Features Extraction from Time Series

A thesis submitted to the University of Sheffield for the degree of Doctor of Philosophy

Xinxin Yao

Department of Automatic Control and Systems Engineering

February 2019

I would like to dedicate this thesis to my family and friends

Declaration

I hereby declare that this submitted thesis is my research work under the guidance of my tutor. Additionally, this dissertation, entitled "**Features Extraction from Time Series**", does not contain research results that have been published by other individuals or groups, or results that have been previously applied for degrees or used for other purposes.

Xinxin Yao February 2019

Acknowledgements

I would like to express my gratitude to my tutor, Dr. Hua-Liang Wei, who offered many suggestions during my research, from the selection of my project to the cogitation of research problems and the conclusion of my dissertation. Without his support and help, I could not have accomplished these tasks throughout my PhD journey.

I am also thankful to my friends, for being like my family when I was far away from home: Mr Jackey Wong, Mr Xi Chen, Dr Chao Liu, Mr Yuanlin Gu, Mr Qifeng Liu and Mr Yunpeng Li.

Finally, I want to thank my parents and my wife. I want to offer deepest appreciation to my parents, who supported me, taught me, and encouraged me during this process. I am so blessed to have such a warm family. Without my parent's support, I would not have been able finish my PhD. I also appreciate my wife's patience and kindness, as she constantly encouraged me when I was in encountered problems and stayed with me during my PhD journey.

Abstract

Time series can be found in various domains like medicine, engineering, and finance. Generally speaking, a time series is a sequence of data that represents recorded values of a phenomenon over time. This thesis studies time series mining, including transformation and distance measure, anomaly or anomalies detection, clustering and remaining useful life estimation.

In the course of the first mining task (transformation and distance measure), in order to increase the accuracy of distance measure between transformed series (symbolic series), we introduce a novel calculation of distance between symbols. By integrating this newly defined method to symbolic aggregate approximation and its extensions, the experimental results show this proposed method is promising.

During the process of the second mining task (anomaly or anomalies detection), for the purpose of improving the accuracy of anomaly or anomalies detection, we propose a distance measure method and an anomalies detection calculation. These proposed methods, together with previous published anomaly detection methods, are applied to real ECG data selected from MIT-BIH database. The experimental results show that our proposed outperforms other methods.

During the course of the third mining task (clustering), we present an automatic clustering method, called AT-means, which can automatically carry out clustering for a given time series dataset: from the calculation of global average time series to the setting of initial centres and the determination of the number of clusters. The performance of the proposed method was tested on 10 benchmark time series datasets obtained from UCR database. For comparison, the K-means method with three different conditions are also applied to the same datasets. The experimental results show the proposed method outperforms the compared K-means approaches.

During the process of the fourth mining task (remaining useful life estimation), all the original data are transformed into low-dimensional space through principal components analysis. We then proposed a novel multidimensional time series distance measure method, called as multivariate time series warping distance (MTWD), for remaining useful life estimation. This whole process is tested on the CMAPSS

(Commercial Modular Aero Propulsion System Simulation) datasets and the performance is compared with two existing methods. The experimental results show that the estimated remaining useful life (RUL) values are closer to real RUL values when compared with the comparison methods.

Our work contributes to the time series mining by introducing novel approaches to distance measure, anomalies detection, clustering and RUL estimation. We furthermore apply our proposed methods and related methods to benchmark datasets. The experimental results show that our methods are better than previously published methods in terms of accuracy and efficiency.

Table of Contents

List of Algorithms

List of Figures

List of Tables

Nomenclature

AID	Average distance Initial centre Determination
AT-means	Automatic Time series clustering
ANMD	Average Non-self Match Distance
ARI	Adjusted Rand Index
AWDD	Adaptive Window based Discord Discovery
BFDD	Brute Force Discord Discovery
CBF	Cylinder Bell and Funnel
DBA	Dynamic time warping Barycentre Averaging
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DTW	Dynamic Time Warping
DWAD	Dual Weight Average Distance
ECG	ElectroCardioGram
ESAX	Extended Symbolic Aggregate Approximation
HDD	Heuristic Discord Discovery
IWGTA	Initialized Weight Global Time series Averaging
MDTW	Modified Dynamic Time Warping
MIT-BIH	Massachusetts Institute of Technology-Beth Israel Hospital
MTWD	Multivariate Time series Warping Distance
NLAAF	Nonlinear Alignment and Averaging Filters
PAA	Piecewise Aggregate Approximation
PCA	Principal Components Analysis
PHM	Prognostic and Health Management
PSA	Prioritized Shape Averaging
ROCK	RObust Clustering using linKs
RUL	Remaining Useful Life
SAX	Symbolic Aggregate Approximation
SAX-TD	Symbolic Aggregate Approximation Trend Distance
SVM	Support Vector Machine
TLB	Tightness Lower Bound

Chapter 1

Introduction

In this chapter, background and motivations of my research are first provided. Then methods previously investigated during my PhD are described. Finally, the outline of this thesis is presented.

1.1 Background and Motivations

"We are living in the information age" is a popular saying, but we are actually living in the data age. Over millions of gigabytes of data pour into our data storage devices every day from finance, science, engineering, medicine, and almost every other aspect of daily life (Han et al 2011). Time is a dimension and measure in which events can be ordered from the past through the present into the future, and also the measure of durations of events and the intervals between them. As time goes by and thanks to the development technologies (e.g. sensor techniques and massive storage techniques), almost all types of data can be stored as time series including a DNA sequence, a video and changes of stock prices.

With this massive data explosion, we often try to make good use of them to discover the most important patterns. These can not only help us find the relationship between different things, but also offer us important evidence to make the right decision. Data mining is such a technique, which is an activity to extract some new information contained in large database. In general, the goal of data mining is to use a method or combination of data techniques to discover hidden patterns, unexpected trends, or other subtle relationships (Sumathi and Sivanandam 2006). Today, this new discipline is widely used in business, science and engineering. For example, personal and financial information can be recorded and minded to help bank to make right decision. Data mining can also help reveal potential locations of some resources, or help establish early warning

systems for disasters like oil spills etc. The applications of data mining are very wide and will grow rapidly in the next few years.

The increasing use of data and the exponential growth of database size, especially the growth of time series, have aroused great interest in the field of data mining. In the e-commerce domain alone, large amounts of time series data as diverse as browsing histories, shopping histories, transferring histories of customers are generated and analysed. Similar works are also applied to industry, education, healthcare, entertainment and virtually every other field of human endeavor. In recent decades, various researches have been attempted in time series data mining, such as similarity measure (Baydogan and Runger 2016, Agrawal et al 1993), anomaly detection (Fujimaki et al 2005, Kumar 2005), clustering (Paparrizos and Gravano 2015, Paparrizos and Gravano 2017), prediction (Zhou et al 2016, Xiao et al 2017), and so on (Fu et al 2006, Zhang et al 2004). Although all of these techniques have long traditions, there still remain unsolved problems that spur further research.

- **Time series representation and distance measure:** How can the fundamental shapes of a time series be represented? How can the distance between two time series be computed? The representation technique should acquire the characteristics of shape by decreasing the dimensionality and preserving the useful information of data. The similarity measure method should have the ability to distinguish any pair of different time series.
- **Anomaly detection of time series:** How can the anomalous parts of a time series be defined? How can the anomalies in one time series be extracted? The anomaly detection algorithm should be able to tell the differences between normal time series and anomalous time series, thus quickly and accurately find the anomalous parts in a time series.
- **Clustering of time series:** How can the number of clusters of a time series dataset be determined? How can the average time series of a set of time series be calculated? How can clusters of time series be generated? An automatic time series clustering algorithm should properly compute the number of clusters in a time series dataset, calculate the average time series of every cluster to represent the characteristics of corresponding cluster, and organize similar time series into related groups according to the distance between testing time series and centre time series.

- **Prognostics and health manage of time series:** How can the health status of a degradation pattern be determined? How can the RUL of equipment be predicted? The health management and prognostic algorithm should find historical patterns that are similar to the testing pattern, and estimate remaining useful life of testing pattern by using real life of historical patterns.

1.2 Methods Investigated in 4 Years

Many approaches for time series data mining tasks depend on pairwise (dis)similarity comparisons of (sub)sequences by means of distance measure (Ding et al 2008, Esling and Agon 2012). Hence my PhD started with the research of time series distance measure. Then anomaly detection of time series was investigated. Afterwards, time series clustering was studied. In the last section of my research, RUL estimation of time series was considered.

1.2.1 Time Series Distance Measure

Given the fact that there is such a large number of complex data in time series, it will inevitably lead to large expenditure of time and funds if we want to directly analyze time series. It is probably worse that the final result may not be accurate or robust. In recent decades, in order to reduce the dimensionality of raw time series while retaining its essential characteristics, many time series representation methods have been proposed. Among these approaches, symbolic aggregate approximation is one of the famous methods with dimensionality reduction, symbolic representation, and distance measure. For distance measure between symbolic series, distances between symbols are defined according to Gaussian distribution and the distance between symbolic series is computed by summing the distances between paired symbols. Although the definition of distances between symbols is uncomplicated, the accuracy of distance measure between symbolic series is influenced if the distances between symbols is not accurate. In addition, due to the first step of symbolic aggregate approximation (time series normalization), different time series with same normalized shape are defined as same. During our research of distance measure between time series, we introduced a calculation of distances between symbols and a distance measure method. By integrating our proposed methods to symbolic aggregate approximation and its extended methods, the performance of distance measure between time series based on symbolic representation is improved.

1.2.2 Time Series Anomaly Detection

Anomaly detection is an important issue in various fields and application domains. In recent decades, many anomaly detection techniques have been developed, some for specific domains while others are more generic (Chandola et al 2009). ECGs, as the most commonly used biological signals in medical field, are easy to collect and are typically used to determine the cardiac structure and function of patients. For cardiovascular diseases, such as myocardial and ischemia, they occur over a certain period with the heart of a patient does not work normally. Because of this, effective detection of anomalous segments in ECG data can make a significant contribution to heart diagnosis. Among the published anomaly detection techniques, brute force discord discovery and adaptive window discord discovery are used to detect anomalous segment in ECGs. For anomaly detection in ECGs, when there is only one disordered segment or several significantly different disordered segments, these two methods can correctly detect the anomalous segment(s) while adaptive window discord discovery outperforms brute force discord discovery in terms of computational efficiency. However, when there are two or more anomalous segments and the distance between anomalies lower than a small value, these two methods cannot correctly detect the anomalies. Furthermore, traditional dynamic time warping distance is used to calculate the distance between time series through directly adding up the distances between paired points. This influences the accuracy of distance measure. During the research of anomalies detection in time series, a modified dynamic time warping distance was proposed to improve the performance of time series distance measure, and anomalies detection method (non-self match average distance) was introduced to detect all anomalies. This proposed anomaly detection method (combined by the proposed distance measure method and the anomalies detection method), together with brute force discord discovery and adaptive window discord discovery, were applied to same ECG data. The experimental results show that the proposed method is promising in terms of calculation complexity and outperforms the two compared methods with regards to the accuracy of anomalies detection.

1.2.3 Automatic Time Series Clustering

The process of dividing a collection of objects into classes, in which objects are similar to each other, is called clustering. In data mining, clustering analysis has long played an important role in a wide range of fields, such as image processing

(Niennattrakul and Ratanamahatana 2007) and forecasting (Sfetsos and Siriopoulos 2004). A special type of clustering is time series clustering (Aghabozorgi et al. 2015). In recent decades, clustering of time series has received significant attention from different aspects, not only because time series clustering can discover valuable patterns from time series datasets, but also saves a lot of unnecessary work and time because the analysis of a large dataset can be achieved by analyzing a relatively smaller structured dataset with the facilitation of clustering techniques. During the research of time series clustering: 1) an initial centers sequence determination method was developed so that the initial centers are located in proper areas; 2) a modified global time series averaging method was introduced to calculate the average sequence of a cluster of time series; and 3) a novel elbow point extraction method was proposed to determine the number of clusters. The combination of these three ideas is used to automatically cluster a set of time series, called AT-means. This proposed automatic time series clustering method and three K-means approaches were applied to 10 real-life time series datasets. The comparison results showed that the proposed method outperforms the three compared K-means approaches in terms of accuracy.

1.2.4 Remaining Useful Life Estimation

Remaining useful life estimation of safety related critical components has embraced a vast number of techniques and algorithms in recent decades, not only because the health statuses of these critical components are closely linked to lives of related people, but also because the repair and maintenance of these components requires substantial resources and funding. Most existing RUL estimation methods build a physical prediction model according to the data. These physical models, which can describe the physical behaviors of a testing system, are effective when the system's degradation process can be well described. However, for complex systems, it is easier to collect data than to build physical models, and hence, a lot of data-driven prognostics haven been published in the past decades. Among the data-driven prognostic approaches, similarity-based approaches, such as RUL estimation based on similar health indicator (Wang et al 2008) and RUL prediction based on degradation shapelets extraction (Malinowski et al 2015), are relatively new but have made promising performances. During my last research, we proposed a multivariate time series distance measure method, called multivariate time series warping distance (MTWD), to properly extract degradation fragments of training

equipment that are similar to that of testing equipment, and estimate the RUL of testing equipment according to the real life of extracted training equipment. The proposed similarity measure method was applied to CMAPSS (Commercial Modular Aero Propulsion System Simulation) datasets and the performance is compared with two existing methods reported by Wang et al (2008) and Malinowski et al (2015). Results generated by the proposed method show that the estimated RUL values are closer to real RUL values when comparing the two methods.

1.3 Outline of This Thesis

This completed thesis includes seven chapters, are outlined as follows:

Chapter 2: Literature Review

In this chapter, some concepts and notions that will be relevant for my research are introduced. This literature review covers a brief introduction of time series representation and transformation, an overview of distance measure between time series, a short survey of time series anomaly detection, a summary of time series clustering, and a retrospect of time series RUL estimation.

Chapter 3: Time Series Distance Measure

In this chapter, a new definition of distances between symbols and a distance measure method are presented for distance measure between symbolic series. The maximum and minimum values in each symbolic area are used to calculate the distances between symbols. These calculated distances are used to generate a look-up table, which is then used to calculate the distance between original time series through inverse calculation of zero-normalization. The look-up table is integrated to SAX and SAX-TD. These two integrated methods and the proposed method are applied to 1000 pairs of benchmark time series. The results show that our proposed method improve the performance of previous published symbolic representation and distance measure methods.

Chapter 4: Anomaly Detection of Time Series

In this chapter, modified dynamic time warping (MDTW) and a new anomaly definition method are proposed for time series anomaly detection. The improved SAX (in Chapter 3) is used to represent a time series, the modified DTW is adapted to calculate

the distance between symbolic series, and the proposed anomaly definition method is used to detect anomalous systems from original time series. This anomaly detection method and another 2 additional famous anomaly detection methods are applied to 30 real ECGs. Experimental results show that this proposed method is promising in terms of calculation complexity and accuracy.

Chapter 5: Automatic Time Series Clustering

In this chapter, an automatic clustering method, called AT-means, is presented. AT-means can automatically carry out clustering for a given time series dataset: from the calculation of global average time series to setting of initial centers and the determination of the number of clusters. The performance of AT-means is tested on 10 benchmark time series datasets obtained from UCR (University of California Riverside) database. For comparison, K-means with three different conditions are also applied to the same datasets. The experimental results show that AT-means outperforms the compared K-means approaches.

Chapter 6: Similarity-Based Remaining Useful Life Estimation

In this chapter, a multidimensional time series similarity measure method is proposed for similarity-based RUL estimation. Principal components analysis (PCA) is applied to transform original multidimensional time series into low-dimensional time series. The proposed distance measure method is applied to extract meaningful degradation patterns from training library, and RUL of testing equipment is computed according to the real life of extracted training patterns. The proposed method is applied to aircraft engines data provided by NASA Prognostic Data Repository, and experimental results of two published similarity-based RUL estimation approaches are used for comparison. The comparison shows that the proposed method is very effective in RUL estimation.

Chapter 7: Conclusion and Future Work

In this chapter, conclusions regarding works presented in this thesis are provided, and an outlook of future work is provided.

1.4 Publications in 4 Years

The content of this thesis builds on the following publications by the author:

Published

1. (Chapter 3)

   Xinxin Yao and Hua-Liang Wei. Walking gestures recognition based on a novel symbolic representation. 22nd International Conference on Automation and Computing (ICAC), 2016.

2. (Chapter 3)

   Xinxin Yao and Hua-Liang Wei. Off-line signature verification based on a new symbolic representation and dynamic time warping. 22nd International Conference on Automation and Computing (ICAC), 2016.

3. (Chapter 5)

   Xinxin Yao and Hua-Liang Wei. Improving K-means clustering performance using a new global time-series averaging method. 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2017.

4. (Chapter 6)

   Xinxin Yao and Hua-Liang Wei. Short-term stock price forecasting based on similar historical patterns extraction. 23nd International Conference on Automation and Computing (ICAC), 2017.

Processing

5. (Chapter 3)

   Xinxin Yao and Hua-Liang Wei. A New Metric for Computing the Distance between Time Series Based on Symbolic Aggregate Approximation. Submit to Pattern Recognition Letters.

6. (Chapter 4)

   Anomaly detection of time series: an application to ECG data.

7. (Chapter 5)

   AT-means: Automatic Time Series Clustering.

8. (Chapter 6)

   A multidimensional time series similarity measure approach for similarity-based remaining useful life estimation.

Chapter 2

Literature Review

In this chapter, we present several reviews on time series mining that are related to my research. The chapter is divided into five major sections: in the first section, time series transformation and representation methods are reviewed; in the second section, popular used distance measure methods are described; in the third section, an overview of anomaly detection methods is provided; in the fourth section, time series clustering methods are reviewed; in the fifth section, a broad review of time series remaining useful life estimation is presented.

2.1 Time Series Representation and Transformation

Through applying a transformation or representation method to a time series, the obtained series has to satisfy following requirements: 1) features that contain useful information are extracted; 2) dimensionality of new series is lower than that of original series. In the last few decades, multiple approaches about time series transformation and representation have been proposed. In this section, we briefly review several approaches that relate to our research, they are piecewise aggregate approximation, symbolic aggregate approximation and principal components analysis.

2.1.1 Piecewise Aggregate Approximation

As early as 1974, Pavlidis and Horouitz proposed a method hereby the original time series can be represented by a series of segments. They also pointed out that the advantages of this method using segments to represent original signal can reduce the dimension of original signal, preserve powerful information and remove noises (Pavlidis and Horowitz 1974). Generally speaking, this kind of time series segmentation and approximation method can be called piecewise linear representation. The basic idea of

this method is using a series of head-tail segments to approximate the original database. In recent decades, with the development of computing technologies, piecewise linear representation has been employed in many applications (Jia et al 2008, Kimura et al 2008).

Piecewise Aggregate Approximation (PAA) is one of the most commonly used piecewise linear representation methods used to separate the original data into several or many segments with equal length and represent all of them by the average values of segments. In this way, PAA improves the efficiency in case where the main objective is to find the matching patterns in a large database that contains a great number of data. A simple example of such a transformation is shown in Figure 2.1, where the upper curve is the original signal and the lower one is the transformed signal. It can be noticed that the transformed series can still describe majority of features of the original time series with a suitable choice of the length of every segment. There are two special areas of concern when applying PAA: 1) when the number of segment is 1, the new series is simply the mean of original time series 2) When the number of segments is equal to N, where N is the number of values in original series, the new series is identical to the original series. Detailed discussions of PAA are provided by (Keogh et al 2001).

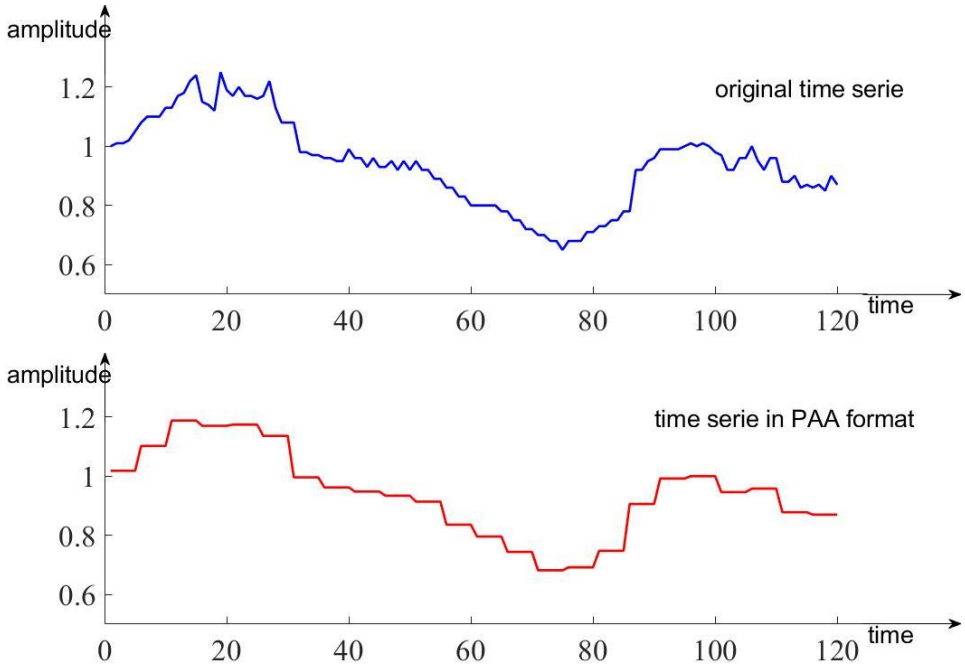


Figure 2.1 Original time series and piecewise aggregate approximation with the length of every segment is 9

Given one time series $Te = [te_1, te_2, \dots, te_i, \dots, te_{n_{Te}}]$, through the processing of

PAA, Te is transformed into $\overline{Te} = [\overline{te_1}, \overline{te_2}, \dots, \overline{te_\iota}, \dots, \overline{te_{m_{Te}}}]$. The calculation of PAA is completed by following equation:

$$\overline{\mathrm{te}_\mathrm{l}} = \frac{\mathrm{m_{Te}}}{\mathrm{n_{Te}}} \sum_{\mathrm{j}=\frac{\mathrm{n_{Te}}}{\mathrm{m_{Te}}}(\mathrm{i}-1)+1}^{\frac{\mathrm{n_{Te}}}{\mathrm{m_{Te}}}*\mathrm{i}} \mathrm{te}_\mathrm{j} \tag{2.1}$$

where i is the time point of te_i in time series Te, $\overline{te_\iota}$ represents the average value of the ith segment, n_{Te} means the number of values in time series Te, m_{Te} means the number of values in $\overline{Te}$.

PAA is efficient in dealing with time series data when the main objective is dimensionality reduction. For one time series, the transformation process of PAA is summarized by following pseudocode:

Algorithm 2.1 Piecewise Aggregate Approximation

Requirements: Input Time Series: Te
Length of Segment: l
An Empty Matrix: B
$n \leftarrow$ length of Te
for $i = 1$ to n with step l **do**
$id = i : i + l - 1$;
$t \leftarrow$ average value of part of time series Te: $Te(id)$;
$B(id) \leftarrow t$ is repeated l times
end for

The input of Algorithm 2.1 is one time series. The output is a new series containing many segments with equal length and each is represented by the average value of the individual segment. But it should be noted that the length of each segment needs to be specified when applying this algorithm to different databases.

In real life situations, such as in stock market, trades, prices change daily, even by minute, but investors normally are interested in weekly features or monthly patterns. Through the application of PAA in stock market, investors can easily find the information they need when the length of every segment is suitably set. As the stock prices of a company listing on Shanghai Stock Exchange as shown in Figure 2.2 (Shanghai Stock Exchange, 2016), the upper time series (containing the daily change of one stock in 240 working days) can be easily transformed to weekly series (the middle curve) and monthly series (the lower curve).

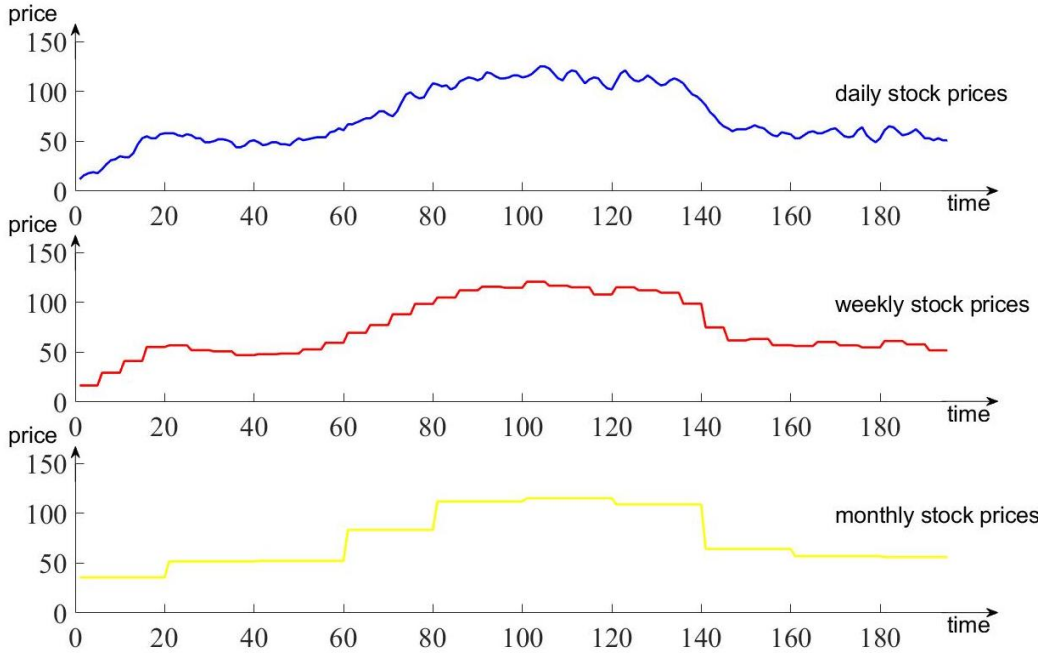


Figure 2.2 Daily, weekly and monthly stock prices

2.1.2 Symbolic Aggregate approximation

Symbolic Aggregate approXimation (SAX) is one of the famous dimensionality reduction and symbolic representation approaches in time series domain (Wang and Megalooikonomou 2008, Sun et al 2014). The basic idea of this approach is to convert a time series of length n to a symbolic series of length m consisting of s alphabets ($s \ll m$ and $m \ll n$). Given one time series Te of length n_{Te}, the operation of transforming Te into SAX manner involves two main steps: i) dimensionality reduction, ii) discretization.

In the first step, Te is normalized firstly, then the normalized time series is divided into m_{Te} equal-sized segments by PAA (described in subsection 2.1.1). In the second step, the transformed time series obtained through PAA is mapped into symbols using a look-up table containing a number of breakpoints (Lin et al 2003, Lin et al 2007). Because normalized time series have a high Gaussian distribution, breakpoints are easy to be obtained and can be defined as a sorted list of numbers, $B = \beta_1, \ldots, \beta_{\alpha-1}$, such that the area under a $N(0,1)$ Gaussian curve from β_i to β_{i+1} is $1/\alpha$. For example, when the original time series is divided into 4 different symbolic areas, there are 3 break points: $\beta_1 = -0.67$, $\beta_2 = 0$, and $\beta_3 = 0.67$; when the original time series is divided into 5 different symbolic areas, there are 4 break points: $\beta_1 = -0.84$, $\beta_2 = -0.25$, $\beta_3 = 0.25$, and $\beta_4 = 0.84$. When the number of symbolic areas is different, some samples of these breakpoints are shown in Table 2.1.

Table 2.1 A Look-Up Table Containing Break Points

	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

Using the defined breakpoints and taking the case with three breaking points as example, the corresponding conversion (from numerical series to symbolic series) is as follows:

$$\begin{aligned} \widetilde{te_\iota} &= 'A' \quad \text{if} \quad p < -0.67 \\ \widetilde{te_\iota} &= 'B' \quad \text{if} \quad -0.67 \le p < 0 \\ \widetilde{te_\iota} &= 'C' \quad \text{if} \quad 0 \le p < 0.67 \\ \widetilde{te_\iota} &= 'D' \quad \text{if} \quad p \ge 0.67 \end{aligned} \tag{2.2}$$

where $\widetilde{te_\iota}$ represents the ith segment of0 the symbolic series, A, B, C and D are the defined symbols, and p is the average value of the ith segment. Let alpha_i denote the ith element of alphabet, i.e. $\text{alpha}_1 = A$, $\text{alpha}_2 = B$, the mapping from a PAA approximation $\overline{Te}$ (original time series Te in PAA format)to $\widetilde{Te}$ (original time series in SAX format) is defined as:

$$\widetilde{\text{te}_\text{ı}} = \text{alpha}_\text{i} \quad \text{if} \quad \beta_{\text{j}-1} \le \overline{\text{te}_\text{ı}} \le \beta_\text{j} \tag{2.3}$$

The whole process of symbolic representation is summarized by following pseudocode:

Algorithm 2.2 Symbolization

Requirement: The length of the new time series after PAA: $numbertemplate$
A same size symbolic series $Asymbol$
Defined symbols
for $i1 = 1$ to $numbertemplate$ **do**
if $Z1(i1) \ge 0.67$ **then**
$A1symbol(i1) = symbolA$

else if $Z1(i1) < 0.67\ and\ \ Z1(i1) \geq 0$ **then**
$A1symbol(i1) = symbolB$
else if $Z1(i1) < 0\ and\ \ Z1(i1) \geq -0.67$ **then**
$A1symbol(i1) = symbolC$
else
$A1symbol(i1) = symbolD$
end if
end for

To use SAX for time series similarity measure, we need to introduce a meaningful distance measure of time series of symbols. Mentioned by Lin et al (2003) and Lin et al (2007), prior to distance measure between symbols, the distances between symbols were defined based on Gaussian distribution, as distances between symbols shown in Table 2.2.

Table 2.2 Look-Up Table of Distances between Symbols based on Gaussian Distribution

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*
A	0	0	0.44	0.75	1.03	1.28	1.53	1.80	2.12	2.56
B	0	0	0	0.32	0.59	0.84	1.09	1.36	1.68	2.12
C	0.44	0	0	0	0.27	0.52	0.77	1.04	1.36	1.80
D	0.75	0.32	0	0	0	0.25	0.55	0.77	1.09	1.53
E	1.03	0.59	0.27	0	0	0	0.25	0.52	0.84	1.28
F	1.28	0.84	0.52	0.25	0	0	0	0.27	0.59	1.03
G	1.53	1.09	0.77	0.50	0.25	0	0	0	0.32	0.75
H	1.80	1.36	1.04	0.77	0.52	0.27	0	0	0	0.44
I	2.12	1.68	1.39	1.09	0.84	0.59	0.32	0	0	0
J	2.56	2.12	1.80	1.53	1.28	1.03	0.75	0.44	0	0

Given two time series $Tr = [tr_1, tr_2, \dots, tr_n]$ and $Te = [te_1, te_2, \dots, te_n]$, based on SAX the distance between Tr and Te is computed as follow:

$$MINDIST(\widetilde{Tr}, \widetilde{Te}) = \sqrt{\frac{n}{w}} * \sqrt{\sum_{i=1}^{w} \left(dis(\widetilde{Tr}_i, \widetilde{Te}_i)\right)^2} \qquad (2.4)$$

where $\widetilde{Tr}$ and $\widetilde{Te}$ are the symbolic series corresponding to Tr and Te respectively, n is the length of original time series (Tr and Te), w is the length of equal-sized segments ($\widetilde{Tr}$ and $\widetilde{Te}$). The $dis(\dots)$ function is implemented using the predefined distances between symbols in Table 2.2 and this function can be expressed by following equation:

$$dis(\widetilde{Tr_i}, \widetilde{Te_i}) = \begin{cases} 0 & \left|\widetilde{Tr_i} - \widetilde{Te_i}\right| \leq 1 \\ \beta_{\max(\widetilde{Tr_i}, \widetilde{Te_i})-1} - \beta_{\min(\widetilde{Tr_i}, \widetilde{Te_i})} & otherwise \end{cases} \quad (2.5)$$

In above equation, $\beta_{...}$ is computed according to Table 2.1. For example, the original time series is represented by 4 different symbols, $\widetilde{Tr_i}$ is represented by symbol *A*, $\widetilde{Te_i}$ is represented by symbol *C*, as 3-1 not equal or less than 1, the distance between them is calculated as follows: $dis(A, C) = \beta_{\max(3,1)-1} - \beta_{\min(3,1)}$. As there are 3 break points, $\beta_{\max(3,1)-1} = 0$ and $\beta_{\min(3,1)} = -0.67$, $dis(A, C)$ is equal to 0.67.

From the publication of SAX, this method has been widely used in many time series data mining applications, such as human action recognition (Junejo and Al Aghbari 2012), financial investment and mobile data management (Hung and Anh 2007).

2.1.3 Principal Component Analysis

Principal component analysis (PCA) is probably the most widely used multivariate time series analysis. (Abdi and Williams 2010). PCA is used to analyzes dataset that is described by several inter-correlated variables, and its goal is to compress the size of dataset and keep the useful information.

The basic idea of PCA is shown in Figure 2.3. The original signal in plane built by A and B can be represented in a new plane built by C and D, where C and D are the linear combination of A and B. It can be found that points in original database project values on C axis and D axis, while the projected values on C axis are almost 0. This means we can ignore the influence of C when we are analyzing the data in the plane built by C and D. In this way, the analyzing process is transformed from 2-dimension to 1-dimension without much information loss.

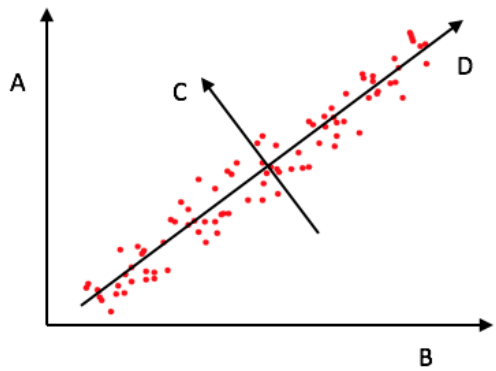


Figure 2.3 Basic idea of principal component analysis

Chapter 2. Literature Review

Given one n-dimension time series $D = (X_1, X_2, \dots, X_i, \dots, X_n)$, where X_i is a 1-dimensional time series with length equal to m and set as $X_i = (X_i^1, X_i^2, \dots, X_i^j, \dots, X_i^m)$. The entire procedure of PCA contains 5 steps:

- Step1: Centre the values in X_i as below:

$$X_i^j = X_i^j - \frac{1}{m}\sum_{j=1}^{m} X_i^j \tag{2.6}$$

- Step 2: Calculate the covariance matrix of the original time series

$$C = \begin{pmatrix} cov(X_1, X_1) & cov(X_1, X_2) & \cdots & cov(X_1, X_n) \\ cov(X_2, X_1) & cov(X_2, X_2) & \cdots & cov(X_2, X_n) \\ \vdots & & \ddots & \vdots \\ cov(X_n, X_1) & cov(X_n, X_2) & \cdots & cov(X_n, X_n) \end{pmatrix} \tag{2.7}$$

  where cov means the covariance between two candidates.

- Step 3: Compute the eigenvectors and eigenvalues of the covariance matrix

$$C - \lambda I = 0 \tag{2.8}$$

$$(C - \lambda I)X = 0 \tag{2.9}$$

  where C is the covariance matrix, I is the n by n identity matrix, λ is eigenvalues and X is eigenvectors.

- Step 4: Choose components and form a feature vector.

$$FeatureVectors = (eig_1, eig_2, \dots, eig_i, \dots, eig_k) \tag{2.10}$$

  eigenvectors that are obtained in step 3 are ordered by eigenvalues, from highest to lowest. In 2.10, eig_i means that its corresponding eigenvalues is the ith value in the ordered eigenvalues vector. In general, the first k eigenvectors are utilized to construct the feature vector, and the value of k is different when applying PCA to different areas.

- Step 5: Derive the new dataset.

$$DataNew = FeatureVector \times DataAdjusted \tag{2.11}$$

  where $DataNew$ is the transformed matrix containing principal components, $DataAdjusted$ is the mean adjusted data obtained in step 1.

The input of PCA is one n-dimensional time series or multivariate time series, the outputs include the mean adjusted matrix, covariance matrix, eigenvalues, eigenvectors, transformation matrix and principal components matrix. In conclusion, there are 4 advantages when applying PCA to similarity measure in multivariate time series database.

- Equal the Size of Different Multidimensional Time Series. Broadly speaking, the number of variables in multidimensional time series is always the same for a given application, but the number of observations is different because of the length of collecting time. Therefore, the traditional data mining method will face big challenges when comparing two multidimensional time series. For PCA, the first k highest eigenvalues and their corresponding eigenvectors are extracted and used to construct new matrix, the dimension of transformed matrix will be the same and the challenge of different size is resolved.
- Dimension Reduction. Because the length of observation is collecting time, the number of observations is far greater than the number of variables. If PCA is used to transform the original database, only a small number of components are used to represent original database, and therefore the dimension is effectively reduced.
- Improve the Accuracy of Data Mining. In general, the data in multidimensional time series is collected from different sources and the database is contaminated by noise because of various reasons, such as outdated equipment. After applying PCA to multidimensional time series, features with the most information are extracted and noise is ignored.
- Analysis as a whole. Variables in one dataset may be dependent with each other. During the process of analyzing multidimensional time series, the whole database has to be treated as a whole because the correlation between variables may be lost if we separate multidimensional time series into multiple 1-dimensional time series. For the application of PCA in multidimensional time series, the complete database is treated as a whole and the correlations between variables are saved.

Benefit by the advantages. In recent years, PCA-based data mining techniques have been used in various applications, such as iris recognition (Huang et al 2002), face recognition (Perlibakas 2004) and jaundice detection (Mansor et al 2011).

Chapter 2. Literature Review

2.2 Distance Measure

It is not possible to find two identical time series in any areas. However, different time series does not mean there is no relationship between each other. In recent decades, many researchers have focused their attention to finding the similarity and dissimilarity between different time series. For example, in the research of handwritten signature verification, conducted by Yao and Wei (2016), similarity measure method is used to identify whether the writer of the testing signature is the same writer of the template signature. In this part, some popular distance measure methods are reviewed.

2.2.1 Euclidean Distances

Euclidean distance, as a tool to take distance measure between time series, was firstly proposed in 1993 (Agrawal et al 1993). In the following decades, as an easy to understand and implement distance measure method, it has been widely used numerous fields, such as detection of outliers (Knorr et al 2000).

Given two vectors $X = [x1, y1]$ and $Y = [x2, y2]$, assume the distance between them is $d_{x,y}$, the square of $d_{x,y}$ is written as:

$$d_{x,y}^2 = (x1 - x2)^2 + (y1 - y2)^2 \tag{2.12}$$

Similarly, for two time series $X = [x1, x2, \dots, xn]$ and $Y = [y1, y2, \dots, yn]$, the Euclidean distance between them is written as equation 2.13, in which xi and yi are the ith instances in X and Y.

$$d_{x,y} = \sqrt{\sum_{i=1}^{n}(xi - yi)^2} \tag{2.13}$$

Minkowski distance, which is also called lp_norm, is an extension of Euclidean distance (Han et al 2011, Cha 2007). It is defined as:

$$Lp(X, Y) = (\sum_{i=1}^{n}(xi - yi)^p)^{1/p} \tag{2.14}$$

where p is called the order of Minkowski distance. In fact, for p=2, Minkowski distance is Euclidean distance.

when p=1, Minkowski distance is Manhattan distance and it is defined as:

$$L1(X, Y) = \sum_{i=1}^{n} |xi - yi| \tag{2.15}$$

Minkowski distance is easy to be implemented and understood, and can be well applied to other data mining problems, such as clustering (De Amorim and Mirkin 2012), but note that Minkowski distance is not applicable to distance measure between time series, this is because similar time series may be presented in different forms.

Vertical shift is one of the reasons that cause similar time series in different forms. As shown in Figure 2.4, the structures of time series A and B are the same with each other, but the amplitude values of them are different. Hence Euclidean distance and its extensions cannot be used to represent the similarity between them.

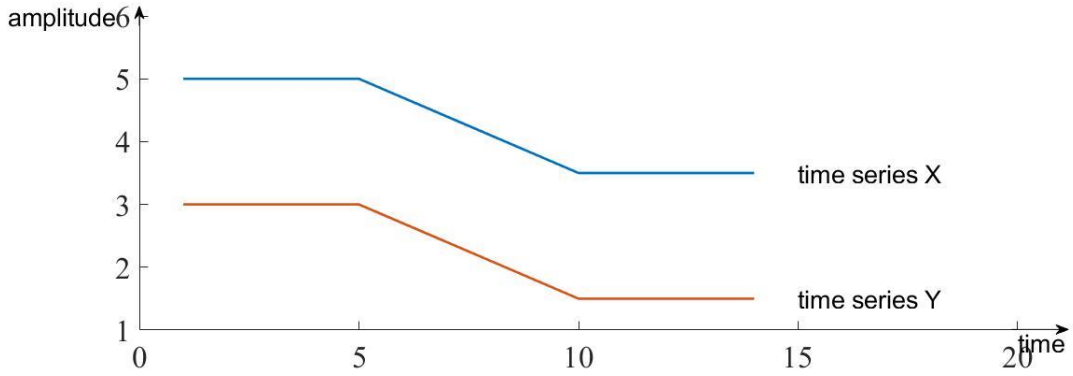


Figure 2.4 Vertical shift of time series

In order to reduce the influence of vertical shift on time series distance measure, a weighted time series similarity measure method, called v-shift, was proposed (Chan and Fu 1999). The definition of this method is: for time series $X = [x1, x2, \dots, xn]$ and time series $Y = [y1, y2, \dots, yn]$, they are defined as similar if they satisfy the following function:

$$d_{x,y} = \sqrt{\sum_{i=1}^{n} \left((xi - yi) - (\bar{X} - \bar{Y})\right)^2} \leq \tau \qquad (2.17)$$

where τ is a predefined threshold, $\bar{X}$ and $\bar{Y}$ are the mean values of time series X and Y.

Time warping is another reason that causes similar time series in different forms. As shown in Figure 2.5, time series A and B are similar with each other, but the peaks of these two time series are not at the same time. Under this condition, Euclidean distance and its extensions still cannot be used to define the similarity between them.

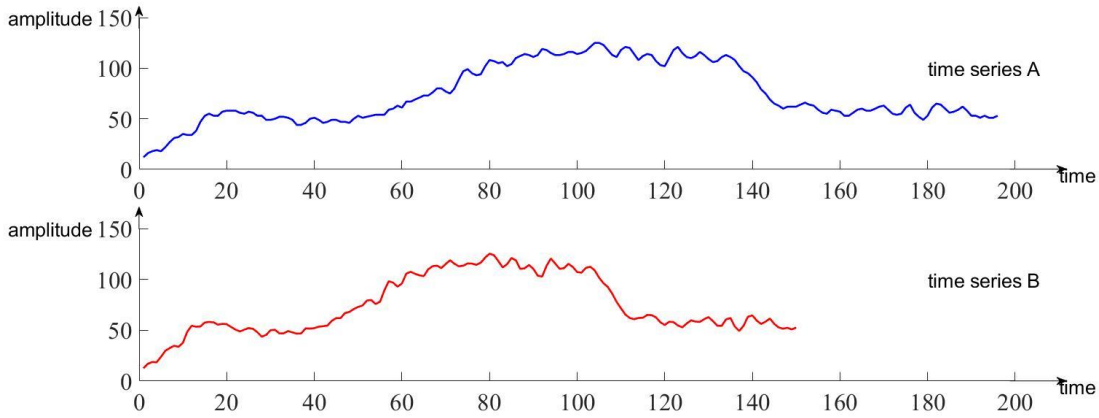


Figure 2.5 Time line warping of time series

2.2.2 Dynamic Time Warping

Due to the existence of timeline warping in time series computation, the final result of similarity computation may be distorted if we directly sum all the distances between corresponding points in a traditional manner. Dynamic time warping is such a method where two time series are warped in a nonlinear fashion and the similarity between the two time series is then measured in some way using the warped version of the time series (Muller 2007), as shown in Figure 2.6. Owing to this, the alignment between two time series will not be influenced by timeline drift, which can often cause error if Euclidean distance is directly used to measure the similarity.

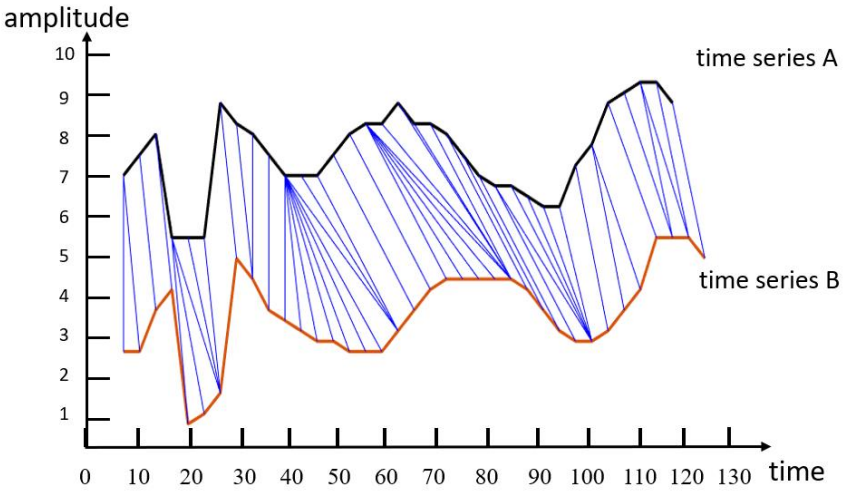


Figure 2.6. Alignment according to dynamic time warping

Dynamic time warping (DTW) was initially proposed for spoken word recognition in 1978 (Sakoe and Chiba 1978). It was used to measure similarity between testing voice and template speech signals, where the time line of voice has to be warped so that the

most similar characteristics can match each other. In the following decades, DTW had been widely used in pattern recognition (Berndt and Cliford 1994), fast similarity measure (Sakurai et al 2005) and genetic research (Aach and Church 2001). Different from traditional distance measure methods (Euclidean distances that are reviewed in subsection 2.2.1), DTW can recover optimal alignments between points in a template and the testing time series. For example, given two time series sequences, $\{a_i\}$ and $\{b_j\}$ $(i = 1, 2, \dots, n;\ j = 1, 2, \dots, m$, n and m are the number of values in $\{a_i\}$ and $\{b_j\}$ respectively) the optimal path between the two sequences, from the position $(1,1)$ to (n, m), is illustrated in Figure 2.7 (Yao and Wei 2016).

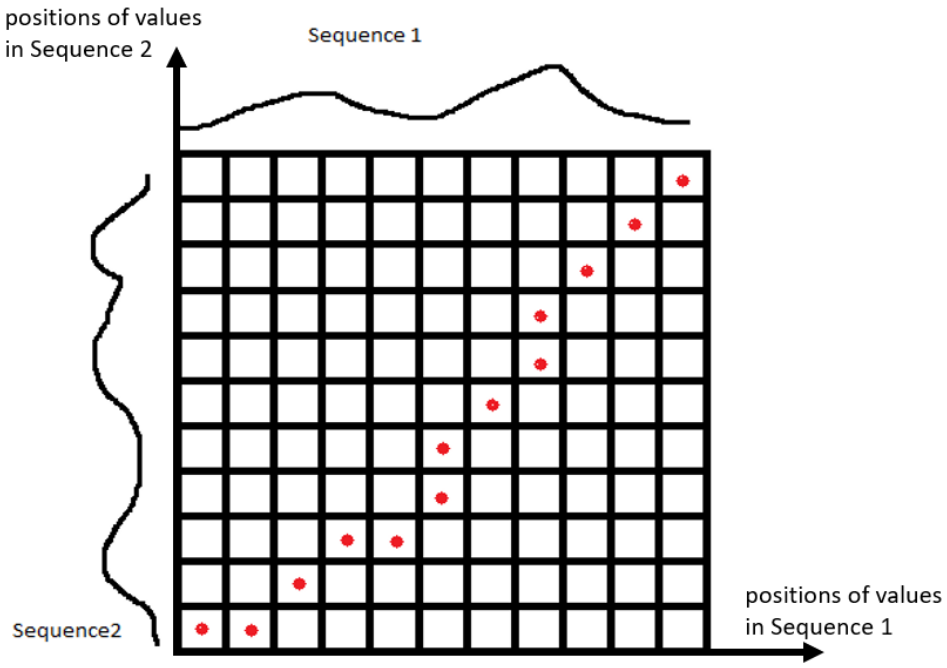


Figure 2.7 Optimal path between two sequence

We know that the time series sequences are:

$$Sequence1 \quad a_1, a_2, \dots, a_i, \dots, a_n$$
$$Sequence2 \quad b_1, b_2, \dots, b_j, \dots, b_m$$

We can build a distance matrix C to store the distance between two aligning points in the two sequences as below:

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,m} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,m} \\ \cdots & \cdots & c_{i,j} & \cdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,m} \end{bmatrix} \tag{2.18}$$

where $c_{i,j}$ represents the distance between a_i and b_j. Then we can get the cumulative distance matrix D:

$$D = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,m} \\ d_{2,1} & d_{2,2} & \dots & d_{2,m} \\ \dots & \dots & d_{i,j} & \dots \\ d_{n,1} & d_{n,2} & \dots & d_{n,m} \end{bmatrix} \tag{2.19}$$

where the elements $d_{i,j}$ is defined as

$$d_{i,j} = c_{i,j} + min\,[d_{i-1,j-1}, d_{i-1,j}, d_{i,j-1}] \tag{2.20}$$

here $d_{i,j}$ represents the minimum cumulative distance between two sequences from the beginning of Sequence1 to a_i and from the beginning of Sequence2 to b_j. Algorithm 2.3 below briefs the calculation procedure of distance between the two sequences and Algorithm 2.4 describes the computation of the optimal align path.

Algorithm 2.3 Distance Calculation

$N \leftarrow$ length of $Sequence1$
$M \leftarrow$ length of $Sequence2$
for $i = 1$ to N **do**
  for $j = 1$ to M **do**
    $C(i,j) \leftarrow sqrt(\left(Sequence1(i) - Sequence2(j)\right)^2)$
  end for
end for
for $i = 2$ to N **do**
  for $j = 2$ to M **do**
    $D(i,j) = C(i,j) + \min\,[D(i-1,j), D(i-1,j-1), D(i,j-1)]$
  end for
end for
$distance = D(N,M)$

The input of Algorithm 2.3 contains two time series sequences and the output is the dynamic time warping distance between these two sequences.

Algorithm 2.4 Optimal Path Finding

Requirement: The distance matrix $\boldsymbol{D}$ obtained in Algorithm 1
  length of $\boldsymbol{Sequence1}$: $\boldsymbol{N}$
  length of $\boldsymbol{Sequence2}$: $\boldsymbol{M}$
while $N + M$ equal to 2 **do**
  if $N - 1$ equal to 0 **do**
$M \leftarrow M - 1$

if $M-1$ equal to 0 **do**
$N \leftarrow N-1$
else $[values, number] \leftarrow min([D(N-1, M), D(N, M-1), D(N-1, M-1)])$
switch $number$
case 1 **do** $N \leftarrow N-1$
case 2 **do** $M \leftarrow M-1$
case 3 **do** $N \leftarrow N-1, M = M-1$
end switch
end if
$k \leftarrow k+1$
$w \leftarrow cat(1,2,[N, M])$
end while

The input of Algorithm 2.4 is dynamic time warping distance matrix and the output is optimal align path.

There are four requirements for DTW:

- Monotonicity. All the data in time series are obtained and stored in sequence. Although DTW has the ability to repeat the points with optimal alignments, the matching between points must abide by the time order.
- Continuity. Assuming that the neighbour points in warping path are $d_k = (i, j)$ and $d_{k-1} = (i', j')$ respectively, then the position information of i and j, $i'$and j' must obey the following rules:

$$\begin{gathered} i - i' = 0 \text{ and } j - j' = 1 \\ \text{or} \\ i - i' = 1 \text{ and } j - j' = 0 \\ \text{or} \\ i - i' = 1 \text{ and } j - j' = 1 \end{gathered} \tag{2.21}$$

  The requirement of continuity can ensure that there are no missing points during the calculation.
- Slope constraints. Every point in one time-series sequence cannot be aligned too many times in the other time series, therefore, slope constraints are needed to avoid large movements in a single direction.
- Boundary conditions. The first points and the end points of two time-series must be aligned to each other (Berndt and Cliford 1994).

Chapter 2. Literature Review

2.3 Anomaly Detection

Anomaly detection is a hot topic that has been discussed in various areas and domains. Most of the proposed methods are specifically proposed and improved for certain applications, while others are more generic. In this section, we try to provide a review of the researches on anomaly detection in recent decades.

2.3.1 classification-based anomaly detection

Classification is a data mining function that assigns items in a collection to target categories or classes (Krishnaiah et al 2014). Similar to the fashion of classification, the training step of classification-based anomaly detection is to create a model using the available data, the testing step of classification is to declare whether the testing instance is anomalous, using the model (generated in training step) (Steinwart et al 2005).

According to the number of classes in dataset, classification-based anomaly detection can be broadly divided into two groups, they are: one-class anomaly detection and multi-class anomaly detection.

- One-class anomaly detection techniques assume that there is only one class in the dataset, any test instance that does not match the model (generated in training step) is identified as anomalous. Such techniques constructed a classification model using a one-class algorithm, such as one-class SVMs (Scholkopf et al 2001).
- Multi-class anomaly detection techniques assume that there are two or more classes in the dataset (De Stefano et al 2000), a test instance that do not classified into any class is considered anomalous.

Broadly speaking, there are 2 advantages and 1 disadvantage of classification-based anomaly detection techniques:

- Advantage 1: Powerful algorithms can be used to distinguish which class the testing instance belongs to.
- Advantage 2: The testing step of classification-based techniques is fast, because each test instance is compared against the pre-obtained model.
- Disadvantage 1: Such techniques rely on the availability of accurate labels for various normal classes, this is usually impossible.

2.3.2 Clustering-based anomaly detection

Clustering is used to group similar data instances into clusters (Shirkhorshidi et al 2014). Although anomaly detection and clustering seem to be fundamentally different, several clustering-based anomaly detection approaches have been proposed. In general, clustering-based anomaly detection techniques are separated into three groups:

- Normal instance belongs to any one of the clusters, while outliers do not belong to any of them. According to this assumption, a known clustering method can be used to the dataset to define whether the testing instance is anomalous or not. Such as DBSCAN (Ester et al 1996), and ROCK (Guha et al 2000).
- Normal instances close to their corresponding nearest cluster centres, while outliers are far away from their nearest centres. According to this assumption, a clustering method is used to separate all the instances, and the distances between instances and their corresponding nearest cluster centres are used as anomaly detection score. Such as Self-Organising Maps (Smith et al 2002), k-means is used to train data and the generated clusters are used to classify testing data.
- Normal instances belong to large and dense clusters, while outliers belong to small or sparse clusters. According to this assumption, methods declare instances belonging to clusters whose size or density is below a threshold, as anomalous. Such as FindCBLOF (He et al 2003), assigns an anomaly score, which captures the size of the cluster to which the instance belongs.

Broadly speaking, there are three advantages and two disadvantages of clustering-based methods, shown as follows:

- Advantage 1: Clustering-based techniques can be operated in an unsupervised mode.
- advantage 2: Clustering-based techniques can be used to detect anomaly or anomalies from complex datasets through plugging in a clustering algorithm that can deal with the particular data type.
- advantage 3: The testing step for clustering-based techniques is not time-consuming because the number of clusters is a small constant.
- disadvantage 1: The effectiveness of clustering algorithm determines the performance of clustering-based anomaly.

- disadvantage 2: For outliers that far away to cluster centres, most clustering methods always force them into clusters. This may cause outliers to be allocated to a large cluster, and thus treated as normal.

2.3.3 Statistical anomaly detection

Normal instances occur in high probability regions, while anomalous instances occur in low probability regions. Under this assumption, statistical techniques fit a statistical model to the dataset and apply statistical techniques to determine whether a testing instance is anomalous. Both parametric and nonparametric techniques have been applied to fit a statistical model (Eskin 2000, Desforges et al 1998).

- As for parametric techniques, such as Gaussian Model-based anomaly detection, the distance between testing instance and estimated mean is defined as anomaly score, a threshold is then used to determine whether the testing instance is anomalous.
- As for non-parametric statistical models, such as histogram-based anomaly detection, for univariate data, a histogram is constructed in the train step, then a testing instance is checked whether it is falls in any one of the bins of the histogram.

In general, there are one advantage and two disadvantages of statistical techniques. They are shown as follows:

- Advantage 1: Statistical techniques can operate in an unsupervised setting without any need for labelled data.
- Disadvantage 1: Such techniques can only work effectively when the data is generated from a particular distribution.
- Disadvantage 2: For multidimensional data, histogram-based techniques cannot be used to capture the interactions between different attributes.

2.3.4 Spectral anomaly detection

Data can be projected into a lower dimensional space where normal instances and anomalous instances are different. Under this assumption, spectral techniques are used to transform the original data and detect the anomalous instances.

As principal components analysis (PCA) is widely used to project data into a low dimensional space, in 1996, (Parra et al.) proposed a such technique, which analyses the projection of each data instance along the principal components.

The advantages and disadvantages of spectral anomaly detection techniques are as follows:

- Advantage 1: Dimensionality reduction is automatically performed by spectral anomaly detection techniques.
- Advantage 2: Spectral anomaly detection techniques can work well in unsupervised settings.
- Disadvantage 1: Only if the normal and anomalous instances can be accurately separated in the lower dimensional space, spectral anomaly detection techniques are applicable.
- Disadvantage 2: It is high computational complexity for most spectral anomaly techniques.

2.3.5 Nearest-neighbour based anomaly detection

Normal instances close to their neighbourhoods, while anomalies are far from their neighbours. According to this assumption, nearest neighbour analysis has been used in several anomaly detection techniques. This technique requires a distance measure method to define the similarity between two instances. For nearest-neighbour based anomaly detection techniques, Euclidean distance is a popular choice, but more complex distance measures can also be used (Boriah et al. 2008; Chandola et al. 2008).

The advantages and disadvantages of nearest neighbour-based techniques are as follows:

- Advantage 1: These techniques can work well in unsupervised settings.
- Advantage 2: Such techniques can be directly applied to most kinds of data as long as a suitable distance measurement method is selected for given data.
- Disadvantage 1: If normal instances do not have enough close neighbours, or if anomalous instances have several close neighbours, the technique is not applicable.

- Disadvantage 2: Because the distance of each instance and all the other instances has to be computed, such techniques are always time-consuming.
- Disadvantage 3: Distance between instances is difficult to be computed when the data is complex..

2.4 Clustering

Clustering is used to identify structures in an unlabelled dataset by separate original data into different groups, where the within-group distance is minimized and between-group distance is maximized. Clustering is necessary when there is no labelled data in the original dataset regardless of what types the data is. Static data, as its name implies, means that the feature values do not change with time. Clustering methods for handing various static data are separated into five major types (Ham et al 2001): partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods. Two of these five methods: specifically partitioning methods and hierarchical methods, have been utilized directly for time series clustering, are reviewed in this section.

2.4.1 Partitioning methods

Given a time series database that contains n unlabelled sequences, partitioning methods can divide the database into k different groups, where each group represents a cluster and all these groups have to meet the following requirements: 1) there is at least one sequence in each group; 2) each sequence can only belong to one exactly cluster. There are two famous partitioning methods, for k-means, each cluster is represented by the average sequence of the group, for k-mediods, each cluster is represented by the sequence whose within-group-distance is the minimum one in the group.

K-means was first proposed almost three decades ago (Jain et al 1999) and has been widely applied in a variety range of domains, such as gene expression and prediction of student performance (Lu et al 2004, Oyelade et al 2010). The basic idea behind the method is to randomly choose k data as the initial cluster centre, using distance measure method to calculate the distances between all the rest data and the randomly selected cluster centres, and classify these data to their nearest cluster. Once all the data have been labelled, calculate the average data of every group so that it can be used as the representation of each cluster. The final result of k-means relies on iterative operation,

which will stop if the new average data is equal or approximate to the previous average data of the cluster.

The whole process of k-means based clustering is described as following 3 steps:

- Distances calculation: Assuming there are n patterns in original database, k-means starts with randomly selected k patterns as initial cluster centre, then calculate the distances between the rest data and these randomly chosen centres. The main part of this step is distance calculation because accuracy level of distance calculation determines the classification accuracy of unlabelled patterns. For most k-means based clustering calculation, Euclidean distance is used to calculate the distance between patterns.
- Average data calculation: Once all the data in k groups have been obtained, the arithmetic mean value of each group is calculated and used to replace the previous cluster centre because it meets the requirement that the within-group-distance of cluster centre is minimum when compare with other patterns in the same group.
- Comparison and decision making: Usually k-means based clustering takes several iterations, and it will stop if the distance between the new cluster centre and the previous cluster centre is less than a predefined small value (the amount of this value is defined according to the requirement of the clustering calculation). On the other hand, if the distance between the new cluster centre and previous centre is greater than the predefined value, clustering calculation should repeat previous steps.

For some isolated data in original database, they are always far away from clustering centre. Even so, k-means based clustering method still forces these data into clusters. Because of this, the average value of the group cannot correctly describe the features of the group and therefore influence future clustering calculation. K-medoids clustering method was proposed to reduce the effect of outliers in clustering procedure (Kaufman and Rousseeuw 2009). Because the advantage of k-medoids clustering method that real data points (medoids) is used as clustering centre and avoids the effect of outlier (Xu and Wunsch 2005), a lot of k-medoids based clustering methods were extended from K-medoids and applied in various domains in recent decades (Zhang and Couloigner 2005, Park and Jun 2009).

Similar with k-means based clustering method, the whole clustering process of k-medoids based clustering methods are also separated into three steps.

- Distance matrix building: Given a set of data, k-medoids based clustering method starts with randomly selected *k* different initial data, then build a distance matrix that contains the distances between the *k* pre-selected data and all the data in database. Euclidean distance is always used in this step to calculate the distances between different data. This is the first step of k-medoids based clustering method and this part is same with the first step of k-means based clustering method.
- Data classification and mediods chosen: For the first part of this step, all the undefined data is organized to their nearest cluster according to the distances matrix that is obtained in previous step. After that, as the basic idea of k-medoids based is to use a real data to represent the features of the group, within-group-distances of every data in the group are computed and the corresponding data with the minimum distance sum is defined as the medoid of the cluster. As introduced in k-means based clustering, the new cluster centre is the average value of the group, here, the new representative of the group is a real instance in the group.
- Comparison and decision: The whole clustering procedure always takes several iterations and it will stop if the new medoids of every cluster are equal to the previous one. Different with k-means based clustering method, for k-medoids, the new medoid must be the same with previous one, if not, the calculation has to keep running.

2.4.2 Hierarchical methods

Hierarchical clustering methods offer a way to build a hierarchical structure tree according to the similarity between different data. The root of the tree represents all the data in database and the top-level of the tree usually expressed by one cluster contains all the data. According to different requirements, the growth directions of hierarchical tree are different and generally can be classified as agglomerative methods and divisive methods. For agglomerative methods, the starting position is that each data in the database represents a cluster and every cluster only contains exactly one object. According to similarity measure between every two objects in database, the nearest two objects can align with each other and construct a new cluster. This alignment process has to repeat

several times and will stop until all the objects are classified to a same group. For divisive clustering methods, the beginning status is that all the data in database belongs to one cluster, then splits the data into two parts and goes on by dividing them further into smaller parts until each cluster only contains one object (Kaufman and Rousseeuw 2009, Xu and Wunsch 2005). Figure 2.8 illustrates the basic structures of agglomerative clustering methods and divisive cluster methods. It can also clearly describe the difference between these two methods.

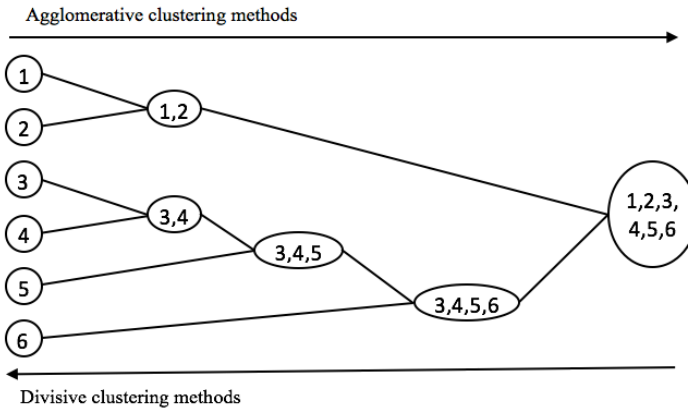


Figure 2.8 Description of agglomerative and divisive clustering methods

The purpose of most clustering calculation is organising a set of unlabelled data into a certain number of groups where the within-group-object distance is minimized and between-group-object distance is maximized. Therefore, agglomerative clustering methods are usually used in practice. Generally, the whole process of agglomerative clustering can be divided into 4 steps.

- Distance matrix: Given a database contains N patterns and using a suitable distance calculation method to calculate the distances between every pattern, the distance values are stored in a distance matrix.
- Match: Find the most nearest two patterns according to the distance matrix that is obtained in step 1, and then combine them to form a new cluster. There may be several pairs of data combined together at this step, and therefore, the result of this step may has several new clusters.
- Update: Once we get a new cluster in step 2, the previous data used to form the new cluster should be replaced by the new data.
- Repeat: Step 2 and step 3 have to keep repeating until all the patterns are classified in one cluster.

Chapter 2. Literature Review

2.5 Remaining Useful Life Estimation

Remaining useful life (RUL) estimation of safety related critical components has received increasing attention in recent decades, not only because the health status of these critical components are closely linked to life of related people, but also because the repair and maintenance of these components requires expenditure of additional money and resources. Remaining useful life estimation can be completed by using two main approaches, namely physics-based and data-driven approaches (Medjaher et al 2012).

2.5.1 Physics based methods

Physics-based methods are suitable for situations where precise theoretical models can be accurately constructed (Baraldi et al 2013). To create a physical model, we must start with the physical equation, the key parameters are then selected according to the target of the model (Bagul et al 2008).

In recent decades, several physical based RUL estimation models are proposed. For example, a physical model, proposed by Oppenheimer and Loparo (2002), is used to predict machine condition, based on crack growth law, this model could be applied to determine RUL of testing machine; developed by Li and Lee (2005), a gear meshing stiffness identification model is used to predict the RUL of a fatigue tooth crack; presented by Waston et al (2005), physics based simulation model and wear prediction model are combined to estimate the RUL of a highly dynamic high-power dry clutch system.

Physics-based models can be constructed according to first principles and physical mechanisms. When physical information is significantly complete, physics-based model will significantly outperform other types of prediction model in terms of RUL estimation. However, due to the lack of understanding of all failure modes, physics-based model can not be used to estimate the RUL of a complex system. Additionally, a physical model is usually created individually. Hence, a physics-based RUL estimation model is not applicable to a different system.

2.5.2 Data Driven Based Methods

Data-driven methods mainly predict RUL based on the equipment status monitoring data and the measurements of similar equipment or systems from health to failure degradation process (Tsui et al 2015, Zhang et al 2015, Heng et al 2009). These methods

only rely on previously observed data and do not need to require complex physical failure mechanism. In recent decades, data-driven based RUL estimation methods have been widely studied.

A neural network provides a means to analysing a complex system without any knowledge about the internal structure, hence these methods are suitable for predicting the RUL of complex equipment. For example, a neural network was developed by Byington et al (2004) to predict the RUL of aircraft components; a neural network model was proposed by Yu et al (2006) to predict the condition of a boring process during its full life cycle; Huang et al (2007) provided an approach to predict the RUL of a ball bearing based on neural network methods; and Guo et al (2017) proposed a health indicator based on recurrent neural network to predict RUL of bearings.

Similarity-based RUL estimation approaches are relatively new but have made promising performance. The basic idea behind these approaches is to extract historical degradation trajectories that are similar or same with that of testing equipment, and calculate RUL of the testing equipment according to RUL of the extracted historical trajectories. For example, a similarity-based RUL estimation approach was proposed by Wang et al (2008) to estimate RUL of aircraft engines; a similarity-based approach was provided by Zio and Maio (2010) for RUL estimation of nuclear system; Zhang et al (2015) proposed a similarity-based RUL estimation method to predict the RUL of high-pressure water pumps.

Data-driven models are implemented only from historical data, and are applicable when historical data is sufficiently abundant. Similar models can also be applied to other systems without understanding the complex physics. However, most results of data-driven models are not easy to explain or to be related to any physical meaning.

2.6 Summary

The reduction of original time series' dimensionality is crucial because most time series mining methods only work well when the number of dimensions is low (Wang and Megalooihonomou 2008). In the first section of this Chapter, we briefly review some time series transformation and representation methods, they are piecewise aggregate approximation, symbolic aggregate approximation and principal components analysis.

Chapter 2. Literature Review

The problem of similarity measure in time series database has attracted a lot of attention recently. This is because similarity measure is the most essential role in time series mining process. In the second section of this chapter, we provide a review of time series distance measure methods, includes Euclidean distance and dynamic time warping.

Anomalies are patterns in data that do not conform a well-defined notion. In some cases, anomalies translate significant and actionable information. In the third section of this chapter, we review some time series anomaly detection methods, including classification-based anomaly detection, clustering based anomaly detection, statistical based anomaly detection, spectral anomaly detection and nearest neighbour based anomaly detection.

Clustering time series data has applied in a wide range of applications and has attracted researches from a wide range of areas. This is because clustering analysis can be used as a pre-processing step for most data mining techniques, such as prediction and anomaly detection. In the fourth section of this chapter, we provide a review of two popular clustering techniques, they are partitioning clustering technique and hierarchical clustering technique.

Remaining useful life prediction has been applied to many applications, such as military, power systems, aerospace systems and manufacturing equipment. This is because accurate RUL estimation methods can increase availability, reliability and safety; and reduce maintenance and logistics cost. In the fifth section of this chapter, we review physical-based RUL estimation model and data-driven RUL estimation model.

In the following chapters, we will introduce the works I have completed during my PhD, effort including a novel calculation of distances between symbols and a distance measure method for similarity measure between symbolic series; a novel anomalies detection method for anomalies detection and extraction from ECG data; an automatic time series clustering (AT-means), from setting the initial centres to determination of number of clusters and generation of clusters; and a new multidimensional time series similarity measure method for similarity-based remaining useful life estimation method.

Chapter 3

Time Series Distance Measure

Due to the fact that there is a large number of complex data in time series, it will inevitably lead to significant expenditure of money and time if we want to directly measure the similarity between time series, and possibly worse, is that the final result may not be accurate or robust. In the past few decades, symbolic representations of time series have been considered in numerous works because such representations help researchers to avail of the wealth of data and improve the performance of distance measure between time series. In this chapter, a novel definition of distances between symbols and a distance measure method are proposed for time series similarity measure. In order to validate the performance of the proposed methods, we integrate the proposed distance table calculation method to symbolic aggregate approximation and its extension methods, and apply both the proposed methods and integrated methods to 1000 pairs of benchmark time series. The experimental results show the performance of time series distance measure is improved by applying our proposed methods.

3.1 Introduction

With the development of high-techniques in the past decades, a significant amount of data is generated and recorded every day from many application domains, such as finance, industry, agriculture, scientific experiments, medical observations, etc. According to an IBM (IBM 2017) report, 2.5 billion gigabytes of data were generated every day in 2012 and were estimated to 2.5 quintillion bytes every day in 2017. As a consequence, many time series representation methods have been proposed with two objectives: i) reducing the dimension of raw data so that the efficiency of data mining can be improved; ii) removing the noise from original data and remaining the main features of the raw data.

With the propositions of time series representation methods, because similarity measure between time series plays an important role in time series mining, many time series distance measure methods were proposed at the same time.

Decades ago, time series representation and dimensionality reduction method based on discrete Fourier transform (DFT) was firstly performed to measure similarity between time series (Agrawal et al 1993). But later, due to the fact that DFT can only preserve the information of the raw data in frequency domain whereas discrete wavelet transform (DWT) has the ability to keep the information of raw data in both frequency and time domain, DWT was introduced as a more powerful alternative for time series distance measure (Chan and Fu 1999, Wu et al 2000). Both of DFT and DWT are used to transform original data from time domain to frequency domain, and the similarity between original data is expressed by the distance between transformed series in frequency domain. So far, most time series representation and similarity measure methods are implemented in the time domain directly. One of the time domain transformation methods is piecewise aggregate approximation (PAA), which uses the segmented means to represent the original time series and such an approximation can be used to improve the efficiency of distance measure between time series (Keogh et al 2001a). Later, in order to adapt to the shape of the time series and make the distance between transformed series tightly close to the Euclidean distance of original data, PAA was extended to an adaptive piecewise constant approximation (APCA), which is used to transform one time series by a set of constant value segments of varying lengths (Keogh at al 2001b). Principal component analysis is another popular transformation method, which constructs a linear combination of the original data so as to represent the original time series in low dimensional space, and the distance between the transformed time series in the projected domain is used to represent the distance between original data (Yang and Shahabi 2004, Karamitopoulos at al 2010). Different from the above traditional time series representation and distance measure methods that use real-valued numbers, another commonly used method for time series representation is symbolic representation, which converts the numeric time series to some symbolic form and uses the distance between symbolic series to represent the similarity between the original data. One of the most popular symbolic representations and distance measure methods is symbolic aggregate approximation (SAX) (Lin et al 2003). (explanation: This part has been partly minimized. This part briefly introduces the

development of time series representation. For part 2 in Chapter 2, that is review of time series representation)

Compared with real-valued time series, symbolic series is more powerful to tackle some specific tasks, such as anomaly detection and motif discovery (Wang et al 2013, Fu 2011). Since the publication of SAX, it has been widely used in many time series data mining applications, such as similarity search on financial time series (Canelas et al 2012), human action recognition (Junejo and Aghbari 2012), mobile data management (Tayebi et al 2011), etc. Note that the original SAX has a number of limitations. For example, the extreme points of every segment during distance calculation are neglected and the trends of raw data are not taken into account. Therefore, modifications and extensions to the original SAX have been proposed to improve the performance for time series mining. In Lkhagva et al's (2006a) and Lkhagva et al's (2006b) researches, in order to keep the information of extreme points in financial time series data, an extended SAX (ESAX) was proposed by adding two new points in equal sized segments. It was shown that the representation and similarity measure defined in ESAX are more precise than SAX in term of high frequency dataset. In the research conduct by Sun et al (2014), in order to improve the SAX representation precision in distinguishing different time series with similar average values while with different trends, a SAX trend distance (SAX-TD) approach was proposed by defining trend distance quantitatively with starting and ending points and replacing the original SAX distance measure with the weighted trend distance. It was demonstrated that SAX-TD can significantly decrease the classification error rate (Sun et al 2014).

It is noteworthy that due to the distances between symbols are defined according to the Gaussian distribution, distance measure between symbolic series calculated using the above mentioned symbolic representations are not accuracy enough. In this work, in order to improve the performance of distance measure between symbolic series, we propose a novel method to define the look-up table and a new method to measure distance between symbolic time series. The two proposed methods are integrated to SAX and SAX-TD, which are referred to as 'improved SAX' and 'improved SAX-TD', respectively, in subsequent sections for convenience of description. In order to validate the performance of the proposed method, we apply the modified methods to 1000 pairs of benchmark time series obtained from UCR time series collection (Chen et al 2015). For comparison

purpose, the original SAX, ESAX and SAX-TD are also applied to the same time series datasets.

The remainder of this chapter is organized as follows. Section 3.2 briefly reviews ESAX and SAX-TD. Section 3.3 illustrates the new definition of look-up table and the new distance measure method, and Section 3.4 reports the experimental results. Finally, Section 5 briefly summarizes this chapter.

3.2 Related Works

In recent decades, SAX representation was applied to many time series data mining problems, and improvements from different aspects were proposed at the same time. In this section, we review two popular extended works (ESAX and SAX-TD).

3.2.1 Extended Symbolic Aggregate Approximation

Financial time series is typically characterized by a few critical points, such as maximum point and minimum point. However, SAX is based on PAA representation for dimensionality reduction and mean value based representation causes a highly possibility to miss some important information. In Lkhagva et al's (2006a) studies, in order to reduce the loss of important information during the representation procedure, Extended SAX (ESAX) was proposed for symbolic representation of financial time series.

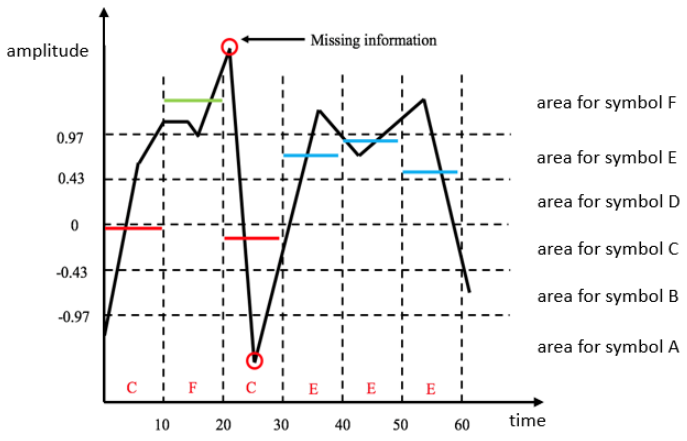


Figure 3.1 SAX representation of financial time series (Lkhagva et al 2006a)

The improvement of ESAX starts at the step of dimensionality reduction. As the SAX representation of one typical financial time series shown in Figure 3.1, the segment from time 20 to time 30 are labelled as symbol C whereas the maximum and minimum point, shown in the small red cycles, located in the area of A and F. For the representation of

one time series, when the segments of financial time series are only represented by their corresponding average values, important points of some segments may be missed. In (Lkhagva et al 2006a), aimed to fully represent time series data, maximum and minimum points are added for the representation of every segment. Take the segment from time 20 to time 30 in Figure 3.1 as an example. Because the maximum point locates in the area of symbol A and the minimum point locate in the area of symbol F, this segment is going to be represented by the combination of symbol A, B and F.

Given two symbolic series, $S1 = [A, B, C]$ and $S2 = [C, A, B]$, although both $S1$ and $S2$ contain the same symbols, we cannot define them as similar because the position of symbols in $S1$ and $S2$ are different. In Lkhagva et al's (2006a) research, because three values are used to represent one segment, their positions have to be defined prior to similarity measure.

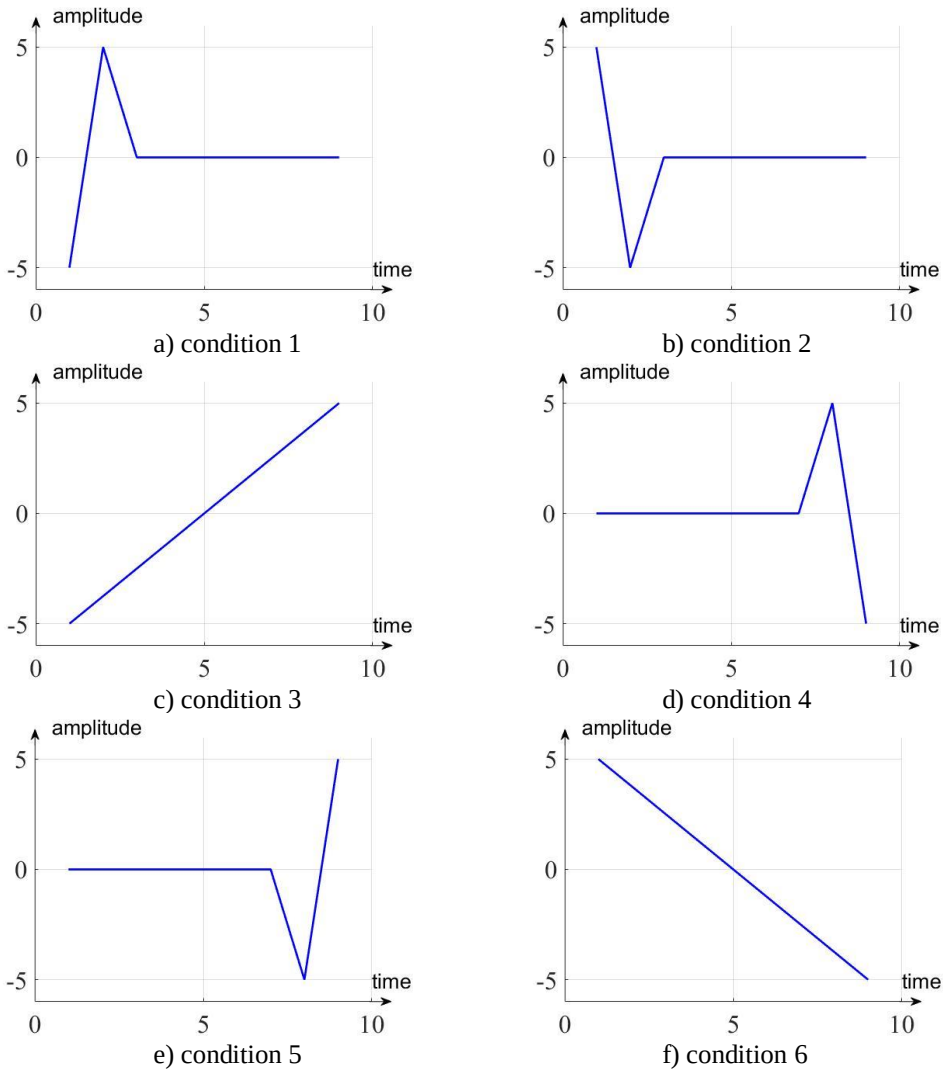


Figure 3.2 Orders of extreme points

For one time series segment, the location of the average value is the middle position of the segment and set as P_{mid}, the position of the maximum point is set as P_{max}, the position of minimum point is set as P_{min}. Depending on the order in which these three values appear, there are 6 different conditions: 1) $P_{min} > P_{max} > P_{mid}$, 2) $P_{max} > P_{min} > P_{mid}$, 3) $P_{min} > P_{mid} > P_{max}$, 4) $P_{mid} > P_{max} > P_{min}$, 5) $P_{mid} > P_{min} > P_{max}$, 6) $P_{max} > P_{mid} > P_{min}$, as shown in Figure 3.2.

According to the 6 different conditions, the ordering calculation can be expressed by following equation (Lkhagva et al 2006a):

$$[S_1 \quad S_2 \quad S_3] = \begin{cases} [\mathrm{S_{min}} \quad \mathrm{S_{max}} \quad \mathrm{S_{mid}}] & \text{if} \quad \mathrm{P_{mid}} < \mathrm{P_{max}} < \mathrm{P_{min}} \\ [\mathrm{S_{max}} \quad \mathrm{S_{min}} \quad \mathrm{S_{mid}}] & \text{if} \quad \mathrm{P_{mid}} < \mathrm{P_{min}} < \mathrm{P_{max}} \\ [\mathrm{S_{min}} \quad \mathrm{S_{mid}} \quad \mathrm{S_{max}}] & \text{if} \quad \mathrm{P_{max}} < \mathrm{P_{mid}} < \mathrm{P_{min}} \\ [\mathrm{S_{mid}} \quad \mathrm{S_{max}} \quad \mathrm{S_{min}}] & \text{if} \quad \mathrm{P_{min}} < \mathrm{P_{max}} < \mathrm{P_{mid}} \\ [\mathrm{S_{mid}} \quad \mathrm{S_{min}} \quad \mathrm{S_{max}}] & \text{if} \quad \mathrm{P_{max}} < \mathrm{P_{min}} < \mathrm{P_{mid}} \\ [\mathrm{S_{max}} \quad \mathrm{S_{mid}} \quad \mathrm{S_{min}}] & \text{if} \quad \mathrm{P_{min}} < \mathrm{P_{mid}} < \mathrm{P_{max}} \end{cases} \tag{3.1}$$

The whole process of ESAX representation is also summarized in Algorithm 3.1.

Algorithm 3.1 Extended Symbolic Aggregate Approximation

Requirements: One time series: T
Length of segment: m, length of T: n
for $i = 1 : n$ step by m **do**
$j \leftarrow i : i + m - 1$
$t_{mid} \leftarrow mean(T(j))$
$t_{max} \leftarrow max(T(j))$
$t_{min} \leftarrow min(T(j))$
$P_{mid} \leftarrow sum(j)/m$
$P_{max} \leftarrow find(T(j) == tmax)$
$P_{min} \leftarrow find(T(j) == tmin)$
if $P_{min} < P_{mid} < P_{max}$ **do**
$b((i - mod(i,m))/m + 1, :) = [tmax, tmid, tmin]$;
if $P_{max} < P_{mid} < P_{min}$ **do**
$b((i - mod(i,m))/m + 1, :) = [tmin, tmid, tmax]$;
if $P_{mid} < P_{max} < P_{min}$ **do**
$b((i - mod(i,m))/m + 1, :) = [tmin, tmax, tmid]$;
if $P_{mid} < P_{min} < P_{max}$ **do**
$b((i - mod(i,m))/m + 1, :) = [tmax, tmin, tmid]$;
if $P_{min} < P_{max} < P_{mid}$ **do**
$b((i - mod(i,m))/m + 1, :) = [tmid, tmax, tmin]$;
if $P_{max} < P_{min} < P_{mid}$ **do**
$b((i - mod(i,m))/m + 1, :) = [tmid, tmin, tmax]$;
end if
end for

The input of Algorithm 3.1 is one time series and the output is represented by ordered maximum, minimum and mean values.

The ordered important values are then symbolized according to the predefined requirements, such as the number of symbols. Once the ESAX representation of time series is obtained, because the distances between symbols have been defined according to Gaussian distribution, as shown in Table 2.2 (page 13), the distance between time series can be calculated.

Given two time series A and B, their corresponding ESAX representation are $\tilde{A} = [\widetilde{a_1}, \widetilde{a_2}, \dots, \widetilde{a_n}]$ and $\tilde{B} = [\widetilde{b_1}, \widetilde{b_2}, \dots, \widetilde{b_n}]$, the distance between A and B is calculated as following equation.

$$D = \sqrt{\frac{n}{k}} * \sqrt{\sum_{i=1}^{k} \left(dist(\widetilde{a_i}, \widetilde{b_i}) \right)} \tag{3.2}$$

where n is the length of A and B, k is the length of $\tilde{A}$ and $\tilde{B}$, the $dist(\dots)$ function is implemented using the predefined distances between symbols in Table 2.2 (page 13).

3.2.2 Symbolic Aggregate Approximation – Trend Distance

During the step of dimensionality reduction in SAX representation, because each segment is represented by its average value, some important points are missed and hence ESAX was proposed by tripling the dimensions of original SAX. Due to the same reason that average values are used to represent segments, the directions of segments are ignored, which results in that the SAX representation cannot distinguish different time series with similar average values. Mentioned by Sun et al (2014), in order to improve the distance calculation of SAX, a modified distance measure by integrating the SAX distance with a weighted trend distance was proposed.

Several typical segments with same average value are shown in Figure 3.3 (the curves represent original time series and the lines are their corresponding average values), with the application of SAX, all of the segments are represented by one symbol. This means these segments are going to be defined as same with each other after the process of similarity measure although that is not true. Because of this, trend plays an important role in the analysis of similarity between time series.

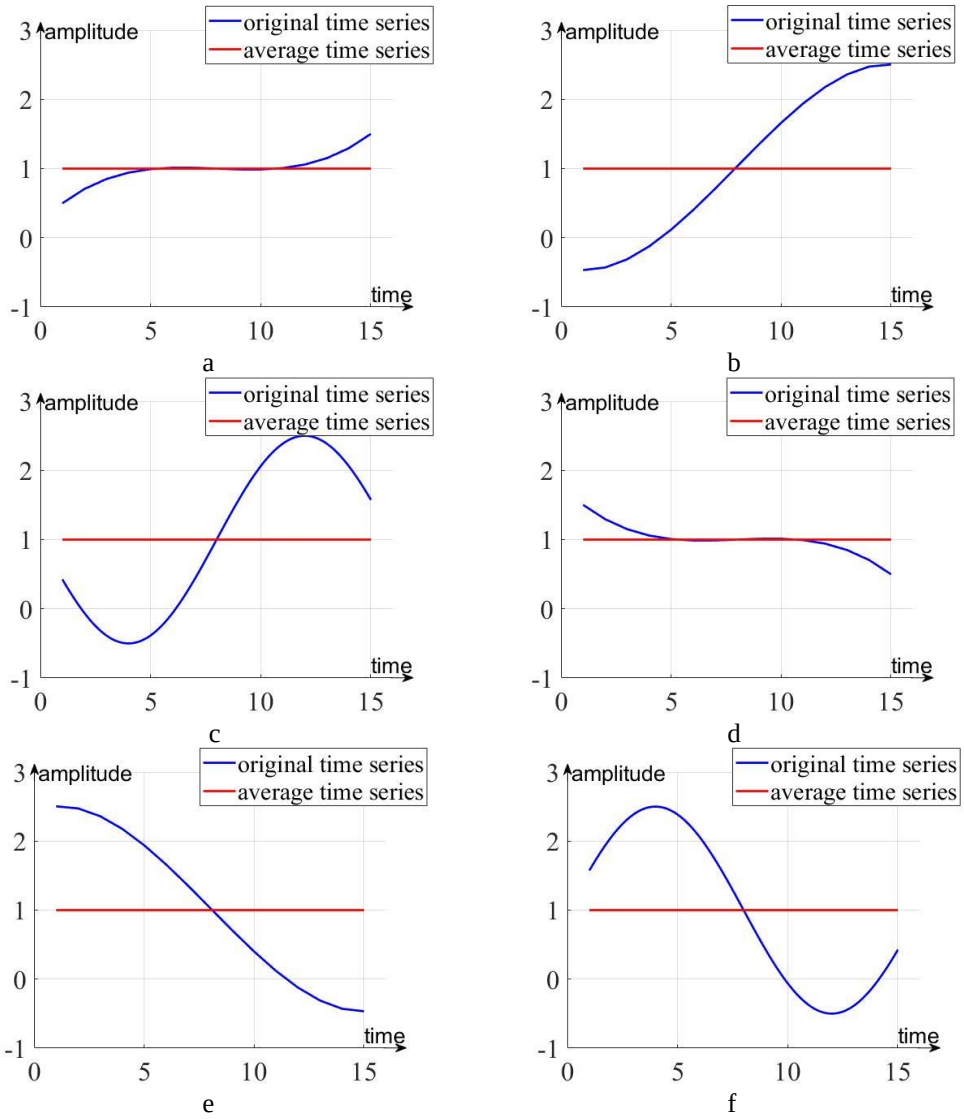


Figure 3.3 Six typical segments with same average value but different trends (Sun et al 2014). a) level and slight up, b) obvious up, c) down, up and down, d) level and slight down, e) obvious down, f) up, down and up.

Given one time series T, its corresponding SAX representation is $Ts = [ts_1, ts_2, \dots, ts_m]$. Mentioned by Sun et al (2014), trend variation of each segment of T is incorporated into Ts, as the representation manner shown below:

$$T_{SAX-TD} = [\Delta T(1), ts_1, \dots, \Delta T(i), ts_i, \Delta T(i+1), \dots, ts_m, \Delta T(m+1)] \qquad (3.3)$$

where ts_i is the ith symbols in the SAX representation of T, $\Delta T(i)$ is trend variation, which is used not only to represent the distance between ending value and average value of the ith segment, but also to represent the distance between staring value and average value of the $(i+1)$ th segment. The whole transformation process of SAX-TD is described by Algorithm 3.2.

Algorithm 3.2 Symbolic Aggregate Approximation Trend Distance

Requirements: One time series: T
Length of segment: m
$n \leftarrow$ length of T
for $i = 1{:}\,m{:}\,n$ **do**
$id = i{:}\,i + m - 1$
$tmid = mean(A(id))$
$At = A(id);$
$ts = At(1);$
$b((i - mod(i,m))/m + 1, :) = [ts, tmid];$
end
$newseries = [reshape(b', [1, number * 2]), A(n)];$

Given two segment of time series, P and Q, their SAX-TD representation are $P_{st} = [\Delta P(ts),\ \tilde{P},\ \Delta P(te)]$ and $Q_{st} = [\Delta Q(ts),\ \tilde{Q},\ \Delta Q(te)]$, the trend distance $td(P,Q)$ is defined as follows:

$$td(P,Q) = \sqrt{\left(\Delta P(ts) - \Delta Q(ts)\right)^2 + \left(\Delta P(te) - \Delta Q(te)\right)^2} \tag{3.4}$$

For the distance measure of two time series (A and B) with same length of n, the SAX-TD based distance calculation is as follows:

$$TDIST(\tilde{A},\tilde{B}) = \sqrt{\frac{n}{m}} * \sqrt{\sum_{i=1}^{m}\left(\left(dist(\tilde{a_i},\tilde{b_i})\right)^2 + \frac{m}{n} * \left(td(a_i,b_i)\right)^2\right)} \tag{3.5}$$

where $\tilde{A}$ and $\tilde{B}$ are the SAX-TD representation of A and B, m is the number of elements in $\tilde{A}$ and $\tilde{B}$, a_i and b_i are the ith segment of A and B, $\tilde{a_i}$ and $\tilde{b_i}$ are the SAX representation of a_i and b_i. For the distance calculation function $dist(\ldots)$, it is the same with the distance measure between symbols in SAX and ESAX, which is implemented using the predefined distances between symbols in Table 2.2 (page 13) (Lin et al 2003).

3.3 Distance Measure between Symbolic Series

In this section, distance measure between symbolic series is introduced as follows: i) definition of distances between symbols, ii) time series distance measure based on the proposed method, iii) proof of lower bound.

3.3.1 Definition of Distances between Symbols

The symbols in symbolic series, transformed from real-valued time series, can only characterize the changes of amplitude, but cannot indicate the real values of their corresponding segments. To calculate the distance between symbolic series, it is necessary to define the distances between symbols. Based on the definition of the concept of distance tables (Lin et al 2003, Lkhagva et al 2006a, Sun et al 2014), Gaussian distribution can be used to define the distances between symbols, as shown in Table 2.2 (page 13) It should be noticed that the distance between two neighbour symbols is defined as 0 although they are not the same. More than that, once two segments are labelled as a same or neighbour symbol, the distances between them are set to 0 although they might not be the same. Such a definition of distances between symbols is easy to understand and implement, but will influence the accuracy of further calculation.

In this subsection, given the number of break points, in order to improve the accuracy of the look-up table and satisfy the requirement of symbolic representation that the distance between two symbolic series is lower than the Euclidean distance of the original two time series, the distances between different symbols are computed based on the maximum and minimum values of PAA representation in each area. For example, SAX representation of one time series is defined in Figure 3.4, the distance between symbol A and symbol B is defined as the distance between the minimum mean value of symbol A (the mean value of the segment in area A) and the maximum mean value of symbol B (mean value of the 3rd segment in area B).

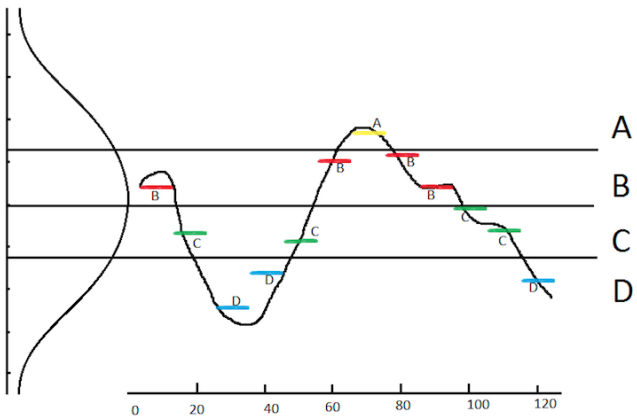


Figure 3.4 The SAX representation of one time series

Defining the number of break points is 3, the pseud-code of the proposed calculation of distances between symbols is given in Algorithm 3.3 below:

Algorithm 3.3 Definition of Distances between Symbols

Requirements: Candidate time series A
Candidate time series B
Length of segment m
$n \leftarrow$ length of candidate time series
$Cseries \leftarrow [A, B]$
$Cseries\_PAA \leftarrow PAA(Cseries)$
$Cseries\_symbol \leftarrow SAX(Cseries)$
for $i = \frac{n}{m} * 2$ **do**
  $location\{symbol\} \leftarrow find(Cseries\_symbol(i) == symbol)$
  $\max(symbol) \leftarrow \max(Cseries\_PAA(location\{symbol\}))$
  $\min(symbol) \leftarrow \min(Cseries\_PAA(location\{symbol\}))$
  for $i = 1{:}4$ **do**
    for $j = 1{:}4$ **do**
      if $i = j$
        $table(i, j) \leftarrow 0$
      else
        $x1 \leftarrow abs(\max(i) - \min(j))$
        $x2 \leftarrow abs(\min(i) - \max(j))$
        $table(i, j) \leftarrow \min(x1, x2)$
      end if
    end for
  end for
end for

The input of Algorithm 3.3 are two real-valued time series and the length (number of points) of every segment. The output is the distance table that contains the distances between symbols, and this table can only be used as the look-up table for the distance calculation of the input time series. This work proposes a new definition of a look-up table that is different from the traditional one as shown in Table 2.2 (page 13) (Lin et al 2003). For example, given two time series A and B (which are the 1st and 12nd time series in CBF (Cylinder Bell and Funnel) dataset (Chen et al 2015), when the number of break points is 8 and the length of segment is 6, the corresponding distance table is shown as Table 3.1.

Compare Table 3.1 and Table 2.2 (page 13), we can find that the distances between symbols in Table 2.2 is not accurate enough. For example, the distance between symbols B and D is 0.36 in Table 3.1 while in Table 2.2 the distance is 0.33. It can also be noticed that the distances between neighbour symbols are not equal to 0 in Table 3.1. For example, the distance between symbols C and D is 0 in Table 2.2 while in Table 3.1 the distance is

0.04. As a consequence, using maximum and minimum values of each area to define the distances between symbols is more reasonable.

Table 3.1 Lookup Table Defined by the Proposed Method

	A	B	C	D	E	F	G	H	I
A	0	0.32	0.59	0.89	1.25	1.60	1.95	2.46	2.54
B	0.32	0	0.06	0.36	0.72	1.07	1.41	1.93	2.01
C	0.59	0.06	0	0.04	0.40	0.75	1.10	1.61	1.69
D	0.89	0.36	0.04	0	0.15	0.50	0.84	1.35	1.44
E	1.25	0.72	0.40	0.15	0	0.18	0.52	1.03	1.12
F	1.60	1.07	0.75	0.50	0.18	0	0.34	0.85	0.94
G	1.95	1.41	1.10	0.84	0.52	0.34	0	0.51	0.60
H	2.46	1.93	1.61	1.35	1.03	0.85	0.51	0	0.07
I	2.54	2.01	1.69	1.44	1.16	0.94	0.60	0.07	0

It should be noted that distances between symbols in Table 2.2 can be treated as a special case of the newly defined distance table here, that is, there is at least one segment in every edge area and there are at least 2 segments in every middle area. In details, the minimum value and maximum value of segments in upper edge area and lower edge area are equal to the maximum break point and minimum break points, respectively, and the maximum and minimum values of segments in every middle area are equal to their corresponding upper and lower break points. Taking the SAX representation in Figure 3.4 as an example, there are 1, 4, 4 and 3 segments in different areas represented by symbols A, B, C and D, respectively. The average value of the segment in area A (the upper edge area) should be equal to upper break point, which is 0.67. The average value of the 3rd segment in area B, which is the segment with largest value among the 4 segments in area B, should be equal to 0.67. The average value of the 4th segment in area B, which is the segment with minimum average value among the 4 segments in area B, should be equal to 0. The average value of the 3rd segment in area C, whose average value is the largest one among 4 segments in area C, should be equal to 0. The 2nd segment in area C, whose average value is the smallest one among the 4 segments in area C, should be equal to -0.67. The 2nd segment in area D, whose average value is the maximum one among the average values of the 3 segments in area D, should also be equal to -0.67.

In order to validate the performance of the proposed method for distance measure, two time series (the 1st and 12nd sequences of ECG dataset provided by Chen et al (2015) are extracted, and both the proposed method and the previously published symbolic representation method are also applied to these two time series to calculate the distance between them. The length of both time series is 960 and the Euclidean distance between them is 10.9904. With a choice of 8 break points for symbolization, the distances calculated by different methods are shown in Table 3.2.

Table 3.2 Distances between Two Time Series based on SAX, ESAX, SAX-TD and The Proposed Method with Different Segment Length (The Euclidean distance is 10.9904).

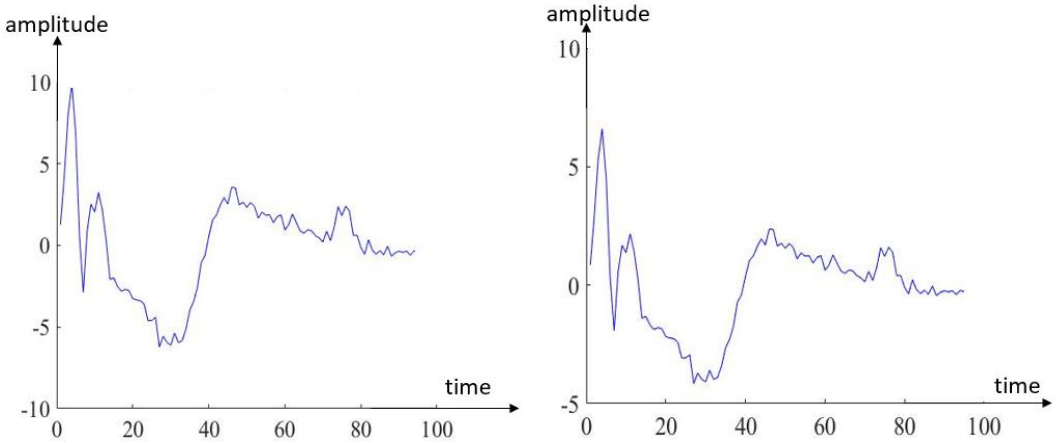
Length of segment	3	6	12	24	48
SAX (Lin et al 2003)	6.6111	6.8039	5.7081	4.5204	0
Improved SAX	7.3456	7.8207	8.1173	7.2524	3.2199
SAX-TD (Sun et al 2014)	7.9677	8.1300	6.0896	4.9174	0.5799
Improved SAX-TD	8.5870	8.9982	8.3899	7.5063	3.2717
ESAX (Lkhagva et al 2006a)	6.4581	6.2862	6.5940	5.0276	12.5179

The first column illustrates the names of symbolic series distance measure methods, in which “improved SAX” and “improved SAX-TD” mean that the look-up table used in distance calculation step of SAX and SAX-TD is replaced by the proposed look-up table. The values in the first row of the remaining columns indicate the length of segments. The other columns from left to right show the calculated distances based on different methods when the length of every segment changes from 3 to 48. It can be noticed that with the replacement of the proposed method, the calculated distance is robust and comparable to the Euclidean distance. For the ESAX, it can be seen that its results float significantly and do not always follow the lower bound Euclidean distance. This may be because ESAX is primarily designed for high-frequency time series with extreme points but not generic for most time series.

3.3.2 Distance Calculation

The first step of SAX is to normalize input series to have mean of zero and a standard deviation of one (Lin et al 2003). This is because that the conversion from PAA representation to symbolic series is based on Gaussian distribution. Following the normalization step, the similarity between input series will be described by distance between normalized series. However, for time series with the same normalized shape but

different amplitudes, such as the time series X and Y defined in Figure 3.5a and 3.5b, we cannot define them as same although the distance between them based on SAX is 0.



a) time series X b) time series Y

Figure 3.5 Two time series with same normalized series but different amplitude

Given one sequence $A = [a_1, a_2, \ldots, a_i, \ldots, a_n]$, with the application of the normalization step in SAX, A is transformed by the equation:

$$anew_i = \frac{a_i - \mu}{\sigma} \tag{3.6}$$

where a_i is the ith value in A, μ is the average value of A, σ is the standard deviation of A, $anew_i$ is the ith value in the normalized series. Given two time series $X = [x_1, x_2, \ldots, x_i, \ldots, x_n]$ and $Y = [y_1, y_2, \ldots, y_i, \ldots, y_n]$, after the normalization, the corresponding series of X and Y become $Xnew = [xnew_1, xnew_2, \ldots, xnew_m]$ and $Ynew = [ynew_1, ynew_2, \ldots, ynew_m]$. Based on SAX, symbolic representation of X and Y are $Xs = [xs_1, xs_2, \ldots, xs_i, \ldots, xs_p]$and $Ys = [ys_1, ys_2, \ldots, ys_i, \ldots, ys_p]$. Let the distance between two symbols in Xs and Ys be:

$$dist1(xs_i, ys_i) = d \tag{3.7}$$

where $dist1(\ldots)$ is implemented via indexing the look-up table.

As mentioned in SAX, because the distance between two symbols is used to express the similarity between their corresponding original values, we have:

$$Ed(xnew_i, ynew_i) = dist1(xs_i, ys_i) = d \tag{3.8}$$

where $Ed(\ldots)$ is Euclidean distance defined below:

$$\sqrt{(xnew_i - ynew_i)^2} = d \tag{3.9}$$

Inserting (3.6) to (3.8) and (3.9), we get:

$$\sqrt{\left(\frac{x_i-\mu_x}{\sigma_x} - \frac{y_i-\mu_y}{\sigma_y}\right)^2} = d \tag{3.10}$$

where x_i and y_i are the ith values in X and Y, μ_x and μ_y are the average values of X and Y, σ_x and σ_y are standard deviation of X and Y.

Distance measure between X and Y based on SAX can be represented by:

$$d_{SAX}(X,Y) = \sqrt{\sum_{i=1}^{n}\left(\frac{x_i-\mu_x}{\sigma_x} - \frac{y_i-\mu_y}{\sigma_y}\right)^2} \tag{3.11}$$

where $d_{SAX}(\dots)$ means distance calculation between input time series based on SAX. For the distance measure between X and Y based on Euclidean distance, it is calculated as follows:

$$Ed(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.12}$$

Note that SAX-based distance measure method is a special case of Euclidean distance with two requirements: i) standard deviations of X and Y have to be equal to 1; ii) the mean values of X and Y have to be equal to each other. In this work, the SAX-based distance measure is extended. Based on the proposed method, there is only one requirement, that is, the standard deviations of input series have to equal each other.

For the distance between x_i and y_i, as the distance between xs_i and ys_i is d, the distance between original points can be calculated using the following equation:

$$Ed1(x_i, y_i) = \begin{cases} \sqrt{\left(\mu_x - \mu_y + \sigma d\right)^2} & if \quad xs_i \geq ys_i \\ \sqrt{\left(\mu_y - \mu_x + \sigma d\right)^2} & if \quad xs_i < ys_i \end{cases} \tag{3.13}$$

where function $\mathrm{Ed1}(\dots)$ means calculate the distance between original points.

The distance between X and Y is defined as:

$$Ed(X,Y) = \sqrt{\sum_{i=1}^{n}\left(Ed1(x_i, y_i)\right)^2} \tag{3.14}$$

To show that this proposed method is more generic for distance measure between time series, we apply our distance measure method, along with the look-up Table 2.2 (page 13), to calculate the distance between time series in Figure 3.5a and 3.5b. The Euclidean distance between these two time series is 92.1727, when the length of each segment is 12. Based on the proposed method, the distance is 69.4890, while the distance is 0 based on SAX.

3.3.3 Proof of Lower Bounding

Thanks to the aforementioned advantages (dimensionality reduction and lower bound of Euclidean distance), SAX is the most famous one among many time series symbolic representation and distance measure methods. In this subsection, we show that with the integration of our proposed look-up table and the new distance measure method, the distance between transformed series is also lower bound the Euclidean distance of the original series.

Given two time series $A = [a_1, a_2, \dots, a_i, \dots, a_n]$ and $B = [b_1, b_2, \dots, b_i, \dots, b_n]$, it was proved by Lin et al (2003) that PAA distance lower bounds the Euclidean distance as:

$$\sqrt{\frac{n}{m}} * \sqrt{\sum_{i=1}^{m}\left(\bar{a}_i - \bar{b}_i\right)^2} \leq \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{3.15}$$

where n is the number of elements in time series A and B, m is the length of every segment, $\bar{a}_i$ and $\bar{b}_i$ are mean values of the segments where a_i and b_i belong.

In order to prove that the proposed method proves a lower bound to Euclidean distance, we have to prove

$$TSdist(A,B) \leq Dpaa(A,B) \tag{3.16}$$

where $TSdist(\dots)$ means the proposed distance measure method, $Dpaa(\dots)$ represents PAA distance function. Note that (3.16) can be rewritten as

$$\sqrt{\frac{n}{m}} * \sqrt{\sum_{i=1}^{m}\left(Ed1(a_i, b_i)\right)^2} \leq \sqrt{\frac{n}{m}} * \sqrt{\sum_{i=1}^{m}\left(\bar{a}_i - \bar{b}_i\right)^2} \tag{3.17}$$

or equivalently, (3.17) and (3.18) can be rewritten as:

$$\left(\overline{a_\iota} - \overline{b_\iota}\right)^2 \geq \left(Ed1(a_i, b_i)\right)^2 \tag{3.18}$$

where $\overline{a_\iota}$ and $\overline{\mathrm{b}_\iota}$ are the average values of corresponding segments in the original time series. Hence we have:

$$\frac{\overline{a_\iota} - \mu_\mathrm{a}}{\sigma} = \overline{a\mathrm{new}_\iota} \tag{3.19}$$

$$\frac{\overline{\mathrm{b}_\iota} - \mu_\mathrm{b}}{\sigma} = \overline{\mathrm{bnew}_\iota} \tag{3.20}$$

Both of these equations can also be written as follows:

$$\overline{a_\iota} = \sigma * \overline{a\mathrm{new}_\iota} + \mu_\mathrm{a} \tag{3.21}$$

$$\overline{\mathrm{b}_\iota} = \sigma * \overline{\mathrm{bnew}_\iota} + \mu_\mathrm{b} \tag{3.22}$$

When $\mathrm{a}_i \geq \mathrm{b}_i$, (3.18) can be written as:

$$\sigma * \left(\overline{a\mathrm{new}_\iota} - \overline{\mathrm{bnew}_\iota}\right) \geq \sigma * d \tag{3.23}$$

When $\mathrm{a}_i \leq \mathrm{b}_i$, (3.18) can be written as:

$$\sigma * \left(\overline{\mathrm{bnew}_\iota} - \overline{a\mathrm{new}_\iota}\right) \geq \sigma * d \tag{3.24}$$

where d is the distance between symbols (correspond to $\overline{a_\iota}$ and $\overline{b_\iota}$). As the definition of the proposed look-up table, the distance between two symbols is the minimum one of following distances:

$$d1 = abs\begin{pmatrix} maximum\ average\ in\ the\ area\ of\ symbol\ A \\ -minimum\ average\ in\ the\ area\ of\ symbol\ B \end{pmatrix} \tag{3.25}$$

$$d2 = abs\begin{pmatrix} minimum\ average\ in\ the\ area\ of\ symbol\ A \\ -maximum\ average\ in\ the\ area\ of\ symbol\ B \end{pmatrix} \tag{3.26}$$

Obviously, the value of d in our defined look-up table is smaller than the absolute value of $\left(\overline{a\mathrm{new}_\iota} - \overline{\mathrm{bnew}_\iota}\right)$ when $\mathrm{a}_i \geq \mathrm{b}_i$ and is smaller than the absolute value of $(\overline{\mathrm{bnew}_\iota} - \overline{a\mathrm{new}_\iota})$ when $\mathrm{a}_i \leq \mathrm{b}_i$. Hence, our proposed symbolic series distance measure method provides a compact lower bound for Euclidean distance.

3.4 Experiments and Comparisons

In order to demonstrate the performance of the proposed method for time series distance measure, we integrate the proposed method to SAX and SAX-TD, and apply these integrated symbolic representation and similarity measure methods to 10

benchmark datasets provided by Chen et al (2015). For comparison purpose, we also apply SAX, ESAX and SAX-TD to the same datasets. In this section, basic information of the benchmark datasets is described firstly, and then comparison of results is proposed.

3.4.1 Dataset Description

UCR time series database, as the largest public collection of class-labelled time series datasets, contains 48 datasets (Chen et al 2015), each of which contains from 36 to 9236 time series sequences. The sequences in each dataset have an equal length, but from one dataset to another the length of sequences varies from 24 to 1882 (Paparrizos and Gravano 2015). In this chapter, 1000 pairs of time series sequences are selected from 10 different datasets to evaluate the performance of our proposed method (100 pairs of time series are selected from each dataset). The basic information of these 10 datasets is listed in Table 3.3.

Table 3.3 Basic Information of the Selected Time Series

No.	Dataset	Length of Selected Time series	Number of Time Series
1	Synthetic Control	60	600
2	CBF	128	930
3	ECG	96	200
4	Yoga	426	3300
5	Fish	463	350
6	Beef	470	60
7	Coffee	286	56
8	Olive Oil	570	60
9	Trace	275	200
10	50 Words	270	955

3.4.2 Comparison of Efficiency

As mentioned by Lin et al (2003), there are two important characteristics of SAX: i) dimensionality reduction; ii) distance measure between symbolic series lower bounds Euclidean distance. An evaluation method, which is commonly used to validate the performance of symbolic representation and distance measure methods, called as tightness of lower bound (Wang et al 2013).

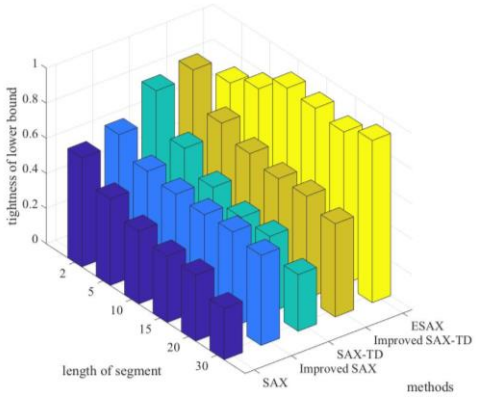
Tightness of Lower Bound (TLB), as its name implies, is the level of how close the calculated distance between symbolic series and the Euclidean distance between original time series. This measure is roughly defined as below:

$$Tightnes\ Lower\ Bound(TLB) = \frac{Lower\ Bound\ Dist(T,S)}{True\ Euclidean\ Dist(T,S)} \tag{3.27}$$

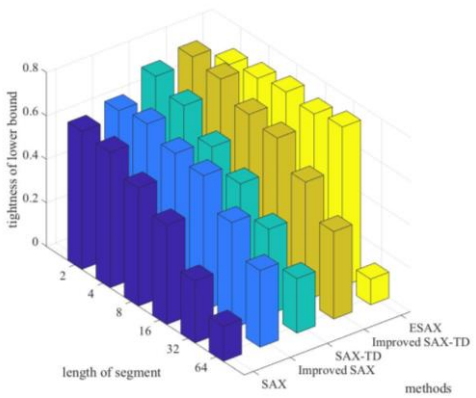
The value of TLB should be in the range of 0 to 1. When the value of TLB is equal to one, it means that a representation and distance measure method can completely replace Euclidean distance-based distance measure. On the other hand, when the value of TLB is close to 0, it means that the symbolic representation and distance measure method used is not ideal as an alternative to Euclidean distance. In general, the higher the value of TLB is, the better the performance of the corresponding representation method.

We integrate the adapted look-up table approach to SAX and SAX-TD, and apply these improved methods to 1000 pairs of time series. For comparison purpose, we also apply the original SAX, SAX-TD and ESAX to the same datasets. The results are shown in Figure 3.6, where z-axis is tightness of lower bound, y-axis describes the length of segments, and x-axis indicates the methods used, with '1' for SAX, '2' for the improved SAX, '3' for SAX-TD, '4' for the improved SAX-TD, and '5' for ESAX.

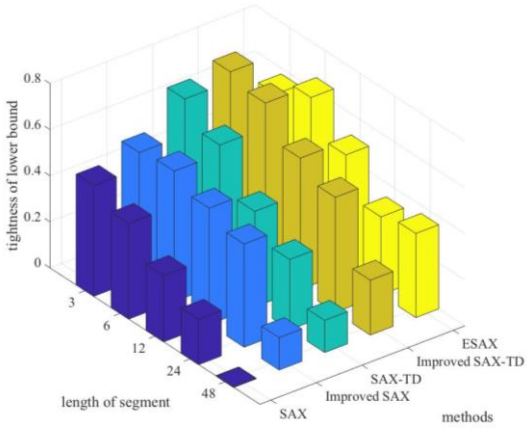
Based on the results of the 5 different methods for the 10 different datasets, with the length of segments equal to each other, the TLB values of the improved methods are higher than that of the corresponding original methods. As mentioned earlier, ESAX is more suitable for time series with high frequency and its performance floats significantly among these 10 datasets. Furthermore, because the distance measure based on ESAX cannot guarantee lower bounds Euclidean distance of the original time series, ESAX for time series representation and similarity measure is not widely acceptable. In summary, our proposed method improves the performance of distance measure between symbolic series and preserve the advantages of time series symbolic representation.



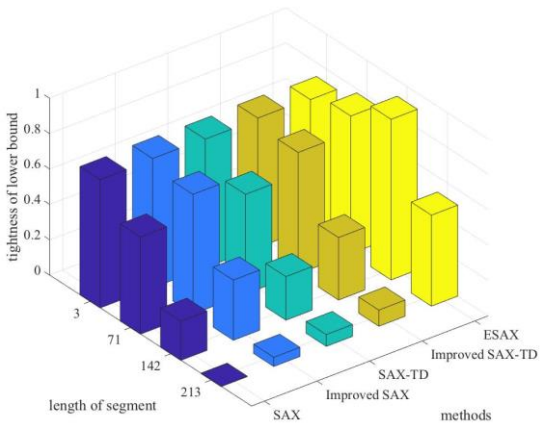
1) Performance of time series representations and distance measure methods on Synthetic Control



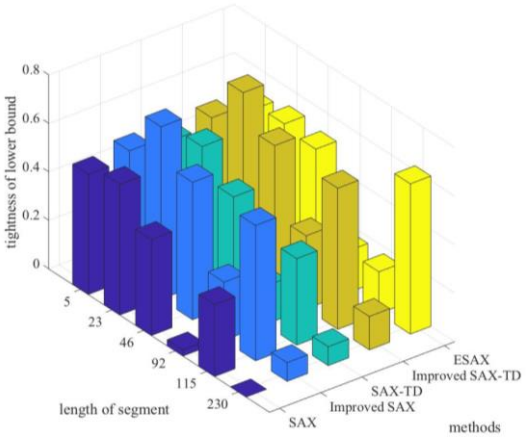
2) Performance of time series representations and distance measure methods on CBF



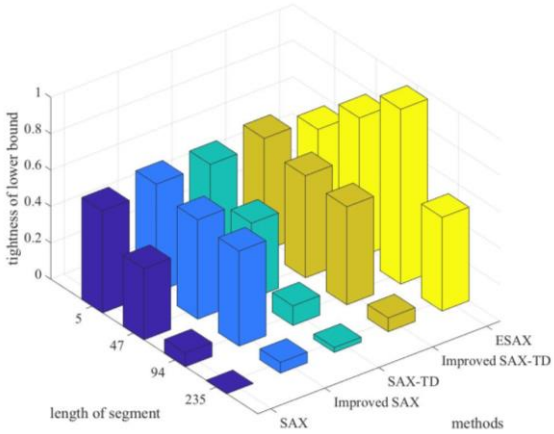
3) Performance of time series representations and distance measure methods on ECG



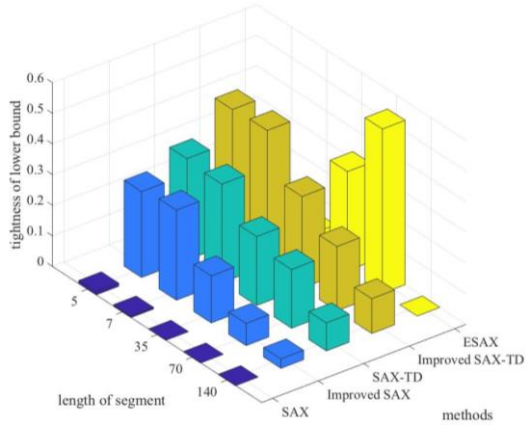
4) Performance of time series representations and distance measure methods on Yoga



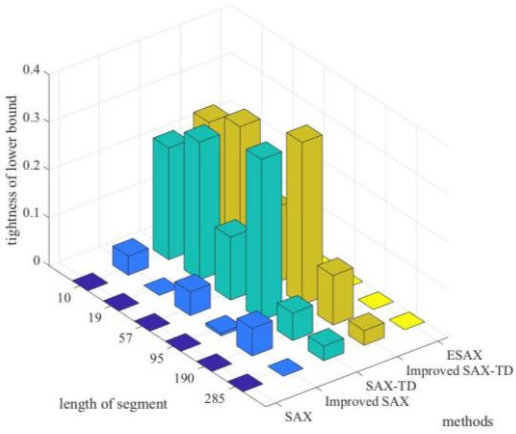
5) Performance of time series representations and distance measure methods on Fish



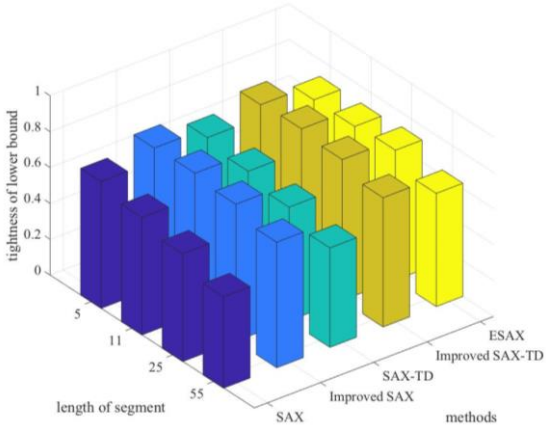
6) Performance of time series representations and distance measure methods on Beef



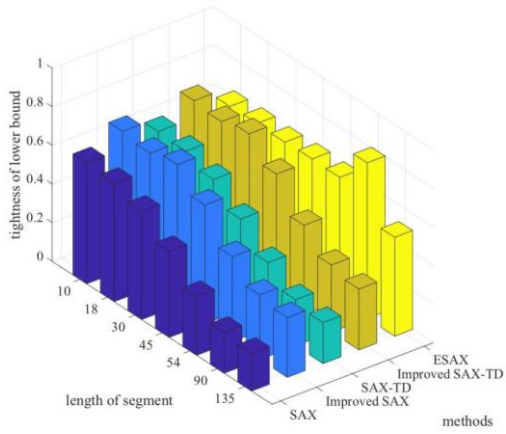
7) Performance of time series representations and distance measure methods on Coffee



8) Performance of time series representations and distance measure methods on Olive Oil



9) Performance of time series representations and distance measure methods on Trace



10) Performance of time series representations and distance measure methods on 50 Words

Figure 3.6 Performance (tightness lower bound) comparison

3.5 Summary

An efficient representation of the original high-dimensional time series can help save a lot of time during the analysis of time series, and correct calculation of the distance between time series can improve the accuracy of similarity measure and provide strong support for further time series mining. In this chapter, in order to improve the performance of distance measure between symbolic series, we proposed a novel definition of distances between symbols and an improved distance measure method. The basic idea of the proposed look-up table is to use the maximum and minimum mean values of all segments in individual areas to calculate the distances between symbols, and calculate the distance between original time series based on the distance between symbolic series. By integrating the new proposed methods to SAX and SAX-TD, the performance of the corresponding original algorithms can be significantly improved, as shown through the case studies on the 1000 pairs of benchmark time series.

Chapter 4

Anomaly Detection of Time Series: an application to ECG data

In this chapter, because timeline warping may exist in the process of time series similarity measure, dynamic time warping is modified by considering the optimal align path in distance calculation. In order to reduce the complexity of similarity measure, the modified dynamic time warping is integrated to the proposed method in chapter 3 to calculate the distance between segments. In addition, because previous proposed anomalies detection methods can only work well when all the anomalies in one time series are significantly different from each other, average non-self match distance is proposed to detect anomalies. To validate the performance of the proposed distance measure and anomalies detection methods, these proposed methods, together with brute force discord discovery and adaptive window discord discovery, are applied to real ECG data selected from MIT-BIH database. The experimental results show that our proposed method outperforms the other methods.

4.1 Introduction

Anomalies are patterns in data that do not conform to a well-defined notion of normal behaviour (Chandola et al 2009). In real life, anomalous information appears in various fields because of different reasons, such as pathological changes in human organs, network intrusion, malfunction of machine, etc. Figure 4.1 gives a visual intuition of an anomalous segment which is highlighted by a ellipse. Such an anomalous segment may not easily be noticed although it may contain critical information, such as it deliverss a message that a person is having a heart attack. In recent decades, researchers in data

mining domains realize that anomalies in some typical fields contain very important information, and hence anomaly detection received extensive attention and became a very hot researching area.

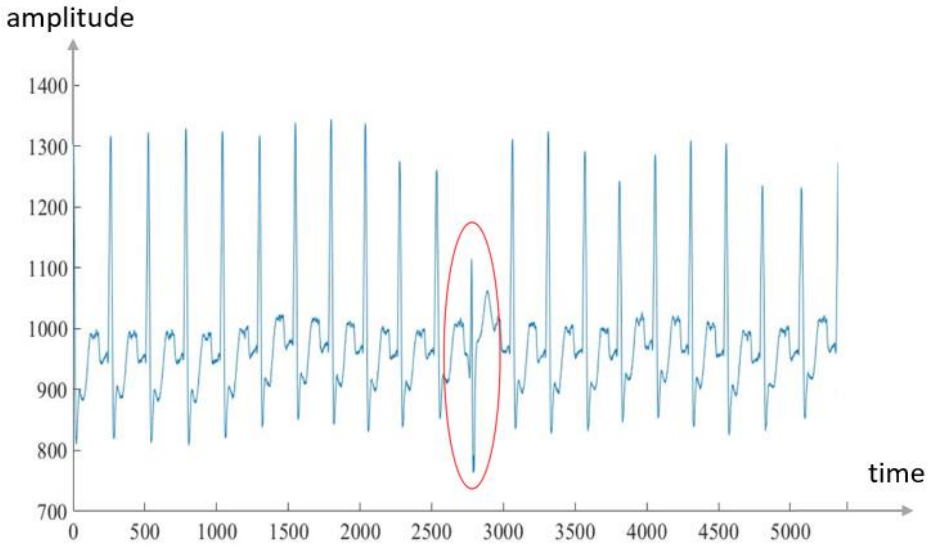


Figure 4.1 Anomaly detection and label in an ECG data (Collected from MIT-BIH Database)

In the past few decades, various kinds of data mining approaches were proposed in order to detect anomalous information from un-investigated data, they are: i) classification-based anomaly detection; ii) clustering-based anomaly detection; iii) statistical anomaly detection; iv) spectral anomaly detection; and v) nearest neighbour-based anomaly detection. In terms of classification-based anomaly detection (Scholkopf et al 2001, De Stefano et al 2000), a classifier is constructed by training the available dataset, then the testing instance is classified as normal or anomalous by using the classifier. For clustering-based anomaly detection (Ester et 1996, Guha et al 2000), based on distance measure methods, instances that are closed to centroid are defined as normal, while that are far away from centroid are set as anomalies. With regard to statistical anomaly detection (Saligrama and Chen 2012), instances that occur in high probability areas are defined as normal, while instances that occur in low probability regions are defined as anomalous. For spectral anomaly detection (Agovic et al 2007), the original data is projected to a lower dimensional space in which normal and anomalous instances can be easily identified. Regarding nearest neighbour-based anomaly detection (Boriah et al 2008, Chandola et al 2008), instances are defined as normal when they are close to their non-self matches. On the contrary, instances are labelled as anomalies when the

distances between them and their nearest non-self matches are greater than a pre-obtained threshold.

Since traditional anomaly detection methods primarily focus on detecting anomalous points while most data that are obtained from real life is recorded in time series, some methods, whose basic ideas are traditional anomaly detection methods, were proposed to focus on anomaly detection of time series. Among these time series anomaly detection methods, brute force discord discovery (BFDD), which belongs nearest neighbour based anomaly detection, is the easiest method to be understood and implemented (Keogh et al 2005). Based on BFDD, the whole process of anomaly detection is achieved with nest iterations, where the outer iteration considers every possible candidate sub-sequence and the inner iteration is a linear scan to identify the candidate's nearest non-self match (Lin et al 2005). This method is not only easy to be understood and implemented, but also has another advantage in that the length of subsequence is the only required input. However, a significant flaw of this method is that this method is time-consuming, which means it is not suitable for even moderately large datasets. In order to improve the efficiency of BFDD, heuristic discord discovery (HDD) was also proposed by Keogh et al (2005). Based on HDD, the first step is to construct the heuristic order of outer iteration, and the second step is to extract each candidate subsequence in heuristic order and find its corresponding nearest non-self subsequence in the inner iteration. The third step is to detect anomalies according to the distances between candidates and their corresponding nearest non-self match subsequences (Keogh et al 2006). Through the construction of heuristic order, HDD can improve the efficiency of BFDD-based anomaly detection. However, it should be noted that the calculation of heuristic order is not uniform, which means there is no guarantee that it is efficient for all kinds of time series data. A special type of time series is ECG, which is the collection of electrical changes of the heart beat over time by external electrodes attached to human body. In the clinical study, cardiovascular diseases (e.g. arrhythmia, myocardial ischemia, etc.) occur when the heart of a cardiac patient does not work normally over a certain period. The corresponding ECG will become different from other normal signals, or in other words, ECG becomes an anomalous segment of the heart beat process. Because of this, effective detection of anomalous segments in ECG data can make a significant contribution to heart diagnosis. In the research conducted by Chuah et al (2007), in order to improve the performance of BFDD-based ECG anomaly detection, adaptive window-based discord discovery

(AWDD) was proposed for detecting anomalous heartbeats, and the experimental results showed that AWDD outperforms BFDD in terms of efficiency of ECG anomaly detection. (explanation: This part has been partly minimized. This part briefly introduces the development of time series anomaly detection based on nearest-neighbour based anomaly detection. For part 3 in Chapter 2, that is review of several types of anomaly detection, such as clustering based anomaly detection, nearest-neighbour based anomaly detection .)

However, both BFDD and AWDD have a common drawback that they cannot directly detect all the anomalies when there are two or more similar anomalies. For example, the anomalies of ECG signals of one patient are always caused by one reason and the caused anomalies are always similar with each other. Another drawback is that Euclidean distance is set as the distance measure method in both BFDD and AWDD. It should be noted, however, that the existence of timeline drift in time series distance measure may distort the accuracy of similarity measure if Euclidean distance is directly used to calculate the distance. In this paper, in order to correctly detect all the anomalous segments, a new calculation, called as average non-self match distance (ANMD), is used to detect anomalous segments from raw data; and a modified dynamic time warping (MDTW) is used to calculate the distance between candidates. To demonstrate the performance of the proposed methods for ECG anomaly detection, a case study on ECGs from MIT-BIH arrhythmia database (Goldberger et al 2000) is carried out. To provide a reference for comparison, BFDD and AWDD are also applied to the same data.

The remainder of this chapter is organized as follows. In the second section, BFDD and AWDD are briefly reviewed. The third section presents the proposed anomaly detection method. In the fourth section, the newly defined anomalies detection method and existing approaches are applied to a series of ECGs, and some comparative analysis results are reported. Finally, this chapter is briefly summarized in the fifth section.

4.2 Basic Notion and Related Works

BFDD and AWDD for anomaly detection, as the least and most successful nearest-neighbour based time series anomaly detection methods, have been shown to achieve correct results. In this section, because the basic idea behind our proposed method is also nearest-neighbour based anomaly detection, basic notion (non-self match) is described firstly in this part, and then BFDD and AWDD are reviewed.

4.2.1 Non-Self Match

Given one time series A and one of its subsequent B beginning at position P, in general, the beginning points of the best matches to B (apart from itself) should be $P \pm 1$ or $P \pm 2$. Therefore, excluding unnecessary matches is an important step prior to detecting anomalies. Otherwise, distance between candidate subsequence and its corresponding best match will lower than a pre-obtained threshold and it is impossible to find anomalies. In the research constructed by Keogh et al (2005), one matching definition, called as *Non-Self Match*, was introduced to remove trivial matches in the process of anomaly detection.

Given one time series T and its two sub-sequences $T1$ and $T2$, the beginning points of $T1$ and $T2$ are P and Q, and the length of both $T1$ and $T2$ are set as n, $T2$ can be defined as a non-self match to $T1$ if the position distance greater or equal to n ($|P - Q| \geq n$). As an example shown in below:

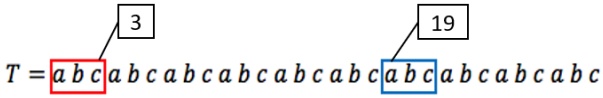


Figure 4.2 One time series contains two non-self match sub-sequences

where $T1$ and $T2$ are labelled by red box and blue box respectively, the length values of them are set as 3, the beginning point of $T1$ is 1 and the beginning point of $T2$ is 19. In this case, $T2$ can be defined as a non-self match to $T1$ because $|19 - 1| \geq 3$. On the contrary, another example shown in below describes that $T2$ cannot be defined as a non-self match to $T1$.

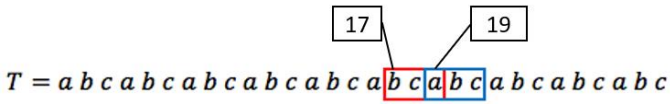


Figure 4.3 One time series contains two subsequences that cannot be defined as non-self match

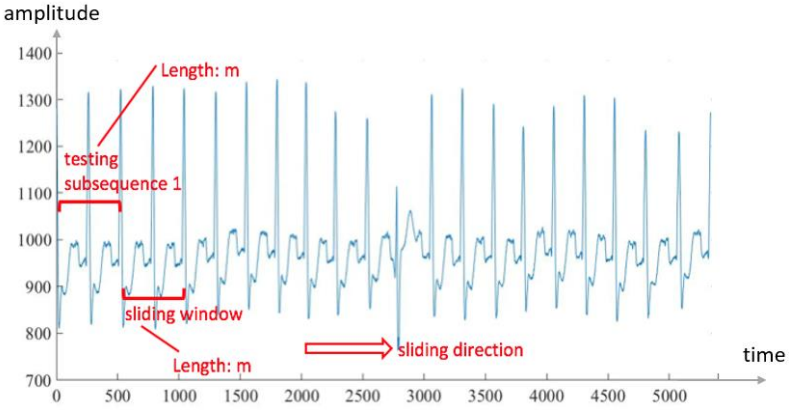
All the settings in Figure 4.3 are the same as those in Figure 4.2. The only difference is that the beginning points of $T1$ and $T2$ are 17 and 19 respectively. It can be noticed that $T1$ and $T2$ are partly folded together. For the position distance, it is $|19 - 17| = 2$ and hence lower than the length of sliding window.

Non-self match sub-sequences of one segment cannot be directly used to define whether the corresponding candidate is anomalous or not. For BFDD and AWDD, the

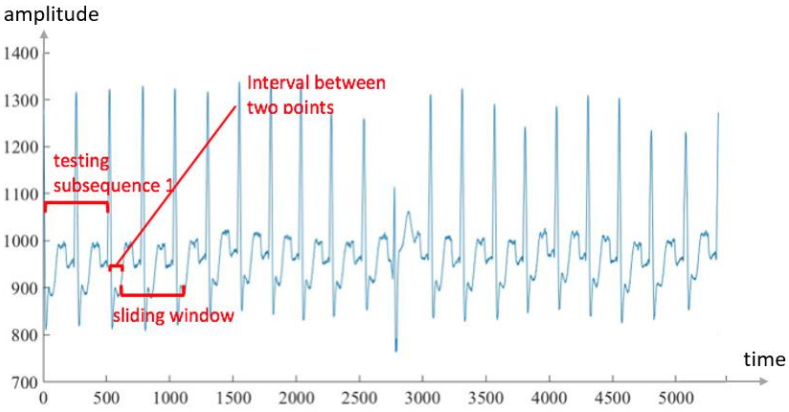
distances between the candidate and each of its non-self matches are calculated firstly, and then the minimum distance is recorded and the corresponding subsequence is defined as the nearest non-self match of the candidate. Given one time series T and one candidate A, the distances between A and all its non-self matches are calculated and recorded as $D = [d1, d2, \dots, dn]$. The nearest non-self match distance is the minimum value in D and the corresponding segment is the nearest non-self match of A. Once the nearest non-self match distance of one candidate is computed, it will be compared with a pre-obtained threshold to identify whether the candidate is anomalous or not.

4.2.2 Brute Force Discord Discovery

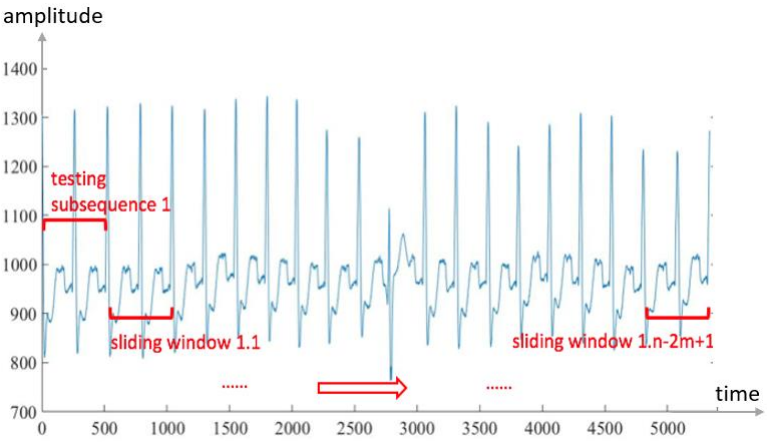
Brute force discord discovery (BFDD) algorithm was initially proposed by Keogh et al (2005) and Lin et al (2005), and the advantage of this algorithm is that it is easy to be understood and implemented. Based on BFDD, the implemented procedure of time series anomaly detection is as follows: 1) the first segment to be tested is the subsequence whose length is equal to that of a sliding window and its first point is the same with that of time series, as shown in Figure 4.4a and Figure 4.4b. 2) After defining testing subsequence, the next process is: sliding down the defined window on sample at a time and calculating the distance between the testing subsequence and the subsequence in the sliding window. This process can help us to find the nearest non-self match of the testing subsequence, as shown in Figure 2c. 3) Once the first nearest non-self match distance and it corresponding subsequence are recorded, the first point of the testing subsequence will move from the first point of the time series to the second point. Meanwhile, the length of testing subsequence is still equal to that of sliding window, as shown in Figure 2d. 4) For every testing subsequence with length m, and an original time series with length n, in order to find the nearest non-self match, it needs to calculate at least $(\mathrm{n} - 3\mathrm{m} + 1)$ times, and sometimes $(\mathrm{n} - 2\mathrm{m} + 1)$ times, as shown in Figure 2e. 5) According to the obtained nearest non-self match distance values, the anomalous segment can be detected via the comparison between the recorded values and a pre-obtained threshold, which is calculated by applying BFDD to the training time series. The mechanism of BFDD-based anomaly detection is illustrated by Figure 4.4.


amplitude
Length: m
testing
subsequence 1
sliding window
Length: m
sliding direction
time

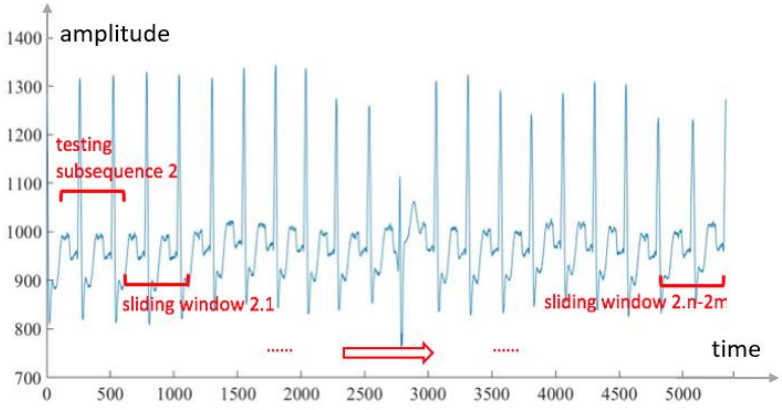
a


amplitude
Interval between
two points
testing
subsequence 1
sliding window
time

b


amplitude
testing
subsequence 1
sliding window 1.1
sliding window 1.n-2m+1
time

c


amplitude
testing
subsequence 2
sliding window 2.1
sliding window 2.n-2m
time

d

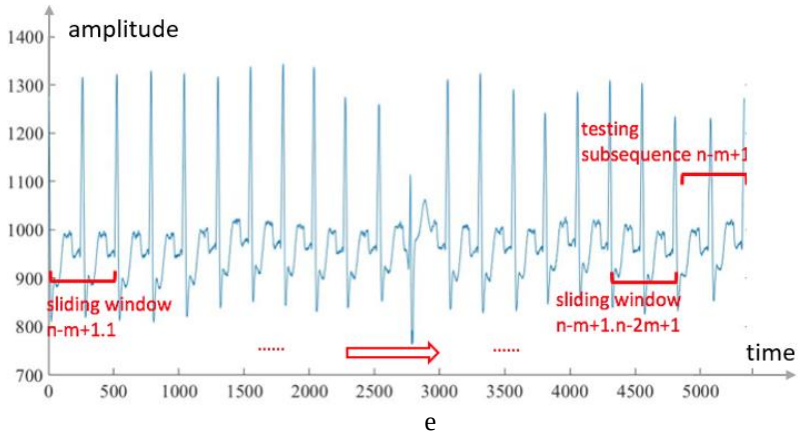


Figure 4.4 Mechanism of BFDD-based anomaly detection. a) defining the first testing subsequence and sliding window; b) sliding down the window on sample at a time; c) every testing subsequence has at least $n - 3m + 1$ non-self match distances; d) moving 1 unit backward of the testing subsequence; e) the testing subsequence keeps moving to the end of the time series.

The pseudo-code and procedure of BFDD-based anomaly detection is also described by Algorithm 4.1.

Algorithm 4.1 Brute Force Discord Discovery

Requirements: A time series: T
The length of sliding window: m
$n \leftarrow$ length of input time series
$best\_so\_far\_dist \leftarrow 0$
for i = 1 to n − m + 1 **do**
  nearest_neighbour_dist = infinity
  for j = 1 to n − m + 1 **do**
    if |i − j| ≥ n **do**
      if Dist((Ti, … , T(i + m − 1)), (Tj, … , T(j + m − 1))) <
        nearest_neighbour_dist **do**
        $\text{nearest}_{\text{neighbour}_{\text{dist(i)}}} =$
          Dist((Ti, … , T(i + m − 1)), (Tj, … , T(j + m − 1)))
        end if
  end for
  if nearest_neighbour_dist (i) > threshold **do**
    best_so_far_loc = i
  end if
end for

The inputs of Algorithm 4.1 include one time series and the length of sliding window. The output of this algorithm is the location of the anomaly.

For example, BFDD is applied to the ECG signal shown in Figure 4.1. The length of sliding window is 300, the corresponding nearest non-self matches distances are shown in Figure 4.5.

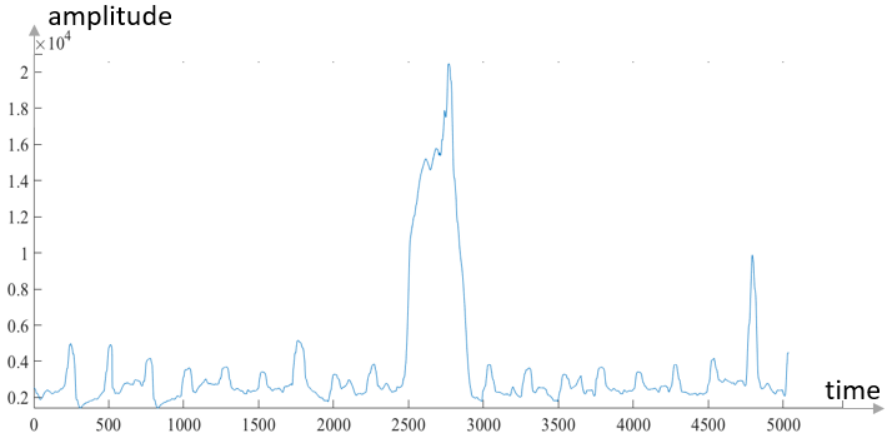


Figure 4.5 Nearest non-self match distances

Once the nearest non-self match distances are computed, all these distances will be compared with a pre-obtained threshold (obtained through applying BFDD to training dataset) and the corresponding segments are going to be identified. It is worth to mention the low computational efficiency of BFDD-based anomaly detection. It can be observed that there is a nested iteration in the calculation process. Every iteration has to calculate the distances between testing subsequence and its non-self sub-sequences at least $(n - 3m + 1)$ times (n is the length of time series and m is the length of sliding window). This will need a huge amount of time for even moderately large datasets. Assuming the length values of sliding window and testing subsequence are set as 300, the ECG signal shown in Figure 4.1 is a 15 seconds record and contains 5400 data points. The whole calculation process must be completed more than 20 million times (which is $(5400 - 300) \times (5400 - 300) > 20\text{million}$). A normal computer requires at least 50 seconds to finish the calculation.

Although BFDD-based time series anomaly detection is time consuming, there is one advantage need to be mentioned, that is universality, which means this method can be used to detect anomalies for various kinds of time series.

4.2.3 Adaptive Window Discord Discovery

For some special types of time series, the application of general anomaly methods may complicate the operation process and distort the calculation accuracy. In terms of ECG data, in order to overcome the heavy computational load involved in BFDD-based anomaly detection, adaptive window discord discovery (AWDD) was proposed by Chuah et al (2007). AWDD separates ECG into a number of segments based on the peak points,

then measures the distances between each segment and determines which subsequence can be treated as anomaly. It should be noted that the segments separated by peak points do not have the same length while the distance measure in AWDD is Euclidean distance. To solve this problem, if the length values of two candidates are different, the longer one should be compressed firstly so that its length is equal to that of the shorter one. In comparison with BFDD, the calculation time is significantly reduced through the application of AWDD without losing detection accuracy. The whole process of AWDD-based anomaly detection is described by Algorithm 4.2.

Algorithm 4.2 Adaptive Window Discord Discovery

Requirements: A time series: T
Location of peak points: P
$n \leftarrow$ length of input time series
$m \leftarrow$ number of peak points

for $i = 1: n - 1$ **do**;
  $outlength = P(i + 1) - P(i)$;
  for $j = 1: n - 1$ **do**;
    $innerlength = P(j + 1) - P(j)$;
  if $outlength > innerlength$ **do**;
    $B = imresize(A(P(i): P(i + 1)), [1, innerlength])$;
    $C = A(P(j): P(j + 1) - 1)$;
  else
    $B = imresize(A(P(j): P(j + 1)), [1, outlength])$;
    $C = A(P(i): P(i + 1) - 1)$;
  end if
  $ddd(j) = dist(B, C)$
end for
  $nearest\_neighbour\_dist(i) = min(ddd)$;
  if nearest_neighbour_dist (i) $>$ threshold **do**
  best_so_far_loc = i
  end if
end for

The inputs of Algorithm 4.2 include one ECG series and the locations of peak points. The output of this Algorithm is the location of anomaly.

AWDD compresses the longer subsequence so that its length is equal to that of the shorter one. This enables the use of Euclidean distance to measure the distance between two candidate sub-sequences. As an illustration, Figure 4.6a provides a simple example of two time series with different length values. In Figure 4.6b, the longer time series is compressed.

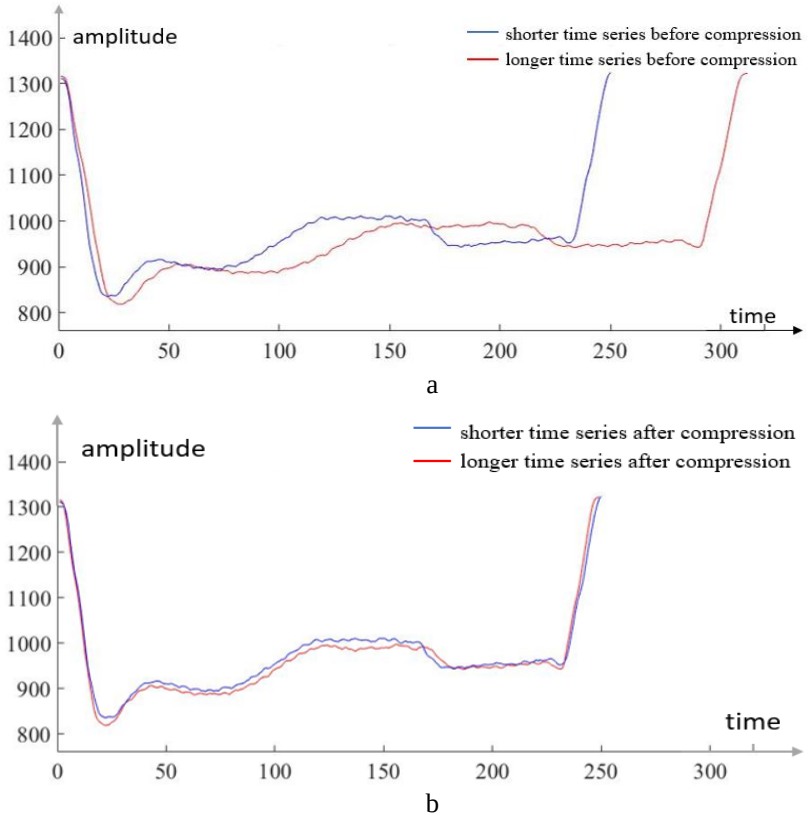


Figure 4.6 Two time series. a: before compression, b: after compression

AWDD is also applied to the ECG data in Figure 4.1. As there are 21 normal peak points, prior to distance measure, the ECG was separated into 20 segments. Then every segment is regarded as one testing subsequence and the corresponding nearest non-self match distances of these 20 segments are recorded, as shown in Figure 4.7.

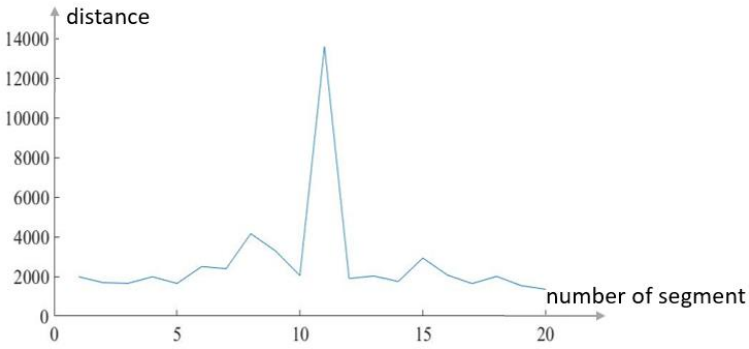


Figure 4.7 Nearest neighbour distance based on AWDD

As the result of BFDD-based anomaly detection shown in Figure 4.5 that the nearest non-self match distances are anomalous between 2500 to 3000, the location of abnormal point of nearest non-self match distances in Figure 4.7 is 11 and the interval between two normal points is almost 260. It is obvious that AWDD-based nearest non-self match distances reserve most information of BFDD-based nearest non-self match distances (can

correctly detect the anomalous segment). For AWDD improvement, the distances are calculated between every segment instead of sliding down the testing subsequence on one sample time. The completed procedure of AWDD-based anomaly detection is only $(20-1)^2$ times and for a normal computer only takes about 0.36 seconds.

4.3 Anomaly Detection of ECG Data

If there is only one disorder segment or several significantly different discorded segments in an ECG signal, both BFDD and AWDD can correctly detect the anomalous segment(s) while AWDD outperforms BFDD in term of computational efficiency, but both methods have two common drawbacks: i) Euclidean distance measure method may influence the accuracy of anomaly detection if timeline drift exits during the process of calculating the non-self match distances; ii) they cannot correctly detect the anomalies when there are two or more anomalous segments that are similar to each other. In this section, in order to improve the accuracy ECG anomaly detection, one modified distance measure method (MDTW) and one new anomalies detection method (ANMD) are proposed for ECG anomaly detection. They are as follows: 1) Modified dynamic time warping, called as MDTW, is presented to improve the accuracy of time series distance measure. 2) Average non-self match distance (ANMD) is proposed to replace nearest non-self match distance. 3) The analysis procedure using the proposed method for ECG anomaly detection is described.

4.3.1 Distance Measure

Traditional DTW-based distance measure (described in subsection 2.2.2) is used to directly calculate the distances between corresponding points, and the sum of them is used as final distance. In this chapter, the distance between two candidates is defined according to the DTW distance and the optimal align path (which is achieved by Algorithm 2.3 and Algorithm 2.4 (page 21)). Given two time series A and B, the distance between them is defined as follows:

$$\mathrm{Dist}(\mathrm{A},\mathrm{B}) = \mathrm{d} + \left(\frac{\mathrm{l}-\mathrm{la}}{\mathrm{la}}\right) * \left(\mathrm{sum}(\mathrm{Anew}) - \mathrm{sun}(\mathrm{A})\right) + \left(\frac{\mathrm{l}-\mathrm{lb}}{\mathrm{lb}}\right) * (\mathrm{sum}(\mathrm{Bnew}) - \mathrm{sum}(\mathrm{B})) \tag{4.1}$$

where $\mathrm{Dist}(\mathrm{A},\mathrm{B})$ represents the distance between A and B, d is the DTW distance between A and B, l is the length of the optimal align path; Anew and Bnew are two new

time series segments which are constructed according to A, B and the optimal align path; la and lb are the length values of Anew and Bnew; the function $\mathrm{sum}(\ldots)$ returns the sum of elements of the input segment. The whole process of this distance measure method is summarized by Algorithm 4.3.

Algorithm 4.3 Distance Calculation

Requirements: Two time series A and B

la ← length of A
lb ← length of B
distance ← DTWdistance (A, B)
optimal path ← DTWdrift (A, B)
wa ← first column of optimal path
wb ← second column of optimal path
l ← length of optimal path

for i = 1 to l **do**
Anew(i) = A(wa(i))
Bnew(i) = B(wb(i))
end for
final distance = distance + ((l − la)/la) ∗ (sum(Anew) − sum(A)) + ((l − lb)/lb) ∗ (sum(Bnew) − sun(B))

The inputs of this algorithm are two time series and the output is the distance between them.

To demonstrate the performance of the proposed method for time series similarity measure, two time series A and B are defined in Figure 4.8, and the proposed method, together with traditional dynamic time warping and Euclidean distance are applied to calculate the distance between A and B.

A 1 2 3 4 5 6 6 7 8 9 10 11 12 13 13

B 1 2 3 4 5 6 7 8 9 9 9 10 11 12 13

Figure 4.8 Template time series

The calculation matching image of Euclidean distance is shown in Figure 4.9 and the distance between A and B is 7. However, we can find that the subsequence from the 7th point to the 10th point in time series A is similar to the subsequence from the 6th point to the 9th point in time series B, but Euclidean distance directly calculates the distances between elements at same time point and sums them up as the distance between these two time series.

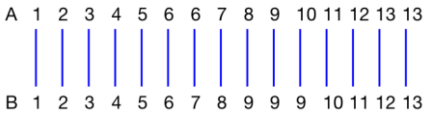


Figure 4.9 Matching image of distance calculation based on Euclidean distance

The matching image of using traditional DTW to measure the distance between A and B is shown in Figure 4.10. It can be noticed that the timeline has been warped and the most similar elements have aligned with each other. However, the final distance between these two time series is 0 although they are not identical.

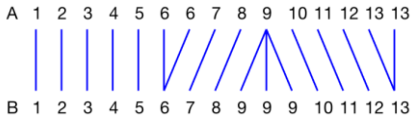


Figure 4.10 Matching image of distance calculation based on DTW

Traditional distance measures such as Euclidean distance and DTW do not work well for the above time series A and B. That is the motivation to propose the new method (Algorithm 4.3) to overcome the disadvantage of traditional distance measures. In Algorithm 4.3, in order to eliminate the impact of the neglect of timeline drift, the distance between two candidates is defined according to three variables: DTW distance, optimal align path between two time series, and the sum of distances between the extended new points and base point (these new points exist when there is timeline drift between two candidates, and vice versa). Compared with the results obtained based on Euclidean distance and traditional DTW, the distance between A and B computed based on Algorithm 4.3 is 4.9333.

4.3.2 Non-self Match Average Distance

To overcome the drawback of BFDD and AWDD whereby they can only work well for anomaly detection when all the anomalies in time series of interest are significantly different from each other, this subsection proposes a new calculation, namely, average non-self match distance (ANMD).

Given one time series T, one of its segment is A, non-self matches of A in T are stored in $A_m = [A_1, A_2, \dots, A_n]$, and the distances between A and all its non-self matches are obtained through the application of the proposed distance measure method and recorded in $D = [d_1, d_2, \dots, d_n]$. In terms of anomaly detection based on BFDD and AWDD, the minimum value in D is recorded as nearest non-self match distance and used

to identify whether A is anomaly or not. Different from nearest non-self match, average value of D is recorded and used in ANMD to identify whether A is anomaly or not.

In order to clearly state the advantage of the proposed method in anomalies detection of multi-anomalous time series, a time series contains two same anomalies is defined in Figure 4.11, which is constructed through repeat the time series in Figure 4.1.

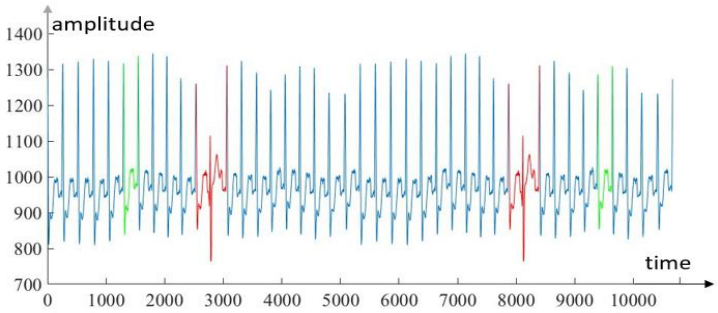


Figure 4.11 Two-anomaly time series

Two normal segments and two anomalous segments (highlighted by green and red in Figure 4.11) are extracted as testing segments, both nearest non-self match and average non-self match distance are applied to these four segments. Table 4.1 illustrates the values that are used to identify whether the input segments are anomalous or not.

Table 4.1 Values Used for Anomalies Identification

	Normal 1	Normal 2	Anomaly 1	Anomaly 2
Nearest Non-self Match Distance	0	0	0	0
Average Non-self Match Distance	1.9730×10^5	1.8508×10^5	6.6383×10^5	6.6383×10^5

In Table 4.1, the second row states the distances between the extracted segments and their corresponding nearest non-self match segments. The third row illustrates the average values of distances between extracted segments and all their corresponding non-self match segments. The values in Table 4.1 show that anomaly detection methods based on nearest non-self match cannot correctly detect anomalies in some special conditions, while our proposed notion is helpful to correctly detect all the anomalous segments.

4.3.3 Anomaly Detection of ECG Data

The analysis procedure using the proposed methods for ECG anomalies detection is as follows: 1) separate the input ECG into several segments based on peak points; 2) transform every segment to a symbolic series; 3) define the anomalies using ANMD.

4.3.3.1 Peak Points Collection

It is known that ECG can be defined as periodical time series because ECG derives from regular heart muscle beat. Because of this, peak points based ECG segmentation is applied prior to distance measure. Algorithm 4.4 briefly describes the procedure of peak points collection.

Algorithm 4.4 Peak Points Collection

Requirements: An ECG signal: T
A defined value: h
$n \leftarrow$ length of input ECG signal
$m \leftarrow 1$
for $i = 1$ to n **do**
if $T(i) \geq h$ **do**
$location(m) = i$
$m \leftarrow m + 1$
end if
end for

The inputs of Algorithm 4.4 include one ECG signal and one threshold. This threshold is obtained through training the available ECG data. The output is a vector containing locations of peak points.

As an example, Algorithm 4.4 is applied to the ECG signal shown in Figure 4.1, and the detected peak points are highlighted by red dots and shown in Figure 4.12.

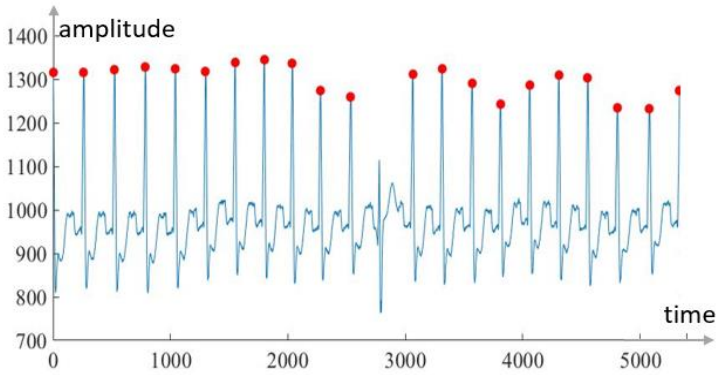


Figure 4.12 Peak points collection

4.3.3.2 Transformation

For a cardiac patient, it is necessary to go to hospital to periodically for examinations, and sometimes the examination may take a long time. Mathematically speaking, an ECG carries out over many hours contains a huge amount of data. The improved symbolic representation and distance measure method, which was introduced in chapter 3, has

proved that it can not only reserve important original data information, but also reduce the dimension of original data. In this part, prior to the analysis of original ECG data, the original data is processed based on symbolic representation.

The application of normal time series representation method is to represent the whole input time series by a low-dimensional time series. In this work, because the ECG is periodical time series, the segment between two peak points is treated as one subsequence and every subsequence is processed based on symbolic representation. Take the first subsequence in the ECG signal in Figure 4.1 as example. The length of each segment is 10 and the number of break points is 9. The representation is shown in Figure 4.13, the ECG time series is transformed into PAA format (the red line), and then represented by symbolic series, which is 'AHJIHHIHHGFEEEEEEEFGGFFFEAA'.

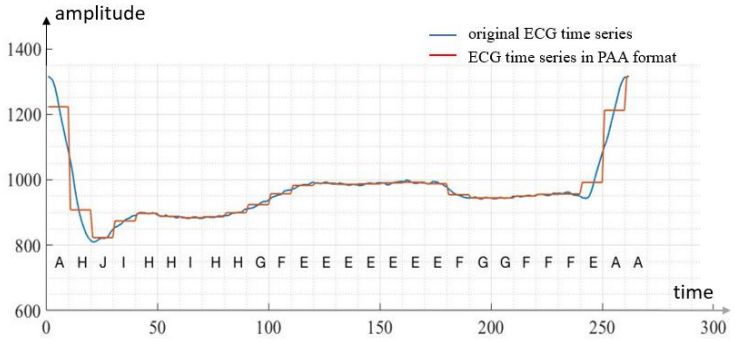


Figure 4.13 Symbolic representation of one ECG subsequence

Through the application of symbolic representation, the length of the new series is 27, while the length of original subsequence is 262. It is obvious that the pre-processing of input ECG data makes great contribution in reducing further analysis workload.

4.3.3.3 Anomaly Detection

As mentioned in the introduction, anomalies are patterns in data that do not conform to a well-defined notion of normal behaviour. At the beginning of anomaly detection, a criteria has to be defined and it can be obtained through applying the proposed anomaly detection method to available training data. For example, ANMDs of all normal segments are stored in D_n, ANMDs of all anomalous segments are stored in D_a. The threshold is always defined as the average value of the minimum value in D_a and the maximum value in D_n.

Once the threshold is obtained, the ANMDs of all the segments in testing signal have to be computed and compared with the threshold. If the value greater than the threshold,

the corresponding segment is defined as anomalous. On the contrary, if the value is lower than the threshold, the corresponding segment is defined as normal. Figure 4.14 shows the average non-self distances of all the segments in ECG signal in Figure 4.1, as the threshold is 1.8×10^4 (which is obtained through applying the proposed method to 10 training ECG signals). The ANMD of the 11st segment is greater than the threshold. Hence the 11st segment is an anomaly and the others are normal segments.

It can be noticed that the average non-self match value of the anomalous segment in Figure 4.14 is significantly greater than the values of normal segments, and the corresponding segment can be defined as anomalous directly without the comparison with the threshold. It should also be noted this is a special condition. In some common cases, comparing with threshold is the best way to correctly detect the anomalous segments.

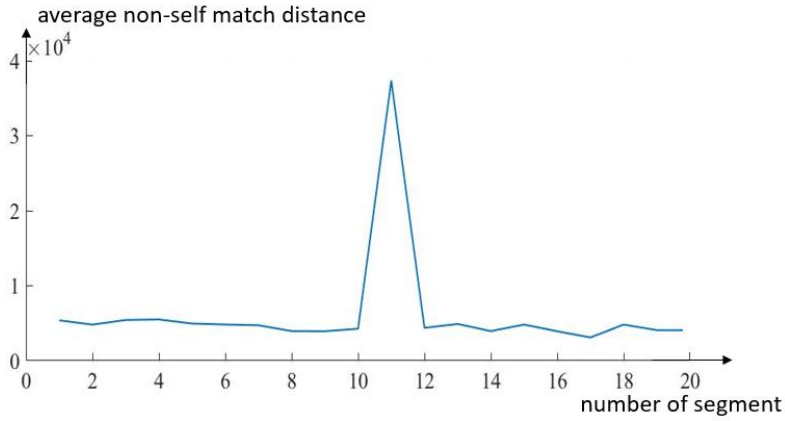


Figure 4.14 Average non-self match distances of time series in Figure 4.1

4.4 Experimental Comparison

In order to validate the performance of the proposed approach, it is applied to ECG signals obtained from public database (MIT-BIH arrhythmia database) (Goldberger et al 2000). The procedure of using the proposed method for anomaly detection from ECG can be separated into 3 steps: 1) compute a threshold by applying the anomaly detection algorithm to the training ECG database, 2) calculate the ANMD of every segment in testing ECG, 3) test whether the ECGs contains anomalous segments. For a comparison, BFDD and AWDD are also applied to the same ECG data.

4.4.1 ECG Database

The resource of ECGs (30 ECGs in total) included in the MIT-BIH database is a set of over 4000 long-term Holter recordings that were obtained by Beth Israel Hospital

Arrhythmia Laboratory from 1975 to 1979. Approximately 60% of the recordings were obtained from inpatients (Goldberger et al 2000). As a result, testing performance of ECG anomaly detection algorithms based on this database has strong conviction. The experimental database used in this study contains 30 ECGs. 10 of them are used as training database to obtain the threshold and the remaining 20 ECGs are used as testing database.

The 30 ECGs used in the study are listed in Table 4.2, where the 1st column is an index of the 10 training ECGs and 20 testing ECGs. The 2nd column illustrates the starting and ending time of corresponding ECG signal. The 3rd column is a location index to show where the anomaly occurs, and the 4th column is an indication of whether there is an anomaly or anomalies in the corresponding ECG or not. The first 5 training ECGs contain no anomalous segment whereas each of the last 5 training ECGs contains one anomalous segment. In testing dataset, in order to demonstrate the ability of the proposed method for detecting multiple anomalous ECGs, the 20 test ECGs were chosen as follows: No. 11-15 contain no anomaly, No. 16-20 contain one anomaly each, No. 21-35 contain 2 significantly different anomalies, and No. 26-30 are constructed through repeating one-anomalous segment and each of them contains 2 same anomalies.

Table 4.2 ECG Excerpts from MIT-BIH Record 109

Training	Start-end points	Anomaly Location	Anomaly Identification
1	140s-180s	NA	NO
2	440s-480s	NA	NO
3	560s-600s	NA	NO
4	700s-740s	NA	NO
5	740s-780s	NA	NO
6	20s-60s	6758	YES
7	80s-120s	4474	YES
8	200s-240s	12890	YES
9	260s-300s	5690	YES
10	520s-560s	3205	YES
Testing	Start-end points	Anomaly Location	Anomaly
11	860s-900s	NA	NO
12	940s-980s	NA	NO
13	980s-1020s	NA	NO

14	1180s-1220s	NA	NO
15	1220s-1260s	NA	NO
16	620s-660s	11630	YES
17	660s-700s	7928	YES
18	820s-860s	9410	YES
19	1300s-1340s	6100	YES
20	1400s-1440s	7170	YES
21	900s-940s	6270,11590	YES
22	1060s-1100s	10320,12920	YES
23	1100s-1140s	2412,11970	YES
24	1140s-1180s	966,9957	YES
25	500s-540s	3655,10410	YES
26		11630, 26030	YES
27		7928, 22328	YES
28		9410, 23810	YES
29		6100, 20500	YES
30		7170, 21570	YES

4.4.2 BFDD Based Anomaly Detection

BFDD is applied to training ECGs (No. 1-10 in Table 4.2) to calculate the threshold. The nearest non-self match distances of normal segments and anomalous segments are shown in Table 4.3

Table 4.3 Threshold Calculation based on BFDD

ECG	Nearest Non-self Match Distances	Anomaly
1	5102	NO
2	4886	NO
3	9206	NO
4	5582	NO
5	6056	NO
6	21171	YES
7	22469	YES
8	16947	YES
9	9996	YES
10	22162	YES

As mentioned in related works in Section 4.2, nearest neighbour distance of candidate is used to define whether the corresponding segment is anomalous. On the basis of the obtained values, the threshold has to clearly state whether the segment is an anomaly. Table 4.3 shows that the maximum value in the second column in relation to the normal segments is 9206, and the minimum value in the second column in relation to the anomalous segments is 9996. As a consequence, the threshold is defined as the average value of 9996 and 9206, which is 9601.

With the threshold, BFDD-based anomaly detection is applied to the testing ECGs. The first step is to calculate and record the nearest non-self match distance of every sliding window, and then compare the recorded values with the threshold to identify the corresponding segment. Table 4.4 shows the anomaly detection results of the 20 testing ECGs.

In table 4.4, the 2nd column shows the location or locations of the detected anomaly or anomalies, and the 3rd column illustrates the nearest non-self match distance or distances that greater than the threshold. The 4th column describes the results of anomaly detection and the last column lists the calculation time used by BFDD. It can be seen that BFDD can correctly define that the ECG is normal or anomalous when there is no anomaly or only one anomaly in one ECG. When there are two significantly different anomalous segments in testing ECG, BFDD can also detect the presences of anomalies, but the accuracy is only 40%. What is worse is when there are two same or similar anomalies, BFDD cannot detect any of them. In terms of computation complexity, BFDD-based anomaly detection is not acceptable, as the length of testing ECG only contains the records in 40 seconds, and the calculation time is over 450 seconds.

Table 4.4 Anomaly Detection based on BFDD

ECG	Detected Location	Nearest Non-self Match Distance	Anomaly Identification	Operation Time
11	NA	6093	NO	461.8
12	NA	7299	NO	458.6
13	NA	5351	NO	461.9
14	NA	5634	NO	459.2
15	NA	3831	NO	456.3
16	11673	24023	YES	458.8
17	7942	22513	YES	461.8
18	9421	17365	YES	455.1

19	6106	23793	YES	459.6
20	7178	20911	YES	460.7
21	6286, 11611	21261, 19876	YES	459.1
22	6750, 7659	8201, 6239	YES	457.8
23	4679, 7730	7299, 9206	YES	461.9
24	971, 9960	18481, 17365	YES	457.8
25	6089, 9409	6285, 6250	YES	460.7
26	NA	0	NO	1719.8
27	NA	0	NO	1843.5
28	NA	0	NO	1843.8
29	NA	0	NO	1855.1
30	NA	0	NO	1863.9

4.4.3 AWDD Based Anomaly Detection

AWDD-based anomaly detection is applied to the same ECG data. Based on Algorithm 4.2, the first step is the same with that of BFDD-based anomaly detection, which is to define the threshold through applying AWDD to training dataset. In this part, the threshold for ECG anomaly detection is 8325. The results of applying AWDD to training ECGs are shown in Table 4.5.

Table 4.5 Threshold Calculation based on AWDD

ECG	Maximum Nearest Non-self Match Distances	Anomaly
1	2326	NO
2	1960	NO
3	2636	NO
4	2133	NO
5	3068	NO
6	1433	YES
7	15990	YES
8	13583	YES
9	13830	YES
10	15164	YES

Once the threshold is known, AWDD is then applied to the 20 testing ECGs. Table 4.6 shows the results of the AWDD-based anomaly detection.

Table 4.6 Anomaly Detection based on AWDD

ECG	Detected Location	Maximum Nearest Non-self Match Distance	Anomaly Identification	Operation Time
11	NA	2224	NO	1.9752

12	NA	2006	NO	1.8634
13	NA	3670	NO	1.7463
14	NA	1873	NO	1.8055
15	NA	1630	NO	1.9323
16	11440	15650	YES	2.1261
17	7800	15346	YES	1.9347
18	9360	13206	YES	1.9914
19	5880	15263	YES	1.7558
20	7020	14696	YES	1.9283
21	6240, 11440	13008, 12578	YES	1.9469
22	10140, 12740	15788, 15678	YES	1.9222
23	2340, 11960	11520, 13250	YES	2.2854
24	780, 9800	14967, 12191	YES	1.8531
25	3640, 10400	8542, 10254	YES	1.8206
26	NA	0	NO	6.9522
27	NA	0	NO	7.0700
28	NA	0	NO	7.3153
29	NA	0	NO	7.8347
30	NA	0	NO	7.3965

Table 4.6 shows that AWDD can correctly tell the normal ECGs (No. 11-15). For testing ECGs (No. 16-20), AWDD can also correctly detect anomalous segments and identify their corresponding location. When there are two different anomalies in testing ECGs (No. 21-25), AWDD can detect the existences of all the anomalies, which outperforms the results of BFDD-based anomaly detection, while for the remaining 5 testing ECGs (No. 26-30), the results are same with BFDD-based anomaly detection and no anomaly is detected. One improvement needing to be mentioned is that the whole process of anomaly detection for every ECGs only takes about1.5 seconds. To summarise, AWDD is more trustworthy and efficient when compared with BFDD in terms of ECG anomaly detection.

4.4.4 Proposed Method Based Anomaly Detection

The whole process of the proposed method based ECG anomaly detection is similar with that of BFDD and AWDD. Specifically, the first step is to compute a threshold through training the available ECGs. The second step is to calculate ANMDs of testing ECGs and identify the testing ECGs through comparing the distances with the threshold. The results generated by applying the proposed method to training ECGs are shown in Table 4.7.

Table 4.7 Threshold Calculation based on Proposed Method

ECG	Average Non-self Match Distances	Anomaly
1	5308	NO
2	5531	NO
3	5832	NO
4	7012	NO
5	5246	NO
6	32031	YES
7	30694	YES
8	30120	YES
9	35691	YES
10	29098	YES

As shown in Table 4.7, a threshold can be computed to allow us to define whether the testing segment is an anomaly. For example, the threshold, defined as the mean value between maximum ANMD of non-anomalous ECGs and minimum ANMD of the anomalous ECGs, is 18055. With the threshold, the new method is applied to testing ECGs. The results are shown in Table 4.8.

Table 4.8 Anomaly Detection based on Proposed Method

ECG	Detected Location	Average Non-self Match Distance	Anomaly Identification	Operation Time
11	NA	4143	NO	3.6727
12	NA	7199	NO	3.3580
13	NA	4865	NO	3.1531
14	NA	4040	NO	3.3303
15	NA	5149	NO	3.1878
16	11440	34477	YES	3.3558
17	7800	33981	YES	3.1755
18	9360	32119	YES	3.3403
19	5880	36618	YES	3.4315
20	7020	41718	YES	3.3230
21	6240, 11440	27768, 26318	YES	3.3121
22	10140, 12740	32760, 31071	YES	3.2435
23	2340, 11960	38546, 34453	YES	3.3711
24	780, 9800	35035, 28146	YES	3.1587
25	3640, 10400	37826, 36183	YES	3.2899
26	10400, 26000	34527, 34527	YES	12.5846
27	7800, 22100	33916, 33916	YES	12.3937
28	9360, 23660	31946, 31946	YES	12.8550

29	5880, 20280	36730, 36730	YES	12.5591
30	7020, 21580	41759, 41759	YES	12.8025

From Table 4.8, it is clear that the new method has the ability to detect all the anomalies when there are more than 1 similar or same anomalies in an ECG. For the 5 ECGs that contain 2 different anomalies and the 5 ECGs contain 1 anomalous segment, this new method can correctly detect the existence or existences of anomaly or anomalies. For the remaninig non-anomalous ECGs, this new method can also correctly identify that they are normal.

As shown in Table 4.4, Table 4.6 and Table 4.8, BFDD and AWDD cannot detect anomalies when there are two or more similar or same anomalies in one ECG, while the proposed method can correctly detect all the anomalies. For ECG containing two or more significantly different anomalies, BFDD-based anomaly detection has the accuracy of 40%, while the proposed method and AWDD has the accuracy of 100%. For ECG only containing one anomalous segment and non-anomalous ECG, these three methods work well, with an accuracy rate of 100% for all of them. In terms of computation complexity, BFDD anomaly takes over 460 seconds while AWDD and the proposed method only take 1.5 seconds and 2.8 seconds respectively. In summary, the proposed methods provide a promising improvement in terms of the accuracy of anomalies from ECG signals. The overall performances of the three methods are briefly summarized in Table 4.9.

Table 4.9 Anomaly Detection Accuracy Comparison

	2 or more anomalies similar or same with each other	2 or more anomalies significantly different from each other	1 anomaly ECG	Non-anomalous ECG
New Method	100%	100%	100%	100%
BFDD (Keogh et al 2005)	0	40%	100%	100%
AWDD (Chuah and Fu 2007)	0	100%	100%	100%

4.5 Summary

Given the fact that cardiovascular disease has been a focus in society and clinical fields for ages, we believe that the application of data mining method to ECG anomaly detection will make a great contribution to the heart disease detection. With the nature of

fast calculation and high accuracy, data mining methods are helpful for patients to get fast and accurate treatment.

In this chapter, we proposed an ECG anomaly detection method based on a new distance measure method (MDTW) and a new anomaly detection calculation (ANMD). For the proposed distance measure method (MDTW), with the purpose of eliminating the error caused by existence of timeline drift and removing the error caused by neglect of timeline drift, the distance between two candidates is calculated according to their DTW distance and the optimal align path between them. For the new anomaly detection calculation (ANMD), in order to correctly detect all the anomalies in one time series, the average value of non-self match distance is used to replace the minimum value of non-self match distances. Through applying the new method and the other two famous anomaly detection methods to 30 actual ECGs, experimental results show that the proposed method is promising in terms of calculation complexity and outperforms the two compared methods regarding the accuracy of anomalies detection.

Chapter 5

Automatic Time Series Clustering

In this chapter, we present an automatic time series clustering method, called AT-means. AT-means can automatically carry out clustering for a given time series dataset, from setting the initial centers to the determination of number of clusters and generation of clusters. The performance of AT-means is tested on 10 benchmark time series datasets obtained from the UCR database. For comparison, the K-means method with 3 different conditions are also applied to the same datasets. The experimental results show that the proposed method significantly outperforms the compared K-means approaches.

5.1 Introduction

Clustering is a data mining technique where similar data are placed into related or homogeneous groups without having prior knowledge of groups' definition (Aghabozorgi et al. 2015; Rai and Singh 2010). In detail, the purpose of clustering is to identify the structure of an unlabelled database by objectively organizing data into different groups, where the within-group-object similarity is minimized and meanwhile the between-group-object dissimilarity is maximized (Liao 2005). Through the application of clustering, some hidden features in the original dataset can be found, which is helpful for future analysis. For example, clustering approach plays an important role in image segmentation (Zheng et al. 2015; Choy et al. 2017), feature selection (Song et al. 2013; Sotoca and Pla 2010), outlier detection (Duan et al. 2009; Pamula et al. 2011). To date, clustering techniques have been extensively studied and applied in a wide range of fields, ranging from information processing, medical sciences, to earth sciences (Xu and Wunsch 2005; Hansen and Jaumard 1997).

Chapter 5. Automatic Time Series Clustering

A special type of clustering is time series clustering (Aghabozorgi et al. 2015). In the last few decades, clustering of time series has received significant attention from different aspects. Example include rule discovery (Fu et al. 2001; Harms et al. 2002), summarization (Appice et al. 2015; Mampaey and Vreeken 2013) and prediction (Chaouch 2014; Chen and Tanuwijaya 2011), not only because time series clustering can discover valuable patterns from time series datasets, but also save a lot of unnecessary work and time because the analysis of a large dataset can be achieved by analysing a relatively smaller structured dataset with the facilitation of clustering techniques. In the reviews of time series clustering in the last few decades (Aghabozorgi et al. 2015; Liao 2005), it was introduced that time series clustering methods can be classified into six groups: partitioning, hierarchical, grid-based, model-based, density-based and multi-step clustering. Among these groups, the most commonly used and easily understood algorithms are partitioning clustering and hierarchical clustering. For partitioning clustering, such as K-means (MacQueen 1967), K-medoids (Kaufman et al. 2009) and Fuzzy c-means (Bezdek 1981), it makes k groups from n unlabelled objects in the way that each group contains at least one object. For hierarchical clustering, for example Chameleon (Karypis 1999), CURE (Guha 1998) and BRIRCH (Zhang et al. 1996), it offers a way to build a hierarchically structured tree according to the similarity between different time series. More detailed description of time series clustering algorithms can be found in a review of time series clustering (Aghabozorgi et al. 2015).

Partitioning-based clustering algorithms may have been the most widely used time series clustering algorithms during most recent few decades, meanwhile, partitioning-based clustering algorithms have been extended in many different ways (Bezdek 2013; Eschrich et al. 2003; Kaufman and Rousseeuw 2009). However, partitioning-based time series clustering methods, for example the traditional K-means, have three major shortcomings: 1) a gradient descent algorithm is often incorporated into the partitioning procedure. This can make the clustering highly sensitive to the initial placement of the cluster centres (Celebi et al. 2013). 2) Most partitioning clustering methods are also sensitive to the value of "means" and there is currently not a best method to calculate the average sequence of a set of time series sequences (Wu et al. 2008). 3) The number of clusters, k, has to be pre-assigned, this is not applicable or feasible for many applications (Aghabozorgi et al. 2015; Antunes 2001; Wang et al. 2006). In order to overcome these shortcomings, some solutions have been proposed in the last few decades. For the first

drawback, numerous initialization methods have been proposed to address this problem (Celebi et al. 2013), such as K-means++ (Arthurand and Vassilvitskii 2007), ROBust INitiallization (Al Hasan et al. 2009) and global K-means method (Likas et al. 2003). Nevertheless, these methods still randomly select the initial points from the dataset by imposing some constraints. The second disadvantage is that some time series averaging methods have been proposed, such as nonlinear alignment and averaging filters (Gupta et al. 1996), prioritized shape averaging (Niennattrakul and Ratanamahatana 2009) and dynamic time warping barycentre averaging (Petijean et al. 2011). Although these methods can improve the accuracy of time series average calculation, they are still not effective enough because they are sensitive to the paring orders or presences of outliers. For the third weakness, several methods were developed to optimally find the number of clusters (Hancer and Karaboga 2017), such as gap statistics (Tibshirani et al. 2001), weighted gap statistics (Yan and Ye 2007), X-means (Pelleg and Moore 2000) and MACE-means (Shahbaba and Beheshti 2014). However, all of these methods cannot be directly applied to time series because the values of time series change as a function of time.

In this chapter, we propose an automatic time series clustering method, called AT-means, aiming at overcoming the disadvantages of the aforementioned methods and improving the performance of time series clustering. The main contributions of this work are threefold. First, a modified global time series averaging method, called as initialised weighted global time series averaging (IWGTA), is proposed to correctly calculate the average sequence of a set of time series. Second, we develop an initial centre sequence determination method, called average initial centre determination (AID), depending on which initial centres will be located in proper areas. Third, a novel elbow point extraction method, called dual weight average distance (DWAD), is introduced and used to determine the number of clusters. To demonstrate the performance of the proposed method for time series clustering, AT-means is applied to 10 benchmark time series datasets obtained from UCR time series collection (Chen et al. 2015). To provide a reference comparison, the performance of AT-means is compared with K-means under 3 different conditions, they are: i) number of clusters is known; ii) number of clusters is known and initial centre setting method is applied; iii) number of clusters is known and the proposed global averaging method is applied. Comparison results show that AT-means outperforms the other three methods in terms of time series clustering.

This Chapter is structured as follows: in Section 2, previously published methods for clustering number determination and global time series average calculation are reviewed; Section 3 illustrates time series sequences clustering based on the proposed method; and Section 4 reports the experimental results; finally, Section 5 concludes this Chapter.

5.2 Related Works

In order to extend K-means to a totally unsupervised time-series clustering technique, one important task is to estimate the proper number of clusters. The other one is to calculate the average time series of a cluster of the time series. In the last few decades, many methods have been proposed to solve these two problems. This section briefly reviews two groups of most commonly used methods: cluster number determination and global time series averaging.

5.2.1 Determination of Cluster Number

In recent years, many methods have been proposed for determining the number of clusters, among which gap statistics (Tibshirani et al 2001) and X-Means (Pelleg and Moore 2000) are two popular and representative methods, which are summarized below.

5.2.1.1 Gap Statistics

Gap statistics is a useful method for determining the proper number of clusters for a time series by comparing the observed weight curve with a reference weight curve (Tibshirani et al 2001). It generates a sample of data representing the observed data, and then calculate the gap between the labelled cluster and reference distribution. The implementation of gap statistics based cluster number estimation is divided into three steps:

- Set a range of k, such as $k_{min} = 1$ and $k_{max} = K$, and then calculate the within-dispersion measures W_k, where $k = 1, 2, \dots, K$.
- Generate a reference dataset and calculate the gap. The gap is defined using the following equation:

$$Gap(k) = E_n^*\{log(W_{kb})\} - log(W_k) \tag{5.1}$$

  where W_k denotes the sum of the pairwise distances of all the points in a cluster in the original dataset, W_{kb} states the sum of the pairwise distances of all the points in

a cluster in reference dataset, E_n^* means the expectation under a sample of size n from the reference distribution.

- Choose the number of clusters according to following equation:

$$\hat{k} = \text{smallest } k \text{ such that } Gap(k) \geq Gap(k+1) - S_{k+1} \tag{5.2}$$

where S_{k+1} means the standard deviation of within-dispersion measures of the reference dataset.

Note that the gap statistics method has two drawbacks that need to be taken into consideration: (i) computationally expensive and (ii) not easy to implement for time series clustering. For the first shortcoming, the upper bound of k has to be big enough in order to guarantee that the actual cluster number is smaller than the pre-specified upper bound, and therefore the whole process is time-consuming if the upper bound is specified too large. For the second shortcoming, the reference time series dataset is not easy to be generated due to the characteristics of time series, such as frequency, amplitude, period and length.

During the last 10 years, some methods have been proposed to overcome the shortcomings and improve the performance of traditional gap statistics. For example, weighted values are used to reduce the influence of points that are far away from the cluster centre (Yan and Ye 2007). However, the problems mentioned above (computational load and unsuitability for time series) are still unsolved.

5.2.1.2 X-means

As a straightforward extension of the K-means algorithm, X-means applies the Bayesian Information Criterion (BIC) to the splitting process (Pelleg and Moore 2000; Hancer and karaboga 2017). In essence, the algorithm starts with a small k (equal to the lower bound of the given range) and continues adding centroids until the value of k reached the upper bound or the BIC value of children clusters is smaller than that of the parent clusters. The procedure of using X-means to determine the number of cluster can be divided into three steps:

- Randomly locate two points as initial centres and separate the remaining points into two clusters according to the distances between points to centres, then replace

previous centres by mean value of each cluster and stop the calculation when the centroid stays fixed.

- Set $k = 2$ (or another small number) and apply k-means to the groups that are obtained from the previous step.
- Compare the BIC value of children clusters and parent cluster. If $BIC(k+1) < BIC(k)$, stop the calculation; if $BIC(k+1) > BIC(k)$, continue the calculation. In case, there is no k that satisfies the stop requirement ($BIC(k+1 < BIC(k))$ for any k in the range, the calculation has to stop when k reaches the upper bound.

It is noteworthy that X-Means works well only for cases where there is plenty of data and well separable spherical clusters. In most recent years, G-means (Hamerly and Elkan 2004), PG-means (Feng and Hamerly 2007) and GX-means (Vatsavai et al. 2011) were proposed to improve the performance of X-means, but each of them cannot work properly if the cluster does not obey the Gaussian distribution or is uniformly distributed (Hancer and Karaboga 2017).

5.2.2 Global Time Series Averaging

It is known that many distance-based clustering methods, such as partitioning clustering and hierarchical clustering, require an averaging scheme and the performance of these methods highly depends on the quality of the averaging scheme adopted (Petijean et al. 2011). This subsection briefly reviews three time series averaging methods proposed in the last two decades, namely, nonlinear alignment and averaging filters (NLAAF) (Gupta et al. 1996), prioritized shape averaging (PSA) (Niennattrakul and Ratanamahatana 2009) and dynamic time warping barycentre averaging (DBA) (Petijean et al. 2011).

5.2.2.1 Nonlinear Alignment and Averaging Filters

The basic principle of nonlinear alignment and averaging filters (NLAAF) is to apply dynamic time warping to calculate the distance between sequences and use an optimal path to calculate the average sequence (Gupta et al. 1996). Given two time series sequences X and Y, where $X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_j, \dots, y_n\}$, the first step of NLAAF based average sequence calculation is to extract the optimal warping path between X and Y. In this step, the align path is represented by $w =$

$\{w_1, w_2, \dots, w_k, \dots, w_K\}$, where w_k stores a pair of indices i and j of data x_i and y_j in the sequences X and Y. The second step is to calculate the average sequence $Z = \{z_1, z_2, \dots, z_k, \dots, z_K\}$ according to the following equation:

$$z_k = \frac{1}{2}(x_i + y_j) \tag{5.3}$$

The whole process of NLAAF based average sequence calculation is depicted in Figure 4.1. The first step is to pair the first and second sequences in the cluster and calculate their average sequence, and then sequentially pair the previous average sequence and next sequence to calculate their average sequence.

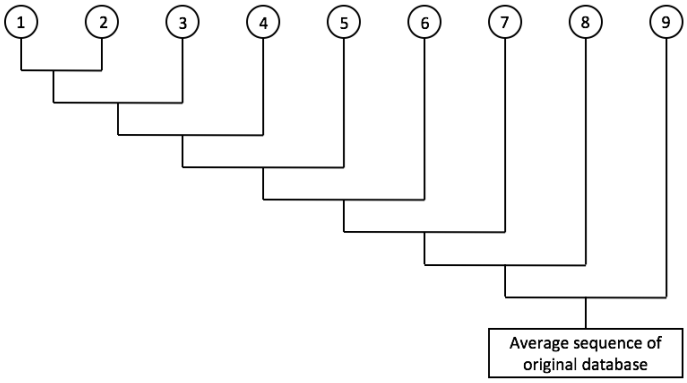


Figure 5.1. Nonlinear alignment and averaging filters

As the average sequence is calculated sequentially from the first sequence to the last sequence, the result depends on the order of the considered sequences. For the final step of average sequence calculation, the weight value of the last sequence is 0.5, which is equal to the sum of weight values of all the previous sequences. That is why the result of NLAAF is not a satisfactory approximation.

5.2.2.2 Prioritized Shape Averaging

The NLAAF based average calculation pairs sequences in order and sets the final result as the average sequence of the cluster, but there is no guarantee that a different order can get the same results. Prioritized shape averaging (PSA) was proposed to resolve the shortcomings of NLAAF (Gupta et al. 1996). In PSA based time series average calculation, a hierarchical structure tree is used to set the pairing order and weight values are calculated according to the number of sequences involved in the averaging. Therefore, this overcomes the influence of pairing order. The process is graphically illustrated in Figure 5.2.

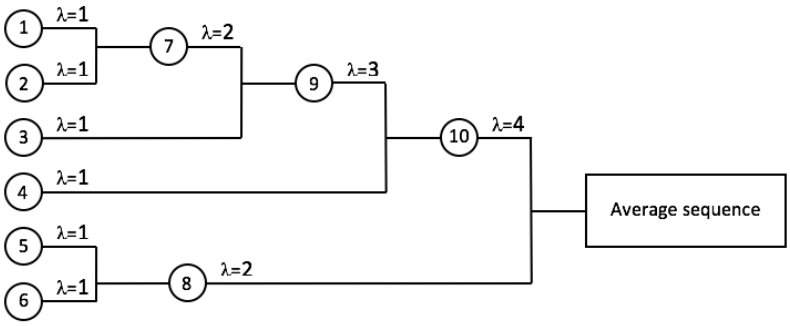


Figure 5.2. Prioritized shape averaging based average sequence calculation

At the beginning of the calculation process, the root nodes of the tree are the sequences in the initial time series database. In the following calculation process, the nearest two sequences ($X = \{x_1, x_2, \dots, x_i, \dots, x_n\}$, $Y = \{y_1, y_2, \dots, y_j, \dots, y_n\}$) are aligned together and each new value z_k on the average sequence ($Z = \{z_1, z_2, \dots, z_k, \dots, z_K\}$) is calculated using the arithmetic mean of x_i and y_j:

$$z_k = \frac{\lambda_x * x_i + \lambda_y * y_j}{\lambda_x + \lambda_y} \tag{5.4}$$

where λ_x and λ_y are the weight values of sequence X and Y. These weight values are calculated from the number of sequences used to generate the averaged sequence (Niennattrakul and Ratanamahatana 2009).

Compare with NLAAF, PSA based average calculation does not only use DTW to reduce the influence of timeline drift, but also solve the problem of pairing orders. However, according to (5.4), the length of the new sequence is K, which is equal to the length of the optimal path, and according to Algorithm 2.4 (page 21), the length of the new sequence is greater than the length of X and Y. This means that the length of the average sequence is greater than that of paired sequences after each iteration of average sequence calculation, and the length of the final average sequence will be several times of that of the initial sequences. As a consequence, the final average sequence cannot correctly characterize the features of the cluster.

5.2.2.3 Dynamic Time Warping Barycentre Averaging

The influence of pairing order is solved through the application of a hierarchical structure tree. However, there is still another issue: the length of the averaged sequences is greater than that of pairing sequences after each iteration, and in some late stages the length of averaged sequence exceeds the length of the original time series sequences. To

solve this issue, an effective global time series averaging method was proposed and it is labelled as dynamic time warping barycentre averaging (DBA) (Petijean et al. 2011). Given a time series dataset containing n sequences, the process of DBA can be divided into four steps.

- Initial sequence finding: randomly select one sequence as the original centre of average sequence calculation.
- Optimal path calculation: based on DTW, calculate the distance between every sequence and the initial sequence. In this step, the associations between centre sequence and all the sequences in the dataset can be found (Petijean et al. 2011).
- Average sequence calculation: assuming there are m sequences with different lengths in the dataset, the centre sequence is $\mathrm{XA} = \{\mathrm{XA}_1, \mathrm{XA}_2, \ldots, \mathrm{XA}_\mathrm{n}\}$, the remaining sequences in the database are $\mathrm{X1} = \{\mathrm{X1}_1, \mathrm{X1}_2, \ldots, \mathrm{X1}_{\mathrm{n1}}\}$, $\mathrm{X2} = \{\mathrm{X2}_1, \mathrm{X2}_2, \ldots, \mathrm{X2}_{\mathrm{n2}}\}$,…, $\mathrm{Xm} = \{\mathrm{Xm}_1, \mathrm{Xm}_2, \ldots, \mathrm{Xm}_{\mathrm{nm}}\}$. According to the optimal paths that are extracted in step 2, the values in the average sequence can be calculated as:

$$li = \frac{sum(align(XA_i))}{count(align(XA_i))} \tag{5.5}$$

  where $align(XA_i)$ means the elements that are aligned with XA_i in the rest of sequences in the dataset, $sum(align(XA_i))$ means the sum of all the elements values that are align with XA_i, $count(align(XA_i))$ means the number of elements that are aligned with XA_i in the remaining sequences.
- Repeat: When comparing the distance between output sequence and the pre-centre, if the distance between them equals 0 then stop the calculation. If the distance between the output sequence and the initial sequence greater than 0, step 2 and 3 have to be repeated.

The average sequence obtained by DBA can represent the features of the original sequences more accurately when compared with the other two methods (NLAAF and PSA). However, since the initial sequence is selected randomly, the whole calculation needs to be repeated several times until the distance between the new average sequence and previous average sequences become zero (or less than a specified small threshold), this is time-consuming. In addition, because the weight values of all the sequences are

the same in the calculation, the result may not correctly represent the features of the cluster if there are some outliers.

5.3 AT-means: Automatic Time Series Clustering

Time series clustering has been ubiquitously used in diverse areas and many clustering methods are available in the literature, among which K-means is most commonly used algorithm. K-means clustering algorithm and its variants, however, have several shortcomings for many real applications (Jain 2010; Oyelade et al. 2010). To overcome these shortcomings, we propose an average sequence based totally automatic time series clustering method, called AT-means.

5.3.1 Initialized Weighted Global Time Series Averaging

For traditional non-time series K-means clustering methods, the distances between data are calculated according to Euclidean distance, and because the mean value of every cluster can represent meaningful information of the cluster, the arithmetic mean value of every cluster is used as the centre of the cluster. However, for time series clustering, because a time series represents a collection of values over time, and the length of different time series are usually different from each other, the average time series of every cluster cannot be simply obtained through computing the arithmetic mean value of every time point.

So far, dynamic time warping barycentre averaging (DBA) is the first and only global approach to averaging a set of sequences (Petijean et al. 2011). However, due to the fact that the initial centre sequences are randomly selected and the weight values of all the sequences are the same during average sequence calculation, the calculation is time demanding and the results of the average sequences are not satisfactory.

In this part, in order to overcome the drawbacks of DBA, a modified global time series averaging method is introduced, called initialized weighted global time series averaging (IWGTA). We propose a novel scheme that can be used to determine the initial centre sequences: this proposed method has an obvious advantage in that it avoids the randomness in the determination of the initial centre sequences. The rationale behind the scheme is that the initial centre should be chosen as the sequence located in the centre area of the cluster. Specifically, it first calculates the distance matrix containing all the distances between every two sequences in the cluster, and then finds the sequence such

that the sum of the distance between the target sequence and all the other sequences is minimized. The pseud-code is given in Algorithm 5.1 below.

Algorithm 5.1 Initial Sequence Setting

Requirement: A cluster of sequences
$[m, n] \leftarrow size(sequences)$
for $i \leftarrow 1: m$ **do**
  for $j \leftarrow 1: m$ **do**
$distance(i, j) \leftarrow DTWdistance(sequences(i, :), sequences(j, :))$;
end for
end for
for $i \leftarrow 1: m$ **do**
  $sumdistance(i) \leftarrow sum\big(distance(i, :)\big)$
end for
$sortdistance \leftarrow sort(sumdistance)$;
$clocation \leftarrow find(sumdistance = sortdistance(1))$;
$centre\_sequence \leftarrow sequences(clocation, :)$

The input of this algorithm is a cluster of sequences and the output is the chosen center sequence.

An anomalous time series is usually far away from the others in a group. Although most of sequences in one cluster have similar patterns and features, there is no guarantee that there is no outlier in the cluster. For a cluster of time series with some anomalous sequences, if the influence of the outliers is at a high level, the quality of the output average sequence will be affected by the existence of the anomalous sequences.

In order to reduce the impacts of outliers, weight values are calculated according to the distances between the centre sequence and the remaining sequences. The pseud-code of the method is given in Algorithm 5.2.

Algorithm 5.2 Weight values calculation

Requirements: A cluster of sequences: $sequences$
Initial sequence: $preaverage$
for $i \leftarrow 1: m$ **do**
  $weightdistance(i) \leftarrow DTWdistance(preaverage, sequences(i, :))$;
end for
$delete \leftarrow find(weightdistance == 0)$;
$weightdistance(delete) \leftarrow inf$;
for $i \leftarrow 1: m$ **do**
  $distancerec(i) \leftarrow 1/weightdistance(i)$;
end for
$sumdistance \leftarrow sum(distancerec)$;

for $i \leftarrow 1: m$ **do**
$lamda(i) \leftarrow distancerec(i)/sumdistance$;
end for

The input of this algorithm is a set of sequences and their corresponding predefined centre sequence, the output is a vector containing the weight value of each sequence.

The amplitudes of elements in time series do not represent the whole information of the sequence because there remain some features that are not revealed by the amplitudes but can be effectively characterized by timeline. For initialized weighted global time series averaging (IWGTA), in order to reduce the influence of timeline drift when aligning two sequences, the associations between centre sequences and other sequences are obtained according to dynamic time warping. After that, the elements in the average sequence are computed according to the initial centre sequence (obtained by Algorithm 5.1), associations (obtained by Algorithm 2.4 (page 21)) and weight values (obtained by Algorithm 5.2). The pseud-code for average sequence calculation is given in Algorithm 5.3 below.

Algorithm 5.3 Average sequence calculation

Requirements: A group of time series: $sequences$
Predefined centre: $preaverage$
A vector contains weight values: $lamda$
The number of time series in the group: m
for $i = 1: m$ **do**
$w\{i\} \leftarrow DTWdrift(preaverage, sequences(i,:))$;
$series \leftarrow sequences(i,:)$;
$locationA \leftarrow w\{i\}(:,2)$;
$locationB \leftarrow w\{i\}(:,1)$;
for $j = 1: n$ **do**
$align(i,j) \leftarrow sum(series(locationA(find(locationB == j))))$
end for
end for
for $i = 1: n$ **do**
$sequencesum(i) \leftarrow sum(align(:,i))$
end for
for $i = 1: n$ **do**
$averagesequence(i) \leftarrow lamda(i) * sequencesum(i)$;
end for

The input of this algorithm is a cluster of sequences and the output is the average sequence of this cluster.

In order to demonstrate the performance of this new time series averaging method and its superiority to other methods, a set of artificial time series, containing 25 similar sequences and 3 white noise sequences, is considered. The sequences are shown in Figure 5.3.

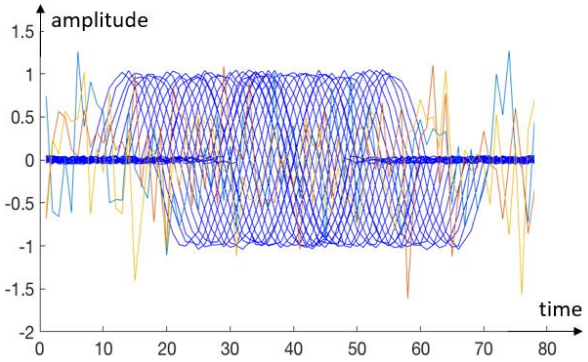


Figure 5.3. Artificial time series

The proposed time series averaging method is applied to the set of time series, and the output sequence is illustrated by Figure 5.4d. For comparison, the additional three methods, namely, nonlinear alignment and averaging filters (NLAAF), prioritized shape averaging (PSA) and dynamic time warping barycentre averaging (DBA) are also applied to the same dataset, and the outputs are shown in Figures 5.4a, 5.4b and 5.4c, respectively.

Figure 5.4a shows the result of the nonlinear alignment and averaging filters approach. Clearly, the output sequence cannot represent the features of the time series in the group because the order of sequences pairing. In addition, due to the length of the average sequence is greater than paired sequences, the result cannot describe even a small fraction of the property of the dataset. For the given dataset in Figure 5.3, the length of the output sequence shown in Figure 5.4a is almost 280, which is about 3.5 times of the initial sequences.

Figure 5.4b illustrates the output sequence of PSA. Because the average sequence calculation process uses weight values, the impact of the order of averaged sequence is reduced, so the final sequence can roughly represent the common structure of the sequences in the dataset. However, this method has a same issue as for NLAAF, that is, the length of final sequence is greater than that of original sequences, and as a consequence the average sequence may not correctly represent the features of the time series in the group. In addition, because the weight values depend on the number of sequences used to generate the averaged sequence, the whole calculation process usually

requires a heavy computational load for large datasets. The number of iterations for the artificial data is 29. In practice, however, the numbers of iterations could be very large, making the calculating process of Algorithm 5.3 time demanding if the sizes of datasets are large.

Figure 5.4c shows the result by DBA: compared with the outputs of NLAAF and PSA, DBA provides a much better representation of the features of the initial time series sequences. Note, however, the initial centre is randomly selected, and the calculation process may need to take many iterations to find the average sequence. So, for large size datasets, it is time demanding if the initial sequences are distributed in a wide range or many of anomalous sequences are far away from the actual centre. Furthermore, there is a scope of delay in the timeline of the result sequence, which can result in error in subsequent calculations.

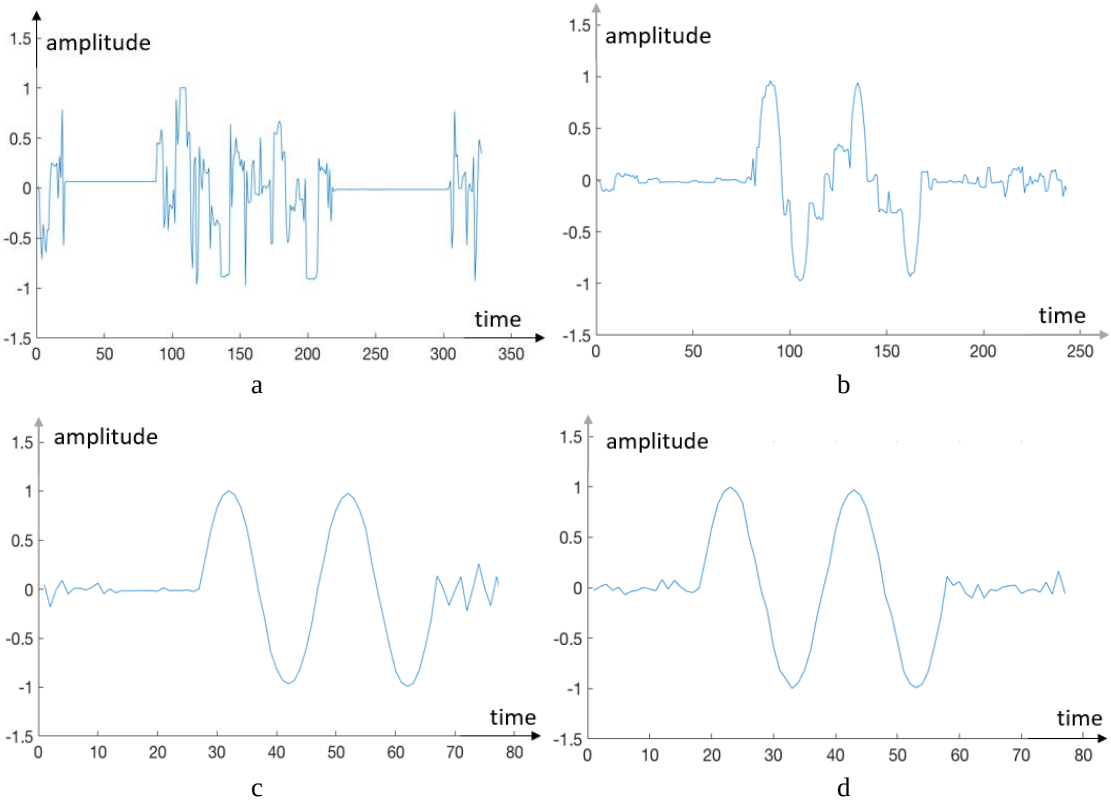


Figure 5.4. Performance comparison of the proposed method and three other methods on the artificial dataset of time series sequences (the amplitude values of figures in Figure 5.4 are average values of time series in Figure 5.3). a) NLAAF based time-series average calculation, b) Prioritized shape averaging based time series averaging calculation, c) DBA based time series average calculation, d) the proposed method based time series average calculation.

The result of the proposed method is shown in Figure 5.4d, from which it is clear that the new method can correctly characterise the structure of the initial sequences and the

timeline is also correctly located at the centre of the dataset. With the application of weight values that are computed according to the distance between initial centre and the remaining sequences, the impact of outliers is reduced or avoided. Specifically, with the application of both DBA and the proposed algorithm to the dataset, the sums of within-group distance are 137.6896 and 128.2563 respectively. By comparing the sum of within-group distance obtained by DBA and the proposed method, the average sequence obtained by the proposed method is closer to the centre of the dataset.

5.3.2 Initial Centre Determination

The traditional K-means algorithm starts with *k* arbitrary centres, typically chosen uniformly at random from the data points. However, due to the centre of each cluster is calculated as the mean of all the points assigned to it, this algorithm is highly sensitive to the selection of initial centres (Celebi et al. 2013; Fahim et al. 2009). In order to overcome the disadvantage, in the proposed AT-means, two sequences are extracted from the unorganized dataset and ensure that these two sequences are not outliers and not close to each other. For easy understanding, Figure 5.5 provides an illustration (in 2 dimension) of how the proposed method works on find two initial centres, where the blue dotted-points represent sequences and the distance between points represent the DTW distance between the corresponding sequences. According to the location of two initial points (in red colour) shown in Figure 5.5, it can determine that initial centres should be located in the two crowed areas and not close to each other.

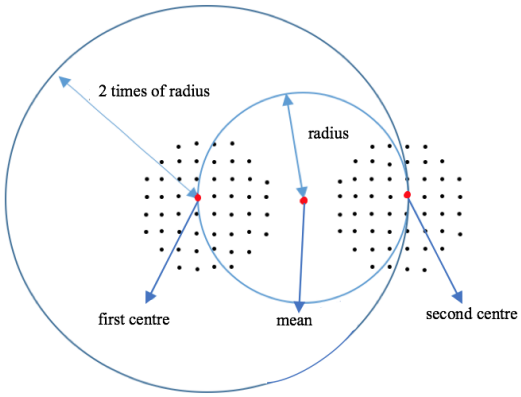


Figure 5.5 Finding initial sequences

The procedure of determining the initial sequences is divided into 6 steps: 1) calculate the average sequence of the unorganized dataset using Algorithm 5.3, 2) calculate the matrix containing the distances between all the sequences in the dataset and the average sequence, 3) set the mean value of distance matrix (obtained in step 2) as radius and the

average sequence as the centre of a circle, 4) find the first sequence that the distance between this sequence and the centre is closest to radius among all the distances in distance matrix, 5) calculate the distances between all the remaining sequences and the first sequence, 6) find the second sequence that the distance between it and the centre is close to the radius and the distance between it and the first sequence is close to twice of radius. The method proposed here is referred as average distance initial centre determination.

Assume there are two clusters in the time series dataset, the procedure of finding initial centres is depicted in Algorithm 5.4 below.

Algorithm 5.4 Finding Initial Sequences

Requirements: A set of sequences: $sequences$
$[number, length] \leftarrow the\ size\ of\ sequences$
$average \leftarrow the\ average\ sequence\ of\ the\ dataset$
for $i = 1: number$ **do**
$distance(i) \leftarrow DTWdistance(sequences(i,:), average)$
end for
$location1 \leftarrow$
$position\ of\ the\ sequence\ that\ the\ distance\ between\ this\ sequence$
$and\ the\ averagesequence\ is\ close\ to\ the\ mean\ value\ of\ distance$
$location2 \leftarrow$
$position\ of\ the\ sequence\ that\ the\ distance\ between\ this\ sequence$
$and\ the\ average\ sequence\ is\ close\ the\ mean\ value\ of\ distance,$
$and\ the\ distance\ between\ this\ sequence, and\ the\ first\ sequence$
$is\ close\ to\ two\ times\ of\ the\ mean\ value\ of\ distance.$
$sequence1 \leftarrow sequences(location1,:)$
$sequence2 \leftarrow sequences(location2,:)$

The input of this algorithm is a set of sequences; the outputs are two sequences, which can be used as the initial center sequences.

To demonstrate the performance of the proposed method for determining initial sequences, a dataset containing two clusters (each contains 100 time series, with a length of 500) and four anomalous sequences are artificially generated, and both the proposed method and the random selection method are applied to the dataset to extract the initial centre sequences. Similar to the description of the sequences in Figure 5.5, all the sequences in the dataset are represented by points and the distance between points is defined as the DTW distance between corresponding sequences; the visual illustrations are shown in Figure 5.6a.

The extracted sequences by the proposed method are shown in Figure 5.6b (represented by red points). It can be noticed that both of the extracted centre sequences are close to the centres of the two clusters although there may exist a very small distance between the extracted centres and the actual centres. Random selection method is also applied to the same dataset to extract initial centres. The method was performed twice, and the corresponding numerical experiments results are shown in Figure 5.6c and Figure 5.6d, respectively. As shown in Figure 5.6c, one centre is an outlier and the other centre locates at the edge of one cluster. In Figure 5.6d, the two centres are close to each other. In fact, these are the two typical and commonly encountered issues when the random selection method is applied to real world problem solving – in many cases it could fail to find the correct or appropriate centres. Our proposed method, however, can overcome these issues.

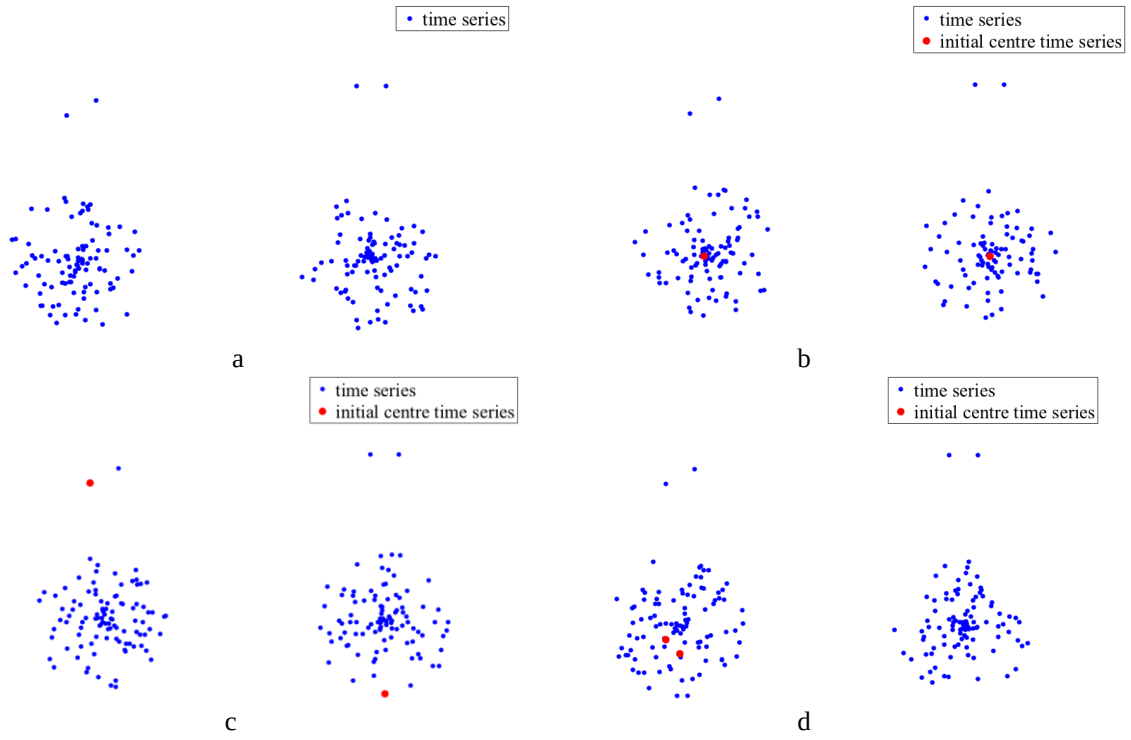


Figure 5.6 Distribution of sequences in dataset and initial sequences. a) a dataset contains two clusters of sequences and four outliers, b) initial centre sequences extraction based on the proposed method, c) first time randomly initialization, d) second time randomly initialization

5.3.3 Elbow Point

The idea of the traditional elbow method (Tibshirani et al 2001) starts with $k = 2$ (the number of clusters).It keeps increasing the number of clusters in each step by 1 and calculating the cost function of each cluster, and stops when the cost function drops dramatically at a value of the number of clusters. In this paper, a modified version of the

elbow method is proposed and used to determine the number of clusters. The proposed version has the following two improved properties: 1) the parent cluster is only separated into two children clusters in each step and only both of the children groups are analysed in next step, 2) the elbow point is the maximum point of the relevant cost function rather than the point where the cost function drops dramatically.

For a time series dataset that contains several classes of sequences, at the beginning of clustering, some sequences are inevitably forced into some groups where they do not belong. Traditional average distance, which is defined as the arithmetic mean of distances between all the sequences in one group and their corresponding average sequence, is always used as the cost function for cluster number determination (for K-means clustering), but it cannot detect whether or not outliers exist when there are a large number of sequences in the group. In this section, in order to underline the impact of outliers via the analysis of average distances, weight values are used to calculate the average distance. The weight values are defined as:

$$\mathrm{w_i} = \frac{\mathrm{d_i}^3}{\sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{d_i}^3} \tag{5.6}$$

where d_i, with $i = 1, 2, \dots, n$, is the distance between the ith sequence and the average sequence, w_i is the weight value of the distance between the ith sequence and the average sequence. The reason the third power of the distance is used is to enhance the impact of outliers during average distance calculation. The weighted average distance is computed as:

$$\mathrm{wadis} = \sum_{\mathrm{i}}^{\mathrm{n}} \mathrm{w_i} * \mathrm{d_i} \tag{5.7}$$

In addition, in order to reveal the sparseness of the group through the analysis of average distance, weighted average distance is multiplied by the distance between the average sequences of two children clusters. In this paper, this average distance calculation method is referred to as the dual weight average distance calculation (DWAD).

Consider a dataset containing 22 points: 20 of them are close to each other and 2 are outliers (see Figure 5.7). By applying the standard elbow method, the average distance of these 22 points is 2.3175, but using the dual weight average distance (DWAD) is 24.5207. Furthermore, if the number of clusters is set to be 2, the overall average distance of the two clusters is 0.0917 (standard elbow method) and 0.1535 (DWAD), respectively.

Clearly, there is a dramatic reduction in the average distance after the dataset is separated into two clusters with the standard elbow method, but comparing the result of DWAD, the reduction rate of traditional elbow method is not significant. Therefore, in order to weight the influence of outliers and increase the changing rate of average distance, DWAD is recommended.

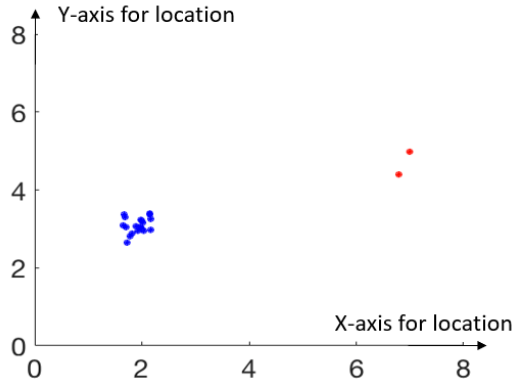
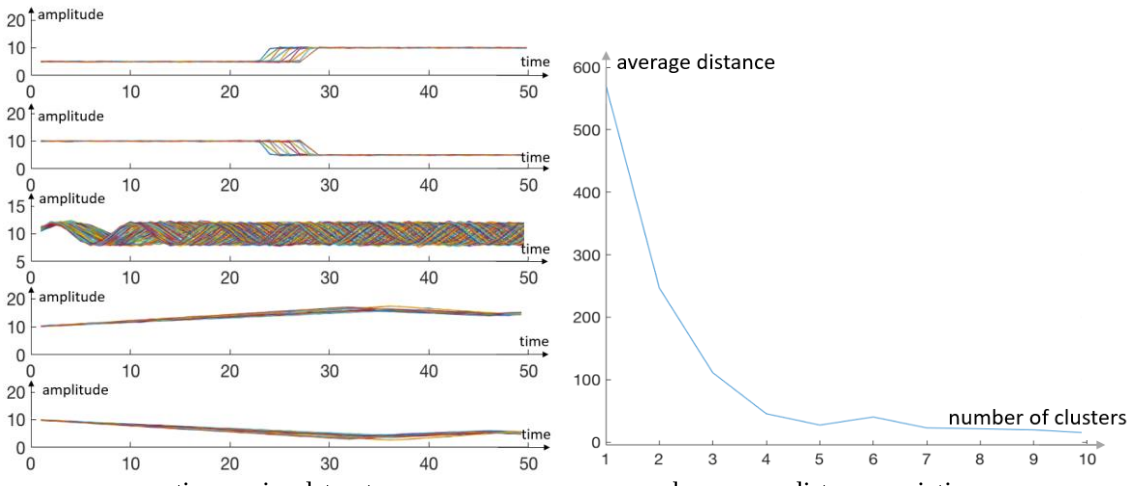


Figure 5.7 20 normal points with 2 outliers

It should be stressed that there could be several elbow points, which cannot always be unambiguously identified by means of the standard elbow method (Kodinariya and Makwana 2013). For example, consider a dataset containing 5 different classes of sequences, as shown in Figure 5.8a. The average distance changing calculated by using the standard elbow method is shown in Figure 5.8b.



a. time-series dataset b. average distance variation curve

Figure 5.8. Time series dataset and its corresponding average distance variation curve

It is well known that the first derivative of a curve at a point means the slope of the curve at that point, and the second derivative at a point represents the change rate of the first derivative series. For the standard elbow method, the point at which the distance

variation curve dramatically drops is considered to be a minimum point of the associated cost function and thus that point is chosen as the cluster number (Bholowalia abd Kumar 2014). This is not a good cluster number determination mechanism due to its lack of robustness (e.g. the determined cluster number could be much smaller or much larger than the true value). Different from standard elbow method, in this chapter, the parent cluster is separated into two children clusters, and the dual weight average distances of parent cluster and children clusters are used to generate the distance variation curve. The splitting process from dataset to children clusters is briefly described in Figure 5.9.

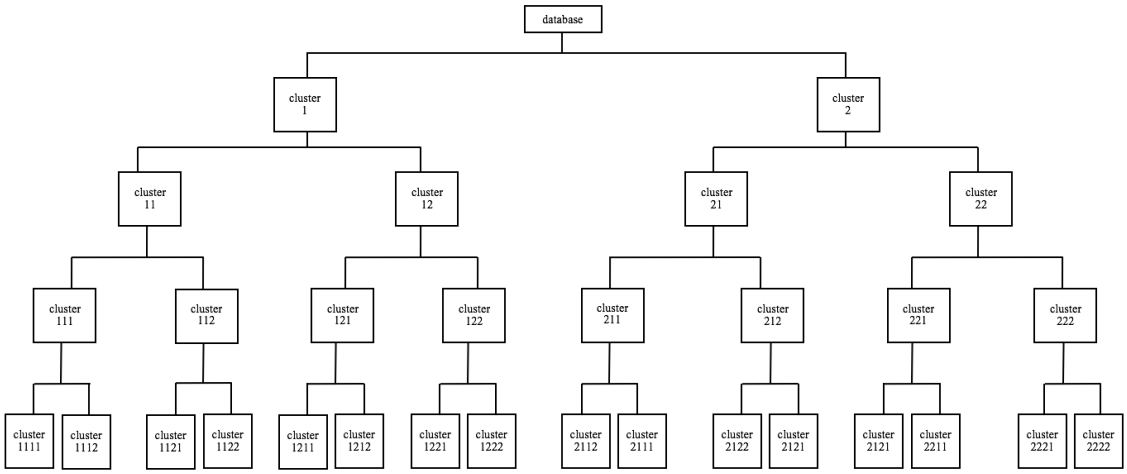


Figure 5.9 Splitting process

Once we get the distance variation curve, the specific values between the dual weight average distances of children clusters and their corresponding parent clusters are set as "first derivatives"; the "second derivatives" can then be obtained through calculating the specific values of the "first derivatives" of children clusters and the parent clusters. Taking the first branch (from database to cluster1111) as an example, the calculation procedure of "second derivative" of this branch is depicted by Figure 5.10.

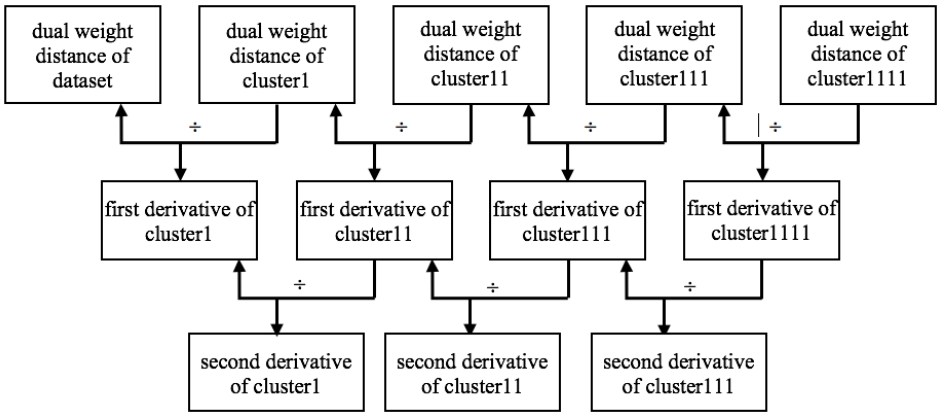


Figure 5.10 Calculation procedure of "second derivative" of first branch

Because the minimum value of cluster number is greater than or equal to 2, the initial value of "second derivative" is set to 0. The "second derivative" variation curve of the first branch is shown in Figure 5.11. Different from normal cluster number determination methods (Tibshirani et al. 2001; Pelleg and Moore 2000) that there is a cost function to define the elbow points, in this work the elbow points are defined as the points where the "second derivative" series drop. For the curve shown in Figure 5.11, it means the splitting should stop at cluster 1.

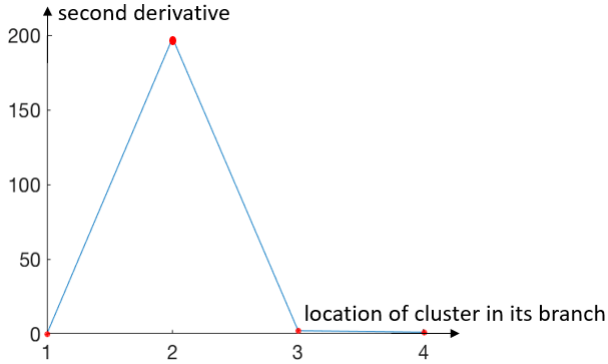


Figure 5.11 "second derivative" of first branch

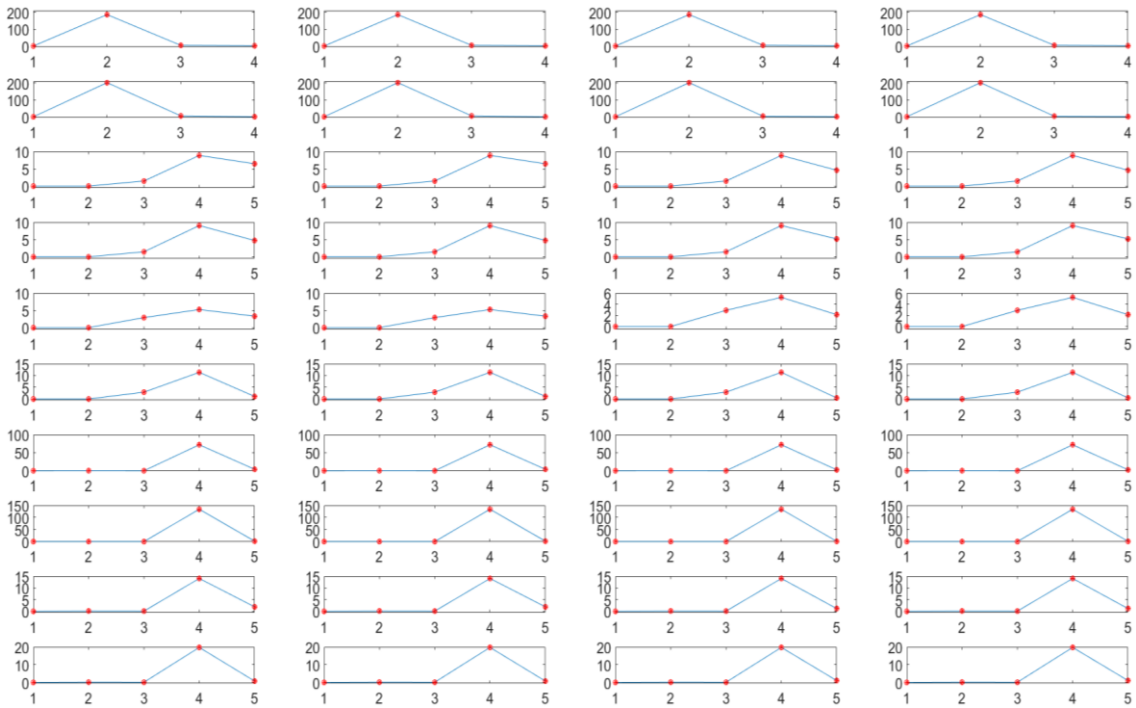


Figure 5.12. "second derivative" series (The vertical coordinates of all above figures are second derivative of corresponding clusters, such as the first one (upper left one), is second derivative series of cluster- 1111, the last one (bottom right one), is second derivative series of cluster-22222. The X-axis of all above figures is location of corresponding cluster in its branch)

We applied the "second derivative" method to the dataset in Figure 5.8a to search and test each of the candidate cluster number. According to the definition of elbow points

in this chapter, and the "second derivative" variation curves of all branches (from database to the bottom clusters) in Figure 5.12, the splitting should stop at cluster 1, cluster 211, cluster 212, cluster 221 and cluster 222 in Figure 5.9.

5.4 Results and Comparison

In order to validate the performance of the proposed AT-means for time series clustering, we applied AT-means to 10 benchmark datasets available in the research conducted by Chen et al. (2015). For comparison purposes, we also applied the following three K-means algorithms to the same datasets: a) *k* is predefined; b) initial centres and k are predefined; c) global sequences averaging method is applied and k are predefined.

5.4.1 Adjusted Rand Index

The Adjusted Rand Index (ARI) (Hubert and Arabie 1985), as an extension of the Rand index (Rand 1971), is one of the most commonly used cluster validation indexes and it was recommended as an index for measuring agreement between two partitions in clustering analysis with different numbers of clusters (Santos and Embrechets 2009; Milligan and Copper 1986). Detailed description of the adjusted Rand index can be found in the paper proposed by Yeung and Ruzzo (2001).

Suppose that *O* and *U* represent two different partitions of the objects under consideration, *O* is the true partition and *U* is K-means result, the notions are illustrated in Table 5.1.

Table 5.1 Notations for Comparing Two Partitions

Group	U_1	U_2	...	U_k	Total
O_1	$A_{1,1}$	$A_{1,2}$	...	$A_{1,k}$	$A_{1,}$
O_2	$A_{2,1}$	$A_{2,2}$	...	$A_{2,k}$	$A_{2,}$
⋮	⋮	⋮	$A_{i,j}$	⋮	⋮
O_m	$A_{m,1}$	$A_{m,2}$	...	$A_{m,k}$	$A_{m,}$
Total	$A_{,1}$	$A_{,2}$	...	$A_{,k}$	

where symbols $A_{i,j}$ (i =1, 2, …, m; j =1, 2, …, k) present the number of sequences that are both in class O_i and cluster U_j. The adjusted Rand index is defined as:

$$\mathrm{ARI} = \frac{\binom{n}{2}(a+d) - ((a+b)(a+c)+(c+d)(b+d))}{\binom{n}{2}^2 - ((a+b)(a+c)+(c+d)(b+d))} \tag{5.8}$$

In (5.8), a is the number of pairs in the same class in O and same cluster in U; b is the number of pairs of objects in the same class in O but not in the same cluster in U; c is the number of pairs of objects in different class in O but in same cluster in U; and d is the number of pairs of objects in different class in O and in different cluster in U. These four numbers are calculated as follows (Santos and Embrechets 2009):

$$a = \sum_{i=1}^{m} \sum_{j=1}^{k} \binom{A_{i,j}}{2} \tag{5.9}$$

$$b = \sum_{i=1}^{m} \binom{A_{i.}}{2} - a \tag{5.10}$$

$$c = \sum_{j=1}^{k} \binom{A_{.j}}{2} - a \tag{5.11}$$

$$d = \binom{n}{2} - a - b - c \tag{5.12}$$

The adjusted Rand index (ARI) is equal to 1 if all the sequences in one dataset are correctly categorized into their own corresponding clusters, and it will be zero if no sequence is correctly classified. Theoretically, the larger the number of correctly categorized sequences, the larger the adjusted Rand index is. In other works, the smaller the number of correctly classified sequences, the smaller the adjusted Rand index is.

5.4.2 Results and Analysis

The experiments considered in this study are based upon the following approaches:

(a) K-means #1: the actual number of clusters is known.

(b) K-means #2: the proposed initial centre setting method is applied, and the actual number is known.

(c) K-means #3: the modified global averaging method is applied, and the actual number of clusters is known

(d) AT-means: Automatic time series clustering.

The experimental results by the AT-means and K-means methods for the 10 datasets are tabulated in Table 5.2, in which the first column gives the names of the 10 benchmark datasets considered in this study and the additional four columns present the corresponding ARI values of the four clustering approaches.

Table 5.2. The Results of K-means with 3 Conditions and AT-means to Time Series Datasets

	Adjusted Rand Index			
Name	K-means #1	K-means #2	K-means #3	AT-means
CBF	0.3149	0.3780	0.4574	0.6821
ECG 200	0.0965	0.1314	0.4452	0.6596
Face All	0.0107	0.3652	0.4263	0.6358
Medical Images	0.0168	0.1052	0.3078	0.4162
Trace	0.1240	0.2334	0.3730	0.5539
ECG Five Days	0.0307	0.1626	0.3014	0.6394
Synthetic Control	0.1187	0.4453	0.4057	0.6858
Proximal Phalanx TW	0.0937	0.1357	0.2968	0.5978
Two Lead ECG	0.1256	0.1351	0.4037	0.5985
Electric Devices	0.5172	0.5183	0.5213	0.6762

From Table 5.2, it can be noticed that the clustering performance is improved by using both the proposed initial centres setting method and the global averaging method.

Firstly, the ARI values listed in column 2 represent the performance of the standard K-means clustering method where the actual number of clusters is assumed to be known. The values given in column 3 show the performance of the standard K-means, where the actual clusters number is known, and the proposed initialization method is applied. It can be noticed that all the values in column 3 are greater than their corresponding values in column 2, meaning that the performance of K-means is improved by applying the proposed initial centres setting approach.

Secondly, the ARI values listed in columns 4 is for K-means #3. It can be seen that all the ARI values in column 4 are greater than their corresponding values in column 2, meaning that the performance of the K-means with the proposed global averaging methods is much better than K-means only with actual number of clusters.

It is interesting to compare column 4 with column 3 and note that values in column 4 are not always greater than their corresponding values in column 3. For example, for the Synthetic Control dataset, the index in column 4 is 0.4057 which is smaller than that (0.4453) in column 3, means that k-means with global averaging may not always outperform that with centre initialization method.

Thirdly, the last column (column 5) presents the ARI values of the proposed AT-means. It is obvious that AT-means achieves outstanding performance for all datasets when compared with the three compared K-means approaches.

These experimental results confirm that AT-means can produce much better clustering performance for time series sequences by adopting and incorporating the following three methods: average distance initial centre determination (AID), initialized weighted global time series averaging (IWGTA) and dual weight average distance (DWAD) calculation. It is worth noting that none of the three compared K-means methods can be treated as an automatic unsupervised clustering approach because all of them require a predefined number of clusters. For AT-means, however, there is no need to pre-specify the number of clusters as the clustering process will automatically terminate when the "second derivative" series begins to decrease, and the turning point is treated to be the number of clusters.

5.5 Summary

In this chapter, we proposed an automatic time series clustering method, called AT-means, which can automatically complete the clustering process for a set of time series. The main contributions of this chapter include three aspects: i) by effectively choosing the initial sequences of the clusters and applying weight values to average sequence calculation, the influence of outliers is reduced or removed and the average time series can more properly represent the information of a set of time series; ii) through setting the average within-group distance as the radius of a cycle, the initial two sequences can be properly extracted from the dataset, iii) by using the dual weight average distance calculation and "second derivative", the number of clusters can be correctly or properly determined without manual pre-specification and intervention.

This new proposed method, along with three K-means approaches (with 3 different conditions), were applied to 10 real-life time series datasets. In terms of accuracy, measured by the adjusted Rand index, the proposed AT-means obviously outperforms the three compared K-means.

Chapter 6

Remaining Useful Life Estimation

In this chapter, a novel similarity-based remaining useful life estimation method is proposed. This method firstly uses low-dimensional time series to replace original multidimensional time series, then both testing and training data are used to build testing folder and training folder. The next is using the proposed multidimensional time series similarity measure method to extract historical fragments when their degradation behaviors are similar with that of testing unit, and finally estimate the remaining useful life of testing units according to the remaining useful life of extracted fragments. To evaluate the performance of the proposed method, it is applied to aircraft engines data provided by NASA Prognostic Data Repository. For comparison, 2 published similarity-based remaining useful life estimation approaches are applied to the same datasets. The experimental results show that the proposed method is very effective in RUL estimation.

6.1 Introduction

In past decades, over thousands of billions of dollars were spent around the world for the maintenance of safety related critical components, such as aircraft engines, nuclear equipment and large industrial machines (Kan et al 2015). Prognostic and health management (PHM), which is used to access the health status of equipment, has received increasing attention. As the main task of PHM, remaining useful life (RUL) estimation is used to provide accurate prediction of the time after which equipment will not be able to meet its operating conditions (Malinowski et al 2015). Through this estimation-based maintenance policy, PHM is not only able to protect the system from faulty causally loss, but also avoid unnecessary maintenance activities and resource wasting (Zhao et al 2017).

In general, approaches dealing with RUL prediction problem are mainly separated into 2 categories: physics-based model and data-driven model (Zhao et al 2017). A

physics-based prognostics model typically involves building a mathematical model to describe the physical behaviours of the system (Heng et al 2009). For example, the crack growth process of gear was simulated and an estimation of the remaining useful life was provided (Zhao et al. 2013); a Kalman filter method was proposed to model the crack growth in a tensioned steel band and predict the health state of the system (Swanson et al. 2000); a damage accumulation model of aircraft engine is implemented to provide the remaining useful assessment (Orsagh et al. 2004). These types of approaches are particularly important if accuracy is a critical factor and testing is restricted, but physics-based models are not easy to construct because it is challenging to obtain the physical degradation of a system. Moreover, because this kind of approach is system specific, they do not have generality. A data-driven approach attempts to derive useful information from past observed run-to-failure data and produces the prediction outputs according to the relationship between collected condition monitoring data and the degradation level of same type equipment (Heng et al 2009). Classical data-driven approaches include neural networks (Heimes 2008, Liu et al 2010), hidden Markov models (Baruah and Chinnam 2005, Camci and Chinnam 2010), Gaussian process regression (Liu et al 2013, Hong and Zhou 2012), support vector machine (Patil et al 2015, Chen et al 2013). This kind of approach is usually easier to obtain and mainly used when a physical model cannot be derived. Data-driven approaches have a wide range of applications where run-to-failure data are available, such as RUL prediction of aircraft engines based on similarity measure (Wang et al 2008).

In many practical cases, it is easier to gather data than to build accurate physical models and hence a lot of data-driven prognostics have been published in the past decades. Among the data-driven prognostic approaches, similarity-based approaches are relatively new but have made promising performances. For example, RUL estimation based on linear regression and Euclidean distance (Wang et al. 2008), RUL prediction based on fuzzy instance (Xue et al. 2008), RUL estimation based on a fuzzy pointwise similarity concept (Zio and Maio. 2010), RUL prediction based on instance-based-learning (Khelif et al. 2014), RUL estimation based on degradation shapelets extraction (Malinowski et al. 2015), and RUL estimation based on similarity of phase space trajectory (Zhang et al 2015). However, for above mentioned similarity-based RUL estimation approaches, the observed data is transformed into 1-dimension space (time series). Some degradation patterns may be lost although most useful information are kept. Moreover, because a same

operation setting cannot guarantee that the working environments are exactly the same, timeline warping may exist in both training and testing datasets. Hence Euclidean distance based similarity measure in both methods are not suitable.

In this chapter, we propose a new similarity-based RUL estimation method, aimed at overcoming the disadvantages of the aforementioned methods and improving the performance of RUL estimation. The basic idea behind the newly proposed method is as follows: First, in order to reduce the dimensionality of the original data space and keep most useful information during the transformation, all the raw data is transformed into same low-dimensional space (through principal components analysis (PCA)) rather than into 1-dimension space (1-D time series). Second, a multidimensional time series distance measure method, called multivariate time series warping distance (MTWD), is proposed to properly extract training fragments that are similar to that of the testing units. The proposed similarity-based RUL estimation method is applied to the CMAPSS datasets (Saxena and Goebel 2008) and the performance is compared with two existing methods reported by (Malinowski et al 2015) and (Wang et al 2008). Results generated by the proposed method show that the estimated RUL values are closer to real RUL values when compared with the two methods.

This chapter is structured as follows. In Section 2, related works are briefly reviewed. Section 3 introduces the process of RUL estimation based on the proposed method and Section 4 reports the experimental results. Finally, Section 5 summaries this chapter.

6.2 Related Works

In recent decades, sensors and storage technologies enable researchers to continuously monitor and record the health statues of operating components. Hence similarity-based RUL estimation generates very accurate results and a lot of similarity-based RUL estimation approaches were proposed. In this section, similarity-based approaches for RUL prediction are briefly reviewed.

Similarity-based RUL estimation approach is to match testing pattern to historical patterns and compute the RUL of testing unit. Given one training dataset and the sensors' readings of one testing unit, the whole process of similarity-based RUL estimation includes two stages: offline stage and online stage. For offline stage, multidimensional monitoring data that are collected from different sensors are first processed. It should be

noted that different types of data correspond to different kinds of processing, such as noise filtering, data fusion, feature extraction, and so on. After data processing step, the obtained data is converted into a one-dimensional time series and this time series represent the fault evolution of the equipment. After the conversion step, data is formalized into instances. For different methods, there are two kinds of instances: 1) instances represent the whole trajectory; 2) instances represent part of the whole trajectory. Hence, during the offline stage, a library of instances is constructed from available historical data. In terms of online stage, the first step is to apply the same processing operations to the sensors' readings of the testing unit. Then the similarity between the testing instance and instances in training library are computed to determine which training instance has best matching score, and the instance with the highest similarity is used to predict the RUL of the testing unit. Figure 6.1 depicts the general framework of similarity-based RUL estimation.

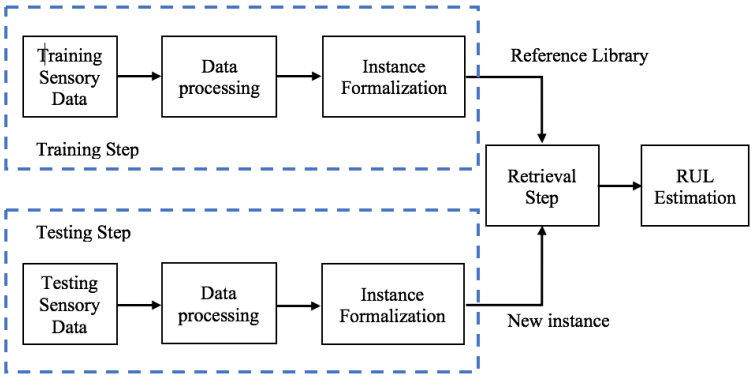


Figure 6.1 General framework of similarity-based RUL estimation (Malinowski et al. 2015)

The similarity-based prognostics approach proposed by (Wang et al. 2008) was used to tackle the problem defined by the 2008 PHM Data Challenge Competition, and the competition result showed that this method was among the top three approaches in the competition. The basic idea of this method is to match degradation pattern, which was represented by modelled health indicator, to the historical run-to-failure dataset, and the final RUL of the testing degradation pattern is computed through weighted sum of the RULs of matched historical patterns. Another similarity-based aircraft engine RUL estimation method was proposed by Xue et al. (2008). Given a testing unit, a local fuzzy model, which is related to kernel regression and locally weighted learning, was used to define a cluster of peers in which each of these peers is a similar instance to this given testing unit with comparable operational characteristics. For the prediction of RUL of the testing unit, it is computed by computing the weighted average of the peers' individual

predictions. A similarity-based approach was proposed by Zio and Maio. (2010) for RUL estimation of nuclear system. During the training step, historical run-to-failure data of the system is used to construct a library of reference trajectory patterns. During the testing step, the pointwise difference between testing and reference patterns is firstly computed. Then the pointwise difference is mapped into values of membership. The next step is to define the weight values of the individual RUL estimates, and finally calculate RUL of the testing system through weighting the RUL of extracted reference patterns. A new similarity measure method was proposed for similarity-based RUL estimation (Khelif et al. 2014). Because late working cycles of a unit at late age are more likely to observe failure patterns, late cycles will be given more weight while the whole testing trajectory is considered. Different from aforementioned similarity-based RUL estimation approaches that the whole testing trajectory is used to match patterns in reference library, discriminative shapelets are used by Malinowski et al. (2015) to predict the RUL of testing units. In the offline stage, discriminative shapelets are collected from reference run-to-failure dataset. In the online stage, shapelets of one testing unit are compared to all the shapelets in reference dataset, and RUL of the testing unit is computed based on RUL of all matched reference shapelets. Proposed by Zhang at al. (2015), similarity of phase space trajectory is used to estimate the RUL of high-pressure water pump. The phase space reconstruction is adopted to build reference degradation patterns dataset from historical run-to-failure data, and the similarities between trajectory of testing unit and trajectories in reference dataset are measure and used to estimate RUL of testing unit.

6.3 Proposed Method for Remaining Useful Life Estimation

In order to efficiently utilize the run-to-failure dataset for estimating the RUL of testing unit, a multidimensional time series similarity measure method is proposed for extracting useful historical sub-sequences. The framework of the proposed similarity-based RUL estimation method is depicted in Figure 6.2 and the analysis procedure contains the following 4 steps: 1) data pre-processing. Both training and testing datasets are transformed by PCA, and a third-order polynomial is used to smooth the sensor values. 2) library construction. The length values (number of points of a time series) of testing and training sub-sequences are flexibly allowed in a suitable range, and these testing and training sub-sequences are stored in testing and training library (this step is detailed in the 4th part of this section although it is the 2nd step during the entire RUL estimation

operation. This is because that a description of the proposed similarity measure method before library construction is helpful to understand why we build the libraries and how to build the libraries). 3) build a library construction model. 50 sub-sequences are randomly selected from training library firstly. Then RUL of each selected subsequence is calculated according to their corresponding testing library and training library, and finally determine the parameters of the model according to the information (starting points, RUL, number of points in the subsequence) of the 50 sub-sequences and their corresponding historical sub-sequences that are acceptable for RUL estimation. 4) similarity measure and RUL estimation. Calculate the RUL of the RUL of testing subsequence (testing subsequence is a part of the whole degradation trajectory of the testing equipment), and the RUL of the testing equipment is the mathematic average of the RUL values of all its corresponding testing sub-sequences.

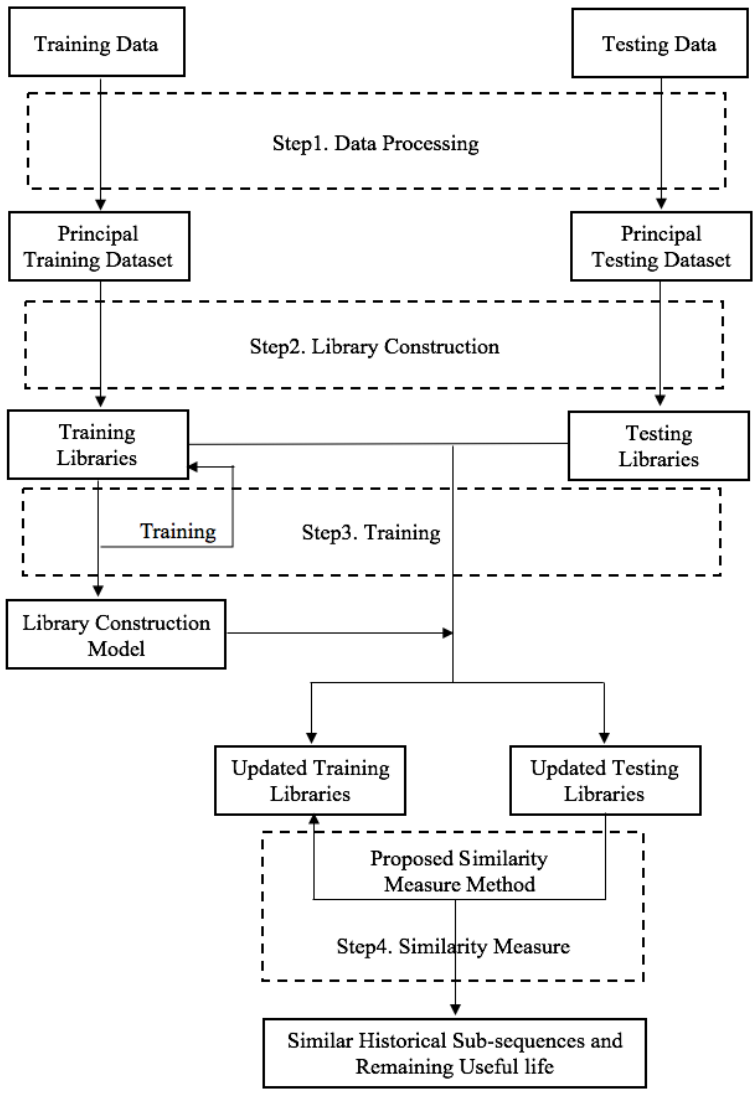


Figure 6.2 Flow chart of RUL estimation

6.3.1 Data Description

The data provided by NASA is generated by using C-MAPSS that can be utilized to simulate the realistic work conditions of the commercial turbofan engine. Under different working conditions, multiple simulations of the same type engines are carried out and the data from a fleet of engines is constructed. For each such simulation, one engine will go through the process from healthy status to failure and all the working cycles in the entire process are used to represent the working life. Because 3 working conditions (altitude, speed and throttle resolver angle) and measurement values from 21 sensors are recorded in every working cycle, the whole process of one engine is represented by a 24-dimensional time series. According to 4 different experiment setups, there are 4 independent datasets provided by NASA and the basic information of these datasets are listed in Table 6.1.

Table 6.1: Basic Information of Datasets

No. of Dataset	1	2	3	4
Number of Fault Modes	1	1	2	2
Number of Operation Conditions	1	6	6	6
Number of Training Units	100	260	100	249
Number of Testing Units	100	259	100	248

As shown in Table 6.1, depending on the number of fault modes, dataset 1 and 2 contain only 1 fault mode while dataset 3 and 4 include 2 different fault modes, depending on the number of operation conditions, dataset 1 and 3 contain 1 operation condition while dataset 2 and 4 include 6 different conditions. These datasets are divided into training and testing subsets. Training subsets include instance with complete run-to-failure data, which is used to construct the matching and estimating model. Testing subsets include instances with data up to a certain cycle prior to failure, which is used to calculate RUL. In this chapter, the 1st and the 4th dataset are used to evaluate the performance of the proposed method.

The first dataset is constructed under one fault mode and one operation condition. The entire working procedure of one unit has a number of working cycles and each working cycle includes unit ID, cycle index operation conditions and measurement values from 21 sensors. Table 6.2 briefly describe the entire life of one unit in dataset 1.

Table 6.2 Run-to-Failure of One Engine in FD01 Training Dataset

Cycle	Operation Setting 1	Operation Setting 2	Operation Setting 3	Sensor1	Sensor2	…	Sensor 21
1	-0.0007	-0.0004	100	518.6700	641.8200	…	23.4190
2	0.0019	-0.0003	100	518.6700	642.1500	…	23.4236
…	…	…	…	…	…	…	…
192	0.0009	0	100	518.6700	643.5400	…	22.9649

The 4th dataset is the most complicated one, which includes 2 kinds of fault modes and 6 different operation modes. The variables of each cycle in this dataset include the same information in the first dataset, but the difference is that the working conditions keep changing during the entire working life while the working conditions in the first dataset stay the same. Table 6.3 gives an example of the entire life of one unit in dataset 4.

Table 6.3 Run-to-Failure from One Engine in FD04 Training Dataset

Cycle	Operation Setting 1	Operation Setting 2	Operation Setting 3	Sensor1	Sensor2	…	Sensor 21
1	42.0049	0.8400	100	445.0000	549.6800	…	6.3670
2	20.0020	0.7002	100	491.1900	606.0700	…	14.6552
…	…	…	…	…	…	…	…
5	25.0063	0.6207	60	462.5400	536.1000	…	8.6754
6	34.9996	0.8400	100	449.4400	554.7700	…	8.9057
7	0.0019	0.0001	100	518.6700	641.8300	…	23.4578
…	…	…	…	…	…	…	…
17	9.9989	0.2506	100	489.0500	603.8000	….	17.1975
…	…	…	…	…	…	…	…
321	42.0058	0.8400	100	445.0000	549.7100	…	6.4590

6.3.2 Data Pre-Processing

The measurements of training data start with different levels of degradation but all of the beginning points are considered as healthy (the equipment or unit works well). The measuring process stops when the equipment reaches a level where its condition is considered not sufficient to meet the associated operating requirement. In order to make

the information of the degradation trend easy to understand, the abscissa of the last cycle of every equipment or unit is set as 0 whereas all the healthy cycles have positive indices. In this way, take measurements of sensor 2 in the first training dataset (Saxena and Goebel 2008) as example, the degradation behaviours of all the 100 equipment are shown in Figure 6.3, where x-axis states RUL and y-axis represents the measurement values of sensors. For example, a point with coordinate (100, 643) in Figure 6.3, the value of x-coordinate means there are 100 working cycles remained before failure and the value of y-coordinate is the measured value of sensor 2.

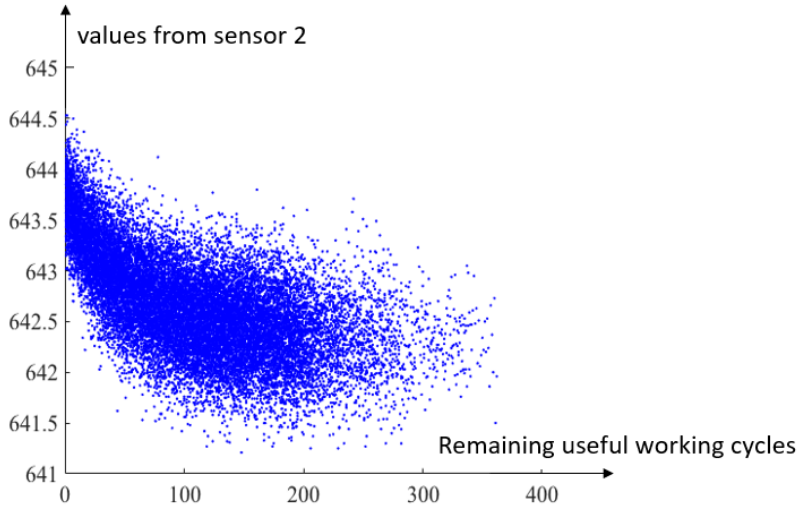
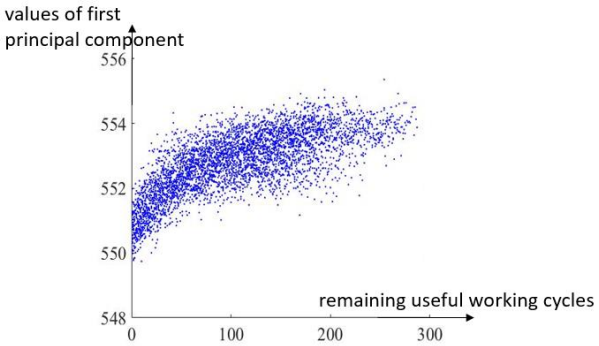
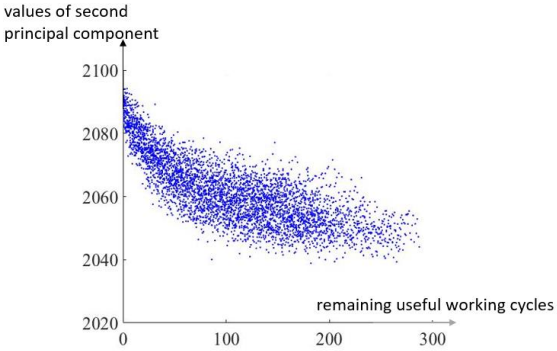


Figure 6.3 Degradation patterns of 100 units collected from sensor 2 in FD01 training dataset

The degradation trend of every unit is described by measurements from 21 sensors, but only few of them exhibit signs of degradation with decreasing of RUL. In this chapter, in order to avoid wasting of time and computing resource, PCA is used to reduce the dimension of original dataset and keep the most useful information. But note here, take the first training dataset as example, because the degradation patterns collected from 21 sensors but only express 4 different change trends, PCA should be separately used to process degradation patterns in 4 different trends. These 21 measurements are organized into 4 groups firstly: a) increasing (sensor 7, 18, 20, 21), b) decreasing (sensor 2, 3, 4, 8, 11, 13, 15, 19), c) hybrid (sensor 9, 12), d) stable (sensor 1, 5, 6, 10, 14, 16, 17). Those measurements that do not change with the decreasing of RUL are ignored in further analysis because these measurements do not show any features of degradation of these units. In addition, PCA is applied to extract principal components from every useful groups. In Figure 6.4, image 6.4a and 6.4b express the degradation trends of 20 units (randomly selected from the first training dataset) on the 2 different principal components.



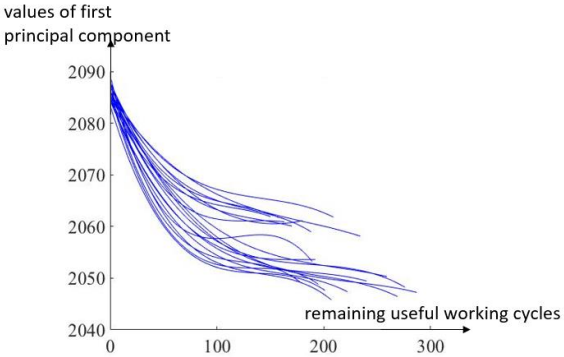
a) principal degradation trend of increasing group



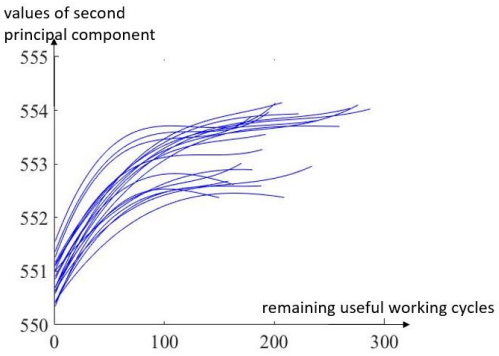
b) principal degradation tend of decreasing group

Figure 6.4 Principal degradation patterns extracted from 2 groups

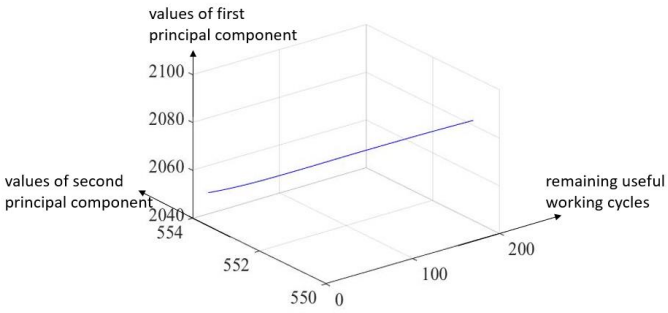
Following the dimension reduction step (through PCA), the degradation behaviours of each unit, represented by a 22-dimensional time series (one is timeline and the others are the sensor measurements), is replaced by a 3-dimensional time series (one is timeline and the others are principal components of the 2 groups). However, given that these sensor data are corrupted by noise, a third-order polynomial is used to smooth the sensor values and the resulting curves are used to represent the trends of the time series. Figure 6.5a and 6.5b show the fitted curves of increasing group and decreasing group, and Figure 6.5c (of 3 dimensional), presents the fitted principal degradation trend of the first unit in the first training dataset.



a) fitted curve of increasing group



b) fitted curve of deceasing group



c) degradation trend of unit 1

Figure 6.5 Fitted principal degradation patterns

6.3.3 Multivariate Time Series Similarity measure

Euclidean distance has been widely used in many similarity-based prognostic approaches for RUL estimation (Wang et al 2008, Malinowski et al. 2015). However, as the working environments are not always the same and the total operation cycles of similar degradation behaviours may be different, a direct use of Euclidean distance may distort the true similarity. Dynamic time warping (DTW) is such a method where two time series are warped in a nonlinear fashion and the similarity between them is then measured using the warped version of the time series. However, it should be noted that the conventional dynamic time warping approach directly calculates the distances between aligning points and sum all the distances as final result. This usually results in error of distance measure because the drift between aligning points is ignored. In this case, modified dynamic time warping (MDTW), which was proposed in Chapter 4, is used to calculate the distance between time series in this chapter.

However, it is known that traditional dynamic time warping is usually applied to 2-dimensional time series (a timeline and a target signal). As an extension of traditional dynamic time warping, MDTW cannot be directly applied to measure similarity among multidimensional time series (more than one target signals). To solve this issue, MDTW

is extended the case of multidimensional time series. The first step is to construct a distance matrix between two multidimensional time series; the second step is to find the optimal alignment path; the third step is to calculate the distance between two candidates according to the proposed method. The pseud-code of step 1, step 2 and step 3 are given in Algorithm 6.1, Algorithm 6.2 and Algorithm 6.3 respectively.

Algorithm 6.1 Construction of Distance Matrix

Requirements: multidimensional time series A
multidimensional time series B
$[mA, nA] \leftarrow$ size of A
$[mB, nB] \leftarrow$ size of B
for $i = 1{:}nA$ **do**
  for $j = 1{:}nB$ **do**
    $C(i,j) \leftarrow$ Euclidean distance between the ith point in A and the jth point in B
  end for
end for

The inputs of Algorithm 6.1 are two multidimensional time series and the output is a distance matrix containing the Euclidean distance between points in two candidates.

Algorithm 6.2 Extraction of Optimal Alignment Path

Requirements: distance matrix C
size of time series A
size of time series B
for $i = 2$ to nA **do**
  for $j = 2$ to nB **do**
    $D(i,j) = C(i,j) + \min\,[D(i-1,j), D(i-1,j-1), D(i,j-1)]$
  end for
end for
$distance = D(nA, nB)$
$k \leftarrow 1$
while $nA + nB$ not equal to 2 **do**
  if $nA - 1$ equal to 0 **do**
    $nB \leftarrow nB - 1$
  elseif $nB - 1$ equal to 0 **do**
    $nA \leftarrow nA - 1$
  else
$[values, number] \leftarrow min\,([D\,(nA-1, nB), D(nA, nB-1), D\,(nA-1, nB-1)])$
  switch $number$
    case 1 **do** $nA \leftarrow nA - 1$
    case 2 **do** $nB \leftarrow nB - 1$
    case 3 do $nA \leftarrow nA - 1, nB = nB - 1$
  end switch
  end if

$k \leftarrow k + 1$
$w \leftarrow cat(1,2,[nA, nB])$
end while

The input of Algorithm 6.2 is a distance matrix, the outputs are dynamic time warping distance and the optimal alignment path.

Algorithm 6.3 Final Distance Calculation

Requirements: size of time series A
size of time series B
dynamic time warping distance between A and B: d
optimal alignment path: w
$l \leftarrow$ length of w
$wa \leftarrow$ first column of w
$wb \leftarrow$ second column w
for $i = 1$ *to* l **do**
$Anew(i,:) = A(wa(i),:)$
$Bnew(i,:) = B(wb(i),:)$
end for
for $i = 1$ *to* nA **do**
$SA(i) = norm(A(i,:))$
end for
for $i = 1$ *to* nB **do**
$SB(i) = norm(B(i,:))$
end for
for $i = 1$ *to* l **do**
$SAnew(i) = norm(Anew(i,:))$
$SBnew(i) = norm(Bnew(i,:))$
end for
$final\ distance = d + ((l - nA)/nA) * (sum(SAnew) - sum(SA)) + ((l - nB)/nB) * (sum(SBnew) - sun(SB))$

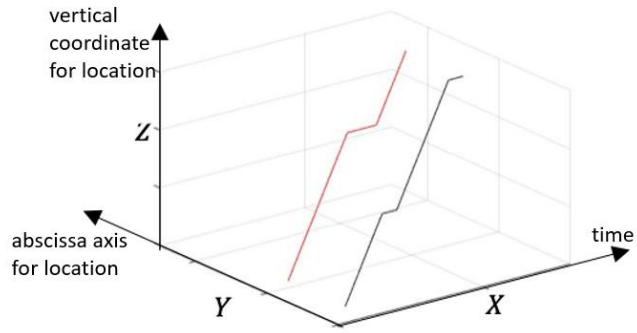
The calculation procedure of Algorithm 6.3 is similar to that of Algorithm 2.3 (page 21) and the output of Algorithm 6.3 is also the distance between two candidates. The only difference is that Algorithm 6.3 can not only be used to calculate the distance between 2-dimensional time series, but can also be used to calculate the distance between multidimensional time series.

As a simple example, consider the two 3-dimensional time series below:

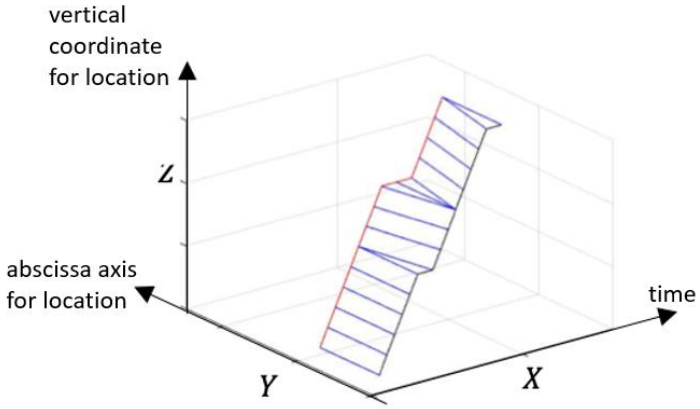
$$A = \begin{bmatrix} 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 14\ 15 \\ 1\ 2\ 3\ 4\ 5\ 6\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 13 \\ 1\ 2\ 3\ 4\ 5\ 6\ 6\ 7\ 8\ 9\ 10\ 11\ 12\ 13\ 13 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 9 & 9 & 10 & 11 & 12 & 13 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 9 & 9 & 10 & 11 & 12 & 13 \end{bmatrix}$$

Points between these two time series are depicted in Figure 6.6. Note that A partly overlaps B. In order to state the alignment between similar points in two time series, A shifts 6 units along the positive direction of y-axis as shown in Figure 6.6a and 6.6b. For the values in A and B, the first row means timeline and its corresponding axis is X. The values in second and third row represent the values on Y axis and Z axis.



a) 2 Multidimensional Time Series



b) Alignment between Multidimensional Time Series

Figure 6.6 Points alignment in 2 multidimensional time series. a) 2 Multidimensional Time Series, b) Alignment between Multidimensional Time Series

As the matching image shows in Figure 6.6b, similar sub-sequences in two candidates are aligned together and the final distance between these two multidimensional time series is 6.9768.

6.3.4 Folder Construction Model

From the discussion on the mechanism of dynamic time warping, if the length of the testing sequence is equal to that of the training sequence, the advantage of the proposed method cannot be utilized because the length of the resulting similar training sequence

may not be equal to that of the testing sequence. In this chapter, the length values of testing and training sequences are flexibly allowed to be in a suitable range rather than a fixed value.

A folder construction model is designed in order to build acceptable folder. This model is defined by a tuple $F = (T, [pte1,\ pte2], [ptr1,\ ptr2])$, where T represents the original testing sequence, $pte1$, $pte2$, $ptr1$ and $ptr2$ mean that when the specific value between the length values of testing sub-sequences and the length value of the original testing sequence is greater than $pte1$ and less than $pte2$, and the specific value between the length of training subsequence and that of its corresponding testing subsequence bigger than $ptr1$ and smaller than $ptr2$, and the average value of gaps between the estimated RUL and the actual RUL is minimum. According to the model, the construction of testing folder and training folder is described by Figure 6.7.

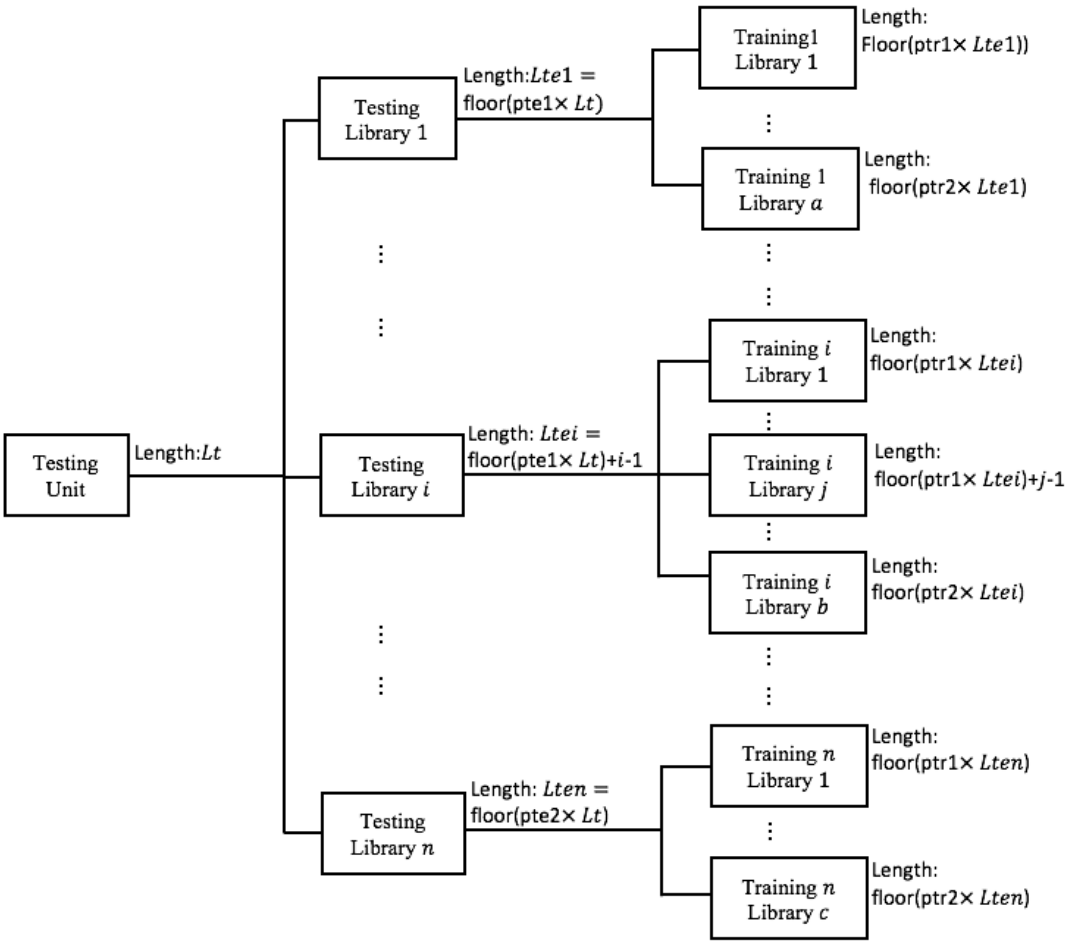


Figure 6.7 Construction of testing folder and training folder

In Figure 6.7, the input is a testing multidimensional time series with length as Lt. Testing Library i is the ith testing library corresponding to the input time series and the length values of all the time series in Testing Library i are set according to equation 6.1. The number of testing libraries corresponds to the input time series is set according to equation 6.2. For Training i Library j, it represents the jth training library corresponds to the ith testing library and the length values of all the training time series in *Training i Library j* are set according to equation 6.3, the number of time series in *Training i Library j* is set according to equation 6.4. Once all the testing and training libraries are constructed, they are stored in testing folder and training folder separately.

$$Ltei = floor(pte1 \times Lt) + i - 1 \tag{6.1}$$

$$n = Lten - Lte1 + 1 \tag{6.2}$$

$$Ltrij = floor(ptr1 \times Ltei) + j - 1 \tag{6.3}$$

$$b = Ltrib - Ltri1 + 1 \tag{6.4}$$

where $floor(...)$ means rounding the input to the next smaller integer, $Ltei$ means the length of time series in the ith testing library, $Ltrij$ means the length of time series in Training i Library j.

In this part, 50 sub-sequences are randomly selected from FD01 training dataset and used for the determination of $pte1$, $pte2$, $ptr1$ and $ptr2$.

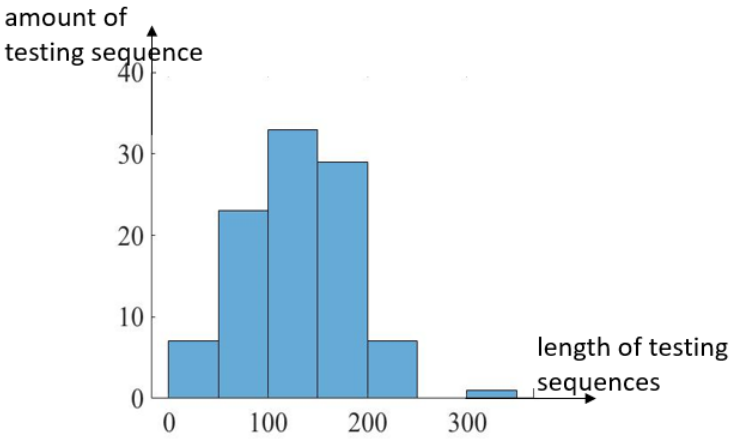


Figure 6.8 Length values of testing sequence in FD01 testing dataset

For different dataset, the values of the four parameters in the model may be different. Take the first dataset as example, the distribution of length values of testing sequences is shown in Figure 6.8, and we can find that the length of most testing sequences in the range from 50 to 200. For the length values of the 50 selected training sub-sequences,

they are defined as 70, 100, 130, 160 and 190, and these sub-sequences are separated into 5 different groups according to their length values. For one selected sequence, because the initial values of $pte1$ and $pte2$ are set as 0.5 and 1, it has a number of corresponding testing libraries. The testing folder construction process of is given by Algorithm 6.4.

Algorithm 6.4 Construction of Testing Subsequence folder

Requirements: Testing Sequence T
$lt \leftarrow$ length of T
for $i = floor(0.5 * lt)$ to lt **do**
  for $j = 1$ to $lt - i + 1$ **do**
    $Test\_folder\{i-1, j\} = T(:, j: j+i-1)$
  end for
end for

The input of Algorithm 6.4 is a multidimensional time series and the output is a folder, in which there are a number of testing libraries and every testing library contains a number of testing sub-sequences. For every testing subsequence, due to the idea behind the proposed similarity measure method that the timeline drift should be less than half of the length of testing sequence, the length values of training sub-sequences are defined from half of the length of input sequence to 1.5 times of the length of the input sequence. The training folder construction procedure is described by Algorithm 6.5.

Algorithm 6.5 Construction of Training Sub-Sequence Folder for One Testing Sub-Sequence

Requirement: one testing sub-sequence from testing sub-sequence library Tn
training dataset $Train$
$number \leftarrow$ number of units in $Train$
$ltrain \leftarrow$ working cycles of all units in $Train$
$ltn \leftarrow$ length of Tn
$traininglength \leftarrow$ from half of ltn to 1.5 times of ltn
for $i = 1$ to $number$ **do**
  $x \leftarrow Train\{i\}$
  if length of x greater or equal to $traininglength$ **do**
    for $j = 1$ to $ltrain(i) - traininglength + 1$ **do**
      $trainlibrary\{j, i\} \leftarrow x(j: j + traininglength - 1, :)$
    end for
  for $j = ltrain(i) - traininglength + 2$ to $\max(ltrain) - traininglength + 2$ **do**
    $train\_library\{j, i\} \leftarrow zero\ vector$
  end for
 else
    for $j = 1$ to $\max(ltrain) - traininglength + 2$ **do**

$train\_library\{j, i\} \leftarrow zero\ vector$
end for
end if
end for

The inputs of Algorithm 6.5 are the original training dataset and one testing subsequence from testing folder. The output is a corresponding training folder.

In order to define the values of the four parameters in the folder construction model, a 5-dimensional matrix is constructed, in which the four variables are $pte1$, $pte2$, $ptr1$ and $ptr2$ ($pte2 > pte1$ and $ptr2 \geq 1 \geq ptr1$), and the value of each point represents the average gap between estimated RUL and actual RUL.

Once the 5-dimension matrix is constructed, the minimum value in the matrix and its corresponding coordinates are utilized to construct the folder construction model. In this chapter, the minimum value in the matrix is 11.32 and its corresponding coordinates are $pte1 = 0.86$, $pte2 = 1$, $ptr1 = 0.93$ and $ptr2 = 1.05$, which means, for the 50 selected sequences, the average gap between estimated RUL and actual RUL is 11.32 when $pte1$, $pte2$, $ptr1$ and $ptr2$ are equal to 0.86, 1, 0.93 and 1.05.

6.3.5 RUL Estimation

For a testing sequence, its RUL is the mathematic average of the estimated RUL values of all the sub-sequences in testing folder, and the RUL of each testing subsequence is the mathematic average of all its own estimated RUL values that are calculated according to its corresponding training folder. Given a testing sequence, one of its subsequence is Te and one of the training libraries corresponds to Te is $Train$. The procedure of calculating RUL of Te is separated into 3 steps: The first step is to build a vector containing the distance values (based on the proposed multidimensional time series distance measure method) between Te and all the training sequences in $Train$. The second step is to extract the closest 200 training sequences and calculate RUL of Te according to the RUL of these extracted sequences. The third step is to construct a sequence S, whose abscissa is the ranking of similarity between training sequences in $Train$ and Te, and ordinate is the estimated RUL of Te. Take the testing sequence of unit 17 in the first testing dataset as example, when $pte1 = 1$, $pte2 = 1$, $ptr1 = 1$ and $ptr2 = 1$, one RUL sequence is shown in Figure 6.9.

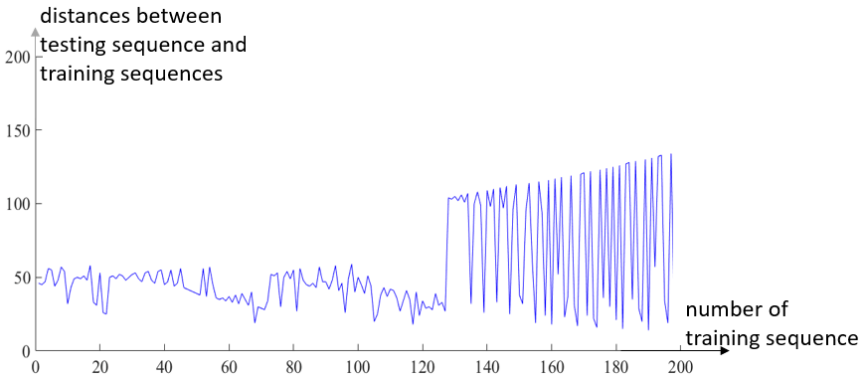


Figure 6.9 Remaining useful life sequence

In general, the training sequence that is closest to Te should be utilized to estimate the RUL of Te, but it should be noted that there is no guarantee that there is no outlier in the training libraries. In addition, due to the characteristics of the proposed similarity measure method that timeline between two candidates can be warped so that most similar points are aligned to each other, there will be a number of training sequences similar to the testing subsequence. In this paper, instead of calculating RUL of Te according to the RUL of the training sequence that is closest to Te, a number of fragments are extracted from sequence S and used for RUL calculation of Te. The pseud-code of this method is given in Algorithm 6.6.

Algorithm 6.6 Fragments Extraction from RUL Sequence

Requirement: RUL sequence S
$lS \leftarrow$length of S
while $right < lS$ **do**
  $saverage = mean(S(left{:}right))$
  for $i = left{:}right$ **do**
    $serror(i) = abs(S(i) - baverage)$
  end for
  if $max(serror) > saverage * 0.2$ **do**
    $AAA = S(left{:}right - 1)$
    $New\{k\} = AAA$
    $left = right$
    $right = right + 2$
    $k = k + 1$
  else do
    $right = right + 1$
  end if
end while
$[mNew, nNew] \leftarrow size(New)$
for $i = 1$ to $nNew$ **do**
  $lNew(i) =$length of $New\{i\}$

end for
$lNewlocation \leftarrow find(lNew > 8)$
$llocation \leftarrow length(lNewlocation)$
for $i = 1$ to $llocation$ **do**
$AAA\{i\} = New\{lNewlocation(i)\}$
end for

The input of Algorithm 6.6 is a RUL sequence and the output of this Algorithm is one or several RUL fragments. Take the RUL sequence in Figure 6.9 example, the output RUL fragments are shown in red color in Figure 6.10.

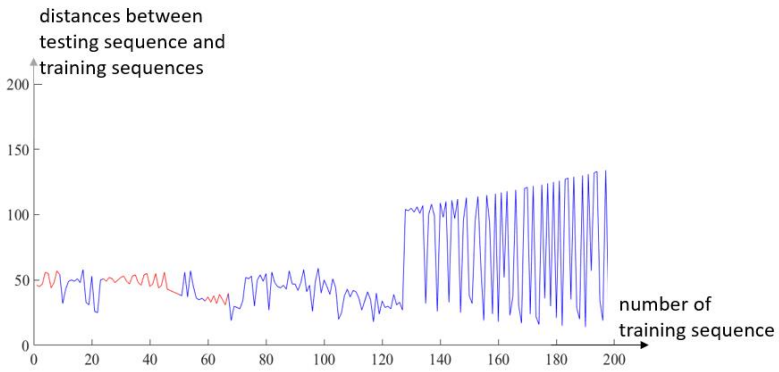


Figure 6.10 Extracted fragments of remaining useful life sequence

Once the fragments that satisfy the extraction requirements are obtained via Algorithm 6.6, RUL of Te is the mathematic average of these fragments. For the testing sequence of unit 17 in the first testing dataset, set as Ta, its actual RUL is 50, when $pte1$, $pte2$, $ptr1$ and $ptr2$ are set equal to 1 ,1 ,1 and 1, the calculated RUL of Ta is 48. When the four parameters are set equal to 0.86, 1, 0.93 and 1.05, the calculated result is 49, which is very promising.

6.4 Case Study

In order to validate the performance of the proposed similarity measure method for prognostic, this method is applied to turbofan engines dataset provided by NASA Prognostic Data Repository (Saxena and Goebel 2008). In this section, basic information of the dataset is described firstly, then the evaluation index is given, and finally we present the performance of our proposed method.

6.4.1 Performance Assessment

Performance assessment plays an important role during the procedure of RUL estimation, especially for safety related components. This is because that performance

assessment can show the estimation quality and give us important evidence to make the right decisions for further works. In this paper, the prediction score that was defined in the PHM08 data challenge competition is applied (Saxena and Goebel 2008), as shown in equation 6.5:

$$S = \begin{cases} \sum_{i=1}^{n} e^{\frac{r_e - r_t}{a_1}} - 1 & \text{for} \quad (r_e - r_t) > 0 \\ \sum_{i=1}^{n} e^{-\frac{r_e - r_t}{a_2}} - 1 & \text{for} \quad (r_e - r_t) \leq 0 \end{cases} \tag{6.5}$$

where S is computed score, which is the sum of scores of all the estimations for the n units, n is the number of units under test; r_e is the estimated RUL, $r_t$is the actual RUL; a_1 is set equal to 10 and a_2 is set equal to 13, this is because an early prediction is preferred over late prediction.

Due to the difference between a_1 and a_2, the score prediction is asymmetric, as shown in Figure 6.11, in which abscissa represents the gap values between estimated RUL and actual RUL. With the increasing of absolute value of gap, the score will increase exponentially. For the performance assessment according to the score values, the smaller the score is, the better the prediction is.

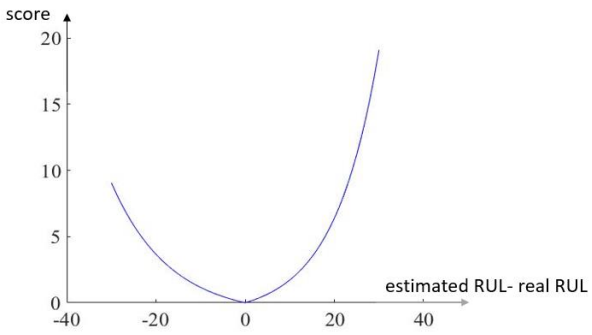
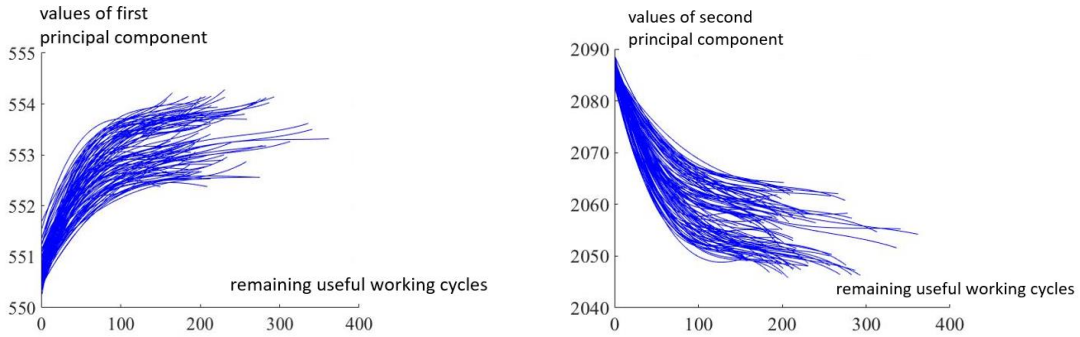


Figure 6.11 Score as a function of gap

6.4.2 Results and Discussion

The proposed similarity-based RUL estimation method is firstly applied to the first dataset. During the procedure of estimating the RUL of testing units in this dataset, on one hand, because there is only one operation condition, only the measurements from 21 sensors are considered and the operation setting is ignored. On the other hand, because there is only one fault mode, with the decreasing of RUL, measurements from sensors that do not change or change in different directions are eliminated. For these 21 measurements in the first dataset, only 12 of them are kept for further analysis.

As the step of pre-processing described in 6.3.2, in order to reduce the complexity of calculation and the impact of noise, the 12 measurements are divided into 2 groups (one with increasing trend and one with decreasing trend). Then PCA is applied to extract principal components from each group, and finally a third-order polynomial is utilized to smooth the sensor values. Figures 6.12a and 6.12b respectively represent the fitted curve of increasing group and decreasing group.



a) fitted curve of increasing group b) fitted curve of decreasing group

Figure 6.12 Fitted curve of principal components

After data pre-processing, the proposed RUL estimation method is applied to extract training time series that are similar with testing time series, and the RUL is estimated according to the information of these extracted training time series. Figure 6.13 shows the histograms of the prediction errors based on the proposed method.

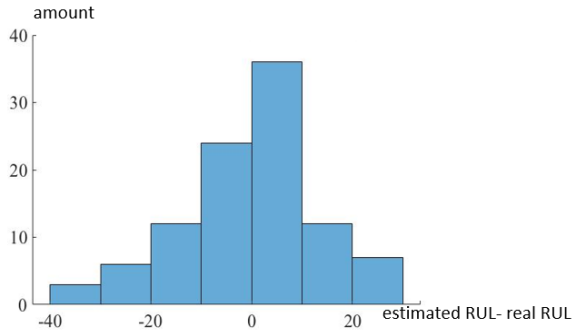


Figure 6.13 Histogram of prediction errors

As the distribution of error values shown in Figure 6.13, we can find that most of the error values are concentrated around 0 and only 3 of them less than -30 and none of them greater than 30. In this part, the distribution of error values is also represented by another type of image, as shown in Figure 6.14, where the red line means the actual RUL. The blue dot line illustrates the estimated RUL, the vertical axis expresses the RUL value, the abscissa axis indicates the 100 independent testing units. For this abscissa axis, the testing units are sorted in decreasing order for better observation (Zhao et al 2017).

The smaller the vertical distance between the blue line and red line, the more accurate the estimated RUL. For the comparison shown in Figure 6.14, where the blue points represent the estimated RUL, the red point represent actual RUL, when the actual RUL value greater than 0 and less than 50, the estimated RUL is close to actual RUL and the average estimation error is 4.8878. This is because the principal components of the 21 sensors have significant changes in this range and it is helpful to extract similar time series from training libraries. When the actual RUL values locate between 50 and 120, the average estimation error is 11.7222 and it is bigger than the average estimation error of last range, this is because the changes of the principal components of the 21 sensors are not obvious. When the actual RUL values bigger than 120, some of estimation errors are over 30 and the average estimation error is 15.4615, it is bigger than last two average values. This is because the change almost equals 0 and it is challenging to extract useful historical time series.

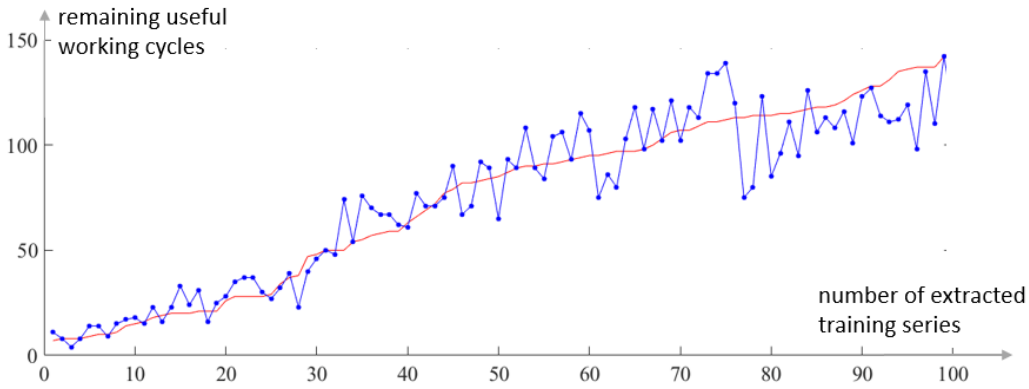


Figure 6.14 Sorted estimation for 100 units in dataset 1

The experimental results by the proposed method and 2 previous published methods for the first dataset are tabulated in Table 6.4, where the first column gives the methods for prediction and the second column presents the average estimation score given in equation 6.5.

Table 6.4 Performance Evaluation for Dataset 1

Method	Average Prediction Score of Eq. (6.5)
Proposed Method	2.41
Estimation based on Method in (Malinowski et al. 2015)	6.52
Estimation based on Method in (Wang et al. 2015)	7.91

Scores of methods in (Malinowski et al. 2015) and in (Wang et al. 2015) are obtained from (Malinowski et al. 2015).

The proposed method is also applied to the 4th dataset for RUL estimation. But it should be noted that: 1) because there are 6 different working conditions of every unit in the dataset and these 6 working conditions are highly independent to each other, the entire working procedure of every unit has to be separated into 6 independent multivariate time series. As the operation setting of all units shown in Figure 6.15, where each red point represents one kind of working condition. 2) As the number of fault modes is 2, with the decreasing of RUL under one condition, measurements do not change or change in 3 or more directions are eliminated. For the measurements from 21 sensors, 15 of them are retained for further analysis.

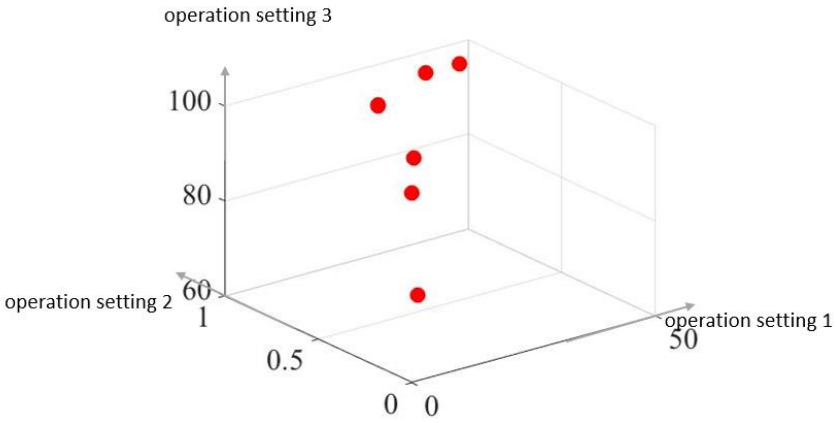


Figure 6.15 Operational settings

Because there is no failure mode nor failure criteria, the only difference between analysing dataset 4 and dataset 1 is the number of operation conditions. The process of similar time series extraction from the 4th dataset is same as that for dataset 1, and there are two more steps for the entire procedure of RUL estimation for dataset 4: 1) separate the entire working procedure of every unit into 6 groups before data pre-processing; 2) estimate the final RUL according to the RUL of extracted time series from 6 different groups. Due to the degradation degrees of measurements from 6 groups are different, weight values are calculated according to the degradation degrees and the final RUL is the weight average of the RUL values from those 6 groups. For example, a fragment, which begins from the 107th cycle with length equal to 160, is extracted from working procedure of the 47th unit in dataset 4. This fragment is then separated into 6 groups and the RUL is calculated according to the degradation patterns of the 6 groups. The RUL of this testing fragment that are calculated according to the 6 groups are 49, 55, 82, 59, 55 and 74, and the weight values of these 6 working conditions are 0.1810, 0.1544, 0.1278,

0.2280, 0.1576 and 0.1513. The final RUL of the testing fragment is 61, as the entire working cycles of the 47th unit in dataset 4 is 375, and the real RUL of the fragment is 63. It can be seen that the result obtained according to our proposed method is promising.

The experimental results by the proposed method and 2 previous methods for the 4th dataset are shown in Table 6.5:

Table 6.5 Performance Evaluation for Dataset 4

Method	Average Prediction Score of Eq. (6.5)
Proposed Method	15.0607
Estimation based on Method in (Malinowski et al. 2015)	37.7097
Estimation based on Method in (Wang et al. 2008)	69.2928

Scores of methods in (Malinowski et al. 2015) and in (Wang et al. 2015) are obtained from (Malinowski et al. 2015).

From Table 6.4 and 6.5, it can be noticed that the average score of the proposed method is smaller than previous two methods, which means the proposed method performs better than the two previous methods in terms of the accuracy of RUL estimation.

6.5 Summary

In this chapter, we proposed a similarity-based RUL estimation method, which can more effectively predict the RUL of critical equipment. The promising performance of this proposed method is mainly benefited by three aspects: i) PCA was introduced to extract useful measurements from original data and the original multidimensional time series is replaced by low-dimensional time series; ii) the construction of testing folder and training folder is helpful to accurately find historical fragments that the degradation patterns of these historical fragments are similar to that of testing units; iii) a multidimensional time series similarity measure method was proposed, which can improve the precision of distance measure between multidimensional time series.

In order to validate the performance of this proposed method, it was applied to the aircraft dataset provided by Prognostic Data Repository, and the final score shows that the proposed method perform very well in terms of RUL estimation.

Chapter 7

Conclusions and Future Work

This chapter concludes the main works of this thesis and offers some proposals for further study.

7.1 Conclusion

Time series is an important class of temporal data objects and can be easily obtained from scientific researches and daily activities. With the explosive growth of time series, we often try to make good use of them to discover the most important patterns, so that these can help us to find the relationship between different things and give us important evidence to make right decision. This thesis presents some studies on time series data mining: time series distance measure, anomalies detection from time series, automatic time series clustering and remaining useful life estimation of time series.

In Chapter 3, in order to improve the performance of similarity measure between time series, we proposed a calculation of distances between symbols and a similarity measure method. The idea behind the calculation of distance between symbols is to use the maximum and minimum mean values of all segments in individual areas to compute the distances between symbols. Additionally, the idea of the similarity measure method is to use the distance between symbolic series to compute the distance between original time series (through back calculation of time series normalization). To validate the performance of the proposed methods for time series distance measure, we integrated the proposed methods to previous popular used symbolic representation and distance measure methods (SAX and SAX-TD), and applied these integrated methods and original algorithms to 1000 pairs of benchmark time series. The experimental results show that the proposed methods improve the similarity measure performance of the corresponding original methods.

Chapter 7: Conclusions and Future Work

Chapter 4 focused on anomalies detection from ECG data. In order to deal with timeline warping during the process of time series similarity measure, dynamic time warping was modified by considering the optimal path in distance measure. Additionally, in order to overcome the drawback of previous published methods (BFDD and AWDD) that they can only work well for anomaly detection when all the anomalies in time series of interest are significantly different from each other. We introduced average non-self match, which is used to replace the minimum value of non-self match distance during the process of anomalies detection. Through applying the introduced method and previously published methods to 30 real ECGs, experimental results show that our proposed methods outperform others in terms of accuracy and efficiency.

In Chapter 5, we provided an automatic time series clustering method, called AT-means, which can be used to automatically complete the clustering process of a set of time series. There are 3 contributions in this chapter: 1) we proposed an initial sequence determination method, based on which the initial centres are close to real centre sequences. 2) We developed a global time series averaging method so that the average sequence can represent the main structure of original time series. 3) We provided an elbow point extraction method to determine the number of clusters. For comparison, AT-means, along with 3 K-means approaches (K-means with 3 different conditions), are applied to 10 real-life time series datasets. The results shown that the performance of AT-means outperform the K-means approaches in terms of accuracy.

Chapter 6 gave attention to similarity-based remaining useful life estimation. Since that high-dimensional time series mining is time-consuming, we firstly used low-dimensional time series to represent original high-dimensional time series (through PCA). Then, we presented a multidimensional time series distance measure method, called multivariate rime series warping distance (MTWD), which can be used to properly extract historical degradation patterns that are similar to that of testing equipment. Next, based on RUL of extracted historical patterns, the RUL of testing equipment was computed. For comparison, this proposed similarity-based RUL estimation method was applied to aircraft dataset provided by Prognostic Data Repository, and the estimation scores show that the proposed method outperforms previously published methods in terms of accuracy.

7.2 Future Works

Feature extraction from time series is a research field full of challenge. In this thesis, we separately proposed several approaches for distance measure, anomalies detection, clustering and remaining useful life estimation. However, there are still many problems need to be solved. Broadly, my future work plan includes three aspects:

- Existing anomalies detection approaches that are shown to perform well in one domain are not guaranteed to perform well in other domains. This is because the nature of time series in different domains is often significantly different. In Chapter 4, the proposed anomalies detection method can only work well in terms of ECG anomalies detection. In further work, a more generic anomalies detection approach is deserved to be researched.
- Time series clustering is a challenging issue because real-life time series are often with large size, this will lead to a heavy computation. In Chapter 5, the main work is to make sure that the proposed method is totally automatic, but does not consider the calculation complexity. In further work, reduce computational complexity should be given top priority.
- Similarity-based RUL estimation approaches are implemented only according to historical data. However, most of similarity-based approaches are not easy to explain the physics-based RUL estimation approaches. Physics-based RUL estimation approaches are derived from the understanding of physical mechanisms. In future work, it is desirable to generate a hybrid model, which is a combination of the proposed similarity-based approach (in Chapter 6) and a physical model.

References

Aach, J. and Church, G.M., 2001. Aligning gene expression time series with time warping algorithms. *Bioinformatics,* 17(6), pp.495-508.

Abdi, H. and Williams, L.J., 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics,* 2(4), pp.433-459.

Al Hasan. M., Chaoji. V., Salem. S., Zaki. M.J., 2009. Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition Letters,* 30(11), pp.994-1002

Aghabozorgi, S., Shirkhorshidi, A.S. and Wah, T.Y., 2015. Time-series clustering–A decade review. *Information Systems,* 53, pp.16-38.

Antunes. C.M., 2001. Temporal data mining: An overview. In *Workshop on Temporal Data Mining with the International Conference Knowledge Discovery and Data Mining*

Agovic, A., Banerjee, A., Ganguly, A.R. and Protopopescu, V., 2007. Anomaly detection in transportation corridors using manifold embedding. In *Proceedings of the 1st International Workshop on Knowledge Discovery from Sensor Data,* pp. 435-455.

Agrawal, R., Faloutsos, C. and Swami, A., 1993. Efficient similarity search in sequence databases. In *International conference on foundations of data organization and algorithms,* pp. 69-84.

Appice. A., Ciampi. A. and Malerba. D., 2015. Summarizing numeric spatial data streams by trend cluster discovery. *Data Mining and Knowledge Discovery*, 29(1), pp.84-136.

Arthurand. D. and Vassilvitskii. S., 2007. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*.

Bagul, Y.G., Zeid, I. and Kamarthi, S.V., 2008. Overview of remaining useful life methodologies. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference,* pp. 1391-1400.

References

Baraldi, P., Cadini, F., Mangili, F. and Zio, E., 2013. Model-based and data-driven prognostics under different available information. *Probabilistic Engineering Mechanics*, 32, pp.66-79.

Baruah, P. and Chinnam, R.B., 2005. HMMs for diagnostics and prognostics in machining processes. *International Journal of Production Research*, 43(6), pp.1275-1293.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers Norwell, MA, USA.

Bezdek. J.C., 2013. Pattern recognition with fuzzy objective function algorithms. *Springer Science and Business Media*.

Berndt, D.J. and Clifford, J., 1994. Using dynamic time warping to find patterns in time series. In *KDD workshop*,10(16), pp. 359-370.

Bholowalia. P. and Kumar. A., 2014. EBK-means: A clustering technique based on elbow method and k-means in WSN. *International Journal of Computer Applications*, 105(9), pp. 17-24.

Boriah, S., Chandola, V. and Kumar, V., 2008. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp. 243-254.

Boriah, S., Chandola, V. and Kumar, V., 2008. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pp. 243-254.

Byington, C.S., Watson, M. and Edwards, D., 2004. Data-driven neural network methodology to remaining life predictions for aircraft actuator components. In *2004 Aerospace Conference Proceedings*, pp. 3581-3589.

Canelas, A., Neves, R. and Horta, N., 2012. A new SAX-GA methodology applied to investment strategies optimization. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pp. 1055-1062.

Celebi, M.E., Kingravi, H.A. and Vela. P.A., 2013. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40(1), pp.200-210.

Cha S.H., 2007. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal Mathematical Models and Methods in Applied Sciences*, 4(1), pp. 300-307.

Chan, K.P. and Fu, A.W.C., 1999. Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering*, pp. 126–133.

Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. *ACM computing surveys*, 41(3), p.15.

Chaouch. M., 2014. Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves. *IEEE Transactions on Smart Grid*, 5(6), pp.411-419.

Chen. S.M. and Tanuwijaya. K., 2011. Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques. *Expert Systems with Applications*, 38(8), pp.10594-10605.

Chen, X., Shen, Z., He, Z., Sun, C. and Liu, Z., 2013. Remaining life prognostics of rolling bearing based on relative features and multivariable support vector machine. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 227(12), pp.2849-2860.

Chen. Y.P., Keogh. E., Hu. B., Begum. V., Bagnall. A., Mueen. A. and Batista. G., 2015. The UCR Time Series Classification Archive. URLwww.cs.ucr.edu/~eamonn/time_series_data/.

Choy. S.K., Lam. S.Y., Yu. K.W., Lee. W.Y. and Leung. K.T., 2017. Fuzzy model-based clustering and its application in image segmentation. *Pattern Recognition*, 68, pp.141-157.

References

Chuah, M.C., Fu, F., 2007. ECG anomaly detection via time series analysis. In *Proceedings of the 2007 international conference on Frontiers of High Performance Computing and Networking-ISPA 2007 Workshops*, pp. 123–135.

De Amorim, R.C. and Mirkin, B., 2012. Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering. *Pattern Recognition*, 45(3), pp.1061-1075.

De Stefano, C., Sansone, C. and Vento, M., 2000. To reject or not to reject: that is the question-an answer in case of neural classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(1), pp.84-94.

Desforges, M.J., Jacob, P.J. and Cooper, J.E., 1998. Applications of probability density estimation to the detection of abnormal conditions in engineering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 212(8), pp.687-703.

Di Maio, F., Tsui, K.L. and Zio, E., 2012. Combining relevance vector machines and exponential regression for bearing residual life estimation. *Mechanical Systems and Signal Processing*, 31, pp.405-427.

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

Duan. L., Xu. L., Liu. Y. and Lee. J., 2009. Cluster-based outlier detection. *Annals of Operations Research*, 168(1), pp.151-168.

Eskin, E., 2000. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning*. pp. 255-262.

Esling, P. and Agon, C., 2012. Time-series data mining. *ACM Computing Surveys*, 45(1), p.12.

Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, *Kdd*, 96(34), pp. 226-231.

Fahim A-M, Salem A-M, Torkey F-A, Saake G, Ramadan M-A (2009) An efficient k-means with good initial starting points. Georgian Electronic Scientific Journal: Computer Science and Telecommunications, 2(19), pp.47-57.

Feng. Y. and Hamerly. G., 2007. PG-means: learning the number of clusters in data. In *Advances in Neural Information Processing Systems*, pp. 393-400.

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

Fu, T.C., Chung, F.L., Ng, C.M., 2006. Financial time series segmentation based on specialized binary tree representation. In *Proceedings of the 2006 International Conference on Data Mining*, pp. 3–9.

Fujimaki, R., Yairi, T. and Machida, K., 2005. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 401-410.

Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov. P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E., PhysioBank, PhysioToolkit, and PhysioNet: *Components of a New Research Resource for Complex Physiologic.* Available at: https://physionet.org/physiobank/database/mitdb/.

Guha. S., Rastogi. R. and Shim. K., 1998. CURE: an efficient clustering algorithm for large databases. In: *ACM SIGMOD Record*. 27(2), pp. 73-84.

Guha, S., Rastogi, R. and Shim, K., 2000. ROCK: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5), pp.345-366.

Guo, L., Li, N., Jia, F., Lei, Y. and Lin, J., 2017. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing*, 240, pp.98-109.

Gupta. L., Molfese. D.L., Tammana. R. and Simos. P.G., 1996. Nonlinear alignment and averaging for estimating the evoked potential. *IEEE Transactions on Biomedical Engineering*, 43(4), pp.348-356.

References

Hamerly. G. and Elkan. C., 2004. Learning the k in k-means. In *Advances in Neural Information Processing Systems*, pp. 281-288.

Han, J., Kamber, M. and Tung, A.K., 2001. Spatial clustering methods in data mining. *Geographic data mining and knowledge discovery*, pp.188-217.

Han, J., Pei, J. and Kamber, M., 2011. *Data mining: concepts and techniques*. Elsevier.

Hancer. E. and Karaboga. D., 2017. A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. *Swarm and Evolutionary Computation*, 32, pp.49-67.

Hansen. P. and Jaumard. B., 1997. Cluster analysis and mathematical programming. *Mathematical Programming*, 79, pp. 191–215.

Harms. S.K., Deogun. J. and Tadesse. T., 2002. Discovering sequential association rules with constraints and time lags in multiple sequences. In *International Symposium on Methodologies for Intelligent Systems*. Springer, Berlin, Heidelberg.

He, Z., Xu, X. and Deng, S., 2003. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10), pp.1641-1650.

Heimes, F.O., 2008, October. Recurrent neural networks for remaining useful life estimation. In *International Conference on Prognostics and Health Management*, pp. 1-6.

Heng, A., Zhang, S., Tan, A.C. and Mathew, J., 2009. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical systems and signal processing*, 23(3), pp.724-739.

Hong, S. and Zhou, Z., 2012. Remaining useful life prognosis of bearing based on Gauss process regression. In *International Conference on Biomedical Engineering and Informatics*, pp. 1575-1579.

Hotelling H., 1933. Analysis of a complex of statistical variables into principal components. *J Educ Psychol*, 25, pp. 417 – 441.

Huang, Y.P., Luo, S.W. and Chen, E.Y., 2002. An efficient iris recognition system. In *Proceedings International Conference on Machine Learning and Cybernetics*, pp. 450-454.

Huang, R., Xi, L., Li, X., Liu, C.R., Qiu, H. and Lee, J., 2007. Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. *Mechanical systems and signal processing*, 21(1), pp.193-207.

Hung, N.Q.V. and Anh, D.T., 2007. Combining SAX and piecewise linear approximation to improve similarity search on financial time series. In *2007 International Symposium on Information Technology Convergence*, pp. 58-62.

Hubert. L. and Arabie. P., 1985. Comparing partitions. *Journal of classification*, 2(1), pp.193-218.

IBM, 2016. 10 Key Marketing Trends for 2017. Available at: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN.

Jain. A.K., 2010. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), pp.651-666.

Jain, A.K., Murty, M.N. and Flynn, P.J., 1999. Data clustering: a review. *ACM computing surveys*, 31(3), pp.264-323.

Jia, P., He, H. and Sun, T., 2008. Error restricted piecewise linear representation of time series based on special points. In *2008 7th World Congress on Intelligent Control and Automation*, pp. 2059–2064.

Junejo, I.N. and Al Aghbari, Z., 2012. Using SAX representation for human action recognition. *Journal of Visual Communication and Image Representation*, 23(6), pp.853-861.

Kan, M.S., Tan, A.C. and Mathew, J., 2015. A review on prognostic techniques for non-stationary and non-linear rotating systems. *Mechanical Systems and Signal Processing*, 62, pp.1-20.

References

Karamitopoulos, L., Evangelidis, G. and Dervos, D., 2010. PCA-based time series similarity search. *Data Mining*, pp. 255-276.

Karypis. G., Han. E.H. and Kumar. V., 1999. Chameleon: hierarchical clustering using dynamic modelling. *Computer*, 32(8), pp. 68-75.

Kaufman, L. and Rousseeuw, P.J., 2009. Finding groups in data: an introduction to cluster analysis, John Wiley & Sons.

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

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

Keogh, E., Lin, J., Fu, A., 2005. HOT SAX: efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining,* pp. 226–233.

Keogh, E., Lin, J., Fu, A.W., Van Herle, H., 2006. Finding Unusual Medical Time-Series Subsequences: Algorithms and Applications. *IEEE Transactions on Information Technology in Biomedicine*, 10(3), pp.429–439.

Khelif, R., Malinowski, S., Chebel-Morello, B. and Zerhouni, N., 2014, June. RUL prediction based on a new similarity-instance based approach. In *23rd International Symposium on Industrial Electronics*, pp. 2463-2468.

Kimura, A., Kashino, K., Kurozumi, T. and Murase, H., 2008. A quick search method for audio signals based on a piecewise linear representation of feature trajectories. *IEEE Transactions onAudio, Speech, and Language Processing*, 16(2), pp.396-407.

Knorr, E.M., Ng, R.T. and Tucakov, V., 2000. Distance-based outliers: algorithms and applications. *The International Journal on Very Large Data Bases*, 8(3-4), pp.237-253.

Kodinariya, T.M. and Makwana, P.R., 2013. Review on determining number of Cluster in K-Means Clustering. *International Journal of Advanced Research in Computer Science and Management Studies*, 1(6), pp.90-95.

Krishnaiah, V., Narsimha, G. and Chandra, N.S., 2014. Survey of classification techniques in data mining. *International Journal of Computer Sciences and Engineering*, 2(9), pp.65-74.

Kumar, V., 2005. Parallel and distributed computing for cybersecurity. *IEEE Distributed Systems Online*, 6(10).

Li, C.J. and Lee, H., 2005. Gear fatigue crack prognosis using embedded model, gear dynamic model and fracture mechanics. *Mechanical systems and signal processing*, 19(4), pp.836-846.

Liao, T.W., 2005. Clustering of time series data: a survey. *Pattern Recognition*, 38(11), pp.1857-1874.

Liao, L. and Kottig, F., 2014. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1), pp.191-207.

Lin, J., Keogh, E., Lonardi, S. and Chiu, B., 2003, June. 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*, pp. 2-11.

Lin, J., Keogh, E., Fu, A., Van Herle, H., 2005. Approximations to magic: finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems*, pp. 329–334.

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

Liu, D., Pang, J., Zhou, J., Peng, Y. and Pecht, M., 2013. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectronics Reliability*, 53(6), pp.832-839.

References

Liu, J., Saxena, A., Goebel, K., Saha, B. and Wang, W., 2010. An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries. In *Annual Conference of the Prognostics and Health Management Society*.

Likas, A., Vlassis, N. and Verbeek, J.J., 2003. The global k-means clustering algorithm. *Pattern Recognition*, 36(2), pp.451-461.

Lkhagva, B., Suzuki, Y. and Kawagoe, K., 2006a. Extended SAX: Extension of symbolic aggregate approximation for financial time series data representation. *DEWS2006*, 4A-i8.

Lkhagva, B., Suzuki, Y. and Kawagoe, K., 2006b. New time series data representation ESAX for financial applications. In *22nd International Conference on Data Engineering Workshops*, pp. s: x115-x115.

Lu, Y., Lu, S., Fotouhi, F., Deng, Y. and Brown, S.J., 2004. Incremental genetic K-means algorithm and its application in gene expression data analysis. *BMC bioinformatics*, 5(1), p.172.

Luo, J., Namburu, M., Pattipati, K., Qiao, L., Kawamoto, M. and Chigusa, S., 2003. Model-based prognostic techniques. In *Proceedings AUTOTESTCON 2003, IEEE Systems Readiness Technology Conference*, pp. 330-340.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the fifth Berkeley symposium Mathematical Statistics Probability*, 1, pp. 281–297.

Malinowski, S., Chebel-Morello, B. and Zerhouni, N., 2015. Remaining useful life estimation based on discriminating shapelet extraction. *Reliability Engineering & System Safety*, 142, pp.279-288.

Mampaey, M., and Vreeken. J., 2013. Summarizing categorical data by clustering attributes. *Data Mining and Knowledge Discovery*, 26(1), pp.130-173.

Mansor, M.N., Yaacob, S., Muthusamy, H. and Nisha, S., 2011. PCA-based feature extraction and k-NN algorithm for early jaundice detection. *System*, 1(1), pp. 25-29.

Medjaher, K., Tobon-Mejia, D.A. and Zerhouni, N., 2012. Remaining useful life estimation of critical components with application to bearings. *IEEE Transactions on Reliability*, 61(2), pp.292-302.

Milligan, G.W. and Cooper. M.C., 1986. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioural Research*, 21(4), pp.441-458.

Muller, M., 2007. Dynamic time warping. *Information Retrieval for Music and Motion*. pp.69-84, Springer Berlin Heidelberg.

Niennattrakul, V. and Ratanamahatana, C.A., 2007. On clustering multimedia time series data using k-means and dynamic time warping. In *Internal Conference on Multimedia and Ubiquitous Engineering*, pp. 733-738.

Niennattrakul, V. and Ratanamahatana, C.A., 2009. Shape averaging under time warping. *In 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Informaton Technology*, pp. 626-629.

Oppenheimer, C.H. and Loparo, K.A., 2002. Physically based diagnosis and prognosis of cracked rotor shafts. In *Component and Systems Diagnostics, Prognostics, and Health Management II*, 4733, pp. 122-133.

Orsagh, R., Roemer, M., Sheldon, J. and Klenke, C.J., 2004. A comprehensive prognostics approach for predicting gas turbine engine bearing life. In *ASME Turbo Expo 2004: Power for Land, Sea, and Air*, pp. 777-785.

Oyelade, O.J., Oladipupo, O.O. and Obagbuwa, I.C., 2010. Application of k Means Clustering algorithm for prediction of Students Academic Performance. *International Journal of Computer Science and Information Security*, 7(1), pp. 292-295.

Pamula, R., Dekaand, J.K. and Nandi, S., 2011. An outlier detection method based on clustering. In *Proceedings of the 2nd International Conference on Emerging Applications of Information Technology,* pp. 253-256.

Paparrizos, J. and Gravano, L., 2015. k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data,* pp. 1855-1870.

References

Paparrizos, J. and Gravano, L., 2017. Fast and accurate time-series clustering. *ACM Transactions on Database Systems*, 42(2), p.8.

Park, H.S. and Jun, C.H., 2009. A simple and fast algorithm for K-medoids clustering. *Expert systems with applications*, 36(2), pp.3336-3341.

Parra, L., Deco, G. and Miesbach, S., 1996. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation*, 8(2), pp.260-269.

Patil, M.A., Tagade, P., Hariharan, K.S., Kolake, S.M., Song, T., Yeo, T. and Doo, S., 2015. A novel multistage Support Vector Machine based approach for Li ion battery remaining useful life estimation. *Applied energy*, 159, pp.285-297.

Pavlidis, T. and Horowitz, S.L., 1974. Segmentation of plane curves. *IEEE transactions on Computers*. 23(8). pp.860-870.

Pelleg, D. and Moore, A.W., 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: *International Conference of Machine Leaning*, pp. 727-734.

Petitjean. F., Ketterlin. A. and Gançarski. P., 2011. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3), pp.678-693.

Perlibakas, V., 2004. Distance measures for PCA-based face recognition. *Pattern recognition letters*, 25(6), pp.711-724.

Rai. P. and Singh. S., 2010. A survey of clustering techniques. *International Journal of Computer Applications*, 7(12), pp.1-5.

Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336), pp.846-850.

Sakoe, H. and Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1), pp.43-49.

Sakurai, Y., Yoshikawa, M. and Faloutsos, C., 2005. FTW: fast similarity search under the time warping distance. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 326-337.

Saligrama, V. and Chen, Z., 2012. Video anomaly detection based on local statistical aggregates. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2112-2119.

Sankavaram, C., Pattipati, B., Kodali, A., Pattipati, K., Azam, M., Kumar, S. and Pecht, M., 2009. Model-based and data-driven prognosis of automotive and electronic systems. In *IEEE International Conference on Automation Science and Engineering*, pp. 96-101.

Santos, J.M. and Embrechts. M., 2009. On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International Conference on Artificial Neural Networks*, pp. 175-184.

Saxena, A. and Goebel, K., C-MAPSS data set. 2008. *NASA Ames Prognostics Data Repository*.

Scholkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J. and Williamson, R.C., 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7), pp.1443-1471.

Sfetsos, A. and Siriopoulos, C., 2004. Time series forecasting with a hybrid clustering scheme and pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 34(3), pp.399-405.

Shahbaba. M. and Beheshti. S., 2014. MACE-means clustering. *Signal Processing*, 105, pp.216-225.

Shanghai Stock Exchange. Available at: http://english.sse.com.cn/.

Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y. and Herawan, T., 2014. Big data clustering: a review. In *International Conference on Computational Science and Its Applications*, pp. 707-720.

References

Smith, R., Bivens, A., Embrechts, M., Palagiri, C. and Szymanski, B., 2002. Clustering approaches for anomaly based intrusion detection. *Proceedings of intelligent engineering systems through artificial neural networks*, pp.579-584.

Song, Q., Niand, J. and Wang, G., 2013. A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(1), pp.1-14.

Sotoca, J.M. and Pla, F., 2010. Supervised feature selection by clustering using conditional mutual information-based distances. *Pattern Recognition*, 43(6), pp.2068-2081.

Steinwart, I., Hush, D. and Scovel, C., 2005. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6, pp.211-232.

Sun, Y., Li, J., Liu, J., Sun, B. and Chow, C., 2014. An improvement of symbolic aggregate approximation distance measure for time series. *Neurocomputing*, 138, pp.189-198.

Sumathi, S. and Sivanandam, S.N., 2006. Introduction to data mining principles. *Studies in Computational Intelligence*, 29. Springer, Berlin, Heidelberg.

Swanson, D.C., Spencer, J.M. and Arzoumanian, S.H., 2000. Prognostic modelling of crack growth in a tensioned steel band. *Mechanical systems and signal processing*, 14(5), pp.789-803.

Tan P.N., Steinbach, M. and Kuma, V., 2013. Data Mining Cluster Analysis: Basic Concepts and Algorithms. In *Introduction to Data Mining*.

Tayebi, H., Krishnaswamy, S., Waluyo, A.B., Sinha, A. and Gaber, M.M., 2011. Ra-sax: resource-aware symbolic aggregate approximation for mobile ECG analysis. In *2011 12th IEEE International Conference on Mobile Data Management*, pp. 289-290.

Tibshirani, R., Walther, G. and Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), pp.411-423.

Tsui, K.L., Chen, N., Zhou, Q., Hai, Y. and Wang, W., 2015. Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*.

Wang, Q. and Megalooikonomou, V., 2008. A dimensionality reduction technique for efficient time series similarity analysis. *Information systems*, 33(1), pp.115-132.

Wang, T., Yu, J., Siegel, D. and Lee, J., 2008. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *2008 International Conference on Prognostics and Health Management*, pp. 1-6.

Wang X, Smith K, Hyndman R (2006) Characteristic-based clustering for time series data. Data Mining and Knowledge Discovery, 13(3), pp.335-364.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P. and Keogh, E., 2013. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2), pp.275-309.

Watson, M., Byington, C., Edwards, D. and Amin, S., 2005. Dynamic modelling and wear-based remaining useful life prediction of high power clutch systems. *Tribology Transactions*, 48(2), pp.208-217.

Wu X, Kumar V, Quinlan J-R, Ghosh J, Yang Q, Motoda H, McLachlan G-J, Ng A, Liu B, Philip S-Y, Zhou Z.H (2008) Top 10 algorithms in data mining. Knowledge and Information Systems, 14(1), pp.1-37.

Xiao, Q., Chu, C.Q. and Li, Z., 2017. Time series prediction using dynamic Bayesian network. *Optik-International Journal for Light and Electron Optics*, 135, pp.98-103

Xu, R. and Wunsch, D., 2005. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3), pp.645-678.

Xue, F., Bonissone, P., Varma, A., Yan, W., Eklund, N. and Goebel, K., 2008. An instance-based method for remaining useful life estimation for aircraft engines. *Journal of failure analysis and prevention, 8(2)*, pp.199-206.

References

Yan M, Ye K (2007) Determining the number of clusters using the weighted gap statistic. Biometrics, 63(4), pp.1031-1037.

Yang, K. and Shahabi, C., 2004, November. A PCA-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases,* pp. 65-74.

Yao, X. and Wei, H.L., 2016, September. Off-line signature verification based on a new symbolic representation and dynamic time warping. In *22nd International Conference on Automation and Computing*, pp. 108-113.

Yeung K-Y, Ruzzo W-L (2001) Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. Bioinformatics17(9), pp.763-774.

Yu, G., Qiu, H., Djurdjanovic, D. and Lee, J., 2006. Feature signature prediction of a boring process using neural network modeling with confidence bounds. *The International Journal of Advanced Manufacturing Technology*, 30(7-8), pp.614-621.

Zhang, H., Ho, T.B., Lin, M.S., 2004. A Non-parametric wavelet feature extractor for time-series classification. In *Proceedings of the Eighth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 595–603.

Zhang, Q. and Couloigner, I., 2005. A new and efficient k-medoid algorithm for spatial clustering. In *International Conference on Computational Science and Its Applications*, pp. 181-189, Springer Berlin Heidelberg.

Zhang, Q., Tse, P.W.T., Wan, X. and Xu, G., 2015. Remaining useful life estimation for mechanical systems based on similarity of phase space trajectory. *Expert Systems with Applications*, 42(5), pp.2353-2360.

Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. In ACM SIGMOD Record. 25(2), pp. 103-114.

Zhao, F., Tian, Z. and Zeng, Y., 2013. Uncertainty quantification in gear remaining useful life prediction through an integrated prognostics method. *IEEE Transactions on Reliability*, 62(1), pp.146-159.

Zhao, Z., Liang, B., Wang, X. and Lu, W., 2017. Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering and System Safety*, 164, pp.74-83.

Zheng Y, Jeon B, Xu D, Wu Q-M, Zhang H, (2015) Image segmentation by generalized hierarchical fuzzy C-means algorithm. Journal of Intelligent and Fuzzy Systems, 28(2), pp.961-973.

Zhou, T., Gao, S., Wang, J., Chu, C., Todo, Y. and Tang, Z., 2016. Financial time series prediction using a dendritic neuron model. *Knowledge-Based Systems*, 105, pp.214-224.

Zio, E. and Di Maio, F., 2010. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. *Reliability Engineering & System Safety*, 95(1), pp.49-57.