DISSERTATION

FUNCTIONAL METHODS IN OUTLIER DETECTION AND CONCURRENT REGRESSION

Submitted by

Michael L. Creutzinger

Department of Statistics

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2024

Doctoral Committee:

Advisor: Daniel Cooley
Co-Advisor: Julia L. Sharp

Matt Koslovsky
Dominik Liebl
Francisco Ortega

ABSTRACT

FUNCTIONAL METHODS IN OUTLIER DETECTION AND CONCURRENT REGRESSION

Functional data are data collected on a curve, or surface, over a continuum. The growing presence of high-resolution data has greatly increased the popularity of using and developing methods in functional data analysis (FDA). Functional data may be defined differently from other data structures, but similar ideas apply for these types of data including data exploration, modeling and inference, and post-hoc analyses. The methods presented in this dissertation provide a statistical framework that allows a researcher to carry out an analysis of functional data from "start to finish".

Even with functional data, there is a need to identify outliers prior to conducting statistical analysis procedures. Existing functional data outlier detection methodology requires the use of a functional data depth measure, functional principal components, and/or an outlyingness measure like Stahel-Donoho. Although effective, these functional outlier detection methods may not be easily interpreted. In this dissertation, we propose two new functional outlier detection methods. The first method, Practical Outlier Detection (POD), makes use of ordinary summary statistics (e.g., minimum, maximum, mean, variance, etc). In the second method, we developed a Prediction Band Outlier Detection (PBOD) method that makes use of parametric, simultaneous, prediction bands that meet nominal coverage levels. The two new outlier detection methods were compared to three existing outlier detection methods: MS-Plot, Massive Unsupervised Outlier Detection, and Total Variation Depth. In the simulation results, POD performs as well, or better, than its counterparts in terms of specificity, sensitivity, accuracy, and precision. Similar results were found for PBOD, except for noticeably smaller values of specificity and accuracy than all other methods.

Following data exploration and outlier detection, researchers often model their data. In FDA, functional linear regression uses a functional response $Y_i(t)$ and scalar and/or functional predictors, $X_i(t)$. A functional concurrent regression model is estimated by regressing Y_i on X_i pointwise

at each sampling point, t. After estimating a regression model (functional or non-functional), it is common to estimate confidence and prediction intervals for parameter(s), including the conditional mean. A common way to obtain confidence/prediction intervals for simultaneous inference across the sampling domain is to use resampling methods (e.g., bootstrapping or permutation). We propose a new method for estimating parametric, simultaneous confidence and prediction bands for a functional concurrent regression model, without the use of resampling. The method uses Kac-Rice formulas for estimation of a critical value function, which is used with a functional pivot to acquire the simultaneous band. In the results, the proposed method meets nominal coverage levels for both confidence and prediction bands. The method we propose is also substantially faster to compute than methods that require resampling techniques.

In linear regression, researchers may also assess if there are influential observations that may impact the estimates and results from the fitted model. Studentized difference in fits ($DFFITS$), studentized difference in regression coefficient estimates ($DFBETAS$), and/or Cook's Distance (D) can all be used to identify influential observations. For functional concurrent regression, these measures can be easily computed pointwise for each observation. However, the only current development is to use resampling techniques for estimating a null distribution of the average of each measure. Rather than using the average values and bootstrapping, we propose working with functional $DFFITS$ ($DFFITS(t)$) directly. We show that if the functional errors are assumed to follow a Gaussian process, $DFFITS(t)$ is distributed uniformly as a scaled Student's t process. Then, we propose using a multivariate Student's t distributional quantile for identifying influential functional observations with $DFFITS(t)$. Our methodology ("Theoretical") is compared against a competing method that uses a parametric bootstrapping technique ("Bootstrapped") for estimating the null distribution of the mean absolute value of $DFFITS(t)$. In the simulation and case study results, we find that the Theoretical method greatly reduces the computation time, without much loss in performance as measured by accuracy (ACC), precision (PPV), and Matthew's Correlation Coefficient (MCC), than the Bootstrapped method. Furthermore, the average sensitivity of the Theoretical method is higher in all scenarios than the Bootstrapped method.

ACKNOWLEDGEMENTS

This work utilized the Alpine high performance computing resource at the University of Colorado Boulder. Alpine is jointly funded by the University of Colorado Boulder, the University of Colorado Anschutz, and Colorado State University.

DEDICATION

I would like to thank the Department of Statistics at Colorado State University and especially my advisors, Dr. Julia L. Sharp and Dr. Dan Cooley, and my committee members, Dr. Dominik Liebl, Dr. Matt Koslovsky, and Dr. Francisco Ortega, for their continued support throughout the years. I would also like to thank all of my family and friends that have supported me along the way.

I am dedicating this dissertation thesis to my partner and best friend in life, Jordan Ann Creutzinger, and my father and mentor, John Lawrence Creutzinger. Without them, none of this would have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Defining Functional Data

Functional data are defined as data collected on a curve, or surface, over a continuum. Similar to time series data, a common functional data continuum is time. Other common examples of continuum used for functional data are spatial location, wavelength, probability, and more (Ramsay and Silverman, 2005). As an example, consider the Sprint Start Kinetics of Amputee and Non-Amputee Sprinters, a data set originally collected and described by Willwacher et al. (2016). The study sample in Willwacher et al. (2016) includes 154 non-amputee sprinters with a wide range of 100-meter sprint performance levels (100m personal records (PRs), 9.58s - 14.00s). Willwacher et al. (2016) used a custom-made instrumented starting block to obtain the force data of a sprint start. The analog force signals were converted to digital at a sampling rate of 10,000 Hz. After force signals were converted from analog to digital, the sampling domain was scaled from 0-100 % of the Push-Off Phase (from starting force to finished force). The resulting data frame consists of 161 functional observations, observed at 101 sampling points. In this example, the continuum is time, like in a time series data set. The Sprint Start Kinetics data can be indexed by the time that each individual sprinter was measured and observed. An entire function measured on an individual in a functional data set is considered one observation, rather than a single observed value at a sampling point for the individual. Considering functions as observations is the key distinction that separates functional data from other forms of data. The collection of 161 sprinters' start force measured is an example of a univariate sample of functional data, where the function measured is force.

Even though a functional observation is only observed or measured at each sampling point, it could hypothetically be measured at an infinite number of sampling points. Functional observations are defined as infinite-dimensional, because they can be measured without limit on the number of sampling points (Ramsay and Silverman, 2005). The resolution, or sampling frequency,

is decided by how many sampling points are observed or measured for each functional observation. The more sampling points that are observed, the higher the resolution of the functional data (Ramsay and Silverman, 2005). Although functional data are considered infinite in dimension, the resolution of functional data is determined by the measurement scale, the way the data are measured, or the environment or behavior of the unit measured. For example, it takes approximately 1-2 minutes to record a single value of blood pressure (National Center for Chronic Disease Prevention and Health Promotion, 2021). Alternatively, an accelerometer device has the ability to record acceleration movements at frequencies of 1000 Hz (or 1000 measurements per second) or more. Neither of these examples measure the functional data with an infinite resolution, but the accelerometer clearly records data at a higher resolution than a blood pressure gauge.

Devices that record functional data at a high resolution, such as an accelerometer, are commonly found in health, environmental, finance, and other research areas. Recent technological advancements have allowed more scientific fields to develop high resolution devices, like the force plates in the Sprint Start Kinetics study. The growing presence of high resolution data has greatly increased the popularity of using methods in functional data analysis (FDA). This rise in popularity has resulted in more recent and continual development of FDA methods. However, there is still a lack of methods that are available to analyze functional data from "start to finish". A common way around this is to estimate the mean/median of each functional observation, so that a non-functional linear model can be used for regression. When a functional observation is reduced to a point estimate, it is possible that too much information is lost to gain reliable inference from the data.

The methods presented in this dissertation provide a statistical framework that allows a researcher to explore the functional data sample, concurrently regress a functional response variable on functional and/or scalar predictor variable(s), make inference from the functional concurrent regression model through simultaneous confidence and prediction bands, and check for functional observations that may be influential to the model. Code is developed in `R` to carry out all the methods presented.

1.2 Methods in Functional Outlier Detection

1.2.1 Defining Functional Outliers

Functional data may be defined differently from other data structures, but data analysis techniques similar to non-functional data are still relevant. For example, even in functional data, there is a need to identify outliers prior to statistical analysis procedures. Febrero et al. (2008) describe two ways that outliers can be present in functional data. First, outliers in functional data can result from a faulty measuring device, observational error, or recording error. Second, an outlier in functional data could represent an observation that was correctly measured and recorded, but varies from the majority of the other observations. An observation can exhibit as an outlier in an isolated area of the sampling domain, or across the entire sampling domain.

A functional outlier can be categorized as a shape outlier, shift outlier, amplitude outlier, or some combination of these types (Hubert et al., 2015). A shape outlier is defined as a functional observation with a different trend across the sampling domain than the majority of the other functional observations. For example, if the majority of functional observations follow a flat trend line centered around zero plus some noise, then a shape outlier could be a functional observation that follows a sinusoidal pattern, centered at zero, plus some noise. A shift outlier is defined as a functional observation whose observed values are consistently larger (or smaller) than the observed values of the majority of the other functional observations. For example, if the majority of functional observations follow a flat trend line centered around zero plus some noise, then a shift outlier would be a functional observation that also follows a flat trend line plus some noise, but centered at a positive/negative number relatively far from zero. An amplitude outlier is defined as a functional observation that has the same shape as the majority of other functional observations, but is different in scale. For example, if the majority of functional observations follows a sinusoidal trend plus some noise, then an amplitude outlier could be a functional observation that also follows a sinuosoidal trend plus some noise, but the peaks and valleys of the trend magnified away from the center of the majority of the other functional observations.

A main concern of outlier detection is the effect of "masking." Any observation that is identified as a potential outlier after first removing obvious and extreme outliers from the data set, is an example of a masked outlier (Bendre and Kale, 1987). Masking often occurs when using a statistical measure that is susceptible to outliers, such as the sample mean and standard deviation. The susceptibility of a statistic to outliers is also related to the "breakdown point." The breakdown point of a statistic can be defined as the proportion of outlying observations that can be present in a random sample before the results of the estimator are no longer correct (Hampel, 1971).

1.2.2 Current Methodology in Functional Outlier Detection

Methods in functional data outlier detection have been well-developed over the years, and an overview can be found in Ojo et al. (2021). Several outlier detection methods use a measure of functional depth. In general, functional depth is an estimator of the relative position of all functional observations, with respect to the "center" of the functional observations (Zuo and Serfling, 2000). Common examples of depth functions include Fraiman and Muniz Depth (FMD) (Fraiman and Muniz (2001)), H-Modal Depth (MD) (Cuevas et al. (2006)), Random Projection Depth (RPD) (Cuevas et al. (2007)), and Tukey's half-space location depth (Tukey (1975)). Huang and Sun (2019) developed the Total Variate Depth (TVD) and Modified Shape Similarity index for functional outlier detection. Febrero et al. (2008) also developed functional outlier detection by comparing depth values.

There has also been development in functional outlier detection methods that do not rely on functional depth measures. Functional outlier detection implemented by Febrero et al. (2007) uses a likelihood ratio test and bootstrapping techniques to identify the outliers. Hyndman and Shahid Ullah (2007) estimates the integrated squared errors of the functional prinicipal components, with large errors associated with outliers. Dai and Genton (2018) developed functional "outlyingness" measures (e.g., Stahel-Donoho estimated on a function) and the Magnitude-Shape (MS) Plot to identify functional outliers (Dai and Genton, 2019). In a similar approach, Azcorra et al. (2018) developed functional "indices" related to shape, magnitude, and amplitude types of

outliers. A large value for an index implies that the observation is a functional outlier in that regard (e.g., a large shape index would imply a functional shape outlier). Lastly, Dai et al. (2020) proposed a sequential transformation method, which can incorporate different functional outlyingness measures, functional depth measures, and several other forms of transformations.

Outlier detection can also be thought of as a clustering technique (for e.g., with two clusters for outliers vs non-outliers), and clustering methods do exist for functional data analysis. A review of functional clustering methods is provided by Zhang and Parnell (2023). However, the majority of clustering algorithms require the user to decide on the number of clusters. If only two clusters are decided, it is not guaranteed that the two clusters identified will represent outliers and non-outliers. Even if a supervised clustering algorithm is leveraged, there could be more than one distinct type of functional outlier present, making it nearly impossible a priori to express the clustering rules ahead of time with a supervised clustering method.

To date, there are no methods in functional outlier detection that make direct use of ordinary summary statistics, such as the mean, median, variance, maximum, minimum, etc. Instead, common functional outlier detection methods require the comprehension of a newly developed functional data measure, which is typically used in a complex detection algorithm.

1.3 Methods in Functional Concurrent Regression

After the data have been examined for outliers, inferential techniques – in particular, simultaneous inference – can be considered. Most inferential procedures currently developed for FDA implement the statistical test at each time point observed along the function. For example, Ramsay and Silverman (2005) developed a two-sample, independent t-test for functional data, which is an extension of the two-sample, independent t-test for non-functional data. The functional t-test developed by Ramsay and Silverman (2005) is independently applied and measured at each sampling point. In this functional t-test, a permutation test is used to generate the critical value for a test of significance at each sampling point. Several other inferential procedures in FDA were created as an extension of well-known statistical tests, which are conducted independently at each

sampling point (Ramsay and Silverman, 2005). The fundamental idea of FDA is that an observation is a measured function(s) on an individual, not a single sampling point along the continuum of the function(s). Treating observations functionally makes it advantageous to use simultaneous inferential procedures, rather than testing at each sampling point, to ensure more robust inference.

A common analysis choice to acquire inferential results is the use of linear regression. In FDA, the time-varying coefficient model (Hastie and Tibshirani, 1993), or functional concurrent regression (Ramsay and Silverman, 2005, Ch. 14), considers a functional linear regression model with a functional response, scalar and/or functional predictors. The model is "concurrent", because it is regressed pointwise at each sampling point in the domain. After estimating a regression model (functional or non-functional), it is common to acquire confidence and prediction intervals for the parameter(s), including the conditional prediction of a new observation. Similar to the t-test, previous applications of confidence and prediction intervals in FDA use non-functional data rules applied at each sampling point (Ramsay and Silverman, 2005). For example, if a univariate sample of functional data is presumed to arrive from a Gaussian process, a confidence band could be constructed at each individual sampling point.

Calculating a confidence interval pointwise for all sampling points in the domain can create an issue of inflated Type I error rate. A common way to extend confidence and prediction intervals for simultaneous inference across the sampling domain is to use resampling methods such as bootstrapping and/or permutation. The use of resampling methods results in a critical value function that allows creation of simultaneous confidence/prediction bands with nominal coverage levels. Olshen et al. (1989), Lenhoff et al. (1999), Degras (2017), Wang et al. (2020), and others all use resampling methods for simultaneous inference. For the previously mentioned methods, the simultaneous bands reach nominal coverage levels. However, implementing resampling methods can require a large amount of time and computing power.

More recently, methods have emerged for simultaneous intervals that do not require resampling methods. Liebl and Reimherr (2023) leverage Kac-Rice formulas from Random Field Theory in order to estimate a critical value function, which is allowed to vary across the sampling domain.

Telschow and Schwartzman (2022) also propose a method for simultaneous confidence bands by leveraging ideas in Random Field Theory. Namely, Telschow and Schwartzman (2022) develop their simultaneous confidence bands by using the Gaussian Kinematic Formula of Student's t processes (tGKF) to estimate the critical value for the band. However, both of these methods for creating simultaneous bands have only been developed for a univariate sample mean or the difference of two sample means. Without the use of resampling techniques, there is currently no parametric methodology to acquire a simultaneous confidence/prediction band for a conditional, predicted mean of a functional concurrent regression model (Torti et al., 2020). The only other parametric methodology for simultaneous bands in functional regression is developed by Ecker et al. (2024). The methodology of Ecker et al. (2024) also builds on Liebl and Reimherr (2023), but Ecker et al. (2024) only considers the case of Gaussian errors.

Conformal inference is a non-parametric approach for creating simultaneous confidence or prediction bands, and conformal inference was recently extended from non-functional linear regression to functional concurrent regression (Vovk and Shafer, 2008; Diquigiovanni et al., 2022; Fontana et al., 2023). However, when using conformal inference for small sample sizes, the resulting simultaneous band tends to be off-centered and too wide for reliable inference. Furthermore, Conformal Inference is not able to condition on the covariate values for a new prediction.

1.4 Functional Influential Measures of Concurrent Regression

After estimating a linear regression model (either in the functional setting or not), it is common to assess the model with diagnostic measures and to check for influential observations. An influential observation is one that substantially influences the fit of the estimated regression model. If the regression model is estimated with the influential point, the estimated coefficients of the model can be substantially different than if the regression model is estimated without the influential point. In some examples, an estimated coefficient could swap signs (e.g. "+" or "-") depending on the presence of the influential point.

In non-functional statistical analyses, a common method to find influential observations in linear regression is to calculate the difference in fit ($DFFIT$) for each observation. The $DFFIT$ of an observation is the change in predicted value of an observation when using the observation in the regression model versus leaving the observation out (Belsley et al., 2005). An observation with a large magnitude of $DFFIT$, may be considered an influential observation. Another common approach is to calculate the studentized $DFFIT$, $DFFITS$.

As shown by Seber and Lee (2003), when the errors are assumed to follow a Gaussian process, $DFFITS$ are distributed as a scaled Student's t with $n - K - 1$ degrees of freedom (for n observations and K covariates). A common rule for deciding influential points in non-functional linear regression, is to compare the estimated $DFFITS$ with a quantile from the scaled Student's t distribution (Belsley et al., 2005). Other measures of influence have been introduced for non-functional linear regression, such as $DFBETA(s)$, Cook's Distance (D), and the "hat" matrix. My dissertation focuses on development of $DFFITS$ in a functional setting.

Some efforts have been made to extend non-functional linear regression influential measures to the functional concurrent regression model. Similar to the development of simultaneous bands for a functional concurrent regression, current methods incorporate resampling techniques to make inference on the estimated influential measures. For example, most recently, Pittman (2022) calculated $DFFITS$, $DFBETAS$, and D, at each sampling point. Then, Pittman (2022) calculated the average of each influential measure, by averaging over the sampling domain. With the average of each influential measure for each observation, Pittman (2022) used bootstrapping techniques to estimate the null distribution of each influential measure, and used percentiles from this estimated null distribution to determine the influential observations in the functional concurrent regression. At this time, there does not exist a parametric approach for inference on estimated functional influential measures, which does not rely on resampling techniques.

1.5 Dissertation Layout

In this dissertation, we propose new methods to assist in the analysis of a functional random sample. In Chapter 2, new methods of outlier detection for a univariate sample of functional data, which has been submitted for publication at The American Statistician (Creutzinger and Sharp, 2024). A new method of simultaneous confidence/prediction bands for the conditional mean of a functional concurrent regression model is presented in Chapter 3 and will be submitted for publication at The Journal of the American Statistical Association, Applications and Case Studies (Creutzinger et al., 2024). In Chapter 4, a new method for detecting influential observations in a functional concurrent regression model is presented (Chapter 4). My dissertation concludes with a discussion on these new methods, including limitations and future work (Chapter 5).

Chapter 2

New Methods for Functional Outlier Detection

Abstract *Existing functional data outlier detection methodology requires the use of a functional data depth measure, functional principal components, likelihood ratio, and/or an outlyingness measure like Stahel-Donoho. Although effective, these functional outlier detection methods may not be easily interpreted. In this paper, two new methods are proposed for functional outlier detection: Practical Outlier Detection (POD) and Prediction Band Outlier Detection (PBOD). POD was developed with the use of summary statistics to be easily interpretable. PBOD was developed with the use of a new simultaneous prediction band for a univariate sample of functional data. POD was found to be as good or better (in terms of accuracy, precision, and Matthew's Correlation Coefficient) than competing methodology, especially for the identification of shape outliers. POD and PBOD were assessed and compared to other methodology using simulation studies and a case study with World Population Growth data.*

2.1 Introduction

2.1.1 Defining Functional Data

Thorough statistical analyses begin with exploratory data analysis, including figures, tables, and summary statistics. Before the data are used for statistical inference or estimation, it is important to consider if outliers are present, because the presence of outliers can potentially lead to biased inferential results and estimates. When data are defined functionally, they are often collected at a high resolution (e.g., frequency of data collection across time, or some other sequential sampling domain) (Ramsay and Silverman, 2005). A functional observation can be denoted as $X = \{X(t) : t \in [0, 1]\}$, where the domain of the sample is restricted to $[0, 1]$ without loss of generality. The sampling points where the function is observed are denoted as t_j for $j = 1, \ldots, T$. Thus, each functional observation can be expressed as a vector of length T and a univariate func-

tional sample can be expressed as $\{X_i\}_{i=1}^n$, for $i = 1, \ldots, n$ observations. That is, at each time point, t_j, the same function is measured on n unique observations.

An "event" can be defined as the behavior of a functional observation between two sampling points, t_j and $t_{j'}$, $j \neq j'$, in the domain (e.g., a person's activity between 4am and 6am). As the resolution of functional data increases (e.g., the value T), the possible number of events that can be estimated also increases. If the function were to be observed continuously, there would be infinitely many events. With infinitely many events, estimation of a sample mean in functional data can be thought of as estimating infinitely many points.

The presence of functional outliers can impact the mean of a functional sample, just like an outlier can influence the mean of a non-functional sample. However, a functional outlier can influence the estimation of infinitely many points along the sampling domain, rather than only a single point for non-functional data. If one or more observations are severe functional outliers, then other, less severe, functional outliers may be "masked". Any observation that is identified as a potential outlier after first removing obvious and severe outliers from the data set, is an example of a masked outlier (Bendre and Kale, 1987). Thus, detecting an outlier in a functional data sample is an important and necessary first step for functional data analysis (FDA).

2.1.2 Outliers in Functional Data and Classification Diagnostics

In FDA, outlying observations can be identified as one or more of the following: magnitude outliers, shape outliers, and/or amplitude outliers (Hubert et al. (2015), Ojo et al. (2021)). Outlying features may be present across the whole domain of data collection (e.g., $t \in [0, 1]$), or in one or more sub-intervals of the domain. Methods in functional data outlier detection have been well-developed to identify outliers in a univariate random sample. An overview of methods can be found in Ojo et al. (2021). In addition, Ojo et al. (2021) created an `R` package, `fdaoutlier`, which includes functions to implement outlier detection methods and functions. The `fdaoutlier` package also has functions (e.g., `simulation_model1()`) for nine data generation models to aid in developing and testing functional outlier detection methods (Figure 2.1). All of the data generation

functions in `R` allow the user to specify the sample size (n), number of sampling points (T), the proportion of sample size created as outliers (Δ), and parameters that control the variation of the functional sample. The nine data generation models represent random samples of functional observations with one or more type (i.e., magnitude, shape, or amplitude) of functional outliers present. For example, a case of magnitude outliers is represented by Model 1, a case of shape outliers is represented by Model 7, and a case of amplitude outliers is represented by Model 9 (Figure 2.1; Ojo et al. (2022)). A variety of generation models can be especially helpful in developing function outlier detection methods, allowing researchers to test methodology on a wide variety of functional outlier types. The variety of models also helps a researcher test how well a given methodology can identify the type of functional outlier, if possible with a given method.

In a sample of functional observations, a "positive" (P) is defined to be an observation that represents an outlier, while a "negative" (N) is defined to be an observation that does not represent an outlier. When using a method to classify the observations as outliers or not, a "true" classification means that the observation is correctly identified as positive or negative. By comparison, a "false" classification means that an observation is incorrectly identified as positive or negative. Combining terms, a "true positive" (TP) is an outlying observation that is correctly classified as an outlier. In contrast, a "false positive" (FP) is a non-outlying observation that is incorrectly classified as an outlier. A "true negative" (TN) and "false negative" (FN) are defined analogously.

A common choice for model classification diagnostics is the true positive rate, $TPR = \frac{TP}{(TP+FN)}$ (sensitivity), and false positive rate, $FPR = \frac{FP}{FP+TN}$ ($1 -$ specificity). If a researcher would like to diagnose how well they can identify either class correctly (positive or negative), then it is common to combine sensitivity and specificity into one measure, such as accuracy or precision, for comparing classification methods. Accuracy is described as how close a given set of measurements are to their true value, computed as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}.$$

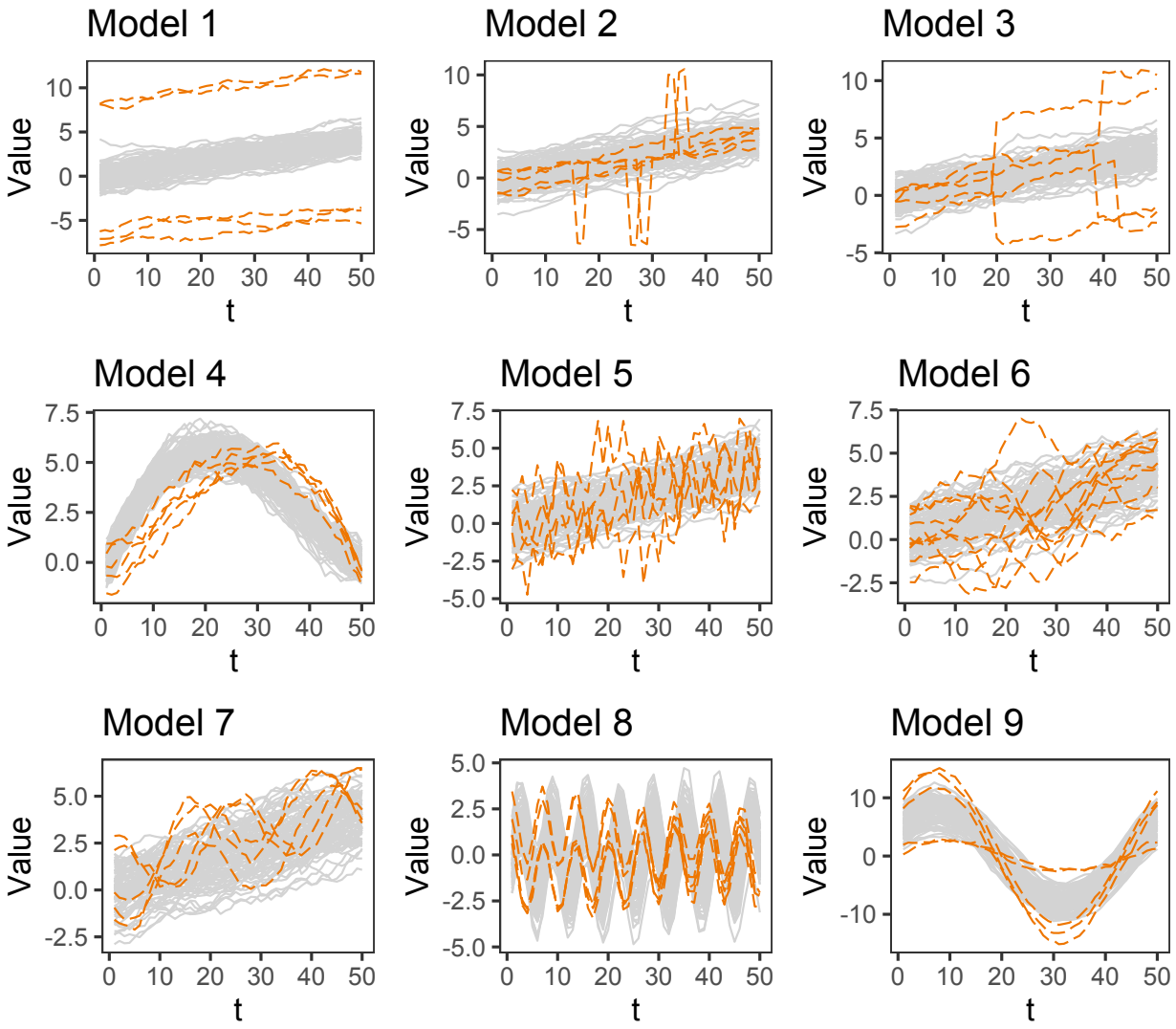


Figure 2.1: A plot of random samples generated by the functions presented in Ojo et al. (2021). The random samples were generated using the default parameters of each function in R. Outliers are plotted as red lines (dark gray in black and white) and all other observations are plotted as light gray.

Precision (or positive predictive value) is described as how close a given set of measurements are to one another, computed as

$$PPV = \frac{TP}{TP + FP}.$$

In this paper, Matthew's Correlation Coefficient (MCC) is used as the main metric for comparison. MCC is described as the correlation between the observed and predicted binary classifications, and is computed as

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}.$$

MCC is a value between -1 (worst) and 1 (best) and is undefined when any component of the denominator (for e.g., TP + FP) is zero. Boughorbel et al. (2017), Delgado and Tibau (2019) , and Chicco and Jurman (2020) show that MCC is regarded as a robust measure, which can be used even if the classes (for e.g., TP, TN, etc) are imbalanced (i.e., the proportion of positives is larger or smaller than the proportion of negatives). By comparison, Wardhani et al. (2019) show that ACC and PPV are not robust to imbalanced classes and may be misleading. The proportion of functional observations considered to be an outlier can be relatively small (for e.g., less than 15% of the observations), which results in imbalanced classes. Thus, ACC and/or PPV may be unreliable as a measure to compare methods.

For example, suppose there are five outliers in 100 functional observations. A method is used to identify outliers ("Method 1") and one of the five outlying observations is correctly classified as an outlier. The remaining 99 observations are not classified as outliers when Method 1 is used. The classification results of Method 1 are $TP = 1$, $TN = 95$, $FP = 0$ and $FN = 4$. The ACC of Method 1 is $\frac{1+95}{100} = 0.96$ and the PPV is $\frac{1}{1+0} = 1$. The sensitivity of Method 1 is 0.20 and the specificity is 1. Yet, the MCC is 0.44. Then, some other method is used ("Method 2") and two of the five outliers are classified correctly. The remaining 98 observations are not classified as outliers using Method 2. The ACC of Method 2 is $\frac{2+95}{100} = 0.97$ and the PPV is $\frac{2}{2+0} = 1$ (the same as Method 1). The sensitivity of Method 2 is 0.40 and the specificity is 1. The MCC of Method 2 is 0.62; a clear increase in performance when compared to Method 1 (Table 2.1). The sensitivity is also noticeably different between the two methods, but this measure does not fully describe and compare the two methods.

2.1.3 Existing Functional Outlier Detection Methods

Several functional outlier detection methods use a measure of functional depth. Zuo and Serfling (2000) first described the main properties of a depth function, with respect to non-functional,

Table 2.1: A comparison of classification diagnostics through a toy example. The toy example has five true outliers out of 100 observations.

Method	Sensitivity	Specificity	ACC	PPV	MCC
Method 1	0.20	1.00	0.96	1.00	0.44
Method 2	0.40	1.00	0.97	1.00	0.62

multivariate data: affine invariance, maximality at center, monotonicity relative to deepest point, and vanishing at infinity. These properties and the definition of a depth function for functional data was later defined by Gijbels and Nagy (2017). Common examples of depth functions include Fraiman and Muniz Depth (FMD) (Fraiman and Muniz (2001)), H-Modal Depth (MD) (Cuevas et al. (2006)), Random Projection Depth (RPD) (Cuevas et al. (2007)), and Tukey's half-space location depth (Tukey (1975)). Febrero et al. (2008) developed a method for functional outlier detection using any of these common functional depth functions, $D_n(X_i)$, for univariate sample $\{X_i\}$. Febrero et al. (2008) propose the following procedure for functional outlier detection:

1. Obtain the functional depths, $D_n(X_i)$, of all functional observations (using FMD, MD, or RPD).

2. If $D_n(X_i) \leq C$, then X_i is a functional outlier; where C is a given cutoff value.

3. Remove all X_i marked as a functional outlier. Repeat these steps until no more outliers are removed.

Recently, Huang and Sun (2019) introduced a functional depth measure called the "Total Variation Depth" (TVD). Given a real-valued functional observation X on interval I with distribution F_X, Huang and Sun (2019) define the pointwise total variation depth as

$$D_f(t) = \text{var}\left\{R_f(t)\right\},$$

where $R_f(t) = \mathbb{I}\{X(t) \leq f(t)\}$. If $p_f(t)$ denotes $E[R_f(t)] = \mathbb{P}(X(t) \leq f(t))$, then $D_f(t) = p_f(t)(1 - p_f(t))$. TVD is then defined as

$$TVD(f) = \int_I w(t) D_f(t) dt,$$

where $w(t)$ is a weight function defined on interval I.

In addition to TVD, Huang and Sun (2019) define the modified shape similarity (MSS) of a function f on interval I. The MSS of f given a fixed time span, Δ, with respect to the distribution F_X is

$$MSS(f; \Delta) = \int_I v(t; \Delta) S_{\tilde{f}}(t; \Delta) dt,$$

where $v(t; \Delta)$ is a weight function and for any pair $(t - \Delta, t)$, $\tilde{f}$ is given by

$$\tilde{f}(s; \Delta) = \begin{cases} \text{median}\{X(s)\} & , s = t \\ f(s) - f(s + \Delta) + \text{median}\{X(s + \Delta)\} & , s = t - \Delta, \end{cases}$$

and $S_{\tilde{f}}(t; \Delta) = \text{var}\left(\mathbb{E}\left[R_{\tilde{f}}(t) | R_{\tilde{f}}(t - \Delta)\right]\right) / D_{\tilde{f}}(t)$.

Huang and Sun (2019) use MSS and TVD for functional outlier detection, specifying the outlier type as shape or magnitude outliers. Huang and Sun (2019) define a magnitude outlier as observations that have very deviated values in some dimensions (i.e., any observation that is a magnitude and/or amplitude outlier). Using a classical boxplot rule (e.g., $F \times IQR$, where F is adjusted by users; typically, $F = 1.5$ in univariate settings), functional observations with MSS below the lower fence are identified as shape outliers and removed. After shape outliers are removed, TVD is used to identify magnitude outliers with a functional boxplot approach (Sun and Genton, 2011).

Aside from outlier detection methods for functional data that use functional depth functions, other methods leverage functional principal components (for e.g., Hyndman and Shahid Ullah (2007)), a likelihood ratio statistic (for e.g., Febrero et al. (2007,0)), and/or outlyingness measures (for e.g., Rousseeuw et al. (2018)). Dai and Genton (2019) propose a measure of functional outlier

detection that makes use of a functional, directional outlyingness measure. Dai and Genton (2019) start by considering a functional observation, X_i, in the space of real continuous functions defined on a compact interval, I. They denote the probability distribution of X as F_X, and the pointwise probability distribution of $X(t)$ as $F_{X_{(t)}}$. Dai and Genton (2019) define the directional outlyingness for multivariate data as

$$O\left(X, F_X\right) = o\left(X, F_X\right) \cdot v,$$

where $o\left(X, F_X\right)$ is the outlyingness of X with respect to F_X and v is the spatial depth defined at point t by

$$v(t) = \frac{|X(t) - Z(t)|}{||X(t) - Z(t)||},$$

with $Z(t)$ being the unique median of $X(t)$ with respect to $F_{X(t)}$ (deepest point of $F_{X(t)}$). In other words, $v(t)$ is a unit vector pointing from $Z(t)$ to $X(t)$.

Dai and Genton (2019) recommend a distance-based measurement for outlyingness, such as the Stahel-Donoho Outlyingness measure. Dai and Genton (2019) define the functional directional outlyingness (FO) as

$$FO\left(X, F_X\right) = \int_I \left|\left|O\left(X(T), F_{X(t)}\right)\right|\right|^2 w(t)dt,$$

where $w(t)$ is a weight function defined on interval I. The mean directional outlyingness (MO) and variation of directional outlyingness (VO) are defined as

$$MO(X, F_X) = \int_I O\left(X(T), F_{X(t)}\right) w(t)dt,$$

and

$$VO(X, F_X) = \int_I \left|\left|O\left(X(T), F_{X(t)}\right) - MO(X, F_X)\right|\right|^2 w(t)dt,$$

which measure the magnitude and shape outlyingness, respectively. Dai and Genton (2019) show that FO can be decomposed into the magnitude and shape outlyingness:

$$FO(X, F_X) = ||MO(X, F_X)||^2 + VO(X, F_X).$$

An MS-Plot is a scatter-plot of the points $\left(MO^T, VO\right)^T$. For outlier detection, Dai and Genton (2019) form a multivariate data set whose columns are MO and VO. With this data set, the Mahalanobis distance is calculated for each $\left(MO^T, VO\right)^T$ pair in the data. A robust covariance matrix is estimated using the minimum covariate determinant (MCD) estimator (Rousseeuw and Driessen, 1999). The distribution of these robust distances is approximated using the F distribution (Hardin and Rocke, 2005). Then, any observation with a robust distance greater than the cutoff obtained from the tails of the F distribution is flagged as an outlier.

Outlier detection using sequential transformations is a method developed by Dai et al. (2020). The method by Dai et al. (2020) makes use of the functional boxplot for detecting outliers after each transformation made to the functional data. Dai et al. (2020) discuss three possible sequences of transformations: shifting and normalization of curves, derivatives of curves, and directional outlyingness. As a result, Dai et al. (2020)'s method is a combination of functional depth and functional outlyingness measures.

A sample of functional observations in the space of continuous functions $C(I)$ defined on interval $I \in \mathbb{R}$, is denoted as $\{X_i\}_{i=1}^n$. The distribution of functional observations, X_i, is denoted as F_X, and a transformation of functional observations is denoted as Γ. The distribution of transformed data, $\{\Gamma(X_i)\}_{i=1}^n$, is denoted as $F_{\Gamma(X)}$. The number and type of transformations can be decided by the user. After a sequence of transformations is chosen, functional outlier detection is performed through several steps.

The first step in Dai et al. (2020)'s functional outlier detection is to identify the magnitude outliers using a functional boxplot and the original sample of functional observations, $\{X_i\}_{i=1}^n$. The set of magnitude outliers identified by the functional boxplot is denoted as S_0, and the magnitude outliers in S_0 are called the Γ_0-outliers. Here, subscript 0 represents no transformation has been

applied. The second step is to apply the first transformation, Γ_1, from the chosen sequence of transformations. Applying Γ_1 to functional sample, $\{X_i\}_{i=1}^n$, gives the transformed functional sample, $\{\Gamma_1(X_i)\}_{i=1}^n$. A functional boxplot is used on the transformed functional sample, $\{\Gamma_1(X_i)\}_{i=1}^n$, and returns a new set of functional outliers, denoted by S_1. The functional outliers listed in S_1 but not in S_0 ($S_1 \backslash S_0$), are called Γ_1-shape outliers. The third step is to apply the second transformation, Γ_2, to the functional sample acquired from the first transformation, $\{\Gamma_1(X_i)\}_{i=1}^n$. Then, a second sample of transformed functional data is retained, $\{\Gamma_2 \circ \Gamma_1(X_i)\}_{i=1}^n$. Once again, a functional boxplot is used to identify a new set of outliers, S_2, after applying transformation Γ_2. The functional outliers listed in S_2 but not listed in S_1 nor listed in S_0, are called the $\Gamma_2 \circ \Gamma_1$-shape outliers. This process continues until all desired transformations have been applied. The final result is a list of all outliers identified, unique to each transformation performed.

In addition to methods that use functional depths, a likelihood ratio statistic, outlyingness measure, or functional principal components, Azcorra et al. (2018) proposed "Massive Unsupervised Outlier Detection" (MUOD). To implement MUOD, three indices for the shape, magnitude, and amplitude, are calculated for each functional observation, X_i. Specifically, the shape index of X_i, denoted $I_S(X_i)$, is defined as

$$I_S(X_i) = \left| n^{-1} \sum_{i'=1}^{n} \hat{\rho}\left(X_i, X_{i'}\right) - 1 \right|, \quad i \neq i',$$

where $\hat{\rho}$ is the Pearson correlation coefficient, given by

$$\hat{\rho}\left(X_i, X_{i'}\right) = \frac{\text{cov}\left(X_i, X_{i'}\right)}{s_{X_i} s_{X_{i'}}},$$

where s denotes the sample standard deviation, and $s_{X_i}, s_{X_{i'}} \neq 0$.

The magnitude and amplitude indices are based on simple linear regression. When calculating the magnitude and amplitude indices for observation X_i, the observed values of the function, $x_{ij} = \{X_i(t_j) : t_j \in [0,1], j = 1, \ldots, T\}$, are defined as the dependent variable. The observed values of all other functions, $x_{i'j}, i \neq i'$, are each used as an independent variable in a simple linear

regression. For each simple linear regression, the slope and intercept are estimated as

$$\hat{\beta}_{i'} = \frac{\text{cov}\left(X_i, X_{i'}\right)}{s^2_{X_{i'}}}, \quad s^2_{X_{i'}} \neq 0, \quad \text{and} \quad \hat{\alpha}_{i'} = \bar{x_i} - \hat{\beta}_{i'}\bar{x_{i'}},$$

where $\bar{x_i}$ denotes the estimated mean of the observed functional values from functional observation X_i. Using the estimated slopes and intercepts obtained, the magnitude and amplitude indices are defined as

$$I_M\left(X_i\right) = \left| n^{-1} \sum_{i'=1}^{n} \hat{\alpha}_{i'} \right|, \quad \text{and} \quad I_A\left(X_i\right) = \left| n^{-1} \sum_{i'=1}^{n} \hat{\beta}_{i'} - 1 \right|,$$

respectively. As with other outlyingness measures, a larger index value, I, is related to a greater chance of the observation being an outlier of that type.

Azcorra et al. (2018) proposed a tangent method ("tan") for determining a cutoff for outlier detection. The tangent method is used by searching for the line tangent to the maximum index (for each index separately). The x-intercept of this tangent line is used as the cutoff value. Any functional observation with an estimated index larger than the cutoff is identified as a functional outlier (of that type). However, Vinue and Epifanio (2021) showed that when using the tangent cutoff method, several observations would be incorrectly identified as an outlier (i.e., false positives). Alternatively, Azcorra et al. (2018) propose using the upper fence of Tukey's classical boxplot rule for the cutoff of each index.

The functional outlier detection methods discussed perform relatively well in terms of TPR (sensitivity), FPR (1 - specificity), and/or ACC (see Section 2.1.2). Huang and Sun (2019) report that TVD had a TPR of 0.98 or higher and a FPR of 0.07 or lower (with the exception of one simulation model that showed a FPR of 0.25). Febrero et al. (2007) report their LRT method had an ACC of 0.97 or higher and a FPR of 0.02 or lower. Hyndman and Shahid Ullah (2007) report their functional principal component method had a TPR of 1.00, but do not describe other measures. Rousseeuw et al. (2018) do not report values of TPR or FPR, but show the functional adaptation of the Stahel-Donoho measure identifies more true outliers than the Adjusted Outlyingness proposed by Brys et al. (2005). Dai et al. (2020) report that sequential transformations had a

TPR of 0.77 or higher and a FPR of 0.15 or lower, when using the L^{∞} depth developed by Long and Huang (2015). Dai and Genton (2019) report that MS-Plot had a TPR of 0.90 or higher and a FPR of 0.01 or lower. Azcorra et al. (2018) report that MUOD had a TPR of 0.95 or higher and a FPR of 0.07 or lower.

Some functional outlier detection methods can be used to only identify outliers (for e.g., Dai and Genton (2019)'s magnitude-shape (MS) Plot), while other methods can be used to identify outliers and the type of each outlier (for e.g., Huang and Sun (2019)'s TVD method). Some can be used to identify all three types: magnitude, shape, and amplitude (for e.g., Azcorra et al. (2018)'s MUOD), while others can be used to only identify magnitude and shape outliers, where amplitude outliers are considered shape outliers (for e.g., Ojo et al. (2022)'s TVD). The LRT method by Febrero et al. (2007), functional principal component method by Hyndman and Shahid Ullah (2007), adjusted directional outlyingness by Rousseeuw et al. (2018), and MS-Plot by Dai and Genton (2019) can not be used to identify the type of functional outlier. None of the previously mentioned methods were assessed via simulations for classifying the type of functional outlier specifically.

2.1.4 New Methods for Functional Outlier Detection

Many methods available to identify outliers rely on statistical measures that are not easily interpretable. For example, a depth measure (functional or non-functional) is not usually covered in an introductory statistics class. The first method proposed here, Practical Outlier Detection (POD), is a practical, easily interpretable approach to outlier detection. The method is developed using summary statistics, which are introduced in nearly all introductory statistics courses, estimated on several disjoint intervals after binning. In this way, any functional observation can be reduced to a finite dimension of statistics describing its behavior, similar to graph-theoretic scagnostics for non-functional data (Tukey and Tukey, 1985; Wilkinson et al., 2005). After implementing the method, a list of outliers, along with the type of functional outlier (i.e., magnitude or shape), is identified.

Functional outlier detection through the use of simultaneous prediction bands is also proposed. In a non-functional setting, the use of a prediction band has been considered for outlier detection (Horn et al., 1988). However, a prediction band is often computed with the use of an estimated mean and standard deviation: two statistics that are susceptible to outliers. Horn et al. (1988) show that the presence of outliers produces a prediction interval that is too wide, compared to when no outliers are present. A second challenge of prediction intervals is the assumed parent distribution. If the assumed distribution is specified incorrectly, the resulting prediction interval will also be incorrect, either in terms of accuracy or precision.

In this paper, a simultaneous prediction band is created with the use of a critical value function created by Liebl and Reimherr (2023). Liebl and Reimherr (2023) leverage Kac-Rice formulas from Random Field Theory in order to estimate a critical value function used to compute the margin of error. The estimated critical value function can then be used to create simultaneous prediction bands in a parametric form, through a pivot, as is typically seen in non-functional data. The Liebl and Reimherr (2023) critical value function allows for creation of simultaneous bands on data with wide-tailed errors, which still hold nominal coverage levels.

The rest of the chapter is organized as follows: the two new methods, Practical Outlier Detection (POD) and Prediction Band Outlier Detection (PBOD), are presented in Section 2.2.1 and 2.2.2, respectively. The simulation studies comparing POD and PBOD to TVD, MS-Plot, and MUOD (Huang and Sun, 2019; Dai and Genton, 2019; Azcorra et al., 2018), are presented in Section 2.3. An overall description of the simulation studies that were performed is given in Section 2.3.1. The first simulation study, which compares overall performance of POD, PBOD, TVD, MS-Plot, and MUOD, is described in Section 2.3.2. The second simulation study, which compares how well POD, TVD, and MUOD classify the type of functional outlier, magnitude or shape, is described in Section 2.3.3. A case study displaying each method's results for identifying functional outliers from the World Population Growth data is presented in Section 2.4. In Section 2.5, a summary of the paper, limitations of the methods proposed, potential future work, and the conclusion are presented.

2.2 Methods

2.2.1 Practical Outlier Detection (POD)

Practical Outlier Detection (POD) is proposed as a univariate, functional outlier detection method. POD is built on fundamental summary statistics, leveraging non-functional practices for functional data. POD is intended to be easily approachable, with easily interpretable results. POD also has the advantage of making no prior assumptions about the structure of the functional data sample. Let $X_i = \{X_i(t); t \in [0,1]\}$, $i = 1, \ldots, n$, be the observed, functional data at time points $t_j \in [0,1]$ for $j = 1, 2, \ldots, T$, where restricting the domain to $[0,1]$ is without loss of generality and common in FDA to ease notation.

2.2.1.1 Statistics Used for POD

The practical approach to functional outlier detection is carried out by estimating summary statistics of each individual curve, including the minimum, maximum, mean, median, range, overall roughness, area under the curve (AUC), variance, and coefficient of variation. The minimum, maximum, mean, and median are measures of location, which can be helpful in identifying all types of functional outliers, especially magnitude outliers. All four measures of location will be extreme for a magnitude outlier, when compared to these same four measures of non-outlying functional observations. A summary statistic of an observation is deemed extreme if the estimated value is an outlier in the sample of estimated summary statistics. For example, if a functional observation is shifted entirely above the rest of the observations in the sample (see Model 1 in Figure 2.1), its maximum, minimum, mean, and median will all be larger than those values of the other observations. If the magnitude outlier is only outlying over a small interval (see Model 2 in Figure 2.1), then the median will not be helpful in identifying the magnitude outlier.

The AUC (or integral of a univariate function) is estimated with spline interpolation. `R`'s `splinefun` function in the `stats` package is used to estimate a natural spline, while the `integrate` function is used to estimate AUC (R Core Team, 2021). Since AUC is altered by the position of the curve relative to the y-axis, it can also help identify magnitude outliers. For

example, if one curve is shifted well below the rest of the curves, its AUC will be smaller than all other curves' AUC values in the data (Model 1 in Figure 2.1). Contrarily, if a functional observation is only a shape or amplitude outlier, then a value of AUC would not be expected to be greatly different from the rest (Model 9 in Figure 2.1).

The range, variance, and coefficient of variation are measures of dispersion, which can be helpful in identifying functional observations that display shape outlyingness. For example, consider a random sample of functional observations, which are generated as a constant function over time, plus some noise. If there is a functional observation with sinusoidal shape, plus noise, plotted alongside the random sample (Model 7 in Figure 2.1), measures of location for the sinusoidal observation would not be too different from the measures of location or the AUC of other observations. The measures of dispersion can help uncover the different shape of the sinusoidal observation, given that the sinusoidal observation will have more inherent variation than the constant observations plus noise. The coefficient of variation (CV) helps further identify shape outlyingness, by scaling the variance with the value of the mean. Thus, if the observation is a magnitude and shape outlier, measures of location and CV should be different from the rest (Model 3 in Figure 2.1).

The roughness is calculated as an empirical estimate of the integrated, squared, second derivative, which is often used when smoothing functional data (Ramsay and Silverman, 2005). Roughness can also be described as the curvature of a functional observation, because it estimates the amount by which a function's shape deviates from a straight line. The larger the value of roughness, the more different the function is from a straight line. The roughness is directly relatable to identifying shape outliers. For example, consider the previous comparison of constant functions versus sinusoidal functions (Model 7 in Figure 2.1). All observations following a sinusoidal function should have a larger value of roughness than the observations following a constant function. If $x_i(t_j)$ represents the value of functional observation X_i at sampling point t_j, for $i = 1, \ldots, n$

observations and $j = 1, \ldots, T$ sampling points, then the roughness of X_i is

$$\text{Rough}_i = \frac{1}{4} \sum_{j=1}^{T} \left(x_i(t_j) - 2 \cdot x_i(t_{j-1}) + x_i(t_{j-2}) \right)^2 \tag{2.1}$$

(Ramsay and Silverman, 2005).

2.2.1.2 Splitting the Sampling Domain into Intervals

The high resolution of functional data cannot be fully captured by these summary statistics, when considering the entire functional domain. For instance, a single functional observation could have a high value of variance, because the functional observation shows high variability across the entire domain, or because the functional observation shows extraordinary local variability. This is important for functional outlier detection, because the outlying behavior of a single function does not always occur across the entire functional domain. However, if each functional observation is considered over a smaller interval of the domain, local changes in behavior can be identified more easily. On the contrary, if an interval of the domain is too small, outlying behavior can be missed or over-exaggerated, resulting in mis-classification of outliers. For example, if variation is calculated over only a few sampling points on the function, the local variance could be underestimated and identification of an outlier could be missed. Therefore, it is beneficial to look at sampling domain intervals of different sizes. Interval size is defined as the percentage of sampling points included within each interval. An interval size of 10% would include 10% of the sampling points within each interval, producing 10 intervals in total.

For the purpose of POD development, two interval sizes are used when implemented and the number of sampling points, T, is used to determine the two interval sizes. The first interval size is used to create A unique intervals, and the second interval size is used to create B unique intervals.

The interval sizes are

$$\begin{cases} 6.67\% \ (A = 15) \ \text{ and } \ 5\% \ (B = 20), & \text{if } \ T \geq 60, \\ 33\% \ (A = 3) \ \text{ and } \ 6.67\% \ (B = 15), & \text{if } \ 45 \leq T < 60, \\ 33\% \ (A = 3) \ \text{ and } \ 12.5\% \ (B = 8), & \text{if } \ 24 \leq T < 45, \end{cases} \tag{2.2}$$

and POD can not be implemented if $T < 24$. Note, the interval sizes in Equation 2.2 were decided by optimizing a previous simulation study (see Appendix A.2.1.1).

As an example, suppose $T = 60$, so interval sizes of 6.67% and 5% are used to break up the sampling domain. With an interval size of 6.67%, each functional observation is binned into $A = 15$ intervals across the sampling domain. These intervals are denoted as Int_l, for $l \in a = \{1, \ldots, A\}$ intervals. For each observation, $X_i(t)$, the observed points, $x_i(t_1), x_i(t_2), x_i(t_3)$, and $x_i(t_4)$ are all within $\text{Int}_{a=1}$. The observed points $x_i(t_5), x_i(t_6), x_i(t_7)$, and $x_i(t_8)$ are all within $\text{Int}_{a=2}$. Placing each observed point in a bin continues sequentially until all observed points are placed within one of the 15 intervals created. Binning the functional observations is repeated for interval size 5%, resulting in $B = 20$ intervals denoted as Int_l, for $l \in b = \{1, \ldots, B\}$. After splitting the data according to both interval sizes, there is a total of $L = A + B = 15 + 20 = 35$ intervals.

2.2.1.3 Calculating Summary Statistics on Each Interval

The minimum, maximum, mean, median, range, overall roughness, area under the curve, variance, and coefficient of variation are estimated for each individual observation, with respect to each interval. Each summary statistic calculated is denoted as S_{type}, for type $\in \{$min, max, mean, median, range, roughness, AUC, variance, and coefficient of variation$\}$ ($\{\cdot\}$ is used as set or collection notation throughout). The summary statistic calculated for a specific observation over a unique interval is denoted as $\text{S}_{type}\left(\text{Int}_l\right)_i$ for all Int_l, $l \in a, b$, for $i = 1, \ldots, n$. The collection, or set, of statistics calculated for all observations of a given type on interval Int_l is denoted as $\{\text{S}_{type}\left(\text{Int}_l\right)\}$. Thus, each functional observation has nine summary statistics estimated over each

of the domain intervals. For example, when $T = 60$, with interval size of 6.67% ($A = 15$), an $n \times 15 \times 9$ data frame of summary statistics is generated, and with an interval size of 5% ($B = 20$), a $n \times 20 \times 9$ data frame of summary statistics is generated.

2.2.1.4 Identifying Extreme Summary Statistics of Each Observation on Each Interval

The next step is to identify and count the number of extreme summary statistics for each observation (see Section 2.1.2). When comparing the summary statistics calculated for each functional observation on each interval, a functional outlier should have more extreme summary statistics than non-outlying functional observations. The total count of extreme summary statistics is a measure of the functional observation's overall outlyingness.

Several options were considered for determining the extreme values of each summary statistic, for each interval, including Tukey's classical boxplot rule ($1.5 \times IQR$), adjusted boxplot (Hubert and Vandervieren, 2008), and a combination of the two. The use of an adjusted boxplot was motivated by the shape of the sampling distribution for a few of the summary statistics (e.g., the range, variance, and roughness are typically positively skewed). Tukey's classical boxplot rule is not robust to skewed distributions (Seo, 2002). In comparison, Hubert and Vandervieren (2008) proposed the adjusted boxplot, which incorporates a robust measure of skewness called the medcouple (Brys et al., 2004). If the distribution is symmetric, the medcouple is zero, and the adjusted boxplot becomes equivalent to Tukey's classical boxplot. Despite the concern of skewness, the use of Tukey's classical boxplot was found to have better performance (in terms of ACC, PPV, and Matthew's correlation coefficient (MCC)) than use of an adjusted boxplot or combination (see Appendix A.2.1.2 for more details).

Tukey's classical boxplot rule is implemented by calculating the first quartile, third quartile, and interquartile range of the computed summary statistics, relative to each statistic and each interval. The first quartile, third quartile, and interquartile range of a statistic in $type$ and interval l are denoted by

$$Q_1\left\{S_{type}\left(\text{Int}_l\right)\right\}, Q_3\left\{S_{type}\left(\text{Int}_l\right)\right\}, \text{ and } IQR\left\{S_{type}\left(\text{Int}_l\right)\right\}, \tag{2.3}$$

respectively. A lower and upper "fence" are computed by

$$\begin{aligned} L\left\{S_{type}\left(\text{Int}_l\right)\right\} =& Q_1\left\{S_{type}\left(\text{Int}_l\right)\right\} - 1.5 \cdot IQR\left\{S_{type}\left(\text{Int}_l\right)\right\} \text{ and} \\ U\left\{S_{type}\left(\text{Int}_l\right)\right\} =& Q_3\left\{S_{type}\left(\text{Int}_l\right)\right\} + 1.5 \cdot IQR\left\{S_{type}\left(\text{Int}_l\right)\right\}, \end{aligned}$$

for summary statistics calculated on interval l.

For example, if $T \geq 60$ and interval sizes 6.67% ($A = 15$) and 5% ($B = 20$) are used, the result of calculating the fences for each summary statistic and interval is a set of 315 unique fences (9 statistics by $L = A + B = 35$ different intervals). The estimated statistics for each functional observation, with respect to an interval and statistic, are compared to their respective fences. If

$$\left[S_{type}\left(\text{Int}_l\right)\right]_i \notin \left(L, U\right)\left\{S_{type}\left(\text{Int}_l\right)\right\}, \tag{2.4}$$

then $\left[S_{type}\left(\text{Int}_l\right)\right]_i$ is declared an extreme summary statistic. If a summary statistic is declared extreme, the statistic is flagged as "out". Otherwise, if a summary statistic is not declared extreme, the statistic is flagged as "in". After all statistics over all intervals are assessed, the total number of summary statistics flagged as "out" are counted for each functional observation:

$$n_{\text{out},i} = \sum_{l} \sum_{\text{type}} \mathbb{I}\left\{\left[S_{type}\left(\text{Int}_l\right)\right]_i \notin \left(L, U\right)\left\{S_{type}\left(\text{Int}_l\right)\right\}\right\} \leq 9 \cdot l. \tag{2.5}$$

Additionally, the total number of summary statistics indicated as "out" with respect to each statistic is counted for each functional observation:

$$n_{\text{out,type},i} = \sum_{l} \mathbb{I}\left\{\left[S_{type}\left(\text{Int}_l\right)\right]_i \notin \left(L, U\right)\left\{S_{type}\left(\text{Int}_l\right)\right\}\right\} \leq l. \tag{2.6}$$

The counts computed in Equations 2.5 and 2.6 are saved in a data frame. The data frame includes a column of indices for the observations, a column of $n_{\text{out},i}$ values, and separate columns

of $n_{\text{out,type},i}$ for each $type$. The collection of $n_{\text{out},i}$ values is denoted as n_{out}, and the collection of $n_{\text{out,type},i}$ values is denoted as $n_{\text{out,type}}$ for each type.

2.2.1.5 Identifying Observations as Functional Outliers

A list of functional outliers is identified by comparing the total count of extreme summary statistics for each observation ($n_{out,i}$) to a threshold. The user chooses between two types of thresholds:

1. Custom threshold (user): the user specifies the expected proportion of the observations that may be outliers as a proportion, $\delta \in [0, 1]$. For example, if $\delta = 0.05$ is specified, approximately 5% of the observations will be flagged as an outlier. Any functional observation with more extreme statistics than the $(1 - \delta) \times 100^{th}$ percentile is identified as an outlier. In other words, if

$$n_{\text{out},i} \geq (1 - \delta) \cdot 100^{th} \left\{n_{\text{out}}\right\} \text{ percentile}, \tag{2.7}$$

   then observation i is a functional outlier.

2. Tukey's classical boxplot threshold (Tuk): any observation with a total count of extreme summary statistics ($n_{out,i}$) higher than the upper fence of Tukey's classical boxplot rule is identified as an outlier. The threshold is applied by calculating the third quartile, interquartile range, and upper fence ($Q_3 + 1.5 \cdot IQR$) of the total counts of extreme summary statistics ($n_{out,i}$). A functional observation with few to no extreme summary statistics would not be considered a functional outlier. Thus, the calculation of the lower fence is not necessary. Any observation with a total count of extreme summary statistics larger than the upper fence is identified as an outlier; i.e. if

$$n_{\text{out},i} \geq Q_3 \left\{n_{\text{out}}\right\} + 1.5 \cdot IQR \left\{n_{\text{out}}\right\}, \tag{2.8}$$

   then observation i is identified as a functional outlier.

2.2.1.6 Classifying the Type of Functional Outlier

As previously discussed in Section 2.1.2, functional outliers are often classified further as magnitude (or shift) outliers, shape outliers, and/or amplitude outliers (for e.g., Sequential Transformation (Dai et al., 2020)). POD was developed to classify observations as magnitude (shift) outliers and/or shape (and amplitude) outliers. An extreme value for the minimum, maximum, median, mean, or AUC tends to be associated with a magnitude outlier, herein called the "location statistics." While, an extreme value for the variance, CV, range, or roughness tends to be associated with a shape or amplitude outlier, herein called the "spread statistics." POD can be used to identify the type of outlier, by counting extreme summary statistics in the location group to the extreme summary statistics in the spread group, specific to intervals a and intervals b (see Section 2.2.1.2). The use of different interval sizes reveals different levels of magnitude and/or shape outlyingness. Thus, it is possible for a functional outlier to be classified as both a magnitude and shape outlier using POD, as is common in functional outlier literature (Ojo et al., 2021).

The number of extreme statistics by type and interval, for each observation, is given by

$$n_{\text{out,type},a,i} = \sum_{l \in a} \mathbb{I}\left\{ \left[S_{type}\left(\text{Int}_l\right) \right]_i \notin \left(L, U\right) \left\{ S_{type}\left(\text{Int}_l\right) \right\} \right\}, \tag{2.9}$$

for intervals in group a, and similarly for intervals in group b. The count of extreme summary statistics in the location statistics is given by

$$n_{\text{out,location},a,i} = \sum_{l \in a} \sum_{\text{type} \in \text{location}} \mathbb{I}\left\{ \left[S_{type}\left(\text{Int}_l\right) \right]_i \notin \left(L, U\right) \left\{ S_{type}\left(\text{Int}_l\right) \right\} \right\}, \tag{2.10}$$

for intervals in group a, and similarly for intervals in group b. The count of extreme summary statistics in the spread statistics is given by

$$n_{\text{out,spread},a,i} = \sum_{l \in a} \sum_{\text{type} \in \text{spread}} \mathbb{I}\left\{ \left[S_{type}\left(\text{Int}_l\right) \right]_i \notin \left(L, U\right) \left\{ S_{type}\left(\text{Int}_l\right) \right\} \right\}, \tag{2.11}$$

for intervals in group a, and similarly for intervals in group b.

For any interval, a (or b), if

$$n_{\text{out,location},a,i} > 2, \tag{2.12}$$

then interval a (or b) is said to be magnitude outlying. Similarly, for any interval, a (or b), if

$$n_{\text{out,spread},a,i} > 1, \tag{2.13}$$

then interval a or b is said to be shape outlying. If one third or more of all intervals, $L = A + B$, are magnitude outlying, then the observation is classified as a magnitude outlier. If one fifth or more of all intervals are shape outlying, then the observation is classified as a shape outlier. If both events occur, then the observation is classified as a magnitude and shape outlier. If neither event occurs, but the observation was identified as an outlier, then the observation is further classified as a shape outlier (see Appendix A.2.1.3 for additional details on how the thresholds were determined).

2.2.2 Prediction Band Outlier Detection (PBOD)

Prediction bands have been used in the past for non-functional outlier detection, with varying levels of success (Horn et al., 1988). A first challenge in using prediction bands for functional outlier detection, is the need to create a parametric, simultaneous band for functional data. Preferably, the prediction band should not rely on any resampling methods, because resampling can greatly slow down computation time and the random sample is assumed to be representative of the population. A parametric prediction band also assumes the random sample is representative of the population. Therefore, if the random sample is not representative of the population, a wide array of bias and variability issues could occur. The outlier detection method, which relies on a simultaneous prediction band, should also be created to avoid common issues of outlier masking (see Section 2.1.1). Development of the simultaneous prediction band and the outlier detection method that relies on this band are detailed in the next two subsections.

2.2.2.1 Creation of Fast and Fair Simultaneous Prediction Bands

Liebl and Reimherr (2023) use a connection between the supremum of a functional observation and the expected Euler characteristic of an exceedance set to create a critical value function. Liebl and Reimherr (2023) first consider a real-valued random function over a closed interval, $X = \{X(t) : t \in [0,1]\}$, where the data domain restricted on $[0,1]$ is without loss of generality. The probability that function X crosses some threshold function $u = \{u(t) : t \in [0,1]\}$ across the domain, at least once, is given by $P(\exists\, t \in [0,1] : X(t) \geq u(t))$. By comparison, an expected Euler characteristic can be calculated by counting the number of events when X crosses u in an upward trajectory ("up-crossings"), which is defined as

$$N_{u,X}([0,1]) := \#\{0 \leq t \leq 1 : X(t) = u(t), X'(t) > u'(t)\}, \tag{2.14}$$

where X' and u' are the first derivative of X and u, respectively. Similar to an up-crossing, a down-crossing is when X crosses u in a downward trajectory.

Liebl and Reimherr (2023) then use Boole's and Markov's inequality to establish the inequality

$$P(\exists\, t \in [0,1] : X(t) \geq u(t)) \leq E[\phi_u(X)], \tag{2.15}$$

where $\phi_u(X) := \mathbb{1}_{X(0)>u(0)} + N_{u,X}([0,1])$ denotes the Euler characteristic of the excursion set $\{t \in [0,1] : X(t) > u(t)\}$. Furthermore, if the starting value of t is not 0, but some value $t_0 \in [0,1]$, then a more general form of the expected Euler characteristic inequality can be implemented, which leverages the symmetry of calculating up-crossings vs down-crossings. The general form of the expected Euler characteristic inequality is given by

$$\begin{aligned} P[\exists\, t \in [0,1] :& X(t) \geq u(t)] \leq \\ & P[X(t_0) \geq u(t_0)] + E[N^{-}_{u,X}([0,t_0])] + E[N_{u,X}([t_0,1])], \end{aligned}$$

where $N^{-}_{u,X}$ is the number of down-crossings:

$$N^{-}_{u,X}\left([0,1]\right) := \#\left\{0 \leq t \leq 1 : X(t) = u(t), X'(t) < u'(t)\right\}. \tag{2.16}$$

Finally, Liebl and Reimherr (2023) use the Kac (1943) and Rice (1945) (Kac-Rice) functions to explicitly calculate the expected Euler characteristic for known distributions. Liebl and Reimherr (2023) also generalize the Kac-Rice formulas by allowing them to vary across the time domain, t. The final outcome is a critical value (threshold) function $u_{\alpha/2}(t)$, which results in a valid $(1-\alpha)\times 100\%$ simultaneous confidence band when used in conjunction with a proper estimator and standard error of the estimator. These results hold for any random sample of elliptical processes, including Gaussian and Student's t random samples. For instance, if $X_1, X_2, \ldots, X_n$ is a random sample of functional observations $\in \mathcal{C}^1[0,1]$, identically distributed as a Gaussian process with unknown mean function $\theta(t) = E\left[X_i\right]$ and unknown covariance function $C_\theta(t,s) = Cov\left(X_i(t), X_i(s)\right)$, then critical value function $u_{\alpha/2}(t)$ can be estimated and a simultaneous confidence band can be formed. Liebl and Reimherr (2023) calculate $u^{\star}_{\alpha/2}(t)$, an estimate for $u_{\alpha/2}(t)$, using an algorithm that leverages the Kac-Rice formulas (Algorithm 1). The 95% simultaneous confidence band for $\theta(t)$ is given by

$$n^{-1}\sum_{i=1}^{n} X_i(t) \pm u^{\star}_{.025}(t)\sqrt{n^{-1}\text{Var}\left(X_i\right)}. \tag{2.17}$$

In this paper, a simultaneous prediction band is created using the critical value function of Liebl and Reimherr (2023) and known pivotal functions. If X_{n+1} is a new, unobserved functional observation from a random sample of functional observations $X_1, X_2, \ldots, X_n \in \mathcal{C}^1[0,1]$, is identically distributed as a Gaussian process with unknown mean function $\theta(t) = E\left[X_i\right]$ and unknown covariance function $C_\theta(t,s) = Cov\left(X_i(t), X_i(s)\right)$, and $W(t) = X_{n+1}(t) - n^{-1}\sum_{i=1}^{n} X_i(t)$, then,

$$E\left[W(t)\right] \overset{ind}{=} E\left[X_{n+1}\right] - n^{-1}\sum_{i=1}^{n} E\left[X_i\right] = 0, \tag{2.18}$$

and

$$\begin{aligned} Var\left[W(t)\right] &= Var\left[X_{n+1}\right] + n^{-2}\sum_{i=1}^{n} Var\left[X_i\right] \\ &= C_\theta + n^{-2}\sum_{i=1}^{n} C_\theta \\ &= \left(1 + n^{-1}\right) C_\theta. \end{aligned} \tag{2.19}$$

Thus, $W(t)$ is distributed as a Gaussian process with zero mean function and variance function $(1 + n^{-1})\, C_\theta$. For each $t \in [0, 1]$, pivotal quantity T can be defined as

$$T(t) = \frac{W(t)}{\sqrt{Var\left[W(t)\right]}} = \frac{X_{n+1} - n^{-1}\sum_{i=1}^{n} X_i}{\sqrt{\left(1 + n^{-1}\right) C_\theta}}, \tag{2.20}$$

where $T(t)$ is a standard Gaussian process. An estimator for C_θ can be defined as:

$$\hat{C}_\theta := (n-1)^{-1}\sum_{i=1}^{n}\left(X_i(t) - n^{-1}\sum_{i=1}^{n} X_i(t)\right)\left(X_i(s) - n^{-1}\sum_{i=1}^{n} X_i(s)\right), \tag{2.21}$$

and an estimator of $T(t)$ is given by

$$\hat{T}(t) = \frac{X_{n+1} - n^{-1}\sum_{i=1}^{n} X_i}{\sqrt{\left(1 + n^{-1}\right)\hat{C}_\theta}}. \tag{2.22}$$

$\hat{T}(t)$ is distributed as a Student's t stochastic process with $n - 1$ degrees of freedom.

If the pivotal quantity T is inverted by solving for new observation X_{n+1}, a simultaneous prediction band for a new, unobserved functional observation in the population, can be constructed by leveraging the critical value threshold function created by Liebl and Reimherr (2023). Let $u^\star_{t_{\alpha/2,n-1}}$ denote the critical value threshold function, calculated using the $(1 - \alpha/2)^{th}$ percentile of the Student's t distribution with $n-1$ degrees of freedom. Then, a $(1-\alpha)\times 100\%$ simultaneous prediction band (i.e., Fast and Fair Simultaneous Prediction Band, or FFSPB) for a new, unobserved function

in the population is given by

$$n^{-1}\sum_{i=1}^{n} X_i \pm u^{\star}_{t_{\alpha/2,n-1}}\sqrt{(1+n^{-1})\,\hat{C}_{\theta}}. \tag{2.23}$$

A simulation study was conducted to show the simultaneous prediction band meets nominal coverage levels (see Appendix A.3.1). Three different means combined with three different covariance structures for data generation were considered in the simulation. The simultaneous prediction band meets nominal coverage levels for small ($n = 15$) and larger ($n = 100$) sample sizes, when assuming Student's t distributed errors. However, if the errors are assumed to be Gaussian, the simultaneous band no longer meets nominal coverage levels for small sample sizes ($n = 15$). This dependence of sample size is similar to that of confidence/prediction bands for non-functional data.

As reported in Liebl and Reimherr (2023), it is also possible to create one-sided simultaneous confidence/prediction bands by setting the generalized, expected Euler characteristic function equal to α, rather than $\alpha/2$. A one-sided simultaneous band can be useful in the case that the functional data is strictly positive and within a close range of zero, for example.

2.2.2.2 Functional Outlier Detection Through Fast and Fair Simultaneous Prediction Bands

Using the FFSPB, a functional outlier detection method is proposed, called Prediction Band Outlier Detection (PBOD). A prediction band can be altered by the presence of an outlier in the random sample. The outlier may widen the prediction band, potentially masking the presence of other outliers in the sample (Horn et al., 1988). To avoid issues of masking with a simultaneous prediction band, a two-step resampling procedure is proposed.

Given a random sample of n functional observations, $\{X_1, \ldots, X_n\}$, the two-step resampling procedure is as follows:

1. A training set and testing set are created from the original random sample by randomly placing half of the observations in each set. The training and testing sets are denoted as

$$\left\{X_i^{\text{train}}\right\}_{i=1}^{n/2} \text{ and } \left\{X_{i'}^{\text{test}}\right\}_{i'=1}^{n/2}, \text{ where } i \neq i'.$$

2. Using the training set, a FFSPB is generated:

$$FFSPB_{iter} = (n/2)^{-1} \sum_{i=1}^{n/2} X_i^{\text{train}} \pm u^{\star}_{t_{\alpha/2,n/2-1}} \sqrt{\left(1 + (n/2)^{-1}\right) \hat{C}_{\theta}}$$

3. Each observation in the testing set is compared to the FFSPB created in step 2. An exceedance is defined as an observation that is not within the FFSPB and is calculated for each sampling point, t_j, of each functional observation, i, as

$$\text{exc}_{i,j} := \mathbb{I}\left\{X_{i'}^{\text{test}}(t_j) \notin FFSPB_{iter}\right\},$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. The total number of exceedances found for each functional observation in the testing set for a single iteration is saved as

$$\text{exc}_{iter,i'} := \sum_{j=1}^{T} \text{exc}_{i',j}.$$

4. Steps 1 through 3 are repeated a total of n times, and the number of exceedances, $\text{exc}_{iter,i}$, are recorded for each functional observation in the testing set, on each iteration.

5. For each functional observation, a resampling weight is computed, based on the exceedances recorded from each iteration:

$$w_i := 1 - (nT)^{-1} \sum_{iter=1}^{n} \text{exc}_{iter,i}.$$

6. Steps 1 through 3 are repeated a second time for $2n$ iterations, resampling with respect to weights w_i such that the probability of observation i being randomly chosen for the training set is given by w_i.

7. The average number of exceedances ($\overline{\text{exc}}$) are calculated and recorded for each observation:

$$\overline{\text{exc}}_i := (2n)^{-1} \sum_{iter=1}^{2n} \text{exc}_{iter,i}. \tag{2.24}$$

8. A list of outlying observations is identified according to a threshold specified by the user, $\gamma \in [0, 1]$. Any functional observation with $\overline{\text{exc}}_i \geq (1 - \gamma) \times 100^{th}$ percentile of $\overline{\text{exc}}$ is identified as a functional outlier.

2.3 Simulations

2.3.1 Simulation Description

Two simulations were used to compare POD and PBOD to MS-Plot (Dai and Genton, 2019), TVD (Huang and Sun, 2019), and MUOD (Azcorra et al., 2018). POD has two thresholds that could be considered: user-specified threshold ("user") and Tukey's classical boxplot ("Tuk") (see Section 2.2.1). Both thresholds ("user" and "Tuk") were used for comparison in the simulation studies. Similarly, MUOD has two thresholds that could be implemented: boxplot rule ("box") and tangent rule ("tan"). Both thresholds for MUOD were used for comparison in the simulation studies. Each simulation considers several different simulation parameters: sample sizes $n = 30, 100,$ and 250, sampling points $T = 30, 45, 60, 100, 365,$ and 1000, and outlier proportions $\Delta = 0.05$ and $\Delta = 0.15$. Ojo et al. (2021) use a covariance kernel for data generation expressed as

$$\gamma(s, t) = \alpha \exp\{-\beta|t - s|^{\nu}\}, \tag{2.25}$$

with default values of $\alpha = \beta = \nu = 1$. The α parameter allows for constant scaling of the covariance, whereas the β parameter allows for exponential scaling based on the distance between

two sampling points. Different levels of covariance roughness measured by $\alpha = 1, 2,$ and 3 or $\beta = 0.1, 0.5,$ and 0.9 (higher values indicate more roughness) were used to generate various samples from each of the nine data models in Ojo et al. (2021).

In the Method Performance Simulation (Section 2.3.2), α is used to adjust the covariance roughness and β is fixed at its default value ($\beta = 1$). In the Classification of Outlier Type Simulation (Section 2.3.3), β is used to adjust the covariance roughness and α is fixed at its default value ($\alpha = 1$). β was used in the second simulation because it allows for easier separation of different outlier types. If only α were adjusted, different simulation models have a tendency of producing more than one type of functional outlier. Clear separation of outlier type in the simulated data is necessary for examining the outlier type classification.

In each simulation, the methods were compared using sensitivity, specificity, ACC, PPV, and MCC, as defined in Section 2.1.2. Furthermore, POD (user) was implemented by setting $\delta = \Delta$ for all simulations, assessing the oracle property of POD functional outlier detection.

2.3.2 Method Performance Simulation

The Method Performance Simulation is used to provide an overall comparison of method performance among POD, PBOD, MS-Plot (Dai and Genton, 2019), MUOD (Azcorra et al., 2018), and TVD (Huang and Sun, 2019). Sample sizes $n = 30, 100,$ and 250, sampling points $T = 30, 45, 60, 100, 365,$ and 1000, outlier proportions $\Delta = 0.05$ and $\Delta = 0.15$, and $\alpha = 1, 2,$ and 3 (Equation 2.25) were used to generate functional data for each of the nine data models in Ojo et al. (2021). When the simulation parameters were combined, a total of 972 unique simulations were considered, or 108 simulations per each of the nine data models. Each individual simulation was implemented with 1000 iterations (or for 24 hours, pending computation time). No fewer than 336 iterations were implemented in any of the simulations. In total, each of the nine data models was simulated for at least 104,056 iterations. The overall average values and standard deviations were calculated by averaging over all simulation parameters: model, n, T, Δ, and α.

The overall averages and standard deviations were presented in Table 2.2. Methods PBOD and MUOD (tan) were outperformed by all other functional outlier detection methods. When using either PBOD or MUOD (tan), there is a tendency of classifying a high proportion of false positives, which is reflected by lower average specificity (PBOD = 0.302 and MUOD (tan) = 0.681) than other methods (range between 0.932 and 0.988) and higher average sensitivity (PBOD = 0.923 and MUOD (tan) = 0.794) than other methods (range between 0.644 and 0.751; Table 2.2). On several iterations, PBOD classified every observation in the sample as an outlier, which results in $TN = FN = 0$ and thus MCC is undefined. For all but Simulation Model 2, PBOD had an undefined MCC in 61,596 iterations or more. Therefore, the MCC was not reported for PBOD. By comparison, all other methods had a defined MCC for at least 95% (or at least 99,887) of the $104,056$ iterations for each data model.

Table 2.2: Average (standard deviation) results of the simulation, averaged over the nine data generation models, sample size ($n = 30, 100, 250$), number of sampling points ($T = 30, 45, 60, 100, 365, 1000$), covariance roughness ($\alpha = 1, 2, 3$), and outlier proportion ($\Delta = 0.05, 0.15$).

Method	Run Times (s)	Sensitivity	Specificity	ACC	PPV	MCC
POD (Tuk)	0.887 (0.616)	0.703 (0.351)	0.967 (0.029)	0.937 (0.050)	0.649 (0.273)	0.633 (0.296)
POD (user)	0.891 (0.635)	0.751 (0.283)	0.962 (0.042)	0.944 (0.061)	0.696 (0.278)	0.691 (0.306)
PBOD	19.413 (28.837)	0.923 (0.215)	0.302 (0.450)	0.371 (0.396)	0.297 (0.336)	–*
MS-Plot	0.060 (0.039)	0.703 (0.353)	0.981 (0.020)	0.952 (0.045)	0.744 (0.307)	0.682 (0.315)
MUOD (box)	0.029 (0.048)	0.684 (0.330)	0.932 (0.040)	0.904 (0.053)	0.501 (0.260)	0.521 (0.279)
MUOD (tan)	0.029 (0.048)	0.794 (0.282)	0.681 (0.243)	0.691 (0.210)	0.266 (0.174)	0.326 (0.217)
TVD	0.016 (0.028)	0.644 (0.427)	0.988 (0.024)	0.953 (0.057)	0.723 (0.408)	0.647 (0.418)

* Average MCC was excluded for PBOD because the MCC was NA or undefined for more than half of the iterations.

The performance of the methods was also examined across the different number of sampling points, T, and the sample size n. The average MCC was calculated by averaging over Δ, α, and all nine data generation models (Figure 2.2). For a sample size of $n = 30$, POD (user) and POD (Tuk) had an average MCC of 0.671 and 0.639, which was greater than the average of all other methods. For larger sample sizes, $n \geq 100$, and at smaller number of sampling points, $T \leq 60$, TVD had the highest average MCC (average MCC range from 0.725 to 0.797; Figure 2.2); otherwise, when

$n \geq 100$ and $T \geq 100$, MS-Plot had the highest average MCC (average MCC range from 0.725 to 0.738; Figure 2.2). In nearly all settings, MUOD (box) had the lowest average MCC (range from 0.514 to 0.532; Figure 2.2). For all methods except TVD, average MCC was consistent across all number of sampling points, T. Average MCC decreased for TVD as the number of sampling points increased. There was a slight increase in average MCC for MS-Plot and TVD as the number of functional observations increased, while the MCC for other methods remained consistent across different sample sizes.

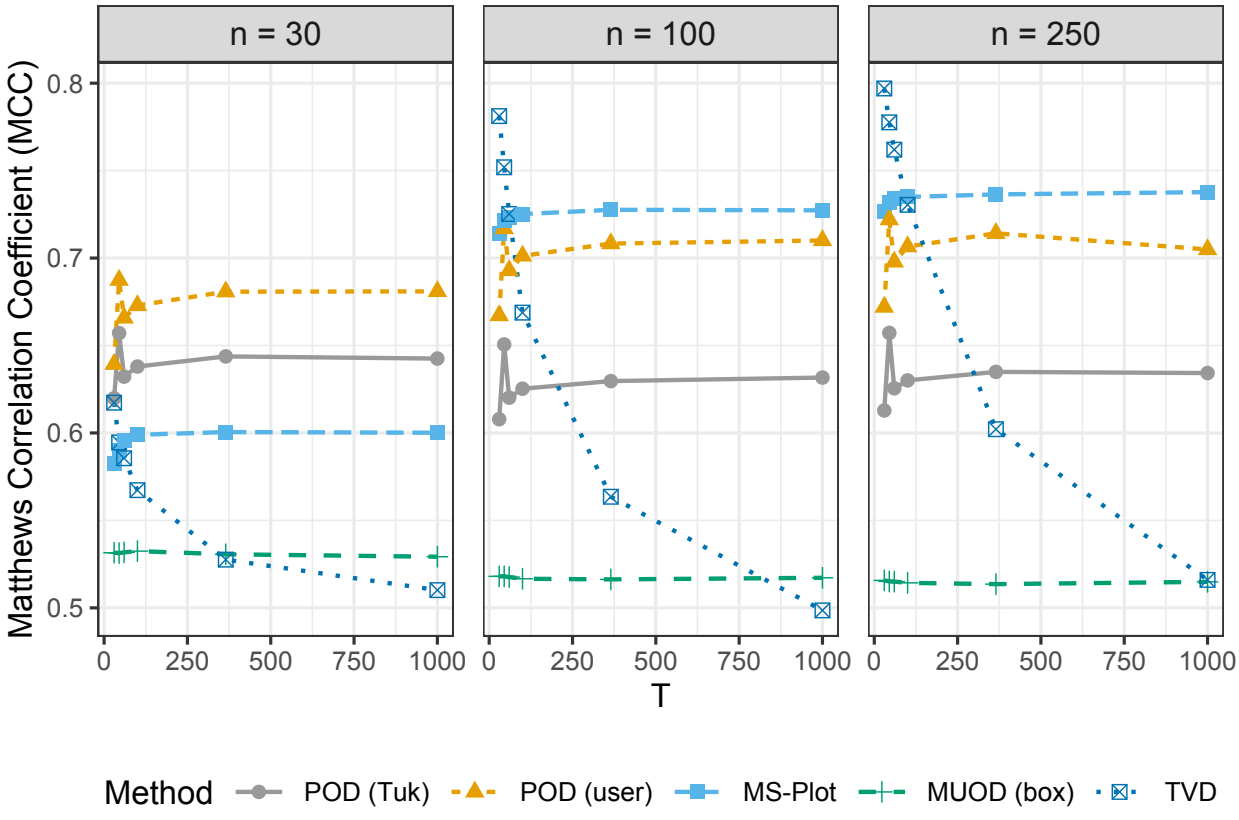


Figure 2.2: Average values of Matthew's Correlation Coefficient (MCC) for varying number of sampling points (T = 30, 45, 60, 100, 365, and 1000), faceted on the sample size (n = 30, 100, 250), and colored by the method (POD (Tuk), POD (user), MS-Plot, MUOD (box), and TVD). PBOD is not included due to missing and/or undefined MCC values. MUOD (tan) is not included, so as to focus on the top performing methods.

Trends with the number of sampling points, T, and the roughness parameter, α, were also compared among the methods. The average MCC was calculated by averaging over n, Δ, and

all nine data generation models (Figure 2.3). When the underlying roughness of the sample of functional data is $\alpha = 1$, MS-Plot had the highest average MCC across all number of sampling points (average MCC range from 0.834 to 0.840; Figure 2.3). When there was more roughness (e.g., $\alpha \geq 2$) and the number of sampling points is $T \leq 60$, TVD showed the highest average MCC value (range from 0.621 to 0.725; Figure 2.3). Otherwise, MS-Plot and POD (user) had the highest average values of MCC (0.684 and 0.680, respectively) for roughness $\alpha = 2$ (Figure 2.3). Both POD (user) and POD (Tuk) showed the highest average MCC (0.616 and 0.555, respectively) when the number of sampling points was large, $T > 100$, and the roughness was $\alpha = 3$ (Figure 2.3). Furthermore, all methods increased in average MCC when the proportion of outliers increased from 5% to 15%, indicating that outlier detection was more accurate when using each method, if more outliers were present.

2.3.3 Classification of Outlier Type Simulation

The second simulation is a comparison of a method's performance in classifying the type of functional outlier: magnitude (shift) or shape (and amplitude) outliers. Methods MS-Plot (Dai and Genton, 2019) and PBOD are not included in this simulation, because they were not developed to classify the type of functional outlier. Simulation model 1 from Ojo et al. (2021) was used to assess the classification of magnitude outliers. Simulation model 7 from Ojo et al. (2021) will be used to assess the classification of shape or amplitude outliers. A new simulation model was created by combining simulation models 1 and 7 from Ojo et al. (2021). This new simulation model was used to generate random functional data that contained only combined (shape and magnitude) outliers in a sample (called model "Combined"; Figure A.1 in Appendix A.1.1) and to generate random functional data that contained all types of functional outliers: shape, magnitude, and combined (called model "Mix"; Figure A.2 in Appendix A.1.1). There were two types of combined outliers generated by the new simulation model:

1. An outlier that shows shape outlyingness and magnitude outlyingness across the entire domain.

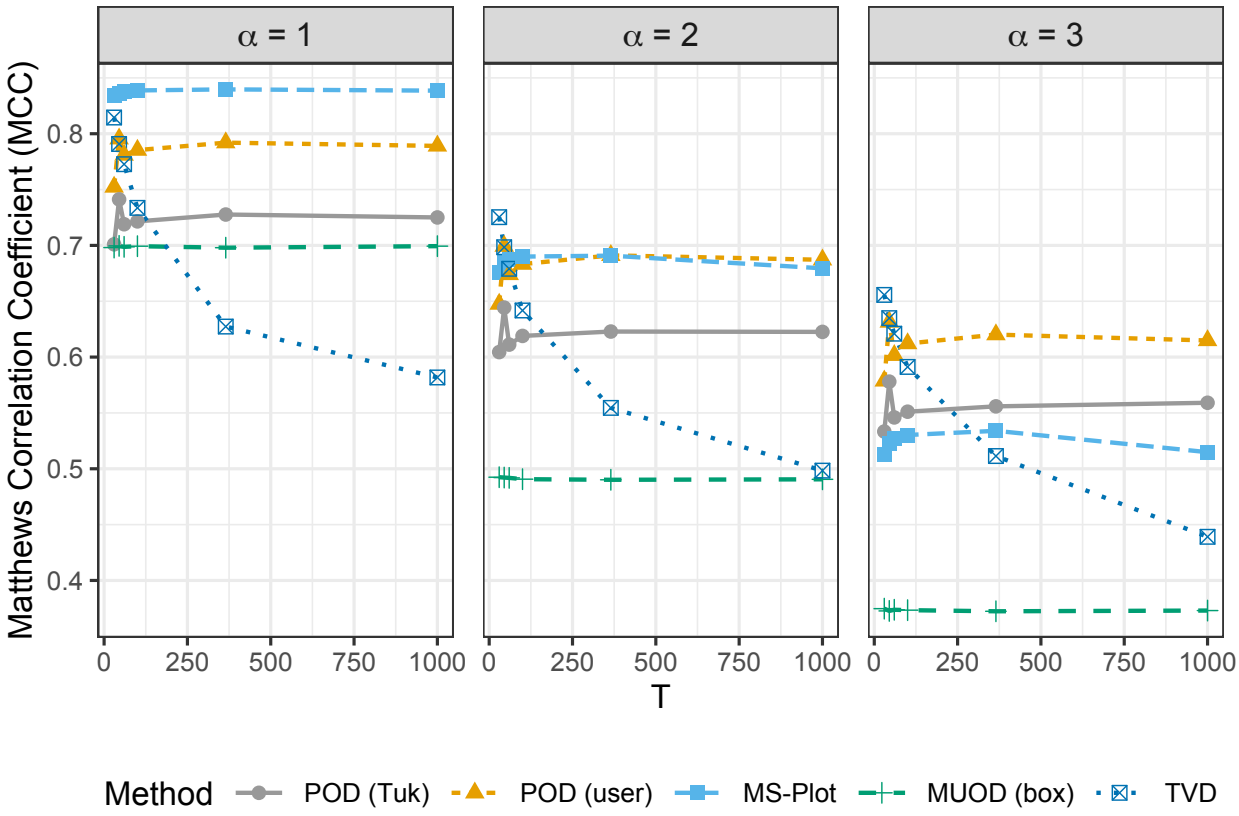


Figure 2.3: Average values of Matthew's Correlation Coefficient (MCC) for varying number of sampling points (T = 30, 45, 60, 100, 365, 1000), faceted on the roughness (covariance α = 1, 2, 3), and colored by the method (POD (Tuk), POD (user), MS-Plot, MUOD (box), TVD). PBOD is not included due to missing and/or undefined MCC values. MUOD (tan) is not included, so as to focus on the top performing methods.

2. An outlier that shows shape outlyingness on one half of the domain magnitude outlyingness on the other half of the domain.

Note, for purposes of this simulation, only persistent outliers were considered. Simulation parameters include $n = 30, 100,$ and 250, number of sampling points $T = 30, 45, 60, 100, 365,$ and 1000, outlier proportion $\Delta = 0.05$ and 0.15, and $\beta = 0.1, 0.5,$ and 0.9 (Equation 2.25). Each simulation was implemented with 1000 iterations.

In each simulation, the number of observations classified as magnitude outliers, and the number of observations classified as shape outliers, were recorded for each method. Then, the sensitivity, specificity, ACC, PPV, and MCC were calculated relative to each outlier type; i.e. the number

of outliers classified as shape and the number of true shape outliers were used to calculate the shape sensitivity, specificity, ACC, PPV, and MCC (similarly for magnitude outliers).

The overall average and standard deviation of each method were reported for both classes of outliers: magnitude and shape (Table 2.3). For each outlier type, the overall average was calculated by averaging over the four models used (i.e., models 1, 7, "Combined", and "Mixed"), sample size ($n = 30, 100,$ and 250), sampling points ($T = 30, 45, 60, 100, 365,$ and 1000), proportion of outliers ($\Delta = 0.05$ and 0.15), and covariance roughness ($\beta = 0.1, 0.5,$ and 0.9). All outlier detection methods had an average MCC of 0.685 or larger when classifying magnitude outliers. POD (user) had the highest average MCC at 0.848 (SD = 0.203) when classifying magnitude outliers (Table 2.3), followed by POD (Tuk) with an average MCC of 0.836 (SD = 0.200). For shape outliers, MUOD (box), MUOD (tan), and TVD had an MCC less than 0.700, while POD (Tuk) and POD (user) had an average MCC greater than 0.700 (Table 2.3).

Table 2.3: Average (standard deviation) results of the Magnitude and Shape outlier classification, averaged over the four simulation models used (i.e., models 1, 7, "Combined", and "Mixed"), sample size ($n = 30, 100,$ and 250), sampling points ($T = 30, 45, 60, 100, 365,$ and 1000), proportion of outliers ($\Delta = 0.05$ and 0.15), and covariance roughness ($\beta = 0.1, 0.5,$ and 0.9).

Magnitude Outliers

Method	Sensitivity	Specificity	ACC	PPV	MCC
POD (Tuk)	0.997 (0.054)	0.979 (0.029)	0.980 (0.030)	0.626 (0.384)	0.836 (0.200)
POD (user)	0.990 (0.074)	0.982 (0.028)	0.982 (0.029)	0.649 (0.392)	0.848 (0.203)
MUOD (box)	0.997 (0.055)	0.976 (0.028)	0.977 (0.029)	0.569 (0.403)	0.829 (0.201)
MUOD (tan)	0.996 (0.058)	0.928 (0.111)	0.930 (0.112)	0.521 (0.375)	0.785 (0.200)
TVD	0.747 (0.392)	0.982 (0.033)	0.963 (0.049)	0.535 (0.445)	0.685 (0.370)

Shape Outliers

Method	Sensitivity	Specificity	ACC	PPV	MCC
POD (Tuk)	0.936 (0.236)	0.967 (0.032)	0.967 (0.033)	0.502 (0.372)	0.714 (0.275)
POD (user)	0.862 (0.288)	0.981 (0.031)	0.978 (0.031)	0.654 (0.397)	0.749 (0.314)
MUOD (box)	0.924 (0.242)	0.936 (0.039)	0.936 (0.039)	0.390 (0.355)	0.645 (0.276)
MUOD (tan)	0.759 (0.335)	0.835 (0.201)	0.830 (0.192)	0.339 (0.356)	0.497 (0.324)
TVD	0.647 (0.430)	0.984 (0.021)	0.966 (0.040)	0.457 (0.465)	0.591 (0.414)

The performance of the methods was also examined across the different number of sampling points, T, and the sample size n. The average (standard deviation) MCC was computed for each of magnitude and shape outliers by averaging over the four models used (i.e., models 1, 7, "Combined", and "Mixed"), proportion of outliers ($\Delta = 0.05, 0.15$), and covariance roughness ($\beta = 0.1, 0.5,$ and 0.9; Figure 2.4). When classifying magnitude outliers, POD (user) had a greater average MCC at all values of sampling points T and all sample sizes n (average MCC ranged from 0.785 to 0.866; Figure 2.4). If $n = 30$ observations, then MUOD (box) had the second highest average MCC and POD (Tuk) had the third highest average MCC across all sampling points (MUOD (box) average MCC ranged from 0.773 to 0.848 and POD (Tuk) average MCC ranged from 0.770 to 0.843). Otherwise, when $n > 30$, POD (Tuk) has the second highest average MCC and MUOD (box) has the third highest average MCC across all sampling points (POD (Tuk) average MCC ranged from 0.828 to 0.851 and MUOD (box) average MCC ranged from 0.815 to 0.841). Then, MUOD (tan) has the fourth highest average MCC across all sample sizes and number of sampling points (average MCC ranged from 0.689 to 0.815), and TVD has the smallest average MCC across all sample sizes and number of sampling points (average MCC ranged from 0.633 to 0.785).

When classifying shape outliers, a similar trend was found with POD (user) having the highest average MCC, followed by POD (Tuk), MUOD (box), and then MUOD (tan) (Figure 2.4). The average MCC for each method ranged from 0.636 to 0.816 for POD (user), from 0.613 to 0.758 for POD (Tuk), from 0.572 to 0.674 for MUOD (box), and from 0.296 to 0.582 for MUOD (tan). One noticeable difference is that TVD has the highest average MCC when $n > 30$ and $T \leq 100$ (average MCC ranged from 0.800 to 0.852). When $n = 30$ and $T \leq 60$, TVD has the fourth highest average MCC, above MUOD (tan) (average MCC ranged from 0.428 to 0.623).

The comparison of methods remained unchanged when looking at different values of β (covariance roughness); and all methods showed a decrease in average MCC as β (or the roughness) increased (for both magnitude and shape outliers). The same comparison of best methods remained mostly unchanged when looking at different values for the proportion of outliers present, Δ. When

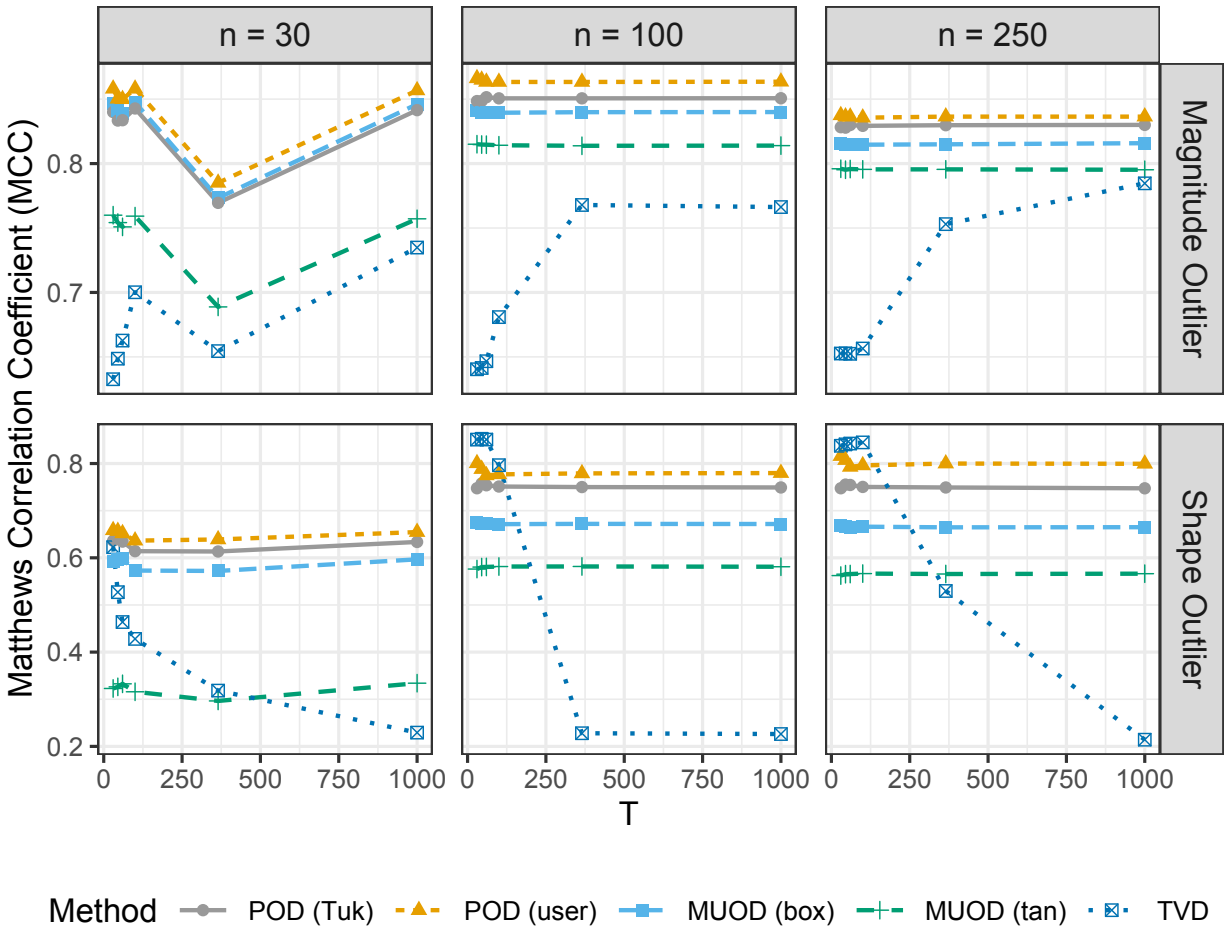


Figure 2.4: Average Matthew's Correlation Coefficient (MCC) for classifying magnitude and shape outliers on a varying number of sampling points (T = 30, 45, 60, 100, 365, 1000), faceted on the sample size (n = 30, 100, 250), and colored by the method type (POD (Tuk), POD (user), MS-Plot, MUOD (box), and TVD.

classifying shape outliers, POD (Tuk) has a higher average MCC (ranged from 0.801 to 0.810) than POD (user) (average MCC ranged from 0.785 to 0.796) for $\Delta = 0.15$. For $\Delta = 0.05$ when classifying shape outliers, TVD has the highest average MCC when $T \leq 100$ (ranged from 0.731 to 0.771). When classifying magnitude outliers, the only difference is that MUOD (box) had the highest average MCC when $\Delta = 0.15$ (ranged from 0.801 to 0.849; see Figure A.3 in Appendix A.1.1).

2.4 Case Study: World Population Growth

The World Population Growth data, collected by United Nations et al. (2016) and described in Dai et al. (2020) and Ojo et al. (2022), were chosen to illustrate and compare the methods in the univariate functional outlier detection setting. Populations were recorded for 105 countries, measured once annually on July 1st between 1950 and 2010 (for e.g., $T = 61$ sampling points per $n = 105$ observations). The 105 countries included in the data had populations greater than 1 million, and less than 15 million people, for the year of 1980. These data are available in `R` package `fdaoutlier` (Ojo et al., 2022).

The World Population Growth data do not have a presumed proportion of functional outliers. Thus, Tukey's classical boxplot threshold decision was used for POD. For this case study, a user specified threshold of 0.10 would return the same results as using Tukey's threshold (for a sample of normally distributed data, Tukey's threshold would be equivalent to a user threshold of 0.00335). Using the POD method, 11 out of the 105 countries were identified as functional outliers. Specifically, using POD, Netherlands was identified as a functional magnitude outlier, Sudan was identified as a functional magnitude and shape outlier, while Uganda, Ghana, Kazakhstan, Afghanistan, Nepal, Malaysia, Iraq, Saudi Arabia, and Australia were identified as functional shape (and amplitude) outliers (Figure 2.5).

PBOD, TVD, MS-Plot, MUOD (box), and MUOD (tan) methods were also applied to the world population growth data. When PBOD was used to identify outliers in the World Population Growth data, all 105 countries were identified as an outlier. Thus, PBOD is not included in the results. The frequency and percentages of outliers and their type (i.e., magnitude and shape) were computed for each of the five methods (Table 2.4). When using POD (Tuk) and TVD, a smaller subset of observations were classified as outliers than when using MS-Plot, MUOD (box), and MUOD (tan) (Table 2.4). Of the 11 observations that were identified as functional outliers when using POD (Tuk), 9 were identified by all other methods. Of the 14 observations that were identified as functional outliers using TVD, all 14 were also identified when using MUOD (box), but only 11 and 12 observations were identified when using MS-Plot and MUOD (tan), respectively.

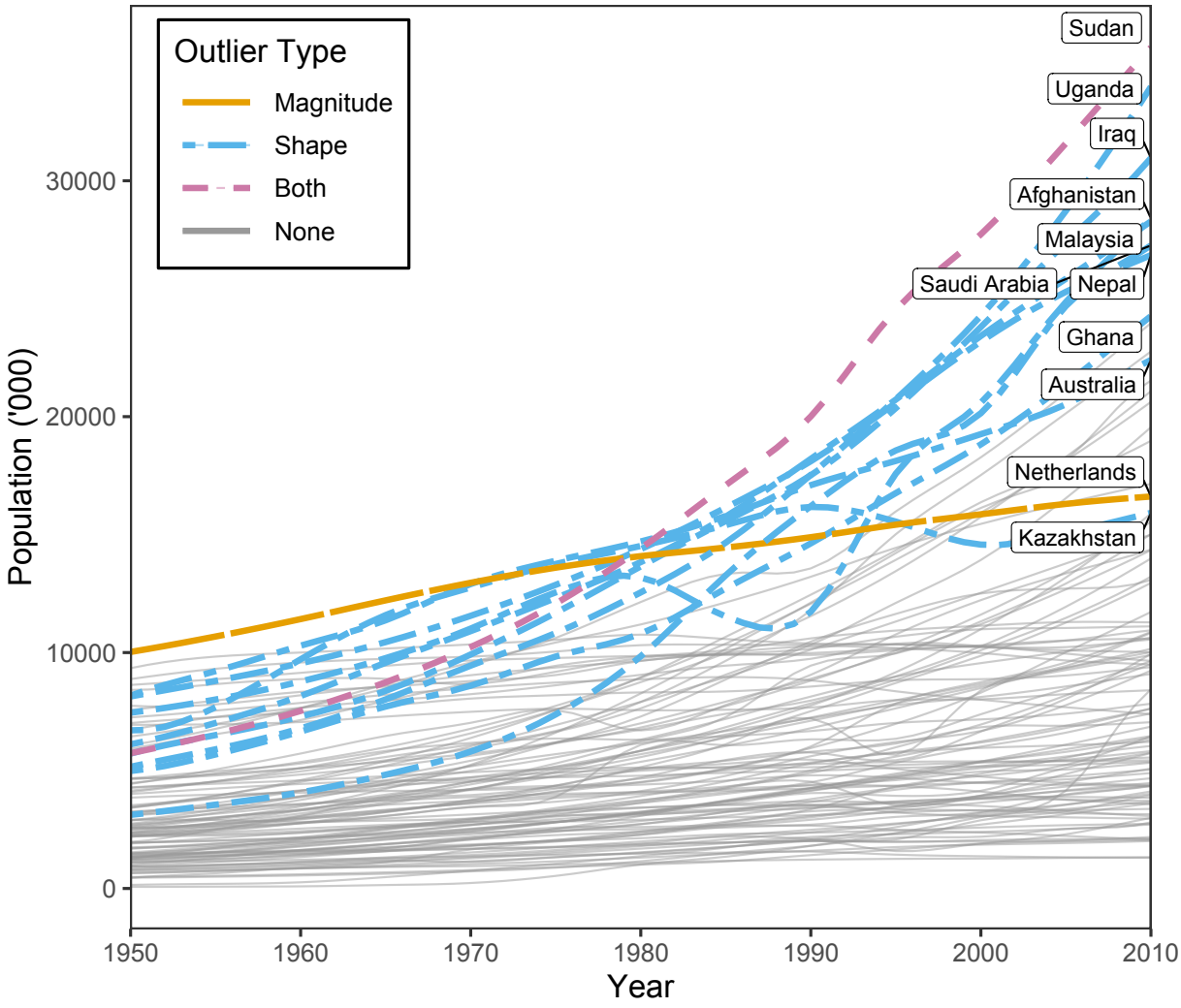


Figure 2.5: Countries identified as a functional outlier using POD (Tuk). The type of functional outlier (magnitude, shape, both, or none) for each country is indicated.

Most outliers were identified as shape outliers using all methods. One or more outliers were identified as magnitude outliers when using POD (Tuk), MUOD (box), and MUOD (tan). Uganda, Sudan, Ghana, Afghanistan, Malaysia, Iraq, and Saudi Arabia populations over the time period were identified as functional outliers using all methods.

Table 2.4: Frequency (%) of outliers identified in the World Population Growth data ($n = 105$) for all five methods, and the agreed number of outliers detected for pairs of the methods (i.e., concordance). The results of PBOD are not included, because all 105 countries were identified as a functional outlier when using PBOD.

Method	n (%)	Magnitude n (%)	Shape n (%)	POD	MS-Plot	TVD	MUOD (box)	MUOD (tan)
POD	11 (0.105)	2 (0.019)	10 (0.095)	–	9	9	9	9
MS-Plot	23 (0.219)	–*	–*	9	–	11	16	13
TVD	14 (0.133)	0 (0.000)	14 (0.133)	9	11	–	14	12
MUOD (box)	27 (0.257)	6 (0.057)	27 (0.257)	9	16	14	–	20
MUOD (tan)	24 (0.229)	9 (0.086)	24 (0.229)	9	13	12	20	–

* Magnitude and Shape outliers cannot be classified when using MS-Plot.

2.5 Conclusion

Two new methods of functional outlier detection were developed in this study: POD and PBOD. Functional outlier detection with POD can be implemented with the use of summary statistics. When implementing POD, classification diagnostics were as good or better than the competing methods (TVD, MS-Plot, and MUOD). As a result, POD is an approachable, functional outlier detection method. Furthermore, the use of POD does not rely on assumptions about the structure of the functional sample, and does not require the sample to be smoothed, allowing the user to implement functional outlier detection on a sample of raw functional data.

PBOD is implemented using a new simultaneous prediction band for the sample mean of a univariate sample of functional data. Simulation (Section 2.3) results also showed that PBOD had the lowest average ACC and average PPV, when compared to the competing methods. A simultaneous prediction band was computed with the use of the estimated functional mean and functional standard deviation: two statistics that are susceptible to outliers. The presence of outliers in a functional sample produces a prediction interval that is too wide, compared to when no outliers are present (Horn et al., 1988). It was hypothesized that a resampling algorithm (Section 2.2.2) could avoid the issue of wide prediction bands, but this did not resolve the interval width issue.

Although all methods vary in identifying functional outliers, MUOD (tan) and PBOD had the lowest ACC, PPV, and MCC (when applicable). MUOD (tan) was found to be prone to false positives, resulting in poor classification diagnostics (Vinue and Epifanio, 2021). When using

PBOD, all functional observations were often identified as outliers, leading to many false positive identifications. The high rate of false positives is reflected in PBOD's high sensitivity and low specificity.

The methodology in this paper and competing literature considers identifying an entire functional observation as an outlier, rather than looking for local features of outlyingness per observation. The competing methods (TVD, MS-Plot, and MUOD) can not be used to identify local features of outlyingness, unless the user specifically subsets the sampling domain before implementation. In contrast, POD returns a list of identified functional outliers and the count of extreme summary statistics per interval. Although not examined in this research, it is possible to identify intervals of outlyingness per observation with the local counts of extreme summary statistics.

If the user has an idea about the expected proportion of observations that should be functional outliers, then POD (user) has better classification diagnostics than POD (Tuk). However, the average MCC is comparable between POD (user; 0.633) and POD (Tuk; 0.691). If a secondary goal of the user is to properly classify the type of functional outlier, then POD has the greatest average MCC for identifying shape and/or magnitude outliers among the methods considered, whether or not the user presumes a proportion of outliers to be present.

POD relies on the use of intervals that split the sampling domain. If there are fewer than three sampling points within each interval, roughness (or curvature) can not be estimated. A random sample of functional data must have at least 24 sampling points ($T = 24$) for each observation in order to use the POD method (see Appendix A.2.1 for more details). Given a large enough number of sampling points for use of POD, the simulation results (Section 2.3) of POD (user) has better classification diagnostics than POD (Tuk) in nearly all cases. However, the user may not have an expected proportion of outliers present in their data, which is necessary for POD (user). In this case, the user is advised to use POD (Tuk) instead.

When classifying the type of functional outlier, POD (Tuk) and POD (user) can be used to classify magnitude and shape outliers. Other methodologies, such as MUOD, can be used to

classify the outliers as magnitude, shape, or amplitude outliers. The methodology used for POD was unable to differentiate shape and amplitude outliers.

The simultaneous prediction band meets nominal coverage levels when the functional sample meets the assumptions. However, a large proportion of observations were identified as outliers when using PBOD (e.g., there were several simulation iterations for which all observations were classified as an outlier when using PBOD). Even with resampling methods, the prediction bands created were often too wide to find any exceedances (see Section 2.1.4). In addition, PBOD was the only method that takes more than one second to implement, on average. Due to inaccuracy and inefficiency, PBOD is not useful for identifying outlying observations in a data set.

Only fixed interval sizes were considered in the development of POD, with approximately the same number of sampling points per interval. It could be possible that POD performs better with interval sizes that differ, based on the underlying roughness of the functional sample. If so, the roughness of the functional sample could be used to decide the size of each interval and the boundary locations. Another area of future exploration includes using a sliding window for an interval, rather than static intervals. A sliding window (or interval) could be created by setting the window size (or number of sampling points in the window) and step size (how many sampling points to move forward). For example, if the window size is 5 sampling points, the first interval would be the first five sampling points. If the step size is one sampling point, the second interval would be the second through the sixth sampling point. For a functional sample with $T = 30$, equidistant intervals of size five sampling points would create six unique intervals. By comparison, a sliding window with a window size of five sampling points and a step size of one sampling point would create 26 unique intervals. This approach could be implemented for POD by finding an optimal (in terms of MCC) window size and step size. The use of a sliding window may be difficult, because the optimal window size and step size may be dependent on the functional data structure. The use of a sliding window technique also requires estimating summary statistics many more times than static intervals, which would increase the computation time. Another potential solution is to use a

The simulation study and case study consider regularly spaced and observed functional data only. The competing methods (MUOD, MS-Plot, and TVD) do not consider irregularly spaced functional data and the code in `R` do not allow for irregularly spaced functional data (Ojo et al., 2021; R Core Team, 2021). However, MUOD, MS-Plot, and TVD could be mathematically extended to include empirical estimates for irregularly spaced functional data. A future step of development for POD would be to test its robustness in a scenario of irregularly spaced functional data. As long as three or more sampling points are present in a specified interval for the observation, all statistics can be estimated. For observations with less than 3 observed points on a specified interval, the estimates could be replaced with NA. Another possible solution would be to consider a dynamic interval.

POD has classification diagnostics as good or better than all other methods, while only relying on ordinary summary statistics. The summary statistics are estimated on several disjoint intervals after binning each functional observation on the sampling domain. This allows for a functional observation to be represented by a finite number of statistics, similar to ideas of graph-theoretic scagnostics (Tukey and Tukey, 1985; Wilkinson et al., 2005). Furthermore, POD has better classification diagnostics (i.e., ACC, PPV, and MCC) than the other methods in comparison for correctly classifying the type of functional outlier (shape and/or magnitude). POD results are easy to interpret and provide a summary of the number of extreme summary statistics of each type for each functional outlier identified. This summary can provide information beyond the identification of outliers. POD results are returned with a computation time less than a second, on average. The classification of the type of outlier simulation study presented illustrates the novelty and accuracy of this new method.

Chapter 3

Simultaneous Confidence and Prediction Band Methods for Estimating the Conditional Mean of a Functional Concurrent Regression Model

Abstract *Rule 6.3.4 in the Competition and Technical Rules of the World Athletics prevents any sprinter with a mechanical aid (e.g., a prosthesis) from participating in competitive sprint events with non-amputee sprinters, unless the sprinter is able to show "on the balance of probabilities" that the use of an aid would not provide a competitive advantage. At this point, there is no accepted statistical methodology that allows checking Rule 6.3.4. We develop and apply a novel* $(1-\alpha)\times 100\%$ *simultaneous prediction band (*SPB*), called fast and fair simultaneous prediction band (*FFSPB*). Our* FFSPB *has three properties relevant to the case study, which are not provided by alternative approaches: 1. fair predictive inference, 2. covariate-adjusted prediction, and 3. prediction in the presence of fat-tailed error term processes. Our* FFSPB *is compared to the competing methodology of conformal inference through a simulation study and the Sprint Start Kinetics case study. In the Sprint Start Kinetics case study, we illustrate the application of our* FFSPB *as a tool for checking whether the force curve of an amputee sprinter differs from the "normal" range of force curves of non-amputee sprinters, conditionally on predictor values that equal those of the amputee sprinter.*

3.1 Introduction

Determining if amputee sprinters have a competitive advantage due to their prostheses is a long-standing, unsolved methodological challenge in sports science. When amputee sprinters want to participate in the sprint events of non-amputee sprinters at the Olympic Games, they must adhere

to Rule 6.3.4 of World Athletics. The rule states that any mechanical aid (such as prostheses) is not allowed ...

> *"... unless on the balance of probabilities the use of an aid would not provide them with an overall competitive advantage over an athlete not using such aid." (World Athletics, 2020b,0)*

To this point, however, there is no accepted statistical methodology for assessing Rule 6.3.4—an issue that has challenged many scientists in the past. The lack of suitable statistical methodology has far-reaching negative consequences for athletes, sports associations, and societies' discussions of inclusivity in general. For instance, when prosthesis sprinter Blake Leeper made an attempt to compete in the Olympic Games in Tokyo (2020), he teamed up with scientists to produce the required scientific evidence to fulfill the requirements of Rule 6.3.4. Yet, due to the use of unsuitable statistical methodology, the scientists were not successful (Court of Arbitration for Sport, 2020). The highest-ranked publication in this context, Taboga et al. (2020), was even retracted by PLOS ONE since the main conclusions are not supported by the statistical analysis.

To judge "on the balance of probabilities" whether a prosthesis provides an amputee sprinter with an "overall competitive advantage," one needs to compare the amputee sprinter's movement and force patterns with a probabilistically justified range of movement and force patterns from *analogous* non-amputee sprinters. That is, one needs a $(1-\alpha)\times 100\%$ (e.g., where $\alpha = 0.1$ or $\alpha = 0.05$) prediction band with the following two main requirements: (a) it needs to be a *conditional* prediction band given the relevant characteristics (body measurements, height, age, gender, etc.) of the amputee sprinter to guarantee comparability, and (b) the band needs be able to process modern motion and force pattern data. Today's motion capture systems and force measurement devices record movement and force patterns at a very high resolution, leading to high-quality function-valued data. However, literature on prediction bands for functional data is scarce and underdeveloped, and none of the existing prediction bands for functional data allows conditioning on predictor values.

We address this methodological gap and contribute a novel $(1-\alpha)\times 100\%$ simultaneous prediction band (SPB), called fast and fair simultaneous prediction band (FFSPB). $\mathrm{FFSPB}_{1-\alpha}(Y_X(t))$ provides a "normal" range for functional data $Y(t)$, conditional on functional or non-functional covariates X. Our FFSPB has several properties relevant to our case study, which are not provided by alternative approaches:

(i) Fair predictive inference. For a given significance level $0 < \alpha < 1$ (e.g., $\alpha = 0.05$), our FFSPB has a global $(1-\alpha)\times 100\%$ simultaneous coverage probability over the total sampling domain $[a, b]$, as well as a local $(1-\alpha\frac{b_l-a_l}{b-a})\times 100\%$ simultaneous coverage probability over (disjoint) sub-intervals $[a_l, b_l] \subset [a, b]$ that partition the domain $[a_1, b_1]\cup\cdots\cup[a_L, b_L] = [a, b]$. Thus, detected effects in $[a, b]$ are significant at level α and effects in $[a_l, b_l]$ are significant at level $\alpha\frac{b_l-a_l}{b-a}$. This improved interpretability allows narrowing down simultaneous significance testing from global to local effects, corresponding to classic post-hoc testing approaches. To achieve this, we adapt the width of the band to the local roughness of the sample paths of the functional data, such that the false positive events are distributed proportionally ("fairly") to each sub-interval $[a_l, b_l]$, $l = 1, \ldots, L$. The latter is important since the sample paths of biomechanical functional data, as used in our case study, typically do not have homogeneous roughness. For this feature, we use the results on fair simultaneous confidence bands in Liebl and Reimherr (2023) and adapt them to the case of fair SPB.

(ii) Covariate-adjusted. Using a concurrent function-on-function regression model

$$Y(t) = \beta_1(t) + \sum_{k=2}^{K} X_k(t)\beta_k(t) + \varepsilon(t), \quad t \in [a, b]$$

allows us to provide SPB conditionally on functional as well as non-functional covariates $X_k(t) = x_k(t)$, $k = 1, \ldots, K$.

(iii) To take into account the possibility of fat-tailed error term processes, we allow for a non-standardized t-distributed error process $\varepsilon = \{\varepsilon(t), t \in [a, b]\}$ and estimate the degrees of freedom parameter from the data.

(iv) In addition to our fair and covariate-adjusted SPBs, denoted FFSPBs, we also contribute fair and covariate-adjusted simultaneous confidence bands (SCBs), denoted FFSCBs for the regression parameter functions $\beta_k = \{\beta_k(t), t \in [a, b]\}$. The FFSCBs do not require a specific distribution assumption, since our asymptotic theory results only require error processes $\varepsilon = \{\varepsilon(t), t \in [a, b]\}$ with finite variances and two-times continuously differentiable sample paths.

In Figure 3.1, we illustrate the application of our FFSPB as a tool for assessing whether the force curve of an amputee sprinter differs from the "normal" range of force curves of non-amputee sprinters, conditional on predictor values that equal those of the amputee sprinter. At the start of a sprint event, athletes rest their front and back foot on an angled starting block. The push-off phase of a sprinter (measured for each foot) starts at zero percent when the foot begins to push against the block and ends at one hundred percent when the maximum force of the sprinter occurs. The force curve data we consider is the vertical force component (measured in Newton per kilograms (N/kg)) generated in the front starting block during the push-off phase of the sprint start. The data set was originally collected and described by Willwacher et al. (2016). The study sample in Willwacher et al. (2016) considers the sprint start kinetics of 154 non-amputee sprinters and 7 amputee sprinters. In Figure 3.1, the force curve of the seventh amputee sprinter (black dashed line) is outside the 90% FFSPB (blue shaded region) just slightly before 50% of the push-off phase. One exceedance or more from our FFSPB is a non-coverage event, and we observe such non-coverage events for 5 of the 7 amputee sprinters. Each of the five non-coverage events happens in the middle part of the push-off phase (from $33.\bar{3}\%$ to $66.\bar{6}\%$) over which our FFSPB is a $(1-0.1/3)\times 100\% = 96.67\%$ SPB. Five non-coverage events is strong evidence for a difference in the force curves of amputee sprinters and non-amputee sprinters in the middle phase of the sprint start.

The literature on SPBs for functional data is initiated in the 1990s (Olshen et al., 1989; Lenhoff et al., 1999). The early works of Olshen et al. (1989) and Lenhoff et al. (1999) are inspired by biomechanic curve data, but do not allow for covariate adjustments. More recently, Franco-Villoria

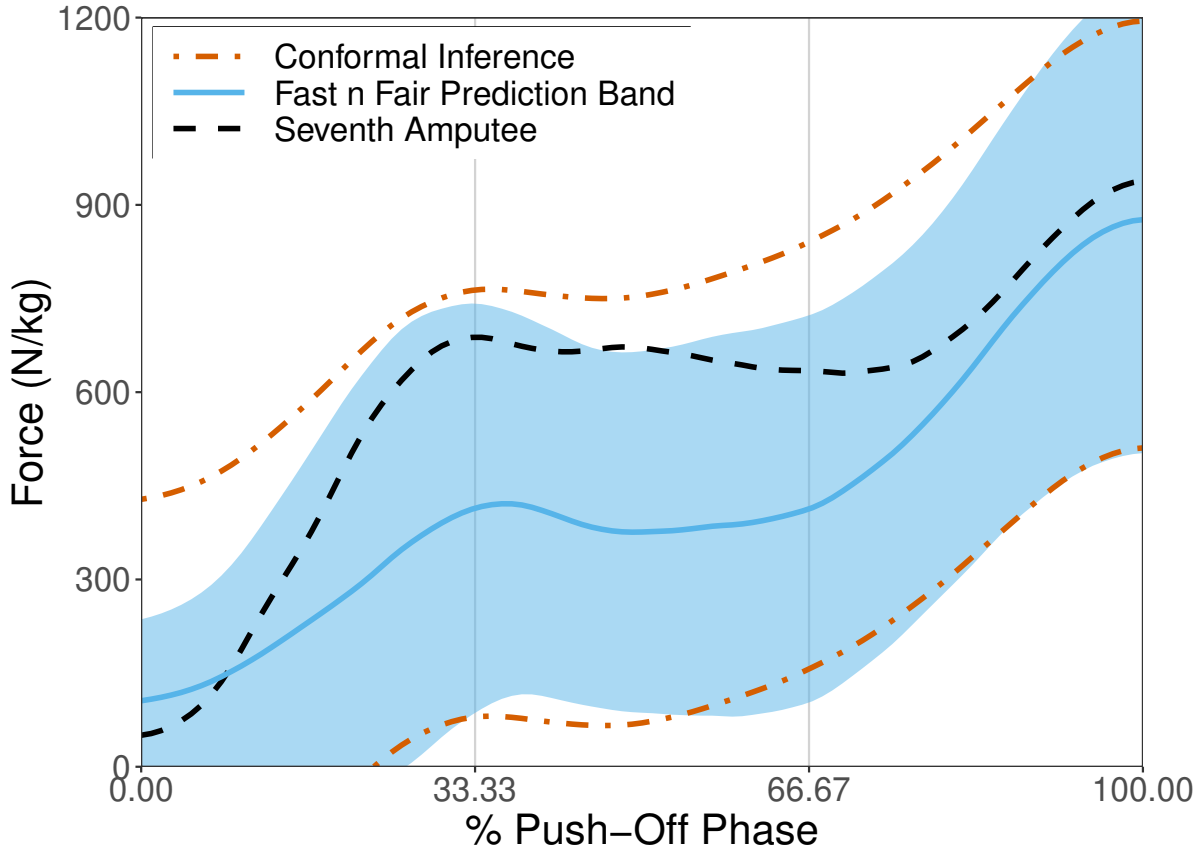


Figure 3.1: Estimated 90% simultaneous prediction bands (SPBs) for a non-amputee sprinter's front vertical force created by using conformal inference (dot-dashed, orange) and fast and fair (solid, blue, shaded region). The solid, blue line represents the predicted vertical front force, $\hat{Y}_{x_7}(t)$, for a sprinter with the same demographics as the seventh amputee sprinter (e.g., $X(t) = x_7(t)$). The dashed, black line represents the observed front vertical force of the seventh amputee sprinter, $Y_{x_7}(t)$.

and Ignaccolo (2017) and Paparoditis and Shang (2023) develop SPBs for spatial functional data and functional time series, respectively, but also these works do not take into account covariate adjustments. In related works, Rathnayake and Choudhary (2016) and de Silva and Choudhary (2023) propose bootstrap-based simultaneous tolerance bands (STB) for Gaussian and exponential family functional data. However, the STBs of Rathnayake and Choudhary (2016) and de Silva and Choudhary (2023) also do not allow for covariate adjustments.

To this date, the only method for obtaining an SPB for a predicted curve from a function-on-function regression model is the conformal inference approach of Diquigiovanni et al. (2022) and

Fontana et al. (2023) which extend conformal inference (Vovk and Shafer, 2008) to the case of functional data with covariates. The recent developments of conformal inference methods provide important contributions to the literature on SPBs, since they are effectively distribution-free. However, as shown in Lei and Wasserman (2014), prediction regions created by conformal inference, such as the SPBs of Diquigiovanni et al. (2022) and Fontana et al. (2023), can only provide marginal coverage guarantees. Using conformal inference does not provide coverage conditionally on given covariate values as is required in our case study. This "marginal vs. conditional on covariate values"-issue can be seen in our simulation studies (Section 4.3), which demonstrate that the SPBs based on conformal inference are overly wide and thus hide important features. The issue is also demonstrated in Figure 3.1, where it can be seen that the conformal inference SPB (orange dashed-dotted) is generally wider than our FFSPB and is not centered around the predicted curve (solid blue line). None of the above literature cited on SPBs for functional data allow for local predictive inference over sub-intervals, $[a_l, b_l] \subset [a, b]$, which turns out to be very useful in our case study where all non-coverage events occur in the middle of the function-domain.

The literature on SCBs is substantially broader than the literature on SPBs, but the majority of contributions also do not allow for covariate adjustments (see, for instance, Bunea et al., 2011; Cao et al., 2012; Cao, 2014; Degras, 2017; Wang et al., 2020; Telschow and Schwartzman, 2022; Liebl and Reimherr, 2023). SCBs that take into account covariate adjustments in a function-on-scalar regression model are contributed by Chang et al. (2017) and Abramowicz et al. (2018). Belloni et al. (2018) develop a very general theory that allows constructing SCBs for functional parameters. For example, Belloni et al. (2018) developed a SCB for a logistic function-on-function regression model in a high-dimensionality context. None of the latter works on SCBs, however, allow for local inference over sub-intervals, $[a_l, b_l] \subset [a, b]$. Most similar to our SCB is the SCB of Ecker et al. (2024), which, like ours, builds upon the work of Liebl and Reimherr (2023). However, the SCB in Ecker et al. (2024) requires a Gaussian error process, while our SCB does not require a specific distributional assumption. As previously mentioned, our asymptotic theory results only

assume error processes $\varepsilon = \{\varepsilon(t), t \in [a, b]\}$ with finite variances and two-times continuously differentiable sample paths.

The rest of the manuscript is organized as follows: the model, estimators, and FFSPB are given in Section 3.2, a simulation comparison between conformal inference and fast and fair is presented in Section 4.3, the Sprint Start Kinetics of Amputee and Non-Amputee Sprinters case study is displayed in Section 3.4, and the manuscript concludes in Section 3.5.

3.2 Theory & Methods

3.2.1 Model and Assumptions

Let us consider the time-varying coefficient model (Hastie and Tibshirani, 1993), also called concurrent function-on-function linear regression model (Ramsay and Silverman, 2005, Ch. 14). "Concurrent" implies the model is regressed pointwise at each sampling point in the domain. The model can be expressed for a single observation as

$$Y(t) = X^T(t)\beta(t) + \varepsilon(t), \quad \text{for} \quad t \in [0, 1] \tag{3.1}$$

where $Y = \{Y(t) \in \mathbb{R},\, t \in [0, 1]\}$ and $X = (X_1, \ldots, X_K)^T$, with $X_k = \{X_k(t) \in \mathbb{R},\, t \in [0, 1]\}$, for $k = 1, \ldots, K$, and $\varepsilon = \{\varepsilon(t) \in \mathbb{R},\, t \in [0, 1]\}$ are continuous stochastic processes with constant $X_1(t) = 1$ for all $t \in [0, 1]$, and continuous parameter functions $\beta = (\beta_1, \ldots, \beta_K)^T$ with $\beta_k = \{\beta_k(t),\, t \in [0, 1]\}$ for $k = 1, \ldots, K$. Considering the standard unit-interval $[0, 1] \subset \mathbb{R}$ is without loss of generality since any compact interval $[a, b] \subset \mathbb{R}$ can be standardized to $[0, 1]$. We consider the case of an independent and identically distributed (iid) random sample $(Y_1, X_1^T), \ldots, (Y_n, X_n^T) \overset{iid}{\sim} (Y, X^T)$, where the error term $\varepsilon = Y - X^T\beta$ is mean independent of X, i.e. $E(\varepsilon(t)|X(t)) = 0$ for all $t \in [0, 1]$. Note that the case of function-on-scalar regression is included in (3.1), since any scalar predictor X_k can be defined as a constant predictor function $X_k(t) = X_k(t')$ for all $t, t' \in [0, 1]$.

In this work, we focus on SPBs and SCBs that may also be used in small samples (e.g., $n = 15$). For small sample sizes, resampling strategies like conformal inference (Vovk and Shafer, 2008; Diquigiovanni et al., 2022; Fontana et al., 2023) do not provide accurate and precise SPBs. Therefore, we take a more traditional route and consider a heavy-tailed stochastic error process, ε, allowing us to construct SPBs and SCBs that can be used in the case of heavy-tailed phenomena, or lead to conservative inference in the case of thin-tailed phenomena. The conservative inference is particularly useful in cases with costly false positives, as in our application (see Section 3.4).

We assume that ε is a Student's t type of process with $\nu_0 > 4$ degrees of freedom defined as

$$\varepsilon \overset{d}{=} Z\sqrt{\frac{\nu_0}{\chi^2_{\nu_0}}},$$

where $\chi^2_{\nu_0}$ is a real Chi-squared distributed random variable with ν_0 degrees of freedom, $Z = \{Z(t),\, t \in [0,1]\}$ is a mean-zero Gaussian process with continuous covariance kernel $E(Z(t)Z(s)) = \sigma_Z(t,s)$, and where Z and $\chi^2_{\nu_0}$ are independent. Thus, the covariance kernel of the error term ε is given by

$$E(\varepsilon(t)\varepsilon(s)) = E(Z(t)Z(s))E\left(\frac{\nu_0}{\chi^2_{\nu_0}}\right) = \sigma_Z(t,s)\frac{\nu_0}{\nu_0 - 2} = \sigma_\varepsilon(t,s).$$

Note that Model (3.1) contains the case of Gaussian errors in the special case where $\nu_0 \to \infty$. Moreover, for any set of $J \in \mathbb{N}$ distinct evaluation points, $t_1 < \cdots < t_J$, the random vector $(\varepsilon(t_1), \ldots, \varepsilon(t_J))$ is multivariate Student's t distributed with ν_0 degrees of freedom and $J \times J$ scaling matrix $(\sigma_Z(t_j, t_{j'}))_{1 \le j,j' \le J}$.

3.2.1.1 Assumptions

The following list contains our theoretical assumptions under which we develop our pointwise and uniform confidence and prediction bands.

A1 Linear model:

$$Y(t) = X^T(t)\beta(t) + \varepsilon(t), \quad t \in [0,1],$$

where $X = \{X(t) \in \mathbb{R}^K, t \in [0,1]\}$ is a K-dimensional vector-valued random function $X(t) = (X_1(t), \ldots, X_K(t))^T$ with intercept $X_1(t) = 1$, for all $t \in [0,1]$, and where $\varepsilon = \{\varepsilon(t), t \in [0,1]\}$ denotes the unobserved error function with $\mathbb{E}(\varepsilon(t)|X(t)) = 0$ for all $t \in [0,1]$.

A2 Sampling: (Y_i, X_i^T), $i = 1, \ldots, n$, is iid as (Y, X^T).

A3 Smoothness and uniform 2nd moments: The random functions $X_1, \ldots, X_K$, and ε are twice continuously differentiable, i.e. $X_1, \ldots, X_K, \varepsilon \in C^2[(0,1)]$ almost surely. Moreover, the means of the random functions, $\mathbb{E}(X_1), \ldots, \mathbb{E}(X_K)$ and the parameters β_j, $j = 1, \ldots, K$, are also twice continuously differentiable; i.e., $\mathbb{E}(X_j), \beta_j \in C^2[(0,1)]$ for each $j = 1, \ldots, K$. Let $G_k^{(d)}$ and $G_{k'}^{(d)}$ denote the dth derivative of the kth and k'th element of the $(K+1)$-dimensional vector-valued random function $G = (X_1, \ldots, X_K, \varepsilon)^T$. For all $d \in \{1, 2\}$ and all $k, k' \in \{1, \ldots, K+1\}$

$$\mathbb{E}\left(\sup_{t\in[0,1]}\left(G_k^{(d)}(t)G_{k'}^{(d)}(t)\right)^2\right) \leq C < \infty,$$

where $0 < C < \infty$ denotes a generic constant depending only on k, k', and d.

The error ε is a Student's t type of process with $\nu_0 > 4$ degrees of freedom defined as $\varepsilon \stackrel{d}{=} Z\sqrt{\frac{\nu_0}{\chi^2_{\nu_0}}}$, where $\chi^2_{\nu_0}$ is a real Chi-squared distributed random variable with ν_0 degrees of freedom, $Z = \{Z(t),\, t \in [0,1]\}$ is a mean zero Gaussian process with continuous covariance kernel $E(Z(t)Z(s)) = \sigma_Z(t,s)$, and where Z and $\chi^2_{\nu_0}$ are independent. The covariance kernel of the error term ε is assumed to not depend on the predictor functions X, i.e.

$$E(\varepsilon(t)\varepsilon(s)|X) = E(\varepsilon(t)\varepsilon(s)) = E(Z(t)Z(s))E\left(\frac{\nu_0}{\chi^2_{\nu_0}}\right) = \sigma_Z(t,s)\frac{\nu_0}{\nu_0 - 2} = \sigma_\varepsilon(t,s).$$

A4 Rank condition: $\operatorname{rank}\left(\mathbb{E}(X^T(t)X(t))\right) = K$ for all $t \in [0,1]$.

Assumption **A1** is our linear model assumption with mean-independent error terms. Assumption **A2** requires an iid random sampling scheme. However, our results also hold for weakly

dependent stationary functional time series. Assumption **A3** generalizes the moment assumptions from the classic regression theory to the case of functional predictors and error terms. We develop a central limit theorem within the space of continuous functions, which is an infinite dimensional problem. Similar to high-dimensional statistical problems, this infinite dimensional nature requires us to impose some structure on the stochastic properties of the sample paths of the functional data. By assuming the first and second derivative functions have uniform second moments, we require the sample paths to be smooth and have finite variances. For example, Brownian motions do not meet the requirement of smooth sample paths. However, the typical applications considered in functional data analysis are concerned with relatively smooth functions and thus fit into our theoretical framework. Assumption **A4** is the usual rank assumption required to identify β.

3.2.2 Pointwise Confidence and Prediction Bands

We begin with the derivation of pointwise $(1-\alpha)\times 100\%$ (e.g., where $\alpha = 0.05$) confidence (CB) and prediction bands (PB) for which the coverage probability of $(1-\alpha)\times 100\%$ is guaranteed to hold pointwise for each $t \in [0,1]$. In Section 3.2.3 we generalize them to SCBs and SPBs for which the coverage probability of $(1-\alpha)\times 100\%$ is guaranteed to hold simultaneously for all $t \in [0,1]$.

3.2.2.1 Pointwise Confidence Band

Let $(Y_1, X_1^T), \ldots, (Y_n, X_n^T) \overset{iid}{\sim} (Y, X^T)$ denote the iid sample that is used for estimating the unknown parameters of Model (3.1), and let $(Y_{x_{new}}, x_{new}^T) \sim (Y, X^T)$ be a new observation from the same distribution. The best (in the mean squared error sense) prediction of $Y_{x_{new}}(t)$ given some observed predictor $X(t) = x_{new}(t)$ is the conditional mean $Y_{x_{new}}(t) = E\left[Y(t)|X(t) = x_{new}(t)\right]$, where under Model (3.1), $Y_{x_{new}}(t) = x_{new}^T(t)\beta(t)$. We estimate the conditional mean response $Y_{x_{new}}(t)$ by

$$\hat{Y}_{x_{new}}(t) = x_{new}^T(t)\hat{\beta}(t), \tag{3.2}$$

where $\hat{\beta}(t) = (\beta_1(t), \ldots, \beta_K(t))^T$ denotes the pointwise ordinary least squares (OLS) estimator

$$\hat{\beta}(t) = \left(\sum_{i=1}^{n} X_i(t) X_i^T(t) \right)^{-1} \sum_{i=1}^{n} X_i(t) Y_i(t) \tag{3.3}$$

that may be evaluated at each time point $t \in [0, 1]$.

From classic asymptotic results about the linear regression model (e.g., Seber and Lee (2003)) it follows that

$$\sqrt{n} \left(x_{new}^T(t)\hat{\beta}(t) - x_{new}^T(t)\beta(t) \right) \rightarrow_D \mathcal{N} \left(0, \sigma_\varepsilon(t,t) x_{new}^T(t) E \left(X_i(t) X_i^T(t) \right)^{-1} x_{new}(t) \right)$$

pointwise for each $t \in [0, 1]$ as $n \rightarrow \infty$. Inverting this asymptotic statistic leads to the pointwise $(1 - \alpha) \times 100\%$ CB for $Y_{x_{new}}(t) = x_{new}^T(t)\beta(t)$

$$\begin{aligned} &\mathrm{CB}_{1-\alpha}(Y_{x_{new}}(t)) = \\ &\quad [x_{new}^T(t)\hat{\beta}(t) \pm q_Z(\alpha/2) \left(\frac{1}{n} \sigma_\varepsilon(t,t) x_{new}^T(t) E \left(X_i(t) X_i^T(t) \right)^{-1} x_{new}(t) \right)^{1/2}], \end{aligned} \tag{3.4}$$

where $q_Z(\alpha/2)$ denotes the $1 - \alpha/2$ quantile of the standard normal distribution.

The feasible version of (3.4) is given by plugging in an estimator, $\hat{\sigma}_\varepsilon(t,t)$, for $\sigma_\varepsilon(t,t)$:

$$\begin{aligned} &\widehat{\mathrm{CB}}_{1-\alpha}(Y_{x_{new}}(t)) = \\ &\quad [x_{new}^T(t)\hat{\beta}(t) \pm q_Z(\alpha/2) \left(\frac{1}{n} \hat{\sigma}_\varepsilon(t,t) x_{new}^T(t) \frac{1}{n} \sum_{i=1}^{n} \left(X_i(t) X_i^T(t) \right)^{-1} x_{new}(t) \right)^{1/2}]. \end{aligned} \tag{3.5}$$

Possible estimators, $\hat{\sigma}_\varepsilon(t,t)$, for $\sigma_\varepsilon(t,t)$ are given below in Equations (3.9) and (3.10).

3.2.2.2 Pointwise Prediction Band

To construct a pointwise $(1 - \alpha) \times 100\%$ PB for $Y_{x_{new}}(t) = x_{new}^T(t)\beta(t) + \varepsilon_{new}(t)$, we need to consider that the error term $\varepsilon_{new}(t)$ in Model (3.1) is pointwise Student's t distributed with ν_0 degrees of freedom. Under our assumptions (see Section 3.2.1.1), it follows from classic regression

theory (e.g., Seber and Lee (2003)) that $|x_{new}^T(t)\hat{\beta}(t) - x_{new}^T(t)\beta(t)| \to_P 0$ pointwise for each $t \in [0,1]$. Given this convergence in probability and our assumptions on the error term (see Section 3.2.1.1), applying Slutsky's theorem results in

$$x_{new}^T(t)\hat{\beta}(t) - x_{new}^T(t)\beta(t) + \varepsilon_{new}(t) \to_D Z(t)\sqrt{\frac{\nu_0}{\chi^2_{\nu_0}}}, \tag{3.6}$$

as $n \to \infty$ pointwise for each $t \in [0,1]$, where $Z\sqrt{\nu_0/\chi^2_{\nu_0}}$ is a scaled Student's t process with ν_0 degrees of freedom and variance covariance function $\sigma_Z(s,t) \cdot \frac{\nu_0}{\nu_0 - 2}$.

An asymptotic, pointwise, $(1-\alpha) \times 100\%$ PB for a new random outcome $Y_{x_{new}}(t) = x_{new}^T(t)\beta(t) + \varepsilon_{new}(t)$ is given by inverting the asymptotic statistic (3.6):

$$\mathrm{PB}_{1-\alpha}(Y_{x_{new}}(t)) = \left[x_{new}^T(t)\hat{\beta}(t) \pm q_{t_{\nu_0}}(\alpha/2)\left(\sigma_Z(t,t)\right)^{1/2}\right], \tag{3.7}$$

where $q_{t_{\nu_0}}(\alpha/2)$ denotes the $1-\alpha/2$ quantile of the Student's t distribution with ν_0 degrees of freedom. To see this, observe that $Y_{x_{new}}(t) = x_{new}^T(t)\hat{\beta}(t) + \left(x_{new}^T(t)\beta(t) - x_{new}^T(t)\hat{\beta}(t)\right) + \varepsilon_{new}(t)$. Thus, we must account for the variability due to $\left(x_{new}^T(t)\beta(t) - x_{new}^T(t)\hat{\beta}(t)\right)$ and $\varepsilon_{new}(t)$, where the former is asymptotically negligible.

The PB in (3.7) is, of course, not feasible since we do not know $\sigma_Z(t,t)$, so we need to use an empirical version. To additionally account for the variability due to $|x_{new}^T(t)\hat{\beta}(t) - x_{new}^T(t)\beta(t)| \neq 0$ in finite samples, we also take into account the standard deviations of $x_{new}^T(t)\hat{\beta}(t)$. Using estimates for the variance components, $\sigma_Z(s,t)$ and $\sigma_\varepsilon(s,t)$, leads to the following feasible version of the pointwise PB with finite-sample correction:

$$\widehat{\mathrm{PB}}_{1-\alpha}(Y_{x_{new}}(t)) = \left[x_{new}^T(t)\hat{\beta}(t) \pm q_{t_{\nu_0}}\left(\frac{\alpha}{2}\right)\left(\hat{\sigma}_Z(t,t) + \hat{\sigma}_\varepsilon(t,t)x_{new}^T(t)\left(\sum_{i=1}^n X_i(t)X_i^T(t)\right)^{-1}x_{new}(t)\right)^{1/2}\right]. \tag{3.8}$$

Our feasible pointwise CB (3.5) and PB (3.8) require estimates, $\hat{\sigma}_\varepsilon(s,t)$ and $\hat{\sigma}_Z(s,t)$, for the typically unknown variance components, $\sigma_\varepsilon(s,t)$ and $\sigma_Z(s,t)$, of the error term. In this paper, we consider the unbiased (ub) and the maximum-likelihood (ml) estimators of $\sigma_\varepsilon(s,t)$:

$$\hat{\sigma}_\varepsilon^{ub}(s,t) = \frac{1}{n-K}\sum_{i=1}^{n} e_i(s)e_i(t) \quad \text{with} \quad e_i(t) = Y_i(t) - X_i^T(t)\hat{\beta}(t). \tag{3.9}$$

and

$$\hat{\sigma}_\varepsilon^{ml}(s,t) = \frac{1}{n}\sum_{i=1}^{n} e_i(s)e_i(t). \tag{3.10}$$

Using the maximum-likelihood estimator, $\hat{\sigma}_\varepsilon^{ml}(s,t)$, results in slightly narrower bands than using the unbiased estimator, $\hat{\sigma}_\varepsilon^{ub}(s,t)$. However, as also shown in our simulations, when the sample size is small (e.g., $n \leq 30$), the unbiased estimator, $\hat{\sigma}_\varepsilon^{ub}(s,t)$, should be used to ensure proper coverage levels. Note that the matrix of covariance estimates, $\hat{\sigma}_\varepsilon(s,t)$, will be singular when $n < T$. Since we are specifying functional concurrent regression, we do not need to invert $\hat{\sigma}_\varepsilon(s,t)$, so this is not a problem.

To estimate $\sigma_Z(s,t)$, we need to estimate the degrees of freedom, ν_0, of the error process. Singh (1988) develops the following estimator for ν_0:

$$\hat{\nu}_0(t) = \frac{2(2\hat{a}(t)-3)}{\hat{a}(t)-3} \quad \text{with} \quad \hat{a}(t) = \frac{n^{-1}\sum_{i=1}^{n} e_i(t)^4}{\left(n^{-1}\sum_{i=1}^{n} e_i(t)^2\right)^2}. \tag{3.11}$$

Given an estimate of ν_0 and σ_ε, we estimate σ_Z as

$$\hat{\sigma}_Z(s,t) = \frac{\hat{\nu}_0 - 2}{\hat{\nu}_0}\hat{\sigma}_\varepsilon(s,t). \tag{3.12}$$

Note, any missing values of $\hat{\nu}_0(t)$ are replaced with $4+\varepsilon$ and $\hat{\nu}_0 = \min_{t\in[0,1]} \hat{\nu}_0(t)$.

3.2.3 Simultaneous Confidence and Prediction Bands

To construct our SCBs and SPBs, we need to estimate the parameter

$$\tau_\varepsilon(t) = Var(\tilde{\varepsilon}'(t)) \quad \text{with} \quad \tilde{\varepsilon}(t) = \frac{\varepsilon(t)}{\sqrt{\sigma_\varepsilon(t,t)}}$$

which can be estimated by

$$\hat{\tau}_\varepsilon(t) = \frac{1}{n}\sum_{i=1}^{n}\left(\tilde{e}_i'(t) - \left(\frac{1}{n}\sum_{i'=1}^{n}\tilde{e}_{i'}'(t)\right)\right)^2 \quad \text{with} \quad \tilde{e}_i(t) = \frac{e_i(t)}{\sqrt{\hat{\sigma}_\varepsilon(t,t)}}.$$

The parameter $\tau_\varepsilon = \{\tau_\varepsilon(t) : t \in [0,1]\}$ measures the local roughness of the standardized sample paths of the error process ε. Thus, τ_ε allows quantifying the extent of the multiple testing problem over the function domain $[0,1]$. For our method, we need to assume that $\tau_\varepsilon(t) > 0$, for all $t \in [0,1]$, which boils down to assuming a certain degree of complexity in the sample paths of the error process ε. For instance, simple error processes like $\varepsilon(t) = A + b\,t$ with random intercepts A, but constant slope b are ruled out, since for such a process we would have that $\tau_\varepsilon(t) = Var(\tilde{\varepsilon}'(t)) = Var(b/Var(A)) = 0$. Thus, requiring that $\tau_\varepsilon(t) > 0$, for all $t \in [0,1]$ rules out processes for which multiple testing across $t \in [0,1]$ is not problematic anyway since test decisions are equal for all $t \in [0,1]$. However, the assumption $\tau(t) > 0$, for all $t \in [0,1]$, includes processes with a stochastic slope at each $t \in [0,1]$. When a stochastic slope is present, it is necessary to account for the multiple testing problem across $t \in [0,1]$, since the test decisions will generally differ across $t \in [0,1]$. Generally, the larger τ_ε, the rougher the sample paths of the error process and the more uncorrelated the test decisions across $t \in [0,1]$.

3.2.3.1 Simultaneous Confidence Bands

Next, we establish the uniform convergence of $\hat{\beta}$, $\hat{\sigma}_\varepsilon^{type}$, and $\hat{\tau}_\varepsilon$; and show that $\hat{\beta}$ satisfies the functional central limit theorem (CLT). The theorem of convergence is stated here:

Theorem 3.2.1 (Uniform Convergences)**.** *Under Assumptions **A1, A2, A3, A4***

(a) $\sup_{k\in\{1,\dots,K\}}\sup_{t\in[0,1]}|\hat{\beta}_k^{(d)}(t) - \beta_k^{(d)}(t)| \xrightarrow{a.s.} 0$ *for each* $d \in \{0,1\}$,

(b) $\sup_{t\in[0,1]} |\hat{\sigma}_{\varepsilon}^{(d,d)}(s,t) - \sigma_{\varepsilon}^{(d,d)}(s,t)| \xrightarrow{a.s.} 0$ *for each* $d \in \{0,1\}$,

(c) $\sup_{t\in[0,1]} |\hat{\tau}_{\varepsilon}(t) - \tau_{\varepsilon}(t)| \xrightarrow{a.s.} 0$.

Next, the functional central limit theorem of the regression parameters is stated here:

Theorem 3.2.2 (CLT for Functional Regression Parameter)**.** *Let X, Y, and* ε *be continuous processes meeting assumptions* ***A1****,* ***A2****, and* ***A4****. Let* $\hat{\beta} = \left\{\hat{\beta}(t), t \in [0,1]\right\}$ *be the ordinary least squares (OLS) estimator of* $\beta = \{\beta(t), t \in [0,1]\}$ *as defined in Equation (3.3), then*

$$\sqrt{n}\left(\hat{\beta} - \beta\right) \rightarrow_D \mathcal{G}(0, c_{\beta}),$$

where $\mathcal{G}(0, c_{\beta})$ *is a mean-zero Gaussian process with covariance function defined as*

$$c_{\beta}(s,t) = \sigma_{\varepsilon}(s,t) E\left[X_i(s) X_i^T(t)\right]^{-1}.$$

Let $x_{new}^T = \left\{x_{new}^T(t) : t \in [0,1]\right\}$. As a result of Theorem 3.2.2, it now follows by Slutksy's functional theorem that

$$\sqrt{n}\left(x_{new}^T\hat{\beta} - x_{new}^T\beta\right) \rightarrow_D \mathcal{G}\left(0, c_{x_{new}^T\beta}\right),$$

where $\mathcal{G}$ is a mean-zero Gaussian process with covariance function defined as $c_{x_{new}^T\beta}(s,t) = \sigma_{\varepsilon}(s,t) x_{new}^T(s) E\left(X_i(s) X_i^T(t)\right)^{-1} x_{new}(t)$. Note that $c_{x_{new}^T\beta}(s,t)$ only requires inversion of $X_i(s)X_i^T(t)$ pointwise, which results in $K \times K$ matrices.

As is shown in the pointwise bands, this asymptotic result can be inverted to acquire pointwise CBs. If we replace the pointwise critical value $q(\alpha/2)$ with a functional critical value, $u_{\alpha/2}^{\star}(t)$, then simultaneous bands can be constructed. Corollary 3.2 from Liebl and Reimherr (2023) establishes estimating equations for computing the fast and fair critical value function, denoted $u_{Z_{\alpha/2}}^{\star}(t)$, when the process is Gaussian. Let $Y_{x_{new}} = \left\{Y_{x_{new}}(t) = x_{new}^T(t)\beta(t) : t \in [0,1]\right\}$,

$u^{\star}_{Z_{\alpha/2}} = \left\{u^{\star}_{Z_{\alpha/2}}(t) : t \in [0,1]\right\}$, and $\sigma_{\varepsilon} = \{\sigma_{\varepsilon}(t,t) : t \in [0,1]\}$. Now, we invert the asymptotic statistic and use $u^{\star}_{Z_{\alpha/2}}$ for the critical value, which leads to the fast and fair $(1-\alpha)\times 100\%$ SCB (FFSCB) for $Y_{x_{new}}$:

$$\text{FFSCB}_{1-\alpha}(Y_{x_{new}}) = \left[x^T_{new}\hat{\beta} \pm u^{\star}_{Z_{\alpha/2}}\left(\frac{1}{n}\sigma_{\varepsilon}\cdot x^T_{new}E\left(X_iX_i^T\right)^{-1}x_{new}\right)^{1/2}\right]. \tag{3.13}$$

The feasible version of Equation 3.13 is given by using an estimator for σ_{ε} and for $u^{\star}_{Z_{\alpha/2}}$:

$$\widehat{\text{FFSCB}}_{1-\alpha}(Y_{x_{new}}) = \left[x^T_{new}\hat{\beta} \pm \hat{u}^{\star}_{Z_{\alpha/2}}\left(\frac{1}{n}\hat{\sigma}_{\varepsilon}\cdot x^T_{new}\left(\frac{1}{n}\sum_{i=1}^{n}X_iX_i^T\right)^{-1}x_{new}\right)^{1/2}\right]. \tag{3.14}$$

3.2.3.2 Simultaneous Prediction Bands

To construct a $(1-\alpha)\times 100\%$ SPB for $Y_{x_{new}}(t) = x^T_{new}(t)\beta(t) + \varepsilon_{new}(t)$, we need to consider that the error term $\varepsilon_{new}(t)$ in Model (3.1) follows a Student's t process with ν_0 degrees of freedoms. Let $\varepsilon_{new} = \{\varepsilon_{new}(t) : t \in [0,1]\}$ and $Z = \{Z(t) : t \in [0,1]\}$, such that Z follows a Gaussian process as described in Section 3.2.1. As a result of the uniform convergences, Theorem 3.2.1,

$$x^T_{new}\hat{\beta} - x^T_{new}\beta + \varepsilon_{new} \to_D Z\sqrt{\frac{\nu_0}{\chi^2_{\nu_0}}}, \tag{3.15}$$

where $Z\sqrt{\frac{\nu_0}{\chi^2_{\nu_0}}}$ is a scaled Student's t process with ν_0 degrees of freedom and variance covariance function $\sigma_Z(s,t)\cdot\frac{\nu_0}{\nu_0-2}$ (see Section 3.2.1). Corollary 3.3 from Liebl and Reimherr (2023) establishes estimating equations for computing the fast and fair critical value function, $u^{\star}_{t_{\nu_0,\alpha/2}} = \left\{u^{\star}_{t_{\nu_0,\alpha/2}}(t) : t \in [0,1]\right\}$, when the process is t distributed with ν_0 degrees of freedom.

Let $\sigma_Z = \{\sigma_Z(t,t) : t \in [0,1]\}$. Inverting the asymptotic statistic (3.15) leads to the asymptotic fast and fair $(1-\alpha)\times 100\%$ SPB (FFSPB) for $Y_{x_{new}} = x^T_{new}\beta + \varepsilon_{new}$:

$$\text{FFSPB}_{1-\alpha}(Y_{x_{new}}) = \left[x^T_{new}\hat{\beta} \pm u^{\star}_{t_{\nu_0,\alpha/2}}\sigma_Z^{1/2}\right]. \tag{3.16}$$

The SPB in (3.16) is not feasible since we do not know σ_Z, but need to plug-in an empirical version. To additionally account for the variability due to $|x_{new}^T\hat{\beta} - x_{new}^T\beta| \neq 0$ in finite samples, we also take into account the standard deviations of $x_{new}^T\hat{\beta}$. Using an estimator for $u^{\star}_{Z_{\alpha/2}}$, σ_Z, and σ_ε leads to the following feasible version:

$$\widehat{\text{FFSPB}}_{1-\alpha}(Y_{x_{new}}) = \left[x_{new}^T\hat{\beta} \pm \hat{u}^{\star}_{t_{\nu_0,\alpha/2}} \left(\hat{\sigma}_Z + \hat{\sigma}_\varepsilon x_{new}^T \left(\frac{1}{n}\sum_{i=1}^{n} X_i X_i^T \right)^{-1} x_{new} \right)^{1/2} \right]. \quad (3.17)$$

The estimators for σ_Z and σ_ε are the same as those described in Section 3.2.2.

3.3 Simulations

The data for the simulation are generated using the model

$$Y_i(t) = \beta_1(t) + x_i(t)\beta_2(t) + \varepsilon_i(t), \quad i = 1, \ldots, n, \quad (3.18)$$

where $\beta_1(t) = 1$, $\beta_2(t) = \sin(8\pi t) \cdot \exp(-3t) + t$, for all $t \in [0, 1]$, and $x_i(t) \in \{0, 1\}$ for all $t \in [0, 1]$. For the first half of the observations, $i = 1, \ldots, n/2$, we set $x_i(t) = 0$, and for the second half of the observations, $i = n/2 + 1, \ldots, n$, we set $x_i(t) = 1$. To mimic functional data, we discretize all curves using an equidistant grid $t = 0, 1/100, 2/100, \ldots, 1$. We generate the random errors, $\varepsilon_i(t)$, from a Student's t Stochastic process with given degrees of freedom, ν_0. We use two different covariance structures for the Student's t Stochastic process: a Matérn covariance structure, given by

$$C_\theta(t, s) = 0.25^2 \left(2^{1-\nu}/\Gamma(\nu)\right) \left(\sqrt{2\nu}\,|t - s|\right)^{\nu} K_\nu \left(\sqrt{2\nu}\,|t - s|\right), \quad (3.19)$$

where Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and where $\nu \geq 0$ controls the roughness of the sample paths. In our simulations, we set $\nu = 3/2$, which leads to continuously differentiable sample paths. The second covariance structure is a non-stationary

Matérn covariance structure, given by

$$C_\theta(t,s) = 0.25^2 \left(2^{1-\nu_{ts}}/\Gamma(\nu_{ts})\right) \left(\sqrt{2\nu_{ts}}\,|t-s|\right)^{\nu_{ts}} K_{\nu_{ts}} \left(\sqrt{2\nu_{ts}}\,|t-s|\right), \tag{3.20}$$

where $\nu_{ts} = 2 + \sqrt{\max(t,s)} \cdot (1/4 - 2)$. The nonstationary Matérn covariance structure results in sample paths that begin smooth in the sampling domain and later become rough. To generate a random sample of Student's t Stochastic errors, we obtain the singular value decomposition (SVD) of the Matérn covariance. Then, we use the eigenvectors as basis vectors and multiply them by Student's t distributed scores to obtain the random errors. An example of the functional data created can be seen in Figure 3.2.

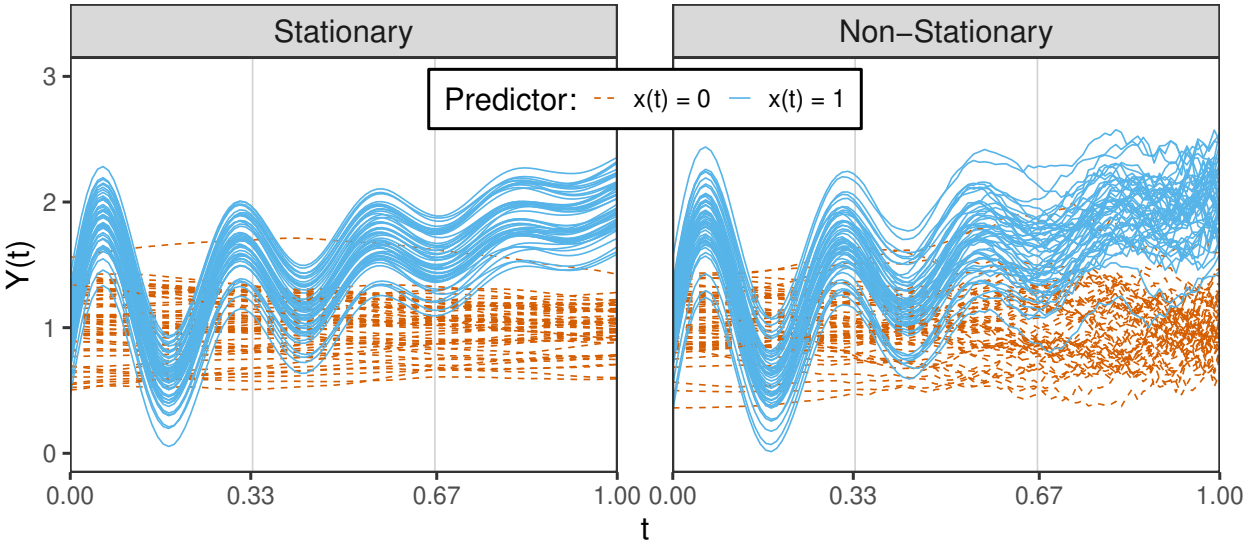


Figure 3.2: A random sample of the data generated by the simulation model (see Section 3.3), with degrees of freedom $\nu_0 = 15$ and sample size $n = 100$. Figure in the left column ("Stationary") shows data generated with a stationary Matérn covariance structure and figure in the right column ("Non-Stationary") shows data generated with a non-stationary Matérn covariance structure. The blue, solid lines represent random observations of the response, $Y(t)$, when the predictor variable is $x(t) = 1$. The orange, dashed lines represent random observations of the response, $Y(t)$, when the predictor variable is $x(t) = 0$.

We consider the scenarios of a small sample size, $n = 30$, and of a large sample size, $n = 100$, a scenario with fat tails, $\nu_0 = 5$, and one with moderately fat tails, $\nu_0 = 15$, a scenario with stationary Matérn covariance for smooth sample paths, and one with non-stationary Matérn covariance for

smooth and rough sample paths. For each of the eight scenarios, we conduct a simulation study with $5,000$ iterations. On each iteration, we generate $n+2$ observations from the simulation model, with observation $i = n + 1$ and $i = n + 2$ designated as hold-out observations for prediction. Specifically, one observation is held out for the prediction of observations with $x_i(t) = 0$ and another is held out for the prediction of observations with $x_i(t) = 1$. Then, we create a 90% prediction band for $x_i(t) = 0$ and $x_i(t) = 1$ using our fast and fair prediction bands for functional concurrent regressions. We create the bands with three equidistant intervals. If any observed point $t = 0, 1/100, 2/100, \ldots, 1$ of the holdout functional observation lies outside the prediction band, the prediction is marked as "out". We calculate the global, simultaneous coverage probability as $1 - \sum_{[0,1]} \frac{\text{out}}{5000}$. If any observed point $t \in [a_l, b_l]$ of the holdout functional observation lies outside the prediction band, the prediction is marked as "out$_l$". We calculate the local coverage probability for each of the three sub-intervals as $1 - \sum_{[a_l,b_l]} \frac{\text{out}_l}{5000}$, for $[a_l, b_l] \in \{[0, 1/3], [1/3, 2/3], [2/3, 1]\}$.

Additionally, we evaluate the prediction bands using the band score, a functional version of the interval score (Gneiting and Raftery, 2007) defined as

$$\max_{t\in[0,1]} \{u(t) - l(t)\} + \frac{\alpha}{2} \max_{t\in[0,1]} (l(t) - y(t)) \, \mathbb{I}_{\{y(t)<l(t)\}} + \frac{\alpha}{2} \max_{t\in[0,1]} (y(t) - u(t)) \, \mathbb{I}_{\{y(t)>u(t)\}}, \quad (3.21)$$

where $u(t)$ denotes the upper bound of the prediction band, $l(t)$ denotes the lower bound of the prediction band, $\mathbb{I}$ is the indicator function, and α denotes the significance level used for the prediction band. If the prediction band $[l(t), u(t)]$ covers the new observation, $y(t)$, then the band score is equivalent to the maximum width of the band. If the prediction band does not cover the new observation, the score is equal to the maximum width of the band plus a measure of how far the new observation was missed by the prediction band.

We compare the performance of our fast and fair band with the conformal inference band for functional data of Diquigiovanni et al. (2022); Fontana et al. (2023). Conformal inference was implemented by using `R` package `conformalInference.fd` (Diquigiovanni et al., 2022; R Core Team, 2021). The results of the simulation are present in Tables 3.1 and 3.2. The prediction bands produced by each method have nominal coverage level, 90%, and fast and fair bands have

more conservative coverage (i.e., $> 90\%$). Despite fast and fair bands having more conservative coverage, the bands have a smaller average band score (see Equation (3.21)) and average maximum band width ($\max_{t\in[0,1]}\{u(t) - l(t)\}$), indicating narrower prediction bands. An example of the 90% prediction bands (for $x_{new}(t) = 1$) generated by using conformal inference and fast and fair is provided in Figure 3.3. The y-intercepts that are plotted in gray lines indicate the three sub-intervals we use when estimating the fast and fair bands.

Table 3.1: Coverage probability, mean max band width, and mean band score of conformal inference prediction bands versus fast and fair prediction bands, when using a stationary Matérn covariance.

	Conformal Inference	Fast and Fair	Conformal Inference	Fast and Fair
	$x_{new}(t) = 0$		$x_{new}(t) = 1$	
	$\nu_0 = 5$			
	$n = 30$			
Coverage	0.94	0.94	0.95	0.95
Mean Max Band Width	2.28	1.64	2.28	1.64
Mean Band Score	2.44	1.76	2.44	1.77
	$n = 100$			
Coverage	0.90	0.95	0.91	0.95
Mean Max Band Width	2.00	1.54	2.00	1.54
Mean Band Score	2.21	1.64	2.23	1.64
	$\nu_0 = 15$			
	$n = 30$			
Coverage	0.95	0.95	0.95	0.95
Mean Max Band Width	2.13	1.45	2.14	1.45
Mean Band Score	2.26	1.55	2.26	1.55
	$n = 100$			
Coverage	0.90	0.95	0.90	0.96
Mean Max Band Width	1.86	1.35	1.86	1.35
Mean Band Score	2.12	1.45	2.10	1.44

Since the fast and fair bands are created with three equidistant intervals, the local coverage levels of each sub-interval should be $\geq 96.67\%$. Regardless of sample size, degrees of freedom, and covariance structure, fast and fair bands have nominal coverage levels for each of the three sub-intervals (Tables 3.3 and 3.4). However, conformal inference does not have a fairness constraint,

Table 3.2: Coverage probability, mean max band width, and mean band score of conformal inference prediction bands versus fast and fair prediction bands, when using a non-stationary Matérn covariance.

	Conformal Inference	Fast and Fair	Conformal Inference	Fast and Fair
	$x_{new}(t) = 0$		$x_{new}(t) = 1$	
	$\nu_0 = 5$			
$n = 30$				
Coverage	0.94	0.95	0.94	0.95
Mean Max Band Width	2.48	2.29	2.48	2.29
Mean Band Score	2.64	2.41	2.66	2.42
$n = 100$				
Coverage	0.90	0.96	0.90	0.95
Mean Max Band Width	2.19	2.07	2.19	2.07
Mean Band Score	2.43	2.15	2.47	2.16
	$\nu_0 = 15$			
$n = 30$				
Coverage	0.94	0.95	0.94	0.95
Mean Max Band Width	2.30	2.02	2.30	2.02
Mean Band Score	2.44	2.13	2.44	2.14
$n = 100$				
Coverage	0.91	0.96	0.91	0.95
Mean Max Band Width	2.02	1.80	2.02	1.80
Mean Band Score	2.23	1.87	2.23	1.88

and the same local coverage levels are not met within each sub-interval. When data are created with a stationary Matérn covariance structure, bands created by conformal inference hold local coverage levels for interval $[1/3, 2/3)$ across all sample sizes, degrees of freedom, and prediction. Local coverage is also met by conformal inference for the interval $[0, 1/3)$ when sample size is $n = 30$ but not when sample size is $n = 100$. When using a non-stationary covariance structure to generate the data, conformal inference bands always hold local coverage levels for intervals $[0, 1/3)$ and $[1/3, 2/3)$. Local coverage levels are never met when using conformal inference for interval $[2/3, 1]$ (see Tables 3.3 and 3.4), which is where the observed functional responses are the most rough (see Figure 3.2).

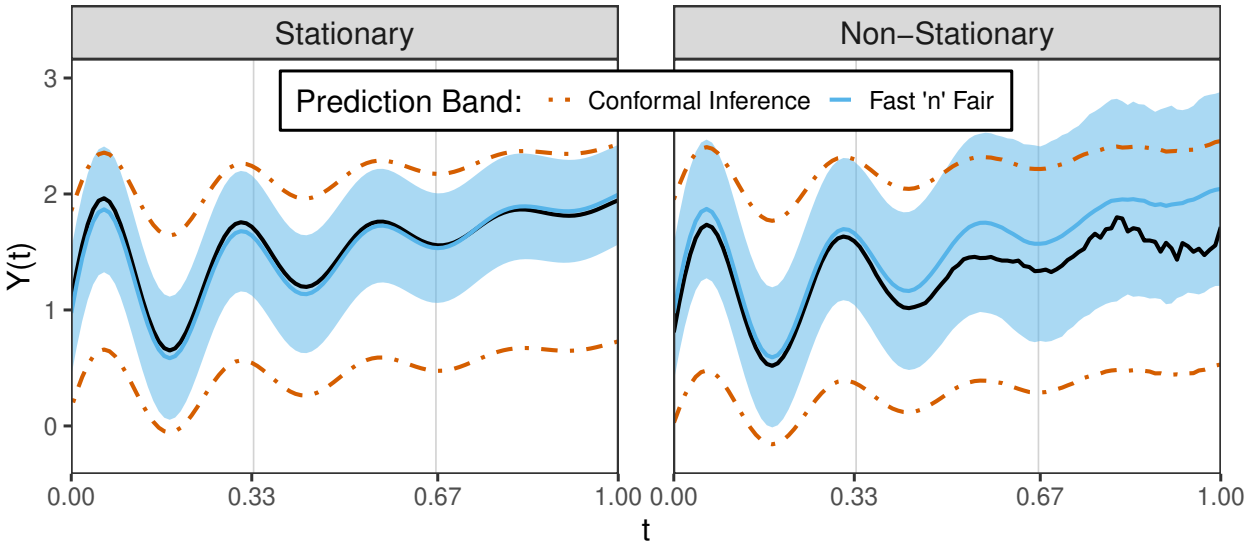


Figure 3.3: One realization of the 90% prediction bands created by using conformal inference (orange, dot-dashed line) and fast and fair (blue, solid shaded region) when predicting $Y(t)$ at $x(t) = 1$. The random sample of data used to estimate the bands was generated with $\nu_0 = 15$ and $n = 100$, using a stationary Matérn covariance structure in the left plot ("Stationary")) and non-stationary Matérn covariance structure in the right plot ("Non-Stationary"). The black, solid line represents the true functional response, $Y(t)$ for which the prediction bands were generated. The blue, solid line is the predicted functional response, $\hat{Y}(t)$, acquired by using fast and fair bands.

3.4 Application: Sprint Start Kinetics

3.4.1 Data Description

Sprint Start Kinetics of Amputee and Non-Amputee Sprinters is a data set originally collected and described by Willwacher et al. (2016). The study sample in Willwacher et al. (2016) includes 154 non-amputee sprinters and 7 amputee sprinters with a wide range of 100-m sprint performance levels (100m personal records (PRs), $9.58s$ - 14.00s for non-amputee sprinters and 11.70 - 12.70s for amputee sprinters). Of the 154 non-amputee sprinters, 103 are males (mean age, 20.8 ± 3.7 years; mean body mass, 74.8 ± 7.5 kg; mean standing height, 1.81 ± 0.06 m), and 51 are females (mean age, 20.0 ± 3.6 years; mean body mass, 60.8 ± 5.6 kg; mean standing height, 1.71 ± 0.06 m). The seven remaining sprinters are male, amputee sprinters (mean age, 28.9 ± 3.27 years; mean body mass, 77.7 ± 6.9 kg; mean standing height, 1.89 ± 0.07 m). The 100 m PR times for all 161 sprinters are acquired prior to data collection. Willwacher et al. (2016) uses a custom-made instrumented starting block to obtain the force data of a sprint start. The analog force signals are

Table 3.3: Local coverage probability over three sub-intervals of conformal inference prediction bands versus fast and fair prediction bands, when using a stationary Matérn covariance.

	Conformal Inference	Fast and Fair	Conformal Inference	Fast and Fair
	$x_{new}(t) = 0$		$x_{new}(t) = 1$	
		$\nu_0 = 5$		
	$n = 30$			
$[0, 1/3)$	0.97	0.97	0.97	0.97
$[1/3, 2/3)$	0.98	0.97	0.99	0.97
$[2/3, 1]$	0.96	0.97	0.96	0.97
	$n = 100$			
$[0, 1/3)$	0.95	0.97	0.95	0.97
$[1/3, 2/3)$	0.98	0.97	0.98	0.97
$[2/3, 1]$	0.94	0.97	0.94	0.97
		$\nu_0 = 15$		
	$n = 30$			
$[0, 1/3)$	0.98	0.97	0.98	0.97
$[1/3, 2/3)$	0.99	0.97	0.99	0.97
$[2/3, 1]$	0.96	0.97	0.96	0.97
	$n = 100$			
$[0, 1/3)$	0.94	0.97	0.95	0.97
$[1/3, 2/3)$	0.97	0.97	0.97	0.97
$[2/3, 1]$	0.93	0.97	0.93	0.97

converted to digital at a sampling rate of $10,000$ Hz. Force signals are recorded on three different axes: x-axis pointed forward along the running surface (horizontal plane), y-axis pointed to the left along the same surface plane, and z-axis pointed vertically upwards.

For purposes of this case study, the vertical force (z-axis) recorded by the front foot plate ("$Y(t)$") is used as the functional response variable in a function on scalar concurrent regression. After $Y(t)$ is converted from analog to digital, the sampling domain is scaled from $0 - 100\%$ of the push-off phase (from starting force to finished force). The resulting data frame is 161 functional observations ($n = 161$), observed at 101 sampling points ($J = 101$). The demographic information of each participant are used as scalar predictor variables. Namely, age, mass, height, and sex are all used as predictor variables. In addition, Willwacher et al. (2016) measures the time from start to finish of vertical force applied ("push-time"). Push-time is also included as a predictor variable.

Table 3.4: Local coverage probability over three sub-intervals of conformal inference prediction bands versus fast and fair prediction bands, when using a non-stationary Matérn covariance.

	Conformal Inference	Fast and Fair	Conformal Inference	Fast and Fair
	$x_{new}(t) = 0$		$x_{new}(t) = 1$	
		$\nu_0 = 5$		
	$n = 30$			
$[0, 1/3)$	0.98	0.97	0.99	0.97
$[1/3, 2/3)$	0.99	0.98	0.99	0.98
$[2/3, 1]$	0.95	0.98	0.95	0.98
	$n = 100$			
$[0, 1/3)$	0.98	0.97	0.97	0.97
$[1/3, 2/3)$	0.99	0.98	0.98	0.98
$[2/3, 1]$	0.92	0.99	0.92	0.98
		$\nu_0 = 15$		
	$n = 30$			
$[0, 1/3)$	0.99	0.97	0.99	0.97
$[1/3, 2/3)$	0.99	0.97	0.99	0.98
$[2/3, 1]$	0.95	0.98	0.95	0.98
	$n = 100$			
$[0, 1/3)$	0.98	0.97	0.98	0.97
$[1/3, 2/3)$	0.99	0.98	0.98	0.97
$[2/3, 1]$	0.93	0.98	0.92	0.98

3.4.2 Data Analysis

We first observe that a sprinter's maximum (max) $Y(t)$ does not align at the same phase of push-off (i.e., max $Y(t)$ occurs at different sampling points for each sprinter). We align all sprinters' functional $Y(t)$'s so their max occurs at the same sampling point. The fall off in $Y(t)$ force at the end of the push-off phase after the max $Y(t)$ force is left out for each sprinter. The remaining functional $Y(t)$ force of each sprinter is then linearly interpolated for $J = 101$ points using `approx` function in `R` (R Core Team, 2021). The realigned functional observations are shown in Figure 3.4.

We then use $Y(t)$ (realigned) as the functional response variable in a function-on-scalar concurrent regression model, with age, height, mass, sex, and push-time as scalar predictor variables. We estimate the model using only the realigned $Y(t)$ force obtained by non-amputee sprinters. After

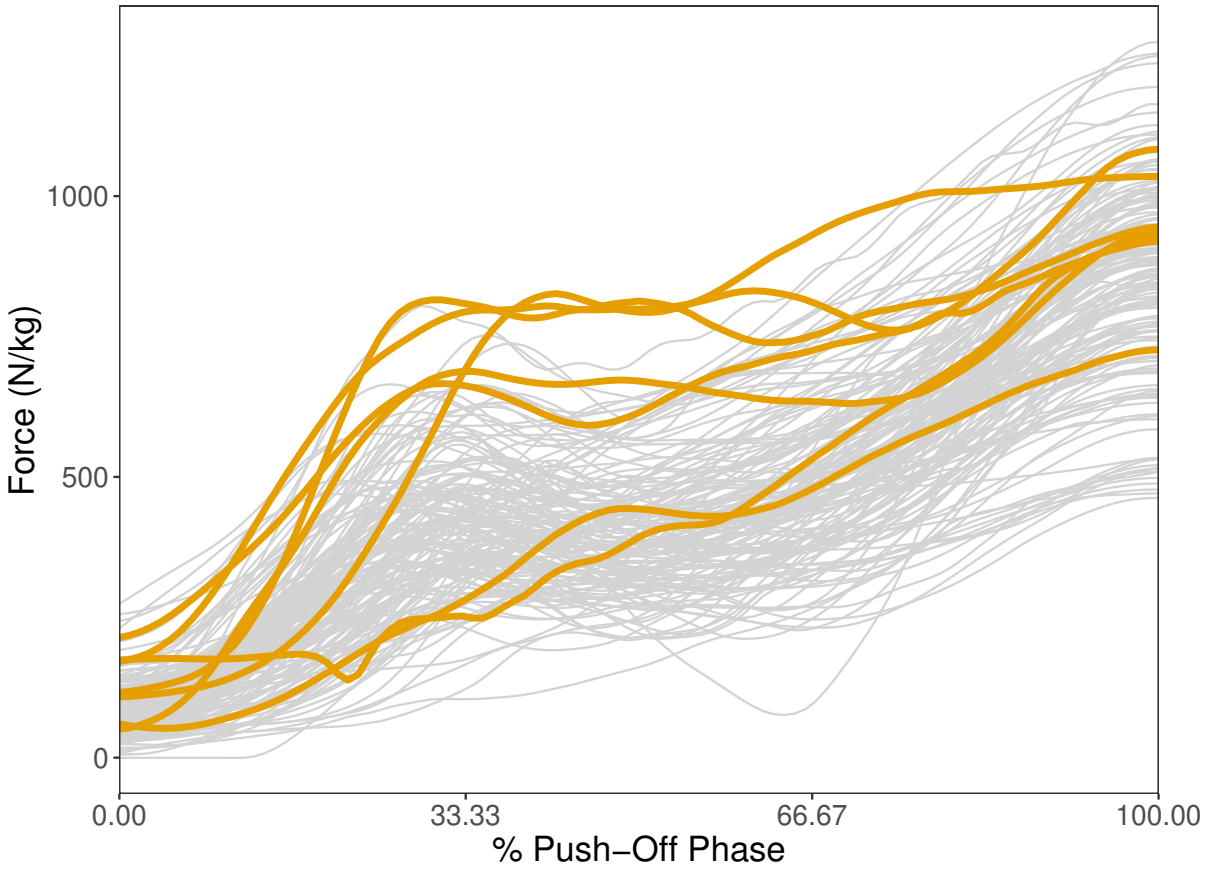


Figure 3.4: A plot of the front vertical force, $Y(t)$, for each sprinter in Sprint Start Kinetics. The front vertical force of each sprinter was realigned with respect to their maximum vertical force during the Push-Off Phase (this corresponds with the right edge of the figure). The orange, wider lines (darker in gray tone) represent the seven amputee sprinters.

the model fit is obtained, the roughness parameter function, τ, and the adaptive critical value function (for three intervals) are estimated ((a) and (b) in Figure 3.5). The estimate $\hat{\tau}$ varies between zero and eight across the sampling domain, motivating the need for a fair band (as described in Liebl and Reimherr (2023)). The estimate $\hat{\tau}$ is correlated with the adaptive critical value function, $u^{\star}_{t_{\nu_0},\alpha/2}$. The more roughness present in the estimation ($\hat{\tau}$), the wider the band should be. To create a wider band, the critical value ($u^{\star}_{t_{\nu_0},\alpha/2}$) needs to be increased. This is reflected in the estimation algorithm of $u^{\star}_{t_{\nu_0},\alpha/2}$, as described in Liebl and Reimherr (2023).

We also check the assumption of wide-tailed, elliptical errors, as is required by our theoretical results to create simultaneous bands. Górecki et al. (2020) investigated multivariate tests of normal-

ity applied to functional data. For a sample size of $n \geq 150$, Górecki et al. (2020) found Mardia's MJB_M test has accurate size and is most powerful when compared to other multivariate tests of normality. Since Sprint Start Kinetics has 154 non-amputee sprinters that we use to fit the model, we apply Mardia's MJB_M test for multivariate normality to the matrix of discretized residuals. The test of Mardia did not find statistical evidence of skewness at the 5% significance level, while the test of Mardia did find statistical evidence of kurtosis. The results of Mardia's test implies that the functional residuals follow a symmetric distribution with wide tails, as required by assumption to create fast and fair simultaneous bands. After checking the assumption of wide-tailed errors and estimating the model, the degrees of freedom, ν_0, were estimated as $\hat{\nu_0} = 7.08$.

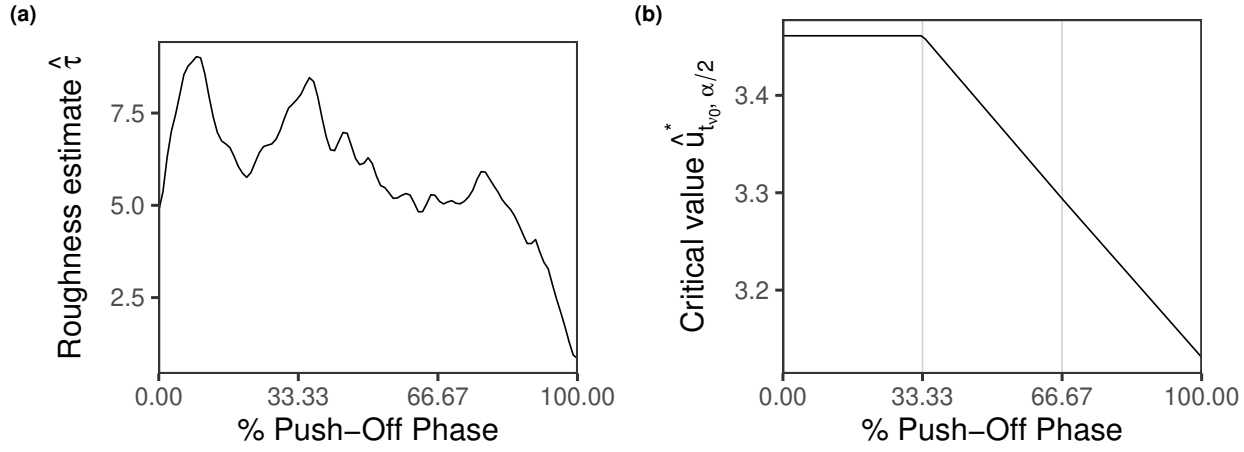


Figure 3.5: (a) Estimated τ roughness parameter for front vertical force, $Y(t)$, realigned for phase shift; and (b) the estimated adaptive critical value function, $u_{t_{\nu_0},\alpha/2}$, for $\alpha = 0.05$.

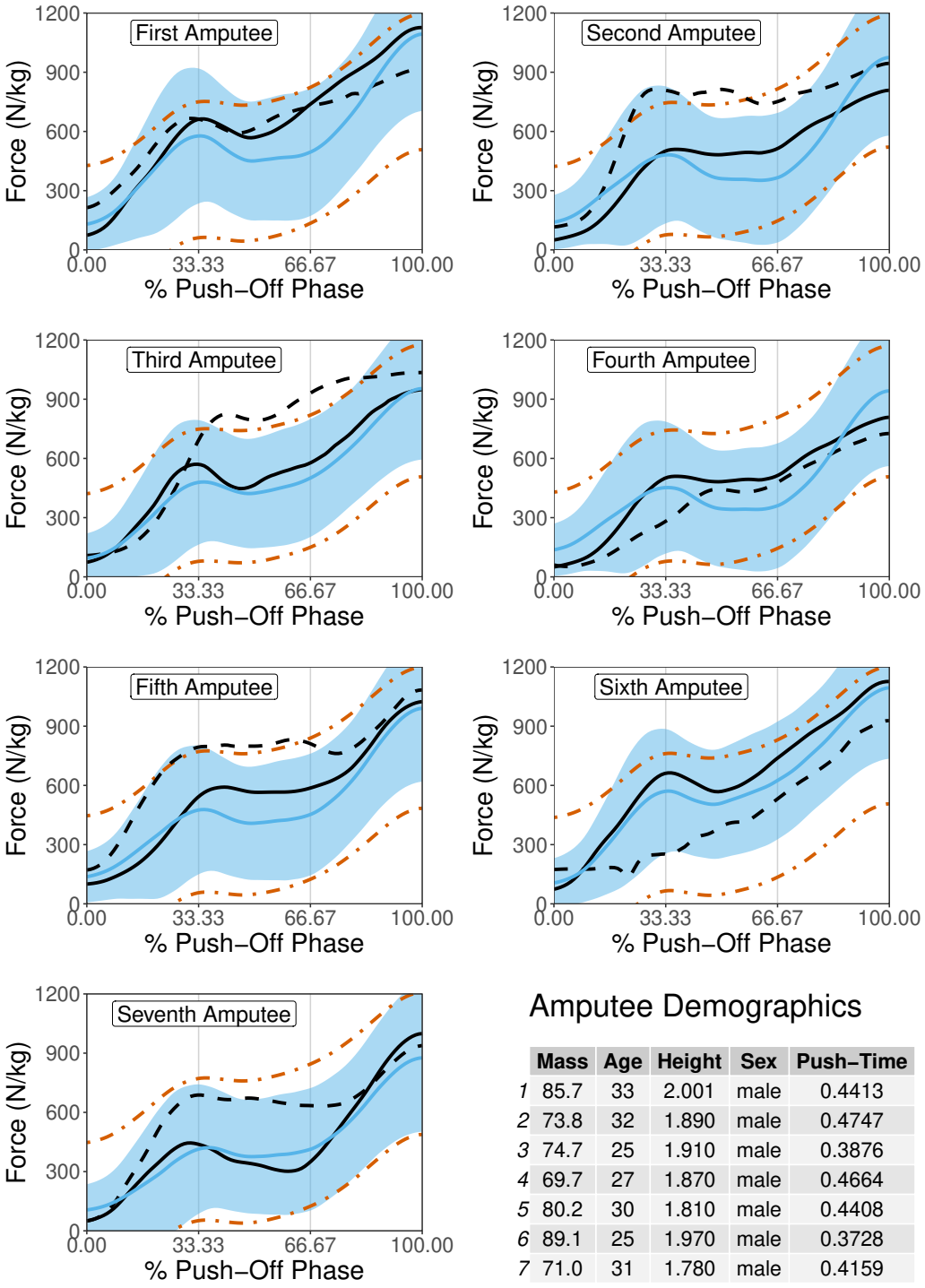
A difficult challenge in professional sprinting is deciding if a prosthesis gives the user any advantage. Previous attempts have been made to show that prostheses do not provide advantage to a sprinter, but were later dismissed (Taboga et al., 2020). We investigate differences between amputee and non-amputee sprinters by estimating 90% FFSPBs for each of the seven amputee sprinters, using equation (3.17) and three intervals to estimate $u^{\star}_{t_{\nu_0},\alpha/2}$. In general, $\hat{u}^{\star}_{t_{\nu_0},\alpha/2}$ is larger at the start of the push-off phase and decreases as the push-off phase progresses (Figure 3.4). Each of the FFSPBs are estimated by using the observed predictor variables of each amputee (Figure

3.6). Since the model is estimated using only the non-amputee sprinters, we expect $Y(t)$ to exceed its relative FFSPB, if the prosthesis truly makes a systematic difference in performance. At a 90% confidence level, if there is no difference between sprinters with and without a prosthesis, we would expect no, or at most one, amputee sprinter to exceed its relative FFSPB.

All except the first and fourth amputee have at least one exceedance from their respective 90% FFSPB. For each amputee sprinter in the data set, we found a similar non-amputee sprinter (in terms of age, height, mass, sex, and push-time) through nearest neighbors. All of these non-amputee counterparts lie entirely within their respective FFSPB, as expected. Thus, observing five exceedances in seven cases (i.e., 71%) is a clear sign of a systematic difference between amputee and non-amputee sprinters. By comparison, only three of the seven (i.e., 43%) amputee sprinters have at least one exceedance from their respective conformal inference SPB. Furthermore, the conformal inference SPBs do not have contextually meaningful lower bounds until 25% (or further) through the Push-off phase, since force can not be negative. The exceedances for the amputee runners all tend to be above the conformal inference SPB, in the middle of the realigned sampling domain (Figure 3.6). Unlike fast and fair simultaneous bands, conformal inference simultaneous bands do not provide suitable statistical methodology and evidence for testing if amputee or non-amputee sprinters have any competitive advantage.

3.5 Conclusion

We create methodology to estimate fast and fair SCBs and SPBs. As of this writing, our fast and fair bands are the only parametric methodology for creating simultaneous bands for the predicted, conditional mean of a response from a functional concurrent regression model with wide-tailed errors. We show that when the functional errors are assumed to follow an elliptical distribution (e.g., Gaussian or Student 's t process), the OLS estimator of the β parameter, estimator of the variance covariance function, and estimator of roughness function, τ, all converge uniformly to their truth (Theorem 3.2.1). As a result, β satisfies the functional Central Limit Theorem (Theorem 3.2.2). A natural extension of this methodology is to develop the bands for functional regression



Amputee Demographics

	Mass	Age	Height	Sex	Push−Time
1	85.7	33	2.001	male	0.4413
2	73.8	32	1.890	male	0.4747
3	74.7	25	1.910	male	0.3876
4	69.7	27	1.870	male	0.4664
5	80.2	30	1.810	male	0.4408
6	89.1	25	1.970	male	0.3728
7	71.0	31	1.780	male	0.4159

Figure 3.6: Estimated 90% simultaneous prediction bands (SPBs) for a non-amputee sprinter's front vertical force created by using conformal inference (dot-dashed, orange) and fast and fair (solid, blue, shaded region). The SPBs were estimated for all seven amputee sprinters and the sprinters' demographics are provided in a table. The solid, blue line represents the predicted vertical front force, $\hat{Y}_{x_{new}}(t)$, for a sprinter with the same demographics as the amputee sprinter (e.g., $X(t) = x_{new}(t)$). The solid, black line represents an observed front vertical force, $Y(t)$, of a non-amputee sprinter with the same, or nearly the same, demographics. The dashed, black line represents the observed front vertical force of the amputee sprinter, $Y_{x_{new}}(t)$.

methods that are not concurrent. A big challenge in functional non-concurrent regression models is that the estimation of the covariance kernel often requires regularization. An additional challenge could be inverting the covariance kernel when $n < T$, because the $T \times T$ covariance matrix would not be full rank. If the inverse of the covariance kernel is required, a pseudo-inverse solution would be necessary.

In the simulation study, we show the fast and fair bands hold nominal coverage levels in all settings. Despite being conservative, our fast and fair methodology produces bands narrower on average than conformal inference. Our fast and fair simultaneous bands also have the advantage of being "fair" across the sampling domain, unlike conformal inference. Thus, the user can be certain that any exceedance of a functional observation from its relative simultaneous confidence/prediction band is evidence of the observation being different from the estimated mean, and not just a spurious point.

Our case study illustrates a potential solution to a long-standing challenge in the world of professional Track and Field. To this point, no suitable methodology had been provided to answer the question, "Do amputee sprinters have any competitive advantage over non-amputee sprinters when using a prosthesis?" We focus on the vertical (z-axis) force trajectory of a sprinter's front foot when pushing off the starting blocks. Of the seven amputees present in the case study, five of them are found to have a vertical force trajectory that exceeds their relative 90% FFSPB (e.g., 71% of the cases showed an exceedance). One can conclude, in this sample, that amputee sprinters with a prosthesis have a systematic difference in front vertical force than non-amputee sprinters. These results can be strengthened by acquiring more observations from other sprinters, whom compete in a variety of different sprint evens. Then, a new predictor variable could be added to the model to denote which sprint event the sprinter usually competes in. The model would then benefit from a larger sample size, while still being possible to make conditional predictions on the amputee sprinters, specific to the event they compete in. Thus, fast and fair simultaneous bands provide a suitable statistical methodology to investigate and test Rule 6.3.4 of the World Athletics.

3.6 Proofs of the Theoretical Results

Let $G_j^{(d)}$ denote the dth derivative of the jth, $j = 1, \ldots, K+1$, element of the $(K+1)$-dimensional vector of random functions,

$$G = (X_1, \ldots, X_K, \varepsilon)^T.$$

In the no derivative case, $G_j^{(0)}$, we usually drop the superscript and write $G_j^{(0)} = G_j$. Define

$$G_{jk}^{(d)} := G_j^{(d)} G_k^{(d)}, \quad j, k = 1, \ldots, K+1.$$

In our proofs, we make use of the following lemma:

Lemma 3.6.1 (Stochastic Lipschitz Continuity)**.** *Under Assumption **A3**, we have for all $d \in \{0, 1\}$ and all $j, k = 1, \ldots, K+1$*

$$\left| G_{jk}^{(d)}(t) - G_{jk}^{(d)}(s) \right| \leq A_{jkd}\, \phi_{jkd}(|t-s|) \quad \textit{for all} \quad t, s \in [0,1],$$

where ϕ_{jkd} is a deterministic nondecreasing continuous function on $[0,1]$ with $\phi_{jkd}(0) = 0$, and where A_{jkd} is a real-valued random variable with $E(|A_{jkd}|^2) < \infty$.

Proof of Lemma 3.6.1

Under Assumption **A3**, $G_{jk}^{(d)}$ is continuously differentiable, and thus, by the Mean Value Theorem,

$$G_{jk}^{(d)}(t) - G_{jk}^{(d)}(s) = G_{jk}^{(d+1)}(\xi)(t-s) \quad \text{for some} \quad \xi \in (s,t)$$

and any $0 \leq s < t \leq 1$. This implies, for all $s, t \in [0,1]$,

$$\left(G_{jk}^{(d)}(t) - G_{jk}^{(d)}(s) \right)^2 \leq \sup_{\xi \in [0,1]} \left(G_{jk}^{(d+1)}(\xi) \right)^2 (t-s)^2$$

Now, if we take the expectation of the left-hand side and apply the uniform 2nd moment assumption in **A3**, it yields for all $s, t \in [0,1]$,

$$\mathbb{E}\left[\left(G_{jk}^{(d)}(t) - G_{jk}^{(d)}(s)\right)^2\right] \leq C_{jkd}(t-s)^2 =: f_{jkd}((t-s)^2), \tag{3.22}$$

where the constant $0 < C_{jkd} < \infty$, and thus also the deterministic function f_{jkd}, only depends on $j, k = 1, \ldots, K+1$, and $d \in \{0,1\}$. Now, observe that for all $j, k = 1, \ldots, K+1$, and $d \in \{0,1\}$

$$\int_0^1 x^{-3/2} f_{jkd}^{1/2}(x)dx = 2C_{jkd}^{1/2} < \infty \tag{3.23}$$

The result of Lemma 3.6.1 follows now directly from (3.22) and (3.23) by applying Theorem 2.3 in Hahn (1977) for the case of $r =$ 2nd moments. □

Proof of Theorem 3.2.1 (a) (Uniform Convergence of the OLS Estimator).

We first consider pointwise convergence for each $t \in [0,1]$ and then expand this to uniform convergence. Using standard arguments, we can express the vector-valued parameter function estimator $\hat{\beta}(t) = (\hat{\beta}_1(t), \ldots, \hat{\beta}_K(t))^T$ as

$$\begin{aligned}\hat{\beta}(t) &= \left(n^{-1}\sum_{i=1}^n X_i(t)X_i^T(t)\right)^{-1} n^{-1}\sum_{i=1}^n X_i(t)Y_i(t) \\ &= \beta(t) + \left(n^{-1}\sum_{i=1}^n X_i(t)X_i^T(t)\right)^{-1} n^{-1}\sum_{i=1}^n X_i(t)\varepsilon_i(t).\end{aligned}$$

By Kolmogorov's strong law of large numbers (SLLN) and the continuous mapping theorem, the second summand converges (a.s.) to the K-dimensional zero vector

$$\left(n^{-1}\sum_{i=1}^n X_i(t)X_i^T(t)\right)^{-1} n^{-1}\sum_{i=1}^n X_i^T(t)\varepsilon_i(t) \xrightarrow{a.s.} 0, \quad n \to \infty,$$

which implies that pointwise for each $t \in [0,1]$

$$\hat{\beta}_j(t) \overset{a.s.}{\to} \beta_j(t), \quad \text{for each} \quad j = 1, \ldots, K. \tag{3.24}$$

Moreover, from $\mathbb{E}[\varepsilon(t)|X(t)] = 0$ and our iid assumption, it follows that the estimator $\hat{\beta}_j(t)$ is unbiased, since pointwise for each $t \in [0,1]$ and every n

$$\mathbb{E}\left[\left(n^{-1}\sum_{i=1}^{n} X_i(t)X_i^T(t)\right)^{-1} n^{-1}\sum_{i=1}^{n} X_i^T(t)\varepsilon_i(t)\right] = 0.$$

Now, we need to expand result (3.24) to uniform convergence across all $t \in [0,1]$. Let $G_{ijk}(t) = G_{ij}(t)G_{ik}(t)$ denote the iid copies of $G_{jk}(t) = G_j(t)G_k(t)$. Lemma 3.6.1 implies for all $j, k = 1, \ldots, K+1$ that

$$|G_{ijk}(t) - G_{ijk}(s)| \leq A_{ijk}\phi_{jk}(|t-s|) \quad \text{for all} \quad t, s \in [0,1]$$

where A_{ijk}, $i = 1, \ldots, n$ is iid as A_{jk} with $E(A_{jk}^2) < \infty$. Define

$$B_{jkn}(t) := n^{-1}\sum_{i=1}^{n} G_{ijk}(t) - E\left[G_{ijk}(t)\right].$$

By SLLN,

$$B_{jkn}(t) \overset{a.s.}{\to} 0 \quad \text{pointwise for each} \quad t \in [0,1]. \tag{3.25}$$

Then,

$$
\begin{aligned}
&|B_{jkn}(t) - B_{jkn}(s)| \\
&= \left| \left(n^{-1} \sum_{i=1}^{n} G_{ijk}(t) - E\left[G_{ijk}(t)\right] \right) - \left(n^{-1} \sum_{i=1}^{n} G_{ijk}(s) - E\left[G_{ijk}(s)\right] \right) \right| \\
&\leq n^{-1} \sum_{i=1}^{n} |G_{ijk}(t) - G_{ijk}(s)| + E\left[|G_{ijk}(t) - G_{ijk}(s)|\right] \quad \text{(Triangle Inequality)} \\
&\leq n^{-1} \sum_{i=1}^{n} A_{ijk} \phi_{jk}(|t-s|) + E\left(A_{ijk}\right) \phi_{jk}(|t-s|) \quad \text{(Lemma 3.6.1)} \\
&\leq \left(n^{-1} \sum_{i=1}^{n} A_{ijk} + E\left(A_{ijk}\right) \right) \phi_{jk}(|t-s|), \qquad (3.26)
\end{aligned}
$$

where by SLLN and Lemma 3.6.1 ($E\left(A_{ijk}\right) < \infty$),

$$
\left(n^{-1} \sum_{i=1}^{n} A_{ijk} + E\left(A_{ijk}\right) \right) \xrightarrow{a.s.} 2E\left(A_{ijk}\right) < \infty.
$$

Result (3.26) implies, by Theorem 22.8 from Davidson (2021), that B_{jkn} is strongly stochastically equicontinuous for every $j, k \in \{1, \ldots, K\}$. This, together with the pointwise consistency (3.25) implies, by Theorem 22.10 from Davidson (2021), that

$$
\sup_{t \in [0,1]} |B_{jkn}(t)| \xrightarrow{a.s.} 0, \quad \text{for all } j, k \in \{1, \ldots, K+1\}.
$$

The latter elementwise result implies the following matrix-valued and vector-valued uniform convergence results,

$$
\begin{aligned}
\sup_{t \in [0,1]} \left| n^{-1} \sum_{i=1}^{n} X_i(t) X_i^T(t) - E\left(X_i(t) X_i^T(t)\right) \right| &\xrightarrow{a.s.} \underset{(K \times K)}{0} \\
\sup_{t \in [0,1]} \left| n^{-1} \sum_{i=1}^{n} X_i(t) \varepsilon_i(t) - E\left(X_i(t) \varepsilon_i(t)\right) \right| &\xrightarrow{a.s.} \underset{(K \times 1)}{0},
\end{aligned}
$$

where $E\left(X_i(t)\varepsilon_i(t)\right) = E\left(X_i(t) E(\varepsilon_i(t)|X_i(t))\right) = 0$ for all $t \in [0,1]$. Under our assumptions, $E\left[X_i(t) X_i^T(t)\right]$ is invertible. Thus, by the functional version of the uniform continuous mapping

theorem, we also have that

$$\sup_{t\in[0,1]}\left|\left(n^{-1}\sum_{i=1}^{n}X_i(t)X_i^T(t)\right)^{-1}n^{-1}\sum_{i=1}^{n}X_i(t)\varepsilon_i(t)\right|\overset{a.s.}{\to}\underset{(K\times 1)}{0}$$

which implies that

$$\sup_{t\in[0,1]}\left|\hat{\beta}(t)-\beta(t)\right|\overset{a.s.}{\to}\underset{(K\times 1)}{0}.$$

Then, by a similar argument, one can show

$$\sup_{t\in[0,1]}\left|\hat{\beta}^{(1)}(t)-\beta^{(1)}(t)\right|\overset{a.s.}{\to}\underset{(K\times 1)}{0},$$

concluding the proof of Theorem 3.2.1 (a). □

Proof of Theorem 3.2.1 (b) (Uniform Convergence of the Variance Estimator).

The estimators $\hat{\sigma}_\varepsilon^{ml}$, $\hat{\sigma}_\varepsilon^{ub}$, and $\hat{\sigma}_\varepsilon^{mm}$ only differ with respect to the scaling parameters $\frac{1}{n}$, $\frac{1}{n-K}$, and $\frac{\nu_0-4}{(\nu_0-2)(n-K+2)}$ and thus are asymptotically equivalent. Therefore, it suffices to consider the ML estimator $\hat{\sigma}_\varepsilon^{ml}\equiv\hat{\sigma}_\varepsilon$. Note that the residuals, $e_i(t)$, can be defined as

$$\begin{aligned}
e_i(t) &= Y_i(t)-X_i^T(t)\hat{\beta}(t)\\
&= Y_i(t)-X_i^T(t)\beta(t)-X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right)\\
&= \varepsilon_i(t)-X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right),\\
e_i^2(t) &= \varepsilon_i^2(t)-2\left(\hat{\beta}(t)-\beta(t)\right)^T X_i(t)\varepsilon_i(t)\\
&\quad+\left(\hat{\beta}(t)-\beta(t)\right)^T X_i(t)X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right),
\end{aligned}$$

and the ML estimator of the variance is

$$\begin{aligned}
\hat{\sigma}_{\varepsilon}(t,t) &= \frac{1}{n}\sum_{i=1}^{n}\left(e_i(t)\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i(t) - X_i^T(t)\hat{\beta}(t)\right)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left(\varepsilon_i^2(t) - 2\left(\hat{\beta}(t)-\beta(t)\right)^T X_i(t)\varepsilon_i(t) + \right. \\
&\qquad \left. \left(\hat{\beta}(t)-\beta(t)\right)^T X_i(t)X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2(t) - 2\left(\hat{\beta}(t)-\beta(t)\right)^T \frac{1}{n}\sum_{i=1}^{n} X_i(t)\varepsilon_i(t) \\
&\quad + \left(\hat{\beta}(t)-\beta(t)\right)^T \frac{1}{n}\sum_{i=1}^{n}\left(X_i(t)X_i^T(t)\right)\left(\hat{\beta}(t)-\beta(t)\right).
\end{aligned}$$

By the SLLN and continuous mapping theorem, the second and third summand converge (a.s.) to zero. This implies that pointwise for each $t \in [0,1]$,

$$\hat{\sigma}_{\varepsilon}(t,t) \overset{a.s.}{\to} \sigma_{\varepsilon}(t,t). \tag{3.27}$$

Now, we need to expand result (3.27) to uniform convergence across all $t \in [0,1]$. The following uniform convergence results were proven in Theorem 3.2.1 (a):

$$\begin{aligned}
&\sup_{t\in[0,1]}\left|\hat{\beta}(t)-\beta(t)\right| \overset{a.s.}{\to} \underset{(K\times 1)}{0}, \\
&\sup_{t\in[0,1]}\left|n^{-1}\sum_{i=1}^{n} X_i(t)X_i^T(t) - E\left(X_i(t)X_i^T(t)\right)\right| \overset{a.s.}{\to} \underset{(K\times K)}{0}, \text{ and} \\
&\sup_{t\in[0,1]}\left|n^{-1}\sum_{i=1}^{n} X_i(t)\varepsilon_i(t) - E\left(X_i(t)\varepsilon_i(t)\right)\right| \overset{a.s.}{\to} \underset{(K\times 1)}{0}.
\end{aligned}$$

Then, by functional continuous mapping theorem,

$$\sup_{t\in[0,1]}\left|\left(\hat{\beta}(t)-\beta(t)\right)^T\frac{1}{n}\sum_{i=1}^{n}X_i(t)\varepsilon_i(t)\right|\overset{a.s.}{\to}0\quad\text{and}$$
$$\sup_{t\in[0,1]}\left|\left(\hat{\beta}(t)-\beta(t)\right)^T\frac{1}{n}\sum_{i=1}^{n}\left(X_i(t)X_i^T(t)\right)\left(\hat{\beta}(t)-\beta(t)\right)\right|\overset{a.s.}{\to}0,$$

which implies that

$$\sup_{t\in[0,1]}|\hat{\sigma}_\varepsilon(t,t)-\sigma_\varepsilon(t,t)|\overset{a.s.}{\to}0. \tag{3.28}$$

Next, we need to show the covariance estimator converges uniformly. Let, without loss of generality, $s<t\in[0,1]$, then

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}e_i(s)e_i(t)\\
&=\frac{1}{n}\sum_{i=1}^{n}\left(\varepsilon_i(s)-X_i^T(s)\left(\hat{\beta}(s)-\beta(s)\right)\right)\left(\varepsilon_i(t)-X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right)\right)\\
&=\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i(s)\varepsilon_i(t)\\
&\quad-\left(\hat{\beta}(s)-\beta(s)\right)^T\frac{1}{n}\sum_{i=1}^{n}X_i(s)\varepsilon_i(t)\\
&\quad-\left(\hat{\beta}(t)-\beta(t)\right)^T\frac{1}{n}\sum_{i=1}^{n}X_i(t)\varepsilon_i(s)\\
&\quad+\frac{1}{n}\sum_{i=1}^{n}X_i^T(s)\left(\hat{\beta}(s)-\beta(s)\right)X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right).
\end{aligned}$$

By the SLLN and continuous mapping theorem, the second, third, and fourth summand converge (a.s.) to zero. This implies that pointwise for each $s,t\in[0,1]$,

$$\hat{\sigma}_\varepsilon(s,t)\overset{a.s.}{\to}\sigma_\varepsilon(s,t). \tag{3.29}$$

Now, we need to establish uniform convergence of the covariance estimator. Observe that

$$X_i^T(s)\left(\hat{\beta}(s)-\beta(s)\right)\varepsilon_i(t)=\varepsilon_i(t)X_i^T(s)\left(\hat{\beta}(s)-\beta(s)\right) \quad \text{and}$$
$$X_i^T(s)\left(\hat{\beta}(s)-\beta(s)\right)X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right)=\left(\hat{\beta}(s)-\beta(s)\right)^T X_i(s)X_i^T(t)\left(\hat{\beta}(t)-\beta(t)\right).$$

It was already proven in Theorem 3.2.1 (a) that

$$\sup_{t\in[0,1]}\left|\hat{\beta}(t)-\beta(t)\right|\overset{a.s.}{\to}\underset{(K\times 1)}{0}.$$

Thus, it suffices to show that $\varepsilon_i(s)X_i^T(t)$ and $X_i(s)X_i^T(t)$ converge uniformly.

Let $G_{ijk}(s,t)=G_{ij}(s)G_{ik}(t)$ for some predictor, j, such that

$$n^{-1}\sum_{i=1}^{n}X_{ij}(s)X_{ik}^T(t)=n^{-1}\sum_{i=1}^{n}G_{ij}(s)G_{ik}^T(t)=n^{-1}\sum_{i=1}^{n}G_{ijk}(s,t).$$

Note that by SLLN,

$$n^{-1}\sum_{i=1}^{n}G_{ijk}(s,t)\overset{a.s.}{\to}E\left[G_{ijk}(s,t)\right], \tag{3.30}$$

pointwise for each j,k,s, and t. Next, define

$$B_{jkn}(s,t):=n^{-1}\sum_{i=1}^{n}G_{ijk}(s,t)-E\left[G_{ijk}(s,t)\right].$$

We need to show that B_{jkn} is strongly stochastically equicontinuous for every $j \in \{1, \ldots, K+1\}$. Let $s, t, u, v \in [0,1]$, such that $s < u$ and $t < v$, without loss of generality. Then,

$$\begin{aligned}
|B_{jkn}(s,t) - B_{jkn}(u,v)| &= \Big| \left(n^{-1} \sum_{i=1}^{n} G_{ijk}(s,t) - E\left[G_{ijk}(s,t)\right] \right) \\
&\quad - \left(n^{-1} \sum_{i=1}^{n} G_{ijk}(u,v) - E\left[G_{ijk}(u,v)\right] \right) \Big| \\
&= \Big| \left(n^{-1} \sum_{i=1}^{n} G_{ij}(s) G_{ik}(t) - n^{-1} \sum_{i=1}^{n} G_{ij}(u) G_{ik}(v) \right) \\
&\quad + \left(E\left[G_{ij}(s) G_{ik}(t)\right] - E\left[G_{ij}(u) G_{ik}(v)\right] \right) \Big| \\
&\leq \left| n^{-1} \sum_{i=1}^{n} \left(G_{ij}(s) G_{ik}(t) - G_{ij}(u) G_{ik}(v) \right) \right| \\
&\quad + \left| \left(E\left[G_{ij}(s) G_{ik}(t) - G_{ij}(u) G_{ik}(v)\right] \right) \right|,
\end{aligned}$$

by the Triangle Inequality. Now, focusing on the first quantity:

$$\begin{aligned}
&\left| n^{-1} \sum_{i=1}^{n} \left(G_{ij}(s) G_{ik}(t) - G_{ij}(u) G_{ik}(v) \right) \right| \\
&\qquad \leq \left| n^{-1} \sum_{i=1}^{n} \left(\left(G_{ij}(s) - G_{ij}(u) \right) G_{ik}(t) - G_{ij}(u) \left(G_{ik}(t) - G_{ik}(v) \right) \right) \right| \\
&\qquad \leq n^{-1} \sum_{i=1}^{n} \left| \left(G_{ij}(s) - G_{ij}(u) \right) \right| \left| G_{ik}(t) \right| + \left| G_{ij}(u) \right| \left| \left(G_{ik}(t) - G_{ik}(v) \right) \right|.
\end{aligned}$$

Again, we focus on the first part of the summation, since the second part is analogous to the first. Using the Cauchy-Schwarz inequality,

$$
\begin{aligned}
& n^{-1} \sum_{i=1}^{n} \left| (G_{ij}(s) - G_{ij}(u)) \right| |G_{ik}(t)| \\
& \leq \left(n^{-1} \sum_{i=1}^{n} \left| (G_{ij}(s) - G_{ij}(u)) \right|^2 \right)^{1/2} \left(n^{-1} \sum_{i=1}^{n} |G_{ik}(t)|^2 \right)^{1/2} \\
& \leq \left(n^{-1} \sum_{i=1}^{n} \left| (G_{ij}(s) - G_{ij}(u)) \right|^2 \right)^{1/2} \left(n^{-1} \sum_{i=1}^{n} \sup_t \left(G_{ik}(t) \right)^2 \right)^{1/2} \\
& \leq \left(n^{-1} \sum_{i=1}^{n} \left[A_{ij} \phi_j \left(|s-u| \right) \right]^2 \right)^{1/2} \left(n^{-1} \sum_{i=1}^{n} \sup_t \left(G_{ik}(t) \right)^2 \right)^{1/2},
\end{aligned}
$$

where the last inequality holds by Lemma 3.6.1 and $A_{ij}, i = 1, \ldots, n$ is iid as A_j with $E\left(A_j^2\right) < \infty$. Then, by the SLLN and functional continuous mapping theorem,

$$
n^{-1} \sum_{i=1}^{n} \left| (G_{ij}(s) - G_{ij}(u)) \right| |G_{ik}(t)| \xrightarrow{a.s.} E\left(A_j^2\right)^{1/2} \phi_j \left(|s-u| \right) \cdot E \left[\sup_t G_k(t)^2 \right]^{1/2}, \tag{3.31}
$$

which is finite, by assumption A3. Similary,

$$
n^{-1} \sum_{i=1}^{n} |G_{ij}(u)| \left| (G_{ik}(t) - G_{ik}(v)) \right| \xrightarrow{a.s.} E\left(A_k^2\right)^{1/2} \phi_k \left(|t-v| \right) \cdot E \left[\sup_u G_j(u)^2 \right]^{1/2}, \tag{3.32}
$$

which is finite, by assumption A3. Therefore,

$$
\begin{aligned}
& \left| n^{-1} \sum_{i=1}^{n} \left(G_{ij}(s) G_{ik}(t) - G_{ij}(u) G_{ik}(v) \right) \right| \\
& \xrightarrow{a.s.} E\left(A_j^2\right)^{1/2} \cdot E \left[\sup_t G_k(t)^2 \right]^{1/2} \cdot \phi_j \left(|s-u| \right) + E\left(A_k^2\right)^{1/2} \cdot E \left[\sup_u G_j(u)^2 \right]^{1/2} \cdot \phi_k \left(|t-v| \right),
\end{aligned} \tag{3.33}
$$

which is also finite. By a similar argument,

$$
\begin{aligned}
&|(E\left[G_{ij}(s)G_{ik}(t) - G_{ij}(u)G_{ik}(v)\right])| \\
&\overset{a.s.}{\to} E\left(A_j^2\right)^{1/2} \cdot E\left[\sup_t G_k(t)^2\right]^{1/2} \cdot \phi_j\left(|s-u|\right) + E\left(A_k^2\right)^{1/2} \cdot E\left[\sup_u G_j(u)^2\right]^{1/2} \cdot \phi_k\left(|t-v|\right).
\end{aligned} \tag{3.34}
$$

Therefore,

$$
\begin{aligned}
|B_{jkn}(s,t) - B_{jkn}(u,v)| \overset{a.s.}{\to} 2\Bigg(& E\left(A_j^2\right)^{1/2} \cdot E\left[\sup_t G_k(t)^2\right]^{1/2} \cdot \phi_j\left(|s-u|\right) + \\
& E\left(A_k^2\right)^{1/2} \cdot E\left[\sup_u G_j(u)^2\right]^{1/2} \cdot \phi_k\left(|t-v|\right)\Bigg) \\
< \infty.
\end{aligned} \tag{3.35}
$$

Result (3.35) implies, by Theorem 22.8 from Davidson (2021), that B_{jkn} is strongly stochastically equicontinuous for every $j, k \in \{1, \ldots, K+1\}$. This, together with the pointwise consistency (3.30) implies, by Theorem 22.10 from Davidson (2021), that

$$
\sup_{s,t\in[0,1]} |B_{jkn}(s,t)| \overset{a.s.}{\to} 0, \quad \text{for all } j,k \in \{1,\ldots,K+1\}. \tag{3.36}
$$

The latter elementwise result implies the following matrix-valued and vector-valued uniform convergence results,

$$
\sup_{s,t\in[0,1]} \left| n^{-1} \sum_{i=1}^{n} X_i(s)X_i^T(t) - E\left[X_i(s)X_i^T(t)\right] \right| \overset{a.s.}{\to} \underset{(K\times K)}{0}. \tag{3.37}
$$

$$
\sup_{s,t\in[0,1]} \left| n^{-1} \sum_{i=1}^{n} \varepsilon_i(s)X_i^T(t) - E\left[\varepsilon_i(s)X_i^T(t)\right] \right| \overset{a.s.}{\to} \underset{(1\times K)}{0}. \tag{3.38}
$$

Furthermore, by the functional continuous mapping theorem,

$$
\sup_{s,t\in[0,1]} \left| n^{-1} \sum_{i=1}^{n} \varepsilon_i(t)X_i^T(s) \left(\hat{\beta}(s) - \beta(s)\right) \right| \overset{a.s.}{\to} 0, \quad \text{and}
$$

$$\sup_{s,t\in[0,1]}\left|n^{-1}\sum_{i=1}^{n}\left(\hat{\beta}(s)-\beta(s)\right)^{T}X_i(s)X_i^{T}(t)\left(\hat{\beta}(t)-\beta(t)\right)\right|\overset{a.s.}{\to}0.$$

The results of uniform convergence, combined with pointwise convergence (3.29), implies that

$$\sup_{t,s\in[0,1]}|\hat{\sigma}_{\varepsilon}(s,t)-\sigma_{\varepsilon}(s,t)|\overset{a.s.}{\to}0. \tag{3.39}$$

Then, by a similar argument, one can show

$$\sup_{t,s\in[0,1]}\left|\hat{\sigma}_{\varepsilon}^{(1,1)}(s,t)-\sigma_{\varepsilon}^{(1,1)}(s,t)\right|\overset{a.s.}{\to}0. \tag{3.40}$$

concluding the proof of Theorem 3.2.1 (b).

□

Proof of Theorem 3.2.1 (c) (Uniform Convergence of the τ Estimator).

Applying functional continuous mapping theorem together with assumption A3 immediately gives the result.

□

Proof for Theorem 3.2.2 (Functional Central Limit Theorem for Functional Concurrent Regression Parameter Estimation)

Given that $\hat{\beta}(t)$ is the OLS estimator of $\beta(t)$, we can write

$$\sqrt{n}\left(\hat{\beta}(t)-\beta(t)\right)=\left(n^{-1}\sum_{i=1}^{n}X_i(t)X_i^T(t)\right)^{-1}\left(\sqrt{n}\left[n^{-1}\sum_{i=1}^{n}X_i(t)\varepsilon_i(t)\right]\right).$$

As was shown in Theorem 2.1(a), we also know that

$$\sup_{t\in[0,1]}\left|n^{-1}\sum_{i=1}^{n}X_i(t)X_i^T(t)-E\left(X_i(t)X_i^T(t)\right)\right|\overset{a.s.}{\to}\underset{(K\times K)}{0}.$$

We need to show that $(\sqrt{n}\,[n^{-1}\sum_{i=1}^{n}X_i(t)\varepsilon_i(t)])$ converges in distribution to a Gaussian process. First, let $\lambda\in\mathbb{R}^K$. Then,

$$\left(\sqrt{n}\left[n^{-1}\sum_{i=1}^{n}\lambda^TX_i(t)\varepsilon_i(t)\right]\right)\in\mathbb{R},$$

with mean zero. By Lemma 3.1, $X_i(t)\varepsilon_i(t)$ is stochastic lipschitz continuous, and meets the integrability assumption (equation 3.23) of Theorem 2.5 in Hahn (1977) for $r=2$. Therefore, $\sum_{i=1}^{n}X_i(t)\varepsilon_i(t)$ satisfies the CLT in $\mathcal{C}^1[0,1]$, since it is also mean zero. Specifically, we have

$$\left(\sqrt{n}\left[n^{-1}\sum_{i=1}^{n}\lambda^TX_i(t)\varepsilon_i(t)\right]\right)\to\mathcal{G}_p\left(0,\lambda^Tc_{X\varepsilon}(s,t)\lambda\right),\tag{3.41}$$

where $\mathcal{G}_p$ is a mean-zero Gaussian process and variance term $c_{X\varepsilon}(s,t)=E\left[\left(X_i(s)\varepsilon_i(s)\right)\left(X_i(t)\varepsilon_i(t)\right)^T\right]$. By the Cramer-Wold device, we now have that

$$\left(\sqrt{n}\left[n^{-1}\sum_{i=1}^{n}X_i(t)\varepsilon_i(t)\right]\right)\to\mathcal{G}_p\left(0,c_{X\varepsilon}(s,t)\right).\tag{3.42}$$

Then, by the functional version of Slutsky's Theorem,

$$\sqrt{n}\left(\hat{\beta}(t)-\beta(t)\right) \to \mathcal{G}_p\left(0, E\left[X_i(s)X_i(t)\right]^{-1} c_{X\varepsilon}(s,t) E\left[X_i(s)X_i(t)\right]^{-1}\right). \tag{3.43}$$

This allows for heteroscedastic errors, where $c_{X\varepsilon}(s,t)$ is a function of X_i. If homoscedastic errors are assumed, then

$$c_{X\varepsilon}(s,t) = E\left[X_i(s)X_i(t)^T\right] c_\varepsilon(s,t),$$

where $c_\varepsilon(s,t) = E\left[\varepsilon_i(s)\varepsilon_i(t)\right]$. Substituting into the covariance for equation 3.43, for homoscedastic errors, we have

$$\sqrt{n}\left(\hat{\beta}(t)-\beta(t)\right) \to \mathcal{G}_p\left(0, c_\varepsilon(s,t) E\left[X_i(s)X_i(t)\right]^{-1}\right). \tag{3.44}$$

□

Chapter 4

Identifying Influential Observations in a Functional Concurrent Regression Model

A common way to investigate influential observations in non-functional ordinary linear regression is to estimate a measure of influence and leverage, such as standardized difference in fits ($DFFITS$) or externally studentized residuals ($t_{i(i)}$). When considering functional concurrent regression, both $DFFITS$ and $t_{i(i)}$ can be estimated pointwise at each sampling point, t, to obtain their functional estimations, $DFFITS(t)$ and $t_{i(i)}(t)$. In this paper, we show that $t_{i(i)}$ follows a Student's t process with $n-K-1$ degrees of freedom, where K is the number of predictor variables (including the intercept), when the functional errors ($\varepsilon(t)$) follow a Gaussian process. Under the same assumption, we also show that $DFFITS(t)$ is distributed as a scaled Student's t process. Then, we propose using a multivariate Student's t distributional quantile for identifying influential functional observations with $DFFITS(t)$. Our methodology ("Theoretical") is compared against a competing methodology that uses a parametric bootstrapping technique ("Bootstrapped") for estimating the null distribution of the mean absolute value of $DFFITS(t)$. In the simulation and case study results, we find that the Theoretical method greatly reduces the computation time, without much loss in accuracy as measured by accuracy (ACC), precision (PPV), and Matthew's Correlation Coefficient (MCC), than the Bootstrapped method. Furthermore, the average sensitivity of the Theoretical method is higher in all scenarios than the Bootstrapped method.

4.1 Introduction

An influential observation in a regression model can be defined as an observation that is outlying in its relationship between the response variable and predictor variable(s) (Belsley et al., 2005). Specifically, an influential observation is one that heavily influences the fit of the estimated regression model. If the regression model is estimated with the influential point, the estimated

coefficients of the model will be different than if the regression model is estimated without the influential point. In some examples, an estimated coefficient could swap signs (e.g. "+" or "-") depending on the presence of the influential point. Methods to asses and identify the influentialness of an observation in non-functional ordinary linear regression have been well studied and developed over the years. A common way to investigate influential observations in non-functional ordinary linear regression is to estimate a measure of influence and leverage, such as standardized difference in fits ($DFFITS$), standardized difference in beta ($DFBETAS$), Cook's distance (D), and the "hat" matrix (Cook, 1977; Welsch and Kuh, 1977).

By comparison, methods for assessing influentialness in functional linear regression are still being developed. Shen and Xu (2007) developed model selection methods and functional influence measures for a function on scalar linear regression model. Specifically, Shen and Xu (2007) used the L^2 norm of the residuals to compute the studentized residuals, externally studentized residuals, and Cook's distance. They also developed a functional F-test for model comparison when assuming Gaussian errors. Febrero-Bande et al. (2010) considered the case of a scalar on function linear regression and compute a functional Cook's measure for prediction, a functional Cook's measure for estimation, and functional Pena's measure for prediction. Febrero-Bande et al. (2010) also implemented a bootstrap technique to estimate the null distribution of each functional influential measure. Influential observations were identified by comparing the influential estimates to a quantile (e.g., $95th$) from their respective null distribution (Febrero-Bande et al., 2010).

Chiou and Müller (2007) considered the case of function on function linear regression with a single predictor variable and treated the hat matrix as a function (e.g., $h_{ii} = \{h_{ii}(t) : t \in [0, 1]\}$. They also implemented functional principal components (FPC) analysis and study the FPC scores as proxies for the residual processes (Chiou and Müller, 2007). Similarly, Chen et al. (2014) considered the function on function concurrent linear regression model. After estimating the unknown coefficient functions with a spline basis, Chen et al. (2014) computed a functional Cook's distance and functional likelihood distance (Cook and Weisberg, 1982). However, rather than assessing if an entire functional observation is influential, Chen et al. (2014) are more concerned about iden-

tifying single influential points among the sampling domain. Pittman (2022) builds on the work done by Chen et al. (2014) and focuses on the influentialness of an entire functional observation, rather than a single observed point of a functional observation. Pittman (2022) considers the case of function on function concurrent linear regression from which they calculate a functional $DFFITS$, functional Cook's distance, and functional $DFBETAS$. The average value of each functional measure is obtained for each functional observation and bootstrapping is used to estimate the null distribution of the average of each functional influential measure. The methodology of Pittman (2022) also relies on functional concurrent regression implemented through regularization. Namely, Pittman (2022) uses a B-spline basis for smoothing the data and parameter estimates, including the $DFFITS$. Yet, regularization and/or smoothing of the functional data prior to identifying influential observations has been shown to alter the identification of influential observations and vice versa (Edith U Umeh and Chinyere I Ojukwu, 2019; Hellton et al., 2023). That is, the presence of influential observations in the random sample when implementing regularization can alter the choice of the penalty parameter and/or the "optimal" number of bases.

In this paper, the function on function concurrent linear regression model, without regularization, is considered (Section 4.2). The functional, externally, studentized residuals and functional $DFFITS$ are estimated and their distributional properties are established (Section 4.2.2). Rather than averaging over functional influential measures and across the sampling domain, the entire functional influential measure is considered. A rule for identifying influential functional observations in a functional concurrent regression is determined using the distributional properties of $DFFITS$. That is, a scaled quantile of a multivariate Student's t process is used as a threshold for $DFFITS$. If any observation's absolute $DFFITS(t)$ is larger than the scaled quantile for any $t \in [0, 1]$, the observation is identified as influential. This new method, denoted the "Theoretical" method, is assessed and compared to Pittman's 2022 methodology, denoted "Bootstrapped" method, via simulation study in Section 4.3. A case study application is presented in Section 3.4 and a discussion of the method is provided in Section 4.5, including its advantages, limitations, and future work.

4.2 Methods

4.2.1 Model

The multiple function-on-function concurrent regression model can be expressed for a single observation as

$$Y(t) = X^T(t)\beta(t) + \varepsilon(t), \quad \text{for} \quad t \in [0,1] \tag{4.1}$$

where $Y_{n\times 1} = \{Y(t) \in \mathbb{R},\, t \in [0,1]\}$ and $X_{k\times n} = (X_1, \ldots, X_K)^T$, with $X_{k,(n\times 1)} = \{X_k(t) \in \mathbb{R},\, t \in [0,1]\}$, for $k = 1, \ldots, K$, and $\varepsilon_{n\times 1} = \{\varepsilon(t) \in \mathbb{R},\, t \in [0,1]\}$ are continuous stochastic processes with constant $X_1(t) = 1$ for all $t \in [0,1]$, and continuous parameter functions $\beta_{k\times 1} = (\beta_1, \ldots, \beta_K)^T$ with $\beta_k = \{\beta_k(t),\, t \in [0,1]\}$ for $k = 1, \ldots, K$. The random sample $(Y_1, X_1^T), \ldots, (Y_n, X_n^T)$ is assumed to be independent and identically distributed (iid) as $\sim(Y, X^T)$, where the error term $\varepsilon = Y - X^T\beta$ is mean independent of X, i.e., $E(\varepsilon(t)|X(t)) = 0$ for all $t \in [0,1]$. Note that the case of function-on-scalar regression is included in (4.1) since any scalar predictor X_k can be defined as a constant predictor function $X_k(t) = X_k(t')$ for all $t, t' \in [0,1]$.

4.2.2 Estimating Functional $DFFITS$

The functional $DFFIT$ (non-standardized) for an observation is defined as

$$DFFIT_i(t) := \hat{Y}_i(t) - \hat{Y}_{i(i)}(t), \tag{4.2}$$

where $\hat{Y}_{i(i)}(t)$ is the prediction of observation $Y_i(t)$ obtained from a model fit without observation $Y_i(t)$, for $t \in [0,1]$. The prediction of $Y_i(t)$ is computed as

$$\hat{Y}_i(t) = X_i^T(t)\hat{\beta}(t), \tag{4.3}$$

where $\hat{\beta}(t) = (\beta_1(t), \ldots, \beta_K(t))^T$ denotes the pointwise least squares estimator

$$\hat{\beta}(t) = \left(\sum_{i=1}^{n} X_i(t)X_i^T(t)\right)^{-1} \sum_{i=1}^{n} X_i(t)Y_i(t) \tag{4.4}$$

that may be evaluated at each time point $t \in [0,1]$. The prediction of $Y_{i(i)}(t)$ is computed in the same way, but with observation i first removed from the sample.

The unbiased estimator of covariance kernel $\sigma_\varepsilon(s,t)$ is given by

$$\hat{\sigma}_\varepsilon(s,t) = \frac{1}{n-K}\sum_{i=1}^{n} e_i(s)e_i(t) \quad \text{with} \quad e_i(t) = Y_i(t) - X_i^T(t)\hat{\beta}(t).$$

The covariance kernel of the error term $\varepsilon(t)$ for the model without observation $Y_i(t)$, is denoted by $\sigma_{\varepsilon(i)}(s,t)$ and has unbiased estimator

$$\hat{\sigma}_{\varepsilon(i)}(s,t) = \frac{1}{n-K-1}\sum_{j\neq i}^{n} e_j(s)e_j(t).$$

The functional leverage of observation $Y_i(t)$ is defined as

$$h_{ii}(t) = X_i^T(t)\left[X(t)X^T(t)\right]^{-1}X_i(t).$$

The functional, standardized $DFFIT$ ($DFFITS$), is then defined as

$$DFFITS_i(t) := \frac{DFFIT_i(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)h_{ii}(t)}} = \frac{\hat{Y}_i(t) - \hat{Y}_{i(i)}(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)h_{ii}(t)}}. \tag{4.5}$$

Alternatively, $DFFITS$ can be defined with the use of externally studentized residuals, $t_{i(i)}$, as

$$DFFITS_i(t) := t_{i(i)}(t)\sqrt{\frac{h_{ii}(t)}{1-h_{ii}(t)}}, \tag{4.6}$$

where $t_{i(i)}(t)$ is defined as

$$t_{i(i)}(t) = \frac{e_i(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)\left(1-h_{ii}(t)\right)}}. \tag{4.7}$$

For more details on the derivation of Equation 4.6, see Appendix C.1 (Seber and Lee, 2003).

4.2.3 Random Errors as Gaussian Process

If the functional errors are assumed to follow a mean-zero Gaussian process with variance or covariance kernel $\sigma_Z(s,t)$, then the studentized residuals, $t_{i(i)}(t)$, are distributed pointwise as a Student's t distribution with $n-K-1$ degrees of freedom. This follows directly from results shown for studentized residuals in non-functional linear regression (Pope, 1976). For each time point, t, the errors are distributed normally with mean zero and variance covariance $\sigma_Z(t,t) \cdot I$. Thus, it can be shown the residuals are also normally distributed pointwise for each timepoint, t:

$$e_i(t) \sim N\left(0, \sigma_z(t,t)\left(1-h_{ii}(t)\right)\right). \tag{4.8}$$

Furthermore, by Cochran's Theorem (Cochran, 1934), this implies

$$\frac{\sum_{i=1}^{n-1} e_{i(i)}^2(t)}{\sigma_Z(t,t)} \sim \chi^2_{n-K-1}, \tag{4.9}$$

pointwise for each timepoint t. Combining results 4.8 and 4.9, the studentized residuals, $t_{i(i)}(t)$, are distributed pointwise as a Student's t distribution with $n-K-1$ degrees of freedom, since

$$t_{i(i)}(t) = \frac{e_i(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)\left(1-h_{ii}(t)\right)}} = \frac{\frac{e_i(t)}{\sqrt{\sigma_Z(t,t)(1-h_{ii}(t))}}}{\sqrt{\frac{\sum_{i=1}^{n-1} e_{i(i)}^2(t)}{(n-K-1)\sigma_Z(t,t)}}} \sim t_{n-K-1}. \tag{4.10}$$

Since the residuals are distributed as a Gaussian process, the theoretical assumptions of Creutzinger Chapter 3 are met (see Section 3.2.1.1 in Chapter 3). By Theorem 3.2.1 of Creutzinger Chapter 3, $\hat{\beta} = \left\{\hat{\beta}(t) : t \in [0,1]\right\}$ and $\hat{\sigma}_{\varepsilon(i)} = \left\{\hat{\sigma}_{\varepsilon(i)}(s,t) : s,t \in [0,1]\right\}$ converge uniformly to their truth, β and $\sigma_{\varepsilon(i)}$. Thus, $t_{i(i)} = \left\{t_{i(i)}(t) : t \in [0,1]\right\}$ follows a Student's t process with $n-K-1$ degrees of freedom. Under assumptions **A1**, **A2**, **A3**, and **A4**, and by Theorems 3.2.1 and 3.2.2 of Chapter 3, the following corollary can be established:

Corollary 4.2.1 (Functional CLT for Externally Studentized Residuals)**.** *Let X, Y, and ε be continuous processes meeting assumptions A1, A2, and A4 in Creutzinger Chapter 3. Let $\hat{\beta} =$*

$\left\{\hat{\beta}(t), t \in [0,1]\right\}$ *be the least squares estimator of* $\beta = \{\beta(t), t \in [0,1]\}$ *as defined in Equation (4.4). Then,*

$$t_{i(i)} \rightarrow_D \mathcal{G}(0,1),$$

where $\mathcal{G}(0,1)$ *is a standard normal Gaussian process.*

Let $DFFITS_i = \{DFFITS_i(t) : t \in [0,1]\}$ and $h_{ii} = \{h_{ii}(t) : t \in [0,1]\}$. Considering Equation 4.6, $DFFITS_i$ follows a scaled Student's t process, given $X_i(t)$. Specifically,

$$DFFITS_i \sim \sqrt{\frac{h_{ii}}{1-h_{ii}}} \cdot t_{n-K-1}. \tag{4.11}$$

If the experimental design is perfectly balanced such that all observations have the same leverage, then $h_{ii}(t) = \frac{K}{n}$, implying that

$$\sqrt{\frac{h_{ii}(t)}{1-h_{ii}(t)}} = \sqrt{\frac{K}{n-K}}.$$

Therefore, functional $DFFITS_i(t)$ can be compared pointwise to $\sqrt{\frac{K}{n-K}} \cdot t_{n-K-1,\alpha}$, where $t_{n-K-1,\alpha}$ is the $1-\alpha$ quantile of a Student's t distribution with $n-K-1$ degrees of freedom. That is, for any observation i at point t_j, $Y_i(t_j)$, if

$$|DFFITS_i(t_j)| > \sqrt{\frac{K}{n-K}} \cdot t_{n-K-1,\alpha}, \tag{4.12}$$

then $Y_i(t_j)$ is identified as potentially influential at that point in the sampling domain for that functional observation. Fewer observations will be identified as potentially influential with smaller α than larger α. If α is too small, then no observations will be identified as potentially influential, even if there is a truly influential observation in the sample. However, if α is too large, then too many potentially influential observations will be identified. A value of 0.100, 0.050, 0.010, or 0.005 is recommended for α.

In the case of identifying potential influential functional observations across the sampling domain, rather than potential influential observations at a particular sampling domain point, the

quantile of the Student's t distribution is replaced with the quantile of a multivariate Student's t distribution with the same degrees of freedom. By Equation (4.11), $DFFITS_i$ follow a scaled Student's t process. As a result, for any sequentially observed points, $t_1, t_2, \ldots, t_T$, the collection $DFFITS_i(t_1), DFFITS_i(t_2), \ldots, DFFITS_i(t_T)$ follows a multivariate Student's t distribution with dimension T. Thus, given a random sample observed on T sampling points, the $1-\alpha$ quantile of a multivariate Student's t distribution of dimension T with $n-k-1$ degrees of freedom is used, denoted $t_{T,n-K-1,\alpha}$. Observation i is identified as a potential influential functional observation if

$$\sum_{j=1}^{T} \mathbb{I}\left\{ |DFFITS_i(t_j)| > \sqrt{\frac{K}{n-K}} \cdot t_{T,n-K-1,\alpha} \right\} > 0. \tag{4.13}$$

This method to identify potential influential observations is denoted the "Theoretical" method. A value of 0.100, 0.050, 0.010, or 0.005 is recommended for α. Note that there is no direct calculation for the quantile of a multivariate Student's t distribution. Instead, a stochastic root finding algorithm described in Bornkamp (2018) can be used to compute an equi-coordinate quantile function of the multivariate Student's t distribution for arbitrary correlation matrices by inverting the distribution function. This can be implemented in `R` with the function `qmvt` from the package `mvtnorm` (R Core Team, 2021; Genz and Bretz, 2009).

4.3 Simulation Study

The data for all simulations are generated using the model

$$\begin{aligned} Y_i(t) &= \beta_1(t) + x_i(t)\beta_2(t) + \varepsilon_i(t), \\ i &= 1, \ldots, n, \quad t = 1, \ldots, T, \end{aligned}$$

where $\varepsilon(t)$ are independent functional errors, which follow a mean-zero, Ornstein-Uhlenbeck process, as generated in Pittman (2022) simulations. The independent $x_i(t)$ are generated by

$$x_i(t) = (t/12) \cdot [a_s \sin([1/k_s][t-d_s]) + c_s] \cdot [a_c \cos([1/k_c][t-d_c]) + c_c], \tag{4.14}$$

after randomly selecting the parameters $a_s, a_c, c_s, c_c, k_s, k_c, d_s$, and d_c. The parameters a_s, a_c, c_s, and c_c are independently sampled from $\{-3, -2, -1, 0, 1, 2, 3\}$, k_s and k_c are sampled from $\{-300, -200, -100, 100, 200, 300\}$, and d_s and d_c are sampled from $\{-100, -50, 0, 50, 100\}$, as done in Pittman (2022). By randomly sampling the parameters for generating $X(t)$, each random sample of predictor functions follow the same underlying curve of $t/12$, but with some variability. An example of $n = 30$ generated $X(t)$ curves is presented in Figure 4.1.

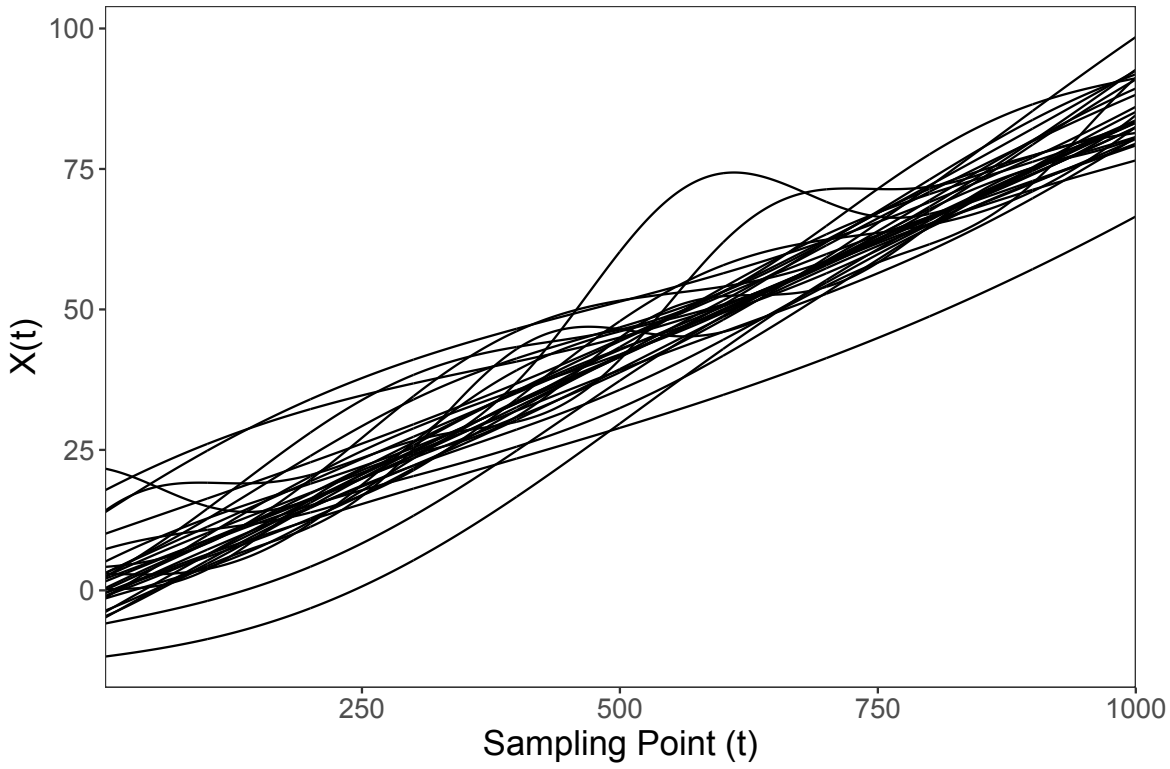


Figure 4.1: An example of $n = 30$ generated $X(t)$ curves using Equation (4.14) and the described method.

Given a random sample of predictor functions $X(t)$, three models are considered for generating the response functions, $Y(t)$. The models differ from one another by altering the parameter functions, $\beta_1(t)$ and $\beta_2(t)$, and are designed to present different proportions of influential sampling

points per influential observation. The first model (hereby, "Model 1") is generated by setting

$$\beta_1(t) = \cos(t/200) + 2 \quad \text{and} \quad \beta_2(t) = \sin(t/200) + 2.$$

Any observation that is generated as an influential observation is obtained by changing $\beta_2(t)$ to be

$$\beta_2(t) = \lambda \times \sin(t/200) + 2,$$

where $\lambda \in \{1, 1.5, 2\}$ controls the magnitude of how influential an observation is (e.g., the further from 1 λ is, the more influential the observation). The second and third models (hereby, "Model 2" and "Model 3", respectively) are generated by setting

$$\beta_1(t) = t/5000 \quad \text{and} \quad \beta_2(t) = t/500.$$

Influential observations are generated for Model 2 by using

$$\beta_2(t) = (\lambda - 1) + t/500,$$

and for Model 3 by using

$$\beta_2(t) = (\lambda - 1) \times \mathbb{I}\{t > 500\} + t/500,$$

where $\lambda \in \{1, 1.5, 2\}$ controls the magnitude of how influential an observation is and $\mathbb{I}$ denotes the indicator function. For all models, if $\lambda = 1$, the observation is not influential. An example of each model with one influential observation generated is presented in Figure 4.2 (a), and the corresponding $DFFITS$ estimated through functional concurrent regression (without regularization or smoothing) is shown in Figure 4.2 (b).

If observation i is generated as a true influential observation and then identified as an influential observation, it is called a true positive (TP). If observation i is generated as a true influential

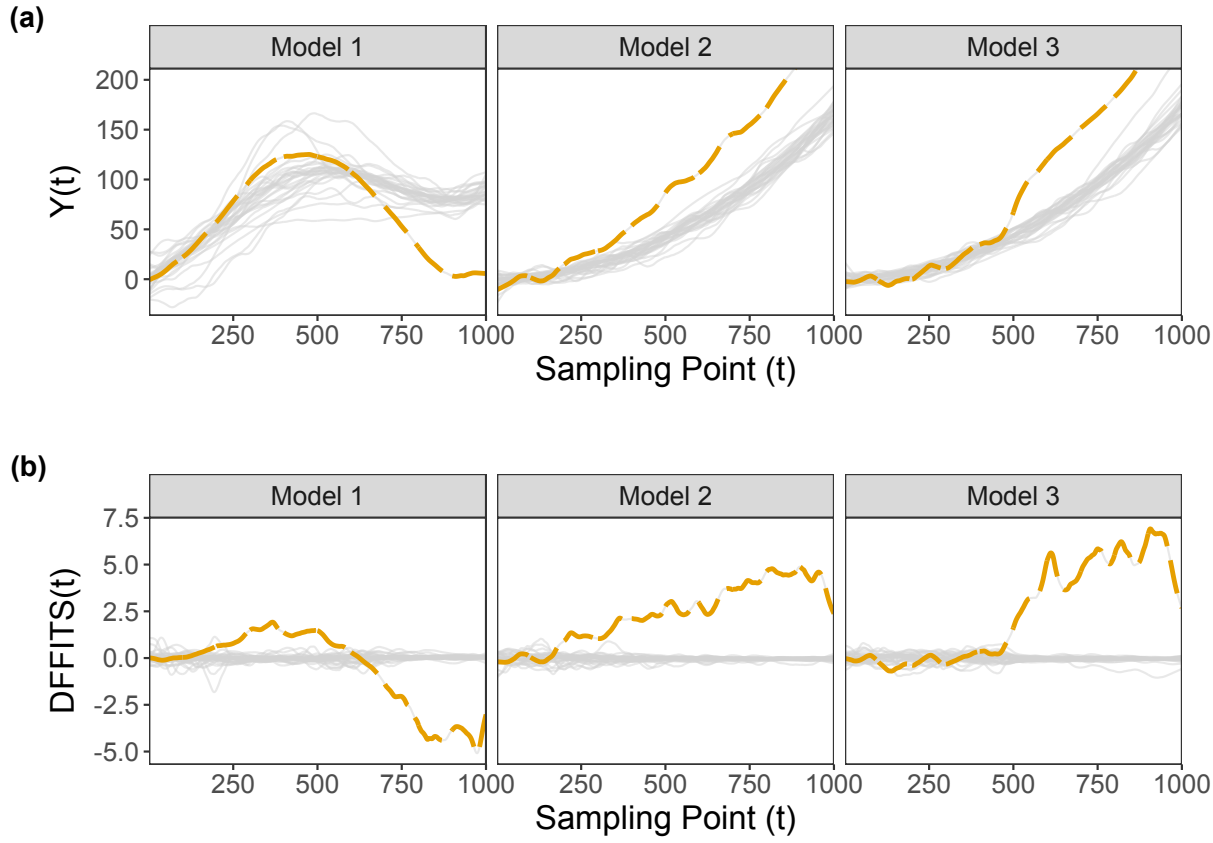


Figure 4.2: (a) An example of $n = 50$ functional responses $Y(t)$ generated by each of the three simulation models, with one observation generated as an influential observation with $\lambda = 1.5$ (dashed, orange line). (b) The corresponding resulting $DFFITS$ estimated without the use of smoothing or regularization.

observation but not identified, it is called a false negative (FN). If observation i is not generated as a true influential observation and not identified, it is called a true negative (TN). Lastly, if observation i is not generated as a true influential observation but is identified as an influential observation, it is called a false positive (FP).

A common choice for model classification diagnostics is the true positive rate, $TPR = \frac{TP}{(TP+FN)}$ (sensitivity), and false positive rate, $FPR = \frac{FP}{FP+TN}$ (1 − specificity). If a researcher would like to diagnose how well they can identify either class correctly (positive or negative), then it is common to combine sensitivity and specificity into one measure, such as accuracy or precision, for comparing classification methods. Accuracy is described as how close a

given set of measurements are to their true value, computed as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.15}$$

Precision (or positive predictive value) is described as how close a given set of measurements are to one another, computed as

$$PPV = \frac{TP}{TP + FP}. \tag{4.16}$$

In this paper, Matthew's Correlation Coefficient (MCC) is used as the main metric for comparison (Matthews (1975)). MCC is described as the correlation between the observed and predicted binary classifications, and is computed as

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{4.17}$$

MCC is a value between -1 (worst) and 1 (best) and is undefined when any component of the denominator (for e.g., TP + FP) is zero. Boughorbel et al. (2017), Delgado and Tibau (2019) , and Chicco and Jurman (2020) show that MCC is regarded as a robust measure, which can be used even if the classes (for e.g., TP, TN, etc) are imbalanced (i.e., the proportion of positives is larger or smaller than the proportion of negatives). By comparison, Wardhani et al. (2019) show that ACC and PPV are not robust to imbalanced classes and may be misleading. The proportion of functional observations considered to be an outlier can be relatively small (for e.g., less than 15% of the observations), which results in imbalanced classes. Thus, ACC and/or PPV may be unreliable as a measure to compare methods.

4.3.1 Simulation Parameter Settings

A simulation study is conducted to examine the accuracy of identifying potential influential functional observations using the Theoretical method (Equation (4.13)) compared to the Bootstrapped method of Pittman (2022). The simulation study is conducted using each of the three models, Model 1, Model 2, and Model 3, with three sample sizes, $n = 10$, 50, and 100, two num-

ber of sampling points, $T = 100$ and 1000, three influential points, $n_{\text{inf}} = 1$, 2, and 3, and three levels of magnitude of how influential an observation is, $\lambda = 1$, 1.5, and 2.

The Bootstrapped method by Pittman (2022) requires bootstrap parameter $\alpha_B \in [0, 0.5]$ and B bootstrap iterations. Parameter α_B is a scaling factor of the selection probability for the parametric bootstrapping algorithm in Pittman (2022). If $\alpha_B = 0$, all observations have an equal selection probability. As α_B increases, the selection probability increases for observations with a smaller value of mean absolute $DFFITS$. Parameter B is set to 100 iterations and three values of α_B are assessed: 0.00, 0.25, and 0.50. In the simulation results, α corresponds to using the $1 - \alpha$ quantile of a multivariate Student's t distribution and/or the bootstrapped distribution of Pittman (2022). Note that Pittman (2022) did not consider the scenario of more than one influential observation present in the sample and only assessed the performance of their method when implemented with smoothing and/or regularization. Thus, for a full comparison, we implement the Bootstrapped method as described by Pittman (2022) (denoted "Bootstrapped (smooth)") and use their estimated $DFFITS$ to apply the Theoretical method (denoted "Theoretical (smooth)"). Then, we implement the Bootstrapped method by Pittman (2022) without any smoothing or regularization (denoted "Bootstrapped (raw)") and compare the results with the Theoretical method implemented without any smoothing or regularization (denoted "Theoretical (raw)"). All methods are assessed using four values of α, $\alpha = 0.100$, 0.050, 0.010, and 0.005.

4.3.2 Simulation Results

When the simulation results are averaged over Model 1, Model 2, and Model 3, the sample sizes $n = 10, 50$, and 100, the number of sampling points, $T = 100$ and 1000, the number of influential points $n_{\text{inf}} = 1, 2$, and 3, and the varying magnitudes of how influential an observation is, $\lambda = 1, 1.5$, and 2, then Bootstrapped (smooth) has a higher average PPV and MCC for all values of α and α_B than Bootstrapped (raw) (see Table C.1 in the Appendix). Additionally, when comparing the results for different values of α_B, Bootstrapped (smooth) has the highest average value of ACC, PPV, and MCC when $\alpha_B = 0.500$ for all values of α. One advantage

of using Bootstrapped (raw) is the average (standard deviation) run time (40.732 (32.883) seconds when $\alpha_B = 0.500$) is considerably faster than the average run time when using Bootstrapped (smooth) (3415.135 (3034.031) seconds when $\alpha_B = 0.500$) (see Table C.1). However, the ACC, PPV, and MCC when implementing Bootstrapped (raw) is lower on average in all simulations than when implementing Bootstrapped (smooth). Therefore, for further simulation comparisons, we consider only the case of Bootstrapped (smooth), implemented with $\alpha_B = 0.500$ and $B = 100$ bootstrap iterations. Similarly, the ACC, PPV, and MCC are all smaller on average when applying Theoretical (smooth) than when applying Theoretical (raw). Thus, the simulation results presented here focus on comparing Theoretical (raw) to Bootstrapped (smooth) with $\alpha_B = 0.500$. See Table C.2 in Appendix C.2.2 for a comparison on the impact of smoothing and regularization from both the Theoretical and Bootstrapped methods.

For both the Theoretical (raw) and Bootstrapped (smooth) methods, as α increases, the sensitivity increases and the specificity decreases on average (calculated by averaging over Model 1, Model 2, and Model 3, the sample sizes $n = 10$, 50, and 100, the number of sampling points, $T = 100$ and 1000, the number of influential points $n_{\text{inf}} = 1$, 2, and 3, and the varying magnitudes of how influential an observation is $\lambda = 1$, 1.5, and 2). The average ACC, PPV, and MCC are highest for Theoretical (raw) when implemented with $\alpha = 0.005$ for the $(1 - \alpha) \times 100\%$ quantile of the multivariate Student's t distribution. The average value of ACC and MCC are highest for Bootstrapped (smooth) when implemented with $\alpha = 0.005$ for the $(1 - \alpha) \times 100\%$ quantile of the bootstrapped distribution acquired with $B = 100$ bootstrap iterations and $\alpha_B = 0.5$ (see Table 4.1). Since both methods have the best ACC and MCC when $\alpha = 0.005$, the remaining simulation results focus on Theoretical (raw) versus Bootstrapped (smooth) with $B = 100$ and $\alpha_B = 0.50$ with $\alpha = 0.005$. Note, the comparison of Theoretical (raw) and Bootstrapped (smooth) has the same trend as the results mentioned next, regardless of α.

Using $\alpha = 0.005$, Bootstrapped (smooth) has a higher average ACC, PPV, and MCC ($0.944, 0.627, 0.688$, respectively) than Theoretical (raw) ($0.913, 0.217, 0.343$, respectively). The lower average value of PPV and MCC for Theoretical (raw) is due to lower average specificity

corresponding to a higher FPR, on average. However, the average (standard deviation) run time for the Bootstrapped (smooth) method is 3415.135 (3034.031) seconds while the average (standard deviation) run time for the Theoretical (raw) method is 0.412 (0.329) seconds.

Table 4.1: The average (s.d.) run time (in seconds), sensitivity, specificity, accuracy (ACC), precision (PPV), and Matthew's Correlation Coefficient (MCC) of Theoretical (raw) and Bootstrapped (smooth) using $B = 100$ bootstraps and $\alpha_B = 0.5$, for each value of $\alpha = 0.005$, 0.025, 0.050, and 0.100. The averages are calculated by averaging over Model 1, Model 2, and Model 3, the sample sizes $n = 10$, 50, and 100, the number of sampling points, $T = 100$ and 1000, the number of influential points $n_{\text{inf}} = 1$, 2, and 3, and the magnitude of how influential an observation is $\lambda = 1$, 1.5, and 2.

	$\boldsymbol{\alpha}$	Sensitivity	Specificity	ACC	PPV	MCC
Theoretical						
Run Time: 0.412 (0.329) s						
	0.005	0.394 (0.459)	0.958 (0.039)	0.913 (0.085)	0.217 (0.331)	0.343 (0.389)
	0.025	0.434 (0.463)	0.931 (0.050)	0.891 (0.088)	0.186 (0.299)	0.299 (0.372)
	0.050	0.455 (0.462)	0.914 (0.057)	0.876 (0.090)	0.173 (0.284)	0.278 (0.362)
	0.100	0.481 (0.458)	0.889 (0.066)	0.855 (0.095)	0.160 (0.264)	0.257 (0.348)
Bootstrapped						
Run Time: 3415.135 (3034.031) s						
	0.005	0.148 (0.334)	0.999 (0.003)	0.944 (0.087)	0.627 (0.483)	0.688 (0.403)
	0.025	0.374 (0.472)	0.989 (0.015)	0.941 (0.086)	0.360 (0.454)	0.522 (0.472)
	0.050	0.424 (0.483)	0.972 (0.029)	0.927 (0.085)	0.293 (0.398)	0.460 (0.450)
	0.100	0.479 (0.471)	0.931 (0.049)	0.893 (0.086)	0.231 (0.346)	0.360 (0.407)

The average specificity is also calculated by averaging over the models, Model 1, Model 2, and Model 3, and the number of sampling points, $T = 100$ and 1000 (see Figure C.10). The average specificity of Theoretical (raw) and Bootstrapped (smooth) decreases as the sample size increases. Yet, regardless of the sample size, n, number of influential observations, or the magnitude of how influential an observation is, λ, Bootstrapped (smooth) with $\alpha = 0.005$ has a larger specificity on average (ranges from 0.997 to 1.000) than Theoretical (raw) with $\alpha = 0.005$ (ranges from 0.937 to 1.000). The average specificity of both models also increases as the magnitude of how influential an observation is, λ, increases. The average specificity of both methods is also calculated by averaging over the models, Model 1, Model 2, and Model 3, and the sample size, $n = 10$, 50, and 100 (see Figure C.10). For $\alpha = 0.005$, the average specificity of both methods, Theoretical (raw)

and Bootstrapped (smooth), increases as the number of sampling points, T, increases, except when no influential observations are present in the sample (e.g., $\lambda = 1$).

The average sensitivity is also calculated by averaging over the models, Model 1, Model 2, and Model 3, and the number of sampling points, $T = 100$ and 1000. When $\alpha = 0.005$, the average sensitivity of Theoretical (raw) and Bootstrapped (smooth) increases as the sample size increases (see Figure C.12). Note that results are not shown for when $\lambda = 1$. This is because when $\lambda = 1$, none of the observations in the random sample are a true influential observation (e.g., total number of $TP = 0$). When $\alpha = 0.005$, Theoretical (raw) has a higher average sensitivity (ranges from 0.035 to 0.562) than Bootstrapped (smooth) (ranges from 0 to 0.522) for every value of sample size, n, number of influential observations, # Influential, or the magnitude of how influential an observation is, λ. The average sensitivity of both methods is also calculated by averaging over the models, Model 1, Model 2, and Model 3, and the sample size, $n = 10$, 50, and 100. The average sensitivity of Theoretical (raw) and Bootstrapped (smooth) increases as the number of sampling points increase when $\alpha = 0.005$ (see Figure C.13). Theoretical (raw) has a higher average sensitivity (ranges from 0.049 to 1) than Bootstrapped (smooth) (ranges from 0.001 to 0.670) when $\alpha = 0.005$ for every value of sample size, n, number of influential observations, # Influential, or the magnitude of how influential an observation is, λ.

The average MCC of Theoretical (raw) and Bootstrapped (smooth), each with $\alpha = 0.005$, is calculated by averaging over the models, Model 1, Model 2, and Model 3, and the number of sampling points, $T = 100$ and 1000 (see Figure 4.3). However, when the sample size $n = 10$, the MCC could not be calculated for the majority of the simulation iterations for either method and was not included. An incalculable MCC when $n = 10$ most often occurred due to none of the observations being identified as a "positive", which leads to $TP = FP = 0$. The denominator of Equation 4.17 is thus zero as well. There is not a clear trend in MCC for either methodology when considering the sample size only.

In all scenarios except for the case when the number of influential observations is 2, the sample size is $n = 50$, and the magnitude of how influential the observation is set to $\lambda = 2$, or when

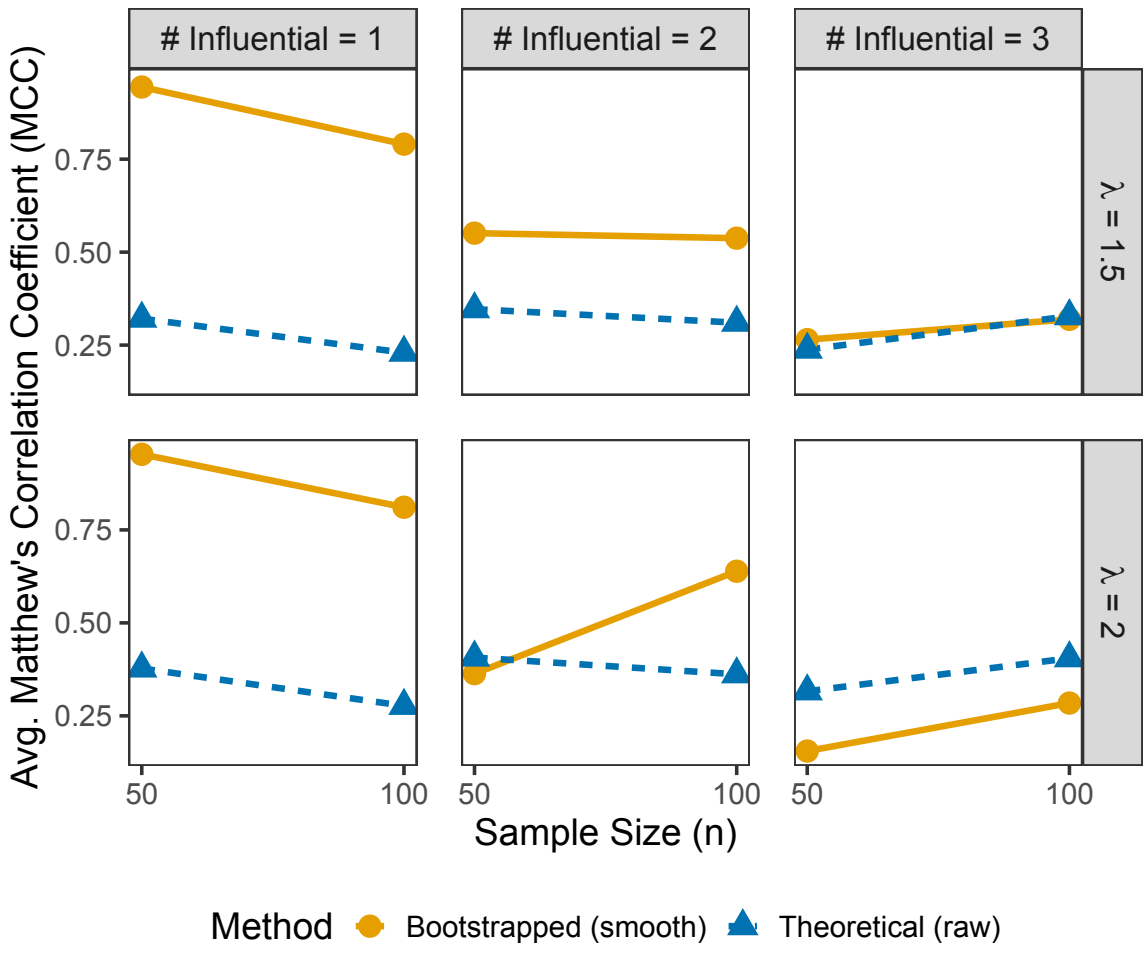


Figure 4.3: Average Matthew's Correlation Coefficient (MCC) of Theoretical (raw) (dashed line with triangle points) and Bootstrapped (smooth) (solid line with circle points) when $\alpha = 0.005$. The average Matthew's Correlation Coefficient (MCC) is calculated by averaging over Model 1, Model 2, and Model 3 and the number of sampling points, $T = 100$ and 1000. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ.

the number of influential observations is 3, the sample size is $n = 100$, and the magnitude of how influential the observation is set to $\lambda = 1.5$, or when the number of influential observations is 3 and the magnitude of how influential the observation is set to $\lambda = 2$, the average MCC of Bootstrapped (smooth) is larger (ranges from 0.265 to 0.953) than Theoretical (raw) (ranges from 0.230 to 0.377). When $\alpha = 0.005$, the average MCC of Theoretical (raw) and Bootstrapped (smooth) is also calculated by averaging over the models, Model 1, Model 2, and Model 3, and the sample size,

$n = 10$, 50, and 100 (see Figure 4.4). The average MCC of both methods, Theoretical (raw) and Bootstrapped (smooth), implemented with $\alpha = 0.005$, increases as the number of sampling points increases. The average MCC of Bootstrapped (smooth) is larger than Theoretical (raw) for all numbers of influential points, magnitude of how influential an observation is set to λ, and number of sampling points, T. The only exception is that Theoretical (raw) has a larger average MCC than Bootstrapped (smooth) when the number of influential observations is 3, the magnitude of how influential the observation is set to $\lambda = 2$, and the number of sampling points $T = 1000$ and when the number of influential observations is 1, the magnitude of how influential the observation is set to $\lambda = 1.5$, and the number of sampling points is $T = 100$.

4.4 Application

The River Floods case study is originally described in Pittman (2022). The data in the River Floods case study represent the river height of Congaree River and Cedar Creek during historical flood events. A flood event occurs when the maximum height of the river reaches 17.85ft or more. From the year 1995 to 2020, only 12 flood events of the Congaree River have occurred. However, of the 12 flood events recorded for Congaree River, Cedar Creek only has observations for ten of them, thus making the sample size $n = 10$. One of the non-observed Cedar Creek dates, October 5th, 2015, is chosen by Pittman (2022) as a specific date of interest with the goal of using functional concurrent regression to predict the height of Cedar Creek during this flood event. The height of Congaree River was measured from October 1st, 2015 to October 21st, 2015 every 15 minutes for a total of 500 hours, which results in $T = 2000$ sampling points.

Pittman (2022) uses a novel Landmark Aligned L_1 (LAL_1) distance approach to approximate the start and end time of the other flood events found, relative to the October 15th flood event. Since not every flood event occurs during the same duration, the newly aligned flood events are interpolated on 2000 sampling points between the start and end date determined by the LAL_1 alignment. The functional heights of the realigned Congaree River flood events are presented in Figure 4.5. Pittman (2022) uses the ten complete flood events for Congaree River as a functional

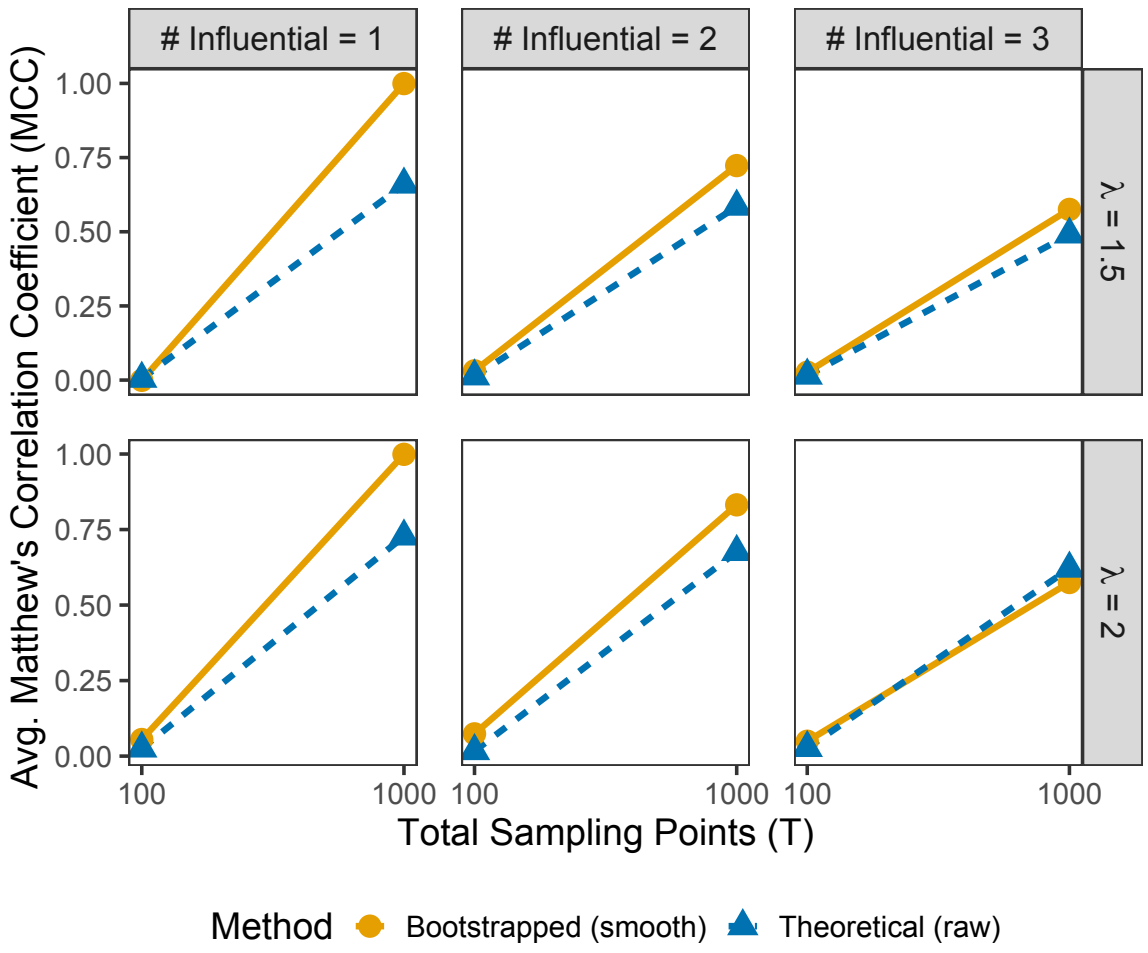


Figure 4.4: Average Matthew's Correlation Coefficient (MCC) of Theoretical (raw) method implemented with $\alpha = 0.005$ (solid line with circle points) and Bootstrapped (smooth) implemented with $\alpha = 0.025$ (dashed line with triangle points). The average Matthew's Correlation Coefficient (MCC) is calculated by averaging over the models, Model 1, Model 2, and Model 3 and the sample size, $n = 10, 50$, and 100. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ.

predictor variable and the ten matching events of Cedar Creek as the functional response variable. Before fitting a functional concurrent regression model, both Congaree River and Cedar Creek flood events are smoothed with a Fourier basis using $\lambda = 10^{-1}$ and $M = 11$ basis functions. The values of λ and M are determined through leave-one-out cross validation. Lastly, Pittman (2022) uses the `fRegress` function from `R` package `fda` to fit a functional concurrent regression model with the smoothed flood events.

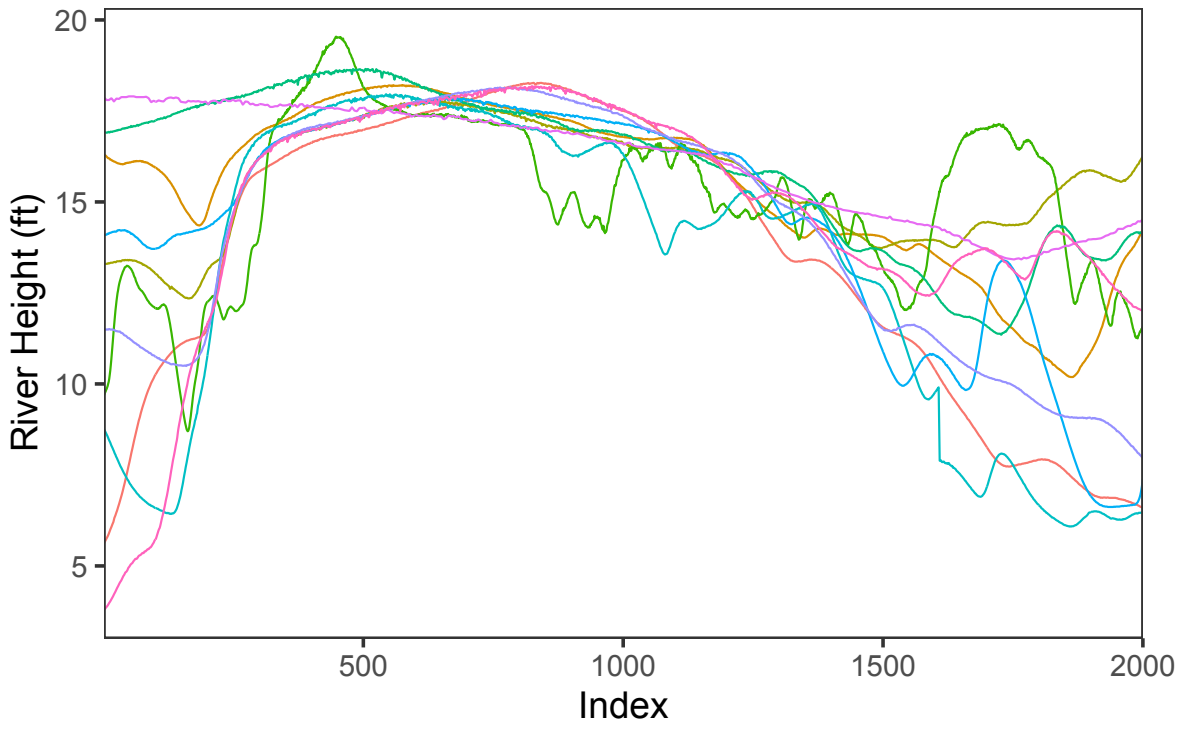


Figure 4.5: The realigned functional river heights (ft) of the ten Congaree River flood events in the sample. The functions are realigned by applying Pittman (2022) LAL_1.

After concurrently regressing Cedar Creek heights on Congaree River heights, Pittman (2022) estimated $DFFITS_i(t_j)$ for each observation i at each sampling point t_j. Then, the estimated $DFFITS$ were smoothed the same way as done originally with the functional data using a Fourier basis with $\lambda = 10^{-1}$ and $M = 11$ basis functions (Figure 4.6). Pittman (2022) then estimated the average value of $DFFITS(t)$ for each observation i by averaging over the sampling points, t_j, and applied their parametric bootstrapping (Bootstrapped (raw) method) to approximate the null distribution of the average $|DFFITS|$. The average $|DFFITS|$ value for each observation can be seen in Table C.3, Section C.3. The average $|DFFITS|$ were compared to a threshold value, which is a quantile from the bootstrapped null distribution. Specifically, an estimate of the 97.5^{th} quantile ($\alpha = 0.025$) of the bootstrapped distribution is 2.51, as obtained by implementing the bootstrap procedure of Pittman (2022) with $B = 100$ iterations and $\alpha_B = 0.5$. Using this

approach, the only Cedar Creek flood event with an average $|DFFITS(t)| > 2.51$ was identified as occurring in February 2020.

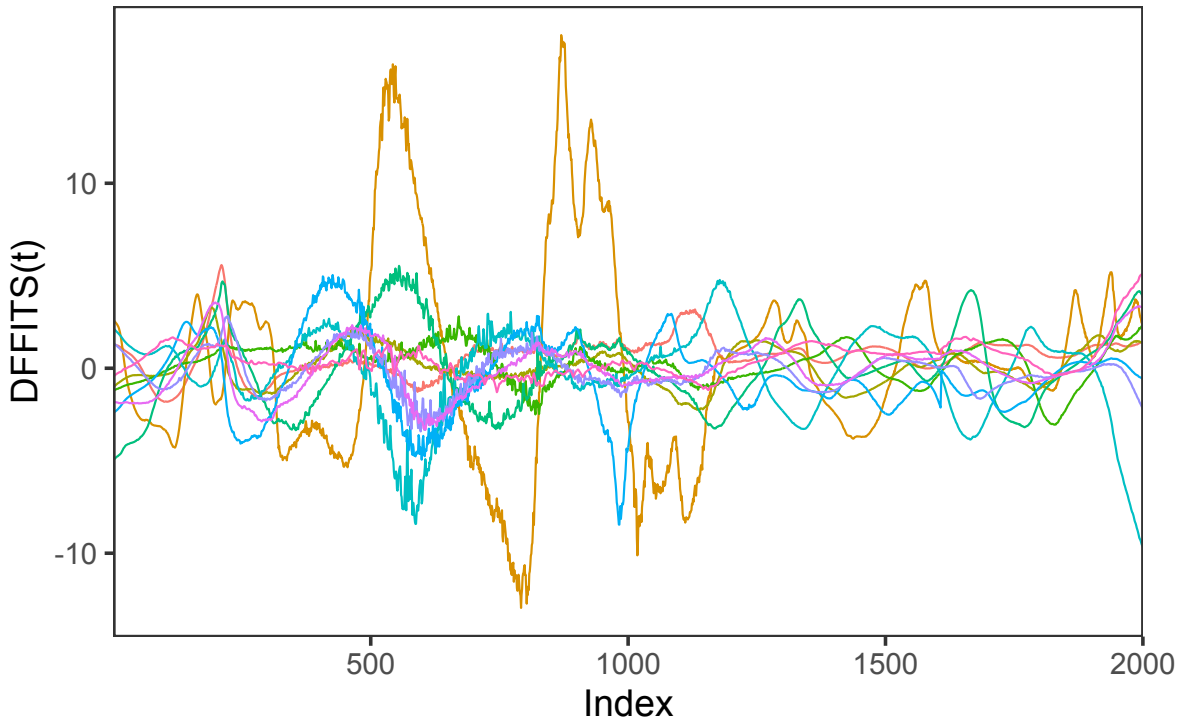


Figure 4.6: Estimates of the functional $DFFITS$, which are acquired by implementing the methodology of Pittman (2022).

The realigned functional river height data are concurrently regressed *without* smoothing or using regularization, and the functional $DFFITS$ are estimated. These newly estimated $DFFITS$, along with the 99^{th} percentile ($\alpha = 0.005$) of multivariate Student's t distribution with $7 = n - K - 1$ degrees of freedom, and dimension $T = 1000$ are shown in Figure 4.7 (Genz and Bretz, 2009). (Note: the functional river height data is actually observed on $T = 2000$ sampling points, so the quantile is an approximation.) When using the original data without smoothing and applying Theoretical (raw), two of the ten flood events (March 2007 and February 2020) have at least one value that exceeds the relevant multivariate Student's t quantile (see Figure 4.7).

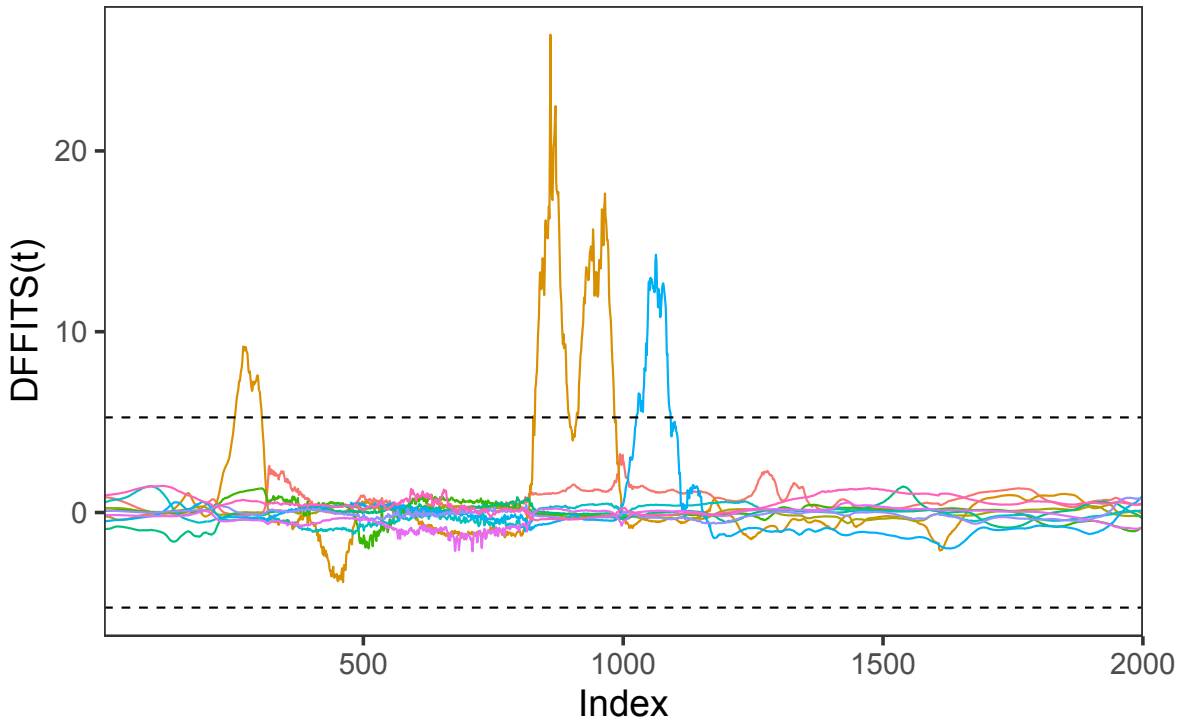


Figure 4.7: Estimates of the functional $DFFITS$, which are acquired by implementing Theoretical (raw). The dashed horizontal lines represent the $\alpha = 0.005$ and $1 - \alpha$ quantile of a multivariate Student's t distribution with $T = 1000$ and $df = 7$.

Similar to the simulation results, more influential observations are identified using the Theoretical (raw) method than the Bootstrapped (smooth) method. However, in this application, the February 2020 flood event is identified as an influential observation by both methods.

4.5 Conclusion

The distributional properties of externally studentized residuals, $t_{i(i)}$, and functional $DFFITS$ are established for the case of functional concurrent regression with functional errors following a mean-zero Gaussian process. A rule for identifying influential observations in a functional concurrent regression is proposed based on the distributional properties of functional $DFFITS$ (see Section 4.2). The distributional-based rule involves estimating a quantile of the multivariate Student's t distribution as an approximation to the quantile of a Student's t process. The multivariate Stu-

dent's t distribution quantile is estimated using the function `qmvt` from the `R` package `mvtnorm`, which is limited to a dimension of 1000. In other words, the multivariate Student's t distribution quantile can only be used to approximate the Student's t process quantile for functions observed on $T = 1000$ sampling points or less (R Core Team, 2021; Genz and Bretz, 2009). Future work could be done to extend the Theoretical method to functional non-concurrent regression models or the case of non-Gaussian errors. The challenge would be identifying the analytical distribution of functional $DFFITS_i$ and estimation of the covariance kernel.

The Theoretical (raw) method for identifying influential observations is compared to Pittman's 2022 Bootstrapped (smooth) method ($B = 100$ and $\alpha_B = 0.5$) in a simulation study (see Section 4.3). The Pittman (2022) method has a larger ACC, PPV, and MCC, on average, than the Theoretical (raw) method. However, Pittman (2022) methodology is computationally expensive and has a lower average sensitivity. Using the Theoretical (raw) method, more observations are identified as potentially influential, on average, than Pittman's 2022 Bootstrapped (smooth). As a result, Theoretical (raw) often results in a higher FPR than Pittman's 2022 Bootstrapped (smooth). Furthermore, the FPR increases on average as sample size n increases, which may be due to the rate of $\sqrt{\frac{K}{n-K}}$ converging to zero and shrinking the estimated multivariate Student's t distribution quantile. A sample size adjustment to scalar $\sqrt{\frac{K}{n-K}}$ when the sample size is large could improve the ACC, PPV, and MCC diagnostic characteristics of the Theoretical method. It may also be possible that using the externally studentized residuals, $t_{i(i)}$, for identifying potentially influential observations would be more accurate than $DFFITS$ for larger sample sizes ($n = 50$ or 100). Implementing Pittman's 2022 Bootstrapped (smooth) can be unstable at a small sample size ($n = 10$), due to non-full rank predictor matrix. Theoretical (raw) is stable at small sample size ($n = 10$), because it does not rely on any resampling techniques.

A challenge in smoothing and regularization techniques can be the presence of influential observations in the random sample being smoothed and/or regularized (Edith U Umeh and Chinyere I Ojukwu, 2019; Hellton et al., 2023). It was found with a simulation study that implementing Pittman's 2022 Bootstrapped (smooth) without smoothing and/or regularization is not as accurate

on average (in terms of ACC, PPV, and MCC) as when implemented with smoothing and regularization. On the contrary, when implementing the Theoretical method presented in this paper, the results are more accurate on average when applied to the raw functional data, without any smoothing or regularization.

Chapter 5

Conclusion

The growing presence of high resolution data has greatly increased the popularity of using methods in functional data analysis (FDA), which has resulted in more recent and continual development of FDA methods. The focus of this dissertation was to develop a toolbox of methodology that allows a researcher to investigate, model, and infer from a random sample of functional data. Given a random sample of functional data, including a functional response variable and functional/scalar predictor variable(s), we developed methodology to identify functional outlying observations, to create simultaneous confidence/prediction bands for the estimated conditional mean of a functional concurrent regression model, and to identify functional influential observations in the concurrent regression model. The methods presented in this dissertation allows a researcher to fully analyze their functional data, without the need to reduce the sample to a non-functional sample.

In FDA, outlying observations can be identified as one or more of the following: magnitude outliers, shape outliers, and/or amplitude outliers (Hubert et al. (2015), Ojo et al. (2021)). Outlying features may be present across the whole domain of data collection (e.g., $t \in [0, 1]$), or in one or more sub-intervals of the domain. Methods in functional data outlier detection have been well-developed to identify outliers in a univariate random sample. An overview of methods can be found in Ojo et al. (2021). However, many methods available to identify outliers rely on statistical measures that are not easily interpretable.

The first method we proposed in this dissertation, Practical Outlier Detection (POD), is a practical, easily interpretable approach to outlier detection. The method is developed using summary statistics that are introduced in nearly all introductory statistics courses. A list of nine summary statistics are estimated for each functional observation on several disjoint intervals of the sampling domain. The estimates are compared across observations and extreme estimates are identified. The functional observation(s) with the most extreme estimates are identified as a functional outlier and

further classified as either a magnitude or shape (including amplitude) outlier. We find POD has classification diagnostics as good or better than all other methods, while only relying on ordinary summary statistics. Furthermore, POD has better classification diagnostics (i.e., accuracy, precision, and Matthew's correlation coefficient) than the other methods in comparison for correctly classifying the type of functional outlier (shape and/or magnitude). POD results are easy to interpret and provide a summary of the number of extreme summary statistics of each type for each functional outlier identified.

We also proposed functional outlier detection through the use of simultaneous prediction bands. In a non-functional setting, the use of a prediction band has been considered for outlier detection (Horn et al., 1988). However, a prediction band is often computed with the use of an estimated mean and standard deviation: two statistics that are susceptible to outliers. Horn et al. (1988) show that the presence of outliers produces a prediction interval that is too wide, compared to when no outliers are present. In this dissertation, a simultaneous prediction band is created with the use of a critical value function created by Liebl and Reimherr (2023). Then, a resampling algorithm is implemented to create a functional outlier detection method, Prediction Band Outlier Detection (PBOD). However, simulation results showed that PBOD had the lowest average ACC and average PPV, when compared to competing methods (including POD).

Given its success over PBOD, we suggest that POD is implemented to find and classify a functional outlying observation in a functional random sample. POD methodology relies on the use of a threshold for separating the functional outliers from the rest of the random sample. The threshold value can be user-specified or can be taken as the upper fence of Tukey's boxplot. The use of POD for identifying functional outliers is most accurate when a user-specified threshold is provided, but this requires the user to know a priori the proportion of functional outliers present in the random sample. Another element that directly affects the results of using POD is the number of disjoint intervals created. The number of sampling points observed in the functional random sample restricts how many disjoint intervals can be created. If there are not enough intervals, or

if too many intervals are created, the classification diagnostics of POD are worse. However, a possible solution would be to consider non-disjoint intervals of different sizes.

After the data have been examined for outliers, inferential techniques – in particular, simultaneous inference – can be considered. A common analysis choice to acquire inferential results is the use of linear regression. In FDA, functional concurrent regression considers a functional linear regression model with a functional response, scalar and/or functional predictors, regressed pointwise at each sampling point in the domain. After estimating a regression model (functional or non-functional), it is common to acquire confidence and prediction intervals for the parameter(s), including a conditional mean. However, literature on prediction bands for functional data is scarce and underdeveloped, and none of the existing prediction bands for functional data allows conditioning on predictor values. In this dissertation, we address this methodological gap and contribute a novel $(1-\alpha)\times 100\%$ Simultaneous Prediction Band (SPB), called Fast and Fair Simultaneous Prediction Band (FFSPB). $\mathrm{FFSPB}_{1-\alpha}(Y_{x_{new}}(t))$ provides a "normal" range for functional data $Y(t)$, conditionally on functional or non-functional covariates $X = x_{new}$. Our FFSPB has three properties relevant to the Sprint Start Kinetics case study, which are not provided by alternative approaches: 1. Fair predictive inference, 2. Covariate-adjustment, and 3. Coverage for wide-tailed errors.

We compared our fast and fair methodology to conformal inference and find that prediction bands created by either methodology meets nominal coverage levels. The coverage level of fast and fair bands is more conservative than conformal inference bands, but the fast and fair bands are narrower with a better band score on average than conformal inference bands. A natural extension of FFSPB methodology is to develop the bands for more functional regression methods (e.g., non-concurrent or concurrent generalized linear regression).

Following the estimation of a linear regression model (either in the functional setting or not), it is common to assess the model with diagnostic measures and to check for influential observations. An influential observation can be defined as an observation that is outlying in its relationship between the response variable and predictor variable(s). Methods to assess and identify an influential

observation in non-functional ordinary linear regression have been well studied and developed over the years, but are still under development for functional linear regression. A common approach in functional linear regression to identify influential observations is to calculate a non-functional measure of influence (e.g., $DFFITS$, $DFBETAs$, Cook's Distance, etc.) at each observed sampling point (t_j). Then, the mean of the influence measure is taken over the sampling domain for each observation, and the null distribution of the mean influence measure is estimated through resampling techniques.

In this dissertation, we established the distributional properties of $DFFITS$ for functional concurrent regression. Namely, we showed that when the error processes are assumed to follow a mean-zero Gaussian process, $DFFITS$ is distributed as a scaled Student's t process. Then, as is suggested in non-functional linear regression, a quantile from the Student's t process can be used as a cutoff point for identifying influential observations. Using a quantile from the Student's t process prevents the need for resampling, saving computation time in the process, while still displaying relatively high accuracy. The next step in developing functional $DFFITS$ is to establish distributional properties when the error process are not assumed to follow a mean-zero Gaussian process. It would also be beneficial to establish $DFFITS$ properties for functional non-concurrent regression.

The developments in this dissertation provide methodology for a researcher to analyze a random sample of functional data from "start to finish." Several of the proposed methods are built on well-founded statistics for non-functional data analysis. By extending non-functional methodology to functional methodology, our methods provide a familiarity to statisticians, even to those without experience in FDA.

References

Abramowicz, K., C. K. Häger, A. Pini, L. Schelin, S. Sjöstedt De Luna, and S. Vantini (2018). Nonparametric inference for functional-on-scalar linear models applied to knee kinematic hop data after injury of the anterior cruciate ligament. *Scandinavian Journal of Statistics 45*(4), 1036–1061.

Azcorra, A., L. F. Chiroque, R. Cuevas, A. Fernández Anta, H. Laniado, R. E. Lillo, J. Romo, and C. Sguera (2018). Unsupervised scalable statistical method for identifying influential users in online social networks. *Scientific Reports 8*(1), 6955.

Belloni, A., V. Chernozhukov, D. Chetverikov, and Y. Wei (2018). Uniformly valid post-regularization confidence regions for many functional parameters in z-estimation framework. *The Annals of Statistics 46*(6), 3643–3675.

Belsley, D. A., E. Kuh, and R. E. Welsch (2005). *Regression Diagnostics Identifying Influential Data and Sources of Collinearity* (1., Auflage ed.). John Wiley & Sons.

Bendre, S. M. and B. K. Kale (1987). Masking effect on tests for outliers in normal samples. *Biometrika 74*(4), 891–896.

Bornkamp, B. (2018). Calculating quantiles of noisy distribution functions using local linear regressions. *Computational Statistics 33*(1), 487–501.

Boughorbel, S., F. Jarray, and M. El-Anbari (2017). Optimal classifier for imbalanced data using Matthew's correlation coefficient metric. *PLOS ONE 12*(6), e0177678.

Brys, G., M. Hubert, and P. J. Rousseeuw (2005). A robustification of independent component analysis. *Journal of Chemometrics 19*(5), 364–375.

Brys, G., M. Hubert, and A. Struyf (2004). A robust measure of skewness. *Journal of Computational and Graphical Statistics 13*(4), 996–1017.

Bunea, F., A. E. Ivanescu, and M. H. Wegkamp (2011). Adaptive inference for the mean of a Gaussian process in functional data. *Journal of the Royal Statistical Society Series B: Statistical Methodology 73*(4), 531–558.

Cao, G. (2014). Simultaneous confidence bands for derivatives of dependent functional data. *Electronic Journal of Statistics 8*(2), 2639–2663.

Cao, G., L. Yang, and D. Todem (2012). Simultaneous inference for the mean function based on dense functional data. *Journal of Nonparametric Statistics 24*(2), 359–377.

Chang, C., X. Lin, and R. T. Ogden (2017). Simultaneous confidence bands for functional regression models. *Journal of Statistical Planning and Inference 188*, 67–81.

Chen, G., C. Huang, and J. Lin (2014). Statistical diagnostics for functional linear regression models with gaussian process errors. *Communication on Applied Mathematics and Computation 28*(1), 118–126.

Chicco, D. and G. Jurman (2020). The advantages of the Matthew's correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics 21*(1), 6.

Chiou, J.-M. and H.-G. Müller (2007). Diagnostics for functional regression via residual processes. *Computational Statistics & Data Analysis 51*(10), 4849–4863.

Cochran, W. G. (1934). The distribution of quadratic forms in a normal system, with applications to the analysis of covariance. *Mathematical Proceedings of the Cambridge Philosophical Society 30*(2), 178–191.

Cook, R. D. (1977). Detection of influential observation in linear regression. *Technometrics 19*(1), 15–18.

Cook, R. D. and S. Weisberg (1982). Criticism and influence analysis in regression. *Sociological Methodology 13*, 313.

Court of Arbitration for Sport (2020). *CAS 2020/A/6807 Blake Leeper v. International Association of Athletics Federations*.

Creutzinger, M. L., D. Liebl, and J. L. Sharp (2024+). (In review) Fair simultaneous prediction and confidence bands for concurrent functional regressions: Comparing sprinters with prosthetic versus biological legs. *Journal of the American Statistical Association*.

Creutzinger, M. L. and J. L. Sharp (2024+). (In review) Practical outlier detection in functional data analysis. *The American Statistician*, 1–31.

Cuevas, A., M. Febrero, and R. Fraiman (2006). On the use of the bootstrap for estimating functions with functional data. *Computational Statistics & Data Analysis 51*(2), 1063–1074.

Cuevas, A., M. Febrero, and R. Fraiman (2007). Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics 22*(3), 481–496.

Dai, W. and M. G. Genton (2018). Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics 27*(4), 923–934.

Dai, W. and M. G. Genton (2019). Directional outlyingness for multivariate functional data. *Computational Statistics & Data Analysis 131*, 50–65.

Dai, W., T. Mrkvička, Y. Sun, and M. G. Genton (2020). Functional outlier detection and taxonomy by sequential transformations. *Computational Statistics & Data Analysis 149*, 106960.

Davidson, J. (2021). *Stochastic Limit Theory: An Introduction for Econometricians* (Second edition ed.). Advanced Texts in Econometrics. Oxford University Press.

de Silva, G. S. and P. K. Choudhary (2023). Tolerance bands for exponential family functional data. *The Canadian Journal of Statistics*.

Degras, D. (2017). Simultaneous confidence bands for the mean of functional data: SCB for the mean of functional data. *Wiley Interdisciplinary Reviews: Computational Statistics 9*(3), e1397.

Delgado, R. and X.-A. Tibau (2019). Why Cohen's kappa should be avoided as performance measure in classification. *PLOS ONE 14*(9), e0222916.

Diquigiovanni, J., M. Fontana, A. Solari, S. Vantini, and P. Vergottini (2022). *conformalInference.fd: Tools for Conformal Inference for Regression in Multivariate Functional Setting*. R package version 1.1.1.

Diquigiovanni, J., M. Fontana, and S. Vantini (2022). Conformal prediction bands for multivariate functional data. *Journal of Multivariate Analysis 189*, 104879.

Ecker, K., X. de Luna, and L. Schelin (2024). Causal inference with a functional outcome. *Journal of the Royal Statistical Society Series C: Applied Statistics 73*(1), 221–240.

Edith U Umeh and Chinyere I Ojukwu (2019). Effects of influential outliers in local polynomial techniques (smoothing techniques). *International Journal of Probability and Statistics 8*(1), 19–22.

Febrero, M., P. Galeano, and W. González-Manteiga (2007). A functional analysis of NOx levels: Location and scale estimation and outlier detection. *Computational Statistics 22*(3), 411–427.

Febrero, M., P. Galeano, and W. González-Manteiga (2008). Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels. *Environmetrics 19*(4), 331–345.

Febrero-Bande, M., P. Galeano, and W. GonzÃ¡lez-Manteiga (2010). Measures of influence for the functional linear model with scalar response. *Journal of Multivariate Analysis 101*(2), 327–339.

Fontana, M., G. Zeni, and S. Vantini (2023). Conformal prediction: A unified review of theory and new challenges. *Bernoulli 29*(1), 1–23.

Fraiman, R. and G. Muniz (2001). Trimmed means for functional data. *Test 10*(2), 419–440.

Franco-Villoria, M. and R. Ignaccolo (2017). Bootstrap based uncertainty bands for prediction in functional Kriging. *Spatial Statistics 21*, 130–148.

Genz, A. and F. Bretz (2009). *Computation of multivariate normal and t probabilities*. Number 195 in Lecture notes on statistics. Springer.

Gijbels, I. and S. Nagy (2017). On a General Definition of Depth for Functional Data. *Statistical Science 32*(4), 630 – 639.

Gneiting, T. and A. E. Raftery (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association 102*(477), 359–378.

Górecki, T., L. Horváth, and P. Kokoszka (2020). Tests of normality of functional data. *International Statistical Review 88*(3), 677–697.

Hahn, M. G. (1977). Conditions for sample-continuity and the Central Limit Theorem. *The Annals of Probability 5*(3), 351–360.

Hampel, F. R. (1971). A general qualitative definition of robustness. *The Annals of Mathematical Statistics 42*(6), 1887–1896.

Hardin, J. and D. M. Rocke (2005). The distribution of robust distances. *Journal of Computational and Graphical Statistics 14*(4), 928–946.

Hastie, T. and R. Tibshirani (1993). Varying coefficient models. *Journal of the Royal Statistical Society: Series B (Methodological) 55*(4), 757–779.

Hellton, K. H., C. LingjÃŠrde, and R. De Bin (2023). Influence of single observations on the choice of the penalty parameter in ridge regression.

Horn, P. S., P. W. Britton, and D. F. Lewis (1988). On the prediction of a single future observation from a possibly noisy sample. *The Statistician 37*(2), 165.

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

Hubert, M., P. J. Rousseeuw, and P. Segaert (2015). Multivariate functional outlier detection. *Statistical Methods & Applications 24*(2), 177–202.

Hubert, M. and E. Vandervieren (2008). An adjusted boxplot for skewed distributions. *Computational Statistics & Data Analysis 52*(12), 5186–5201.

Hyndman, R. J. and M. Shahid Ullah (2007). Robust forecasting of mortality and fertility rates: A functional data approach. *Computational Statistics & Data Analysis 51*(10), 4942–4956.

Kac, M. (1943). A correction to "On the average number of real roots of a random algebraic equation". *Bulletin of the American Mathematical Society 49*(12), 938–939.

Lei, J. and L. Wasserman (2014). Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology 76*(1), 71–96.

Lenhoff, M. W., T. J. Santner, J. C. Otis, M. G. Peterson, B. J. Williams, and S. I. Backus (1999). Bootstrap prediction and confidence bands: A superior statistical method for analysis of gait data. *Gait & Posture 9*(1), 10–17.

Liebl, D. and M. Reimherr (2023). Fast and fair simultaneous confidence bands for functional parameters. *Journal of the Royal Statistical Society Series B: Statistical Methodology 85*(3), 842–868.

Long, J. P. and J. Z. Huang (2015). A study of functional depths. *arXiv preprint arXiv:1506.01332*.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme". *Biochimica et Biophysica Acta (BBA) - Protein Structure 405*(2), 442–451.

National Center for Chronic Disease Prevention and Health Promotion (2021). Measure your blood pressure.

Ojo, O., R. E. Lillo, and A. F. Anta (2021). Outlier detection for functional data with R package fdaoutlier. *arXiv preprint arXiv:2105.05213*.

Ojo, O. T., A. Fernández Anta, R. E. Lillo, and C. Sguera (2022). Detecting and classifying outliers in big functional data. *Advances in Data Analysis and Classification 16*(3), 725–760.

Olshen, R. A., E. N. Biden, M. P. Wyatt, and D. H. Sutherland (1989). Gait analysis and the bootstrap. *The Annals of Statistics 17*(4), 1419–1440.

Paparoditis, E. and H. L. Shang (2023). Bootstrap prediction bands for functional time series. *Journal of the American Statistical Association 118*(542), 972–986.

Pittman, R. (2022). Using concurrent functional regression to reconstruct river stage data during flood events and identify influential functional measurements.

Pope, A. J. (1976). *The Statistics of Residuals and the Detection of Outliers*. US Department of Commerce, National Oceanic and Atmospheric Administration.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Ramsay, J. O. and B. W. Silverman (2005). *Functional Data Analysis* (2nd ed ed.). Springer series in statistics. Springer.

Rathnayake, L. N. and P. K. Choudhary (2016). Tolerance bands for functional data. *Biometrics 72*(2), 503–512.

Rice, S. O. (1945). Mathematical analysis of random noise. *Bell System Technical Journal 24*(1), 46–156.

Rousseeuw, P. J. and K. V. Driessen (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics 41*(3), 212–223.

Rousseeuw, P. J. and A. Leroy (1987). Robust regression and outlier detection. *John Wiley & Sons*.

Rousseeuw, P. J., J. Raymaekers, and M. Hubert (2018). A measure of directional outlyingness with applications to image data and video. *Journal of Computational and Graphical Statistics 27*(2), 345–359.

Seber, G. A. F. and A. J. Lee (2003). *Linear Regression Analysis* (2nd ed ed.). Wiley Series in Probability and Statistics. J. Wiley.

Seo, S. (2002). A review and comparison of methods for detecting outliers in univariate data sets. pp. 59.

Shen, Q. and H. Xu (2007). Diagnostics for linear models with functional responses. *Technometrics 49*(1), 26–33.

Singh, R. S. (1988). Estimation of error variance in linear regression models with errors having multivariate student-t distribution with unknown degrees of freedom. *Economics Letters 27*(1), 47–53.

Sun, Y. and M. G. Genton (2011). Functional boxplots. *Journal of Computational and Graphical Statistics 20*(2), 316–334.

Taboga, P., O. N. Beck, and A. M. Grabowski (2020). Prosthetic shape, but not stiffness or height, affects the maximum speed of sprinters with bilateral transtibial amputations. *PLOS ONE 15*(2), e0229035.

Telschow, F. J. and A. Schwartzman (2022). Simultaneous confidence bands for functional data using the Gaussian kinematic formula. *Journal of Statistical Planning and Inference 216*, 70–94.

Torti, A., A. Pini, and S. Vantini (2020). Modelling time-varying mobility flows using function-on-function regression: Analysis of a bike sharing system in the city of milan. *Journal of the Royal Statistical Society 70*(1), 226–247.

Tukey, J. W. (1975). Mathematics and the picturing of data. *Proceedings of the International Congress of Mathematicians 2*, 523–531.

Tukey, J. W. and P. A. Tukey (1985). Computer graphics and exploratory data analysis: An introduction. In *Proceedings of the sixth annual conference and exposition: computer graphics*, Volume 85, pp. 773–785.

United Nations, Department of Economic and Social Affairs, and Population Division (2016). World population prospects: The 2015 revision.

Vinue, G. and I. Epifanio (2021). Robust archetypoids for anomaly detection in big functional data. *Advances in Data Analysis and Classification 15*(2), 437–462.

Vovk, V. and G. Shafer (2008). A tutorial on conformal prediction. *The Journal of Machine Learning Research 9*, 371–421.

Wang, Y., G. Wang, L. Wang, and R. T. Ogden (2020). Simultaneous confidence corridors for mean functions in functional data analysis of imaging data. *Biometrics 76*(2), 427–437.

Wardhani, N. W. S., M. Y. Rochayani, A. Iriany, A. D. Sulistyono, and P. Lestantyo (2019). Cross-validation metrics for evaluating classification performance on imbalanced data. In *2019 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 14–18. IEEE.

Welsch, R. E. and E. Kuh (1977). Linear regression diagnostics. Technical report, National Bureau of Economic Research.

Wilkinson, L., A. Anand, and R. Grossman (2005). Graph-theoretic scagnostics. In *Information Visualization, IEEE Symposium on*, pp. 21–21. IEEE Computer Society.

Willwacher, S., V. Herrmann, K. Heinrich, J. Funken, G. Strutzenberger, J.-P. Goldmann, B. Braunstein, A. Brazil, G. Irwin, W. Potthast, and G.-P. Brüggemann (2016). Sprint start kinetics of amputee and non-amputee sprinters. *PLOS ONE 11*(11), e0166219.

World Athletics (2020a). *Amendments to Rule 6.3.4 of the Technical Rules*.

World Athletics (2020b). *Competition and Technical Rules*.

Zhang, M. and A. Parnell (2023). Review of clustering methods for functional data. *ACM Trans. Knowl. Discov. Data 17*(7).

Zuo, Y. and R. Serfling (2000). General notions of statistical depth function. *Annals of Statistics 28*(2), 461–482.

Appendix A

New Methods for Functional Outlier Detection

A.1 Additional Figures

A.1.1 Simulation Figures

Figure A.1: Random sample generated from Simulation Model "Combined", which generates combined magnitude and shape outliers. The random sample presented has $n = 100$ observations, $T = 100$ sampling points, and outlier rate$\Delta = 0.05$.

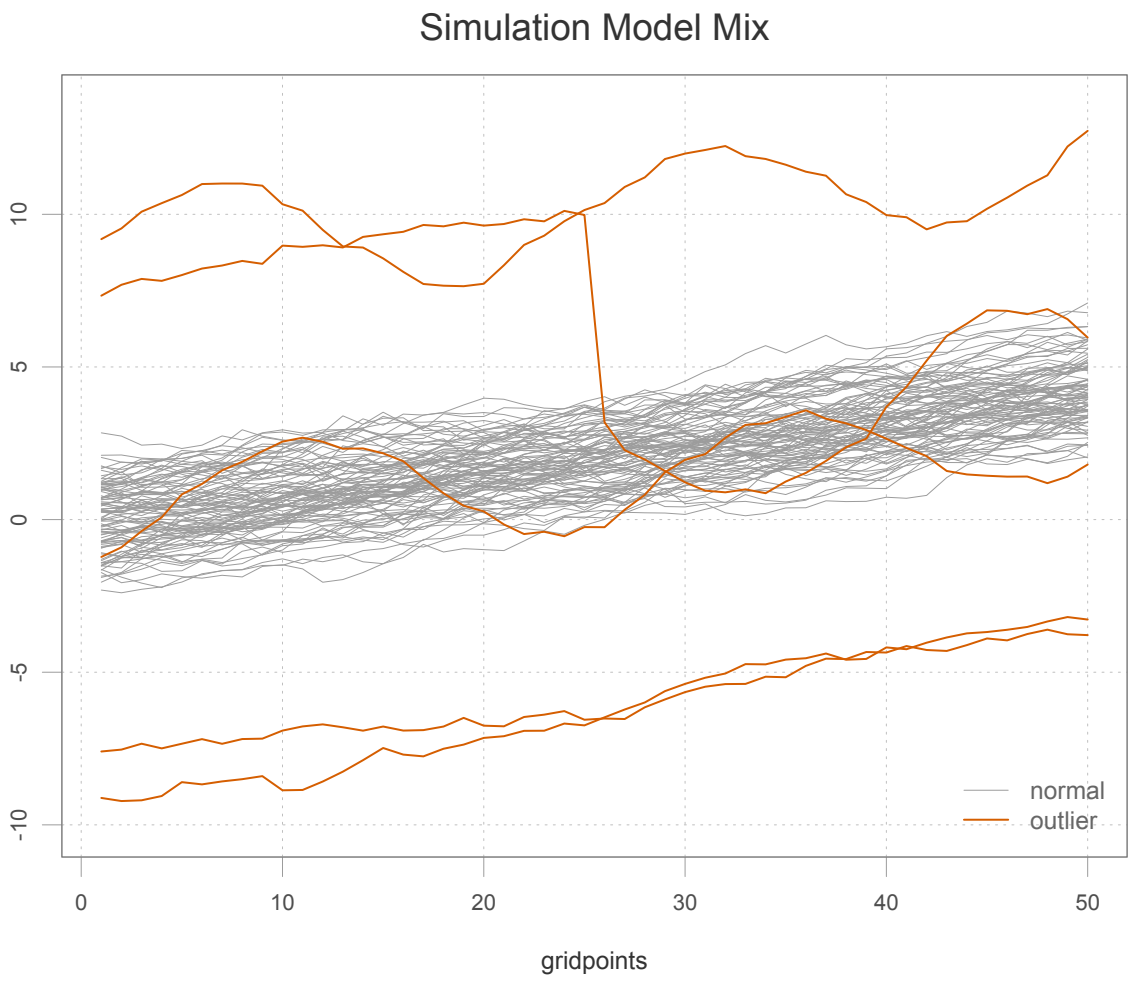


Figure A.2: Random sample generated from Simulation Model "Mix", which generates magnitude outliers, shape outliers, and combined magnitude and shape outliers. The random sample presented has $n = 100$ observations, $T = 100$ sampling points, and outlier rate$\Delta = 0.05$.

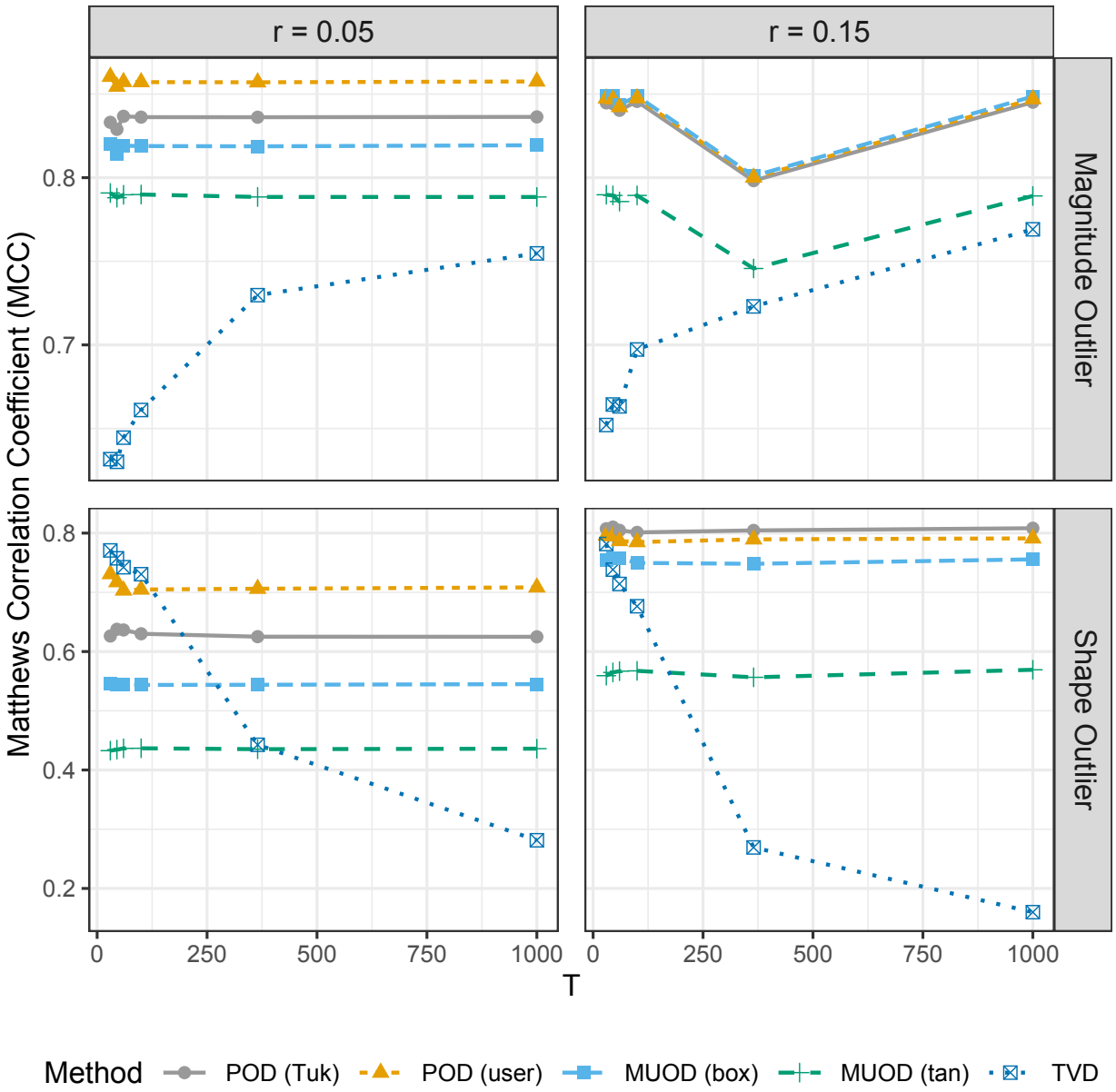


Figure A.3: Average Matthew's Correlation Coefficient (MCC) for classifying magnitude and shape outliers on a varying number of sampling points ($T = 30, 45, 60, 100, 365, 1000$), faceted on the proportion of outliers ($\Delta = 0.05, 0.15$), and colored by the method type (POD (Tuk), POD (user), MS-Plot, MUOD (box), and TVD.

A.2 Practical Outlier Detection (POD)

A.2.1 Development of the Method

A.2.1.1 Deciding the Number of Intervals

The first priority in developing our practical outlier method was to minimize the number of tuning parameters. Specifically, the number of intervals used to bin the sampling points of each observation, had a direct effect on the performance of the method. Rather than allowing a user to specify the number of intervals, it was decided to optimize the algorithm in terms of accuracy, precision, recall (or sensitivity), etc. with respect to the number of intervals used to bin the data. When developing the simulation, the whole domain, one interval, two intervals, ..., up to a number of intervals that would allow at least three sampling points per interval, were all decided. For each number of intervals, a sequence of 30 threshold values, using δ from 0.01 to 0.99, was used for deciding the outlying observations. Using a sequence of thresholds allowed the calculation of Precision-Recall Area Under the Curve ($PR - AUC$) to compare the performance of each number of intervals used. Precision-Recall was preferred over the traditional Receiver-Operating-Characteristic curve because outliers tend to represent a small proportion of total observations.

The first consideration was to look for patterns in summary statistics collected on the whole domain, with respect to how many intervals would be the optimal choice for maximizing $PR - AUC$. Then, the data would drive the methodology, without direct user influence. Clear patterns when investigating trends of summary statistics against $PR - AUC$ and the number of intervals used were not observed. However, patterns in the number of intervals used and measurements of classification diagnostics (precision, recall, selectivity, etc.) were observed. In general, as the number of intervals used to bin the data increases, the number of outliers identified also increases. In all nine simulation models, the Practical Outlier method improved when using two intervals versus the whole domain, and when using three intervals versus two intervals. Yet, for several simulation settings, after reaching enough intervals used, the performance of Practical Outlier would drop off: precision, recall, and accuracy all decreased. The unexpected drop in performance

led to further investigation of interval sizes based off the number of sampling points, T. Further investigation ultimately led to the interval rule described in Section 2.2.1.

A second simulation was conducted to decide between the use of 15 intervals, 20 intervals, and two techniques of combining their results (when T $\geq$ 60). One way to combine the results is to identify a list of outliers using 15 intervals, identify a list using 20 intervals, then combine the two lists as the final list of outliers. The second technique is to implement the method with 15 intervals and save the data frame containing the total count of each observation's "out" features. Repeat the same steps with 20 intervals used. Then, sum up the total count of "out" features found with 15 and with 20 intervals. Lastly, use this data frame of counts to identify the outliers.

In the simulation comparing combination of results, it was found that combining the counts of "out" features, before identifying the outliers with a threshold cutoff, outperformed the first technique of identifying two separate lists of outliers first, and then combining.

A.2.1.2 Determining Extreme Summary Statistics

The collection of summary statistics considered for POD do not all have symmetric sampling distributions. For example, if the population of a random sample is normally distributed, the sampling distribution of the sample variance will be χ^2, which is right-skewed. Therefore, in order to find the extreme values of a summary statistic, it seems necessary to use a rule that is robust to skewed distributions. Seo (2002) reviews and compares common methods in outlier detection for non-functional, univariate data sets. Of particular interest, Hubert and Vandervieren (2008) introduced the adjusted boxplot, which alters Tukey's original boxplot rule by leveraging a robust measure of skewness, called the medcouple. Medcouple was first introduced and compared to other robust measures of skewness (namely, quartile skewness (QS) and octile skewness (OS)) by Brys et al. (2004). When compared, the medcouple has the sensitivity of OS to detect skewness and the robustness of QS towards outliers that are present (Seo, 2002). It is also important to note that the medcouple is a bounded measure of skewness with a breakdown value of 25% (Brys et al., 2004). That is, it would require that 25%, or more, of the observations are replaced with outlying values, before medcouple becomes useless (Rousseeuw and Leroy, 1987).

To calculate the adjusted boxplot "fences," the first and third quartiles, the inner quartile range, and the medcouple must all be estimated first. Denote the first and third quartiles, for each summary statistic and each interval, as $Q_1\left\{S_{type}\left(\text{Int}_a\right)\right\}$ and $Q_3\left\{S_{type}\left(\text{Int}_a\right)\right\}$. The medcouple is estimated with respect to each summary statistic and each interval (resulting in 90 measures of medcouple). Denote the medcouple as $MC\left\{S_{type}\left(\text{Int}_a\right)\right\}$, which is defined as $MC\left\{S_{type}\left(\text{Int}_a\right)\right\} :=$

$$\text{med}\left\{\frac{\left(\left[S_{type}\left(\text{Int}_a\right)\right]_i - \text{med}\left\{S_{type}\left(\text{Int}_a\right)\right\}\right) - \left(\text{med}\left\{S_{type}\left(\text{Int}_a\right)\right\} - \left[S_{type}\left(\text{Int}_a\right)\right]_{i'}\right)}{\left[S_{type}\left(\text{Int}_a\right)\right]_{i'} - \left[S_{type}\left(\text{Int}_a\right)\right]_i}\right\},$$

for $\left[S_{type}\left(\text{Int}_a\right)\right]_i \leq \text{med}\left\{S_{type}\left(\text{Int}_a\right)\right\} \leq \left[S_{type}\left(\text{Int}_a\right)\right]_{i'}$.

The "fences" for the adjusted boxplot, with respect to each summary statistic and interval, are denoted as $(L, U)\left\{S_{type}\left(\text{Int}_a\right)\right\}$. Each fence is defined as

$$\begin{cases} (Q_1 - 1.5 \cdot \exp\left(-3.5 \cdot MC\right) \cdot IQR, Q_3 + 1.5 \cdot \exp\left(4 \cdot MC\right) \cdot IQR,) & \text{if } MC \geq 0 \\ (Q_1 - 1.5 \cdot \exp\left(-4 \cdot MC\right) \cdot IQR, Q_3 + 1.5 \cdot \exp\left(3.5 \cdot MC\right) \cdot IQR,) & o.w. \end{cases},$$

where Q_1, Q_3, and MC are calculated with respect to same summary statistic and interval, $\left\{S_{type}\left(\text{Int}_a\right)\right\}$. Note that the adjusted boxplot is equivalent to Tukey's classical boxplot, when the medcouple is zero (symmetric distribution).

Three different methods were investigated for deciding the extreme summary statistics: use of Tukey's classical boxplot for all statistics, use of adjusted boxplot for all statistics, and the use of Tukey's classical boxplot for roughly symmetric sampling distributions, and otherwise an adjusted boxplot. Through statistical theory, backed by an empirical study, the sampling distributions of the mean, minimum, maximum, median, AUC, and coefficient of variance were found to be symmetric, while the sampling distributions of the variance, range, and roughness were found to be positively skewed.

Of all three methods, it was found that using Tukey's classical boxplot for all summary statistics outperformed using an adjusted boxplot for all summary statistics and using a combination of Tukey's classical and adjusted boxplot.

A.2.1.3 Classifying the Type of Outlier

In Section 2.2.1, it was mentioned that the measures of location would be ideal for identifying magnitude outliers, while the measures of dispersion and roughness would be ideal for identifying shape (and amplitude) outliers. Simulation studies were used to observe the frequency of extreme summary statistics for each observation identified as a functional outlier by POD, and then to identify patterns in the count of extreme summary statistics found.

Four different data generation models were considered: a model with magnitude outliers only (see Model 1, Figure 2.1), a model with shape outliers only (see Model 7, Figure 2.1), a model with combined (magnitude and shape) outliers only, and a model with all types present (see Figure A.2). The model with combined outliers only is similar to Figure A.2, without the strictly magnitude and/or shape outliers present. Each model was simulated using all combinations of sample size $n = 30, 45, 60$, number of sampling points $T = 30, 100, 250$, rate of outliers $\Delta = 0.05, 0.15$, and covariance roughness $\beta = 0.1, 0.5, 0.9$. Each unique combination of model, sample size (n), number of sampling points (T), rate of outliers (Δ), and covariance roughness (β) was simulated 250 iterations. On each iteration, a random sample of functional data is generated by the chosen model with the chosen simulation parameters. POD is implemented to identify the extreme statistics of every observation on every interval. Only the data pertaining to the true functional outliers in the sample, as identified by the simulation model, is retained.

Investigation of the results found a pattern related to the count of extreme summary statistics per interval and the class of functional outlier. Specifically, the total number of extreme summary statistics in each group, location vs variation, found per interval was computed. For each observation, an interval that has more than two extreme location stats was identified as Magnitude outlying, and an interval that has more than one extreme variation stat was identified as Shape outlying. If at least one third of the intervals were identified as Magnitude outlying, then the functional outlier is

classified as a Magnitude outlier. If at least one fifth of the intervals are identified as Shape outlying, then the functional outlier is classified as a Shape outlier. In rare cases, if a functional outlier has neither one third or more Magnitude outlying intervals nor one fifth or more Shape outlying intervals, then the functional outlier was identified as a Shape outlier. This typically occurs when a functional outlier is strictly Magnitude outlying over one part of the domain and strictly Shape outlying over another part of the domain.

This pattern gave the ability to accurately classify 96.5% of all 373,400 simulated functional outliers. Of the 16,803 simulation functional outliers that were improperly classified, 3116 of them were truly both Shape and Magnitude outlying, but falsely identified as Magnitude outlier only; 2956 of them were truly Magnitude outlying only, but falsely identified as both Shape and Magnitude outlying; 10,722 of them were truly Shape outlying only, but falsely identified as both Shape and Magnitude outlying; and nine of them were truly Shape outlying only, but identified as a Magnitude outlier only. Since this pragmatic rule was able to reach more than 95% accuracy, it was chosen to be implemented in the methodology of POD for classification of type.

A.3 Prediction Band Outlier Detection (PBOD)

A.3.1 Simulated Coverage Probabilities for Simultaneous Prediction Bands

The coverage probabilities of the simultaneous prediction bands were assessed via simulation study. A total of nine different data generation methods were considered: three different population mean structures and three different population covariance structures (further details can be found in Liebl and Reimherr (2023)). Each data generation model was used for $n = 15$ and $n = 100$, $T = 101$, and stochastic Student's t errors. On each iteration, a simultaneous prediction band was made assuming Gaussian errors and assuming Student's t errors.

The results confirm the expected behavior. At the smaller sample size, only the prediction band assuming Student's t errors meets the nominal coverage probability. In all other cases, the simulta-

neous prediction band using Gaussian errors and the simultaneous prediction band using Student's t errors both meet nominal coverage levels. Yet, the nominal coverage level of the simultaneous prediction band using Student's t errors is larger on average than the simultaneous prediction band using Gaussian errors.

A.3.2 Development of the Method

Similar to development of the practical outlier method, parameters for functional outlier detection through FFSPB's were first optimized in terms of $PR - AUC$. Given sample size n, the method showed the best performance, when the number of resampling iterations for the first step is equivalent to the sample size, n, and the number of resampling iterations for the second step is equivalent to twice the sample size, $2n$. The optimal number of iterations was found by testing different multiples of the sample size for the number of resampling iterations in each step (e.g. $0.25n$, $0.5n$, n, $2n$, and $4n$).

In the same simulation, the holdout percentage was optimized for the resampling steps. A range of values from 5% to 50% were considered, and PBOD performed best when using a holdout percentage of 50%.

A.4 Supplementary Material

GitHub Repository:

`https://github.com/creutzml/practical_outlier_detection`

All relevant code, data, and figures can be found in my GitHub repository and are described in detail in the README.md document.

Publication: This chapter, without the Prediction Band Outlier Detection results, has been submitted for publication to The American Statistician, General under the name "Practical Outlier Detection in Functional Data Analysis" by Creutzinger and Sharp (2024+).

Appendix B

Simultaneous Confidence and Prediction Band Methods for Estimating the Conditional Mean of a Functional Concurrent Regression Model

B.1 Additional Definitions

Definition B.1.1. *A stochastic process* $Z = \{Z(t) \in \mathbb{R},\, t \in [0,1]\}$ *is a* $\mathcal{L}^b$*-Lipschitz process, if there exists a random variable* A *satisfying* $\mathbb{E}[|A|^b] < \infty$*, such that*

$$|Z(t) - Z(t')| \leq A\,|t - t'| \quad \textit{for all} \quad t, t' \in [0,1]$$

and

$$\int_0^1 \sqrt{\log(N([0,1],u))}du < \infty,$$

where $N([0,1],u)$ *is the minimal number of intervals of length* u *needed to cover the domain* $[0,1]$.

Side note: Since $\log(N([0,1],u)) \leq \log(C/u) \leq (C/u) - 1 \leq (C/u)$ for some $0 < C < \infty$, then

$$\begin{aligned}\int_0^1 \sqrt{\log(N([0,1],u))}du &\leq C\int_0^1 u^{-1/2}du \\ &= C[2u^{1/2}]_0^1 \\ &= 2C < \infty\end{aligned}$$

B.2 Additional Figures and Tables

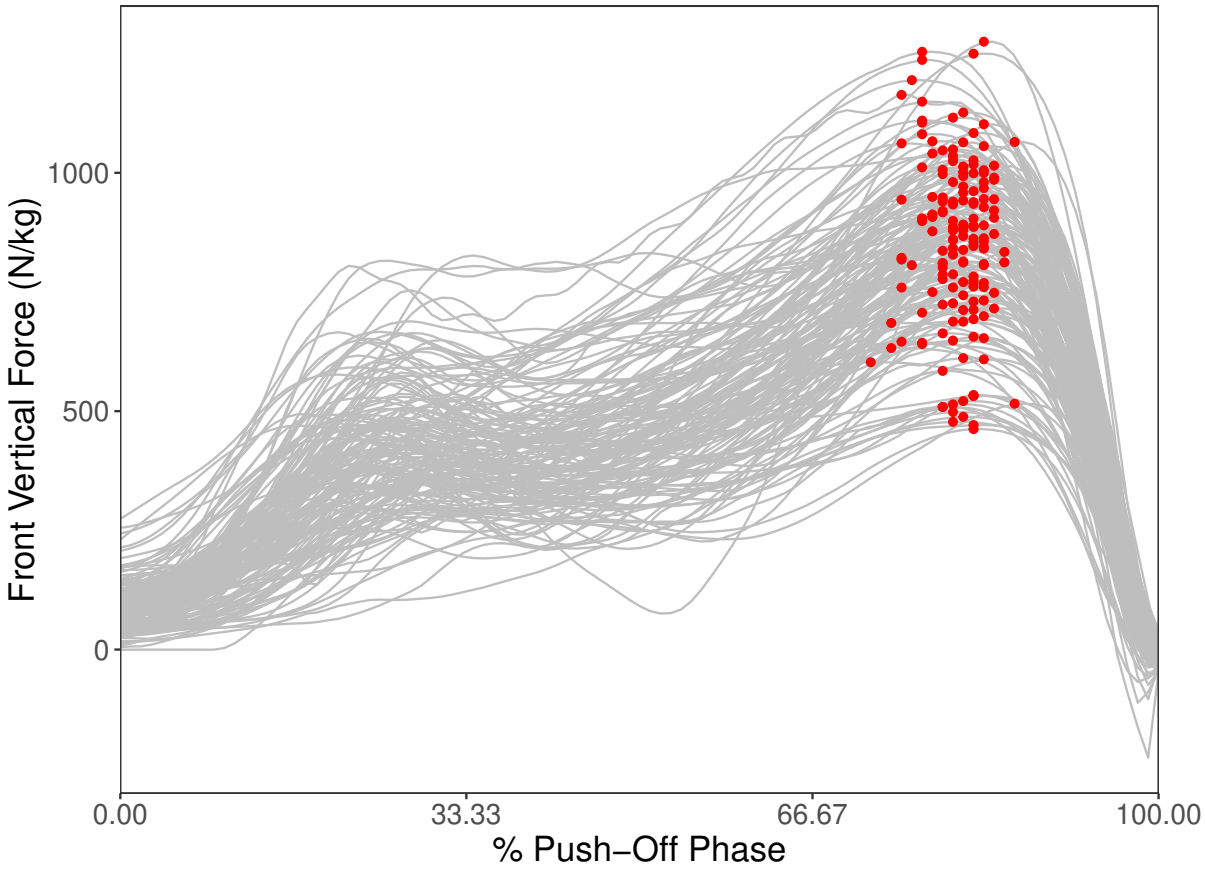


Figure B.1: A plot of the original vertical force $Y(t)$ for each sprinter, with their maximum vertical force plotted in red.

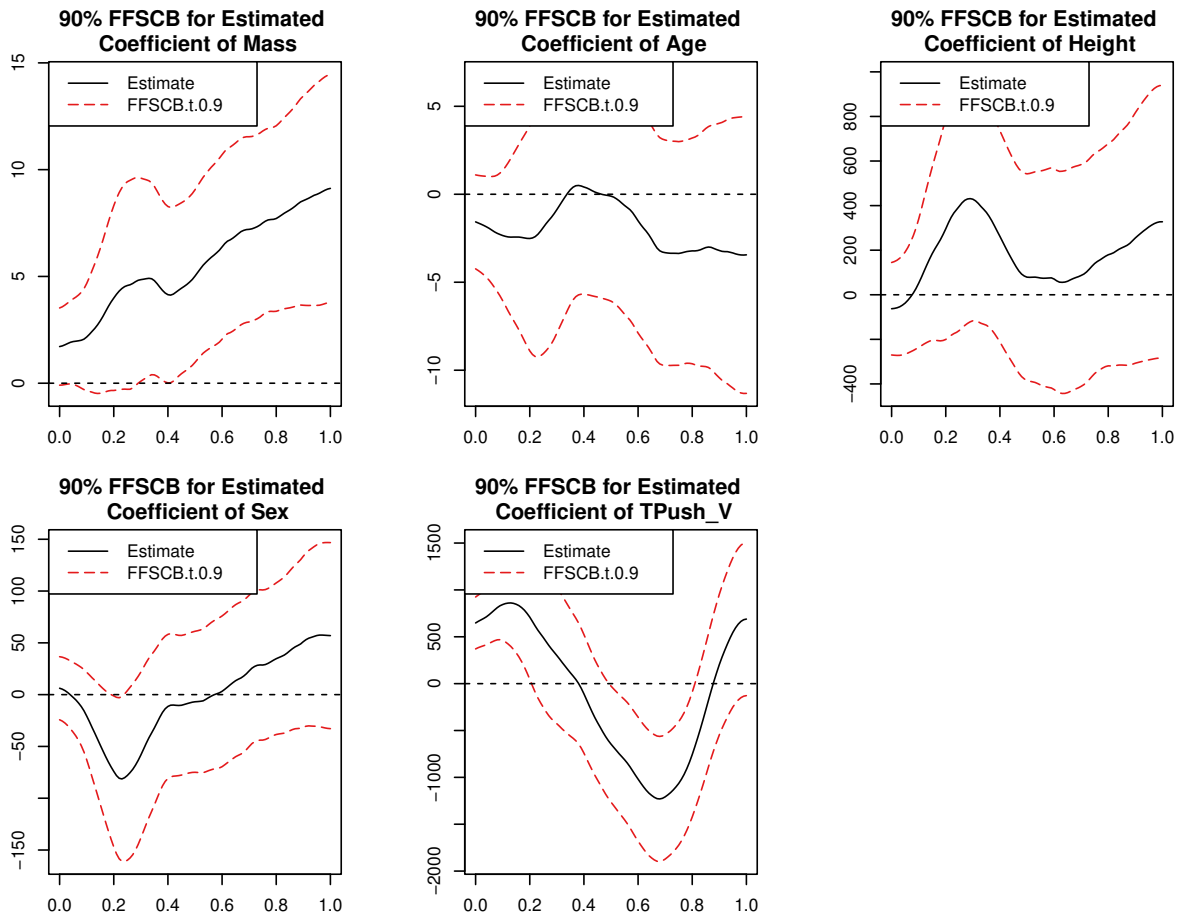


Figure B.2: 90% simultaneous confidence bands for the estimated coefficients of the functional concurrent regression model. The bands are made by FFSCBs, using three intervals.

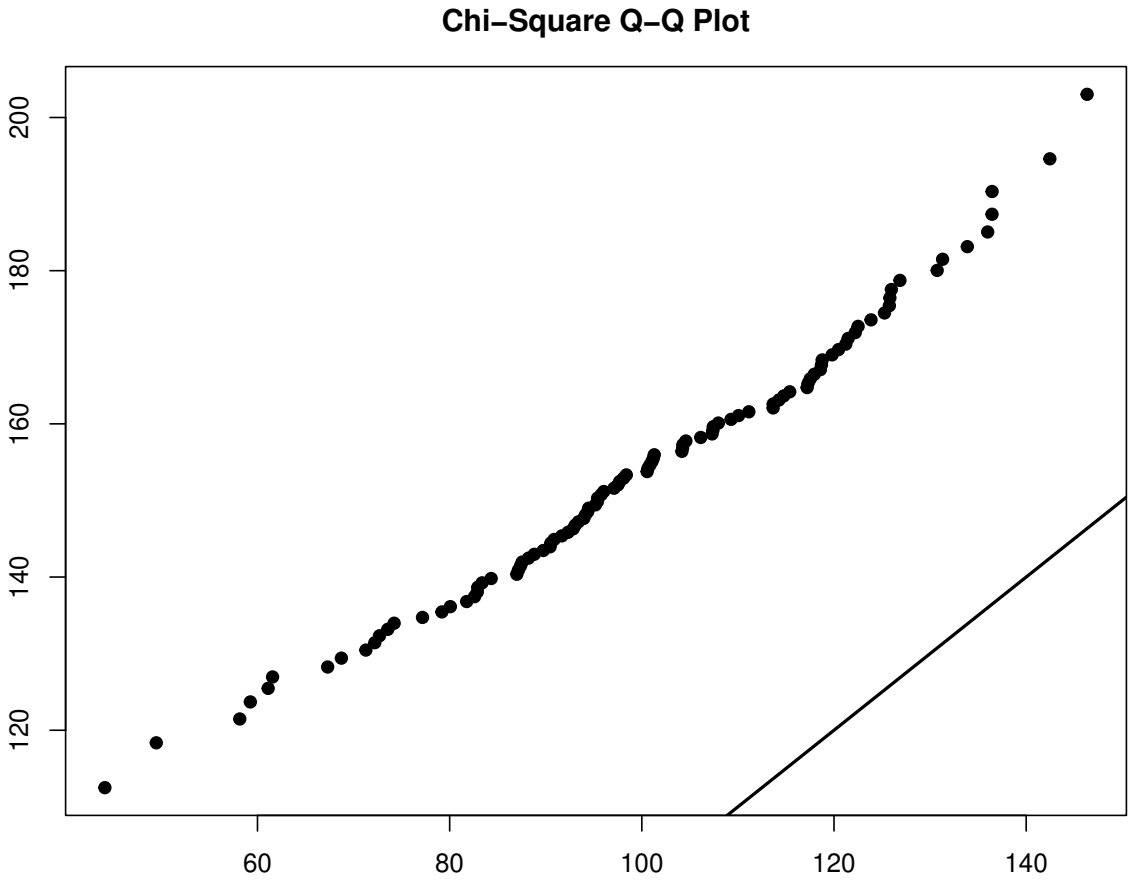


Figure B.3: QQPlot obtained by the Mardia multivariate normality test implemented in R.

Table B.1: Coverage probability of fast and fair simultaneous confidence bands over $[0, 1]$ (with 3 intervals and $\alpha = 0.10$), average max band width, and average band scores across $5,000$ Monte Carlo runs, when using the stationary and non-stationary Matérn covariance.

	Stationary Matérn	Non-Stationary Matérn	Stationary Matérn	Non-Stationary Matérn
	$x_{new}(t) = 0$		$x_{new}(t) = 1$	
	$\nu_0 = 5$			
$n = 30$				
Coverage	0.96	0.97	0.97	0.97
Mean Max Band Width	0.49	0.64	0.49	0.64
Mean Band Score	0.51	0.66	0.50	0.66
$n = 100$				
Coverage	0.97	0.98	0.97	0.98
Mean Max Band Width	0.24	0.32	0.24	0.32
Mean Band Score	0.25	0.33	0.25	0.33
	$\nu_0 = 15$			
$n = 30$				
Coverage	0.97	0.97	0.97	0.97
Mean Max Band Width	0.43	0.57	0.43	0.57
Mean Band Score	0.45	0.58	0.45	0.58
$n = 100$				
Coverage	0.97	0.98	0.97	0.97
Mean Max Band Width	0.22	0.28	0.22	0.28
Mean Band Score	0.23	0.28	0.22	0.29

Table B.2: Local coverage probabilities of fast and fair simultaneous confidence bands (with 3 intervals and $\alpha = 0.10$) over the three sub-intervals, when using a stationary and non-stationary Matérn covariance.

	Stationary Matérn	Non-Stationary Matérn	Stationary Matérn	Non-Stationary Matérn
	$x_{new}(t) = 0$		$x_{new}(t) = 1$	
	$\nu_0 = 5$			
$n = 30$				
$[0, 1/3)$	0.98	0.98	0.98	0.98
$[1/3, 2/3)$	0.98	0.98	0.98	0.99
$[2/3, 1]$	0.98	0.99	0.98	0.99
$n = 100$				
$[0, 1/3)$	0.98	0.99	0.98	0.98
$[1/3, 2/3)$	0.98	0.99	0.98	0.99
$[2/3, 1]$	0.98	0.99	0.98	0.99
	$\nu_0 = 15$			
$n = 30$				
$[0, 1/3)$	0.98	0.98	0.98	0.98
$[1/3, 2/3)$	0.98	0.98	0.98	0.99
$[2/3, 1]$	0.98	0.99	0.98	0.99
$n = 100$				
$[0, 1/3)$	0.98	0.99	0.98	0.98
$[1/3, 2/3)$	0.98	0.99	0.98	0.99
$[2/3, 1]$	0.98	0.99	0.98	0.99

B.3 Supplementary Material

GitHub Repository: `https://github.com/creutzml/FunctionalPrediction`

R-package: The GitHub repository is a forked repository of `R` package `ffscb`. The package contains functions for estimating a fast n fair simultaneous confidence or prediction band for eiter a univariate random sample of functional data or for the parameters of a functional concurrent regression model. `R` package `ffscb` is originally created by Dr. Dominik Liebl and Dr. Matthew Reimherr.

Publication: This chapter has been submitted for publication to the Journal of the American Statistical Association: Case Studies, under the name "Fair Simultaneous Prediction and Confidence Bands for Functional Concurrent Regressions: Comparing Sprinters with Prosthetic versus Biological Legs" by Creutzinger, Liebl, and Sharp (2024+).

Appendix C

Identifying Influential Observations in A Functional Concurrent Regression Model

C.1 Deriving $DFFITS$ in Terms of Externally Studentized Residuals

Continuing from Section 4.2.2, it can be shown that

$$\begin{aligned}\left(X_{(i)}(t)X_{(i)}^T(t)\right)^{-1} &= \left(X(t)X^T(t) - X_i(t)X_i^T(t)\right)^{-1} \\ &= \left(X(t)X^T(t)\right)^{-1} + \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)X_i^T(t) \left(X(t)X^T(t)\right)^{-1}}{1 - h_{ii}(t)},\end{aligned}$$

where the last equality follows from linear algebra properties. It can also be shown that

$$X_{(i)}(t)Y_{(i)}(t) = X(t)Y(t) - X_i(t)Y_i(t).$$

Combining these two results, the least squares estimator of $\hat{\beta}_{(i)}(t)$ (model fitted without observation i), can be expressed as

$$\begin{aligned}
\hat{\beta}_{(i)}(t) &= \left(X_{(i)}(t)X_{(i)}^T(t)\right)^{-1} X_{(i)}(t)Y_{(i)}(t) \\
&= \left[\left(X(t)X^T(t)\right)^{-1} + \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)X_i^T(t)\left(X(t)X^T(t)\right)^{-1}}{1-h_{ii}(t)}\right] \cdot \\
&\quad [X(t)Y(t) - X_i(t)Y_i(t)] \\
&= \left(X(t)X^T(t)\right)^{-1} X(t)Y(t) - \left(X(t)X^T(t)\right)^{-1} X_i(t)Y_i(t) \quad + \\
&\quad \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)X_i^T(t)\left(X(t)X^T(t)\right)^{-1}}{1-h_{ii}(t)} X(t)Y(t) \quad - \\
&\quad \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)X_i^T(t)\left(X(t)X^T(t)\right)^{-1}}{1-h_{ii}(t)} X_i(t)Y_i(t) \\
&= \hat{\beta}(t) - \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)}{1-h_{ii}(t)} \Big[(1-h_{ii}(t))Y_i(t) - X_i^T(t)\left(X(t)X^T(t)\right)^{-1} X(t)Y(t) + \\
&\qquad\qquad X_i^T(t)\left(X(t)X^T(t)\right)^{-1} X_i(t)Y_i(t)\Big] \\
&= \hat{\beta}(t) - \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)}{1-h_{ii}(t)} \Big[Y_i(t) - h_{ii}(t)Y_i(t) - \hat{Y}_i(t) + h_{ii}(t)Y_i(t)\Big] \\
&= \hat{\beta}(t) - \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)e_i(t)}{1-h_{ii}(t)}.
\end{aligned}$$

As a result,

$$\hat{\beta}(t) - \hat{\beta}_{(i)}(t) = \frac{\left(X(t)X^T(t)\right)^{-1} X_i(t)e_i(t)}{1-h_{ii}(t)},$$

and

$$\begin{aligned}
DFFIT_i(t) &= \hat{Y}_i(t) - \hat{Y}_{i(i)}(t) \\
&= X_i^T(t)\left(\hat{\beta}(t) - \hat{\beta}_{(i)}(t)\right) \\
&= \frac{X_i^T(t)\left(X(t)X^T(t)\right)^{-1} X_i(t)e_i(t)}{1-h_{ii}(t)} \\
&= \frac{h_{ii}(t)}{1-h_{ii}(t)} e_i(t).
\end{aligned}$$

Next, substituting this form of $DFFIT_i(t)$ into the numerator for $DFFITS_i(t)$ gives

$$\begin{aligned} DFFITS_i(t) &= \frac{DFFIT_i(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)h_{ii}(t)}} \\ &= \frac{\frac{h_{ii}(t)}{1-h_{ii}(t)}e_i(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)h_{ii}(t)}}. \end{aligned}$$

Finally, the residuals are expressed in terms of the externally studentized residuals,

$$t_{i(i)}(t) = \frac{e_i(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)\,(1-h_{ii}(t))}} \quad \Rightarrow \quad e_i(t) = t_{i(i)}(t)\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)\,(1-h_{ii}(t))},$$

and plugged into the most recent definition of $DFFITS_i(t)$:

$$\begin{aligned} DFFITS_i(t) &= \frac{\frac{h_{ii}(t)}{1-h_{ii}(t)}e_i(t)}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)h_{ii}(t)}} \\ &= \frac{\frac{h_{ii}(t)}{1-h_{ii}(t)}t_{i(i)}(t)\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)\,(1-h_{ii}(t))}}{\sqrt{\hat{\sigma}_{\varepsilon(i)}(t,t)h_{ii}(t)}} \\ &= t_{i(i)}(t)\sqrt{\frac{h_{ii}(t)}{1-h_{ii}(t)}}. \end{aligned}$$

C.2 Additional Simulation Results

C.2.1 Pointwise Simulation

As previously mentioned in the Introduction (see Section 4.1), different methodologies consider different scopes of the sampling domain when searching for influential points or observations. The focus of functional $DFFITS$ is to identify an entire functional observation as influential, rather than just a single, observed point. However, in development of functional $DFFITS$, it is necessary to first assess the pointwise performance of Equation (4.12). A simulation study is conducted by generating a random sample of functional data, concurrently regressing the functional response variable on the functional predictors, computing the functional $DFFITS$ for each observation, and then comparing them to the proposed cutoff (Equation (4.12)). For each sam-

pling point, t_j, if Equation (4.12) is true for observation i, then $Y_i(t_j)$ is identified as an influential point. If observation i is originally generated as an influential observation in the random sample and then identified as an influential point at t_j, then functional observed point $Y_i(t_j)$ is called a true positive (TP). If observation i is originally generated as an influential observation in the random sample and then not identified as an influential point at t_j, then functional observed point $Y_i(t_j)$ is called a false negative (FN). Similarly, if observation i is originally generated as a non-influential ("ordinary") observation in the random sample and then identified as an influential point at t_j, then functional observed point $Y_i(t_j)$ is called a false positive (FP). Lastly, if observation i is originally generated as an ordinary observation in the random sample and then not identified as an influential point at t_j, then functional observed point $Y_i(t_j)$ is called a true negative (TN).

The simulation is conducted using each of the three models, Model 1, Model 2, and Model 3, with multiple sample sizes, $n = 10$, 50, and 100, observed at $T = 1000$ sampling points, different numbers of influential points, $n_{\text{inf}} = 1$, 2, and 3, and varying levels of influentialness, $\lambda = 1$, 1.5, and 2. Equation (4.12) is applied using four values of α, $\alpha = 0.100$, 0.050, 0.010, and 0.005. Note that each of the three models produce an influential observation with different proportions of the sampling domain exhibiting influentialness (see Figure 4.2 (b)). Model 1 produces an influential observation with influential points present for $t_j \in \{(150, 600), (750, 1000]\}$; Model 2 produces an influential observation with influential points present for $t_j \in (125, 1000]$; and Model 3 produces an influential observation with influential points present for $t_j \in (500, 1000]$. Therefore, in Model 1 and Model 3, there are several sampling points at which the influential functional observation is truly *not* influential. As a result, it is expected that the average MCC will be lower at the non-influential sampling points than at the influential sampling points. When $\lambda = 1$, none of the functional observations are generated as true influential observations. Therefore, we expect the accuracy to be constant across the entire sampling domain when $\lambda = 1$ (Note: when no influential points are present, MCC is often NA due to the denominator in Equation (4.17) being zero). In each simulation, the methods were compared using sensitivity, specificity, ACC, PPV, and MCC, as previously defined.

The average MCC of the pointwise simulation for Model 1 can be seen in Figures C.1 ($n = 10$), C.2 ($n = 50$), and C.3 ($n = 100$); for Model 2 can be seen in Figures C.4, C.5, and C.6; and for Model 3 can be seen in Figures C.7, C.8, and C.9. The average MCC is largest for all values of α at sampling points where the influential observations are truly influential (see Figure 4.2 (b)). For example, in Model 1, when $\lambda = 2$ and #Influential $= 1$, the average MCC for $\alpha = 0.005$ is In all scenarios, as λ increases from 1.5 to 2, the average MCC also increases. The average MCC when using Equation (4.12) is associated with the proportion of influential observations present in the random sample. The average MCC is highest across the sampling domain when the proportion of influential observations is approximately between 0.04 and 0.10 (#Influential $= 2$ and 3 in Figure C.2, C.5, or C.8 is the result for a proportion of 0.04 and 0.06, and #Influential $= 1$ in Figure C.1, C.4, or C.7 is the result for a proportion of 0.10). The α quantile used for implementing Equation (4.12) is also associated with the average MCC. When the proportion of influential observations in the sample is large (e.g., 0.20), the use of Equation (4.12) with $\alpha = 0.100$ results in the highest average MCC. When the proportion of influential observations in the sample is small (e.g., 0.01), the use of Equation (4.12) with $\alpha = 0.005$ results in the highest average MCC. To summize, the smaller α is when applying Equation (4.12), the more conservative the results (e.g., as α decreases, the sensitivity decreases and the specificity increases on average).

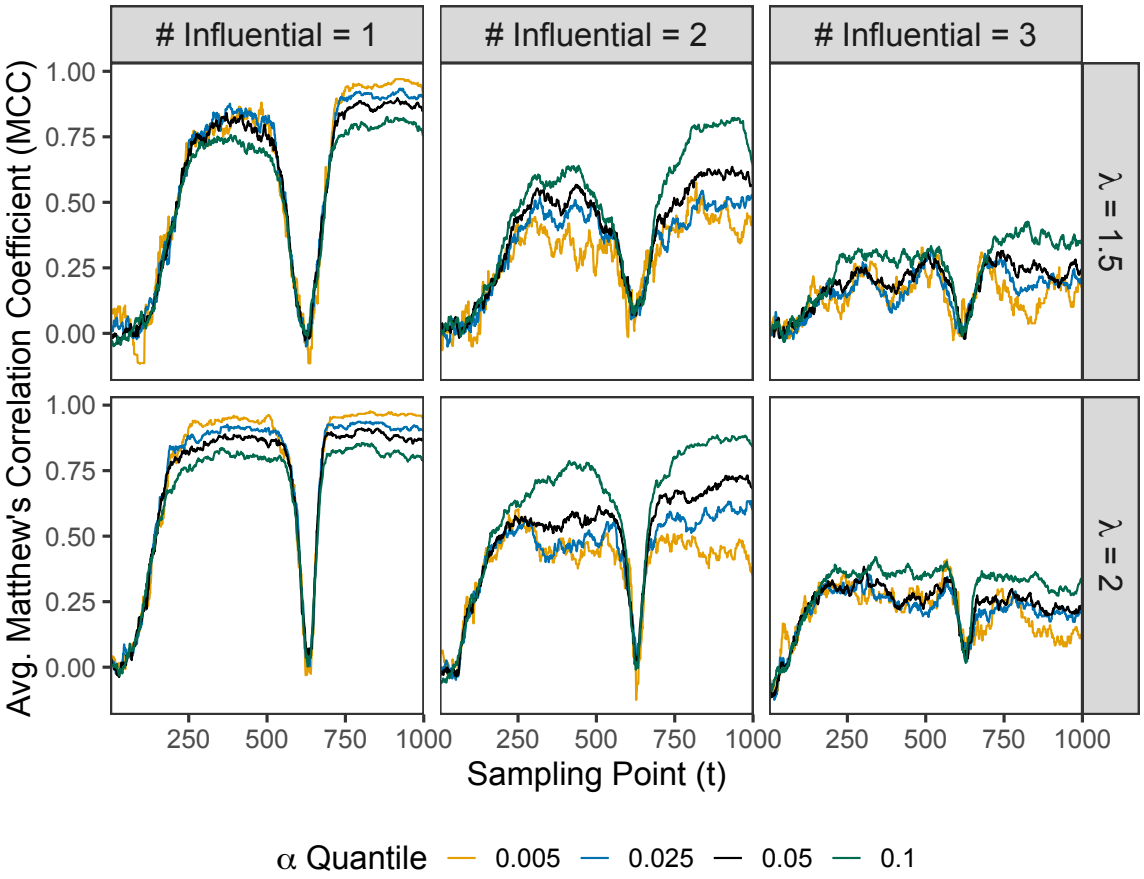


Figure C.1: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 1 and sample size $n = 10$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

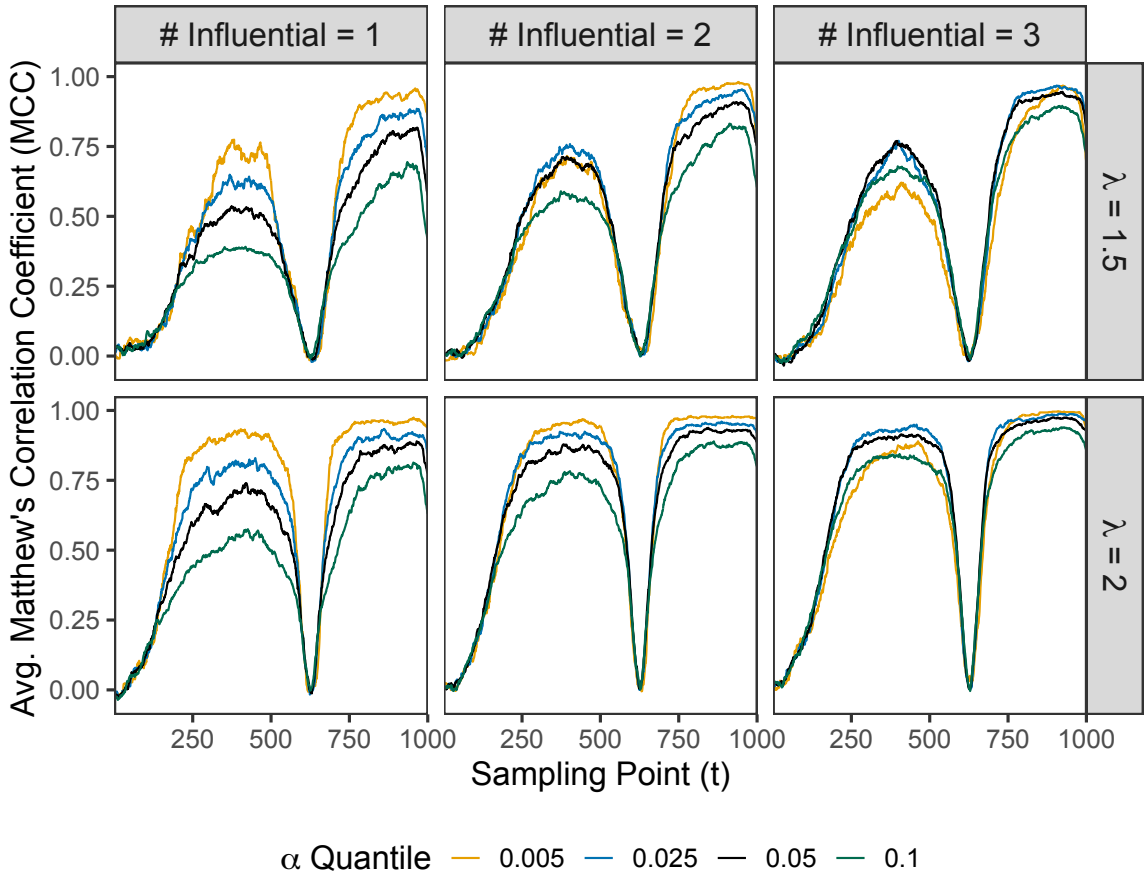


Figure C.2: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 1 and sample size $n = 50$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

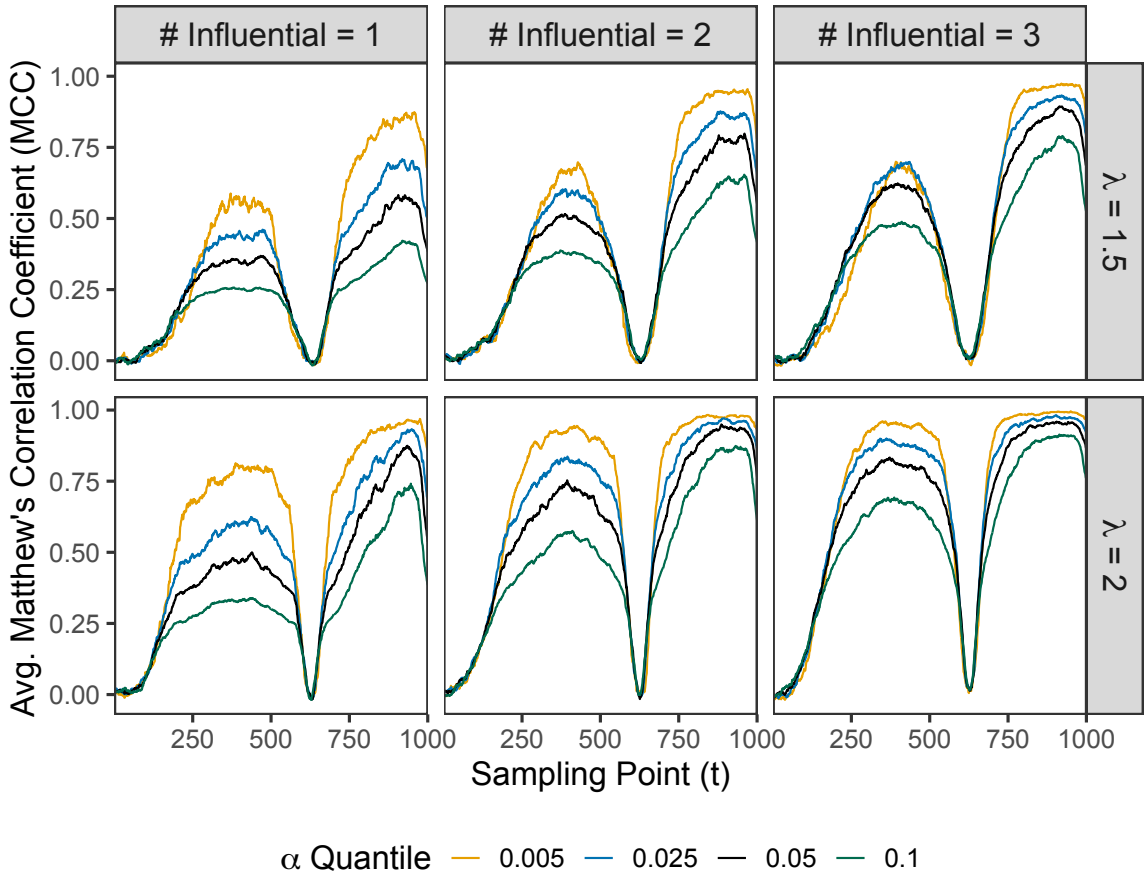


Figure C.3: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 1 and sample size $n = 100$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

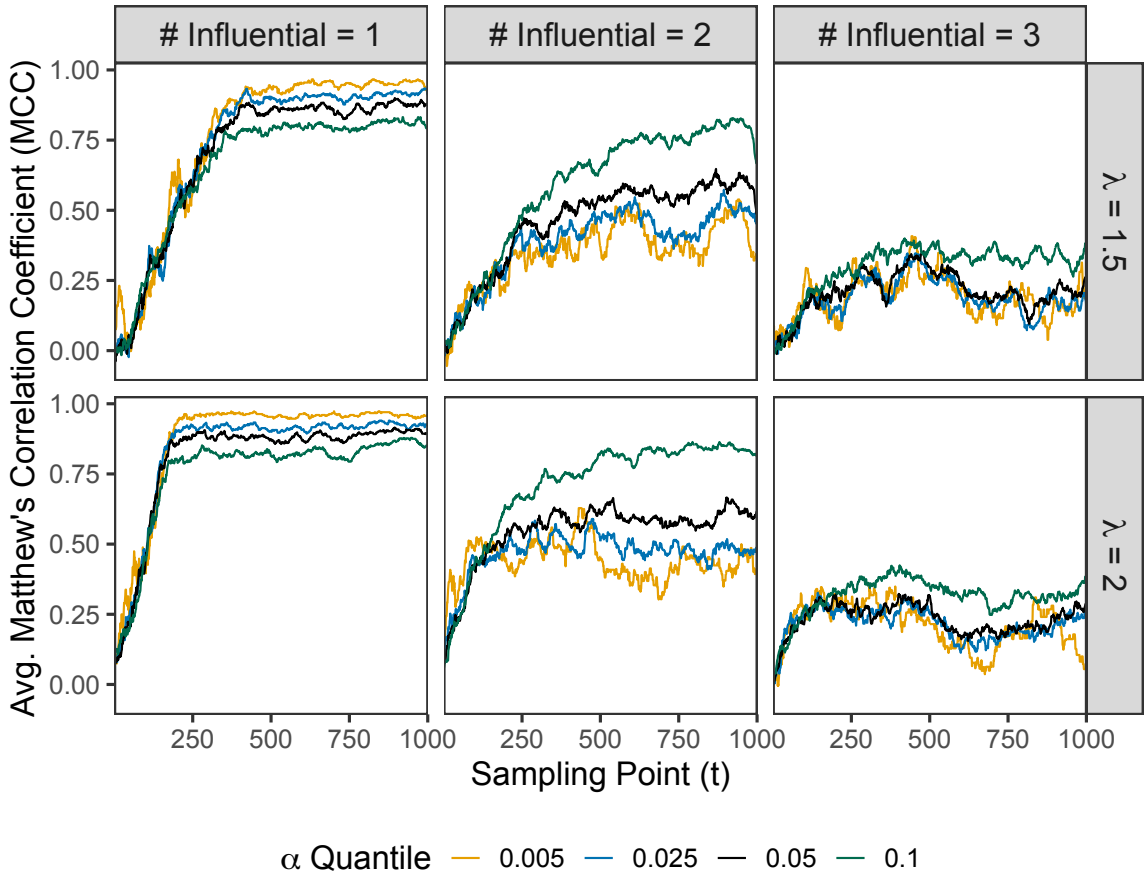


Figure C.4: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 2 and sample size $n = 10$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

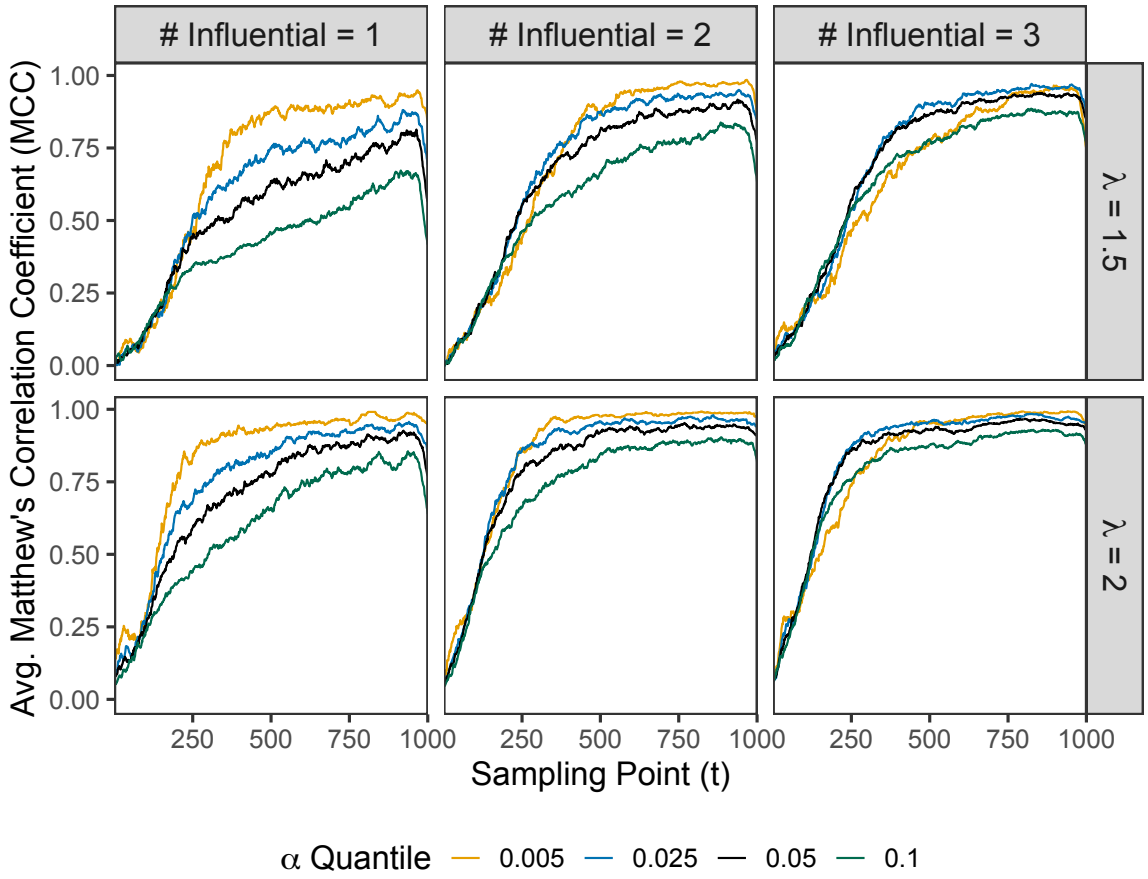


Figure C.5: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 2 and sample size $n = 50$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

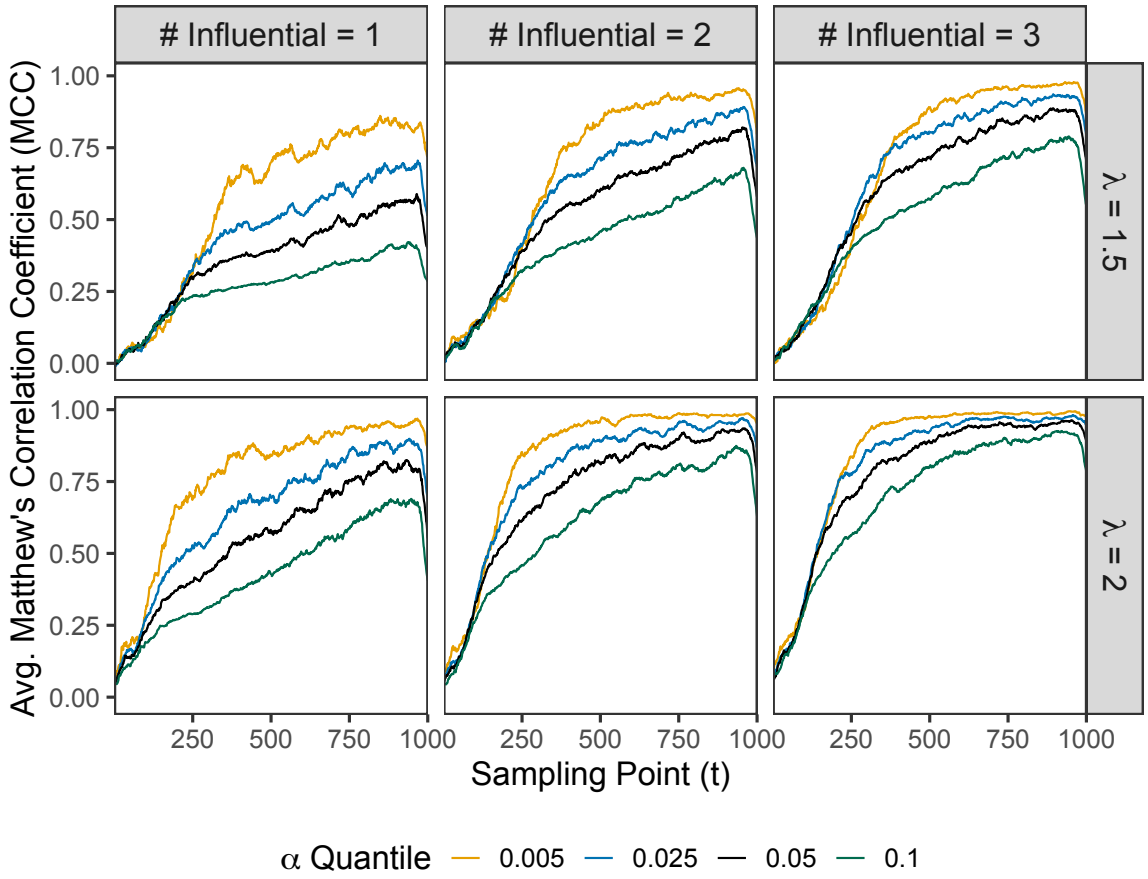


Figure C.6: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 2 and sample size $n = 100$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

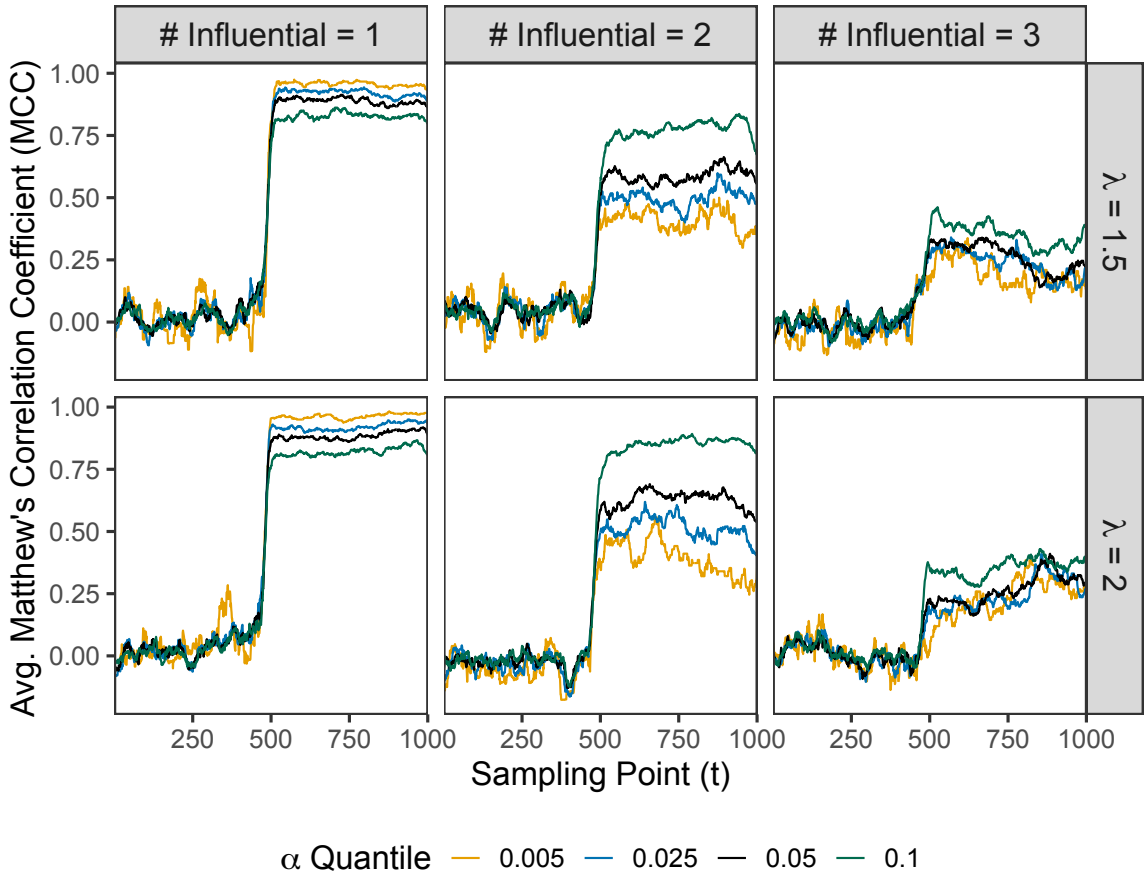


Figure C.7: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 3 and sample size $n = 10$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

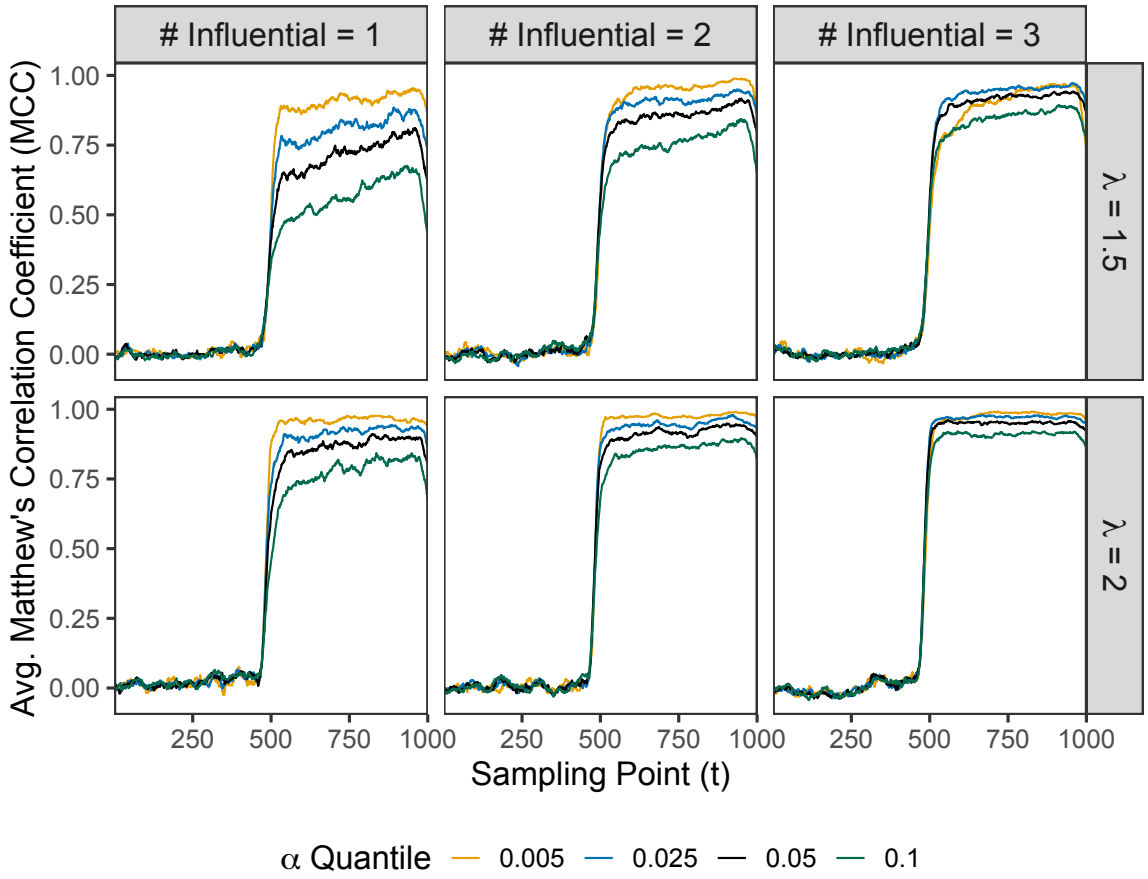


Figure C.8: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 3 and sample size $n = 50$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

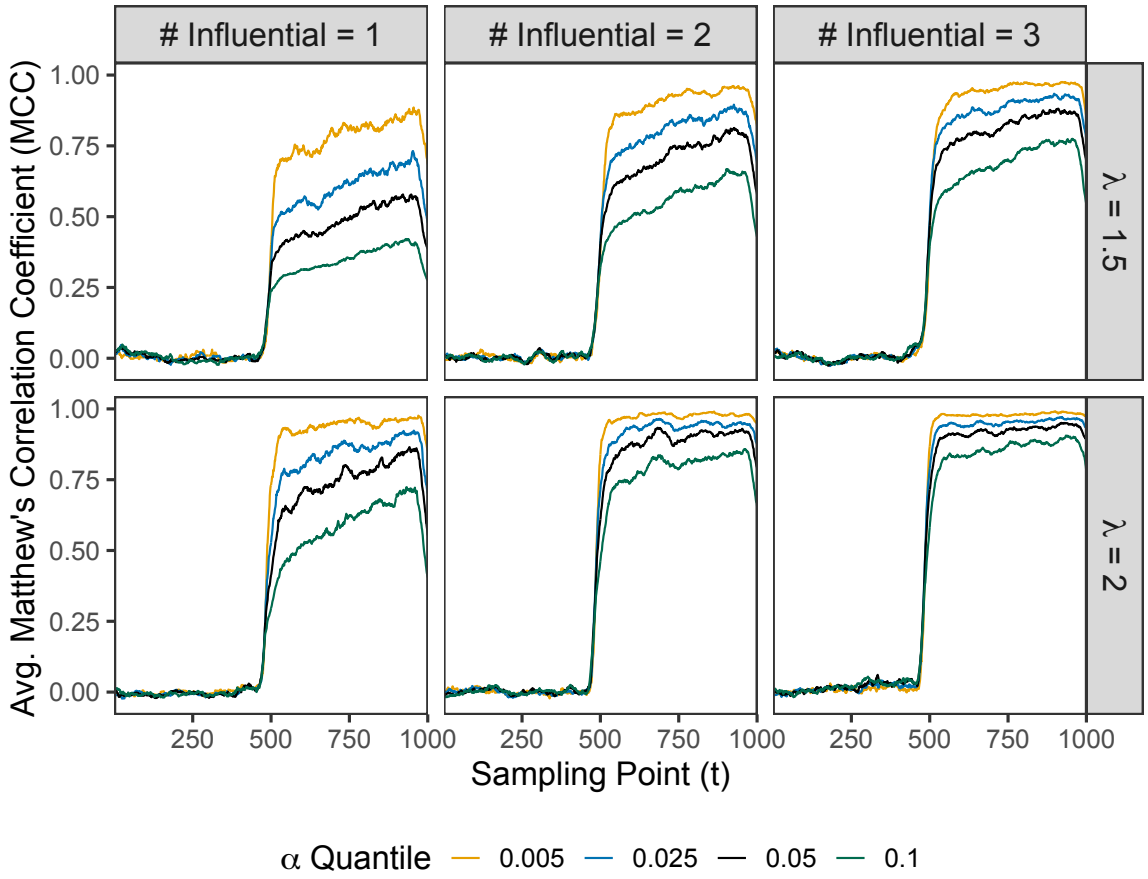


Figure C.9: Pointwise mean Matthew's Correlation Coefficient (MCC) of applying Equation (4.12) for identifying influential functional points, when the data is generated using Model 3 and sample size $n = 100$. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ. Note that when $\lambda = 1$, none of the functional observations are generated as true influential observations, implying that the top row of figures are all equivalent simulations. Lastly, the α quantile used in Equation (4.12) is represented by the color of the line.

C.2.2 Functional Simulation

Several figures of additional simulation results are given below:

Table C.1: The average (s.d.) run time (in seconds), sensitivity, specificity, accuracy (ACC), precision (PPV), and Matthew's Correlation Coefficient (MCC) of Pittman (2022) methodology without (Bootstrapped (raw)) and with smoothing (Bootstrapped (smooth)), for each value of $\alpha = 0.005$, 0.025, 0.050, and 0.100 and for each value of $\alpha_B = 0.00, 0.25, 0.50$. The average results are calculated by averaging over Model 1, Model 2, and Model 3, the sample sizes $n = 10$, 50, and 100, the number of sampling points, $T = 100$ and 1000, the number of influential points $n_{\text{inf}} = 1$, 2, and 3, and the varying magnitude of how influential an observation is, $\lambda = 1$, 1.5, and 2.

$\boldsymbol{\alpha = 0.005}$

Method	α_B	Run Times (s)	Sensitivity	Specificity	Accuracy	PPV	MCC
Bootstrapped (raw)	0.000	44.112 (35.328)	0.330 (0.370)	0.969 (0.045)	0.930 (0.096)	0.359 (0.480)	0.370 (0.435)
Bootstrapped (raw)	0.250	42.356 (36.224)	0.329 (0.369)	0.969 (0.044)	0.930 (0.095)	0.358 (0.479)	0.369 (0.434)
Bootstrapped (raw)	0.500	40.732 (32.883)	0.330 (0.370)	0.969 (0.044)	0.930 (0.095)	0.359 (0.480)	0.371 (0.434)
Bootstrapped (smooth)	0.000	3883.562 (3302.809)	0.112 (0.278)	0.999 (0.004)	0.942 (0.087)	0.453 (0.498)	0.496 (0.420)
Bootstrapped (smooth)	0.250	3413.447 (3066.575)	0.106 (0.284)	0.999 (0.003)	0.943 (0.087)	0.510 (0.500)	0.583 (0.431)
Bootstrapped (smooth)	0.500	3415.135 (3034.031)	0.148 (0.334)	0.999 (0.003)	0.944 (0.087)	0.627 (0.483)	0.688 (0.403)

$\boldsymbol{\alpha = 0.025}$

Method	α_B	Run Times (s)	Sensitivity	Specificity	Accuracy	PPV	MCC
Bootstrapped (raw)	0.000	44.112 (35.328)	0.454 (0.449)	0.958 (0.041)	0.922 (0.093)	0.298 (0.420)	0.389 (0.448)
Bootstrapped (raw)	0.250	42.356 (36.224)	0.453 (0.449)	0.958 (0.041)	0.922 (0.092)	0.297 (0.420)	0.389 (0.447)
Bootstrapped (raw)	0.500	40.732 (32.883)	0.454 (0.449)	0.959 (0.041)	0.922 (0.092)	0.300 (0.421)	0.391 (0.447)
Bootstrapped (smooth)	0.000	3883.562 (3302.809)	0.336 (0.451)	0.987 (0.016)	0.936 (0.085)	0.342 (0.455)	0.484 (0.469)
Bootstrapped (smooth)	0.250	3413.447 (3066.575)	0.337 (0.454)	0.988 (0.014)	0.937 (0.085)	0.339 (0.449)	0.485 (0.466)
Bootstrapped (smooth)	0.500	3415.135 (3034.031)	0.374 (0.472)	0.989 (0.015)	0.941 (0.086)	0.360 (0.454)	0.522 (0.472)

$\boldsymbol{\alpha = 0.050}$

Method	α_B	Run Times (s)	Sensitivity	Specificity	Accuracy	PPV	MCC
Bootstrapped (raw)	0.000	44.112 (35.328)	0.481 (0.462)	0.946 (0.037)	0.911 (0.088)	0.247 (0.371)	0.349 (0.415)
Bootstrapped (raw)	0.250	42.356 (36.224)	0.479 (0.462)	0.946 (0.037)	0.910 (0.088)	0.246 (0.370)	0.347 (0.413)
Bootstrapped (raw)	0.500	40.732 (32.883)	0.481 (0.461)	0.946 (0.037)	0.911 (0.088)	0.249 (0.372)	0.350 (0.414)
Bootstrapped (smooth)	0.000	3883.562 (3302.809)	0.427 (0.482)	0.969 (0.031)	0.924 (0.085)	0.285 (0.394)	0.449 (0.446)
Bootstrapped (smooth)	0.250	3413.447 (3066.575)	0.420 (0.483)	0.971 (0.029)	0.925 (0.085)	0.284 (0.391)	0.451 (0.445)
Bootstrapped (smooth)	0.500	3415.135 (3034.031)	0.424 (0.483)	0.972 (0.029)	0.927 (0.085)	0.293 (0.398)	0.460 (0.450)

$\boldsymbol{\alpha = 0.100}$

Method	α_B	Run Times (s)	Sensitivity	Specificity	Accuracy	PPV	MCC
Bootstrapped (raw)	0.000	44.112 (35.328)	0.498 (0.458)	0.915 (0.036)	0.881 (0.078)	0.192 (0.326)	0.283 (0.360)
Bootstrapped (raw)	0.250	42.356 (36.224)	0.496 (0.456)	0.915 (0.036)	0.881 (0.077)	0.192 (0.327)	0.282 (0.359)
Bootstrapped (raw)	0.500	40.732 (32.883)	0.497 (0.457)	0.916 (0.036)	0.881 (0.077)	0.194 (0.328)	0.284 (0.361)
Bootstrapped (smooth)	0.000	3883.562 (3302.809)	0.491 (0.467)	0.924 (0.048)	0.888 (0.084)	0.211 (0.327)	0.331 (0.383)
Bootstrapped (smooth)	0.250	3413.447 (3066.575)	0.481 (0.468)	0.928 (0.048)	0.890 (0.085)	0.218 (0.332)	0.340 (0.392)
Bootstrapped (smooth)	0.500	3415.135 (3034.031)	0.479 (0.471)	0.931 (0.049)	0.893 (0.086)	0.231 (0.346)	0.360 (0.407)

Table C.2: The average (s.d.) run time (in seconds), sensitivity, specificity, accuracy (ACC), precision (PPV), and Matthew's Correlation Coefficient (MCC) of Theoretical (raw), Theoretical (smooth), Bootstrapped (raw), and Bootstrapped (smooth) using $B = 100$ and $\alpha_B = 0.5$, for each value of $\alpha = 0.005$, 0.025, 0.050, and 0.100. The average results are calculated by averaging over Model 1, Model 2, and Model 3, the sample sizes $n = 10$, 50, and 100, the number of sampling points, $T = 100$ and 1000, the number of influential points $n_{\text{inf}} = 1$, 2, and 3, and the varying levels of influentialness $\lambda = 1$, 1.5, and 2.

$\boldsymbol{\alpha = 0.005}$

Method	Run Times (s)	Sensitivity	Specificity	ACC	PPV	MCC
Theoretical (raw)	0.412 (0.329)	0.394 (0.459)	0.958 (0.039)	0.913 (0.085)	0.217 (0.331)	0.343 (0.389)
Theoretical (smooth)	0.412 (0.329)	0.404 (0.458)	0.947 (0.045)	0.905 (0.088)	0.197 (0.312)	0.312 (0.380)
Bootstrapped (raw)	40.732 (32.883)	0.330 (0.370)	0.969 (0.044)	0.930 (0.095)	0.359 (0.480)	0.371 (0.434)
Bootstrapped (smooth)	3415.135 (3034.031)	0.148 (0.334)	0.999 (0.003)	0.944 (0.087)	0.627 (0.483)	0.688 (0.403)

$\boldsymbol{\alpha = 0.025}$

Method	Run Times (s)	Sensitivity	Specificity	ACC	PPV	MCC
Theoretical (raw)	0.412 (0.329)	0.434 (0.463)	0.931 (0.050)	0.891 (0.088)	0.186 (0.299)	0.299 (0.372)
Theoretical (smooth)	0.412 (0.329)	0.445 (0.460)	0.917 (0.057)	0.879 (0.092)	0.171 (0.281)	0.271 (0.361)
Bootstrapped (raw)	40.732 (32.883)	0.454 (0.449)	0.959 (0.041)	0.922 (0.092)	0.300 (0.421)	0.391 (0.447)
Bootstrapped (smooth)	3415.135 (3034.031)	0.374 (0.472)	0.989 (0.015)	0.941 (0.086)	0.360 (0.454)	0.522 (0.472)

$\boldsymbol{\alpha = 0.050}$

Method	Run Times (s)	Sensitivity	Specificity	ACC	PPV	MCC
Theoretical (raw)	0.412 (0.329)	0.455 (0.462)	0.914 (0.057)	0.876 (0.090)	0.173 (0.284)	0.278 (0.362)
Theoretical (smooth)	0.412 (0.329)	0.467 (0.458)	0.897 (0.065)	0.862 (0.096)	0.159 (0.263)	0.254 (0.348)
Bootstrapped (raw)	40.732 (32.883)	0.481 (0.461)	0.946 (0.037)	0.911 (0.088)	0.249 (0.372)	0.350 (0.414)
Bootstrapped (smooth)	3415.135 (3034.031)	0.424 (0.483)	0.972 (0.029)	0.927 (0.085)	0.293 (0.398)	0.460 (0.450)

$\boldsymbol{\alpha = 0.100}$

Method	Run Times (s)	Sensitivity	Specificity	ACC	PPV	MCC
Theoretical (raw)	0.412 (0.329)	0.481 (0.458)	0.889 (0.066)	0.855 (0.095)	0.160 (0.264)	0.257 (0.348)
Theoretical (smooth)	0.412 (0.329)	0.495 (0.454)	0.870 (0.076)	0.839 (0.101)	0.149 (0.244)	0.237 (0.334)
Bootstrapped (raw)	40.732 (32.883)	0.497 (0.457)	0.916 (0.036)	0.881 (0.077)	0.194 (0.328)	0.284 (0.361)
Bootstrapped (smooth)	3415.135 (3034.031)	0.479 (0.471)	0.931 (0.049)	0.893 (0.086)	0.231 (0.346)	0.360 (0.407)

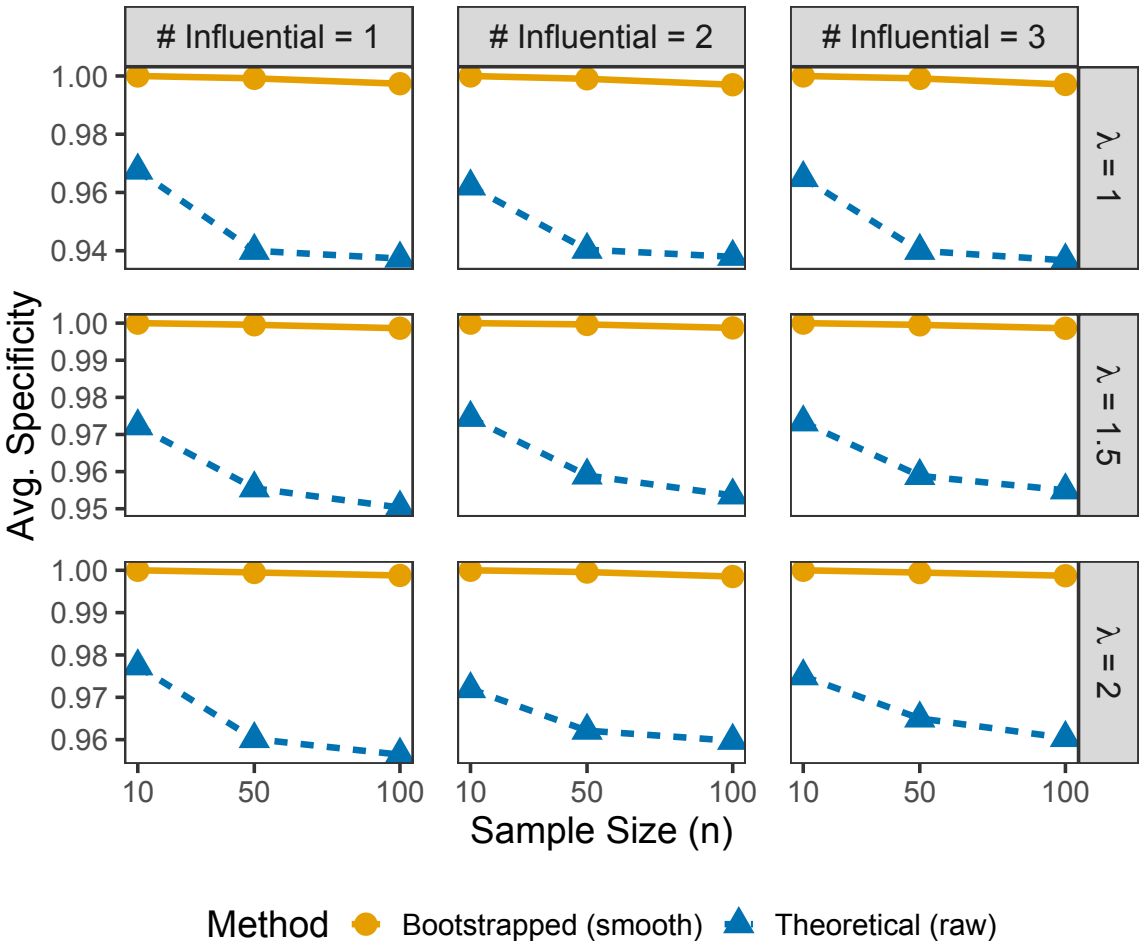


Figure C.10: Average specificity of Theoretical (raw) method implemented with $\alpha = 0.005$ (solid line with circle points) and Bootstrapped (smooth) implemented with $B = 100$, $\alpha_B = 0.5$, and $\alpha = 0.025$ (dashed line with triangle points). The average specificity is calculated by averaging over the models, Model 1, Model 2, and Model 3 and the number of sampling points, $T = 100$ and 1000. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ.

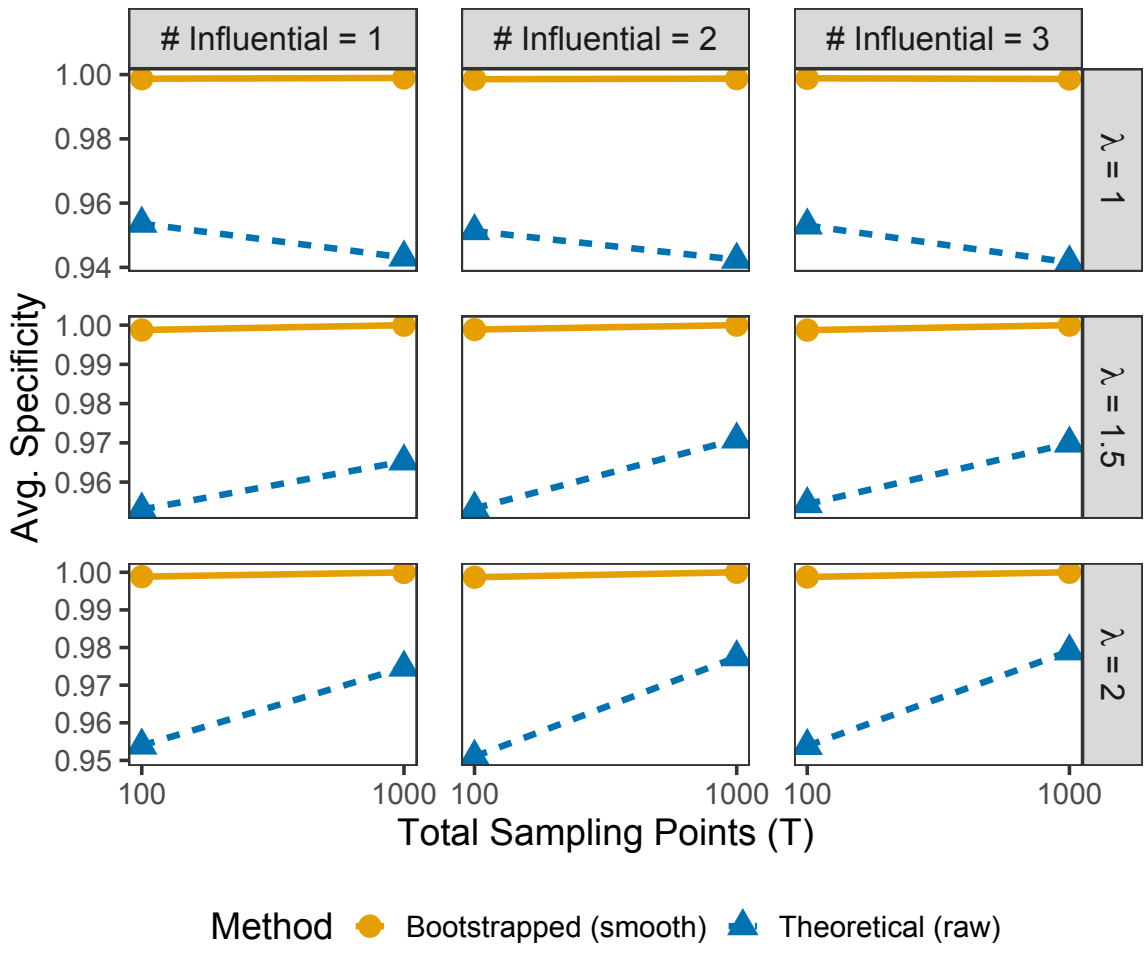


Figure C.11: Average specificity of Theoretical (raw) method implemented with $\alpha = 0.005$ (solid line with circle points) and Bootstrapped (smooth) implemented with $B = 100$, $\alpha_B = 0.5$, and $\alpha = 0.025$ (dashed line with triangle points). The average specificity is calculated by averaging over the models, Model 1, Model 2, and Model 3 and the sample size, $n = 10$, 50, and 100. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ.

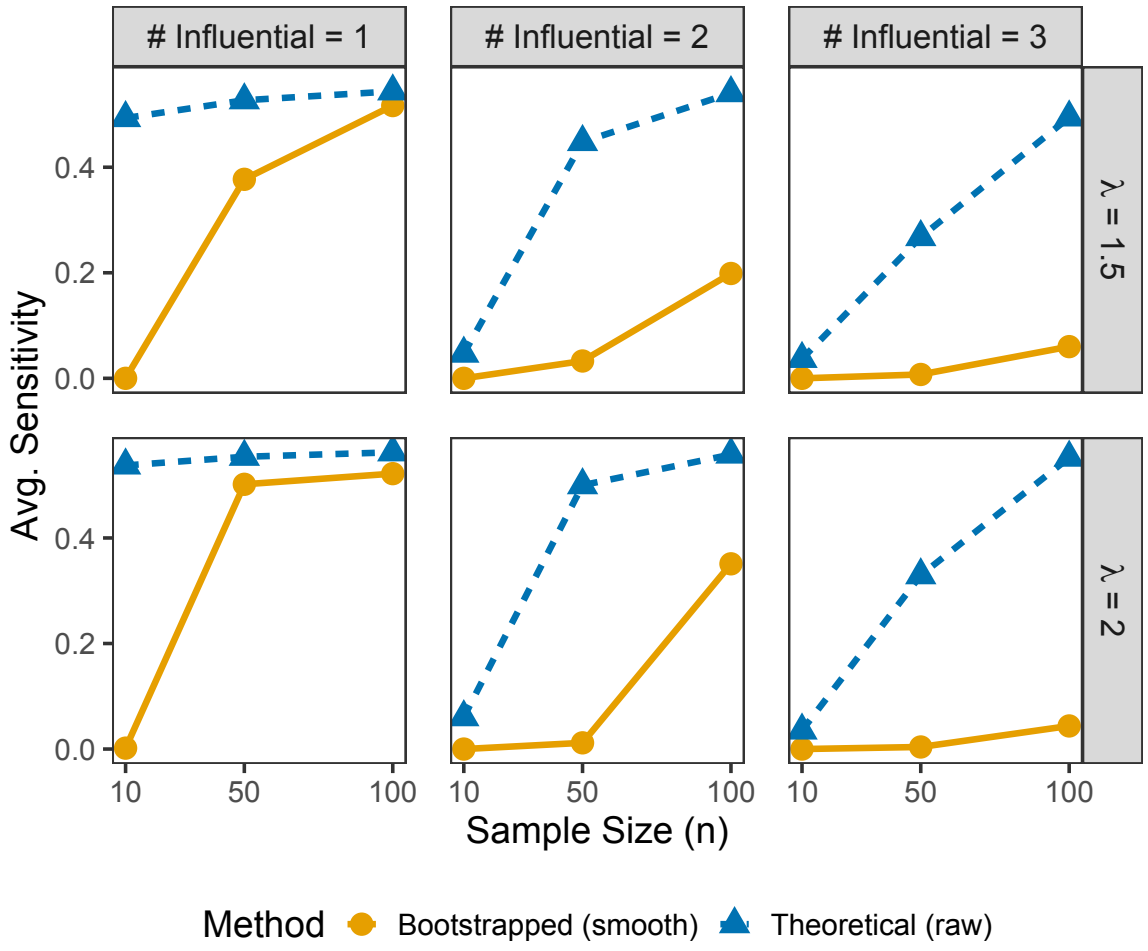


Figure C.12: Average sensitivity of Theoretical (raw) method implemented with $\alpha = 0.005$ (solid line with circle points) and Bootstrapped (smooth) implemented with $B = 100$, $\alpha_B = 0.5$, and $\alpha = 0.025$ (dashed line with triangle points). The average sensitivity is calculated by averaging over the models, Model 1, Model 2, and Model 3 and the number of sampling points, $T = 100$ and 1000. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ.

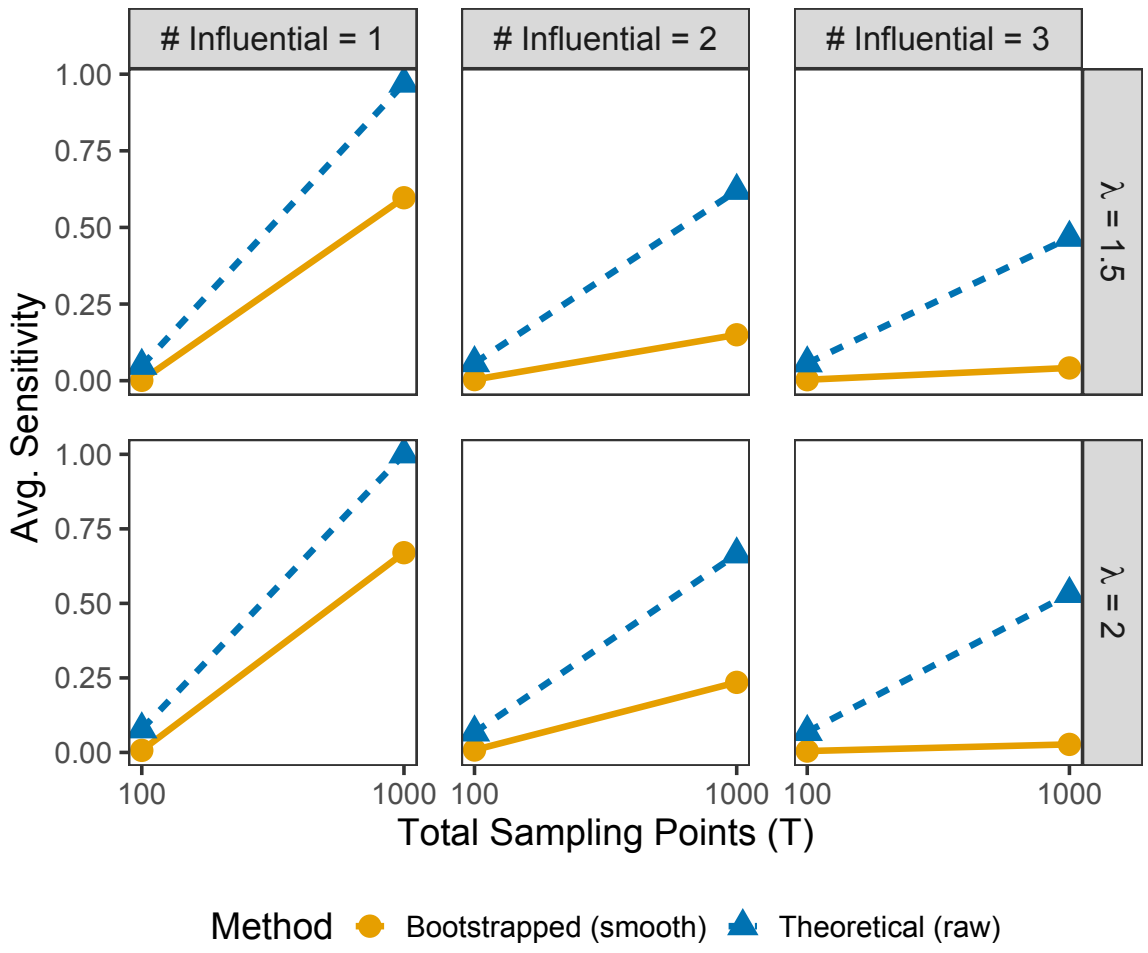


Figure C.13: Average sensitivity of Theoretical (raw) method implemented with $\alpha = 0.005$ (solid line with circle points) and Bootstrapped (smooth) implemented with $B = 100$, $\alpha_B = 0.5$, and $\alpha = 0.025$ (dashed line with triangle points). The average sensitivity is calculated by averaging over the models, Model 1, Model 2, and Model 3 and the sample size, $n = 10$, 50, and 100. The plot is faceted by the number of influential points generated in the random sample, # Influential, and the magnitude of how influential an observation is, λ.

C.3 Additional Application Results

Additional table related to the case study:

Table C.3: The average $DFFITS$, calculated by averaging the functional $DFFITS$ over sampling domain T. The functional $DFFITS$ are first estimated by implementing Pittman (2022) methodology.

Flood Event	Avg. $\lvert DFFITS(t)\rvert$
August 1995	0.96
February 1998	0.82
March 2003	0.88
May 2003	1.78
September 2004	1.78
March 2007	1.71
February 2010	0.85
May 2013	1.07
November 2018	0.74
February 2020	4.01

C.4 Supplementary Material

GitHub Repository: `https://github.com/creutzml/fct_dffits`

The GitHub repository contains all necessary code to reproduce the results presented in "Identifying Influential Observations in A Functional Concurrent Regression Model."

Appendix D

License

Colorado State University LaTeX Thesis Template

by Elliott Forney – 2017

This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or distribute this software, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

In jurisdictions that recognize copyright laws, the author or authors of this software dedicate any and all copyright interest in the software to the public domain. We make this dedication for the benefit of the public at large and to the detriment of our heirs and successors. We intend this dedication to be an overt act of relinquishment in perpetuity of all present and future rights to this software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.