University of Cape Town

Statistics Honours Project

Functional Linear Regression with Truncated Path Signatures

Authors:
Trentin Petersen
Daniela Stevenson

Supervisors:
Andrew Paskaramoorthy
Jake Stangroom

November 14, 2024

Abstract

This project explores the concept of incorporating path signatures into functional linear regression to address the limitations faced by traditional regression methods when dealing with high-dimensional, sequential data. Replicating Adeline Fermanian's 2022 work, which utilised the signature linear model as an alternative to basis expansion techniques, this project implements her methodology on the original dataset used in her paper and extends it to a high-dimensional, real-world dataset. The theoretical component covers key concepts, including functional data analysis and the signature method, with a focus on how path signatures capture higher-order interactions within functional data. Empirically, we compare traditional regression models, specifically those using Fourier, B-spline, and functional principal components basis expansions, with the signature linear model applied to two datasets: the Air Quality dataset and the Appliance Energy Prediction dataset. Our results indicate that while the traditional models perform well in lower-dimensional cases, the signature linear model demonstrates superior performance in high-dimensional settings. This project also assesses Fermanian's truncation order estimation method, highlighting its impact on both computational efficiency and performance. Ultimately, the signature linear model presents itself as an effective approach for high-dimensional functional data, offering a compelling alternative to traditional regression techniques.

Contents

1 Introduction **1**
1.1 Aim . . . 2
1.2 Background Research . . . 3
1.3 Project Outline . . . 3

2 Functional Data Analysis **4**
2.1 Transforming Discrete Data into Smooth Functions . . . 6
2.1.1 Fourier Basis . . . 7
2.1.2 B-Splines . . . 7
2.1.3 Functional Principal Components Analysis . . . 7
2.2 Functional Linear Regression . . . 8
2.2.1 Scalar-on-Function Regression . . . 8
2.2.2 Function-on-Scalar Regression . . . 8
2.2.3 Function-on-Function Regression . . . 9
2.3 SoFR in the Univariate Case . . . 9
2.4 General Assumptions and Limitations of FDA . . . 11
2.5 Key Assumptions and Limitations of FLR . . . 11

3 The Signature Method **12**
3.1 Defining Path Signatures . . . 12
3.1.1 Order of Path Signatures . . . 14
3.1.2 Dimensionality of Truncated Path Signatures . . . 16
3.2 Key Properties and Theoretical Foundations . . . 17
3.2.1 Invariance under Time Reparametrisations and Translation . . . 17
3.2.2 Chen's Identity . . . 17
3.2.3 Propositions Motivating the Signature Linear Model . . . 18
3.3 The Signature Linear Model . . . 20
3.3.1 Model Formulation . . . 20
3.3.2 Comparison of SLM and SoFR . . . 21
3.3.3 Key Parameters: Truncation and Coefficient Estimation . . . 21
3.3.4 Truncation Order Estimation . . . 22
3.3.5 Path Signature Computation . . . 24

4 Experimental Results **25**
4.1 Air Quality Dataset . . . 25
4.2 Appliance Energy Prediction Dataset . . . 27

5 Conclusion **30**

1 Introduction

Functional data analysis (FDA) is a branch of statistics that analyses data that is inherently functional rather than discrete (20). The relevance of FDA has grown with technological advancements that enable the capture of multidimensional sequential data such as financial time series and medical diagnostics (8; 19).

A primary application of FDA is functional linear regression (FLR), which extends traditional linear regression into the functional domain, modelling relationships between predictors and responses that vary continuously over domains, such as time or space. In FLR responses, predictors, and regression coefficients can be functions (10). This project focuses on scalar-on-function regression (SoFR), where functional predictors are used to model a scalar response. These functional components are typically represented using a finite number of basis functions, allowing for smoothing, and analysis (20). However, FLR relies on restrictive assumptions of linearity and independence across dimensions of the data. As a result, it faces several limitations, particularly when handling high-dimensional data and non-linear relationships (5).

To address these limitations, Adeline Fermanian incorporated path signatures into the FLR framework in her 2022 paper by utilising the signature linear model (SLM) (5). This novel approach builds on the foundational work of Levin, Lyons and Ni (12), who explored the application of path signatures in linear regression and for non-parametric statistical inference in sequential data. Unlike traditional FLR, the SLM does not assume independence between predictor dimensions, allowing it to model high-order interactions and temporal relationships effectively. This makes the SLM particularly suitable for analysing high-dimensional datasets (5).

Path signatures are a powerful mathematical tool for analysing sequential data, capturing the non-linear geometric and analytic properties through an infinite collection of iterated integrals (11). However, working with full path signatures is computationally infeasible. To overcome this, truncating the path signature yields a finite-dimensional representation, reducing the infinite set of signatures to a manageable subset. This method prioritises the essential features of the signature, reducing dimensionality while preserving critical information.

This project examines both the theoretical foundations and practical applications of FLR and the SLM. We begin by elucidating key concepts in FDA and FLR, including the application of basis expansions such as Fourier, B-splines, and functional principal components analysis (FPCA) to model functional data. After laying this groundwork, we introduce the signature method, exploring the concept of path signatures and their key properties and propositions that define the SLM, such as its invariance under time reparametrisation and translation, and its ability to approximate functional relationships. These concepts position the SLM as an effective alternative to traditional FLR methods.

In the practical component of this project, we replicate Fermanian's analysis using her original Air Quality dataset and extend this analysis to include a higher-dimensional real-world dataset–the Appliance Energy Prediction dataset. This em-

pirical study compares the SLM against three traditional FLR models, specifically SoFR models using Fourier bases and B-splines, as well as functional principal components regression (FPCR). The results highlight the strengths and limitations of applying the SLM to real-world data, conveying insights into its performance relative to traditional FLR techniques.

Truncation is a key concept in this project, essential for applying the SLM to real-world datasets, as it balances model complexity and computational efficiency. While Fermanian's objective was to establish a rigorous methodology for estimating truncation order, her evaluations were confined to a low-dimensional setting. This project extends this assessment to a high-dimensional setting, testing the practical performance of her truncation estimation procedure in these environments.

Aim

The primary aim of this project is to serve as a complementary guide to Fermanian's paper on FLR with truncated signatures (5). Fermanian used the SLM, a novel approach that uses path signatures in FLR as an alternative to traditional basis expansion methods. Her focus was on establishing a method for estimating truncation order, which she tested on low-dimensional real-world data, while also comparing SLM with traditional FLR methods.

This project has two key objectives:

1. **Theoretical Foundation**: To provide an accessible introduction to key topics such as FDA, FLR, and path signatures. We aim to guide the reader through the foundational properties and key propositions that motivate the definition of the SLM, in order to position it as an alternative to traditional FLR methods. Fermanian assumes a high level of prior knowledge of many theoretical concepts in her paper, which are typically not covered at the honours level. Therefore, this project aims to serve as a resource for honours students by offering accessible explanations of these foundational topics, ensuring readers are well-prepared for the subsequent replication work.
2. **Application and Replication**: To replicate Fermanian's results by applying FLR methods (Fourier, B-splines, FPCR) and the SLM to her original dataset. Additionally, we will extend the application of these methods to a high-dimensional real-world dataset to evaluate their practical performance. Unlike Fermanian's focus on proving the truncation order estimation, we aim to test its effectiveness by applying it in a real-world, high-dimensional setting, assessing both the SLM and truncation estimation method in this new context.

Background Research

The signature method, originally conceptualised by Chen in the 1950s, was later refined in rough path theory by Terry Lyons (2; 17), and has since evolved into a multifaceted technique for analysing multidimensional sequential data. Its applications span various domains such as quantitative finance, medicine, and machine learning.

For example, Gyurkó, Lyons, Kontkowski and Field (2013) (8) demonstrated the use of path signatures in financial time series to extract essential features for use in supervised machine learning algorithms. Their study revealed that signatures outperformed traditional statistical methods, such as linear regression, in filtering valuable information from noisy market data, significantly improving the accuracy of classification predictions.

In healthcare, Moore, Lyons and Gallacher (2019) (19) used path signatures to predict the progression of Alzheimer's disease by capturing non-linear interactions in MRI data. This approach distinguished individuals whose diagnoses progressed to Alzheimer's from those initially diagnosed as healthy or with mild cognitive impairment (MCI) by isolating predictive features in sequential medical data. The study demonstrated the effectiveness of path signatures in identifying non-linear predictive features and interactions without manual selection, highlighting their utility as a data processing tool.

Lyons and McLeod (2022) (16) demonstrated the wide applicability of the signature method in machine learning tasks such as human action recognition and anomaly detection. Their study introduced the log-ODE method, a key concept in rough path theory, which effectively handles complex temporal behaviour and non-linearities in sequential data. The study showed that path signatures are a powerful tool for capturing the structure of multidimensional, non-stationary data.

Ultimately, the papers discussed illustrate the signature method's wide applicability and its potential to address the limitations of traditional FDA techniques. This research provides empirical evidence that the signature method is potentially an effective alternative approach for methods such as FLR.

Project Outline

This project is structured into five main chapters. Chapter 2 discusses the fundamentals of FDA and FLR, covering various FLR types and basis expansion methods, alongside their limitations. Chapter 3 introduces the signature method, path signatures and the SLM, including Fermanian's truncation order estimation. Chapter 4 details the empirical applications of the SLM and SoFR models to the Air Quality and Appliance Energy Prediction datasets, comparing their performance. Chapter 5 concludes by summarising our findings, discussing their implications, and proposing directions for future research. The code for this project is available at: https://gitfront.io/r/trentin/5jexFgsnNwXF/Stats-Honours-Project/

2 Functional Data Analysis

FDA is a branch of statistics that focuses on analysing data represented as functions, rather than discrete observations. Each functional datum is a *set of measurements* along a continuum, treated as a single observation and regarded as a single entity. This typically represents a function, curve, or surface that varies over a continuous domain such as time (13). Functional data is also commonly referred to as time series data, repeated measures, or longitudinal data.

Traditional multivariate statistics typically analyse samples of discrete points, capturing variations such as measurement errors and sampling variability (13). In contrast, FDA treats each observation as a continuous function or curve drawn from a larger population of curves, accounting for variability both within and between curves.

Consider the process of writing a digit, such as '9'. Instead of viewing it as a static, two-dimensional image, we can think of it as a continuous function–a dynamic process where each point is defined by continuous X and Y coordinates over time. This perspective allows us to examine the pen's movement, offering a detailed analysis of how the digit is formed.

A well-known example is the MNIST stroke sequence dataset (4), which contains thousands of handwritten digits frequently used in machine learning. Figure 1, depicts the digit '9' from this dataset. The $X(t)$ and $Y(t)$ coordinates of the digit '9', as shown in Figure 1, represent the discrete path of the pen's movement in each coordinate.

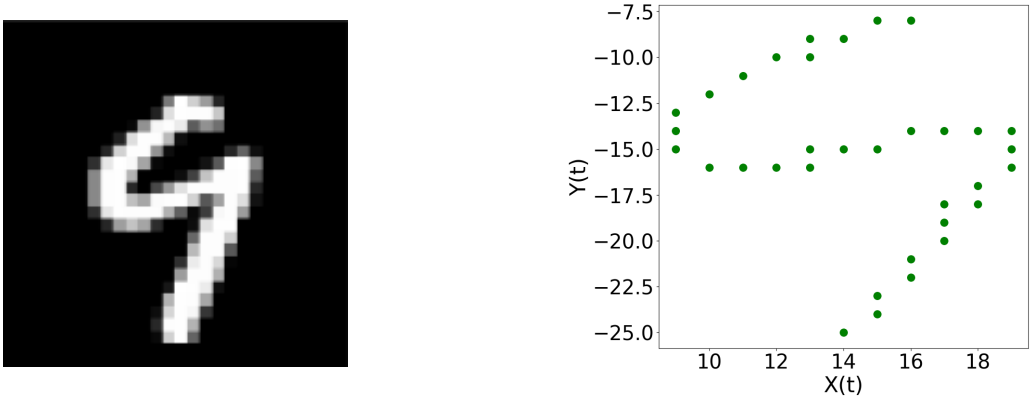

Figure 1: The left figure shows a digit '9' image from the MNIST stroke sequence dataset (4). The right displays the digit's discretised path in a coordinate system.

FDA is based on the assumption that the underlying data-generating process is inherently smooth (13). Smoothing techniques are applied to reduce noise, highlight the continuous structure of the data, and represent each observation as a functional object.

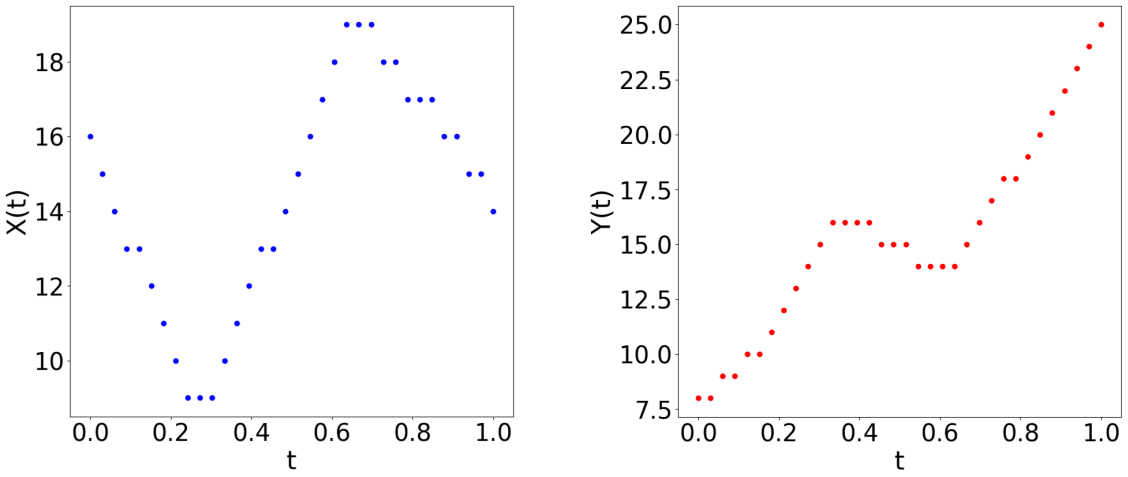

Figure 2: $X(t)$ and $Y(t)$ coordinates plotted separately over time, t.

Figure 2 above shows the $X(t)$ and $Y(t)$ coordinates plotted separately over time, t. To capture the flow of writing, smoothing techniques interpolate these points into a continuous curve, as illustrated in Figure 3. This provides a seamless representation of the pen's movement, highlighting its continuous progression rather than focusing on discrete points.

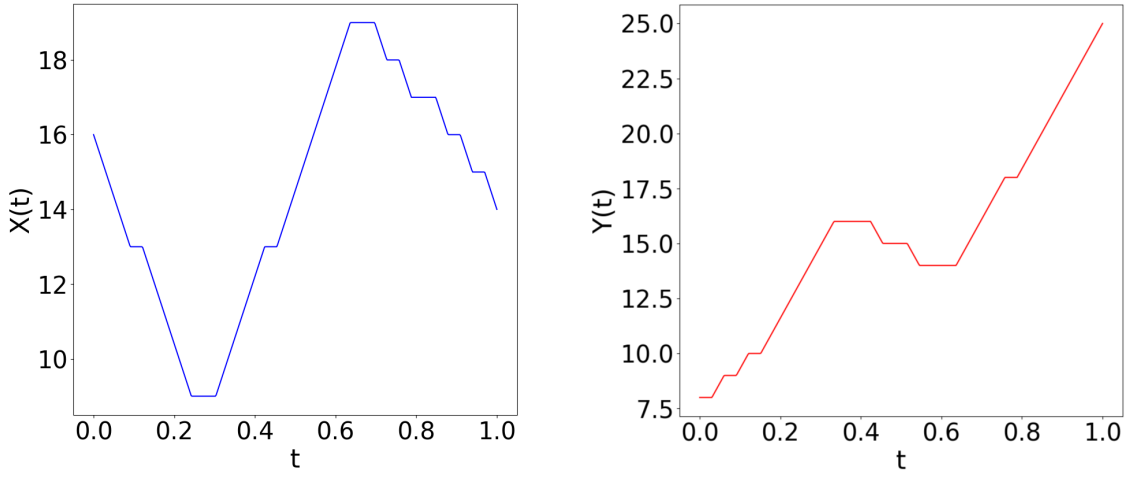

Figure 3: Smoothed $X(t)$ and $Y(t)$ coordinates plotted separately over time, t.

While FDA theoretically models data as continuous, in practice, data is collected as discrete points. Even with high sampling rates, such as handwriting capture devices operating at $100 - 120$ Hz, the observations remain inherently discrete (13). These discrete points are used to approximate the smooth underlying function.

Figure 4 illustrates multiple replications of the digit '9' These replications are samples from multiple participants or multiple samples from the same individual (13). They improve our understanding of the curve's shape, allowing for a more accurate reconstruction of the pen's path.

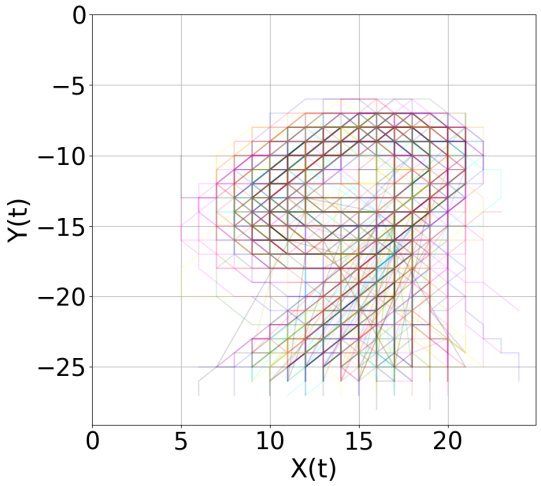

Figure 4: Multiple replications of the handwritten '9' from the MNIST stroke sequence dataset (4) represented as smoothed data points.

Ultimately, FDA's approach to representing pen movements as continuous functions rather than discrete points provides greater insights into the dynamic processes of handwriting. This is especially beneficial for applications such as signature verification, animation, or handwriting recognition, where the continuity and smoothness of pen strokes are important. Unlike traditional methods, which may overlook these subtle dynamics by focusing on the discrete nature of data points, FDA captures the entire path of the pen more effectively, ensuring more accurate depictions and analyses.

Transforming Discrete Data into Smooth Functions

One common preprocessing step is *basis expansion*, which fits a curve to the discrete observations to approximate the underlying continuous process (24). Each observation $X(t)$ is transformed into a smooth function using basis functions,

$$X_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t),$$

where $\phi_k(t)$ are the *basis functions*, and c_{ik}, the *basis coefficients* that determine the relative contribution of each basis function to the curve. The choice of basis functions depends on the data's nature (smooth, periodic, etc.)(23).

Fourier Basis

Fourier basis functions are typically used when dealing with periodic (or near-periodic) data, such as long time series with seasonal trends. These functions, consisting of sine and cosine terms, are ideal for modelling cyclical patterns, especially functions with constant curvature and weak local characteristics. However, they are unsuitable for functions with data discontinuities or low-order derivatives (21).

The Fourier series, which forms the basis for this expansion, is represented as (21):

$$\begin{aligned}\phi_1(t) &= 1\\ \phi_2(t) &= \sin(\omega t)\\ \phi_3(t) &= \cos(\omega t)\\ \phi_4(t) &= \sin(2\omega t)\\ \phi_5(t) &= \cos(2\omega t)\\ &\vdots\end{aligned}$$

Here ω is related to a given period T by $\omega = 2\pi/T$. T is often taken as the range of t values where the data is observed (18).

B-Splines

B-splines are piecewise polynomials, making these more flexible and more complicated than finite Fourier series (21). Thus, these bases are used for non-periodic and complex curves, that cannot have their features captured by low-order polynomials. They consist of polynomial segments joined at *knots*, which allow for varying smoothness along the fitted curve. Different orders provide different levels of smoothness, e.g., *Order 4* B-splines are often used in standard smoothing as they appear reasonably smooth (13).

Functional Principal Components Analysis

Functional principal components analysis (FPCA) reduces the dimensionality of functional data while retaining the most important modes of variation. Unlike B-splines and Fourier basis expansion, which use predefined functions, FPCA derives the basis functions directly from the data. This is beneficial when the structure of the data is not sufficiently captured by traditional basis expansions (13). It extends multivariate PCA, identifying principal components that capture the principal variation among the observed functions.

Functional Linear Regression

Functional linear regression (FLR) is a primary application of FDA, where responses, predictors, and regression coefficients can be functions (10). FLR has three primary forms depending on the nature of the predictor and response variables (23): (*i*) Scalar-on-function regression (SoFR) (*ii*) Function-on-scalar regression (FoSR) (*iii*) Function-on-function regression (FoFR).

Scalar-on-Function Regression

In SoFR, functional predictors are used to model a scalar response:

$$Y_i = \beta_0 + \int_{\mathcal{T}} \beta(t) X_i(t) dt + \epsilon_i,$$

where, Y_i is the scalar response, $X_i(t)$ is the functional predictor, and $\beta(t)$ is the functional regression coefficient, and $\epsilon_i \sim N(0, \sigma^2)$. The set $\mathcal{T}$ represents the domain, typically time, over which $X_i(t)$ and $\beta(t)$ are defined.

Here, $\beta(t)$ describes which regions of the functional predictor, $X_i(t)$, influence the response, Y, capturing how $X(t)$ affects Y. For each, t, the product of the $X(t)$ and $\beta(t)$ represents the incremental change in the response (24). The integral accumulates the effect of $X_1(t)$ on the Y_1 across the entire domain, $\mathcal{T}$. This form is central to our study as we replicate Fermanian's methodology.

An example from Reiss et al. (2017) (24) used pain intensity as the scalar response, Y, and brain activity data as the functional predictor, $X(t)$. Specifically, Y is the self-reported pain intensity after thermal stimuli, and $X(t)$ is the blood oxygen level-dependent (BOLD) signal from fMRI. This model captures how pain is affected by time-varying brain activity, with the functional coefficient $\beta(t)$ identifying which time intervals of the BOLD signal best predict pain intensity.

Function-on-Scalar Regression

In FoSR, the predictor is scalar and the response is functional:

$$Y_i(t) = \beta_0(t) + \int_{\mathcal{T}} \beta(t) X_i dt + \epsilon_i(t),$$

where $Y_i(t)$ is the functional response, $\beta_0(t)$ is the baseline outcome function, and $\epsilon_i(t)$ is the functional error term. The set $\mathcal{T}$ represents the domain, over which $Y_i(t)$, $\beta_0(t)$, $\beta(t)$, and $\epsilon_i(t)$ are defined.

An example of FoSR is seen in the work of Goldsmith et al. (2016) (6), where they modelled children's physical activity over a 24-hour period using scalar predictors such as season, sex, BMI score and asthma diagnosis. The functional response $Y_i(t)$ represents minute-by-minute physical activity, while the scalar predictors account for the variations in activity levels across the day.

Function-on-Function Regression

FoFR models both the predictor and the response as functions. Here, $X : \mathcal{S} \to \mathbb{R}$. The model can be written as:

$$Y_i(t) = \beta_0(t) + \int_S \beta(s,t) X_i(s) ds + \epsilon_i(t),$$

where $\beta(s,t)$ is a coefficient surface describing how changes in $X_i(s)$ affects $Y_i(t)$ over the domain (9).

The example mentioned in FoSR can be extended to FoFR by replacing the scalar predictors with functional predictors. For example, we can model the relationship between the physical activity patterns over a 24-hour period and a time-varying predictor, such as a person's heart rate or specific environmental factors measured throughout the same 24-hour period.

SoFR in the Univariate Case

In this section, we review a SoFR model in the univariate case to illustrate how basis expansions can simplify functional regression models.

In the univariate case, where $d = 1$, the predictor is a real-valued function $X_i : [0,1] \to \mathbb{R}^d$, and the response Y_i is a scalar, $Y_i \in \mathbb{R}$ (5). This model includes a scalar intercept term $\beta_0 \in \mathbb{R}$, a functional coefficient $\beta(t) : [0,1] \to \mathbb{R}$, and an error term $\epsilon_i \sim N(0, \sigma^2)$. The SoFR model can be expressed as,

$$Y_i = \beta_0 + \int_0^1 X_i(t)\beta(t)\, dt + \epsilon_i, \tag{1}$$

In this model, $\beta(t)$ describes how different regions of the functional predictor $X_i(t)$ affect the response Y_i. The integral captures this cumulative effect over the domain.

To estimate this model, both $X_i(t)$ and $\beta(t)$ are expanded using a finite set of basis functions:

$$X_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t), \quad \beta(t) = \sum_{k=1}^{K} b_k\phi_k(t), \tag{2}$$

where $\phi_k(t)$ are the chosen real-valued basis functions (e.g., Fourier or B-splines), c_{ik} are the basis coefficients corresponding to $X(t)$, and b_k are the coefficients associated with $\beta(t)$ (5). Substituting the expansions from Equation 2 into the SoFR, we obtain:

$$Y_i = \beta_0 + \sum_{k=1}^{K} c_{ik} b_k \int_0^1 \phi_k(t)\phi_k(t) dt + \epsilon_i.$$

Another approach is functional principal components regression (FPCR), where $X(t)$ and $\beta(t)$ are expanded using the orthonormal basis functions $\phi_k(t)$ and $\phi_{k'}(t)$, respectively, obtained from FPCA (24).

The coefficients c_{ik} describe how the predictor $X_i(t)$ is represented in terms of the basis functions $\phi_k(t)$, while the coefficients $b_{k'}$ represent the contribution of each basis function $\phi_{k'}(t)$ to the functional coefficient $\beta(t)$.

Now the basis functions are orthonormal, meaning:

$$\int_0^1 \phi_k(t)\phi_{k'}(t)\,dt = \begin{cases} 1, & \text{if } k = k', \\ 0, & \text{if } k \neq k'. \end{cases}$$

This orthonormality simplifies the SoFR model:

$$Y_i = \beta_0 + \sum_{k=1}^{K}\sum_{k'=1}^{K} c_{ik}b_{k'} \int_0^1 \phi_k(t)\phi_{k'}(t)dt + \epsilon_i,$$

which reduces to the traditional multivariate regression model:

$$Y_i = \beta_0 + \sum_{k=1}^{K} c_{ik}b_k + \epsilon_i,$$

since the integral is 1 when $k = k'$. This simplification reduces the complexity of the model, making FPCR computationally efficient.

Ultimately, by expanding both $X(t)$ and $\beta(t)$ using basis functions or principal components, the functional regression problem in Equation 1, is transformed into a traditional multivariate regression problem. The use of basis expansion simplifies parameter estimation, allowing for the application of standard techniques, such as ordinary least squares (OLS), to estimate c_{ik} and b_k. The choice of basis functions and number of components K is critical for model performance, ensuring the model captures the essential features without overfitting.

General Assumptions and Limitations of FDA

FDA operates under the assumption that the underlying data-generating process is smooth, meaning that the data is represented as a continuous and differentiable function over some domain, such as time or space.

Additionally, FDA requires a sufficient number of discrete observations to accurately model the continuous process. This sufficiency is often determined by the researcher's experience, intuition, and trial and error (13). If the dataset is too small, it may not provide enough information to capture the dynamics of the process adequately, thereby limiting FDA's effectiveness. Hence, FDA tends to be most beneficial when applied to large datasets.

Furthermore, the selection of an appropriate type and number of basis functions plays a critical role in model performance. Selecting too many basis functions risks overfitting, where the model captures noise in the data rather than the true underlying patterns. Conversely, using too few basis functions may result in underfitting, where the model oversimplifies the data and misses important nuances in the functional relationship (13).

FDA assumes that the smoothness of the functional data can be captured by the chosen basis, typically B-splines, Fourier, or other functional representations. This assumption, however, may not hold for all datasets, especially those with erratic or highly non-linear behaviours over short intervals (22).

Key Assumptions and Limitations of FLR

In FLR, a key assumption is the projection of functional predictor X onto a lower-dimensional space using basis functions, which simplifies the analysis. However, Fermanian points out that when these predictors are vector-valued ($d > 1$), each coordinate of X is expanded independently, with the model assuming no interaction between coordinates (5). While this assumption simplifies the model, when the coordinates are correlated it is not an efficient representation.

For example, in longitudinal data analysis, repeated measurements are often modelled independently using parametric approaches like ANOVA (5). This works well when each coordinate is fairly independent, but in scenarios where the coordinates are different (e.g., multiple financial stocks or the spatial coordinates of a pen's path), interactions between them are important. Ignoring these correlations reduces the effectiveness of the model in capturing the true dynamics of the data (5).

Another important assumption in FLR is the smoothness of the coefficient function $\beta(t)$, which governs how predictor variables influence the response over time. While smoothness in $\beta(t)$ ensures that the model is not overly sensitive to short-term fluctuations, it also limits the model's ability to detect rapid changes or abrupt shifts in predictor effects over short periods. This is particularly limiting in cases where sharp changes in the relationship between predictors and the outcome variable occur, such as in financial data during market crashes or sudden economic events.

3 The Signature Method

In the previous chapter, we focused on traditional FDA and FLR models, which rely on assumptions of linearity and independence between predictor dimensions. While these methods have provided a foundation for analysing functional data, they can have limitations in fully capturing the complexities of high-dimensional sequential data, such as non-linear and higher-order relationships between variables.

This chapter introduces the signature method, which utilises path signatures, a powerful mathematical tool for analysing sequential data. This method offers greater flexibility by constructing a basis for functions of $X(t)$ that captures the essential geometric and analytic properties of sequential data (5). This approach forms the foundation of the signature linear model (SLM), which incorporates path signatures into the FLR framework. The SLM is particularly useful for time-dependent data, addressing some of the limitations of traditional FDA methods, such as FLR, by reducing reliance on linear assumptions and independence (5).

Ultimately, this chapter serves as a guide through the foundational properties of path signatures, contextualising them within Fermanian's work by providing explanations of key definitions and propositions that motivate the development of the SLM.

Before we can present the SLM, we need to define a path signature and understand its theoretical foundations, i.e., its orders, dimensionality, and key properties. These are all necessary steps to define and present the SLM as a compelling alternative approach to traditional FLR methods with basis expansions.

Defining Path Signatures

A path signature is best defined by clearly understanding what paths and path integrals are.

A **path**, X, is a continuous mapping from an interval $[a, b]$ to some $d-$dimensional Euclidean space, $\mathbb{R}^d$ (3). The path depends on a parameter, typically denoted as t, which often refers to time. We denote the path at time t as $X_t = X(t)$. In $d-$dimensions $X_t \in \mathbb{R}^d$ can be represented as,

$$X : [a, b] \to \mathbb{R}^d,\ X_t = \{X_t^1, X_t^2, \ldots, X_t^d\},$$

The paths $(X_t^1, X_t^2, \ldots, X_t^d)$ are known as coordinate paths and each $X^i : [a, b] \to \mathbb{R},\ \forall i \in \{1, \ldots, d\}$, is a real-valued path.

This illustrates the concept of a path being a curve or trajectory in a multidimensional space. It encompasses anything from measurements of a time series to the movement of a point in space.

In this project paths and functions are synonymous as they can both be viewed as mappings from some time domain to $\mathbb{R}^d$.

Before we can discuss path integrals, we must first define the total variation of a path. Total variation is a measure that quantifies how a path 'moves' or 'varies' as it evolves. Establishing bounds on this variation is important because it helps control the complexity of path signatures and ensures they remain mathematically tractable.

Definition 3.1 (Total Variation of a Path (5))
Let $X : [0,1] \to \mathbb{R}^d$ and let $\mathcal{P}$ denote the set of all partitions of $[0,1]$. The total variation of X is defined as,

$$\|X\|_{TV} = \sup_{p \in \mathcal{P}} \left\{ \sum_{i=1}^{n_p} \|X_{t_i} - X_{t_{i-1}}\| \right\}.$$

Note that the supremum is taken over all finite subdivisions of $[0,1]$, where $t_0, \ldots, t_n$ represents the partition points along the interval, and $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^d$.

X is said to have bounded variation if $\|X\|_{TV} < \infty$; furthermore, the set of all paths of bounded variation is denoted by $BV(\mathbb{R}^d)$.

This project assumes that the paths discussed are smooth and of bounded variation. Smooth paths are those which have derivatives of all orders, meaning they are piece-wise and infinitely differentiable. The assumption of bounded variation is not restrictive upon application, as the observed data utilised is a discretisation of the continuous path, thus it naturally has bounded variation.

Having discussed the makings of a path, a **path integral** generalises the concept of integrating a function along a path to integrating any real-valued path against another. Consider the two real-valued paths, $X, Y : [a,b] \to \mathbb{R}$. The path integral of Y against X is defined as,

$$\int_a^b Y_t dX_t.$$

Since this project assumes all paths are of bounded variation the above integral can be expressed as a Riemann-Stieltjes integral:

$$\int_a^b Y_t dX_t = \int_a^b Y_t \frac{dX_t}{dt} dt = \int_a^b Y_t \dot{X}_t dt.$$

The 'upper-dot' notation indicates taking a derivative with respect to one variable.

A Riemann-Stieltjes integral evaluates how the path Y accumulates along the trajectory defined by X. It encapsulates how Y interacts with X over the interval $[a,b]$.

For example, if we set $X_t = t, \forall t \in [a, b]$, then the path integral becomes the standard Riemann integral of Y that measures the area under a curve,

$$\int_a^b Y_t dX_t = \int_a^b Y_t dt.$$

Alternatively, if we set $Y_t = 1, \forall t \in [a, b]$ then it becomes the increment of X,

$$\int_a^b Y_t dX_t = \int_a^b 1 \times dX_t = \int_a^b \dot{X}_t dt = X_b - X_a.$$

Now that we have explained the concept of a path and path integral, we are ready to define a path signature.

Definition 3.2 (Path Signatures (3))
The signature of a given path, $X : [a, b] \to \mathbb{R}^d$, is denoted by, $S(X)_{a,b}$, is a collection of iterated integrals that comprehensively captures the path's behaviour and properties. It is constructed by considering all of its possible iterated integrals. Formally, it is the sequence of real numbers,

$$S(X)_{a,b} = \left(1, S^1(X)_{a,b}, \dots, S^d(X)_{a,b}, S^{(1,1)}(X)_{a,b}, S^{(1,2)}(X)_{a,b}, \dots\right),$$

where, by convention, the "zeroth" term is equal to 1, and the superscripts run along the set of all multi-indices:

$$W = \{(i_1, \dots, i_k) \mid k \geq 1, i_1, \dots, i_k \in \{1, \dots, d\}\}.$$

Further, the signature of X truncated at order m is the sequence $S^m(X)_{a,b}$ containing all the signature coefficients of order lower than or equal to m (5),

$$S^m(X)_{a,b} = (1, S^1(X)_{a,b}, S^2(X)_{a,b}, \dots, S^{\overbrace{(d, \dots, d)}^{\text{length m}}}(X)_{a,b}).$$

Order of Path Signatures

Now, we will explore the construction of path signatures, starting with the first-order and progressing to higher-order integrals. These signatures provide a hierarchical representation of the path, with each successive order offering deeper insight into its structure.

The **first-order** of iterated integrals are simply the increments of the path. The signature for any single index $i \in \{1, \dots, d\}$ is,

$$S^i(X)_{a,t} = \int_{a<s<t} dX_s^i = X_t^i - X_a^i.$$

The **second-order**, also known as the *double-iterated integral* of X, this is defined for any pair of indices $i, j \in \{1, \ldots, d\}$ as (3),

$$S^{(i,j)}(X)_{a,t} = \int_{a<s<t} S^i(X)_{a,s} dX_s^j = \int_{a<r<s<t} dX_r^i dX_s^j = \int_a^t \int_a^s dX_r^i dX_s^j$$

This measures the interaction between the i^{th} and j^{th} components of the path as they evolve over time, by capturing the interaction between the i^{th} component at time r and the j^{th} component at time s, where $r < s$. This shows that for each fixed s, you integrate with respect to r first, then integrate with respect to s (3).

This pattern is continued recursively as the order increases. **Higher-order** iterated integrals involve multiple integrals over different components of the path.

For any collection of indices, $i_1, \ldots, i_k \in \{1, \ldots, d\}, \forall k \geq 1$, a path integral is defined as (3),

$$S^{(i_1,\ldots,i_k)}(X)_{a,t} = \int_{a<s<t} S^{(i_1,\ldots,i_{k-1})}(X)_{a,s} \ldots dX_s^{i_k}, \tag{3}$$

which is equivalent to,

$$S^{(i_1,\ldots,i_k)}(X)_{a,t} = \int_{a<t_1<t_2<\cdots<t_k<t} dX_{t_1}^{i_1} dX_{t_2}^{i_2} \ldots dX_{t_k}^{i_k}.$$

The *K-fold iterated integral* of X along the collection of indices, $i_1, \ldots, i_k$, is then the real number $S^{(i_1,\ldots,i_k)}(X)_{a,t}$ (3).

The **complete signature** is the infinite sequence of all iterated integrals, which captures the information from the first-order integrals to the higher-level interactions. As the order of iterated integrals increases, each additional term, while increasing in computational complexity, contributes progressively less new information about the path's overall shape. This reflects a principle of diminishing returns in the utility of the signature terms (7).

Furthermore, a critical feature of path signatures is their ability to capture the geometric properties of the path (5). The first-level iterated integrals capture the overall direction and length of the path and higher-level integrals, such as second-order, capture geometric features like areas (e.g., the area enclosed by a path in two dimensions) (3).

As illustrated in Figure 5, the first order of coefficients, $S^{(i)}(X)$ and $S^{(j)}(X)$ represent the increments of the path in each coordinate, i and j, respectively. Further, the second order of coefficients, $S^{(i,j)}(X)$ and $S^{(j,i)}(X)$ correspond to the areas outlined by the path, i.e. the orange and blue regions respectively.

Higher orders of signature coefficients encode not only the individual changes in each coordinate but also how the different coordinates change together over time, capturing the interactions between them.

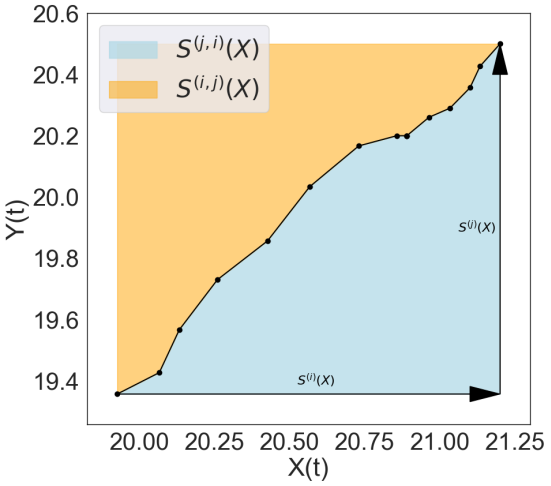

Figure 5: Geometric interpretation of the first and second signature coefficients orders using a path sampled from the Appliance Energy Prediction dataset (15).

Dimensionality of Truncated Path Signatures

Next, we consider the dimensionality of truncated path signatures, which reflects how the number of signature coefficients grows as we increase the order of truncation. This plays a central role in determining the computational complexity and scalability of path signatures, particularly when applied to high-dimensional data.

By changing the integration bounds in Equation (3), the definition of a truncated path signature can be extended to paths defined on any interval $[s,t] \subset \mathbb{R}$ (5). There are d^k signature coefficients of order k (5). The size of the signature of a path $X \in BV(\mathbb{R}^d)$ truncated at order m is $s_d(m)$. The signature truncated at order m is thus a vector of dimension $s_d(m)$ where,

$$s_d(m) = \begin{cases} \sum_{k=0}^{m} d^k = \frac{d^{m+1}-1}{d-1} & \text{if } d \geq 2, \\ \\ m+1 & \text{if } d = 1. \end{cases}$$

For $d = 1$, the size of $S^m(X)$ grows linearly since the only path dimensions are the scalar increments at each order. Thus, as long as $d \geq 2$, the size of $S^m(X)$ increases exponentially with m and grows polynomially with d, as demonstrated by the values in Table 1.

m	$d=1$	$d=2$	$d=3$	$d=6$	$d=10$
1	2	3	4	6	11
2	3	7	13	42	111
5	6	63	364	9330	111111
7	8	255	3280	335923	11111111

Table 1: Typical values of $s_d(m)$.

Key Properties and Theoretical Foundations

The signature method has numerous important properties that justify its use in modern data analysis. In this section, we will examine several properties and propositions from Fermanian's paper (5), which form the theoretical foundation for implementing signatures as a tool used to improve the performance of traditional functional linear models, such as SoFR. For further details, see Chevyrev and Kornmilitzin's primer on the signature method in machine learning (3).

Invariance under Time Reparametrisations and Translation

Based on the signature's definition as an integral, it is clear the signature views functions as purely geometric objects (5), making it invariant under time reparametrisation and by translation. This is a fundamental property of the signature method as it ensures that path signatures consistently represent the geometric structure of functional data, regardless of how the path is traversed in time or its starting position. Thus, information about sampling frequency, speed or travel time is unnecessary, enhancing the SLM's generalisation abilities.

Moreover, this property underpins several important propositions, which will be discussed in detail once the relevant propositions have been introduced. Essentially, the property ensures a stable, shape-preserving representation that is crucial in multivariate data analysis.

Chen's Identity

Chen's identity plays an important role in the theory of path signatures, as it elucidates how signatures comprehensively encode path information. Before stating Chen's identity, it is necessary to first define the concatenation of paths.

Definition 3.3 (Concatenation of Paths (3))
For two paths, $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$, their concatenation can be defined as, $X * Y : [a, c] \to \mathbb{R}^d$ such that,

$$(X * Y)_t = \begin{cases} X_t & \text{for } t \in [a, b], \\ X_b + (Y_t - Y_b) & \text{for } t \in (b, c]. \end{cases}$$

This definition of path concatenation is essential for understanding Chen's identity, which reveals how the signature operation transforms the 'concatenation product' $*$ of the paths into the product of their signatures (3). This result illustrates how paths can be described by their component parts.

Theorem 1 (Chen's Identity (3))
Let $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ be two paths. Then,

$$S(X * Y)_{a,c} = S(X)_{a,b} \cdot S(Y)_{b,c}.$$

Chen's identity allows the algebraic manipulation of signatures to represent concatenated paths, making it possible to handle paths constructed from multiple path segments by using their individual signatures. In sequential or time-series data, where paths often consist of several smaller segments representing different stages or phases of a process, each segment can be analysed independently.

As we have already discussed, path signatures encode the essential geometric information of a path. However, computing the signature of long, intricate paths directly would be computationally expensive and inefficient. Chen's identity addresses this issue. Firstly, it allows the signature for each segment to be computed separately and in parallel, improving computational efficiency. Secondly, Chen's identity provides a mathematical formula to combine the signatures of the individual segments into a single signature that represents the entire path. This is done through algebraic operations that preserve the path's sequential and geometric properties.

Overall, this identity is a valuable tool in applications such as handwriting recognition, time-series analysis, financial modelling, and motion tracking, where the data often spans multiple sequential stages. This practical utility is further explored later in our *Path Signature Computation* section at the end of the chapter.

Propositions Motivating the Signature Linear Model

Now we will examine a series of propositions that motivate the definition of the SLM.

Proposition 1 (Uniqueness of Path Signatures (5))
Let $X \in BV(\mathbb{R}^d)$ such that it contains at least one monotone coordinate, then its signature, $S(X)$, can uniquely characterise the path up to translations and reparameterisations.

This proposition ensures that the signature of a path provides a unique representation of that path, meaning no two different paths – with differences in their structure or order of events – can share the same signature. This is provided that the path contains at least one monotone coordinate, typically time. Importantly, this monotone coordinate does not need to represent the exact timing of events, but must simply preserve the order in which the events occur.

The time reparametrisation invariance property allows us to focus on the sequence of events without concern for the precise timing or the speed at which different segments of the path are traversed. This feature is particularly useful in real-world datasets, where the exact timing may be unknown, but the order of events is available.

The next proposition functions as a universal approximator for continuous functions on paths, drawing a strong parallel to the capabilities of the universal approximation theorem in neural networks, which asserts that a neural network can approximate any function given sufficient complexity and proper configuration.

Proposition 2 (Universal Approximation Property (5))
Let D be the set of all compact paths in $BV(\mathbb{R}^d)$. Suppose $X \in D \subset BV(\mathbb{R}^d)$ with the initial condition $X_0 = 0$, and let $f : D \to \mathbb{R}$ be a continuous function defined on D. Consider the time-augmented path $\tilde{\mathbf{X}} = (X_t, t)'$ defined over $t \in [0, 1]$. Then for every $\epsilon > 0$, there exists an order $m^* \in \mathbb{N}$ such that, for any $m \geq m^*$ there exists a vector of coefficients $\boldsymbol{\beta}^* \in \mathbb{R}^{S_d(m)}$ such that,

$$|f(X) - \boldsymbol{\beta}^* \cdot S^m(\tilde{\mathbf{X}})| < \epsilon.$$

Here $\cdot$ represents the dot product on $\mathbb{R}^{S_d(m)}$.

This proposition is arguably the most important as it establishes the core motivation of the SLM, showing that signatures have the capacity to linearise and approximate continuous functions on entire paths within a bounded error margin (5).

A key property supporting this approximation is the signature's invariance under time reparametrisation and translation. Since the signature depends only on the path's geometric shape, it provides a stable, invariant representation, enabling the signature method to approximate continuous functions (such as regression functions), regardless of specific path characteristics. Translation invariance allows us to assume the initial condition $X_0 = 0$ (5), enabling the signature method to model paths from different starting points. This is essential in multivariate data analysis, such as in financial and biological data, where paths often start at varying levels.

Proposition 3 (Bounded Norms of Truncated Signatures (5))
Let $X : [0, 1] \to \mathbb{R}^d$ be a path in $BV(\mathbb{R}^d)$. Then, for any $m \geq 0$,

$$\|S^m(X)\| \leq \sum_{k=0}^{m} \frac{\|X\|_{TV}^k}{k!} \leq e^{\|X\|_{TV}}.$$

This proposition offers a bound on the norm of truncated signatures, which ensures the rate of decay of higher-order signature coefficients can be controlled (5). Specifically, the bound indicates that as the order of truncation m increases, the influence of the higher-order coefficients diminishes. This implies that the information captured by these coefficients decays exponentially (14).

While truncation is already necessary due to the infinite nature of the signature, this proposition provides a justification for estimating an optimal truncation order. It demonstrates that the contribution of the higher-order coefficients has a diminishing return. As a result, truncating the signature not only reduces computational complexity but also ensures that the majority of the essential information is preserved with the lower signature orders.

Ultimately, each proposition, along with the signature's invariance under time reparametrisation and translation property, directly supports the practical application of signatures in complex data environments. Thus promoting the SLM as a powerful tool in data analysis. In the next section, we will explore some of these properties further, examining their impact on the SLM.

The Signature Linear Model

Having established the theoretical framework outlined in the preceding sections, which reviewed the core properties, propositions, and capabilities of path signatures, we can now introduce the SLM.

Model Formulation

One of Fermanian's primary goals was to model the functional relationships between a real random response variable, $Y \in \mathbb{R}$, and a random input path $X \in BV(\mathbb{R}^d)$ (5). Now we assume that $d \geq 2$ and that a temporal dimension has been incorporated meaning we augment X with a time component t, such that one coordinate of X is $t \mapsto t$. Furthermore, from Proposition 2, we assume there exists a truncation order $m \in \mathbb{N}$ and corresponding coefficients $\boldsymbol{\beta}_m^* \in \mathbb{R}^{S_d(m)}$, which motivates the following model (5):

$$\mathbb{E}[Y|X] = \boldsymbol{\beta}_m^* \cdot S^m(X), \quad Var(Y|X) \leq \sigma^2 < \infty. \tag{4}$$

We now consider the smallest truncation order $m^* \in \mathbb{N}$ for which there exists coefficients $\boldsymbol{\beta}_{m^*}^* \in \mathbb{R}^{S_d(m)}$ that satisfy the model:

$$\mathbb{E}[Y|X] = \boldsymbol{\beta}_{m^*}^* \cdot S^{m^*}(X).$$

We now include the error term, expressing the model as:

$$Y = \boldsymbol{\beta}_{m^*}^* \cdot S^{m^*}(X) + \epsilon.$$

In simpler terms, we assume a regression model where the relationship between the variables is a linear combination of the signature $S^{m^*}(X)$ (5). This model indicates that the chosen order, m^*, and coefficients, $\boldsymbol{\beta}_{\boldsymbol{m^*}}^*$ are sufficient to capture the expected response, Y given the input path X.

Proposition 1, the uniqueness of path signatures property, ensures the consistency and interpretability of the SLM by guaranteeing that each path has a unique signature. This is crucial because, without uniqueness, different paths could produce the same signature, leading to ambiguity and reduced interpretability. The uniqueness property prevents this, ensuring that the model behaves predictably and reliably in practical applications.

Moreover, Proposition 2 highlights the signature's approximation capabilities, underscoring the practical utility of the SLM in various applications. By approximating continuous functions to some desired degree of accuracy, signature models can effectively capture and predict dynamics inherent in sequential data. This proposition is foundational to the model's ability to handle non-linear relationships and interactions, offering flexibility without the restrictive assumptions often required by traditional methods like FLR.

Comparison of SLM and SoFR

It is instructive to compare the SLM, presented in Equation 4, to the SoFR model as described in Equation 1 (5). Firstly, the SLM does not require the assumption that the conditional expectation $E[Y|X]$ is linear in X, unlike the SoFR model, which assumes a linear relationship between X and Y. This allows the SLM to capture non-linear relationships between the input and output variables.

Additionally, the SLM operates under different assumptions regarding X. Specifically, X needs only to be of bounded variation, while the SoFR model assumes X has a smooth functional representation, necessitating a finite basis expansion (5). Furthermore, while SoFR requires a sufficient number of observations and smoothness of the coefficient function $\beta(t)$, the SLM imposes no such requirements (5).

The SLM is tailored to the vector-valued case, enhancing its applicability (5). In contrast, the SoFR model is designed for function-valued data, making it less applicable to scenarios where simpler vector representations are preferred. Moreover, the SLM relies directly on a finite vector of signature-derived coefficients, $\beta^*_{m^*}$, whereas the SoFR model constructs β using basis functions (5). Notably, the selection of these basis functions must be customised for each specific application, adding complexity to the model. In contrast, the SLM depends only on two parameters, making it a more generally applicable model with fewer hyperparameters (5).

Lastly, the regression coefficients in the SLM correspond directly to the signature coefficients, which are geometrically interpretable up to the second level of the signature. In contrast, the SoFR model produces functional regression coefficients, $\beta(t)$, which span the entire domain of t and are therefore less intuitive to interpret.

Key Parameters: Truncation and Coefficient Estimation

Notably, since the first term of any signature is 1, the SLM naturally includes an intercept. This means that when the truncation order $m^* = 0$, model 4 simplifies to a constant model (5).

The two unknown variables in the SLM are m^* and $\boldsymbol{\beta}^*_{m^*}$ (5). The truncation order m^* of the signature of X determines the size of our model, while $\boldsymbol{\beta}^*_{m^*}$ represents the regression coefficients, whose dimensions, $S_d(m^*)$, are dependent on m^*.

The truncation order m^* is a crucial parameter for the SLM as it controls the number of coefficients used, which directly impacts the computational viability of the entire signature method. As highlighted in Fermanian's analysis (5), this critical aspect is often not rigorously addressed in existing literature, with smaller truncation values typically chosen without sufficient justification tied to the structure of the model. This underscores the need for a systematic approach to determining m^* that ensures the model's effectiveness while maintaining computational practicality.

Ultimately, Fermanian's objective was to establish a rigorous method for estimating m^* and, in doing so, define a consistent estimator of m^* (5). Consequently, she obtained a simple estimator for $\beta^*_{m^*}$, thereby enabling the estimation of the regression function (5).

Truncation Order Estimation

The truncation order m^* determines the complexity and the dimensionality of the SLM. Fermanian defines m^* as the point where the balance between model complexity and empirical risk minimisation is optimised. This is achieved by exploring a sequence of nested model spaces indexed by m, where each space corresponds to a signature truncated at m. We discuss an overview of this topic; for a more in-depth examination, see Theorem 1 in Fermanian's work (5).

The *theoretical risk* for a specific truncation m is defined as,

$$R_m(\beta) = \mathbb{E}[(Y - \beta \cdot S^m(X))^2].$$

This equation represents the expected squared error between the observed outcomes and the model's predictions under that truncation order. As m increases, the model's accuracy increases, which subsequently increases the risk of overfitting due to the increased complexity. Therefore, the optimal truncation order m^* is the smallest m where the gain in predictive accuracy justifies the additional complexity.

Conversely, *empirical risk* is calculated based on actual data and is defined as,

$$R_{n,m}(\beta) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta \cdot S^m(X_i))^2.$$

Minimising this empirical risk across different truncation values, m helps select the optimal truncation order, m^*. This ensures an appropriate trade-off between accurately fitting the data and maintaining model simplicity.

The objective is to identify a truncation order, $\hat{m}$, that closely approximates the optimal order m^*. This estimator of the true m^* is determined by the trade-off between the decreasing empirical risk and the increasing function that penalised the number of coefficients. Meaning $\hat{m}$ is defined by,

$$\hat{m} = \min\left(\arg\min_{m\in\mathbb{N}}\left(\hat{L_n}(m) + \mathrm{pen}_n(m)\right)\right), \tag{5}$$

where $\hat{L_n}(m)$ is the minimised empirical risk for truncation m, and $\mathrm{pen}_n(m)$ is a penalty function. This approach is analogous to Ridge regression, as it involves regularising the number of parameters to avoid overfitting, a prevalent issue in high-dimensional settings.

The penalty function, $\text{pen}_n(m)$, is used to prevent the model from becoming overly complex by penalizing the model based on the order of truncation, m. This plays a role in preventing overfitting of the model and crucially ensures that the model is computationally feasible.

$\text{pen}_n(m)$ is defined by Fermanian (5) as follows:

$$\text{pen}_n(m) = K_{pen} n^{-\rho} \sqrt{s_d(m)}, \tag{6}$$

where n is the number of samples, $s_d(m)$ is the size of the signature, ρ is a parameter of the penalization which is chosen such that $0 < \rho < \frac{1}{2}$ and K_{pen} is an arbitrary penalization constant which is calibrated using the slope heuristics method (1), $K_{pen} > 0$.

Only K_{pen} needs to be estimated; the other parameters are either chosen or selected from an analysis of the data or model. An overview of the procedure for choosing K_{pen} as implemented by Fermanian (5) is described below.

1. Choose a sequence of candidate K_{pen} values.
2. Use slope heuristics method (1) to calibrate a plot of K_{pen} vs. $\hat{m}$.
3. Identifying the value of K_{pen} which corresponds to the first big jump of $\hat{m}$
4. Set K_{pen} to be equal to $2 \times K_{pen}$ identified in Step 3.

This can be demonstrated through a brief example. Given a range of K_{pen} values, the following plot is produced.

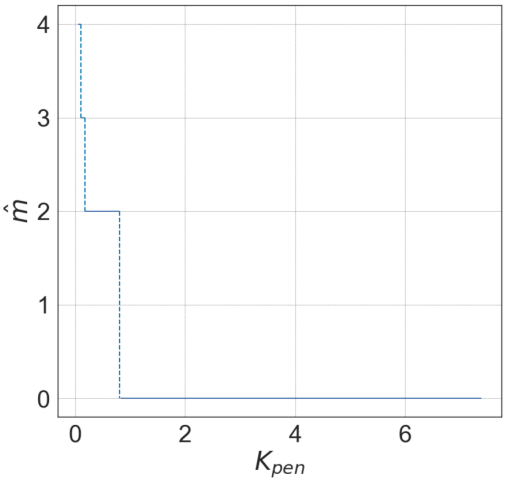

Figure 6: K_{pen} vs. $\hat{m}$

The value of K_{pen} that corresponds to the first big jump of $\hat{m}$ is approximately $K_{pen} = 0.05$. Therefore K_{pen} is set to $K_{pen} = 0.1$.

Once a value for K_{pen} has been identified, $\text{pen}_n(m)$ can be calculated using equation 6 and used to estimate an appropriate truncation order using equation 5.

Path Signature Computation

Building upon the theoretical groundwork of Chen's identity, discussed in earlier in this chapter, we now explore the practical application of this concept within the SLM. As outlined in our previous discussion on FDA in Chapter 2, theoretical models typically assume continuous paths, while real-world data is inherently discrete. Since the signature is defined as an integral over a continuous path, it cannot be directly calculated from discrete data points. This discrepancy necessitates a method for transforming discrete data into a continuous form, enabling the application of continuous path-dependent models like the SLM.

To address this issue, *interpolation* is utilised to transform discrete points into a smooth, continuous path approximation. This interpolation is linear and aims to construct a continuous path that passes through or near the discrete set of known points. This continuous path approximation is crucial as it allows the application of Chen's identity to the data. Most signature computation libraries, including `iisignature` (which we use in this project), utilise linear interpolation for their computational efficiency when leveraging Chen's identity.

Following interpolation, Chen's identity is applied recursively to concatenated paths. This is succinctly demonstrated in the modified version of Theorem 1, adapted from Fermanain's Proposition 4 in her paper (5), which formed the central computational element of her SLM method.

Theorem 2 (Chen's Identity (5))
Let $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ be two paths with bounded variation. Then, for any multi-index $(i_1, \ldots, i_k) \subset \{1, \ldots, d\}^k$, the k^{th} order signature of the concatenated path $X * Y$ is given by,

$$S^{(i_1,\ldots,i_k)}(X * Y) = \sum_{\ell=0}^{k} S^{(i_1,\ldots,i_\ell)}(X) \cdot S^{(i_{\ell+1},\ldots,i_k)}(Y).$$

This theorem illustrates how the signature of a concatenated path can be decomposed into the signatures of its individual segments, simplifying the application of linear models to path-dependent data. By applying interpolation to each segment, we can approximate the entire path and compute its corresponding signature, enhancing the computational efficiency of the SLM.

4 Experimental Results

With the theoretical foundations for FLR and the SLM established in Chapters 3 and 5, we now shift our focus to implementing these theoretical models on two real-world datasets using Python. The first dataset is the Air Quality dataset(25), as used by Fermanian (5), and the second is the Appliance Energy Prediction dataset (15). Both datasets can be found and accessed through the UCI Machine Learning Repository.

To analyse the datasets, we employed Fourier and B-spline basis expansions to fit SoFR models, conducted FPCR, and applied the SLM.

For the Fourier FLR, B-spline FLR and FPCR, the optimal number of basis functions (*nbasis*) used in the basis expansion of the observations was determined through 5-fold Cross-validation. The models were fit to the data using different values for *nbasis* such that the optimal *nbasis* value is selected based on the value that produces the lowest cross-validated MSE. For the Fourier and B-spline FLR models, *nbasis* was set within the range of 4 to 13, inclusive. For the FPCR model, a basis expansion using 7 B-spline components was applied to the functional predictors, after which *nbasis* was selected from the range of values between 1 and 6. For the SLM, the estimator of the optimal truncation parameter, $\hat{m}$, was chosen using the algorithm proposed and developed by Fermanian (5).

The Fourier FLR, B-spline FLR, FPCR, and SLM models were fitted to 20 randomised train/test splits, and the test Mean Squared Errors (MSEs) were computed for each split, producing a distribution of test MSEs for each model.

This methodology is a direct replication of the approach discussed and implemented by Fermanian. This ensures consistency and enables a direct comparison of our results with those reported by Fermanian.(5).

Air Quality Dataset

The Air Quality dataset is comprised of the hourly averaged responses from an Air Quality Chemical Multi-sensor Device, which includes five metal oxide chemical sensors. The sensors were used to capture measurements of various gas concentrations, such as nitrogen dioxide, in the air over the course of a year from March 2004 to February 2005. Additionally, the dataset includes the "ground truth" concentrations for several gasses (CO, NO2, etc.). These measurements are provided by a certified reference analyser located in the same area as the multi-sensor and serve as a benchmark for evaluating the performance of the previously mentioned multi-sensor.

Fermanian proposed implementing SoFR on this dataset using the past 7 days of hourly nitrogen dioxide measurements ($PT08.S4(NO2)$) from the multi-sensor to predict the ground truth concentration of nitrogen dioxide ($NO2(GT)$) in the following hour. This represents a univariate scenario, $d = 1$, as only one functional predictor is being used to predict the scalar response; however, it can easily be extended to the multivariate scenario, $d = 3$, by introducing additional functional

predictors. In this case, the hourly averaged temperature (T) and relative humidity (RH) are introduced, and alongside $PT08.S4(NO2)$, are used to predict $NO2(GT)$.

Implementing the various models for both the univariate and multivariate scenarios yielded the distributions of MSEs, as shown in the boxplots in the figure below.

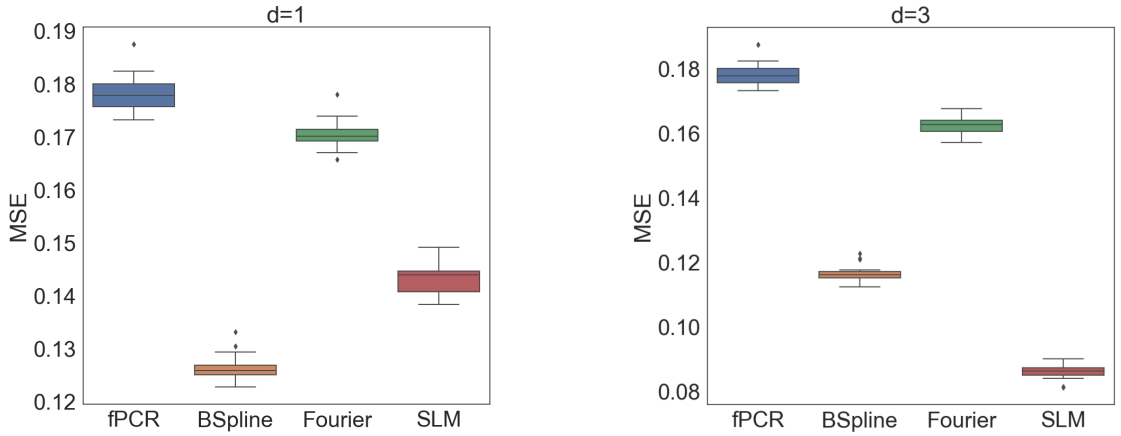

Figure 7: Test MSE distributions for regression models fitted to the Air Quality dataset (25) using $d = 1, 3$.

In the univariate case, while the SLM outperforms the Fourier FLR and the FPCR models, the B-spline FLR achieves the lowest MSE and is, therefore, the best performer amongst the four models. Since the paths in the univariate case are one-dimensional, there is a great loss in the complexity of the data, making the benefits of the SLM less pronounced. As a result, the far simpler B-spline model performs better in this scenario.

In contrast, the multivariate SoFR reveals a notable reduction in MSEs for the SLM, with the SLM outperforming the three traditional models. This is likely due to the signature's ability to capture more complex structures and interactions/dependencies between the different functional predictors in the data.

The results presented in Figure 7 are consistent with Fermanian's findings and demonstrate the SLM's strength in handling more complex, higher-dimensional functional predictors. Minor differences between the test MSEs in our analysis and Fermanian's analysis can be attributed to the different seeds used for the train/test splits.

The regression coefficients produced by the SLM, $\hat{\boldsymbol{\beta}}_{\hat{m}}$, are represented using a heatmap, as shown in Figure 8.

The heatmap reveals that the coefficients corresponding to the signature terms $S^{(1)}(X)$ and $S^{(1,1)}(X)$ are the largest. This is expected in the context of the problem, as both of these terms represent the change in $PT08.S4(NO2)$ concentration over the seven-day period.

Additionally, $S^{(3,1)}(X)$, which represents the area under the curve of Relative Humidity (RH) and $PT08.S4(NO2)$, has a fairly large corresponding regression coefficient.

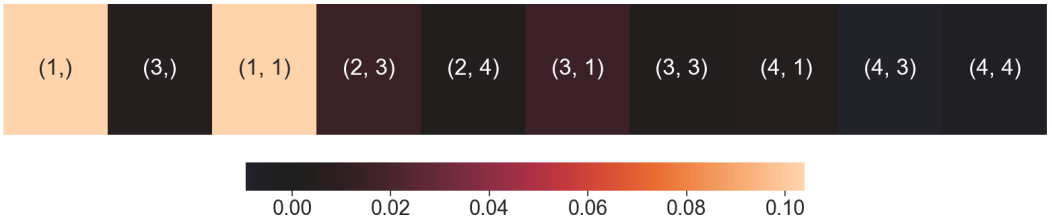

Figure 8: Heatmap of regression coefficients associated with the signature terms for the first two orders of the signature.

This indicates an interaction between RH and $PT08.S4(NO2)$ that plays an important role in predicting $NO2(GT)$ in the next hour. This interaction suggests that an increase in both RH and $PT08.S4(NO2)$ results in a larger measured concentration of $NO2(GT)$. Furthermore, Figure 8 shows that the cumulative $PT08.S4(NO2)$ measured over time, represented by $S^{(4,1)}(X)$, also has a slightly positive corresponding regression coefficient.

The magnitude of the regression coefficients in our model differs somewhat from those reported by Fermanian (5), but this can likely be attributed to the use of different regularisation values.

Appliance Energy Prediction Dataset

Building on our findings from the analysis of the Air Quality dataset, we applied traditional SoFR techniques, as well as the SLM, to the Appliances Energy Prediction dataset, focusing on the performance of the SLM when the truncation order is limited due to the high dimensionality of the data.

The dataset consists of temperature, humidity, energy consumption and other weather data collected over a period of approximately 4.5 months at 10-minute intervals. The temperatures and humidities in different rooms in the house were captured using a ZigBee wireless sensor network every 3.3 minutes and were then averaged into 10-minute intervals. The energy consumption by household appliances and lights was recorded at 10-minute intervals using m-bus energy meters. The outdoor weather conditions data was sourced from the weather station at Chievres Airport in Belgium and integrated with the data captured by the household sensors.

Similar to the Air Quality implementation, we considered both univariate and multivariate cases for the Fourier FLR, B-spline FLR, FPCR, and SLM. For the univariate case, $d = 1$, the temperature outside the house (T_out) for the past 24 hours was used to predict the energy consumed by appliances ($Appliances$) in the house for the next 10 minutes.

We considered two multivariate cases: $d = 8$ and $d = 9$. For $d = 8$, in addition to T_out, we included the temperatures from seven of the rooms in the house ($T1 - T5$ and $T8 - T9$) to predict $Appliances$ in the next 10 minutes. For $d = 9$, $T7$ was also included as a functional predictor alongside the functional predictors used for $d = 8$.

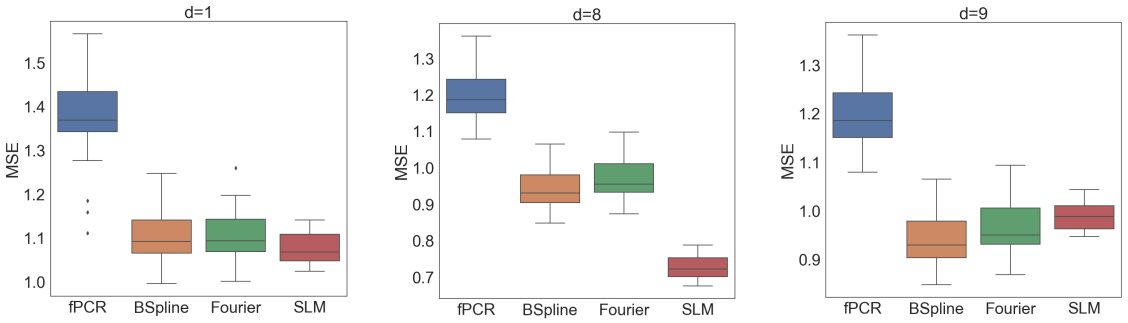

Figure 9: Test MSE distributions for the regression models fitted to the Appliance Energy Prediction dataset (15) using $d = 1, 8, 9$.

The distribution of the MSEs in Figure 9 above indicates that, in the univariate case, the B-spline FLR, Fourier FLR and SLM models perform approximately equally, with the FPCR performing the worst.

Figure 9 shows that for the multivariate case, when $d = 8$ is chosen, the SLM has the best performance amongst the models. Notably, we observe that when $d = 8$, the truncation order selected was $\hat{m} = 3$ using Fermanian's truncation order estimation method (5). Adding another functional predictor, thereby increasing the dimensionality to $d = 9$, results in the truncation order being limited to $\hat{m} = 2$. The impact of this selected truncation order is reflected in the SLM's performance, as the test MSEs when $d = 9$ show an increase compared to $d = 8$, resulting in the SLM failing to outperform the B-spline FLR and the Fourier FLR models.

Upon closer inspection of Fermanian's truncation order estimation method, we observed that the maximum truncation order (max_k) of the signature is determined using the following equation:

$$max\_k = \left\lfloor \frac{\log(max\_features \times (d-1)+1}{\log(d)} - 1 \right\rfloor,$$

where k is the truncation order of the signature, d is the dimensionality of the time-augmented path and $max\_features$ is the size of the vector of coefficients (5). Fermanian sets $max\_features = 1000$.

We considered increasing $max\_features = 2000$, which not only added complexity to the model by allowing for more regression coefficients but also enabled an increase in max_k from 2 to 3. By doing so, the truncation order is selected as $\hat{m} = 3$ using Fermanian's estimation method. Figure 10 shows that by allowing a larger truncation order, the SLM model with d increased to $d = 11$ was able to extract additional information from the higher-dimensional data, resulting in lower test MSEs. Thus, we observe that increasing the model complexity by allowing a higher $\hat{m}$ has improved model performance at the cost of higher computational capacity.

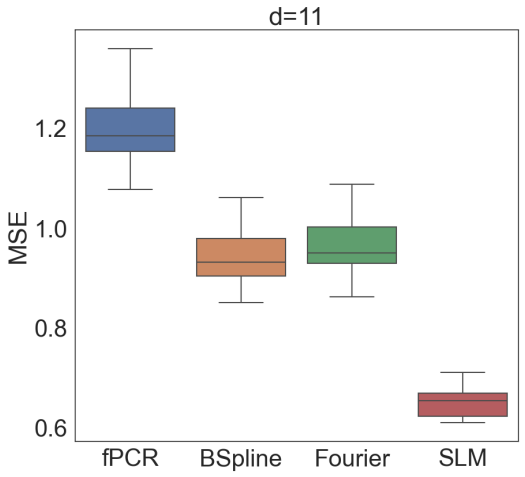

Figure 10: Test MSE distributions for the regression models fitted to the Appliance Energy Prediction dataset using $d = 11$ and a modified truncation order estimation method.

As with the analysis on the Air Quality dataset, the regression coefficients $\hat{\boldsymbol{\beta}}_{\hat{m}}$ for the $d = 11$ SLM fitted to the Appliances Energy Prediction dataset can also be plotted as a heatmap. However, due to the high dimensionality, only the regression coefficients corresponding to the first two orders of the signature and those with $\hat{\beta}_{\hat{m}} \geq 0.35$ are plotted. This highlights the signature terms that are of most importance in the SLM.

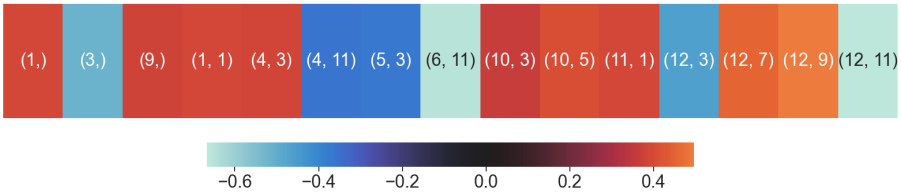

Figure 11: Heatmap of the regression coefficients corresponding to the signature terms for the first two orders of the signature.

In Figure 11, large regression coefficients are assigned to the signature terms $S^{(1)}(X)$ and $S^{(1,1)}(X)$, which represent the variation in outside temperature over the past day. It is reasonable to expect that we can use the outside temperature to predict household appliance energy consumption.

The term $S^{(4,3)}(X)$, representing the area under the curve for the temperatures in the laundry room ($T3$) and the living room ($T2$), is also assigned a large regression coefficient. This suggests that if the temperatures inside both rooms increase, household appliance energy rises. This is logical, as higher temperatures often lead to the increased use of appliances such as air conditioning.

Unsurprisingly, some cumulative temperatures across different rooms in the house are associated with large regression coefficients. Notably, the teenager's room ($T8$), represented by the signature coefficient $S^{(12,9)}(X)$, has a regression coefficient ≥ 0.4, indicating its strong contribution to predicting the appliance energy consumption.

5 Conclusion

In this project, we synthesised key concepts from FDA, FLR, path signatures and the SLM into a comprehensive and cohesive narrative, making Fermanian's work accessible to readers with an honours-level background in statistics.

We began by introducing FDA, distinguishing it from traditional multivariate analysis, and illustrating how functional data can be represented and analysed using basis expansion techniques like Fourier, B-splines and FPCA. Next, we explored three functional linear regression models, emphasising SoFR the primary FLR model used in Fermanian's paper. We discussed the assumptions and limitations of FDA and FLR, providing a rationale for the SLM as a less restrictive approach. We then introduced path signatures, examining key properties and propositions, which motivate and support the use of the SLM. The comparison of the SLM and SoFR models highlighted areas where the SLM should theoretically outperform traditional SoFR.

In the empirical section, we implemented both the SLM and traditional FLR models (SoFR with B-splines and Fourier bases, and FPCR), closely following Fermanian's methodology and applying them to two real-world datasets: the Air Quality dataset originally used by Fermanian (5; 25) and the Appliance Energy Prediction dataset (15), which allowed for high-dimensional predictors.

Our findings showed that in simpler, univariate cases, traditional SoFR models, particularly those using B-splines, performed effectively, and the SLM's advanced modelling capabilities were unnecessary. However, in multivariate cases, the SLM excelled in handling high-dimensional data by capturing relationships that FLR models, constrained by assumptions of independence between dimensions, could not. This highlights the critical role of path signatures in capturing additional information at higher dimensions, where traditional methods fall short.

The SLM's superior performance was especially evident in our analysis of the higher-dimensional Appliance Energy Prediction dataset, where it outperformed the SoFR models as the dimensionality increased. This advantage likely stems from the SLM's less restrictive assumptions, allowing it to capture and model interactions that FLR assumptions prevent from being captured.

However, as dimensionality increased further, the SLM's performance became limited by the choice of truncation order, which was influenced by parameters set in Fermanian's truncation order estimation method. Our investigation revealed that tuning these parameters could significantly impact the model's ability to capture important information at higher levels of dimensionality. We suggest that future work could focus on refining the parameter selection within Fermanian's truncation order estimation method to balance model performance and computational feasibility.

Overall, this project demonstrates the SLM's considerable advantages in capturing intricate, non-linear, high-dimensional patterns in data, while highlighting the need for further refinement in truncation parameter selection to ensure optimal performance in practical applications and further establish the SLM as a standard modelling technique.

References

[1] Birgé, L., and Massart, P. Minimal penalties for gaussian model selection. *Probability Theory and Related Fields 138* (2007), 33–73.

[2] Chen, K.-T. Integration of paths–a faithful representation of paths by non-commutative formal power series. *Transactions of the American Mathematical Society 89*, 2 (1958), 395–407.

[3] Chevyrev, I., and Kormilitzin, A. A primer on the signature method in machine learning. *arXiv preprint arXiv:1603.03788* (2016).

[4] de Jong, E. D. Mnist stroke sequence data set, 2016.

[5] Fermanian, A. Functional linear regression with truncated signatures. *Journal of Multivariate Analysis 192* (2022), 105031.

[6] Goldsmith, J., Liu, X., Jacobson, J., and Rundle, A. New insights into activity patterns in children, found using functional data analyses. *Medicine and science in sports and exercise 48*, 9 (2016), 1723.

[7] Graham, B. Sparse arrays of signatures for online character recognition. *arXiv preprint arXiv:1308.0371* (2013).

[8] Gyurkó, L. G., Lyons, T., Kontkowski, M., and Field, J. Extracting information from the signature of a financial data stream. *arXiv preprint arXiv:1307.7244* (2013).

[9] Ivanescu, A. E., Staicu, A.-M., Scheipl, F., and Greven, S. Penalized function-on-function regression. *Computational Statistics 30* (2015), 539–568.

[10] James, G. M., Wang, J., and Zhu, J. Functional linear regression that's interpretable. *The Annals of Statistics* (2009).

[11] Lee, D., and Ghrist, R. Path signatures on lie groups. *arXiv preprint arXiv:2007.06633* (2020).

[12] Levin, D., Lyons, T., and Ni, H. Learning from the past, predicting the statistics for the future, learning an evolving system. *arXiv preprint arXiv:1309.0260* (2013).

[13] Levitin, D. J., Nuzzo, R. L., Vines, B. W., and Ramsay, J. Introduction to functional data analysis. *Canadian Psychology/Psychologie canadienne 48*, 3 (2007), 135.

[14] Liao, S. *Log signatures in machine learning.* PhD thesis, UCL (University College London), 2022.

[15] Luis M. Candanedo, Véronique Feldheim, D. D. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings 140* (2017), 81–97.

[16] Lyons, T., and McLeod, A. D. Signature methods in machine learning. *arXiv preprint arXiv:2206.14674* (2022).

[17] Lyons, T. J. Differential equations driven by rough signals. *Revista Matemática Iberoamericana 14*, 2 (1998), 215–310.

[18] Montesinos López, O. A., Montesinos López, A., and Crossa, J. Functional regression. In *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Springer, 2022, pp. 579–631.

[19] Moore, P., Lyons, T., Gallacher, J., and Initiative, A. D. N. Using path signatures to predict a diagnosis of alzheimer's disease. *PloS one 14*, 9 (2019), e0222212.

[20] Morris, J. S. Functional regression. *Annual Review of Statistics and Its Application 2* (2015), 321–359.

[21] Ramsay, J., Hooker, G., and Graves, S. Functional data analysis with r and matlab. science+ business media. *Inc., New York* (2009).

[22] Ramsay, J. O., and Silverman, B. W. *Functional Data Analysis*. Springer, New York, 1997.

[23] Ramsay, J. O., and Silverman, B. W. *Functional Data Analysis*, 2 ed. Springer Series in Statistics. Springer, New York, NY, 2005.

[24] Reiss, P. T., Goldsmith, J., Shang, H. L., and Ogden, R. T. Methods for scalar-on-function regression. *International Statistical Review 85*, 2 (2017), 228–249.

[25] S. D. Vito, E. Massera, M. P. L. M. G. F. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical* (2008).