TREND DETECTION AND PATTERN RECOGNITION IN FINANCIAL TIME SERIES

by

Seunghye J. Wilson
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Statistical Science

Committee:

_____ Dr. James E. Gentle, Dissertation Director

_____ Dr. Daniel B. Carr, Committee Member

_____ Dr. Guoquing Diao, Committee Member

_____ Dr. Jessica Lin, Committee Member

_____ Dr. William F. Rosenberger, Department Chair

_____ Dr. Kenneth S. Ball, Dean, The Volgenau School of Engineering

Date: _____ Spring Semester 2016
George Mason University
Fairfax, VA

Trend Detection and Pattern Recognition in Financial Time Series

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Seunghye J. Wilson
Master of Science
Rutgers University, 2010
Bachelor of Science
Yonsei University, 2003

Director: Dr. James E. Gentle, Professor
Department of Statistics

Spring Semester 2016
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to my parents *Taewoong Jung* and *Sunbok Lee*, and my better half *Justin A. Wilson* for their unconditional love and support.

Acknowledgments

First of all, I would like to express my deep gratitude to my advisor Dr. James E. Gentle for his supporting, encouraging, and extensive knowledge. I was very fortunate to be his student. He is not only a great advisor for research but also gentle-hearted mentor for my life. Through the priceless time with him, I was inspired to have a deeper passion for statistics and research. He directed me to grow as an independent researcher.

I also would like to thank committee members, Dr. Daniel Carr for his advice for graph visualization, Dr. Guoqing Diao for his time and support, and Dr. Jessica Lin for her guidance and vast knowledge about time series data mining. Without their valuable support and advice, this dissertation would not have been possible.

I am grateful for Dr. Rosenberger and his leadership of the department, and also want to especially thank Mrs. Elizabeth Quigley for her thoughtful, caring, and kind assistance of students.

I had a wonderful graduate student life thanks to my classmates, graduate teaching assistant buddies, and friends in Room 2616, Engineering Building who are all passionate PhD students.

I personally would like to thank Dr. Sijung Yun, CEO of Yotta Biomed LLC., who encouraged me to stay strong at all times.

Also I am grateful for Mr. Byungjoon Lee, Senior Customer Operation Manager of Microsoft Inc., who was my manager and a great mentor for my career. Without his help and recommendation, I could not have started my new life as a statistician.

For my parents, I cannot really thank them enough. They dedicated their life for their three children and family, which enriched my life to be happy and confident.

Last but not least, I would like to thank my husband Justin who always stays with me by my side in weal and woe. His support, encouragement, and love are definitely the greatest resource of my life and achievement.

Table of Contents

List of Tables

List of Figures

Abstract

TREND DETECTION AND PATTERN RECOGNITION IN FINANCIAL TIME SERIES

Seunghye J. Wilson, PhD

George Mason University, 2016

Dissertation Director: Dr. James E. Gentle

One major interest of financial time series analysis is to identify changepoints of trends and recognize patterns that can be used for classification and clustering of time series. Because of the large amounts of data, nonlinear relationship of the data elements, and the presence of random noise, some method of data reduction is necessary. The data reduction, however, must preserve the important characteristics of the original data. Many representation methods in the time domain or frequency domain have been suggested to accomplish efficient extraction of information. These include, for example, piecewise linear approximation, symbolic representation, and discrete wavelet transformation (DWT). However, most of the existing methods do not take into consideration time information of trends and/or depend on user-defined parameters, for example the number of segments for piecewise approximation.

We introduce *alternating trend smoothing* (ATS) and *piecewise band smoothing* (PBS) for data representation based on up/down direction change as it has h (step size) additional data points and linear regression using small sets of current data points respectively. The proposed method is flexible and interpretable in the sense that it allows the acquisition and addition of new data points (online method) to detect meaningful trends and changepoints.

Changepoints are confirmed once new data points stray far enough outside of the band, creating a reduced dataset of changepoints to utilize. Next, we define patterns from the reduced data which preserve trends and the length of a trends duration. In addition to the definition of patterns, some distance metrics are suggested as similarity measures that are suitable for reduced data by our data representation. Finally, we demonstrate applications of clustering, classification, indexing, and prediction using methods suggested, and discuss conclusions and future work.

Chapter 1: Introduction

1.1 Problem Statement

Traditional financial times series models, such as an ARIMA and frequency domain model approaches, assume underlying parametric models. These models describe the data generating process that governs the global datasets and also provide significant data reduction. However, it is not feasible to describe data with a single global model as the size of data dramatically increases. As a result, various methods for data reduction, or representation, have been proposed; piecewise approximation, preserving critical points, symbolic transformation and so forth. One of the primary goals of these and other data reduction methods is to adequately preserve the key information we want to analyze.

Considering that trends, generally "up" and "down", and their changepoints are of major interest in financial time series analysis, it is often desirable that the methods for data representation preserve the information on trends and the changepoints of trends. Nevertheless, only a few existing methods for data representation seem to be able to approximate data, preserving these trend characteristics. Bao (2008) [10] suggests *critical points model* (CPM) for financial time series representation. CPM model smooths data by identifying local minimal/maximal points based on some threshold criteria. Chung et al. (2001) [12] propose a *perceptually important point* (PIP) method to represent financial data by preserving salient points that contribute to the shape of the data. Given datasets, these methods can provide information on trends of data fairly well, however, the data is smoothed based on the shape of batch data for trend changepoints or processed for trend information.

Once the original data is represented, we measure distance between time series for classification or clustering. There are various methods to measure similarity/dissimilarity

between two time series: L_p, correlation, and *dynamic time warping* (DTW), to name a few. The choice of similarity measure considerably relies on the structure of represented data. For example, although Euclidean distance is one of the most widely used methods, measuring similarity between two time series by directly using Euclidean distance may yield counter-intuitive results because it is sensitive to noise. Rather DTW is often more popular for comparing similarity among time series, such as pattern recognition. Of course, there is no single distance measure that is superior over all other methods. The choice of distance measure may not be the same depending on the purpose of the analysis even if the data is approximated by the same representation methods. These are methods that treat data representation and similarity/dissimilarity as one combined problem.

In this research, we introduce new methods of data representation (dimensionality reduction), defining financial data trends, and distance measures.

- **Data Representation**

  Our data representation methods focus on detecting trends and trend changepoints. We assume that the large size time series data consists of a sequential data generating process over the sequence of non-overlapped time domain rather than governed by one single model. By identifying trend changepoints, we find trend regimes sequentially. To identify changepoints, we introduce parameters that set criteria of "change" and define "trend" of the regime.

- **Defining Patterns**

  The original data with length N becomes a sequence with length n ($n \ll N$) by data representation. Each element in the sequence has some form of underlying model of data generating process, a linear fit, on the non-overlapped piece of time domain, that is called *trend regime*. From the coefficients of a linear function or the length of the trend regime in the reduced datasets, we suggest various methods to define patterns - numerical and categorical - and then derive some statistical properties of the length of trend regimes and the changepoints.

- **Distance Measures**

  We introduce new methods that measure the distance between two times series based on their trends and trend changepoints over time corresponding to appropriate types of patterns, and then discuss their properties.

- **Applications**

  We demonstrate applications in classification, clustering, indexing, and prediction with real-world financial data incorporating the methods suggested in this research.

1.2 Dissertation Outline

In Chapter 2, we will review background and relevant literature. Chapter 3 will provide new methods of data representation for financial time series by identifying changepoints. Also, parameters that define "change" of trends and their properties are addressed. Chapter 4 addresses various methods to define patterns from the reduced data and corresponding distance measures, and statistical properties of the length of trends and changepoints. In Chapter 5, we demonstrate applications examples, clustering, classification, indexing, and prediction, using our new methods discussed in Chapter 3 and 4. Chapter 6, will provide our conclusions, challenges, and future work.

Chapter 2: Background and Literature Review

2.1 General Approaches for Data Approximation

Function approximation has been widely used and developed in many applications. One approach to find a simpler representation when the known underlying function is too complicated for practical use. Another approach is to use nonparametric methods. Nonparametric methods estimate the unknown function using interpolation and extrapolation given observed data (Goodman (2006) [23]). For very large size datasets, it is not feasible to describe the character of the original data with one single global either parametric or nonparametric model, thereby the sequence of local parametric functions may be a better approach. We incorporate these methods to represent for lengthy and noisy financial time series. In this section, we review function approximations according to the size and availability of data, and loss function and regularization to find optimal parameters. The mathematical notation follows that of Hastie, Tibshirani, and Friedman (2009) [25].

2.1.1 Global Models: Regressions

Linear Regression

Linear regression is one of the most extensively used statistical techniques for data description, parameter estimation, and prediction because of its interpretability, among other reasons. Let $X^T = (X_1, X_2, \cdots, X_p)$ and Y be an input vector and a real-valued output vector respectively. Then the liner regression assumes the underlying model of data

$$Y = f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \epsilon \tag{2.1}$$

where the β_0, β_j's are unknown parameters and ϵ is independent random error vector with zero mean and finite variance. X_j can be numerical data or basis expansions, such as $X_2 = X_1^2, X_3 = X_1^3$ in polynomial regression models. These unknown parameters are generally estimated by *ordinary least squares* (OLS), that is by minimizing the *residual sum of squares* (RSS),

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \tag{2.2}$$

and the best linear unbiased estimate $\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$ is obtained by normal equation where $\mathbf{y}$ is a vector of observed responses corresponding to input vector $\mathbf{X}$.

Regression Through the Origin

Consider simple linear regression with $p = 1$. Then in (2.1), the parameter $\hat{\beta}_0$ is the intercept and $\hat{\beta}_1$ is the slope of the straight line in the X-Y plane. In some applications, a *no-intercept regression model* is more appropriate providing sensible interpretation in analyzing data (Eisenhauer (2003) [15]). In economics literature, Theil (1971) [57] also argues "From an economic point of view, a constant term usually has little or no explanatory virtues." Nevertheless, careful data exploration is necessary to avoid misuse of the no-intercept model, especially when the data lies at some distance from the origin because the relationship between X and Y near the origin and distant from the origin may be different. Given N observations (x_i, y_i) $(i = 1, \cdots, N)$, the simple linear regression model without intercept has a form,

$$Y = f(X) = X^T\beta_1 + \epsilon \tag{2.3}$$

where X and Y are an input vector and real-valued output vector respectively, and ϵ is independent random error vector with zero mean and finite variance. The least-squares is

$$\text{RSS}(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} (y_i - x_i\beta_i)^2 \tag{2.4}$$

and the least-square estimator of the slope is given by

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N} y_i x_i}{\sum_{i=1}^{N} x_i^2}. \tag{2.5}$$

Note that by forcing the line to go through the origin, the sum of the residuals is not zero, which implies the estimator in (2.5) is biased.

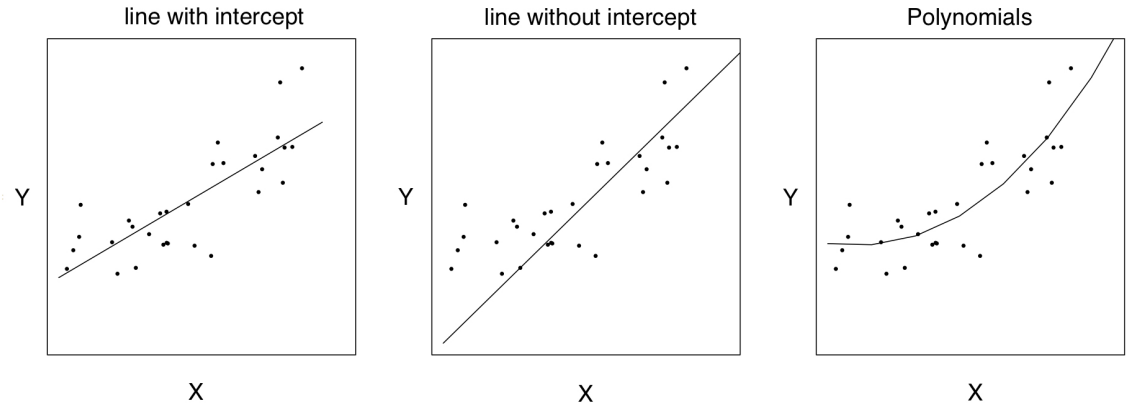

Figure 2.1: Linear regression with and without intercept, and polynomial regression

Polynomial Regression

Polynomial regression is the extension of simple linear regression to the extent where the relationship between the predictors and the response is non-linear. The form of polynomial regression is given by

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 \cdots + \beta_d x_i^d + \epsilon_i \tag{2.6}$$

where ϵ_i is the i-th error term. Although the fitted line is allowed to be a non-linear curve

with high degree d, it is rare that d is higher than 3 or 4. The estimation of coefficients in (2.6) is the same as that of ordinary linear regression in (2.2). Usually the individual coefficients are not of interest in polynomial regression but rather the relationship between the predictor and the response over the whole range of the predictor.

2.1.2 Piecewise Polynomials and Splines

Although linear or polynomial regression models have been popular because of many nice statistical properties such as consistency, efficiency, and unbiasness of the estimates, it is not likely possible to build a single global linear or polynomial regression model on very large size datasets. Piecewise polynomial functions and splines are useful approximation methods for lengthy and evolving data in random manners. They are obtained by dividing the domain X into contiguous intervals and building local models.

Denote $h_m(X) : \mathbb{R}^p \mapsto \mathbb{R}$ as the m th transformation of X, $(m = 1, 2, 3 \cdots, M)$. Piecewise polynomial splines has a form,

$$f(X) = \sum_{m=1}^{M} \beta_m h_m(X) \tag{2.7}$$

where $\{h_1(x), h_2(x), \cdots, h_M(x)\}$ is a set of known linear basis functions. The form of bias function h_m is allowed to be various transformations of X, for example, log transformation or indicator function that defines range of X. Here, we review some forms of basis function $h_m(X)$ relevant to linear or cubic approximation functions of X. The piecewise constant

polynomial is the simplest linear approximation of data. Its basis functions are,

$$\begin{aligned} h_1(X) &= I(X < \xi_1) \\ h_2(X) &= I(\xi_1 \le X < \xi_2) \\ &\vdots \\ h_M(X) &= I(\xi_{M-1} \le X) \end{aligned} \tag{2.8}$$

The points $\xi_1, \xi_2, \cdots, \xi_{M-1}$ are *knots* where the coefficients $\beta_i,\ (i = 1, 2, \cdots, M)$ change and the new piecewise model starts. In piecewise constant splines, the coefficients $\beta_i,\ (i = 1, 2, \cdots, M)$ is the mean of data in the i-th interval (Figure 2.2 left).

Piecewise linear splines fits a linear models in each segment. Since two parameters, intercept and slope, are required for the linear model per each segment, the total number of parameters is simply $2M$, where M is the number of segments (Figure 2.2 middle). With continuity restrictions, there are constraints such that $f(\xi_i^-) = f(\xi_i^+),\ i = 1, 2, \cdots, M-1$ at every knot. For example, $\beta_1 + \beta_4\xi_1 = \beta_2 + \beta_5\xi_1$ at knot ξ_1 and now we have three parameters to estimate for these two successive segments because of this restriction. Generally, with continuity restrictions, the number of parameter (or *degrees of freedom*) for piecewise approximation is

$$Mp - (M-1)c \tag{2.9}$$

where M is the number of segments, p is the number of parameters required per each segment, and c is the number of constraint at each knot. Thus in the example of Figure 2.2 right, there are $4 \times 2 - 3 \times 1 = 5$ parameters to estimate. Direct representation can be made for piecewise approximation using *truncated power basis functions*.

Definition 2.1. A n degree truncated power basis function is defined by

$$h(x-\xi) = (x-\xi)_+^n = \begin{cases} (x-\xi)^n, & \text{if } x > \xi \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

The function in the right of Figure 2.2 can be written as a linear combination of $h_1(X) = 1,\ h_2(X) = X,\ h_3(X) = (X-\xi_1)_+,\ h_4(X) = (X-\xi_2)_+$, and $h_5(X) = (X-\xi_3)_+$. These piecewise approximations are able to be more smooth globally by increasing the order of degree of polynomials. Use of *cubic splines* is a common approach to obtain a globally smoothed function. In cubic splines each segment allows to fit cubic polynomials. Generally, splines of orders higher than three (cubic) are seldom used for function approximation because they tend to overfit and requires intensive computation. For continuity restrictions, cubic splines may have two or three constraints at knots. The cubic splines with three constraints at knot ξ,

$$\begin{aligned} &(i) \quad f(\xi^-) = f(\xi^+) \\ &(ii) \quad f'(\xi^-) = f'(\xi^+) \\ &(iii) \quad f''(\xi^-) = f''(\xi^+) \end{aligned} \tag{2.11}$$

is called *natural cubic splines*. Cubic splines with two constraints (i) and (ii) does not guarantee a smooth connection at knots. Natural cubic splines can be optimized by satisfying,

$$\min RSS(f,\lambda) \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int \left(f''(t)\right)^2 dt \tag{2.12}$$

where λ is a fixed *smoothing parameter* (penalty for roughness). It is also known as *regression splines* when knots are fixed. Intuitively, knots should be located where the function changes rapidly rather than where it is stable. There have been various ways suggested to

determine the number of knots and their locations. The number of knots are not fixed in *penalized splines*, or *P-splines*. Instead, they use fixed quantiles of independent variables with fixed roughness penalty (Ruppert (2012) [51]). *B-splines* minimize the number of segments with respect to a given degree and smoothing penalty from the augmented knots (De Boor (1978) [13]).

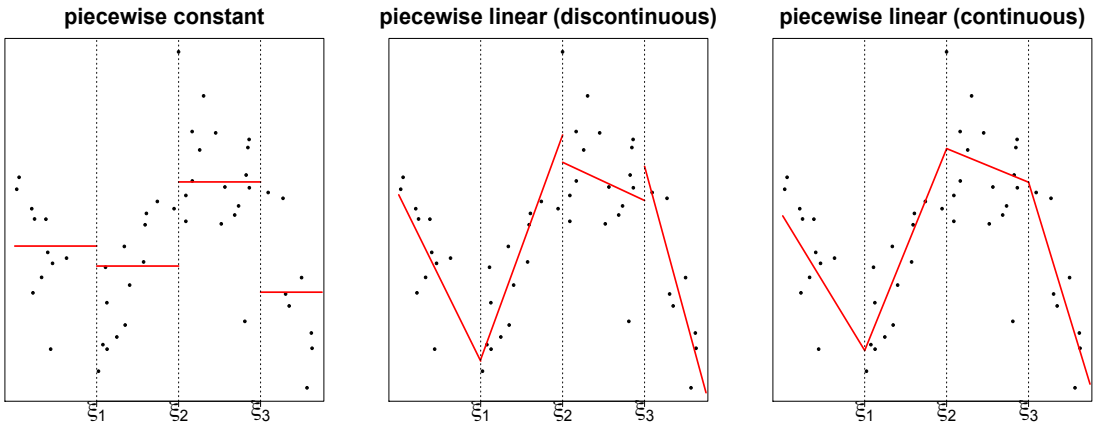

Figure 2.2: Piecewise linear approximation

2.1.3 Batch and Online Processing Methods

Datasets may be preprocessed in different ways based on the their availabilty. There are three common processing methods; batch processing, online processing, and a combination of the two processing methods (Shelly et al. (2009) [53]). Here, we discuss batch processing and online processing related to our approaches to represent data introduced in Chapter 3 and Chapter 4.

Batch Processing

Batch processing may be used when all the data are available during all the computations.

Therefore, it is possible to understand characteristics and structure of data before analyzing it. Gentle (2016) [21] discusses two approaches of piecewise constant models by batch method. One simple approach to approximate massive dataset is to take representative values, for example means, sum of variations (Lee et al. (2003) [40]) or volatility, of each segment and use sample quantiles of data corresponding models.

Another way is to build a model for the whole data and then detect outliers to identify influential points on the shape of data. By repeating this process recursively with selected points and stopping based on optimization criteria, the full dataset can be approximated preserving its shape character. Chung et al.[12] propose perceptually important points (PIP) as a data reduction process to find technical analysis patterns in financial applications. PIPs are detected based on the deviation from the most recently found PIPs recursively. Generally, the computing cost of the batch method is less than that of online method.

Online Processing

Online processing analyzes data immediately while datasets are provided, thus the user obtains the results in less time. The online method is preferred in time series and massive data mining because most data are accumulated continuously over time. Online processing allows new datasets during all the computations. Online processing has an advantage that no additional post-processing is necessary although it costs more than batch method for computation.

2.1.4 Loss Function and Regularization

A loss function, also known as a cost function, provides a quantitative assessment of estimates. Although we should be clear with the difference between *estimation* and *approximation* of functions (Gentle (2002) [29]), we may use them interchangeably in the sense that both aim to describe the characteristics of the original data in simpler and more efficient forms while minimizing the loss of information. The most popular loss function is

squared-error loss. Squared-error loss function is defined by

$$L(y, f(x)) = (y - f(x))^2 \tag{2.13}$$

at point x.

Risk (or *mean squared error* (MSE)) is the expectation of squared loss. The solution that minimizes squared error loss is $f(x) = \mathrm{E}(Y|x)$. Another popular loss function is *absolute loss*

$$L(y, f(x)) =\mid y - f(x) \mid \tag{2.14}$$

and its solution is given by $f(x) = \text{median}(Y|x)$. With the finite number of samples, squared-error loss is widely used because it yields unique unbiased estimates in closed form while absolute loss may not have a unique solution and requires expensive computation to obtain solutions in general. However, squared-error loss is sensitive to outliers in the modeling process, and thus is less robust than absolute loss and poorly performs with long-tailed error distributions. Often, the optimization problem is to balance between preserving the information and overfitting. Generally, the objective function for optimization has the form as follows

$$\min \sum_{i=1}^{N} L\left(y_i, f(x_i)\right) + \lambda R(f) \tag{2.15}$$

where $R(f)$ is regularization term and λ is a tuning parameter which determine the importance of the regularization.

2.2 Time Series Data Representation

Before we define patterns of the data, data representation or data preprocessing is required to reduce the dimension of data to manageable size. That is, the amount of data points are reduced by data representation. Each element in this reduced data sequence may be a transformed value that represents important features of the original data. Time series data can be represented in time domain or frequency domain. The latter represents time series data in frequency domain using discrete Fourier transforms (DFT) or discrete wavelet transforms (DWT). Although frequency domain transformation methods have been popular for periodic data or for image data analysis since Agrawal et al. (1993) [6] proposed, time domain models would be better approaches for up/down pattern analysis because time information is an important feature with trends. In this chapter, time domain data representation methods are discussed. We classify the data representation methods into three categories based on the criteria of data reduction referring to Fu (2011)'s classification [18].

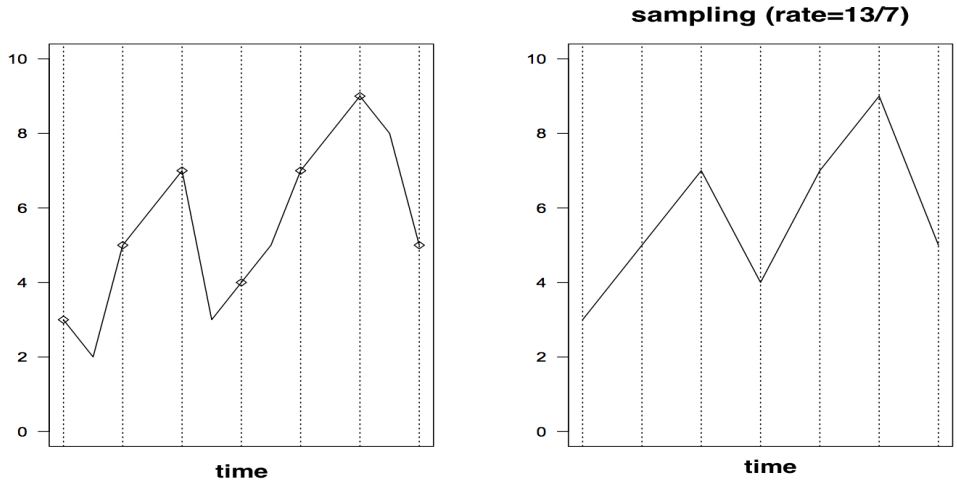

Figure 2.3: Data approximation by sampling with rate = 13/7

2.2.1 Piecewise Approximation

Generally the first step of piecewise approximation is to divide the time series into some number of non-overlapping segments. There might be two ways to represent each segment. One method is using any statistic of the data within the segment. The other method is fitting a straight line.

Piecewise Information Summary

The idea behind of piecewise information summary is to split the whole data into some number of segments and use a representative information, for example mean value or variance, of each segment. Astrom (1969) [8] proposes sampling points at equal spacing h which defines sampling rate $\frac{N}{h}$, where N is the length of time series and show there is an optimal choice of h. A drawback of sampling method is when h deviates from the optimal value, it misrepresents the shape of the original data. (Figure 2.3) Lee et al. (2003) [40] propose SSV-indexing. They use *segmented sum of variation* (SSV) for data representation which does not need vertical shifting for data registration. (Figure 2.4 left) Ratanamahatana et al. (2005) [49] propose using bit level data for approximation. Bit level approximation converts each data point,

$$p_i = \begin{cases} 1 & \text{if } p_i > \mu \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

where μ is the mean of given time series, and then represents the converted binary data by run length encoding (RLE). "@" and "!" are used to identify whether the converted binary data begins with 0 or 1. For example, if we have converted data sequence of bit, **0001100011110**, it can be represented as **@3,2,3,4,1** which means that it starts with three zeros, two ones, three zeros, and so forth (Figure 2.4 right). The mean value is also commonly used to represent each segment of data. Yi et al. (2000)[62] use *segmented mean* of equal spaced intervals. Keogh et al. (2001a) [33] propose *piecewise aggregate approximation*

(PAA) for large time series databases. It also divides the original sequence with length n into equal sized N intervals (Figure 2.5). As an enhanced idea of PAA, *adaptive piecewise constant approximation* (APCA) is proposed (Keogh et al. (2001a) [34]). APCA uses Haarr wavelet transformation to identify breakpoints and thus allows various lengths of segmented intervals.

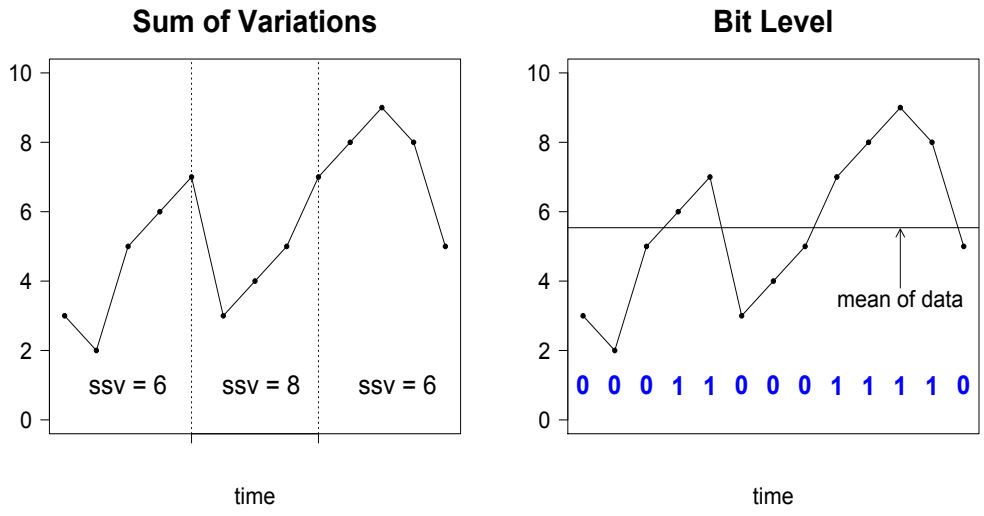

Figure 2.4: Sum of variations = (6,8,6) and bit level approximation

Piecewise Linear Approximation

Piecewise linear approximation is representing each interval as a line based on linear interpolation or regression. In this sense PAA can be considered a piecewise linear approximation as well. *Piecewise linear representation* (PLR) is linear interpolation proposed by Keogh (1997) [32] and Smyth and Keogh (1997) [55]. PLR uses a *bottom-up* algorithm. The algorithm divides the original data so that each segment contains a minimum of three data points, and connects adjacent breakpoints. Then it keeps merging adjacent segments sequentially based on a merging criteria, that is whether the standard deviation of total residuals is reduced by merging, recursively until the optimal number of segments is achieved. It chooses two adjacent segments that yield the greatest reduction in the standard deviation of total

residuals and repeats this process until the standard deviation of total residuals begins to increase. Keogh and Pazzani [37] [38] enhance PLR idea combining weighting influence and user feedback on segment. Shatkay and Zdonik (1996) [52] propose a notion of *generalized approximate queries* which approximates time series data using linear regression by online algorithm.

Figure 2.5: PAA for the daily stock price of Amazon Inc. (11/02/2011 − 10/25/2015)

2.2.2 Preserving Important Points

While piecewise approximation methods summarize a handful of data points in segments of the original data and represent them as a subsequence of information, data representation methods preserve important observations by selecting a group of meaningful points. Chung et al. (2001) [12] introduce *perceptually important points* (PIP). Searching process for PIPs

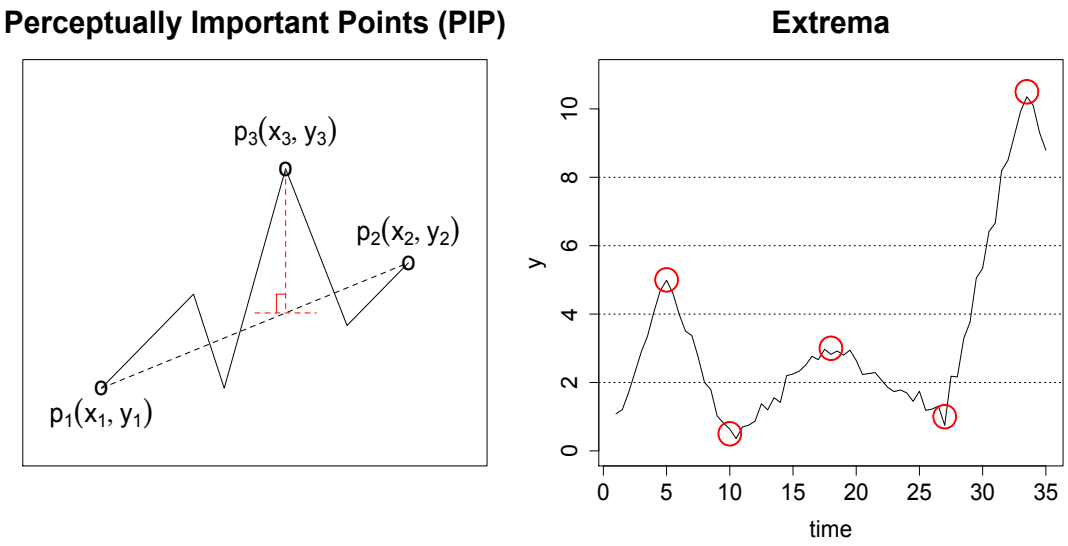

Figure 2.6: PIP by vertical distance (left) and Extrema (right)

is as follows. For time series data with length n, $x_1, x_2, \cdots, x_n$, the first and the last two points $P_1 = x_1$ and $P_2 = x_n$ are always PIPs. The third PIP P_3 is the point that has the maximum vertical distance from the straight line between P_1 and P_2. The fourth PIP is the point that locates in the maximum vertical distance from either $\overline{P_1P_3}$ or $\overline{P_2P_3}$ where $\overline{P_iP_j}$ is the straight line between P_i and P_j. In this fashion, it proceeds to search k-th PIP that has maximum vertical distance from $\overline{P_iP_{i+1}}$, $i = 1, 2, \cdots, k-1$ until it reaches predefined number of PIPs (Figure 2.6, left). Fu et al. (2008) [19] employ Euclidean and perpendicular distance to identify PIPs. This method is a batch process. Pratt and Fink (2002, 2003) [47] [17] propose "important" points which cause major fluctuation and represent the original time series for only these selected important points. Extracting important extrema can be performed by batch or online method. Bao (2008) [10] represents financial data by connecting critical points to identify technical patterns of stock prices. Critical points are selected based on the amount of oscillation that exceeds the thresholds. In other words, these points contribute most critically to summarize the shape of data.

2.2.3 Discretizing Numeric Time Series

Data representation methods by discretizing numeric time series transform numeric data into categorical data. It is also called *symbolic representation* because often, symbols are used as categorical variables. Yang and Zhao (1998) [61] and Yang et al. (1999) [60] propose a symbolic data representation technique that converts time series data into symbols, either 0 or 1, based on a threshold function. *Shape Description Alphabet* (SDA) is proposed by Jonsson and Badal (1997) for *blurry matching*. SDA transforms data into a few number of letters that represent magnitude of transition from one point to the next point using slope value. Another popular symbolic representation method is *symbolic aggregate approximation* (SAX) proposed by Lin et al. (2003) [42] [43]. SAX preprocesses the data in two steps. First, it represents normalized data by PAA and next converts the sequence of PAA coefficients into alphabetical strings. Hence, SAX representation depends on two parameters, symbol size a and the dimension of the reduced data $w \ll n$ (the length of PAA coefficients) where n is a length of time series. Symbol size a is the number of letters available that corresponds to the number of equal probability area under standard normal distribution and the breakpoint set. Specifically, with symbol size a, we have a set of letters $\{L_1, L_2, \cdots, L_a\}$ and a breakpoint set $\{\beta_1, \beta_2, \cdots, \beta_{a-1}\}$ such that $P(Z < \beta_1) = P(\beta_1 \leq Z < \beta_2) = \cdots = P(\beta_i \leq Z < \beta_{i+1}) = \cdots = P(\beta_{a-1} \leq Z)$ where $Z \sim N(0,1)$. In the second step of SAX, the sequence of PAA coefficients $\{\bar{c}_1, \bar{c}_2, \cdots, \bar{c}_w\}$ are transformed to a string sequence $\{\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_w\}$ by the following (2.17).

$$\hat{c}_i = L_j \quad \text{if and only if} \quad \bar{c}_i \in [\beta_{j-1}, \beta_j) \tag{2.17}$$

A new distance metric MINDIST in SAX has some advantages from the use of PAA (Figure 2.7).

Definition 2.2. Suppose a symbol size parameter is a. Given two time series Q and C of the same length n, let $\hat{Q} = \{\hat{q}_1, \hat{q}_2, \cdots, \hat{q}_w\}$ and $\hat{C} = \{\hat{c}_1, \hat{c}_2, \cdots, \hat{c}_w\}$ be transformed

string sequences of Q and C respectively by (2.17), where w is a parameter the number of segments. Then the MINDIST function is defined by

$$\text{MINDIST}(\hat{Q}, \hat{C}) \equiv \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} \left(dist(\hat{q}_i, \hat{c}_i)\right)^2} \tag{2.18}$$

where $dist()$ is a function implemented by a $r \times c$ lookup table. The value in the cell (r, c) for lookup table is decided by the following expression.

$$cell_{(r,c)} = \begin{cases} 0 & : \text{ if } |r - c| \geq 1 \\ \beta_{\max(r,c)-1} - \beta_{\min(r,c)} & : \text{otherwise} \end{cases} \tag{2.19}$$

where $\beta_1, \beta_2, \cdots, \beta_{a-1}$ are breakpoints such that the area under the standard normal distribution from β_i to $\beta_{i+1} = 1/a$ (β_0 and β_a are defined as $-\infty$ and ∞, respectively).

MINDIST lower bounds Euclidean distance, that is for two given time series Q and C, $\text{MINDIST}(Q, C) \leq D_{EU}(Q, C)$, where $\text{MINDIST}(Q, C)$ and $D_{EU}(Q, C)$ are distances between Q and C by MINDIST and Euclidean distance respectively. This is an important property for indexing time series. For example, when a time series C in database is very similar to the query series Q by true Euclidean distance, any distance measures that do not lower bound Euclidean distance may fail to retrieve C (false-negative). MINDIST guarantees no false-negative by lower bounding Euclidean distance. Additionally, symbolic representation enables not only to save memory space efficiently but also to compute more quickly by using bits (character) rather than double format (real-values).

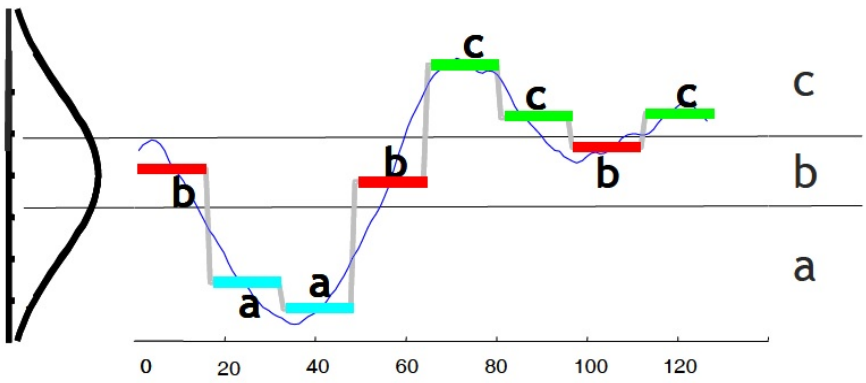

Figure 2.7: Coefficients from PAA approximation are mapped into SAX symbols. The number of raw data points=128, $w = 8$, and alphabet size $a = 3$. The time series represented by a sequence of string **baabccbc** (Lin et al. [43]).

2.3 Distance Measures and Pattern Recognition

The use of appropriate similarity/distance measures is crucial to obtain meaningful data mining results. Indeed, the unique characteristics of time series data, large size and the presence of noise, have led to various methods for measuring distance to be developed. In time series data mining, the choice of distance measure is closely relevant to data representation because we measure distance between two represented datasets not raw datasets. Specifically, if the represented data are sequences of different length, one may have to look for a distance measure that allows two objects with different lengths, for example dynamic time warping (DTW) as I wil discuss on page 23, instead of L^p distance. In this section we review several similarity/distance measures related to trends analysis of financial time series and pattern recognition. More about distance measures of time series are discussed in Liao (2005) [41] and Aggarwal et al. [5].

2.3.1 Metric and Distance Measures

The measures of the distance between two observations or variables are often a form of metric function Δ from $\mathbb{R}^m \times \mathbb{R}^m$ into $\mathbb{R}$ which satisfies the following properties (Gentle (2002) [29]).

- $\Delta(x_1, x_2) \geq 0$ for all $x_1, x_2 \in \mathbb{R}^m$ (non-negativity)
- $\Delta(x_1, x_2) = 0$ if and only if $x_1 = x_2$ (Identity of indiscernibles)
- $\Delta(x_1, x_2) = \Delta(x_2, x_1)$ for all $x_1, x_2 \in \mathbb{R}^m$ (symmetry)
- $\Delta(x_1, x_3) \leq \Delta(x_1, x_2) + \Delta(x_1, x_2)$ for all $x_1, x_2, x_3 \in \mathbb{R}^m$ (triangular inequality)

where x_1, x_2 and x_3 are observed data points in $\mathbb{R}^m$.

The L_p-norm is the most commonly used distance metric. There is no single superior method over all others. The decision of the most appropriate distance measure in time series mining depends on its form of representation and goal of application.

L_p Distance

The dimension of two data must be the same to use L_p norm. Let T_1 and T_2 each be a p dimensional vector. The L_p norm distance between T_1 and T_2 is defined by

$$D_L(p)(T_1, T_2) = \left(\sum_{i=1}^{n} (T_{1i} - T_{2i})^p \right)^{\frac{1}{p}} \tag{2.20}$$

where T_{ki} $(k = 1, 2)$ is i-th element of vector T_k. Euclidean distance, maximum distance, and Manhattan distance are special cases when $p = 2,\ p \to \infty$ and $p = 1$ respectively.

Distance Based on Pearson's Correlation Coefficient

Correlation or covariance is a simple method to measure similarity for numerical data. To use correlation coefficient, the length of data also must be the same. Let T_i and T_j each be a p dimensional vector. Pearson's correlation coefficient between T_i and T_j, $\rho(T_i, T_j)$ is defined by

$$\rho(T_i, T_j) = \frac{\sum_{k=1}^{p} (T_{ik} - \overline{T}_i)(T_{jk} - \overline{T}_j)}{S_{T_i} S_{T_j}} \tag{2.21}$$

where

$$\overline{T}_i = \frac{1}{p}\sum_{k=1}^{p} T_{ik} \quad \text{and} \quad S_{T_i} = \left(\sum_{k=1}^{p}(T_{ik} - \overline{T}_i)\right)^{\frac{1}{2}} \tag{2.22}$$

Golay et al. (1998) [22] propose two cross-correlation-based distances in the fuzzy c-means algorithm as follows.

$$d^1 = \left(\frac{1-\rho}{1+\rho}\right)^{\beta} \quad \beta > 0 \quad \text{and} \quad d^2 = 2(1-\rho) \tag{2.23}$$

Dynamic Time Warping (DTW)

Dynamic time warping (DTW) is a popular algorithm for measuring similarity in time series data mining. It searches the optimal alignment between two time series by warping time satisfying tree conditions, *boundary, monotonicity*, and *step size* conditions. Originally DTW has been developed in studies of automatic speech recognition. DTW is a useful method to compare two time dependent data with various time intervals.

DTW is not a metric because triangular inequality is not satisfied. When a distance measure does not satisfy triangular inequality, it may result in a false negative response. However, this does not imply DTW is less appropriate than other distance metric considering its advantages and great performance in some applications such as speech recognition.

For two time series $X = \{x_1, x_2, \cdots, x_M\}$ and $Y = \{y_1, y_2, \cdots, y_N\}$, DTW aligns two time series to have the minimal distance between X and Y. To achieve this, we need *local cost measure*. Formally, let $\mathcal{F}$ be sample space of time series data and $X_m, Y_n \in \mathcal{F}$ and $1 \leq n < N,\ 1 \leq m < M$. The *local cost measure* c is defined by

$$c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}^+ \tag{2.24}$$

Local cost measure evaluates the cost by calculating Euclidean distance between each pair of X and Y and thus *cost matrix* $C \in \mathbb{R}^{M \times N}$ can be obtained from a local cost measure. DTW searches the warping path that costs least from C_{11} to C_{MN} where C_{ij} is i-th row

and j-th column of cost matrix C. There are three conditions for warping path alignment. Müller (2007) [44] formalizes as follows.

Definition 2.3. Given time series data $X = \{x_1, x_2, \cdots, x_M\}$ and $Y = \{y_1, y_2, \cdots, y_N\}$, a warping path is a sequence $p = (p_1, p_2, \cdots, p_L)$ with $p_l = (m_l, n_l)$ where $1 \leq m \leq M,\ 1 \leq n \leq N$ and $1 \leq l \leq L$, satisfies the following three conditions.

- Boundary condition: $p = (1, 1)$ and $p_L = (M, N)$
- Monotonicity condition: $m_1 \leq m_2 \leq \cdots \leq m_L$ and $n_1 \leq n_2 \leq \cdots \leq n_L$
- Step size condition: $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $1 \leq l \leq L - 1$

With three constraints in definition 2.3 DTW searches the warping path p^* that has the minimal cost among all possible paths. Hence the *DTW distance* is defined by

$$DTW(X, Y) = c_{p^*}(X, Y) = \min \{c_p(X, Y) |\ p \text{ is an } (M, N)\text{-warping path}\} \tag{2.25}$$

where c_p is the local cost measure $c_p(X, Y) = \sum_{l=1}^{L} c(x_{ml}, y_{nl})$. It is a great advantage of DTW to be able to apply different lengths of data and provide reliable time alignment for pattern recognition. However, computational cost to search the optimal alignment among all possible warping path is very high especially for large amounts of data. To speed up computation in DTW algorithms, some variations have been proposed by modification of step size condition or imposing global constraint regions. *Sakoe-Chiba band* and *Itakura parallelogram* are two well-known global constraint regions.

Short Time Series Distance (STS)

Möller-Levet et al. (2003) propose *short time series distance* (STS) for data represented in piecewise linear functions. STS defines the distance by the sum of the squared differences of slopes given a sequence of segments of time. Let $X = \{x_0, x_1, \cdots, x_n\}$ be a time series data with length n and $x(t) = m_k t + b_k$ be the linear function between two successive breakpoints

t_k and t_{k+1} where,

$$m_k = \frac{x_{k+1} - x_k}{t_{k+1} - t_k}, \qquad \text{and} \quad b_k = \frac{t_{k+1}s_k - t_k x_{k+1}}{t_{k+1} - t_k} \tag{2.26}$$

The STS between two time series $X = \{x_0, x_1, \cdots, x_n\}$ and $Y = \{y_0, y_1, \cdots, y_n\}$ is defined by,

$$d^2_{STS}(X, Y) = \sum_{k=0}^{n-1} \left(\frac{y_{k+1} - y_k}{t_{k+1} - t_k} - \frac{x_{k+1} - x_k}{t_{k+1} - t_k} \right)^2 \tag{2.27}$$

STS is particularly suited for short-time series data with unevenly distributed sample points and the same length. Standardization of the original data is recommended since the STS and the Euclidean distance are both sensitive to scaling.

2.3.2 Pattern Recognition in Financial Time Series

According to the efficient market hypothesis, it is impossible to forecast asset prices because they are fully determined by present information which neither can be known ahead of time nor predictable from the past. That is, asset prices evolve in random manner, so-called *random walk* (Bachelier (2011) [9]). Nevertheless, there have been continuous endeavors to develop statistical forecasting models in the domain of finance and economics in the belief of existence of rules or patterns in financial data, and some of the models seem to be performing reasonably in practice. There are two main methods in financial market data analysis; fundamental analysis and technical analysis.

Fundamental analysis requires *fundamental* indicators such as earnings and growth which represent *intrinsic values* of companies to evaluate the values of the assets. The actual values of assets, for example stock prices, would be compared to these fundamental indicators so that investors determine their position to make profit. On the other hand, technical analysis depends only on *technical* information generated from market activity such as open, high, low, and close prices. Technical analysis does not use "fundamental" information. Instead, it uses technical indicators, for instance moving average, trend,

volatility etc. Although technical analysis should be ineffective under the efficient market hypothesis, it is believed that *momentum* caused by psychological, social and emotional factors at times enables the prediction of asset prices and their trends by technical analysis with some degree of accuracy. This behavioral economic theory has motivated researchers to develop data-driven modeling with financial data.

Conventional parametric statistical ARIMA or GARCH types of models have been broadly used for prediction and investigation of data generating process. However, these methods do not seem plausible to predict long-term trends, a major interest of financial data analysis. Thus, data mining techniques for trends analysis have been attracted along with the enhancement of computing technologies. In this section we review methods of pattern recognition in the context of financial data analysis.

Shape-Based Pattern Recognition

In financial times series data analysis, an overarching interest is to forecast *trends*. Generally trends of financial data are explained by directions of evolving data, for example "up" , "down" and "stable". Shape-based pattern recognition techniques aim to identify the directions or patterns of data evolution based on the shape of the original data. As a result, in most shape-based pattern recognition the original data are represented by piecewise function approximation or preserving important points that intend to contain the characteristics of the shape. Guo et al. (2007) [24] represent data by piecewise linear approximation by bottom-up algorithm and classify every seven segments (or less than seven) into one of the eighteen typical technical patterns, for example descending triangle, head and shoulder etc. Eight breakpoints including the starting and ending of the seven are used for input feature for neural network classifier. Bao (2008) [10] uses *critical points model* (CPM) to recognize three trend type patterns; the broadening pattern, the triangular pattern, and the rectangular pattern.

Attribute-Based Pattern Recognition

Attribute-based pattern recognition models address patterns of some feature values rather

than the “up” or “down” direction of data, for example ranges of daily returns or even raw price. Thereby, in the attribute-based model, data representation depends on some statistics of data for example piecewise volatility or variations, or some times immediately use raw data values without data preprocessing. Moreover, while shape-based pattern recognition model is unlikely to use fundamental information as a predictor, attribute-based models often use fundamental indicators, such as dividend ratio and earning per share. Pathak (2014) [46] uses open, high, low, close, and volume data for predictive model of stock price using neural network. Quah (2008) [48] uses eleven fundamental indicators as input variables of neural networks for classification of appreciation of equities.

Chapter 3: Identification of Trend Changepoints

In piecewise function approximation, identifying breakpoints and fitting the local function in each segment are closely related to the characteristics of data. In case all of the data is available, it is possible to understand the underlying distribution of data so that we may decide the breakpoints by predefined quantity or parameters. However, data representation by predetermined breakpoints is not likely to represent suitable local trends and features efficiently, particularly for continuously updated data with new observations. In this chapter, we propose new data representation methods by piecewise linear smoothing; *alternating trends smoothing* (ATS) and *piecewise band smoothing* (PBS), and introduce important parameters and their properties. Then we discuss the extended basis function to high degree polynomials and compare the empirical results of two data representation methods; piecewise approximation (1) by linear basis and (2) by linear and quadratic basis.

3.1 Alternating Trends Smoothing

In technical analysis of financial market data, identifying "increasing" or "decreasing" trends and their changepoints is of major interest. A *point and figure chart* is a primitive technical analysis tool to represent price movements. The point and figure chart represents "up" and "down" trends by rising "X" column and falling "O" column respectively as it is shown in Figure 3.1 (right). While the point and figure chart is simple and efficient to represent "up" and "down" direction of data and identify their changepoints, it does not contain the time information between changepoints since it represents data based on price action not on the time.

Gentle (2012) [20] introduces *Alternating Trends Smoothing* (ATS), a piecewise linear smoothing method by alternating up and down trends. It detects trend changepoints by

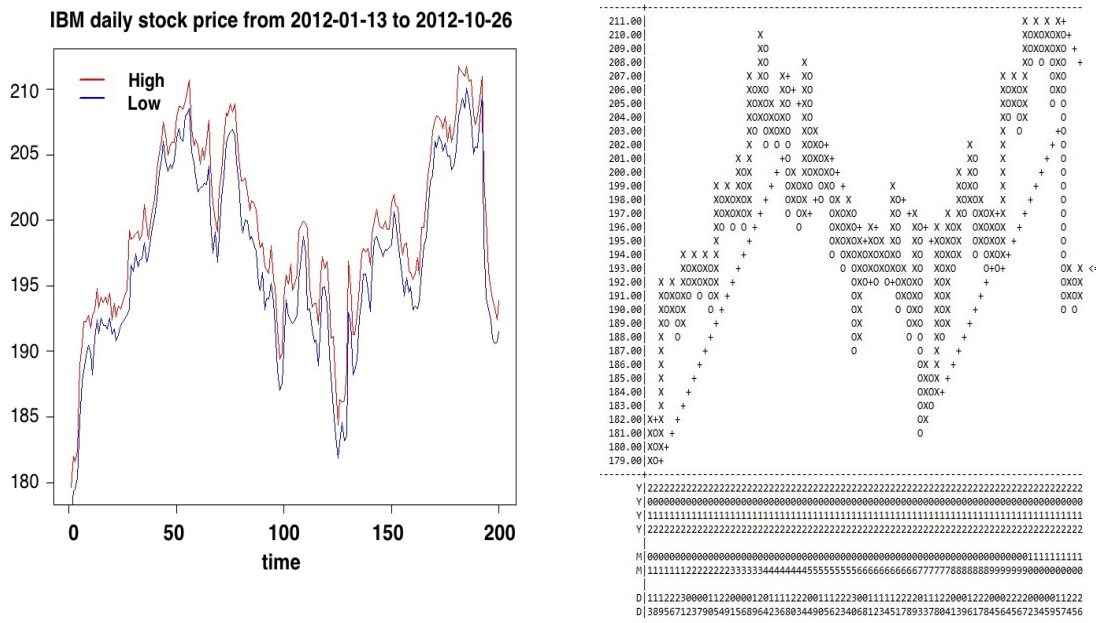

Figure 3.1: IBM stock price and *point and figure* chart

examining whether any point among newly updated h data points (a parameter of step size) contributes to a change in the current trend direction. The algorithm of ATS is given in **Algorithm** 3.1.

Given time series data $X = \{x_{t_1}, x_{t_2}, \cdots, x_{t_N}\}$ the output of **Algorithm** 3.1 is

$$(b_1, c_1), (b_2, c_2), \cdots, (b_{n-1}, c_{n-1}), \quad (n \ll N) \tag{3.1}$$

where $b_1 = t_1, b_2 = t^{(2)}, \cdots, b_k = t^{(k)}, \cdots, b_n = t^{(n)}$ and $c_1 = x_{t_1}, c_2 = x_{t^2}, \cdots, c_k = x_{t^{(k)}}, \cdots, c_n = x_{t^{(n)}}$. That is, b_k is the time when the $(k-1)$-th trend change occurs, and c_k is the value at b_k. Furthermore, we obtain the sequence of trends, $S = \{s_1, s_2, \cdots, s_n\}$ by defining

$$s_1 = \frac{c_2 - c_1}{b_2 - b_2}, \cdots, s_k = \frac{c_{k+1} - c_k}{b_{k+1} - b_k}, \cdots, s_{n-1} = \frac{c_n - c_{n-1}}{b_n - b_{n-1}}, \quad (n \ll N). \tag{3.2}$$

The s_k is the slope of k-th trend regime by simply connecting two adjacent change-points. The method approximates the original time series data by alternating increasing

Algorithm 3.1 Alternating Trends Smoothing

Set $d \leftarrow 1$ (changepoint counter)
Process data within first time step:
while more data **do**
    for $i = 1, 2, \cdots, m$ ($m = h$ if h additional data available, or m is last data item) **do**
        input x_i;
        $b_d \leftarrow 1; c_d \leftarrow x_1$
        Determine $j_+, j_-, x_{j_+}, x_{j_-}$ such that
            $x_{j_+} = \max(x_1, x_2, \cdots, x_h)$ and $x_{j_-} = \min(x_1, x_2, \cdots, x_h)$
        Set $s = (x_k - x_i)/(k - i)$ and $r = sign(s)$
        while $r = 0$ **do**
            Continue inputting more data; stop with error at end of data
        end while
    end for
end while

Set $j \leftarrow i$ (index of last datum in previous step); and set $d \leftarrow d + 1$
while more data **do**
    for $i = j + 1, j + 2, \cdots, j + m$ ($m = h$ if h additional data available, or $j + m$ is last data item) **do**
        Input x_i;
        while $sign(s) = r$ **do**
            Set $k \leftarrow \min(i + h, n)$ where n is the number of data points
            if k=i **then** break
            end if
            Set $s \leftarrow (x_k - x_j)/(k - j)$
            Set $j \leftarrow k$
        end while
        Determine j_+ such that $rx_{j_+} \leftarrow \max(rx_{j+1}, \cdots, rx_{j+m})$
        Set $b_d \leftarrow j_+$; and set $c_d \leftarrow x_{j_+}$
        Set $d \leftarrow d + 1$; set $j \leftarrow j_+$; and set $r \leftarrow -r$
    end for
    Set $b_d \leftarrow j_+$; and set $c_d \leftarrow x_{j_+}$
end while

and decreasing straight lines and reduces its original dimension N to n, $(n \ll N)$ as seen in (3.2), where n is the length of the reduced data.

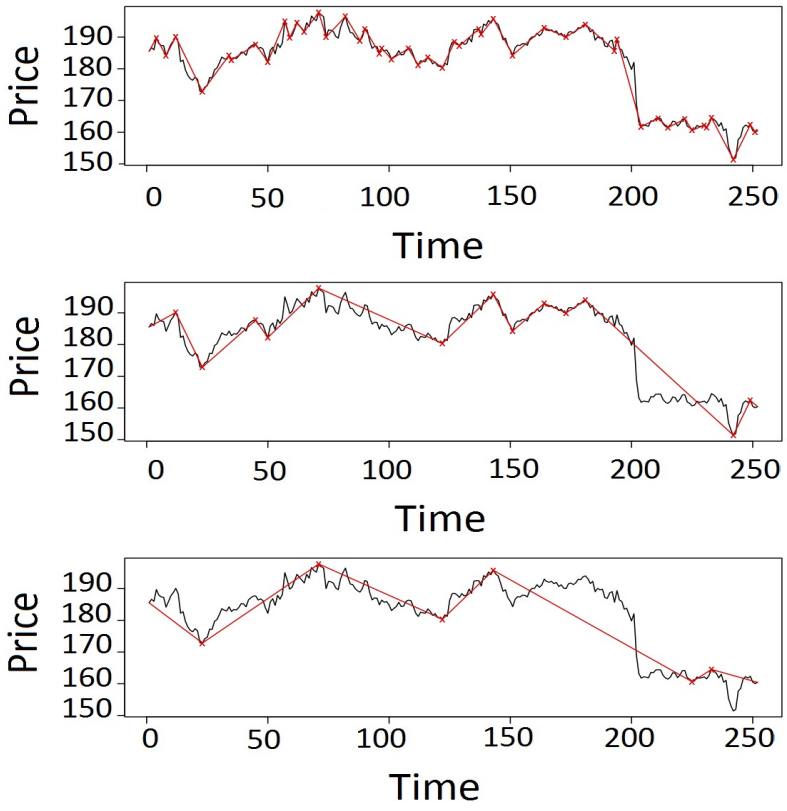

Figure 3.2: IBM stock price with various step sizes of ATS

The Tuning Parameter: Step Size h

ATS approximates data based on the tuning parameter h which specifies "step size". It moves h data points at a time to look for changepoints and once it finds a changepoint, it moves h steps again from the changepoint. The larger value h tends to find fewer changepoints because with a large step size, it has less opportunities to check if a point is changepoint. However, the distance between changepoints can be smaller than h even as small as one step. Figure 3.2 shows ATS representations of daily closing prices of International Business Machine Corporation (NYSE:IBM) from January 1, 2014 to December 31, 2014

with $h = 3$, $h = 10$ and $h = 30$ respectively. There are only 6 changepoints in ATS representation with $h = 30$ compare to 45 changepoints with $h = 3$. Note that the distances between changepoints has a range from 5 (time point 45 to 50) to 61 (time point 81 to 242) with $h = 10$ in Figure 3.2 (middle). Default step size is one tenth if all data is available, however, other step sizes can be utilized based on experimental needs.

If the smoothed data still seems too noisy, we may apply ATS recursively instead of increasing step size. That is, we smooth the changepoints obtained from smoothing the original data in (3.1). Figure 3.3 shows a recursive smoothing example of time series data. The data with 100 points is smoothed by ATS with $h = 4$ (red) and then 26 changepoints obtained from the first smoothing is smoothed again by ATS with $h = 4$ (blue).

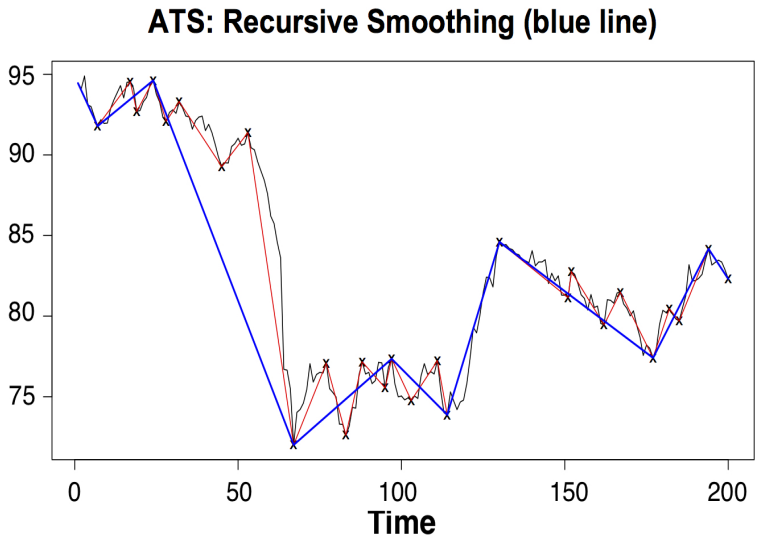

Figure 3.3: Time series with length 100 is recursively smoothed by ATS with $h = 5$.

The Limitations and Extensions of ATS

Although ATS detects changepoints based on individual points, in the first regime the aggregate behavior of data is more influential because there is no previous trend for trend comparison given that ATS identifies changepoints based on changing trends. This may result in the first simply being determined by connecting the first observation x_1 and the h-step ahead observation x_{1+h} (Figure 3.4). This issue can be resolved with a simple modification to ATS, that is, applying some other criterion to the first regime. For example, least-squared linear fit through x_1 - because of the continuity constraints - might be a possible approach.

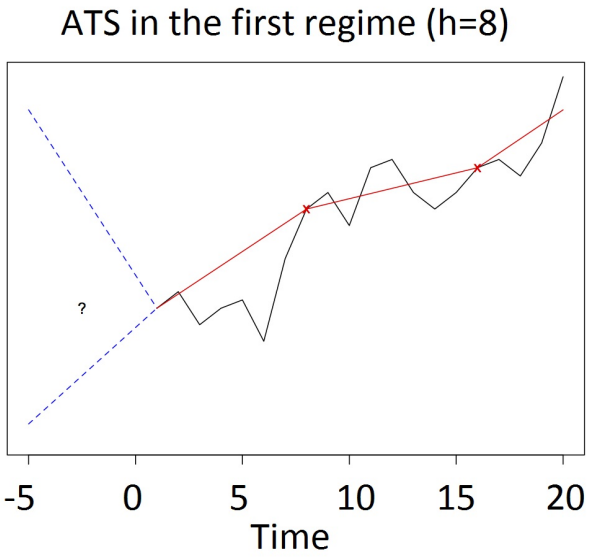

Figure 3.4: An issue in the first regime - ATS

Another possible issue is that ATS sometimes overshoots the peaks and valleys because the identification of a changepoint is delayed until the trend change is confirmed by a true trend as seen in Figure 3.2 (middle) and (bottom). This can be modified by post-processing after ATS. One simple method is to identify changepoints based on the deviations from the trend line and segment the regime into several sub-regimes within any regime. Or, the algorithm might be modified to detect changepoints considering not only the direction of

the trend change but the magnitude of the trends. This would require some additional parameters that specify the "significant" quantity in change of the magnitude of trends. However, then, the smoothed patterns of trends may no longer alternate because the algorithm identifies changepoints where the magnitude of trends change significantly while the signs may not. *Piecewise Band Smoothing* (PBS) in the next section, incorporates these ideas for modification to ATS.

3.2 Piecewise Band Smoothing (PBS)

Although ATS is simple and efficient for data reduction of online data, there are some limitations as discussed in Section 3.1. The patterns obtained by ATS is always alternating because it detects the changepoints only when the direction of trends changes. *Piecewise band smoothing* (PBS) considers more various quantities of trends to identify changepoints. While the represented data by ATS has only alternating trends, from increasing to decreasing or from decreasing to increasing, the smoothed data by PBS may have some partial sequence of the same directional patterns. In this section, we introduce identification of changepoints by PBS and its parameters, and discuss parameter selection based on goodness of fit of piecewise functions.

3.2.1 Detecting Trend Changepoints

Initial Window Size for Linear Regression (w)

The general approximation method of time series data proposed by Shatkay et al. (1996) [52] is very efficient for data reduction, however, it still requires domain expertise to specify criteria for identification of breakpoints and feature-preserving representation, and has been rarely discussed in finance. *Picewise band smoothing* (PBS) is an online processing method for piecewise linear approximation that detects changepoints where the trends change, not only the directions but "significant" magnitude. It assumes that the entire set of time series data consists of multiple data generating processes, trend regimes. Each trend regime has

a linear underlying model, such that

$$x_{it} = \beta_{i0} + \beta_{i1}t + \epsilon_{it} \tag{3.3}$$

for i-th trend regime, where ϵ_{it}'s are independent with zero mean and a finite variance $Var(\epsilon_i) = \sigma_i^2$.

Initially, some number of observations, $x_1, x_2, \cdots, x_w$, are required to fit a linear line,

$$\hat{x_t} = \hat{\beta}_0 + \hat{\beta}_1 t \tag{3.4}$$

to determine the trend ($\hat{\beta}_1$) of the current regime. Once the trend of the current regime is determined, PBS examines whether the next point x_{w+1} is under the current trend regime or not by comparing the value of $|x_{w+1} - \hat{x}_{w+1}|$ and *bandwidth* ($B > 0$), the admissible deviation range.

$$|x_{w+1} - \hat{x}_{w+1}| \leq B \quad : \text{if } x_{w+1} \text{ holds the current trend.}$$

$$|x_{w+1} - \hat{x}_{w+1}| > B \quad : \text{if } x_{w+1} \text{ does not hold the current trend.}$$

Bandwith (B) is a predefined parameter, a tolerable range of data fluctuation. That is, if x_{w+1} does not fall in the range between $\hat{x}_{w+1} - B$ and $\hat{x}_{w+1} + B$, we consider that the trend change has occurred between time w and $w + 1$, identifying x_w as a *changepoint*. If x_{w+1} falls in the range between $\hat{x}_{w+1} - B$ and $\hat{x}_{w+1} + B$, x_{w+1} is considered to hold the current trend and PBS examines the next points $x_{w+i},\ i = 2, 3, \cdots$ until it identifies a changepoint that deviates more than B from its fitted value by (3.4).

Initial window size, $w > 3$, is a parameter to specify the number of observations used for linear fit. Smaller w tends to find more changepoints than ATS does. However, unlike ATS, the length of any trend regime is always greater or equal to w because once the current trend is determined using w observations, PBS does not look back at these w points to identify changepoints.

Choice of Bandwidth: Adaptive vs. Constant (B)

Since PBS identifies changepoints by comparing the deviation of the observation from its fitted value of the current trends line to the bandwidth B, a wide bandwidth tends to identify fewer changepoints than a narrow bandwidth. The data representation results by PBS with bandwidth 6 and 2 are shown in Figure 3.5. PBS detects 6 and 14 changepoints when bandwidth are 2 and 14 in USD respectively.

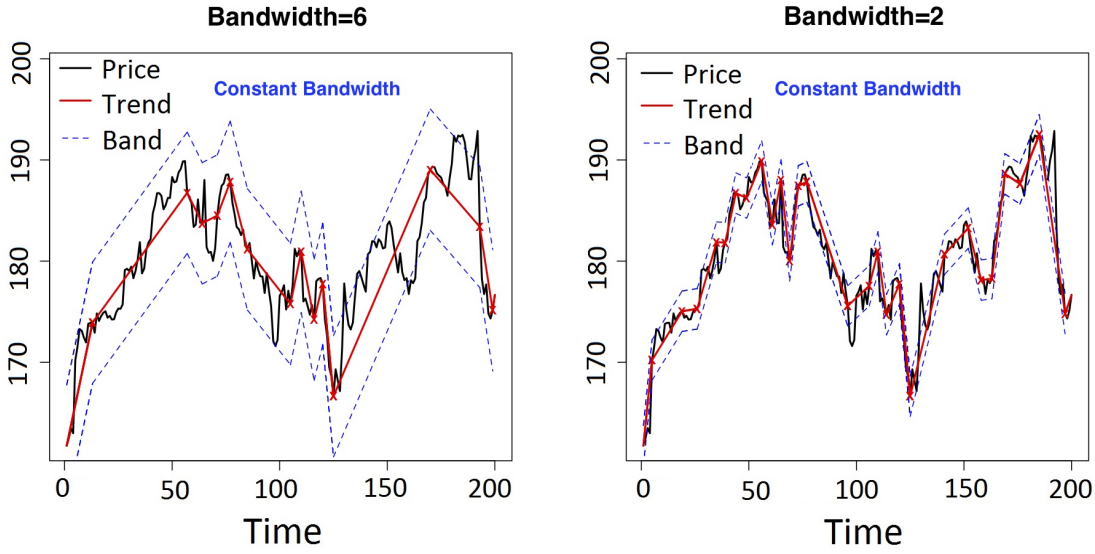

Figure 3.5: Data representation by PBS with bandwidth 6 (left) and 2 (right) - IBM daily closing price from January 13, 2012 to October 26, 2012

In financial data time series data, we often observe the variation of data is not constant but rather it varies over time. Especially, very large sized datasets usually do not maintain a constant range of data variation over time. So, it is difficult to choose one single "good" bandwidth to identify trend regimes for long time periods. Thus, PBS allows using an *adaptive bandwidth*.

Definition 3.1. (Adaptive Bandwidth) Let $L = \{l_1, l_2, \cdots, l_m\}$ be a set of the fitted piecewise linear lines such that

$$l_i : \hat{x}_{it} = \alpha_i + \beta_i t, \quad \mathbf{e_i} = \{\epsilon_{i1}, \epsilon_{i2}, \cdots, \epsilon_{in_i}\}$$

where $\hat{x}_{it}$ and $\mathbf{e_i}$ denote the fitted value at t and the residuals in the i-th regime respectively. n_i is the number of observations in the i-th regime ($n_i > 3$). Then the adaptive bandwidth of $(i+1)$-th regime $B_a^{(i+1)}$ is

$$B_a^{(i+1)} = \sum_{j=1}^{n_i} \frac{\epsilon_j^2}{(n_i - 2)} \tag{3.5}$$

The *Adaptive bandwidth* uses the information from the most recent trend regime, that is, the standard deviation of the residuals from the linear model in the previous regime. In this frame, it is assumed that the variation pattern of the observations in the most recent regime tends to continue in the current regime. Hence, the bandwidth keeps changing over time as the data evolves, and thereby the *adaptive bandwidth* works more flexibly to identify up/down trends for lengthy and noisy datasets.

If the variations of observations are relatively small and consistent over the entire time, using a *constant bandwidth* might yield a good approximation by PBS. However, in practice, especially non-stationaty financial time series data, it is unlikely that datasets are generated within constant variation rather the variation of data tends to evolve over time [58]. Figure 3.6 illustrates the represented data by PBS with *constant bandwidth* (left) and *adaptive bandwidth* (right). The *constant bandwidth* identifies 4 changepoints between time 16 and 47 although the data points between 16 and 47 seem to be under the same trend regime with a high bit of variation. This is because the *constant bandwidth* does not recognize that the variation of observation has changed in that time interval. Thus, using *adaptive bandwidth* in PBS approximation would be more appropriate for datasets with various variations of observations over time. The procedure for PBS with the parameters initial window size (w) and the bandwidth(B) is given in **Algorithm** 3.2.

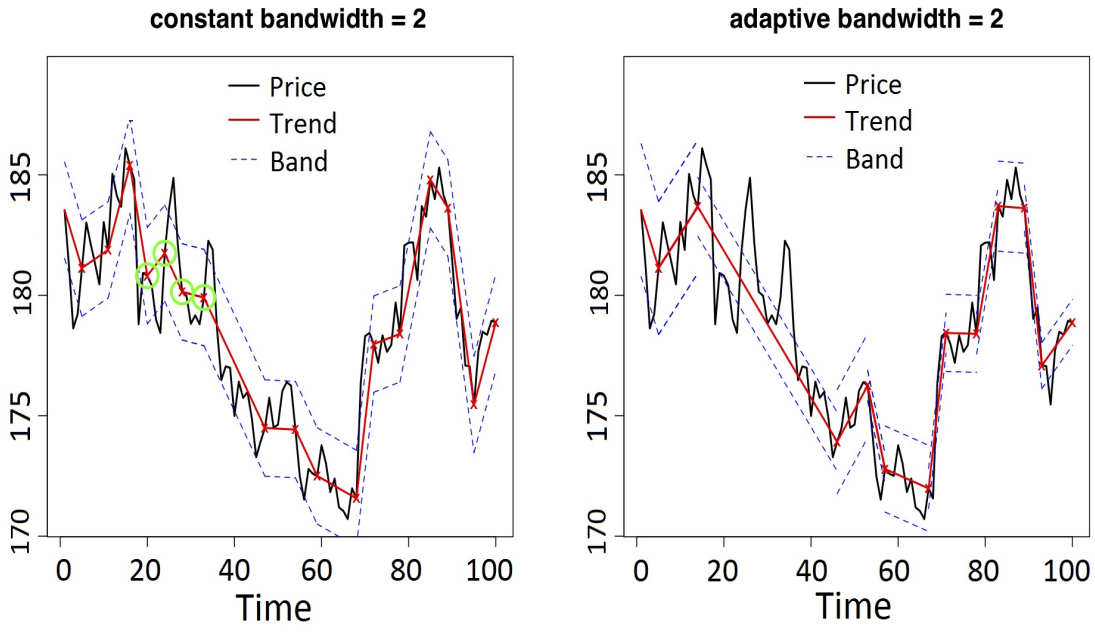

Figure 3.6: Data representation by PBS with constant bandwidth (left) and adaptive bandwidth (right) - IBM daily closing price from March 25, 2014 to August 14, 2014

Sometimes, *adaptive bandwidth* has similar issues as the constant bandwidth does when the variation of data changes dramatically. For example, if the variation of observations in the current regime is very small compared to that in the previous regime, PBS with *adaptive bandwidth* possibly fails to identify the changepoint in the end of the current regime because the bandwidth is too wide. To resolve these remaining issues, we introduce parameters *change ratio* (R), and *angle restriction (A)*.

Algorithm 3.2 Piecewise Band Smoothing with B and w

1: Set $i \leftarrow 1$ (index of data points)
2: Set $d \leftarrow 1$ (changepoint counter, the first data point at $t = 1$)

3: Specify bandwidth option (*band.wd*), either "adaptive" or "constant".
4: Specify bandwidth quantity $B > 0$
5: Specify initial window size $w > 0$

6: **while** more data **do**
7: Set $j \leftarrow 1$
8: Fit the linear trend line with w initial data points $\{x_i, x_{i+1}, \cdots, x_{i+w-1}\}$
9: **while** $|x_{i+w+j-1} - \hat{x}_{i+w+j-1}| < B$, where $\hat{x}_{i+w+j-1}$ is the fitted value of the trend **do**
10: Set $j \leftarrow j + 1$
11: **end while**

12: Save $i + w + j - 2$ as an index of changepoint and $d \leftarrow d + 1$
13: Set $i \leftarrow i + w + j - 2$, a new starting point of new trend regime
14: Go to 7
15: **end while**

Trend Change Ratio (R)

The *change ratio* (R) is a constraint on the magnitude of change when two adjacent regimes have the same direction in trend. As mentioned, it is possible that PBS miss the changepoints when the variations of data in two contiguous trend regimes are very different. For example, if the standard deviation of the residuals in the linear model of the i-th regime σ_i is much larger than that of the linear model of the next $(i+1)$-th regime $\sigma_i \gg \sigma_{i+1}$, PBS might miss some important changepoints because of too wide a bandwidth. For the other way around, $\sigma_i \ll \sigma_{i+1}$, PBS possibly identifies wrong points as changepoints. The *change ratio* (R) specifies the ratio of the magnitudes of two contiguous trends that is considered a "meaningful" change.

Definition 3.2. (Change Ratio) Let $A = \{\alpha_1, \alpha_2, \cdots, \alpha_m\}$ be a set of the trend (slope values) by piecewise banding smoothing from 3.1. When two contiguous trends $sign(\alpha_i) = sign(\alpha_{i+1})$, then the change ratio $R > 1$ is

$$\begin{aligned} R &= \min\left(\frac{\alpha_{i+1}}{\alpha_i}\right) \quad \text{if} \quad \left(\frac{\alpha_{i+1}}{\alpha_i}\right) > 1 \\ &= \min\left(\frac{\alpha_i}{\alpha_{i+1}}\right) \quad \text{if} \quad \left(\frac{\alpha_{i+1}}{\alpha_i}\right) < 1 \end{aligned} \tag{3.6}$$

equivalently,

$$\frac{1}{R} = \max\left(\frac{\alpha_{i+1}}{\alpha_i}\right) \quad \text{if} \quad \left(\frac{\alpha_{i+1}}{\alpha_i}\right) < 1$$

A larger value of the change ratio R tends to identify fewer number of changepoints. Figure 3.7 shows the results of data representation by PBS with the change ratio $R = 2$ (left) and $R = 5$ (right). PBS approximation with smaller change ratio $R = 2$ results in identifying 32 changepoints while PBS with $R = 5$ identifies 27 changepoints. These 5 changepoints (circled) additionally identified by PBS with $R = 2$ does not seem to be turning points where the linear trends are changing, but rather the trends before and after

these points seem to be the same or very similar. These issues can be relieved considerably by change ratio constraint.

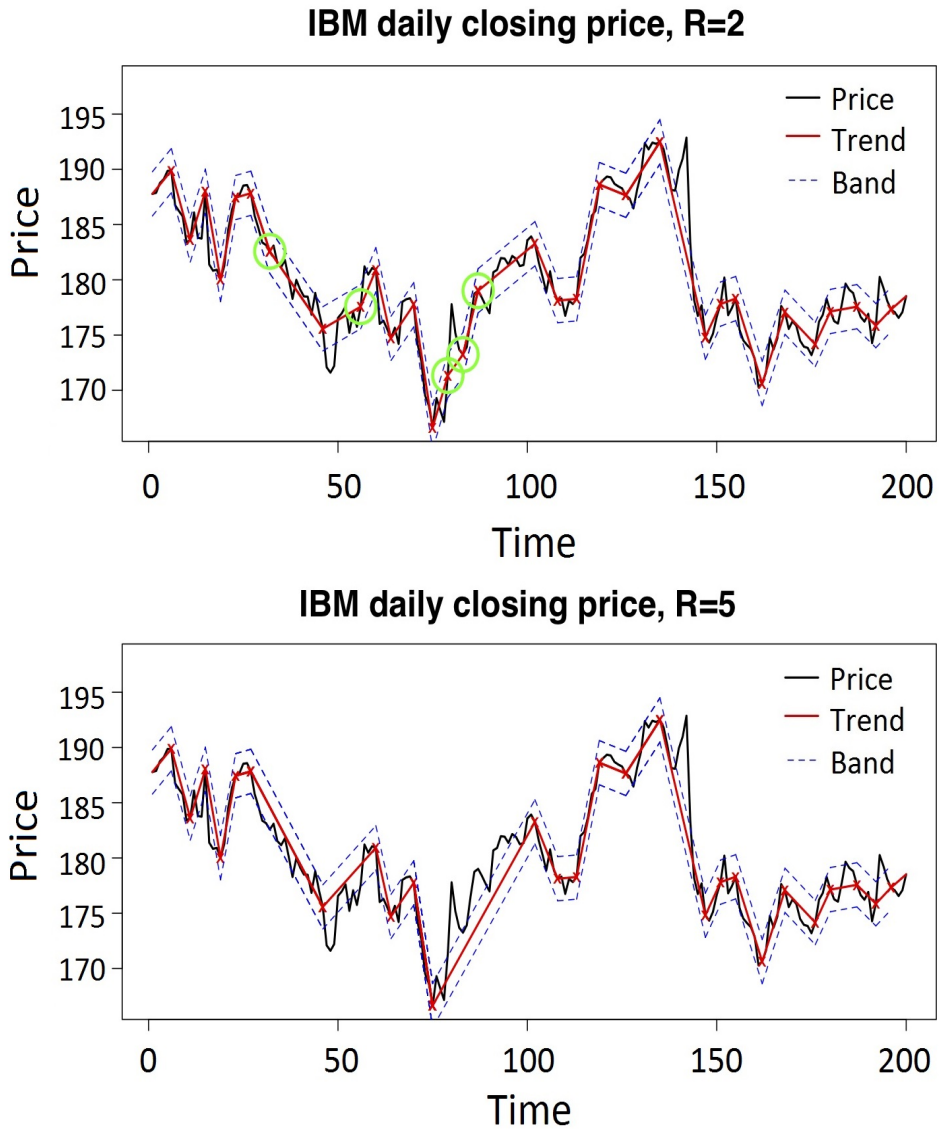

Figure 3.7: Data representation by PBS with the change ratio $R = 2$ (left) and $R = 5$ (right) - IBM daily closing price from March 27, 2012 to January 11, 2013

Angle Restriction for Small Magnitude Trends (A)

When the magnitude of the trends are very small (close to zero) or very large, even large value of change ratio does not differentiate trend change clearly as seen in Figure 3.8. This issue leads to the requirement of angle restriction since the angle between two trends varies for the same change ratio. The *angle restriction* A is the maximum angle θ^* $(0 < \theta^* < \pi)$ between two trend lines. Some degree of "visual" significance in trend change is guaranteed by angle restriction.

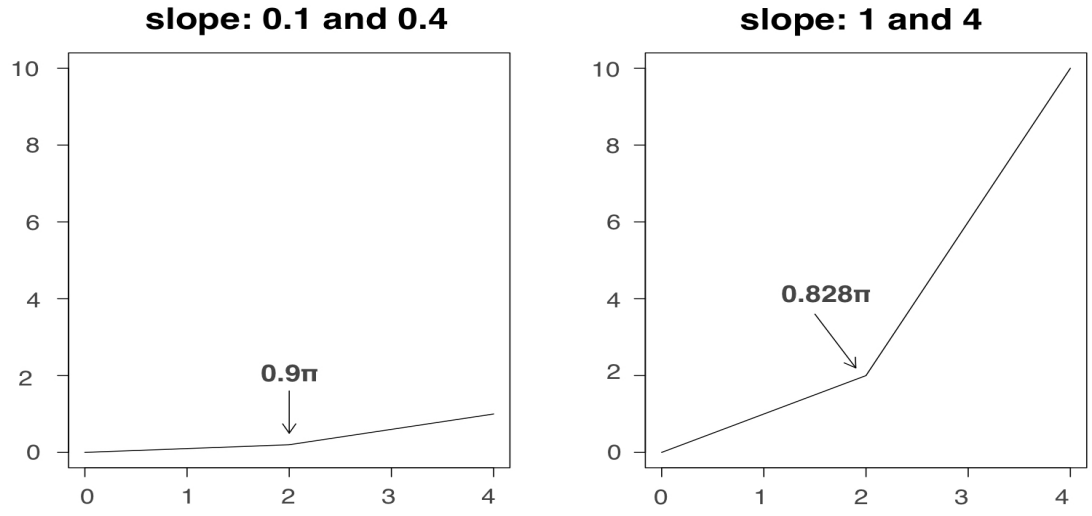

Figure 3.8: For the same $R = 4$, the angles are different.

Definition 3.3. (Angle Restriction) Let θ_i be the angle between i-th trend line l_i and time axis such that $\alpha_i = \tan\theta_i$ where α_i denotes the slope value of l_i. Then the angle between two contiguous lines l_i and l_{i+1} is

$$\gamma_i = \left|\tan^{-1}(\alpha_{i+1}) - \tan^{-1}(\alpha_i)\right|, \quad 0 < \gamma_i < \pi$$

and the angle restriction A is defined by

$$A = \max(\pi - \gamma_i) \tag{3.7}$$

The *angle restriction* is particularly useful when the signs of trends change but the magnitudes of the trends are very small. The issue that PBS may fail to identify the important changepoints or misidentify changepoints can be resolved by employing angle restriction. The PBS algorithm including all parameters, initial window size w, bandwidth B, change ratio R, and angle restriction A is given in **Algorithm** 3.3

Algorithm 3.3 Piecewise Band Smoothing with B, w, R and A

1: Set $i \leftarrow 1$ (index of data points)
2: Set $d \leftarrow 1$ (changepoint counter, the first data point at $t = 1$)
3: Specify bandwidth option (*band.wd*), either "adaptive" or "constant".
4: Specify $B > 0, w > 0, R > 1$ and A
5: where B, w, R and A is bandwidth, initial window size, change ratio and angle restriction respectively.

6: Fit the first linear trend line l_1 with $\{x_1, x_2, \cdots, x_w\}$
7: **if** bandwidth option = adaptive bandwidth **then**
8: $B^{(1)} \leftarrow$ residual standard deviation of l_1
9: **end if**
10: **if** bandwidth option = constant bandwidth **then**
11: $B^{(d)} \leftarrow c$ for all $d \in \mathbb{N}$, where c is specified in line 4.
12: **end if**
13: **while** more data **do**
14: Set $j \leftarrow 1$
15: **while** $|x_{w+j} - \hat{x}_{w+j}| < B^{(1)}$ **do**
16: where $\hat{x}_{w+j}$ is the fitted value of the line l_1 at $t = w + j$
17: Set $j \leftarrow j + 1$
18: **end while**
19: Save $(w + j - 1)$ as the first changepoint; $d \leftarrow d + 1$
20: Set $i \leftarrow (w + j - 1)$ a starting point of new trend regime
21: Set $j \leftarrow 1$
22: Fit the linear trend l_d with $\{x_i, x_{i+1}, \cdots, x_{i+w-1}\}$
23: **while** $|x_{i+w+j-1} - \hat{x}_{i+w+j-1}| < B^{(d-1)}$ **do**
24: where $\hat{x}_{i+w+j-1}$ is the fitted value of the line l_d at $t = i + w + j - 1$
25: Set $j \leftarrow j + 1$
26: **end while**

27: **if** $d < 3$ **then**
28: go back to 14
29: **end if**
30: **if** $d \leq 0$ **then**
31: **if** sign(slope of d-th regime) = sign(slope of $(d-1)$-th regime) **then**
32: **if** $\left(\frac{s_{d+1}}{s_d} > R \text{ or } \frac{s_{d+1}}{s_d} < \frac{1}{R}\right)$ and $\theta_d < A$ **then**
33: where s_d = the slope of d-th regime and θ_d = angle between s_d and s_{d+1}
34: Confirm and save $(i + w + j - 2)$ as a $(d-1)$-th index of changepoints
35: Otherwise, delete $(i + w + j - 2)$ from the changepoint sequence
36: **end if**
37: **end if**
38: **if** sign(slope of d-th regime) != sign(slope of $(d-1)$-th regime) **then**
39: **if** $\theta_d < A$ **then**
40: confirm and save $(i + w + j - 2)$ as a $(d-1)$-th index of changepoints
41: Otherwise, delete $(i + w + j - 2)$ from the sequence of changepoints.
42: **end if**
43: **end if**
44: **end if**
45: **end while**

Continuity Constraint - Linear Regression without Intercept

The fitted lines by least-squared method may or may not have the intercept. Although generally the fitted line with intercept is recommended when behavior of data near the origin is uncertain, in PBS fitting the line without intercept provides more sensible because otherwise every changepoint has two fitted values. Also, by fitting the line without intercept (*Regression Through Origin, RTO*), the continuity constraint of smoothing can be achieved. Note that in RTO fitting the mean of residuals is non-zero because the regression line is generally inconsistent with the best fit [15]. However, with the finite number of data points within the regime, in practice the use of RTO is more beneficial for interpretablity in the context with adjacent regimes at changepoints (Figure 3.9).

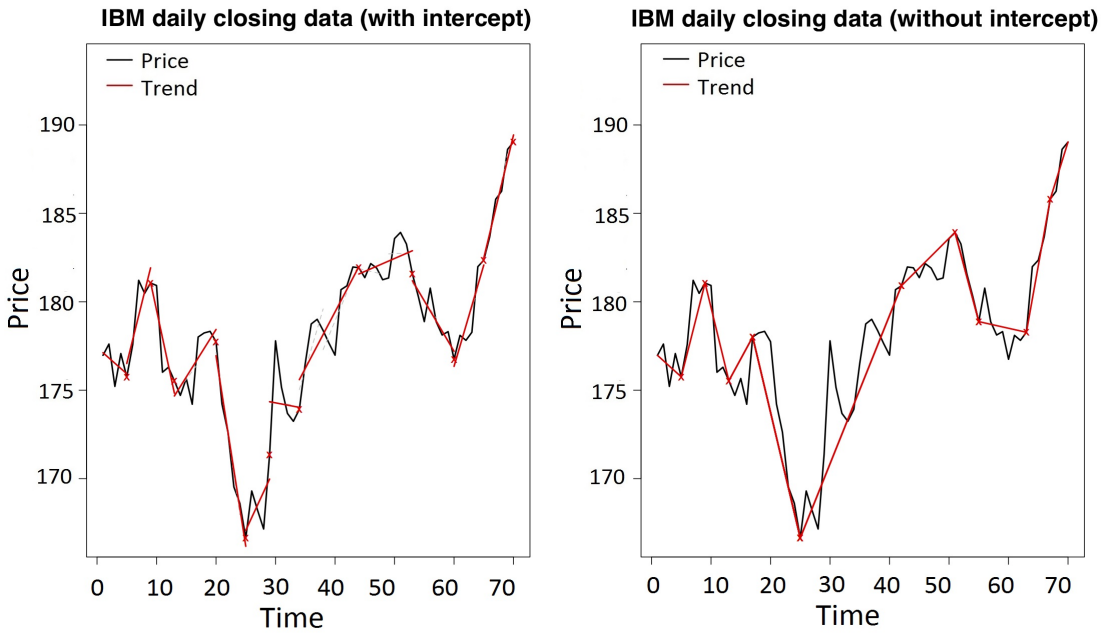

Figure 3.9: Piecewise linear lines with and without intercept

3.2.2 Criterion of Data Representation

In piecewise band smoothing for financial time series data, our objective is not only to identify trend changepoints but also to reduce the original data to a manageable size. Selecting parameters that yields the smallest squared sum of residuals (SSR) might be considered a decision criteria in terms of goodness-of-fit, but this criteria will always choose the smallest window size and bandwidth, and as a result there are too many trend regimes, or always more than a certain number of regimes in the reduced data. Therefore the regularization on the number of trend regime is necessary to reconcile our objectives, preserve the information of the original data, and reduce dimensionality efficiently. The criterion of parameter selection can be given as follows: let $X = \{x_{ik} \mid x_{11}, x_{12}, \cdots, x_{1n_1}, \cdots, x_{i1}, \cdots, x_{in_i}, \cdots\}$ be a set of original time series data indexed by PBS such that x_{ik} denotes the k-th point in i-th trend regime.

$$\begin{aligned} &\min \sum_{i=1}^{\infty} (\text{SSR per trend in } i\text{-th regime}) + \lambda \text{ (the number of changepoints)} \\ &= \min \sum_{i=1}^{\infty} \sum_{k=1}^{n_i} \left\{ \frac{(x_{ik} - \hat{x}_{ik})^2}{n_i} + \lambda I(x_{ik} = x_{i1}) \right\} \end{aligned} \tag{3.8}$$

where $\hat{x}_{ik}$ is the fitted value of k-th point, n_i is the number of data points in i-th regime and λ is a tuning parameter of regularization term. The first term of (3.8) implies the loss function of the data while the second term penalizes for the number of changepoints to approximate smoothing. Figure 3.10 and Figure 3.11 show the changes of SSR and the number of the changepoints as the initial window size w and the bandwidth B increase (IBM daily closing price from October 15, 2012 to October 15, 2015). X axis in Figure 3.10 implies that k-fold adaptive bandwidth. In Figure 3.10, the SSR tends to increase and the number of the changepoints decrease as the bandwidth broadens although both tendencies are not quite monotonic. Similarly, the SSR tends to increase as the initial window size w increases while the number of the changepoints decreases. Note that the number of the

trend regimes, or changepoints, decrease as the initial window size w increases.

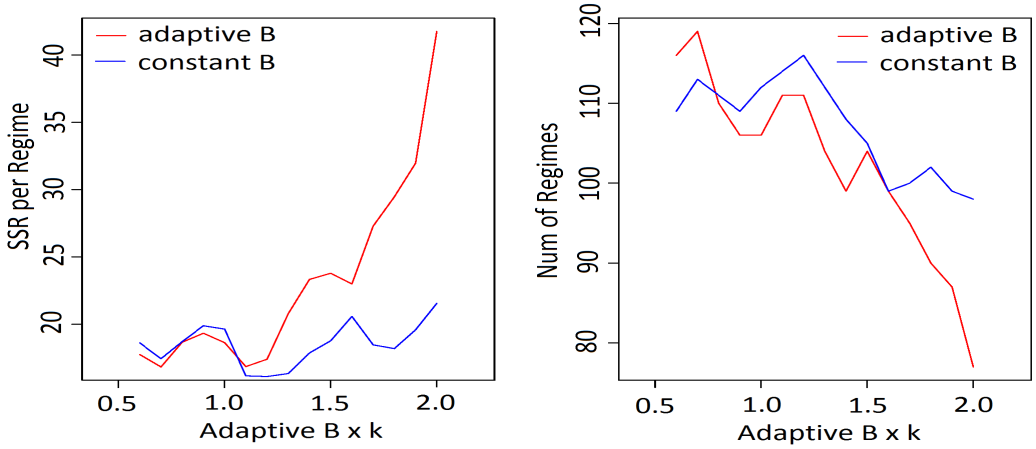

Figure 3.10: SSR per regime and the number of regimes for various B

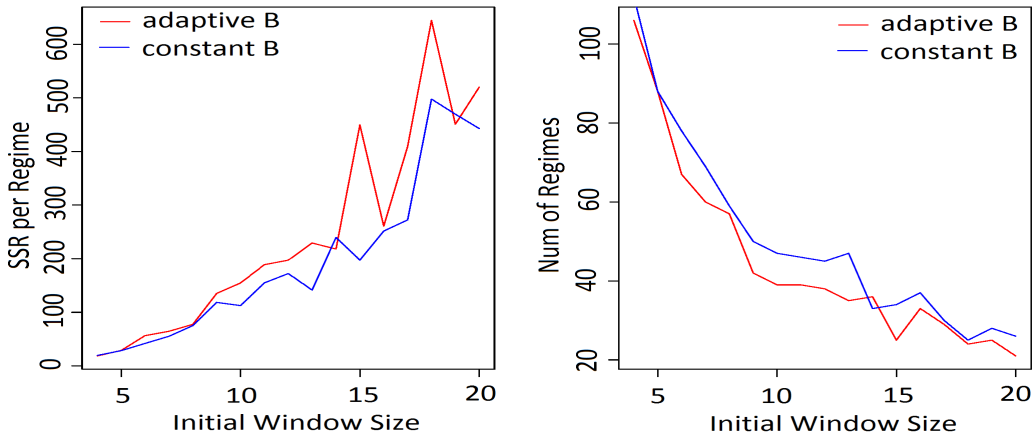

Figure 3.11: SSR per regime and the number of regimes for various w

Table 3.1: SSR per regime and the number of changepoints

	SSR/regime								**No. changepoints**							
Band	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
$w = 4$	17.75	18.66	18.63	17.40	23.33	22.99	29.47	41.76	116	110	106	111	99	99	90	77
$w = 5$	24.84	25.64	28.92	37.62	45.27	49.78	57.18	77.63	98	92	88	79	73	69	69	66
$w = 6$	43.52	46.17	56.52	55.98	69.59	78.85	100.29	93.48	76	72	67	71	67	62	59	63
$w = 7$	75.61	59.90	64.63	70.80	86.48	117.95	111.18	95.81	53	66	60	59	52	48	46	48
$w = 8$	94.76	89.78	77.48	104.10	87.37	160.30	144.44	155.08	51	54	57	53	55	40	43	42
$w = 9$	122.46	134.31	135.25	128.72	127.95	146.55	189.59	186.38	53	46	42	45	49	46	40	43
$w = 10$	112.11	150.16	154.73	182.94	142.32	174.54	237.79	273.34	50	47	39	41	43	41	37	35
$w = 11$	150.69	180.93	188.90	195.23	191.96	198.72	265.95	238.92	42	43	39	39	36	38	32	30

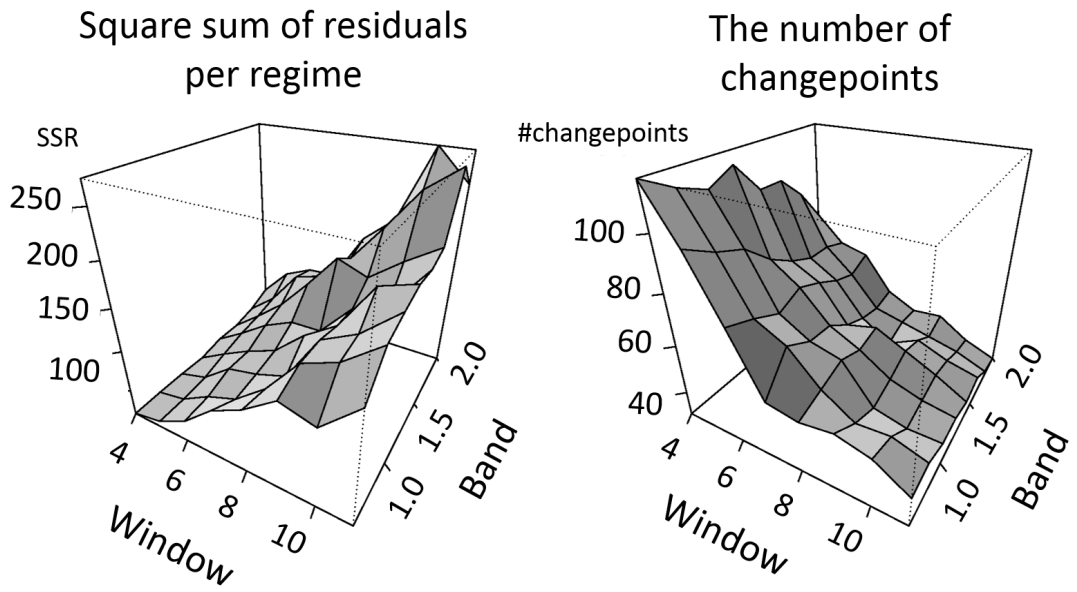

Figure 3.12: SSR per regime and the number of regimes for various B and w

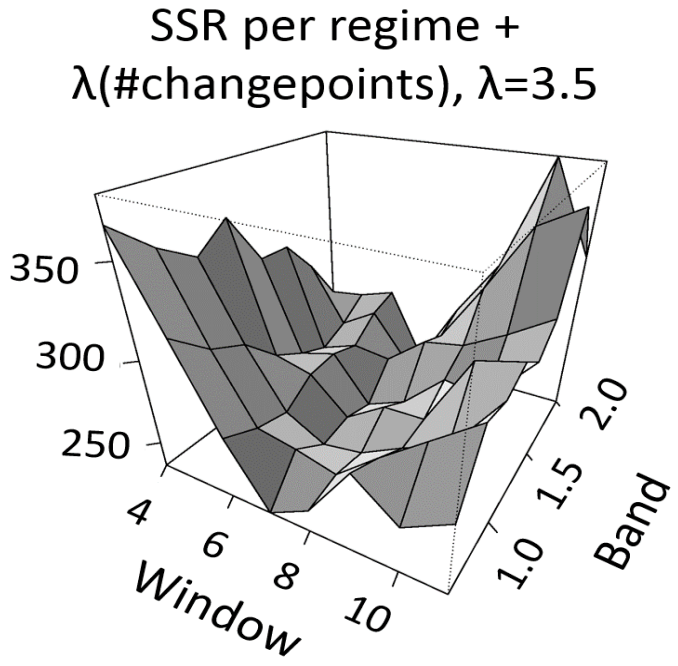

Figure 3.13: Criteria function combining the loss function and regularization ($\lambda = 3.5$)

In financial time series data, the range of SSR of individual assets varies. For example, from January 13, 2012 to January 12, 2015, the range of IBM daily closing prices is 53.81 USD while that of Microsoft Corporation (NASDAQ:MSFT) is 23.57 USD. Thereby, the range of the individual asset prices should be considered to choose an effective tuning parameter λ in (3.8). Standardization of the data can be one approach. Or we can use the equivalent form to the (3.8) as follows:

$$\min \sum_{i=1}^{\infty} \sum_{k=1}^{n_i} \frac{(x_{ik} - \hat{x}_{ik})^2}{n_i} \quad \text{subject to} \quad d_1 \leq \sum_{i}^{\infty} \sum_{k=1}^{n_i} I(x_{ik} = x_{i1}) \leq d_2 \tag{3.9}$$

The price variation of assets also influences the choice of the angle restriction. The smaller variation of data is more robust to the angle restriction. Thus, a larger angle is recommended for the data with small variation. Table 3.2 shows the result of parameter selection with $d_1 = 60$ and $d_2 = 70$ based on the criteria shown in (3.8) for stock prices of nine companies and Dow Industrial Average data from January 13, 2012 to January 12, 2015. Stock prices of Pfizer Inc. (NYSE:PFE) shows the smallest SSR per regime after data representation, and it requires the largest angle parameter ($A = \frac{44}{45}\pi$) for PBS smoothing since the standard deviation of Pfizer stock price is the smallest. On the other hand, a smallest angle ($\frac{9}{10}\pi$) is enough to smooth the stock price of Goldman Sachs Group, Inc. (NYSE:GS) because its standard deviation is the largest among nine asset prices except Dow Jones Industrial Average (DJI: DJIA).

Table 3.2: Parameter selection based on criteria given in (3.8)

	SSR/regime	**No. changepoints**	(w, B, A)	**std**(x)
IBM	49.78	69	$(5, 1.6, \frac{11}{12}\pi)$	9.73
ORCL	2.50	69	$(6, 0.8, \frac{23}{24}\pi)$	4.79
MSFT	2.50	70	$(5, 0.8, \frac{19}{20}\pi)$	6.70
PFE	0.84	68	$(6, 0.6, \frac{44}{45}\pi)$	3.67
GSK	2.44	68	$(6, 0.8, \frac{29}{30}\pi)$	4.41
JNJ	5.84	68	$(4, 1.0, \frac{11}{12}\pi)$	15.17
VISA	6.45	70	$(4, 1.6, \frac{11}{12}\pi)$	10.70
JPM	5.54	67	$(5, 0.6, \frac{11}{12}\pi)$	8.83
GS	48.33	69	$(6, 0.6, \frac{8}{9}\pi)$	28.93
DJIA	261033	63	$(6, 1.4, \frac{9}{10}\pi)$	1654239

Table 3.3: Stock symbols

Symbol	**Company**
IBM	International Business Machine (NYSE)
ORCL	Oracle Corporation (NYSE)
MSFT	Microsoft Corporation (NASDAQ)
PFE	Pfizer Inc. (NYSE)
GSK	GlaxoSmithKline (NYSE)
JNJ	Johnson & Johnson (NYSE)
VISA	Visa Inc. (NYSE)
JPM	JPMorgan Chase (NYSE)
GS	Goldman Sachs (NYSE)
DJIA	Dow Jones Industrial Average

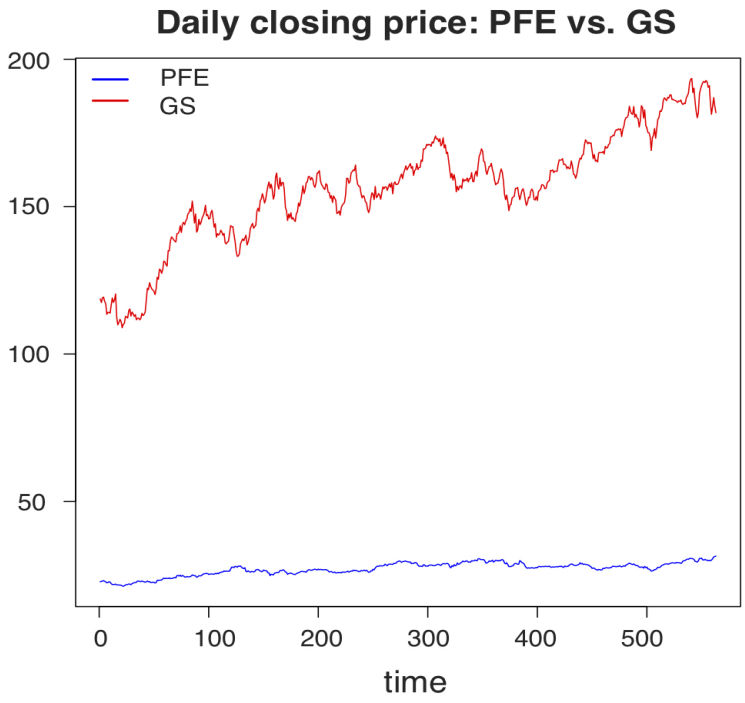

Figure 3.14: Daily stock closing prices: PFE vs. GS

3.3 Extend to Higher Degree Polynomials

Instead of a linear line, higher degree polynomial functions such as quadratic and cubic polynomials may be considered for piecewise functions (Figure 3.15). This approximation does not require two parameters, change ratio R and angle restriction A. Of course as the degree of polynomials increases the SSR per regime decreases and thus representation is closer to data with reasonable number of trend regimes. Nevertheless, it is not feasible to identify patterns of the regimes while applying continuity restrictions. Moreover, it might be less meaningful in the viewpoint of financial data trend analysis because two or more directions can exist within a regime.

Higher degree polynomials can be modified by flexible choice of piecewise functions. For example, given initial w points in the beginning of the regime, PBS fits both quadratic and linear models. Then it compares F-statistics or p-value of two models and chooses the more

significant one but in case that the quadratic fit is more significant it also must be monotonic, otherwise it chooses the linear fit. Figure 3.16 shows represented data of Dow Jones Industrial Average (INDEXDJX:DJIA) from January 13, 2012 to January 11, 2013 by this "flexible" algorithm. Not surprisingly given the reasons discussed earlier, every represented regime is fitted by a linear line. This might imply that linear representation is sufficient to summarize trend sequences from financial data but further research is required.

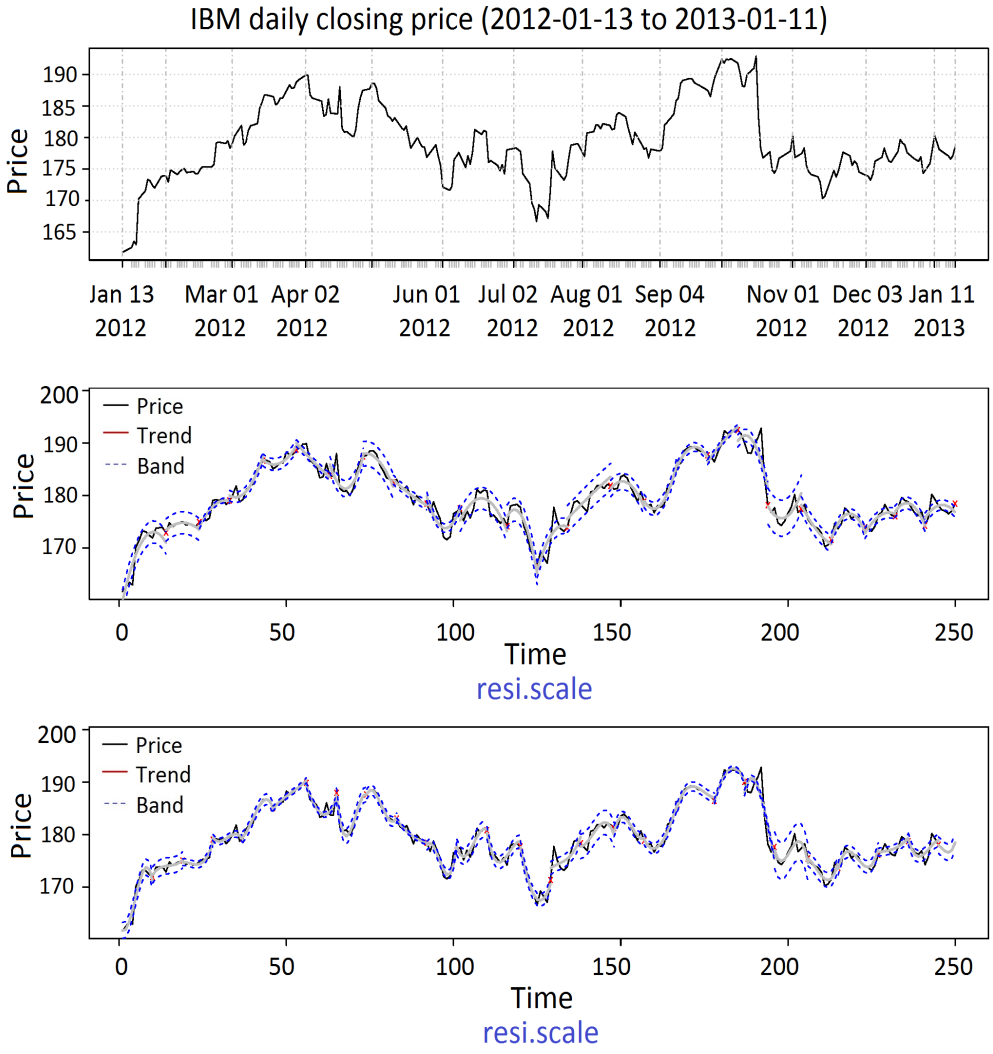

Figure 3.15: Piecewise higher degree polynomials

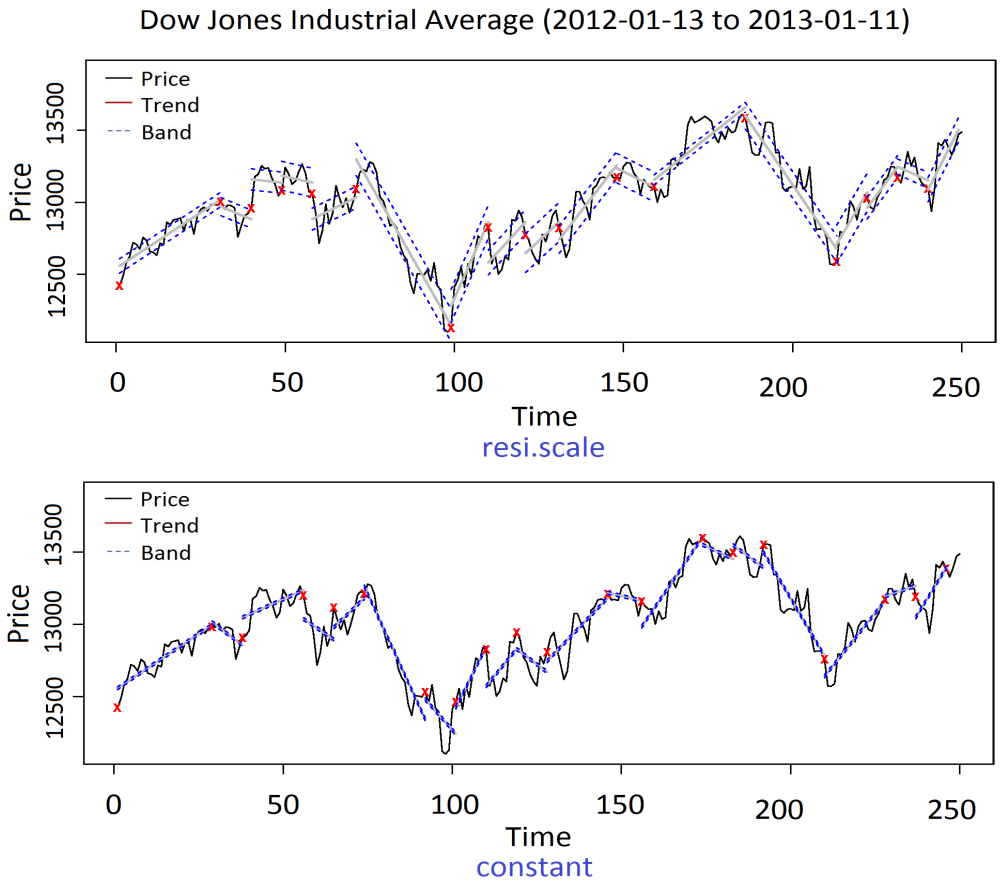

Figure 3.16: Piecewise flexible approximation

Chapter 4: Patterns of Trends and Distance Measures

Pattern recognition for large size data consists for two important steps, (1) data representation and (2) similarity/disimilarity measures. The most important reason for data representation is to reduce the dimensionality of data substantially since measuring similarity between two lengthy time series directly is usually not feasible. Another important reason would be to find patterns efficiently from the reduced data. There are some patterns used widely such as trend and seasonality. Besides, patterns can be defined based on the purpose of the analysis. For example, in technical analysis of financial market data, various patterns are discussed to look for trading opportunities (Figure 4.1).

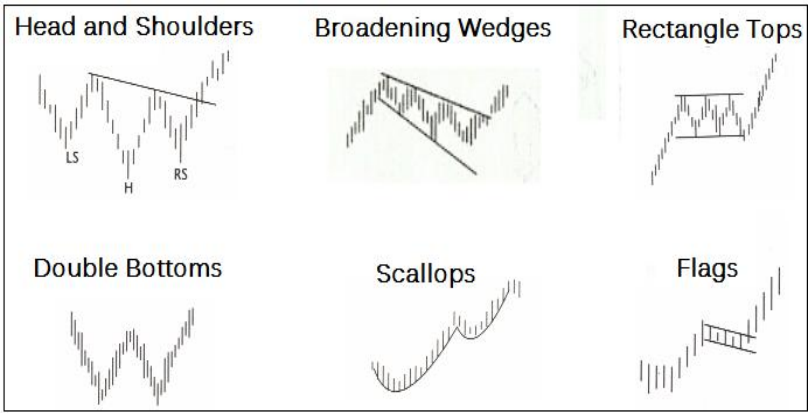

Figure 4.1: Various patterns used in stock market analysis. http://www.forexblog.org [2]

Well-defined patterns from the reduced data and similarity measures can result in good performance in data analysis, such as classification and clustering. In this chapter, we discuss defining two types of patterns, continuous and discretized, and propose some methods for similarity measures motivated by Müller (2007) [44] and Lin (2007) [43].

4.1 Sequence of Local Functions

Alternating trends smothing (ATS) or piecewise band smoothing (PBS) reduces the dimensionality of data a substantially by by fitting sequential local models.

Mathematically, the data reduction process by ATS or PBS can be addressed as follows. Let $X = \{x_{t_1}, x_{t_2}, \cdots, x_{t_N}\}$ be the original time series data and ϕ be an operation of ATS or PBS. Then ϕ maps the original data $X \in \mathcal{X}$ to the reduced space $\mathcal{X}_{re}$. That is,

$$\phi : \mathcal{X} \longrightarrow \mathcal{X}_{re} \tag{4.1}$$

where $\mathcal{X} \in \mathbb{R}^N$ $\mathcal{X}_{re} \in \mathbb{R}^n$, $(n \ll N)$. and . Each element in $X_{re} = \{x_{b_1}, x_{b_2}, \cdots, x_{b_n}\} \in \mathcal{X}_{re}$, where b_i's are the i-th changepoint, can be decomposed to X-Y cartesian coordinates, for example,

$$X_{re} = \{x_{b_1}, x_{b_2}, \cdots, x_{b_n}\} = \{(b_1, c_1), (b_2, c_2), \cdots, (b_n, c_n)\} \tag{4.2}$$

where b_k and c_k are the time index of k-th changepoints and the value at time b_k respectively. Patterns can be defined using some features of a local function $\mu_i(t)$ for i-th regime defined between b_i and b_{i+1}. Local $\mu_i(t)$ can have various forms of functions. For piecewise constant modeling,

$$\mu_i(t) = \alpha_i I[b_i, b_{i+1}](t) \tag{4.3}$$

where $I_A(t) = 1$ if $t \in A$ and $I_A(t) = 0$ otherwise. For piecewise linear approximation by ATS or PBS, $\mu_i(t)$ can be written as

$$\mu_i(t) = (\alpha_i + \beta_i(t))I[b_i, b_{i+1}](t) \tag{4.4}$$

where

$$\alpha_i = c_i, \quad \beta_i = \frac{c_{i+1} - c_i}{b_{i+1} - b_i}, \quad \text{and} \quad b_i < b_{i+1} \text{ for all } i.$$

α_i is the value at the starting point of i-th trend regime and b_i is the trend (slope).

Obviously the global model of the represented data can be written as

$$\sum_{i=1}^{n} \mu_i(t) \tag{4.5}$$

4.2 Types of Patterns and Distance Measures

Well-defined patterns that describe interesting features of data enable effective similarity measures between two time series and lead to valid classification or clustering. The features can be obtained from piecewise local functions. In piecewise constant approximation in (4.3), there are two pieces of information available in each regime: (1) the representative constant α_i and (2) the length of the regime (duration) $b_{i+1} - b_i$. Similarly, piecewise linear representations in (4.4), provide three pieces of information in each regime: (1) the constant α_i, (2) a slope (trend) β_i, and (3) the length of the regime (duration) $b_{i+1} - b_i$. Although the piecewise local function contains multiple pieces of information, it is not easy to incorporate all the information for a similarity measure. Rather it might be more efficient to use just one or two important features of the regime.

4.2.1 Numerical Patterns and Distance Measures

In financial data analysis, it is obvious that the major interests are "up/down trend" and "the length of the trends." The sequence of the trends and the length of trends can be written as the slope of the linear fit and the difference between the starting and the ending times of the regime.

Now, we address transforming a time series data with length N into the reduced sequence of trends, and measuring distance between two reduced data in detail. Let $T = \{x_{t_1}, x_{t_2}, \cdots, x_{t_N}\}$ and $Q = \{y_{t_1}, y_{t_2}, \cdots, y_{t_N}\}$ be two time series with the same length N. By piecewise linear smoothing, T and Q are mapped to $\widehat{T}$ and $\widehat{Q}$ such that,

$$\begin{aligned}\widehat{T} &= \{(s_1, d_1), (s_2, d_2), \cdots, (s_m, d_m)\}, \quad (m \ll N) \\ \widehat{Q} &= \{(v_1, l_1), (v_2, l_2), \cdots, (v_n, l_n)\}, \quad (n \ll N)\end{aligned} \tag{4.6}$$

where (s_i, d_i) and (v_j, l_j) are the trend slope and the length of the trend in the i-th and the j-th regime of T and Q respectively. The i-th length of the trend, d_i, can be written using changepoints as well, that is, $d_i := (b_{i-1}, b_i)$ where b_{i-1} is the starting point and b_i are the last point of the i-th trend regime. Then another form of (4.6) is possible as follows.

$$\begin{aligned}\widehat{T} &= \{(s_1, d_1), (s_2, d_2), \cdots, (s_m, d_m)\} \\ &= \{(s_1, (b_0, b_1)), (s_2, (b_1, b_2)), \cdots, (s_m, (b_{m-1}, b_m))\} \\ \widehat{Q} &= \{(v_1, l_1), (v_2, l_2), \cdots, (v_n, l_n)\} \\ &= \left\{(v_1, (b_0, b_1^*)), (v_2, (b_1^*, b_2)), \cdots, (v_n, (b_{n-1}^*, b_n^*))\right\}\end{aligned} \tag{4.7}$$

where $\{b_0, b_1, b_2, \cdots, b_m\}$ and $\{b_0^*, b_1^*, b_2^*, \cdots, b_n^*\}$ are sets of the identified changepoints of T and Q ($b_0 = b_0^* = t_1,\ b_m = b_n^* = t_N$).

Now, for similarity measure between $\widehat{T}$ and $\widehat{Q}$, the *short time series distance* (STS) method [44] can be extended by allowing the different lengths of reduced data. We define the step functions of the trends over time for given smoothed time series $\widehat{T}$ and $\widehat{Q}$,

$$f_{\widehat{T}}(t) = s_i I([b_{i-1},\ b_i]), \quad i = 1, 2, \cdots, m \tag{4.8}$$

$$f_{\widehat{Q}}(t) = v_j I(\left[b_{j-1}^*,\ b_j^*\right]), \quad j = 1, 2, \cdots, n \tag{4.9}$$

where $I_A(t) = 1$ if $t \in A$; otherwise $I_A(t) = 0$. The *trends step function* (TSF) distance

between T and Q is defined by

$$\begin{aligned} \mathrm{TSF}(\widehat{T},\widehat{Q}) &= \left(\sum_{i=1}^{m+n}\int_{a_{i-1}}^{a_i}\left(f_{\widehat{T}}(t)-f_{\widehat{Q}}(t)\right)^2 dt\right)^{\frac{1}{2}} \\ &= \left(\sum_{i=1}^{m+n}\left(f_{\widehat{T}}(a_i)-f_{\widehat{Q}}(a_i)\right)^2 (a_i-a_{i-1})\right)^{\frac{1}{2}} \end{aligned} \tag{4.10}$$

where $\{a_0, a_1, a_2, \cdots, a_{m+n}\}$, $(a_0 = t_1,\ a_{m+n} = t_N,\ a_i \le a_{i+1}$ for all $i)$ is the combined set of changepoints of $\widehat{T}$ and $\widehat{Q}$ in order. Figure 4.2 illustrates the smoothed time series by PBS $\widehat{T}$ and $\widehat{Q}$ (top) and the step functions transformed of $\widehat{T}$ and $\widehat{Q}$. The TSF distance between T and Q is described as the summation of square of the differences between two step functions (bottom). The TSF distance in (4.10) allows two reduced datasets to have different lengths. Also, it is a metric function discussed in Section 2.3.1 of Chapter 2.

We prove triangular inequality of TSF here since it is obvious that TSF distance satisfies properties of the metric, non-negativity, identify of indiscernibles, and symmetry.

Claim 1. *The trends step function* (TSF) distance in (4.10) satisfies triangular inequality,

$$\mathrm{TSF}(\widehat{T},\widehat{R}) \le \mathrm{TSF}(\widehat{T},\widehat{Q}) + \mathrm{TSF}(\widehat{Q},\widehat{R}) \tag{4.11}$$

where $\widehat{T},\widehat{Q}$ and $\widehat{R}$ are smoothed time series by ATS or PBS.

Proof: Let $T = \{x_{t_1}, x_{t_2}, \cdots, t_{t_N}\}$, $Q = \{y_{t_1}, y_{t_2}, \cdots, y_{t_N}\}$ and $R = \{z_{t_1}, z_{t_2}, \cdots, z_{t_N}\}$ be time series with length t_N. Suppose each time series is smoothed by ATS or PBS and we obtain the sets of changepoints $T_b = \{b_0, b_1, b_2, \cdots, b_l\}$, $Q_b = \{b_0^*, b_1^*, \cdots, b_m^*\}$, and $R_b = \{b'_0, b'_1, \cdots, b'_n\}$ for T, Q and R respectively. Let $A = \{a_0, a_1, \cdots, a_{l+m+n}\}$ denote a combined set of T_b, Q_b and R_b in order, where $a_0 = b_0 = b_0^* = b'_0 = t_1$ and $a_{l+m+n} = b_l = b_m^* = b'_n = t_N$. Then the sum of distances between $\widehat{T}$ and $\widehat{Q}$, and $\widehat{Q}$ and $\widehat{R}$ satisfies

$$
\begin{aligned}
&\mathrm{TSF}(\widehat{T},\widehat{Q}) + \mathrm{TSF}(\widehat{Q},\widehat{R}) \\
&= \left(\sum_{i=1}^{l+m+n} \left(f_{\widehat{T}}(t) - f_{\widehat{Q}}(t) \right)^2 (a_i - a_{i-1}) \right)^{\frac{1}{2}} + \left(\sum_{i=1}^{l+m+n} \left(f_{\widehat{T}}(t) - f_{\widehat{Q}}(t) \right)^2 (a_i - a_{i-1}) \right)^{\frac{1}{2}} \\
&= \left(\sum_{i=1}^{l+m+n} \left(\left(f_{\widehat{T}}(a_i) - f_{\widehat{Q}}(a_i) \right) \sqrt{a_i - a_{i-1}} \right)^2 \right)^{\frac{1}{2}} + \left(\sum_{i=1}^{l+m+n} \left(\left(f_{\widehat{Q}}(a_i) - f_{\widehat{R}}(a_i) \right) \sqrt{a_i - a_{i-1}} \right)^2 \right)^{\frac{1}{2}} \\
&\geq \left(\sum_{i=1}^{l+m+n} \left(\sqrt{a_i - a_{i-1}} \left(f_{\widehat{T}}(t) - f_{\widehat{Q}}(t) + f_{\widehat{Q}}(t) - f_{\widehat{R}}(t) \right) \right)^2 \right)^{\frac{1}{2}} \text{ by Minkoski inequality} \\
&= \left(\sum_{i=1}^{l+m+n} \left(f_{\widehat{T}}(t) - f_{\widehat{R}}(t) \right)^2 (a_i - a_{i-1}) \right)^{\frac{1}{2}} = \mathrm{TSF}(\widehat{T},\widehat{R})
\end{aligned}
\tag{4.12}
$$

■

Another possible approach to measure distance for numerical patterns is to compare the change amounts in values and length of the trend regimes for all pairs. To use this similarity measure, two reduced datasets must be the same length. In (4.6), s_i and v_j can be substituted for $\Delta_i^T = x_{b_i} - x_{b_{i-1}}$ and $\Delta_j^Q = y_{b_j^*} - y_{b_{j-1}^*}$ respectively then the sequence of the patterns becomes

$$
\begin{aligned}
\widehat{T} &= \{(s_1, d_1), (s_2, d_2), \cdots, (s_m, d_m)\} \\
&:= \left\{ (\Delta_1^T, d_1), (\Delta_2^T, d_2), \cdots, (\Delta_m^T, d_m) \right\} \\
\widehat{Q} &= \{(v_1, l_1), (v_2, l_2), \cdots, (v_n, l_n)\} \\
&:= \left\{ (\Delta_1^Q, l_1), (\Delta_2^Q, l_2), \cdots, (\Delta_m^Q, l_m) \right\}
\end{aligned}
\tag{4.13}
$$

and thus the distance by change amounts comparison in the length of trends and range of the values in a trend regime is given by

$$\text{Dist}_{\Delta}(\widehat{T},\widehat{Q}) = \sum_{i=1}^{m} \sqrt{(\Delta_i^T - \Delta_i^Q)^2 + (d_i - l_i)^2}. \tag{4.14}$$

Since the distance in (4.14) is based on the Euclidean distance metric, it satisfies the properties of metric function given in Section 2.3.1 of Chapter 2.

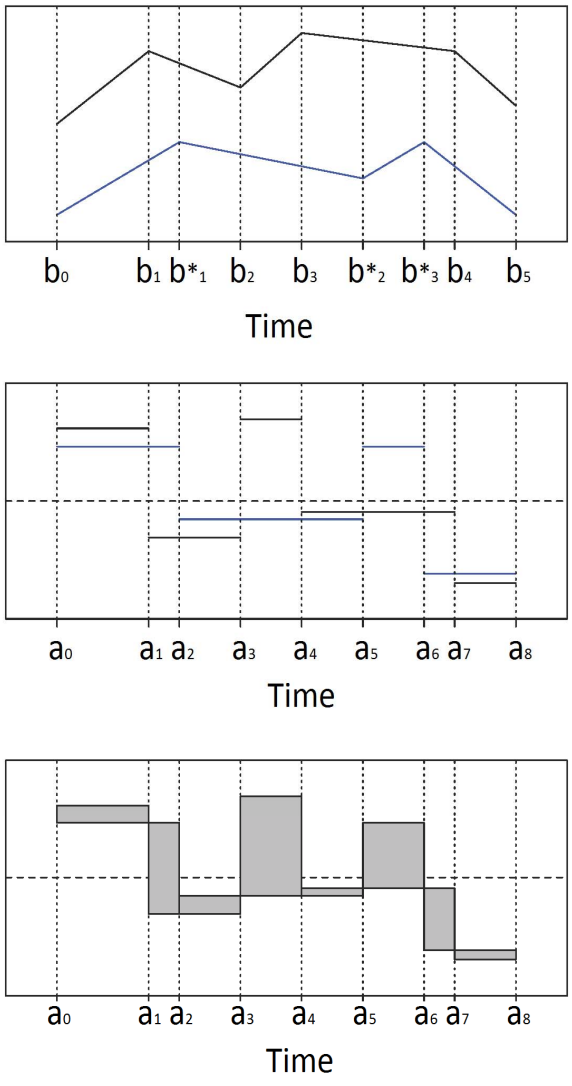

Figure 4.2: Step functions of the trend sequences

4.2.2 Discretized Patterns and Distance Measures

The numerical patterns in Section 4.2.1 can also be transformed to the discretized patterns by categorization of the data. We categorize the numerical sequences of patterns in order since our interests of financial time series are ordinal. We use two alphabet symbols, consonant and vowels, to represent each trend regime. A pair of consonant and vowel represent the direction, the length of the trend, and the magnitude of the trend respectively. For example, with *trend size* = 3, consonant "J", "K", and "L" are assigned on the decreasing trends for short period, moderate length of period, and long period respectively. Similarly, so are 'P", "Q", and "R" assigned for the increasing trends. With the *magnitude size* = 3 three vowels "A", "E", "I" are used to represent the magnitude of the trends in order (Table 4.1). The *trend size* c and *magnitude size* v are predefined parameters. The trend size c and magnitude size v separate categories according to $\frac{i}{c}, (i = 1, \cdots, c)$ and $\frac{j}{v}, (j = 1, \cdots, v)$ quantiles of the distribution of c and v values. By discretization function $\mathbf{D}$ with the parameters c and v, a pair of numerical trends (s_i, d_i) in (4.6) is mapped to a pair of symbols that consists of a consonant and a vowel.

$$\mathbf{D} : (s_i, d_i) \longrightarrow (C_i, V_i) \equiv C_i V_i \tag{4.15}$$

where

$$C_i \in \mathbf{C}^+ \text{ or } C_i \in \mathbf{C}^-, \text{ and } V_i \in \mathbf{V}$$

$$\mathbf{C}^+ = \left\{C_1^+, C_2^+, \cdots, C_c^+\right\}$$

$$\mathbf{C}^- = \left\{C_1^-, C_2^-, \cdots, C_c^-\right\}$$

$$\mathbf{V} = \{V_1, V_2, \cdots, V_v\}$$

C_i^+ $(i = 1, 2, \cdots, c)$ and C_i^- $(i = 1, 2, \cdots, c)$ are consonants to represent the increasing and decreasing trends according to the length of their trends respectively. V_i $(i = 1, 2, \cdots, v)$ is a vowel to represent the magnitude of the i-th trend regime. The value for the consonant C_i of (s_i, d_i) is determined by the direction of s_i and the location of the duration d_i among

$c-1$ quantile values of the distribution of d_i. Specifically, symbol values, C_i and V_i, can be written as follows.

$$\begin{aligned}
C_i =& I(\text{sign}(s_i) \geq 0) \left\{ C_1^+ I\left(d_i < Q_d\left(\frac{1}{c}\right)\right) + C_2^+ I\left(Q_d\left(\frac{1}{c}\right) \leq d_i < Q_d\left(\frac{2}{c}\right)\right) + \cdots \right. \\
& \left. + C_c^+ I\left(Q_d\left(\frac{c-1}{c}\right) \leq d_i\right) \right\} \\
& + I(\text{sign}(s_i) < 0) \left\{ C_1^- I\left(d_i < Q_d\left(\frac{1}{c}\right)\right) + C_2^- I\left(Q_d\left(\frac{1}{c}\right) \leq d_i < Q_d\left(\frac{2}{c}\right)\right) \right. \\
& \left. + \cdots + C_c^- I\left(Q_d\left(\frac{c-1}{c}\right) \leq d_i\right) \right\} \\
=& I(\text{sign}(s_i) \geq 0) \left\{ \sum_{k=1}^{c} C_k^+ I\left(Q_d\left(\frac{k-1}{c}\right) \leq d_i < Q_d\left(\frac{k}{c}\right)\right) \right\} \\
& + I(\text{sign}(s_i) < 0) \left\{ \sum_{k=1}^{c} C_k^- I\left(Q_d\left(\frac{k-1}{c}\right) \leq d_i < Q_d\left(\frac{k}{c}\right)\right) \right\}
\end{aligned} \tag{4.16}$$

where $Q_d\left(\frac{i}{c}\right)$ $(i = 1, 2, \cdots, c-1)$ is the i-th quantile value of distribution of d_is and $I(\cdot)$ is an indicator function. Similarity, V_i can be written as follows.

$$\begin{aligned}
V_i =& V_1 I\left(|s_i| < Q_s\left(\frac{1}{v}\right)\right) + V_2 I\left(Q_s\left(\frac{1}{v}\right) \leq |s_i| < Q_s\left(\frac{2}{c}\right)\right) + \cdots \\
& + V_v I\left(Q_s\left(\frac{v-1}{v}\right) \leq |s_i|\right) \\
=& \sum_{k=1}^{v} V_i I\left(Q_s\left(\frac{k-1}{v}\right) \leq |s_i| < Q_s\left(\frac{k}{v}\right)\right)
\end{aligned} \tag{4.17}$$

where $Q_s\left(\frac{i}{v}\right)$ $(v = 1, 2, \cdots, v-1)$ is the i-th quantile value of distribution of s_i and $I(\cdot)$ is an indicator function. The example is shown in Figure 4.3.

Since the symbolic patterns are discretized based on the ordinal categories, some ordinal numbers may be assigned on these categories for similarity measure. Also $(+)$ or $(-)$ sign

can be used to identify the trend direction. In the example, $1, 2$, and 3 are assigned on "J", "K", and, "L" respectively as they are sorted by the length of trends. Similarly, $-1, -2$ and -3 are assigned for "P", "Q", and, "R" since they represent decreasing trends. The numbers to represent the magnitude of the trends also can be chosen in the same way as they were for the length of the trends, but their signs should coincide with the sign of the trends (Figure 4.4). For two sequences $T^* = \left\{(C_i^1, V_i^1)|\ i = 1, 2, \cdots, m\right\}$ and $Q^* = \left\{(C_i^2, V_i^2)|\ i = 1, 2, \cdots, m\right\}$ transformed by $\mathbf{D}$ in (4.15), the *discrete pattern* (DP) distance can be written as follows.

$$\text{DP distance} = \sum_{i=1}^{m} \sqrt{(C_i^1 - C_i^2)^2 + (V_i^1 - V_i^2)^2} \tag{4.18}$$

The distance measure in (4.18) does not satisfy *identity of indiscernables property* of metric function because we aggregate various patterns based on some cutoff values to the one category. Nevertheless, the distance measure in (4.18) has some advantages. It facilitates simple and fast computation for similarity measure between large scale data. Moreover, it does not require that two time series have the same length as long as the number of trend regimes by ATS or PBS is the same while TSF distance and the distance by change amount comparison are valid for the data registered with respect to time.

Table 4.1: Symbolic Discretized Patterns

J	Up short length	**P**	Down short length	**A**	Small in magnitude
K	Up moderate length	**Q**	Down moderate length	**E**	Medium in magnitude
L	Up long length	**R**	Down short long	**I**	Large in magnitude

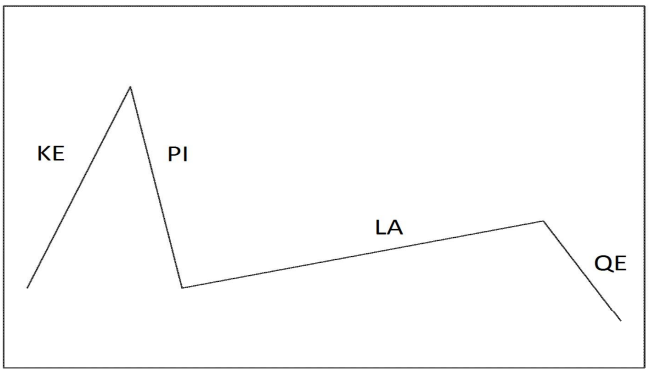

Figure 4.3: Discretized Symbol Patterns

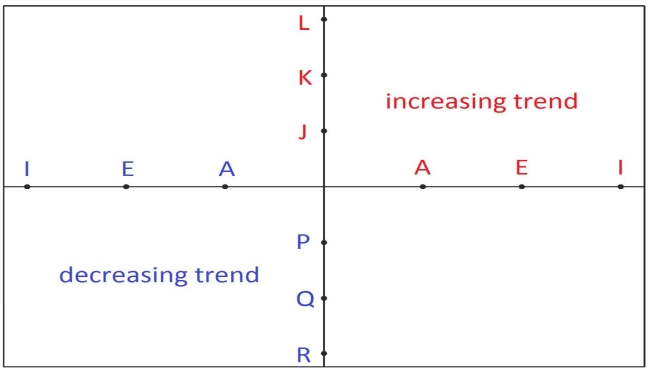

Figure 4.4: Distances for discretized patterns

Example: Matching Discrete Patterns

Figure 4.5 is the result of indexing the best matching sequence of patterns using discrete patterns. Daily closing price data of International Business Machine Corp. (IBM) from October 15, 2010 to October 15, 2015 is used to find two types of patterns, **M** and **W** patterns. First of all, the stock data is smoothed by PBS and transformed to a pair of sequences, the sequence of slope values and the sequence of the length of the trend regimes. Two parameters, *trend size* $(c) = 4$ and *magnitude size* $(v) = 4$, are specified to discretize patterns as seen in Table 4.2. Based on Table 4.2, **M** pattern can be described $\{MO, SO, MO, SO\}$ while a sequence $\{MA, RO, LI, SO\}$ indicates **W** pattern. To find the best matching subsequence of **M** or **W** patterns, we searched sequentially all subsequences (*sliding windows*) and measured distance using *discrete pattern* (DP) distance in (4.18).

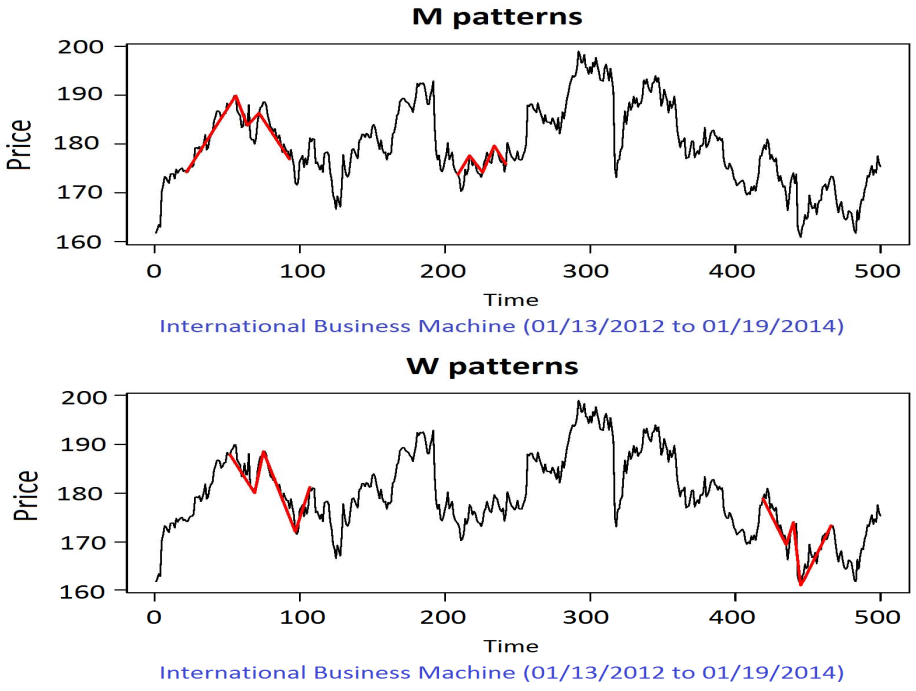

Figure 4.5: Search **M** and **W** patterns from the stock price data (IBM).

Table 4.2: Discretized Patterns for pattern matching

J	Up for very short length	**P**	Down for very short length	**A**	Very small in magnitude
K	Up for short length	**Q**	Down for short length	**E**	Small in magnitude
L	Up for long length	**R**	Down for long length	**I**	Large in magnitude
M	Up for very long length	**S**	Down for very long length	**O**	Very large in magnitude

4.2.3 Adjustments of the Length of the Reduced Data

The distance measures for smoothed data by piecewise linear representation, TSF distance (D_{TSF}) in (4.10), the distance by change amount comparison (D_{Δ}) in (4.14), and the distance of discrete patterns (D_{DP}) in (4.18) require some conditions about data registration and the dimension equality as shown in Table 4.3. In case that the length of the two smoothed sequence of data are not equal, we may adjust one of the sequences so that they have the same length by some post-process. As it has been suggested in many literature, we propose a *merging method* based on a bottom-up algorithm. Let $\widehat{T} = \{s_1, s_2, \cdots, s_m\}$ and $\widehat{Q} = \{v_1, v_2, \cdots, v_n\}$ be sequences of the trends smoothed by ATS of PBS where $m < n$. Then by removing one changepoint from the longer sequence $\widehat{Q}$, the length of $\widehat{Q}$ is reduced to $n - 1$. The step is applied recursively until the lengths of $\widehat{Q}$ is equal to that of $\widehat{T}$. The removal changepoints is based on the perpendicular distance P_d, from the i-th changepoint to the line between $(i-1)$-th and $(i+1)$-th changepoints (Figure 4.6).

Table 4.3: Distance Measures for Smoothed Data

	Required $\widehat{T}$ and $\widehat{Q}$ to be	
	registered w.r.t time	the same length
$D_{TSF}(\widehat{T}, \widehat{Q})$	Yes	No
$D_{\Delta}(\widehat{T}, \widehat{Q})$	Yes	Yes
$D_{DP}(\widehat{T}, \widehat{Q})$	No	Yes

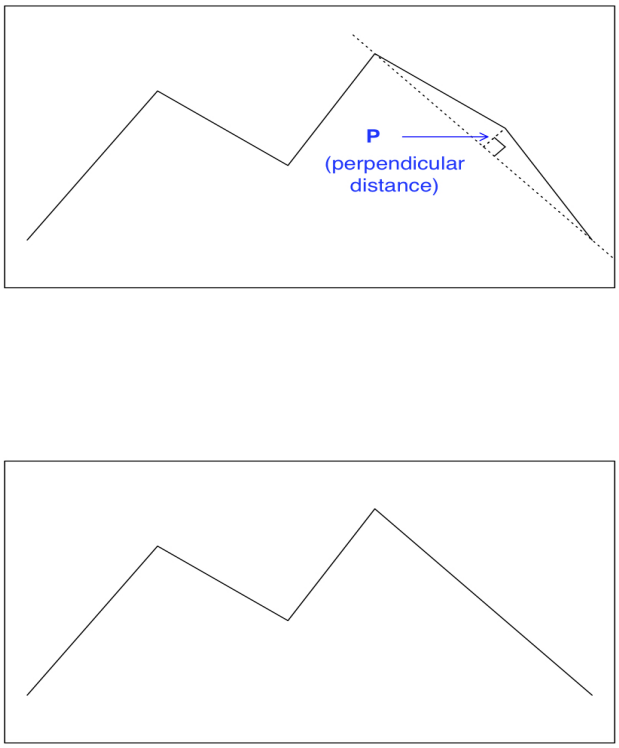

Figure 4.6: Merging two trend regimes

4.3 Properties of the Length of the Trend Regime

In piecewise band approximation, the quantities that describe a trends of a regime such as the direction and magnitude of the linear slope, and the lengths of trends can vary based on the choice of parameters, initial window size w, bandwidth B, change ratio R, and angle restriction A. It is observed that when we smooth a time series data by PBS with initial window size w, the lengths of the trend regime often tend to have close values to w, while the the trends (slopes values) show a consistently similar distribution, symmetric with zero mean regardless of the choice of parameters. In this section, we investigate the properties of the length of a trend regime given initial window size w.

4.3.1 Assumptions about Data and Models

As mentioned in Chapter 1, we assume that a large size time series data consists of non-overlapping sequential data generating processes, such as linear models. Specifically, suppose a time series T consists of K linear trend regimes,

$$T = \{x_{11}, x_{12}, \cdots, x_{1n_1}, \cdots, x_{K1}, x_{K2}, \cdots, x_{Kn_K}\}.$$

Assume that each trend regime has a underlying linear model for all K as follows.

$$x_{it_j} = \beta_{i0} + \beta_{i1} t_j + \epsilon_{it_j} \quad (i = 1, 2, \cdots, K \text{ and } j = 1, 2, \cdots, n_K) \tag{4.19}$$

where ϵ_{it_j}'s are independent with zero mean and finite variance $Var(\epsilon_i) = \sigma_i^2 < \infty$. Hence, each trend regime can be fitted by

$$\hat{x}_{it_j} = \hat{\beta}_{i0} + \hat{\beta}_{i1} t_j \quad (i = 1, 2, \cdots, K \text{ and } j = 1, 2, \cdots, n_K) \tag{4.20}$$

and we have residuals $r_{it_j} = x_{it_j} - \hat{x}_{it_j}$ from the fitted model. Recall that given the initial window size $w > 3$ and bandwidth $B > 0$, we compare the residual of the observation x_{w+1}, $|x_{w+1} - \hat{x}_{w+1}|$ to B to determine if x_w is the trend changepoint or not. Therefore, the probability of the length of a trend regime $L = w$ can be written by

$$\begin{aligned} \Pr(L = w) &= \Pr(|x_{w+1} - \hat{x}_{w+1}| > B) \\ &= \Pr(|r_{w+1}| > B) \\ &= p \end{aligned} \tag{4.21}$$

where $1 - p = \Pr(|x_{w+1} - \hat{x}_{w+1}| \leq B) = \Pr(|r_{w+1}| \leq B)$. Similarly, the probability of the length of a trend regime $L = w + 1$ is,

$$\begin{aligned}\Pr(L = w+1) &= \Pr(|x_{w+2} - \hat{x}_{w+2}| > B)\ \Pr(|x_{w+1} - \hat{x}_{w+1}| \leq B) \\ &= \Pr(|r_{w+2}| > B)\ \Pr(|r_{w+1}| \leq B) \\ &= p(1-p)\end{aligned} \tag{4.22}$$

because r_i's are independent.

In the same fashion, the probability of the length of a regime $L = w + l$ $(l \geq 0)$ can be written by

$$\Pr(L = w + l) = p(1-p)^l \tag{4.23}$$

Therefore, given an initial window size w, the length of a trend regime $L(\geq w)$ follows geometric distribution with $p = \Pr(|x_{w+i} - \hat{x}_{w+i}| > B)$ where $i \geq 0$. Figure 4.7 illustrates the distribution of trend regimes after smoothing IBM daily closing price data from January 13, 2005 to January 12, 2015 by PBS with $w = 4$ and constant bandwidth $B = 2$. The distribution of the length of trend regimes seems to be approximately geometric distribution with $p = 0.21$.

4.3.2 The Properties of p

Based on the assumption that we made about data in Section 4.3.1, the probability that PBS identifies the true changepoint can be estimated as follows. Suppose there are two trend regimes,

$$\begin{aligned}\text{Trend 1 } (l_1): \ x_{1t_j} &= \beta_{10} + \beta_{11} t_j + \epsilon_{1t_j} \\ \text{Trend 2 } (l_2): \ x_{2t_j} &= \beta_{20} + \beta_{21} t_j + \epsilon_{2t_j}\end{aligned} \tag{4.24}$$

where ϵ_{it_j} has a zero mean and a finite variance σ_i $(i = 1, 2)$, and Trend 1 changes at time c; that is x_c is the true changepoint. To satisfy the continuity constraints, we will consider only $\beta_{i1},\ (i = 1, 2)$ for a trend in the i-th regime. Let p be the probability of x_c being

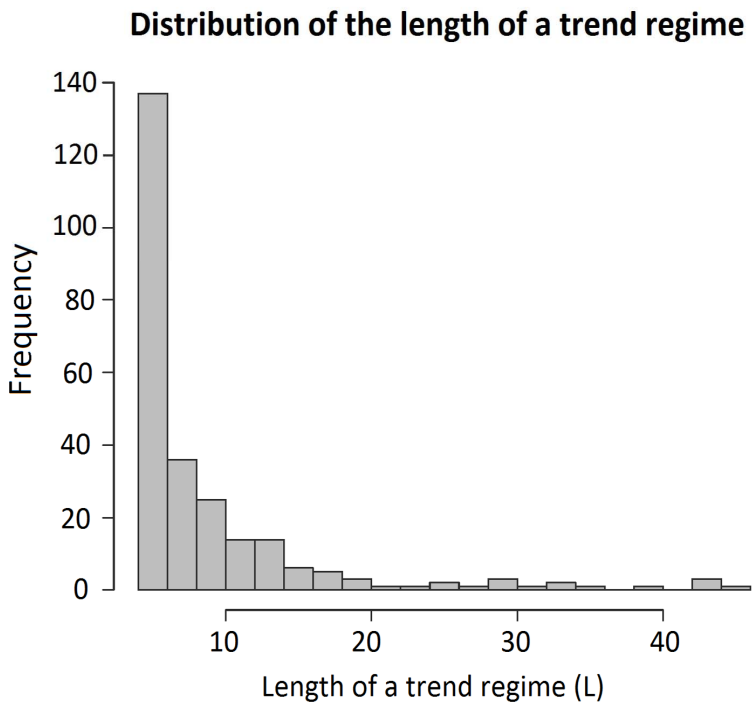

Figure 4.7: The distribution of the length of a trend regime with $w = 4$

identified as the trend changepoint by PBS. Since PBS determines a changepoint based on the distance between the future observation x_{c+1} and the current trend line l_1, the farther distance of the x_{c+1} from l_1 implies the higher probability of x_{c+1} being identified as a changepoint. That is, as the difference between two trends $d = |\beta_{21} - \beta_{11}|$ increases, p also increases (Figure 4.8).

Suppose ϵ_{it_j} $(i = 1, 2)$ is an identically independent normal distribution with zero mean and variance σ_i $(i = 1, 2)$, and $\beta_{11} \neq \beta_{21}0$. Then, the observation at $c + i$ can be written as $x_{c+i} = \beta_{20} + \beta_{21}(c + i) + \epsilon_{2,c+i}$ $(i > 0)$ and $x_{c+i} = \beta_{10} + \beta_{11}(c + i) + \epsilon_{1,c+i}$ $(i \leq 0)$.
Let

$$Y = \begin{cases} 1 & \text{if } \hat{x}_c = x_c | \hat{x}_c \neq x_p \quad (p = w, w+1, \cdots, c-1) \\ 0 & \text{if } \hat{x}_c \neq x_c | \hat{x}_c \neq x_p \quad (p = w, w+1, \cdots, c-1) \end{cases} \tag{4.25}$$

where $\hat{x}_t$ is a changepoint identified by PBS at t. Consider how p changes when $d = B$, where $d = |\beta_{21} - \beta_{11}|$ and B are the difference between two trends and the bandwidth of

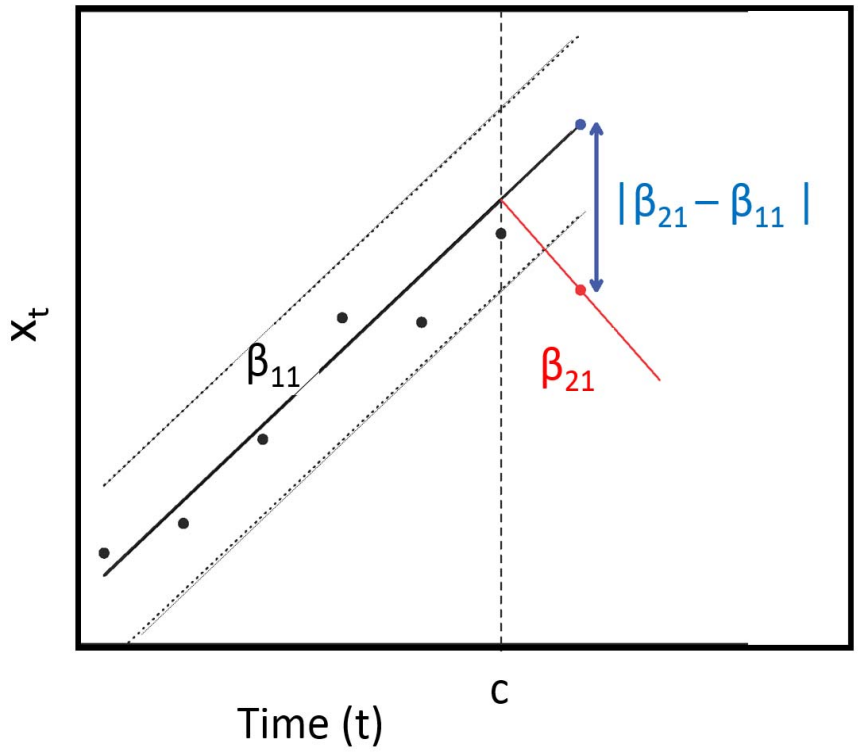

Figure 4.8: The probability that PBS identifies a true changepoint

the first regime respectively. Given the bandwidth B, $Y = 1$ when $x_{c+1} < \beta_{20} + \beta_{21}(c+1)$, that is $\epsilon_{2,c+1} < 0$. Therefore $Y = 1$ with probability 0.5. If $d < B$, $Y = 1$ when $x_{c+1} < \beta_{20} + \beta_{21}(c+1) - d'$ $(d' > 0)$, that is $\epsilon_{c+1} < -d' < 0$ and thus $Y = 1$ with probability less than 0.5. Similarly, if $d > B$, $Y = 1$ with probability greater than 0.5. Hence, it is easily inferred that,

$$\Pr(Y = 1) \longrightarrow 1 \quad \text{as} \quad (d - B) \longrightarrow \infty \tag{4.26}$$

and

$$\Pr(Y = 1) \longrightarrow 0 \quad \text{as} \quad (d - B) \longrightarrow -\infty \tag{4.27}$$

Indeed, $P(Y = 1)$, the probability that PBS identifies true changepoint, has a logic regression curve as seen in Figure 4.9 and Figure 4.10. The graphs on the left side in Figure 4.9 and Figure 4.10 show the simulation results about the change of the probability

$P(Y = 1)$ as d increases given $B = 3$ and $B = 5$ respectively. The red curves indicate the logistic regression curve,

$$p = P(Y = 1) = \frac{e^{d-B}}{1 + e^{d-B}} \quad or \quad \log \frac{p}{1-p} = d - B \tag{4.28}$$

The graphs on the right side are simulation results when PBS identifies changepoints with change ratio constraints. With change ratio constraints, the probability p is always lower than the curve by (4.28). This is because PBS rejects any identified points that do not satisfy change ratio constraints although they fall outside of the bandwidth.

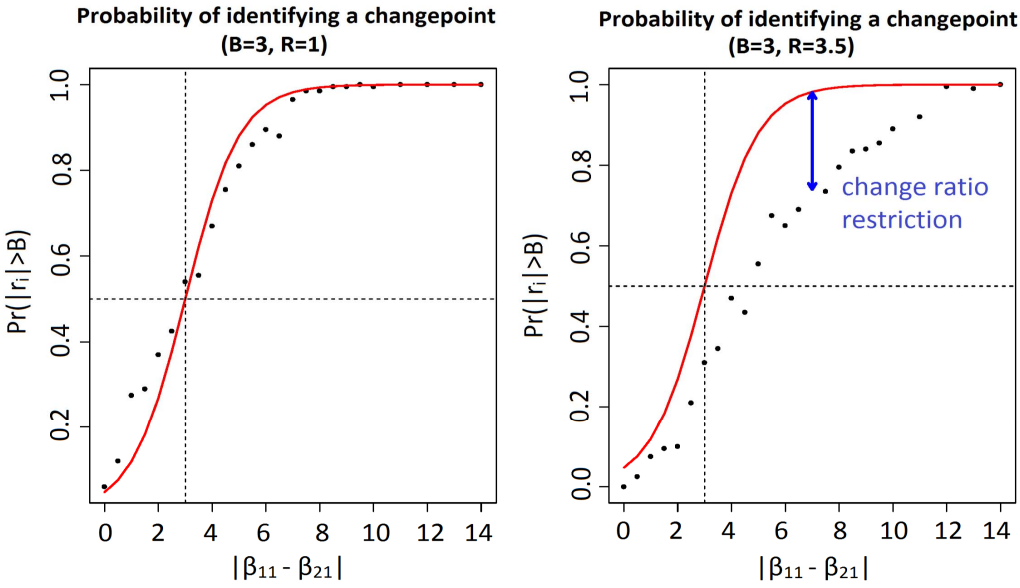

Figure 4.9: The distribution of p with bandwidth $B = 3$ (left) and with change ratio restriction (right)

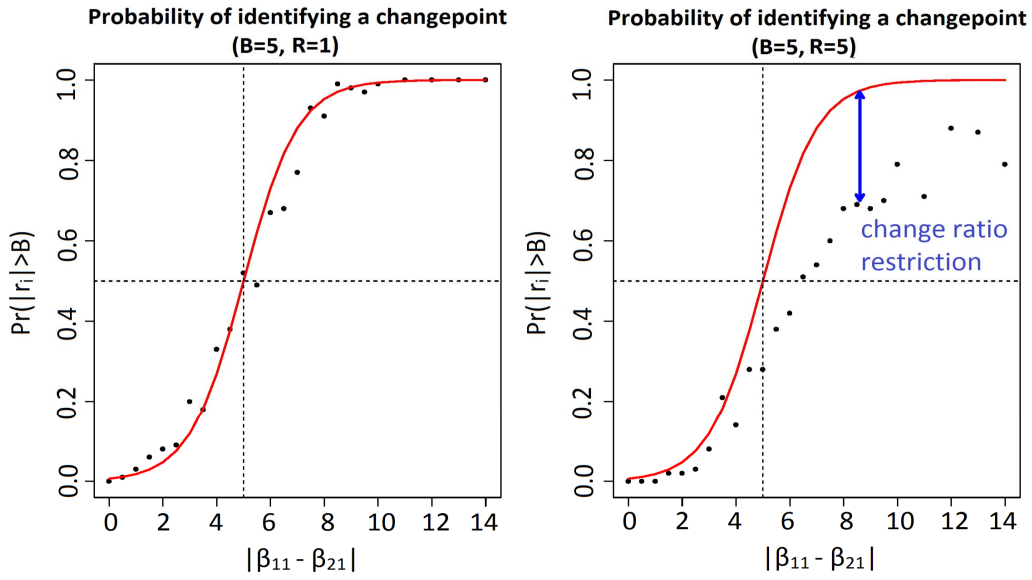

Figure 4.10: The distribution of p with bandwidth $B = 5$ (left) and with change ratio restriction (right)

Now, we consider p for more general cases. Let $p_k = \Pr(\hat{x}_c = x_{c+k})$ be the probability of x_{c+k} being identified as a changepoint by PBS ($c+k \geq w$), where x_c is the true changepoint. Specifically, $k < 0$ implies that PBS identifies a trend changepoint in k advance of true changepoint x_c, and $k > 0$ implies that the k ahead point is identified as a changepoint. Figure 4.11, Figure 4.12, and Figure 4.13 illustrate simulation results of the change of p_k under three different scenarios, (i) $\sigma_1 = \sigma_2$, (ii) $\sigma_1 > \sigma_2$, and (iii) $\sigma_1 < \sigma_2$. For all three scenarios, $p_0 = \Pr(\hat{x}_c = x_c)$ - the probability that PBS identifies a changepoint correctly - increases as d increases. Note that PBS sometimes identifies points as changepoints before it reaches the true changepoint. This is because we use only w points to fit the current trend line. Thus, the gap between the true trend and the fitted trend is getting larger as PBS moves forward along with the fitted line, the points in the current trend regime are possibly identified as changepoints (see Figure 4.14 left). Under the scenarios $\sigma_1 = \sigma_2$ and

$\sigma_1 < \sigma_2$ when d is small, PBS tends to identify changepoints far after it passes the true changepoint (Figure 4.11 and Figure 4.13 left). It can be easily understood from Figure 4.14 (middle). When d is small, the bandwidth in the first regime possibly covers many points in the second regime near the true changepoint, then PBS cannot identify any point as a changepoint until it moves further toward the end of the second regime. Under the scenario $\sigma_1 > \sigma_2$, when d is small, PBS tends not to identify any points even until it moves to the end of the second regime (Figure 4.12 left). This is because of the same reason as the case in Figure 4.14 middle. Additionally the wider bandwidth in the first regime embraces most points in the second regime, therefore PBS may not identify any deviated points outside of the bandwidth (Figure 4.14 right). These issues may be alleviated by some modifications in defining adaptive bandwidth as we will discuss in Chapter 6.

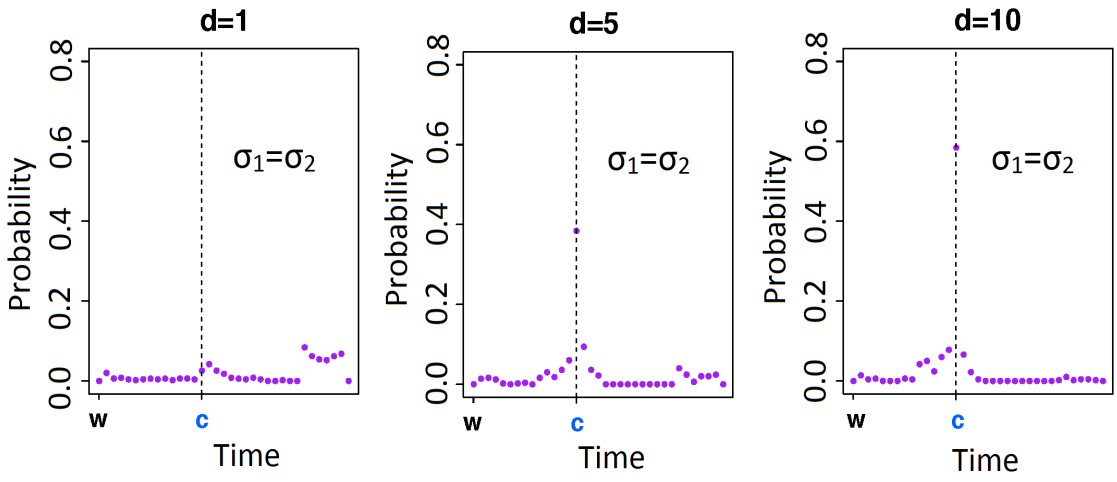

Figure 4.11: The probabilities of the points being identified as the changepoint with $d = 1, 5$ and 10.

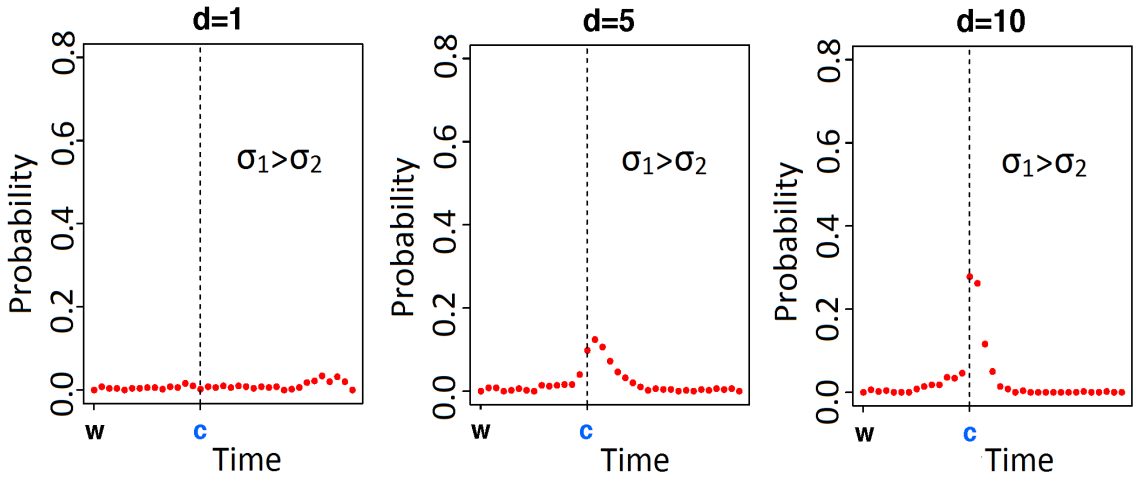

Figure 4.12: The probabilities of the points being identified as the changepoint with $d = 1, 5$ and 10.

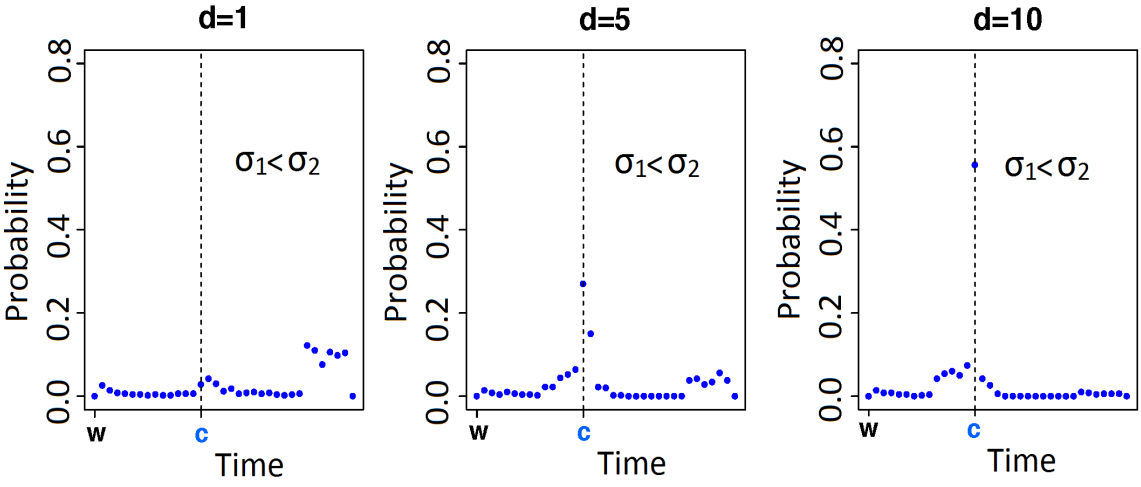

Figure 4.13: The probabilities of the points being identified as the changepoint with $d = 1, 5$ and 10.

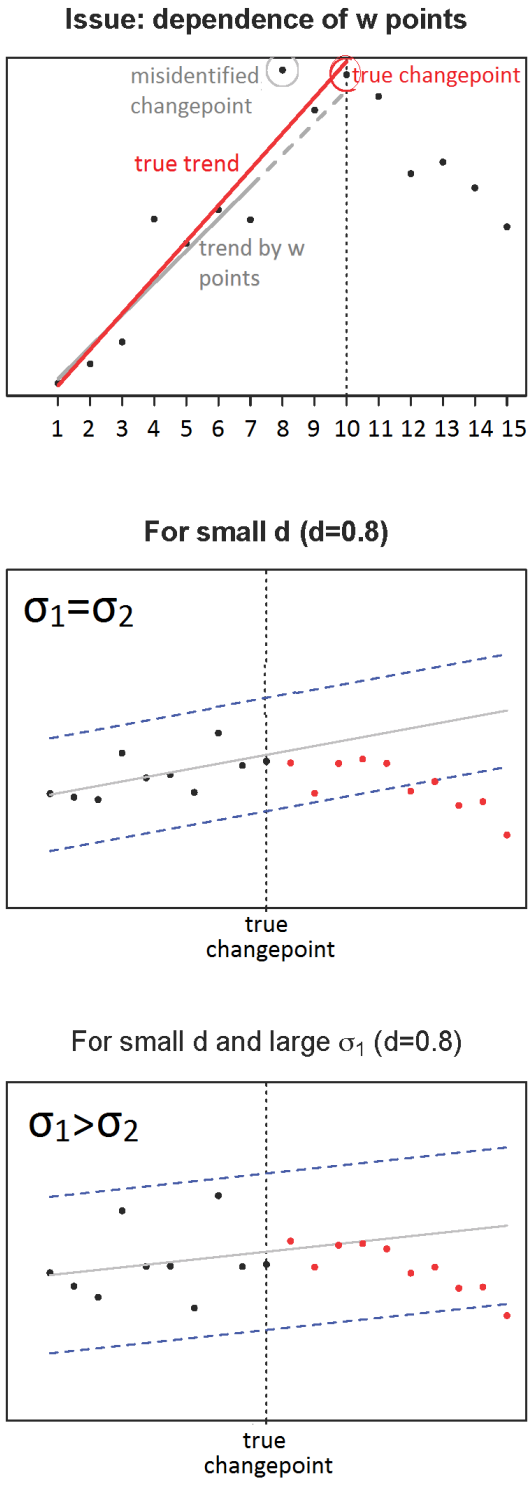

Figure 4.14: The fitted trend line using w points (top), small d (middle), and small d and $\sigma_1 > \sigma_2$ (bottom).

Chapter 5: Application Examples and Experimental Evaluation

In this chapter, we demonstrate two examples of pattern recognition in time series using *alternating trends smoothing* (ATS) and *piecwise band smoothing* (PBS) for stock price datasets [1]. The two datasets used in the examples were selected subjectively and grouped based on the similarity of their up and down trends over time. The purpose of these application examples is not to evaluate the performance of ATS and PBS.

After these application examples, we evaluate our methods for classification and segmenting by comparing other methods. The evaluation is performed based on Keogh et al. (2003, 2004) [35] [36] and Ding et al. (2008) [14].

5.1 Application Examples

5.1.1 Example 1: Clustering Groups with Similar Trends

Clustering, also known as *unsupervised learning*, aims to categorize the observations into natural groups by minimizing intra-cluster distances and maximizing inter-cluster distances. The members categorized in the same group are desirably homogeneous. The number of categories may or may not be pre-determined. If there are the labels of the natural groups, the labels are not used in the clustering. In time series analysis, *partitioning* and *hierarchical clustering* are the most popular approaches for clustering [50]. Here, we use hierarchical clustering.

The purpose of the example is to illustrate a method of clustering of time series; not to evaluate the performance of the method.

Hierarchical Clustering

While the widely-used K-means clustering method requires a pre-specified number of clusters, hierarchical clustering does not require information on the number of clusters or the initial conditions. Hierarchical performs clustering either in an *agglomerative* (bottom-up) or divisive (top-down) way. Here, we discuss *agglomerative* clustering. Agglomerative clustering starts from individual observations and keeps merging them to the nearest cluster based on similarity. The steps for agglomerative clustering are described as follows [30].

1. Begin with n observations and measure distances between all possible $\binom{n}{2} = \frac{n(n-1)}{2}$ pairs of observations. Treat each observation as its own cluster.
2. For $i = n, n-1, \cdots$:
    a. Measure all pairwise inter-cluster dissimilarities among i clusters, identify the pair of clusters that has the minimum inter-cluster distance, and then combine these two clusters.
    b. Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Generally, Euclidean distance is used as the distance measure in step 1 while there are several methods of measuring inter-cluster dissimilarities, referred to *linkage*, in step 2a. Some commonly used linkages are as follows:

- **Complete linkage:** Maximal inter-cluster dissimilarity.
- **Single linkage:** Minimal inter-cluster dissimilarity.
- **Average linkage:** Mean inter-cluster dissimilarity.
- **Ward's method**: The sum of squared deviations from points to centroid.

The hierarchical clustering represents the result of a tree-based diagram branching out downward, called a *dendrogram*. The dendrogram not only provides a record of the clustering process but also facilitates the choice of the number of clusters, by cutting off the links

at some height. Note that the proximity among observations along the horizontal axis in a dendrogram does not indicate the degree of similarity, but rather the degree of similarity is indicated by vertical distances among clusters.

Datasets (1): Daily stock price data from two industries (2 clusters)

As an example, we perform hierarchical clustering with stock price data from two industries, airlines and restaurants. Each time series is smoothed by alternating trends smoothing and transformed to a sequence of the magnitude of its trends. The motivation for choosing this example is that prices of stocks in similar industries may exhibit similar patterns of price movements. Prices of stocks in different industries may reflect different sensitivities to market changes. Daily closing price data of 24 stocks listed on major stock exchanges from January 1, 2014 to December 31, 2014 are used for clustering analysis. Twelve stocks are from the airline industry and twelve from the restaurant industry. The length of each time series is 252. The datasets from these two industries are presented in Figure 5.1. The name and group of data are referred to in **Appendix A** Table A.1.

Datasets (2): Daily stock price data from four industries (4 clusters)

As a second example, we use daily closing price data for 27 companies listed in NYSE from October 15, 2010 to October 15, 2015, and each time series belongs to one of four market sectors, Utilities-Electricity (**U**), Finance Insurance (**F**), Investment Mortgage & Bank (**M**), and Drug Manufacturer (**D**). The original data are shown in **Appendix A** Table A.2. The length of each time series is 1,259. The plots of datasets are shown in Figure 5.2.

Data Representation using (ATS) and the Choice of Parameters

We represent these stock prices by, the alternating trends smoothing (ATS) method to transform the original data to a sequence of the magnitude of their trends; that is, the absolute values of their slopes. The step size parameter (h) is 4 for 2-cluster example as *a week* step size, and 20 is used for 4-cluster example based on the relative size of the datasets.

Similarity Measure and Linkage for Hierarchical Clustering

We use the agglomerative method for hierarchical clustering of our data. In most of our work, we measure dissimilarities between time series by simple Euclidean distance but here we measure similarities of all the pairwise transformed sequences using dynamic time warping (DTW) with a Sakoe-Chiba band instead of simple Euclidean distance because (1) the length of transformed sequences may be different and (2) we suppose that the overall shape evolving the trends' magnitudes in the same sector are similar by warping the horizontal axis. For linkage, we use Ward's method which clusters based on "information loss" by minimizing the sum of squared errors of any two clusters.

Results of Hierarchical Clustering

Figure 5.3 shows the results of clustering using ATS representation. Based on the clustering of smoothed data, 8 out of 12 stock data in airline companies are clustered in the same group. For the restaurants' stock data, 10 out of 12 are grouped in the same cluster. This seems to work quite well although the size of the dataset is small. Note that the variation of vertical distances in the right cluster (airline industry) seems to be smaller than that in the left cluster (restaurant industry). This is relevant to the compactness of clusters. Indeed, the intra-cluster distance for airline group is 0.072 and that of restaurants is 0.141. *P*-values via bootstrap resampling for clusters are shown in Figure 5.5 to assess the clustering analysis results. The numbers on the left (red) and right (green) are *approximately unbiased* (AU) *p*-values and *bootstrap probability* (BP) obtained by ordinary bootstrap resampling respectively. AU is calculated by multiscale bootstrap resampling, and it is superior over BP in bias. See [54] for details. AU can have a value between 0 and 100, and a higher AU indicates that the cluster is strongly supported by data.

In the result of the 4-cluster problem (Figure 5.4), an outlier (Goldman Sachs Group Inc.) in Mortgage group (the first cluster from the left) influenced the clustering process.

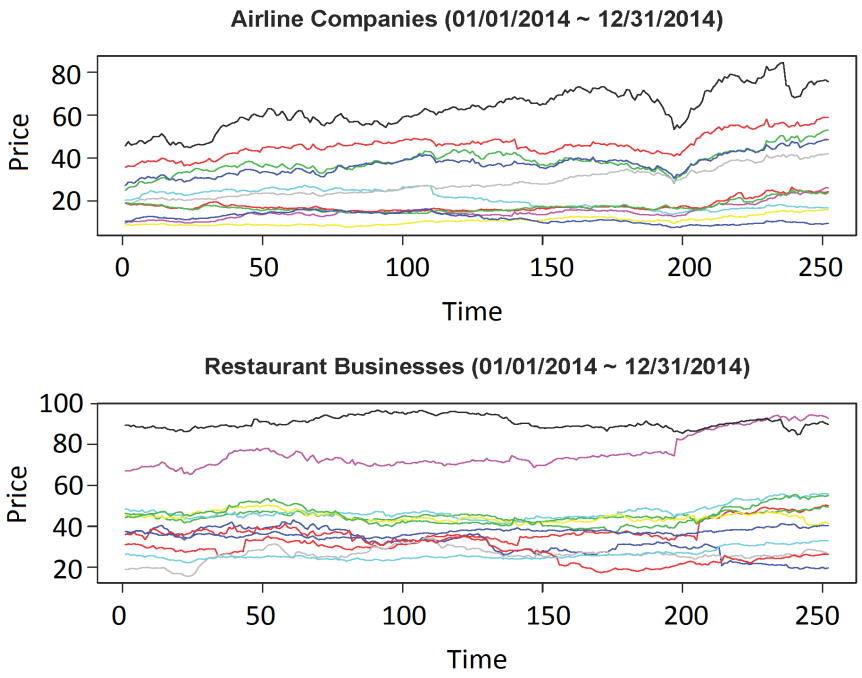

Figure 5.1: Daily closing prices from airline and restaurant industries.

Except for this outlier, the datasets in Mortgage and Drug groups seem to be more homogeneous compared to the other two groups. Based on our assumption and motivation, this result may be interpreted that stocks in the same industry might tend to react and evolve similarly given the market condition. The purpose of this example, however is just to illustrate how our smoothing and data reduction techniques can be used to cluster time series data.

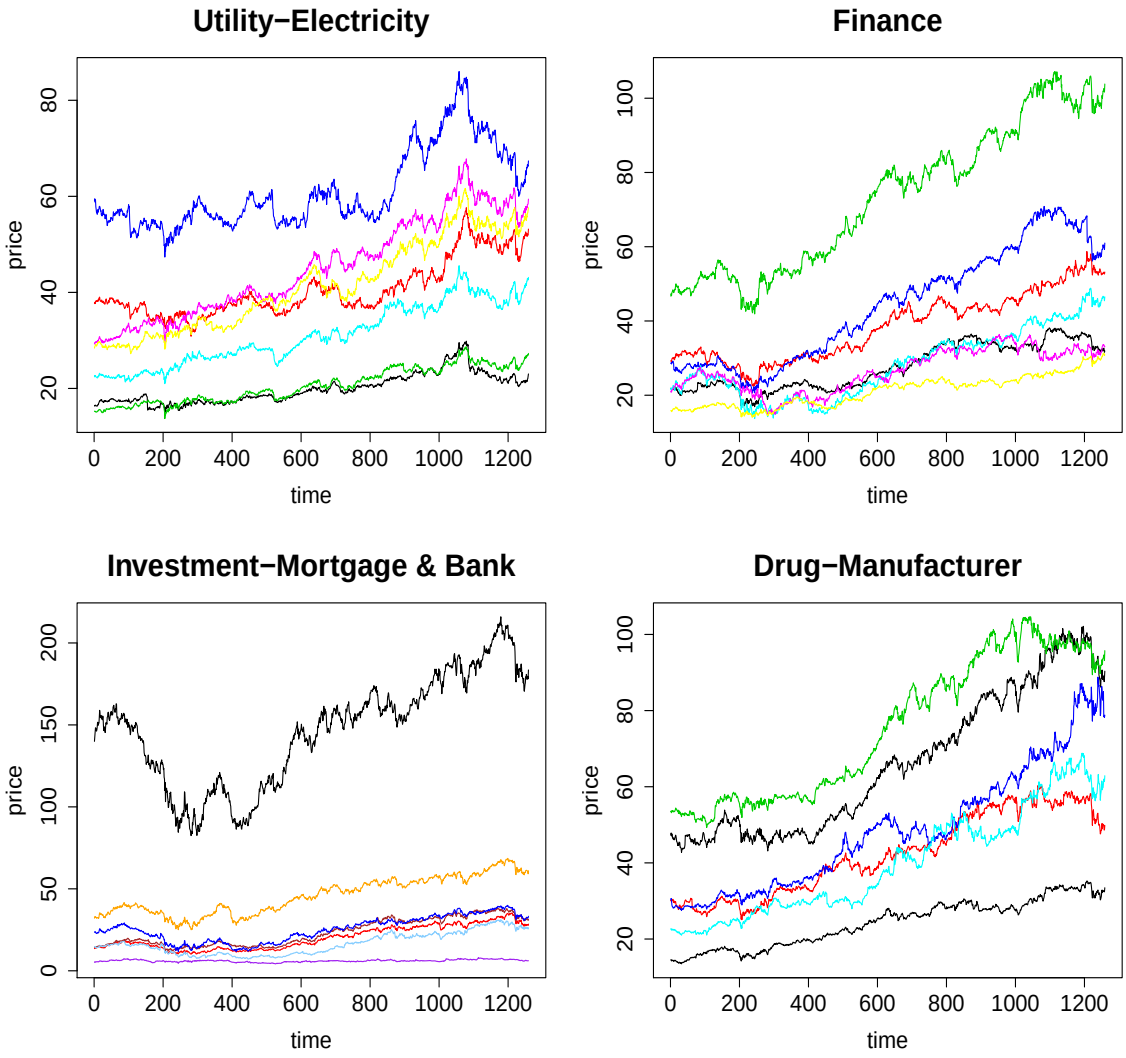

Figure 5.2: Daily closing price of 27 stocks in 4 market sectors

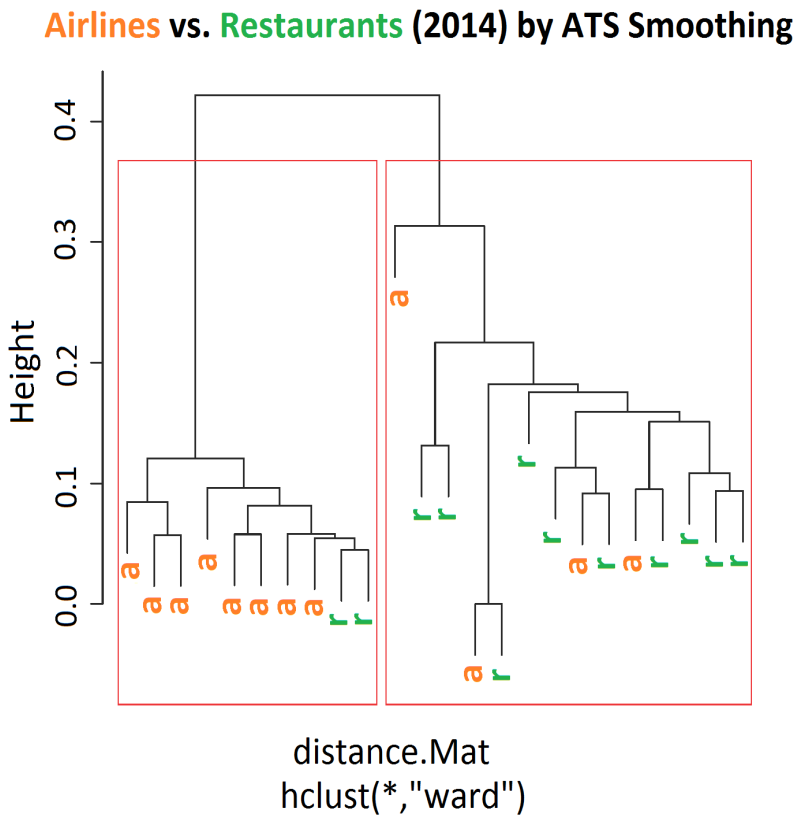

Figure 5.3: Hierarchical clustering (2 clusters)

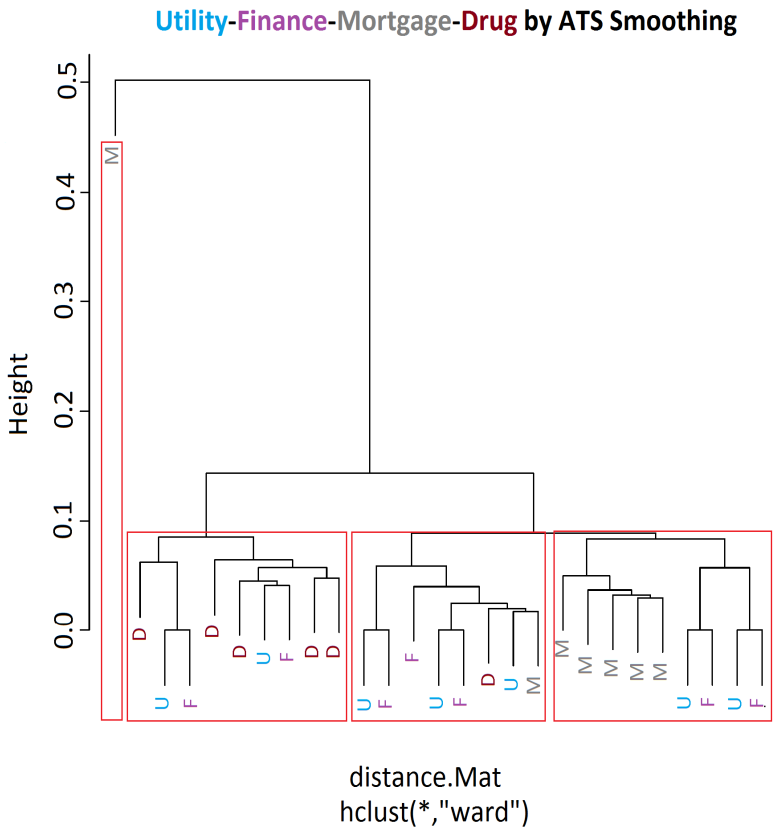

Figure 5.4: Hierarchical clustering (4 clusters).

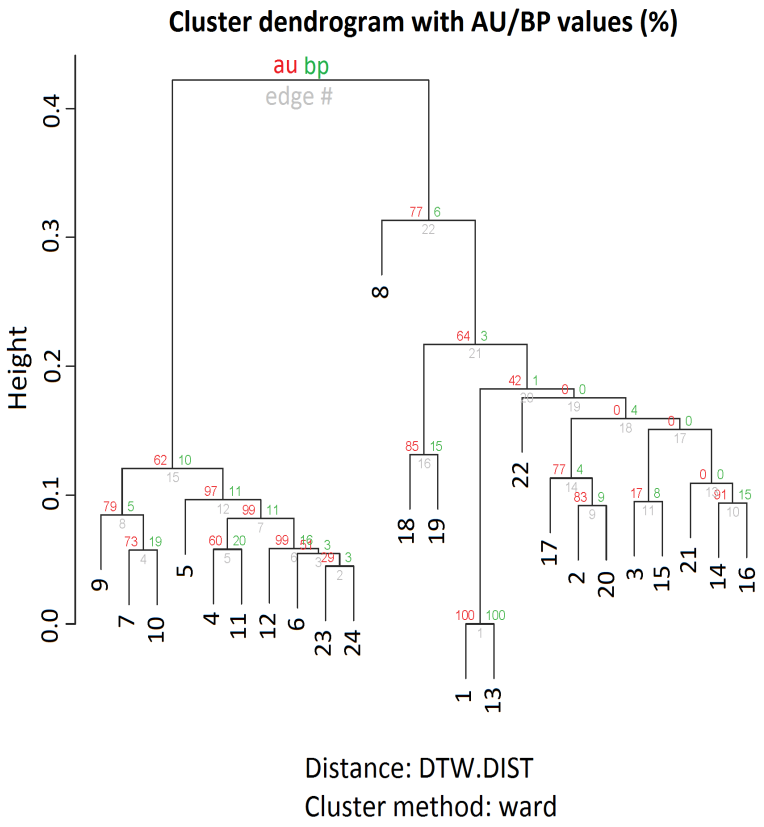

Figure 5.5: Assessment of the clustering of smoothed data by ATS with DTW distance. Airline stocks are from 1 to 12 and restaurant stocks are from 13 to 24.

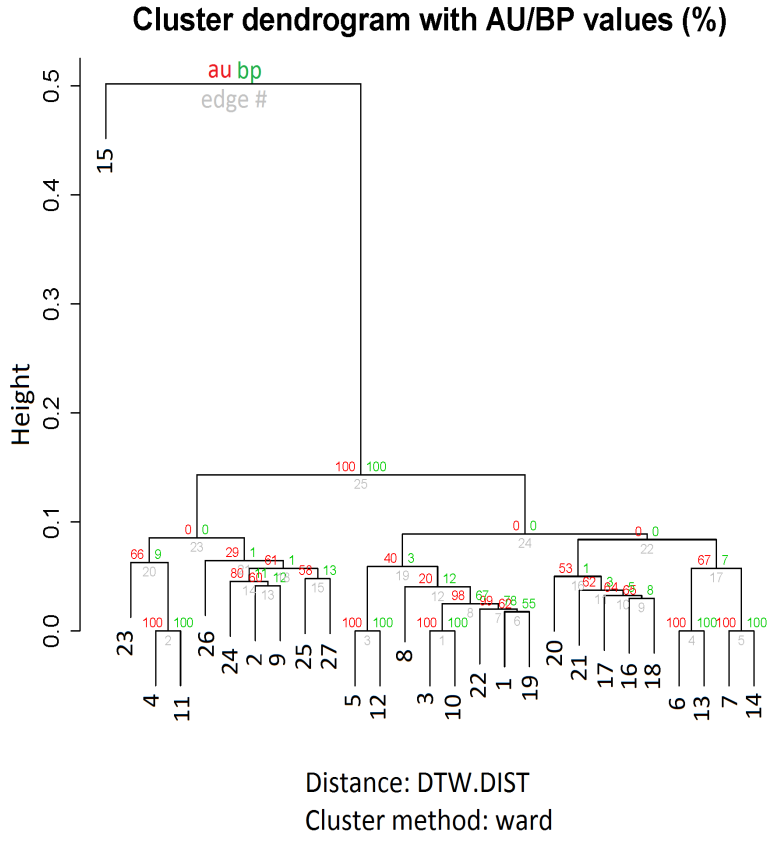

Figure 5.6: Assessment of the clustering of smoothed data by ATS with DTW distance. From 1 to 7 are stocks for Utilities-Electricity group, from 8 to 14 for Finance-Insurance group, from 15 to 21 for Investment Mortgage & Bank group, and from 22 to 27 for Drug-Manufacturer group.

5.1.2 Example 2: Stock Market Sector Classification

Classification is another important technique for learning about relationships in data. Classification assigns observations to one of the pre-defined (or *labeled*) groups. Some sets of labeled data are used to train the classifier. Many classifiers have been used in time series, including K-nearest neighbors (K-NN), support vector machine (SVM), decision trees, and so forth. Here we utilize the simplest classification approach K-NN and demonstrate the result of stock market sector classification using the 1-nearest neighbor method.

Again, as we stated in the beginning of Chapter 5, this classification example is not intended to be an evaluation of the performance of *piecewise band smoothing* (PBS). For this example, we selected some subsets of stocks that have "visually similar" patterns of prices. The purpose of this example is to illustrate our method for classification and to see if our smoothing method would detect this similarity.

K-Nearest Neighbors

The K-nearest neighbors (K-NN) method of classification assigns a new observation x_0 to the class corresponding to the most frequent class of the K-nearest classified observations to x_0. Let $\mathcal{C} = \{c_j \mid j = 1, 2, \cdots, J\}$ be a set of classes. Then K-NN identifies K closest data to x_0 based on the given distance measure, denoted by ν, and estimates the conditional probability of class j from the proportion of classes in ν as follows.

$$P(Y = j | X = x_0) = \frac{1}{K} \sum_{i \in \nu} I(y_i = j) \tag{5.1}$$

In time series analysis, 1-NN with DTW has been widely used for classification [16] [14]. In this section, we show the result of stock market sector classification using 1-NN with *transformed step function* (TSF) distance.

Datasets: Daily stock price data from four market sectors

We perform 4-trend class classification using 1-NN *transformed step function* (TSF) in

(4.10). Datasets in each class are chosen based on their "visual similarity" of pattern from a market sector. Of course, in practice, prices of stocks in the same sector do not always evolve with the same or even similar patterns for a given time period. However, in the belief that there might exist some industry-characterized patterns given the same market condition, for example how sensitively an industry reacts to a news story, we collect stock price datasets from four different sectors because it is easy to find a group of stocks that show "visually similar" patterns in the same sector.

For the classification example, the datasets used were from the 4-cluster example in Section 5.1.1. The "visual trend similarities" are determined based on local features of time series data at specific time intervals. Local features of time series in each class are as follows.

- Stock price data in Utilities-Electricity class have local features (Figure 5.7)
  1. a sharp valley around time 200
  2. slow increase after short drop and then a "M" pattern between time 500 and 700 ("M" pattern means that the data is shaped similar to the letter "M")
  3. a sharp peak around time 1100

- Stock price data in Finance-Insurance class have local features (Figure 5.8)
  1. a drop - short flat - jump pattern between time 190 and 270
  2. short drop and increasing pattern between time 800 and 850

- Stock price data in Investment-Mortgage & Bank class have local features (Figure 5.9)
  1. a peak between time 270 and 410
  2. a "W" pattern between time 950 and 1150 ("W" pattern means that the data is shaped similar to the letter "W")

  (Without *Goldman Sachs*, the detail shape of other stock price data seen in Figure 5.10)

- Stock price data in Drug-Manufacturer class have local features (Figure 5.11)
  1. a jump - flat - drop pattern between time 100 and 200
  2. two sharp drops (a "W" pattern) between time 970 and 1020

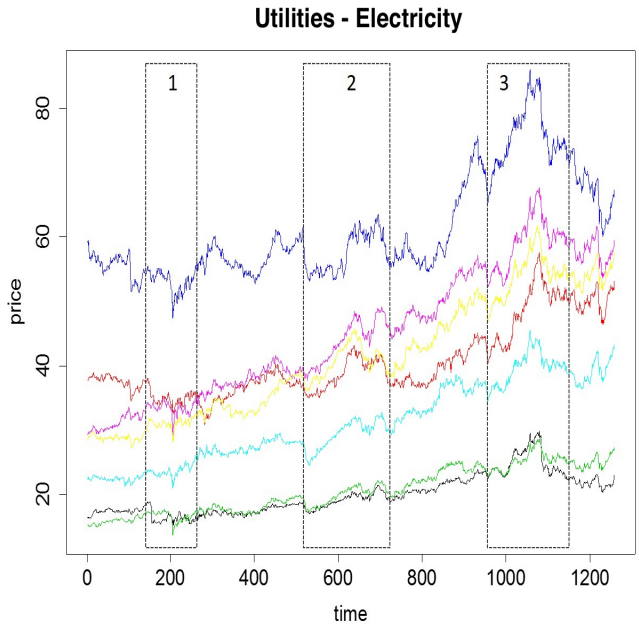

Figure 5.7: Local features of data in Utilities-Electricity class

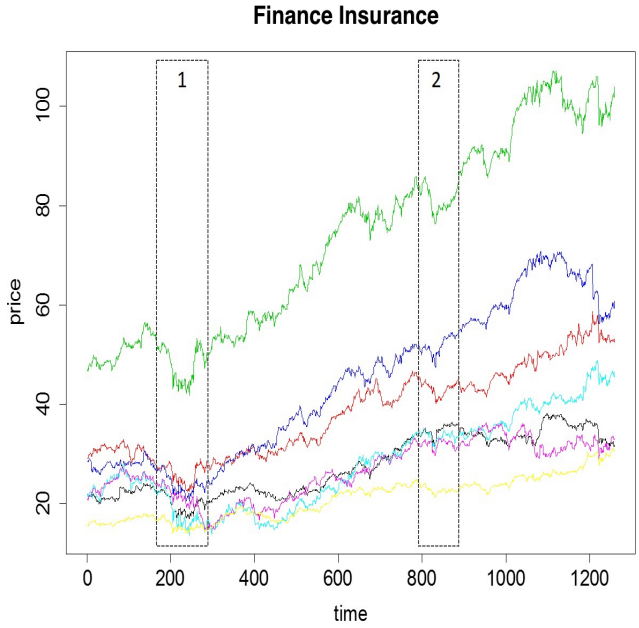

Figure 5.8: Local features of data in Finance-Insurance class

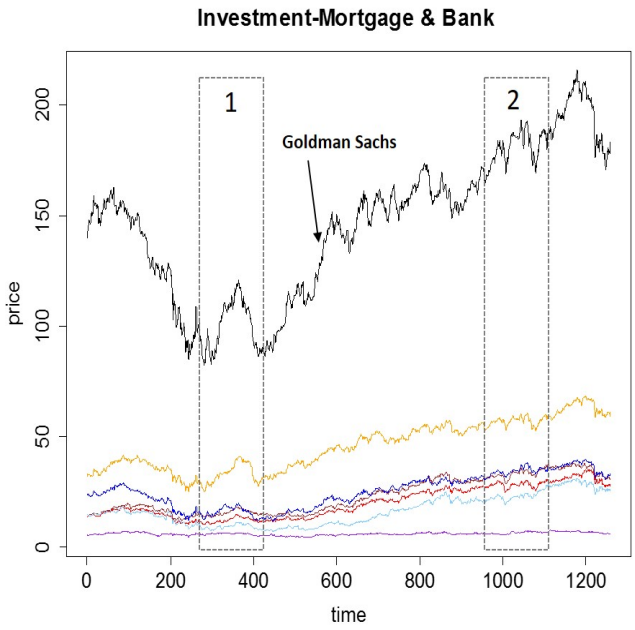

Figure 5.9: Local features of data in Investment-Mortgage & Bank class

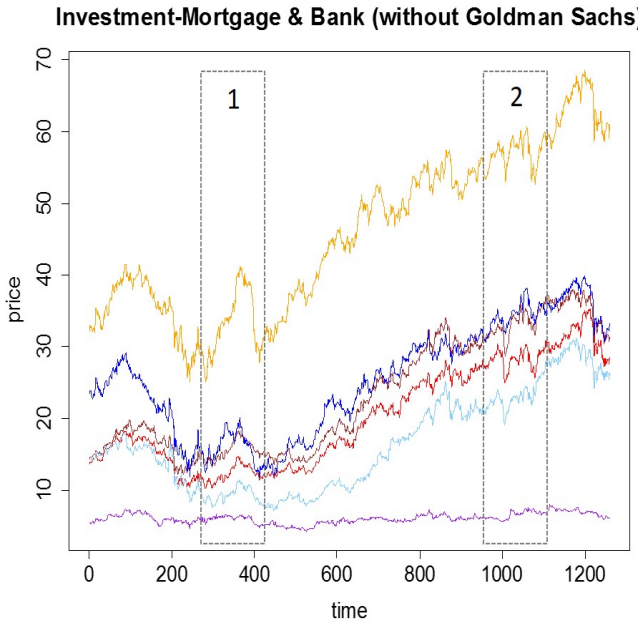

Figure 5.10: Local features of data in Investment-Mortgage & Bank class (without *Goldman Sachs* data)

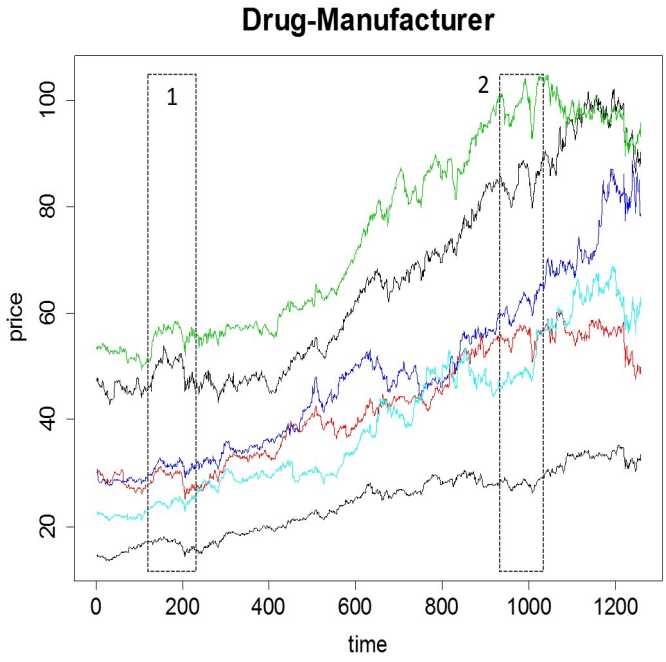

Figure 5.11: Local features of data in Drug-Manufacturer class

Data Representation using PBS and the Choice of Parameters

Each dataset is smoothed by *piecewise band smoohting* (PBS) based on the smoothing criteria in (3.8) with $d_1 = 60$ and $d_2 = 70$, and using the adaptive bandwidth and the change ratio $R = 3.5$. Adaptive bandwidth is the standard deviation of the residuals of the linear line in the previous trend regime. For the first regime, the standard deviation of the residuals of linear line fitted by the initial w points is the adaptive bandwidth. For initial window size w and bandwidth multiplier k, we chose $w \in S$ and $k \in K$, where $S = \{w \mid 4 \leq w \leq 15,\ w \text{ is an integer}\}$ and $K = \{0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$, that yield the smallest sum of the squared error in (3.8). In summary, each stock price data is smoothed using parameters initial window size w, $k\times$(adaptive bandwidth), and change ratio $R = 3.5$ under the constraint $60 \leq d \leq 70$ where d is the number of changepoints in the smoothed data.

Similarity Measure for 1-Nearest Neighbor Classification (1-NN)

We classify times series data using 1-NN and the *transformed step function* (TSF) distance in (4.10) is used as the similarity measure between two reduced datasets (the sequence of slopes). This method is applied to the raw data and normalized data. To compare performance of our smoothing methods and similarity measure, we perform classification using Euclidean distance on the normalized data to remove the effect of amplitude scale.

Results of 1-NN Classification

The results of the classification for three cases are shown in Table 5.1, Table 5.2, and Table 5.3. All methods seem to perform fairly well, particularly it is noticeable that the TSF distance for normalized data outperforms with accuracy 1. Using the raw data with the same method, PBS smoothing and TSF distance works less well than when using normalized data because the distance measurements are highly influenced by the amplitude scale of data. These results suggest that standardization of data is necessary to classify data merely based on the similarity of patterns excluding the effect of their units.

Table 5.1: The result of 1- NN classification for raw data by Euclidean Distance

		Predicted				row
		U	**F**	**M**	**D**	total
Actual	**U**	7	0	0	0	7
	F	0	4	3	0	7
	M	0	0	7	0	7
	D	0	2	0	4	6
		total				**27**
Error Rate = 0.19						

Table 5.2: The result of 1-NN classification for raw data by PBS with TSF distance

		Predicted				row
		U	**F**	**M**	**D**	total
Actual	**U**	6	0	1	0	7
	F	1	3	3	0	7
	M	1	0	6	0	7
	D	0	0	1	5	6
		total				**27**
Error Rate = 0.26						

Table 5.3: The result of 1-NN classification for normalized data by PBS with TSF distance

		Predicted				row
		U	**F**	**M**	**D**	total
Actual	**U**	7	0	0	0	7
	F	0	7	0	0	7
	M	0	0	7	0	7
	D	0	0	0	6	6
		total				**27**
Error Rate = 0.00						

5.2 Experimental Evaluations

Although the examples of clustering and classification in the previous section yielded results that are consistent with our knowledge and understanding of the datasets chosen, it is worthwhile to evaluate their performance by comparing other methods with objective measurements. Practically, it is impossible to compare all the methods or algorithms for time series data mining. Nevertheless, we evaluated our methods for classification, clustering, and segmentation against benchmarks from several papers for time series data mining ([36] [35] [14]).

5.2.1 Evaluation: Similarity Measure for Classification and Clustering

Classification and clustering problems are closely connected with the similarity measures used. If the similarity measures fail to capture the characteristics of the original time series, the classification and clustering results are meaningless. As discussed in Chapter 2, there have been many similarity measures and methods for data representation suggested, there is no one superior measure of method. Rather, because of the unique characteristic and variety in types of time series, some methods perform better for some specific types of time series while others do better for others. Therefore, we performed classification using TSF distance in (4.10) for two publicly available datasets [4] [3] and compared the result with 12 other methods.

Datasets for Benchmarks

- **Cylinder-Bell-Funnel:** This dataset contains 30 and 900 observations in the training set and test set respectively. Each observation belongs to one of three classes, *cylinder, Bell* or *Funnel*, based on its shape (Figure 5.12). The length of each individual time series is 128. This dataset is available in the UCR Time series data archive [4].

- **Synthetic Control:** This dataset contains 600 observations with length 60. There are 6 classes as seen in Figure 5.13. This dataset is available in the UCI Data Archive

[3].

1-Nearest Neighbor Classification

We perform 1-Nearest Neighbor (1-NN) classification, evaluated using *leaving-one-out* cross validation and compared 12 other methods for similarity measure including Euclidean distance as Keogh (2003, 2008) suggested. Before using TSF distance, we represented data by piecewise band smoothing (PBS).

Evaluation Results

The results of classification for *Cylinder-Bell-Funnel* (CBF) datasets and *synthetic control* dataset are summarized in Table 5.4 and Table 5.6. It is noticeable that TSF distance performs poorly, particularly for "flat" patterns such as "cylinder" and "normal" patterns. This result is somewhat natural because piecewise band smoothing and TSF distance relies on the change of slopes of piecewise linear lines, "flat" patterns are not captured as often as "up" of "down" patterns. Therefore, its performance for datasets without "cylinder" and "normal" patterns is better as seen in Table 5.5 and Table 5.7.

Sometimes PBS misses changepoints where sudden jumps or drops occur within the initial w points for current trend fitting. For this reason many observations in the "cylinder" class seem to be misclassified as a "Bell" or "Funnel" pattern. Likewise, many observations with "normal" patterns in the synthetic control dataset are misclassified as other patterns.

In *Bell-Funnel* pattern classification in Table 5.5, the accuracy for the "Bell" pattern is only 0.52 while it classifies "funnel" patterns perfectly. This is reasonable for the following two reasons: (1) there can be higher probability that the initial w for current trend fitting covers the sudden dropping point as PBS moves farther from the first time index, and (2) the length of each observation is not long enough for PBS to distinguish overall patterns thus it becomes sensitive to noises from the main trend lines. Among various types of synthetic datasets, TSF distance seems to perform *best* for "cyclic" data based on the results in Table 5.6 and Table 5.7.

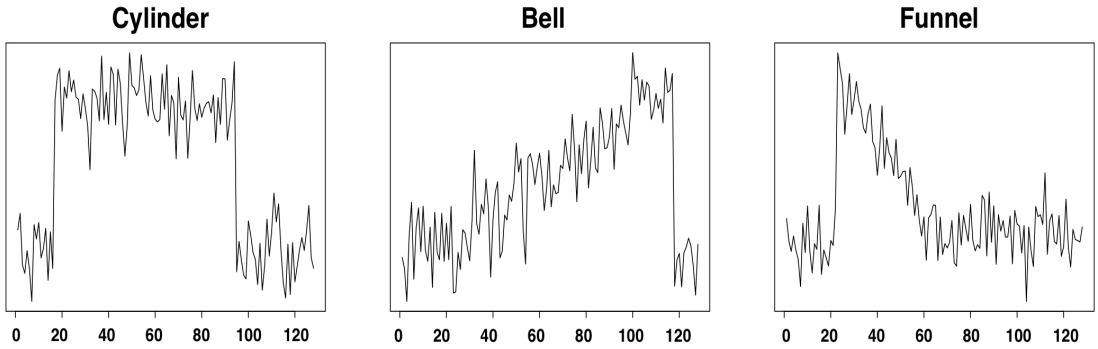

Figure 5.12: Cylinder-Bell-Funnel datasets

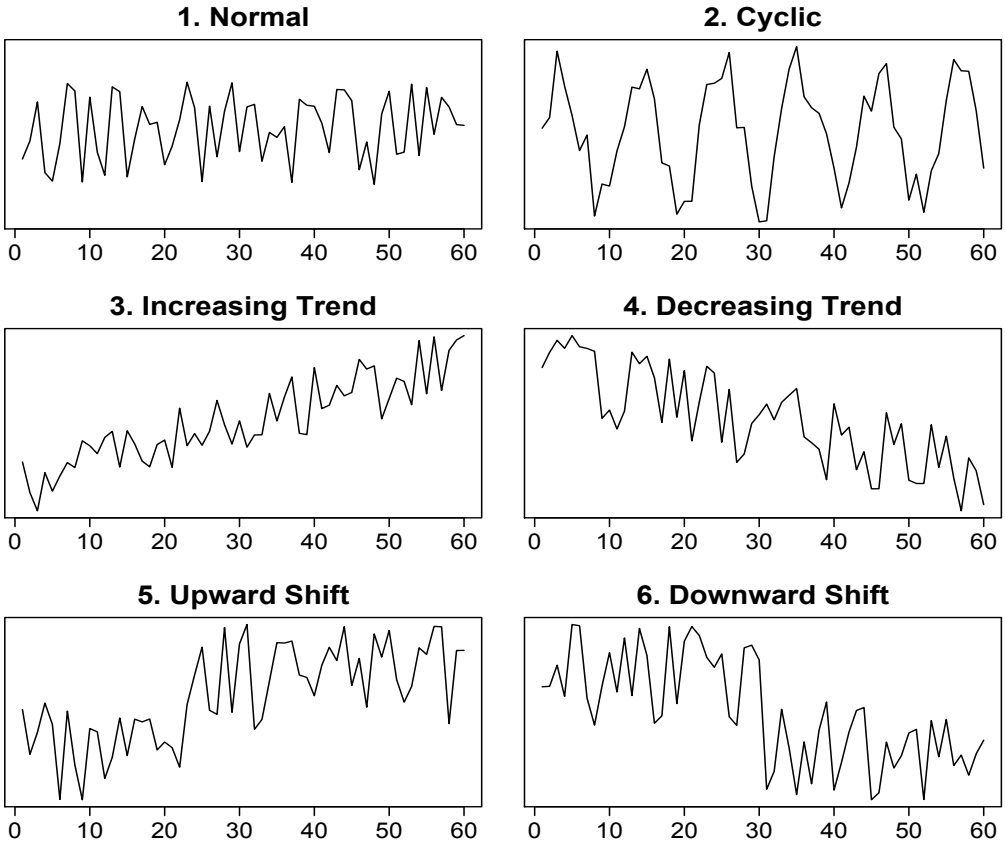

Figure 5.13: Synthetic control datasets

Table 5.4: Classification result for Cylinder-Bell-Funnel datasets using TSF distance

		Predicted			row
		Cylinder	**Bell**	**Funnel**	total
Actual	**Cylinder**	124	55	121	300
	Bell	131	126	41	300
	Funnel	80	2	220	300
		total			**900**
Error rate = 0.477					

Table 5.5: Classification result for Bell-Funnel datasets using TSF distance

		Predicted		row
		Bell	**Funnel**	total
Actual	**Bell**	157	143	300
	Funnel	0	300	300
		total		**600**
Error rate = 0.238				

Table 5.6: Classification result for Synthetic Control datasets using TSF distance

		Predicted						row
		C1	**C2**	**C3**	**C4**	**C5**	**C6**	total
Actual	**C1**	27	3	18	14	27	11	100
	C2	3	93	2	0	1	1	100
	C3	2	0	63	1	31	3	100
	C4	0	0	1	81	2	16	100
	C5	1	0	24	1	73	1	100
	C6	3	0	1	21	1	74	100
		total						**600**
Error rate = 0.315								
C1: Normal **C2:** Cyclic **C3:** Increasing **C4:** Decreasing **C5:** Downward Shift **C6:** Upward Shift								

Table 5.7: Classification result for Synthetic Control datasets without *Normal* patterns using TSF distance

		Predicted					row
		C2	**C3**	**C4**	**C5**	**C6**	total
Actual	**C2**	96	2	0	1	1	100
	C3	0	64	1	32	3	100
	C4	0	1	81	2	16	100
	C5	0	25	1	73	1	100
	C6	0	1	23	1	75	100
		total					**500**
Error rate = 0.222							
C2: Cyclic **C3:** Increasing **C4:** Decreasing **C5:** Downward Shift **C6:** Upward Shift							

Table 5.8: The error rate of various similarity measures for *Cylinder-Bell-Funnel* and *Synthetic Control* datasets [36]

Distance measure	**Cylinder-Bell-Funnel**	**Synthetic Control**
Euclidean Distance	0.003	0.013
Aligned Subsequence **(M1)** [45]	0.451	0.623
Piecewise Normalization **(M2)** [28]	0.130	0.321
Autocorrelation Functions **(M3)** [59]	0.380	0.116
Cepstrum **(M4)** [31]	0.570	0.458
String **(M5)** [26]	0.206	0.578
Important Points **(M6)** [47]	0.387	0.478
Edit Distance **(M7)** [11]	0.603	0.622
String Signature **(M8)** [7]	0.444	0.695
Cosine Wavelet **(M9)** [27]	0.130	0.371
Hölder **(M10)** [56]	0.331	0.593
Piecewise Probabilistic **(M11)** [39]	0.202	0.321
Trend step function (TSF)	**0.477**	**0.315**

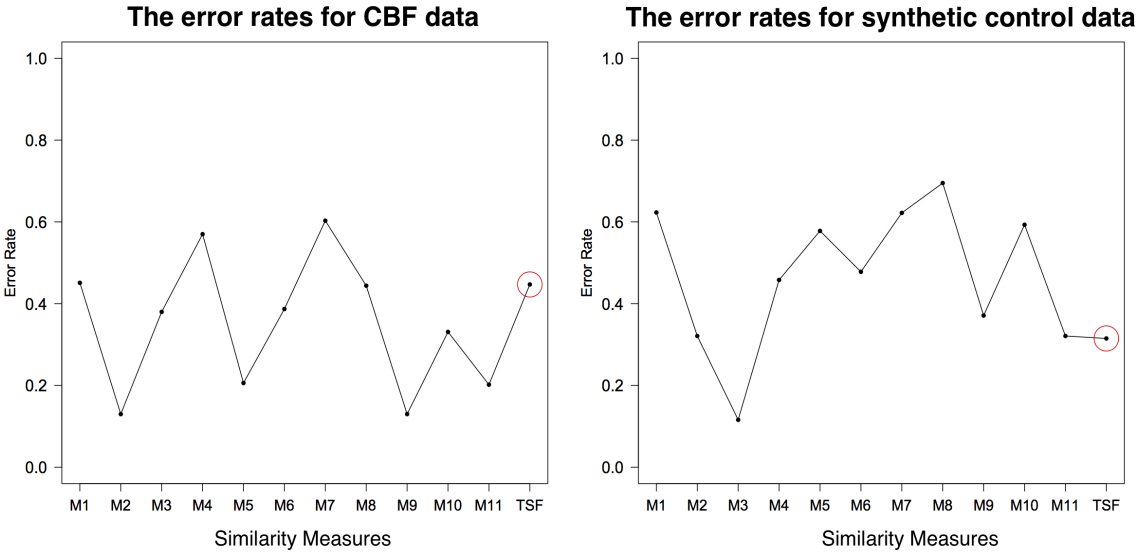

Figure 5.14: The error rate of various similarity measures for *Cylinder-Bell-Funnel* and *Synthetic Control* data

5.2.2 Evaluation: Identification of Changepoints (Segmentation)

Piecewise linear or polynomial representation methods are usually introduced for segmentation or used as a subroutine [36]. When these representation methods aim to segment large size time series, often segmentation is performed by detecting changepoints. Most segmentation algorithms can be grouped in one of three types of algorithms; (1) *sliding windows*, (2) *bottom-up*, or (3) *top-down*. To evaluate the quality of a segmentation, we perform segmentation for a fixed number of regimes using PBS and other segmentation algorithms, and then obtain the sum of squared errors (SSE) for all algorithms. These SSEs were compared to evaluate how well the algorithm detects changepoints. The lower SSE indicates the better an algorithm performs for segmentation.

Datasets for Benchmarks

We use six different types of time series data as seen in Figure 5.15. The length of each dataset is 1,024. These datasets are publicly available [4].

Algorithms for Segmentation

PBS belongs to the *sliding windows* algorithm. We use *top-down* and *bottom-up* algorithms for piecewise linear approximation shown in Keogh et al. (2004) [35], and *perceptually important points* (PIP) (bottom-up). Each time series is segmented to 64 pieces by all algorithms and the sum of squared errors are compared to evaluate performance.

Evaluation Results

Table 5.9 shows the sum of squared errors by four different segmentation algorithms. PBS does not perform well for "flat" pattern datasets such as *Balloon* datasets in this benchmark. We can conclude that PBS performs best for "PH data" data because SSEs of segmented data by PBS and by bottom-up algorithm are not significantly different although the bottom-up algorithm performs slightly better. Overall, PBS seems to perform fairly well particularly for datasets with "cyclic" or "up/down" patterns, *Soil Temperature*, *Darwin*,

PH data, and *Winding*, except datasets with "flat" patterns.

From the results, it can be concluded that there is no superior segmentation algorithm over others like there was for the similarity measure benchmark, but rather each algorithm has its own advantages for specific types of time series.

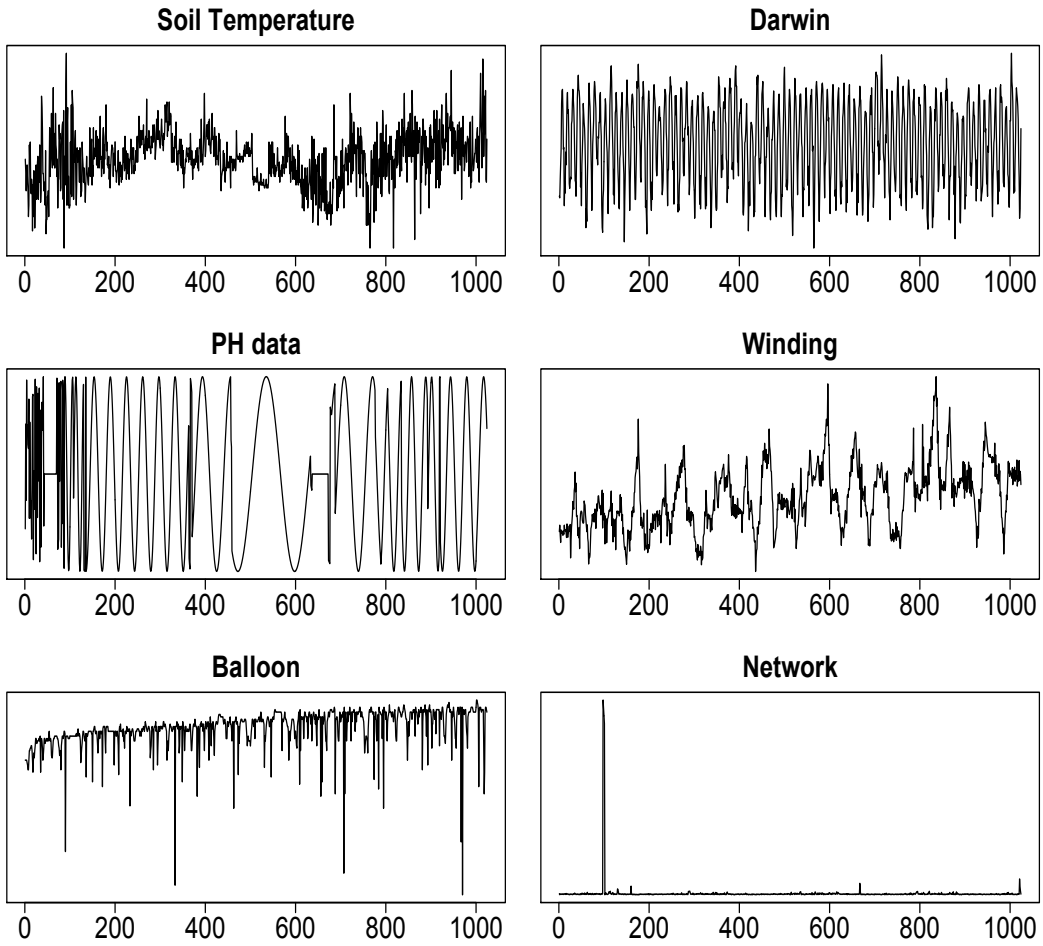

Figure 5.15: Datasets used for segmentation evaluation

Table 5.9: Sum of squared errors of approximated data by various algorithms for segmentation

Dataset	**PIP (TD)**	**PLA (TD)**	**PLA (BU)**	**PBS (SW)**
Soil Temperature	1307.66	536.99	559.79	656.33
Darwin	9364.29	6064.15	9122.203	6463.66
PH data	848.34	981.15	347.644	397.12
Winding	152.09	266.313	83.67	143.58
Balloon	14.45	18.60	18.64	21.6
Network	3598.98	836184.4	115148.1	302426.2
TD: top-down	**BU:** bottom-up	**SW:** sliding-windows		

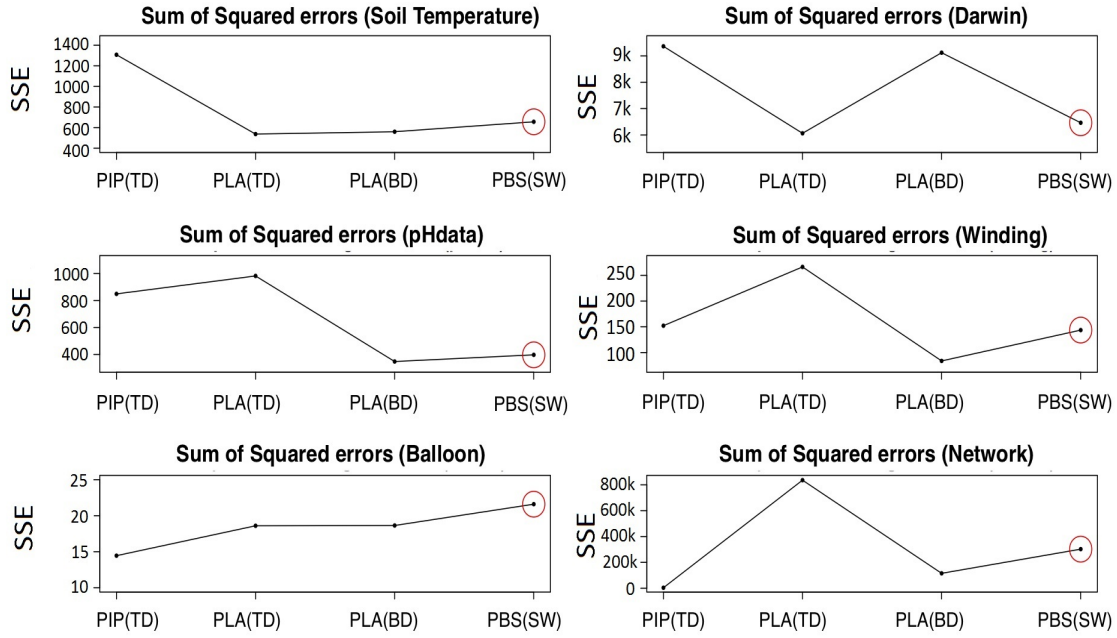

Figure 5.16: The sum of squared errors for approximated data by various segmentation algorithms

Chapter 6: Conclusions and Future Work

In this dissertation, we researched various problems in smoothing time series, including piecewise smoothing, pattern recognition, classification, and clustering. An important consideration in modeling time series is detection of points in time at which the underlying data generating process changes ("*changepoints*"). Detection of changepoints is inextricably associated with the process of building models or smoothing of the time series on either side of the changepoints. Following identification of changepoints at smoothing, we can continue the analysis of the time series data by identifying patterns in the data, and then further by associating a given time series with other time series exhibiting similar patterns.

Contributions

Contributions made in this research are as follows:

- We developed and studied two new methods of smoothing and identification of changepoints in time series. They are piecewise linear smoothing methods, *alternating trends smoothing* (ATS) and *piecewise band smoothing* (PBS). The represented data by ATS and PBS is a sequence of linear trends.

- Assuming the time series is a sequence of realizations of a random variable, the identified changepoint is a discrete random variable. We determined the probabilistic distribution of the random variable defined as the difference between two successive changepoints (the length of the trend regime) under various scenarios for the underlying data generating process.

- We suggested defining patterns: the sequence of numerical patterns and discrete patterns. These patterns contain the trend and time information.

Future Work

- In this research, we assumed that a large time series dataset consists of many data generating processes, *trend regimes*, and the observations are independent within the regime. However, in practice, observations in financial time series data more likely correlated. Moreover, the variance of data, or *volatility*, evolves rather than stays constant or stable over time. Thus, the models can be modified to be suitable for practical data in the future.

- In piecewise band smoothing (PBS), we used only w initial points to fit the current trend line. Obviously, $w > 3$ might not be enough points to fit the current trend as discussed in Section 4.3.2. One possible approach is updating the fitted trend line as we add more data points after initial w points. This may complement the drawback that the current trend overly depends on w points except when a trend changes gradually (Figure 6.1).

- The length of trends cannot be less than w in represented data since PBS does not look back at those initial w points for fitting the current trend to identify potential changepoints. This drawback may cause missing identification of *real* changepoints that might be included in w points. This issue can be resolved by modification of determining a changepoint step in the algorithm.

- The adaptive bandwidth can be determined in various ways rather than just depending on the previous regime. For example, all residuals in the past k previous regimes can be used. Or w residuals in the current regime also can be used with the residuals in the previous regimes.

- Models for multivariate time series data can be researched.

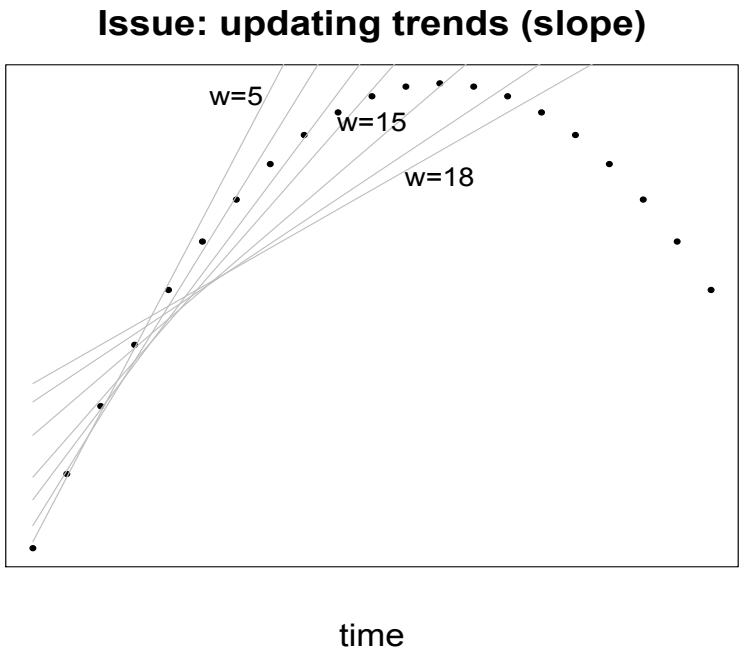

Figure 6.1: Gradually changing trend

Appendix A: An Appendix

Table A.1: Ticker symbols of stocks (clustering using ATS)

Airline Industry	
ALK	Alaska Air Group, Inc.
AAL	American Airlines Group Inc.
DAL	Delta Air Lines, Inc.
DLAKY	Deutsche Lufthansa Aktiengesellschaft
HA	Hawaiian Holdings Inc.
JBLU	JetBlue Airways Corporation
LUV	Southwest Airlines Co.
CEA	China Eastern Airlines Corp. Ltd.
ZNH	China Southern Airlines Co. Ltd.
AFLYY	Air France-KLM SA
SKYW	SkyWest Inc.
Restaurant Industry	
BJRI	BJ's Restaurants, Inc.
CAKE	The Cheesecake Factory Incorporated
CHUY	Chuy's Holdings, Inc.
DRI	Darden Restaurants, Inc.
DPZ	Domino's Pizza, Inc.
DNKN	Dunkin' Brands Group, Inc.
DAVE	Famous Dave's of America Inc.
MCD	McDonald's Corp.
NDLS	Noodles & Company
PZZA	Papa John's International Inc.
SBUX	Starbucks Corporation
TXRH	Texas Roadhouse, Inc.

Table A.2: Ticker symbols of stocks (classification using PBS)

Sector (**U**): Utilities - Electricity	
EDE	The Empire District Electric Company
PCG	PG & E Corporation
GXP	Great Plains Energy Incorporated
ETR	Entergy Corporation
AEE	Ameren Corporation
LNT	Alliant Energy Corporation
AEP	American Electric Power Co., Inc.
Sector (**F**): Finance Insurance	
CNA	CNA Financial Corporation
AXS	The Cheesecake Factory Incorporated
TRV	The Travelers Companies, Inc.
ALL	The Allstate Corporation
HIG	The Hartford Financial Services Group, Inc.
SLF	Sun Life Financial Inc.
PGR	Progressive Corp.
Sector (**M**): Investment Mortgage & Bank	
GS	The Goldman Sachs Group, Inc.
SCHW	The Charles Schwab Corporation
AMTD	TD Ameritrade Holding Corporation
MS	Morgan Stanley
JMP	JMP Group LLC
ETFC	E∗TRADE Financial Corporation
JPM	JPMorgan Chase & Co.
Sector (**D**): Drug Manufacturer	
PFE	Pfizer Inc.
NVS	Novartis AG
MRK	Merck & Co. Inc.
JNJ	Johnson & Johnson
LLY	Eli Lilly and Company
BMY	Bristol-Myers Squibb Company

Bibliography

[1] NYSE daily stock price data. `http://finance.yahoo.com`, accessed April, 2016.

[2] Patterns in technical analysis. `http://www.forexblog.org`, accessed April, 2016.

[3] UCI machine learning repository. `http://archive.ics.uci.edu/ml/datasets.html`, accessed April, 2016.

[4] UCR time series data archive. `http://www.cs.ucr.edu/~eamonn/time_series_data`, accessed April, 2016.

[5] Aggarwal, C. C., and Reddy, C. K. *Data clustering: algorithms and applications*. CRC Press, 2013.

[6] Agrawal, R., Faloutsos, C., and Swami, A. *Efficient similarity search in sequence databases*. Springer, 1993.

[7] André-Jönsson, H., and Badal, D. Z. Using signature files for querying time-series data. In *Principles of Data Mining and Knowledge Discovery*. Springer, 1997, pp. 211–220.

[8] Astrom, K. J. On the choice of sampling rates in parametric identification of time series. *Information Sciences 1*, 3 (1969), pp. 273–278.

[9] Bachelier, L. *Louis Bachelier's theory of speculation: the origins of modern finance*. Princeton University Press, 2011.

[10] Bao, D. A generalized model for financial time series representation and prediction. *Applied Intelligence 29*, 1 (2008), pp. 1–11.

[11] Bozkaya, T., Yazdani, N., and Özsoyoğlu, M. Matching and indexing sequences of different lengths. In *Proceedings of the sixth international conference on Information and knowledge management* (1997), ACM, pp. 128–135.

[12] Chung, F.-L., Fu, T.-C., Luk, R., and Ng, V. Flexible time series pattern matching based on perceptually important points. In *International joint conference on artificial intelligence workshop on learning from temporal and spatial data* (2001), pp. 1–7.

[13] De Boor, C. *A practical guide to splines*, vol. 27. Springer-Verlag New York, 1978.

[14] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment 1*, 2 (2008), pp. 1542–1552.

[15] Eisenhauer, J. G. Regression through the origin. *Teaching Statistics 25*, 3 (2003), pp. 76–80.

[16] Esling, P., and Agon, C. Time-series data mining. *ACM Computing Surveys (CSUR) 45*, 1 (2012), pp. 12.

[17] Fink, E., Pratt, K. B., and Gandhi, H. S. Indexing of time series by major minima and maxima. In *SMC* (2003), IEEE, pp. 2332–2335.

[18] Fu, T.-c. A review on time series data mining. *Engineering Applications of Artificial Intelligence 24*, 1 (2011), pp. 164–181.

[19] Fu, T.-c., Chung, F.-l., Luk, R., and Ng, C.-m. Representing financial time series based on data point importance. *Engineering Applications of Artificial Intelligence 21*, 2 (2008), pp. 277–300.

[20] Gentle, J. E. Mining for patterns in financial time series. In *JSM 2012 Proceedings* (Alexandria, VA, 2012), American Statistical Association, pp. 2978–2988.

[21] Gentle, Härdle, L. S. *Handbook of Big Data Analytics*. Springer International Publishing, 2017.

[22] Golay, X., Kollias, S., Stoll, G., Meier, D., Valavanis, A., and Boesiger, P. A new correlation-based fuzzy logic clustering algorithm for fmri. *Magnetic Resonance in Medicine 40*, 2 (1998), pp. 249–260.

[23] Goodman, J. *Principles of Scientific Computing*. `http://www.cns.nyu.edu/~fan/docs/GoodmanBook.pdf`, 2006.

[24] Guo, X., Liang, X., and Li, X. A stock pattern recognition algorithm based on neural networks. In *Natural Computation, 2007. ICNC 2007. Third International Conference on* (2007), vol. 2, IEEE, pp. 518–522.

[25] Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.

[26] Huang, Y.-W., and Yu, P. S. Adaptive query processing for time-series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining* (1999), ACM, pp. 282–286.

[27] Huhtala, Y., Karkkainen, J., and Toivonen, H. T. Mining for similarities in aligned time series using wavelets. In *AeroSense'99* (1999), International Society for Optics and Photonics, pp. 150–160.

[28] Indyk, P., Koudas, N., and Muthukrishnan, S. Identifying representative trends in massive time series data sets using sketches. In *VLDB* (2000), pp. 363–372.

[29] James, E. Gentle. elements of computational statistics. *Statistics and Computing. Springer* (2002).

[30] James, G., Witten, D., Hastie, T., and Tibshirani, R. *An introduction to statistical learning*, vol. 112. Springer, 2013.

[31] Kalpakis, K., Gada, D., and Puttagunta, V. Distance measures for effective clustering of arima time-series. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on* (2001), IEEE, pp. 273–280.

[32] Keogh, E. Fast similarity search in the presence of longitudinal scaling in time series databases. In *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on* (1997), IEEE, pp. 578–584.

[33] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems 3*, 3 (2001), pp. 263–286.

[34] Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. Locally adaptive dimensionality reduction for indexing large time series databases. *ACM SIGMOD Record 30*, 2 (2001), pp. 151–162.

[35] Keogh, E., Chu, S., Hart, D., and Pazzani, M. Segmenting time series: A survey and novel approach. *Data mining in time series databases 57* (2004), pp. 1–22.

[36] Keogh, E., and Kasetty, S. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery 7*, 4 (2003), pp. 349–371.

[37] Keogh, E. J., and Pazzani, M. J. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *KDD* (1998), vol. 98, pp. 239–243.

[38] Keogh, E. J., and Pazzani, M. J. Relevance feedback retrieval of time series data. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (1999), ACM, pp. 183–190.

[39] Keogh, E. J., and Smyth, P. A probabilistic approach to fast pattern matching in time series databases. In *KDD* (1997), vol. 1997, pp. 24–30.

[40] Lee, S., Kwon, D., and Lee, S. Dimensionality reduction for indexing time series based on the minimum distance. *Journal of Information Science and Engineering 19* (2003), pp. 697–711.

[41] Liao, T. W. Clustering of time series dataa survey. *Pattern recognition 38*, 11 (2005), pp. 1857–1874.

[42] Lin, J., Keogh, E., Lonardi, S., and Chiu, B. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* (2003), ACM, pp. 2–11.

[43] Lin, J., Keogh, E., Wei, L., and Lonardi, S. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery 15*, 2 (2007), pp. 107–144.

[44] Müller, M. Dynamic time warping. *Information retrieval for music and motion* (2007), pp. 69–84.

[45] Park, S., Kim, S.-W., and Chu, W. W. Segment-based approach for subsequence searches in sequence databases. In *Proceedings of the 2001 ACM symposium on Applied computing* (2001), ACM, pp. 248–252.

[46] Pathak, A. Predictive time series analysis of stock prices using neural network classifier. *International Journal of Computer Science and Engineering Technology* (2014), pp. 2229–3345.

[47] Pratt, K. B., and Fink, E. Search for patterns in compressed time series. *Int. J. Image Graphics 2*, 1 (2002), pp. 89–106.

[48] Quah, T.-S. Djia stock selection assisted by neural network. *Expert Syst. Appl. 35*, 1-2 (July 2008), pp. 50–58.

[49] Ratanamahatana, C., Keogh, E., Bagnall, A. J., and Lonardi, S. A novel bit level time series representation with implication of similarity search and clustering. In *Advances in knowledge discovery and data mining*. Springer, 2005, pp. 771–777.

[50] Ratanamahatana, C. A., Lin, J., Gunopulos, D., Keogh, E., Vlachos, M., and Das, G. Mining time series data. In *Data mining and knowledge discovery handbook*. Springer, 2009, pp. 1049–1077.

[51] Ruppert, D. Selecting the number of knots for penalized splines. *Journal of computational and graphical statistics 11*, 4 (2002), pp. 735–757.

[52] Shatkay, H., and Zdonik, S. B. Approximate queries and representations for large data sequences. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on* (1996), IEEE, pp. 536–545.

[53] Shelly, G., and Rosenblatt, H. J. *Systems analysis and design*. Cengage Learning, 2011.

[54] Shimodaira, H., et al. Approximately unbiased tests of regions using multistep-multiscale bootstrap resampling. *The Annals of Statistics 32*, 6 (2004), pp. 2616–2641.

[55] Smyth, P., and Keogh, E. Clustering and mode classification of engineering time series data. In *Proc. of the 3rd Int. l Conf. on KDD* (1997), pp. 24–30.

[56] Struzik, Z. R., and Siebes, A. The haar wavelet transform in the time series similarity paradigm. In *Principles of Data Mining and Knowledge Discovery*. Springer, 1999, pp. 12–22.

[57] Theil, H. *Principles of econometrics*. J. Wiley and sons, New York, London, Sydney, 1971.

[58] Tsay, R. S. *Analysis of financial time series*, vol. 543. John Wiley & Sons, 2005.

[59] Wang, C., and Sean Wang, X. Supporting content-based searches on time series via approximation. In *Scientific and Statistical Database Management, 2000. Proceedings. 12th International Conference on* (2000), IEEE, pp. 69–81.

[60] Yang, O., Jia, W., Zhou, P., and Meng, X. A new approach to transforming time series into symbolic sequences. In *Proceedings of the First Joint Conference between the Biomedical Engineering Society and Engineers in Medicine and Biology* (1999), vol. 974.

[61] Yang, Z., and Zhao, G. Application of symbolic techniques in detecting determinism in time series. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (1998), vol. 20, pp. 2670–2673.

[62] Yi, B.-K., and Faloutsos, C. Fast time sequence indexing for arbitrary L_p norms. VLDB.

Biography

Seunghye Jung Wilson was born and grew up in Seoul, South Korea. She graduated Younsei University, Seoul in 2003 with B.S. in Atmospheric Science and received M.S in statistics from Rutgers University, New Brunswick in 2010.