Title
Uncover the Rhythmic and Arrhythmic Dynamics in Complex Systems

Author
Wang, Xiaodong

Publication Date
2021

Peer reviewed|Thesis/dissertation

Uncover the Rhythmic and Arrhythmic Dynamics in Complex Systems

By

XIAODONG WANG

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

BIOSTATISTICS

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Fushing Hsieh, Chair

Patrice Koehl, Member

Emilio Ferrer, Member

Committee in Charge

2021

To my family.

Contents

List of Figures

List of Tables

Uncover the Rhythmic and Arrhythmic Dynamics in Complex Systems

Abstract

In this dissertation, I computationally analyze the dynamic pattern for time series belonging to two distinct settings: rhythmic and arrhythmic, and resolving a Machine Learning topic: Multiclass Classification (MCC). The primary study of arrhythmic signals is carried out via change-point analysis on dynamics of various complex systems, while the study of rhythmic patterns is focused on gait dynamics. For arrhythmic time series with unknown and unspecified non-stationarity, an approach is proposed to partition the whole time span into homogeneous periods where the underlying distribution is identical. In contrast, for the rhythmic time series, the goal is to detect all rhythmic cycles precisely. Although an encoding-and-decoding technique is implemented from local to global, the methodologies and the information content under the two settings are rather different. Under the arrhythmic setting, the number and temporal locations of all identified change points are the primary parts of information. Especially, a group of time points is marked as events of interest for characterizing segments in high and low intensity. The number of change-points is determined by information criteria based on maximum likelihood functions derived based on Geometric distributions of recursive time between two successive events. While under the rhythmic setting, structural components constitute a rhythm, so the cyclic structures and rhythmic patterns are the major parts of information. The rhythmic time series is discretized by a set of symbols, and the deterministic pattern is embraced so that the symbolic trajectory is in low Kolmogorov or Lempel-Ziv complexity. So far, the homogeneous segments or rhythmic cycles are detected without explicit labeling. For the completeness of the study, geometric structure of structured data with labels is investigated as a basis of facilitating error-free classification with potential multiple candidate labels.

Acknowledgments

I would like to express my deepest gratefulness to my Ph.D. advisor, Professor Fushing Hsieh, for his intelligent guidance and endless support during my Ph.D. career. His novel idea of thinking, enthusiasm in investigating different research fields, and rigorous attitude towards data analysis deeply influenced me and inspired me to explore widely and get my hands dirty in real data. We have finished six papers in the past few years, and now I can lead projects and conduct my research independently. Without his generous help and kindness, I cannot make so much progress. It is a great pleasure for me to work under his supervision.

I would like to express my appreciation to my committee members, Professor Patrice Koehl and Professor Emilio Ferrer, for their help and constructive comments on my dissertation and oral exam.

I would gratefully thank my collaborators in the UC Davis Center for Mind and Brain- Patrick Dwyer, Professor Clifford Saron, and Professor Susan Rivera for their fruitful collaborations and discussions. Their solid background and knowledge excite my interest to resolve the scientific problem in psychology.

I would sincerely thank Professor Jie Peng for her encouragement and thoughtful suggestions during my graduate study. I would thank other faculty members in the statistics department from whom I have learned a lot. I am also thankful to all my graduate school friends. It is them who make my life meaningful and joyful in Davis. The department for me is not only a place to absorb knowledge but more like a family.

Last but not the least, I would like to thank my parents, Jianxin Wang and Caiping Xiao, and my fiancee, Chunzi Pan, for their unconditional love and invaluable companions in my life. The dissertation is dedicated to them.

CHAPTER 1

Introduction

It has been claimed by John Tukey early in 1962 that data analysis is "... procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise and more accurate, and all the machinery and results of statistics which apply to analyzing data". Nowadays, with the rapid increase in the amount of information, real-world data analysis has been unprecedentedly challenged by either the amount and the complexity of the data resources. Major issues of data analysis are still remained and needed to be resolved accordingly.

The signals coming from the real world often have very complex dynamics. In psychology, auditory stimuli of different intensities provide the dynamics of electroencephalographic(EEG) activity differing from typical development and autism spectrum development. In finance, the volatility of the price of a single stock would transit the uncertainty of risk to other associated stocks. In the musculoskeletal analysis of gait, dynamic pattern of movement is reflected by wearable sensors measuring accelerator and gyroscope from different joints. One undergoing challenge is then to manage information and understand the underlying dynamics from a complex system.

A complex system consists of a set of possible states which determine how the system changes. Especially, the analysis of deterministic and stochastic patterns is a basic topic in nonlinear dynamic fields. As a tool for analyzing dynamical systems, symbolic dynamics was studied by [**87**] in partitioning the space into finite phases and encoding a trajectory with a discrete alphabet. The encoding procedure which is designed to reduce redundancy and extracting features is now a basis of various information processing and a key to understanding properties of the dynamical system. However, it is unresolved to define a good encoding in real data application [**46**], and more importantly, to reveal the dynamic signals based on the symbolic trajectories. If it is mysterious to discover the rule that generates the real data, can we extract the underlying information and at least understand the complex system better?

In this dissertation, I aim to propose data-driven signal processing approaches to investigate symbolic dynamics and heterogeneity property of data systems from different fields of complex systems. The dynamic pattern can be basically divided into two settings- arrhythmic and rhythmic. Following the concepts of symbolic dynamics, I analyzed both types of dynamics by encoding the time series from local parts of the system into a symbolic trajectory and then decoding or summarizing the information globally. It is noted that the encoded trajectory is able to filter out irrelevant or noisy signals and present the information simply and concisely. In the decoding phase, the deterministic structures are extracted such as the cutpoints between two consecutive time segments and the transition probability between symbols within each segment. The motivation of applying the algorithm from locally to globally is based on the integration property of complex systems. Specifically, the components of a system interact with each other and form collective behaviors. The idea of dividing-and-concurring or from-local-to-global exactly reflects the escalation process of understanding a system- from the simple to the complex.

Though similar procedures are employed under arrhythmic and rhythmic settings, the information is extracted and collected differently. For arrhythmic signals, the number and locations of change points play a crucial role in revealing the abrupt distributional changes. The idea is then refined as a generalized methodology for change point analysis. By iteratively subsampling a group of data points and aggregating the change point locations for each binary process, the proposed approach is able to detect distributional changes without prior knowledge of the underlying distributions. While, for rhythmic signals, the recurrent pattern of gait time series implicates the deterministic behavior. Especially, there exists a particular landmark by which the gait time span can be sectioned into rhythmic cycles although the duration of each cycle evolves over time stochastically. It claims that this aspect of difference between these two settings critically relies on whether structural components exist and further constitute a rhythm.

Motivation on Stock Dynamics and Network Analysis. Since the major financial crisis in the last century, a huge amount of attention from researchers or scientists has been received to work on financial markets in order to avoid such risks in the future. As a complex system of finance, it has not been well understood so far that how stocks are interacting with each other and transit volatility into the whole market. Volatility of financial stock is referring to the degree

of uncertainty or risk embedded within a stock's dynamics. The study of volatility is essential to model the dynamic in stock or a financial market. A nonparametric regime-switching model is firstly proposed to study the stochastic volatility for a single stock and then a network is established to illustrate which stocks stimulate or even promote volatility on others. The regime-switching model was advocated by [**48**, **52**] to incorporate stochastic volatility into stock price modeling, such as the Hidden Markov Model (HMM) [**95**]. Specifically, the distributional changes between different volatility regimes are determined under the assumption of Markov Chain. However, these models usually required too many parameters or a strong assumption of the underlying dynamic structure, which makes it hard to interpret the analysis result. In contrast, my nonparametric approach is able to detect distributional changes between low and high volatility without assuming any underlying distribution or Markovian property. After that, the causality between two stocks is measured based on the recovered dynamic trajectories. Finally, networks dealing with distinct financial implications are established to represent different aspects of global connectivity among all stocks in S&P500.

Motivation on Multivariate Financial Analysis. The volatility modeling for a single stock is extended to multivariate settings. My motivation is because there are multiple assets highly associated and their nonlinear dependence is necessary to be captured. For instance, transaction volume had been increasingly used as a cause of stock return volatility. Thus, it is a barrier to study the stock price without taking into account the trading number and volume. Moreover, the pair-wise dependence in the S&P500 network is measured based on the recovered dynamics of every single process, so the nonlinear dependence can be easily overwhelmed by the integrated microstructure noises. In this chapter, a nonparametric approach is proposed to dissect multivariate time series in order to discover multiple dynamics phases when the joint distribution varies. It shows that this expanded approach can successfully not only map out volatile periods but also provide potential associative links between stocks.

Motivation on Change Point Analysis. Under the direction of regime-switching model, the research problem is then generalized to change point analysis. The goal is to estimate the number of change points and their locations so long as the consecutive distributions are different. Change points as temporal locations of such occurrences and their multiplicity are key parts of

deterministic structures of the time series under study. The task of change point detection is more general since it does not restrict the number of hidden states. So far, it has been playing a crucial role in diverse fields including bioinformatics, behavioral science, neuroimage, climate science, finance, and speech recognition. In this chapter, a nonparametric approach is proposed to detect the abrupt changes of joint distribution without imposing prior distributional knowledge. A structural subsampling procedure is firstly developed such that the observations are encoded into multiple sequences of Bernoulli variables. So, the maximum likelihood approach can be applied to detect change points on each Bernoulli process separately. Then, aggregation statistics are proposed to collect change-point results from all individual univariate time series. The theoretical work shows that the proposed statistic is favorable in terms of controlling false change-point discovery and holds consistency property as the sample size goes into infinity.

Motivation on Gait Analysis. As an application for rhythmic signals, the idea of from-local-to-global is extended to unravel the systematic dynamics in gait analysis. The local movement trajectories from an apart of skeleton or joint are first discovered and then aggregated to reconstruct a global system for an individual's gait dynamic. It is known that rhythmic biomechanics and the dependency between different joints form the primary deterministic structures. One of the scientific goals is to detect the early-stage illness or disorder condition for the clinical trial. For example, an imbalanced correspondence between left and right food may reflect some potential diseases like Alzheimer's. The rhythmic cycle is firstly discovered by encoding the gait time series from one joint separately. Then, the joint-to-joint dependency is visualized by stacking the resulted code sequences. Specifically, the encoding procedure is designed to represent the rhythmic pattern in a low Kolmogorov's complexity, and the visualization implies evident deterministic structure within each rhythmic cycle. Another scientific goal of gait analysis is identification. The goal is to recognize one's walking style as a signature of gait. Since such a signature is hard to be faked, it can be used to strengthen security inspection in the future as an alternative to facial or finger recognition. Another encoding scheme is proposed to symbolize the gait time series. The most frequent symbols are then extracted as the key information for gait identification. It shows that the selected symbols work as a gait signature that can be used to perfectly separate one individual from others.

Motivation on Multiclass Classification. Nowadays, Machine Learning grew out of the field of artificial intelligence and has gained a stronghold in data analysis. Given that human experts hardly make mistakes in classification or categorization, the machine is still impossible to compete with human intelligence until very recently. It motivates us to consider why such mistakes and how to decrease the risk of decision-making. I demonstrate that possible answers under Multiclass Classification (MCC) setting. MCC is defined by a collection of labeled point clouds specified by a feature set. The group of features is chosen to capture certain system characters or to reflect certain existing experiences or knowledge pertaining to the system. However, the expert-made labeling may contain certain information that the machine is unable to learn. For example, the distinction between the labels and the presence of heterogeneity within a label group. In this chapter, I aim to construct a label-embedding tree to illustrate the labeling geometry. Such tree geometry in turn sheds light on explainable knowledge on why and how labeling comes about and facilitates error-free prediction with potential multi-scale candidate labels supported by the data.

Organization. The dissertation consists of five chapters which are organized as follows. In Chapter 2, an encoding-and-decoding procedure is present to reveal the volatility dynamics of every single stock of S&P500. After that, stock networks dealing with distinct financial implications are established to represent different aspects of global connectivity among all stocks in S&P500. The encoding approach is then generalized to multivariate stock settings in Chapter 3. I analyze the relationship from returns, trading volume, and transaction number of a single, as well as of multiple stocks. In Chapter 4, the approach is employed under the setting of change point detection, and theoretical work is shown accordingly. In Chapter 5, I studied the deterministic and stochastic pattern embedded in the gait system, and resolve two questions specifically- how to precisely differentiate gait signatures of many individuals, and how to represent an individual's gait dynamics for authentication. In Chapter 6, the data-driven intelligence is applied by demonstrating coarse- and fine-scale geometric information content of MCC in Major League Baseball(MLB). A conclusion and some future research topics related to this dissertation are organized in the last chapter.

CHAPTER 2

From Single Stock Volatility to Networks in S&P500

2.1. Introduction

To discover the mystery of the stock dynamics, financial researchers focus on stock returns or log returns. Black and Scholes **[19]** proposed in their seminal work to use stochastic processes in modeling stock prices. One particular model of focus is the geometric Brownian motion (GBM), which assumes all log-returns being normally distributed. That is, if a time series of the price of a stock is denoted as $\{P_t\}_t$, the GBM modeling structure prescribes that

$$log\frac{P_t}{P_{t-1}} \sim N(\mu, \sigma^2)$$

Later Merton **[85]** extended Black and Scholes' model by involving time-dependent parameters for accommodating potential serial correlations. Further, in order to go beyond normal distribution, models belonging to a broader category, including a general Levy process or particular geometric Levy process model **[38]**, become popular and appropriate alternatives by embracing stable distribution with heavy tails. Since the independent increments property of Brownian motion or Levy process, returns over disjoint equal-length time intervals remain identically independently distributed (i.i.d). So, the independent increment property restricts modeling stochasticity to be invariant across the entire time span. However, it is well known that the distributions of returns are completely different over various volatility stages. Thus, these models are prone to fail in capturing extreme price movements **[52]**.

Research attempts from various perspectives have experimented to make stock price modelings more realistic. One fruitful perspective is to incorporate stochastic volatility into stock price modeling. From this perspective, regime-switching or hidden-state models are proposed to govern the stock price dynamics. The regime-switching model can be represented by the distributional

changes between a low-volatility regime and a more unstable high-volatility regime. In particular, different regimes are characterized by distinct sets of distributional modeling structures. One concrete example of such modeling is the Hidden Markov Model(HMM). HMM appeals increasingly to researchers due to its mathematical tractability under the assumption of Markovian. Its parametric approach has gained popularity and its parameter estimation procedures have been discussed comprehensively. For instance, Hamilton [**48**] described autoregressive (AR) models under Markov regime-switching structure. Hardy [**52**] offered Markov regime-switching lognormal model by assuming different normal distribution within each state,

$$log\frac{P_t}{P_{t-1}}|s \sim N(\mu_s, \sigma_s^2)$$

where s indicates the hidden states for $s = 1, 2, ...$ Fine et al. [**41**] developed hierarchical HMM by putting additional sources of dependency in the model. So, every state is composed of substates with multiple levels of dependencies. To further increasing the degree of modeling complexity in stochastic volatility, another well-known financial model was related to volatility clustering [**36**]. For instance, GARCH models have been studied to model the time-varying conditional variance of asset returns [**20**]. However, such a complicated dynamic structure usually involves a large number of parameters. This modeling complexity renders the model hard to interpret. In contrast, nonparametric approaches are still scarce in the literature due to the lack of tractability and involvement of many unspecified characteristics [**112**].

In this chapter, we take up a standpoint right in between the purely parametric and nonparametric modelings. We adopt the research platform of regime-switching models but aim to develop an efficient nonparametric procedure to discover the dynamic volatility underlying any asset returns. The idea is motivated by a nonparametric approach, named Hierarchical Factor Segmentation(HFS) [**59**,**61**], to mark extreme large returns as 1 and others 0, and then partition the resultant 0-1 Bernoulli sequence into alternating homogeneous segments. HFS takes advantage in transforming the returns into a 0-1 process with time-varying Bernoulli parameters, so parametric approaches such as likelihood-based function can be applied to fit each segment respectively. However, it is unclear in HFS to define a "large" return that should be marked, which makes the implementation

limited in application. Another limitation of HFS is that there are only two kinds of regimes alternatingly evolving across the entire temporal span. This limitation is sharply contrasting with the shared limitation of the regime-switching models or HMM, in which there is no data-driven way of determining the number of underlying regimes or hidden states.

We propose an encoding-and-decoding approach to resolve the issues tied to the aforementioned limitations simultaneously. The encoding procedure is done by iteratively marking the returns at different thresholding quantile levels, so the time series can be transformed into multiple 0-1 processes. In the decoding phase, a searching algorithm in conjunction with model selection criteria is developed to discover the dynamic pattern for each 0-1 process separately. Finally, the underlying states are revealed by aggregating the decoding results via cluster analysis. It is remarked that the nonparametric approach is able to discover both light-tail and heavy-tail distributional changes without assuming any dynamic structure or Markovian properties. Though the proposed method is favorable under independence or exchangeability conditions, our numerical experiments show that the approach still works for settings with the presence of weak serial dependency, which can be checked by testing the significance of lagged correlation in practice.

Another contribution is that a searching algorithm is developed to partition a 0-1 process into segments with different probability parameters. Therefore, our computational development is a change point analysis on a sequence of Bernoulli variables with the number of change points being large and unknown. For such a setting and its like, the current searching algorithm is infeasible, such as bisection procedures [**91**, **118**]. As an alternative to the hierarchical searching strategy, our proposed search algorithm concurrently generates multiple segments with only a few parameters. The optimal partition of homogeneous segments is ultimately obtained via model selection.

The chapter is constructed as follows. In Section 2.2, we introduce the asymptotic theory for recurrent time distributions. In Section 2.3, we review the HFS and develop the new searching algorithm that can handle multiple-states decoding. In Section 2.4, we present the main approach in modeling distributional changes. In Section 2.5, real data analysis is performed and various networks with nonlinear association are established to illustrate diverse aspects of relational patterns among S&P500 stocks. A conclusion and remarks are given in Section 2.6.

2.2. Recurrent Time Distribution

2.2.1. Homogeneous Time Series. Given a large data of stock price at an even interval of time and its consequential calculated returns with length N, we can encode the continuous time series into a 0-1 binary sequence of length N where 1 indicates an observation of a rare event, and 0 otherwise. In such one-dimensional stock time series, the rare event is defined by extremely large values of absolute stock returns, so that the binary sequence can represent the frequency of return volatility. Note that a period with a high-frequent appearance of 1's may indicate a volatility clustering.

As advocated by the Black-Scholes model, stock's return stochastic process is often modeled by geometric Brownian motion, or more generally geometric Levy processes. The returns are i.i.d. or exchangeable under the model assumptions. It motivates an invariance theorem for the waiting time between successive extreme large returns. If we look at the time of observing a successive 1's under the assumption of exchangeable returns, it was proved that the waiting time is asymptotically independent and its finite empirical distribution converges almost surely to a geometric distribution [**24**].

Consider a N-length series of stock returns $\{X_t\}_{t=1}^N$. Suppose M out of N objects are selected randomly as 1's, and the unseleted $N-M$ as 0's. Denote the recurrent time of two successive 1's as R^N, so there obtained $M+1$ recurrent time sequence $R_1^N, R_2^N, ..., R_{M+1}^N$. Assume the waiting time can be 0 if two 1's appear consecutively, $R_1^N = 0$ if $X_1 = 1$, and $R_{M+1}^N = 0$ if $X_N = 1$. Due to the exchangeability assumption of $\{X_t\}_{t=1}^N$, $R_1^N, R_2^N, ..., R_{M+1}^N$ are exchangeable as well.

THEOREM 2.2.1. *If $N \to \infty$ and $M \to \infty$ in a way such that $\frac{M}{N} \to p \in (0,1)$, then, for any $t \geq 1$,*

$$(R_1^N, R_2^N, ..., R_t^N) \xrightarrow{d} (R_1, R_2, ..., R_t) \tag{2.1}$$

where $(R_1, R_2, ..., R_t)$ are independent and identically geometric distribution with parameter p.

The proof sees Theorem 2.1 in [**24**]. When N and M go to infinity in a way that $M \sim Np$, the recurrent time becomes asymptotically independent and converge to a geometric distribution with parameter $p = M/N$.

In one-dimensional stock returns, a fixed proportion from all the time points can be selected as events of interest. For example, α and β quantile is set to cut the lower and upper tail of the distribution, where $0 < \alpha < 0.5 < \beta < 1$ and $\alpha + (1 - \beta) < 1$. So that all time stamp has the same probability $p = \alpha + (1 - \beta)$ to be marked as 1. An excursion process is defined by,

$$E_t = \begin{cases} 1 & X_t \leq \alpha\text{-}quantile,\ X_t \geq \beta\text{-}quantile \\ 0 & Otherwise \end{cases} \tag{2.2}$$

where $\{E_t\}_t$ is the resultant 0-1 sequence after labeling absolute large return as 1 and 0 otherwise.

However, the exchangeability of successive returns is easily violated due to the well-known stochastic volatility in finance. Returns should only be considered exchangeable locally, and rapid time-varying volatility is evidently observed **[23]**.

2.2.2. Mixed Geometric Distribution. Complicated models are investigated to evaluate the mechanism of stock return. Markov property, more or less, plays a significant role in modeling time-varying volatility. Instead, Hierarchical Factor Segmentation(HFS) is proposed to search for alternating hidden regions without assuming any Markov property **[59]**. The idea is to admit distributional heterogeneity embedded behind the distribution of stock returns. After encoding the time series in the same way described above, HFS is implemented to label each time point by an index of hidden regions. By assuming that the conditional distributions are exchangeable within a hidden region, a corollary for heterogeneous time series is shown as a direct consequence of Theorem 2.2.1.

Assume there are only k hidden regions, denoted by $S_1, S_2, ..., S_k$. Select a fix size of samples M from all the samples. Denote the sample selected from region S_j having size M_j, for $j = 1, 2, ..., k$. So, $\sum_{j=1}^{k} M_j = M$.

COROLLARY 2.2.1. *If $N \to \infty$ and $M \to \infty$ in a way such that $\frac{M_j}{N} \to p_j \in (0, 1)$, for $j = 1, 2, ..., k$, then, for any $t \geq 1$,*

$$(R_1^N, R_2^N, ..., R_t^N | S_j) \xrightarrow{d} (R_1, R_2, ..., R_t | S_j) \tag{2.3}$$

where $(R_1, R_2, ..., R_t|S_j)$ *are independent and identically geometric distribution with parameter* p_j *given hidden region* S_j*.*

Moreover, by further assuming identicality of conditional distribution given a hidden region, i.e. the cumulative distribution function denoted by F_j given region S_j, with an appropriate choice of α and β advocated in (2.2), ratio $\frac{M_j}{N}$ converges to a constant almost surely,

$$\frac{M_j}{N} \to p_j = (\int_{-\infty}^{\alpha\text{-}quantile} + \int_{\beta\text{-}quantile}^{\infty}) dF_j \tag{2.4}$$

It is easy to generalize the thresholds to involve one-sided tale excursions, for instance, to set $\alpha = 0$ and $\beta \in (0, 1)$ to focus on positive returns or upper tail excursions. If the thresholds are set too extreme, then only fewer excursive returns can stand out. As a result, the excursion process is too simple to preserve enough information about the volatility dynamics due to the reduction of sample size. While, if the quantile value is set close to the median, then the dynamic pattern is overwhelmed by irrelevant information or noise. There is an inevitable trade-off between the magnitude of the sample size and the amount of information about excursive events. Our remedy to this problem is to systematically apply a series of thresholds and encode the time series returns into multiple binary (0-1) excursion processes. For the completeness of the analysis, we will discuss a searching algorithm in conjunction with model selection criteria in the section below, which is the key in the decoding phase.

2.3. Region Switching Model

2.3.1. The searching algorithm. Suppose a 0-1 excursion process has been obtained. In this subsection, we discuss how to search for potential temporal segmentation. As the study involving multiple change points, we aim to detect abrupt distributional changes from one segment of low-volatility regime to another segment of high-volatility regime. To properly accommodate a potentially large number of unknown change points due to the recursive emissions of volatility, and to effectively differentiate the alternating volatility-switching patterns, the Hierarchical Factor Segmentation(HFS) was employed to partition the excursion process into a sequence of high and low event-intensity segments [**59**]. The original HFS assumes that there exist only two kinds of hidden

states within the returns corresponding to low-volatility and high-volatility regimes. Though the assumption is plausible within a short time period, its potential becomes limited when the time series of returns is lengthy and embracing more complicated regime-specific distributions. In this subsection, we expand the HFS by incorporating a more generalized searching algorithm to handle the scenarios of multiple states.

Denote the entire 0-1 excursion process sequence as $\{E_t\}_{t=1}^{N}$. The recursive recurrent time between two successive 1's of $\{E_t\}_{t=1}^{N}$ is recorded into a sequence, denoted as $\{R_t\}_{t=1}^{M^*}$. It is noted that the recurrent time can be 0 if two 1's appear consecutively. As such, the length of $\{R_t\}_{t=1}^{M^*}$ is $M^* = M+1$ where M is the number of 1's in $\{E_t\}_{t=1}^{N}$. To make the notations consistent, we denote $\{E_t^i\}_{t=1}^{M^*}$ as the i-th coding sequence for $i = 1, 2, ...$ and its corresponding recurrent time sequence as $\{R_t^i\}_{t=1}^{M^{i*}}$ for $i = 1, 2, ...$, respectively.

Suppose that the number of internal states is k and $k > 1$. Then, there are k tuning parameters are required in the searching algorithm given below. Denote the first thresholding parameter vector as $T = (T_1, T_2, ..., T_{k-1})$ where $T_1 < T_2 < ... < T_{k-1}$, and the second thresholding parameter as T^*. The algorithm is described in **Algorithm 1**.

Algorithm 1 multiple-states searching

1.Define events of interest and encode the time series of return into a 0-1 digital sequence $\{E_t\}_{t=1}^{N}$ with 1 indicating an event and 0 otherwise.

2.Calculate the recurrence time in $\{E_t\}_{t=1}^{N}$ and denote the resultant sequence as $\{R_t\}_{t=1}^{M^*}$.

3. For loop: cycle through $i = 1, 2, ..., k-1$:

i. Transform $\{R_t\}_{t=1}^{M^*}$ into a new 0-1 digital strings $\{E_t^i\}_{t=1}^{M^*}$ via the second-level coding scheme:

$$E_t^i = \begin{cases} 1 & R_t \geq T_i \\ 0 & otherwise \end{cases}$$

ii. Upon code sequence $\{E_t^i\}_{t=1}^{M^*}$, take code digit 1 as another new event and recalculate the event recurrence time sequence $\{R_t^i\}_{t=1}^{M^{i*}}$

iii. If a recursive time $R_t^i \geq T^*$, then record its associated time segment in $\{E_t^i\}_{t=1}^{M^*}$, denoted as Seg_i where $Seg_i \subset \{1, ..., n\}$.

4. The k internal states are returned by $S_1 = Seg_1$, $S_2 = Seg_2 \backslash Seg_1$, ..., $S_{k-1} = Seg_{k-1} \backslash Seg_{k-2}$, and $S_k = \{1, ..., N\} \backslash Seg_{k-1}$.

A sequence of Gaussian distributed observations are generated with mean 0 and variance varying under different unknown states in Figure 2.1(A). A pair of thresholds α-*quantile* $= -2$ and β-*quantile* $= 2$ are applied to code the observations via (2.2), so a sequence of recursive time is obtained in Figure 2.1(B). The first-level parameter T_i are set to control the event-intensity that we aim to partition for $i = 1, ..., m-1$, see thresholds T_1 and T_2 in Figure 2.1(B). If T_i takes its maximum T_{k-1}, then a high-intensity segment is separated from other level segments, see T_2 in Figure 2.1(B). Decreasing the value of T_i from T_{k-1} to T_1 to implement a series of partitions, so the multiple intensity levels of phases can get separated. In this example, T_1 is set to partition high- and median-intensity from the low-intensity segment.

In the second level of recursive time calculation, $\{R_t^i\}_{t=1}^{M^{i*}}$ are calculated for $i = 1, ..., k-1$. If R_t^i is above the second-level threshold T^*, the segment corresponds to a period with low-intensity events. So, for a fixed T_i, T^* is set to decide which phases having relatively low intensity, so the rests are in high intensity. It is noticed that $Seg_j \subset Seg_i$, for $j > i$. It is because if a recursive time is greater than T_j, it is greater than T_i as well. So, by applying the same parameter T^*, the low-intensity segment Seg_j is wider than Seg_i. In Figure 2.1(B), for example, Segment2 is wider than Segment1, so the median-intensity segment is obtained by $Seg2 \backslash Seg1$.

2.3.2. Model selection. Multiple volatility phases of time series S_j, $j = 1, ..., k$ are computed and achieved by applying the searching algorithm. Assuming join distribution is exchangeable within each volatility phase, the recursive distribution $\{R_t\}_{t \in S_j}$ converges to a geometric distribution with parameter p_j as the sample size goes to infinity. Maximized Likelihood Estimation(MLE) and Method of Moment(MOM) give the same estimator for p_j,

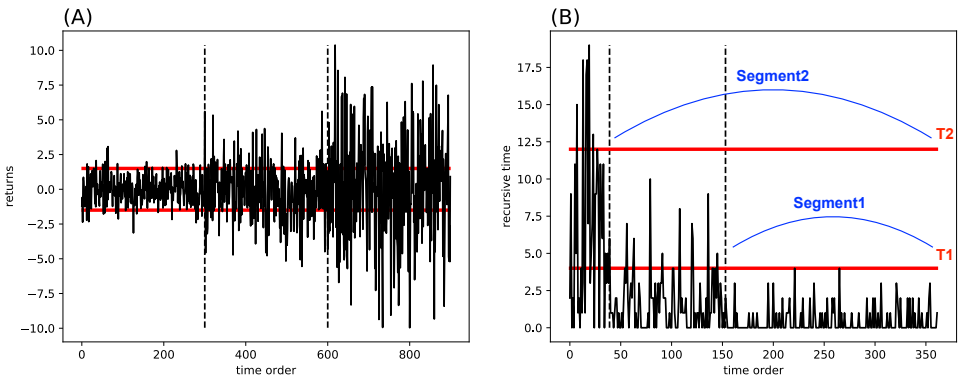

FIGURE 2.1. Simple example to illustrate the implementation of **Algorithm 1**

$$\hat{p}_j = \frac{1}{\sum_{t \in S_j} R_t} \tag{2.5}$$

for $j = 1, ..., k$. The searching algorithm actually advocates a way to partition the process into segments with k different intensities. With enough sample size, geometric distributions are appropriate to model the k phases with estimated parameter $\hat{p}_j$. Moreover, there are k parameters required, say $\theta = T_1, ..., T_{k-1}$, and T^*. To measure goodness-of-fit for a potential hidden state sequence, model selection is done by fitting geometric distribution within each hidden region, while penalizing the total number of switching regions as the model complexity. Information criteria AIC or BIC can be utilized for this purpose. Parameter p_j is estimated by MLE $\hat{p_j} = M_j/N$. So, the negative penalized likelihood or loss function can be written as,

$$Loss(\theta) = -2 \sum_{j=1}^{k} [\sum_{t \in S_j^\theta} E_t log \hat{p_j} + \sum_{t \in S_j^\theta} (1 - E_t) log(1 - \hat{p_j})] + \phi(N) Q_k \tag{2.6}$$

where E_t is a 0-1 discrete process after applying binning strategy (2.2); k is the number of hidden states; Q_k is the total number parameters from k conditional geometric distributions; $\phi(N)$ is a penalty coefficient. For instance, $\phi(N) = 2$ corresponds to Akaike Information Criterion(AIC), and $\phi(N) = log(N)$ corresponds to Bayesian Information criterion(BIC). In this paper, we consistently use BIC in all the experiment. The optimal parameters θ^* are tuned such that the loss can achieves its minimum, so

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, Loss(\theta) \tag{2.7}$$

Thus, the segments are ultimately achieved by applying θ^*. The computation cost is expensive if all possible $T_1, ..., T_{k-1}$ combinations are considered. In practice, a random grid-search strategy can be applied.

A toy dataset is simulated to illustrate how the algorithm works. Independent Normal data points are simulated with mean 0 but time-varying variance. $\sigma = 1$ when time $t \in [1, 200] \bigcup [400, 600]$, denoted as "state1", and $\sigma = 1.5$ for the rest of time, denoted as "state0". The purpose here is to discover the underlying switching pattern of σ. The simulated time series is shown in Figure 2.2. A pair of thresholds α and β is chosen as cutting lines to mark extremely large values (red dots). After that, the limiting distribution of waiting time between successive extreme events is analyzed, and segmentation is done via model selection with AIC. P-P plots in Figure 2.3 show a goodness-of-fit for the waiting time variables in both regions. And the segmentation result (yellow line) can almost perfectly capture the true dynamic pattern of σ.

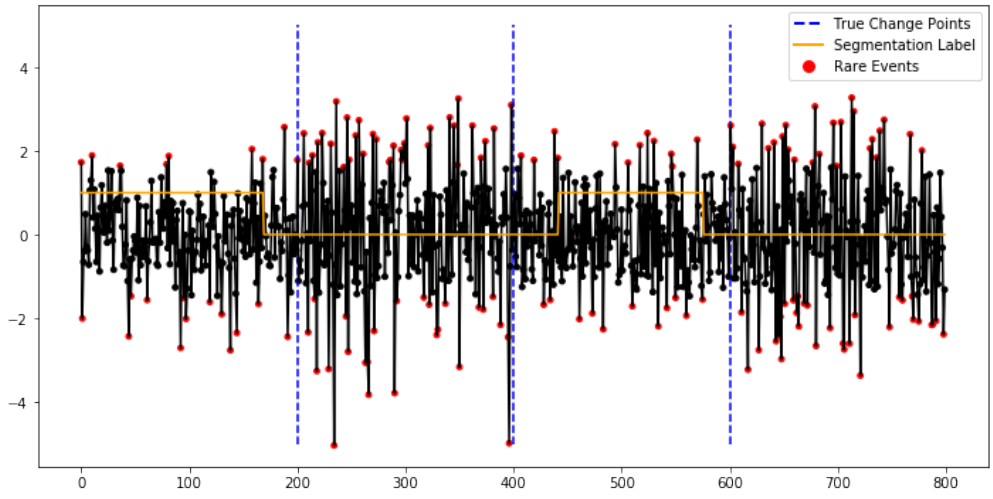

FIGURE 2.2. Independently Normal distributed process with $\mu = 0$ and $\sigma = 1$ or 1.5 varying over time. The vertical dashed line indicates the real change points; the yellow solid line indicates estimated segmentation label; red dots indicate the events of interest

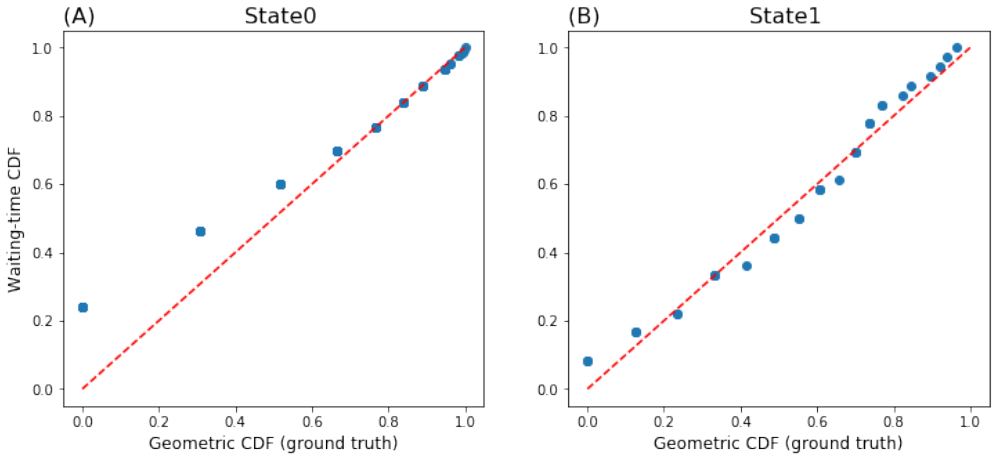

FIGURE 2.3. P-P plot for the geometric distribution with true parameter versus empirical waiting time between successive events; (A) within "state0"; (B) within "state1"

2.3.3. Simulation. Numerical experiments are done to demonstrate the model performance. In the first case, data is generated according to Hidden Markov Model(HMM) with 2 hidden states. The decoding error rate is calculated and compared for the proposed method and the Viterbi's path [**117**].

In the application of Viterbi's, since the true transition probability and emission probability are unknown in reality, EM algorithm in Baum–Welch type [**13**] is firstly implemented to estimate the parameters. Generally, an initial condition is needed in implementing the EM algorithm. Here, we assume that the true emission probability is known, but initialed with two kinds of transition probability- one is the true transition probability denoted as A_{true} and the other is a random initial denoted as A_0. For convenience, we set $p_{11} = p_{22}$ in transition probability matrix A_{true}, so A_{true} is controlled by only one parameter p_{12},

$$A_{true} = \begin{bmatrix} 1 - p_{12} & p_{12} \\ p_{12} & 1 - p_{12} \end{bmatrix}$$

Random initial transition A_0 are generated with all entry value 0.5,

$$A_0 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

Observations are simulated via different transition matrix A_{true} and emission (p_1, p_2). The experiments are repeated 100 times, and the mean and standard deviation of decoding error rates are reported in Table1. It shows that the overall decoding errors are decreasing as p_{12} is lower, and our method gets largely improved when p_{12} decreases to 0.005. It can be explained by the fact that model selection criteria, like BIC, favors a "simple" model with fewer alternating segments but with a longer length of each segment. In this settings, Viterbi's in conjunction with the true initial matrix performs better when p_{12} is moderate. However, the result becomes much worse if a random initial is applied. It challenges the application of Viterbi's or Baum–Welch's to select an appropriate initial. Instead, without assuming any prior knowledge, the proposed method is more robust and competitive with Viterbi's even under Markovian conditions.

TABLE 2.1. Decoding Error Rate

		p_1=0.2, p_2=0.1			p_1=0.1, p_2=0.05		
		Viterbi(A_0)	Viterbi(A_{true})	HFS	Viterbi(A_0)	Viterbi(A_{true})	HFS
p_{12}=0.005	N=2000	0.4358 (**0.050**)	0.2378 (0.114)	**0.2290** (0.093)	0.4550 (**0.062**)	0.3726 (0.110)	**0.2689** (0.090)
	N=5000	0.3940 (0.054)	**0.1850 (0.051)**	0.2809 (0.164)	0.4256 (**0.076**)	**0.3169** (0.077)	0.3171 (0.097)
p_{12}=0.01	N=2000	0.4405 (**0.045**)	**0.2838** (0.069)	0.3376 (0.109)	0.4722 (**0.044**)	**0.3569** (0.082)	0.3710 (0.089)
	N=5000	0.4504 (**0.042**)	**0.2890** (0.058)	0.3584 (0.048)	0.4889 (**0.053**)	0.4219 (0.076)	**0.4064** (0.078)
p_{12}=0.05	N=2000	**0.4407** (**0.031**)	0.4526 (0.045)	0.4518 (0.034)	0.4718 (**0.036**)	0.4825 (0.037)	**0.4652** (0.051)
	N=5000	0.4398 (**0.020**)	**0.4187** (0.051)	0.4752 (0.029)	0.4751 (0.021)	0.4832 (0.033)	**0.4738** (**0.014**)
p_{12}=0.10	N=2000	**0.4415** (0.019)	0.4543 (0.034)	0.4728 (**0.017**)	**0.4729** (0.025)	0.4778 (0.032)	0.4744 (**0.017**)
	N=5000	**0.4464** (**0.014**)	0.4650 (0.026)	0.4920 (0.025)	**0.4757** (**0.021**)	0.4792 (0.031)	0.4853 (0.022)

In the second case, data is simulated under a regime-switching model with 3 hidden states embedded behind. Suppose that the observations (log returns) are $\{X_t\}_{t=1}^{8000}$ and there are 8 alternating segments over time:

$$S_t = \begin{cases} 1 & t \in [1, 1000], [4001, 5000], [7001, 8000] \\ 2 & t \in [1001, 2000], [3001, 4000], [6001, 7000] \\ 3 & t \in [2001, 3000], [5001, 6000] \end{cases}$$

The index of the hidden states is alternating like $1, 2, 3, 2, 1, 3, 2, 1$. In the first example, observations are generated of Gaussian distribution with mean 0 and variance varying under different states in

Figure 2.4(B), so

$$X_t \sim \begin{cases} N(0, \sigma_1^2) & S_t = 1 \\ N(0, \sigma_2^2) & S_t = 2 \\ N(0, \sigma_3^2) & S_t = 3 \end{cases}$$

where standard deviations $\sigma_1 = 1$,$\sigma_2 = 2$, and $\sigma_3 = 3$. In the second example, heavy-tail distribution, student-t, is considered. The simulation is shown in Figure 2.4(D) by

$$X_t \sim \begin{cases} t(df_1) & S_t = 1 \\ t(df_2) & S_t = 2 \\ t(df_3) & S_t = 3 \end{cases}$$

where degree of freedoms $df_1 = 1$, $df_2 = 2$, and $df_3 = 5$. In the Gaussian setting, we set $|\alpha\text{-}quantile| = |\beta\text{-}quantile| = 2$ which corresponds to 0.9 quantile of the observations; In the student-t setting, a larger threshold is considered, $|\alpha\text{-}quantile| = |\beta\text{-}quantile| = 3$, which corresponds to 0.95 quantile of the observations.

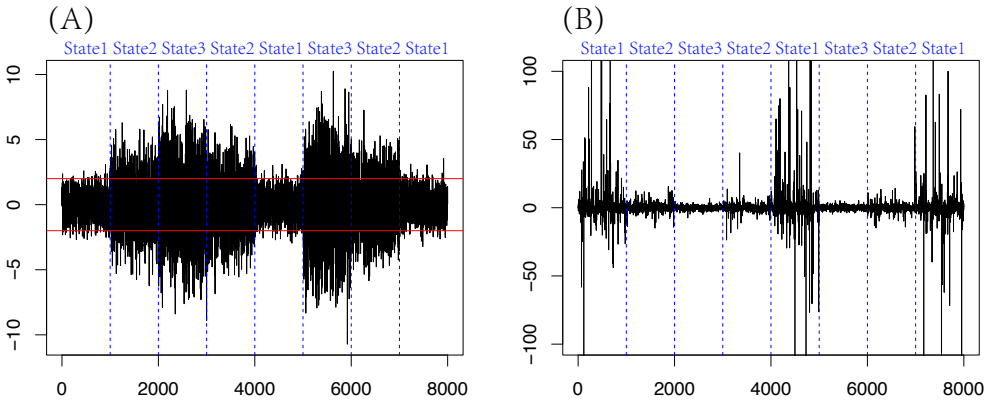

FIGURE 2.4. Data is simulated via conditional distribution given a hidden state: (A) Gaussian distribution (B) student-t distribution. 8 underline phases alternates over time where 3 kinds of hidden states are embedded. The horizontal lines in (A) indicate the thresholds α-*quantile* and β-*quantile*.

The recovered segment (marked in different colors) shows that with appropriate choice of thresholds, the proposed method can successfully detect the alternating hidden states. The error only appears around the change points. Besides, we obtain good estimations of emission probability

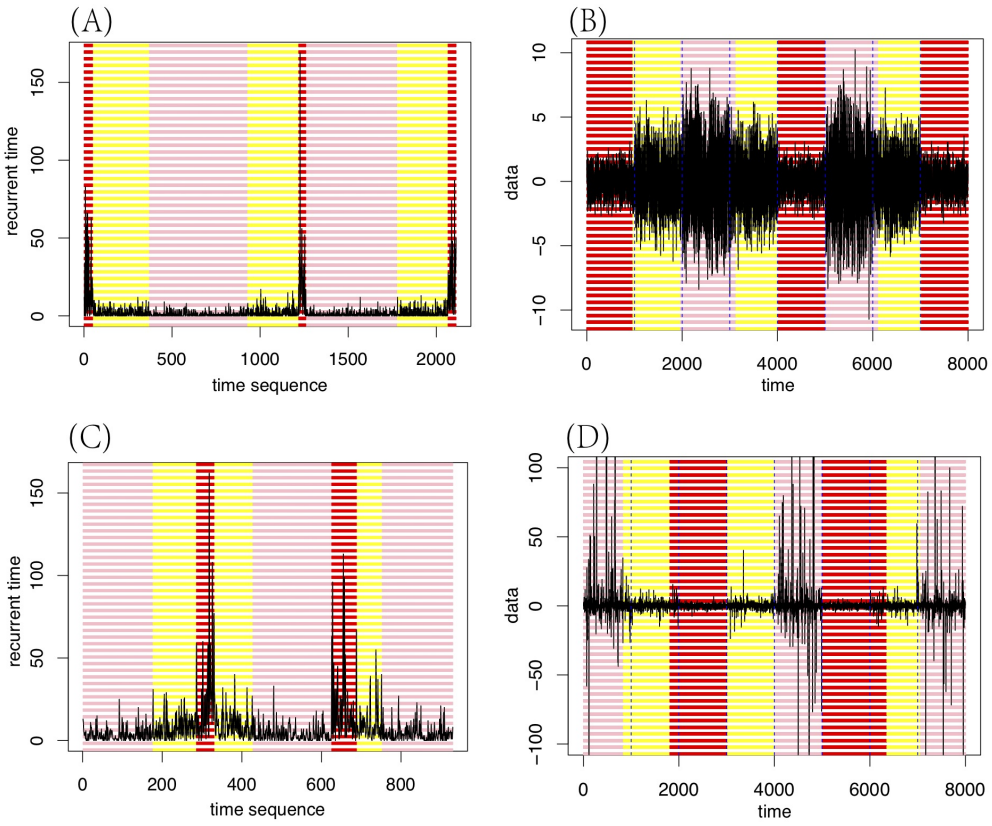

FIGURE 2.5. Simulation with Normal distribution(A)(B), student-t distribution (C)(D). (A),(C): Recursive time; (B),(D): raw data with colored decoding states. "red", "yellow", and "pink" 3 colors indicates 3 different kinds of states.

under different hidden states. The estimators are $(0.0463, 0.3210, 0.4739)$ in the Gaussian setting and $(0.0420, 0.1008, 0.1997)$ in the student-t setting, respectively. They are close to the theoretical parameter

$$2 * (\Psi_1(-2), \Psi_2(-2), \Psi_3(-2)) = (0.0455, 0.3173, 0.5049)$$

and

$$2 * (F_{t1}(-3), F_{t2}(-3), F_{t3}(-3)) = (0.0300, 0.0954, 0.2048)$$

where Ψ_j is the CDF of Gaussian distribution under the j-th hidden state, and F_{tj} is the CDF of student-t distribution under the j-th hidden state, for $j = 1, 2, 3$.

2.4. Encoding-and-Decoding Procedure

2.4.1. The method. As described in the previous section, the choice of threshold in defining an event or large returns is of vital importance. A natural question is that what if the threshold fluctuates, is the decoding result of the 0-1 processes still stable? The answer is positive. For example, if an observation is marked with a return below a threshold π, the intensity parameter in the geometric distribution is $p_s(\pi) = F_s(\pi)$ under a hidden state s. By assuming the continuity of the underlying distribution F_s, p_s is also continuous for π. Thus, the emission probability under a hidden state would not fluctuate much if π varies slightly. Indeed, our experiment shows that the estimated emission probability is not sensitive to π. To make the notation consistent, we will use $p^\pi(t)$ or $F^\pi(t)$ if both t and π are present.

The idea of dealing with a stochastic process with continuous observations is described as follows. In the encoding phase, we iteratively switch the excursion threshold from an extreme value lower to 0 and discretize the time series into a 0-1 binary process at each iteration. After that, we implement the searching algorithm to decode the processes, respectively. As a consequent result, a vector of estimated emission probability $\hat{p}^\pi(t)$ is obtained at time t with a different choice of π. It actually gives an estimator of the empirical Cumulative Distribution Function(eCDF) at time t, $\hat{F}^\pi(t) = \hat{p}^\pi(t)$ where $\hat{F}^\pi(t)$ is a function of π given a fixed t. Following up the simulated data with t distribution, Figure 2.6 shows a series of eCDF with a change point embedded in the middle though it is hard to detect by eyes. Lastly, all the decoding information is aggregated by clustering the time points with comparable eCDFs. Suppose that $\{E_t^\pi\}_t$ is a 0-1 coding sequence obtained by applying a threshold π upon the returns, and Π is a pre-determined threshold set, for example, Π can be a series of quantiles of the mixed distributions $\Pi = \{0.99\,\textit{quantile}, 0.97\,\textit{quantile}, 0.95\,\textit{quantile}, ...\}$. The encoding-and-decoding algorithm is described in **Algorithm 2**.

Algorithm 2 Encoding-and-Decoding

---

1. For loop: cycle threshold π through $\Pi = \{\pi_1, \pi_2, ...\}$:

Define events and code the whole process as a 0-1 digital string $\{E_t^\pi\}_{t=1}^N$,

$$E_t^\pi = \begin{cases} 1 & X_t \leq -\pi \ or \ X_t \geq \pi \\ 0 & otherwise \end{cases} \quad (symmetric)$$

Or

$$E_t^\pi = \begin{cases} 1 & X_t \leq \pi \ \ if \ \pi < 0 \\ 1 & X_t \geq \pi \ \ if \ \pi > 0 \\ 0 & otherwise \end{cases} \quad (asymmetric)$$

Repeat the step 3 & 4 in **Algorithm 1** and estimate the probability $\hat{p}^\pi(t)$ by (2.5).

End For

2. Stack the estimated probability at t in a vector $\vec{p}(t) := (\hat{p}^{\pi_1}(t), \hat{p}^{\pi_2}(t), ...)$.

3. Merge time points with comparable $\vec{p}(t)$ together via clustering analysis.

It is surprising that how an eCDF can get returned based on the only observation at each time stamp. Indeed, the eCDF does not hold an asymptotic property here. Because of the limitation of data information, increasing the number of thresholds π would not provide a good estimation of the distribution. The estimated distribution function here depends on how well the decoding algorithm can separate the different states and at what intensity levels. Specifically, if there exists a threshold π by which the underlying distribution can be separated well, then the decoding achieves a good result to reflect the distributional changes. On the other hand, if the π is set not appropriate, for example, the permission probability of the underlying distribution at state s and $s^{'}$ are very close to each other or $\hat{F}_s^\pi \cong \hat{F}_{s^{'}}^\pi$, then the decoding algorithm fails to separate the two states with such a threshold applied. There is an ongoing discussion about how to choose a good threshold to discretize a sequence of continuous observations. A heuristic idea is to tune the optional value of π such that the estimated probabilities under different states are far apart from each other. For example, consider the max-min estimator of π,

$$\hat{\pi} = \underset{\pi}{\operatorname{argmax}} \min_{s,s^{'}} |\hat{p}_s(\pi) - \hat{p}_{s^{'}}(\pi)| \tag{2.8}$$

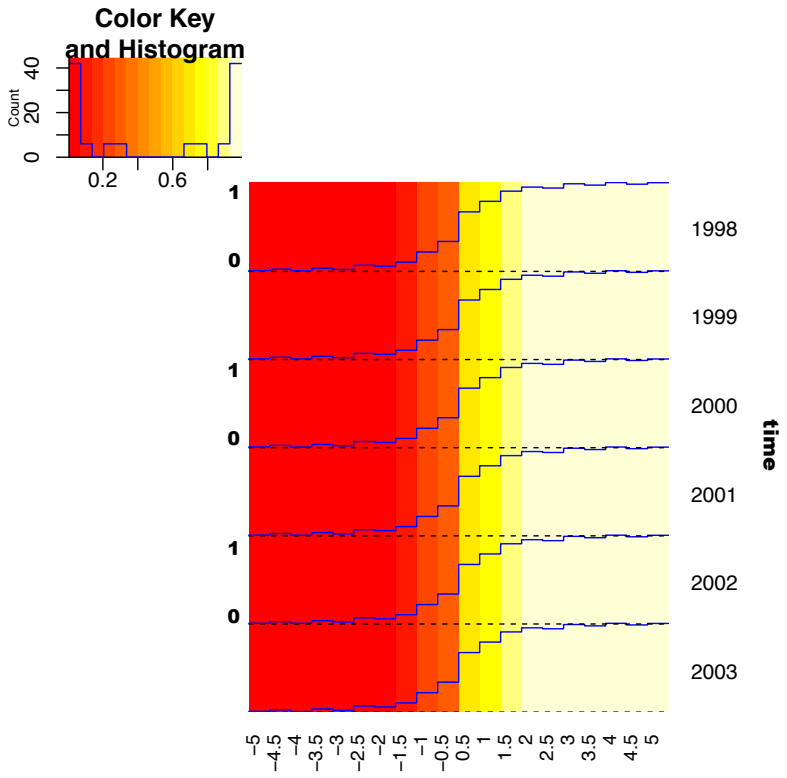

FIGURE 2.6. For t-distributed simulation, eCDF for time point from 1998 to 2003. Data from 1998 to 2000 follows t distribution with degree of freedom 2; data from 2001 to 2003 follows t distribution with degree of freedom 5

It is remarked that the proposed procedure avoids the issue of tuning parameters by imposing a series of thresholds and aggregating all the decoding results together. We claim that the information of distributional changes is reserved onto the vector of the emission probabilities. Moreover, a noisy result by applying an unreasonable threshold would not affect the aggregation result much. For example, if $\hat{F}_s^{\pi} \cong \hat{F}_{s'}^{\pi}$, then there is no distributional changes are detected in the process, so $\hat{p}^{\pi}(t)$ is a constant for any t. Based on our numerical experiments, if π is set close to the median, the emission probabilities are not separated enough, so there is a big "jump" in the middle of the eCDFs. In summary, the continuous decoding algorithm can be applied by shifting π value from high to low to obtain a sequence of eCDFs, although only fewer π are meaningful in sense of decoding. By combining the decoding results of several discrete processes, the aggregated information sheds a light in differentiating the underlying distributions.

2.4.2. The number of hidden states. The next question is how to summarize the information among all the eCDFs, and how many underlying distributions can represent the patterns of distribution switching. It raises a realistic question to all the regime-switching models to determine the number of underlying states. Generally, the more states are taken into consideration, the less tractable the model becomes. For example, a 2-state lognormal Markov Model contains 6 parameters, while a 3-state model increases the number to 10. The number of states is usually decided subjectively or tuned with an extra criterion. Given the estimated probability vector $\vec{p}(t)$ in **Algorithm 2**, the problem is resolved by clustering the similar time points together such that the eCDFs within each cluster are comparable to each other. Moreover, the number of underlying states can be determined by searching through the number of clusters embedded in the probability vectors or eCDFs. Hierarchical clustering is implemented to cluster similar time points shown in Figure 2.7(A). One can visualize the dendrogram to discover the number of clusters, or employ criteria to quantify the quality of clustering.

The numerical data with student-t distributions is reused in this example. The dendrogram shows that 2 or 3 clusters may be embedded inside the observations. If we cut the dendrogram into 3 clusters, the trajectory of cluster indices can almost perfectly represent the alternating hidden states, see Figure 2.7(B). If 2 clusters are taken rather than 3, the result also makes sense since cluster2 and cluster3 are combined together as a contradiction to the high-intensity cluster or cluster1. By calculating the average function of eCDFs for each cluster, one can compare the estimated probability function with the theoretical distributions, respectively. Figure 2.7 (C)-(E) show that the average function is a goodness-of-fit for each cluster.

It is also claimed that the model performance is not sensitive to the number of hidden states that we supposed in decoding each 0-1 process. Without the prior knowledge of 3 states embedded in the observations, if we implement a 2-states or 4-states decoding schedule, a 3-cluster clustering result can still represent the distributional changes well. The clustering result of 2-state decoding is shown in Figure 2.8 though there is a short period (between 1000 and 2000) in which state2 is misclassified as state3. The result of 4-state decoding is more accurate, see Figure A.1 in Appendix A.

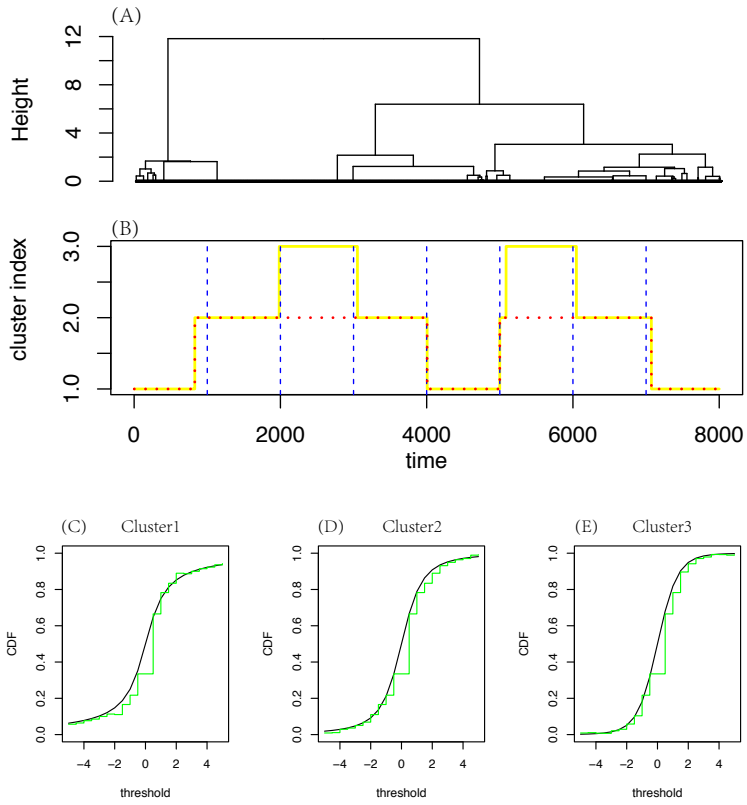

FIGURE 2.7. 3-states continuous-distribution decoding in simulation data. (A) Hierarchical Clustering Tree; (B) cluster index switching over time; (C),(D),(E): median eCDFs versus true CDFs, in cluster 1,2,3, respectively

2.5. Real Data Application

In the real data application, we analyze the tick-by-tick data of S&P500 index. The stock returns are calculated in a market time scale which is measured by transaction rather than the real-time clock. The idea was firstly suggested by **[80]**, and then worked thoroughly by **[32]**. A well-known example is the random-walk model suggesting that the variance of returns depend on the number of transactions. Following the idea above, we apply the tiniest sampling rate to alleviate the serial dependency. It is reasonable to assume that the stock returns are exchangeable within a certain number of transactions.

2.5.1. Single index dynamics. The encoding-and-decoding algorithm is implemented to discover the volatility dynamics for a single stock. Since the result is not sensitive to the number

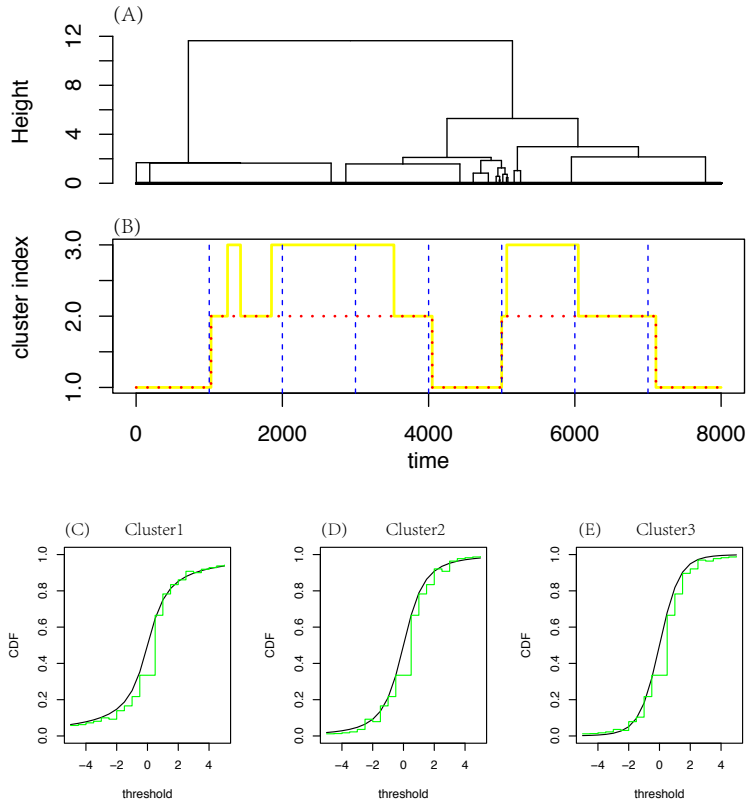

FIGURE 2.8. 2-states continuous-distribution decoding in simulation data. (A) Hierarchical Clustering Tree; (B) cluster index switching over time; (C),(D),(E): median eCDFs versus true CDFs, in cluster 1,2,3, respectively

of hidden states, a 2-state decoding procedure is applied and the number of clusters is determined according to the tree height of hierarchical clustering. It turns out that there are 3 potential clusters embedded in the returns of IBM index in 2006. The average function of eCDFs for the 3 clusters are shown in Figure 2.9. The distribution of cluster3 with a heavier tail reflects a high-volatility phase; cluster1 indicates a phase with low volatility. As a phase in the middle, cluster2 shows an asymmetric distribution with a heavy tail on the left but a light tail on the right. Instead, cluster1 and cluster3 look more symmetric on both sides. The result shows that an asymmetric distribution could be embedded in reality, which is usually missed by researchers.

We then present volatility dynamics by showing the cluster index varying over the transaction time. The dynamic pattern of IBM in January 2006 are shown in Figure 2.10. According to the

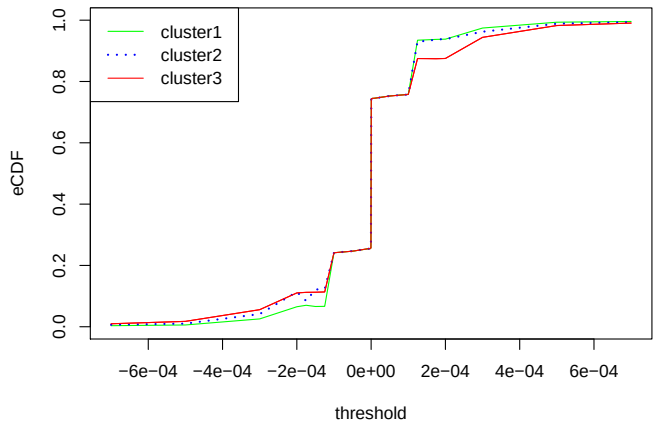

FIGURE 2.9. The average functions of eCDFs from the 3 clusters of IBM in January 2006

previous notation, cluster1 indicates a low-volatility phase, cluster2 is for a median stage, and cluster3 presents a high-volatility phase. Based on the daily segments, it is clear that the unstable volatility mostly appears at the beginning of a stock market, and usually shows up twice or three times per day.

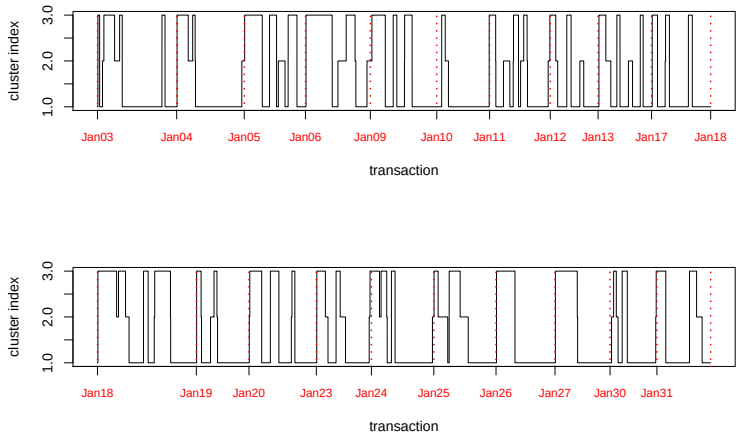

FIGURE 2.10. Recovered volatility trajectory of IBM in January 2006

2.5.2. Nonlinear Dependency. Beyond detecting the volatility dynamics for a single stock, we further consider the network among all the S&P500 to present how one stock's returns is related

to another stock's return. Such a relationship can be quantified by calculating the cross-correlation between two stock time series [**29**, **81**] and correlation matrix was investigated via random matrix theory [**103**] or clustering analysis [**63**]. Conditional correlation [**40**] and partial correlation [**68**] was also studied to provide information about how the relationship of two stocks is eventually influenced by other stocks. However, since the empirical distribution of returns is very different from Gaussian, and correlation is only adaptive for a linear relationship, a distribution-free and nonlinear measurement is studied to measure the financial connection. Transfer Entropy(TE) [**57**, **107**], as an extension of the concept of Granger causality [**12**], was proposed to measure the reduction of Shannon's entropy in forecasting a target variable via the past value of a source variable. Denote the target variable at time t as Y_t and the source variable at t as X_t. The Transfer Entropy from X to Y in terms of past l lags is defined by,

$$TE_{X->Y}(l) = \sum_{t=l+1}^{n} P(Y_t, Y_{(t-l):(t-1)}, X_{(t-l):(t-1)}) log \frac{P(Y_t|Y_{(t-l):(t-1)}, X_{(t-l):(t-1)})}{P(Y_t|Y_{(t-l):(t-1)})}$$

It is remarked that the measure is asymmetric, generally, $TE_{X->Y} \neq TE_{Y->X}$.

However, it is computationally infeasible to calculate the exact TE value due to the difficulty in estimating a conditional distribution or joint distribution especially when l is large. In the application of finance, people commonly cut the observation range into disjoint bins and assign a binning symbol to each data point [**35**, **102**, **104**]. However, a straightforward implementation of binning with equal probability for every symbol will lead to sensible results [**82**]. To the best of our knowledge, it still lacks in the literature to digitize the stock returns and effectively reveal the dynamic volatility. The simple binning methods such as histogram or clustering fail to catch the excursion of large returns, so only the trend of the returns is studied but the dynamic pattern or volatility is missing in it. We claim that the encoding-and-decoding procedure remedies the drawbacks of simple binnings. Indeed, the recovered volatility states can be easily applied to calculate the TE values. Moreover, the network is improved in terms of measuring the causality of stock volatility rather than the similarity of trend.

The proposed procedure is implemented to detect the volatility trajectory of the tick-by-tick returns. However, since the return patterns are recorded in transactions, the decoded sequences are not directly conjunct with each other. It is required to transform the decoded trajectories back

into the real-time scale before calculating the Transfer Entropy. Suppose that the intensity level of volatility is indicated by ordinal number 1, 2, or 3 meaning low-, median-, or high-volatility state, respectively. If there exists a tiny time scale in which at most one transaction happens, a time unit can be labeled by symbol 0 if no transaction present or an ordinal number if a transaction present. Thus, the decoded pattern from different stocks can share the same chronological time. It seems that we attempt to choose a time scale as small as possible, but the pairwise dependency is weakened due to the increasing number of 0's. To balance the proportion of symbols and alleviate the sparsity, we summarize the recovered pattern by scanning a time block from the beginning to the end of the time axis to select the maximal ordinal number. So, uninformative 0's are filtered out, while volatility stages are kept. Suppose that the recovered symbol sequence is $\{S_t\}_{t=1}^N$ where $S_t \in \{0, 1, 2, 3\}$. The sequence is then summarized via a time block with length w by

$$S_t^* = \max\{S_t : t \in (t - \lfloor \frac{w}{2} \rfloor, t + \lfloor \frac{w}{2} \rfloor)\} \tag{2.9}$$

for $t = \lfloor \frac{w}{2} \rfloor, ..., n - \lfloor \frac{w}{2} \rfloor$ where $\{S_t^*\}_t$ is the summarized symbolic sequence. It is noted that a minute-level scale w is too rough to reflect the tick-by-tick volatility pattern. A block with $w = 5$-seconds is an admissible choice.

According to the way we summarize the symbolic trajectory, a nonlinear measure is developed as a variant of TE. Different from the classic TE, this measure takes both lag and lead effects into account instead of only the lag effect. Denote the summarized symbolic sequence of X and Y as S_t^X and S_t^Y, respectively. The lag-and-lead information flow from X to Y is defined by

$$TE^*_{X->Y}(w) = \sum_t P(S_t^Y = 3, S_t^X) log \frac{P(S_t^Y = 3 | S_t^X)}{P(S_t^Y = 3)} \tag{2.10}$$

We use $TE^*_{X->Y}$ to differentiate it from the classic TE and w is omitted without confusion. The measure is interpreted by how much uncertainty Y is affected due to the lag-and-lead effect of X such that Y is under its volatility states (state3). The higher the value, the stronger the impact that X promotes volatility phases of Y.

The summarized symbolic sequence $S_X(t)^*$ and $S_Y(t)^*$ are shown in Figure 2.11. In the first example, Figure 2.11(A) shows that MXIM (Maxim Integrated Products Inc.) and NTAP (NetApp Inc.) share a large intersection in volatility phases. Especially, when MXIM is in volatility, the price

of NTAP has a high probability to be in state3. The value of the dependency TE^* from MXIM to NTAP is 0.039 and 0.016 in reverse. In the second example, Figure 2.11(B) shows that TWX has a stronger influence on the volatility stages of BRCM. The value of TE^* from TWX to BRCM is 0.036 and 0.026 in reverse.

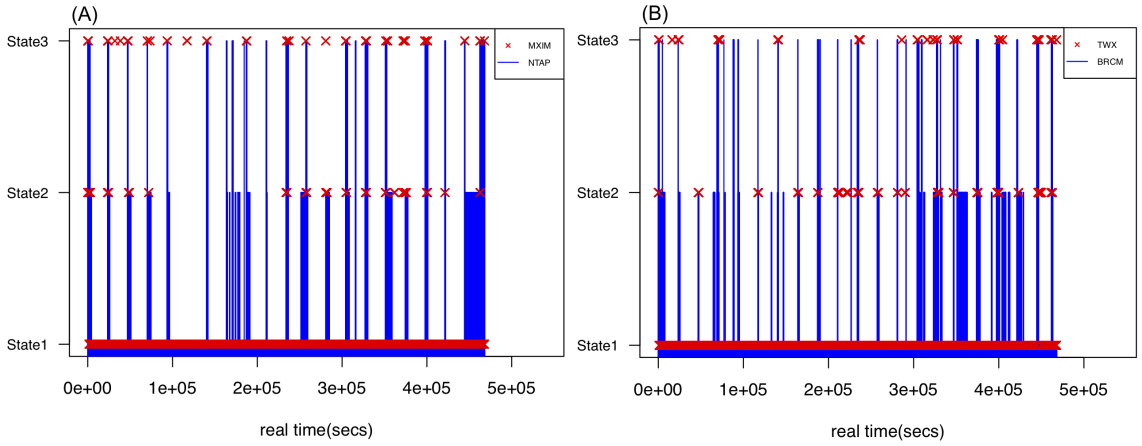

FIGURE 2.11. A pair of volatility trajectories summarized in real time: (A) MXIM v.s NTAP; (B) TWX v.s BRCM

Once the Transfer Entropy is calculated for all pairs of indices of S&P500, the result can be recorded via a 500×500 asymmetric matrix with the entry value of the i-th row and the j-column as the information flow from the i-th stock to the j-th stock. We rearrange the rows and columns such that the sum of rows and the sums of columns are in ascending order, respectively. The reordered TE matrix is shown in Figure 2.12. The idea of reordering follows the discussion about the node centrality for directed networks in [**104**]. Two types of node strength are considered for incoming and outgoing edges. The incoming node strength at node i denoted as NS^i_{in}, is defined by the sum of the weights of all the incoming edges to i,

$$NS^i_{in} = \sum_j TE^*_{j->i} \tag{2.11}$$

Similarly, the outgoing node strength, denoted as NS^i_{out}, is defined by the sum of the weights of all the outgoing edges from i,

$$NS^i_{out} = \sum_j TE^*_{i->j} \tag{2.12}$$

If a stock has a larger value of incoming node strength, it receives more information flow, which means the stock is strongly influenced by other indices; while, if a stock has a larger value of outgoing node strength, it sends more impacts to other stocks. The top30 stocks with the largest incoming and outgoing node strength values are reported in TableA.1 in Appendix A. If we take the intersection between the top30 incoming nodes and the top30 outgoing nodes, a group of most central stocks gets returned. The central stocks can be regarded as the intermediate nodes which connect all the other stocks in the S&P500 network. The central stocks include CHK(Chesapeake Energy), VLO(Valero Energy), NTAP(NetApp, Inc.), BRCM(Broadcom, Inc.), and TWX(Time Warner, Inc.), which are all big manufacturer, retailer, supplier, or media covering the important fields in the United States.

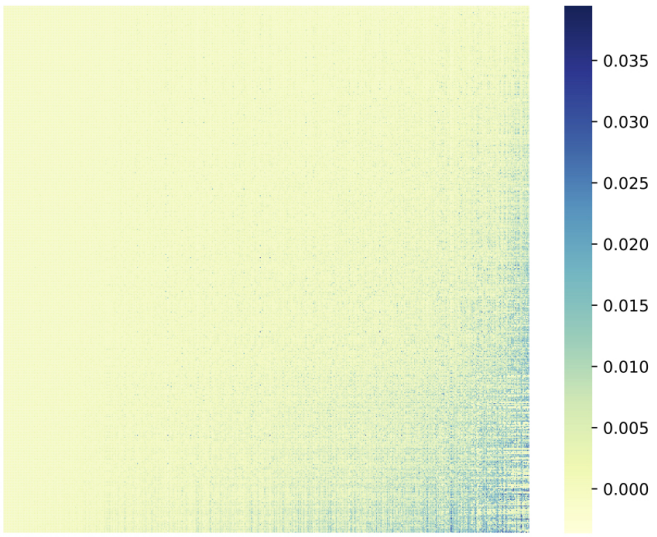

FIGURE 2.12. Transfer Entropy matrix for S&P500 in 2006. The rows and columns are rearanged such that the row sum and column sum are in ascending order

2.5.3. S&P500 Networks. In this subsection, we present two different types of networks to illustrate the volatility connection among the S&P500 in 2006. A weighted directed network is

established by regarding each stock as a node, the information flow from one node to another as an edge, and the Transfer Entropy value as the weight of the edge. Nodes with weak dependency are filtered out, so only the strongest edges and their conjunct nodes are shown in Figure 2.13. Apart from the central stocks such as CHK, VLO, NTAP, and BRCM, the result shows that big investment corporations, such as JPM(JPMorgan), BAC(Bank of America), and C(Citigroup) also heavily depend on other indices. Instead, TWX(Time Warner, Inc.), MXIM(Maxim Integrated Products Inc.), APC(Apple inc.), EBAY(eBay Inc.), and YHOO(Yahoo! Inc.) have a primary impact on others S&P500.

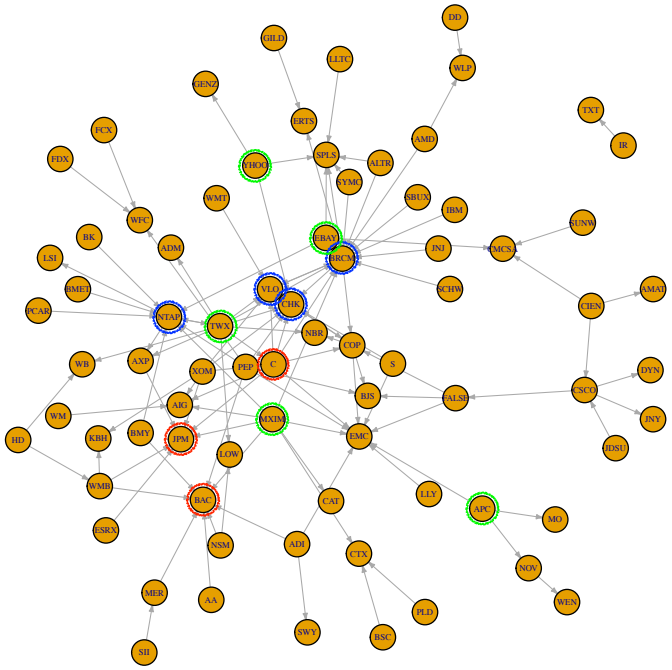

FIGURE 2.13. A directed network of S&P500: edges with the strongest weights and the conjunct nodes are shown; blue nodes: central stocks; red nodes: stocks with strong incoming strength; green nodes: stocks with strong outgoing strength.

Another way to represent the network is to transform the asymmetric information flow into a symmetric dissimilarity measure. The similarity between the i-th and the j-th nodes can be defined by the average of the two asymmetric TE values,

$$Sim(i,j) = (TE^*_{i->j} + TE^*_{i->j})/2 \tag{2.13}$$

If the range of similarity is rescaled between 0 and 1, the dissimilarity can be simply defined by,

$$Dis(i,j) = 1 - \frac{Sim(i,j) - \min_{i,j} Sim(i,j)}{\max_{i,j} Sim(i,j) - \min_{i,j} Sim(i,j)} \tag{2.14}$$

So, the range of dissimilarity is scaled between 0 and 1. The symmetric dissimilarity matrix of S&P500 is present in Figure 2.14(A) with a hierarchical clustering tree imposed on the row and column sides. The idea is similar to Multidimensional Scaling, which has been widely used to visualize the financial connectivity in a low-dimensional space **[55]**. We claim that the dendrogram provided by hierarchical clustering is more informative in terms of illustrating how the S&P500 are agglomerated hierarchically from the bottom to the top according to their dissimilarity. Intuitively, companies under a similar industrial category should be merged into a small cluster branch. One of the branches with relatively low mutual distance is extracted and shown in Figure 2.14(B). It looks that the cluster mainly includes technology companies including internet retail (EBAY and AMZN), manufacturer of integrated circuits(LLTC), video games(ERTS), information technology(ALTR), network technology(TLAB), biotechnology(GILD and GENZ), etc. Besides, we notice that energy corporations, such as VLO, COP, and CHK, are also merged into a small cluster.

2.6. Conclusion

Starting from a definition of large or relative extreme returns, we firstly propose a searching algorithm to segment stock returns into multiple levels of volatility phases. This is an extension of Hierarchical Factor Segmentation(HFS). Then, we advocate a data-driven method, named encoding-and-decoding, to discover the embedded number of hidden states and represent the stock dynamics. By encoding the continuous observations into a sequence of 0-1 variables, a maximum likelihood approach is applied to fit the limiting distribution of the recurrence time series. Though the assumption of exchangeability within each hidden state is required, our numerical experiments show that our proposed approach still works when the assumption is slightly violated, for example, a weak transaction probability is imposed under the Markovian condition. This demonstration of robustness with respect to various conditions makes our approach valuable in real-world finance researches and practices.

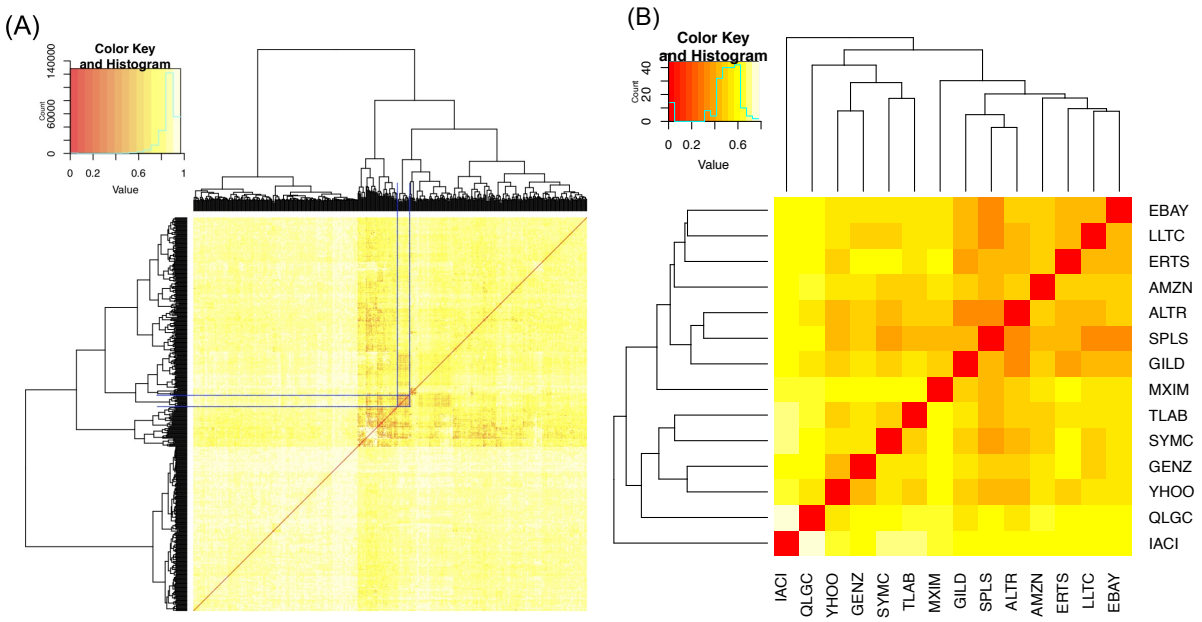

FIGURE 2.14. Heatmap of the symmetric dissimilarity matrix with a hierarchical clustering tree imposed on the row and column sides; Ward linkage is applied in the hierarchical clustering algorithm; (A) a matrix for S&P500; (B) a submatrix extracted from (A)

In real data application, it was reported by [**24**] that stock returns are only exchangeable in a short period. With this assumption holds, our proposed method is implemented on tick data to alleviate the serial dependency. Moreover, it is beneficial to investigate the fine-scale volatility, so the established network can illustrate which stocks stimulate or even promote volatility on others. It is also noted that the nonparametric regime-switching framework can work in conjunction with other financial models. For example, Peak Over Threshold(PoT) [**75**] can be implemented to analyze the extreme value distribution for the discovered homogeneous regimes. We hope these networks would be somehow stimulating for researchers and practitioners in finance.

CHAPTER 3

Multiple Phases of Dynamics in Multivariate Financial Time Series

3.1. Introduction

It has received increasing research interests and attention in studying nonlinear stochastic dynamics in quantitative finance. Researchers and practitioners have realized that it is rather important to understand the joint behaviors of multiple aspects of one single stock or asset as well as one common aspect of multiple assets. One dynamic issue that has been making quantitative finance experts wondering even up to now is the joint dependence among returns, trading volume, and transaction numbers [**78**,**128**]. Another well-known dynamic issue is about how volatility clustering comes to exist and where to look for it [**37**]. Since volatility is measured by conditional variance and it changes over time for one single stock or asset, how to compute and visualize volatility in concert with a form of clustering to a great extend is still mysterious. Computational and data-driven approaches for both issues are not yet well established or reported in the literature.

For instance, GARCH models have been proposed to study and to model the time-varying volatility of asset returns [**20**], and their variants have been extended to multivariate time series cases by modeling the correlation dynamics [**21**, **40**, **115**]. However, they require too many parameters and a large collection of prior knowledge about the dynamic structure. Such modeling and required structures make the model interpretation rather complicated. A more effective methodology was proposed to incorporate realized volatility [**49**, **50**] and realized covariance [**108**, **126**]. However, its results could be biased due to noises' hard to be accommodated nature. Further, often a long time window is usually required implement such a methodology.

Recently, a data-driven approach named Hierarchical Factor Segmentation(HFS) is developed by characterizing volatility fluctuation directly [**59**]. HFS, to some extent, is similar to the regime-switching model advocated by [**48**, **52**]. Both likewise assume regime-switches being somehow away from the beginning and ending time points of the involved time span. HFS computationally attempts

to detect all time-points, at which the dynamics phase shifts from one episode to another by revealing distinct dynamic behavior. Its chief computing device tracks the recurrence of "extreme" events, i.e. large returns, defined by a chosen threshold. Consequently, latent regions with different event-intensities are segmented. Via this way, dynamic tail behaviors are successfully discerned. Along the direction of threshold choice, HFS was extended to study the empirical tail distribution by applying a series of thresholds in [**120**].

Compared with region-switching models, HFS takes advantage of offline analysis to decode dynamics patterns without assuming any underlying distribution or Markovian structure. So HFS is in the category of nonparametric change point detection in time series. Nonparametric change point detection has a wider range of applications than parametric [**53**]. Characteristically, it often relies heavily on the estimation of density functions [**67**], see details in a recent survey being available in [**6**]. The key difference between HFS and the change point approach, in general, is that we assume the underlying distributional changes at a certain point and may come back in afterward in a recurrent fashion, which makes more sense in the case that volatility clustering comes and leaves recurrently in financial data.

So far, the nonparametric approach in discovering the recurrent switch patterns underlying multivariate time series is still scarce. One underlying reason is attributed to the fact that the nonlinear dependence among them is the key and necessary knowledge. Thus, missing or lacking such knowledge underlying all involving time series or processes has become a barrier that hinders the potential research advances in this direction. Such dependence needs to be measured based on the latent phases revealed from each single process separately [**60**, **120**]. However, beyond the multiplicity and complexity of global dynamic patterns, the nonlinear dependence can be easily overwhelmed by the integrated microstructure noises. In this chapter, we extend the idea of HFS to discern the temporal switching patterns underlying a collection of assets. This extended computational approach proceeds in three steps. Firstly, a chosen $\mathbb{R}^p$ dimensional region is created based on observed time series data and then partitioned into B subareas. Upon each subarea, its chronological emergence is tracked along the temporal axis of the involved time series. Secondly, the limiting distribution of recurrent time between successive events according to each subarea-specific chronological emergence is analyzed. Then, a confusion matrix is constructed by stacking B estimated

permission rate vector resulted from each subarea. Lastly, clustering analysis is applied to group similar time points as if they are sharing the same phases of the dynamics. Via such clustering, the dynamic patterns of hidden phases are revealed by the cluster index. This is the fundamental idea underlying our proposed methodological extension of HFS.

The chapter is organized as follows. In Section 3.2, we proposed our main method in encoding multivariate processes. In Section 3.3, feature-weighting techniques of clustering are proposed for choosing potentially informative "extreme" events. Simulation experiments and real data analysis on multiple time series of one stock and multiple stocks in S&P500 index are performed in Section 3.4 and Section 3.5, respectively. A conclusion of this chapter is present in Section 3.6.

3.2. Multivariate Decoding

Based the discussion of the excursion process in Chapter 2, there are still at least three shortcomings for this approach: (i) independence of returns is assumed without considering time dependence; (ii) decoding in multivariate settings yet to be developed; (iii) a more data-driven way to define an "event" is required.

Here, we study this nonparametric region switching model under the assumption that X_t are independent. Such independence assumption is practically needed for computational purposes, though it might be often violated in real settings. Indeed, this assumption allows us to connect the asymptotic conclusion with real data analysis. Results may also be useful when the assumption is slightly violated. We later discuss modifications to our proposed computational approach such that it could accommodate small degree of violation of this independence assumption. For the last two shortcomings, a novel decoding method is proposed to discover the stochastic dynamic among multivariate time series.

3.2.1. The Method. Following the discussion of the excursion process, we shall extend the strategy from one dimension to multivariate. In single-dimensional time series, we consider volatility as a temporal aggregation of absolute large returns, so that extreme returns can be marked with appropriate choice of α and β in (2.2), and then the dynamic pattern is revealed by decoding the resultant 0-1 sequence. However, without any clear definition of an event of interest, it raises a problem in multivariate settings. The event should be defined to reflect the dependence or at least

local dependence of the multiple time series, for example, to mark data points contained in a pre-determined Euclidean subarea in $\mathbb{R}^p$. Intuitively, the subarea is of most interest if it contains data points exclusively from one underlying hidden state. A time period is currently under the control of this state if the subarea-specific events emerge chronologically in a high frequency.

Motivated by the idea of exploring local dependence, a new encoding and decoding approach is proposed as follows. In the encoding phase, a series of (rough) Euclidean "ball" is generated in $\mathbb{R}^p$ to mark points of interest, and so a series of 0-1 binary sequences can get returned. In the decoding phase, we treat the information of dynamics obtained from each "ball" as a feature and aggregate all pieces of information as one. The global pattern is ultimately discovered by clustering time points with similar feature sets.

Consider multivariate time series $\{X_t\}_{t=1}^N$. Let $B^{(v)}$ be the $v\text{-}th$ "ball" with pre-fixed boundary. A new excursion process is defined by,

$$E_t^{(v)} = \begin{cases} 1 & X_t \in B^{(v)} \\ 0 & Otherwise \end{cases} \tag{3.1}$$

Under the assumption of Theorem 2.2.1, the waiting time between two successive 1's in $E_t^{(v)}$ converges to a geometric distribution. The emission probability of 1's given hidden state S_j now becomes

$$p_j^{(v)} = \int_{B^{(v)}} dF_j \tag{3.2}$$

where F_j is the conditional CDF given S_j. A series of alternating hidden regions, for example $(S_1, S_1, S_2, S_1, ...)$, can be computed in model selection, and the corresponding region-based permission probability, which is $(p_1^{(v)}, p_1^{(v)}, p_2^{(v)}, p_1^{(v)}, ...)$ in this example, can be estimated by MLE of geometric distribution. Denote the N-length estimated probability vector as $\hat{p}^{(v)}$ which is the feature generated by $B^{(v)}$. Iteratively generate subarea $B^{(v)}$ for $v = 1, 2, ..., V$, then V resultant features can get obtained.

Note that $\hat{p}^{(v)}$ could be a vector with only a single value if the true permission probability are comparable given different hidden regions, for example, when $\int_{B^{(v)}} dF_0 \cong \int_{B^{(v)}} dF_1$. In this

case, features are less relevant or even redundant. In contrast, features may contain significant information about the dynamics when $\int_{B^{(v)}} dF_0$ differs a lot from $\int_{B^{(v)}} dF_1$.

3.2.2. Ball Generation. To make features more representative and less correlated, the "balls" should ideally be generated mutually disjointed and samples should get selected only once. In real data analysis with finite samples, it is neither efficient nor effective to determine the boundary for each "ball", especially when the dimension is high. Instead, we turn to select a fixed proportion of samples at each iteration and make each group of the samples less overlapping.

To generate fewer overlapping sample groups, cluster analysis can be applied for the purpose. K-Means would be the most appropriate method due to its scalability and property of getting relatively balanced clusters. Assume V clusters get returned via K-Means, then a rough "ball" can be generated by searching for M nearest neighbors starting from the centroid of each cluster. The reason that we fix the size is to make sure there is enough data selected in each cluster. There is actually a tradeoff between the sample size of recurrent time and the magnitude of the excursion. We will keep using the proportion that is advocated in one-dimensional settings, say $\frac{M}{N} = 0.1$. As a result, V subarea gets returned, and each includes exactly M data points. The V is chosen very large in practice, say 100, so a sample is chosen 10 times on average. Here, we lose less information but via involving more correlated features.

Let $\mathbb{X} = [X_1, X_2, ..., X_N]^T$ be a $N \times p$ matrix that records the time series $\{X_t\}_{t=1}^N$ where $X_t \in \mathbb{R}^p$. The feature generation algorithm is described in **Algorithm 3**. In the end, we simply stack all the features into a $N \times V$ matrix $\mathbb{P} = [\hat{p}^{(1)}, \hat{p}^{(2)}, ..., \hat{p}^{(V)}]$ as the output. The next task is resolved by feature selection or feature weighting techniques discussed in the next section.

Algorithm 3 Feature Extraction

Input: Data matrix $\mathbb{X}$

1. Apply K-Means to $\mathbb{X}$, and get V cluster centroids $C_1, C_2, ..., C_V$.
2. Loop: cycle through every C_v
   a. Search for its M nearest neighbors in $\mathbb{X}$, denoted as $B^{(v)}$
   b. Generate a 0-1 excursion process via (3.1), denoted as $\{E_t^{(v)}\}_{t=1}^N$

c. Apply **Algorithm 1** to $\{E_t^{(v)}\}_{t=1}^N$ to get emission probability $\hat{p}^{(v)}$

3. Stack all $\hat{p}^{(v)}$'s into a $(N \times V)$ matrix $\mathbb{P} = [\hat{p}^{(1)}, \hat{p}^{(2)}, ..., \hat{p}^{(V)}]$,

and record $B^{(v)}$'s in a set $\mathbb{B} = \{B^{(1)}, B^{(2)}, ..., B^{(V)}\}$.

Output: a confusion matrix $\mathbb{P}$ and a set $\mathbb{B}$

3.3. Feature Weighting

The decoding result is finally achieved by clustering similar time points in $\mathbb{P}$. We will use K-Means as an example to illustrate the idea. K-Means minimizes the sum of within-cluster error via iteratively assigning each object by its closest centroid and updating each centroid consequently. Define Y_{iv} is the v-th feature in the i-th sample, for $v = 1, 2, ..., V$, and C_j is the centroid of the j-th cluster S_j, for $j = 1, 2, ..., k$. The optimization problem can be specified as to minimizing the following quantity,

$$\sum_{j=1}^{k} \sum_{i \in S_j} \sum_{v=1}^{V} D(Y_{iv}, C_{jv}) \tag{3.3}$$

where $D(.)$ is a metric.

As what is discussed before, features may have different degrees of relevance, but K-Means treats every single feature equally, regardless of the actual relevance. As a consequence, clustering results could be greatly biased by the irrelevant features, while the more relevant features are overwhelmed. This weakness can be resolved by feature selection or feature weighting which is discussed as follows.

The research in feature weighting of clustering can be traced back to 1984. Different from feature selection, feature weighting approaches usually lead to better performance by iteratively conducting clustering and adjust feature weights based on the result in the last step. A survey on feature weighting of K-Means is available in [**8**]. Typically, the goal is to minimize the within clustering dispersion by updating the feature weight w_v for feature position v. The optimization problem (3.3) is then rewritten as,

$$\sum_{j=1}^{k}\sum_{i\in S_j}\sum_{v=1}^{V} w_v D(Y_{iv}, C_{jv}) \tag{3.4}$$

Usually, w_v is set so that $\sum_{v=1}^{V} w_v = 1$. Note that w_v may also vary in different clusters, and metric $D(.)$ can be generalized to non-Euclidean distance, like Minkowski's [**9**], but they are beyond our focus in this paper.

3.3.1. Related Works. Feature Weight Self-Adjustment mechanism(FWSA) [**114**] is designed to adjust feature weight to simultaneously minimize the separations within clusters and maximize the separations between clusters. The importance of a feature to the clustering quality is measured based on a function of sum of separations within clusters, denoted as a_v, and sum of separations between clusters, denoted as b_v, and feature weight is updated, iteratively,

$$w_v^{(c+1)} = w_v^{(c)} - \eta\,(w_v^{(c)} - \frac{b_v^{(c)}/a_v^{(c)}}{\sum_u b_u^{(c)}/a_u^{(c)}}) \tag{3.5}$$

where c indicates the current step, and $c+1$ is the next step; η is the learning rate. The updated weight vector still sums up to 1. In the original paper, η is set as 0.5. FWSA mechanism significantly improves clustering quality in experiments. In addition, it takes considerable advantage that no extra parameter is required to be specified.

The second method weights features according to mutual information. As Shannon Entropy is widely used as criteria of clustering quality, its variant, Mutual information, measures the amount of information obtained about the clusters which can be interpreted through another random variable. The minimum of MI is 0 if a particular feature does not contribute any new information about what its cluster might be. Maximum mutual information is reached when a feature can perfectly recreate the clusters. A drawback of MI is that a feature with numerical value has to be categorized before applying the discrete-version formula, and entropy tends to increase with the number of categories. The Normalized Mutual Information(NMI) solves the problem by standardizing the MI number always between 0 and 1. Fortunately, no extra binning is required in matrix $\mathbb{P}$ since each feature

is a sequence of discrete probability numbers. Feature weights are updated based on the idea that more relevant features to the current clustering result weights more than redundant features.

$$w_v^{(c+1)} = w_v^{(c)} - \eta\,(w_v^{(c)} - \frac{NMI(L^{(c)}, Y_{.v}^{(c)})}{\sum_u NMI(L^{(c)}, Y_{.u}^{(c)})}) \tag{3.6}$$

where $Y_{.v} = (Y_{1v}, Y_{2v}, ..., Y_{Nv})^T$, $L^{(c)}$ is the cluster labels returned at the current step, and η is the learning rate. As an unsupervised algorithm, the quality of clustering may get out of control, especially when signal-to-noise ratio is relatively low, so the noise may get exaggerated in the iteration.

3.3.2. Feature Weighting Clustering for Decoding. A new feature-weighting clustering algorithm is designed for the decoding procedure. As is claimed in one-dimensional settings, the decoding result is reliable if the true permission rates difference between two hidden states is large. It is the reason that α and β in (2.2) are tuned to enlarge the difference between the tailedness of underlying distributions. Inspired by this idea, the feature importance can also be measured by the estimated permission rate delta.

Recall the time series data $\{X_t\}_{t=1}^N$. In a iterative fashion, let $L_t^{(c)}$ be the cluster label for data point X_t in the current step. When $k = 2$, $L_t^{(c)}$ only takes two values corresponding to two hidden states, say "state0" and "state1". Denote the v-th feature is generated by a selection area $B^{(v)}$, then permission probability $p_j^{(v)(c)}$ upon $B^{(v)}$ given hidden state j can be further estimated by,

$$\hat{p_j}^{(v)(c)} = \frac{\sum_{t=1}^N 1\{L_t = j, X_t \in B^{(v)}\}}{\sum_{t=1}^N 1\{L_t = j\}} \tag{3.7}$$

Especially when $k = 2$, the feature importance for feature v can be quantified based on $|\hat{p_1}^{(v)(c)} - \hat{p_0}^{(v)(c)}|$. The greater the absolute difference, the more important feature v is. The feature weight can be simply updated by,

$$w_v^{(c+1)} = w_v^{(c)} - \eta\,(w_v^{(c)} - \frac{|\hat{p_1}^{(v)(c)} - \hat{p_0}^{(v)(c)}|}{\sum_u |\hat{p_1}^{(u)(c)} - \hat{p_0}^{(u)(c)}|}) \tag{3.8}$$

Without any prior information, let's assume the size of the two hidden states is balanced. Then, the estimated delta is simply measured by the proportion of the two cluster labels in $B^{(v)}$. The more purity of cluster in $B^{(v)}$, the more important feature v should be. It actually enlighten us to look at the Shannon entropy in $B^{(v)}$ as a smooth approximation to $|\hat{p_1}^{(v)} - \hat{p_0}^{(v)}|$.

$$w_v^{(c+1)} = w_v^{(c)} - \eta \left(w_v^{(c)} - \frac{\mathcal{F}(H_c(B^{(v)}))}{\sum_{v=1}^{V} \mathcal{F}(H_c(B^{(v)}))} \right) \tag{3.9}$$

where $\mathcal{F}(x) = 1 - \frac{x - \min(x)}{\max(x) - \min(x)}$ and $H_c(B^{(v)})$ denote the Shannon entropy of cluster labels of data points in $\{B^{(v)}\}$ at the current step. The feature weight is measured by one minus the scaled entropy of clusters in $B^{(v)}$. Note that the entropy-type feature weighting procedure can be easily generalized when $k > 2$. Moreover, it takes advantages in geometric interpretation, which is illustrated in the simulation study.

3.4. Simulation Experiments

3.4.1. Independent Processes. Independent Bivariate Normal processes are simulated with mean 0 and 2 types of covariance matrix varying over time. The data is generated with a covariance matrix Cov_0 in a short period of time, then switching to the other matrix Cov_1 for a period and switching back, so on and so forth. Each short period indicates a state hidden behind the time series, and the conditional distribution given a state is identical. There are 10 alternating periods in total, and the time length for each period is uniformly distributed by $Unif([200, 400])$.

Consider 5 different simulation scenarios, named "Case1", up to "Case5". The detail about the simulated covariance matrix is reported in Appendix B. A confusion matrix is firstly obtained via feature generation (**Algorithm 3**), and then feature weighting K-Means is applied to clustering time points in hidden states. Figure 3.1 illustrates a decoding result in "Case1". It shows that the underlying dynamics pattern can be almost perfectly discovered although some stamps around 1000 are misclassified (accuracy is 0.87).

Four feature weighting clustering algorithms described in (3.5), (3.6), (3.8), and (3.9) are compared. For the convenience of comparison, clustering accuracy is calculated and used to measure the quality of decoding. Denote the first feature-weighting algorithm in (3.8) as "MethodA", and

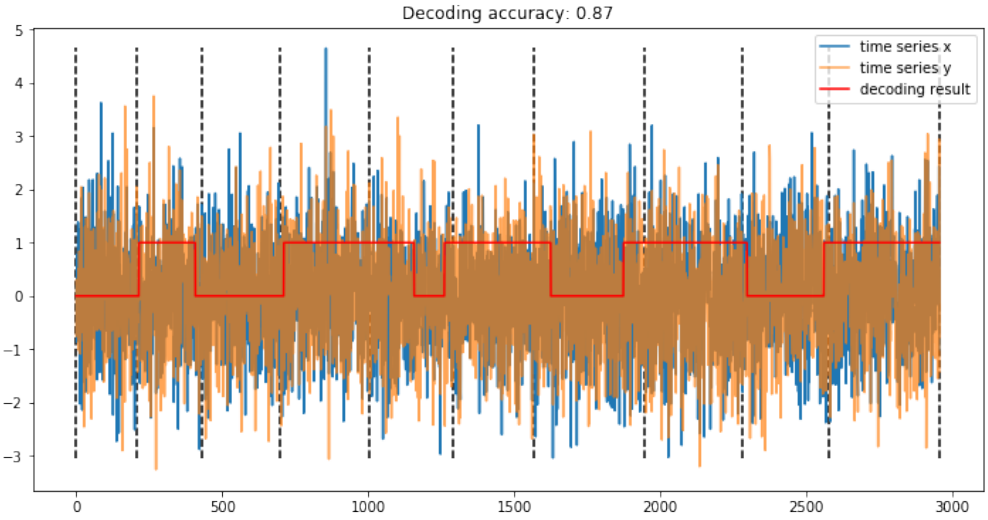

FIGURE 3.1. Dataset simulated from bivariate Gaussian "Case1"; vertical dashed line indicates the true change points; red solid line reflects the segmentation result via (3.9)

the second one in (3.9) as "MethodB". Dataset is simulated for at least 100 times, and the decoding accuracy is reported in Table 3.1.

TABLE 3.1. Decoding Accuracy

Simulation Case	FWSA	NMI	MethodA	MethodB
Case1	0.8048 (0.1046)	0.8021 (**0.0960**)	0.8145 (0.1111)	**0.8286** (0.1026)
Case2	0.9377 (0.0190)	0.9415 (0.0186)	**0.9502** (0.0145)	0.9489 (**0.0145**)
Case3	0.9354 (0.0192)	0.9389 (0.0170)	0.9480 (0.0176)	**0.9493** (**0.0146**)
Case4	0.9213 (0.0218)	0.9249 (0.0217)	0.9378 (**0.0142**)	**0.9390** (0.0142)
Case5	0.8920 (0.0490)	0.8764 (0.0541)	0.9030 (**0.0476**)	**0.9110** (0.0579)

It turns out that the feature weighting methods are adapt to the decoding framework well. "MethodB" that weights features according to the entropy of each Euclidean "ball" outperforms others. In "Case1", the join distribution given a hidden state is Gaussian with a unit variance but different correlations. The join distribution in the two states can be visualized from Figure 3.2(A). "Balls" with relatively high feature weights are highlighted in Figure 3.2(B). It looks that the algorithm is trying to pay more attention to the "balls" located in the right-up and left-bottom corners, in which the distributions differ a lot, see Figure 3.3. While in "Case2", weights are concentrated to "balls" located around the four corners. It claims that our new feature weighting strategy can

truly find out the key difference between the joint distributions, and "balls" with high weights play a significant role in detecting the distribution changes.

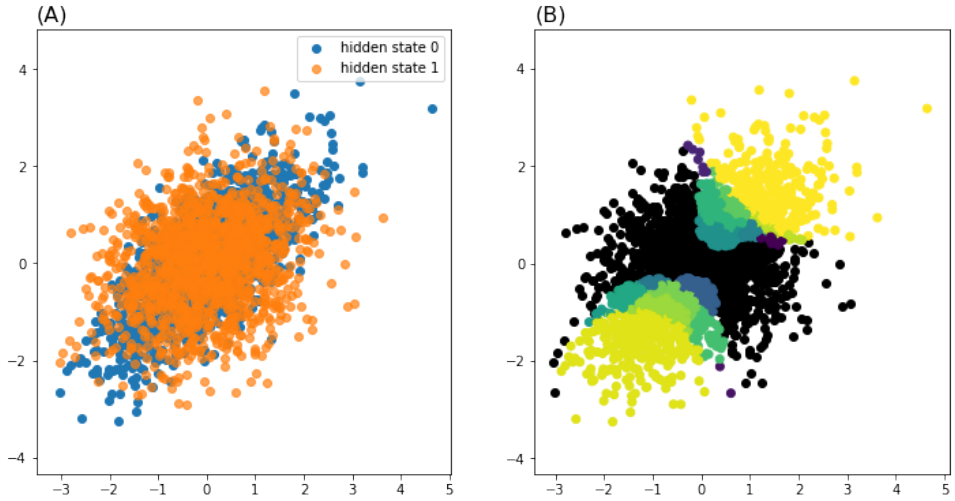

FIGURE 3.2. Dataset simulated from bivariate Gaussian "Case1"; (A) scartterplot from two hidden states; (B) data points are plotted in back; "balls" with high weights are painted in different color

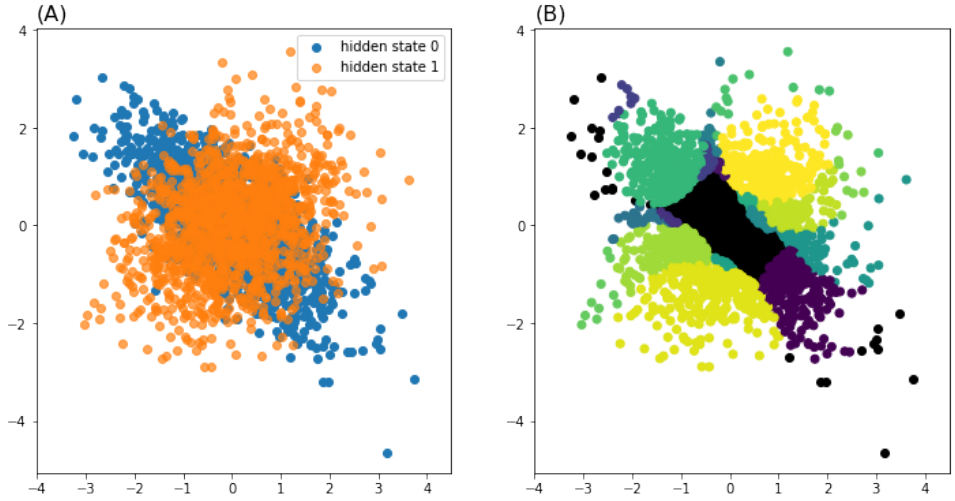

FIGURE 3.3. Dataset simulated from bivariate Gaussian "Case2"; (A) scartterplot from two hidden states; (B) data points are plotted in back; "balls" with high weights are painted in different color

3.4.2. Serial Dependent Processes. In this section, we discuss some extensions to our approach when serial dependence is present in the time sequence. This problem is related to change

point detection in time series models. To detect structural breaks in variance, authors in **[62]** studied cumulative sums of squares in case of independent sequence. Later on, the test statistic is modified by looking into the stability breaks of the autocovariance function $\gamma(r) = E[X_t X_{t+r}]$ where r is the time lag **[16]**.

Motivated by the idea, we extend the multivariate decoding procedure to a single time series with weak serial dependence. A multivariate process is made up by coupling time point with its r-lags, say $\{Z_t\}_t = \{(X_t, X_{t+1}, ...X_{t+r})\}_{t=1}^{N-r}$. We suppose the $(r+1)$-dimensional variables can represent the covariance structure, and modify the decoding algorithm as follows. For the validation of the independence assumption, it is necessary to break the local dependence. Time sequence $\{Z_t\}_t$ is partitioned by l-length window, so $\lfloor \frac{N-r}{l} \rfloor$ time pierces are obtained. Time points in each window are then randomly permuted and denote the new sequence as $\{\tilde{Z}_t^l\}_t$. The choice of l could be very tricky. A too small l has nothing to do with breaking the dependence, while a too large l tremendously destroys the true dynamic pattern. We find l around 30 is proper given that the size of a hidden region is at least 300.

Datasets are simulated based on AR(1) and AR(2) models. Independent standard normal variables were used as innovations. In AR(1) settings, parameters are set $\phi|state0 = 0.3$ and $\phi|state1 = 0.7$ given hidden state "state0" and "state1", respectively. In AR(2), the pair of parameters is $(\phi_1, \phi_2)|state0 = (0.3, 0.2)$ and $(\phi_1, \phi_2)|state1 = (0.5, 0.3)$. The switching pattern of the hidden states is generated in a fashion similar to that in Section 5.1. We choose $r = 1$ and 2 to make up new time sequence $\{Z_t\}_t$ in AR(1) and AR(2) settings, respectively. The average decoding accuracy for simulation in AR(1) is 0.7791 (0.0861), and 0.8045 (0.0705) in AR(2).

3.5. Real Data Application

3.5.1. Triplet Time Series. The relationship between returns, trading volume, and transaction numbers has been received great amounts of attention in finance. Under one old Wall Street adage that "it takes volume to move prices", volume had been increasingly used as a cause of return volatility. It can be explained that volume can reflect the extent of disagreement about a security's value in stock price. However, it would be modified later that it is the number of trades but their

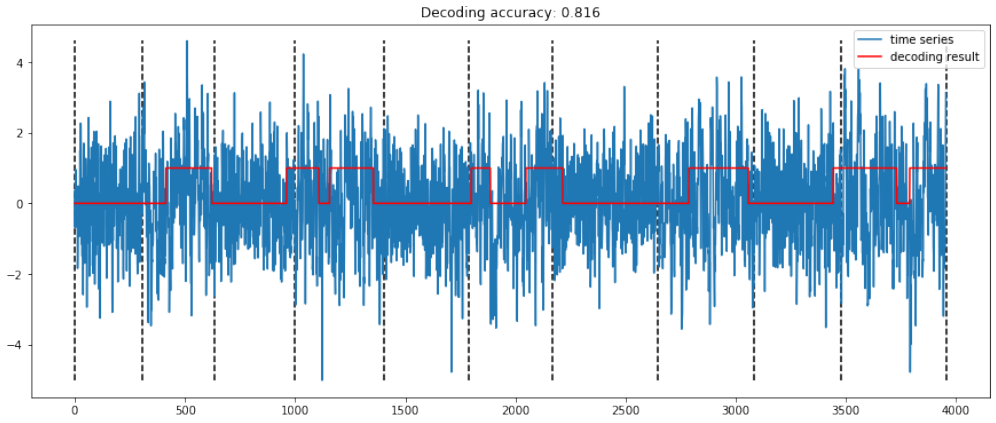

FIGURE 3.4. Dataset simulated from AR(1) with $\phi|state0 = 0.3$ and $\phi|state1 = 0.7$

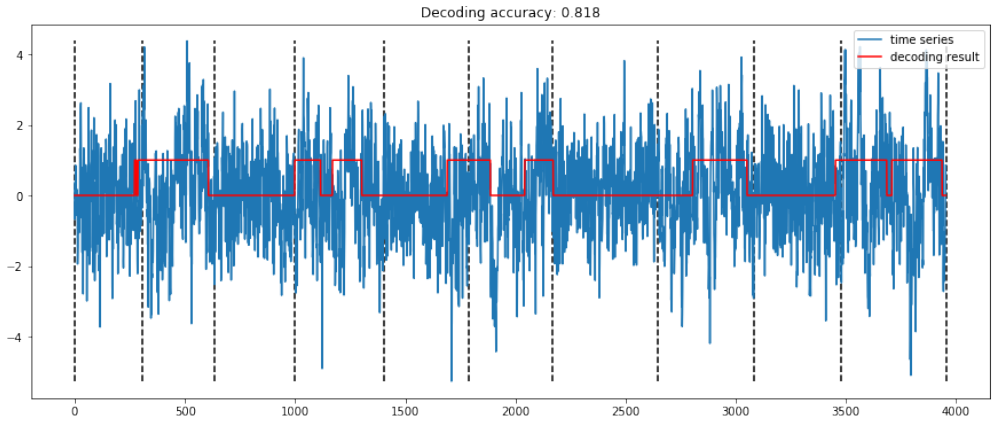

FIGURE 3.5. Dataset simulated from AR(2) with $(\phi_1, \phi_2)|state0 = (0.3, 0.2)$ and $(\phi_1, \phi_2)|state1 = (0.5, 0.3)$

sizes that generate volatility **[78]**. It would also be shown that to recover normality in asset returns, the number of trades is a better time change than the traditionally used trading volume.

It is claimed in **[60]** that direct modeling may have difficulty capturing the intricate dynamic structure, especially given the lack of goodness-of-fit in dynamic linear regression. A nonparametric approach was advocated to explore each of the three dimensions separately by segmenting high-volatility and low-volatility states, and then combine them to reflect a single stock dynamics. However, the idea of divide-and-conquer may fail to capture the real association among the three but be biased by the integrated microstructure noises.

In the experiment, we track the 3-dimensional time series of a single stock from S&P500. The log return, volume, and transaction number at every 1-min interval are recorded. To mitigate the influence of activities near opening and closing, We truncate the transaction time from 10am to 4pm, so there are 360 data points per business day. Again, no prior knowledge about the stochastic mechanism needs to be assumed. Via our proposed method, the time axis is segmented into equilibrium and off-equilibrium periods to represent the latent state-space trajectory underlying the single stock's dynamics.

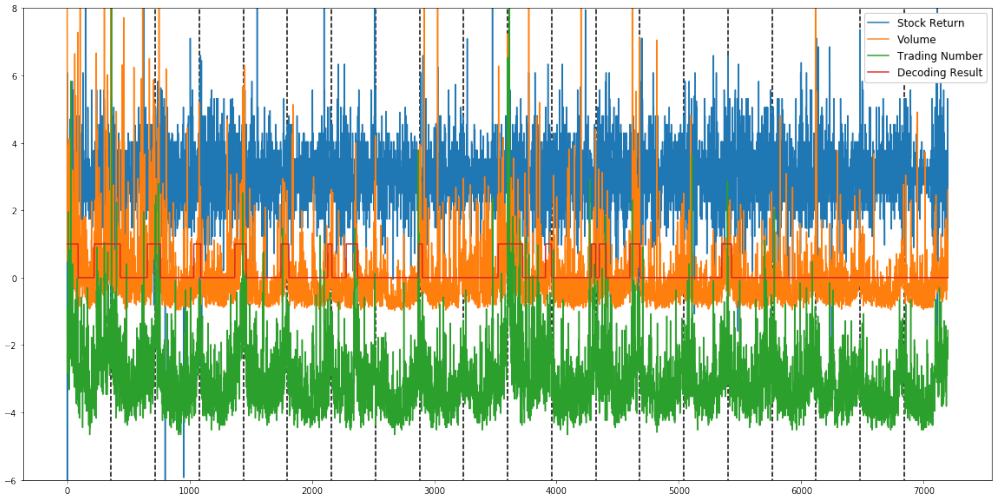

FIGURE 3.6. Trivariate time series of IBM

Figure 3.6 shows minutely trivariate time series of IBM in January 2006. Each of the dimensions is standardized to have a mean 0 and standard deviation of 1. A constant is added or subtracted to returns and trading numbers for better visualization so that the 3 time series are clearly viewed in one panel. The vertical dashed line indicates a date change. It shows that volume and trading number are highly correlated. They would rhythmically go up and down simultaneously. The decoding result (0-1 sequence) obtained by our method is plotted in a red line, which represents the two hidden states switching throughout the whole period. It looks that the segmentation can successfully capture the time when volume and trading number both increase heavily, see state code "1". If the increment is not that much, it is marked as in equilibrium state, see the right part in Figure 3.6.

The next question is, what is the association among these 3 time series given different hidden states? 2-D scatterplots in Figure 3.7 can roughly illustrate the answer. The correlation between volume and trading number is much higher in "state1". The surprising pattern is that the corresponding stock returns in the same period have a much lower deviation than that in "state0". That is to say, stock return tends to stay in low-volatility once volume and trading number are significantly going up together. This phenomenon is shown more clearly in Adobe's stock, see Figure 3.8. Our findings contradict the previous argument that volatility is highly correlated to volume or trading numbers.

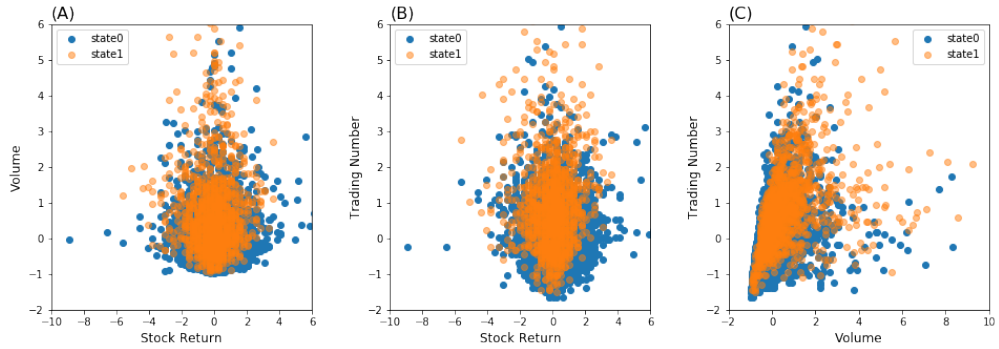

FIGURE 3.7. 2-D scatterplot for IBM: (A) returns v.s volume; (B) returns v.s trading numbers (C) volume v.s trading numbers

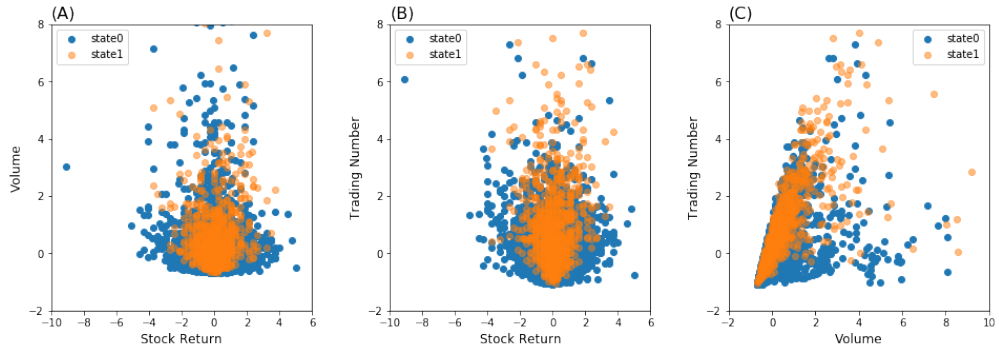

FIGURE 3.8. 2-D scatterplot for ADBE: (A) returns v.s volume; (B) returns v.s trading numbers (C) volume v.s trading numbers

3.5.2. Multivariate Returns. In this experiment, we apply the decoding approach to discover the time-varying dependence among bivariate and multivariate stock returns. In the first

example, a pair of indexes is chosen from one of the categories of S&P500 based on Global Industrial Classification Standard(GICS). For example, 'Amazon' and 'Ebay' are coupled together as a pair of representatives for internet retails. The price returns are calculated in 1-min time interval. The volatility segmentation result is shown in Figure 3.9. Volitility state shows up rhythmically every day in the first business week, and then the frequency tends to disappear in the second week. The return fluctuates even severely in the third week and then goes back to the first state in the end. Kernel density estimations for returns in high-volatility and low-volatility stages are plotted separately in Figure 3.10. It shows that both distributions have their tails heavier when in volatility stage.

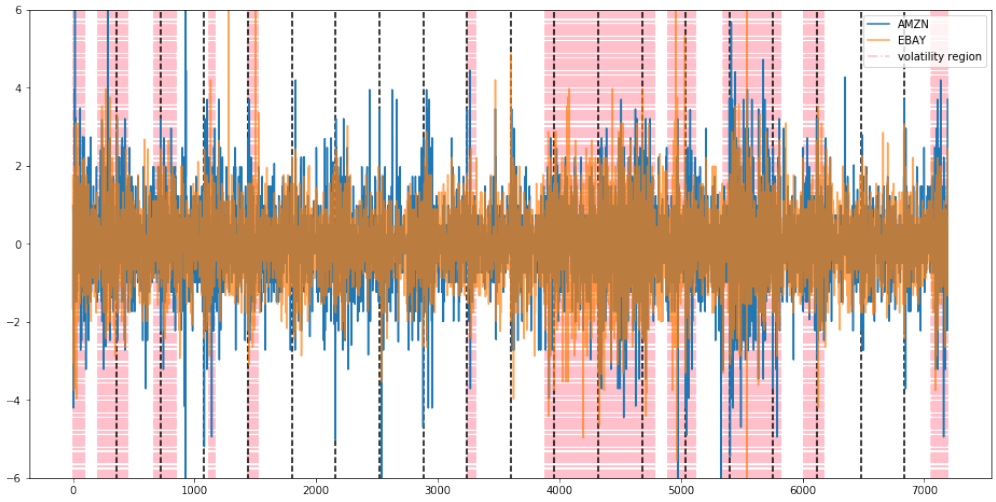

FIGURE 3.9. Bivariate returns of Amazon and Ebay in January 2006

In the second example, we pick up 9 semiconductor indexes from S&P500 and segment the time axis into high-volatility and low-volatility regions. To measure the heavy-tailedness, we calculate the probability with which return X goes beyond the z-standard deviation limits, for $z = 1, 2, 3$,

$$P(X < z\,\sigma) + P(X > z\,\sigma)$$

The heavy-tailedness is calculated for high-volatility and low-volatility, respectively. The delta values between high-volatility and low-volatility is reported in Figure 3.11. All the positive delta values indicate that the 9 indexes would have a heavier tail simultaneously when in the volatility period,

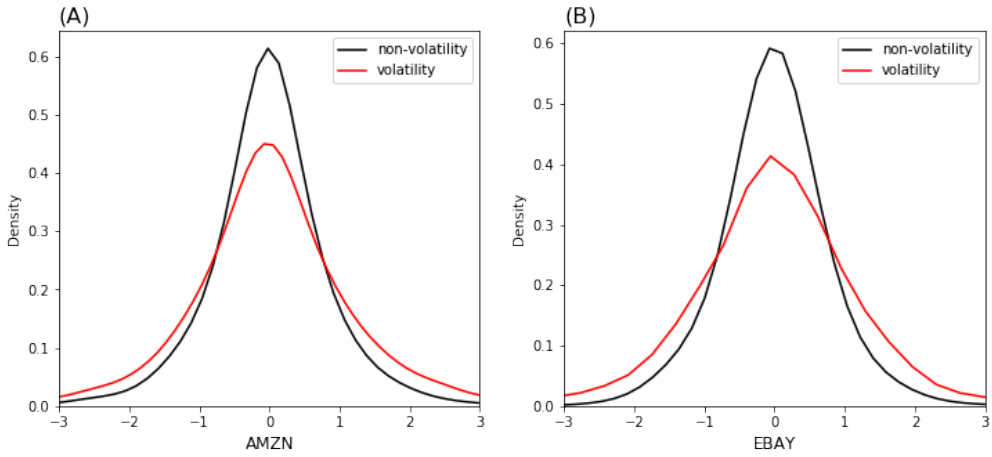

FIGURE 3.10. Kernel density estimation for data points on high-volatility and low-volatility region; (A) Amazon; (B) Ebay

but the heavy-tailedness is quite different. For example, 'Advanced Micro Devices(AMD)' and 'Intel(INTC)' have relatively stable returns when in volatility; while returns of 'Qualcomm(QCOM)' and 'Nvida(NVDA)' fluctuate more heavily.

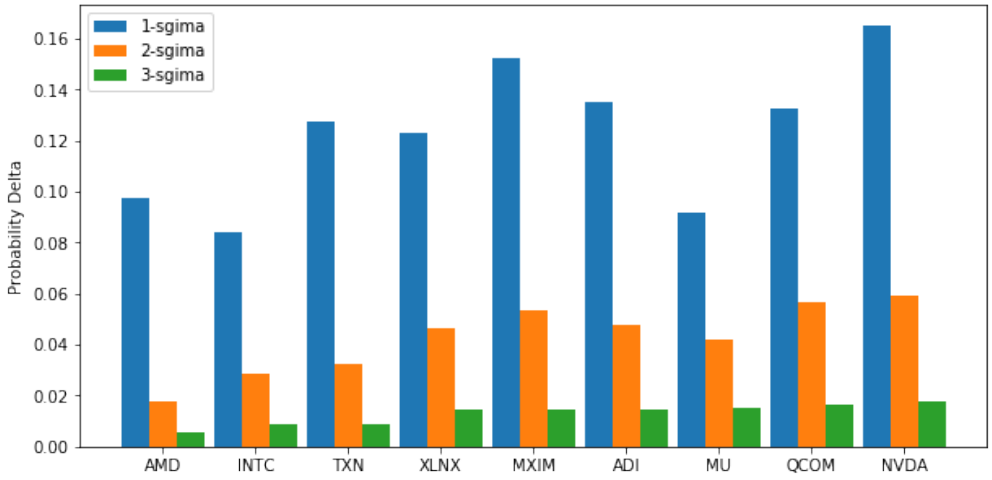

FIGURE 3.11. Heavy-tailedness delta between high-volatility and low-volatility; 9 indexes from left to right is 'AMD', 'INTC', 'TXN', 'XLNX', 'MXIM', 'ADI', 'MU', 'QCOM', and 'NVDA'

3.6. Conclusion

In the chapter, we try to break down the complicated model framework and to directly investigate the volatility dynamic patterns underlying multivariate stock time series. A feature engineering strategy is proposed from feature extraction to feature weighting, and our clustering results can successfully detect the switching region in which the nonlinear dependence differs a lot. In the real data experiment, we revised the former claim on the relationship among returns, trading volume, and transaction numbers, and measure the association in multiple return time series. Despite the weakness in modeling long-term serial dependence and forecasting, the data-driven approach established a platform to study distributional heterogeneity, which is commonly observed in reality. In the future, it can incorporate time series models, like GARCH, to perform a more detailed analysis.

CHAPTER 4

Multiple Change Point Analysis and Stability Detection

4.1. Introduction

Change point analysis aims to detect abrupt distribution changes in time-ordered observations and partition such time series into homogeneous segments. The study can be traced back to [**28**, **64**, **92**]. So far, it has been playing a crucial role in diverse fields including bioinformatics [**88**, **94**], behavioural science [**58**, **98**], neuroimage [**22**], climate science [**97**], finance [**111**], and speech recognition [**79**]. Generally, the analysis can be conducted via either parametric or nonparametric approaches. Parametric approaches rely heavily upon the assumption that the underlying distributions belonging to a known family, and likelihood or penalized likelihood functions are generally involved [**11**,**26**,**127**]. Instead, the nonparametric analysis makes the least assumption about the distributions, so they can be used in a wider variety of applications.

A vast number of attention has been received in the nonparametric literature in the past decade. For instance, authors in [**67**, **77**] attempted to estimate the likelihood ratio using KL divergence; Chen and Zhang [**25**] proposed a graph-based approach and applied it in multivariate non-Euclidean data. Zou et al. [**130**] developed an empirical likelihood approach to discover an unknown number of change points via BIC. Matteson and James [**83**] present a U-statistic to quantify the difference between the characteristic functions of two segments. Lung-Yut-Fong et al. [**1**] generalized Mann-Whitney rank-based statistic to multivariate settings. Arlot et al. [**100**] improved the kernel-based method by [**51**] with a generalized model-selection penalty. However, most the existing nonparametric research focused on the single change-point problem, and the extension of multiple change point detection is achieved via dynamic programming [**1**, **51**, **100**] or bisection procedure [**83**, **90**, **118**]. It is still scarce in the literature to efficiently discover multiple change points under multivariate settings, especially when the covariance structure changes in chronological order.

In this chapter, a new nonparametric approach is proposed to detect multiple distributional changes within independent time-ordered observations. The basic idea is to systematically select a subset of the data points at each iteration and encode the continuous observations into a sequence of Bernoulli variables. The number of change points and their locations are estimated by aggregating all the dynamics information discovered from the Bernoulli processes. Instead of working on the unknown distribution directly, the proposed approach takes advantage of dividing the problem into several easier tasks, so that the maximum likelihood approach can be applied to analyze the Bernoulli processes respectively. We claim that the divide-and-concur framework is robust to any underlying distributions and can be implemented in conjunction with other parametric approaches.

Another important extension of the aggregation technique is stability change point detection. Stability selection introduced by [**84**] was designed to improve the performance of variable selection and provide control for false discoveries. We demonstrate that the idea of aggregating results by applying a procedure to subsamples of the data can be well implemented under our framework. One can aggregate the estimation from the Bernoulli sequences, and select the estimated change point locations with votes beyond a predetermined threshold. As far as we know, this could be the first method in the change point literature that holds both asymptotic property and finite-sample control of false discoveries.

The chapter is organized as follows. We start with an efficient algorithm for searching multiple change points within a change-in-parameter Bernoulli sequence in Section 4.2. In Section 4.3, we propose the main divide-and-concur framework to analyze multivariate observations. In Section 4.4, the stability detection technique is applied under our change point framework. In Section 4.5, a strategy is provided to weighting the results from different sample sets for practical usage. Numerical experiments are shown in Section 4.6 to compare the model performance with that of other nonparametric approaches, and real data applications including categorical and continuous data in univariate and multivariate settings are reported in Section 4.7. We note that the proposed approach can be easily generalized to categorical or ordinal data though we mainly discuss continuous observations in this chapter.

4.2. Sequence of Bernoulli variables

4.2.1. Background. Consider a sequence of 0-1 independent Bernoulli variables $\{E_t\}_{t=1}^{N}$. Suppose that k change points are embedded within the sequence at locations $0 = \tau_0^* < \tau_1^* < ... < \tau_k^* < \tau_{k+1}^* = N$, so the observations are partitioned into $k+1$ segments. Observations within segments are identically distributed but observations between adjacent segments are not. Specially, $E_t \overset{iid}{\sim} Bern(p_i)$ for $E_t \in \{E_{\tau_i^*+1}, ..., E_{\tau_{i+1}^*}\}$, for $i = 0, ..., k$. Now, given the number of change points k, one task of change point detection is to estimate the k locations. In the most general case, both number of change points and their locations need to be estimated.

Change point analysis in a Bernoulli-variable sequence was well studied when $k = 1$. Hinkley and Hinkley [**56**] provided asymptotic distributions of likelihood ratio statistics for testing the existence of a change point. Pettitt [**93**] introduced CUSUM statistics and showed its asymptotical equivalence to the maximum likelihood estimator. Miller and Siegmund [**86**] investigated maximally selected chi-square statistics for two-sample comparison in a form of 2×2 table. Later on, Halpern [**72**] advocated a statistic based on the minimum value of Fisher's Exact Test. When $k > 1$, Fu and Curnow [**42**] firstly attempted to search for optimal change points such that the likelihood function is maximized. However, it still lacks a computationally efficient algorithm especially when k is large.

In this section, we present a new algorithm to address the problem of performing multiple change points detection within a Bernoulli-variable sequence. An exhaustive searching procedure is proposed but with relatively feasible time complexity. The idea is motivated by HFS [**59**] which was designed to detect dynamics phase shifts from one episode to another in financial data. By tracking the recurrence of 1's in the time axis, observations are partitioned into disjoint segments with different emergence intensities in a fashion of dynamic programming. Thus, change points between adjacent segments are detected such that the likelihood or penalized-likelihood functions are maximized.

4.2.2. Multiple Change Points Searching Algorithm. For simplicity of computation, we only consider the situation in which change point locates at the emergence position of 1's. Suppose that the number of 1's in the i-th segment is M_i, so the total number of 1's is $M = \sum_{i=1}^{k+1} M_i$ and the total number of 0's is $N - M$. By further supposing that the recurrent time can be 0 if two

1's appear consecutively, and $R_1 = 0$ if $E_1 = 1$, and $R_{M+1} = 0$ if $E_N = 1$, the Bernoulli-variable sequence can be represent by a sequence of recurrent time between consecutive 1's, denoted as $\{R_t\}_{t=1}^{M+1}$. Especially, there are $M_i + 1$ recurrent times in the i-th segment where $R_t \sim Geom(p_i)$. The task then becomes to search for the change points within the recurrent-time sequence.

The searching procedure is done by iteratively taking off the smallest number R_{min} from the rest R_t's and combine the time points within R_{min}. For example, if R_{min} is the recurrent time between j and j', we combine the locations from $(j+1)$ to j' as a time window, denoted as $w_{(j+1)\to j'}$. Here, we suppose that $E_j = 1$, $E_{j'} = 1$, and $E_t = 0$ for $t \in (j, j')$. In the next step, if the smallest R_t is taken from the recurrent time between j' and j'', a new time window is recorded from $(j'+1)$ to j'', named $w_{(j'+1)\to j''}$. We can further combine the two consecutive time windows $w_{(j+1)\to j'}$ and $w_{(j'+1)\to j''}$ into $w_{(j+1)\to j''}$. Indeed, we iteratively merge a pair of nearest 1's at each step and update the recorded time windows according to their connectivity. The recorded time windows contain recurrent time with relatively smaller values, which corresponds to a period with high frequency of 1's. Hence, the boundaries of the time windows can be extracted as potential change point locations that partition the observations into segments with low and high Bernoulli parameters.

So far, the algorithm works very similarly to the hierarchical clustering with a single-linkage, by merging two closest single 1's or two groups from bottom to top. However, it is known that this greedy algorithm does not guarantee global optimization. Our remedy is to set a tuning parameter C^* to control the minimal length of the recorded high-intensity segments. Additionally, we count the number of R_t absorbed within each recorded time window. Continuing with the above example, the count of recurrent time for window $w_{(j+1)\to j'}$ and $w_{(j+1)\to j''}$ is denoted as $C_{(j+1)\to j'} = 1$ and $C_{(j+1)\to j''} = 2$, respectively. The recorded time window, for example, $w_{\cdot\to\cdot\cdot}$ is regarded as a high-intensity segment only if its count $C_{\cdot\to\cdot\cdot}$ is greater than the threshold C^*. Hierarchical clustering with a single-linkage is just a special case that $C^* = 0$. Another most extreme case is when $C^* = M$, so there is no period having a count number above C^*, thus no change point exists. Without any prior knowledge about the minimal length of the segments, we run over all the choice of C^* starting from 0 to M to generate all possible partitions. The optimum is returned to fit the Bernoulli or Geometric observations best.

Suppose the observations are partitioned into $\tilde{k}+1$ segments via $\tilde{k}$ time window boundaries or change points $\tilde{\tau}_1, ..., \tilde{\tau}_{\tilde{k}}$. The Bernoulli parameter $\tilde{p}_i$ between $\tilde{\tau}_{i-1}$ and $\tilde{\tau}_i$ can be estimated by MLE $\hat{\tilde{p}}_i = \frac{\{\#\text{ of } 1's \in (\tilde{\tau}_{i-1}, \tilde{\tau}_i)\}}{\tilde{\tau}_i - \tilde{\tau}_{i-1}}$. To measure the goodness-of-fit, model selection is done by maximizing log-likelihood function within each segment, while penalizing the number of change points k and related estimation parameters. The penalized function or loss can be written by,

$$L(\tilde{\tau}_1, ..., \tilde{\tau}_{\tilde{k}}) = -2 \sum_{i=1}^{\tilde{k}+1} \sum_{t \in (\tilde{\tau}_{i-1}, \tilde{\tau}_i)} [E_t log \hat{\tilde{p}}_i + (1 - E_t) log(1 - \hat{\tilde{p}}_i)] + \phi(N) Q_{\tilde{k}} \tag{4.1}$$

where Q_k is the total number parameters; $\phi(N)$ is the penalty coefficient; $\phi(N) = 2$ for AIC and $\phi(N) = log(N)$ for BIC.

Suppose that $W(.)$ is a mapping that records the corresponding time window of R_t. For example, $W(R_t) = w_{(j+1) \to j'}$ where R_t is the recurrent time between j and j'. It is marked that the segmentation and the loss function can be updated based on the results in the last step. After applying a big loop cycling through C^* from 0 to M, the total time complexity now becomes $O(M^2)$. As a result, an optimal window set is returned, so the change points locations are estimated by their boundaries. The multiple change points searching algorithm is described in **Algorithm 4**.

Algorithm 4

Input: unmarked recurrence time $\{R_t\}_t$ and a threshold C^*

Loop: cycle R_t through order statistics $R_{(1)}$, $R_{(2)}$, ..., $R_{(M+1)}$

1. Initial an empty set $\mathbb{W}$ recording the high-intensity time windows
2. Consider 4 "if" conditions and obtain a new window w,
   a. If neither R_{t-1} or R_{t+1} is marked:

      $w = W(R_t)$

   b. If R_{t-1} is marked but R_{t+1} is not:

      merge $W(R_{t-1})$ and $W(R_t)$ into one window,

      $w = \{W(R_{t-1}) \bigcup W(R_t)\}$

   c. If R_{i-1} is not marked but R_{t+1} is:

merge $W(R_t)$ and $W(R_{t+1})$ into one window,

$w = \{W(R_t) \bigcup W(R_{t+1})\}$

d. If both R_{t-1} and R_{t+1} are marked:

merge $W(R_{t-1})$, $W(R_t)$, and $W(R_{i+1})$ into one window,

$w = \{W(R_{t-1}) \bigcup W(R_t) \bigcup W(R_{i+1})\}$

3. Update the recorded window set $\mathbb{W}$ with w and mark R_t.

4. If window length $|w|$ is greater than C^*:

extract the boundaries of windows in $\mathbb{W}$ as $\tilde{\tau}_1, ..., \tilde{\tau}_{\tilde{k}}$

update loss function $L(\tilde{\tau}_1, ..., \tilde{\tau}_{\tilde{k}})$

Output: optima boundaries $\hat{\tau}_1, ..., \hat{\tau}_{\hat{k}}$

4.3. MCP for multivariate time series

A large part of change point detection literature deals with continuous observation. In this section, we firstly proposed an encoding approach to categorize continuous time series into multiple Bernoulli sequences, and then analyze change points embedded within the multivariate process. The idea of categorizing real-value observations aims to extract more relevant information and filter out noise. It is claimed that the proposed approach is robust to encode any underlying distributions and is easily generalized to either categorical or continuous observations.

4.3.1. Encoding continuous time series. In the analysis of single stock returns, the authors in **[59]** utilized a pair of thresholds to mark absolutely large stock returns as 1 and 0 otherwise, then revealed the volatility pattern behind the resultant 0-1 sequence. The encoding process is written as

$$E_t = \begin{cases} 1 & X_t \leq \alpha\text{-}quantile,\ X_t \geq \beta\text{-}quantile \\ 0 & Otherwise \end{cases} \tag{4.2}$$

where $\{E_t\}_t$ is an excursion sequence by marking the stock returns. Later, authors in **[119]** proposed an encoding method to explore the local dependence of observations when $X_t \in \mathbb{R}^p$. Following up the

idea, we partition $\mathbb{R}^p$ space into V disjoint subarea, denoted as $B^{(v)}$ for $v = 1, 2, ..., V$, and transform the continuous observations $\{X_t\}_{t=1}^T$ into V Bernoulli sequences or a V-dimensional multinominal process $\{(E_t^{(1)}, E_t^{(2)}, ..., E_t^{(V)})\}_{t=1}^T$, such that

$$E_t^{(j)} = \begin{cases} 1 & X_t \in B^{(j)} \\ 0 & Otherwise \end{cases} \tag{4.3}$$

Here, subarea $B^{(j)}$ plays an important role to reserve the change-point pattern into a Bernoulli process. Denote the Bernoulli parameter in the i-th segments of $\{E_t^{(j)}\}$ as $p_i^{(j)}$. So,

$$p_i^{(j)} = \int_{B^{(j)}} dF_i \tag{4.4}$$

where F_i corresponds to the CDF of $\{X_t\}_t$ in the i-th time segments. Consider two consecutive homogeneous time segments i and $i+1$. The change point detection becomes easier if $p_i^{(j)}$ is far apart from $p_{i+1}^{(j)}$, and vice versa. There is actually a tradeoff between the size and the total number of the subareas. Larger number of subareas with smaller size can discover the distributional difference more precisely but with sacrifice of the power of statistics due to the reduced sample size. In the following subsections, we would assume that V is fixed and $B^{(j)}$ are determined. The implementation of the encoding procedure is discussed in Section 4.5.

4.3.2. Single Change Point Detection. Starting with a simplest setting, let's assume that there exists a single change point at τ^*. Specifically, $\{X_t\}_{t=1}^{\tau^*} \overset{iid}{\sim} F_1$ and $\{X_t\}_{t=\tau^*+1}^{N} \overset{iid}{\sim} F_2$ where F_1 and F_2 are two unknown CDFs. The goal is to test the homogeneity between the two sample sets. Following the encoding procedure above, we obtain a multinomial process $\{(E_t^{(1)}, E_t^{(2)}, ..., E_t^{(V)})'\}_{t=1}^N$ where $\{E_t^{(j)}\}_{t=1}^{\tau^*} \sim Bern(p_{1,\tau^*}^{(j)})$ and $\{E_t^{(j)}\}_{t=\tau^*+1}^{N} \sim Bern(p_{2,\tau^*}^{(j)})$.

Robbins et al. [**97**] extent the multivariate CUSUM statistics with uncorrelated components to the multinomial settings and derived its asymptotic distributions under the null hypothesis. The estimators of Bernoulli parameters at a hypothesized time location τ is defined by

$$\hat{p}_{1,\tau}^{(j)} = \sum_{t=1}^{\tau} \mathbb{1}\{E_t^{(j)} = 1\}/\tau \tag{4.5}$$

and

$$\hat{p}_{2,\tau}^{(j)} = \sum_{t=\tau+1}^{N} \mathbb{1}\{E_t^{(j)} = 1\}/(N-\tau) \tag{4.6}$$

for $j = 1, 2, ..., V$. Then, a chi-square statistic proposed by [**97**] is written as,

$$\chi_\tau^2 = \sum_{j=1}^{V} \frac{(\sum_{t=1}^{\tau} \mathbb{1}\{E_t^{(j)} = 1\} - \hat{p}_{1,\tau}^{(j)})^2}{\hat{p}_{1,\tau}^{(j)}} + \frac{(\sum_{t=\tau+1}^{N} \mathbb{1}\{E_t^{(j)} = 1\} - \hat{p}_{2,\tau}^{(j)})^2}{\hat{p}_{2,\tau}^{(j)}}$$

Moreover, if there exists no change point under the null hypothesis, the maximally selected chi-square statistics $\chi_{\hat{\tau}}^2$ converges to a Brownian motion asymptotically.

4.3.3. Multiple Change Points Detection. Now, we consider multiple change point detection when number of change point k is known. Suppose the change point locations are $0 = \tau_0^* < \tau_1^* < ... < \tau_k^* < \tau_{k+1}^* = N$. Specifically, $\{X_t\}_{t=\tau_i^*}^{\tau_{i+1}^*} \overset{iid}{\sim} F_i$ for $i = 0, 1, ..., k$, and consecutive CDFs F_i and F_{i+1} are different. A naive method to search for $O(N^k)$ possible change point locations is computationally intractable. Bisection procedure as in [**90**,**118**], dynamic programming [**51**], or the one we proposed in **Algorithm 4** can work for the purpose. It is claimed that our searching algorithm is favorable in exploring the global optima, but it is designed only adapting to a single-dimensional Bernoulli-variable sequence.

A divide-and-concur approach is proposed as a remedy to the multivariate problem. Denote $\{E_t^{(j)}\}_{t=1}^N$ as the j-th Bernoulli process after encoding the observations via $B^{(j)}$, and $p_i^{(j)}$ as the true parameters of $E_t^{(j)}$ defined by (4.4). We firstly apply **Algorithm 4** to estimate the change point locations within $\{E_t^{(j)}\}$, for $j = 1, 2, ..., V$, respectively. Suppose the estimated change point locations in the j-th sequence is $0 = \hat{\tau}_0^{(j)} < \hat{\tau}_1^{(j)} < \hat{\tau}_2^{(j)} < ... < \hat{\tau}_{\hat{k}^{(j)}}^{(j)} < \hat{\tau}_{\hat{k}^{(j)}+1}^{(j)} = N$. Note that the number of change points $\hat{k}^{(j)}$ does not necessarily equals k. It should depend on the way that we encode the observations and the choice of penalty coefficient in (4.1). So, the observations are partitioned into $\hat{k}^{(j)} + 1$ segments and within-segment points are sharing the same estimator of parameter. After that, a vector of length N is generated to record the estimated Bernoulli parameter, denoted as $\{\hat{r}_t^{(j)}\}_{t=1}^N$. Let $\hat{p}_i^{(j)}$ be the estimated parameter when t is between $\hat{\tau}_{i-1}^{(j)}$ and

$\hat{\tau}_i^{(j)}$, so

$$\hat{p}_i^{(j)} = \frac{\sum_{t=\hat{\tau}_i^{(j)}+1}^{\hat{\tau}_{i+1}^{(j)}} \mathbb{1}\{E_t^{(j)} = 1\}}{\hat{\tau}_{i+1}^{(j)} - \hat{\tau}_i^{(j)}}$$

for $i = 0, 1, ..., \hat{k}^{(j)}$ and $j = 1, 2, ..., V$. Thus, there are $\hat{\tau}_i^{(j)} - \hat{\tau}_{i-1}^{(j)}$ duplicates of $\hat{p}_i^{(j)}$ in $\{\hat{r}_t^{(j)}\}_{t=1}^N$, and $\hat{r}_t^{(j)} = \hat{p}_i^{(j)}$, for $t \in (\hat{\tau}_{i-1}^{(j)}, \hat{\tau}_i^{(j)}]$. Repeating the above procedure through the V sequences, we can eventually obtain a sequence of V-dimensional estimated parameters, denoted as $\{\hat{r}_t\}_t = \{(\hat{r}_t^{(1)}, \hat{r}_t^{(2)}, ..., \hat{r}_t^{(V)})'\}_t$.

Generated by marking samples from subarea $B^{(j)}$, the Bernoulli-variable sequence $E_t^{(j)}$ partially reserves the distributional changes from the raw observations. Indeed, the switching pattern of Bernoulli-parameter recorded in $\hat{r}_t^{(j)}$'s is relevant to the distributional changes, while some $\hat{r}_t^{(j)}$'s or at least some subsequences may work as irrelevant noise, especially, when $\int_{B^{(j)}} dF_i \cong \int_{B^{(j)}} dF_{i+1}$. An aggregation statistic is present to combine all pieces of information from $j = 1, 2, ..., V$, and weight each $\{E_t^{(j)}\}_{t=1}^N$ according to its degree of relevance. In this section, we will treat every sequence equally for theoretical purpose. The weighting procedure will be described in Section 4.5.

Different from the CUSUM statistics, we consider the within-group variance in $\{\hat{r}_t\}_t$. Given k hypothesized change point locations $\tau_1, \tau_2, ..., \tau_k$, the statistic is written as,

$$\hat{G}(\tau_1, \tau_2, ..., \tau_k) := \sum_{i=0}^{k} \sum_{t=\tau_i+1}^{\tau_{i+1}} \frac{||\hat{r}_t - \bar{r}_i||^2}{\tau_{i+1} - \tau_i} \tag{4.7}$$

where $\bar{r}_i = \sum_{t=\tau_i+1}^{\tau_{i+1}} \hat{r}_t/(\tau_{i+1} - \tau_i)$ for $i = 0, 1, ..., k$. Change point locations are then estimated as the ones that minimize the within-group variance, so

$$\hat{\tau}_1, \hat{\tau}_2, ..., \hat{\tau}_k = \underset{\tau_1, \tau_2, ..., \tau_k}{\operatorname{argmin}} \hat{G}(\tau_1, \tau_2, ..., \tau_k) \tag{4.8}$$

It is shown in the next section that consistency holds for the statistic. Moreover, it is cheap in the computation when $k > 1$. A hierarchical clustering algorithm with $k + 1$ clusters obtained can be implemented to search for multiple change point locations.

Stack the estimated parameters $\{\hat{r}_t\}_{t=1}^N$ in a $N \times V$design matrix denoted as $\mathcal{M}$, in other words,

$$\mathcal{M}_{N\times V} = [\mathcal{M}(t, j)]_{t,j} = [\hat{r}_t^{(j)}]_{t,j} \; for \; t = 1, ..., N; \; j = 1, ..., V$$

A time-order-kept agglomerate hierarchical clustering algorithm is applied upon $\mathcal{M}$ to cluster time locations (rows) with comparable V-dimensional covariables. We modify the classical hierarchical clustering algorithm in the sense that only consecutive time points or groups are agglomerated at each iteration, so the original time order is kept. A Wald's type of linkage is applied for the purpose of minimizing the within-group variance. As a result, $k+1$ consecutive time point clusters get returned, so k change point locations can be detected accordingly.

4.3.4. Consistency. We present the consistency of the estimated change point locations obtained from our proposed procedure. It shows that if some of the likelihood-based estimators of a single Bernoulli sequence are consistent, then the estimators derived by the aggregation statistic in (4.7) can also converge the true change point locations. We firstly demonstrate the consistency property in the case of a single change point and then do the same for the multiple change points setting.

Suppose the true change point location is τ^*, so $\{E_t^{(j)}\}_{t=1}^{\tau^*} \sim Bern(p_{1,\tau^*}^{(j)})$ and $\{E_t^{(j)}\}_{t=\tau^*+1}^{N} \sim Bern(p_{2,\tau^*}^{(j)})$. By definition,

$$\hat{r}_t^{(j)} = \begin{cases} \hat{p}_{1,\hat{\tau}^{(j)}}^{(j)} & , t \in [1, \hat{\tau}^{(j)}] \\ \hat{p}_{2,\hat{\tau}^{(j)}}^{(j)} & , t \in (\hat{\tau}^{(j)}, N] \end{cases} \tag{4.9}$$

where $\hat{\tau}^{(j)}$ is the estimated change point locations in $\{E_t^{(j)}\}_t$. To prove the consistency, people typically assume that the size of the two half time sequence cut by τ^* goes into infinity as $N \to \infty$, and the proportion of the first half converges to a constant $\gamma^* \in (0,1)$, a.k.a. $\tau^*/N \to \gamma^*$ $(N \to \infty)$. The within-group variance of (4.7) at any proportion cut γ can be written as,

$$\hat{G}(\gamma) = \frac{\sum_{t=1}^{\lfloor N\gamma \rfloor} ||\hat{r}_t - \bar{r}_1||^2}{\lfloor N\gamma \rfloor} + \frac{\sum_{t=\lfloor N\gamma \rfloor+1}^{N} ||\hat{r}_t - \bar{r}_2||^2}{N - \lfloor N\gamma \rfloor} \tag{4.10}$$

where $\bar{r}_1 = \frac{\sum_{t=1}^{\lfloor N\gamma \rfloor} \hat{r}_t}{\lfloor N\gamma \rfloor}$ and $\bar{r}_2 = \frac{\sum_{t=\lfloor N\gamma \rfloor+1}^{N} \hat{r}_t}{N - \lfloor N\gamma \rfloor}$. The estimated change point location now becomes

$$\hat{\tau} = \underset{\tau}{\operatorname{argmin}} \ \hat{G}(\tau/N) \tag{4.11}$$

in the finite-sample situation.

The theorem below shows that if some of the estimators $\hat{\tau}^{(j)}$ are consistent, then $\hat{\tau}$ consistently converges to τ^*. Assume that if $p^{(j)}_{1,\tau^*} \neq p^{(j)}_{2,\tau^*}$, then $\hat{\tau}^{(j)}/N$ converges to γ^* asymptotically; otherwise, $\hat{\tau}^{(j)}/N$ converges to 0 or 1 meaning no change point exists in $\{E^{(j)}_t\}_t$. We further assume that there exist at least one encoded Bernoulli sequence such that $p^{(j)}_{1,\tau^*} \neq p^{(j)}_{2,\tau^*}$. Without loss of the generalization, we suppose that a change point exists in $\{E^{(j)}_t\}_t$ for $j = 1, 2, ..., u$, and no change point exists for $j = (u+1), ..., V$ where $1 \leq u \leq V$.

THEOREM 4.3.1. *Under the assumption above and furthermore, for any $\epsilon > 0$,*

$$P(|\hat{\tau}/N - \gamma^*| < \epsilon) \to 1$$

as $N \to \infty$.

PROOF. Let $\hat{\gamma}^{(j)} = \hat{\tau}^{(j)}/N$. For any $\gamma \in (0, 1)$, rewrite

$$\hat{G}(\gamma) = \sum_{j=1}^{V} g(\hat{\gamma}^{(j)}, \gamma)(\hat{p}^{(j)}_{1,\hat{\tau}^{(j)}} - \hat{p}^{(j)}_{2,\hat{\tau}^{(j)}})^2$$

where

$$g(\hat{\gamma}^{(j)}, \gamma) = \frac{\hat{\gamma}^{(j)}}{\gamma}(1 - \frac{\hat{\gamma}^{(j)}}{\gamma})\mathbb{1}\{\gamma \geq \hat{\gamma}^{(j)}\} + \frac{1-\hat{\gamma}^{(j)}}{1-\gamma}(1 - \frac{1-\hat{\gamma}^{(j)}}{1-\gamma})\mathbb{1}\{\gamma < \hat{\gamma}^{(j)}\}$$

For $j = 1, ..., u$, with the consistency of $\hat{\tau}^{(j)}$, we can have

$$g(\hat{\gamma}^{(j)}, \gamma)(\hat{p}^{(j)}_{1,\hat{\tau}^{(j)}} - \hat{p}^{(j)}_{2,\hat{\tau}^{(j)}})^2 \to g(\gamma^*, \gamma)(p^{(j)}_{1,\tau^*} - p^{(j)}_{2,\tau^*})^2$$

While for $j = (u+1), ..., V$, it shows

$$g(\hat{\gamma}^{(j)}, \gamma)(\hat{p}^{(j)}_{1,\hat{\tau}^{(j)}} - \hat{p}^{(j)}_{2,\hat{\tau}^{(j)}})^2 \to 0$$

since $g(0, \gamma) = g(1, \gamma) = 0$. Therefore,

$$\hat{G}(\gamma) \to \sum_{j=1}^{u} g(\gamma^*, \gamma)(p^{(j)}_{1,\tau^*} - p^{(j)}_{2,\tau^*})^2 = g(\gamma^*, \gamma)||p^u_{1,\tau^*} - p^u_{2,\tau^*}||^2 = G(\gamma)$$

as $N \to \infty$, uniformly in γ. Let $\hat{\gamma} = \hat{\tau}/N$. It follows that

$$\hat{G}(\hat{\gamma}) < \hat{G}(\gamma^*)$$

Additionally, the minimum value of $g(\gamma^*, \gamma)$ is attained when $\gamma = \gamma^*$. For any $\epsilon > 0$, there exists $\eta > 0$, such that $G(\gamma) - G(\gamma^*) > \eta$, for all γ with $|\gamma - \gamma^*| \geq \epsilon$. Therefore,

$$\begin{aligned} P(|\hat{\gamma} - \gamma^*| > \epsilon) &\leq P(G(\hat{\gamma}) - G(\gamma^*) > \eta) \\ &= P(G(\hat{\gamma}) - \hat{G}(\hat{\gamma}) + \hat{G}(\hat{\gamma}) - G(\gamma^*) > \eta) \\ &\leq P(G(\hat{\gamma}) - \hat{G}(\hat{\gamma}) + \hat{G}(\gamma^*) - G(\gamma^*) > \eta) \\ &\leq P(|G(\hat{\gamma}) - \hat{G}(\hat{\gamma})| > \eta/2) + P(|\hat{G}(\gamma^*) - G(\gamma^*)| > \eta/2) \to 0 \end{aligned}$$

as N goes into infinity. □

The assumption ensures that $\hat{\tau}^{(j)}$ is a consistent estimator if a change point exists in $\{E_t^{(j)}\}_t$. So long as $u \geq 1$, the distributional discrepancy is captured by $\hat{\tau}$. For a Bernoulli-variable sequence, the change point analysis is relatively easier. One can test the existence of a single change point and plug in a consistent estimator if reject.

In the more general case of multiple change points, suppose that the observations are independent and distributed from $k+1$ distributions $\{F_i\}_{i=0}^k$. Let $\tau_i^*/N \to \gamma_i^*$ as $N \to \infty$, and $0 = \gamma_0^* < \gamma_1^* < ... < \gamma_k^* < \gamma_{k+1}^* = 1$. Since $\{E_t^{(j)}\}_t$ may only reserve partial information of the distributional discrepancy, the number of change points in $\{E_t^{(j)}\}_t$ could be smaller than k and varies for different j. By further assuming the existence of consistent estimator in the Bernoulli-variable sequence, the theorem below shows the consistency of the aggregation statistic when the number of change point $k > 1$.

THEOREM 4.3.2. *Define that* $C_i = \{j : \hat{\tau}_i^{(j)}/N \to \gamma_i^* \text{ as } N \to \infty\}$. *Suppose that* $|C_i| \geq 1$ *and* $\hat{\tau}_i^{(j)}$ *is none if* $j \in \{1, ..., V\}/C_i$. *Further assume that* $\zeta_i + \zeta_{i+1} < \tau_{i+1}^* - \tau_i^*$ *where* $\zeta_i = \max_{j \in C_i} |\hat{\tau}_i^{(j)} - \tau_i^*|$, *for* $i = 1, ..., k$. *Then, for any* $\epsilon > 0$,

$$P(\max_{i=1,...,k} |\hat{\tau}_i/N - \gamma_i^*| < \epsilon) \to 1$$

as $N \to \infty$.

PROOF. Denote $\tau_i^* = N\gamma_i^*$. Consider a group of change point locations that $\tilde{\tau}_i = \tau_i^* + \zeta_i$, for $i = 1, 2, ..., k$. By the definition of ζ_i, it follows that

$$\begin{aligned}\hat{G}(\tilde{\tau}_1, ..., \tilde{\tau}_k) &\leq \sum_{i=0}^{k-1} \sum_{j \in C_{i+1}} \frac{2|\hat{\tau}_i^{(j)} - \tau_i^*|}{\tau_{i+1}^* - \tau_i^*}(1 - \frac{2|\hat{\tau}_i^{(j)} - \tau_i^*|}{\tau_{i+1}^* - \tau_i^*})(\hat{p}_{i+1}^{(j)} - \hat{p}_i^{(j)})^2 \\ &\leq \sum_{i=0}^{k-1} |C_{i+1}| \frac{2\zeta_i/N}{\gamma_{i+1}^* - \gamma_i^*}(1 - \frac{2\zeta_i/N}{\gamma_{i+1}^* - \gamma_i^*})\end{aligned}$$

Then, denote $\Theta = \{(\tau_1, ..., \tau_k) : \max_{i=1,...,k} |\tau_i/N - \gamma_i^*| \leq \epsilon\}$. It shows that, for any $\epsilon > 0$,

$$(*) \qquad P(\max_{i=1,...,k} |\hat{\tau}_i/N - \gamma_i^*| \geq \epsilon) \leq P(G(\hat{\tau}_1, ..., \hat{\tau}_k) \leq \min_{(\tau_1,...,\tau_k) \in \Theta} G(\tau_1, ..., \tau_k))$$

Moreover, since ζ_i is consistent to 0, uniformly in i, by the assumption. So,

$$P(\max_{i=1,...,k} \zeta_i > \epsilon) \to 0$$

Therefore,

$$\begin{aligned}(*) \quad &\leq P(G(\hat{\tau}_1, ..., \hat{\tau}_k) \leq \min_{(\tau_1,...,\tau_k) \in \Theta} G(\tau_1, ..., \tau_k) | \max_i \zeta_i < \epsilon) P(\max_i \zeta_i < \epsilon) \\ &+ P(G(\hat{\tau}_1, ..., \hat{\tau}_k) \leq \min_{(\tau_1,...,\tau_k) \in \Theta} G(\tau_1, ..., \tau_k) | \max_i \zeta_i \geq \epsilon) P(\max_i \zeta_i \geq \epsilon) \\ &\leq P(G(\hat{\tau}_1, ..., \hat{\tau}_k) \leq G(\tilde{\tau}_1, ..., \tilde{\tau}_k)) + P(\max_i \zeta_i \geq \epsilon) \\ &\to P(G(\hat{\tau}_1, ..., \hat{\tau}_k) \leq 0) + 0 = 0\end{aligned}$$

as N goes into infinity. □

The theorem requires that the estimator is consistent if it exists, and there exists at least one estimator over the V Bernoulli sequences according to a true change point. Though the assumption is strong theoretically, we actually transform the change point detection for unknown underlying distributions into an analysis of a Bernoulli-variable sequence. The task becomes easier since an explicit likelihood function exists without further assumption of the distribution family. So, parametric approaches is involved and fitted under the framework. In practice, the searching algorithm advocated in **Algorithm 4** can be employed to detect the change points for each Bernoulli process.

Another advantages by applying the encoding-and-aggregation is that the error rate of change point detection can get controlled at the same time, which is present in the next section.

4.4. Stability Change Point Analysis

Finally, it comes to the most general case that the number of change points and their locations are unknown. The current approaches can be divided into two types: model selection and multi-stage testing. A searching algorithm is usually applied in conjunction with a model selection procedure to explore a possible number of change points starting from 1 to a large number. Multi-stage testing is conducted to test the null hypothesis of no additional change point needed by inserting another change point at each stage. However, none of the approaches provides a control for the discovery error of the change point detection. Indeed, the result is sensitive to the objective function of model selection or the significance level in multi-stage testing.

4.4.1. The Stability Detection Method. In this section, we borrow the idea of stability variable selection and propose a robust change point detection framework, named stability detection. Stability selection was firstly advocated by **[84]** to enhance the robustness and control the false discovery rate of variable selection. Half of the samples are randomly selected to feed into a base model at each iteration. The relevant variables are ultimately discovered based on the votes aggregated over all the variable selection results. Later on, authors in **[14]** extent the stability selection by sampling disjoint subsets of samples.

Similar to the strategy of subsampling, we select but not randomly a subset of samples in $B^{(j)}$ to generate a Bernoulli sequence, and then estimate the number and locations of change points within each of the Bernoulli sequences, respectively. By treating each time location as a variable, the stability selection framework can be employed here to aggregate the estimated change points over $B^{(j)}$ for $j = 1, 2, ..., V$. The successive change points are the ones with votes or selected probability above a pre-determined threshold. However, it could be unrealistic to break down the chronological order and treat each time point as separate from others. The locations near the true change points are considered as acceptable results.

Denote that $\mathcal{S}^{(j)}$ is a set of change points detected based on Bernoulli sequence $\{E_t^{(j)}\}_t$, and $p^{(j)}(t)$ is the probability that a time point t is selected, i.e. $p^{(j)}(t) = P(t \in \mathcal{S}^{(j)})$. After aggregating

all the change points sets $\mathcal{S}^{(j)}$ for $j = 1, 2, ..., V$, the probability of selection for time point t is defined by

$$\Pi^V(t) = \frac{\sum_{j=1}^{V} \mathbb{1}\{t \in \mathcal{S}^{(j)}\}}{V} \tag{4.12}$$

Then, we can obtain the output of stability change point detection by thresholding the quantity with a threshold $\pi \in (0, 1)$,

$$\mathcal{S}_\pi^V = \{t : \Pi^V(t) \geq \pi\} \tag{4.13}$$

4.4.2. Error Control. To evaluate the false discovery rate, we need to define the noisy time points that we should exclude from the admissive set. Especially, we believe that time points around the truth change point τ^* are admissive, but time points far away from τ^* should get excluded. Define $\mathcal{A} = \{t : t \in (\tau_i^* - w_\mathcal{A}, \tau_i^* + w_\mathcal{A})\ i = 1, 2, ...\}$ as a set of admissive change points including true change points and their close neighbors. Here, $w_\mathcal{A}$ is an admissive window width and it can change over i. Similarly, define $\mathcal{N} = \{t : t \notin (\tau_i^* - w_\mathcal{N}, \tau_i^* + w_\mathcal{N})\ i = 1, 2, ...\}$ as a set of noisy time points which is outside from the neighbors of the true change points where $w_\mathcal{N}$ is a noisy window width. Note the window width $w_\mathcal{A}$ can be narrower than $w_\mathcal{N}$, such that $\mathcal{A} \subset \mathcal{N}^C$. Suppose that the following assumptions hold for appropriate $w_\mathcal{A}$ and $w_\mathcal{N}$.

(1) $\sum_{j=1}^{V} p^{(j)}(t)/V$ are identical for any $t \in \mathcal{N}$,
(2) $\sum_{j=1}^{V} p^{(j)}(t)/V$ are identical for any $t \in \mathcal{A}$.

Here, we assume that the noisy time points have the same expected probability to be selected, and so do the admissive time points. Under these assumptions, the next theorem is shown to bound the expectation of false positive rate or false negative rate of change point detection, depending on the choice of threshold π.

THEOREM 4.4.1. *Under the assumption (1) and (2), denote* $p_\mathcal{N}^V = \sum_{j=1}^{V} p^{(j)}(t)/V$ *for* $t \in \mathcal{N}$ *and* $p_\mathcal{A}^V = \sum_{j=1}^{V} p^{(j)}(t)/V$ *for* $t \in \mathcal{A}$. *Let* $\pi \in (0, 1)$ *be the selection threshold.*

For any $0 < \xi < 1/p_\mathcal{N}^V - 1$, *if* $\pi > (1 + \xi)p_\mathcal{N}^V$ *we have*

$$\frac{E[|\mathcal{S}_\pi^V \cap \mathcal{N}|]}{|\mathcal{N}|} \leq \frac{1 - (1 + \xi)p_\mathcal{N}^V}{\pi - (1 + \xi)p_\mathcal{N}^V}\ exp(-\frac{\xi^2 V}{\xi + 2}p_\mathcal{N}^V) \tag{4.14}$$

For any $0 < \xi < 1$, *if* $\pi < (1-\xi)p_{\mathcal{A}}^V$ *we have*

$$\frac{E[|(\mathcal{S}_\pi^V)^C \cap \mathcal{A}|]}{|\mathcal{A}|} \leq \frac{(1-\xi)p_{\mathcal{A}}^V}{(1-\xi)p_{\mathcal{A}}^V - \pi}\ exp(-\frac{\xi^2 V}{\xi+2}p_{\mathcal{A}}^V) \tag{4.15}$$

PROOF. For any $0 < \xi < V/\sum_{j=1}^V p^{(j)}(t) - 1$, denote $\pi_{\mathcal{N}} = (1+\xi)\sum_{j=1}^V p^{(j)}(t)/V$, so that $\pi_{\mathcal{N}} \in (0,1)$.

It is easy to show that $\Pi^V(t) \leq (1-\pi_{\mathcal{N}})\mathbb{1}\{\Pi^V(t) \geq \pi_{\mathcal{N}}\} + \pi_{\mathcal{N}}$ for a fix $t \in \{1,2,...,N\}$. Thus,

$$\begin{aligned}
P(\Pi^V(t) \geq \pi) &\leq P((1-\pi_{\mathcal{N}})\mathbb{1}\{\Pi^V(t) \geq \pi_{\mathcal{N}}\} + \pi_{\mathcal{N}} \geq \pi) \\
&= P(\mathbb{1}\{\Pi^V(t) \geq \pi_{\mathcal{N}}\} \geq \frac{\pi - \pi_{\mathcal{N}}}{1-\pi_{\mathcal{N}}}) \\
&\leq \frac{1-\pi_{\mathcal{N}}}{\pi - \pi_{\mathcal{N}}} P(\Pi^V(t) \geq \pi_{\mathcal{N}}) \\
&= \frac{1-\pi_{\mathcal{N}}}{\pi - \pi_{\mathcal{N}}} P(\sum_{j=1}^V \mathbb{1}\{t \in \mathcal{S}^{(j)}\} \geq (1+\xi)\sum_{j=1}^V p^{(j)}(t))
\end{aligned}$$

The last inequality holds based on Markov's inequality and the condition that $\pi > \pi_{\mathcal{N}}$.

Moreover, $\mathbb{1}\{t \in \mathcal{S}^{(j)}\}$ are independent for $j = 1,2,...,V$. It holds because that we select disjoint samples to make up $\{E_t^{(j)}\}_t$ so for a fixed time t, its selection does not reply on the iteration index j. The resultant probability can be further bounded via Chernoff upper bound,

$$P(\sum_{j=1}^V \mathbb{1}\{t \in \mathcal{S}^{(j)}\} \geq (1+\xi)\sum_{j=1}^V p^{(j)}(t)) \leq exp(-\frac{\xi^2}{\xi+2}\sum_{j=1}^V p^{(j)}(t))$$

Hence,

$$\begin{aligned}
\frac{E[|\mathcal{S}_\pi^V \cap \mathcal{N}|]}{|\mathcal{N}|} &= \frac{\sum_{t\in\mathcal{N}} P(\Pi^V(t) \geq \pi)}{|\mathcal{N}|} \\
&\leq \sum_{t\in\mathcal{N}} \frac{1-\pi_{\mathcal{N}}}{\pi - \pi_{\mathcal{N}}}\ exp(-\frac{\xi^2}{\xi+2}\sum_{j=1}^V p^{(j)}(t))\ /\ |\mathcal{N}|
\end{aligned}$$

By further assuming identical $\sum_{j=1}^V p^{(j)}(t)$ for $t \in \mathcal{N}$, we can cancel $\mathcal{N}$ for both numerator and denominator, so the inequality (4.14) is obtained. Inequality (4.15) can be proved similarly via the lower bound of Chernoff's. □

As the bound of false positive rate or false negative rate decays with V, we are tempting to choose the number of iterations as large as possible. But it will significantly harm the power of

change point detection due to the reductive sample size of the recurrent times. In order to control the false discovery rate from both sides, one should increase the signal-selection rate $p_{\mathcal{A}}^V$ and decrease the noise-selection rate $p_{\mathcal{N}}^V$. It is ideal to set threshold in between, $(1+\xi)p_{\mathcal{N}}^V < \pi < (1-\xi)p_{\mathcal{A}}^V$. Recall the definition of the selection set $\mathcal{S}^{(j)} := \{\hat{\tau}_i^{(j)},\ i = 1, ..., \hat{k}^{(j)}\}$ where $\hat{\tau}_i^{(j)}$ is the i-th estimator of the j-th Bernoulli sequence. Then, $p_{\mathcal{A}}^V$ can be simplified by $\sum_{j=1}^V P(\hat{\tau}_i^{(j)} \in (\tau_i^* - w_{\mathcal{A}}, \tau_i^* + w_{\mathcal{A}}))/V$. Thus, with a fixed width of $w_{\mathcal{A}}$, a good estimator $\hat{\tau}_i^{(j)}$ is favored so that it is close the true change point location with a higher probability.

Another way to increase $p_{\mathcal{A}}^V$ is to sightly expand the selection set $\mathcal{S}^{(j)}$, that is to say, selecting the estimators and their neighbors. So, $\mathcal{S}^{(j)} = \{t : t \in neig(\hat{\tau}_i^{(j)}),\ i = 1, ..., \hat{k}^{(j)}\}$. A wider neighbor set $neig()$ is better in adsorbing admissive change points but endures the risk of involving noise. In the change point analysis of a sequence of Bernoulli variables, it is illustrated in Section 4.3 that a change point is estimated at the locations of 1's. A conservative way of expanding the selection set is to involve the estimator and the locations between the last and the next 1's.

4.5. Subsampling and Weighting Strategy

From an application perspective, there are still two real problems to be addressed. Firstly, how to generate a series of subarea $\{B^{(j)}\}_{j=1,...,V}$ in the encoding phase. Secondly, how to weight the contribution for each encoded Bernoulli sequence $\{E_t^{(j)}\}_t$ based on its degree of relevance. A follow-up question is that how to measure the goodness-of-fit for each $\{E_t^{(j)}\}_t$ and weighting their contributions accordingly. In this section, we resolve both problems via a subsampling weighting technique.

To address the first one, a natural way is to apply clustering analysis to obtain V disjoint clusters as $\{B^{(j)}\}$. But it raises another problem related to the robustness of a different number of clusters and the second question becomes even hard due to the unbalanced cluster size. Model selection criterion in (4.1) can be used to measure the goodness-of-fit if the cluster sizes are balanced. To ensure robustness and efficiency, we attempt to generate a larger number of clusters but with fixed cluster size, so overlappings are present here. Our numerical experiments show that the method is not sensitive to the choice of V. It is advocated to choose a larger number $V = 50$ with a fixed subsampling proportion $M/N = 0.1$, so a sample is expected to be selected 5 times.

Denote $\mathbb{X} = [X_1, X_2, ..., X_N]^{'}$ as a $N \times p$ matrix recording the time series $\{X_t\}_{t=1}^{N}$ where $X_t \in \mathbb{R}^p$. The subsampling algorithm is described as follows. We firstly apply K-Means upon $\mathbb{X}$ to get V cluster centroids. Then, cycle through every centroid to search for its M nearest neighbors in $\mathbb{X}$. We mark the M samples as 1 and the other $N - M$ as 0 at each iteration, so V Bernoulli sequences get returned. If without confusion, let's denote the M marked samples in the j-th step as $B^{(j)}$.

Since the weight is inversely proportional to the model selection criterion values or loss in (4.1), one can consider a mapping function $\mathcal{F} : \mathbb{R} \mapsto \mathbb{R}$ to scale the quantity,

$$\mathcal{F}(x) = 1 - \frac{x - \min(x)}{\max(x) - \min(x)}$$

so, the weight $w^{(j)}$ measuring the importance of j-th Bernoulli sequence is defined by,

$$w^{(j)} = \frac{\mathcal{F}(L^{(j)})}{\sum_{j=1}^{V} \mathcal{F}(L^{(j)})} \tag{4.16}$$

where $L^{(j)} = L(\hat{\tau}_1^{(j)}, \hat{\tau}_2^{(j)}, ...)$ is the loss of the j-th sequence. Thus, a $N \times V$ weighted design matrix $\mathcal{M}^{weighted}$ is fed into the time-order-kept hierarchical clustering algorithm mentioned in Section 4.3.3,

$$\mathcal{M}^{weighted} = \mathcal{M}_{N \times V} \times diag(w^{(1)}, ..., w^{(V)}) \tag{4.17}$$

Another weighting technique is based on the iterative weighting algorithm proposed in **[119]**. In the simple case that only one change point exists within a Bernoulli process, it is hard or even impossible to detect the parameter change if the two Bernoulli parameters are too close. Indeed, one can qualify the goodness-of-fit via the difference between $p_{1,\tau^*}^{(j)}$ and $p_{2,\tau^*}^{(j)}$ or the estimated delta $|\hat{p}_{1,\hat{\tau}}^{(j)} - \hat{p}_{2,\hat{\tau}}^{(j)}|$ in practice. If we assume that the size of the two segments is equal, the estimated delta is then simplified by measuring the proportion of the two recovered segments in $B^{(j)}$. The more purity of $B^{(j)}$, the better $E_t^{(j)}$ can be fitted. It enlightens us to measure the Shannon entropy in $B^{(j)}$ as an approximation when $k > 1$.

Denote the weight of the j-th sequence at the current step as $w_c^{(j)}$ and the entropy of set $B^{(j)}$ at the current step as $H_c(B^{(j)})$. We can iteratively apply clustering algorithm upon the weighted matrix in (4.17) and update the the entropy $H_c(B^{(j)})$ based on the recovered segments in the current

step. So, the weight in the next step can be updated by

$$w_{c+1}^{(j)} = 0.5\ w_c^{(j)} + 0.5\ \frac{\mathcal{F}(H_c(B^{(j)}))}{\sum_{j=1}^{V} \mathcal{F}(H_c(B^{(j)}))} \tag{4.18}$$

iteratively until convergence. Here the 0.5 is set to smooth the learning curve and make the sum of weights equal 1.

4.6. Numerical Experiment

In this section, we simulate various univariate and multivariate distributions with a known and unknown number of change points to illustrate our model performance.

When k is known, the time-order-kept hierarchical clustering algorithm is implemented with the weighting techniques proposed in (4.16) and (4.18). To differentiate the two weighting techniques, we denote (4.16) as 'simple weighting' and (4.18) as 'iterative weighting'. We compare the performance of our approaches with other nonparametric methods: E-Divisive by [**83**], Kernel Multiple Change Point(KernelMCP) by [**100**], and MultiRank by [**1**]. For the fairness of the comparison, all the procedures are conducted with the number of change point k known. The results are reported in Section 4.6.1 and Section 4.6.2 for univariate and multivariate settings, respectively. When k is unknown, since it is hard to quantify the false discovery rate, only stability detection is applied in Section 4.6.3.

Our method was implemented with $V = 50$, cluster proportion $M/N = 0.1$, and $\phi(N) = 2$ (AIC). In the iterative weighting, we further set iteration number $R = 150$ and stop criteria when the weights do not change for 10 steps. E-Divisive was implemented via *ecp* package with the tuning parameter $\alpha = 1$, $R = 499$ which was advocated in the paper. KernelMCP was implemented by Python package named *Chapydette* using the default setting (Gaussian kernel with the Euclidean distance, bandwidth = 0.1, and $\alpha = 2$). For MultiRank, we implement R codes provided in the supplementary file of [**83**].

To quantify the performance of a change point detection result, we calculate the Adjusted Rand Index(ARI) [**71**] between the recovered segments and the true segments. Rand Index(RI) [**96**] was originally used to measure the similarity between two data clustering results. As a corrected version of RI, ARI was designed to adjust for the chance of grouping elements. An ARI value of

1 corresponds to a perfect result, while negative or 0 values imply that the recovered segments are different from the underlying segments.

4.6.1. Univariate Simulation. In this section, we simulate univariate distributions with different variance or tailedness. Three data segments are generated sequentially with distributions $\mathcal{N}(0,1)$, $\mathcal{G}$, $\mathcal{N}(0,1)$, respectively. For changes in variance, $\mathcal{G} \sim \mathcal{N}(0,\sigma^2)$; and for changes in tailedness, $\mathcal{G} \sim t_{df}(0,1)$. Unbalanced segments are generated with the sample size n, $2n$, n, respectively. The size n is also varied $n = 100, 200, 300$ while the proportion for the three segments are kept the same.

Given the number of change points, ARI values as recovery accuracy are compared for the proposed approaches, E-Divisive, KernelMCP, and RankMCP in Table 4.1 and Table 4.2. Results show that E-Divisive outperforms others in the setting of changes in variance. The overall performance is worse in the setting of changes in tailedness, but the iterative weighting approach performs slightly better than others. KernelMCP takes advantage if $\mathcal{G}$ is Gaussian distributed but fails otherwise. It shows that the kernel-type method is very sensitive to the choice of kernel. As a nonparametric approach designed for changes in mean, RankMCP consistently fails in both settings.

It is remarked that our approach is implemented under unfavorable conditions since we attempt to clustering univariate observations with large cluster numbers. Indeed, the encoding phase can be modified accordingly by applying quantile thresholds and marking extreme observations below or above the thresholds. More discussions about encoding a single-dimensional process are referred to [**120**].

TABLE 4.1. ARI values in univariate Gaussian setting

		univariate distribution with changes in variance				
n	σ	simple weighting	iterative weighting	E-Divisive	KernelMCP	RankMCP
100	1.5	0.4216 (0.1668)	**0.5308** (0.1883)	0.5122 (0.2118)	0.3146 (0.1888)	0.3341 (0.1012)
	2	0.6239 (0.1993)	0.6463 (0.1677)	**0.8214** (0.1999)	0.5248 (0.3274)	0.3253 (0.0832)
	4	0.8179 (0.1372)	0.7634 (0.1242)	**0.9724** (0.0382)	0.9510 (0.0784)	0.3281 (0.0782)
200	1.5	0.5852 (0.2208)	0.6808 (0.1911)	**0.6697** (0.2706)	0.4355 (0.2601)	0.3068 (0.1085)
	2	0.7788 (0.1381)	0.7653 (0.1367)	**0.9536** (0.0605)	0.8942 (0.1554)	0.3208 (0.0878)
	4	0.9184 (0.0425)	0.8821 (0.0850)	**0.9872** (0.0176)	0.9815 (0.0157)	0.3246 (0.0798)
300	1.5	0.7311 (0.1747)	0.7674 (0.1586)	**0.7905** (0.2393)	0.6048 (0.3100)	0.3429 (0.0933)
	2	0.8375 (0.1019)	0.8189 (0.1054)	**0.9758** (0.0274)	0.9610 (0.0338)	0.3449 (0.0861)
	4	0.9440 (0.0311)	0.9243 (0.0480)	**0.9935** (0.0078)	0.9889 (0.0088)	0.3449 (0.0861)

TABLE 4.2. ARI values in univariate student-t setting

		univariate distribution with changes in tailedness				
n	df	simple weighting	iterative weighting	E-Divisive	KernelMCP	RankMCP
100	1	0.5527 (0.2242)	0.6426 (0.1780)	**0.6880** (0.2533)	0.2764 (0.1630)	0.3297 (0.1110)
	2	0.3742 (0.1582)	**0.4930** (0.1980)	0.4542 (0.1851)	0.2930 (0.1488)	0.2954 (0.0944)
	5	0.3045 (0.1380)	**0.3910** (0.1316)	0.3767 (0.1139)	0.2564 (0.1380)	0.3194 (0.1247)
200	1	0.7695 (0.1585)	0.7762 (0.1407)	**0.8419** (0.2123)	0.3391 (0.2089)	0.3205 (0.0879)
	2	0.4332 (0.2181)	**0.6097** (0.1824)	0.5055 (0.2267)	0.2672 (0.1648)	0.3212 (0.0967)
	5	0.2887 (0.1254)	**0.3696** (0.1666)	0.3606 (0.1204)	0.2401 (0.1600)	0.2899 (0.1280)
300	1	0.8395 (0.0926)	0.8220 (0.1065)	**0.8927** (0.1709)	0.4675 (0.2891)	0.3325 (0.1010)
	2	0.5010 (0.2481)	**0.6605** (0.2129)	0.6547 (0.2564)	0.3026 (0.1965)	0.3288 (0.1191)
	5	0.3101 (0.1211)	**0.4263** (0.1713)	0.3484 (0.1237)	0.2655 (0.1494)	0.2890 (0.1166)

4.6.2. Multivariate Simulation. Following the generation step above, we simulate multivariate observations in this section. The observations are distributed from $\mathcal{N}_d(0, I)$, $\mathcal{N}_d(0, \Sigma)$, $\mathcal{N}_d(0, I)$, respectively. In the first part, we consider binormal distributions, in which $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ with different correlation ρ. The ARI values for E-Divisive and KernelMCP are compared in Table 4.3. Given a moderate ρ value, it shows that the weighting procedures have comparable ARI values and outperform E-Divisive and KernelMCP. When ρ is extremely large and the sample size is greater, the binormal distribution actually degrades to an univariate Gaussian, which explains why the ARIs of E-Divisive and KernelMCP come from behind at $\rho = 0.9$ and $n = 300$.

In the second part, we simulate observations with dimension $d = 3, 5, 10$. Since KernelMCP is not easily adoptive when the dimension is more than 2, we only compare the performance of simple weighting, iterative weighting, and E-Divisive. Two types of Σ are imposed for the generation. Σ_1 is set with diagonal elements 1 and off-diagonal elements ρ; Σ_2 is set with diagonal elements 1 and ± 1-off-diagonal elements ρ. Table 4.4 shows that the simple weighting is more favorable in identifying the change point locations in the case of Σ_1; in the more complicated case of Σ_2, the iterative weighting performs the best.

4.6.3. Simulation for Stability Detection. Stability detection is applied when the number of change points is unknown. Suppose we encode the continuous observations into V Bernoulli sequence components, so there are V change point sets obtained in total. Instead of weighting each voting set equally, we involve the simple weighting techniques and weight the votes according to the goodness-of-fit.

TABLE 4.3. ARI values in 2-dim Gaussian setting

		2-dim Gaussian with changes in correlation			
n	ρ	simple weighting	iterative weighting	E-Divisive	KernelMCP
100	0.5	0.3673 (0.1593)	**0.4671** (0.1531)	0.3904 (0.1303)	0.2862 (0.1458)
	0.7	0.4726 (0.1935)	**0.5542** (0.1862)	0.4309 (0.1695)	0.2877 (0.1411)
	0.9	**0.6993** (0.1882)	0.6612 (0.1803)	0.5985 (0.2550)	0.3453 (0.2032)
200	0.5	0.3982 (0.1830)	**0.5386** (0.1950)	0.3960 (0.1592)	0.2776 (0.1533)
	0.7	**0.7186** (0.1790)	0.6535 (0.1912)	0.5025 (0.2340)	0.2944 (0.1494)
	0.9	0.8316 (0.1435)	0.7665 (0.1409)	**0.8614** (0.2108)	0.6924 (0.2816)
300	0.5	0.5171 (0.2343)	**0.5809** (0.2228)	0.3897 (0.1643)	0.2872 (0.1435)
	0.7	**0.8209** (0.1099)	0.7311 (0.1558)	0.7013 (0.2852)	0.3196 (0.1411)
	0.9	0.8753 (0.1188)	0.8160 (0.1233)	**0.9461** (0.1403)	0.9305 (0.1404)

TABLE 4.4. ARI values in d-dim Gaussian setting

		d-dim Gaussian with off-diagonal correlation 0.5			d-dim Gaussian with ±1-off-diagonal correlation 0.5		
n	d	simple weighting	iterative weighting	E-Divisive	simple weighting	iterative weighting	E-Divisive
100	3	0.4074 (0.1965)	**0.5520** (0.1855)	0.4505 (0.1772)	0.3805 (0.1565)	**0.5276** (0.2014)	0.4367 (0.1673)
	5	0.5660 (0.2095)	**0.6364** (0.1760)	0.4704 (0.2018)	0.4056 (0.1904)	**0.5160** (0.1836)	0.4222 (0.1581)
	10	**0.7826** (0.1595)	0.7132 (0.1785)	0.5777 (0.2563)	0.3919 (0.1426)	**0.5362** (0.1913)	0.4306 (0.1641)
200	3	0.5056 (0.2296)	**0.6500** (0.2027)	0.4281 (0.1976)	0.5013 (0.2267)	**0.5899** (0.1859)	0.4041 (0.1802)
	5	**0.8078** (0.1312)	0.7706 (0.1649)	0.6051 (0.2646)	0.4594 (0.2090)	**0.5977** (0.2101)	0.4419 (0.1849)
	10	**0.8819**(0.0957)	0.8386 (0.1255)	0.7875 (0.2644)	0.4148 (0.1936)	**0.6229** (0.1983)	0.4459 (0.1905)
300	3	0.6150 (0.2422)	**0.7480** (0.1828)	0.5166 (0.2285)	0.6258 (0.2461)	**0.6319** (0.2108)	0.4780 (0.2199)
	5	**0.8649** (0.0796)	0.8322 (0.1089)	0.8007 (0.2601)	0.5690 (0.2563)	**0.6836** (0.2066)	0.5429 (0.2343)
	10	0.8973 (0.0744)	0.8679 (0.1059)	**0.9170** (0.1787)	0.5586 (0.2469)	**0.7149** (0.1784)	0.4969 (0.2224)

Denote a binormal distribution $\mathcal{N}_2(0, \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix})$ as $\mathcal{G}_2$. In the first scenario, 3 binormal distributions are generated by $\mathcal{N}_2(0, I)$, $\mathcal{G}_2$, $\mathcal{N}_2(0, I)$ with sample size 300, 600, 300, respectively. In the change point analysis of each Bernoulli sequence, model selection criterion is applied with a different penalty coefficient $\phi(N)$. $\phi(N)$ is set from 2, a value corresponding to AIC, to $log(N)$, a value corresponding to BIC. We partition the time axis into disjoint time bins with the same length. The probability of selection is then calculated based on the summed votes falling in each bin. It shows that the results are not sensitive to the choice of penalty term. There are 6 or 7 curves always above the others regardless of the penalty coefficient. Their corresponding time bins are marked in Figure 4.1. The first and second bins are (320,360) and (360, 400) which are close to the first change point located at 300; the third and fourth time bins (280,320) and (880,920) cover the true change point locations. The probability of selection for all the time points are shown in Figure 4.2((A)

for AIC and (B) for BIC). The two big spikes indicate that the number of change points is 2. By further setting a threshold at 0.1, we can obtain two consecutive time windows containing the truth change point locations.

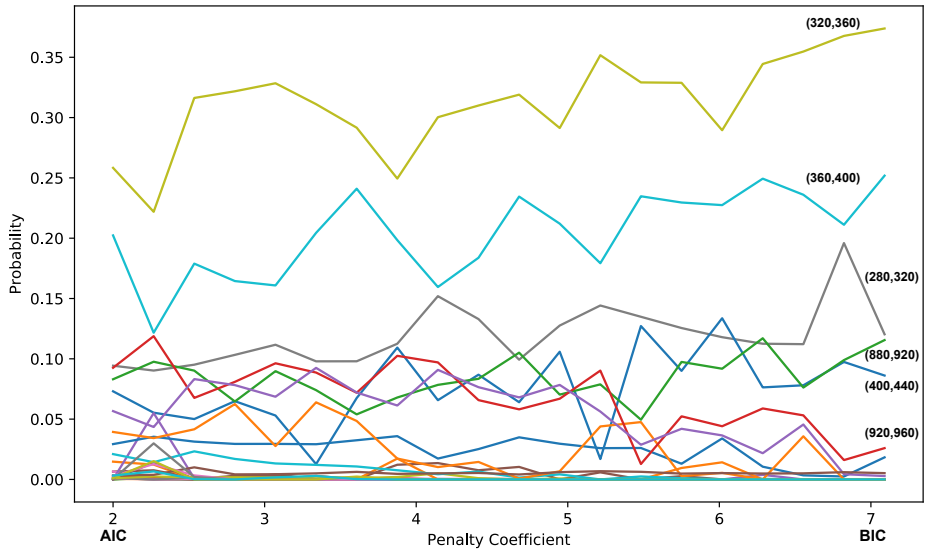

FIGURE 4.1. probability of selection with different penalty coefficient $\phi(N)$; different time bins are plotted in different curves

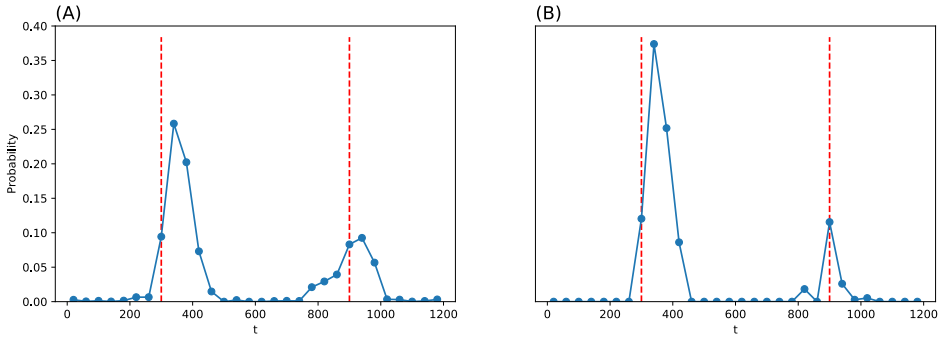

FIGURE 4.2. (A) probability of selection with $\phi(N) = 2$ as AIC; (B) probability of selection with $\phi(N) = log(N)$ as BIC. True change point locations are plotted in vertical lines

In the second scenario, we make the problem more complicated by generating 7 segments $\mathcal{N}_2(0, I)$, $\mathcal{G}_2$, $\mathcal{N}_2(0, I)$, $\mathcal{G}_2$, $\mathcal{N}_2(0, I)$, $\mathcal{G}_2$, $\mathcal{N}_2(0, I)$ with equal sample size n. BIC is applied for model selection of each Bernoulli sequence. There turns to be 6 obvious spikes after smoothing

the probability curve. Especially, when n increases to 500, the local maximas can almost perfectly detect the true change points.

Indeed, stability detection gives a new perspective to discover the number of change points and measure the confidence bound simultaneously. To estimate the change point locations, one can search for the top$k$ local maxima or go back to the hierarchical clustering procedure with an estimated change point number k. Based on our experiment, the two estimation results are very similar. We claim the consistency should also hold based on the result in Figure 4.3.

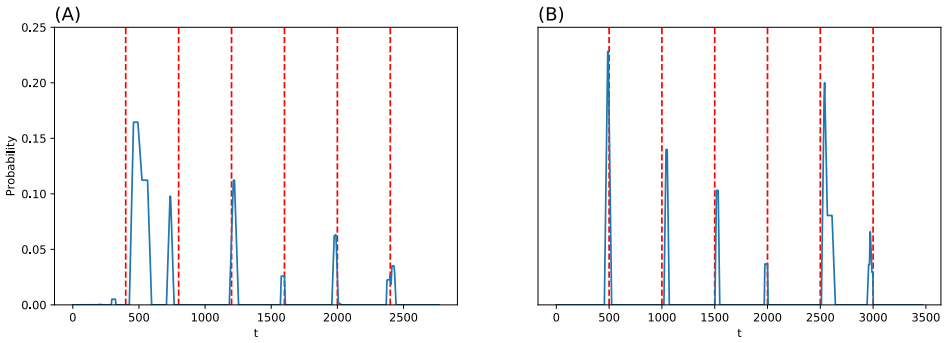

FIGURE 4.3. (A) probability of selection with $n = 400$; (B) probability of selection with $n = 500$

4.7. Real Data Application

4.7.1. Genome Data. CpG dinucleotide clusters or 'CpG islands' are genome subsequences with a relatively high number of CG dinucleotides (a cytosine followed by a guanine). They are observed close to transcription start sites [**101**] and play a crucial role in gene expression regulation and cell differentiation [**18**]. There were developed many computational tools for CpG island identification. A sliding window is typically employed to scan the genome sequence to figure out CpG islands based on some filtering criteria. However, the criteria are set with subjective choice (G+c proportion, observation versus expectation ratio, etc) and it has evolved over time. It commonly happens that different CpG island finders would provide various results.

In this section, we implement our change point detection approach in the categorical nucleotide sequence. We believe that the proposed algorithm is able to detect an abrupt change in C-G patterns, and the estimated change point locations may help researchers to identify potential CpG islands. A contig (accession number *NT_000874.1*) on human chromosome 19 was taken as an

example for CpG island searching. The dataset is available on the website of National Center for Biotechnology Information(NCBI).

Denote the genome sequence as $\{X_t\}_{t=1}^N$ with $X_t \in \{A, G, T, C\}$. In the encoding phase, a 0-1 sequence $\{E_t\}_t$ is generated such that $E_t = 1$ if $X_t = C \& X_{t+1} = G$ and $E_t = 0$ otherwise, for $t = 1, ..., N-1$. **Algorithm 4** is implemented to search for multiple change points in the Bernoulli sequence. Results from a CpG island searching software CpGIE [**125**] are shown as a benchmark for comparison. Criteria advocated by the authors are employed in the usage of CpGIE (length $\geq$ 500 bp, G + C content $\geq$ 50% and CpG O/E ratio $\geq$ 0.60). Note that our algorithm does not need any assumption or tuning parameter. The result in Figure 4.4 shows that there is a high proportion of overlapping segments between ours and CpGIE's. Our approach can also find extra genome subsequence with a higher number of C-Gs which are misspecified by CpGIE.

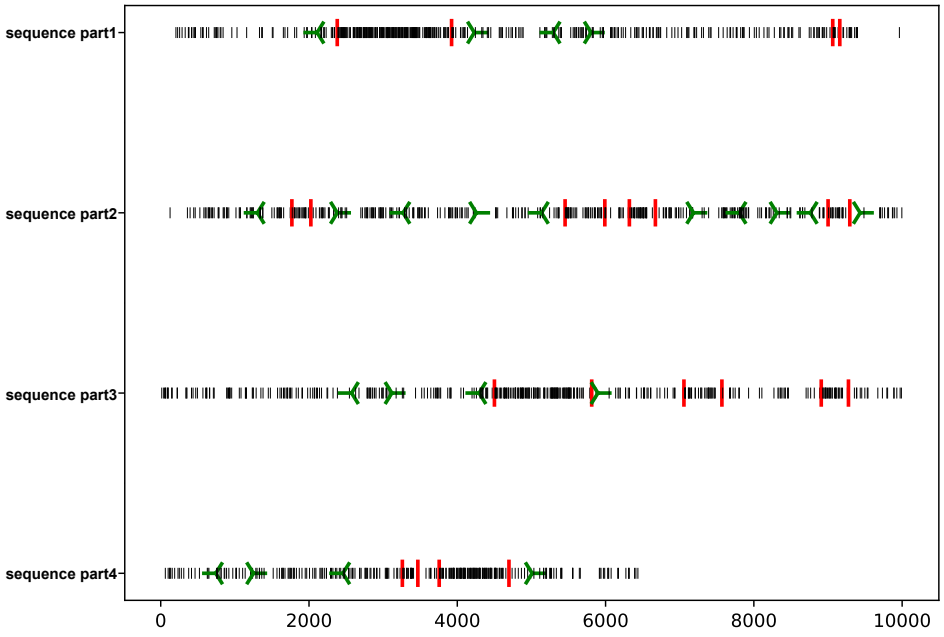

FIGURE 4.4. encoded DNA sequence-CG dinucleotides are marked in black; the CpG islands discovered by CpGIE are marked in green; the estimated change point locations are marked in red

4.7.2. Hurricane Data. It was widely recognized that the global temperature has risen due to anthropogenic factors, such as increased carbon dioxide emissions and other human activities. According to NOAA's 2020 global climate report, the annual temperature has increased globally

at an average rate of 0.14 degrees Fahrenheit per decade since 1880 and over twice that rate (0.32 degrees Fahrenheit) since 1981. It was argued by climatologists that the warmer sea surface leads to an increasing number of stronger tropical cyclones **[39, 106]**. However, it is claimed by **[74]** that the warmer sear surface increases only weak cyclones which are short and even hard to be detected. In this section, we studied the number of cyclones between 1851 and 2019. We are interested to detect potential change points embedded within the tropical cyclone history.

The dataset HURDAT2 recording the activities of cyclones in the Atlantic basin is available on the website of National Oceanic Center(NHC). NHC tracked the intensity of each tropical cyclone per 6 hours every day (at 0, 6, 12, and 18). The intensity level is categorized based on wind strength in knots, such as hurricane (intensity greater than 64 knots), tropical storm (intensity between 34 and 63 knots), tropical depression (intensity less than 34 knots). Different from **[97]** in categorizing cyclones, we summarize the number of time units that a category is observed, so the count is at most 4×31 in a month. The monthly frequency of tropical storm-level and higher-level cyclones is reported in Figure 4.5(A). If we apply 5 change points which is detected by the local maxima of stability detection in Figure 4.5(B), the time range is then partitioned based on the variation of storm count. Figure 4.5(A) shows that storms are more active in the 1880s, 1960s and after 2000. Though the global temperature trends to go upward since 1980, the storms are relatively sparse between 1980 and 2000. Thus, we tend to believe that no firm conclusion can be made yet that higher temperatures would increase the number of hurricanes.

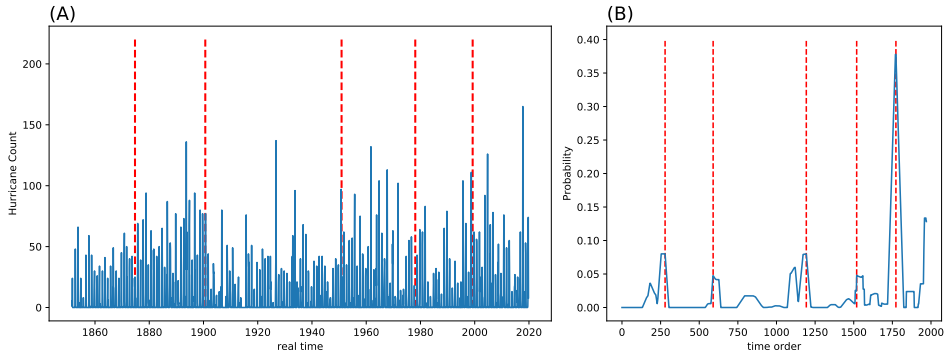

FIGURE 4.5. (A) monthly hurricane counts in Atlantic basin from year 1851 to 2019; estimated change points are plotted in vertical lines. (B) probability of selection for all the time points; local maximas are plotted in vertical lines

4.7.3. Financial Data. Lastly, the proposed approach is applied to detect the abrupt time-varying dependence within bivariate stock log returns. CTSH and IBM are chosen as representative of IT Consulting subcategories of S&P500 based on Global Industrial Classification Standard(GICS). The first and last hours in the transaction time are filtered out (so it is from 10am to 4pm), and the hourly price returns are calculated in the business days of the year 2006. A constant is added to the returns of CTSH for a better visualization in Figure 4.6(A), but the raw return series are analyzed. It was noted that the lagged correlation statistics are not significant based on the sample autocorrelation function of stock returns. Conditional heteroskedasticity can be studied by a more complicated time series model, like GARCH, but it is out of our concentration.

We encode the bivariate time series and apply stability detection techniques. Figure 4.6(B) shows that there exist 3 or 4 change points within the returns. The top3 change point locations with the highest probability are marked by vertical lines in Figure 4.6(A). It shows that the returns are partitioned into segments with different volatility levels. If we further look into the scatterplot between CTSH and IBM under different time partitions (left, middle, right segments) in Figure 4.7, both returns in the middle phase are relatively high, and their correlation is even stronger.

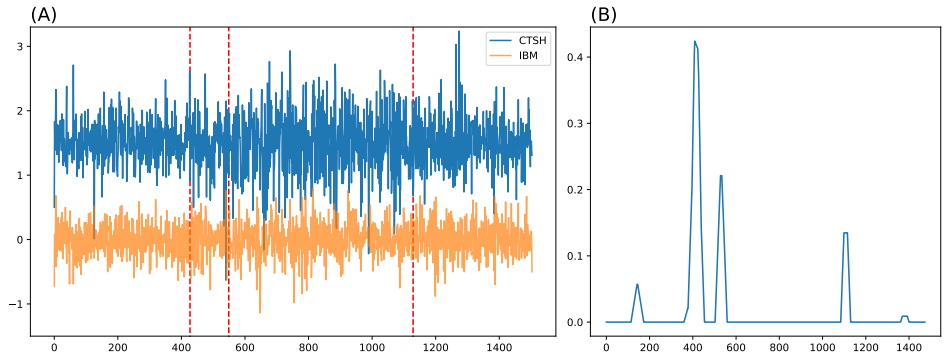

FIGURE 4.6. (A) hourly index returns of CTSH and IBM in 2006; top3 change points with the highest probability of selection are plotted in vertical lines. (B) probability of selection for all time points

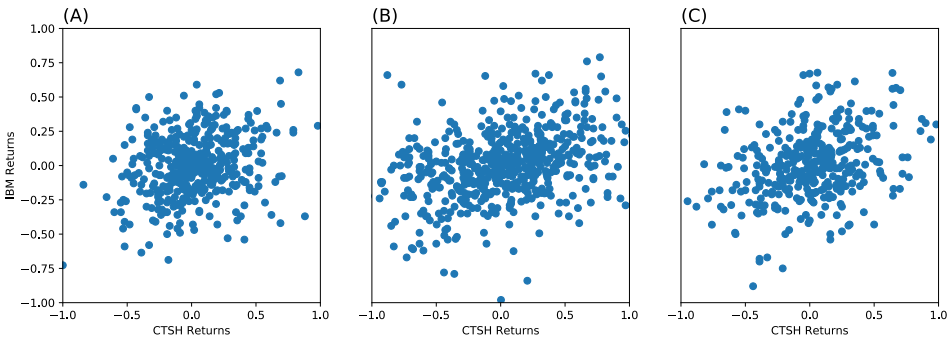

FIGURE 4.7. scatterplot of returns of CTSH versus IBM; (A) observations on the left segment; (B) observations on the middle segment; (C) observations on the right segment

4.8. Conclusion

In the chapter, we have established a framework to encode a sequence of continuous observations into several Bernoulli processes and proposed approaches for change point detection in univariate and multivariate settings with or without a known number of change points. Theoretical work shows that the proposed method can hold both asymptotic property and finite-sample error control. Numerical and real experiments show that the approach is able to detect any type of distributional changes and can be applied to categorical, ordinal, and continuous data. Furthermore, the computational expense is reasonable with time complexity at the most expensive part $O(VM^2)$ or $O(VN^2)$, and parallel programming is applicable to decrease the complexity to $O(N^2)$.

CHAPTER 5

Gait Identification and Individuals' Gait Dynamics

5.1. Introduction

It seems ordinary that we recognize our close friends and family members by their distinctive walking "styles", so-called signatures of gaits. With the complexity of neural and musculoskeletal systems in mind [**124**], the gait dynamics is not at all simple. Unlike high speed camera, our eyes surely miss all gait patterns of fine temporal scales. So, this ability of ours is not at all ordinary. Even though we human are anatomically identical by sharing the same structural skeleton and muscle constructs, and any gait dynamics must obey the universal biomechanics governing our musculoskeletal system, what make up individual signatures of gaits as biometric traits is still not yet well understood.

Majority of gait related research works is in the category of modeling-based gait analyses. The whole gait dynamics is never the focus. Any model based on only a few characteristics of gait dynamics typically not only is prone to make mistakes, but also difficult to apply to large number of healthy people. For instance, many works mainly aim for either Parkinson disease predictions or risk evaluations for the elderly [**2**, **44**, **73**, **113**, **123**]. Such top-down approaches are of limited used for surveillance since they don't embrace diverse spectra of gait characteristics. For instance, the fuzzy finite state machine [**5**] needs to incorporate expert opinions and judgements for specifying relevant states. Further transitions between states are governed by fuzzy logics [**129**].

Recently data collecting technologies have drastically evolved with recent advances in Micro-electromechanical systems (MEMS), such as low-cost, light-weight, easy-to-use inertial measurement units (IMU), such as accelerometer and gyroscope sensors [**110**]. These sensors nowadays are integrated with mobile devices, which enable us to collect gait time series data outside of gait laboratory, see figures of human wearing sensors in [**69**, **89**]. However, the capacity of precisely differentiating

many subjects' gait signatures and seeing a person's multiscale gait dynamics in full are not yet available in literature.

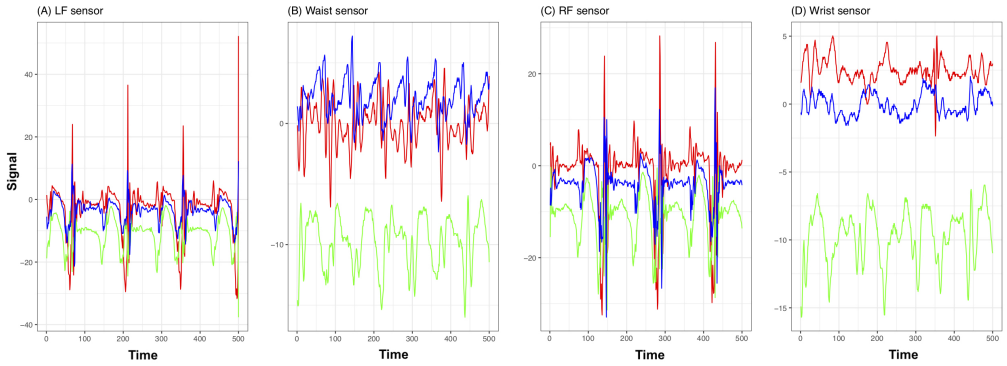

FIGURE 5.1. Gait time series data of subject #5 from four sensors: (A)Left foot; (B)Waist; (C)Right foot; (D)Wrist. $X-$dimension is Red color-coded, $Y-$dimension is Green and $Z-$dimension is Blue.

In this chapter, we develop computing and data-driven algorithms suitable for addressing two questions. 1) How to find and embrace large and diverse spectra of gait characteristics for identification purpose? 2) How to discover and recreate a person's gait dynamics in full?

The first theme of our data-driven developments is to compute and find many principle directions or vectors that implicitly capture many important aspects of above structural dependency-based heterogeneity across many people. We consider one manifestation of structural dependency through temporal patterns via a very simple and coarse coding scheme, called Principle System-State Analysis (PSSA). This dependency manifestation of coarse scale pattern is indeed very versatile for classifying among all subjects. We conjecture that this kind of dependency manifestation is potentially close to how our brains learn gait signatures.

As a complex system, and the intelligence of musculoskeletal system is embraced by its multiscale heterogeneity **[10]**. It is well known that any real "rhythmic" biomechanics is far from being completely deterministic and it naturally embraces stochastic structures across all rhythmic cycles as well **[33]**. Here it is worth emphasizing the evidently visible, but inexplicable stochasticity. Since this stochasticity is chiefly constrained by deterministic structures, it is not completely random. Therefore extracting stochastic structures of gait dynamics is at least as equally important as extracting the deterministic counterparts.

For explicitly extracting such multiscale deterministic and stochastic information contents, we turn to and focus on the system's fundamental structural dependency among all observed gait time series. It is clear that such structural dependency is lost to a great degree in the so-called resultant acceleration signal [**65**, **105**]:

$$A_{res}[t] = \sqrt{X^2[t] + Y^2[t] + Z^2[t]},$$

where X, Y, and Z indicate the 3-dimensional accelerations. This fact is evident through our motivating Lampel-Ziv complexity experiments, see details in the next section. Results from such experiments imply how to build a symbolic coding scheme to retain structural dependency of multiple time series.

Based on such motivation, our second theme of data-driven computing paradigm is developed as an unsupervised learning based multi-layer coding scheme, called Local-first and Global-second (L1G2) coding scheme. We apply L1G2 to build a 2D code sequence pertaining to the [Left-foot + Right-foot] system. We also develop a landmark partition algorithm to dissect such a 2D code sequence into rhythmic cycles consisting of visible biomechanical states. Such rhythmic patterns confirm that this subsystem indeed dictates the contents of a rhythmic cycle, its period and most importantly its evolving process. That is, the entire musculoskeletal system should function by coupling others subsystems upon [Left-foot + Right-foot] system.

To further show L1G2 effectively capturing multiscale gait dynamics, via graphic display, we simply stack all resultant color-coded rhythmic cycles aligned with the landmarks into a 3D cylinder. This rotatable 3D cylinder coherently reveals multiscale deterministic and stochastic rhythmic patterns as multiscale structural dependency across all rhythmic cycles. Such a 3D cylinder is the very foundation of further researches of gait-mimicking. It is also good for clinical diagnosis, and can be used as a "passtensor" for cybersecurity.

Two known gait time series databases are analyzed as the real data experiments. 1) MAREA database [**69**] with 4 sensors; 2) HuGaDB database [**27**] with 6 sensors. Both databases are created on healthy subject's gait when subjects wear with multiple sensors performing various activities on different kinds of surfaces. The sampling rate in MAREA is 128Hz, and is less in HuGaDB. That is, the time series in these databases contains patterns of centisecond (10 mini-second) scale.

We focus only on accelerometer in this study. It picks up accelerations of linear motions of body parts, where the sensors are fixed, upon $X-$, $Y-$ and $Z-$axial orientations. The 3-dim measurements are referencing to the coordinate system of human body: anterior-posterior (forward vs backward), superior-inferior (vertical up vs down along gravity direction) and left-right **[45]**. Our developments can easily accommodate gyroscope-based time series. In MAREA database, each subject wore a 3-axes Shimmer3 (Shimmer Research, Dublin, Ireland) accelerometer (+- 8g). In HUGA database, the information of accelerometer is described in **[27]**.

The chapter is organized as follows. In Section 5.2, we propose an encoding procedure to capture the deterministic structure of multiple accelerometer time series. In Section 5.3, we resolve the task of identifying gait signatures of different individuals. In Section 5.4 and Section 5.5, rhythmic cycles are detected and the gait authentication is done by constructing an individual's gait dynamics. A conclusion and several remarks are given in Section 5.6.

5.2. Revelations of Structural Dependency

To set the stage for our computational developments for exploring an individual's gait dynamics in full, we give an overview of the two contrasting manifestations of structural dependency contained in multi-dimensional gait time series. Firstly, from the 3-second recording of 12 dimensional time series of a MEARA subject's walking on flat ground in Figure 5.1, we see that each sensor's triplet directional time series exhibit diverse scales of relational patterns, which evolve within each visible cycle and recurrently appear across evident rhythmic cycles. Secondly, when we compare such patterns across different sensors, we also discover various scales of recurrent pattern-to-pattern correspondences. Such pattern-to-pattern correspondences are especially evident between panel (A) of Left-foot and panel (C) of Right-foot of Figure 5.1 across the evident cycles. Pattern-to-pattern correspondences between panel (B) of Waist and either one of Left-foot or Right-foot are also apparent, but not between panel (D) of Wrist with the rest of panels. These visible temporal-oriented relational patterns within cycles and complex pattern-to-pattern correspondences across cycles constitute multiscale structural dependency of gait dynamics contained in the 12 dimensional time series.

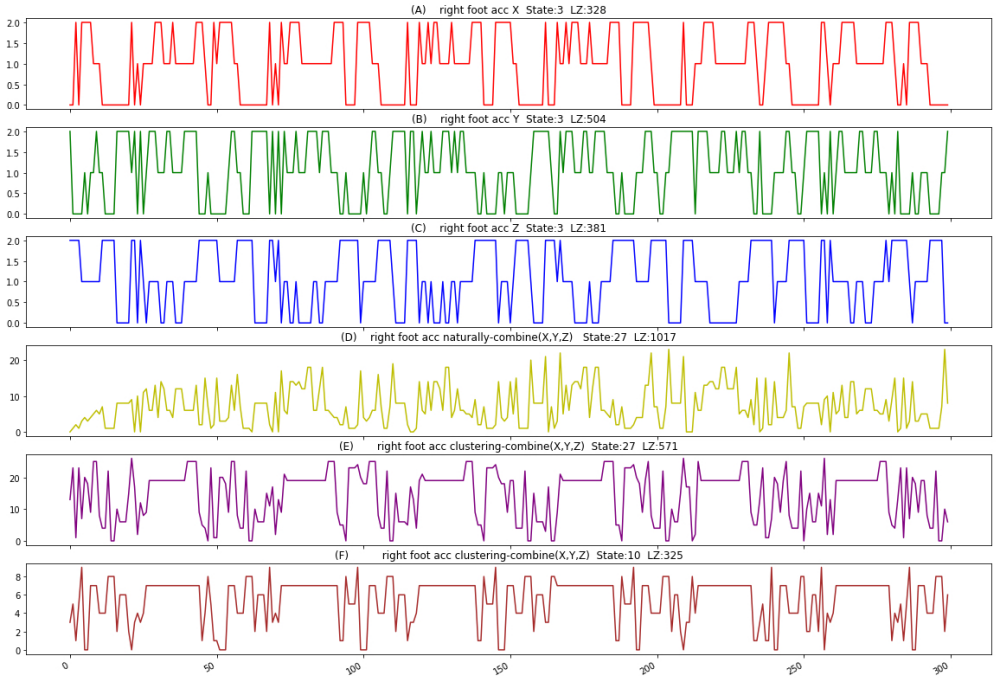

FIGURE 5.2. (A),(B),(C) 3-state code sequences for X-,Y-,Z- accelerometer time series based on 5.1, respectively. (D) is a natural combination of X,Y,Z, and the resultant sequence is coded by 27 ($3 \times 3 \times 3$) states. (E),(F) are sequences based on our clustering-based way of combination; (E) is coded by 27 states (clusters), the same number of states as (D), while its LZ complexity reduces by half. (F) a 10-states code sequence can show the rhythmic pattern clear enough, and its LZ complexity is as low as that of one-dim time series case.

In computational theory of computer science, the concept of Kolmogorov complexity is used in evaluating and exploring hidden structural patterns embraced within symbolic or digital time series. Its conceptual shortest universal computer program for regenerating a time series at hand is recognized to embrace all deterministic and stochastic structures. Unfortunately, Kolmogorov complexity cannot be calculated in general. We employ Lempel-Ziv complexity to give an approximate measure by only using 'copy' and 'insert' two operations. This complexity can be efficiently computed, see [**66**]. So, Lempel-Ziv is used in our complexity experiments. Before our complexity experiment, all the continuous time series must be categorized and transformed into a finite and discrete state sequence.

As shown in each panel of Figure 5.1, each triplet time series of (X, Y, Z) directions of an accelerometer reveal varying mechanism-specific gait dynamic patterns. Thus, we make use of this

data transformation requirement to naturally link the concept of structural dependency among time series to system-states of its dynamics. The idea of system-state can be seen as follows. We develop two tempo-sensitive digital-coding schemes upon gait time series along the temporal axis. The first scheme is to perform digital-coding upon each of the triplet directional time series individually and then couple the three digital code sequences into one sequence of vectors. The second scheme is to apply Hierarchical clustering algorithm based on Euclidean distance and Ward linkage on the temporal (column) axis of a data matrix representing the triplet time series with 3 rows. Based on the resultant clustering tree, a composition of clusters is chosen. A cluster of 3D vectors can be regarded as a symbolic code for a system state. Hence the specific mechanism pertaining to an accelerometer along the temporal axis is represented by a 1D symbolic code sequence. Color-coded examples of such code sequences are given in Figure 5.4. The computing cost of the first approach is much less than that of second approach. But, unlike the second approach, the first approach can only capture relatively coarse structural dependency.

We compare these two coding schemes in a set of Lampel-Ziv complexity experiments based on a short temporal segment $[0, 300]$. Results of such experiments are summarized in Figure 5.2, also see Figure C.2 and Figure C.3 in Appendix C for more details. The top three panels of Figure 5.2 respectively give the three directional symbolic code sequences. Each code sequence has 3 states and a value of Lampel-Ziv complexity. By coupling these three code sequences along the temporal axis, as shown in panel (D), the resultant code sequence with 27 state is seen nearly without any recognizable recurrent patterns. It has a complexity value 1017. In comparison, the second scheme with 27 clusters results into code sequence, as shown in the panel (E), that shows very evident recurrent and rhythmic patterns with a complexity value 571. Further, even if only 10 clusters are used to form the set of states, as seen in the bottom panel (F), the resultant code sequence is as evidently rhythmic as the one with 27 states in (E). With such rhythmic patterns in view, it is not surprising that its Lampel-Ziv complexity value is even lower. Evidently it captures the rhythmic dynamics well. Such experimental results confirm the presence of structural dependency among the three directional gait time series, and at the same time imply that the second coding scheme is way of extracting detailed dependency patterns in gait dynamics. Nonetheless, the first coding scheme has its own merit in identifying among many subjects as seen in the next section.

5.3. Principle System-State Algorithm (PSSA) for Identification

A simple way of having a glimpse of structural dependency among sensor-direction specific D dimensional gait time series is to transform and couple them into a D-dimensional digital vector trajectory. Here D is equal to 12 for 4 sensors used in MAREA database and 18 in for 6 sensors used in HuGaDB database. This digital trajectory is to exhibit rough manifestations of rhythmic cycles. So we manage to have a representation with relative small algorithmic complexity about the gait dynamics. This idea is simple and intuitive. Here we develop data-driven computations via a coarse coding scheme to realize this concept. By doing so, we get away from the necessity of man-made system-states and requirements of their transition rules. The simple computational results are capable of identifying many subjects simultaneously on a single platform. Thus we speculate such a simple algorithm is potentially what our brain actually performs in recognizing friends and relatives' gait signatures. To this aim, we propose an algorithm, called the Principle System-State Algorithm (PSSA), that attempts a single-layer coarse structural dependency among many individuals' D dimensional gait time series simultaneously.

5.3.1. The PSSA algorithm. For the purpose of identification, we expect to identify an individual by only glimpsing his/her short time of walking. Each individual's specific gait time series is subdivided into replicates of period in equal length l. we assume that in the test set, each unlabeled individual would have sample size exceeding l. The choice of l is supposed to be small while the signal is strong enough. Here we set $l = 1000$ time points, which lasts about 7.8 seconds with respect to the sampling rate being set at 128Hz. Consider that each individual at each time point has a D dimensional measurements (with the same unit m/sec^2): 3 directional (X-, Y-, Z-) accelerations from each of accelerometer sensors. We stack such D dimensional vectors together across all individuals' time points into a large data matrix with 12 rows. After that, the PSSA algorithm is applied which is described below.

Firstly, encode each sensor-direction specific 1-dim time series by using 3-digit alphabets.

$$S_d(t) = \begin{cases} 1 & X_d(t) \le \alpha \\ 2 & \alpha < X_d(t) \le \beta \\ 3 & X_d(t) > \beta \end{cases} \tag{5.1}$$

where $X_d(t)$ is the variable at time stamp t and $d = 1, 2, ..., D$ indicating dimension. So a D-dimensional digital system-state (vector), say $S(t) = (S_1(t), ..., S_D(t))'$, is formed at each time point t. The tuning parameter α and β ($\alpha < \beta$) are chosen based on the quantile of each 1-dim empirical distribution of pooled data across all involving subjects. Based on the consideration that the extreme values of each distribution played an important role in identifying different subjects. We choose $\alpha < 0.5 < \beta$ and α and β are closer to their extremes 0 and 1, respectively. As a result, the complexity of resultant digital code time series becomes smaller.

Secondly, collect all distinct system-states $S(.)$ and calculate their corresponding frequency f. There will be at most 3^D possibilities. Sort the distinct states with respect to frequency from the most frequent to the least $S^{(1)}(.), ..., S^{(N)}(.)$ with highest frequency $f^{(1)}$ to the lowest one $f^{(N)}$. Select a set of N^* states with top highest frequency as principle system-states (PSS).

Thirdly, cut the gait time series from the training set into short-temporal segments in length l, and convert each segment to a N^*-vector of proportion of PSS occurring within the period. That is to say, we extract N^* from each of the segment which represent the frequency of the appearance of the principle system states.

Finally, an $m \times N^*$ rectangle matrix Σ_{PSS} is built by stacking all involving proportion vectors along the row-axis, where m is the total number of segments, and N^* is the number of principle components. The entry (i, j) of Σ_{PSS} can be explained as the frequency of the j-th principle state found in the i-th segment. Apply hierarchical clustering analysis on row and column axes of Σ_{PSS}, respectively. Find the corresponding 'key' PSS for each individual such that the PSS can be used as a new feature (group) to exclusively identify the individual from others.

PSSA achieves a huge reduction on temporal dimensionality from $l = 1000$ to N^*. More importantly, such a N^*-dim vector is in the category of structural data, that is, each component can be treated as a feature variable. So any classic machine learning techniques can come in and work on the structured matrix Σ_{PSS}.

With a chosen pair of tuning parameter α and β ($\alpha < 0.5 < \beta$). the complexity digital coded D-dim time series can be seen via the curve of proportion of coverage on all involving trajectories as:

$$r(N^*) = \sum_{i=1}^{N^*} f^{(i)}/N,$$

The selection of N^* principle system-states $S^{(1)}(.), ..., S^{(N^*)}(.)$ can be also based on this curve.

5.3.2. PSSA on real databases. Two examples of coverage proportion curves with respect to N^* principle system-states are given Figure C.1 in Appendix C for MAREA database and HuGaDB database.

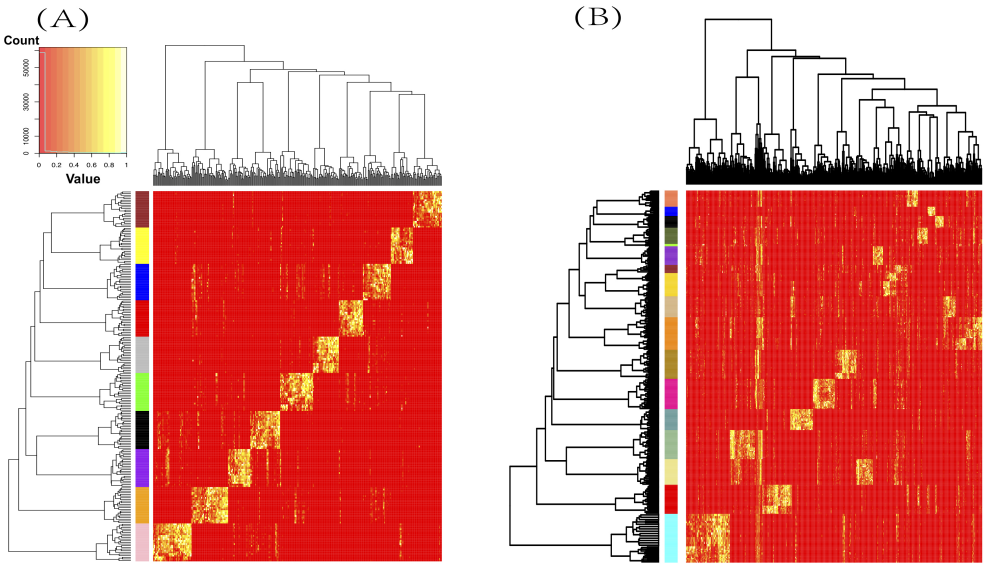

FIGURE 5.3. Identification via heatmap of Σ_{PSS}. Each row indicates a segment of gait time and rows from the same subject are labeled in the same color; each column indicates a selected PSS. (A) MAREA database: 10 subjects. The quantiles $\alpha = 0.3$ and $\beta = 0.7$. $N^*(= 300)$ principle system-states based on 9 dimensions of gait time series derived from three sensors fixed at Left foot and Right foot and wrist; (B)HuGaDB database: 17 subjects with 6 sensors tied to left and right thighs, shins and feet. The quantiles $\alpha = 0.1$ and $\beta = 0.9$. $N^*(= 500)$ principle system-states based on 18 dimensions of gait time series.

Both results in the training set are perfectly classified without any error among all 10 subjects' replicates in MAREA database, and 17 subjects' replicates in HuGaDB database, see Figure 5.3. By selecting one significant states block or cluster for each individual, a simple decision tree can achieve perfect classification result in the test set. That is to say, the principle states take the shape of feature selection, and they are the keys in Gait identification.

Here we make a remark on how to scale a big ensemble of individuals via PSSA. When the ensemble of individuals is big in size, the PSSA needs a strategy to scale down the computing loading. That is, if such an ensemble is taken as being homogeneous, then PSSA will need a

large collection of system-state vectors to cover enough complexity in identification task. Or the percentages α and β are chosen to be close their extremes. On the other hand, if heterogeneity is naturally present in any human ensemble, it implies the necessity of partitioning the whole ensemble into homogeneous sub-ensembles, and then PSSA is applied respectively. This is a typical divide-and-conquer strategy. For instance, the database in [**89**] consists of more than 700 individuals. It is sensible to divide the whole ensemble with respect to available demographic information.

In summary, our PSSA algorithm apparently is able to identify a set of system-states as signatures for each individual subject via relatively easy computations, and then perfectly classify among these subjects. Such visible signatures are indeed between-subject characteristics in nature. Since the computing behind such signatures is so simple, it is postulated why our brain can capture such signatures seemingly with easy after lengthy observations.

5.4. Authentication via Structural Dependency

Here if we agree that different sets of triplet time series from different sensors give rise to different aspects of gait dynamics pertaining to our musculoskeletal system, then to authentically recreate gait dynamics is equivalently to compute the multiscale structural dependency based on all available time series data.

Let the local scale refer to various body components of musculoskeletal system, such as Left-foot, Right-foot, Waist and Wrist. Each component contributes a fixed series of nearly deterministic biomechanical phases. Each biomechanical phase involves with a specific type of stochasticity: either in lengths or compositional contents. It is worth noting that such stochastic structures are somehow constrained by deterministic structures.

Let the global scale refer to how different components of musculoskeletal system couple and work out gait dynamics. Due to their dual symmetry, we particularly focus on how Left-foot relationally works with Right-foot via an evolving process. The [Left-foot + Right-foot] subsystem is rather distinct from their relations to Waist as the center of mass with the musculoskeletal system. That is, within the entire musculoskeletal system, the [Left-foot + Right-foot] system indeed functionally coordinates with different subsystems.

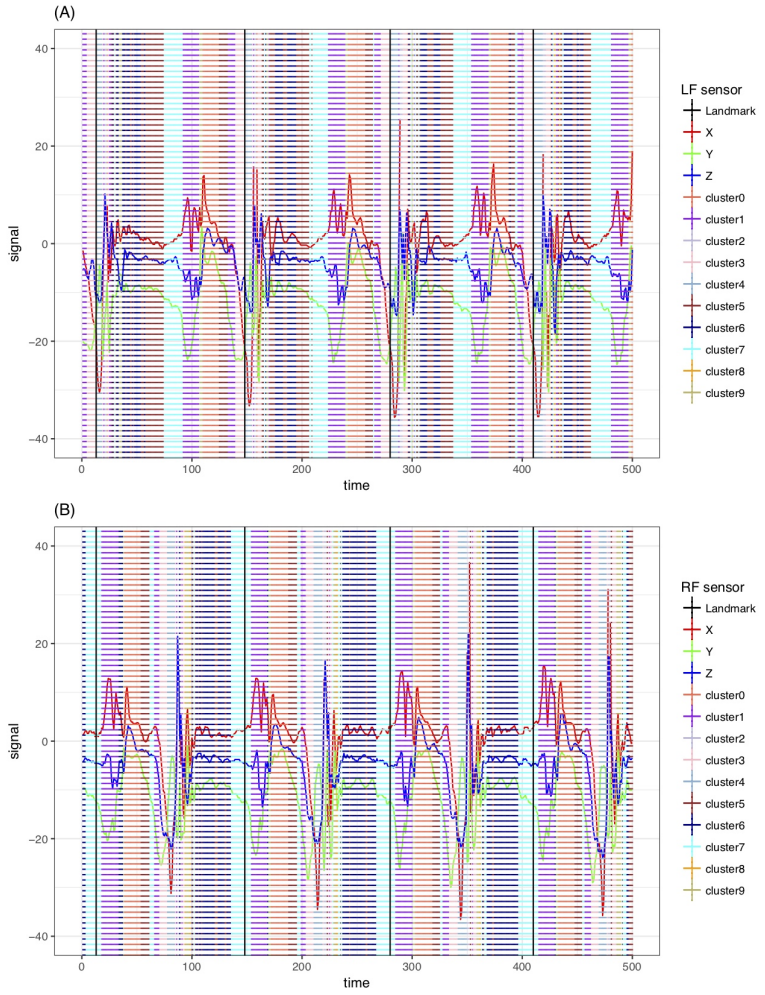

FIGURE 5.4. 3D time series superimposed with color coding on temporal period [1, 500]: (A) Left-foot sensor; (B) Right-foot sensor. Color coding of the 10 selected clusters are listed on the right hand side. The landmarks are calculated and marked with vertical black line.

5.4.1. L1G2 and landmark partition algorithms. We reiterate that Left-foot and Right-foot play dual roles, on one hand, and are comparable or even symmetric, on the other hand. Their two sets of triplet time series are highly associated. We denote the [Left-foot + Right-foot] as the L+R, for short. Thus, we will encode L+R system locally first, and then integrate L+R system with Waist or Wrist. That is, we make the L+R system a foundation to grow the integrated musculoskeletal system. For this integrative task, we develop a rather simple algorithm based "local-first and global-second (L1G2)" coding scheme in this section.

This L1G2 coding scheme is devised by first applying HC algorithm onto the stacked version of $X-$, $Y-$ and $Z-$ directional time series from the Left-foot and Right-foot sensors to generate a clustering tree. Upon this tree, we pick a 10-cluster composition to form a set of 10 code-words. Accordingly, Left-foot's triplet time series are transformed into a 1D symbolic code sequence, so is the Right-foot's. We then simply couples these two code sequences into a 2D L+R system-state trajectory. This choice of 10 code-words is supported by results of complexity evaluations in our Lampel-Ziv experiments in Figure 5.2.

Algorithm 5: Local-first & Global-second (L1G2) Coding

Input: $\{(X_L(t), Y_L(t), Z_L(t)),\ 1 \le t \le T\}$ from Left-foot sensor

$\{(X_R(t), Y_R(t), Z_R(t)),\ 1 \le t \le T\}$ from Right-foot sensor

1. Stack two time series and build a $3 \times 2T$ matrix,

$M_{L+R}[\cdot, 1:T] = \{(X_L(t), Y_L(t), Z_L(t)),\ 1 \le t \le T\}$

$M_{L+R}[\cdot, (T+1):2T] = \{(X_R(t), Y_R(t), Z_R(t)),\ 1 \le t \le T\}$

2. Apply HC on the temporal (column) axis of M_{L+R} to obtain H clusters, coded as $\{a_1, ..., a_H\}$, which represent local-system states.

3. Represent 3D time series $\{(X_L(t), Y_L(t), Z_L(t))\}$ and $\{(X_R(t), Y_R(t), Z_R(t))\}$ as 1D H-digital time sequence $\{S_L(t)\}$ and $\{S_R(t)\}$, respectively.

4. Couple the two local system-state time series of Left-foot and Right-foot in a 2D L+R system-state time series with 2D vector $S_{L+R}(t) = (S_L(t), S_R(t))'$, for $t = 1, 2, ..., T$.

5. Integrate encoded Waist and encoded L+R system by a 3D $(L+R)+W$ system-state time series with 3D vector $S_{(L+R)+W}(t) = (S_L(t), S_R(t), S_W(t))'$.

Output: $S_{L+R}(t)$ and $S_{(L+R)+W}(t)$.

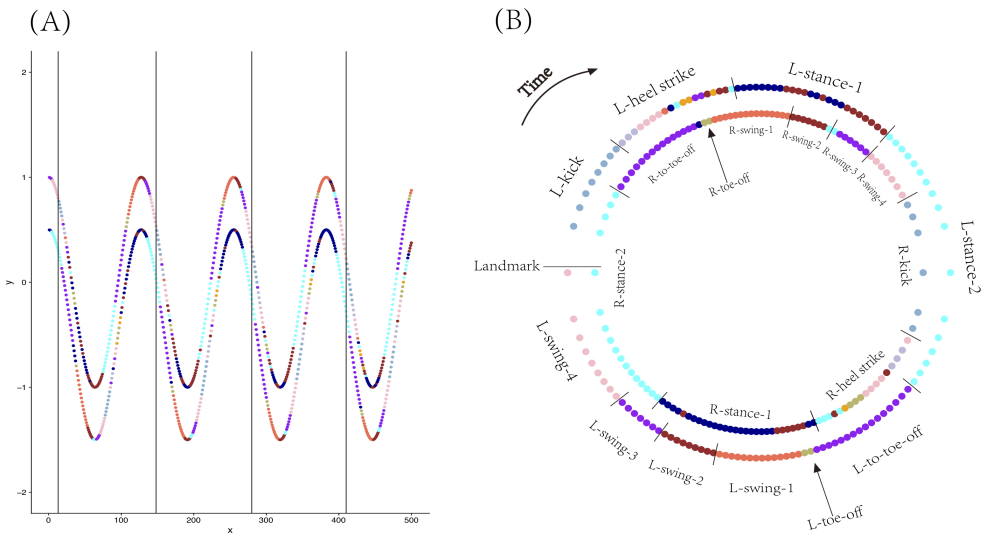

FIGURE 5.5. Color-coded rhythmical cycles in L+R system of subject #5 marked with serial biomechanical phases. (A) The coupled color coding time series on temporal period [1, 500] (Upper curve for Left-foot, Lower curve for the Right-foot. The landmarks are marked with vertical black lines; (B) Rhythmic cycle, the 3rd one in panel (A), is represented by two concentric rings (Outer ring for Left-foot, and inner right for Right-foot). The temporal coordinates go clockwise.

Next we develop a landmark algorithm to partition symbolic system-state trajectories into rhythmic cycles. The algorithm is described in **Algorithm 6**.

Algorithm 6: Landmark Partition

Denote a Run_i as a temporal segment that one specific state i consecutively repeats itself.

Input: the 2D L+R system-state time series $\{S_{L+R}(t)\}$

1. Calculate variance of the size of Run_i.

2. Calculate variance of the recurrence time of Run_i.

3. Choose the system-state i^* as a "landmark",

$i^* = argmin_i\{Var(\text{size of } Run_i) + Var(\text{recurrence time of } Run_i)\}$

4. Employ the landmark i^* to partition the entire system-state trajectory into pieces of rhythmic cycles.

Output: a series of rhythmic cycles.

Throughout our experimental explorations across many subjects, we found that rhythms in the L+R system are rather stable, while Waist and Wrist sensors' system-state are also rhythmic, but their stability are weak. Further computed landmarks are found to coincide with the beginning of a system-state in L+R system, which is defined by a code-word pertaining to either Left-foot or Right-foot sensors, see Figure 5.4. This uncertainty is likely due to some degrees of asymmetry between left foot and right foot.

5.4.2. Color coded rhythmic cycles. We apply the L1G2 algorithm onto the L+R system of subject #5 on temporal period $[1, 10000]$. The Local coding scheme is worked out on a stacked 3×20000 matrix. The 10 code-words are color-coded, so that the identified system-states of L+R system are visible and readable with biomechanical meanings, as shown in Figure 5.4.

Each colored code sequences of Left-foot and Right-foot sensors respectively achieves a dimension reduction: from 3 to 1. By coupling the two colored-codes sequences, as shown in panel (A) of Figure 5.5, L1G2 algorithm results cosine function like rhythm under L+R system. The symmetry on both feet are also explicit. We then apply the landmark computing algorithm on such a 2D coupled colored-code sequence on the temporal period $[1, 10000]$ to result 77 rhythmic cycles. The average period length and standard deviation as calculated as 127.56 ± 2.31.

To better visualize the progressing of system-state of L+R system via coupled colored-codes, a rhythmic cycle is specifically represented by two concentric circles: Outer one for Left-foot and inner one for Right-foot, starting from the marked landmark located at the 9 o'clock position, as shown in panel (B) of Figure 5.5. Biomechanical phases on both feet are annotated. Indeed the gait dynamics within a rhythmic cycle is evidently revealed with deterministic and stochastic structures as characterized as follows:

Deterministic structures:

A. The process of 2D coupling-phases as its state trajectory (with clockwise temporal coordinates) is nearly deterministic throughout all computed cycles:

Starting from "landmark" ⇒ (LF-Kick, RF-Stance2) ⇒ (LF-HeelStrike, RF-toToeOff) ⇒ (LF-HeelStrikeEnd, RF-ToeOff) ⇒ (LF-Stance1, RF-Swing1) ⇒ (LF-Stance1, RF-Swing2) ⇒ (LF-Stance1, RF-Swing3) ⇒ (LF-Stance2, RF-Swing4) ⇒ (LF-Stance2, RF-Kick) ⇒ (LF-ToeOff, RF-HealStrike) ⇒ (LF-ToeOff, RF-HeelStrikeEnd) ⇒ (LF-Swing1, RF-Stance1) ⇒(LF-Swing2, RF-Stance1) ⇒ (LF-Swing2, RF-Stance1) ⇒ (LF-Swing3, RF-Stance2) ⇒ (LF-Swing4, RF-Stance2) ⇒ End at next "landmark";

B. A Toe-off phase of one foot has to happen after the end of Heel-strike phase of the other foot;

C. The end of kick phase as the ending phase of swing process on one foot coincide with the beginning of "to-Toe-off" phase.

Stochastic structures:

A. Each 2D coupling-phase varies with lengths (seen through the 3D plot of rhythmic cycles from #3 to #70). This is the median-scale aspect of stochasticity within a rhythmic cycle;

B. The fine-scale stochasticity is seen in the phases of "heel-strike" of both left foot and right foot. The variations are far from being completely random;

C. There are some orders involving with a limited number of colored nodes. The large-scale of stochasticity is seen via one or two distinct colored nodes being inserted between two phases specifically located at the two concentric circles;

D. There is also evident asymmetry on color coding of stance between the left foot and right foot.

5.5. Graphic Display of Gait Dynamics

The explicit deterministic and stochastic structures in panel (B) of Figure 5.5 prescribe the structural dependency of gait dynamics in L+R system. Such a concentric-ring representation of a rhythmic cycle within L+R system is indeed very stable. Two more rhythmic cycles: one is from the middle and another one from the end of the temporal period $[1, 10000]$ among the 77 cycles, are rather similar, as shown in panels (A) and (B) of Figure 5.6. The great degree of stability of gait dynamics pertaining to the L+R system is also seen through a 3D cylinder representation in panel (A) of Figure 5.6.

Such stability implies remarkable adaptability and precision of gait dynamics and its underlying structural dependency. The adaptability is primarily due to the interplay of deterministic and

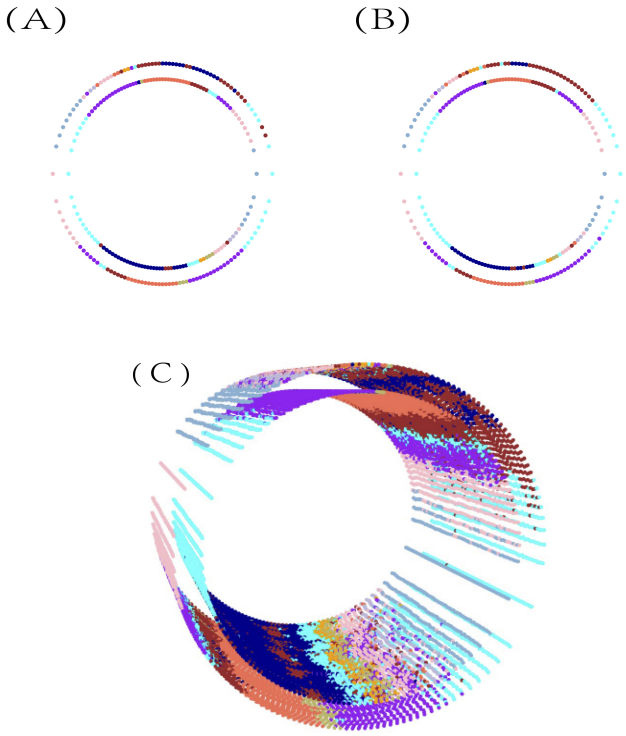

FIGURE 5.6. 3D cylinder representation of evolution of rhythmical cycles in L+R system of subject #5. (A) Concentric-ring for a rhythmic cycle from the middle of $[1, 10000]$; (B) Concentric-ring for a rhythmic cycle from the final part of $[1, 10000]$; (C) 3D cylinder representation of evolution of rhythmic cycles from the 3rd to the 70th.

stochastic structures on the left and right foot. The deterministic structures give rise to a "typical" 2D coupling phase trajectory, while stochastic ones seemingly allows variations in lengths to happen among many components (or phases) of the typical cycle with total precision being about 36ms (=:4600/128). Such a precision is possible only when the deterministic structures are governed strictly by the biomechanics of human musculoskeletal system.

5.5.1. Integrating waist sensor into L+R system. After constructing the rhythmic gait dynamics in L+R system, we then integrate it with the waist sensor. By applying the L1G2 algorithm on the 3D time series from Waist sensor, the resultant local coding sequence is reported in panel (A) of Figure 5.7, while the results derived from the global coding scheme is reported in panel (B) of Figure 5.7 for one rhythmic cycle with 3 layers of concentric circles. A 3D cylinder from 3rd to 70th rhythmic cycles is built and reported in panel (C) of Figure 5.7. It is clear that

3D time series from Waist sensor is rhythmic. But the rhythm is not symmetric with respect to dynamics in L+R system. Likewise, the Wrist sensor can be integrated with L+R system as well.

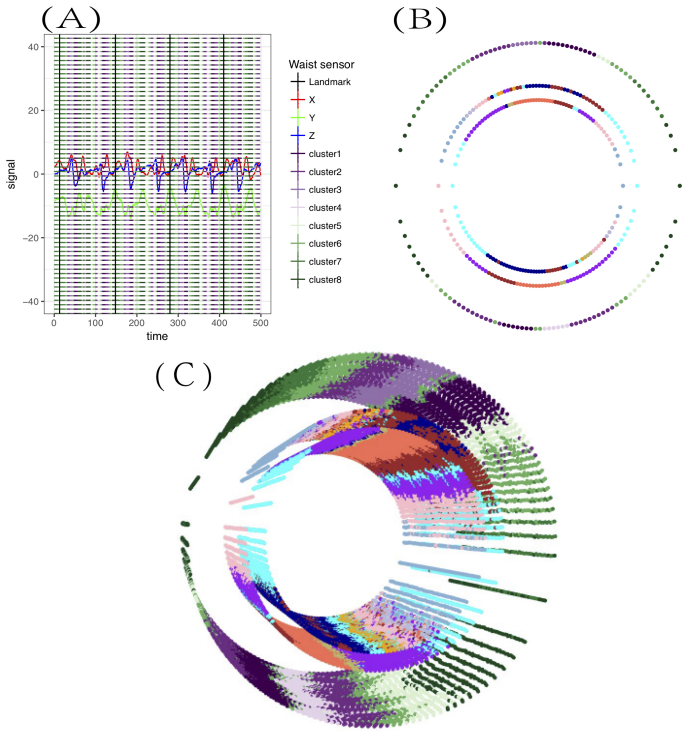

FIGURE 5.7. Integrated gait dynamics of Waist and L+R system. (A) Color coded 3D time series from waist with 8 clusters resulted from the local coding scheme of L1G2 algorithm. (B) Result of L1G2 algorithm represented by 3 layers of concentric-ring pertaining to the 3rd rhythmic cycle on the temporal period $[1, 10000]$; (C) 3D cylinder representation of evolution of rhythmic cycles from the 3rd to the 70th of this integrated system of three sensors.

5.5.2. Passtensors for individual authentications. The applications of coherently computed gait dynamics are rather wide and diverse. Here we mention two essential one in passing without going into details, and then focus on cybersecurity. The first comment is that this L1G2 algorithm will allow us to integrate acceleration sensors with gyroscope sensors. By combining the two kinds of sensors, the resultant gait dynamic system will be rather complex, but extremely interesting. The second comment is obvious that such a 3D representation can be utilized as a platform for mimicking the entire gait dynamics captured by time series data derived from the four acceleration sensors. Such a task of building realistic mimicry of a complex system is technically very challenging, while is scientifically very important, for instance in robotics. Up to now, robots still

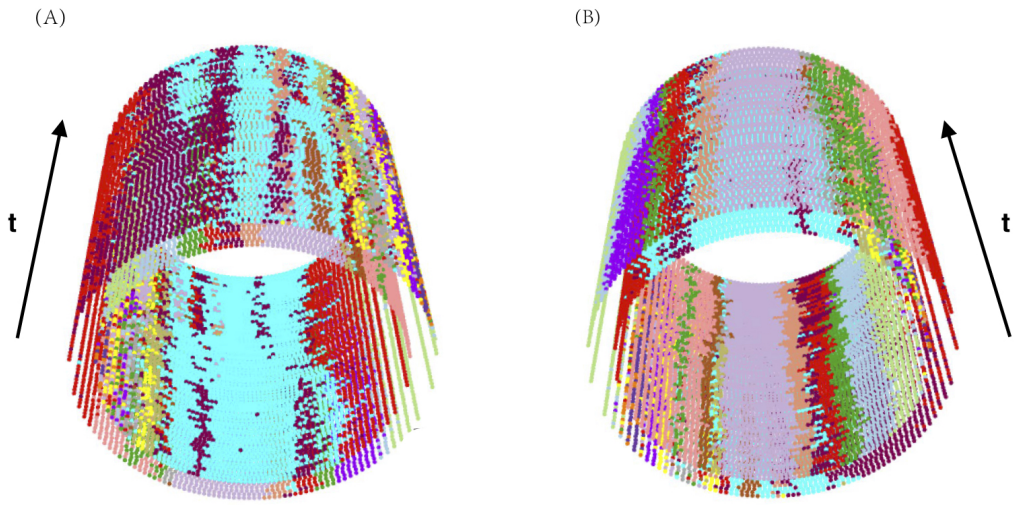

FIGURE 5.8. Two angle-views of 3D passtensor constructed from subject #5's treadmill walking with slope changes in the middle of the temporal period in t. The slope changes cause very subtle change on (A).

walk in very unhuman-like fashions. This issue might be resolved to great extent by incorporating gait dynamics.

Now we turn to cybersecurity, clinical diagnosis and self-evaluating individual health statuses. It becomes clear that, based on our 3D graphic displays of gait dynamics, an individual's process of rhythmic cycle is characterized by the evolution of cyclic deterministic phases with individual specific twists as well as idiosyncratic stochastic deviations associated with all phases. Hence, a 3D cylinder indeed becomes a basis for authenticating this particular individual. For this use, such a 3D cylinder is called "passtensor". More specifically speaking, a L+R system's deterministic cycle of 2D biomechanical phases: from one landmark proceeding to the next one, in indeed provides a rigid frame, while the stochastic phases' lengths and presence or absence of some color codes between adjacent phases provide the soft frames for the purposes of authentications. This authentication capacity further illustrated as follows. For instance, consider the subject #5 in MAREA walked on a treadmill with slope change: from horizontal(0°) to 5° during a recording period. This person's 3D passtensor corresponding to this period is shown in Figure 5.8 with two views from two different angles. The angle specific view in panel (A) of Figure 5.8 reveals visible changes. Such changes are likely critical patterns for authentication purposes.

Here we briefly reiterate the practical uses of our 3D cylinder graphic display of gait dynamics in self-evaluating individual health statuses. By stacking two temporal segments of gait time series from two different temporal periods, we can examine the degrees and aspects of similarity and differences in regarding to deterministic and stochastic structures between these two temporal segments. This is an effective way of finding out subtle and minute discrepancies to serve the early warning purposes.

5.6. Conclusion

5.6.1. Conclusion in system complexity. Our first theme of data-driven computing paradigm, PSSA, allow us to include many principle gait states as a collective of key characteristics for identifying as many people as we want. From many aspects, this identification approach is indeed very distinct from identifications based on facial and voice recognitions, finger-print or retina scanning. It is much easier to achieve social unbiasedness. It is much more difficult to imitate or to fake.

Our second theme of data-driven computing paradigm, consisting of L1G2 coding and landmark algorithms, enables us to explicitly manifest multiscale dynamic patterns of gait dynamics. The graphic displays of single rhythmic cycle and collective 3D passtensor clearly demonstrate how the deterministic circle of biomechanical phase couples with stochastic variations sprinkling between consecutive phases, and offer a whole-view of an individual's gait dynamics. Such intricate coupling relations between deterministic and stochastic structures are the backbones of structural dependency of gait dynamics. They retain essential basis for mimicking an individual's gait dynamics in animation. Its practical uses in clinical diagnosis and cybersecurity are also evident. In fact, the original motivations of this gait study is aiming at detecting relative minor changes in gait dynamics for healthy peoples and gesture tuning for athletes. These two topics require very detailed structures within personal dynamics.

From computational science perspective, our PSSA and L1G2 coding algorithm rests on the crucial fact: different time series have different functions linking to different subsystems of a complex system of interest, so they should not be treated equally and uniformly. Such a rationale is a key for revelations of multiscale structural dependency. It is also the key rationale for recreating a

system's authentic dynamics. Overall, good design of graphic displays definitely pave avenues for true understanding onto a complex system.

5.6.2. Conclusion in security issue. PSSA is purely developed for individual identification within a close community, such as a company or agency that needs a high degree of security. Within a close community or company, PSSA is an effective alternative to facial recognition. Since it does not suffer problems due to shading on images or shadowing and cause social biases. And any individual outside of this community will be identified as outliers. Its application beyond a close community is still in a stage of theoretical research. In theory, it might be possible to convert a 3D video recording data into an accelerometer-based data format. But this technique is still not yet available. In fact, at the current state of technologies, any real-world recording via one camera, for example, CCTV is unlikely to create an authentic 3D recording because of missing data.

For individual gait dynamics, ours developments are geared to help individual to do self-detections for minor gesture changes when walking or doing activities. Such analysis and results are highly personal. So, they intend to be kept and used only by the owner of data. Our potential role would be limited to pointing out where minor changes might have taken place. Even this step is still under intensive researches.

CHAPTER 6

Heterogeneous Geometric Information of Multiclass Classification

6.1. Introduction

Nowadays Machine Learning (M.L.) based Artificial Intelligence (A.I.) researches are by-and-large charged to endow machines with various human's semantic categorizing capabilities **[99]**. Given that human experts hardly make semantic categorizing mistakes, should machine also help to explain: How and Why, to human? We demonstrate that possible answers are computational and visible under any Multiclass Classification (MCC) setting. The keys are: first compute the pertinent information content without artificial structure; secondly, graphically display such information content via multiscale geometries, such as a tree, a network or both, to concisely organize and deliver pattern-based knowledge or intelligence contained in data to human attentions.

Multiclass Classification is one major topic **[7**,**15**,**30**,**47**,**116]** of associating visual images or text articles with semantic concepts **[34**, **76**, **121]**. Its two popular techniques: flat and hierarchical, are prone to make mistakes **[4**, **31**, **54]**. Since a machine is primarily forced to assign a single candidate label toward a prediction. No less, no more. Such a forceful decision-making to a great extent ignores the available amount of information supported by data. With such kind of M.L. in the heart of A.I., it is beyond reasonable doubt that A.I. is bound to generate fundamental social and academic issues in the foreseeable future, if its error-prone propensity is not well harnessed in time.

If completely error-free A.I. is not possible at current state of technology, then at least it should tell us its decision-making trajectory leading up to every right or wrong decision. It is in the same sense as the recommended fourth rule of robotics: "a robot or any intelligent machine-must be able to explain itself to humans" to be added to Asimov's famous three. Since we need to see why, how and where errors occur in hope of knowing what causes, and even figuring out how to fix it.

Such a quality prerequisite on A.I. and M.L. is also coherent with concurrent requirements put forth by many governments around the world: transparent explanation upon each A.I. based decision

is required. Now it is a critical time point to think about how to coherently build and display data's authentic information content that can afford the making of explainable error-free decisions. So such information content with pertinent graphic display can be turned into Data-driven Intelligence. In this chapter, we specifically demonstrate Data-driven Intelligence for Multiclass Classification. This choice of M.L. topic is in part due to that classification is human's primary way of acquiring intelligence, and also in part due to its fundamental importance in science and industry.

On the road to Data-driven Intelligence, we begin by asking the following three simple questions. First, the naive one is: where is relevant information in data? Secondly, what metric geometry is suitable to represent such information content? Finally, how to make perfect, or at least nearly perfect empirical inference or predictive decision-making? We address these three non-hypothetical questions thoroughly based on a model-free label-embedding tree. Here we explicitly show the nature of information content under Multiclass Classification as: multiscale heterogeneity. Such information heterogeneity can be rather intertwined and opaque when its three data-scales: numbers of label, feature and instance, are all big.

The chapter is organized as follows. In Section 6.2, we describe the background and related work of MCC. In Section 6.3, we develop a new label-embedding tree constructed via partial ordering and a classification schedule. In Section 6.4, we illustrate a tree-decent procedure with early stop and represent the error flow. In Section 6.5, we explore the heterogeneity embedded within labels. A conclusion and remarks are given in Section 6.6.

6.2. Multiclass Classification

A generic Multiclass Classification (MCC) setting has three data scales: the number of label L, the number of feature K and total number of subjects N. Each label specifies a data-cloud. A data-cloud is an ensemble of subjects. Each subject is identified by a vector of K feature measurements. The complexity of data and its information content under any MCC setting is critically subject to L, K and N. The goal of Multiclass Classification is to seek for the principles or intelligence that can explain label-to-feature linkages. Such linkages are intrinsically heterogeneous as being blurred by varying degrees of mixing among diverse groups within the space of labeled data-clouds. Since

such data mixing patterns are likely rather convoluted and intertwined, so the overall complexity of information content must be multiscale in nature.

Specifically speaking, its global scale is referred to which label's point-cloud is close to which, but far away from which. Though such an idea of closeness is clearly and fundamentally relative, it is very difficult to define or evaluate precisely. That is, such relativity essence can't be directly measured with the presence of two point-clouds, but it can be somehow reflected only in settings involving three or more point-clouds. From this perspective, all existing distance measures commonly suffer from missing the data-clouds' essential senses of relative closeness locally and globally. For instance, recently Gromov-Wasserstein distance via Optimal Transport has been proposed as a direct evaluation of distance between two point-clouds [**109**]. But it suffers from the known difficulty in handling high dimensionality (large K). So this distance measure likely misses the proper senses of relative closeness among point-clouds, especially when K is big.

In this chapter, we propose a simple computing approach to capture the relative closeness among all involving point-clouds without directly and explicitly evaluating pairwise cloud-to-cloud distance. The key idea is visible as follows: through randomly sampling a triplet of singletons from any triplet of point-clouds, we extract three partial ordering among the three pairs of cloud-to-cloud closeness. By taking one partial ordering as one win-and-loss in a tournament involving $\binom{L}{2}$ teams, we can build a dominance matrix that leads to a natural label embedding tree as a manifestation of heterogeneity on the global scale. Such a triplet-based brick-by-brick construction for piecing together a label embedding tree seems intuitive and natural. Indeed such a model-free approach is brand new to M.L. literature [**15**, **17**]. The existing hierarchical methods build a somehow symbolic label embedding tree by employing a bifurcating scheme that nearly completely ignores the notion of heterogeneity [**3**, **15**, **70**].

After building a label embedding tree on the space of L labels, we further derive a predictive graph, which is a weighted network with precisely evaluated linkages. This graph offers the detailed closeness from the predictive perspective as another key aspect of geometric information content of MCC. To further discover the fine scale information content of MCC, we look into heterogeneity embraced by each label. Clustering analysis is applied on each label's point-cloud to bring out a natural clustering composition, and then label each cluster pertaining to a sublabel. By doing

so across all labels, we result in a space of sublabel with much larger size than L. Likewise we compute a sublabel embedding tree and its corresponding predictive graph. These two geometries then constitute and represent the fine scale information content of MCC.

A real database, Major League Baseball (MLB) $PITCHf/x$, is analyzed for the purpose of application. The availability of data is mentioned in Appendix D. Since 2008 the PITCHf/x database of MLB has been recording each every single pitch delivered by MLB pitchers in all games at its 30 stadiums. A record of a pitch is a measurement vector of 21 features. A healthy MLB pitcher typically pitches around 3000 pitches, which are categorized into one of pitch-types: Fastball, Slider, Change-up, curveball and others types. We collect data from 14 ($= L$) MLB pitchers, who threw around 1000 Fastball or more during the 2017 season. As one pitcher is taken as a label, his seasonal fastball collection is a point-cloud. It is noted that each pitcher tunes his Fastball slightly and distinctively when facing different batters under different circumstances of game. That is, multi-scale heterogeneity is inherently embedded into each point-cloud.

A potential feature set is selected based on permutation-based feature importance measure. The importance score is defined as the reduction in the performance of Random Forest after permuting the feature values. All real data illustrations for the entire computational developments throughout this chapter is done with respect to a feature set consisting of 3 features: horizontal and vertical coordinates, and horizontal speed of a pitch at the releasing point. Results on other larger feature-sets are reported in Appendix D.

6.3. Label Embedding Tree

We develop a computing paradigm to nonparametrically construct the label- and sub-label embedding trees in this chapter. This paradigm is designed to be scalable to the three factors: L, K and N. With a label-triplet, say (La, Lb, Lc), in the brick-by-brick construction, partial ordinal relations are referred to: $D(La, Lb) < D(La, Lc)$ for example, where $D(.,.)$ is the unspecified "distance" between two label clouds. It is emphasized that the algorithm is devised to extract such relations without explicitly computing the three pairwise distances $D(.,.)$. These relations found among three point-clouds are stochastic in nature.

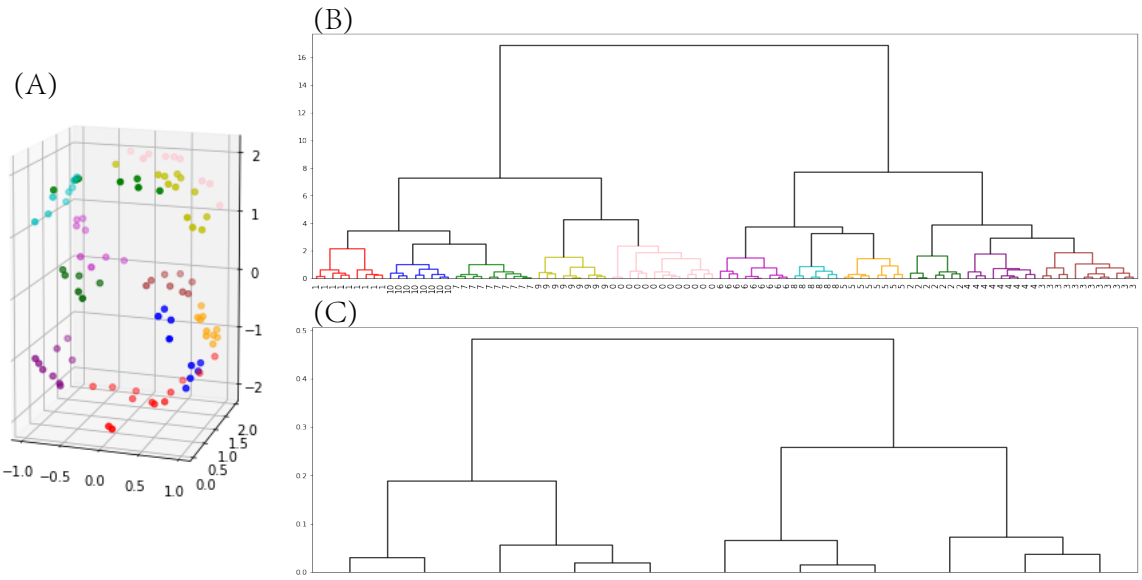

FIGURE 6.1. Illustrating example for **Algorithm 7**. (A) the 3D scatter plot of data; (B) the 11 labeled data-clouds defined by a HC tree; (A) and (B) share the same labeling numbers with the same color; (C) the embedding tree.

Given a triplet of labels La, Lb, Lc, if we randomly sample three singleton vectors in R^K, say X_{La}, X_{Lb} and X_{Lc}: one from each of three labels, separately. A piece of information of partial ordering within the triplet can be shed by inequalities among Euclidian distances $d(.,.)$ among 3 singletons X_{La}, X_{Lb} and X_{Lc}. That is, inequality $d(x_{La}, x_{Lb}) < d(x_{La}, x_{Lc})$ provides a small piece of information about Labels La and Lb being closer than La to Lc and Lb to Lc. By iteratively randomly sampling vector-triplets for a large number of times, say T, the probability of this relative closeness between La and Lb can be estimated as $\hat{P}(D(La, Lb) < D(La, Lc)) = \sum_{t=1}^{T} 1_{d(x_{La}, x_{Lb}) < d(x_{La}, x_{Lc})}/T$.

Via law of large number, we arrive at the relative closeness information by aggregating partial ordering among all possible combination of three labels. Let H be a square dominant matrix with $\binom{L}{2} = L(L-1)/2$ rows and columns. Each entry of H records a probability that "this unspecified distance $D(.,.)$ of a label-pair" is dominated by the same unspecified distance of another label-pair. Denote i_{xy} is the index of a label pair Lx and Ly. The entry of H in the i_{ab}th row and the i_{cd}th column records the related probability between these two label pairs,

$$H[i_{ab}, i_{cd}] = P(D(La, Lb) < D(Lc, Ld)) \tag{6.1}$$

It is noted that $H(i,j) + H(j,i)$ is equal to 1. In this way, H realizes the partial ordering among all pairs of labels.

Here we illustrate the validity of this algorithm through a small example, as shown in Figure 6.1. A S-shape data set is simulated in R^3 space, see panel (A). Hierarchical clustering is implemented and a dendrogram is shown in panel (B). 11 clusters are obtained by cutting the dendrogram at a certain tree height and each cluster is marked with different color. Consider each cluster as a label, and a label embedding tree is created via **Algorithm 7** to show the hierarchical structure among those 11 classes in (C). It shows that our labeling tree built by only using partial ordering can reflect the original hierarchy among labels very well. In short, our dissimilarity matrix makes more sense in showing the natural label-cloud hierarchical dependency, which is the most advantage to distinguish our labeling tree from others.

There is a natural way to do classification based on this triplet partial ordering. We can simply assign a singleton or a batch of unlabeled sample with a new label L_{new}, which never appears in the previous label set. So there is supposed to be $L+1$ labels in total. Then, the triplet-version dissimilarity $\binom{L+1}{2} \times \binom{L+1}{2}$ matrix H_{new} can be calculated for all those $L+1$ labels. The classified label is just the one that is the closest to the new label, see **Algorithm 8**. Actually, given the previous $\binom{L}{2} \times \binom{L}{2}$ matrix H pre-trained, it is only necessary to calculate the rest $\binom{L+1}{2} \times L$ sub-matrix. That is to say, we randomly sample two singletons X_{La} and X_{Lb} from two labels La and Lb, respectively, and sample one unlabeled sample X_{new} from L_{new}. The partial ordering now turns out to compare $d(X_{La}, X_{new})$ and $d(X_{Lb}, X_{new})$. Via a large number of sampling, we gain information about $P(D(La, L_{new}) < D(Lb, L_{new}))$ and its counterpart. Let H_{new} record all newly added probabilities of such dominance. Then the label-pairwised dissimilarity matrix is calculated via the column sum of H_{new}. Therefore, the classification procedure is equivalent to aggregating all binary classifiers and vote according to the sum of probability, which is exactly one-versus-one classification with a soft vote strategy. One brand new property is that, when L_{new} represents a unlabeled data-cloud, the geometry of this data-cloud is fully used in this predictive decision-making.

Algorithm 7 Label Embedding Tree

Denote: H is a $\binom{L}{2} \times \binom{L}{2}$ ranking matrix,

$H[i_{ab}, i_{cd}] = P(D(La, Lb) < D(Lc, Ld))$

where i_{ab} is the index of label pair La and Lb, $D(La, Lb)$ is their dissimilarity which is inaccessible.

Initialize: H with all entries 0

for (La, Lb, Lc) in all unique label triplets:

Randomly sampling a triplet of data for T times with replacement, denoted as $(X_{La}^{(1)}, X_{Lb}^{(1)}, X_{Lc}^{(1)})$, $(X_{La}^{(2)}, X_{Lb}^{(2)}, X_{Lc}^{(2)})$, ..., $(X_{La}^{(T)}, X_{Lb}^{(T)}, X_{Lc}^{(T)})$

where X_L is a single sample of data with label $y = L$

for t in $1, ..., T$:

if $d(X_{La}^{(t)}, X_{Lb}^{(t)}) < d(X_{La}^{(t)}, X_{Lc}^{(t)})$: $H[i_{ab}, i_{ac}] += 1/T$

else $H[i_{ac}, i_{ab}] += 1/T$

if $d(X_{La}^{(t)}, X_{Lb}^{(t)}) < d(X_{Lb}^{(t)}, X_{Lc}^{(t)})$: $H[i_{ab}, i_{bc}] += 1/T$

else $H[i_{bc}, i_{ab}] += 1/T$

if $d(X_{La}^{(t)}, X_{Lc}^{(t)}) < d(X_{Lb}^{(t)}, X_{Lc}^{(t)})$: $H[i_{ac}, i_{bc}] += 1/T$

else $H[i_{bc}, i_{ac}] += 1/T$

end for

end for

Calculate $K \times K$ labeling dissimilarity matrix $\bar{D}$

$\bar{D}(La, Lb) = E_{Lx,Ly}\{P(D(Lx, Ly) < D(La, Lb))\} = \sum_j H(j, i_{ab}) / \binom{L}{2}$

Output: a hierarchical clustering tree based on the dissimilarity matrix $\bar{D}$

We can also sample X_{La} and X_{Lb} from the neighbors of X_{new} to extract the partial ordering locally. Let's choose M-nearest neighbors of X_{new} constrained in the data with label La, denoted as $X_{M|La} = (X_{La}^{(1)}, X_{La}^{(2)}, ..., X_{La}^{(M)})$, and so is $X_{M|Lb}$. We look at whether there are relatively more La's compared with Lb's in the M nearest neighbors. This classification becomes k-Nearest Neighbor with tuning parameter k chosen to be M. If we repeat the aforementioned procedure for a large number of times, we have another way of extracting information of $P(D(La, L_{new}) < D(Lb, L_{new}))$.

Thus, **Algorithm 8** is equivalent to one-versus-one classification with k-NN as its classifier. These properties also explain why our triplet comparison is so important.

Besides, **Algorithm 8** can indicate where the unknown label is located within the previous label embedding tree. The label embedding tree with an unknown label embedded is clear to view which labels are mixed with the unknown label in a small branch and which labels is far away. See Figure 6.2 for an illustration.

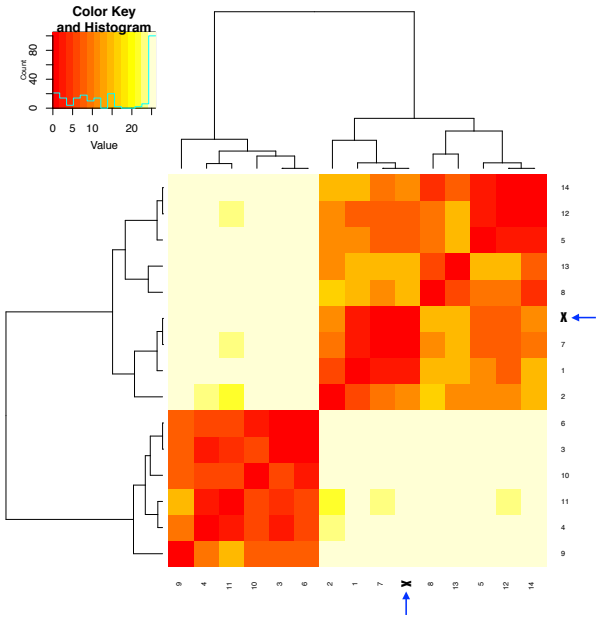

FIGURE 6.2. Label embedding tree of 14 pitchers with a heatmap of "distance" derived from a computed H and an illustrating example of classifying an unknown label **X**; the truth label is 7.

The number of sampling iteration T is supposed to be as large as possible. In practice, T should be chosen dependent on the sample size of each label. If the data is balanced, $T = N/L$, otherwise, $T = max_i N_i$ to cover the biggest label data cloud, where N_i is the sample size for label i. So the time complexity is $O(NKL^2)$.

When L is small or moderate, consider a setting with the number of all possible triplets, $\binom{L}{3}$, being not overwhelmingly big. We perform **Algorithm 7** on all possible triplets to fill up the $\binom{L}{2} \times \binom{L}{2}$ dominance matrix, H. Each of column sum of H tells how many times a label-pair's distance is dominated by distances of all other pairs. So the bigger a column sum is, the larger degree of similarity of this label pair is. Therefore the $\binom{L}{2}$-vector of column sums of H can be

transformed into a natural $L \times L$ similarity matrix, say $\bar{S}$, among all involving labels. In contrast, the $\binom{L}{2}$-vector of row sums of H is a distance (dissimilarity) matrix, say $\bar{D}$, of all labels. Such a $\bar{S}$ or $\bar{D}$ will afford a hierarchy, which is the label embedding tree.

Algorithm 8: Classify X_{new} with an unknown label Lx

Input: a $\binom{L}{2} \times \binom{L}{2}$ matrix H obtained from **Algorithm 7**

Initialize: a $\binom{L+1}{2} \times \binom{L+1}{2}$ ranking matrix H_{new}

$H_{new}[1 : \binom{L}{2}, 1 : \binom{L}{2}] = H$ and the rest entry 0.

for (La, Lb) in all unique label pairs:

Randomly sampling a pair of data for T times with replacement, and concatenate it with X_{new} to make a triplet, denoted as $(X_{La}^{(1)}, X_{Lb}^{(1)}, X_{new})$, $(X_{La}^{(2)}, X_{Lb}^{(2)}, X_{new})$, ..., $(X_{La}^{(T)}, X_{Lb}^{(T)}, X_{new})$ where X_L is a single sample of data with label $y = L$

for t in $1, ..., T$:

if $d(X_{La}^{(t)}, X_{new}) < d(X_{Lb}^{(t)}, X_{new})$, $H_{new}[i_{ax}, i_{bx}] += 1/T$

else $H_{new}[i_{bx}, i_{ax}] += 1/T$

where i_{ax} and i_{bx} are indices for label pair (La, Lx) and (Lb, Lx)

end for

end for

Output1: Classification result a^*, if

$i^* = i_{a^*x}$, $i^* = \text{argmin}_i \sum_j H(j, i)$

Get $(L+1) \times (L+1)$ dissimilarity matrix $\bar{D}_{new}$

$\bar{D}_{new}(La, Lb) = \sum_j H(j, i_{ab}) / \binom{L+1}{2}$

Output2: a hierarchical clustering tree on $\bar{D}_{new}$ and the branch in which the unknown label Lx locates from the previous labeling tree.

When it is too expansive to compute a full version of H, then we start with a sparse version, says H'. By applying the transitivity property in dominance relationship, we can resolve the sparsity issue by making product matrix like $H' \times H'$ to record all indirect dominance with one intermediate **[43]**, see Algorithm D in Appendix D. By embracing such transitivity, as confirmed in our experiment, a reliable distance dominance matrix H can be resulted.

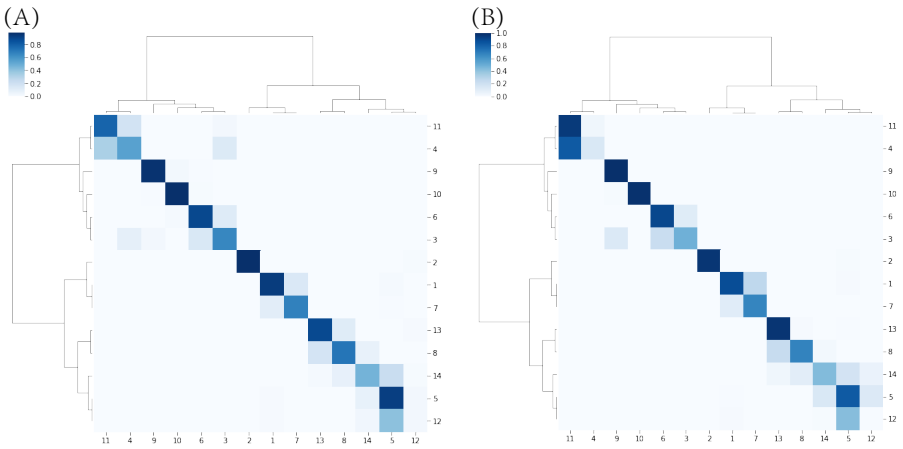

FIGURE 6.3. Label embedding tree superimposed on its confusion matrix: (A) Classification being driven to the tree bottom with a singleton label candidate; (B) Classification can stop early at a tree inter-node.

6.4. Tree-descent Schedule and Error Flow

With a label embedding tree, a very efficient decision-making process can be devised via tree descend framework as depicted in **Algorithm 9**. This algorithm works for any bi-class classifier by making a chain of decisions from top-to-bottom levels of the label embedding tree. So our label embedding tree becomes a scalable platform for decision-making with respect to the number of labels (L). In fact the tree somehow provides an ideal setting for distance metric learning **[122]**, because similar labels have been clustered together.

For prediction purpose, ideally the tree's binary branching structure can allow us to arrive at a singleton label at the bottom of tree, or a small set of label as a small tree branch by avoiding any risk of making any major mistake. A threshold θ defined in **Algorithm 9** works for risk control. If the probability of classification is less than θ, say 0.8, we have less confidence to descend the

labeling tree further, so early stop the iteration and return a label set. This fact can be visualized from our construction of predictive graph below.

Algorithm 9 Classify X_{new} via descending label embedded tree with an early stop

---

Input: a label embedding tree B; a trained Binary Classifier F; a threshold θ to stop descending tree

Denote:

B_{Left} and B_{Right} are the left and right branch on the root node of tree B

$F_L(X_{new})$ returns the probability of classifying X_{new} into Left branch

$F_R(X_{new})$ returns the probability of classifying X_{new} into Right branch

while $(|B| > 1$ & $max\{F_L(X_{new}), F_R(X_{new})\} > \theta)$:

    if $F_L(X_{new}) > F_R(X_{new})$, **then** $B \leftarrow B_{Left}$

    else $B \leftarrow B_{Right}$

end while

Output: label(s) under the current tree B

Let $Y = \{L_j\}_1^L$ and $F = \{f_i\}_1^K$ be the ensembles of label and feature, respectively. Denote a computed label embedding tree as $B[F]$. We derive a label predictive graph, denoted by $G[F]$, based on a confusion matrix. All classification results are collectively summarized into an asymmetric error-flow matrix $E[F] = [e_{i,j}]$ with directed error-flows $(e_{i,j}, e_{j,i})$ between any label pair (L_i, L_j) are the percentages of wrong decisions by predicting L_i to be L_j, and vice versa. $G[F]$ is a weighted network or graphic representation of $E[F]$, see Figure 6.3 for two predictive graphs of 14 MLB pitcher-labels.

The essence of $G[F]$ is that its pairwise directional links $\{(e_{i,j}, e_{j,i})\}$ realistically reflects unequal mixing configurations of labels L_i from L_j. The utility of $G[F]$ is that it allows a smallest predictive label set, while achieving a nearly perfect precision. Such an asymmetry, See Figure 6.3, is invaluable in understanding the MCC setting and in explaining decision-making. This perspective is completely lost when an undirected distance measure is forcefully employed.

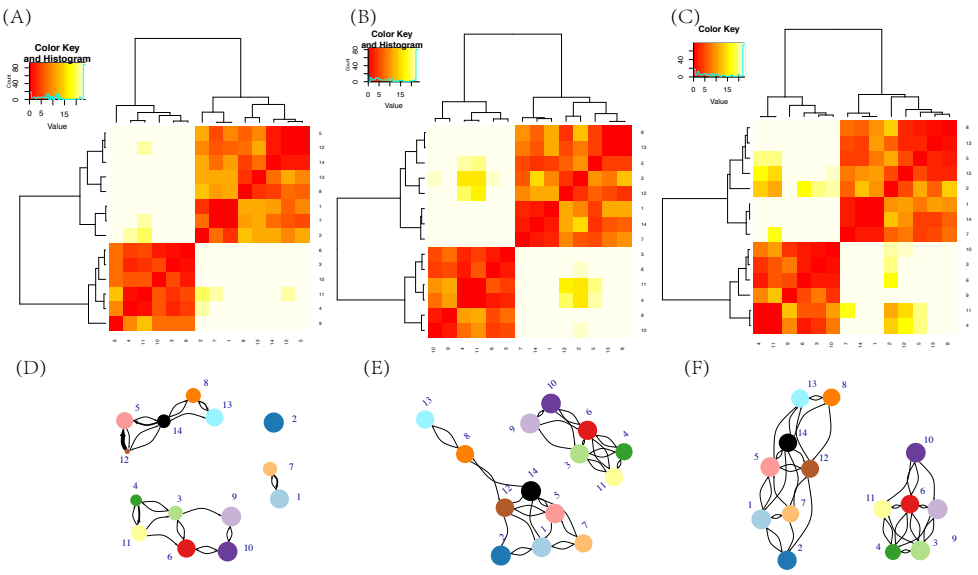

FIGURE 6.4. Dissimilarity matrix and predictive graphs calculated on 3 different Feature Groups with increasing sizes (see Group 1, 3 and 4 in Appendix D)). (A),(B),(C) illustrate the dissimilarity matrix with a label embedding tree embedded on the row and column axis. The label number is the index of a baseball pitcher. There are 14 different pitcher, labeled from 1 to 14; (D),(E),(F) are predictive graphs that visualize the bi-class cut tree descending result.

With the two explicit and visible geometries embraced by the computed label embedding tree and its corresponding predictive graph as the MCC information content with respect to feature set F, the linkages between the label space Y and the collection of point-clouds defined by feature set F become evidently explainable. It is clear to see that the predictive graph is possible to guide us to error-free decision-making if our decision is in a form of a set of potential label candidates, rather than restricted to a singleton. This fact leads us to reflect on the common phenomenal issue: why predicting an unlabeled singleton has to be prone to error? There are at least two key reasons. First, a predictive object can be caught deep within some point-clouds of wrong labels, not just the right one. Therefore, involving all labels' data-clouds at once for such prediction is not ideal. To ameliorate such a situation, a decision-making process descents from the top of a label embedding tree is strategic since MCC's information content is fully used. The second reason is that we ignore what amount of information is available, and simultaneously force ourselves to make a single pick of label.

6.5. Fine Scale Information Content

It is known that each label's point-cloud contains its own label specific heterogeneity. Discovering and accommodating such heterogeneity into MCC's information content in a collective fashion is another essential part of our data-driven computational endeavors. Since our label embedding tree can represent the natural hierarchical structure among separated data clouds, it is straightforward for us to decompose one label's point cloud into separate sublabel clusters and then implement **Algorithm 7**. The sublabel clusters are empirically discovered from each label through a hierarchical clustering tree built upon this label's point could. On the MLB pitching MCC setting, 139 sublabels are generated. We then likewise construct a sublabel embedding tree and its corresponding 139×139 confusion matrix.

Both geometries of fine scale MCC's information content are shown in the three panels of Figure 6.5. They explicitly reveal detail and complex mixing patterns among the 139 sublabel specific point-clouds. Such fine scale information to a great degree reflect the coarse scale information, but at the same time shed new light on its own. For instance, we see how diverse subtypes are belonging to a pitcher's fastball. If all his subtypes are located in a relative small branch of the sublabel embedding tree, then this pitcher fastball pitches are rather uniform. In contrast, if his subtypes are located across several far apart branches, then this pitcher's fastball pitches are difficult to predict. Further we examine in explicit detail how his subtypes are mixing with other pitchers' via a predictive graph. Such examinations allow us to discover how and why this pitcher is in common with which pitchers, and how and why he is distinct with which pitchers. That is, these two geometries are platforms for discovering and establishing many ways of comparing MLB pitchers from many aspects. All these discoveries as diverse parts of the collective knowledge made possible by the fine scale of information content of MCC.

6.6. Conclusion

The coarse and fine scales of information contents of MCC afford us to zoom-in and zoom-out to discover Data-driven Intelligence(D.I.) in visible and explainable fashion. The implied nearly perfect decision-making allows researchers to be responsible. We hope such a D.I. mindset can prevail from sciences to health industries, and beyond. Promoting D.I. is same as promoting truth

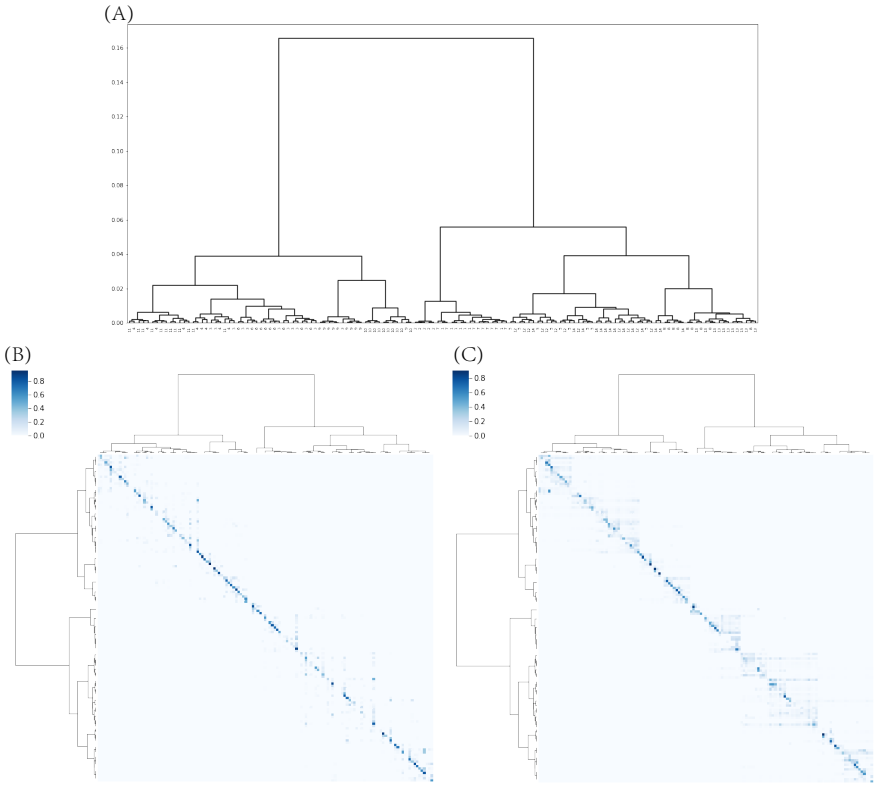

FIGURE 6.5. Fine scale multiscale geometry of 139 sublabels, which belong to 14 pitchers labeled from 1 to 14. (A)The sublabel embedding tree; (B)the confusion matrix with a singleton label candidate; (C) predictions stop early at a tree internode.

and knowledge already contained in data. Human might have been very wasteful in casting away invaluable knowledge by only focusing on forceful prediction.

Finally we make a remark on feature selection. Our standpoint here is that perfect decision-making is the prerequisite on any prediction issue occurring in sciences and health industries. Over these fields, any prediction needs to rightly reflect the amount of information available from data. At the same time, all decision-makers have to be responsible on what they decide. Their subject-matter sensitive criteria can be easily based on the two geometries of MCC's information content. That is, the task of feature selection shall be based on the F and be subject-matter sensitive. Such a standpoint is illustrated in Figure 6.4. By comparing the three sets of geometric information contents pertaining to three feature-sets (feature information given in Appendix D), we gain different

understanding and knowledge regarding the 14 MLB pitchers. We explain such D.I. pertaining to different sets of feature. That is why a prediction is better feature-set sensitive.

In summary, at least under MCC settings, Data-driven Intelligence is one basic principle objective of machine learning in Data Science as well as in Artificial Intelligence.

CHAPTER 7

Conclusion and Future Work

The deterministic and stochastic structure of arrhythmic and rhythmic dynamics have been studied through this dissertation. Upon segment-wised non-stationary time series of arrhythmic pattern, change points as temporal locations of abrupt distributional changes are the key part of the deterministic structure. Within the frame of the change-point skeleton, statistical randomness forms the stochastic part in each stationary segment. By further assuming the recurrence of the stationary segment, regime-switching model or Hidden Markov model can be implemented to investigate the scientific meaning for each underlying regime. While, upon cyclic time series of rhythmic pattern, landmarks and the stable trajectory within each cycle form the crucial information. In gait analysis, the variation of time that a person spends finishing a walking cycle implies the existence of stochasticity.

The deterministic structure plays a significant role in understanding and analyzing the dynamics in complex systems. In financial analysis, by quantifying the correspondence between change point locations resulted from different stock returns, one can measure causal effects- whether one stock's volatility causes an abrupt change to the price of another stock. Such pair-wise dependency is further collected to link stocks of S&P500 into a whole system. In gait analysis, the deterministic structure constitutes the basis of gait identification and authentication. The existence of gait signature and the recurrent pattern motivates us to select system-states with high frequency. It is demonstrated that the principle system-states can be easily applied to differentiate a particular individual as a gait signature. For gait authentication, a stable transition within each circle makes it possible to detect early-stage illness or disorder condition once such stability is found to be broken.

The heterogeneity in real data is fully discussed through this dissertation. For example, in financial data, it is well known that the volatility stage is not homogeneous, so it would be more beneficial to study the regime-switching model compared with the Black-Schole's. In the Multiclass Classification (MCC) setting, there might exist different subtypes in one label cloud, which is

attributed to classification errors of machine learning. In the dissertation, unsupervised learning is implemented to segment the mixed-type data into homogeneous groups. The segment-wised non-stationary stock time series is taken into account and the segmentation is achieved based on the number and location of change points. In MCC, the label-embedding tree provides the geometric connection among the labels and their subtypes, which illustrates why and how the classification error is made by the machine.

So far, the offline method is primarily studied in the dissertation. That is to say, time segmentation is performed after all samples have been collected. For example, the S&P500 network is established based on the correspondence between the retrospective dynamic of a pair of stocks. One of the future works is to generalize the method to online study. The online analysis can be used on streaming data and it is able to react to changes in real time. A possible application is to forecast the volatility regime in the future based on the historical stock price to decrease investment risk.

Another future work is to gain information from an individual to a population. In gait analysis, the fine-scale gait dynamics are represented via the color-coded cylinder under the individual level. With the availability of a larger gait database, it brings the possibility to compare the gait dynamic between different population groups, like male versus female, or youth versus elder. On the other hand, there exists a biometric trait for each participant to make the individual's gait signal very different. The existence of such population-level gait characteristics is still under mystery. In the finance data, it has been demonstrated that the stocks from the same industrial subcategory, such as semiconductor, may transit from low- to high-volatility regimes simultaneously. It motives a potential research topic that whether such local relationship exists in the stock population, or how to figure out the highly associated stock groups in a data-driven way.

APPENDIX A

Appendix of Chapter2

Additional Figures and Tables.

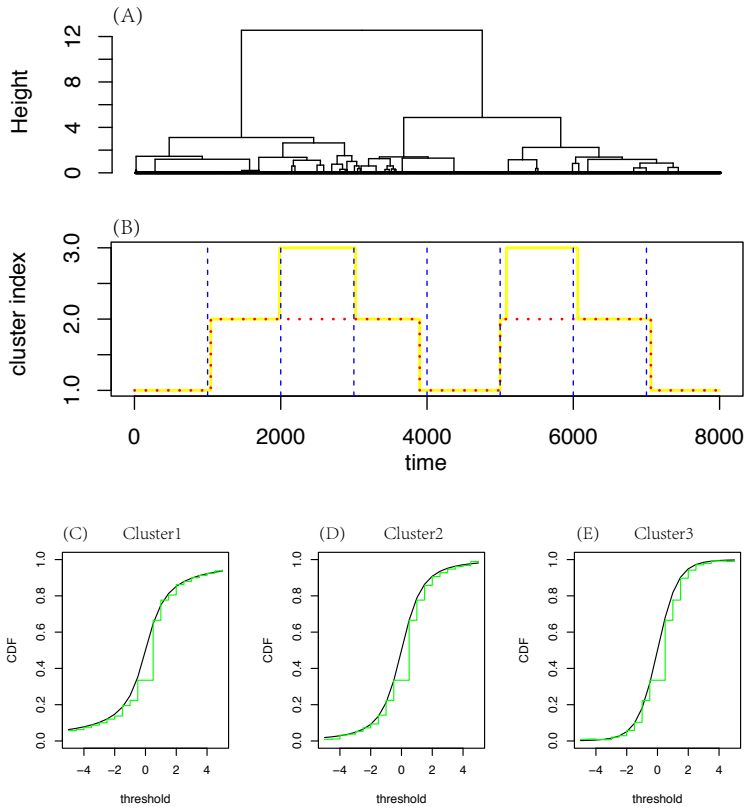

FIGURE A.1. 4-states continuous-distribution decoding in simulation data. (A) Hierarchical Clustering Tree; (B) cluster index switching over time; (C),(D),(E): median eCDFs versus true CDFs, in cluster 1,2,3, respectively

TABLE A.1. Top30 indices with the strongest node strength

incoming		outgoing	
Index	NS	Index	NS
EMC	4.3349	TWX	3.3975
BAC	4.2760	BRCM	3.3148
NTAP	4.2245	NTAP	3.2715
JPM	4.0252	GILD	3.0749
WFC	3.8934	ALTR	2.9504
NBR	3.7955	VLO	2.8755
HON	3.6844	EBAY	2.8178
BRCM	3.6363	HD	2.7764
AIG	3.6327	WMT	2.7619
KBH	3.6230	NVLS	2.7501
CAT	3.6097	CHK	2.7336
WB	3.5598	AMD	2.7331
CTX	3.5570	MXIM	2.7227
WAG	3.4599	YHOO	2.6252
BJS	3.4538	JNJ	2.5732
WLP	3.4503	SCHW	2.5519
LOW	3.3636	IBM	2.5448
SWY	3.3279	XLNX	2.5334
AXP	3.2317	BIIB	2.5313
NOV	3.1355	LLTC	2.5274
BUD	3.1273	MU	2.5269
CHK	3.1227	NVDA	2.5108
DOW	3.0809	BMET	2.4964
KSS	3.0608	TXN	2.4894
VLO	3.0252	C	2.4635
TWX	3.0220	ADBE	2.4633
MO	3.0072	CELG	2.4574
DE	2.9712	TGT	2.4286
COP	2.9598	KLAC	2.3873
TRUE	2.9564	ESRX	2.3860

APPENDIX B

Appendix of Chapter3

Simulation Data.

Denote the two hidden states as "state0" and "state1", and their corresponding covariance matrix "Cov_0" and "Cov_1", respectively. In Section 5, datasets are simulated in 5 different cases described as following.

Simulation Case1

$$Cov_0 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

$$Cov_1 = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$$

Simulation Case2

$$Cov_0 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

$$Cov_1 = \begin{bmatrix} 1 & -0.7 \\ -0.7 & 1 \end{bmatrix}$$

Simulation Case3

$$Cov_0 = \begin{bmatrix} \sigma_1^2 & r * \sigma_1 * \sigma_2 \\ r * \sigma_1 * \sigma_2 & \sigma_2^2 \end{bmatrix}$$

$$Cov_1 = \begin{bmatrix} \sigma_2^2 & r * \sigma_1 * \sigma_2 \\ r * \sigma_1 * \sigma_2 & \sigma_1^2 \end{bmatrix}$$

where σ_1=1, σ_2=1.5, r=0.6.

Simulation Case4

$$Cov_0 = \begin{bmatrix} \sigma_1^2 & r * \sigma_1 * \sigma_2 \\ r * \sigma_1 * \sigma_2 & \sigma_2^2 \end{bmatrix}$$

$$Cov_1 = \begin{bmatrix} \sigma_2^2 & r * \sigma_1 * \sigma_2 \\ r * \sigma_1 * \sigma_2 & \sigma_1^2 \end{bmatrix}$$

where σ_1=1, σ_2=1.5, r=0.2.

Simulation Case5

$$Cov_0 = \begin{bmatrix} 1 & 0.3 \\ 0.3 & 1 \end{bmatrix}$$

$$Cov_1 = \begin{bmatrix} 1 & -0.3 \\ -0.3 & 1 \end{bmatrix}$$

Additional Figures.

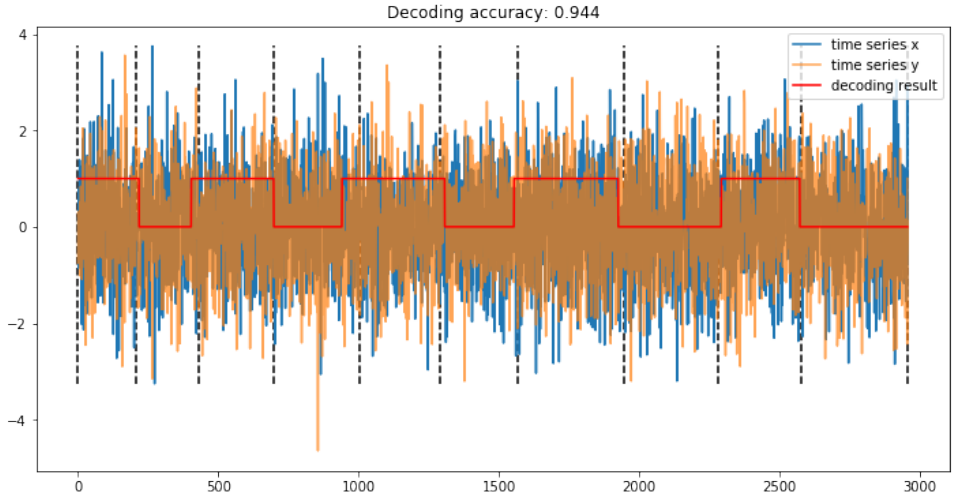

FIGURE B.1. Dataset simulated from bivariate Gaussian "Case2"; vertical dashed line indicates the true change points; red solid line reflects the segmentation result

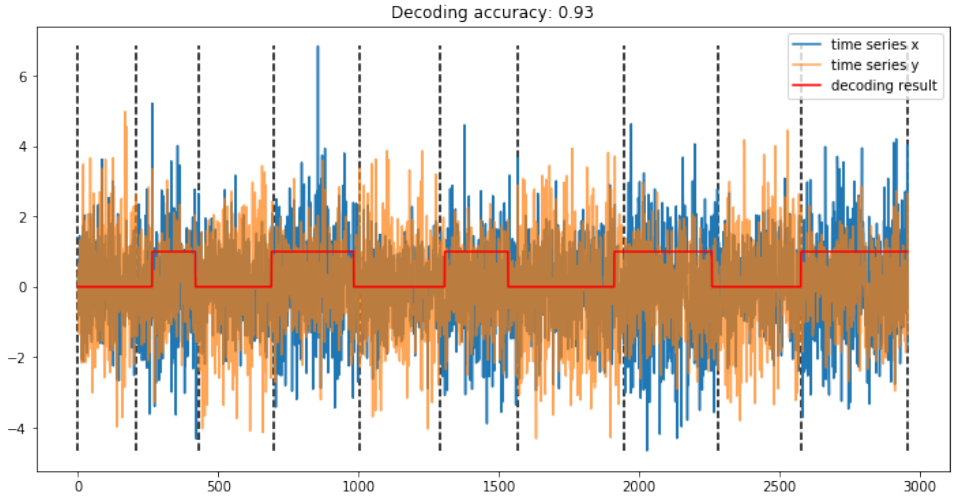

FIGURE B.2. Dataset simulated from bivariate Gaussian "Case3"; vertical dashed line indicates the true change points; red solid line reflects the segmentation result

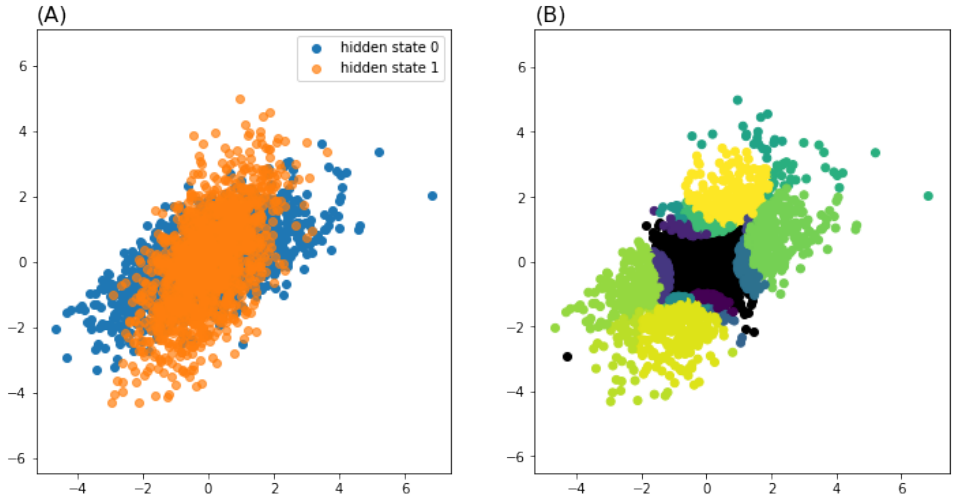

FIGURE B.3. Dataset simulated from bivariate Gaussian "Case3"; (A) scartterplot from two hidden states; (B) data points are plotted in back; "balls" with high weights are painted in different color

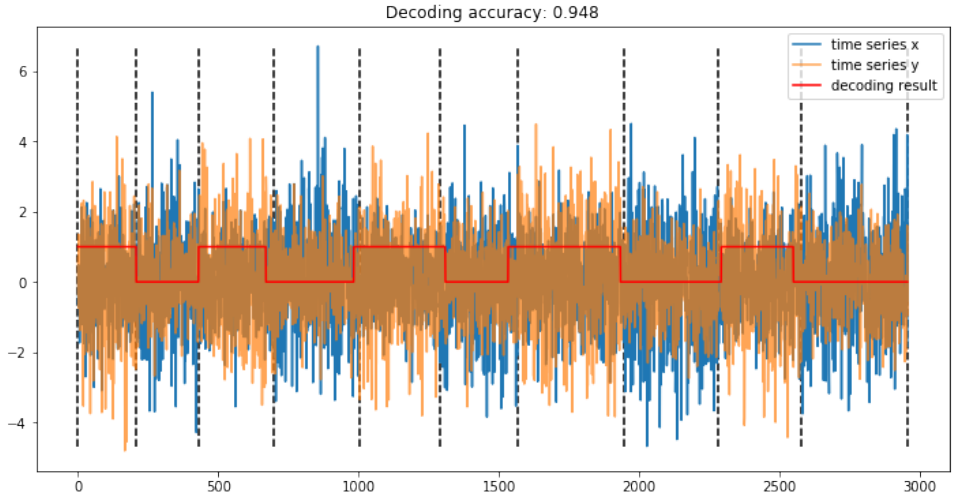

FIGURE B.4. Dataset simulated from bivariate Gaussian "Case4"; vertical dashed line indicates the true change points; red solid line reflects the segmentation result

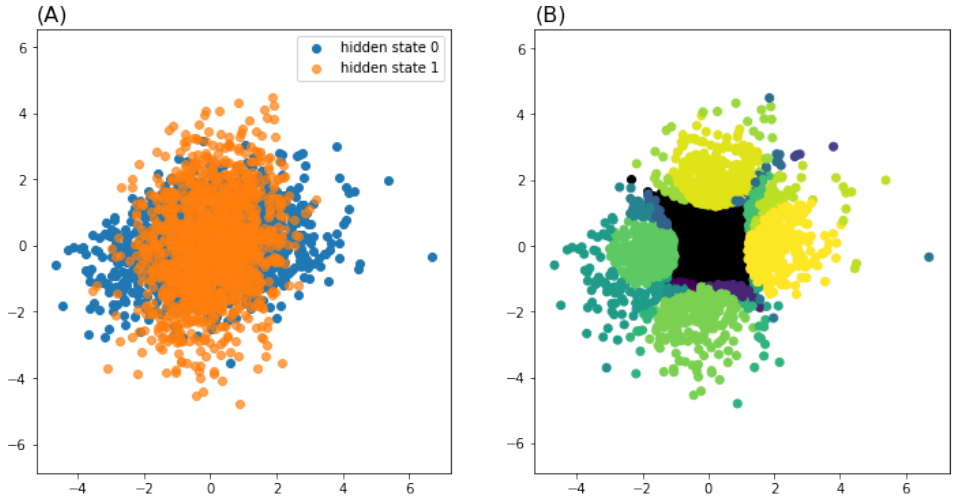

FIGURE B.5. Dataset simulated from bivariate Gaussian "Case4"; (A) scartterplot from two hidden states; (B) data points are plotted in back; "balls" with high weights are painted in different color

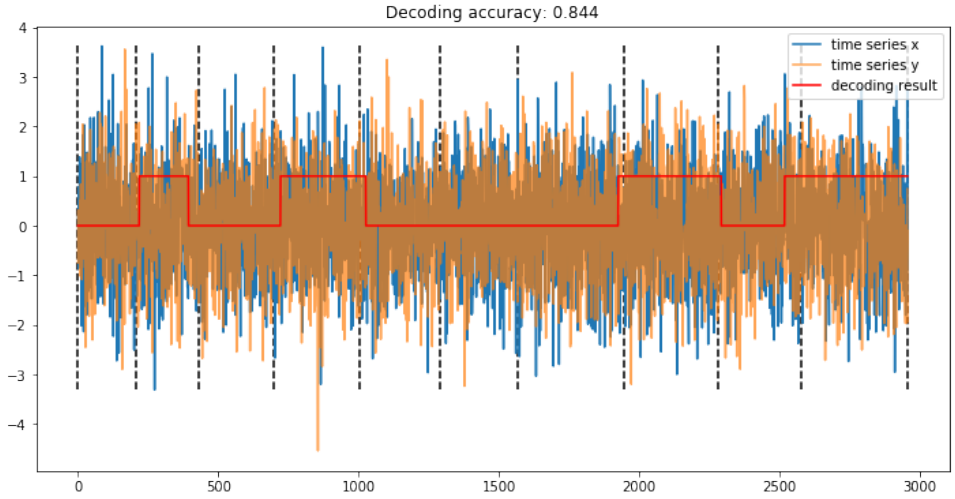

FIGURE B.6. Dataset simulated from bivariate Gaussian "Case5"; vertical dashed line indicates the true change points; red solid line reflects the segmentation result

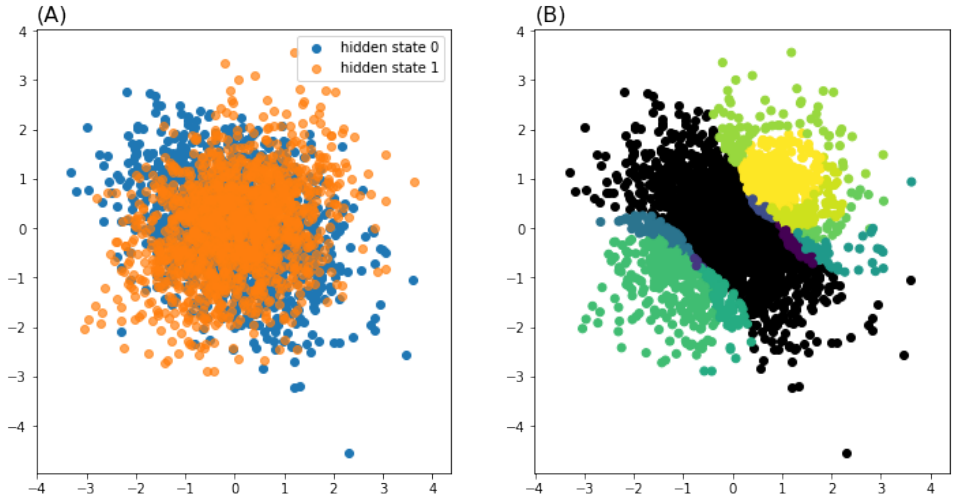

FIGURE B.7. Dataset simulated from bivariate Gaussian "Case5"; (A) scartterplot from two hidden states; (B) data points are plotted in back; "balls" with high weights are painted in different color

FIGURE B.8. Trivariate time series of ADBE

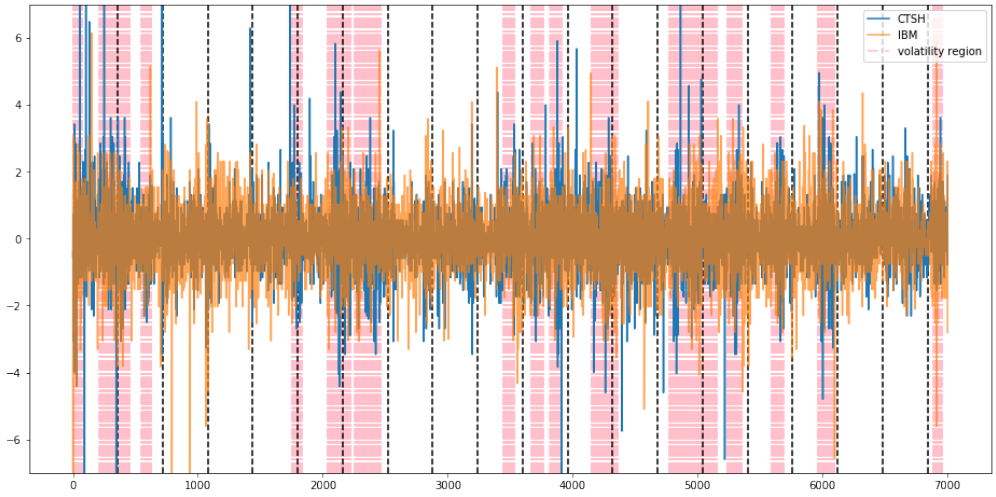

FIGURE B.9. Bivariate returns of CTSH and IBM

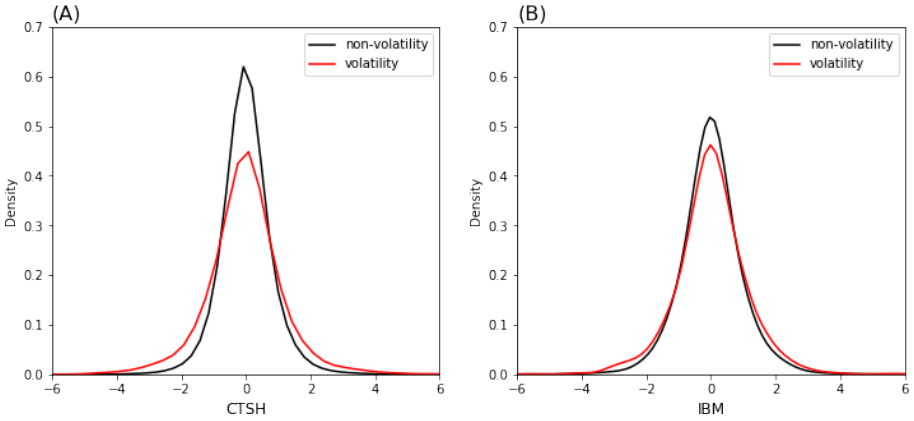

FIGURE B.10. Kernel density estimation for data points on volatility and non-volatility region; (A) CTSH; (B) IBM

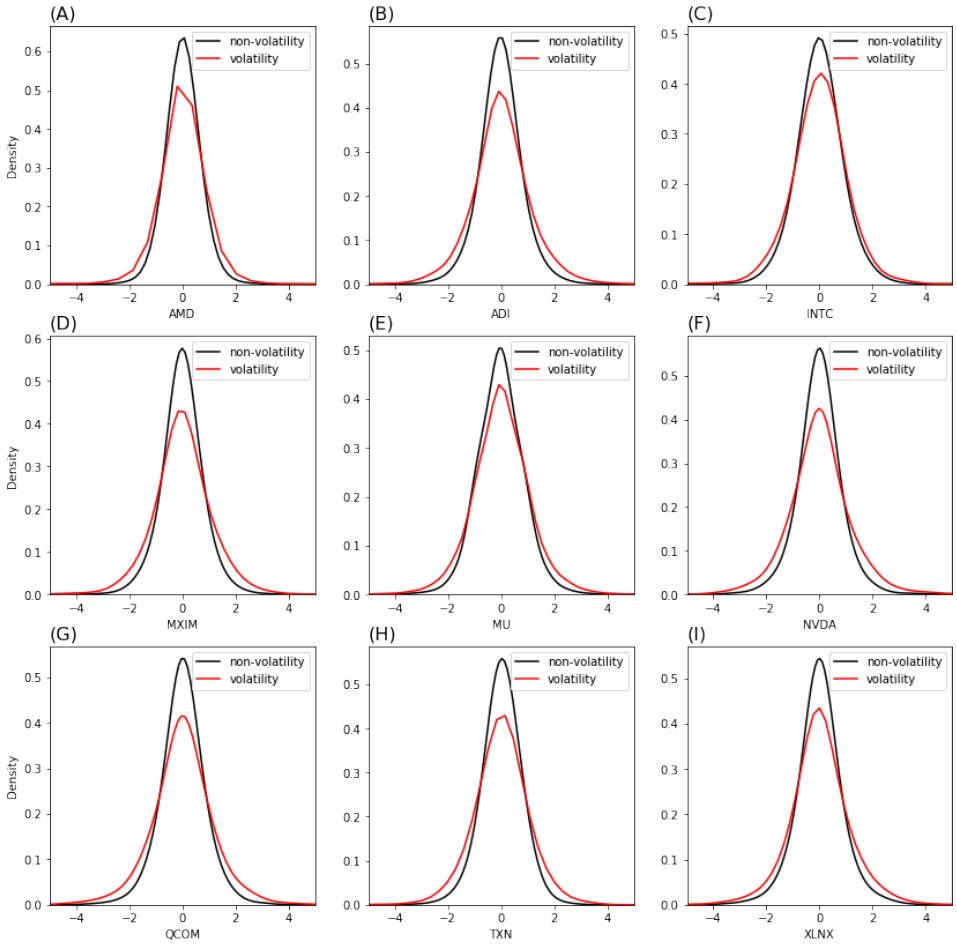

FIGURE B.11. Kernel density estimation for data points on volatility and non-volatility region for 9 semiconductor indexes

APPENDIX C

Appendix of Chapter5

Data Source. The MAREA Gait Database is available at: `http://islab.hh.se/mediawiki/Gait_database`. The Human Gait Database(HuGaDB) is available at: `https://github.com/romanchereshnev/HuGaDB`.

Additional Figures.

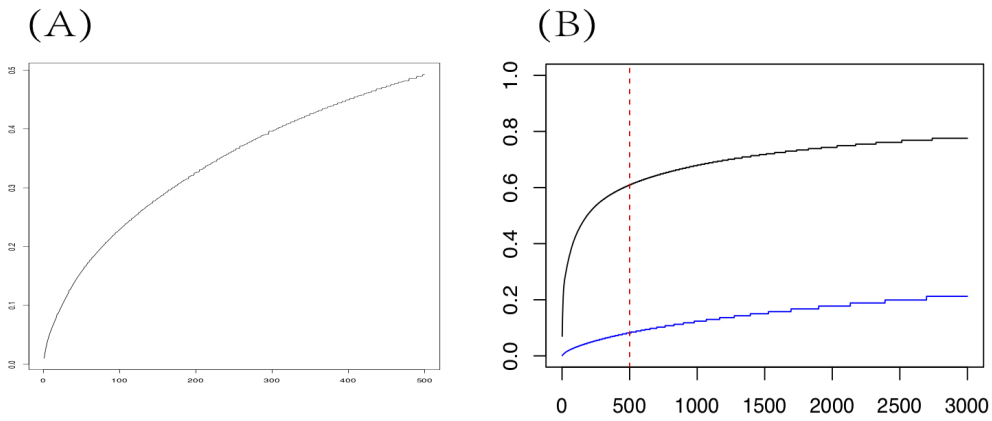

FIGURE C.1. (A): $r(N^*)\, v.s\, N^*$ from 9-dim gait time series from 3 sensors fixed at Left foot and Right foot and wrist among 10 subjects in MAREA database. The triple coding is based on $\alpha = 0.3$ and $\beta = 0.7$ quantiles.
(B): $r(N^*)\, v.s\, N^*$based on 18-dim gait time series derived from 6 sensors fixed to left and right thighs, shines and feet in HuGaDB database. The black curve is pertaining to the triple coding based on $\alpha = 0.1$ and $\beta = 0.9$ quantiles, while the blue curve is based on $\alpha = 0.3$ and $\beta = 0.7$ quantiles

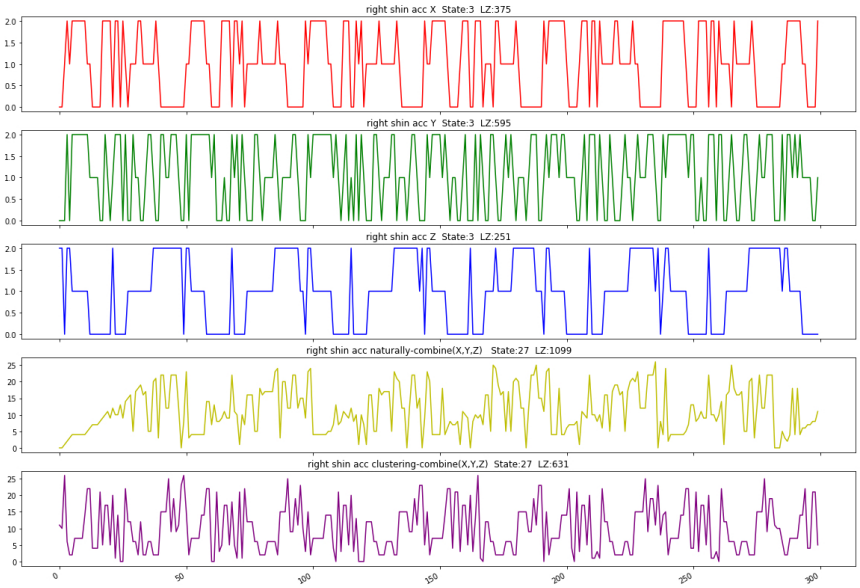

FIGURE C.2. From top to bottom, code each accelerometer time series from **right shine** separately and combine them into one sequence in two different ways; one is a natural way of combination (the second last to the bottom), the other is our clustering-based combination (the last)

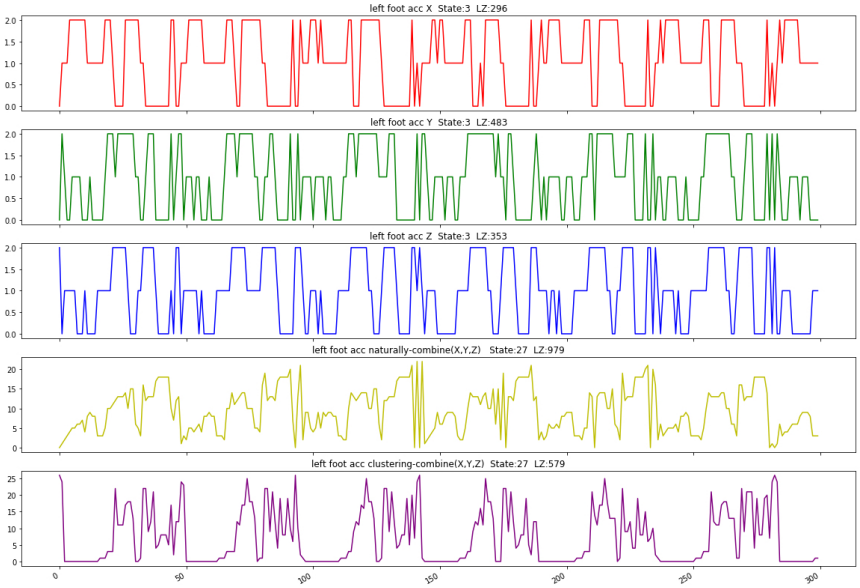

FIGURE C.3. From top to bottom, code each accelerometer time series from **left foot** separately and combine them into one sequence in two different ways; one is a natural way of combination (the second last to the bottom), the other is our clustering-based combination (the last)

APPENDIX D

Appendix of Chapter6

Data Source. The pitching data is available in PITCHf/x database belonging to Major League Baseball via `http://gd2.mlb.com/components/game/mlb/`.

Feature explanation from PITCHf/x. A pitched baseball flight captured by 20 pairs of images via a pair of 60Hz cameras, which have orthogonal optical axes and cover the field of view between pitcher's mound and home plate, are determined with respect to the field coordinates. These images and estimated coordinates are converted into 21 features to characterize the flight's aerodynamics. The 21 features are briefly described as follows.

- The starting speed ("start speed") is measured when the ball is at the point 50 fts away from the home plate, which is very close to the horizontal and vertical coordinates of release point $(x0, z0)$ of a pitch.
- The spin direction ("spin dir") is determined by assuming spin-axis being perpendicular to the movement direction, while spin rate ("spin rate") is the number of rotations per minute.
- Vertical and horizontal movement measurements, denoted by "pfx-z" and "pfx-x", respectively. Topspin and backspin cause positive and negative vertical movements "pfx-z". Therefore this feature has a high association with "start speed" for pitchers, who has the high speed fastball as his chief pitch-type in his repertoire, than for pitchers, who doesn't. The feature "pfx-z" is also associated with features related to how a baseball trajectory curves.
- A baseball trajectory from release point to the home plate is coupled with two straight lines: the tangent line at the release point (x_0, z_0) and the line links the release point and the trajectory's end point. The angle between these two lines is termed "break angle", while the maximum distance between the baseball trajectory and the second straight line

is called and denoted as “break length”. Therefore the three features: “pfx-z”, “break angle” and “break length”, are highly associated with each other.

- The remaining features are three directions of speeds and accelerations at the release point, named “vx0, vy0,vz0” and “ax, ay, az”, respectively, or play only auxiliary roles, like “break y”, “x”, and “y”.

Definition of Feature Groups.

- Feature Group1: “x0", “z0", and “vx0"
- Feature Group2: “x0", “z0", “vx0", “vy0", “start-speed", “end-speed", and “spin-dir"
- Feature Group3: “x0", "z0", “vx0", “vy0", “start-speed" “end-speed", “spin-dir", “spin-rate", “break-angle", “pfx-x", and “pfx-z"
- Feature Group4: all 21 features

Additional Algorithms.

Algorithm D Label Embedding Tree (Sparse)

Alg.1 is applied to get the dominance matrix $H^{'}$ with a smaller sampling iteration T

$H = H^{'} + H^{'} \times H^{'}$

$H(i,j) = min\{H(i,j), 1\}$

$\hat{D}(La, Lb) = \sum_j H(j, i_{ab}) / \binom{L}{2}$

Output: a label embedding tree based on $\hat{D}$

Bibliography

[1] L.-Y.-F. A., L.-L. C., and C. O., *Homogeneity and change-point detection tests for multivariate data using rank statistics*, Journal of the French Statistical Society, 156 (2015), pp. 133–162.

[2] H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, *Identifying people from gait pattern with accelerometers*, Proc. SPIE, 5779 (2005), pp. 7–14.

[3] Aixin Sun and Ee-Peng Lim, *Hierarchical text classification and evaluation*, in Proceedings 2001 IEEE International Conference on Data Mining, 2001, pp. 521–528.

[4] E. L. Allwein, R. E. Schapire, and Y. Singer, *Reducing multiclass to binary: a unifying approach for margin classifiers*, Journal of Machine Learning Research, 1 (2000), pp. 113–141.

[5] A. Alvarez-Alvarez, G. Trivino, and O. Cordon, *Human gait modeling using agenetic fuzzy finite state machine*, IEEE Transactions on Fuzzy Systems, 20 (2012), pp. 205–223.

[6] S. Aminikhanghahi and D. J. Cook, *A survey of methods for time series change point detection*, Knowledge and Information Systems, 51 (2017), pp. 339–367.

[7] Y. Amit, M. Fink, N. Srebro, and S. Ullman, *Uncovering shared structures in multiclass classification*, in Proceedings of the 24th International Conference on Machine Learning, ICML '07, New York, NY, USA, 2007, Association for Computing Machinery, p. 17–24.

[8] R. C. Amorim, *A survey on feature weighting based k-means algorithms*, Journal of Classification, 33 (2015), pp. 210–242.

[9] R. C. Amorim and M. B., *Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering*, Pattern Recognition, 45 (2012), pp. 1061–1075.

[10] P. W. Anderson, *More is different*, Science, 177 (1972), pp. 393–396.

[11] J. Bai and P. Perron, *Computation and analysis of multiple structural change models*, Journal of Applied Econometrics, 18 (2003), pp. 1–22.

[12] L. Barnett, A. B. Barrett, and A. K. Seth, *Granger causality and transfer entropy are equivalent for gaussian variables*, Physical Review Letters, 103 (2009), p. 238701.

[13] L. Baum, T. Petrie, G. Soules, and N. Weiss, *A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains*, Annals of Mathematical Statistics, 41 (1970), p. 164–171.

[14] A. Beinrucker, U. Dogan, and G. Blanchard, *Extensions of stability selection using subsamples of observations and covariates*, Statistics and Computing, 26 (2016), p. 1059–1077.

[15] S. Bengio, J. Weston, and D. Grangier, *Label embedding trees for large multi-class tasks*, in Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1, NIPS'10, Red Hook, NY, USA, 2010, Curran Associates Inc., p. 163–171.

[16] I. Berkes, E. Gombay, and L. Horvath, *Testing for changes in the covariance structure of linear processes*, Journal of Statistical Planning and Inference, 139 (2009), p. 2044–2063.

[17] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, *Sparse local embeddings for extreme multi-label classification*, in Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, Cambridge, MA, USA, 2015, MIT Press, p. 730–738.

[18] A. Bird, *Dna methylation patterns and epigenetic memory*, Genes and Development, 16 (2002), pp. 6–21.

[19] F. Black and M. S. Scholes, *The pricing of options and corporate liabilities*, Journal of Political Economy, University of Chicago Press, 81 (1973), pp. 637–654.

[20] T. Bollersley, *Generalized autoregressive conditional heteroskedasticity*, Journal of Econometrics, 31 (1986), pp. 307–327.

[21] T. Bollersley, R. R. Engle, and J. M. Wooldridge, *A capital asset pricing model with time varying covariances*, Journal of Political Economy, 96 (1988), pp. 116–131.

[22] M. Bosc, F. Heitz, J. Armspach, I. Namer, D. Gounot, and L. Rumbach, *Automatic change detection in multimodal serial mri: application to multiple sclerosis lesion evolution*, NeuroImage, 20 (2003), p. 643–656.

[23] L. B. Chang, A. Goswami, F. Hsieh, and C. R. Hwang, *An invariance for the large-sample empirical distribution of waiting time between successive extremes*, Bulletion of the Institute of Mathematics Academia Sinica, 8 (2013), pp. 31–48.

[24] L. B. Chang, G. Stuart, F. Hsieh, and C. R. Hwang, *Invariance in the recurrence of large returns and the validation of models of price dynamics*, Physical Review E, 88 (2013), p. 022116.

[25] H. Chen and N. R. Zhang, *Graph-based change-point detection*, The Annals of Statistics, 43 (2015), pp. 139–176.

[26] J. Chen and A. K. Gupta, *Testing and locating variance changepoints with application to stock prices*, Journal of the American Statistical Association, 92 (1997), p. 739–747.

[27] R. Chereshnev and A. Kert'esz-Farkas, *Hugadb: Human gait database for activity recognition from wearable inertial sensor networks*, in International Conference on Analysis of Images, Social Networks and Texts, Springer, 2017, pp. 131–141.

[28] H. Chernoff and S. Zacks, *Estimating the current mean of a normal distribution which is subjected to changes in time*, The Annals of Mathematical Statistics, 35 (1964), pp. 999–1018.

[29] K. T. Chi, L. J., and F. C. Lau, *A network perspective of the stock market*, J Empir Financ, 17 (2010), p. 659–667.

[30] M. Cissé, N. Usunier, T. Artieres, and P. Gallinari, *Robust bloom filters for large multilabel classification tasks*, in Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Red Hook, NY, USA, 2013, Curran Associates Inc., p. 1851–1859.

[31] M. M. Cisse, *Efficient Extreme Classification. Data Structures and Algorithms*, PhD thesis, Université Pierre et Marie Curie - Paris VI, 2014. https://tel.archives-ouvertes.fr/tel-01142046/document.

[32] P. K. Clark, *A subordinated stochastic process model with finite variance for speculative prices*, Econometrica, 41 (1973), pp. 135–155.

[33] J. P. Crutchfield, *Between order and chaos*, Nat. Phys., 8 (2012), pp. 17–24.

[34] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei, *What does classifying more than 10,000 image categories tell us?*, in Computer Vision – ECCV 2010, Berlin, Heidelberg, 2010, Springer Berlin Heidelberg, pp. 71–84.

[35] T. Dimpfl and F. J. Peter, *Using transfer entropy to measure information flows between financial markets*, in In Proceedings of Midwest Finance Association 2012 Annual Meetings, New Orleans, LA, USA, 2012, p. 21–24.

[36] Z. Ding and C. W. J. Granger, *Modeling volatility persistence of speculative returns: A new approach*, Journal of Econometrics, 73 (1996), pp. 185–215.

[37] ——, *Modeling volatility persistence of speculative returns: A new approach*, Journal of Econometrics, 73 (1996), pp. 185–215.

[38] E. Eberlein, *Application of generalized hyperbolic lévy motions to finance*, in Lévy Processes, O. E. Barndorff-Nielsen, S. I. Resnick, and T. Mikosch, eds., Birkhäuser, Boston, MA, Boston, MA, 2001, pp. 173–204.

[39] K. A. Emanuell, *Increasing destructiveness of tropical cyclones over the past 30 years*, Nature, 436 (2005), pp. 686–688.

[40] R. F. Engle, *Dynamic conditional correlation: a simple class of multivariate generalized autoregressive conditional heteroskedasticity models*, Journal of Business and Economic Statistics, 20 (2002), pp. 339–350.

[41] S. Fine, W. Singer, and N. Tishby, *The hierarchical hidden markov model: analysis and applications*, Machine Learning, 32 (1998), pp. 41–62.

[42] Y. Fu and R. N. Curnow, *Maximum likelihood estimation of multiple change points*, Biometrika, 77 (1990), pp. 563–573.

[43] H. Fushing and K. Fujii, *Mimicking directed binary network for exploring systemic sensitivity: Is ncaa fbs a fragile competition system*, Frontiers in Applied Mathematics and Statistics, 2 (2016), p. 9.

[44] D. Gafurov, K. Helkala, and T. Sondrol, *Biometric gait authentication using accelerometer sensor*, Journal of Computers, 1 (2006), pp. 51–59.

[45] M. Gietzelt, S. Schnabel, K.-H. Wolf, F. Busching, B. Song, S. Rust, and M. Marschollek, *A method to align the coordinate system of accelerometers to the axes of a human body: The depitch algorithm*, Computer Methods and Programs in Biomedicine, 106 (2012), pp. 97–103.

[46] P. GRASSBERGER, T. SCHREIBER, AND C. SCHAFFRATH, *Nonlinear time sequence analysis*, International Journal of Bifurcation and Chaos, 1 (1994), pp. 521–547.

[47] M. R. GUPTA, S. BENGIO, AND J. WESTON, *Training highly multiclass classifiers*, Journal of Machine Learning Research, 15 (2014), pp. 1461–1492.

[48] J. D. HAMILTON, *A new approach to the economic analysis of nonstationary time series and the business cycle*, Econometrica, 57 (1989), pp. 357–384.

[49] P. R. HANSEN, Z. HUANG, AND H. H. SHEK, *Realized garch: A joint model for returns an realized measures of volatility*, Journal of Applied Econometrics, 27 (2012), pp. 877–906.

[50] P. R. HANSEN, A. LUNDE, AND V. VOEV, *Realized beta garch: A multivariate garch model with realized measures of volatility*, Journal of Applied Econometrics, 29 (2014), pp. 774–799.

[51] Z. HARCHAOUI AND O. CAPPE, *Retrospective change-point estimation with kernels*, in In IEEE Workshop on Statistical Signal Processing, Madison, WI, USA, 2007, 2007, pp. 768–772.

[52] M. R. HARDY, *A regime-switching model of long-term stock returns*, North American Actuarial Journal, 5 (2001), pp. 41–53.

[53] S. B. HARIZ, J. J. WYLIE, AND Q. ZHANG, *Optimal rate of convergence for nonparametric change-point estimators for nonstationary sequences*, The Annals of Statistics, 35 (2007), p. 1802–1826.

[54] T. HASTIE AND R. TIBSHIRANI, *Classification by pairwise coupling*, The Annals of Statistics, 26 (2001), pp. 451–471.

[55] J. HE, P. SHANG, AND H. XIONG, *Multidimensional scaling analysis of financial time series based on modified cross-sample entropy methods*, Physica A, 500 (2018), pp. 210–221.

[56] D. V. HINKLEY AND E. A. HINKLEY, *Inference about the change-point in a sequence of binomial variables*, Biometrika, 57 (1970), p. 477–488.

[57] K. HLAVÁČKOVÁ-SCHINDLER, M. PALUS, M. VEJMELKA, AND J. BHATTACHARYA, *Causality detection based on information-theoretic approaches in time series analysis*, Physics Reports, 441 (2000), pp. 1–46.

[58] A. HOOVER, A. SINGH, S. FISHEL-BRIWN, AND E. MUTH, *Real-time detection of workload changes using heart rate variability*, Biomedical Signal Processing and Control, 7 (2012), pp. 333–341.

[59] F. HSIEH, S. C. CHEN, AND C. R. HWANG, *Discovering stock dynamics through multidimensional volatility phases*, Quantitative Finance, 12 (2012), p. 213–230.

[60] ——, *Single stock dynamics on high-frequency data: From a compressed coding perspective*, PLoS ONE, 9 (2014), p. e85018.

[61] F. HSIEH, C. R. HWANG, H. C. LEE, Y. C. LAN, AND S. B. HORNG, *Testing and mapping non-stationarity in animal behavioral processes: a case study on an individual female bean weevil*, Journal of Theoretical Biology, 238 (2006), pp. 805–816.

[62] C. Inclan and G. C. Tiao, *Use of cumulative sums of squares for retrospective detection of change of variance*, Journal of the American Statistical Association, 89 (1994), p. 913–923.

[63] T. Isogai, *Clustering of japanese stock returns by recursive modularity optimization for efficient portfolio diversification*, J Complex Netw, 2 (2014), p. 557–584.

[64] Z. Kander and S. Zacks, *Test procedures for possible changes in parameters of statistical distributions occurring at unknown time points*, The Annals of Mathematical Statistics, 37 (1966), p. 1196–1210.

[65] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, *Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring*, IEEE Transactions on Information Technology in Biomedicine, 10 (2006), pp. 156–167.

[66] F. Kaspar and H. G. Schuster, *Easily calculable measure for the complexity of spatiotemporal patterns*, Physical Review A, 36 (1987), pp. 842–848.

[67] Y. Kawahara and M. Sugiyama, *Sequential change-point detection based on direct density-ratio estimation*, Statistical Analysis and Data Mining, 5 (2011), p. 114–127.

[68] D. Y. Kenett, X. Huang, I. Vodenska, S. Havlin, and H. E. Stanley, *Partial correlation analysis: applications for financial markets*, Quantitative Finance, 15 (2015), pp. 569–578.

[69] S. Khandelwal and N. Wickstrom, *Evaluation of the performance of accelerometer-based gait event detection algorithms in different real-world scenarios using the marea gait database*, Gait and Posture, 51 (2017), pp. 84–90.

[70] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, *Evaluation measures for hierarchical classification: a unified view and novel approaches*, Data Mining and Knowledge Discovery, 29 (2015), pp. 820–865.

[71] H. L. and A. P., *Comparing partitions*, Journal of Classification, 2 (1985), p. 193–218.

[72] H. A. L., *Minimally selected p and other tests for a single abrupt changepoint in a binary sequence*, Biometrika, 55 (1999), pp. 1044–1050.

[73] D. T. Lai, R. K. Begg, and M. Palaniswami, *Computational intelligence in gait research: a perspective on current applications and future challenges*, IEEE Trans. Inf. Technol. Biomed., 13 (2009), p. 687–702.

[74] C. W. Landsea, G. A. Vecchi, L. Bengtsson, and T. R. Knutsin, *Impact of duration thresholds on atlantic tropical cyclone counts*, Journal of Climate, 23 (2010), pp. 2508–2519.

[75] M. R. Leadbetter, *On a basis for 'peaks over threshold' modeling*, Statistics and Probability Letters, 12 (1991), p. 357–362.

[76] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86 (1998), pp. 2278–2324.

[77] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, *Change-point detection in time-series data by relative density-ratio estimation*, Neural Networks, 43 (2013), p. 72–83.

[78] J. C. M., G. Kaul, and L. M. L., *Transactions, volume, and volatility*, Review of Financial Studies, 7 (1994), pp. 631–651.

[79] R. Malladi, G. P. Kalamangalam, and B. Aazhang, *Online bayesian change point detection algorithms for segmentation of epileptic activity*, in In Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2013, 2013, pp. 1833–1837.

[80] B. Mandelbrot and H. M. Taylor, *On the distribution of stock price differences*, Operations Research, 15 (1967), pp. 1057–1062.

[81] R. Mantegna, *Hierarchical structure in financial markets*, The European Physical Journal B, 11 (1999), p. 193–197.

[82] R. Marschinski and H. Kantz, *Analysing the information flow between financial time series - an improved estimator for transfer entropy*, The European Physical Journal B, 30 (2002), p. 275–281.

[83] D. S. Matteson and N. A. James, *A nonparametric approach for multiple change point analysis of multivariate data*, Journal of the American Statistical Association, 109 (2014), pp. 334–345.

[84] N. Meinshausen and P. Buhlmann, *Stability selection*, Journal of the Royal Statistical Society. Series B, 72 (2010), p. 417–473.

[85] R. C. Merton, *Theory of rational option pricing*, The Bell Journal of Economics and Management Science, 4 (1973), pp. 141–183.

[86] R. Miller and D. Siegmund, *Maximally selected chi square statistics*, Biometrics, 38 (1982), pp. 1011–1016.

[87] M. Morse and G. A. Hedlund, *Symbolic dynamics*, American Journal of Mathematics, 60 (1938), pp. 815–866.

[88] V. M. Muggeo and G. Adelfio, *Efficient change point detection for genomic sequences of continuous measurement*, Bioinformatics, 27 (2011), pp. 161–166.

[89] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, *The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication*, Pattern Recognition, 47 (2014), pp. 228–237.

[90] A. B. Olshen and E. Venkatraman, *Segmentation for the analysis of array-based dna copy number data*, Biostatistics, 5 (2004), p. 557–572.

[91] E. Olshen, A. B. Venkatraman, *Segmentation for the analysis of array-based dna copy number data*, Biostatistics, 5 (2004), p. 557–572.

[92] E. S. Page, *Continuous inspection schemes*, Biometrika, 41 (1954), p. 100–115.

[93] A. N. Pettitt, *A simple cumulative sum type statistic for the change-point problem with zero-one observations*, Biometrika, 67 (1980), pp. 79–84.

[94] F. Picard, S. Robin, M. Lavielle, V. C., and D. J., *A statistical approach for array cgh data analysis*, BMC Bioinformatics, 6 (2005).

[95] L. R. Rabiner, *A tutorial on hidden markov models and selected applications in speech recognition*, Proceedings of the IEEE, 77 (1989), pp. 257–286.

[96] W. M. Rand, *Objective criteria for the evaluation of clustering methods*, Journal of the American Statistical Association, 66 (1971), p. 846–850.

[97] M. W. Robbins, R. B. Lund, G. C. M., and L. Q., *Changepoints in the north atlantic tropical cyclone record*, Journal of the American Statistical Association, 106 (2011), pp. 89–99.

[98] D. Rosenfield, E. Zhou, F. H. Wilhelm, A. Conrad, W. T. Roth, and A. E. Meuret, *Change point analysis for longitudinal physiological data: Detection of cardio-respiratory changes preceding panic attacks*, Biological Psychology, 84 (2010), p. 112–120.

[99] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (3nd ed.)*, Prentice Hall, Upper Saddle River, NJ, USA, 2009.

[100] A. S., C. A., and H. Z., *A kernel multiple change-point algorithm via model selection*, Journal of Machine Learning Research, 20 (2019), pp. 1–56.

[101] S. S., B. P., and B. D., *A genome-wide analysis of cpg dinucleotides in the human genome distinguishes two distinct classes of promoters*, Proceedings of the National Academy of Sciences of the United States of America, 103 (2006), pp. 1412–1417.

[102] L. J. Sandoval, *Structure of a global network of financial companies based on transfer entropy*, Entropy, 16 (2014), pp. 4443–4482.

[103] L. J. Sandoval and I. D. P. Franca, *Correlation of financial markets in times of crisis*, Physica A, 391 (2012), pp. 187–208.

[104] L. J. Sandoval, A. Mullokandov, and D. Y. Kenett, *Dependency relations among international stock market indices*, Journal of Risk and Financial Management, 8 (2015), pp. 227–265.

[105] A. Sant'Anna and N. Wickström, *Developing a motion language: Gait analysis from accelerometer sensor systems*, in 2009 3rd International Conference on Pervasive Computing Technologies for Healthcare, 2009, pp. 1–8.

[106] M. A. Saunders and A. S. Lee, *Large contributions of sea surface warming to recent increase in atlantic hurricane activity*, Nature, 451 (2008), pp. 557–560.

[107] T. Schreiber, *Measuring information transfer*, Physical Review Letters, 85 (2000), p. 461–464.

[108] S. Shirota, Y. Omori, H. F. Lopes, and H. Piao, *Cholesky realized stochastic volatility model*, Economics and Statistics, 3 (2017), pp. 34–59.

[109] J. Solomon, *Optimal transport on discrete domains*, (2018). arxiv.org/abs/1801.07745.

[110] S. Sprager and M. B. Juric, *Inertial sensor-based gait recognition: A review*, Sensors, 15 (2015), pp. 22089–22127.

[111] M. Talih and N. Hengartner, *Structural learning with time-varying components: tacking the cross-section of financial time series*, Journal of the Royal Statistical Society. Series B, 67 (2005), pp. 321–341.

[112] A. Tenyakov, *Estimation of Hidden Markov Models and Their Applications in Finance*, PhD thesis, The University of Western Ontario, 2014. https://ir.lib.uwo.ca/etd/2348.

[113] G. Trivino, A. Alvarez-Alvarez, and G. Bailador, *Application of the computational theory of perceptions to human gait pattern recognition*, Pattern Recognition, 43 (2010), pp. 2572–2581.

[114] C. Y. Tsai and C. C. Chiu, *Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm*, Computational Statistics and Data Analysis, 52 (2008), pp. 4658–4672.

[115] Y. K. Tse and A. K. C. Tsui, *A multivariate garch model with time-varying correlations*, Journal of Business and Economic Statistics, 20 (2002), pp. 351–362.

[116] G. Tsoumakas, I. Katakis, and I. Vlahavas, *A review of multi-label classification methods*, in In Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery (ADMKD 2006), 2006, pp. 99–109.

[117] A. J. Viterbi, *Error bounds for convolutional codes and an asymptotically optimal decoding algorithm*, IEEE Transactions on Information Theory, 13 (1967), p. 260–269.

[118] L. Vostrikova, *Detection disorder in multidimensional random processes*, Soviet Mathematics Doklady, 24 (1981), pp. 55–59.

[119] X. Wang and F. Hsieh, *Discovering multiple phases of dynamics by dissecting multivariate time*, (2021). arxiv.org/abs/2103.04615.

[120] ——, *Unraveling s&p500 stock volatility and networks - an encoding and decoding approach*, (2021). arxiv.org/abs/2101.09395.

[121] K. Q. Weinberger and O. Chapelle, *Large margin taxonomy embedding for document categorization*, in Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008, Curran Associates, Inc., 2008, pp. 1737–1744.

[122] K. Q. Weinberger and L. K. Saul, *Distance metric learning for large margin nearest neighbor classification*, Journal of Machine Learning Research, 10 (2009), pp. 207–244.

[123] M. W. Whittle, *Gait Analysis: An Introduction 4th ed.*, Butterworth-Heinemann, Edinburgh, UK, 2008.

[124] D. A. Winter, *The Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological*, University of Waterloo, Waterloo, ON, Canada, 1991.

[125] W. Y. and L. F. C., *An evaluation of new criteria for cpg islands in the human genome as gene markers*, Bioinformatics, 20 (2004), p. 1170–1177.

[126] Y. Yamaucgi and Y. Omori, *Multivariate stochastic volatility model with realized volatilities and pairwise realized correlations*, Journal of Business and Economic Statistics, 38 (2020), pp. 839–855.

[127] Y.-C. Yao, *Estimating the number of change-points via schwarz' criterion*, Statistics and Probability Letters, 6 (1988), p. 181–189.

[128] C. C. Ying, *Stock market prices and volumes of sales*, Econometrica, 34 (1966), pp. 676–685.

[129] L. A. Zadeh, *Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic*, Fuzzy Sets Syst., 90 (1997), pp. 111–127.

[130] C. Zou, G. Yin, L. Feng, and Z. Wang, *Nonparametric maximum likelihood approach to multiple change-point problems*, The Annals of Statistics, 42 (2014), p. 970–1002.