IMPERIAL

Imperial College London

Department of Mathematics

Signature Coefficient Recovery via Kernels

Author: Daniil Shmelev (CID: 01857518)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2023-2024

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgements

I would like to thank my supervisor, Dr. Cristopher Salvi, for his unwavering support and insightful guidance throughout the course of this project. His expertise and enthusiasm for rough path theory inspired my initial interest in the subject and motivated me to pursue this thesis.

I would also like to thank my friends and family for their continued support throughout my project.

Abstract

Central to the study of rough path theory is the *signature transform* of a path, a tensor of infinite dimension composed of iterated integrals of the underlying path. The signature poses an effective way to capture information from a path, thanks both to its rich analytic properties and its universality when used as a basis to approximate functions on path space. Whilst a truncated version of the signature can be efficiently computed using Chen's relation [7], there is a lack of efficient methods for computing a single coefficient deep within the signature. We aim to solve this problem by leveraging the *signature kernel*, defined as the inner product of two signature transforms, which is computable efficiently as the solution of a Goursat PDE [18]. By forming a "filter" in signature space with which to take kernels, one can effectively isolate specific groups of signature coefficients and, in particular, a singular coefficient at any depth of the transform. We show that such a filter can be expressed as a linear combination of suitable signature transforms and demonstrate empirically the effectiveness of such an approach.

Contents

List of Figures

List of Tables

Introduction

Recently, the theory of rough paths and the signature transform have become important tools in the processing of sequential data and effective feature selection. When concerned with these problems, the signature transform of a path, consisting of a series of iterated integrals, is an attractive choice of feature map owing to its many analytic and algebraic properties. In practice, the simplicity of the signature of a linear path, combined with an algebraic property called *Chen's relation*, allows us to compute the signature of an input data stream as the concatenation of signatures of linear paths interpolating the data points.

A fundamental result in the study of the signature transform is that of its *universal non-linearity*. That is, the *coefficients* of the signature transform can be viewed as a set of non-linear basis functionals on which we can form a Taylor-like expansion of any continuous function on path space. In certain contexts, such as when the function depends on few coefficients or when considering machine learning models on signature space, we may wish to focus on a small set of signature coefficients. A natural question is how we might compute these specific coefficients within the signature, possibly at a very deep level k of the transform. Naively, one may compute the entire transform up to level k and extract the desired coefficients. However, we note that for a discrete input stream of length L and dimension d, this computation would have a time complexity of $\mathcal{O}(Ld^k)$, and would require the computation of a large number of unused coefficients. A better option is to directly compute the iterated integral which defines the signature coefficient, which can be done in $\mathcal{O}(Lk^2)$ time thanks to the iterative structure of the integral. Whilst an improvement, this method still suffers from a quadratic complexity in the depth of the coefficient k.

The methodology we propose is to compute signature coefficients by taking an inner product of the signature with a suitable filter. If this filter can be formed in signature space, then the resulting inner product is expressible as a linear combination of inner products of signatures, called *signature kernels*. These kernels are readily computable as the solution of a Goursat PDE, allowing practical computation of the filtered signature. Our method will have a base complexity of $\mathcal{O}(L2^k)$, but with the important benefit that the dependence on k can be parallelised away, resulting in an algorithm of complexity $\mathcal{O}(L)$.

We begin in Chapter 1 with a brief overview of fundamental definitions and results. We provide some motivation for our approach to constructing the filter in Chapters 2 and 3 based on a weighted generalisation of the signature kernel called the ϕ-signature kernel. Taking inspiration from these results, in Chapter 4 we propose a practical way to isolate signature coefficients up to permutation of the multi-index. In Chapters 5 and 6, we explore suitable paths from which to construct the filter in such a way that forces a specific permutation of the multi-index. Combined with the previous results, this will give us the desired filter for coefficient isolation. Finally, in Chapters 7 and 8, we provide numerical results to support our methodology, as well as considering potential generalisations of the method to computing more complicated structures.

Chapter 1

The Path Signature Transform

We begin by giving a brief introduction to some fundamental definitions and results. Throughout, we will take $(V, \|\cdot\|_V)$ to be a finite-dimensional Banach space over $\mathbb{R}$, equipped with an inner product $\langle \cdot, \cdot \rangle_V$ and an orthonormal basis $\{e_i : 1 \leq i \leq d\}$ with corresponding dual basis $\{e_i^* : 1 \leq i \leq d\}$.

1.1 The Signature

Before defining the path signature transform, we should define the spaces on which it acts and endow these with consistent inner products. Having done this, we will move on to define the signature, composed of a collection of iterated integrals in $\mathbb{R}$ referred to as the *coefficients* of the transform. The inner products we define will underpin the construction of the *signature kernel*, defined in Section 1.3.

Definition 1.1.1. *Denote by $C_p([a,b], V)$ the space of continuous paths from $[a,b]$ to V of finite p-variation, written $C_p(V)$ when the interval can be inferred from context.*

Two natural operations on $C_p(V)$ are path concatenation

$$(x * y)_t := \begin{cases} x_t, & t \in [a,b], \\ y_t - y_b + x_b, & t \in [b,c] \end{cases}$$

for $x \in C_p([a,b], V)$, $y \in C_p([b,c], V)$, where $x * y \in C_p([a,c], V)$, and the time reversal of a path

$$\overleftarrow{x}_t = x_{a+b-t},$$

for $x \in C_p([a,b], V)$, where $\overleftarrow{x} \in C_p([a,b], V)$.

Definition 1.1.2. *Define $T(V) = \bigotimes_{i=0}^{\infty} V^{\otimes i}$ to be the tensor algebra of formal polynomials over V endowed with the usual operations of $+$ and $\otimes$.*

Other than the tensor product $\otimes$, we may also define the *shuffle product* on $T(V)$.

Definition 1.1.3 (Shuffle Product [5, Definition 1.3.9])**.** *Define $\sqcup\!\sqcup : T(V) \times T(V) \to T(V)$ inductively by*

$$\begin{aligned} u \sqcup\!\sqcup r = r \sqcup\!\sqcup u = ru, \quad \forall r \in \mathbb{R}, u \in V, \\ u \sqcup\!\sqcup v = (u_- \sqcup\!\sqcup v) \otimes a + (u \sqcup\!\sqcup v_-) \otimes b \end{aligned}$$

for any $u \in V^{\otimes n}$, $v \in V^{\otimes m}$ of the form $u = u_- \otimes a$, $v = v_- \otimes b$ for $a, b \in V$. The function $\sqcup\!\sqcup$ then extends uniquely to an algebra product on $T(V) \times T(V)$ by linearity.

We will see the relevance of this product once we have defined the signature transform.

Definition 1.1.4 (Hilbert-Schmidt inner product)**.** *Given an inner product $\langle \cdot, \cdot \rangle_V$ on V, equip $V^{\otimes k}$ with the inner product*

$$\langle u, v \rangle_{V^{\otimes k}} = \prod_{i=1}^{k} \langle u_i, v_i \rangle_V$$

for any $u = u_1 \cdots u_k$, $v = v_1 \cdots v_k \in V^{\otimes k}$ and define an inner product on $T(V)$ by

$$\langle A, B \rangle_{T(V)} = \sum_{k=0}^{\infty} \langle A_k, B_k \rangle_{V^{\otimes k}}$$

for any $A = (A_0, A_1, \ldots)$, $B = (B_0, B_1, \ldots) \in T(V)$. Define the extended tensor algebra $T((V))$ to be the completion of $T(V)$ with respect to $\langle \cdot, \cdot \rangle_{T(V)}$.

Definition 1.1.5 (Signature Transform)**.** *Let $x \in C_p([a, b], V)$ for $p \in [1, 2)$. Then for any $[s, t] \subseteq [a, b]$, define the k^{th} level of the signature transform as the iterated Young integral*

$$S(x)^{(k)}_{[s,t]} = \int_{s<t_1<\cdots<t_k<t} dx_{t_1} \otimes dx_{t_2} \otimes \cdots \otimes dx_{t_k} \in V^{\otimes k}$$

and define the signature transform of x on $[s, t]$ to be the formal series

$$S(x)_{[s,t]} = \left(1, S(x)^{(1)}_{[s,t]}, \ldots, S(x)^{(k)}_{[s,t]}, \ldots\right) \in T((V)).$$

We will often drop the subscript $[s, t]$ when it is clear from context.

Definition 1.1.6 (Signature Coefficient)**.** *Let $I = (i_1, \ldots, i_k)$ be a multi-index of integers in $\{1, \ldots, d\}$. Let $e^*_I := e^*_{i_1} \cdots e^*_{i_k} \in (V^*)^{\otimes k}$. By viewing e^*_I as an element of $T((V))^*$, define the scalar*

$$S(x)^I = e^*_I(S(x)) \in \mathbb{R},$$

which we may write as the iterated integral

$$\int_{s<t_1<\cdots<t_k<t} dx^{(i_1)}_{t_1} dx^{(i_2)}_{t_2} \cdots dx^{(i_k)}_{t_k}, \tag{1.1.1}$$

where $x^{(i)}$ denotes the i^{th} channel of the path $x_t = (x^{(1)}, \ldots, x^{(d)})_t$.

1.2 Properties of the Signature Transform

It is not immediately clear why iterated integrals should be used to represent a path. As it turns out, the signature has several analytic and algebraic properties which make it well suited as a *"feature set"* of a path. The first of these is invariance to reparametrisations of time.

Proposition 1.2.1 (Reparameterisation invariance [5, Lemma 1.2.1])**.** *Let $x \in C_p([a, b], V)$ for $p \in [1, 2)$ and let $\lambda : [c, d] \to [a, b]$ be a continuous non-decreasing surjection. Then $S(x)_{[a,b]} = S(x \circ \lambda)_{[c,d]}$.*

This invariance removes an infinite dimensional group of symmetries on path space. The presence of this symmetry can often cause problems in feature extraction, making the signature transform an appealing choice of feature map. In cases where the parametrisation of time is important, such as financial data, time can be added as a channel of the path by taking $\tilde{x}_t = (t, x_t)$. $\tilde{x}$ is then referred to as the *time-augmented path.* Given the infinite-dimensional nature of the signature, one may question its practical use if it cannot be computed fully. Luckily, the terms of the transform decay factorially, meaning lower levels of the signature capture a large proportion of the information.

Lemma 1.2.2 (Factorial Decay [15, Lemma 5.1]). *Let $x \in C_1([a,b],V)$ and $k \in \mathbb{N}$. Then*

$$\left\| S(x)^{(k)} \right\|_{V^{\otimes k}} \leq \frac{\|x\|_{1,[a,b]}^k}{k!},$$

where $\|x\|_{1,[a,b]}$ is the 1-variation of x on $[a,b]$.

The final important analytic property that we might require is uniqueness of the transform. This holds up to so-called *tree-like equivalence* of paths.

Definition 1.2.3 (Tree-Like Equivalence [11, Definition 1.3]). *$x \in C_1([a,b],V)$ is said to be Lipschitz tree-like if there exists a continuous function $h : [a,b] \to \mathbb{R}^+$ of bounded variation such that $h(a) = h(b) = 0$ and*

$$\|x_t - x_s\|_V \leq h(s) + h(t) - 2 \inf_{u \in [s,t]} h(u). \tag{1.2.1}$$

For $x, y \in C_1([a,b],V)$, say x and y are tree-like equivalent and write $x \sim y$ if $x * \overleftarrow{y}$ is Lipschitz tree-like. Then $\sim$ defines an equivalence relation on $C_1([a,b],V)$.

A tree-like path can be thought of as a path which retraces its trajectory exactly back to the start point. The function h can then be interpreted as a *"height function"* measuring the distance of a point along the tree from the start point or *"base node"*. For a tree-like path, if the *"height"* at time s and t is equal and the path only goes deeper into the tree between s and t, that is

$$h(s) = h(t) = \inf_{u \in [s,t]} h(u),$$

then $x_s = x_t$. The above condition is equivalent to Condition (1.2.1). The notion of tree-like equivalence is extended in [3] to *weakly geometric rough paths*, which includes the $p \in (1,2)$ case.

Definition 1.2.4. *Let $\sim$ denote the tree-like equivalence relation on $C_p([a,b],V)$, as defined in [3; 11]. Then $\mathcal{C}_p([a,b],V) = C_p([a,b],V)/\sim$ is the class of unparametrised paths of finite p-variation.*

With these concepts in hand, the signature can be shown to be unique on unparametrised paths.

Theorem 1.2.5 (Injectivity on $\mathcal{C}_p$ [3]). *Let $x, y \in C_p(V)$ for $p \in [1,2)$. Then $S(x) = S(y)$ if and only if $x \sim y$. If $p = 1$, then every equivalence class $[x] \in \mathcal{C}_1(V)$ has an element of minimal length, called the tree-reduced representative.*

As with time-reparametrisations, if we wanted to distinguish between two paths which are tree-like equivalent then we could instead consider the time-augmented paths. Having appreciated the rich analytic properties of the signature, it is important to consider how one might efficiently compute signatures in practice. For a piecewise linear interpolation of data, this is straightforward thanks to the simplicity of the signature of a linear path and an algebraic property called Chen's relation.

Proposition 1.2.6 ([5, Section 1.3.1]). *Let $x \in C_1([a,b],V)$ be the linear path*

$$x_t = x_a + \frac{t-a}{b-a}(x_b - x_a).$$

Let $x_{a,b}$ denote the canonical inclusion of $x_b - x_a \in V$ into $T((V))$. Then

$$S(x)_{[a,b]} = \exp(x_{a,b}),$$

where exp *denotes the tensor exponential*

$$\exp(v) := \sum_{k=0}^{\infty} \frac{v^{\otimes k}}{k!}.$$

Proposition 1.2.7 (Chen's relation [5, Lemma 1.3.1; 7]). *Let $x \in C_p([a,b],V)$ and $y \in C_p([b,c],V)$ for $p \in [1,2)$. Then*

$$S(x * y)_{[a,c]} = S(x)_{[a,b]} \otimes S(y)_{[b,c]},$$

where $x * y$ denotes the path concatenation of x and y.

Another useful algebraic property of the signature is the shuffle identity.

Theorem 1.2.8 (Shuffle Identity [5, Theorem 1.3.10]). *Let $x \in C_p([a,b],V)$ for $p \in [1,2)$ and $f, g \in T((V))^* \cong T(V^*)$. Then*

$$f(S(x)) \cdot g(S(x)) = (f \sqcup\!\sqcup g)(S(x)).$$

1.3 The Signature Kernel

Whilst factorial decay allows us to justify the use of a *truncated* signature transform which is only computed up to a given level, the problem of dimensionality persists, with the k^{th} level of the transform containing d^k many coefficients. A potential solution when comparing the signatures of two paths is to apply a *"kernel trick"* and consider instead the inner product of the signatures as elements of $T((V))$.

Definition 1.3.1 (Signature Kernel). *Let $x \in C_1([a,b],V)$ and $y \in C_1([c,d],V)$. The signature kernel $\mathbf{k}_{x,y} : [a,b] \times [c,d] \to \mathbb{R}$ is given by*

$$\mathbf{k}_{x,y}(s,t) = \left\langle S(x)_{[a,s]}, S(y)_{[c,t]} \right\rangle_{T((V))}.$$

We will often write $\mathbf{k}_{x,y}$ when the kernel is taken over the entire interval.

By applying the Cauchy-Schwarz inequality, we get the following corollary of Lemma 1.2.2.

Corollary 1.2.2.1. Let $x, y \in C_1([a,b],V)$ and $k \in \mathbb{N}$. Then

$$\left| \left\langle S(x)^{(k)}_{[a,b]}, S(y)^{(k)}_{[c,d]} \right\rangle_{V^{\otimes k}} \right| \leq \frac{\|x\|^k_{1,[a,b]} \|y\|^k_{1,[c,d]}}{(k!)^2}.$$

Occasionally, it will be helpful to refer to a truncated signature kernel in which we only consider the signature up to the n^{th} level.

Definition 1.3.2 (Truncated Signature Kernel). *Let $x \in C_1([a,b],V)$ and $y \in C_1([c,d],V)$. The truncated signature kernel $\mathbf{k}^n_{x,y} : [a,b] \times [c,d] \to \mathbb{R}$ is given by*

$$\mathbf{k}^n_{x,y}(s,t) = \sum_{k=0}^{n} \left\langle S(x)^{(k)}_{[a,s]}, S(y)^{(k)}_{[c,t]} \right\rangle_{V^{\otimes k}}.$$

Efficient approaches for computing truncated signature kernels for a linearly interpolated data stream of length L are presented in [13], although these are typically non-linear in L or reliant on low-rank approximations. We may wish to generalise the idea of signature kernels by applying a weighting $\phi(k)$ to the inner product $\langle \cdot, \cdot \rangle_{V^{\otimes k}}$ at level k. We do this by introducing the ϕ-inner product on $T(V)$.

Definition 1.3.3 (ϕ-inner product). *For a given weight function $\phi : \mathbb{N}_0 \to \mathbb{R}^+$, the ϕ-inner product on $T(V)$ is given by*

$$\langle A, B \rangle_\phi = \sum_{k=0}^{\infty} \phi(k) \langle A_k, B_k \rangle_{V^{\otimes k}} \tag{1.3.1}$$

for any $A = (A_0, A_1, \ldots)$, $B = (B_0, B_1, \ldots) \in T(V)$. Define $T_\phi((V))$ to be the completion of $T(V)$ with respect to $\langle \cdot, \cdot \rangle_\phi$.

To define a weighted signature kernel using this inner product, we must first ensure that $T_\phi((V))$ contains the signatures that we wish to take inner products with. This is guaranteed by the next lemma under certain conditions on ϕ.

Lemma 1.3.4 ([5, Lemma 2.1.2]). *Let $\mathcal{S}$ denote the image of $C_1(V)$ under the signature transform. If the function $\phi : \mathbb{N}_0 \to \mathbb{R}^+$ is such that for any $C > 0$ the series $\sum_{k\in\mathbb{N}} \frac{C^k \phi(k)}{(k!)^2}$ converges, then $\mathcal{S} \subset T_\phi((V))$.*

Definition 1.3.5 (ϕ-Signature Kernel). *Let $x \in C_1([a,b],V)$ and $y \in C_1([c,d],V)$. Let ϕ satisfy the condition of Lemma 1.3.4. Then the ϕ-signature kernel $\mathbf{k}^\phi_{x,y} : [a,b]\times[c,d] \to \mathbb{R}$ is given by*

$$\mathbf{k}^\phi_{x,y}(s,t) = \left\langle S(x)_{[a,s]}, S(y)_{[c,t]} \right\rangle_\phi .$$

The untruncated signature kernel is of little practical use if we have no way to compute it. Fortunately, it can be shown to be the solution to a hyperbolic PDE belonging to a class of PDEs called Goursat problems.

Theorem 1.3.6 (Signature Kernel PDE [18, Theorem 2.5]). *Let $x \in C_1([a,b],V)$ and $y \in C_1([c,d],V)$. Then $\mathbf{k}_{x,y}$ is the solution of the Goursat PDE*

$$\frac{\partial^2 \mathbf{k}_{x,y}}{\partial s \partial t} = \langle \dot{x}_s, \dot{y}_t \rangle_V \, \mathbf{k}_{x,y}, \quad \mathbf{k}_{x,y}(a,\cdot) = \mathbf{k}_{x,y}(\cdot,c) = 1, \tag{1.3.2}$$

the existence and uniqueness of a solution to which follows from [14, Theorems 2 & 4].

In [18] it was shown that for paths obtained by linearly interpolating discrete input streams of length L and dimension d, computation of the signature kernel can be achieved in $\mathcal{O}(L^2 d)$ time. The advantage of computing the full signature kernel over the truncated version is that numerical PDE schemes lend themselves well to parallelisation. As such, the complexity may be reduced to $\mathcal{O}(Ld)$ on suitable GPU hardware.

1.4 Universal Nonlinearity

Arguably one of the most important results in the study of the signature transform is that of its *univeral nonlinearity*, which states that continuous functions on path space can be well approximated by linear functionals on the signature.

Theorem 1.4.1 (Universal Nonlinearity [16, Theorem 3.3]). *Given a suitable topology on $\mathcal{C}_1([a,b],V)$, let $\mathcal{K} \subset \mathcal{C}_1([a,b],V)$ be compact and $f : \mathcal{K} \to \mathbb{R}$ a continuous function. Then for any $\varepsilon > 0$ there exists a truncation level $k \in \mathbb{N}$ and $\alpha_{i,I} \in \mathbb{R}$ such that for all $x \in \mathcal{K}$*

$$\left| f(x) - \sum_{i=0}^{k} \sum_{I\in\{1,\dots,d\}^i} \alpha_{i,I} \, S(x)^I_{[a,b]} \right| \leq \varepsilon.$$

This is extended to $\mathcal{C}_p$ in [5, Theorem 1.4.7]. A discussion of suitable topologies on $\mathcal{C}_p$ can be found in [5; 6]. Theorem 1.4.1 provides our main motivation for computing isolated signature coefficients. Suppose for a given function f and tolerance ε we know the coefficients $\alpha_{i,I}$. Moreover, suppose the $\alpha_{i,I}$ mostly vanish. Let x be a piecewise linear path obtained from interpolating an input data stream of length L. To compute $f(x)$ one can, of course, compute $S(x)$ up to level k to get an approximate value for $f(x)$. However, for a path of length L this would have a computational complexity of $\mathcal{O}(Ld^k)$ [12; 17] and would involve unnecessary computation of unused signature coefficients. Alternatively, one could compute the required signature coefficients by directly computing the integral

in Equation (1.1.1). As we will see below, a coefficient at depth k can be computed in this way in $\mathcal{O}(Lk^2)$ time, which is an improvement over computing the entire signature up to level k, but still quadratic in depth k.

The problem we focus on is that of efficiently computing a single coefficient within level k of the signature transform, without needing to compute the entire transform. In the formulation above, this is the case where all but one of the $\alpha_{i,I}$ are zero. Our approach will have a base complexity of $\mathcal{O}(L2^k)$, but with the benefit of being easily parallelisable down to $\mathcal{O}(L)$.

1.5 Practical Motivations

The motivation given in the previous section is highly abstract, so we provide a concrete use case rooted in machine learning.

First introduced in the seminal paper *Attention is all you need* [21], transformer models form the foundation of modern GPT models and have natural applications to time series forecasting in areas such as quantitative finance. At the heart of the transformer model is the *"self-attention"* mechanism, which enables the model to dynamically focus on different parts of the input sequence to capture inherent dependencies. Unlike other models operating on sequential data, such as recurrent neural networks (RNNs) [23], the self-attention mechanism is pivotal in allowing the transformer model to capture long-range dependencies in the sequence. A major bottleneck of the self-attention mechanism is its quadratic complexity in sequence length. Several recent papers [2; 8; 22] have presented methods for linearizing transformers using various approximations of self-attention. Whilst successful in some areas, on average these models underperform the vanilla transformer in long-context tasks, as illustrated in the *Long-Range Arena* (LRA) benchmark [20].

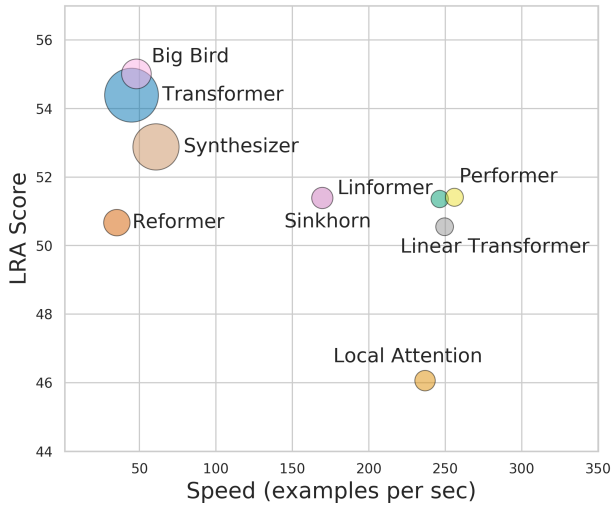


Figure 1.1: LRA benchmark performance and speed of different transformer models. Circle size denotes memory footprint. Source: [20, Figure 3]

When considering long financial time series, a potential solution to the quadratic complexity in sequence length is to consider transformer models on the level of signature coefficients, rather than the underlying sequence. In such a setup, the transformer would focus on specific signature coefficients throughout the training, leveraging signature universality to capture the necessary information from the path. For this to be viable in practice, we must be able to efficiently compute isolated signature coefficients. If this is possible, then we note that, since signature coefficient computation is linear in sequence length, the resulting mechanism would break the quadratic complexity which hinders classical transformers.

1.6 Naive Integration and a Lower Bound on Complexity

A naive approach to computing the signature coefficient is to directly compute the integral (1.1.1). We can exploit the iterative structure of the coefficient by computing recursively

$$S(x)_{[0,t]}^{(i_1,\dots,i_m)} = \int_0^t S(x)_{[0,u]}^{(i_1,\dots,i_{m-1})} dx_u^{(i_m)}$$

for all discretisation points $0 = t_0, t_1, \dots, t_L = 1$, by noting that if x is piecewise linear, then $S(x)_{[0,t]}^{(i_1,\dots,i_m)}$ must be piecewise polynomial of degree m. Since integrating a path which is piecewise polynomial of degree m can be done in $\mathcal{O}(Lm)$ time, we can compute $S(x)^{(i_1,\dots,i_k)}$ recursively in $\mathcal{O}(Lk^2)$ time. At the cost of accuracy, we may also choose to approximate the integral by numerical integration techniques, which can reduce the complexity down to $\mathcal{O}(Lk)$. We should note that, for instance, the signature coefficient $S(x)^{(1,\dots,k)}$ depends on each of the Lk many input data points of the path x. Therefore, a fundamental lower bound on the unparallelised complexity of computing this coefficient must be Lk. We will show that the kernel methods which we develop for computing a signature coefficient allow us to parallelise away the dependence on k, resulting in a complexity of $\mathcal{O}(L)$. Moreover, the tools we develop will extend to computing certain sums of coefficients which may not be easily computable by direct integration.

1.7 General Methodology

Given a multi-index $(i_1, \dots, i_k) \in \{1, \dots, d\}^k$ representing the signature coefficient which we aim to compute, the main idea of the approach is to create a filter $F \in \text{span}(\mathcal{S})$ such that

$$F^{(j_1,\dots,j_m)} = \begin{cases} 1 & \text{if } (j_1, \dots, j_m) = (i_1, \dots, i_k), \\ 0 & \text{otherwise.} \end{cases}$$

We then note that, for $x \in C_1(V)$, $\langle S(x), F \rangle = S(x)^{(i_1,\dots,i_k)}$. If F can be expressed linearly in terms of signature transforms of paths, then we may rewrite the above inner product as a linear combination of signature kernels, which by Theorem 1.3.6 are computable efficiently as solutions of Goursat PDEs. We introduce some definitions which will aid with our construction of F.

Definition 1.7.1. *Let $\mathcal{P}(j_1, \dots, j_m)$ denote the set of multi-indices which are permutations of $(j_1, \dots, j_m)$.*

Example 1.7.1. $\mathcal{P}(1, 2, 2) = \{(1, 2, 2), (2, 1, 2), (2, 2, 1)\}$.

Definition 1.7.2. *For a set $\mathcal{I}$ of multi-indices, let $S(x)^{\mathcal{I}}$ denote the sum $\sum_{I \in \mathcal{I}} S(x)^I$.*

Within our construction of F, we will extensively make use of component-wise path scalings. We will denote these as follows.

Definition 1.7.3. *Let $\lambda = (\lambda_1, \dots, \lambda_d) \in \mathbb{R}^d$. For $z \in C_p(V)$ given by $z_t = \sum_{i=1}^d z_t^{(i)} e_i$, denote by $\lambda \odot z \in C_1(V)$ the path given by $(\lambda \odot z)_t = \sum_{i=1}^d \lambda_i z_t^{(i)} e_i$.*

The problem of forming F using signature transforms can be split into 3 sub-problems:

1. Level isolation: How do we zero all levels of a signature other than the k^{th}?
2. Permutation class isolation: Within level k, how do we zero all coefficients given by multi-indices outside of $\mathcal{P}(i_1, \dots, i_k)$?

3. Order isolation: How do we zero coefficients given by multi-indices within $\mathcal{P}(i_1, \dots, i_k)$ whilst setting the coefficient at index $(i_1, \dots, i_k)$ to 1?

Without loss of generality, we may assume that $(i_1, \dots, i_k) = (1, \dots, k)$ and $\dim(V) = k$. If this is not the case, then one can reorder the channels of the path x, cloning or removing channels as necessary. We make this assumption throughout. In our construction of F, we will consistently make use of the linear path $y \in C_1([0,1], V)$ given by $y_t = t\mathbf{1}$, where $\mathbf{1} = e_1 + \cdots + e_k$. By Proposition 1.2.6, the signature coefficients of y are given by

$$S(y)^{(i_1,\dots,i_m)} = \frac{1}{m!}$$

for any multi-index $(i_1, \dots, i_m)$. The simple form of this signature will allow us to manipulate y and $S(y)$ to form F.

Chapter 2

Motivation: Randomised Scalings and Integral Transforms

We begin with the problems of level and permutation class isolation. Specifically, we aim to compute $S(x)^{\mathcal{P}(1,\dots,k)}$ through signature kernels. To motivate our approach to this, we recall several well-known results concerning integral transforms applied to signature kernels. In particular, these concern expected kernels given a random path scaling and, more generally, integration with respect to (signed) measures.

2.1 Measures and Moment-Weighted Kernels

Proposition 2.1.1 (Moment-Weighted Kernel [4, Proposition 4.3])**.** *Let π be a random variable with finite moments of all orders. Let $\phi(k) = \mathbb{E}[\pi^k]$ and $\psi(k) = \mathbb{E}[|\pi^k|]$ $\forall k \geq 0$ and suppose ψ satisfies the condition of Lemma 1.3.4. Then for any $x, y \in C_1([a,b], V)$ the ϕ-signature kernel is well defined and satisfies*

$$\mathbf{k}^{\phi}_{x,y}(s,t) = \mathbb{E}[\mathbf{k}_{\pi x, y}(s,t)] = \mathbb{E}[\mathbf{k}_{x,\pi y}(s,t)].$$

For the purpose of isolating level k in the signature, we would ideally want a distribution such that $\phi(i) = \delta_{i,k}$, or at least one which has a moment sequence close to this. Unfortunately, the set of probability distributions, or even more generally Radon measures, is not rich enough to produce such a moment sequence, as we demonstrate in the following.

Theorem 2.1.2 (Solution of the Hamburger Moment Problem [19, Theorem 1.12])**.** *For any real sequence $\phi = (\phi(n))_{n\in\mathbb{N}_0}$, the following are equivalent:*

1. *There is a Radon measure μ on $\mathbb{R}$ such that*

$$\phi(n) = \int_{\mathbb{R}} x^n \, d\mu(x), \quad \forall n \in \mathbb{N}_0.$$

   That is, ϕ is a Hamburger moment sequence.

2. *The sequence ϕ is positive semidefinite.*

3. *All Hankel matrices*

$$H_n(\phi) = \begin{pmatrix} \phi(0) & \phi(1) & \cdots & \phi(n) \\ \phi(1) & \phi(2) & \cdots & \phi(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(n) & \phi(n+1) & \cdots & \phi(2n) \end{pmatrix}$$

   for $n \in \mathbb{N}_0$ are positive semidefinite.

4. *The Riesz linear functional L on $\mathbb{R}[x]$ defined by $L(x^n) = \phi(n)$ is positive. That is, $L(p^2) \geq 0$ for all $p \in \mathbb{R}[x]$.*

Corollary 2.1.2.1. For $\varepsilon \geq 0$, let $\phi_{k,\varepsilon}$ be a sequence such that $\phi_{k,\varepsilon}(k) = 1$ and $\phi_{k,\varepsilon}(n) < \varepsilon$ for all $n \geq 1$, $n \neq k$. Then there exists $\varepsilon_0 > 0$ such that $\phi_{k,\varepsilon}$ is not a Hamburger moment sequence for any $\varepsilon \leq \varepsilon_0$.

Proof. Consider the cases for $\phi_{k,\varepsilon}(0)$.

- If $\phi_{k,\varepsilon}(0) < 0$, then clearly $\phi_{k,\varepsilon}$ cannot be the moment sequence of an unsigned measure.
- If $\phi_{k,\varepsilon}(0) = 0$, let $p = x^k - 1 \in \mathbb{R}[x]$. Then

  $$L(p^2) = L(x^{2k} - 2x^k + 1) < \varepsilon - 2.$$

  Thus, choosing $\varepsilon \leq \varepsilon_0 = 2$ gives $L(p^2) < 0$, and so by Theorem 2.1.2, $\phi_{k,\varepsilon}$ is not a Hamburger moment sequence for $\varepsilon \leq \varepsilon_0$.
- If $\phi_{k,\varepsilon}(0) > 0$, let $p = x^k - \phi_{k,\varepsilon}(0)^{-1} \in \mathbb{R}[x]$. Then

  $$L(p^2) = L(x^{2k} - 2\phi_{k,\varepsilon}(0)^{-1}x^k + \phi_{k,\varepsilon}(0)^{-2}) < \varepsilon - \phi_{k,\varepsilon}(0)^{-1}.$$

  Thus, choosing $\varepsilon \leq \varepsilon_0 = \phi_{k,\varepsilon}(0)^{-1}$ gives $L(p^2) < 0$, and so by Theorem 2.1.2, $\phi_{k,\varepsilon}$ is not a Hamburger moment sequence for $\varepsilon \leq \varepsilon_0$.

□

2.2 Signed Measures

Whilst unsigned measures are insufficient, signed measures are rich enough to at least produce approximate behaviour, as we will show below. First, however, we should note that Proposition 2.1.1 generalizes to finite signed Borel measures. The following theorem is presented in [4, Theorem 4.11].

Theorem 2.2.1. *Let μ be a finite signed Borel measure on $\mathbb{R}$. Let $\phi : \mathbb{R} \to \mathbb{R}$ be such that*

$$\phi(m) = \int_{\mathcal{D}} \pi^m \mu(d\pi), \quad \forall m \in \mathbb{N}_0$$

for a suitable domain $\mathcal{D} \subseteq \mathbb{R}$. Let $x, y \in C_1([a,b], V)$. For $s, t \in [a,b]$, $m \in \mathbb{N}_0$ define

$$a_m(s,t) = \left\langle S(x)^{(m)}_{[a,s]}, S(y)^{(m)}_{[a,t]} \right\rangle_{V^{\otimes m}}$$

and assume that $\forall s, t \in [a,b]$

1. *$\int_{\mathcal{D}} |\pi^m \mu(d\pi)| < \infty$ for all $m \geq 0$, and*
2. *$\sum_{m \geq 0} a_m(s,t) \int_{\mathcal{D}} |\pi^m \mu(d\pi)|$ converges absolutely,*

then

$$\mathbf{k}^{\phi}_{x,y}(s,t) = \int_{\mathcal{D}} \mathbf{k}_{\pi x, y}(s,t)\mu(d\pi) = \int_{\mathcal{D}} \mathbf{k}_{x, \pi y}(s,t)\mu(d\pi).$$

The proof of Theorem 2.2.1 follows simply by noting that conditions (1) and (2) allow for an application of Fubini's Theorem to interchange the sum in the ϕ-inner product and the integral with respect to μ. Proposition 2.1.1 then follows as a direct corollary. To see that signed Borel measures can approximate the desired behaviour for signature level isolation, we take ψ_σ to be a centered Gaussian density with standard deviation σ and let $\mu_{k,\sigma}$ be the finite signed Borel measure given by

$$\mu_{k,\sigma}(A) = \int_A \frac{(-1)^k}{k!}\psi_\sigma^{(k)} d\Lambda \quad \forall A \in \mathcal{B}(\mathbb{R}),$$

where Λ is the Lebesgue measure. The following result shows that this class of measures is suitable for isolating the k^{th} level of the signature.

Proposition 2.2.2. *Let $\mathcal{D} = [-1, 1]$ in Theorem 2.2.1. For any $\varepsilon > 0$ and integer $k \geq 1$, there exists $\sigma > 0$ such that the measure $\mu_{k,\sigma}$ satisfies the conditions of Theorem 2.2.1 for any choice of $x, y \in C_1(V)$ and has moment sequence*

$$\phi_{k,\sigma}(m) = \int_{-1}^{1} \pi^m \mu_{k,\sigma}(d\pi)$$

such that $|\phi_{k,\sigma}(k) - \delta_{m,k}| < \varepsilon$ for all $m \in \mathbb{N}_0$.

Proof. See Appendix A, Section A.1. □

This allows us to achieve approximate level isolation with signed measures. In fact, we can extend this beyond just level isolation by considering a component-wise scaling of the path. If we scale each component of a path by an independent variable and then take signatures followed by integrals with respect to the signed measure $\mu_{1,\sigma}$, then by construction we are left with precisely those signature coefficients which contain each component of the path exactly once, that is, the signature coefficients given by multi-indices in $\mathcal{P}(1, \dots k)$. We formalise this below by applying said scaling to y and taking signature kernels with x to compute $S(x)^{\mathcal{P}(1,\dots,k)}$.

Corollary 2.2.1.1. Let $x \in C_1([0, 1], V)$ and let $y \in C_1([0, 1], V)$ be the linear path given by $y_t = \mathbf{1}t$. Let μ_σ be the measure on $[-1, 1]^k$ defined as the product measure $\mu_{1,\sigma}^{\otimes k}$. Then for all $\varepsilon > 0$ there exists $\sigma > 0$ such that

$$\left| k! \int_{[-1,1]^k} \mathbf{k}_{x,\pi\odot y}\, \mu_\sigma(d\pi) - S(x)^{\mathcal{P}(1,\dots,k)} \right| < \varepsilon.$$

Proof. By Fubini's Theorem and linearity, one can interchange integrals and inner products to consider the integral on levels of the signature of $\pi \odot y$. One then notes that by Proposition 2.2.2,

$$\left| k! \int S(\pi \odot y)^{(i_1,\dots,i_m)} \mu_\sigma(d\pi) - \mathbf{1}\left\{(i_1, \dots, i_m) \in S(x)^{\mathcal{P}(1,\dots,k)}\right\} \right| < \varepsilon.$$

For a detailed proof, see Appendix A, Section A.2. □

Whilst these results achieve the goal of permutation class isolation, the integral transforms used are particularly difficult to compute in practice due to dimensionality issues and the limiting behaviour of μ_σ as $\sigma \to 0$. Nonetheless, we now have an idea of how to approach permutation class isolation via path scalings.

Chapter 3

Signature Coefficients as Derivatives

In the above discussion, ψ_σ was intentionally chosen to be a nascent delta function, since the moment sequence when integrating against the Dirac delta derivative $\delta^{(k)}$ exhibits precisely the behaviour that we are looking for due to the property

$$\int_{\mathbb{R}} g(u)\delta^{(k)}(u)du = (-1)^k g^{(k)}(0)$$

for any suitable smooth function g. We may therefore bypass the measure theoretic construction and state the result directly as a derivative.

Proposition 3.0.1. *Let $x \in C_1([0,1],V)$ and let $y \in C_1([0,1],V)$ be the linear path $y_t = t\mathbf{1}$. Let $\lambda = (\lambda_1,\dots,\lambda_k) \in \mathbb{R}^k$. Then*

$$\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\mathbf{k}_{x,\,\lambda\odot y}\bigg|_{\lambda=0} = \frac{1}{k!}\,S(x)^{\mathcal{P}(1,\dots,k)}. \tag{3.0.1}$$

Proof. The proof follows from the fact that

$$S(\lambda\odot z)^{(i_1,\dots,i_m)} = S(z)^{(i_1,\dots,i_m)}\prod_{j=1}^{m}\lambda_{i_j}.$$

The full proof is detailed in Appendix A, Section A.3. □

Remark 3.0.2. *Let E be the Banach space $E = \{v \in T((V)) : \|v\| < \infty\}$, and $S^{-1}(E) \subset C_1(V)$ be the pre-image of E under the signature transform acting on $C_1(V)$. Then by considering the restriction of S to $S^{-1}(E)$, we can restate the above result without relying on deterministic scalings by considering instead the Gateaux derivative*

$$D^k S(0)\{x^{(1)}e_1,\dots,x^{(k)}e_k\},$$

where $x^{(i)}e_i$ is understood to be the path $(x_t^{(i)}e_i)_t \in C_1([0,1],V)$. Intuitively, we see that the only coefficients remaining after evaluating such a derivative are those of first order in $x^{(1)},\dots,x^{(k)}$, that is, precisely those given by multi-indices in $\mathcal{P}(1,\dots,k)$.

As a consequence of Proposition 3.0.1, we obtain the approximation using a forward finite difference

$$\left|\frac{k!}{h^k}\sum_{H\in\{0,h\}^k}(-1)^{\mathrm{sgn}(H)}\,\mathbf{k}_{x,H\odot y} - S(x)^{\mathcal{P}(1,\dots,k)}\right| = o(h), \tag{3.0.2}$$

where $\text{sgn}(H) = \sum_i \mathbf{1}\{H_i = 0\}$. This is obvious intuitively if the sum is brought into the inner product of the kernel and we instead look at

$$\frac{k!}{h^k} \sum_{H \in \{0,h\}^k} (-1)^{\text{sgn}(H)} S(H \odot y) \tag{3.0.3}$$

to consider the sum component-wise in the signature.

Example 3.0.1. *Let $k = 3$ and consider the $(1,2,3)$ coefficient of the sum in (3.0.3). We get*

$$\begin{aligned}
\frac{3!}{h^3} \sum_{H \in \{0,h\}^3} (-1)^{sgn(H)} S(H \odot y)^{(1,2,3)} &= \frac{3!}{h^3} \sum_{H \in \{0,h\}^3} (-1)^{sgn(H)} H_1 H_2 H_3 S(y)^{(1,2,3)} \\
&= \prod_{i=1}^{3} \left(\frac{1}{h} \sum_{H_i \in \{0,h\}} (-1)^{\mathbf{1}\{H_i = 0\}} H_i \right) \\
&= 1,
\end{aligned}$$

whereas the $(1,2,3,2)$ coefficient is

$$\begin{aligned}
\frac{3!}{h^3} \sum_{H \in \{0,h\}^3} (-1)^{sgn(H)} S(H \odot y)^{(1,2,3,2)} &= \frac{3!}{h^3} \sum_{H \in \{0,h\}^3} (-1)^{sgn(H)} H_1 H_2^2 H_3 S(y)^{(1,2,3,2)} \\
&= \frac{1}{4} h,
\end{aligned}$$

and the $(1,1,2)$ coefficient is

$$\begin{aligned}
\frac{3!}{h^3} \sum_{H \in \{0,h\}^3} (-1)^{sgn(H)} S(H \odot y)^{(1,2,2)} &= \frac{3!}{h^3} \sum_{H \in \{0,h\}^3} (-1)^{sgn(H)} H_1 H_2^2 S(y)^{(1,2,2)} \\
&= 0.
\end{aligned}$$

As mentioned in Section 1.3, the signature kernel can be computed as the solution of a PDE in $\mathcal{O}(Lk)$ time, giving the approximation in Equation (3.0.2) a computational complexity of $\mathcal{O}(Lk2^k)$. Whilst this is an improvement over the integral transform approaches in terms of computational ease, computing cross-derivatives of high order is still a notoriously difficult numerical task and often unstable as $h \to 0$. At the cost of a slightly higher computational complexity, we will offer a way to mitigate this instability in the next section.

Chapter 4

Vandermonde Systems

Notice that the finite difference approximation in Equation (3.0.3) is exact on linear functions. Thus, by considering component-wise sums as in Example 3.0.1, we see that for any choice of h the sum still gives the exact desired behaviour up to the k^{th} level. Past this, we get unwanted non-zero coefficients such as the $(1, 2, 3, 2)$ coefficient in Example 3.0.1. To mitigate the numerical instability associated with taking high dimensional derivatives, we fix $h = 1$ and look for another way to zero the higher levels. We could, of course, disregard higher levels of the signature entirely and simply consider the truncated signature kernel up to level k.

Proposition 4.0.1. *Let $x \in C_1([0,1], V)$ and let $y \in C_1([0,1], V)$ be the linear path $y_t = t\mathbf{1}$. Then*

$$k! \sum_{\lambda \in \{0,1\}^k} (-1)^{sgn(\lambda)} \mathbf{k}^k_{x,\, \lambda \odot y} = S(x)^{\mathcal{P}(1,\dots,k)}, \tag{4.0.1}$$

where $sgn(\lambda) = \sum_i \mathbf{1}\{\lambda_i = 0\}$ and $\mathbf{k}^k$ is the truncated signature kernel up to level k.

Proof. The result follows immediately from Proposition 3.0.1 and Equation (3.0.2) by noting that the forward difference approximation of $\partial/\partial\lambda_i$ is exact on linear functions of λ_i. □

It can be useful to reformulate the above in terms of a general weighted signature kernel, as we define below.

Definition 4.0.2. *For a weight function $\omega : \mathbb{N} \to \mathbb{R}^+$, let $\langle \cdot, \cdot \rangle_\omega$ denote the weighted inner product on V given by*

$$\langle u, v \rangle_\omega = \sum_{i=1}^{d} \omega(i) u_i v_i,$$

for any $u = (u_1, \dots, u_d)$, $v = (v_1, \dots, v_d) \in V$. Let $\langle \cdot, \cdot \rangle_{\omega\text{-}\phi}$ denote the ϕ-inner product on $T(V)$ constructed under the inner product $\langle \cdot, \cdot \rangle_\omega$ on V.

It is easy to see that if $(V, \|\cdot\|_V)$ is a Banach space endowed with an inner product $\langle \cdot, \cdot \rangle_V$ with respect to which e_i is an orthonormal basis, then $(V, \|\cdot\|_\omega)$ is a Banach space for any weight function $\omega : \mathbb{N} \to \mathbb{R}^+$, where $\|\cdot\|_\omega$ is the norm induced by $\langle \cdot, \cdot \rangle_\omega$.

Definition 4.0.3 (ω-ϕ Signature Kernel)**.** *Let $x \in C_1([a,b], V)$ and $y \in C_1([c,d], V)$. Let ϕ satisfy the condition of Lemma 1.3.4. We define the ω-ϕ signature kernel $\mathbf{k}^{\omega\text{-}\phi}_{x,y} : [a,b] \times [c,d] \to \mathbb{R}$ given by*

$$\mathbf{k}^{\omega\text{-}\phi}_{x,y}(s,t) = \left\langle S(x)_{[a,s]}, S(y)_{[c,t]} \right\rangle_{\omega\text{-}\phi}.$$

Denote by $\mathbf{k}^{\omega\text{-}\phi,n}_{x,y}(s,t)$ the ω-ϕ signature kernel truncated at level n.

Remark 4.0.4. *Let $\phi(i) = \beta^i$ for some $\beta \in \mathbb{R}^+$. Then the ω-ϕ signature kernel can be reduced to the signature kernel*

$$\mathbf{k}^{\omega\text{-}\phi}_{x,y}(s,t) = \mathbf{k}_{x,\ \beta\omega\odot y}(s,t) = \mathbf{k}_{\beta\omega\odot x,\ y}(s,t),$$

in which case $\mathbf{k}^{\omega\text{-}\phi}_{x,y}(s,t)$ is easily computable as the solution of a Goursat PDE by Theorem 1.3.6. In this case, we will write $\mathbf{k}^{\omega\text{-}\beta}$ to mean $\mathbf{k}^{\omega\text{-}\phi}$ with $\phi(i) = \beta^i$.

Viewing the vector λ as a function $\lambda(i) = \lambda_i$, Proposition 4.0.1 can then be rewritten as

$$k! \sum_{\lambda \in \{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)} \mathbf{k}^{\lambda\text{-}\mathbf{1},k}_{x,y} = S(x)^{\mathcal{P}(1,\dots,k)}. \tag{4.0.2}$$

If $\langle \cdot, \cdot \rangle_\lambda$ is viewed as the inner product taken after an orthogonal projection $x \mapsto \lambda \odot x$, then Equation (4.0.2) can be interpreted as a signed sum of signature kernels formed under orthogonal projections onto lower dimensional space. Whilst this is an exact form for $S(x)^{\mathcal{P}(1,\dots,k)}$ in terms of truncated kernels, one might prefer to consider untruncated kernels as these are easier to compute as PDE solutions. If we could zero the first few levels after the k^{th}, we may then rely on the factorial decay of signature coefficients to minimize the error from higher levels. To do this, we return to the idea of random path scalings for inspiration. Suppose we hope to scale the path by a random variable π, such that π has moment sequence satisfying $\phi(k) = 1$ and $\phi(n) = 0$ for all $k < n \leq k + M$, where $M \geq 0$ is a parameter which we call the *depth* of the scaling. For levels deeper than $k + M$, we rely on the factorial decay in Lemma 1.2.2 to keep the error low. Suppose π is finitely supported on the set $\{\beta_0, \dots, \beta_M\}$ and let $\alpha_i = \mathbb{P}(\pi = \beta_i)$. Then we require that

$$\begin{aligned} \sum_{i=0}^{M} \alpha_i &= 1, \\ \sum_{i=0}^{M} \alpha_i \beta_i^m &= \delta_{m,k}, \quad \forall k \leq m \leq k + M. \end{aligned}$$

In fact, we may drop the requirement that α_i define a probability distribution, and instead of the expectation of a random variable we consider a general weighted sum. We may then rewrite the above equations as the Vandermonde matrix equation

$$B_{k,M} \cdot \alpha = \begin{pmatrix} \beta_0^k & \beta_1^k & \cdots & \beta_M^k \\ \beta_0^{k+1} & \beta_1^{k+1} & \cdots & \beta_M^{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_0^{k+M} & \beta_1^{k+M} & \cdots & \beta_M^{k+M} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_M \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{4.0.3}$$

Such generalized Vandermonde matrices $B_{k,M}$ are invertible for distinct $\beta_i > 0$. An explicit inversion is given by the following.

Proposition 4.0.5. *For distinct $\beta_i > 0$, an explicit solution to Equation* (4.0.3) *is given by*

$$\alpha_i = \frac{(-1)^M}{\beta_i^k} \prod_{\substack{j=0 \\ j \neq i}}^{M} \frac{\beta_j}{\beta_i - \beta_j}.$$

Proof. See [1] and Appendix A, Section A.4. □

Note that if any of the β_i are chosen greater than 1, there will be signature coefficients in levels beyond the $(k+M)^{th}$ which are scaled by a high power of β_i, which may cause high error. To exclude this possible source of error, the β_i should be chosen in $(0,1]$ to ensure that β_i^m does not grow as $m \to \infty$. Beyond that, the method is not particularly sensitive to the choice of β_i as long as these are chosen reasonably, as per the conditions of Proposition 4.0.6 below. Combining this scaling with Equation (3.0.2) for $h = 1$, we get:

Proposition 4.0.6. *Fix $k \geq 1$ and let $\beta_i = \beta_i(M) \in (0,1]$ be chosen such that for any positive constant $C > 0$,*

$$\max_{0\leq i\leq M}|\alpha_i| = \max_{0\leq i\leq M} \frac{1}{\beta_i^k} \prod_{\substack{j=0 \\ j\neq i}}^{M} \frac{\beta_j}{|\beta_i - \beta_j|} = \mathcal{O}\left(\frac{[(k+M)!\,]^2}{M} C^M\right) \tag{4.0.4}$$

as $M \to \infty$. Let $y \in C_1([0,1], V)$ be the linear path $y_t = t\mathbf{1}$. Then for any $x \in C_1([0,1], V)$,

$$k! \sum_{i=0}^{M} \sum_{\lambda\in\{0,1\}^k} (-1)^{sgn(\lambda)} \alpha_i \, \mathbf{k}_{x,y}^{\lambda\text{-}\beta_i} \to S(x)^{\mathcal{P}(1,\ldots,k)} \tag{4.0.5}$$

as $M \to \infty$, where $sgn(\lambda) = \sum_i \mathbf{1}\{\lambda_i = 0\}$.

Proof. Since V is equipped with an inner product, we have $\|\lambda \odot y\|_1 \leq \|y\|_1 = \sqrt{k}$ for all $\lambda \in \{0,1\}^k$ (see the proof of Proposition 2.2.1.1 in Appendix A, Section A.2). Then by Proposition 4.0.1,

$$\begin{aligned}
&\left| k! \sum_{i=0}^{M} \sum_{\lambda\in\{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)} \alpha_i \, \mathbf{k}_{x,y}^{\lambda\text{-}\beta_i} - S(x)^{\mathcal{P}(1,\ldots,k)} \right| \\
&= \left| k! \sum_{i=0}^{M} \sum_{\lambda\in\{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)} \alpha_i \sum_{j=k+M+1}^{\infty} \left\langle S(x)^{(j)}, S(\beta_i \lambda \odot y)^{(j)} \right\rangle_{V^{\otimes j}} \right| && \text{(Proposition 4.0.1)} \\
&\leq k! \sum_{i=0}^{M} \sum_{\lambda\in\{0,1\}^k} |\alpha_i| \sum_{j=k+M+1}^{\infty} \left| \left\langle S(x)^{(j)}, S(\beta_i \lambda \odot y)^{(j)} \right\rangle_{V^{\otimes j}} \right| \\
&\leq k!\, 2^k \sum_{i=0}^{M} |\alpha_i| \sum_{j=k+M+1}^{\infty} \frac{\beta_i^j \, \|x\|_1^j \, k^{j/2}}{(j!\,)^2} && \text{(Corollary 1.2.2.1)} \\
&\leq k!\, 2^k (M+1) \max_{0\leq i\leq M} |\alpha_i| \sum_{j=k+M+1}^{\infty} \frac{\|x\|_1^j \, k^{j/2}}{(j!\,)^2} \\
&\leq k!\, 2^k (M+1) \max_{0\leq i\leq M} |\alpha_i| \frac{(\|x\|_1 \sqrt{k})^{k+M}}{((k+M)!\,)^2} \sum_{j=1}^{\infty} \frac{((k+M)!\,)^2}{((k+M+j)!\,)^2} \, \|x\|_1^j \, k^{j/2} \\
&\to 0. && \text{(Condition (4.0.4))}
\end{aligned}$$

□

Remark 4.0.7. *For a fixed path x, it is clearly sufficient to take $C = (\|x\|_1 \sqrt{k})^{-1}$ in Condition (4.0.4). The more general condition will prove convenient in later results (see Theorems 5.3.4 and 7.2.1).*

Example 4.0.1. *The uniform choice $\beta_i = (i+1)/(M+1)$ satisfies the above condition since*

$$\begin{aligned}\max_{0\le i\le M}|\alpha_i| &= \max_{0\le i\le M}\frac{1}{\beta_i^k}\prod_{\substack{j=0\\ j\neq i}}^{M}\frac{\beta_j}{|\beta_i-\beta_j|}\\ &= \max_{0\le i\le M}\left(\frac{M+1}{i+1}\right)^k\prod_{\substack{j=0\\ j\neq i}}^{M}\frac{j+1}{|i-j|}\\ &= \max_{0\le i\le M}\left(\frac{M+1}{i+1}\right)^{k+1}\binom{M}{i}\\ &= \mathcal{O}(M^{k+1}2^M).\end{aligned}$$

Remark 4.0.8. *Choosing $M=0$, $\beta_0 = h$ and $\alpha_0 = 1/h^k$ corresponds to the discretisation in Equation* (3.0.2)*.*

In practice, when k is large we can choose β_i such that the k^{th} power is uniform to minimize numerical errors associated with overly strong path scalings and a blow-up of the $1/\beta_i^k$ factor in α_i. That is, we choose $\beta_i = [(i+1)/(M+1)]^{1/k}$. This can be shown to satisfy Condition (4.0.4) for all k.

Remark 4.0.9. *When evaluating sums such as* (4.0.5) *numerically, the error can be reduced by subtracting 1 from every kernel, thus effectively only considering level 1 and above of the signature. It is easy to see that this change does not affect the result, but may significantly reduce the magnitude of each term of the sum and hence reduce the numerical error.*

The resulting algorithm involves $(M+1)2^k$ kernel evaluations, giving it a computational complexity of $\mathcal{O}(LkM2^k)$. Figure 4.1 shows the average error when computing $S(x)^{\mathcal{P}(1,\dots,k)}$ using the above sum of kernels for 1,000 random paths x constrained to $[0,1]^d$ for $k=2$ and 4. We report the average absolute error when compared against the exact value

$$\begin{aligned}S(x)^{\mathcal{P}(1,\dots,k)} &= \sum_{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)}\int_{0<t_1<\cdots<t_k<1} dx_{t_1}^{(i_1)}\cdots dx_{t_k}^{(i_k)}\\ &= \int_{[0,1]^k} dx_{t_1}^{(1)}\cdots dx_{t_k}^{(k)}\\ &= \prod_{i=1}^{k}\left(x_1^{(i)}-x_0^{(i)}\right).\end{aligned}$$

As reference, we note that the average magnitude of $\left|S(x)^{(1,\dots,k)}\right|$ is 1.07×10^{-1} for $k=2$ and 1.18×10^{-2} for $k=4$. We see that for $k=2$, there is no improvement in error after $M=4$, whereas for $k=4$ choosing $M=1$ is sufficient since the effect of factorial decay is stronger in the deeper levels of the signature. We note that M does not need to be very large, so the effect on complexity is minimal. Indeed, M is only particularly relevant for small values of k and can be set to 0 for k sufficiently large. As such, we do not include it in any reasoning about computational complexity from now on.

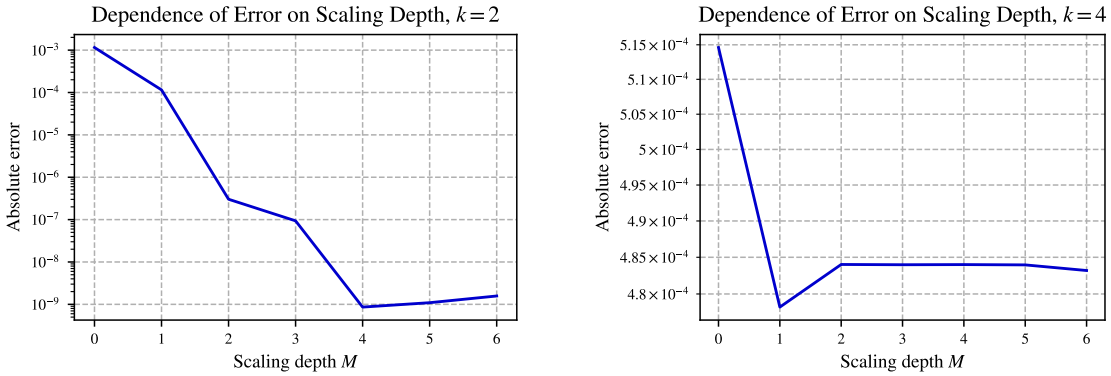


Figure 4.1: Average errors for computing $S(x)^{\mathcal{P}(1,\dots,k)}$ over 1,000 random paths x constrained to $[0,1]^d$, with path length $L = 50$ and coefficient depth $k = 2$ and 4. Dyadic order for the PDE finite difference scheme [18] is fixed at 4. Average magnitude of $S(x)^{\mathcal{P}(1,\dots,k)}$ is 1.07×10^{-1} for $k = 2$ and 1.18×10^{-2} for $k = 4$.

Chapter 5

High Order Monomial Maps

We now have a strong grasp on how to isolate $S(x)^{\mathcal{P}(1,\dots,k)}$, so we move on to address the problem of isolating coefficients within $\mathcal{P}(1,\dots,k)$. We approach this in two ways. Initially, we consider applying a monomial transformation to the path y. We will see that there exists an optimal choice of exponents for the monomial that best approximates the isolating behaviour that we are looking for. Having done this, we will note in the subsequent chapter how the monomials limit to a simple axis path, which can be substituted into the kernel instead of y. Although directly using the axis path is clearly simpler and less error-prone, the monomial approach provides an interesting framework for potentially isolating more complicated patterns of signature coefficients for which the optimal path to substitute for y may not be as obvious. See for instance the discussion in Chapter 9.

5.1 Motivating Example

We will look for a map $p : \mathbb{R}^k \to \mathbb{R}^k$ such that

$$\begin{aligned} S(p(y))^{(i_1,\dots,i_k)} &= \int_{0<t_1<\cdots<t_k<1} dp(y)_{t_1}^{(i_1)} \cdots dp(y)_{t_k}^{(i_k)} \\ &= \int_{0<t_1<\cdots<t_k<1} \dot{p}(y)_{t_1}^{(i_1)} \cdots \dot{p}(y)_{t_k}^{(i_k)} dt_1 \cdots dt_k \end{aligned}$$

is maximised over $(i_1,\dots,i_k) \in \mathcal{P}(1,\dots,k)$ by $(1,\dots,k)$. An analytically tractable choice of map whose action on path signatures has been studied in great detail is the polynomial map. In [9], it is shown that the signature coefficients of a polynomial transformation applied to a path can be expressed in terms of signature coefficients of the original path. Moreover, the corresponding map is an algebra homomorphism on the shuffle algebra $(T(V), \sqcup\!\sqcup)$, with several algebraic properties which aid in its computation. For our purposes, we will consider the simpler class of monomial maps

$$\begin{aligned} p : \mathbb{R}^k &\to \mathbb{R}^k \\ v = \sum_{i=1}^k v_i e_i &\mapsto \sum_{i=1}^k v_i^{n_i} e_i \end{aligned}$$

for $n_i \in \mathbb{N}$. We are interested in finding $n_1,\dots,n_k$ such that the product of time derivatives $\dot{p}(y)_{t_1}^{(1)} \cdots \dot{p}(y)_{t_k}^{(k)}$ is largest in the region $0 < t_1 < \cdots < t_k < 1$.

Example 5.1.1. *Figure 6.1 shows an example of the desired behaviour with $k = 3$, $n_1 = 1, n_2 = 2$ and $n_3 = 4$, whereby the integral*

$$\int_{\mathcal{D}} dt_1 d(t_2^2) d(t_3^4) = \int_{\mathcal{D}} (2t_2)(4t_3^3) dt_1 dt_2 dt_3$$

is maximised over simplexes $\mathcal{D}$ by $\{t_i : 0 < t_1 < t_2 < t_3 < 1\}$, meaning that $S(p(y))^{(1,2,3)}$ will be the largest coefficient in $\mathcal{P}(1,2,3)$. A detailed breakdown of this example into regions is shown in Figure B.1 in Appendix B.

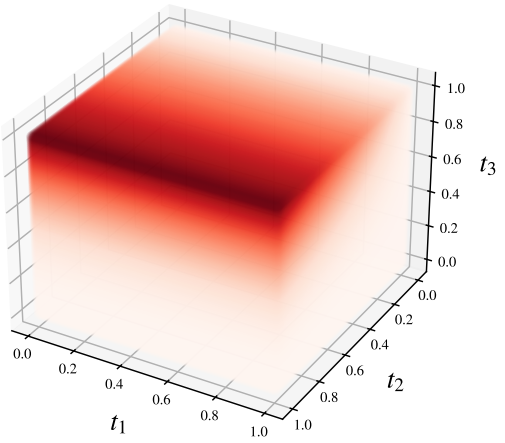


Figure 5.1: $\dot{p}(y)^{(1)}_{t_1} \dot{p}(y)^{(2)}_{t_2} \dot{p}(y)^{(3)}_{t_3}$ for $n_1 = 1, n_2 = 2, n_3 = 4$

5.2 Optimal Exponents

Before attempting to optimize over the exponents n_i, we must determine the form of the general signature coefficient $S(p(y))^{(i_1,\dots,i_k)}$.

Proposition 5.2.1. *Let $p : V \to V$ be the map such that $p(v) = \sum_{i=1}^k v_i^{n_i} e_i$ for $v = \sum_{i=1}^k v_i e_i \in V$ and $n_i \in \mathbb{N}$. Denote by $p(y) \in C_1([0,1], V)$ the path given by a point-wise application of p to y, where y is the linear path $y_t = t\mathbf{1}$. Then*

$$S(p(y))^{(i_1,\dots,i_k)}_{[0,1]} = \frac{\prod_{j=1}^k n_{i_j}}{\prod_{m=1}^k \sum_{j=1}^m n_{i_j}}. \tag{5.2.1}$$

Proof. We will prove by induction on k that

$$S(p(y))^{(i_1,\dots,i_k)}_{[0,t]} = \frac{\prod_{j=1}^k n_{i_j}}{\prod_{m=1}^k \sum_{j=1}^m n_{i_j}} t^{\sum_{j=1}^k n_{i_j}}.$$

The base case of $k = 1$ is trivial. For $k > 1$,

$$\begin{aligned}
S(p(y))^{(i_1,\dots,i_k)}_{[0,t]} &= \int_0^t S(p(y))^{(i_1,\dots,i_{k-1})}_{[0,u]} dp(y)^{(i_k)}_u \\
&= \int_0^t \left[\frac{\prod_{j=1}^{k-1} n_{i_j}}{\prod_{m=1}^{k-1} \sum_{j=1}^m n_{i_j}} u^{\sum_{j=1}^{k-1} n_{i_j}} \right] n_k u^{n_k - 1} du \\
&= \left[\frac{\prod_{j=1}^k n_{i_j}}{\prod_{m=1}^{k-1} \sum_{j=1}^m n_{i_j}} \right] \int_0^t u^{\sum_{j=1}^k n_{i_j} - 1} du \\
&= \frac{\prod_{j=1}^k n_{i_j}}{\prod_{m=1}^k \sum_{j=1}^m n_{i_j}} t^{\sum_{j=1}^k n_{i_j}},
\end{aligned}$$

which completes the induction. □

In the context of forming the filter F, we are interested in the choice of $n_1, \dots, n_k$ minimizing the quantity

$$\max_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \frac{S(p(y))^{(i_1,\dots,i_k)}}{S(p(y))^{(1,\dots,k)}} \tag{5.2.2}$$

$$= \max_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \frac{\prod_{m=1}^{k}\sum_{j=1}^{m} n_j}{\prod_{m=1}^{k}\sum_{j=1}^{m} n_{i_j}}. \tag{5.2.3}$$

We are not interested in the behaviour outside of $\mathcal{P}(1,\dots,k)$, as this is already dealt with in Section 4. Note that it is clear from Proposition 5.2.1 that we must have $n_1 < n_2 < \cdots < n_k$, so we assume this from now on.

Lemma 5.2.2. *Let $n_1 < \cdots < n_k$ and let $\tau(1,\dots,k) \subset \mathcal{P}(1,\dots,k)$ denote the set of multi-indices*

$$\tau(1,\dots,k) := \{(1,\dots,j-1,j+1,j,j+2,\dots,k) : j = 1,\dots,k-1\}$$

obtained from $(1,\dots,k)$ by a transposition of contiguous indices $j, j+1$. Then

$$\min_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \prod_{m=1}^{k}\sum_{j=1}^{m} n_{i_j} = \min_{(i_1,\dots,i_k)\in\tau(1,\dots,k)} \prod_{m=1}^{k}\sum_{j=1}^{m} n_{i_j}.$$

Proof. For $I = (i_1,\dots,i_k)$, let $R_m(I) := \sum_{j=1}^{m} n_{i_j}$. For $m = k$, $R_m(I)$ is fixed over permutations $(i_1,\dots,i_k)$ of $(1,\dots,k)$. For $m < k$, since $n_1 < \cdots < n_k$, clearly the minimum value $R_m(I)$ can take over permutations $(i_1,\dots,i_k)$ is $R_m^0 := \sum_{j=1}^{m} n_j$. Moreover, the second least value $R_m(I)$ can take is $R_m^1 := \sum_{j=1}^{m-1} n_j + n_{m+1}$. We conclude that

$$\min_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \prod_{m=1}^{k} R_m(I) \geq \min_{j=1,\dots,k-1} R_1^0\cdots R_{j-1}^0 R_j^1 R_{j+1}^0\cdots R_k^0.$$

Moreover, we have precisely that

$$\{R_1^0\cdots R_{j-1}^0 R_j^1 R_{j+1}^0\cdots R_k^0 : j = 1,\dots,k-1\} = \left\{\prod_{m=1}^{k} R_m(I) : (i_1,\dots,i_k)\in\tau(i_1,\dots,i_k)\right\}$$

since $R_1^0\cdots R_{j-1}^0 R_j^1 R_{j+1}^0\cdots R_k^0$ is attained by the product $\prod_{m=1}^{k} R_m(I)$ when $(i_1,\dots,i_k) = (1,\dots,j+1,j,\dots,k)$. It follows that

$$\begin{aligned}
\min_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \prod_{m=1}^{k} R_m(I) &\leq \min_{(i_1,\dots,i_k)\in\tau(1,\dots,k)} \prod_{m=1}^{k} R_m(I)\\
&= \min_{j=1,\dots,k-1} R_1^0\cdots R_{j-1}^0 R_j^1 R_{j+1}^1\cdots R_k^0\\
&\leq \min_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \prod_{m=1}^{k} R_m(I).
\end{aligned}$$

□

This result significantly simplifies the problem of minimizing Expression (5.2.2). Using Lemma 5.2.2, we may now reduce the problem to a set of simultaneous equations, which we later use to approximate n_i. In order to simplify the statement and proof, we will assume that $n_i \in \mathbb{R}^+$ in the following corollary.

Corollary 5.2.2.1. For a fixed N, the choice of $n_i \in \mathbb{R}^+$ minimizing

$$\max_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \frac{\prod_{m=1}^{k}\sum_{j=1}^{m} n_j}{\prod_{m=1}^{k}\sum_{j=1}^{m} n_{i_j}}$$

with $1 \leq n_1 < \cdots < n_k = N$ is such that $n_1 = 1$ and the fractions

$$f(j) := \frac{\sum_{i=1}^{j} n_i}{\sum_{i=1}^{j-1} n_i + n_{j+1}}$$

are equal for $1 \leq j < k$.

Proof. From Lemma 5.2.2, we have that

$$\begin{aligned}\max_{\substack{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots k)\\(i_1,\dots,i_k)\neq(1,\dots,k)}} \frac{\prod_{m=1}^{k}\sum_{j=1}^{m} n_j}{\prod_{m=1}^{k}\sum_{j=1}^{m} n_{i_j}} &= \max_{(i_1,\dots,i_k)\in\tau(1,\dots k)} \frac{\prod_{m=1}^{k}\sum_{j=1}^{m} n_j}{\prod_{m=1}^{k}\sum_{j=1}^{m} n_{i_j}} \\ &= \max_{j=1,\dots,k-1} \frac{\sum_{i=1}^{j} n_i}{\sum_{i=1}^{j-1} n_i + n_{j+1}} < 1\end{aligned}$$

which we wish to minimize over n_i. Since the denominator is strictly larger than the numerator, adding a constant term to both will strictly increase the fraction. We may therefore fix $n_1 = 1$. To see that the optimal solution equates all the fractions, note first that the fraction $f(j)$ is strictly decreasing in n_{j+1}, strictly increasing in $n_1, \dots, n_j$ and independent of $n_{j+2} \dots, n_k$. Suppose the fractions are not equal. We run the following algorithm: let $k_1 \in [1, k)$ denote the largest index such that

$$f(k_1) > \max_{j>k_1} f(j). \tag{5.2.4}$$

If $k_1 < k-1$, then increase n_{k_1+1} in order to decrease $f(k_1)$ and increase $f(j)$ for all $j > k_1$ until (5.2.4) is an equality. If $k_1 = k-1$, then decrease n_{k_1} in order to decrease $f(k_1)$ until $f(k_1) = \max_{j<k_1} f(j)$. In both cases, we leave $f(j)$ for $j < k_1$ unaffected and strictly reduce $\max_{j\geq k_1} f(j)$. Repeating these steps, the algorithm must terminate at the optimal n_i, and it is clear that this choice of n_i will equate $f(j)$. □

Proposition 5.2.3. *An approximate solution for $n_i \in \mathbb{N}$ is given by*

$$n_j = round\left(N^{\frac{j-1}{k-1}}\right), \quad \forall 1 \leq j \leq k. \tag{5.2.5}$$

Proof. Consider equating $f(j) = f(1)$, that is, set

$$\frac{\sum_{i=1}^{j} n_i}{\sum_{i=1}^{j-1} n_i + n_{j+1}} = \frac{1}{n_2}, \quad \forall\, 2 \leq j < k$$

recalling that $n_1 = 1$. By rearranging, the above is equivalent to the system of equations

$$\begin{aligned} n_1 &= 1, \\ n_{j+1} &= (n_2+1)n_j - n_{j-1}, \quad \forall 1 < j < k, \\ n_k &= N, \end{aligned}$$

an explicit real-valued solution to which is given by

$$n_j = 2^{-j}\left(C_1\left((n_2+1)-\sqrt{(n_2+1)^2-4}\right)^j + C_2\left((n_2+1)+\sqrt{(n_2+1)^2-4}\right)^j\right).$$

for some constants C_1, C_2. Note that when n_2 is large, $(n_2+1)-\sqrt{(n_2+1)^2-4}$ is small, and so a further approximation is to take n_j of the form $n_j = a\lambda^j$. Substituting boundary conditions and rounding gives the result. □

In the construction of F, we must also make sure that $F^{(1,\dots,k)} = 1$, and so we must scale by the coefficient in Equation (5.2.1). To ease notation, we choose to scale by the k^{th} root of this coefficient before applying the signature transform and absorb this into the map p, but we could just as well apply the scaling after taking signatures. From now on, we refer to $p_N : V \to V$ as the map

$$p_N(v) = \left(\frac{\prod_{m=1}^{k}\sum_{j=1}^{m} n_i}{\prod_{j=1}^{k} n_i}\right)^{1/k} \sum_{i=1}^{k} v_i^{n_i} e_i$$

for $v = \sum_{i=1}^{k} v_i e_i \in V$, where $n_j = \text{round}\left(N^{\frac{j-1}{k-1}}\right)$.

5.3 Error Bounds and Results

Having found the optimal choice of exponents for a fixed N, we can now show that p_N can approximate the desired behaviour up to arbitrary precision and combine this with Proposition 4.0.6 to compute the signature coefficient $S(x)^{(1,\dots,k)}$.

Remark 5.3.1. *Under the approximation given by* (5.2.5),

$$\max_{\substack{(j_1,\dots,j_k)\in\mathcal{P}(1,\dots k)\\(j_1,\dots,j_k)\neq(1,\dots,k)}} S(p_N(y))^{(j_1,\dots,j_k)} \approx \frac{1}{n_2} \approx N^{-\frac{1}{k-1}} \to 0$$

as $N \to \infty$.

More precisely, we have:

Proposition 5.3.2. *Let* N *be the* $(k-1)^{th}$ *power of a positive integer. Then*

$$\max_{\substack{(j_1,\dots,j_k)\in\mathcal{P}(1,\dots k)\\(j_1,\dots,j_k)\neq(1,\dots,k)}} S(p_N(y))^{(j_1,\dots,j_k)} \leq \frac{1}{N^{\frac{1}{k-1}}-1}$$

Proof. A direct computation of

$$\max_{1\leq j<k} \frac{\sum_{i=1}^{j} n_i}{\sum_{i=1}^{j-1} n_i + n_{j+1}}$$

using the fact that $n_i = N^{\frac{i-1}{k-1}}$ gives the result. See Appendix A, Section A.5 for details. □

Example 5.3.1. *Table 5.1 shows the resulting values of* $S(p_N(y))^{(j_1,j_2,j_3)}$ *for* $(j_1,j_2,j_3) \in \mathcal{P}(1,2,3)$ *when* $k=3$ *and* $N = 10^3, 10^6$ *and* 10^9. *We see that* p_N *gives a very close approximation of the isolating behaviour we are looking for when* N *is large.*

N	(1,2,3)	(1,3,2)	(2,1,3)	(2,3,1)	(3,1,2)	(3,2,1)
10^3	1	3.2×10^{-2}	3.2×10^{-2}	1.0×10^{-3}	3.2×10^{-5}	3.1×10^{-5}
10^6	1	1.0×10^{-3}	1.0×10^{-3}	1.0×10^{-6}	1.0×10^{-9}	1.0×10^{-9}
10^9	1	3.2×10^{-5}	3.2×10^{-5}	1.0×10^{-9}	3.2×10^{-14}	3.2×10^{-14}

Table 5.1: Coefficient isolation within $\mathcal{P}(1,2,3)$

Remark 5.3.3. *In practice, when N is large or path length L is small, $S(p_N)$ can suffer heavily from discretisation error. To mitigate this, we can make use of the time reparametrisation invariance of the signature given in Proposition 1.2.1 and, for example, discretise p_N on time points $t_i = (1 - 2^{-(i-1)})/(1 - 2^{-L})$ for $i = 1, \dots, L$, thereby concentrating the discretisation points around $t = 1$ where the time derivative of p_N is large in magnitude.*

We can now recover signature coefficients. The following is a corollary of Propositions 4.0.6 and 5.3.2.

Theorem 5.3.4. *Let β_i be chosen to satisfy Condition 4.0.4 and α_i be as in Proposition 4.0.5. Let $y \in C_1([0,1], V)$ be the linear path $y_t = t\mathbf{1}$. Then for any $x \in C_1([0,1], V)$,*

$$\sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{sgn(\lambda)} \alpha_i \mathbf{k}^{\lambda\text{-}\beta_i}_{x,\, p_N(y)} \to S(x)^{(1,\dots,k)}$$

as $N, M \to \infty$, where $sgn(\lambda) = \sum_i \mathbf{1}\{\lambda_i = 0\}$.

Proof. See Appendix A, Section A.6. □

Chapter 6

Limiting Axis Paths

As we saw in the previous chapter, monomial transforms are a malleable and analytically tractable tool for reproducing patterns within the signature such as order isolation. As it happens, the monomial paths we have considered tend to a simple axis path as $N \to \infty$. We will derive this limit and show that we can consider the limiting path directly, removing the need for monomials and, crucially, the approximating parameter N.

6.1 Order Isolation with Axis Paths

Proposition 6.1.1. *The limit of $p_N(y)$ as $N \to \infty$ is, up to reparametrisation of time, the axis path given by*

$$z_t = (e_1 * e_2 * \cdots * e_k)_t$$

for $t \in [0,1]$, where $(e_i)_t$ is understood to be the linear path from $\mathbf{0}$ to e_i.

Proof. Set $m = N^{\frac{1}{k-1}}$. Let $t_0 = 0$, $t_k = 1$ and

$$t_i = m^{\left(m^{i-1} - m^i\right)^{-1}} \in (0,1)$$

for $i = 1, \ldots, k-1$. Then we have

$$t_i^{N^{\frac{j-1}{k-1}}} = m^{\frac{m^j - i}{1-m}} \to \mathbf{1}\{i \geq j\}$$

for all $i, j = 1, \ldots, k$ as $N \to \infty$. It follows from the definition of p_N that

$$p_N(y)_{t_i}^{(j)} \to \mathbf{1}\{i \geq j\},$$

and so $p_N(y)_{t_i}$ tends to the i^{th} vertex of the axis path z as $N \to \infty$. The result follows from the component-wise monotonicity of $p_N(y)$. □

Alternatively, it is easy to show directly that the path z_t exhibits the desired behaviour by noting that

$$S(z)^{(i_1,\ldots,i_k)} = [\exp(e_1) \otimes \cdots \otimes \exp(e_k)]^{(i_1,\ldots,i_k)} = \mathbf{1}\{(i_1, \ldots, i_k) = (1, \ldots, k)\}$$

for all $(i_1, \ldots, i_k) \in \mathcal{P}(1, \ldots, k)$, by Proposition 1.2.6 and Chen's relation. As a direct corollary of Theorem 5.3.4 and Proposition 6.1.1, we have the following.

Theorem 6.1.2. *Let β_i be chosen to satisfy Condition 4.0.4 and α_i be as in Proposition 4.0.5. Let z denote the axis path of Proposition 6.1.1. Then for any $x \in C_1([0,1], V)$,*

$$k! \sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{sgn(\lambda)} \alpha_i \, \mathbf{k}_{x,z}^{\lambda\text{-}\beta_i} \to S(x)^{(1,\ldots,k)}$$

as $M \to \infty$, where $sgn(\lambda) = \sum_i \mathbf{1}\{\lambda_i = 0\}$.

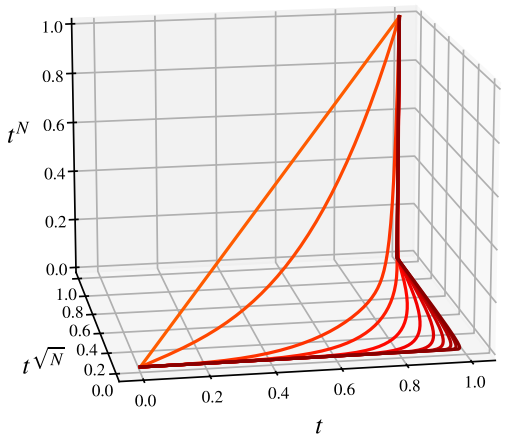


Figure 6.1: Convergence of $p_N(y)$ for $k = 3$

Proof. The proof follows from that of Theorem 5.3.4 but with z in place of $p_N(y)$. □

Remark 6.1.3. *Returning to the original formulation of our approach, the filter F with which we effectively take inner products can now be written as*

$$F = k! \sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{sgn(\lambda)} \alpha_i \, S(\beta_i \lambda \odot z) \in span(\mathcal{S}).$$

Theorem 6.1.2 then states that $\langle S(x), F \rangle \to S(x)^{(1,\dots,k)}$ as $M \to \infty$.

6.2 The Kernel PDE with Axis Paths

Recall the signature kernel PDE (1.3.2) is given by

$$\frac{\partial^2 \mathbf{k}_{x,y}}{\partial s \partial t} = \langle \dot{x}_s, \dot{y}_t \rangle_V \, \mathbf{k}_{x,y}, \qquad \mathbf{k}_{x,y}(0,\cdot) = \mathbf{k}_{x,y}(\cdot,0) = 1$$

for general $x, y \in C_1([0,1], V)$. In [18, Section 3.1], the proposed finite difference scheme for solving the above Goursat PDE takes the form

$$\begin{aligned} \widehat{\mathbf{k}}(s_{i+1}, t_{j+1}) = \widehat{\mathbf{k}}(s_{i+1}, t_j) + \widehat{\mathbf{k}}(s_i, t_{j+1}) - \widehat{\mathbf{k}}(s_i, t_j) \\ + \frac{1}{2} \left\langle x_{s_{i+1}} - x_{s_i}, y_{t_{j+1}} - y_{t_j} \right\rangle \left(\widehat{\mathbf{k}}(s_{i+1}, t_j) + \widehat{\mathbf{k}}(s_i, t_{j+1}) \right) \end{aligned} \tag{6.2.1}$$

over the dyadically refined grid $P_\lambda = \{(s_i, t_j)\}_{0 \le i \le 2^\lambda L_x,\, 0 \le j \le 2^\lambda L_y}$ of order λ, where L_x and L_y are the lengths of the discrete data streams x and y respectively. The complexity of the above finite difference scheme can be reduced to $\mathcal{O}(Lk)$ after a suitable parallelisation of the computation, where $L = \max\{L_x, L_y\}$. The dependence on k arises from computing the inner product $\left\langle x_{s_{i+1}} - x_{s_i}, y_{t_{j+1}} - y_{t_j} \right\rangle$. We note, however, that when y is taken to be the axis path z of Proposition 6.1.1 parametrised uniformly such that $\dot{z}_t = k e_m$ for $t \in \left(\frac{m-1}{k}, \frac{m}{k}\right)$, we have

$$\left\langle x_{s_{i+1}} - x_{s_i}, z_{t_{j+1}} - z_{t_j} \right\rangle = k \left(x^{(m)}_{s_{i+1}} - x^{(m)}_{s_i} \right)$$

for $(t_j, t_{j+1}) \subset \left(\frac{m-1}{k}, \frac{m}{k}\right)$, meaning that the complexity of the finite difference scheme reduces to $\mathcal{O}(L)$. Note that each term of the sum of Theorem 6.1.2 is a signature kernel

of x with an axis path, weighted by $k!\,(-1)^{\mathrm{sgn}(\lambda)}\alpha_i$. Moreover, the terms can be computed independently of each other. Thus, we may parallelise the computation of the terms to reduce the complexity to $\mathcal{O}(L)$.

Remark 6.2.1. *In practice, instead of computing each kernel completely independently, consider the following. Let* $\lambda^+ = (\lambda_1, \dots, \lambda_{k-1}, 1) \in \{0,1\}^k$ *and* $\lambda^- = (\lambda_1, \dots, \lambda_{k-1}, 0) \in \{0,1\}^k$. *For* $i = 0, \dots, k$, *let* $t_i \in [0,1]$ *denote the time at which* z *attains its* i^{th} *vertex. Then*

$$\mathbf{k}_{x,\lambda^- \odot z}(s,t) = \begin{cases} \mathbf{k}_{x,\lambda^+ \odot z}(s,t), & t \in [0, t_{k-1}], \\ \mathbf{k}_{x,\lambda^+ \odot z}(s,t_{k-1}), & t \in (t_{k-1}, 1] \end{cases}$$

for all $s \in [0,1]$. *Therefore, having computed* $\mathbf{k}_{x,\lambda^+ \odot z}$, *we immediately recover* $\mathbf{k}_{x,\lambda^- \odot z}$. *This observation halves the number of kernels which we need to compute.*

Chapter 7

Generalisations

7.1 Sums of Signature Coefficients

The tools we have developed to isolate a single signature coefficient can be applied to isolate patterns of coefficients within the signature. The following two remarks summarise the fundamental two patterns which, when overlayed, produce coefficient isolation.

Remark 7.1.1. *Let z denote the axis path given in the previous section. Then the kernel*

$$\mathbf{k}_{x,z} = \sum_{\substack{J=(j_1,\dots,j_m) \\ j_1 \leq \cdots \leq j_m}} \frac{1}{\#_1(J)! \cdots \#_k(J)!} S(x)^J,$$

where $\#_i(J)$ is the number of times index i appears in J, isolates the coefficients of the signature which are given by an ordered multi-index. The set of coefficients appearing in this sum can be viewed as a path transform which is variant to path channel permutations.

Remark 7.1.2. *For any $x, y \in C_1(V)$, we may define a permutation class kernel*

$$\begin{aligned} \mathbf{k}^{\mathcal{P}}_{x,y} &:= \sum_{I \in \mathcal{P}(1,\dots,k)} S^I(x) S^I(y) \\ &= \frac{\partial^k}{\partial \lambda_1 \cdots \partial \lambda_k} \mathbf{k}^{\boldsymbol{\lambda}\text{-}\mathbf{1}}_{x,y}, \end{aligned}$$

computable as the sum of truncated kernels

$$k! \sum_{\lambda \in \{0,1\}^k} (-1)^{sgn(\lambda)} \mathbf{k}^{\boldsymbol{\lambda}\text{-}\mathbf{1},k}_{x,y},$$

or approximated by

$$k! \sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{sgn(\lambda)} \alpha_i \mathbf{k}^{\boldsymbol{\lambda}\text{-}\boldsymbol{\beta}_i}_{x,y}$$

for M sufficiently large.

7.2 Block-Ordered Coefficients

The axis path z may be generalized further to compute sums of signature coefficients with "blocks" of unordered indices. Suppose, for example, we wish to compute

$$S(x)^{(1,2,3)} + S(x)^{(2,1,3)},$$

that is, the sum of coefficients given by a multi-index where 1 and 2 appear before 3, but the order of 1 and 2 is not important. Instead of considering the path $z = e_1 * e_2 * e_3$, we consider $z = (e_1 + e_2) * e_3$ to remove the order constraint on 1 and 2. It is easy to see by Chen's relation that

$$\begin{aligned} S(z)^{(1,2,3)} = S(z)^{(2,1,3)} &= 1/2, \\ S(z)^{(1,3,2)} = S(z)^{(2,3,1)} = S(z)^{(3,1,2)} = S(z)^{(3,2,1)} &= 0, \end{aligned}$$

giving the desired isolation pattern on $\mathcal{P}(1,2,3)$. To generalise this idea, we introduce the *concatenation* of multi-indices. For two multi-indices $I = (i_1, \dots, i_{m_1})$ and $J = (j_1, \dots, j_{m_2})$, denote the concatenation of I and J by $I * J = (i_1, \dots, i_{m_1}, j_1, \dots, j_{m_2})$. For two sets of multi-indices $\mathcal{I}$ and $\mathcal{J}$, denote by $\mathcal{I} * \mathcal{J}$ the set

$$\mathcal{I} * \mathcal{J} = \{I * J : I \in \mathcal{I}, J \in \mathcal{J}\}.$$

The following result follows immediately from Chen's relation and Proposition 4.0.6:

Theorem 7.2.1. *Let β_i be chosen to satisfy Condition 4.0.4 and let α_i be as in Proposition 4.0.5. Suppose $I_1, \dots, I_m$ are multi-indices of lengths $l_1, \dots, l_m$ respectively, such that $I_1 * \cdots * I_m = (1, \dots, k)$. Let $j_i = \sum_{p=1}^{i} l_p$ for $i = 1, \dots, m$ and denote by z the path*

$$z_t = ((e_1 + \cdots + e_{j_1}) * (e_{j_1+1} + \cdots + e_{j_2}) * \cdots * (e_{j_{m-1}+1} + \cdots + e_k))_t$$

for $t \in [0,1]$, where $(e_i)_t$ is understood to be the linear path from $\mathbf{0}$ to e_i. Then

$$\sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{sgn(\lambda)} \alpha_i \, \mathbf{k}_{x,z}^{\lambda\text{-}\beta_i} \to \frac{1}{l_1! \, l_2! \cdots l_m!} \, S(x)^{\mathcal{P}(I_1) * \mathcal{P}(I_2) * \cdots * \mathcal{P}(I_m)}$$

as $M \to \infty$, for any $x \in C_1([0,1], V)$.

Proof. See Appendix A, Section A.7. □

Remark 7.2.2. *The single coefficient case corresponds to $I_j = (j)$, since then $\mathcal{P}(I_1) * \mathcal{P}(I_2) * \cdots * \mathcal{P}(I_m) = (1, 2, \dots k)$. Similarly, isolation of $S(x)^{\mathcal{P}(1,\dots,k)}$ corresponds to $I_1 = (1, \dots, k)$.*

Remark 7.2.3. *Recall that, as is the case throughout, the assumption that $I_1 * I_2 * \cdots * I_m = (1, \dots, k)$ is purely for simplicity and the result can easily be shown to extend to a general multi-index $I_1 * I_2 * \cdots * I_m = (i_1, \dots, i_k)$.*

Theorem 7.2.1 has a simple interpretation. Let

$$\begin{aligned} \tilde{x}_t^i &:= \int_{[0,t]^{l_i}} \prod_{j \in I_i} dx_{u_j}^{(j)} \\ &= \prod_{j \in I_i} \left(x_t^{(j)} - x_0^{(j)} \right) \end{aligned}$$

and define $\tilde{x} = (\tilde{x}^1, \dots, \tilde{x}^m)$, where $\tilde{x}$ can either be viewed as a path of l_i-dimensional integrals with respect to the channels of x given by indices in I_i, or equivalently as a product of channels as above. Then we may write

$$S(x)^{\mathcal{P}(I_1) * \mathcal{P}(I_2) * \cdots * \mathcal{P}(I_m)} = S(\tilde{x})^{(1,\dots,m)}.$$

Chapter 8

Numerical Results

We present some numerical results to showcase the accuracy of our method. Throughout, we will compute signature coefficients on a random sample of paths and report the absolute error, as well as the average absolute magnitude of the coefficients. It should be noted that average percentage error is not a suitable metric in the case of signature coefficients, which may naturally be very close or equal to 0. Instead, for a measure of error relative to coefficient magnitude, we consider a "scaled error", which we define as the mean absolute error divided by the average magnitude of a coefficient.

Definition 8.0.1. *In the results that follow, for a random sample of paths* $\{x_i\}_{i=1,\dots,m}$*, we define*

$$\textit{Scaled Error} \coloneqq \frac{\frac{1}{m}\sum_{i=1}^{m}|\hat{S}(x_i)^{(1,\dots,k)} - S(x_i)^{(1,\dots,k)}|}{\frac{1}{m}\sum_{i=1}^{m}|S(x_i)^{(1,\dots,k)}|},$$

to be used as a suitable substitute for percentage error, where $\hat{S}(x_i)^{(1,\dots,k)}$ *denotes the value obtained using Theorem 5.3.4 or Theorem 6.1.2.*

Figure 8.1 shows the average errors when computing $S(x)^{(1,\dots,k)}$ using the monomial approximation p_N and their dependence on coefficient depth k, scaling depth M and monomial order N. The average is taken over 1,000 random paths of length $L = 150$ constrained to $[0,1]^d$, where the true value of the signature is computed using the iisignature package [17]. Similarly to what we noted in Chapter 4, for $k = 2$ we benefit from increasing M all the way to $M = 6$, whereas for $k = 5$ it suffices to take $M = 2$ because of the strong effects of factorial decay. From Proposition 5.3.2, we expect the error to decay exponentially with monomial order N. Indeed, we see that this is the case up to about $N \approx 10^8$, after which the plateau in the error is likely caused either by discretisation error in $p_N(y)$ or by other sources of error unrelated to p_N, such as the scaling depth M or the dyadic order of the finite difference scheme (6.2.1). We note that when using the monomial approximation p_N, the algorithm performs poorly for larger coefficient depths k, since the chosen monomial order of 10^{10} is no longer sufficient to produce adequate isolating behaviour at these levels.

Figure 8.2 shows the dependence of error on coefficient depth k using the axis path z, where $M = 2$. Here we observe much lower errors for higher depths k than with the monomial approximation p_N, since the isolating behaviour produced by the axis path is exact. Even for the choice of $M = 2$, we attain remarkably low error relative to the absolute magnitude of coefficients. It should be noted at this point that the error is mainly due to the dyadic order of the PDE scheme (6.2.1). When we increase this in Figure 8.2, we observe even lower errors.

As we discussed in Section 1.4, the complexity of computing the signature of a d dimensional path of length L up to level k is $\mathcal{O}(Ld^k)$. In our case, since we assume $d = k$ this becomes

$\mathcal{O}(Lk^k)$. In light of this, computing coefficients for a large sample of paths using iisignature becomes infeasible for high levels k. To test our algorithm on levels beyond $k = 7$, we test the error on a sample of random linear paths starting at $\mathbf{0}$ and ending at a random point in $[0.5, 1]^k$, whose signature is computable easily using Proposition 1.2.6. The restriction on the endpoint ensures that deep coefficients are not too small. In light of our observations about the scaling depth M, we let it decay as k increases. Specifically, we take

$$M = \begin{cases} 2, & k \leq 6, \\ 1, & 7 \leq k \leq 10, \\ 0, & k \geq 11. \end{cases}$$

For all values of k, we fix a dyadic order of 6 for the PDE scheme. Figure 8.3 shows the resulting absolute error, the absolute magnitude of coefficients and the scaled error as defined above. We see that at deeper levels, the algorithm recovers coefficients up to a scaled error of roughly 0.05.

Given that signature terms decay factorially as per Lemma 1.2.2, we may naturally question why Figure 8.3 shows the scaled error increasing with coefficient depth. After all, the only source of error in Theorem 6.1.2 is that from levels of the signature beyond the $(k + M)^{th}$, and so we would expect this error to decay away factorially with depth k. In practice, however, the discretisation error arising from the numerical scheme (6.2.1) becomes increasingly significant as the magnitude of the target coefficient decays. This becomes the main driver for the error seen in Figure 8.3. As discussed with Figure 8.2, we can reduce this error by increasing the dyadic order of the scheme.

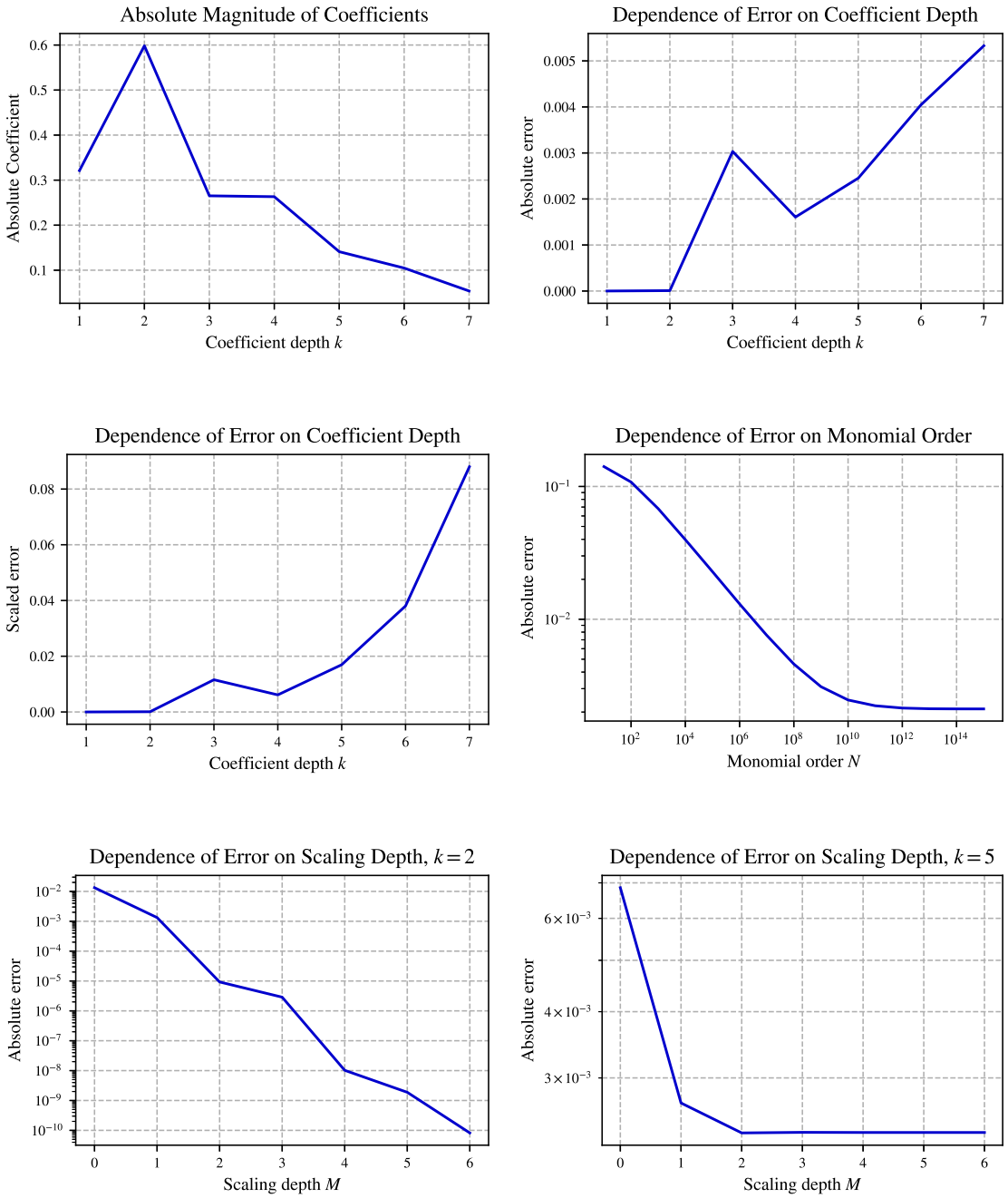


Figure 8.1: Average errors for computing $S(x)^{(1,\dots,k)}$ over 1,000 random paths constrained to $[0,1]^d$ using $p_N(y)$. Unless stated otherwise, we take path length $L = 150$, coefficient depth $k = 5$, monomial order $N = 10^{10}$ and scaling depth $M = 2$. The dyadic order for the kernel PDE solver is fixed at 2.

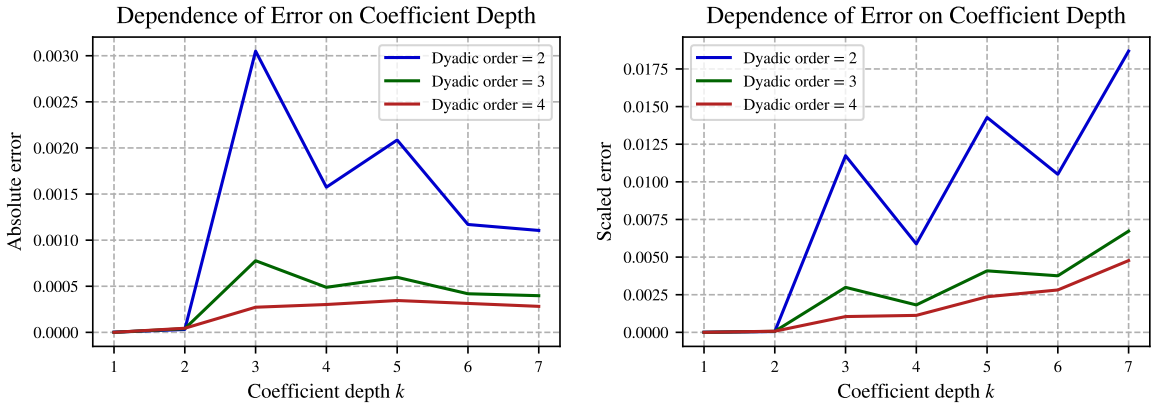


Figure 8.2: Average errors for computing $S(x)^{(1,\dots,k)}$ over 1,000 random paths constrained to $[0,1]^d$ using the axis path z, with path length $L = 150$ and scaling depth $M = 2$. Dyadic order for the kernel PDE solver is set to 2 (blue), 3 (green) and 4 (red).

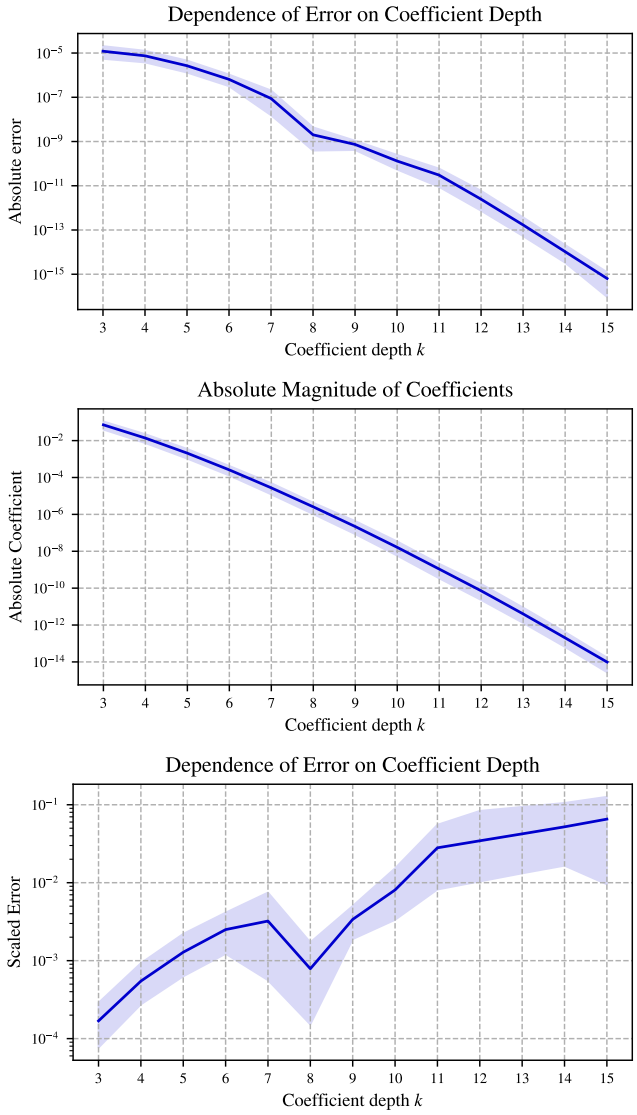


Figure 8.3: Average errors for computing $S(x)^{(1,\dots,k)}$ over 100 random linear paths starting at $\mathbf{0}$ with end points in $[0.5,1]^k$, with decaying scaling depth and a dyadic order of 6. Shaded area shows the region between the 10% and 90% quantiles.

Chapter 9

Conclusion and Future Work

We have developed a framework for efficiently computing signature coefficients through the signature kernel. In Chapter 2, we motivated our approach to permutation class isolation with existing results concerning randomised signature kernels and integral transforms with respect to unsigned and signed measures. In Chapter 3 we argued that, instead of integral transforms, we can recover $S(x)^{\mathcal{P}(1,\dots,k)}$ by considering a suitable derivative of a kernel with respect to a component-wise scaling of the underlying path. By considering a finite difference approximation, we recovered the permutation class as a sum of signature kernels. In Chapter 4, we proposed a different scheme, based on the finite difference, which avoids the numerical instabilities associated with approximating a high-order derivative. Having found a stable algorithm for permutation class recovery, we moved on in Chapters 5 and 6 to the problem of order isolation via monomial approximations and axis paths. We argued that, when using the axis paths, the algorithm may be parallelised down to a complexity of $\mathcal{O}(L)$, which boasts an improvement over the naive methods discussed in Sections 1.4 and 1.6. Finally, we considered some generalisations of the approach in Chapter 7 and presented numerical results supporting the efficacy of our method in Chapter 8.

There are potential ways in which our algorithm may be improved, but which we did not explore here due to time constraints. For example, as we discussed in Chapter 8, the error of our algorithm is largely due to the discretisation of the signature kernel PDE. Whilst for simplicity we have used the same numerical scheme as in [18], it is likely that we can choose a simpler scheme when one of the paths in the signature kernel is an axis path, as in Theorem 6.1.2. If this is the case, we stand to significantly reduce the associated discretisation error. A study of suitable numerical schemes is left for future work.

Another interesting question for future research is whether our approach can be extended to cover, for instance, recovery of the product of two signature terms, which can be written using the Shuffle Identity (Proposition 1.2.8) as

$$S(x)^I \cdot S(x)^J = S(x)^{I \sqcup\!\sqcup J},$$

where $S(x)^{I \sqcup\!\sqcup J} := (e_I^* \sqcup\!\sqcup e_J^*)(S(x))$ is a sum of signature coefficients. In Chapter 7, we considered "block-ordered" coefficients, where we isolate the sum of coefficients whose multi-indices are formed with "blocks" of unordered coefficients, with the blocks themselves being ordered such that all indices in one block precede all indices in the next block. The shuffle product is the opposite of this, where coefficients within each block are ordered, but the blocks themselves are not and can shuffle into each other. In this case, the optimal path to take signature kernels with is less obvious, but we may try to leverage the monomial framework we have developed in Chapter 5 to derive an optimal solution.

In Section 1.5, we discussed potential applications to machine learning and transformer models on signature space. Another path transform which has seen use cases in machine

learning is the *"log-signature"*, defined as the tensor logarithm of the signature. The log-signature offers a more compact representation of the signature, taming the curse of dimensionality from which the standard signature suffers. If we are successful in computing the shuffle product and subsequently products of coefficients, then we may reasonably try to recover log-signature coefficients in an efficient manner, since these can be expressed as a sum of products of standard signature coefficients.

Appendix A Technical Proofs

A.1 Proof of Proposition 2.2.2

Lemma A.1.1 ([10]). *Let ψ_σ denote the centered Gaussian density with standard deviation σ. Then $\psi_\sigma^{(n)}$ admits the representation:*

$$\psi_\sigma^{(n)}(x) = \left[\sum_{m=0}^{\lfloor n/2 \rfloor} C_\sigma(n,m) x^{n-2m}\right] \psi_\sigma(x),$$

where

$$C_\sigma(n,m) = \binom{n}{2m} 2^m \frac{\Gamma\left(\frac{2m+1}{2}\right)}{\Gamma\left(\frac{1}{2}\right)} \left(\frac{1}{\sigma}\right)^{2(n-m)} (-1)^{n+m}.$$

Proof. A simple induction gives the result. □

Lemma A.1.2. *For $n \geq 1$, ψ_σ restricted to $[0,1]$ has n^{th} moment M_n where*

$$M_n \leq \frac{1}{\sigma\sqrt{2\pi}} \left[\frac{1}{n+1} - \frac{1}{2\sigma^2(n+3)} + \frac{1}{8\sigma^4(n+5)}\right].$$

Proof. The proof follows simply from applying the bound $e^{-\frac{1}{2}(x/\sigma)^2} \leq 1 - \frac{1}{2}\left(\frac{x}{\sigma}\right)^2 + \frac{1}{8}\left(\frac{x}{\sigma}\right)^4$. □

Proof of Proposition 2.2.2: Condition (1) clearly holds. Moreover, from the above two lemmas

$$\begin{aligned}
\int_{\mathcal{D}} |\pi^m \mu_{k,\sigma}(d\pi)| &= \frac{1}{k!} \int_{-1}^{1} |\pi^m \psi_\sigma^{(k)}(\pi) \Lambda(d\pi)| \\
&= \frac{1}{k!} \int_{-1}^{1} \left| \left[\sum_{i=0}^{\lfloor k/2 \rfloor} C_\sigma(k,i) \pi^{m+k-2i}\right] \psi_\sigma(\pi) \right| \Lambda(d\pi) \qquad \text{(Lemma A.1.1)} \\
&\leq \frac{2}{k!} \int_0^1 \sum_{i=0}^{\lfloor k/2 \rfloor} |C_\sigma(k,i)|\, \pi^{m+k-2i} \psi_\sigma(\pi) \Lambda(d\pi) \\
&= \frac{2}{k!} \sum_{i=0}^{\lfloor k/2 \rfloor} |C_\sigma(k,i)|\, M_{m+k-2i} \\
&= \mathcal{O}\left(\frac{1}{m}\right). \qquad \text{(Lemma A.1.2)}
\end{aligned}$$

By Corollary 1.2.2.1, we also have that

$$|a_m(s,t)| \leq \frac{\|x\|^m_{1,[a,s]} \|y\|^m_{1,[a,t]}}{(m!)^2}.$$

Thus, there exists $C > 0$ such that

$$\sum_{m \geq 0} |a_m(s,t)| \int_{-1}^{1} |\pi^m \mu(d\pi)| \;\leq\; C \sum_{m \geq 0} \frac{\|x\|^m_{1,[a,s]} \|y\|^m_{1,[a,t]}}{m(m!)^2} < \infty,$$

and so condition (2) holds for any choice of $x, y \in C_1(V)$. Moreover, since $\psi_\sigma^{(k)}$ is a nascent delta function, it follows that

$$\lim_{\sigma\to 0}\int_{\mathcal{D}} \pi^m \mu_{k,\sigma}(d\pi) = \lim_{\sigma\to 0}\frac{(-1)^k}{k!}\int_{-1}^{1}\pi^m \psi_\sigma^{(k)}(\pi)d\pi = \delta_{m,k},$$

and this convergence can easily be shown to be uniform across m by applying Lemma A.1.2. □

A.2 Proof of Corollary 2.2.1.1

Proof. Since we assume V is equipped with an inner product with respect to which the basis e_i is orthonormal, we must have that $\|v\|_V = \sqrt{\sum_{i=0}^k \left(v^{(i)}\right)^2}$ for all $v = \sum_{i=0}^k v^{(i)} e_i \in V$. Thus we have that

$$\begin{aligned}
\|\pi \odot y\|_1 &= \sup_{\mathcal{D}\in\mathcal{D}[0,1]} \sum_{t_i\in\mathcal{D}} \left\|(\pi\odot y)_{t_{i+1}} - (\pi\odot y)_{t_i}\right\|_V \\
&= \sup_{\mathcal{D}\in\mathcal{D}[0,1]} \sum_{t_i\in\mathcal{D}} \sqrt{\sum_{j=1}^k \pi_j^2 \left(y_{t_{i+1}}^{(j)} - y_{t_i}^{(j)}\right)^2} \\
&\leq \sup_{\mathcal{D}\in\mathcal{D}[0,1]} \sum_{t_i\in\mathcal{D}} \sqrt{\sum_{j=1}^k \left(y_{t_{i+1}}^{(j)} - y_{t_i}^{(j)}\right)^2} \quad = \|y\|_1
\end{aligned}$$

for all $\pi \in [-1,1]$, where $\mathcal{D}[0,1]$ is the set of partitions of $[0,1]$. Note that,

$$\begin{aligned}
&\sum_{m=0}^{\infty}\int_{[-1,1]^k} \left|\left\langle S(x)^{(m)}, S(\pi\odot y)^{(m)}\right\rangle_{V^{\otimes m}}\right| \mu_\sigma(d\pi) \\
&\leq \sum_{m=0}^{\infty}\int_{[-1,1]^k} \frac{\|x\|_1^m \|\pi\odot y\|_1^m}{(m!)^2}\mu_\sigma(d\pi) \qquad \text{(Corollary 1.2.2.1)} \\
&\leq \sum_{m=0}^{\infty}\int_{[-1,1]^k} \frac{\|x\|_1^m \|y\|_1^m}{(m!)^2}\mu_\sigma(d\pi) \\
&= \mu_\sigma\left([-1,1]^k\right)\sum_{m=0}^{\infty}\frac{\|x\|_1^m\|y\|_1^m}{(m!)^2} \quad < \infty.
\end{aligned}$$

Thus by Fubini's Theorem, we have

$$\begin{aligned}
&\left|k!\int_{[-1,1]^k} \mathbf{k}_{x,\pi\odot y}\mu_\sigma(d\pi) - S(x)^{\mathcal{P}(1,\ldots,k)}\right| \\
&= \left|k!\int_{[-1,1]^k}\sum_{m=0}^{\infty}\left\langle S(x)^{(m)}, S(\pi\odot y)^{(m)}\right\rangle_{V^{\otimes m}}\mu_\sigma(d\pi) - S(x)^{\mathcal{P}(1,\ldots,k)}\right| \\
&= \left|k!\sum_{m=0}^{\infty}\int_{[-1,1]^k}\left\langle S(x)^{(m)}, S(\pi\odot y)^{(m)}\right\rangle_{V^{\otimes m}}\mu_\sigma(d\pi) - S(x)^{\mathcal{P}(1,\ldots,k)}\right|,
\end{aligned}$$

where

$$
\begin{aligned}
&\int_{[-1,1]^k} \left\langle S(x)^{(m)}, S(\pi \odot y)^{(m)} \right\rangle_{V^{\otimes m}} \mu_\sigma(d\pi) \\
&= \int_{[-1,1]^k} \sum_{(i_1,\ldots,i_m)\in\{1,\ldots,k\}^m} S(x)^{(i_1,\ldots,i_m)} S(\pi \odot y)^{(i_1,\ldots,i_m)} \mu_\sigma(d\pi) \\
&= \frac{1}{m!} \sum_{(i_1,\ldots,i_m)\in\{1,\ldots,k\}^m} S(x)^{(i_1,\ldots,i_m)} \int_{[-1,1]^k} \pi_{i_1} \cdots \pi_{i_m} \mu_\sigma(d\pi)
\end{aligned}
$$

for $m \geq 1$, and 0 for $m = 0$. Let $\phi_{1,\sigma}(i_1, \ldots, i_m) := \int_{[-1,1]^k} \pi_{i_1} \cdots \pi_{i_m} \mu_\sigma(d\pi)$. Then as a direct consequence of Proposition 2.2.2, there exists σ such that

$$
\left| \phi_{1,\sigma}(i_1, \ldots, i_m) - \mathbf{1}\left\{ (i_1, \ldots, i_m) \in S(x)^{\mathcal{P}(1,\ldots,k)} \right\} \right| < C\varepsilon,
$$

$$
C := \left(k! \sum_{m=1}^{\infty} \frac{\|x\|_1^m}{(m!)^2} \right)^{-1},
$$

for which

$$
\begin{aligned}
&\left| k! \sum_{m=0}^{\infty} \int_{[-1,1]^k} \left\langle S(x)^{(m)}, S(\pi \odot y)^{(m)} \right\rangle_{V^{\otimes m}} \mu_\sigma(d\pi) - S(x)^{\mathcal{P}(1,\ldots,k)} \right| \\
&= \left| k! \sum_{m=1}^{\infty} \frac{1}{m!} \sum_{(i_1,\ldots,i_m)\in\{1,\ldots,k\}^m} S(x)^{(i_1,\ldots,i_m)} \phi_{1,\sigma}(i_1, \ldots, i_m) \ - S(x)^{\mathcal{P}(1,\ldots,k)} \right| \\
&= \left| k! \sum_{m=1}^{\infty} \frac{1}{m!} \sum_{(i_1,\ldots,i_m)\in\{1,\ldots,k\}^m} S(x)^{(i_1,\ldots,i_m)} \left(\phi_{1,\sigma}(i_1, \ldots, i_m) \ - \mathbf{1}\left\{ (i_1, \ldots, i_m) \in \mathcal{P}(1, \ldots, k) \right\} \right) \right| \\
&\leq k! \sum_{m=1}^{\infty} \frac{1}{m!} \sum_{(i_1,\ldots,i_m)\in\{1,\ldots,k\}^m} \left| S(x)^{(i_1,\ldots,i_m)} \right| \left| \phi_{1,\sigma}(i_1, \ldots, i_m) \ - \mathbf{1}\left\{ (i_1, \ldots, i_m) \in \mathcal{P}(1, \ldots, k) \right\} \right| \\
&\leq k! \sum_{m=1}^{\infty} \frac{1}{m!} \sum_{(i_1,\ldots,i_m)\in\{1,\ldots,k\}^m} \left| S(x)^{(i_1,\ldots,i_m)} \right| C\varepsilon \\
&\leq k! \sum_{m=1}^{\infty} \frac{1}{m!} \left\| S(x)^{(m)} \right\| C\varepsilon \\
&\leq k! \sum_{m=1}^{\infty} \frac{\|x\|_1^m}{(m!)^2} C\varepsilon \\
&= \varepsilon.
\end{aligned}
$$

Thus,

$$
\left| k! \int_{[-1,1]^k} \mathbf{k}_{x,\pi\odot y} \mu_\sigma(d\pi) - S(x)^{\mathcal{P}(1,\ldots,k)} \right| \leq \varepsilon.
$$

□

A.3 Proof of Proposition 3.0.1

We will first prove a lemma detailing how the derivative acts on levels of the signature, before moving on to the proof of the theorem itself.

Lemma A.3.1.

$$\left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} S(\lambda\odot z)^{(m)}_{[0,1]} = \begin{cases} \sum_{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)} S(z)^{(i_1,\dots,i_k)}_{[0,1]} e_{i_1}\cdots e_{i_k}, & m=k \\ \mathbf{0}, & m\neq k. \end{cases}$$

Proof. For $m=k$,

$$\begin{aligned}
\left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} S(\lambda\odot z)^{(m)}_{[0,1]} &= \left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} \int_{0<t_1<\cdots<t_k<1} d(\lambda\odot z)_{t_1}\otimes\cdots\otimes d(\lambda\odot z)_{t_k} \\
&= \left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} \int_{0<t_1<\cdots<t_k<1} (\lambda\odot dz_{t_1})\otimes\cdots\otimes(\lambda\odot dz_{t_k}) \\
&= \sum_{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)} \int_{0<t_1<\cdots<t_k<1} (e_{i_1}\odot dz_{t_1})\otimes\cdots\otimes(e_{i_k}\odot dz_{t_k}) \\
&= \sum_{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)} S(z)^{(i_1,\dots,i_k)}_{[0,1]} e_{i_1}\cdots e_{i_k}.
\end{aligned}$$

By a similar calculation, it is easy to see that the derivative is $\mathbf{0}\in V^{\otimes m}$ if $m\neq k$. □

Proof of Proposition 3.0.1:

$$\begin{aligned}
\left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} \mathbf{k}_{x,\lambda\odot y} &= \left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} \left\langle S(x)_{[0,1]},\ S(\lambda\odot y)_{[0,1]}\right\rangle \\
&= \left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} \sum_{i=0}^{\infty} \left\langle S(x)^{(i)}_{[0,1]},\ S(\lambda\odot y)^{(i)}_{[0,1]}\right\rangle_{V^{\otimes i}} \\
&= \sum_{i=0}^{\infty} \left\langle S(x)^{(i)}_{[0,1]},\ \left.\frac{\partial^k}{\partial\lambda_1\cdots\partial\lambda_k}\right|_{\lambda=0} S(\lambda\odot y)^{(i)}_{[0,1]}\right\rangle_{V^{\otimes i}} \\
&= \left\langle S(x)^{(k)}_{[0,1]},\ \sum_{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)} S(y)^{(i_1,\dots,i_k)}_{[0,1]} e_{i_1}\cdots e_{i_k}\right\rangle_{V^{\otimes k}} \\
&= \sum_{(i_1,\dots,i_k)\in\mathcal{P}(1,\dots,k)} S(x)^{(i_1,\dots,i_k)}_{[0,1]} S(y)^{(i_1,\dots,i_k)}_{[0,1]} \\
&= \frac{1}{k!} S(x)^{\mathcal{P}(1,\dots,k)}_{[0,1]},
\end{aligned}$$

where the interchange of summation and differentiation is justified by the uniform convergence of the series of derivatives.

□

A.4 Proof of Proposition 4.0.5

Proof. We define $\mathbf{e}_j^{(M+1)}(\mathbf{x})$, $\mathbf{e}_{j,l}^{(M+1)}(\mathbf{x})$ and d_j as in [1] and write $V_{M+1,k} = B_{k,M}$, $x_j = \beta_{j-1}$ to match the notation. By [1, Theorem 2] we have that

$$\alpha_i = (B_{k,M})_{i,1}^{-1} = (V_{M+1,k})_{i+1,1}^{-1} = \frac{(-1)^M \mathbf{e}_{M+1,i+1}^{(M+1)}(\mathbf{x})}{d_{i+1}}.$$

By [1, Equation 6], we have

$$\begin{aligned}
\mathbf{e}_{M+1,i+1}^{(M+1)}(\mathbf{x}) &= \frac{\partial}{\partial x_{i+1}} \mathbf{e}_{M+1}^{(M+1)}(\mathbf{x}) \\
&= \frac{\partial}{\partial x_{i+1}} \prod_{j=1}^{M+1} x_j \\
&= \prod_{\substack{j=1 \\ j \neq i+1}}^{M+1} x_j.
\end{aligned}$$

Substituting in $d_{i+1} = x_{i+1}^k \prod_{\substack{j=1 \\ j \neq i+1}}^{M+1} (x_{i+1} - x_j)$ and $x_j = \beta_{j-1}$ gives the result. □

A.5 Proof of Proposition 5.3.2

Proof. From the proof of Proposition 5.2.2.1, we have that

$$\max_{\substack{(j_i,\ldots,j_k) \in \mathcal{P}(1,\ldots k) \\ (j_i,\ldots,j_k) \neq (1,\ldots,k)}} S(p_N(y))^{(j_1,\ldots,j_k)} = \max_{1 \leq j < k} \frac{\sum_{i=1}^{j} n_i}{\sum_{i=1}^{j-1} n_i + n_{j+1}}.$$

Since $n_i = N^{\frac{i-1}{k-1}}$, we have

$$\begin{aligned}
\max_{1 \leq j < k} \frac{\sum_{i=1}^{j} n_i}{\sum_{i=1}^{j-1} n_i + n_{j+1}} &= \max_{1 \leq j < k} \frac{\sum_{i=1}^{j} N^{\frac{i-1}{k-1}}}{\sum_{i=1}^{j-1} N^{\frac{i-1}{k-1}} + N^{\frac{j}{k-1}}} \\
&= \max \left\{ N^{-\frac{1}{k-1}}, \ \max_{1<j<k} \frac{(N^{\frac{j}{k-1}} - 1)/(N^{\frac{1}{k-1}} - 1)}{(N^{\frac{j-1}{k-1}} - 1)/(N^{\frac{1}{k-1}} - 1) + N^{\frac{j}{k-1}}} \right\} \\
&= \max \left\{ N^{-\frac{1}{k-1}}, \ \max_{1<j<k} \frac{N^{\frac{j}{k-1}} - 1}{N^{\frac{j-1}{k-1}} - 1 + N^{\frac{j+1}{k-1}} - N^{\frac{j}{j-1}}} \right\} \\
&\leq \max \left\{ N^{-\frac{1}{k-1}}, \ \max_{1<j<k} \frac{N^{\frac{j}{k-1}}}{N^{\frac{j-1}{k-1}} + N^{\frac{j+1}{k-1}} - N^{\frac{j}{j-1}}} \right\} \\
&= \max \left\{ N^{-\frac{1}{k-1}}, \ \frac{1}{N^{-\frac{1}{k-1}} + N^{\frac{1}{k-1}} - 1} \right\} \\
&\leq \frac{1}{N^{\frac{1}{k-1}} - 1}.
\end{aligned}$$

□

A.6 Proof of Theorem 5.3.4

Proof. Let N be the $(k-1)^{th}$ power of an integer. By the construction of Chapter 4 and $p_N(y)$, we have

$$\begin{aligned}
&\left| \sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)} \alpha_i \sum_{j=0}^{k+M} \left\langle S(\beta_i \lambda \odot x)^{(j)}, S(p_N(y))^{(j)} \right\rangle_{V^{\otimes j}} - S(x)^{(1,\ldots,k)} \right| \\
&= \left| \sum_{(j_1,\ldots,j_k) \in \mathcal{P}(1,\ldots,k)} S(x)^{(j_1,\ldots,j_k)} S(p_N(y))^{(j_1,\ldots,j_k)} - S(x)^{(1,\ldots,k)} \right| \\
&= \left| \sum_{\substack{(j_1,\ldots,j_k) \in \mathcal{P}(1,\ldots,k) \\ (j_1,\ldots,j_k) \neq (1,\ldots,k)}} S(x)^{(j_1,\ldots,j_k)} S(p_N(y))^{(j_1,\ldots,j_k)} \right| \\
&\leq \frac{1}{N^{\frac{1}{k-1}} - 1} \left| \sum_{\substack{(j_1,\ldots,j_k) \in \mathcal{P}(1,\ldots,k) \\ (j_1,\ldots,j_k) \neq (1,\ldots,k)}} S(x)^{(j_1,\ldots,j_k)} \right| \qquad \text{(Proposition 5.3.2)} \\
&\to 0
\end{aligned}$$

as $N \to \infty$. It is easy to see that $\|p_N(y)\|_1 \leq k$, for instance, by considering the limiting axis path z of Proposition 6.1.1 and noting that

$$\|p_N(y)\|_1 \uparrow \|z\|_1 = k$$

as $N \to \infty$. We have, therefore, that

$$\begin{aligned}
&\left| k! \sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)} \alpha_i \sum_{j=k+M+1}^{\infty} \left\langle S(\beta_i \lambda \odot x)^{(j)}, S(p_N(y))^{(j)} \right\rangle_{V^{\otimes j}} \right| \\
&\leq k! \sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} |\alpha_i| \sum_{j=k+M+1}^{\infty} \left| \left\langle S(\beta_i \lambda \odot x)^{(j)}, S(p_N(y))^{(j)} \right\rangle_{V^{\otimes j}} \right| \\
&\leq k!\, 2^k \sum_{i=0}^{M} |\alpha_i| \sum_{j=k+M+1}^{\infty} \frac{\beta_i^j \|x\|_1^j k^j}{(j!)^2} \qquad \text{(Corollary 1.2.2.1)} \\
&\leq k!\, 2^k (M+1) \max_{0 \leq i \leq M} |\alpha_i| \sum_{j=k+M+1}^{\infty} \frac{\|x\|_1^j k^j}{(j!)^2} \\
&\leq k!\, 2^k (M+1) \max_{0 \leq i \leq M} |\alpha_i| \frac{(\|x\|_1 k)^{k+M}}{[(k+M)!]^2} \sum_{j=1}^{\infty} \frac{((k+M))^2}{((k+M+j)!)^2} \|x\|_1^j k^j \\
&\to 0 \qquad \text{(Condition (4.0.4))}
\end{aligned}$$

as $M \to \infty$. It follows that

$$\left| \sum_{i=0}^{M} \sum_{\lambda \in \{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)} \alpha_i \, \mathbf{k}_{x,\, p_N(y)}^{\lambda\text{-}\beta_i} - S(x)^{(1,\ldots,k)} \right| \to 0$$

as $N, M \to \infty$. □

A.7 Proof of Theorem 7.2.1

Proof. By Chen's relation (Proposition 1.2.7) and Proposition 1.2.6, it is easy to see that

$$S(z)^{(j_1,\dots,j_k)} = \frac{1}{l_1!\cdots l_m!}\mathbf{1}\{(j_1,\dots,j_k) \in \mathcal{P}(I_1) * \cdots * \mathcal{P}(I_m)\}$$

for all $(j_1,\dots,j_k)$. By the construction of Chapter 4, we have

$$\begin{aligned}
&\sum_{i=0}^{M} \sum_{\lambda\in\{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)}\alpha_i \sum_{j=0}^{k+M} \left\langle S(\beta_i\lambda \odot x)^{(j)}, S(z)^{(j)} \right\rangle_{V^{\otimes j}} \\
&= \sum_{(j_1,\dots,j_k)\in\mathcal{P}(1,\dots,k)} S(x)^{(j_1,\dots,j_k)} S(z)^{(j_1,\dots,j_k)} \\
&= \frac{1}{l_1!\cdots l_m!} S(x)^{\mathcal{P}(I_1)*\cdots*\mathcal{P}(I_m)}.
\end{aligned}$$

By the same steps as in the proof of Theorem 5.3.4 in Appendix A, Section A.6, we have

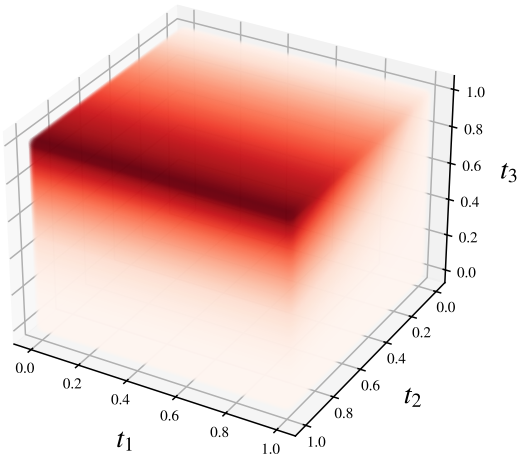
$$\left| k! \sum_{i=0}^{M} \sum_{\lambda\in\{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)}\alpha_i \sum_{j=k+M+1}^{\infty} \left\langle S(\beta_i\lambda \odot x)^{(j)}, S(z)^{(j)} \right\rangle_{V^{\otimes j}} \right| \to 0$$

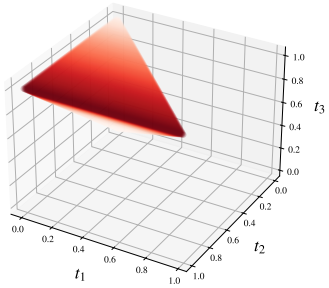
as $M \to \infty$, for any β_i satisfying Condition (4.0.4). It follows that

$$\left| \sum_{i=0}^{M} \sum_{\lambda\in\{0,1\}^k} (-1)^{\mathrm{sgn}(\lambda)}\alpha_i \, \mathbf{k}_{x,\,z}^{\lambda\text{-}\beta_i} - \frac{1}{l_1!\cdots l_m!} S(x)^{\mathcal{P}(I_1)*\cdots*\mathcal{P}(I_m)} \right| \to 0.$$

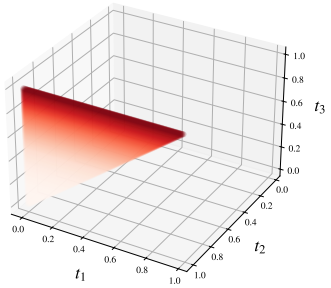
□

Appendix B A Detailed Breakdown of Example 5.1.1

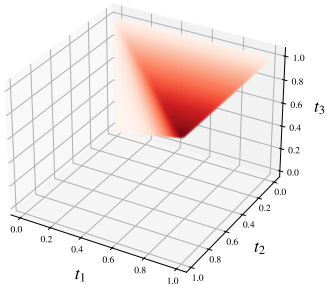




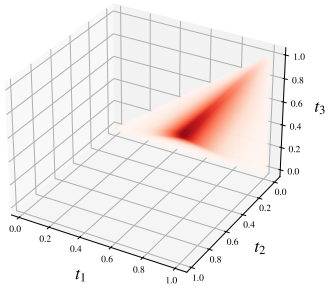
(a) $0 < t_1 < t_2 < t_3 < 1$



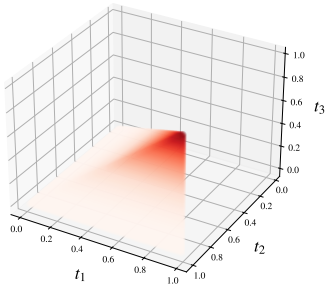
(b) $0 < t_1 < t_3 < t_2 < 1$



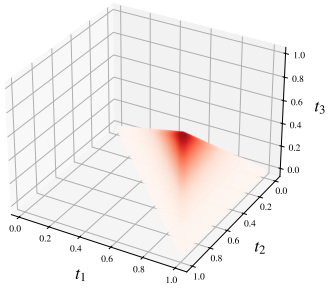
(c) $0 < t_2 < t_1 < t_3 < 1$



(d) $0 < t_2 < t_3 < t_1 < 1$



(e) $0 < t_3 < t_1 < t_2 < 1$



(f) $0 < t_3 < t_2 < t_1 < 1$

Figure B.1: $\dot{p}(y)^{(1)}_{t_1}\dot{p}(y)^{(2)}_{t_2}\dot{p}(y)^{(3)}_{t_3}$ for $n_1 = 1, n_2 = 2, n_3 = 4$, split by sections corresponding to signature coefficients in $\mathcal{P}(1, 2, 3)$.

Bibliography

[1] A. Arafat and M. El-Mikkawy, *A fast novel recursive algorithm for computing the inverse of a generalized vandermonde matrix*, Axioms, 12 (2022), p. 27.

[2] I. Beltagy, M. E. Peters, and A. Cohan, *Longformer: The long-document transformer*, arXiv:2004.05150, (2020).

[3] H. Boedihardjo, X. Geng, T. Lyons, and D. Yang, *The signature of a rough path: uniqueness*, Advances in Mathematics, 293 (2016), pp. 720–737.

[4] T. Cass, T. Lyons, and X. Xu, *Weighted signature kernels*, The Annals of Applied Probability, 34 (2024), pp. 585–626.

[5] T. Cass and C. Salvi, *Lecture notes on rough paths and applications to machine learning*, arXiv preprint arXiv:2404.06583, (2024).

[6] T. Cass and W. F. Turner, *Topologies on unparameterised path space*, Journal of Functional Analysis, 286 (2024), p. 110261.

[7] K.-T. Chen, *Iterated integrals and exponential homomorphisms*, Proceedings of the London Mathematical Society, 3 (1954), pp. 502–512.

[8] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, et al., *Rethinking attention with performers*, arXiv:2009.14794, (2020).

[9] L. Colmenarejo and R. Preiss, *Signatures of paths transformed by polynomial maps*, Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry, 61 (2020), pp. 695–717.

[10] M. A. de Oliveira and R. H. Ikeda, *Representation of the n-th derivative of the normal pdf using bernoulli numbers and gamma function*, Applied Mathematical Sciences, 6 (2012), pp. 3661–3673.

[11] B. Hambly and T. Lyons, *Uniqueness for the signature of a path of bounded variation and the reduced path group*, Annals of Mathematics, (2010), pp. 109–167.

[12] P. Kidger and T. Lyons, *Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU*, arXiv preprint arXiv:2001.00706, (2020).

[13] F. J. Király and H. Oberhauser, *Kernels for sequentially ordered data*, Journal of Machine Learning Research, 20 (2019), pp. 1–45.

[14] M. Lees, *The goursat problem*, Journal of the Society for Industrial and Applied Mathematics, 8 (1960), pp. 518–530.

[15] T. Lyons, *Rough paths, signatures and the modelling of functions on streams*, International Congress of Mathematicians, Seoul, (2014).

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

[17] J. Reizenstein and B. Graham, *The iisignature library: efficient calculation of iterated-integral signatures and log signatures*, arXiv preprint arXiv:1802.08252, (2018).

[18] C. Salvi, T. Cass, J. Foster, T. Lyons, and W. Yang, *The signature kernel is the solution of a Goursat PDE*, SIAM Journal on Mathematics of Data Science, 3 (2021), pp. 873–899.

[19] K. Schmüdgen, *Ten lectures on the moment problem*, arXiv preprint arXiv:2008.12698, (2020).

[20] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler, *Long range arena: A benchmark for efficient transformers*, arXiv preprint arXiv:2011.04006, (2020).

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, Advances in neural information processing systems, 30 (2017).

[22] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, *Linformer: Self-attention with linear complexity*, arXiv:2006.04768, (2020).

[23] K. Wen, X. Dang, and K. Lyu, *RNNs are not transformers (yet): The key bottleneck on in-context retrieval*, arXiv:2402.18510, (2024).