國立臺灣大學理學院統計與數據科學研究所

碩士論文

Institute of Statistics and Data Science

College of Science

National Taiwan University

Master's Thesis

從多變量方法延伸至函數型數據的線上轉折點偵測

Online Change Point Detection in Function Data: An Extension from Multivariate Techniques

吳岱錡

Dai-Chi Wu

指導教授: 陳裕庭 博士

Advisor: Yu-Ting Chen Ph.D.

中華民國 113 年 8 月

August, 2024

doi:10.6342/NTU202403079

Acknowledgements

從開始閱讀文獻和設計模擬實驗，到後來提出研究方法、修正思路及撰寫論文的每一步，都離不開我的指導教授陳裕庭教授的悉心指導。陳教授在整個研究過程中，無論是初期的文獻搜索還是後期的模擬設計和論文撰寫，他都給予了無數寶貴的建議和幫助。他的耐心指導和豐富的經驗協助我克服了許多研究中的挑戰，對此我由衷地表示我的感謝。

特別感謝我的家人和朋友們，他們的支持和鼓勵給了我無比的信心和動力。每當我遇到困難和挫折時，他們總是給予我無限的關懷和幫助，讓我能夠堅持到底。

此外，我也要感謝我同為統計所的同學們，他們在我研究過程中的討論溝通讓我受益匪淺。與他們的交流不僅啟發了我的思路，也讓我在研究過程中體會到了群體智慧的力量。

最後，感謝參與口試的楊鈞澔教授及李百齡教授，您們的寶貴建議和意見使我的論文更加完善和嚴謹。正是因為有了您們的耐心聆聽與詳細指點，我的研究才能順利完成。

再次感謝所有在我研究過程中給予我幫助和支持的人，這篇論文凝聚了大家的智慧和心血，對此我深感榮幸和感激。

doi:10.6342/NTU202403079

摘要

函數性數據，例如醫學領域中的心電圖 (ECG) 信號，或氣象領域中隨時間記錄的天氣變量，通常以連續且無限維度的曲線形式呈現，已成為當今最常見的數據形式之一。然而，儘管其重要性日益增加，針對這類數據的線上轉折點偵測 (OCPD) 方法在現有文獻中的討論相對有限。相比之下，多變量數據則擁有許多已發展成熟的 OCPD 方法。因此，我們在仔細審視現有文獻後，決定將四種無分配假設的多變量 OCPD 方法延伸應用至函數性數據，希望能有效地檢測出連續數據流中的異常。模擬實驗中，我們使用兩個常用的基準——平均運行長度和平均檢測延遲——來評估這些方法的表現，並通過這兩個指標來驗證其可靠性和效率。最後，我們通過比較這些改進方法之間的效能差異，並進一步探討導致這些差異的可能因素。這項研究不僅為函數性數據的線上監控系統的發展作出了貢獻，也為實時分析函數性數據提供了有價值的見解和一些潛在可行的方法。

關鍵字：函數型數據、線上轉折點分析

doi:10.6342/NTU202403079

Abstract

Functional data, such as Electrocardiogram (ECG) signals in the medical field, or weather-related variables recorded over time in the meteorological field, are often presented as continuous and infinite-dimensional curves, and have become a common form of data. However, despite its relevance, online change point detection (OCPD) for this datatype has received limited attention in the existing literature. On the other hand, multivariate data have plenty of its own OCPD methods; therefore, after a thorough survey, we decided to extend four nonparametric multivariate OCPD methods to accommodate functional data characteristics, aiming to successfully detect those anomalies in continuous data streams. Eventually, we evaluate the performance of these adapted methods against two benchmarks commonly employed in online settings: average run length and average detection delay. These metrics provide insights into the reliability and efficiency of these methods. Our work compares the performance between the extended methods and native functional data OCPD techniques via simulations, and further discusses the differences. This research contributes to the ongoing development of robust online monitoring tools for functional data, and offers valuable perspectives for potential method candidates and practical implementations in real-time analysis for functional data.

Keywords: Functional data, Online Change-Point Detection

doi:10.6342/NTU202403079

Contents

doi:10.6342/NTU202403079

doi:10.6342/NTU202403079

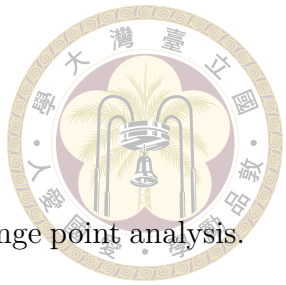
Chapter 1 Introduction

The ubiquity of online change point analysis in data streams has become a focal point in modern statistical research due to the dynamic nature of countless real-time data acquisitions nowadays. For instance, in the field of environmental monitoring, data are continuously collected from sensors based on various indicators, such as air quality or water pollution levels. Detecting change points in those collected data can alert people to investigate the reasons behind the sudden shifts in environmental conditions, which further enables prompt responses to prevent potential hazards.

However, the inherent feature of functional data, which often involve high-dimensional structures and complex dependencies, pose significant challenges to the development of Online Change Point Detection (OCPD) methods within this domain. The intricacies of these data require sophisticated algorithms that can handle the continuous arrival of information and identify changes in real-time without significant delays. Furthermore, the fact that current OCPD methods available for functional data are scarce reflects the difficulty of designing tools that are both efficient and reliable in such settings. Thus, we aim to derive an effective method for functional data on the foundation of multivariate data counterparts.

doi:10.6342/NTU202403079

1.1 Online VS. Offline



First, we contrast the core objectives of online and offline change point analysis. Online change point detection is operated in real-time, which means data are processed sequentially as soon as they become available. Therefore, those circumstances required an immediate response, such as anomaly detection in manufacturing processes or intrusion detection in network security. As stated above, one main focus of online detection is its need to take actions promptly, which is often assessed by expected detection delay (EDD). Nevertheless, another emphasis is on false alarm rate, which needs to be controlled in terms of average run length (ARL).

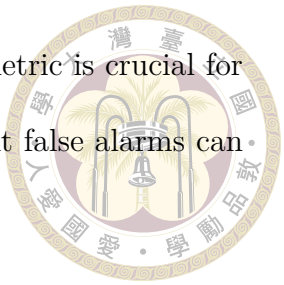
Offline change point detection, on the other hand, analyzes a complete set of data retrospectively. This approach is used in situations where precision is more critical than immediacy, such as analyzing historical climate data or conducting market research. Offline methods often utilize the entire dataset to identify change points, allowing for more complex computations and thorough analysis, with the main goal being the accurate detection of these change points.

1.2 ARL & EDD

Average Run Length (ARL) refers to the average number of observations taken between false alarms, which occurs when a algorithm alerts to a change that did not happened. Essentially, it reflects the algorithm's ability to avoid false alarms under the assumption that no change is present. In other words, a higher ARL indicates a lower false alarm rate, suggesting that the detection method is less prone

doi:10.6342/NTU202403079

to misinterpret small fluctuations in the data as changes. This metric is crucial for maintaining the credibility of the detection system, since frequent false alarms can simultaneously break user trust and lead to inefficiency.



On the other hand, Expected Detection Delay (EDD) measures how quickly a detection method identifies a change point after it occurs. It is typically defined as the average number of observations required from the actual change point to the moment the change is detected by the algorithm. A shorter EDD indicates a more responsive detection system, capable of rapidly responding to changes in the data stream, which is vital in scenarios where a swift response to changes is critical.

1.3 Review of methods

As mentioned above, there are quite a few offline methods for functional data, e.g., those addressing At Most One Change (AMOC) issues, such as the CUSUM test suggested in Berkes et al. (2009) on the constancy of the mean function for independent functional data and the extension of the previous test to weakly dependent functional data by Hörmann and Kokoszka (2010), and a fully functional procedure revealing mean breaks without the application of dimension reduction techniques in Aue et al. (2018). Additionally, for multiple change point problems, for instance, Chiou et al. (2019) introduced the Dynamic Segmentation and Backward Elimination (DSBE) algorithm to find the optimal set of change points in the mean functions, and Harris et al. (2020) combined robust segmentation through an augmented fused lasso procedure with optimal detection via a powerful cumulative sum (CUSUM) statistic. On the contrary, there is not much available for the online

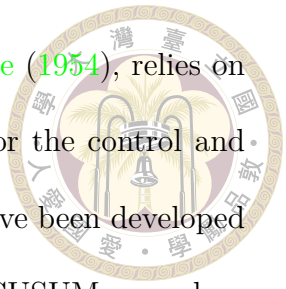
doi:10.6342/NTU202403079

setting.

As a result, we surveyed a variety of multivariate OCPD techniques, in hopes that through some straightforward modifications, they could accommodate the intrinsic continuity and complexities of functional observations, i.e., could be extended to the realm of functional data. Our work aims to provide some intuitive thoughts for extending OCPD methods designed for multivariate data to functional data, by standing on the shoulders of giants.

Through our investigations, we found that numerous methods have been designed for specific types of change. For example, Chen et al. (2022) and Chena et al. (2023) addressed changes in the mean of a p-variate Gaussian data streams. Other scenarios include changes in linear regression models (Geng et al., 2019), in high-dimensional covariance structure (Li and Li, 2023), in sequences of distributions (Horvath et al., 2021), in sensor networks under adversarial attacks (Fellouris et al., 2018), and in situations where both the pre-change and post-change distributions involve unknown parameters (Mei, 2006). In addition, some methods have been proposed to cope with certain data types; for instance, Xie and Siegmund (2013) developed a mixture procedure to monitor parallel data streams for a change point which affects only a subset of them, often sparsely. Several other methods address different variants of data or topics, such as G¨osmann et al. (2022), which concentrates on high-dimensional time series, and Lin et al. (2023), which deals with high-dimensional dynamic systems; moreover, topics like social networks were explored in Raginsky et al. (2012). Moving forward, we will primarily focus on more general methods that are not tailored to specific task, aiming to identify method candidates that could potentially be extended to function data.

doi:10.6342/NTU202403079

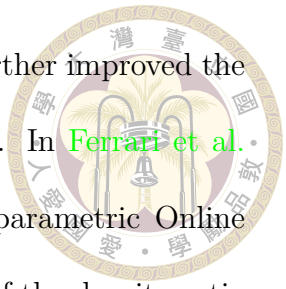


CUSUM, as a classic approach first introduced by Page Page (1954), relies on the log-likelihood ratio between two known distributions: one for the control and one for the anomaly. Since its introduction, plenty of methods have been developed on this foundation. For instance, Cao and Xie (2017) formed a CUSUM procedure after solving a convex optimization problem to identify appropriate parameters for pre- and post-change distributions. Xie et al. (2020) proposed Subspace-CUSUM, Kurt et al. (2021) provided a CUSUM anomaly detection algorithm based on some univariate summary statistics without any restrictive model assumptions on both the high-dimensional data stream and the extracted summary statistics. Rauhameri et al. (2022) discussed the multivariate Max-CUSUM algorithm and proposed Matrix Form CUSUM algorithm, which significantly shortens computation time. Lastly, Window-Limited CUSUM was suggested in Xie et al. (2023).

Another classic approach is the Generalized Likelihood Ratio (GLR) statistic-based procedure, which finds the maximum likelihood estimate (MLE) of the post-change parameter and inserts it back into the likelihood ratio to form the test statistic. For instance, Cao et al. (2018) developed an online mirror descent-based GLR procedure to update the estimate of the unknown post-change parameter with arriving data, and Cao et al. (2019) used GLR statistics to form linear sketches for high-dimensional data, where sketching is a common strategy for reducing data dimensionality.

Aside from the nonparametric algorithm (Kurt et al., 2021) above, there still are many types of distribution-free change detection technique developed with various tools. For example, Austin et al. (2023) introduced Non-Parametric UNbounded Changepoint (NUNC) testing for a change in the empirical cumulative distribution

doi:10.6342/NTU202403079



function (eCDF) within a rolling window. Recently, they have further improved the non-parametric procedure NUNC by building a GLR test for it. In Ferrari et al. (2023), a kernel-based OCPD method namely NOUGAT (Nonparametric Online chanGepoint detection AlgoriThm) is built on direct estimation of the density ratio over consecutive time intervals. Additionally, Guo and Modarres (2022) proposed two nonparametric algorithms consisting of energy statistic and Mahalanobis depth, where the former is applied using a sliding window algorithm with efficient training and updating procedures, and the latter is combined with an algorithm-determined threshold that offers the desired protective ability against false alarms. Last but not least, Chen et al. (2016) construct a series of conditionally distribution-free test statistics to monitor the location parameters.

In Wang and Xie (2023), Xie et al. (2021), and Lai (1995), numerous classical sequential OCPD methods were reviewed; for instance, all three cover procedures like CUSUM, Shiryaev-Roberts, GLR, and Shewhart chart. While Xie et al. (2021) surveyed a wide array of extensions and modern applications of sequential CPD, and Wang and Xie (2023) discussed the trade-off between computation and statistical power, more importantly, both provide insights into previous studies on nonparametric multivariate OCPD methods that don't require explicit distribution assumptions. These approaches might meet our needs, such as the *scan* $B-$statistic formed via kernel maximum mean discrepancy (MMD) in Li et al. (2019).

On the other hand, similarity graph is also a useful strategy in the category of distribution-free methods. For example, Chen (2019) and Chu and Chen (2022) proposed a K-nearest-neighbors-based statistic to detect changes in sequences of multivariate observations or non-Euclidean data objects, such as network data. Marenco

doi:10.6342/NTU202403079

et al. (2022) integrates the directed and weighted Random Dot Product Graphs (RDPG) models for OCPD, and Chen and Chu (2023) reviewed a nonparametric change point analysis framework that utilizes graphs to represent the similarity between observations.

1.4 Organization

The rest of the article is organized as follows: In Section 2, we propose four adapted detection procedures, and the modifications that need to be made before implementation. Simulation settings and method comparisons are presented in Section 3. The paper is concluded in Section 4.

doi:10.6342/NTU202403079

Chapter 2 Adapted Methods

As suggested in Section 1, we surveyed a variety of existing multivariate OCPD methods. Next, we consider some of these methods in hopes of extending them to functional data. The results in Section 3 show that even though these methods are not specifically designed for functional data, they can still be applicable with certain modifications.

In Section 2.1, we first consider a straightforward method—the control chart —by directly treating our functional data as multivariate inputs. From Section 2.2 to Section 2.4, we explore methods closely related to distance metrics to distinguish dissimilarities. Specifically, Section 2.2 and Section 2.3 are based on kNN graphs , while Section 2.4 utilizes a distance-based statistic to signal a change. As for the distance measure, owing to the better performance of Meng et al. (2018) over Euclidean distance in our simulation results of Section 2.2, we believe this derivative information-inspired distance metric can better capture the characteristics of functional data. Thus, we adopt the distance concept from Meng et al. (2018) for the methods following this approach, with definitions provided in Section 2.2.

Apart from the control chart method in Section 2.1, other methods follow the same problem settings, which can be formulated as follows: Denote the functional

doi:10.6342/NTU202403079

data sequence by $\{X_i\}_{i=1}^{\infty}$, we consider the null and alternative hypotheses:

$$H_0: \quad X_i \sim F_0 \quad \text{for } i = 1, 2, \ldots,$$

$$H_1: \quad X_i \sim F_0 \quad \text{for } i < \tau, \quad \text{and } X_i \sim F_1 \quad \text{for } i \geq \tau$$

where F_0, F_1 are different probability measures, and τ is the location of the change point.

Throughout these methods, for the sake of comparison, the parameter settings are all adjusted to achieve the same in-control run length (ICRL) level. The specific settings will be elaborated on in their respective methodologies.

2.1 Control Chart

First of all, when each of the functional data in the sequence is observed at the same time points, we can simply view the data set as multivariate data. And since we're seeking methods that can handle high-dimensional data ($p > n$), the most intuitive approach is to use a control chart for our OCPD problem. Particularly in this technique, we manage our potential high-dimensional data by sequentially applying conditionally distribution-free two-sample tests to construct a monitoring system.

Consider a slightly different change point model with historical observations $X_1, \ldots, X_{N_0}$ in $\mathbb{R}^p$, $p \geq 1$; the ith observation, $\mathbf{X}_i = (X_{1i}, \ldots, X_{pi})^T$, follows the

doi:10.6342/NTU202403079

multivariate location change point model:

$$X_i \overset{\text{iid}}{\sim} \begin{cases} F_0(x;\mu_0) & \text{for } i = 1,\ldots,\tau, \\ F_1(x;\mu_1) & \text{for } i = \tau+1,\ldots, \end{cases}$$

where $\mu_0 = (\mu_{10},\ldots,\mu_{p0})^T$ and $\mu_1 = (\mu_{11},\ldots,\mu_{p1})^T$ are location parameters, and τ is the unknown change point. When the (N_0+n)th observation is collected, we can then construct a location test. Denote $\mathcal{X}_{k,j}^{n} = \{X_{jk},\ldots,X_{jn}\}$ as the jth component of the segment from index k to n. A charting statistic can be constructed as $T_n(w,\lambda) = \sum_{j=1}^{p} T_{jn}^2(w,\lambda)$, where

$$T_{jn}(w,\lambda) = \sum_{i=n-w+1}^{n} (1-\lambda)^{n-i} \frac{R_{jni} - w\left(N_0+n+1\right)/2}{\sqrt{w\left(N_0+n+1\right)\left(N_0+n-w\right)/12}}$$

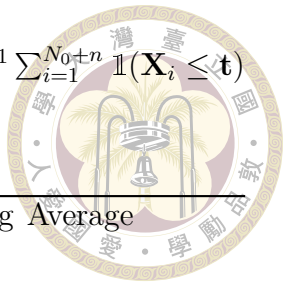
Here, w is the window size, and λ is the smoothing parameter, chosen to balance robustness to nonnormality and detection ability for various shift magnitudes. R_{jni} is the rank of X_{ji} among all current observations $\mathcal{X}_{1,j}^{N_0+n}$. Naturally, a large $T_n(w,\lambda)$ value will lead to the rejection of H_0. Essentially, $T_{jn}(w,\lambda)$ is a weighted version of the two-sample Wilcoxon rank-sum statistic for testing the equality of the locations of the sample $\mathcal{X}_{1,j}^{N_0+n-w}$ and $\mathcal{X}_{N_0+n-w+1,j}^{N_0+n}$. Different rank observations in $T_{jn}(w,\lambda)$ are weighted as in a common exponentially weighted moving average (EWMA) chart, meaning more recent observations receive greater weight, which decays exponentially over time.

Meanwhile, the control limits $H_n(\alpha)$'s are determined by solving the equation:

$$\Pr\left(T_n(w,\lambda) > H_n(\alpha) \mid T_i(w,\lambda) < H_i(\alpha), \max\{1, n-w+1\} \le i < n, \hat{F}_n\right) = \alpha$$

doi:10.6342/NTU202403079

where α is the pre-specified false alarm rate and $\hat{F}_n(\mathbf{t}) = (N_0+n)^{-1}\sum_{i=1}^{N_0+n}\mathbb{1}(\mathbf{X}_i \leq \mathbf{t})$ is the empirical CDF. The algorithm is then provided as follows:



Algorithm 1: Distribution-Free Exponetially Weighted Moving Average (DFEWMA)

Data: Dataset $\{\mathbf{X}_1, \ldots, \mathbf{X}_{N_0}\}$, parameters w, λ, b, α
Result: Threshold values $\hat{H}_1(\alpha), \ldots, \hat{H}_n(\alpha)$

for $n = 1$ **do**
  Generate a random permutation of $\{1, \ldots, N_0\}$, say $\{i_1, \ldots, i_{N_0}\}$.
  Obtain the corresponding $T_1^v(w, \lambda)$ based on permuted samples $\{\mathbf{X}_{i_1}, \ldots, \mathbf{X}_{i_{N_0}}\}$.
  Repeat this procedure b times to obtain $T_1^1(w, \lambda), \ldots, T_1^b(w, \lambda)$.
  Find the threshold value, $\hat{H}_1(\alpha)$, by the $(1-\alpha)$ empirical quantile from samples $T_1^v(w, \lambda), v = 1, \ldots, b$.
for $n > 1$ **do**
  while number of valid $T_k^v(w, \lambda)$ samples $< b$ **do**
    Generate a random permutation and calculate the test statistics $T_k^v(w, \lambda)$, $\max\{1, n-w+1\} \leq k < n$.
    Determine the control limit $H_k(\alpha)$ by solving $\Pr\left(T_k^v(w, \lambda) > H_k(\alpha) \mid T_i(w, \lambda) < H_i(\alpha), \max\{1, k-w+1\} \leq i < k, \hat{F}_k\right) = \alpha$
    if $T_k^v(w, \lambda) < H_k(\alpha)$ **then**
      Calculate $T_n^v(w, \lambda)$.
    else
      Discard this permutation.
  Find the limit $\hat{H}_n(\alpha)$ as the $(1-\alpha)$ empirical quantile from samples $T_n^v(w, \lambda), v = 1, \ldots, b$.
  if $T_n(w, \lambda) \geq \hat{H}_n(\alpha)$ **then**
    Stop the iteration and declare a change at $t = n$.

Since the ICRL distribution of DFEWMA follows a Geometric distribution, the average run length of the proposed chart is $1/\alpha$ when there is no change. For instance, if the desired ICRL is 500, the false alarm rate α should be set to 0.002. Although primarily designed for detecting location changes, this method can be adapted to other scenarios, as demonstrated by our simulations.

Instead of directly using the entire dataset as the algorithm's input, we also

doi:10.6342/NTU202403079

considered performing functional principal component analysis (FPCA) on the historical functional data beforehand and transforming the upcoming data using the initially obtained eigenfunctions. In this way, we successfully use the FPC scores, which is of the form of multivariate data, as proxies for the original functional data. However, possibly due to the different weights in these FPC estimates, when they are treated as general multivariate input and ranks are computed accordingly, the results appear unsatisfactory. Thus, this approach proved unsuitable for this control chart technique.

Additionally, since the core concept involves combining the ranks of different components, we recall that functional data have a wide variety of depth measures designed for different scenarios, which can be unified to create different ranks. Since no single notion of functional depth consistently outperforms the others in all situations, we strive to utilize all the information gathered from these various data depths. Thus, we ultimately chose five functional data depths to transform the original data points into a new dataset, which yielded viable results. These depths include extremal depth (ED) (Narisetty and Nair, 2016), linfinity depth (L^∞D) (Long and Xie, 2016), modified band depth (MBD) (López-Pintado and Romo, 2009), projection depth (PD) (Zuo and Serfling, 2000), and total variation depth (TVD) (Huang and Sun, 2019), with their corresponding definitions provided in the appendix. By integrating the strengths of these various depths, we aim to detect online change points more effectively than with the original algorithm.

In this variant, we first input the algorithm with the respective data depths of the nominal dataset. Each time a new observation arrives, we update the data depths of the entire current data and compute their ranks accordingly. This approach yields

doi:10.6342/NTU202403079

superior results compared to the original DFEWMA control chart.

2.2 Graph-Based Method

As mentioned in previous section, distribution-free methods, such as graph-based methods, are an effective strategy to describe the dissimilarities between observations. Therefore, we will consider the methods described in Chen (2019) and Chu and Chen (2022) in the following adapted method.

Denote the functional data sequence as $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n, \ldots$, where $\mathbf{X}_n$ represents the observation at time n. It is assumed that there are N_0 historical observations with no change point. To introduce the main stopping rule from Chen (2019), we focus on the L most recent observations: $\mathbf{X}_{n-L+1}, \ldots, \mathbf{X}_n$, and refer to the number of edges in the k nearest neighbors between two observations as edge-counts.

That is, for $i, j \in n_L \triangleq \{n-L+1, \ldots, n\}$, we let $A^+_{n_L,ij} = \sum_{r=1}^{k} A^{(r)}_{n_L,ij}$, where $A^{(r)}_{n_L,ij} = \mathbb{1}(X_j \text{ is the } r\text{th NN of } X_i \text{ among observations } X_{n-L+1}, \ldots, X_n)$. And the between-sample edge-count is defined as

$$R_L(t,n) = \sum_{i,j \in n_L} \left(A^+_{n_L,ij} + A^+_{n_L,ji} \right) B_{ij}\left(t, n_L\right)$$

with $B_{ij}\left(t, n_L\right) = b_{\mathbf{P}_{n_L}(i)\mathbf{P}_{n_L}(j)}(t)$, where $b_{ij}(t,n) = \mathbf{I}((i \le t, t < j \le n) \text{ or } (t < i \le n, j \le t))$ and $\mathbf{P}_{n_L}(\cdot)$ is a random permutation among indices $\{n-L+1, \ldots, n\}$. $R_L(t,n)$ quantifies the number of connections between two observations when they

doi:10.6342/NTU202403079

are each other's top k nearest neighbors, yet they belong to different groups. Let

$$Z_{L|\mathbf{x}}(t,n) = -\frac{R_L(t,n) - \mathrm{E}\left(R_L(t,n)\right)}{\sqrt{\mathrm{Var}\left(R_L(t,n) \mid \mathbf{x}\right)}}$$

Therefore, a large value of $R_L(t,n)$ suggests a substantial number of edge-counts between two groups, implying that the disparity between the historical observations and the recent L observations is not particularly significant. This, in return, would lead to a smaller value of $Z_{L|\mathbf{x}}(t,n)$. We then use it to construct the stopping rule:

$$T_Z\left(b_Z\right) = \inf\left\{n - N_0 : \left(\max_{n-n_1 \le t \le n-n_0} Z_{L|\mathbf{x}}(t,n)\right) > b_Z, n \ge N_0\right\}$$

where n_0, n_1 and L are pre-specified values. Specifically, n_0 is set to be small to detect the change as soon as possible, but not too small to avoid the high fluctuations at the very ends. n_1 is set to $L - n_0$ as suggested by the paper. While b_Z is chosen such that, given the stopping rule $T_Z(b_Z)$, the average run length $E_\infty(T_Z(B_Z))$ meets a pre-specified value, i.e., the false discovery rate is controlled at a predetermined level. Here, $E_\infty(\cdot)$ denotes the average run length determined by the stopping rule under the assumption of no change.

However, $Z_{L|\mathbf{x}}(t,n)$ may fail to function properly when facing certain circumstances. Hence, Chu and Chen (2022) suggested three new stopping rules, with still using the concept of directed nearest neighbor graphs from Chen (2019) to construct the three improved two-sample test statistics, namely $S_{L|\mathbf{x}}(t,n)$ from the Generalized Edge-Count Test (S), $W_{L|\mathbf{x}}(t,n)$ from the Weighted Edge-Count Test (W) and $M_{L|\mathbf{x}}(t,n)$ from the Max-Type Edge-Count Test (M). While $S_{L|\mathbf{x}}(t,n)$ is designed to capture the deviation in the within-sample edge-counts from its null expectation,

doi:10.6342/NTU202403079

$W_{L|\mathbf{x}}(t,n)$ is the standardized value of the weighted combinations of the within-sample edge-counts $R_{1,L}(t,n)$ and $R_{2,L}(t,n)$. On the other hand, $M_{L|\mathbf{x}}(t,n)$ is the maximum between $\xi W_{L|\mathbf{x}}(t,n)$ ($\xi \geq 0$, a predetermined value), and the standardized value of the difference between $R_{1,L}(t,n)$ and $R_{2,L}(t,n)$. Their respective stopping threshold b_S, b_W and b_M are also computed analytically to ensure that the ARL for each stopping rule is controlled at a given value.

Essentially, $S_{L|\mathbf{x}}(t,n)$ and $M_{L|\mathbf{x}}(t,n)$ are designed to address the problem of curse-of-dimensionality, while $W_{L|\mathbf{x}}(t,n)$ resolves the issue of increased detection delay resulting from a variance boosting problem. The rest of the detailed definition is in Chu and Chen (2022). In addition, $M_{L|\mathbf{x}}(t,n)$ is expected to perform best among these statistics, as it can cope with location and scale changes simultaneously and offers a more accurate analytical expression for the ARL for false discovery control.

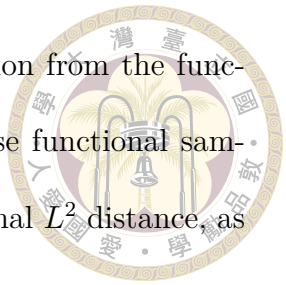
Since the essence of this graph-based method lies in calculating the dissimilarities between observation x_i and x_j to define nearest neighbors, we propose adapting this approach to functional data by implementing distance measures particularly established for functional samples, rather than relying on the default L^2 distance $d\left(x_i, x_j\right) = \sqrt{\int (x_i(t) - x_j(t))^2 dt}$.

First, we utilize a novel distance measure proposed by Meng et al. (2018), defined as

$$d(x_i(t), x_j(t)) = \sqrt{\int_T (x_i(t) - x_j(t))^2 dt + \int_T (Dx_i(t) - Dx_j(t))^2 dt} \tag{2.1}$$

which serves as the similarity metric between two functional samples $x_i(t)$ and $x_j(t)$, where the $Dx_i(t)$ represents the first-order derivative of the ith functional sample.

doi:10.6342/NTU202403079



The distance metric above incorporates derivative information from the functional curves, which intuitively corresponds to the shape of these functional samples. Therefore, this measure is expected to outperform the original L^2 distance, as demonstrated in Section 3.

2.3 GEM Statistics

The third method from Kurt et al. (2021) involves extracting Geometric Entropy Minimization (GEM)-based summary statistics from the collected data for anomaly detection. Anomalies are identified as persistent outliers in these univariate summary statistics, which accumulate over time to indicate a change, falling outside the acceptance region defined by α.

The GEM method relies on bipartite kNN graphs derived from the dataset. It distinguishes anomalous data by comparing the nearest neighbor statistics of nominal data with newly arrived data. Given a nominal dataset X of size N_0 and a chosen significance level α, the dataset is uniformly partitioned into two subsets, S_1 and S_2, with sizes N_1 and $N_2 = N_0 - N_1$, respectively.

For each observation $x_j \in S_2$, the kNNs among set S_1 are determined. Denoting the distance obtained using equation (2.1) from x_j to its ith nearest neighbor in S_1 by $e_j(i)$, the sum of distances from x_j to its kNNs is computed as follows:

$$d_j \triangleq \sum_{i=1}^{k} e_j(i) \tag{2.2}$$

After collecting $\{d_j : x_j \in S_2\}$, these distances are sorted in ascending order, and the largest α fraction of d_j's corresponds to the reject region. Whenever a new

doi:10.6342/NTU202403079

data point x_t arrives, its sum of distances to its kNNs among S_1, denoted by d_t, is compared to the smallest $(1-\alpha)$ fraction of d_j's. The point x_t is then considered an outlier if:

$$\frac{\sum_{x_j \in S_2} \mathbb{1}\{d_t > d_j\}}{N_2} > 1 - \alpha$$

where $\mathbb{1}\{\cdot\}$ is an indicator function.

For online change point detection, we seek to declare an anomaly if such outliers are consistently detected. Consequently, we denote $\hat{p}_t$ as the fraction of nominal summary statistics $\{d_j : x_j \in S_2\}$ greater than d_t. If $\hat{p}_t < \alpha$, x_t is considered an outlier at the pre-specified significance level α. Finally, let:

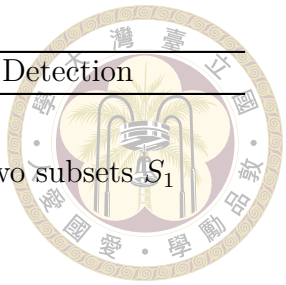
$$\hat{s}_t \triangleq log(\frac{\alpha}{\hat{p}_t})$$

Hence, if x_t is an outlier, we have $s_t > 0$, and for a non-outlier x_t, $s_t \leq 0$. The GEM-statistics CUSUM change point detection algorithm is then formulated as follows:

$$\Gamma = inf\{t : g_t \geq h\}, \quad g_0 = 0, \quad g_t = max\{0, g_{t-1} + \hat{s}_t\}.$$

Pseudo code is then:

doi:10.6342/NTU202403079



Algorithm 2: GEM-Based Real-Time Nonparametric Anomaly Detection

Offline Phase:

Uniformly randomly partition the nominal dataset $\mathcal{X}$ into two subsets S_1 and S_2 with sizes N_1 and N_2, respectively.

for $j : x_j \in S_2$ **do**

  Search for the kNNs of x_j among the set S_1.

  Compute d_j using equation (2.2).

end

Sort $\{d_j : x_j \in S_2\}$ in ascending order.

Online Detection Phase:

Initialization: $t \leftarrow 0$, $g_0 \leftarrow 0$.

while $g_t < h$ **do**

  $t \leftarrow t + 1$. Obtain the new data point x_t.

  Search for the kNNs of x_t among the set S_1 and compute d_t using equation (2.2).

  $\hat{p}_t \leftarrow \frac{1}{N_2} \sum_{x_j \in S_2} \mathbb{1}\{d_j > d_t\}$.

  $\hat{s}_t \leftarrow \log(\alpha/\hat{p}_t)$.

  $g_t \leftarrow \max\{0, g_{t-1} + \hat{s}_t\}$.

end

Declare an anomaly and stop the procedure.

The algorithm shows that if the sequentially acquired data consistently deviate from the nominal data, it is likely that an anomaly has occurred. The statistic g accumulates the measure of outlierness, and a change is thereby declared when g surpasses a certain threshold h.

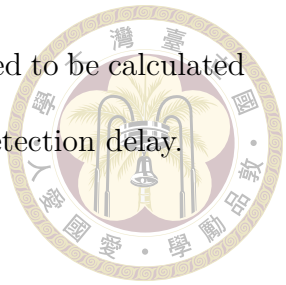
For the test threshold h, we use the analytical solution under the assumption that $N_2 \to \infty$, meaning the nominal dataset is sufficiently large. Here, A is the pre-specified ARL level:

$$h = \frac{log(A/g(\alpha))}{1 - W(\alpha g(\alpha))/g(\alpha)},$$

where $W(c)$ denotes the Lambert-W function providing solution z to the equation $ze^z = c$; while the values of $g(\alpha)$ are derived from the Monte Carlo simulation results in Kurt et al. (2021). In this case, with $\alpha = 0.2$, $g(\alpha)$ is determined to be 10.1.

doi:10.6342/NTU202403079

This method is not only easy to compute—since d_j values need to be calculated only once—but it also claimed to have a relative low expected detection delay.



2.4 Energetic Statistic

Next, we consider a nonparametric two-sample method presented in Guo and Modarres (2022), which incorporates a sliding window algorithm with the energetic statistic proposed in GáborJ.Székely and MariaL.Rizzo (2013). The core concept is to utilize the sliding window to perform homogeneity tests over the Baseline window $\mathscr{B}$ and the Current window $\mathscr{C}_t$, with the statistic based on interpoint distances designed to process high-dimensional data.

Suppose $\{X_i\}_{i=1}^{n}$ is a data stream of independent random vectors in $\mathbb{R}^d$. The baseline window $\mathscr{B}$ contains the first N_0 historical observations, i.e. $\mathscr{B} = \{X_i\}_{i=1}^{N_0}$, while the current window contains the latest N_C observations, i.e., $\mathscr{C}_t = \{X_i\}_{i=t}^{t+N_C-1}$, for $t = N_0 + 1, N_0 + 2, \ldots$.

doi:10.6342/NTU202403079

Algorithm 3: Sliding-Window Algorithm

Input: $\{X_i\}_{i=1}^{\infty}$ with baseline window width N_0, the current window width N_C.

Output: The locations of the change points in the original data stream $\tau_1, \tau_2, \tau_3, \ldots$, $s = 1$, $\tau_0 = 0$.

while not at end of the data stream **do**
  $t = 0$ $\mathscr{B} \leftarrow \{X_1, \ldots, X_{N_0}\}$
  do
    $t = t + 1$;
    $\mathscr{C}_t \leftarrow [X_{N_0+t}, \ldots, X_{N_0+t+N_C-1}]$
    Calculate the statistic $L(\mathscr{B}, \mathscr{C}_t)$ based on $\mathscr{B}$ and $\mathscr{C}_t$
  until $L(\mathscr{B}, \mathscr{C}_t) > h$;
  A change point is detected and recorded as $\tau_s = \tau_{s-1} + N_0 + t$
  Discard the observations before $X_{N_0+t+N_C}$ and re-index the observations $X_{N_0+t+N_C}, X_{N_0+t+N_C+1}, \ldots$ by $1, 2, 3, \ldots$
end
return $\tau_1, \tau_2, \tau_3, \ldots$

Most importantly, the homogeneity of $\mathscr{B}$ and $\mathscr{C}_t$ is tested with the energy statistic

$$L_t = L(\mathscr{B}, \mathscr{C}_t) = 2\hat{\mu}_{BC} - \hat{\mu}_{BB} - \hat{\mu}_{CC},$$

where

$$\hat{\mu}_{BC} = (N_0 N_C)^{-1} \sum_{i=1}^{N_0} \sum_{j=N_0+t}^{N_0+t+N_C-1} \|X_i - X_j\|,$$

$$\hat{\mu}_{BB} = \binom{N_0}{2} \sum_{i=1}^{N_0-1} \sum_{j=i+1}^{N_0} \|X_i - X_j\|,$$

$$\hat{\mu}_{CC} = \binom{N_C}{2} \sum_{i=N_0+t}^{N_0+t+N_C-2} \sum_{j=i+1}^{N_0+t+N_C-1} \|X_i - X_j\|.$$

These are the averages of interpoint distances between and within two groups, measured with the derivative information-inspired distance metric; therefore, L_t assesses whether the discrepancy between the baseline and current window is more significant than the variation within both groups. The greater this difference is, the more likely a change point exists. Furthermore, the statistic L_t is updated by a itera-

doi:10.6342/NTU202403079

tive formula, with details provided in Guo and Modarres (2022), which significantly reduces the computing complexity.

Concerning the estimation of the threshold h, the criterion used in the referenced paper—based on the "Protective ability against false alarms (RL_α)", which represents the αth quantile of the ICRL—does not align with the evaluation standards used in our other methods. Therefore, we adopt a different procedure to find the threshold based on ICRL as follows:

We start by setting the initial threshold h to 100, based on empirical knowledge. We then bootstrap the historical observations to somewhat exceed the desired ARL, for instance, by adding 100 to the desired ARL. This additional 100 allows some buffer for the run length to be greater than the desired ARL, as we can't directly assume the run length is equal to the desired ARL if we only have exactly that many samples. Next, we execute our algorithm on the generated samples 20 times to obtain empirical average run lengths.

We then employ a learning rate method to iteratively adjust the threshold. Specifically, we compare the empirical ARL to the target ARL level. If the former moderately surpasses the latter (by 20 to 50), we report the current h as our final threshold to ensure it is sufficiently conservative. However, if the empirical ARL significantly exceeds the desired ARL, indicating that the threshold is too conservative, we adjust the threshold downwards by 10; otherwise, we adjust it upwards by 10.

doi:10.6342/NTU202403079

Chapter 3 Simulations

As stated in Section 1.2, the performance of a change point detection procedure with stopping time T is evaluated by the average stopping time after a change, subject to a false alarm constraint. Let $\mathbb{E}_\tau$ be the expectation under the hypothesis that the true change point occurs at τ, with $\tau = \infty$ indicating no change. To be precise, the expected detection delay, representing the algorithm's ability to detect a change as soon as possible, is defined as:

$$EDD(T) := \mathbb{E}_\tau[T - \tau | T \geq \tau],$$

which represents the conditional expectation of the lag between the stopping time and the actual change location, subjecting to a fixed average run length, or in-control run length (IC-RL)

$$ARL(T) := \mathbb{E}_\infty[T] \geq \gamma$$

for some large constant $\gamma > 0$. Here, $\mathbb{E}_\infty[T]$ is the expectation under the hypothesis of no change.

In this section, we compare our three proposed modified methods against two benchmark procedure, which includes performing control chart and its variant on the functional samples. The evaluation would be based on their power and EDD.

doi:10.6342/NTU202403079

3.1 Simulation Settings

In the subsequent simulations, we adopt the function data settings used in Chiou et al. (2019). With N as our functional data sequence length, the random functions $Y_i(t)$ are generated by the basis expansion:

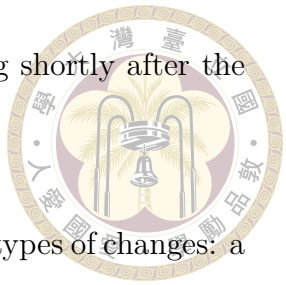
$$Y_i(t) = \psi(t) + \sum_{\ell=0}^{B} \sqrt{\lambda_\ell}\gamma_{i,\ell}\phi_\ell(t), \quad i = 1, \ldots, N,$$

where $\psi(t)$ is the mean function, $(\lambda_\ell, \phi_\ell(\cdot))$ are predetermined eigenvalue–eigenfunction pairs, and $\{\gamma_{i,\ell} : i = 1, \ldots, N\}$, $\ell = 0, \ldots, B$, are random coefficients. The sequence $\{Y_i(t)\}$ depends on the sequence of these random coefficients. For each ℓ, the coefficients $\{\gamma_{i,\ell} : i = 1, \ldots, N\}$ are generated by an AR(1) model such that $\gamma_{i,\ell} = \rho\gamma_{i-1,\ell} + \varepsilon_{i,\ell}$, with $\varepsilon_{i,\ell}$ being the standard normal random variate. The autocorrelation parameter ρ is set to 0.2, indicating low dependence scenario throughout the simulation. Then we set $B = 150$, $N = 200$, $\lambda_\ell = 0.7 \times 2^{-\ell}$, time grid points $t = (0.001, 0.002, \cdots, 1)$, and we assume the change occurs at $i = 150$, corresponding to the third quartile of the observations. The mean function $\psi(t)$ is chosen as $\psi(t) = 0.5 - 100(t-0.1)(t-0.3)(t-0.5)(t-0.9) + 0.8sin(1+10\pi t)$. And the eigenfunctions $\{\phi_\ell(t) : \ell = 1, \ldots, B\}$ are selected as the Fourier basis $\sqrt{2}\sin(2\pi kt - \pi)$ for $\ell = 2k - 1$ and $\sqrt{2}\cos(2\pi kt - \pi)$ for $\ell = 2k, k = 1, \ldots, B/2$, with $\phi_0(t) = 1$, a constant that ensures $\int_T \phi_0^2(t)dt = 1$.

In another case, we set $N = 150$ and assume the change occurs at $i = 100$, while keeping the other parameter settings unchanged. In both change scenarios, the number of historical observations, N_0, is set to 75. The former scenario simulates a change taking place after the algorithm has process some data in the online phase

doi:10.6342/NTU202403079

(denote as case **A**), whereas the latter simulates a change arising shortly after the online phase begins (denote as case **B**).



We then assess the performance of all methods under various types of changes: a change in mean function $\psi_i(t)$, a scale change represented by the size of the multiplier for the random error $\varepsilon_{i,\ell}$ in $\gamma_{i,\ell}$, and a covariance structure change interpreted by the distribution difference in the $\varepsilon_{i,\ell}$'s.

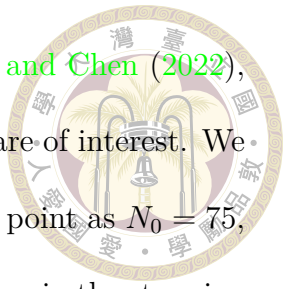
Our detection criterion for success rate in all our methods is established as follows: If the algorithm signals a change shortly after the real change point, specifically by triggering the alert at $N = 150 \sim 200$ in case **A**, or at $N = 100 \sim 150$ in case **B**, we recognize it as a successful detection. Our power is represented by the success rate over 100 replications. Additionally, we calculate the expected detection delay of these successful detections for further comparisons.

3.2 Parameter Settings

Throughout our 100 simulation runs, the ARL is set to 500, 1000, and 1500 to represent different level of false discovery rate control. The detection threshold in our methods is selected accordingly.

For the control chart settings in Chen et al. (2016), α is suggested to be $1/ARL_0$ to achieve the desired false alarm rate. The number of historical observations N_0 is set to 75. The smoothing parameter λ is chosen as the recommended value of 0.05, while the window size is set to at least 5 to produce sufficient distinct values of $T_n^*(w, \lambda)$ for small n. And the number of permutations required is set to 500.

doi:10.6342/NTU202403079



In the kNN based method settings in Chen (2019) and Chu and Chen (2022), for simplicity of comparison, we set k to 3, as only large changes are of interest. We specify the number of historical observations without any change point as $N_0 = 75$, set the window size to $L = 50$, and determine the prespecified values in the stopping rules, n_0, n_1, to be $0.2L$ and $L - n_0 = 0.8L$, as recommended by the reference manual.

Regarding the GEM statistic method in Kurt et al. (2021), we use the analytical solution as our test threshold h:

$$h = \frac{log(A/g(\alpha))}{1 - W(\alpha g(\alpha))/g(\alpha)}$$

Based on the Monte Carlo simulation results, with $\alpha = 0.2$, $g(\alpha)$ is 10.1. Here, the size of S_1, N_1, is set to 25, and thus the size of S_2, $N_2 = N_0 - N_1$, is 50. Since we are capturing the local interactions between points, we chose a moderate value of k to be 4.

Lastly, for the settings in the energy statistic method (GáborJ.Székely and MariaL.Rizzo, 2013), the baseline window width, embodying the historical observation amount, N_0 is set to 75, while the current window width N_C is set to 10. The threshold h is obtained via a learning rate method mentioned in the 2.4, with an initial value of $h = 100$ and using 50 runs to get the empirical ARL.

For the method names in the following performance tables: "Graph+Eucli" refers to the graph-based method using Euclidean distance, and "Graph+Deriv" denotes the graph-based method using derivative distance. "Energy" represents the energy statistic method, "GEM" refers to the GEM statistic method, "DFEWMA"

doi:10.6342/NTU202403079

stands for the DFEEMA control chart method, and "CC+Depth" denotes the control chart method variant with functional depth transformation.

3.3 Mean Change

This denotes a change in the mean function $\psi_i(t)$. The initial mean function is given by: $\psi(t) = 0.5 - 100(t-0.1)(t-0.3)(t-0.5)(t-0.9) + 0.8sin(1+10\pi t)$, and it changes to $\psi(t) = 1 + 3t^2 - 5t^3$ when the change occurs.

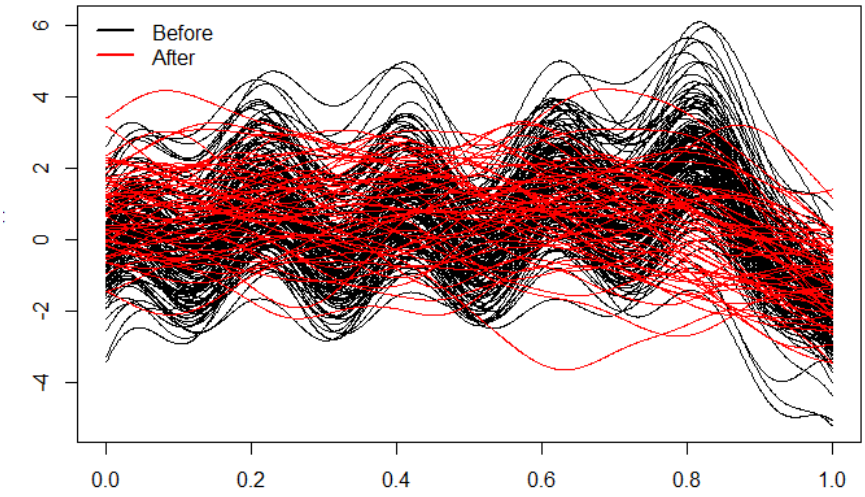


Figure 3.1: Visualization of Mean Change (case **A**)

As shown in Table 3.1, energy statistic method has only an outcome. This is due to its time-consuming threshold finding process and its underwhelming results. Although it is an intuitive technique for examining the differences between the nominal dataset and the current window dataset, its drawbacks prevent us from implementing it. Therefore, we will not consider this technique for the following simulations other than the specific case of case **A** and ARL=500.

We can observe that as ARL rises, all methods generally achieve more successful

doi:10.6342/NTU202403079

Table 3.1: Success rate comparison under Mean Change - Success rate, $(\cdot)$ indicates instances where the algorithm failed to detect any change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	0.60(0)	0.78(0)	0.82(0)	0.79(0)	0.89(0)	0.93(0)
Graph+Eucli(W)	0.61(0)	0.83(0)	0.90(0)	0.82(0)	0.94(0)	0.95(0)
Graph+Eucli(S)	0.65(0)	0.82(0)	0.87(0)	0.87(0)	0.93(0)	0.95(0)
Graph+Deriv(Z)	0.41(0)	0.57(0)	0.71(0)	0.66(0)	0.81(0)	0.83(1)
Graph+Deriv(W)	0.44(0)	0.65(0)	0.72(0)	0.65(1)	0.81(0)	0.83(1)
Graph+Deriv(S)	0.52(0)	0.69(0)	0.72(0)	0.67(0)	0.85(0)	0.90(1)
Energy	0.73(0)					
GEM	0.85(0)	0.91(0)	0.93(0)	0.97(0)	0.99(0)	0.98(0)
DFEWMA	0.41(0)	0.41(0)	0.54(0)	0.68(0)	0.78(0)	0.83(0)
CC + Depth	0.73(0)	0.83(0)	0.76(0)	0.88(3)	0.93(3)	0.86(6)

detections. However, although the DFEWMA control chart with functional depth outperforms most other methods, its success count in both case **A** and case **B** drops slightly when moving from ARL=1000 to ARL=1500. Additionally, in case **B**, it experiences more failures where no change is detected at all when ARL reaches 1500. This indicates that the threshold has become stricter for the algorithm to signal a change.

From the failure count provided in Table 3.1, we can see that most detection failures are attributed to false early detection by the algorithm. This suggests that the asymptotic threshold obtained analytically might be loose and causing these false alarms.

On the whole, all methods perform better in case **B**, as there is less chance of getting a false alarm when the change occurs shortly after the online phase starts. Besides, the GEM statistic method has the best performance among all. Not only

doi:10.6342/NTU202403079

does it excel in case **A**, where the change occurs a while after the algorithm starts, but it also captures almost all anomalies in case **B**.

Table 3.2: EDD comparison under Mean Change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	10.35	10.962	10.829	9.962	10.18	10.817
Graph+Eucli(W)	11.016	11.639	11.556	10.793	10.872	11.632
Graph+Eucli(S)	10.785	11.39	11.253	10.885	10.71	11.179
Graph+Deriv(Z)	6.244	6.193	7.127	5.712	6.321	6.88
Graph+Deriv(W)	6.068	6.015	6.917	5.538	6.21	6.651
Graph+Deriv(S)	6.25	6.261	7.25	6.179	6.459	6.933
Energy	8.082					
GEM	3.847	4.549	4.71	3.784	4.707	4.673
DFEWMA	9.561	8.829	8.463	9.309	9.41	10.133
CC + Depth	8.082	8.735	8.566	9.045	9.28	9.477

As stated in 2.3, Table 3.2 shows that the GEM statistic attains the lowest expected detection delay among all methods. In contrast, all the graph-based methods using the Euclidean distance measure to form kNN graphs have the longest expected detection delay in this circumstance. Furthermore, although Section 2.2 claims that $W_{L|y}(t, n)$ would resolve the increased detection delay issue of the original statistic $Z_{L|y}(t, n)$, it appears to have the same performance as $Z_{L|y}(t, n)$, even in its recommended scenario, where there is only a locational change.

3.4 Scale Change

This refers to a change in the scale of the random error $\varepsilon_{i,\ell}$ in $\gamma_{i,\ell}$, where the multiplier changes from 1 to 2 when the change occurs.

doi:10.6342/NTU202403079

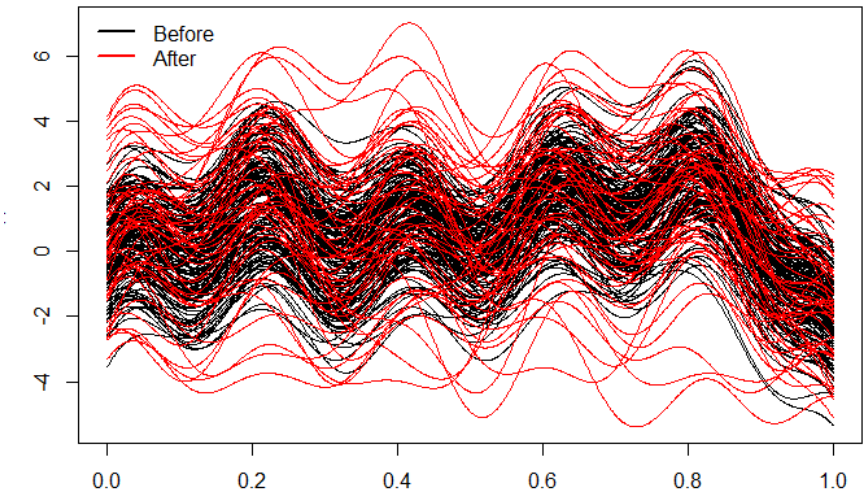


Figure 3.2: Visualization of Scale Change (case **A**)

Table 3.3: Success rate comparison under Scale Change - Success rate, $(\cdot)$ indicates instances where the algorithm failed to detect any change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	0.54(0)	0.51(1)	0.58(4)	0.77(1)	0.76(1)	0.75(0)
Graph+Eucli(W)	0.57(0)	0.54(1)	0.54(4)	0.82(1)	0.72(1)	0.63(0)
Graph+Eucli(S)	0.63(0)	0.69(0)	0.73(0)	0.87(1)	0.87(1)	0.84(0)
Graph+Deriv(Z)	0.42(1)	0.49(0)	0.62(0)	0.73(0)	0.77(2)	0.83(0)
Graph+Deriv(W)	0.47(0)	0.54(2)	0.57(0)	0.74(0)	0.74(2)	0.78(0)
Graph+Deriv(S)	0.54(0)	0.64(0)	0.68(0)	0.81(0)	0.79(0)	0.89(0)
Energy	0.11(0)					
GEM	0.92(0)	0.92(0)	0.96(0)	0.96(0)	0.99(0)	0.98(0)
DFEWMA	0.30(7)	0.33(12)	0.45(9)	0.61(7)	0.67(17)	0.62(19)
CC + Depth	0.72(0)	0.79(0)	0.82(0)	0.93(0)	0.95(0)	0.92(0)

In Table 3.3, besides achieving similar results in the mean change scenario, where all methods perform better in case **B** than in case **A**, the GEM statistic also proves its effectiveness in the scale change scenario, with only a few failures due to early detection. In comparison, the control chart with functional data depth not only secures the second place in overall results but also shows a significant performance

doi:10.6342/NTU202403079

improvement from case **A** to case **B**. This indicates that it is more suitable for online scenarios where changes happen frequently.

Table 3.4: EDD comparison under Scale Change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	16.148	14.745	16.017	13.688	16.066	17.227
Graph+Eucli(W)	22.316	21.519	23.222	19.293	22.875	23.222
Graph+Eucli(S)	13.397	13.173	13.698	11.966	13.276	14.643
Graph+Deriv(Z)	13.738	14.673	16.596	13.274	14.636	16.398
Graph+Deriv(W)	21.979	23.241	25.754	21.622	23.338	26.359
Graph+Deriv(S)	13.019	12.672	14.088	12.099	12.608	13.326
Energy	2.091					
GEM	4.217	5	5.094	4.479	5.081	5.367
DFEWMA	19.667	20.697	19.511	19.361	19.403	17.597
CC + Depth	6.444	6.405	6.939	6.452	6.789	6.88

From the results in Table 3.4, we can observe that detecting a change in scale often requires more iterations. Since the control chart is originally designed for capturing locational changes, it sometimes fails to detect scale changes, resulting in a large number of undetected changes. However, in this scenario, the control chart variant with functional data depth exhibits a relatively low expected detection delay compared to the other methods. This improvement might result from the functional depths chosen in Section 2.1, such as TVD and MBD, which are effective in detecting magnitude changes.

Last but not least, the GEM statistic method once again achieves the lowest expected detection delay, proving its superiority over the other techniques.

On the other hand, $W_{L|y}(t, n)$ attains the longest expected detection delay

doi:10.6342/NTU202403079

among all methods, which is due to the fact that it is specifically designed for locational changes.

3.5 Mean & Scale Change

This indicates that when the change occurs, both situations described in the previous scenarios take place.

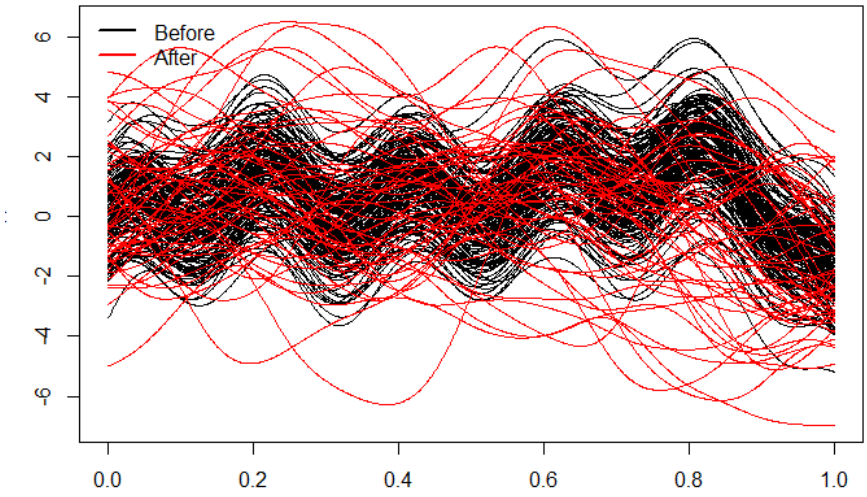


Figure 3.3: Visualization of Mean & Scale Change (case **A**)

As shown in Table 3.5, the graph-based methods with two different distance measures yield similar results, with the Euclidean distance performing slightly better. Moreover, in case **A**, the control chart method variant has almost twice the number of successful detections compared to its counterparts, further proving the effectiveness of applying functional data depth.

In addition, as stated in the paper, $S_{L|y}(t, n)$ performs slightly better than the other two statistics in this general change circumstance.

In Table 3.6, the graph-based method with the derivative information distance

doi:10.6342/NTU202403079

Table 3.5: Success rate comparison under Mean & Scale Change - Success rate, $(\cdot)$ indicates instances where the algorithm failed to detect any change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	0.54(0)	0.67(1)	0.72(0)	0.83(0)	0.89(0)	0.93(0)
Graph+Eucli(W)	0.60(0)	0.74(1)	0.75(0)	0.85(0)	0.87(0)	0.92(0)
Graph+Eucli(S)	0.65(0)	0.73(1)	0.76(0)	0.87(0)	0.87(0)	0.91(0)
Graph+Deriv(Z)	0.43(5)	0.57(1)	0.75(0)	0.72(0)	0.85(1)	0.87(1)
Graph+Deriv(W)	0.48(1)	0.66(1)	0.71(0)	0.73(0)	0.87(1)	0.89(1)
Graph+Deriv(S)	0.56(1)	0.69(1)	0.74(0)	0.81(0)	0.91(1)	0.90(1)
Energy	0.03(0)					
GEM	0.90(0)	0.98(0)	0.98(0)	0.97(0)	0.99(0)	1.00(0)
DFEWMA	0.31(0)	0.39(0)	0.43(2)	0.83(0)	0.83(1)	0.75(2)
CC + Depth	0.73(0)	0.86(0)	0.74(0)	0.94(0)	0.92(0)	0.94(0)

Table 3.6: EDD comparison under Mean & Scale Change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	10.889	11.94	12.333	11	12.135	12.505
Graph+Eucli(W)	13.6	14.905	16.36	13.671	15.598	15.761
Graph+Eucli(S)	11.077	11.548	11.566	10.678	11.506	12.055
Graph+Deriv(Z)	6.744	6.912	7.627	6.639	7.282	7.655
Graph+Deriv(W)	6.812	7.242	7.577	6.986	7.414	7.831
Graph+Deriv(S)	7.089	7.231	7.824	7.099	7.67	7.9
Energy	0.333					
GEM	3.867	4.684	4.908	3.897	4.657	4.75
DFEWMA	11.516	13.564	14.116	12.217	13.229	13.293
CC + Depth	5.767	6.233	6.216	5.894	6.141	6.245

measure has a lower expected detection delay compared to the Euclidean distance across all cases. This indicates its superior performance in capturing the attributes of functional data when dealing with general changes.

doi:10.6342/NTU202403079

Furthermore, the GEM statistic again achieves the lowest expected detection delay, indicating its ability to handle general changes effectively. Similarly, the control chart method with functional depth also performs well in detecting general changes.

3.6 Covariance Structure Change

This indicates a change in the distribution of the random error $\varepsilon_{i,\ell}$ from a standard normal distribution $N(0,1)$ to an exponential distribution $Exp(1)$ when the change occurs, which also involves a shift in the mean, as shown in the figure.

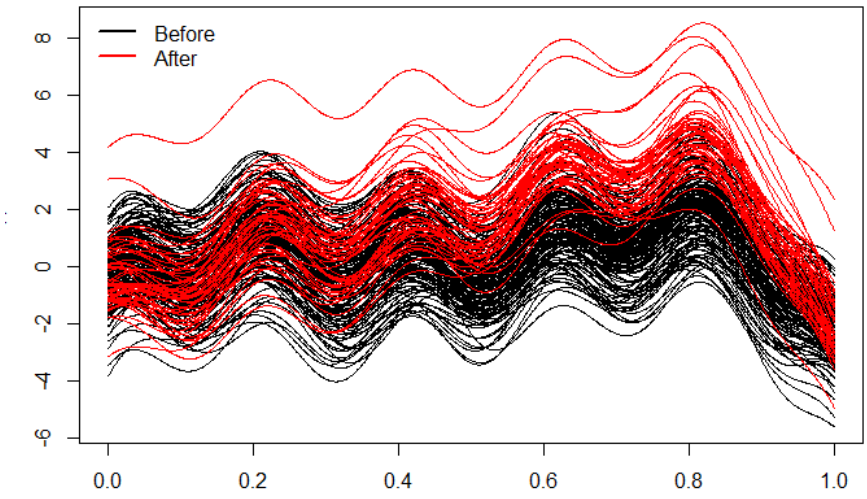


Figure 3.4: Visualization of Covariance Structure Change (case **A**)

Surprisingly, in Table 3.7, the control chart method variant shows a large number of detection failures, primarily due to no detection at all. Furthermore, the increasing failures from ARL=1000 to ARL=1500 in both cases indicate that the threshold is already too high for the algorithm to detect a change.

As shown in Table 3.8, although the average run length between methods does

doi:10.6342/NTU202403079

Table 3.7: Success rate comparison under Covariance Structure Change - Success rate, $(\cdot)$ indicates instances where the algorithm failed to detect any change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	0.54(0)	0.73(0)	0.79(0)	0.75(0)	0.87(0)	0.91(0)
Graph+Eucli(W)	0.61(0)	0.73(0)	0.81(0)	0.82(0)	0.90(0)	0.90(0)
Graph+Eucli(S)	0.67(0)	0.72(0)	0.82(0)	0.84(0)	0.90(0)	0.88(0)
Graph+Deriv(Z)	0.44(0)	0.63(0)	0.68(1)	0.76(1)	0.75(0)	0.91(0)
Graph+Deriv(W)	0.53(0)	0.72(0)	0.71(1)	0.81(1)	0.76(0)	0.91(0)
Graph+Deriv(S)	0.60(0)	0.79(0)	0.72(1)	0.81(1)	0.78(0)	0.92(0)
Energy	0.57(5)					
GEM	0.92(0)	0.95(0)	0.92(0)	1.00(0)	0.95(0)	0.98(0)
DFEWMA	0.31(0)	0.58(0)	0.54(0)	0.69(0)	0.82(0)	0.84(0)
CC + Depth	0.68(4)	0.83(3)	0.68(13)	0.80(13)	0.78(16)	0.64(31)

Table 3.8: EDD comparison under Covariance Structure Change

	case **A** ($\tau = 150$)			case **B** ($\tau = 100$)		
In-Control RL	500	1000	1500	500	1000	1500
Graph+Eucli(Z)	9.315	9.89	10.241	9.227	10.184	10.264
Graph+Eucli(W)	9.246	10.096	10.358	9.305	10.056	10.178
Graph+Eucli(S)	9.642	10.222	10.317	9.452	10.189	10.239
Graph+Deriv(Z)	7.705	8.302	8.515	7.5	8.693	9.121
Graph+Deriv(W)	7.642	8.417	8.408	7.457	8.539	8.868
Graph+Deriv(S)	8.267	8.468	8.639	8	8.731	8.891
Energy	11.088					
GEM	8.152	9.453	10.489	7.73	9.811	9.418
DFEWMA	6.032	6.069	6.352	6.116	6.366	6.726
CC + Depth	12.338	13.012	12.971	13.363	15.218	12.656

not differ significantly, the GEM statistic method does not have the lowest average run length in this scenario. Instead, the original control chart method achieves the lowest average run length. Despite its unsatisfactory results in case **A**, it outperforms

doi:10.6342/NTU202403079

its variant in terms of detection power and detection delay in **B**.

All in all, although the graph-based method with derivative information-inspired distance measure generally performs slightly worse or comparably to its Euclidean distance counterpart, its expected detection delay (EDD) tends to be lower. This might result from the newly considered shape information in the functional samples, which magnifies the random variation in the distance matrix. Consequently, the chances of false detections increases, yet the EDD becomes lower.

3.7 Verification of respective ARLs

In this section, aside from the lack of an analytical threshold for the energy statistic method, we will test whether the recommended thresholds h for the other three methods can achieve the nominal run length claimed in their respective papers. Assume nominal ARL=500.

For the graph-based method, the prespecified ARL is provided in the official package. Therefore, we simply generate a sufficient number of data points and test the algorithm by setting the ARL parameter in the function accordingly. In 50 runs, we obtained an empirical ARL of 235.6, indicating the threshold is too loose for detecting changes.

For the GEM CUSUM method, we modify the algorithm to generate data with no change distribution as the newly arrived data points. If the algorithm stops at a certain time t, this time point is considered its run length. In 50 runs, we obtained an empirical ARL of 996.04, indicating the threshold is too strict for detecting changes.

doi:10.6342/NTU202403079

Lastly, for the control chart method, we test it by setting IC-RL$\doteq \frac{1}{\alpha}$ and then using a dataset with no change point to run 50 times to determine its average run length. We obtained an empirical ARL of 158.28, indicating the threshold is too loose for detecting changes as well.

3.8 Remark

In section 2.2, with a minor modification, we can expand the functionality from merely reporting the alert locations to estimating the location of the change. This is achieved by identifying the location where the test statistics reach their maximum, namely by adding $n - n_1 - 1$ to the alert locations. For instance, we can check whether the algorithm alerts consecutively for a certain number of times, suggesting a strong signal.

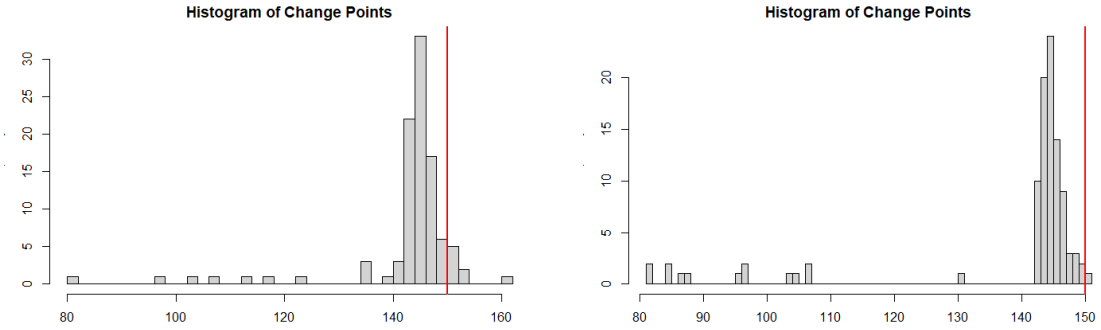
To illustrate, suppose we first check if there are alerts for three consecutive observations. When verifying the change point locations, if we notice that these alerts consistently point to nearly the same location (say A), then we can report that a change has occurred at position A. This method helps ensure that the identified change point is accurate.

Thus, if we want to pinpoint the exact time event that causes the algorithm to signal a change, the graph-based method proves to be valuable. By cross-referencing the alert patterns with the estimated change point locations, we can enhance the reliability of our change detection process.

Given the underwhelming power of the energy statistic method in the scale change and mean & scale change scenarios as shown in Fig.3.5, we further examined

doi:10.6342/NTU202403079

the estimated change point locations and found that most of the change points are actually relatively close to the real change ($i = 150$). Thus, we can improve the algorithm by adjusting the threshold-finding algorithm to yield a more conservative threshold, namely a bigger h.



(a) Scale Change (case **A**) (b) Mean & Scale Change (case **A**)

Figure 3.5: Estimated Change Point Locations (red line as the True Change Point)

On the other hand, the GEM statistics method in section 2.3 provides a solution with a much lower expected detection delay, proving itself to be the best in terms of EDD and power. Consequently, if we need to respond to general changes as quickly as possible, this method would be the best choice.

Furthermore, the control chart method not only provides an intuitive approach to dealing with functional data, but also performs better in most cases when the functional samples is summarized using functional data depth. In the future, we can research which combination of these functional data depth yields the best result.

The approximate execution times for each method under 100 replications are as follows: the original DFEWMA control chart takes about 1 to 3 hours; the functional depth variant of control chart method takes 15 to 30 minutes. The graph-based method usually takes 5 to 10 minutes. The Gem statistic method has the shortest runtime of 1 to 3 minutes. The most time-consuming is the energy statistic method,

doi:10.6342/NTU202403079

which takes around 5 to 10 hours depending on the scenario.

doi:10.6342/NTU202403079

Chapter 4 Conclusion

Our motivation is that there doesn't seem to be any online change point detection method solely designed for functional data. Therefore, we identified some feasible existing nonparametric techniques for multivariate data and made appropriate modifications to apply them to our scenario. Moreover, the promising performance obtained from the control chart with data depth indicates a potential path for future OCPD methods on functional data.

Given the inconsistency between the empirical ARL and nominal ARL, future work should focus on reducing this gap. This difference may stem from the different autocorrelation settings in our simulation compared to the papers' settings, where the assumption is an independent case. For the energetic method, tuning the threshold training algorithm or increasing the number of runs under a prespecified IC-RL, while addressing the time-consuming issue, would be beneficial for achieving more satisfactory results.

While using the graph-based method, we might consider the results from the new derivative distance metric when determining the threshold in the future. Besides, we could try to improve the power while keeping the EDD low by making a few adjustments to the distance function or altering it to other concepts, such as

doi:10.6342/NTU202403079

optimal L^2. Regarding the GEM statistic method, future research could explore using other statistics that are more applicable to functional data or applying different stopping rules besides CUSUM to signal a change.

Additionally, our research only considers the autocorrelation parameter ρ as 0.2, indicating a low dependence scenario. It would be valuable to also consider scenarios with $\rho = 0$ (independent case) and scenarios with strong dependence ($\rho = 0.5$) to obtain a comprehensive understanding of these methods' overall performance. In the independent case ($\rho = 0$), the methods are expected to perform well, as they generally assume independence between data points. Moreover, our simulation settings are relatively ideal, which further contributes to their expected strong performance in the independent scenario.

Future research could consider more realistic conditions, such as functional data collected on inconsistent time grids. This would better reflect real-world scenarios and provide insights into how these methods perform under less ideal conditions. In such scenarios, it may be necessary to employ techniques like interpolation and smoothing to standardize the data. However, smoothing can introduce its own challenges, such as measurement errors, which need to be carefully managed. By incorporating these additional scenarios and complexities, we can gain a more thorough understanding of the robustness and applicability of the proposed methods.

Regarding the composition of the depth measures in the control chart method variant, we have only tested one combination, based on the intuitive idea of consolidating their strengths. However, there is potential to experiment with various different combinations of these measures in future research. By exploring various

doi:10.6342/NTU202403079

configurations, we can better understand their individual and collective contributions to the overall performance and possibly identify more effective combinations.

In conclusion, our approach is not merely about applying existing methods but rather about carefully considering the unique characteristics of functional data. For instance, we incorporated a distance metric that includes first-order derivatives to account for shape variations, which are not considered in multivariate data. On top of that, we applied functional data depths to the control chart, combining the various information summarized by these different depth measures, which also turns out to be the second best method in our simulations. These thoughtful integrations ensure that the nuances of functional data are captured, leading to more accurate and effective online change point detection.

doi:10.6342/NTU202403079

References

Alexander Aue, Gregory Rice, and Ozan Sönmez. Detecting and dating structural breaks in functional data without dimension reduction. J. R. Statist. Soc. B, 80 (3):509–529, 2018.

Edward Austin, Gaetano Romano, Idris A. Eckley, and Paul Fearnhead. Online non-parametric changepoint detection with application to monitoring operational performance of network devices. Computational Statistics and Data Analysis, 177, 2023.

István Berkes, Robertas Gabrys, Lajos Horváth, and Piotr Kokoszka. Detecting changes in the mean of functional observations. J. R. Statist. Soc. B, 71(5):927–946, 2009.

Yang Cao and Yao Xie. Robust sequential change-point detection by convex optimization. In IEEE International Symposium on Information Theory (ISIT), 2017.

Yang Cao, Liyan Xie, Yao Xie, and Huan Xu. Sequential change-point detection via online convex optimization. Entropy, 20(2), 2018.

Yang Cao, Andrew Thompson, Meng Wang, and Yao Xie. Sketching for sequential change-point detection. Journal on Advances in Signal Processing, 42, 2019.

doi:10.6342/NTU202403079

Hao Chen. Sequential change-point detection based on nearest neighbors. The Annals of Statistics, 47(3):1381–1407, 2019.

Hao Chen and Lynna Chu. Graph-based change-point analysis. Annual Review of Statistics and Its Application, 10:475–499, 2023.

Nan Chen, Xuemin Zi, and Changliang Zou. A distribution-free multivariate control chart. Technometrics, 58(4):448–459, 2016.

Yudong Chen, Tengyao Wang, and Richard J. Samworth. High-dimensional and multiscale online changepoint detection. J. R. Statist. Soc. B, 84(1):234 – 266, 2022.

Yudong Chena, TengyaoWang, and Richard J. Samworth. Inference in high-dimensional online changepoint detection. Journal of the American Statistical Association, 0(0):1–12, 2023.

Jeng-Min Chiou, Yu-Ting Chen, and Tailen Hsing. Identifying multiple changes for a functional data sequence with application to freeway traffic segmentation. The Annals of Applied Statistics, 13(3):1430–1463, 2019.

Lynna Chu and Hao Chen. Sequential change-point detection for high-dimensional and non-euclidean data. IEEE Transactions on Signal Processing, 70, 2022.

Georgios Fellouris, Erhan Bayraktar, and Lifeng Lai. Efficient byzantine sequential change detection. IEEE Transactions on Information Theory, 64(5), 2018.

André Ferrari, Cédric Richard, Anthony Bourrier, and Ikram Bouchikhi. Online change-point detection with kernels. Pattern Recognition, 133, 2023.

doi:10.6342/NTU202403079

Jun Geng, Bingwen Zhang, Lauren M. Huie, and Lifeng Lai. Online change-point detection of linear regression models. IEEE Transactions on Signal Processing, 67(12):3316–3329, 2019.

Lingzhe Guo and Reza Modarres. Two multivariate online change detection models. Journal of Applied Statistics, 49(2):427–448, 2022.

Josua G¨osmann, Christina Stoehr, Johannes Heiny, and Holger Dette. Sequential change point detection in high dimensional time series. Electronic Journal of Statistics, 16:3608–3671, 2022.

GáborJ.Székely and MariaL.Rizzo. Energy statistics: A class of statistics based on distances. Journal of Statistical Planning and Inference, 143(8):1249–1272, 2013.

Trevor Harris, Bo Li, and J. Derek Tucker. Scalable multiple changepoint detection for functional data sequences. Environmetrics, 33:509–529, 2020.

Lajos Horvath, Piotr Kokoszka, and Shixuan Wang. Monitoring for a change point in a sequence of distributions. The Annals of Statistics, 49(4):2271–2291, 2021.

Huang Huang and Ying Sun. A decomposition of total variation depth for understanding functional outliers. Technometrics, 61(4):445–458, 2019.

Siegfried Hörmann and Piotr Kokoszka. Weakly dependent functional data. The Annals of Statistics, 38(3):1845–1884, 2010.

Mehmet Necip Kurt, Yasin Yılmaz, and Xiaodong Wang. Real-time nonparametric anomaly detection in high-dimensional settings. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(7), 2021.

doi:10.6342/NTU202403079

Tze Leung Lai. Sequential changepoint detection in quality control and dynamical systems. J. R. Statist. Soc. B, 57(4):613–658, 1995.

Lingjun Li and Jun Li. Online change-point detection in high-dimensional covariance structure with application to dynamic networks. The Journal of Machine Learning Research, 24(1):2144–2187, 2023.

Shuang Li, Yao Xie, Hanjun Dai, and Le Song. Scan b-statistic for kernel change-point detection. Sequential Analysis, 38(4):503–544, 2019.

Sen Lin, Gianmarco Mengaldo, and Romit Maulik. Online data-driven changepoint detection for high-dimensional dynamical systems. Chaos, 33(10), 2023.

James P. Long and Yao Xie. A study of functional depths. In arXiv preprint arXiv:1506.01332v3, 2016.

Sara López-Pintado and Juan Romo. On the concept of depth for functional data. Journal of the American Statistical Association, 104(486):718–734, 2009.

Bernardo Marenco, Paola Bermolen, Marcelo Fiori, Federico Larroca, and Gonzalo Mateos. Online change point detection for weighted and directed random dot product graphs. IEEE Transactions on Signal and Information Processing over Networks, 8:144–159, 2022.

Yajun Mei. Sequential change-point detection when unknown parameters are present in the pre-change distribution. The Annals of Statistics, 34(1):92–122, 2006.

Yinfeng Meng, Jiye Liang, Fuyuan Cao, and Yijun He. A new distance with derivative information for functional k-means clustering algorithm. Information Sciences, page 166–185, 2018.

doi:10.6342/NTU202403079

Naveen N. Narisetty and Vijayan N. Nair. Extremal depth for functional data and applications. Journal of the American Statistical Association, 111(516):1705–1714, 2016.

E. S. Page. Continuous inspection schemes. Biometrika, 41(1/2):100–115, 1954.

Maxim Raginsky, Rebecca M. Willett, Corinne Horn, Jorge Silva, and Roummel F. Marcia. Sequential anomaly detection in the presence of noise and limited feedback. IEEE Transactions on Information Theory, 58(8):5544–5562, 2012.

Anton Rauhameri, Katri Salminen, Jussi Rantala, Timo Salpavaara, Jarmo Verho, Veikko Surakka, Jukka Lekkala, Antti Vehkaoja, and Philipp Müller. A comparison of online methods for change point detection in ion-mobility spectrometry data. Array, 14, 2022.

Haoyun Wang and Yao Xie. Sequential change-point detection: Computation versus statistical performance. In arXiv preprint arXiv:2210.05181v3, 2023.

Liyan Xie, Yao Xie, and George V. Moustakides. Sequential subspace change point detection. Sequential Analysis, 39(3):307–335, 2020.

Liyan Xie, Shaofeng Zou, Yao Xie, and Venugopal V. Veeravalli. Sequential (quickest) change detection: Classical results and new directions. IEEE Journal on Selected Areas in Information Theory, 2:494–514, 2021.

Liyan Xie, George V. Moustakides, Life Senior, and Yao Xie. Window-limited cusum for sequential change detection. IEEE Transactions on Information Theory, 69 (9), 2023.

doi:10.6342/NTU202403079

Yao Xie and David Siegmund. Sequential multi-sensor change-point detection. The Annals of Statistics, 41(2):670–692, 2013.

Yijun Zuo and Robert Serfling. General notions of statistical depth function. The Annals of Statistics, 28(2):461–482, 2000.

doi:10.6342/NTU202403079

Appendix A — Depth definitions

In terms of the corresponding definitions of the selected depths used in 2.1:

Extremal depth (ED) Narisetty and Nair (2016): Before defining ED, we first illustrate how its left-tail stochastic ordering works. Let $S := \{f_1(t), f_2(t), \ldots, f_n(t)\}$ be a collection of n functional observations with $t \in [0,1]$, and $g(t)$ be a given function that may or may not be a member of S. For each fixed $t \in [0,1]$, define the pointwise depth of $g(t)$ with respect to S as

$$D_g(t,S) := 1 - \left| \frac{1}{n} \sum_{i=1}^{n} [\mathbb{1}\{f_i(t) < g(t)\} - \mathbb{1}\{f_i(t) > g(t)\}] \right|,$$

and let $\Phi_g(\cdot)$ be the cumulative distribution function (CDF) of the distinct values taken by $D_g(t,S)$ as t varies in $[0,1]$. Supposed having two functions g and h with depth CDFs Φ_g and Φ_h. Let $0 \leq d_1 < \ldots < d_M \leq 1$ be the ordered elements of their depth levels combined. If $\Phi_h(d_1) > \Phi_g(d_1)$, then $h \prec g$, meaning h is more extreme than g, and vice versa. If $\Phi_h(d_1) = \Phi_g(d_1)$, we move to d_2 and make a similar comparison. This process is repeated until the tie is broken. If for all $i = 1, \ldots, M$, $\Phi_h(d_i) = \Phi_g(d_i)$, the two functions are viewed as equivalent in terms of depth and are denoted as $g \sim h$. Finally, we have ED of a function g with respect to the sample

doi:10.6342/NTU202403079

functions $S = \{f_1, \ldots, f_n\}$ defined as:

$$ED(g, S) = \frac{\#\{i : g \succeq f_i\}}{n}.$$

where $g \succeq f_i$ if either $g \succ f_i$ or $g \sim f_i$.

Linfinity depth (L^∞D) Long and Xie (2016): This depth is simply the generalization of the L^p multivariate depth to functional case. Let I be some compact interval of $\mathbb{R}$ and $C(I)$ be the set of continuous functions on I. Let $X = \{X(t) : t \in I\}$ be a process in $C(I)$ with distribution P. For any functions $g, h \in C(I)$, define $\|g - h\|_\infty = \sup_{t \in C(I)} |g(t) - h(t)|$. The L^∞ depth for functional data is then

$$L^\infty D(x, P) = (1 + \mathbb{E}[\|x - X\|_\infty])^{-1}.$$

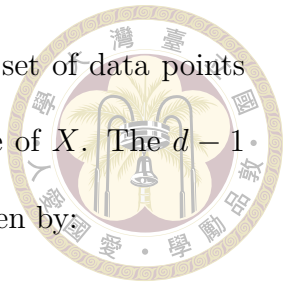
Modified band depth (MBD) López-Pintado and Romo (2009): This is a more flexible version of band depth by measuring the set where the function is inside the corresponding band. For any of the functions f in $f_1, \ldots, f_n$ and for $2 \leq j \leq n$, let $A_j(f)$ be the set in the interval I where the function f is in the band determined by the functions $f_{i_1}, \ldots, f_{i_j}$. If λ is the Lebesgue measure on I, $\lambda_r(A_j(f)) = \lambda(A_j(f))/\lambda(I)$ stands for the "proportion of time" that f is in the band. Now,

$$MBD_n^{(j)}(f) = \binom{n}{j}^{-1} \sum_{1 \leq i_1 < i_2 < \cdots < i_j \leq n} \lambda_r(A(f; f_{i_1}, f_{i_2}, \ldots, f_{i_j})), \quad 2 \leq j \leq n.$$

Let J be a fixed value with $2 \leq j \leq n$. For functions $f_1, \ldots, f_n$, the MBD of any of these curves f is

$$MBD_{n,J}(f) = \sum_{j=2}^{J} MBD_n^{(j)}(f).$$

doi:10.6342/NTU202403079



Projection depth (PD) Zuo and Serfling (2000): Suppose a set of data points $X = (x_1, \ldots, x_n)$ and let $\mathbb{P}$ be the empirical probability measure of X. The $d-1$ dimensional unit sphere in $\mathbb{R}^d$ is denoted by $\mathbb{S}^{d-1}$. The PD is given by:

$$D_P(z \mid X) = \min_{\mathbb{P} \in \mathbb{S}^{d-1}} \left(1 + \frac{|\langle z, \mathbb{P}\rangle - \text{med}(\langle X, \mathbb{P}\rangle)|}{\text{MAD}(\langle X, \mathbb{P}\rangle)}\right)^{-1},$$

where $\langle X, \mathbb{P}\rangle$ is the univariate data set obtained by projecting each point of X on $\mathbb{P}$, med is the univariate median, and MAD is the median absolute deviation from the median.

Total variation depth (TVD) Huang and Sun (2019): Let X be a real-valued stochastic process on $\mathcal{T}$ with distribution F_X, where $\mathcal{T}$ is an interval in $\mathbb{R}$. Denote f as a function, and $f(t)$ as the functional value at a given t. Define $R_f(t) = \mathbb{1}\{X(t) \leq f(t)\}$. Therefore, we have $p_f(t) = \mathbb{E}[R_f(t)] = \mathbb{P}(X(t) \leq f(t))$, which represents the relative position of $f(t)$ w.r.t. $X(t)$. For a given function $f(t)$ at each fixed t, we introduce the pointwise TVD of $f(t)$ as $D_f(t) = \text{var}(R_f(t)) = p_f(t)(1 - p_f(t))$. Eventually, we define the functional TVD for the given function $f(t)$ on $\mathcal{T}$ as:

$$\text{TVD}(f) = \int_{\mathcal{T}} w(t) D_f(t) dt,$$

where $w(t)$ is a weight function defined on $\mathcal{T}$.

doi:10.6342/NTU202403079