chart patterns in Section 4. In Section 5, we propose an algorithm for Accelerating Classification with Prioritized Rules (ACPR). In Section 6, we report the experimental results based on the datasets from Hong Kong stock market. In Section 7, we sum up our findings and discuss the future research directions.

2. Related Work

In financial trading, two types of analysis method are commonly used. The first method is called the "fundamental analysis" and the second method widely known as "technical analysis". These methods are used by analysts to analyze future price fluctuations. In the fundamental analysis, analysts predict the price trend based on the company's financial status and the latest situations effecting the social and political environment. As for the technical analysis, analysts predict the price trend based on the historical daily prices and the trading volumes. In both types of analysis, time series is usually used to represent the historical data.

According to [1], there are 53 chart patterns commonly used in technical analysis. Appearance of chart patterns in financial time series is commonly treated as a signal. There are two types of approaches for detecting chart patterns from financial time series. The first approach uses a similarity score (distance) to compare the query pattern (subsequence) with a template pattern. The second approach classifies patterns based on pre-trained models.

2.1 Distance based approaches for time series classification

In the first approach, to detect the presence of a pattern in a given time series, pattern matching algorithms need to calculate the similarity between a subsequence and a chart pattern template. Similarity calculation often uses Euclidean Distance (ED) approach to compare the subsequence with the template. Dynamic time warping (DTW) and DTW-D [8] are also commonly used for similarity calculation. DTW can map one data point in a time series to more than one data point in other time series.

In Rule-based (RB) [7], Template-based (TB) [7] and Hybrid (HY) [9] approaches, the query segment should contain the same number of data points as the template pattern. RB identifies the segments by a set of predefined rules. In this paper, we focus on RB approach for pattern matching due to its simplicity, comprehensiveness, and efficiency properties in performing fast pattern classification. TB calculates the temporal and amplitude distances of the points to measure the similarity between the pattern template and the query segment. HY ranks the segment based on the Spearman's correlation coefficient. In these methods, segmentation is often needed to reduce the data points in a time series. Several segmentation approaches were proposed for time series classification in literature. Among these approaches, PIP [3], PLA [5], TP [4], and PAA [6] methods are commonly used to reduce the data points in a time series.

In RB, TB and Hybrid (HY) approaches, the length of the subsequence must be equal to the length of a template which is usually defined in advance based on the pattern definition. To achieve this objective, segmentation methods are used to decrease data points. For example, Perceptually Important Points (PIP) method [3] extracts important points which are called "PIPs" from the input time series. In [10], a variant of PIP segmentation method called Perceptually Important Point with Binary tree (PIP-Btree) was proposed. PIP-Btree takes advantage of a standard binary tree for improving the efficiency without compromising the accuracy of PIP method. PIP-Btree supports self-updating when a new data point arrives in a streaming time series. In [11], Chen et al. proposed a PIP-based evolutionary approach for time-series segmentation and pattern discovery. In their approach, a time series was divided into segments by a genetic process. Next, it used the PIP approach to find appropriate critical points and slopes to represent each segment. According to [11], PIP-based evolutionary approach was able to solve the problems of information loss, distortion of segments, and the generation of meaningless patterns. In [7], Fu et al. compared three variants of PIP algorithms which include Euclidean distance (PIP-ED), perpendicular distance (PIP-PD) and vertical distance (PIP-VD). Fu et al. reported that PIP with Vertical Distance (PIP-VD) achieved best performance among all variants.

In [9], Zhang et al. proposed an online pattern-matching scheme based on sliding window. In their approach, a modified Perceptually Important Point (PIP) was integrated with a sliding window. For classification, Spearman's Rank Correlation Coefficient was used to classify the query patterns. Experiment results revealed that proposed scheme outperforms Euclidean Distance based method and Slope Distance based method in differentiating the patterns including distorted patterns.

A formal approach for financial time series chart pattern classification was proposed in [2]. The proposed approach contained three steps. In the first step, subsequences were extracted using a sliding window. Next, PIPs were extracted by PIP-VD from the subsequences. In the third step, RB method was used to classify the chart pattern. Although the proposed approach was effective in classifying majority of the patterns, experiment results revealed that PIP-VD with the sliding window cannot be effectively used to classify the curved-shaped chart patterns.

The other methods that extract important points form the time series for segmentation are PLA (Piecewise Linear Approximation) [5] and the TP (Turning Points) [4]. In [4], a segmentation method based on Turning Points (TPs) was proposed. The proposed method selects TPs from the financial time series in question based on their degree of importance. In [4], an Optimal Binary Search Tree (OBST) was used to store the selected TPs for reconstructing the reduced sample time series. Experiment results showed the proposed method was able to maintain the shape of the original time series and preserve more trends while maintaining the minimum average retrieval cost. Also in [4], Si et al. investigated the performance of TP, PIP, and PLA in preserving the trends and fluctuation of time series. Experiment results revealed that PLA approach produced the least amount of error and the fewest trends whereas TP approach preserved more trends than two other approaches.

Recent studies have found that different segmentation methods can affect the pattern matching results. In [12], the efficiency and the effectiveness of TB, RB and PAA when they are paired with PIP-VD were investigated. According to [12], the performance of TB and RB were superior to the PAA-based method. It was also reported that the TB pattern matching method was more effective than RB pattern matching method, while the RB method can better describe the template pattern. In [13], curve fitting and trend preservation performance of the PIP and TP algorithms were evaluated. According to [13], PIP produces less error than TP. However, TP can preserve more trends than PIP. They found that PIP approach was designed to keep the overall shape of the time series whereas the objective of TP was to extract as many trends as possible from the time series. In addition, it was reported that TPs were more sensitive to the movement of the stock than PIPs. Besides, PIPs tend to maintain the shape of the curve while TPs can be used to identify the change in the trend of the curve.

In addition to the approaches which extract important points from the time series, some methods use different form of representations for time series segmentation. For example, Symbolic Aggregate Approximation (SAX) [14] was proposed to transform a time series into a series of symbols. SAX first obtains segments by using PAA and then uses symbols for representation. In [15], Wu et al. proposed a novel trend-based segmentation method called TBSM for financial time series forecasting. In their approach, Wu et al. modified the Piecewise Linear Representation (PLR) (also known as PLA) segmentation using the trend tendency in a specific time period. Next, Support Vector Regression (SVR) was used to capture the trading knowledge using the trading signals derived from these trading trends.

2.2 Model-based approaches for time series segmentation

In addition to the segmentation methods which reply on extraction of important points from the input time series, a number of other approaches for segmentation were also proposed in literature. In [16], Palivonaite et al. proposed an adaptive algebraic level-set segmentation algorithm for financial time series. The proposed algorithm is based on the algebraic one step-forward predictor with internal smoothing. In their approach, Particle Swarm Optimization (PSO) algorithm was used for the detection of a base algebraic fragment of the time series. Finally, a combinatorial algorithm was used to detect intervals where predictions were lower than a predefined level. In [17], Yin et al. proposed a segmentation algorithm based on Jensen–Shannon divergence. The proposed algorithm can be used to segment the sequence into statistically significant as well as compositionally homogeneous patches. The proposed approach was evaluated on US and Chinese stock indices. In [18], Wang et al. used a complexity measure based on the entropic segmentation called sequence compositional complexity (SCC) for segmenting financial time series. The main objective of their approach was to measure the complexity of a sequence by dividing it into segments to maximize the compositional divergence between the compositional domains. In [19], Durán-Rosal et al. introduced a time series segmentation based on a hybrid genetic algorithm and a clustering technique. The main objective of their approach was to automatically group common patterns from financial time series and addressed the problem of identifying stock market prices trends.

Pattern matching is a key step in the analyzing time series. Second category of methods for classification of chart patterns in time series involves training models with synthetic datasets and using these trained models for detection. The methods for classification of chart patterns in this category include Hidden Semi-Markov Model (HSMM) [20], Adaptive Neuro-Fuzzy System [21], and Support Vector Machine [22]. In [20], a hidden semi-Markov model was trained with a Viterbi algorithm to detect chart patterns. The aim of their approach was to simplify the traditional way of training an HSMM as well as to reduce potential biases in parameter initialization. The model was tested on a set of templates selected from 53 chart patterns. Experiments on a synthetic dataset revealed that the proposed approach achieved the highest average accuracy and recall among TB, RB, ED, DTW, SVM pattern matching approaches. In [21], Wan et al. proposed a method to calculate the thresholds for pattern matching with a trained ANFIS model. The effectiveness and efficiency of the ANFIS model were compared with SVM, TB, RB, ED, and DTW on both synthetic datasets and real datasets. Experimental results showed that the ANFIS model was effective in classifying different chart patterns when compared with other pattern matching approaches. In [22], Gong et al. extended the subsequence search algorithm UCR Suite [23] with a Support Vector Machine (SVM) to train a classifier for chart pattern-matching. The experimental results revealed that the proposed approach achieved significant improvement over 6 existing subsequence-matching approaches (TB, RB, Hybrid, ED, DTW, UCRS) in terms of speed and accuracy. In [24], a two-layer feed-forward network with tan-sigmoid transfer functions in both the hidden layer and the output layer was used for classification of Head & Shoulders pattern. Experiment results revealed that the proposed network outperformed the RB method.

From the above discussion, we can observe that a wide range of segmentation methods for financial time series segmentation were proposed in recent years. These methods can be categorized into distance-based and model-based algorithms. However, to the best of authors' knowledge, there is no reported work on extracting important points from financial time series for identification of curve-shaped chart patterns. Although exiting segmentation methods could be applied to extract the curve-shaped segments from the time series, the extracted segments may not contain the important points which are essential in preserving the shape of the curves. To address this problem, in this paper, we propose three novel methods for segmentation and classification of curve-shaped chart patterns based on Direct Least Squares Fitting of Ellipses.

3. Preliminaries

3.1 Definitions and functions.

In this section, we describe the definitions and functions used for the proposed algorithms.

Definition 1. $\mathbf{T}[(\boldsymbol{x_o}, \boldsymbol{y_o}), \dots (\boldsymbol{x_z}, \boldsymbol{y_z})]$ is a time series that contains a sequence of data points. x_i and y_i are the time and price coordinate of a data point.

Definition 2. $\mathbf{S}[(\boldsymbol{x_o}, \boldsymbol{y_o}), \dots (\boldsymbol{x_z}, \boldsymbol{y_z})]$ is a subsequence of *T*. *S* can be considered as a query segment for classification. x_i and y_i are the timestamp and price of the important data point s_i. The first important point is *0* and the last important point is *n*. s_i is used to represent (x_i, y_i). $\mathbf{S}[s_i,\ s_j]$ is a subsequence between s_i and s_j of T.

Definition 3. $\mathbf{P}(\boldsymbol{x}, \boldsymbol{y})$ is a new data point to a time series. x and y are the timestamp and price coordinate of a data point.

Definition 4. Anchor is the left or the right boundary of the first segment, usually the first or the last data point of a time series.

Definition 5. WS is the size of a sliding window.

Definition 6. Min Len is the minimum length of a sliding window.

Definition 7. Max Len is the maximum length of a sliding window.

Function 1. Function $\boldsymbol{len}(S)$ returns the length of the subsequence S.

Function 2. Function $\boldsymbol{get\_rules}(pattern, i)$ returns rules by the pattern name and priority i. The rules with related priority of the 16 patterns are listed in Table 2 to Table 17.

Function 3. Function $\boldsymbol{slope}(s_i, s_j)$ returns the slope of the line connecting the two points s_i and s_j. $slope\left(s_i, s_j\right) = \frac{y_j - y_i}{x_j - x_i}$.

Function 4. Function $\boldsymbol{add}(P)$ adds a new data point at the end of a time series.

Function 5. Function $\boldsymbol{create\_segment}(S)$ creates subsequence S.

Function 6. Function $\boldsymbol{rule\_selecting}(pattern, S)$ uses rules to select 16 curve-shaped chart patterns with $\boldsymbol{slope}(\boldsymbol{s_i}, \boldsymbol{s_j})$.

Function 7. Function $\boldsymbol{rule\_selecting\_ellipse}(pattern, S)$ uses rules to select 16 curve-shaped chart patterns based on $\boldsymbol{direct\_fit\_ellipse}(S)$.

3.2 Existing segmentation methods

Segmentation is commonly used as a preprocessing step for reducing the number of data points and to smooth out a time series. The Perceptually Important Points (PIP) method [3] is a commonly used segmentation method. There are several variants of PIP reported in literature. They are perpendicular distance PIP, PIP-VD and PIP-ED. Fu et al. [7] reported that the PIP-VD method is superior in terms of its efficiency and effectiveness. Therefore, in our experiments, we choose PIP-VD as a segmentation method. In a time series *T*, the first two PIPs are the first and the last data point. The data point in *T* with maximum vertical distance to the line joined by the first two PIPs is the third PIPs. The data point in *T* with maximum vertical distance to the line joined by either between the first and second PIPs or between the second and the last PIPs is the fourth PIPs. This process repeats until the length of the subsequence equals to the input sequence *Q*. The distance from the next PIPs $(x2, y2)$ to the two adjacent $(x1, y1)$ and $(x3, y3)$ can be measured by Eq. (1). The pseudo code of PIP is given in Algorithm 1 (Appendix). In Figure 3, we show an illustration of the selection process of five PIPs.

$$Dis = |\hat{y}2 - y2| = \left|y1 + \frac{(y3-y1)\times(x2-x1)}{x3-x1} - y2\right|. \qquad (1)$$

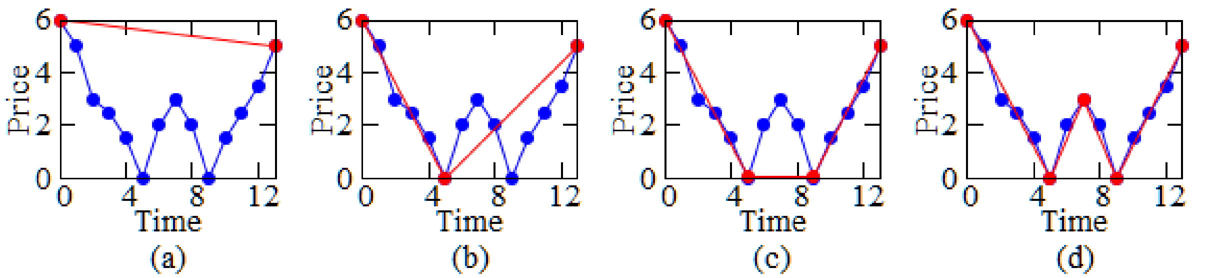

Figure 3. An illustration of PIP method. In (a)-(d), five PIPs are selected. The blue line represents the input time series and the red line in (d) is the final segmentation result.

In Piecewise Linear Approximation (PLA) [5], a time series T is approximated by *K* straight lines with length *n*. The pseudo code of PLA-Bottom-up (PLA-BU) is described in Algorithm 2. In this algorithm, a set of segments in the first WHILE loop represents a time series. In the second WHILE loop, the cost of merging the adjacent segments are calculated. In the third WHILE loop, the algorithm combines the adjacent segments which have the lowest merge-cost until the minimum merge cost exceeds the threshold. The merging process repeats until the number of the salient points in the time series is equal to the number of data points in the chart pattern. An illustration of the selection of five points by PLA is shown in Figure 4.

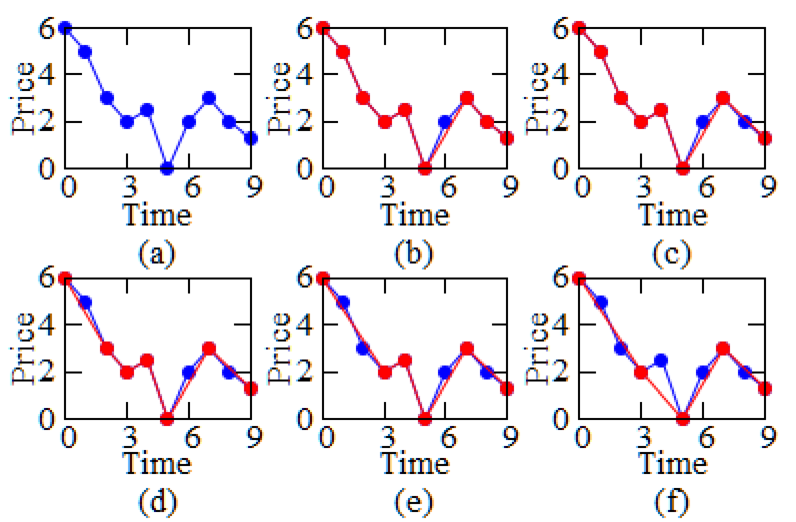

Figure 4. An illustration of the PLA-BU method. (a) Input time series in blue colour. In (b)-(f), the data points in the time series are removed gradually for selecting five points. The red line in (f) is the final segmentation result.

In Piecewise Aggregate Approximation (PAA) [6], the compressed time series T' is used to represent a time series T of length N (i.e. $T' = (x'_0, \ldots, x'_N)$ represents $T = (x_0, \ldots, x_n)$). Time series T is divided into N equal-sized parts, and each part is represented by the mean value of the data points. The i^{th} element of T' can be calculated by using Eq. (2).

$$x'_i = \frac{N}{n} \sum_{j=s_i}^{e_i} x_j \quad (2)$$

where s_i and e_i denote the start and end point of the i^{th} part. Figure 5 shows the selection process of five points with PAA.

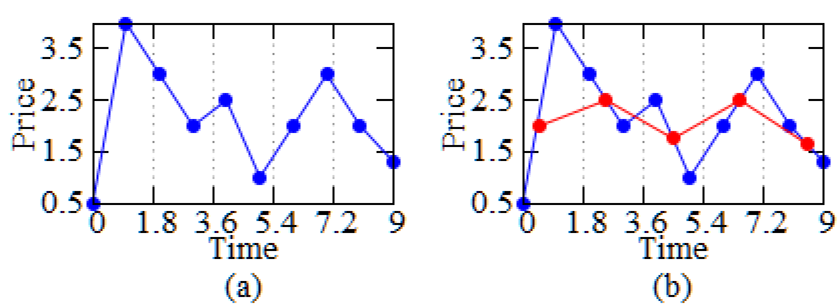

Figure 5. An illustration of PAA method. (a) Input time series in blue colour (b) Each part is then represented by a red point which is calculated from the mean of the blue points in that part. The red line is the segmentation result of PAA.

The Turning Points (TP) method [4] includes the identification phase and the evaluation phase. In this paper, TPs represents the local minimum and maximum points on a time series. The pseudo code of TP is given in Algorithm 3. Figure 6 shows an illustration of the selection of seven TPs.

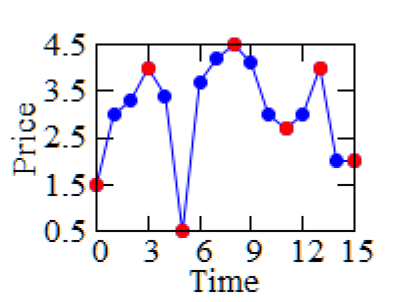

Figure 6. An illustration of TP method. Input time series is depicted in blue colour and the selected TPs are represented as seven red points.

The above segmentation methods play a crucial role in pattern matching in financial time series since the template-based (TB) [7], hybrid (HY) [25], and rule-based (RB) [7] pattern matching methods require the input sequence to be pre-processed to reduce the data points until its length equals to the length of the query pattern.

As illustrated in previous sections, PIP-VD, PLA-BU, PAA and TP are mainly designed to deal with straight-line segments connected by important points in the time series. Therefore, these methods do not perform well for curve-shaped chart patterns in C3. To alleviate this problem, we propose new segmentation approaches which are based on detection of Ellipses. In this section, we introduce the an algorithm called Direct Least Squares Fitting of Ellipses [26].

3.3 Direct Least Squares Fitting of Ellipses

In mathematics, an ellipse is the surrounding two focal points curve in a plane and the sum of the distances to the two focal points for every point on the curve is constant. The equation of the ellipse can be defined as $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$. An illustration of the ellipse definition is depicted in Figure 7.

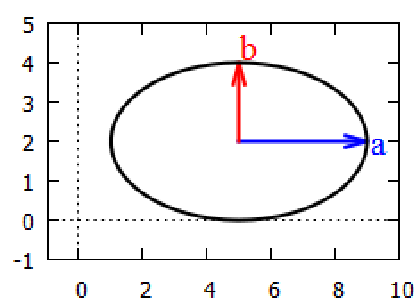

Figure 7. An illustration of the ellipse definition. a is the Semi-major Axis and b is the Semi-minor Axis. The equation of a standard ellipse centered at the origin is $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Direct Least Squares Fitting of Ellipses algorithm [26] can be described in 3 steps. The first step is the calculation of the coefficients *a, b, c, d, e,* and *f* of the ellipse. The second step is the calculation of the center of ellipse (x_0, y_0), semi-axis lengths a' and b', and the counterclockwise rotation angle $\emptyset$ from the x-axis to the major axis of the ellipse. The third step is the calculation of the lowest point y_l by the tangent line and the point of tangency, and the curve depth d_1, d_2, d_3, d_4 of the ellipse.

The pseudo code of the Direct Least Squares Fitting of Ellipses [26] is given in Algorithm 4. The Flowchart for Algorithm 4 is shown in Figure 8. In this paper, we implemented a program for Direct Least Squares Fitting of Ellipses algorithm based on the FitEllipse program[1].

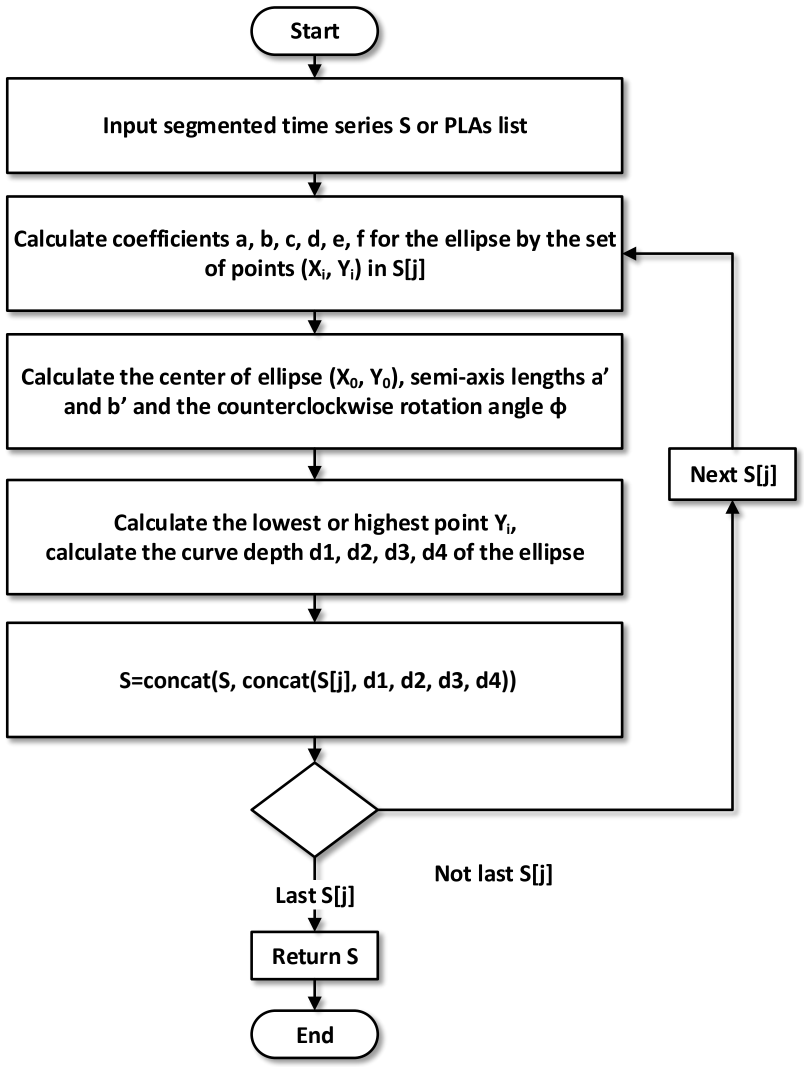

Figure 8. Flowchart for Direct Least Squares Fitting of Ellipses algorithm.

3.4 Sliding window for extracting subsequences

[1] https://github.com/mdoube/BoneJ/blob/master/src/org/doube/geometry/FitEllipse.java.

The SW algorithm uses the leftmost boundary (Anchor) of the given time series. It is usually the first data point of a time series and *i* is the length of the segment from the range [*min*, *max*] which is set by the trader. The *min* and *max* correspond to the minimum and maximum length (width) of the pattern going to be identified from the time series. The lengths vary from one pattern to another and they are usually input by the trader based on his/her preference. Starting from the leftmost anchor point, the SW algorithm first extracts subsequences of length *i* from the giving time series by gradually shifting the window to the rightmost point of the time series. This process is shown in Figure 10. Once the window reaches the rightmost point of the time series, the length of the segment *i* is increased by 1 and the whole process is repeated again. The process continues until the segment length is equal to maximum length (*max*). The pseudo code of Sliding Window algorithm is given in Algorithm 5. The flowchart of the algorithm is depicted in Figure 9.

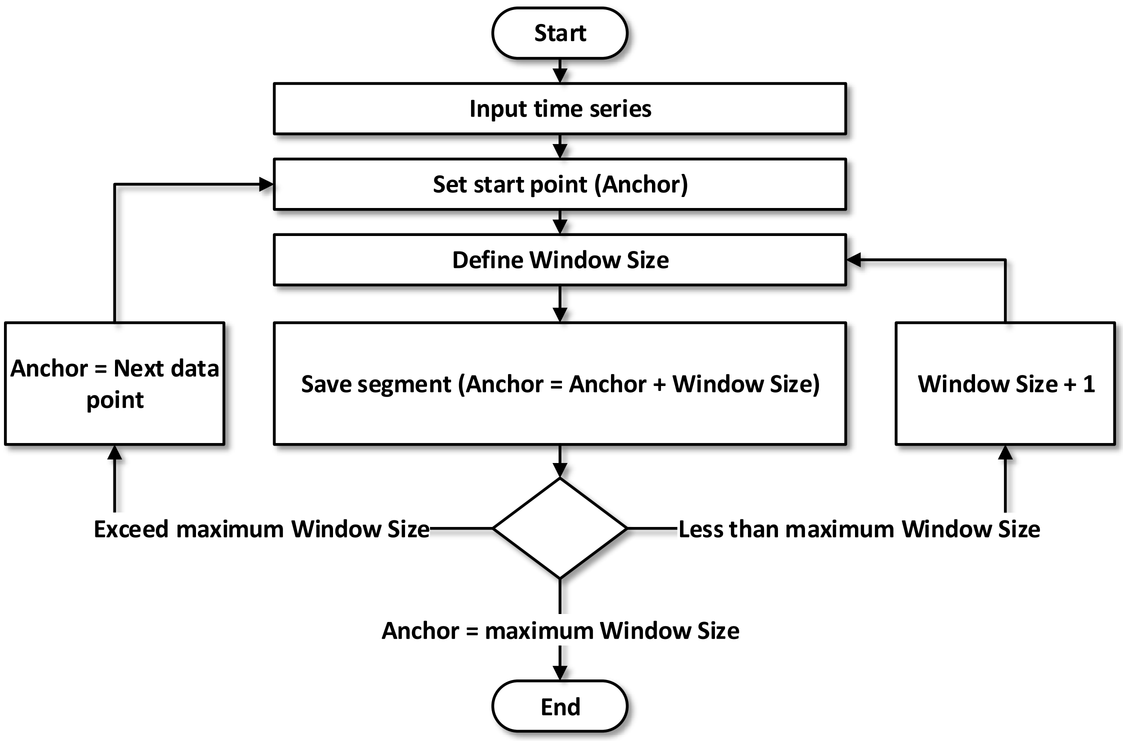

Figure 9. Flowchart for Sliding Window algorithm.

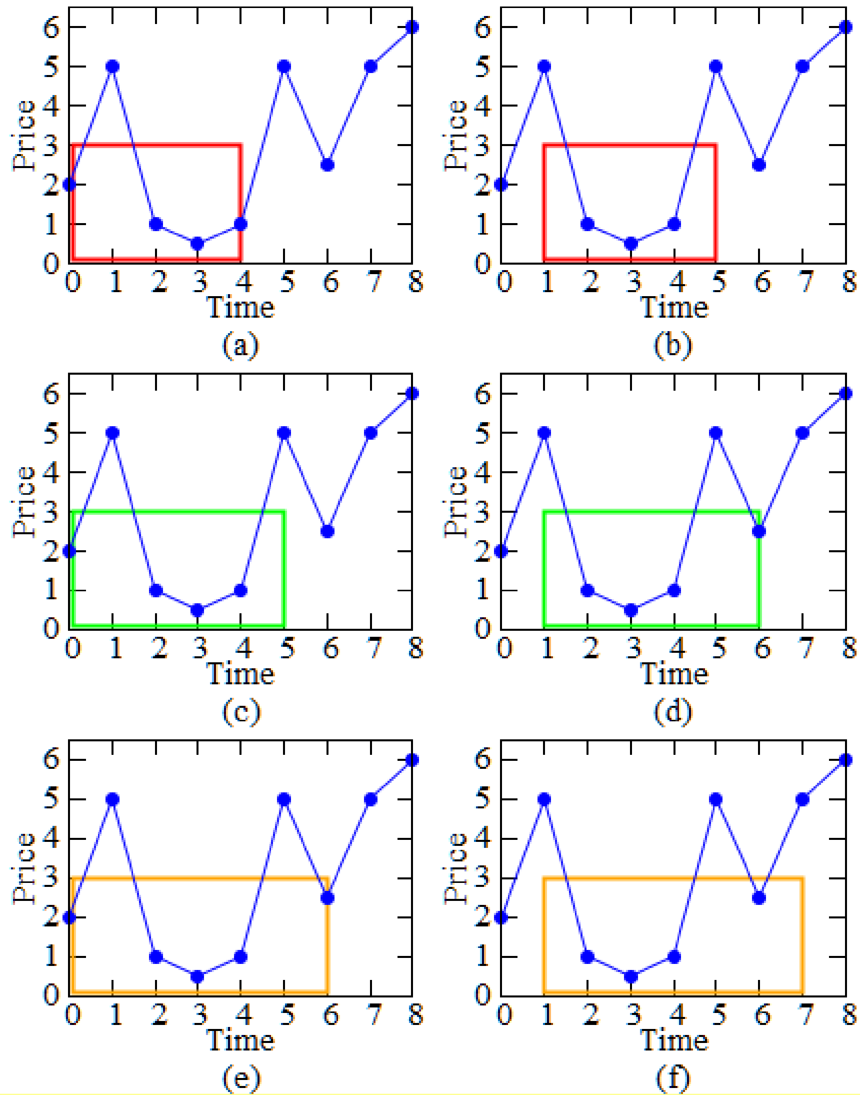

Figure 10. An illustration of SW. Input time series is depicted in a blue line and the sliding window is depicted as a colored rectangle (a) Sliding window of size 4 and left boundary anchored at $x = 0$ (b) Shifting the window to the right (c) Sliding window of size 5 and left boundary anchored at $x = 0$ (d) Shifting the window to the right (e) Sliding window of size 6 and left boundary anchored at $x = 0$ (f) Shifting the window to the right.

4. Proposed segmentation methods for curve-shaped chart patterns

Segmentation is an important pre-processing step for classification of chart patterns. There are several segmentation methods proposed in recent years. These methods include Perceptually Important Points (PIP) method, Turning Points (TP) method, Piecewise Aggregate Approximation (PAA) method, and Piecewise Linear Approximation (PLA) method. However, these segmentations methods do not perform well for 16 curve-shaped chart patterns in category C3 since important point extraction procedure in these methods are not designed to preserve the shape of the curves in the output subsequences. As a result, curves within the subsequences are lost or simply smoothed out into straight lines after the segmentation process. In other words, any curves in the input time series could be lost after the segmentation procedure.

To alleviate this problem, in the following sections, we propose three new segmentation methods to classify the curve-shaped chart patterns from C3. These methods are SE, TSE, and PSE.

1. Segmentation method based on sliding window and direct least squares fitting of ellipses (SE).
2. Segmentation method based on turning points and sliding window and direct least squares fitting of ellipses (TSE).
3. Segmentation method based on the bottom-up piece wise linear approximation and sliding window and direct least squares fitting of ellipses (PSE).

In these methods, we combine the idea of a sliding window, existing segmentation methods for extracting important points (TP, PLA), and direct least squares fitting of ellipses approach. To evaluate the effectiveness of the proposed methods, extensive experiments are conducted in the Experiment section to compare against a baseline method called SP which is based on a sliding window and PIP-VD method. In the following sections, we detail the high-level descriptions of the proposed algorithms.

4.1 Segmentation method based on sliding window and direct least squares fitting of ellipses (SE)

SE approach combines the sliding window and direct least squares fitting of ellipses algorithms. SE segmentation method consists of three main steps:

1. In the first step, subsequences are extracted from the given time series using a sliding window.
2. In the second step, Direct Least Squares Fitting of Ellipses algorithm [26] is applied to calculate the parameters necessary for identifying the potential ellipses.
3. In the third step, rules defined in [2] for each pattern as well as the parameters calculated in step 2 are used to classify the patterns.

The flowchart of SE algorithm is depicted in Figure 11. The pseudo code of SE is given in Algorithm 6.

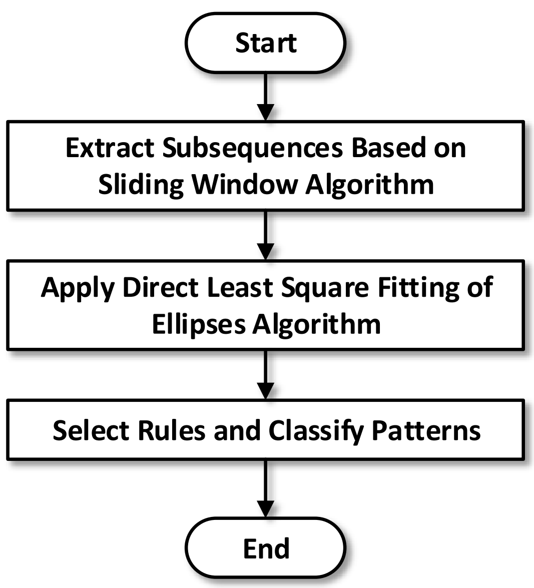

Figure 11. Flowchart for SE algorithm.

4.2 Segmentation method based on turning points and sliding window and direct least squares fitting of ellipses (TSE)

TSE approach combines the turning points identification, sliding window for subsequence extraction, and direct least squares fitting of ellipses algorithm. TSE segmentation algorithm consists of four main steps:

1. First, turning points (TP) are identified from the given time series by the TP approach [4].
2. In the second step, subsequences are extracted from the extracted TP using a sliding window.
3. In the third step, Direct Least Squares Fitting of Ellipses algorithm [26] is applied to calculate the parameters necessary for identifying the potential ellipses.
4. In the fourth step, rules defined in [2] for each pattern as well as the parameters calculated in step 3 are used to classify the patterns.

The flowchart of TSE algorithm is depicted in Figure 12. The pseudo code of TSE is given in Algorithm 7.

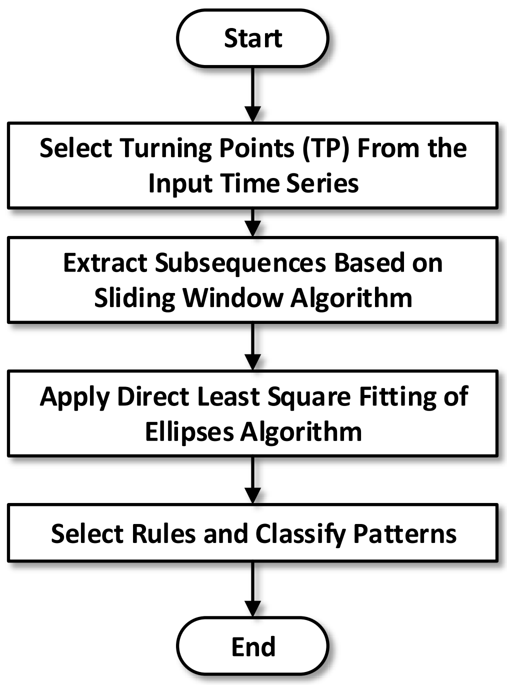

Figure 12. Flowchart for TSE algorithm.

4.3 Segmentation method based on the bottom-up piece wise linear approximation and sliding window and direct least squares fitting of ellipses (PSE)

PSE approach combines the bottom-up piece wise linear approximation method, sliding window subsequence extraction and direct least squares fitting of ellipses. PSE segmentation algorithm consists of following four main steps:

1. In the first step, subsequences are extracted from the given time series using a sliding window.
2. In the second step, PLA-BU (see Algorithm 2) is applied to extract PLAs from the subsequences.
3. In the third step, Direct Least Squares Fitting of Ellipses algorithm [26] is applied to calculate the parameters necessary for identifying the potential ellipses.
4. In the fourth step, rules defined in [2] for each pattern as well as the parameters calculated in step 3 are used to classify the patterns.

The flowchart of PSE algorithm is depicted in Figure 13. The pseudo code of PSE is given in Algorithm 9.

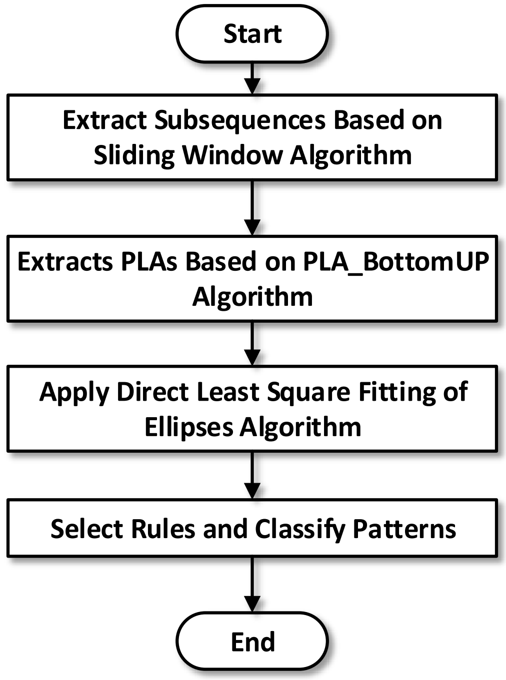

Figure 13. Flowchart for PSE algorithm.

4.4 Baseline approach: segmentation with sliding window and vertical distance perceptually important points (SP)

To evaluate the performance of the proposed segmentation methods, we implanted a baseline approach called SP based on the Sliding Window (SW) algorithm and vertical distance perceptually important points (PIP-VD). In SP, a sliding window is first used to extract subsequences from the given time series.
Next, vertical distance perceptually important points (PIP-VD) approach is used to select important points from the extracted subsequences. In SP, three PIPs are used to detect whether there is a curve in the extracted subsequences. The slopes of the two lines forming the curve should be in a specified range. Specifically, the slope of the line connecting the first and the second salient points should be less than 0 (i.e. tan 0°) and greater than −1 (i.e. tan 135°). At the same time, the slope of the line connecting the second and third salient points must be less than 1 (i.e. tan 45°) and greater than 0 (i.e. tan 0°). SP segmentation algorithm consists of following steps:

1. In the first step, subsequences are extracted from the given time series using a sliding window.
2. Next, PIP-VD is used to extract PIPs from the subsequences. For example, for the "Cup with Handle" pattern (see Figure 1(c)) and "Double Bottoms, Eve & Eve" pattern (see Figure 1(g)), 7 PIPs and 9 PIPs are extracted from each segments.
3. In the third step, rules defined in [2] for each pattern are used to classify the patterns.

The flowchart of SP is depicted in Figure 14. The pseudo code of SP is given in Algorithm 9.

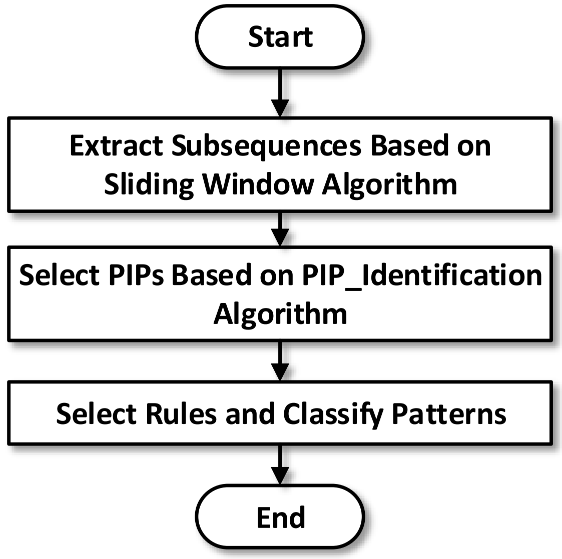

Figure 14. Flowchart for SP algorithm.

4.5 Complexity

In summary, SP contains three steps. The first step extracts subsequences from the given time series by a sliding window. The second step extracts PIPs based on PIP-VD method. The third step uses a rule-based method to classify the chart patterns. The method's overall complexity is $O(n^2)$.

SE contains three steps. The first step extracts subsequences from the given time series by a sliding window. The second step uses direct least squares fitting of ellipses. The third step uses a rule-based method to classify the chart patterns. The method's overall complexity is $O(n)$.

TSE contains four steps. The first step extracts TPs. The second step extracts subsequences from the TPs by using a sliding window. The third step uses direct least squares fitting of ellipses. The fourth step uses a rule-based method to classify the chart patterns. The method's overall complexity is $O(n)$.

PSE contains four steps. The first step extracts subsequences from the given time series based on a sliding window. The second step extracts PLAs by PLA-BU method. The third step uses direct least squares fitting of ellipses. The fourth step uses a rule-based method to classify the chart patterns. The method's overall complexity is $O(n^2)$.

5. Accelerating Classification with Prioritized Rules (ACPR)

Given the interconnected nature of stock markets and commodity trading environment, real-time monitoring of price data and relevant technical indicators become increasingly important for financial analysts. Therefore, it is crucial to design efficient pattern classification methods which are suitable for processing real-time streaming stock market data. In traditional Rule-based (RB) classification method based on PIP segmentation, rules defined for the pattern are sequentially checked for the segment. The segment is classified as a positive pattern when all of these rules are completely satisfied. However, in majority of the cases, a high proportion of these rules cannot be satisfied. Therefore, in traditional RB method based on PIP segmentation, the classifier may spend considerable amount of time on validating conditions which are later found to be false. Such sequential validating process is extremely time consuming and inefficient. To alleviate this problem, in this paper, we propose an algorithm called Accelerating Classification with Prioritized Rules (ACPR). In ACPR, rules defined for each pattern are prioritized based on their level of importance in identifying the pattern. To achieve fast classification outcomes, validation with the prioritized rules is immediately abandoned when one of the rules is unfulfilled. In the experiment section, we compare the execution time of SP, SE, TSE and PSE with or without using ACPR.

Recall that there are 16 curve-shaped chart patterns in C3 category. In ACPR, we divide the rules of the 16 patterns into 6 categories. We set the priority for each category from 1 to 6. These categories are listed in Table 1. The idea behind these categories are originated from the five colored sections of a pattern shown in Figure 15. G1 is the "leading section" before a pattern appears (the blue rectangle). G2 is the "confirmation section" after the pattern (the red rectangle). G3 is the "lips section" of an inverted U or a U shape (the two green rectangles). G4 is the divided parts of the pattern (the two purple rectangle). For example, in the "Cup with Handle" pattern, the left part is a U shaped cup and the right part is a V shaped handle. G5 is the "middle section" of the pattern (the yellow rectangle). The "middle section" starts at the lowest point of the left cup and ends at the lowest point of the right handle.

Categories	Priority
G1: Trend line before pattern rules.	1
G2: Confirmation price after pattern rules.	2
G3: Lips section of a U or an inverted U shape rules.	3
G4: Each divided parts of the pattern rules.	4
G5: Middle section of the pattern rules.	5
G6: Remaining rules.	6

Table 1. Six categories of rules and the corresponding priority.

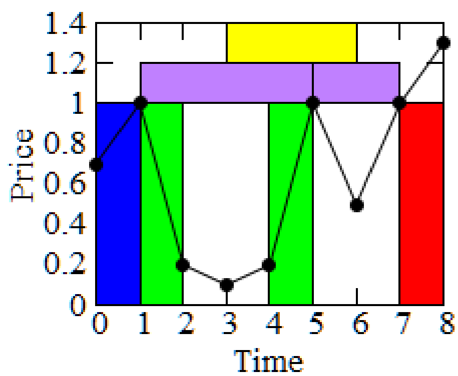

Figure 15. Five sections of a curve-shaped pattern. Sections G1-G5 are represented in five different colors.

In ACPR, three meta-rules are used for designing the priorities of the rules. They are: (a) rules covering the leading and confirmation sections are given highest priority since they can be easily identified, (b) commonly used rules among the patterns are given higher priority, and (c) rules that contain numerical values are given higher priority. All 16 curve-shaped patterns contain the rules from G1, G2 and G3. However, only 10 out of 16 patterns contain the rules which belong to G4 and G5. Therefore, the priority of G1, G2 and G3 is higher than G4 and G5. G1 and G2 cover the leading and confirmation sections of the pattern. Upon validating the corresponding rules for these sections, further rule checking can be immediately abandoned if the conditions are not satisfied. In other words, by checking the rules for G1 and G2, complex calculation for validating the curves and other conditions can be avoided. Therefore, the priorities of G1 and G2 are higher than G3. Some rules in G1 have numeric value in their conditions. For example, in "Cup with Handle" pattern in Table 4, rule R1 "Rise before pattern is at least 30%" contains a numeric value "30%". In contrast, rules for G2 require confirmation of a downward or an upward trend. In this case, it is more efficient to check the rule by using numerical value than calculating the slope of the trend lines. Therefore, the priority for G1 is higher than G2. All the rules in G6 are for recognizing curves. Therefore, they require complex calculation. Therefore, rules for G6 have the lowest priority. Prioritized rules for the 16 patterns are listed in Table 2, Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11, Table 12, Table 13, Table 14, Table 15, Table 16, and Table 17.

Rule	A "Bump-and-Run Reversal Bottoms"	Priority
R1	Prices trend upward to the pattern.	1
R2	The price becomes higher after the bump phase.	2
R3	The angle of inclination of the pattern's start point and end point trend line usually is less than 45 degrees.	3
R4	The price which pierces the trend line and the data point between them are not higher than the trend line.	
R5	The handle should last for at least 1 month.	4
R6	The angle of the tilt of the "handle-form" trend line is less than 45 degrees.	
R7	At least twice the lead-in height (the widest vertical distance from the handle-form trend line to the daily low in the handle) to the bump height (the vertical distance between the handle-form trend line and the lowest low in the bump).	5
R8	The left part is a V-Shaped handle. The right part is a U-Shaped frying pan.	6

Table 2. The priority of the rules for the "Bump-and-Run Reversal Bottoms" pattern.

Rule	A "Bump-and-Run Reversal Tops"	Priority
R1	Prices trend downward to the pattern.	1
R2	After the bump phase, the price moves lower. The price forms a downward breakout which pierces the handle-form trend line.	2
R3	The angle of inclination of pattern start point and end point trend line usually rises at about 30–45 degrees.	3
R4	The price which pierces the trend line and the data point between them are not lower than the trend line.	
R5	The handle should last at least 1 month.	4
R6	A trend line connecting the lows rises steadily which no horizontal or near-horizontal trend lines. Usually at about 30–45 degrees rises for the trend line.	5
R7	At least twice the lead-in height (the widest vertical distance from the handle-form trend line to the highest high in the handle) to the bump height (the vertical distance between the handle-form trend line and the highest high in the bump).	
R8	The left part is an inverted V-Shaped handle. The right part is an inverted U-Shaped frying pan.	6

Table 3. The priority of the rules for the "Bump-and-Run Reversal Tops" pattern.

Rule	A "Cup with Handle"	Priority
R1	"Rise" before the pattern is at least 30%.	1

R2	The price which pierces the horizontal trend line will lead to an upward breakout.	2
R3	Approximately at the same price of the cup lips.	3
R4	The price which pierces the horizontal trend line and the data point between them are not higher than the horizontal trend line.	
R5	The cup should last approximately 7–65 weeks.	4
R6	The handle should last at least 1 week.	
R7	The handle should be formed in the upper half of the cup.	5
R8	The left part is a U-Shaped cup. The right part is a V-Shaped handle.	6
R9	The U-Shaped cup depth is between 12% and 50%.	

Table 4. The priority of the rules for the “Cup with Handle” pattern.

Rule	A “Cup with Handle, Inverted”	Priority
R1	Prices trend downward to the pattern.	1
R2	The price which pierces the horizontal trend line will forms a downward breakout.	2
R3	The cup lips are approximately at the same price.	3
R4	The price which pierces the horizontal trend line and the data point between them are not lower than the horizontal trend line.	
R5	The cup should last at approximately 7–65 weeks.	4
R6	The median of the duration of the handle (from the right cup lip to the breakout) should be 40 days.	
R7	The handle should be formed in the under half of the cup.	5
R8	The left part is an inverted U-Shaped cup. The right part is an inverted V-Shaped handle.	6
R9	The height of Cup tops should be double of the height of the handle.	

Table 5. The priority of the rules for the “Cup with Handle, Inverted” pattern.

Rule	A “Double Bottoms, Adam & Eve”	Priority
R1	Prices trend downward to the pattern, which should not drift below the left bottom.	1
R2	The price must close above the confirmation line, which is equivalent to the horizontal line at the highest high point.	2
R3	The pattern’s start point and end point are approximately at the same price.	3
R4	The confirmation line which is equivalent to the horizontal line at the highest high point and the data point between them are not higher than the horizontal trend line.	
R5	The duration is approximately 2–6 weeks between the two bottoms.	5
R6	A rise between the bottoms, which should be at least 10% higher than the lowest valley.	
R7	The left bottom is a V-Shaped bottom. The right bottom is a U-Shaped bottom.	6
R8	The two bottoms have similar prices.	

Table 6. The priority of the rules for the “Double Bottoms, Adam & Eve” pattern.

Rule	A “Double Bottoms, Eve & Adam”	Priority
R1	Prices trend downward to the pattern, and should not drift below the left bottom.	1
R2	The price must close above the confirmation line which is equivalent to the horizontal line at the highest high point.	2
R3	The pattern’s start point and end point are approximately at the same price.	3
R4	The confirmation line which is equivalent to the horizontal line at the highest high point and the data point between them are not higher than the horizontal trend line.	
R5	The duration is approximately 2–7 weeks between the two bottoms.	5
R6	A rise between the bottoms should be at least 10% higher than the lowest valley.	
R7	The left bottom is a U-Shaped bottom and the right bottom is a V Shaped bottom.	6
R8	The difference between the two bottoms is less than 4%.	

Table 7. The priority of the rules for the “Double Bottoms, Eve & Adam” pattern.

Rule	A “Double Bottoms, Eve & Eve”	Priority
R1	Prices trend downward to the pattern and should not drift below the left bottom.	1
R2	The price must close above the confirmation line, which is equivalent to the horizontal line at the highest high point.	2
R3	The pattern’s start point and end point are approximately at the same price.	3
R4	The confirmation line, which is equivalent to the horizontal line at the highest high point and the data point between them are not higher than the horizontal trend line.	
R5	The duration is approximately 2–7 weeks between the two bottoms.	5
R6	A rise between the bottoms, which should be at least 10% higher than the lowest valley.	
R7	The two bottoms are rounded and wide, similar to U-shaped bottoms.	6
R8	The difference between the two bottoms must be less than 6%.	

Table 8. The priority of the rules for the “Double Bottoms, Eve & Eve” pattern.

Rule	A “Double Tops, Adam & Eve”	Priority
R1	Price trends upward leading to the pattern and should not form a third peak, nor should the twin peaks be part of the same consolidation pattern. Look for two distinct minor highs.	1
R2	The price must close below the confirmation line which is equivalent to the horizontal line at the lowest low point.	2
R3	The pattern’s start point is approximately at the same price as the end point.	3
R4	The confirmation line which is equivalent to the horizontal line at the lowest low point and the data point between them are not lower than the horizontal trend line.	
R5	The duration is approximately 2–7 weeks between the two tops.	5
R6	A dip between the tops is approximately 10-20% lower than the highest top.	
R7	The left top is a V-shaped top and the right top is a U-shaped top.	6
R8	The two tops have similar prices.	

Table 9. The priority of the rules for the “Double Tops, Adam & Eve” pattern.

Rule	A "Double Tops, Eve & Adam"	Priority
R1	Price trends upward leading to the pattern, and should not form a third peak, nor should the twin peaks be part of the same consolidation pattern. Look for two distinct minor highs.	1
R2	The price must close below the confirmation line, which is equivalent to the horizontal line at the lowest low point.	2
R3	The pattern's start point is approximately at the same price as the end point.	3
R4	The confirmation line, which is equivalent to the horizontal line at the lowest low point and the data point between them are not lower than the horizontal trend line.	
R5	The duration is approximately 2–6 weeks between the two tops.	5
R6	A dip between the tops should be approximately 10-25% lower than the highest top.	
R7	The left top is a U-shaped top and the right top is a V-shaped top.	6
R8	The two tops have similar prices.	

Table 10. The priority of the rules for the "Double Tops, Eve & Adam" pattern.

Rule	A "Double Tops, Eve & Eve"	Priority
R1	Price trends upward leading to the pattern, and should not form a third peak, nor should the twin peaks be part of the same consolidation pattern. Look for two distinct minor highs.	1
R2	The price must close below the confirmation line, which is equivalent to the horizontal line at the lowest low point.	2
R3	The pattern start point is approximately at the same price as the end point.	3
R4	The confirmation line, which is equivalent to the horizontal line at the lowest low point and the data point between them should not be lower than the horizontal trend line.	
R5	The duration is approximately 2–6 weeks between the two tops.	5
R6	A dip between the tops should be approximately 10-20% lower than the highest top.	
R7	The left top is a U-shaped top and the right top is a V-shaped top.	6
R8	The two tops have similar prices.	

Table 11. The priority of the rules for the "Double Tops, Eve & Eve" pattern.

Rule	A "Rounding Bottoms"	Priority
R1	Price trends upward leading to the pattern.	1
R2	The price must close below the confirmation line, which is equivalent to the horizontal line at the lowest low point.	2
R3	The pattern start point is approximately at the same price as the end point.	3
R4	The confirmation line, which is equivalent to the horizontal line at the highest high point and the data point between them are not higher than the horizontal trend line.	
R5	The bottom is a U-shaped bottom.	6

Table 12. The priority of the rules for the "Rounding Bottoms" pattern.

Rule	A “Rounding Tops”	Priority
R1	Price trends downward leading to the pattern.	1
R2	The price must close below the confirmation line, which is equivalent to the horizontal line at the lowest low point.	2
R3	The pattern’s start point is approximately at the same price as the end point.	3
R4	The confirmation line, which is equivalent to the horizontal line at the lowest low point and the data point between them are not lower than the horizontal trend line.	
R5	The top is an inverted U-shaped top.	6

Table 13. The priority of the rules for the “Rounding Tops” pattern.

Rule	A “Scallops, Ascending”	Priority
R1	The start is usually a rising price trend and it is rarely a declining trend.	1
R2	It signals an upward breakout when the price closes above the highest high in the pattern, or it signals a downward breakout when the price closes below the lowest low in the pattern.	2
R3	The rounded top portion of the pattern usually retraces not less than 200%.	3
R4	The price, which pierces the trend line and the data point between them are not higher than the trend line.	
R5	The pattern resembles the shape of a letter J. It is almost like a U-shaped bottom.	6

Table 14. The priority of the rules for the “Scallops, Ascending” pattern.

Rule	A “Scallops, Ascending and Inverted”	Priority
R1	Select patterns that appear in an upward price trend or at the bullish turning point of a downward price trend.	1
R2	The price must close below the lowest low in the pattern.	2
R3	The rounded top portion of the pattern usually retraces larger than 50%.	3
R4	The price which pierces the trend line and the data point between them are not lower than the trend line.	
R5	The pattern resembles the upside down version of letter J after it is mirrored on y axis. It is almost like an inverted U-shaped top.	6

Table 15. The priority of the rules for the “Scallops, Ascending and Inverted” pattern.

Rule	A “Scallops, Descending”	Priority
R1	Look for a downward price trend leading to the scallop. That is rare when they do occur in price uptrends.	1
R2	The break up or down depends on whether the price closes above or below the highest high or lowest low.	2
R3	The end (right peak) usually retraces larger than 60%.	3

R4	The price which pierces the trend line and the data point between them are not lower than the trend line.	
R5	The pattern resembles the shape of a letter J after mirroring on the y-axis. It is almost like a U-shaped top.	6

Table 16. The priority of the rules for the “Scallops, Descending” pattern.

Rule	A “Scallops, Descending and Inverted”	Priority
R1	Most scallops appear in a downward price trend or at bearish turning points.	1
R2	The price must close below the lowest low in the pattern.	2
R3	The end (right peak) usually retraces larger than 50%.	3
R4	The price which pierces the trend line and the data point between them are not lower than the trend line.	
R5	The pattern resembles the shape of a letter J after mirroring on the *x*-axis. It is almost like a U-shaped top.	6

Table 17. The priority of the rules for the “Scallops, Descending and Inverted” pattern.

The pseudo code of ACPR is given in Algorithm 10. ACPR algorithm uses rules with the relevant priority to accelerate the classification chart patterns. In this algorithm, a nested loop is used for checking rules based on their priorities. When it fails to satisfy a rule, validation of remaining rules for the given segment is abandoned immediately to speed up the calculation.

6. Experiments on batch classification with historical datasets

In this section, we evaluate the performance of four segmentation methods (SP, SE, TSE and PSE) using real datasets. “Cup with Handle” and “Double Bottoms, Eve & Eve” pattern are used for comparison. The shape of the “Cup with Handle” pattern contains two parts. The left part is a U shape and the right part is a V shape. According to [1], the minimum and maximum lengths (window size) of the left part are 7 and 65 weeks. The minimum length of the right part is greater than or equal to 1 week and there is no limit on the maximum length. The prioritized rules for “Cup with Handle” pattern are listed in Table 4. In a similar way, we can divide the “Double Bottoms, Eve & Eve” pattern into two parts. Both left and the right parts are U shape. According to [1], the two bottoms have similar prices. A rise between the bottoms should be at least 10% higher than the lowest valley. The duration between the two bottoms is approximately 2–7 weeks. The confirmation line is equivalent to the horizontal line at the highest high point. The price must close above the confirmation line. The prioritized rules for “Double Bottoms, Eve & Eve” pattern are listed in Table 8.

To obtain more accurate classification results, we adopt a simple approach for discarding duplicate instances of a pattern found in a closed proximity. In this approach, two kinds of instances are eliminated. If an instance A is enclosed within another instance B, A will be eliminated. If instance B intersects with instance C and if B and C are enclosed by instance A, both B and C will be eliminated.

6.1. Real data experiment

In the experiments, we use the historical datasets of bank, insurance and casino stocks from Hang Seng Index (HSI) from 2 January 2007 to 31 October 2017. Experiments were conducted on a MacBook Pro with 2.9 GHz Intel Core i5 CPU with 8GB 1867 MHz DDR3 Memory. The algorithms were developed in jdk1.8.0 and MySQL Server 5.7.16 was installed to provide database service.

6.1.1 The number of patterns found

We compare the numbers of patterns classified by the four different methods in this experiment. The results are listed in Table 18. In Figure 16, we summarize the number of patterns found in each category. According to Figure 16, SP identified the least number of patterns among all 4 methods. The number of patterns found by TSE is higher than SP. PSE classified the highest number of patterns from the datasets. To better understand the performance of each approach, we further analyzed the experiment results and our findings are summarized as follows:

- 2 chart patterns (shown in #) from stock codes 1111 and 6889 can only be classified by SP and PSE.
- 24 chart patterns (shown in *) can be classified by SE and PSE but they cannot be identified by SP. Because of the curves, these chart patterns cannot be recognized by traditional segmentation methods such as SP.
- 12 chart patterns (shown in ^) can only be classified by PSE.
- 1 chart pattern (shown in $) from stock code 0582 is wrongly classified by SP. However, SE, TSE and PSE can exclude it correctly.

Stock classification	Stock code	"Cup with Handle"				"Double Bottoms, Eve & Eve"			
		SP	SE	TSE	PSE	SP	SE	TSE	PSE
Bank	0005	0	0	0	0	0	0	0	0
	0011	0	0	0	**1**^	0	0	0	0
	0023*	1	2	2	2	0	0	0	0
	0416	0	0	0	1	0	0	0	0
	0440*	2	3	3	3	0	0	0	0
	0626	0	0	0	**1**^	0	0	0	0
	0939	1	1	1	**2**^	0	0	0	0
	0998*	0	2	2	2	0	0	0	0
	1111	**1**#	0	0	1	0	0	0	0
	1216	0	0	0	0	0	0	0	0
	1288*	0	1	1	1	0	0	0	0
	1398	1	1	1	1	0	0	0	**1**^
	1578	0	0	0	0	0	0	0	0
	1658	0	0	0	0	0	0	0	0
	1963	0	0	0	0	0	0	0	0
	1988	1	1	1	1	0	0	0	**1**^
	2016	0	0	0	0	0	0	0	0
	2066	0	0	0	0	0	0	0	0
	2356	2	2	2	2	0	0	0	**1**^
	2388*	0	1	1	1	0	0	0	0
	2888*	0	1	1	1	0	0	0	0

	3328	0	0	0	0	0	0	0	0
	3618	1	1	1	1	0	0	0	**1**^
	3698	0	0	0	0	0	0	0	0
	3866	0	0	0	0	0	0	0	0
	3968	1	1	1	**2**^	0	0	0	0
	3988	1	1	1	1	0	0	0	0
	6122	0	0	0	0	0	0	0	0
	6138	0	0	0	0	0	0	0	0
	6196	0	0	0	0	0	0	0	0
	6818	0	0	0	0	0	0	0	0
Insurance	0662*	0	1	1	1	0	0	0	0
	0945	1	1	1	1	0	0	0	0
	0966*	0	1	0	1	0	0	0	0
	1299	0	0	0	0	0	0	0	0
	1336	0	0	0	0	0	0	0	0
	2318*	2	3	1	3	0	0	0	0
	2328	2	2	2	2	0	0	0	0
	2378	1	1	1	1	0	0	0	0
	2628	0	0	0	0	0	0	0	0
	6060	0	0	0	0	0	0	0	0
	6161	0	0	0	0	0	0	0	0
Casino	0027*	2	3	2	3	0	0	0	0
	0070	0	0	0	0	0	0	0	0
	0102*	0	1	1	1	0	0	0	0
	0200*	3	5	5	5	0	0	0	0
	0296&	2	1	1	1	0	0	0	0
	0326	0	0	0	0	0	0	0	0
	0555*	1	2	1	2	0	0	0	0
	0582$	1	0	0	0	0	0	0	0
	0880*	0	1	1	1	0	0	0	0
	0959	1	1	1	1	0	0	0	0
	1076	0	0	0	0	0	0	0	0
	1128	0	0	0	0	0	0	0	0
	1159*	2	3	3	3	0	1	0	1
	1180*	0	1	1	**2**^	0	0	0	0
	1182*	1	2	2	1	0	0	0	0
	1245	0	0	0	0	0	0	0	0
	1371	1	1	0	1	0	0	0	0
	1655	0	0	0	0	0	0	0	0

	1680*	0	1	1	1	0	0	0	0
	1928*	1	2	2	2	0	0	0	0
	2282*	0	1	1	1	0	0	0	0
	3918	1	1	1	1	0	0	0	0
	6889	**1#**	0	0	1	0	0	0	0
	8156*	0	1	1	1	0	0	0	0
	8198	1	1	1	1	0	0	0	0
	8279	0	0	0	0	0	0	0	0
Total		36	55	49	62	0	1	0	5

Table 18. The number of chart patterns found by SP, SE, TSE and PSE.

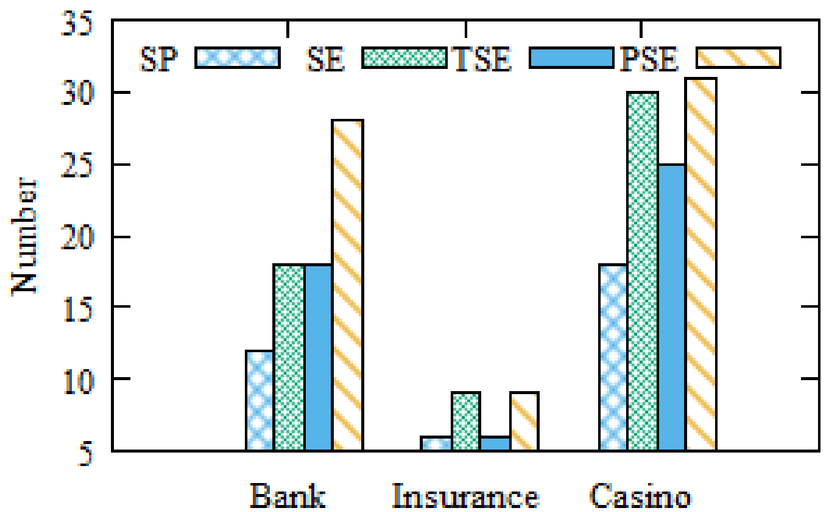

Figure 16. Total number of chart patterns found by SP, SE, TSE and PSE.

In this experiment, we found some special cases:

1) There are 2 instances of patterns (shown in #) from stock codes 1111 and 6889 can be classified by SP and PSE but ignored by SE and TSE due to the curve depth. In Figure 17, we show the instance from the stock 1111 as an example. The pattern is from 26 August 2014 to 30 March 2015 of stock 1111[2].
    - As for SP, the algorithm found that -1 ≤ Slope (P2, P3) ≤ 0 and 0 ≤ Slope (P3, P4) ≤ 1. The calculated curve depths $d_1 = (p_1 - p_3)/p_1$ equals to 15.164%, $d_2 = (p_5 - p_3)/p_1$ equals to 13.150%, $d_3 = (p_1 - p_3)/p_5$ equals to 15.476%, $d_4 = (p_5 - p_3)/p_5$ equals to 13.420%. Since all these values are between 12% and 50%, it is classified as a “Cup with Handle” pattern.
    - As for SE, the center of ellipse (x_0, y_0) equals to $(70.128, 17.098)$. Semi-axis length a' equals 62.100 and semi-axis length b' equals to 1.115. The counterclockwise rotation angle Phi equals to 0.003 and the lowest or highest point y_l of the ellipse equals to 15.970. The calculated curve depth d_1 equals to 11.025%, d_2 equals to 9.011%, d_3 equals to 11.252%, and d_4 equals

[2] Yahoo: 1111.HK. https://uk.finance.yahoo.com/quote/1111.HK/?p=1111.HK

to 9.196%. Since all of them are not between 12% and 50%, this instance is not classified as a "Cup with Handle" pattern.

- As for TSE, the center of ellipse (x_0, y_0) equals to $(70.128, 17.098)$. Semi-axis length a' equals 62.100 and semi-axis length b' equals to 1.115. The counterclockwise rotation angle Phi equals to 0.003 and y_l equals to 15.970. The calculated curve depth d_1 equals to 11.025%, d_2 equals to 9.011%, d_3 equals to 11.252%, and $d_4$equals to 9.196%. Since all of them are not between 12% and 50%, this instance is not classified as a "Cup with Handle" pattern.
- As for PSE, the center of ellipse (x_0, y_0) equals to $(81.293, 17.253)$. Semi-axis length a' equals 81.615 and semi-axis length b' equals to 2.520. The counterclockwise rotation angle Phi equals to 3.144 and y_l equals to 14.852. The calculated curve depths d_1 equals to 19.473%, d_2 equals to 14.932%, d_3 equals to 20.399%, and d_4 equals to 15.642%. Since all of them are between 12% and 50%, it is classified as a "Cup with Handle" pattern.

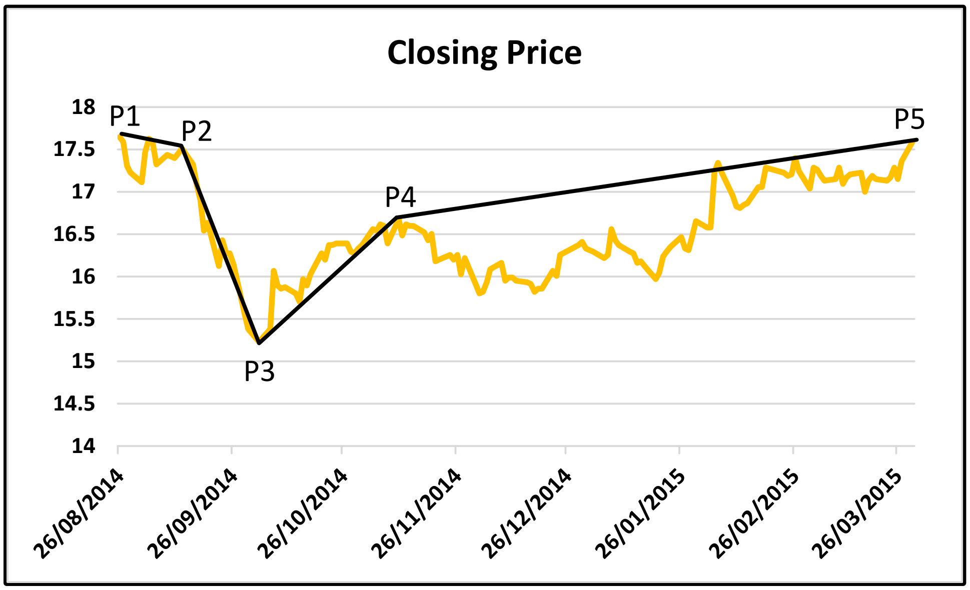

Figure 17. The instance can be classified by SP and PSE but cannot be identified by SE and TSE.

2) The instances of patterns (shown in *) cannot be classified by SP but can be classified by SE, TSE and PSE correctly due to the Direct Least Squares Fitting of Ellipses. For example, the instance of the pattern from 19 April 2016 to 18 August 2016 of stock 2282 is shown in Figure 18.
   - As for SP, since the instance cannot satisfy the rule $-1 \leq Slope\ (p_2, p_3) \leq 0$ and $0 \leq Slope\ (p_3, p_4) \leq 1$, it is not classified as a "Cup with Handle" pattern.
   - As for SE, the center of ellipse (x_0, y_0) equals to $(40.162, 11.262)$. Semi-axis length a' equals 1.017 and semi-axis length b' equals to 34.376. The counterclockwise rotation angle Phi equals to 1.571 and y_l equals to 10.244. The calculated curve depth d_1 equals to 16.440%, d_2 equals to 13.831%, d_3 equals to 16.881%, and d_4 equals to 14.201%. Since all of them are between 12% and 50%, it is classified as a "Cup with Handle" pattern.
   - As for TSE, the center of ellipse (x_0, y_0) equals to $(40.162, 11.262)$. Semi-axis length a' equals 1.017 and semi-axis length b' equals to 34.376. The counterclockwise rotation angle Phi equals to 1.571 and y_l equals to 10.244. The calculated curve depth $d_1$equals to 16.440%, d_2 equals to 13.831%, d_3 equals to 16.881%, and d_4 equals to 14.201%. Since all of them are between 12% and 50%, it is classified as a "Cup with Handle" pattern.

- As for PSE, the center of ellipse (x_0, y_0) equals to $(40.571, 11.152)$. Semi-axis length a' equals 1.171 and semi-axis length b' equals to 59.870. The counterclockwise rotation angle Phi equals to 3.137 and y_l equals to 10.084. The calculated curve depth d_1 equals to 17.751%, d_2 equals to 15.141%, d_3 equals to 18.227%, and d_4 equals to 15.547%. Since all of them are between 12% and 50%, it is classified as a "Cup with Handle" pattern.

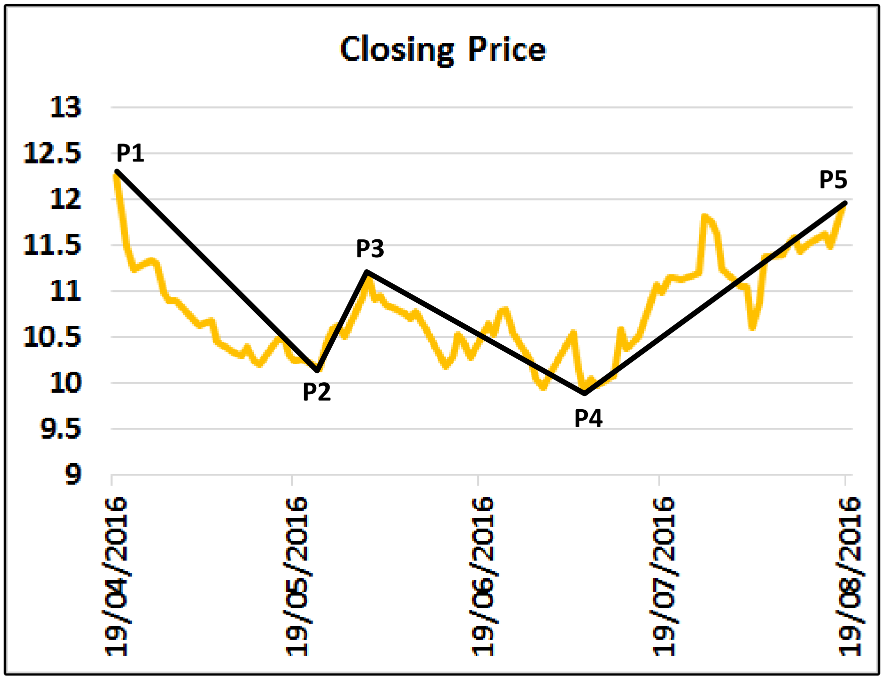

Figure 18. The instance which cannot be classified as "Cup with Handle" pattern by SP.

3) For the instance from 09 January 2015 to 23 March 2015 of stock 3968[3] indicated in "^" from Table 18, only PSE can classify as a "Cup with Handle" pattern. The instance of the pattern is shown in Figure 19.
    - As for SP, since the instance cannot satisfy the rule $-1 \leq Slope\ (p_2, p_3) \leq 0$ and $0 \leq Slope\ (p_3, p_4) \leq 1$, it is not classified as a "Cup with Handle" pattern.
    - As for SE, the center of ellipse (x_0, y_0) equals to $(25.921, 18.140)$. Semi-axis length a' equals 0.861 and semi-axis length b' equals to 19.894. The counterclockwise rotation angle Phi equals to 1.566 and y_l equals to 17.274. The calculated curve depth d_1 equals to 9.274%, d_2 equals to 8.013%, d_3 equals to 9.392%, and d_4 equals to 8.116%. Since all of them are not between 12% and 50%, the instance is not classified as a "Cup with Handle" pattern.
    - As for TSE, the center of ellipse (x_0, y_0) equals to $(25.921, 18.140)$. Semi-axis length a' equals 0.861 and semi-axis length b' equals to 19.894. The counterclockwise rotation angle Phi equals to 1.566 and y_l equals to 17.274. The calculated curve depth d_1 equals to 9.274%, d_2 equals to 8.013%, d_3 equals to 9.392%, d_4 equals to 8.116%. Since all of them are not between 12% and 50%, the instance is not classified as a "Cup with Handle" pattern.
    - As for PSE, the center of ellipse (x_0, y_0) equals to $(32.455, 18.879)$. Semi-axis length a' equals 1.985 and semi-axis length b' equals to 31.452. The counterclockwise rotation angle Phi equals to 3.139 and y_l equals to 16.899. The calculated curve depth d_1 equals to 15.082%,

[3] Yahoo: 3968.HK. https://uk.finance.yahoo.com/quote/3968.HK/?p=3968.HK.

d_2 equals to 14.177%, d_3 equals to 15.220%, and, d_4 equals to 14.307%. Since all of them are between 12% and 50%, it is classified as a “Cup with Handle” pattern.

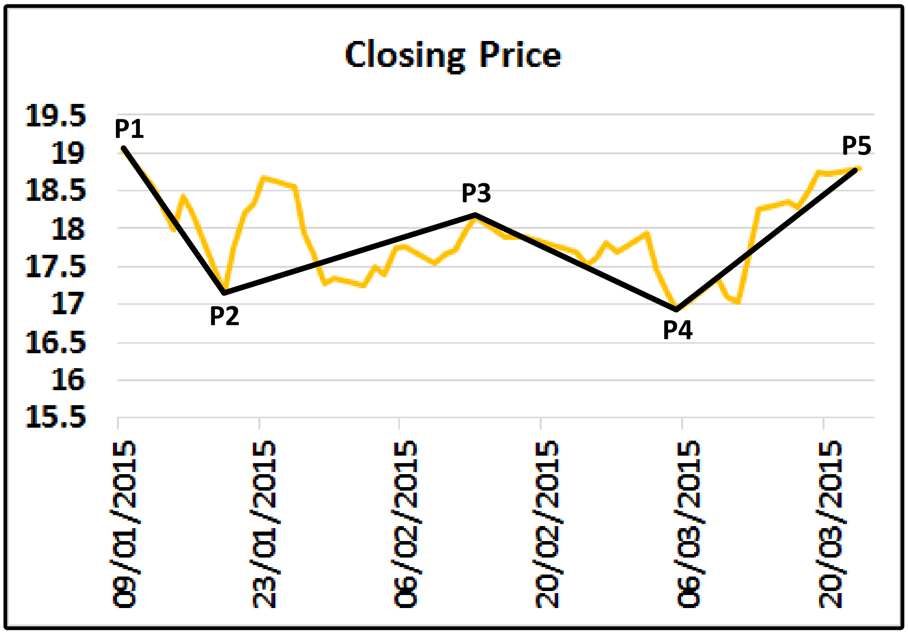

Figure 19. The instance can only be classified by PSE as a “Cup with Handle” pattern.

6.1.2 Error comparison

In this section, we compare the error of the four segmentation methods. The smaller the error, the better the representation by the segmentation method.

We can calculate the error based on the distance between two points (x_i, y_i) and $(\widehat{x_i}, \widehat{y_i})$ as follows [27]:

$$d_i = \sqrt{(x_i - \widehat{x_i})^2 + (y_i - \widehat{y_i})^2} \quad (1)$$

(x_i, y_i) is the point in the original time series. As for PIP-VD, $(\widehat{x_i}, \widehat{y_i})$ is the point in the straight line construct by the adjacent PIPs. As for SE, TSE and PSE, $(\widehat{x_i}, \widehat{y_i})$ is the point in the direct fit ellipse. There are two steps in this calculation.

1) Calculation of $\widehat{y_i}$:

As for PIP-VD, we use the slope of a line to calculate $\widehat{y_i}$. In this paper, we obtain five PIPs from PIP-VD process and there are four slopes in a segment:

We calculate $\widehat{y_i}$ by $\frac{\widehat{Xi}-Xm}{\widehat{Yi}-Ym} = \frac{Xn-Xm}{Yn-Ym}$ (n = 2…5, m= n-1, i = 0…Len (T)). (2)

As for SE, TSE and PSE, we calculate the center of ellipse (C_x, C_y), semi-axis lengths al and bl, and the counterclockwise rotation angle ∅ from the x-axis to the major axis of the ellipse by using Direct Least Squares Fitting of Ellipses algorithm [26]. Next, we use rotating ellipse formula to represent the direct fit ellipse:

$$\frac{((X-Cx)*cos\theta+(Y-Cy)*sin\theta)^2}{al^2} + \frac{((X-Cx)*sin\theta-(Y-Cy)*cos\theta)^2}{bl^2} = 1 \quad (3)$$

Because $\widehat{x_i}$ in this paper is equal to x_i which is in the order of 0, 1, 2…, we can calculate $\widehat{y_i}$ by

$$\frac{((Xi-Cx)*cos\theta+(\widehat{Yi}-Cy)*sin\theta)^2}{al^2} + \frac{((Xi-Cx)*sin\theta-(\widehat{Yi}-Cy)*cos\theta)^2}{bl^2} = 1 \quad (4)$$

2) Calculation of error:

$$Error = \sum_i d_i = \sum_i \sqrt{(x_i - \widehat{x_i})^2 + (y_i - \widehat{y_i})^2} \quad (5)$$

Error rate equals to the error divided by the sum of the distance between (x_i, y_i) and the x-axis:

$$Error\ rate = Error/\sum_i \sqrt{(x_i - x_i)^2 + (y_i - 0)^2} \quad (6)$$

The pseudo code of the error calculation algorithm is given in Algorithm 11.

In the experiment, we investigate the error rate of four segmentation methods. To calculate the error, we use the results of "Cup with Handle" pattern listed in Table 18. Since only a few instances of "Double Bottoms, Eve & Eve" pattern are found in the real dataset by all the four methods, this pattern is not used in the error comparison. We calculate the error rate of SP, SE, TSE and PSE by Algorithm 11. The results of the calculation is listed in Table 19. From the experiment results, we can see that the error rate of SP is large than SE and TSE but small than PSE. The result of SE and TSE are the same and PSE has the highest error rate.

Stock Classification	"Cup with Handle" SP	SE	TSE	PSE
Bank	0.036	**0.029**	**0.029**	0.067
Insurance	0.052	**0.033**	**0.033**	0.074
Casino	0.057	**0.040**	**0.040**	0.102

Table 19. Error rate of SP, SE, TSE and PSE.

6.1.3 Results on execution time

In this experiment, we compare the execution time of four methods. The execution time for each pattern is listed Table 20. The bottom row shows the total execution time. The total execution time for each method is summarized in Figure 20. From Figure 20, we can see that SP and SE have similar execution time. PSE and TSE have the longest and shortest execution time. TSE has shortest execution time since turning point selection is performed as the first step, which significantly reduces the number of points required for processing. PSE has the longest execution time since PLA-BU method is computationally expensive.

Stock classification	"Cup with Handle" SP	SE	TSE	PSE
Bank	28816.733	28816.629	2288.320	28998.890
Insurance	11423.557	11422.983	1243.089	11572.113

Casino	28479.106	28479.296	2286.386	28569.337
Total	68719.396	68718.908	**5817.750**	69130.340

Stock classification	“Double Bottoms, Eve & Eve”			
	SP	SE	TSE	PSE
Bank	3735.808	3724.242	454.944	3800.403
Insurance	1532.735	1534.743	169.262	1568.243
Casino	3965.255	3997.840	369.737	4169.672
Total	9233.798	9255.825	**1000.943**	9439.318

Table 20. The execution time of SP, SE, TSE and PSE (in Seconds).

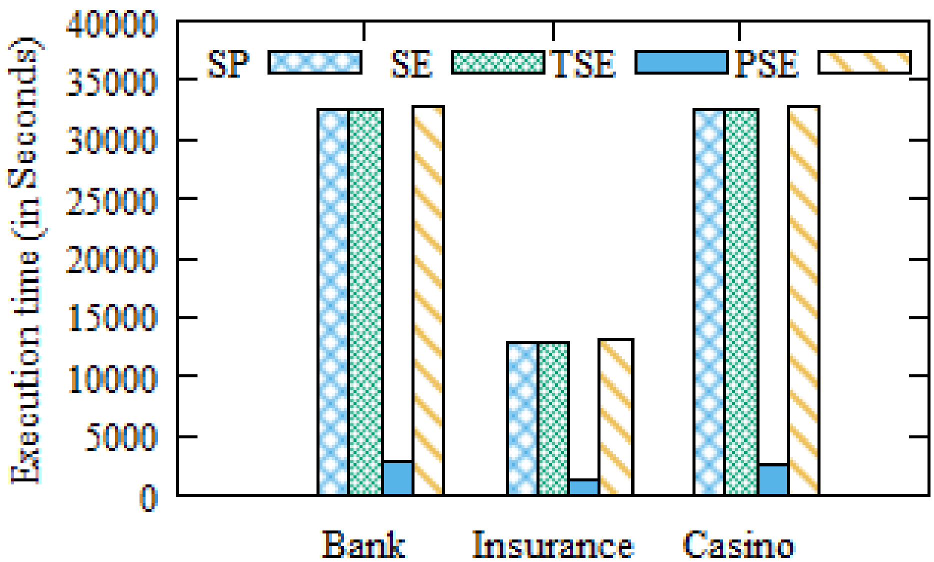

Figure 20. The execution time of SP, SE, TSE and PSE (in Seconds).

6.2 Comparison on the execution time with/without ACPR

In this experiment, we use the same real data sets described in section 6.1. To further analyse the efficiency of the proposed methods, we have added the classification of 5 more patterns (Bump-and-Run Reversal Bottoms”, “Bump-and-Run Reversal Tops”, “Scallops, Ascending”, “Rounding Bottoms”, and “Rounding Tops”) patterns. In this experiment, SP, SE, TSE and PSE represent the segmentation methods without ACPR and SP_ACPR, SE_ACPR, TSE_ACPR, and PSE_ACPR represent the methods with ACPR. The results of the experiments are listed in Table 21 and Table 22. For the purpose of simplification, we rounded down the results to remove the decimal values. The total execution time is depicted in Figure 21. From the experiment results, we can see that the segmentation methods with ACPR can reduce the total execution time in half.

Chart patterns	SP	SE	TSE	PSE
“Bump-and-Run Reversal Bottoms”	27805	27835	5033	32367
“Bump-and-Run Reversal Tops”	29724	29753	5788	34653

"Cup with Handle"	68719	68718	5817	69130
"Double Bottoms, Eve & Eve"	9233	9255	1000	9439
"Scallops, Ascending"	11672	11675	2863	12345
"Rounding Bottoms"	12491	12532	3312	14645
"Rounding Tops"	13194	13542	3522	16112
Total	172838	173310	**27335**	188691

Table 21. The execution time of segmentation methods without ACPR algorithm (in Seconds).

Chart patterns	SP_ACPR	SE_ACPR	TSE_ACPR	PSE_ACPR
"Bump-and-Run Reversal Bottoms"	15456	15475	2745	17039
"Bump-and-Run Reversal Tops"	16017	16045	3122	18114
"Cup with Handle"	30595	30602	2698	30990
"Double Bottoms, Eve & Eve"	4559	4567	494	4564
"Scallops, Ascending"	9180	9195	1317	10335
"Rounding Bottoms"	9338	9410	2562	11123
"Rounding Tops"	7429	10033	2712	12233
Total	92574	95327	**15650**	104398

Table 22. The execution time of segmentation methods with ACPR algorithm (in Seconds).

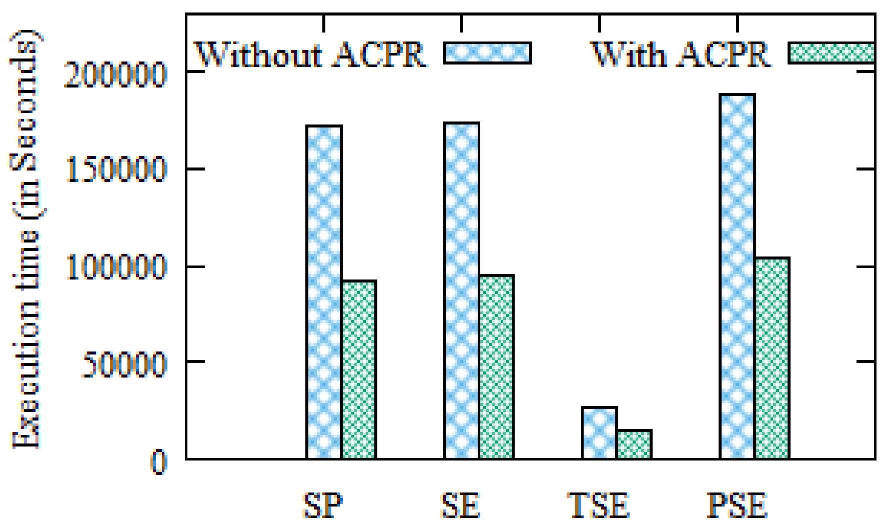

Figure 21. The execution time of segmentation methods with/without ACPR (in Seconds).

7. Conclusions

Segmentation methods are important pre-processing steps for chart pattern classification in financial time series. Classification methods such as RB, TB, and HY rely on these segmentation methods to extract important data points for identification process. However, existing segmentation methods such as PIP, TP, PLA, and PAA are less effective in classifying 16 curve-shaped chart patterns from financial time series.

In this paper, we propose three novel segmentation methods (SE, TSE, and PSE) for classification of curve-shaped chart patterns based on direct least squares fitting of ellipses. These methods are implemented based on the principles of sliding windows, turning points, and bottom-up piece wise linear approximation. To further enhance the efficiency in classifying chart patterns from real-time streaming data, we propose a novel algorithm called Accelerating Classification with Prioritized Rules (ACPR). In this algorithm, rules defined for each curve-shaped pattern are assigned with different priorities for evaluation. ACPR is designed to abandon the evaluation process once a prioritized rule is found to be false. In this way, classification process can be significantly accelerated by avoiding unfruitful comparisons.

Extensive experiments based on real datasets from Hong Kong stock market are used to evaluate the proposed approaches. In the experiments, we use SP as a baseline method to evaluate the performance of the three new segmentation methods. From the experiment results, we can observe that SP finds the least number of segments since it is unable to detect curves in a large number of cases. SP is suitable for low fluctuation series and its error rate is larger than SE and TSE. SE finds more segments than SP but its error rate is smaller than SP. TSE finds similar number of segments as SP does. However, TSE is much faster than SP but its error rate is smaller than SP. Both SE and TSE are also suitable for high fluctuation series. PSE finds the highest number of segments than others. However, it also has the highest error rate. The experiment results with/without ACPR also reveal that by using the proposed approach, the execution time for classification can be significantly reduced.

As for the future work, we are planning to explore image-based feature extraction and pattern classification approaches proposed in graphics and computer vision areas. One potential approach is to capture the segmented subsequence from the input time series as an image and extract the relevant features for pattern classification. A similar feature extraction approach based on Canny edge detection [28] and modified CART algorithm [29] was proposed in [30] by Chen et al. for defect classification in color filter and micro-lens manufacturing. In their approach, 10 candidate features from the image were used for defect classification. Another potential approach for classification of curve-shaped chart patterns is to use deep learning neural networks such as Convolutional Neural Networks (CNN) for recognizing the patterns from the extracted images. However, image and computer vision-based approaches require substantial amount of computation. Since real-time classification capability is essential for technical analysis in stock markets, minimizing the computation complexity would be one of the key challenges in adopting above technologies.

In addition, we are also planning to investigate similar solutions for accelerating classification of patterns from other categories. Besides, we are also planning to extend the current work for classification of chart patterns from intra-day time series.

Acknowledgement

This research was funded by University of Macau (File no. MYRG2019-00136-FST and MYRG2018-00246-FST).

Appendix A.

Algorithm 1. Pseudo code of the PIP identification.

Function: PIP_Identification (T)
  Input: Time series T [1...i]
  Output: PIPs List L [1...i]
Begin
  Set L [1] = T [1], L [2] = T [i]

```
    Repeat until L [1...i] all filled
    Begin
        Point T [j] with maximum distance was selected to the
        Adjacent points in PIPs List (L [1] and L [2] initially)
        Append T [j] To L
    End
    Return
End
```

Algorithm 2. Pseudo code of the PLA-bottom up.

```
Function: PLA_BottomUP (T, max_error)
    Input: Time series T [1...i], max error max_error
    Output: PLAs List S [1...i]
Begin
    // Create initial fine approximation
     While i between 1 and  len (T)
          S = concat (S, create_segment (Ti, i+1));
     End

    // Find the cost of merging each pair of segments
     While i between 1 and  len (S)-1
          merge cost (i) = calculate error ([merge (S (i), S (i +1))]);
     End

    While min (merge cost) < max error
        i = min (merge cost); // Find the cheapest pair to merge
        S (i) = merge (S (i), S (i + 1)); // Merge them
        Delete (S (i + 1));                    //Update records
        Merge cost (i) = calculate error (merge (S (i), S (i + 1)));
        Merge cost (i-1) = calculate error (merge (S (i-1), S (i)));
    End
    Return S
End
```

Algorithm 3. Pseudo code of the TP identification.

```
Function: TPs_identification (T)
    Input: Time series T [1...o]
    Output: Turing points TP [1…q]
Begin
    For every point j in T [1…o]
        If T [j] > T [j-1] and T [j] > T [j+1]
            Or T [j] < T [j-1] and T [j] < T [j+1]
            Put T [j] into TP
        End If
    End
    Return TP
End
```

Algorithm 4. Pseudo code of the Direct Least Squares Fitting of Ellipses.

Function direct_fit_ellipse (S)
Input: Segmented time series S
or PLAs List for segmented time series S
Output: Segmented time series
or PLAs List for segmented time series with related curve depth S
Begin
For every segments or PLAs list for segments in S[j]
//1ST step
Calculate the coefficients, a, b, c, d, e, f for the ellipse by the set of points (x_i, y_i) in S[j].

//2nd step
Calculate the center of ellipse (x_0, y_0), semi-axis lengths a' and b' and the counterclockwise rotation angle $\emptyset$.

//3rd step
Calculate the lowest or highest point $y_l$of the ellipse using the tangent line and point of tangency.
Calculate the curve depth d_1, d_2, d_4, d_4 of the ellipse and concatenate to S[j].
$S = concat\ (S, concat(S[j], d_1, d_2, d_4, d_4))$
End
//Return Segmented time series or PLAs List for segmented time series with related curve depth S.
Return S
End

Algorithm 5. Pseudo code of the SW algorithm.

Function: Sliding_Window (T, min_number, max_number)
Input: Time series T, min segment length min_number, max segment length max_number
Output: Segmented time series S
Begin
Anchor = 1;
While len (T) – Anchor >= max_number
i = min_number
While i between min_number and max_number
S = concat (S, create_segment (T [Anchor: Anchor + i]));
i += 1;
End;
Anchor += 1;
End;
Return S;
End;

Algorithm 6. Pseudo code of SE.

Function SE (T, min_number, max_number, pattern)
Input: Time series T, min segment length min_number, max segment length max_number,

```
        pattern name pattern
  Output: Segmented time series S
Begin
  // 1st step get segments
  Anchor = 1;
  While len (T) – Anchor >= max_number
    i = min_number
    While i between min_number and max_number
      S = concat (S, create_segment (T [Anchor: Anchor + i]));
      i += 1;
    End;
    Anchor += 1;
  End;

  // 2nd step direct least squares fitting of ellipses
  S = direct_fit_ellipse (S);

  //3rd step use rule to classify chart patterns
  S = rule_selecting_ellipse (pattern, S);

  Return S;
End;
```

Algorithm 7. Pseudo code of TSE.

```
Function TSE (T, min_number, max_number, pattern)
  Input: Time series T, min segment length min_number, max segment length max_number,
        pattern name pattern
  Output: Segmented time series S
Begin
  // 1st step identifying the turning points by TP
  TP_List [1...m] = TPs_identification (T)

  // 2nd step get segments
  Repeat until TP_List [1...m] all selected
    Anchor = 1;
    While len (T) – Anchor >= max_number
      i = min_number
      While i between min_number and max_number
        S = concat (S, create_segment (T [Anchor: Anchor + i]));
        i += 1;
      End;
      Anchor = TP_List[j];
    End;
  End;

  // 3rd step direct least squares fitting of ellipses
  S = direct_fit_ellipse (S);

  //4th step use rule to classify chart patterns
  S = rule_selecting_ellipse (pattern, S);
```

```
    Return S;
End;
```

Algorithm 8. Pseudo code of PSE.

```
Function PSE (T, min_number, max_number, pattern)
    Input: Time series T, min segment length min_number, max segment length max_number,
           pattern name pattern
    Output: Segmented time series S
Begin
    // 1st step get segments
    Anchor = 1;
    While len (T) – Anchor >= max_number
        i = min_number
        While i between min_number and max_number
            S = concat (S, create_segment (T [Anchor: Anchor + i]));
            i += 1;
        End;
        Anchor += 1;
    End;

    //2nd step get PLAs by PLA-BU
    S = PLA_BottomUP (S);

    //3rd step direct least squares fitting of ellipses
    S = direct_fit_ellipse (S);

    //4th step use rule to classify chart patterns
    S = rule_selecting_ellipse (pattern, S);

    Return S;
End;
```

Algorithm 9. Pseudo code of SP.

```
Function SP (T, min_number, max_number, pattern)
    Input: time series T, min segment length min_number, max segment length max_number,
           pattern name pattern
    Output: Segmented time series S
Begin
    // 1st step get segments
    Anchor = 1;
    While len (T) – Anchor >= max_number
        i = min_number
        While i between min_number and max_number
            S = concat (S, create_segment (T [Anchor: Anchor + i]));
            i += 1;
        End;
        Anchor += 1;
```

```
    End;

    // 2nd step get PIPs by PIP-VD
    S = PIP_Identification (S);

    //3rd step use rule to classify chart patterns
    S = Rule_Selecting (pattern, S);

    Return S;
End;
```

Algorithm 10. Pseudo code of ACPR.

```
Function ACPR (pattern, seq)
    Input: pattern name, time series sequence seq
    Output: Segmented time series S
Begin
    //Loop rule priority i from 1 to 5
    i = 1
    S = seq
    While i <= 5
        //Get rules R1, R2…Rm by the pattern name and priority i.
        get_rules (pattern, i)

        //Nested loop for Ri checking
        For Ri in (R1, R2…Rm)
            //Exit the function and return null when the Ri is false
            If Ri is false Then
                S = Null
                Exit
            End If
        End For
        i += 1
    End
    Return S
End
```

Algorithm 11. Pseudo code of error calculation algorithm.

Function: Error $(\mathbf{T}, \mathbf{Tlis}, \mathbf{C_x}, \mathbf{C_y}, \mathbf{al}, \mathbf{bl}, \boldsymbol{\theta}, \mathbf{Med})$

Input: Time series T $[(x_o, y_o), \ldots (x_z, y_z)]$,

PIPs list $TLis$ $[(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)]$,

Direct fit ellipse center point $\left(C_x, C_y\right)$, a length al, b length bl, counterclockwise θ,

Segmentation method Med

Output: Error rate

Begin

For every point (x_i, y_i) in T $[(x_o, y_o), \ldots (x_z, x_z)]$

//1ST step calculating $\widehat{Y\iota}$

If Med in (PIP-VD) **Then**

If x_i between x_1 and x_2 **Then**
Calculate $\widehat{yi}$ by $\frac{Xi-x1}{\widehat{Yi}-y1} = \frac{x2-x1}{y2-y1}$
Else If x_i between x_2 and x_3 **Then**
Calculate $\widehat{yi}$ by $\frac{Xi-x2}{\widehat{Yi}-y2} = \frac{x3-x2}{y3-y2}$
Else If x_i between x_3 and x_4 **Then**
Calculate $\widehat{yi}$ by $\frac{Xi-x3}{\widehat{Yi}-y3} = \frac{x4-x3}{y4-y3}$
Else If x_i between x_4 and x_5**Then**
Calculate $\widehat{yi}$ by $\frac{Xi-x4}{\widehat{Yi}-y4} = \frac{x5-x4}{y5-y4}$
End IF
Else If Med in (SE, TSE, PSE) **Then**
Calculate $\widehat{yi}$ by $\frac{((Xi-Cx)*cos\theta+(\widehat{Yi}-Cy)*sin\theta)^2}{al^2} + \frac{((Xi-Cx)*sin\theta-(\widehat{Yi}-Cy)*cos\theta)^2}{bl^2} = 1$
End If

//2nd calculating error by summing up the distance between two points (x_i, y_i) and $(\widehat{x_i}, \widehat{y_i})$

$$Error = \sum_i d_i = \sum_i \sqrt{(x_i - \widehat{x_i})^2 + (y_i - \widehat{y_i})^2}$$

$$Error\ rate = Error / \sum_i \sqrt{(x_i - x_i)^2 + (y_i - 0)^2}$$

End For
Return Error rate
End

References

[1] N. Bulkowski, *Encyclopedia of chart patterns*, Second ed.: Wiley, 2005.

[2] Y. Wan and Y.-W. Si, "A formal approach to chart patterns classification in financial time series," *Information Sciences*, vol. 411, pp. 151-175, 2017/10/01/ 2017.

[3] F. L. Chung, T.-C. Fu, R. Luk, and V. Ng, *Flexible time series pattern matching based on perceptually important points*, 2001.

[4] Y.-W. Si and J. Yin, "OBST-based segmentation approach to financial time series," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 2581-2596, 2013/11/01/ 2013.

[5] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani, "An Online Algorithm for Segmenting Time Series," presented at the Proceedings of the 2001 IEEE International Conference on Data Mining, 2001.

[6] E. J. Keogh and M. J. Pazzani, "A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases," presented at the Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications, 2000.

[7] 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, pp. 347-364, 2007/04/01/ 2007.

[8] Y. Chen, B. Hu, E. Keogh, and G. E. A. P. A. Batista, "DTW-D: time series semi-supervised learning from a single example," presented at the Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, Chicago, Illinois, USA, 2013.
[9] Z. Zhang, J. Jiang, X. Liu, R. Lau, H. Wang, and R. Zhang, "A real time hybrid pattern matching scheme for stock time series," presented at the Proceedings of the Twenty-First Australasian Conference on Database Technologies - Volume 104, Brisbane, Australia, 2010.
[10] Q. Sun and Y.-W. Si, "An Efficient Segmentation Method: Perceptually Important Point with Binary Tree," Cham, 2020, pp. 350-365.
[11] C.-H. Chen, V. S. Tseng, H.-H. Yu, and T.-P. Hong, "Time series pattern discovery by a PIP-based evolutionary approach," *Soft Computing,* vol. 17, pp. 1699-1710, 2013/09/01 2013.
[12] Y. Wan, X. Gong, and Y.-W. Si, "Effect of segmentation on financial time series pattern matching," *Applied Soft Computing,* vol. 38, pp. 346-359, 2016/01/01/ 2016.
[13] J. Yin, Y. Si, and Z. Gong, "Financial time series segmentation based on Turning Points," in *Proceedings 2011 International Conference on System Science and Engineering*, 2011, pp. 394-399.
[14] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery,* vol. 15, pp. 107-144, 2007/10/01 2007.
[15] J.-L. Wu and P.-C. Chang, "A Trend-Based Segmentation Method and the Support Vector Regression for Financial Time Series Forecasting," *Mathematical Problems in Engineering,* vol. 2012, p. 615152, 2012/05/29 2012.
[16] R. Palivonaite, K. Lukoseviciute, and M. Ragulskis, "Algebraic Level-Set Approach for the Segmentation of Financial Time Series," Berlin, Heidelberg, 2014, pp. 239-250.
[17] Y. Yin, P. Shang, and J. Xia, "Compositional segmentation of time series in the financial markets," *Applied Mathematics and Computation,* vol. 268, pp. 399-412, 2015/10/01/ 2015.
[18] H. Wang, P. Shang, and J. Xia, "Compositional segmentation and complexity measurement in stock indices," *Physica A: Statistical Mechanics and its Applications,* vol. 442, pp. 67-73, 2016/01/15/ 2016.
[19] A. M. Durán-Rosal, M. de la Paz-Marín, P. A. Gutiérrez, and C. Hervás-Martínez, "Identifying Market Behaviours Using European Stock Index Time Series by a Hybrid Segmentation Algorithm," *Neural Processing Letters,* vol. 46, pp. 767-790, 2017/12/01 2017.
[20] Y. Wan and Y.-W. Si, "A hidden semi-Markov model for chart pattern matching in financial time series," *Soft Computing,* vol. 22, pp. 6525-6544, 2018/10/01 2018.
[21] Y. Wan and Y.-W. Si, "Adaptive neuro fuzzy inference system for chart pattern matching in financial time series," *Applied Soft Computing,* vol. 57, pp. 1-18, 2017/08/01/ 2017.
[22] X. Gong, Y.-W. Si, S. Fong, and R. P. Biuk-Aghai, "Financial time series pattern matching with extended UCR Suite and Support Vector Machine," *Expert Systems with Applications,* vol. 55, pp. 284-296, 2016/08/15/ 2016.
[23] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, *et al.*, "Searching and mining trillions of time series subsequences under dynamic time

warping," presented at the Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, Beijing, China, 2012.
[24] A. Zapranis and P. Tsinaslanidis, "Identification of the Head-and-Shoulders Technical Analysis Pattern with Neural Networks," Berlin, Heidelberg, 2010, pp. 130-136.
[25] Z. Zhang, J. Jiang, X. Liu, R. Lau, H. Wang, and R. Zhang, "A real time hybrid pattern matching scheme for stock time series," presented at the Proceedings of the Twenty-First Australasian Conference on Database Technologies - Volume 104, Brisbane, Australia, 2010.
[26] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct Least Square Fitting of Ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 21, pp. 476-480, 1999.
[27] G. Taricco, "The Distance Between Two Points Moving On A Graph," *International Journal of Modelling and Simulation,* vol. 15, pp. 120-124, 1995/01/01 1995.
[28] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-8, pp. 679-698, 1986.
[29] L. Breiman, Friedman, J.H., Olshen, R.A., & Stone, C.J., *Classification And Regression Trees,* First ed.: CRC Press, 1984.
[30] Y.-J. Chen, T.-H. Lin, K.-H. Chang, and C.-F. Chien, "Feature extraction for defect classification and yield enhancement in color filter and micro-lens manufacturing: An empirical study," *Journal of Industrial and Production Engineering,* vol. 30, pp. 510-517, 2013/12/01 2013.