Signatures, Rough Paths and Applications in Machine Learning

Shengyang Zhou

a thesis submitted to the Department of Mathematics
at Utrecht University in partial fulfillment of the requirements for the degree of

Bachelor in Mathematics

supervisor: Dr. Karma Dajani

June 13, 2019

Universiteit Utrecht

Abstract

In the context of rough paths theory, the signature is a fundamental object that captures information about paths. Recent developments in this area has motivated the use of signatures as a nonparametric feature set for machine learning.

With this thesis, we aim to provide an accessible introduction to signatures from both a theoretical and applied point of view. On the theoretical side, we review algebraic, analytic and geometric properties of the signature. This touches on many different topics in pure mathematics, including differential equations, multilinear algebra and Lie theory. On the applied side, we review how these properties enable signatures to be effective feature sets in supervised learning problems. We study general algorithms and implementations for computing signatures, as well as specialized transformations for machine learning. Several classical problems such as handwritten digit recognition are explored in detail. Our main contribution to the existing literature is a benchmark of signatures for time series classification problems from the UCR repository.

Contents

1 Introduction

A notion that is encountered in virtually all branches of modern science is that of paths. There are many examples of objects that can be considered paths. A path could be a physical path in space to move from point A to B. A path could also be a sequence of events that describes some change in an observed quantity, e.g., the price movement of a financial instrument. Even more abstract, a path could represent a paragraph of text. In short, paths are ubiquitous and they seem to come in all sizes and shapes.

When dealing with paths, one is often interested in finding features that describe a path well. For instance, when a financial analyst is studying price movements of an instrument, they tend to search for signals that indicate important activity, be that a large order taking place or a sudden spike in volatility. Similarly, when a person (or machine) is reading a text, they will find themselves looking for keywords and sentences that contain more information than others. Likewise, a machine that is able to successfully classify handwritten characters will certainly be trained to look for spatial features and patterns that uniquely identify each character.

In all of these domains, specialized methods and routines are usually deployed to solve the respective problems at hand. However, from a mathematical point of view, there is not much difference between the paths. Whether the path represents a piece of text, a physical path, or a price process, it can always be realized as a function indexed by time. A natural question that arises is then: does there exist a more universal method to achieve this goal of information extraction, regardless of what the underlying application domain may be? Is there perhaps a general mathematical object that is able to succinctly extract and summarize key features of a path? While questions like these may sound vague and finding such object may be overly ambitious, in recent years, research in the novel field of rough paths has brought a fundamental object to light – one that has some of these desired capabilities.

1.1 Rough paths

Rough path theory was originally developed by T. Lyons in the 1990s [1] to study *controlled* differential equations of the form

$$dY_t = f(Y_t)dX_t, \quad Y_0 = y_0. \tag{1.1}$$

Here, f is a vector field and X_t is a known path, called the 'driving signal', and Y is the path to solve for, known as the 'response signal'. In the case that X and Y are smooth, the above equation could be equivalently written in a more familiar ordinary differential equation, obtained symbolically by 'dividing' both sides of the equation by dt,

$$\dot{Y}_t = f(Y_t)\dot{X}_t, \quad Y_0 = y_0.$$

For instance, a physical interpretation could be the following: Y_t is the position of a car at time t and X_t is the turning degree of the steering wheel at time t, while

f is a function that dictates the dynamics of the interaction. Solving the equation for Y given X would amount to finding how the position of the car changes in response to rotation of the steering wheel. Assuming X and f to be smooth, the classical Picard-Lindelöf theorem for ODEs tells us that not only does a unique solution exists, for fixed f the solution Y depends continuously on X in the uniform topology.

Rough path theory attempts to systematically extend (1.1) to the highly non-smooth setting. Typically, nonsmooth driving signals would occur in stochastic modelling. Readers familiar with the topic might view (1.1) as a *stochastic* differential equation (SDE), in which X and Y are random processes. A typical example of a process modelled in this framework is the price movement of a financial instrument. For instance, one can think of Y_t as the value of a financial derivative (e.g., an option), and X_t the price of the underlying instrument, both indexed by time. The solution of equation (1.1) then answers the question: How does the value of an option change when the value of the underlying stock changes?

Although sample paths of many stochastic processes are continuous, they are almost never smooth. For instance, a common choice for a random driving signal X is the Brownian motion B_t, whose sample paths are in fact nowhere differentiable. It is therefore a nontrivial task to even define what equation (1.1) means for random processes. This is precisely the content of the theory of Itô calculus. In essence, (1.1) is made rigorous by using a stochastic version of integration, the Itô integral. A random process solves the SDE (1.1) if and only if it solves the corresponding Itô integral equation,

$$Y_t = Y_0 + \int_0^t f(Y_s)\, dX_s.$$

The important fact to note here is that Itô's framework is probabilistic. There are fundamental reasons why developing a deterministic framework of integration for highly nonsmooth paths cannot work. Itô's framework circumvents these limits by exploiting certain probabilistic niceties (i.e., martingale and semi-martingale properties) present in random processes. One drawback of the theory is that the integral as a map $(X, Y) \mapsto \int_0^T Y_s\, dX_s$ lacks any form of continuity; we cannot equip the path space with a meaningful topology such that the integral is continuous [2, Prop 1.1]. Even so, Itô calculus is hugely successful and has been a major advancement in 20th century mathematics.

The theory of rough paths has been developed partly to address some of these issues present in stochastic integration. Most notably, rough paths answer the following main question that the stochastic framework cannot address: If X and X' are driving signals of (1.1) that are 'nearby', what can we say about the respective solutions Y and Y'? In other words, do we have a meaningful topology or metric on the space of paths such that the solution map $I_f : X \mapsto Y$ of equation (1.1) is uniformly continuous? The answer is positive, and we briefly outline the approach.

Rough path theory achieves continuity in the solution map by 'enhancing' the paths with additional information. Given a path X, we include a special, higher-

order process $\mathbb{X}$ and form the pair $\mathbf{X} = (X, \mathbb{X})$. The enhanced object $t \mapsto \mathbf{X}_t$ is a path taking values in a larger space, and is called the *rough path* obtained by *lifting* X. Given this additional data, it is possible to define what it means to integrate another path Y against the rough path $\mathbf{X}$. Furthermore, under the so-called *rough path metric*, the integral is a continuous map [1, Thm 3.2.2], [2, Thm 4.10]. Once a suitable theory of integration is established, it is possible to consider rough differential equations of the form

$$dY_t = f(Y_t)d\mathbf{X}_t, \quad Y_0 = y_0.$$

The universal limit theorem [1, Thm 4.1.1], [2, Thm 8.4] then guarantees that a unique solution exists, and that the solution map (also known as the Itô-Lyons map) $\tilde{I}_f : \mathbf{X} \mapsto (Y, Y')$ is continuous in the rough path metric. Here, the second object Y' could be interpreted as a 'derivative' of Y. The idea is summarized in the following diagram.

$$\begin{array}{ccc} (X, \mathbb{X}) & \xrightarrow{\quad\tilde{I}_f\quad} & (Y, Y') \\ \text{lift}\uparrow & & \downarrow\text{project} \\ X & \dashrightarrow^{I_f} & Y \end{array}$$

The map I_f can for instance represent the Itô map for the classical SDE, i.e., (1.1). The dashed line emphasizes the fact that I_f is not continuous.

Clearly, many crucial details are omitted in this brief outline. However, the goal of this discussion is not to motivate the theory of rough paths itself, instead, the emphasis lies in the procedure of gaining stronger results by including additional information. The question that begs for an answer is therefore: What is exactly this additional information about a path that we need for this to work?

1.2 The signature

Let us consider a simplified version of the controlled differential equation (1.1). In particular, suppose $f = A$ is linear and X is smooth, then (1.1) becomes the linear equation

$$dY_t = AY_t dX_t, \quad Y_0 = y_0. \tag{1.2}$$

A well-known method to approximate solutions to such classical ODEs is through Picard iterations, which are defined recursively as follows:

$$Y_t^0 = y_0, \qquad Y_t^n = Y_0 + \int_0^t AY_s^{n-1}\, dX_s.$$

It is the content of the Picard-Lindelöf theorem that $(Y^n)_n$ converges uniformly to the solution Y. Since A is linear, we can in fact write explicitly

$$Y_t = \left(\sum_{n=0}^{\infty} A^n \int_{\{0<s_1<\cdots<s_n<t\}} dX_{s_1} \otimes \cdots \otimes dX_{s_n} \right) y_0$$

As a result, the solution to the linear equation is only dependent on A and this collection of iterated integrals in X.

Let us call this collection the *signature* of the smooth path X, and denote it by $S(X)$,

$$S(X)_{0,t} = \left(1, x^1, x^2, \ldots, x^n, \ldots\right)$$

with

$$x^n = \int_{\{\,0<s_1<\cdots<s_n<t\,\}} dX_{s_1} \otimes \cdots \otimes dX_{s_n}.$$

At first sight, this object may not seem very special. However, a stronger and perhaps much more surprising result is true: for any fixed smooth function f, not necessarily linear, the solution Y to (1.1) is completely determined by the signature of X. This key observation by Chen [3] is in fact the foundation of the theory of rough paths. Indeed, for the case of smooth paths, the object $\mathbb{X}$ mentioned in the previous discussion is precisely the signature.

In this view, the signature is a powerful object that is able to fully determine the response of a system given an input signal. This conclusion should serve as a first indication as to why the signature fits the description of being a nonparametric way to summarize a path, as alluded in the beginning of this introduction. In the rest of this thesis, our goal will be to further expand and elaborate on this idea.

1.3 Outline of thesis

The current thesis is divided in two chapters, one that deals with the theoretical aspect of signatures and the other that deals with applications. Our goal is to provide a self-contained introduction to signatures that attempts to combine both theoretical groundwork and the extension to practical applications, with particular focus on machine learning.

The theoretical Chapter 2 builds the foundation of signatures. In this chapter, we introduce the signature for $\mathbf{R}^d$-valued paths of bounded variation. Section 2.1 introduces the space on which signatures live, known as the extended tensor algebra over $\mathbf{R}^d$, which is an associate algebra constructed from a direct sum of vector spaces. In Section 2.2 and 2.3, we define the signature and introduce its main algebraic properties, most notably that the signature is a special morphism between the path space and the tensor algebra. Section 2.4 is concerned with functions on the space of signatures. Here, we formulate and prove the remarkable property that products of linear functionals acting on signatures are again linear functionals. In Section 2.5 and 2.6, we explore analytic and geometric properties of the range of the signature. We will show that the space of signatures has a rich geometric structure, being a special type of Lie group.

One important topic we have omitted is the theory of rough paths, as described in the first section of the introduction. Although signatures have traditionally been a fundamental part of rough paths, we have settled for a treatment of signatures independent from rough paths. As such, the paths we deal with are not 'rough'

paths, but ones that are rather smooth. Nevertheless, the theory of signatures for these non-rough paths is rich and nontrivial. It touches on many different topics of mathematics, notably including Lie theory, sub-Riemannian geometry and Hopf algebras.[1] For a thorough introduction to rough paths, we refer to the book [2].

On the practical side, we focus on recent applications of signatures in machine learning. Our goal is to illustrate using several simple but meaningful examples that signatures can be effective feature sets. In the introduction of Chapter 3 we present main principles that make signatures effective from a theoretical viewpoint. Here we introduce the general machine learning workflow and link key results from the previous theoretical exposition to practical implications in machine learning. Section 3.1 then treats various data preprocessing steps for the signature method. Most methods here are a variant of turning discrete paths into continuous paths, for which we can compute the signature. Section 3.2 focuses on the computation of signatures. Through mostly examples, we illustrate a few algorithms for computing signatures. In Section 3.3, we consider two concrete machine learning problems. In the first example, we use a simple linear regression model on signature features to predict the terminal value of a stochastic process driven by a Brownian motion. The second example attempts to classify handwritten digits using signatures. Both examples have been briefly discussed in different texts already, our discussion will provide a different perspective by including techniques discussed in Section 3.1.

Finally, in Section 3.4, we perform an extensive benchmark of signature features on 114 datasets of the UCR Time Series Classification repository [4]. With this numerical experiment, we aim to better understand the effectiveness of signatures for machine learning in a broader context. This aspect still seems lacking in the current literature, as most papers published so far tend to only consider one specific machine learning problem. We are not aware of any published work that attempts to survey the performance of signatures across a range of different problems and datasets. Fortunately, the UCR repository has been crafted for precisely this purpose in mind. In our experiments, we will not only test signatures under different configurations, but also compare them against the results of other state-of-the-art classification algorithms.

[1] These advanced topics are out of the scope for this thesis, however, we provide references for interested readers.

2 Theory of Signatures

In this chapter we treat a key object in the study of paths: the signature. The goal of this chapter is to provide an accessible introduction to the theory of signatures, complete with proofs.

Let us introduce the notation we use in this chapter. We use capital letters X, Y and Z to denote paths. Most of the time, these are functions from a compact interval $[0, T]$ to d-dimensional Euclidean space $\mathbf{R}^d$, i.e., $X : [0, T] \to \mathbf{R}^d$. We use the subscript notation X_t to denote the value of the path at time t. Paths will always be assumed to be continuous and of bounded variation, see Definition A2. The set of all such paths is denoted by $\mathcal{V}([0, T], \mathbf{R}^d)$.

2.1 Tensor series

Before we define the signature of paths, we introduce the space they live in. Let V be a finite dimensional vector space. Recall that for any nonnegative integer n, the nth tensor power of V is defined to be the tensor product of V with itself n times,

$$V^{\otimes n} = V \otimes \cdots \otimes V.$$

By convention, the zeroth tensor power $V^{\otimes 0}$ is taken to be $\mathbf{R}$. An introduction to tensors can be found in any book on multilinear algebra, see for instance [5, Chapter 12]; the exposition will be omitted here. However, it is beneficial to state the notation we will use. Suppose $\{e_1, \dots, e_d\}$ is a basis of V, then the corresponding basis of $V^{\otimes n}$ is given by $\{ e_{i_1} \otimes \cdots \otimes e_{i_n} \mid 1 \le i_1, \dots, i_n \le d \}$. Hence, any $v \in V^{\otimes n}$ can be written in coordinates as

$$v = \sum_{1 \le i_1, \dots, i_n \le d} v^{i_1, \dots, i_n} e_{i_1} \otimes \cdots \otimes e_{i_n}.$$

Definition 1. Consider a finite dimensional vector space V over $\mathbf{R}$. The *extended tensor algebra* $T((V))$ over V is defined as the space

$$T((V)) = \left\{ a = (a_0, a_1, \dots) \;\middle|\; a_n \in V^{\otimes n}, \text{ for all } n \ge 0 \right\}.$$

It is equipped with the usual element-wise addition and scalar product: for $\lambda \in \mathbf{R}$ and $a = (a_0, a_1, \dots)$ and $b = (b_0, b_1, \dots)$ in $T((V))$,

$$a + b = (a_0 + b_0,\ a_1 + b_1, \dots) \quad \text{and} \quad \lambda a = (\lambda a_0,\ \lambda a_1, \dots).$$

Furthermore, $T((V))$ has a product operation, given by

$$a \otimes b = (c_0, c_1, \dots)$$

where for each $n \ge 0$,

$$c_n = \sum_{k=0}^{n} a_k \otimes b_{n-k}. \tag{2.1}$$

⊘

It is important to note that the multiplication defined above is noncommutative, i.e., $a \otimes b \neq b \otimes a$, because the tensor product is noncommutative. With respect to $\otimes$, $T((V))$ has a unit element $1 = (1, 0, 0, \dots)$. The space $T((V))$ therefore has the structure of an *associative algebra*.

The product operation may remind the reader of products of power series. Indeed, if we have two (formal) power series $\sum_{n=0}^{\infty} a_n z^n$ and $\sum_{n=0}^{\infty} b_n z^n$ then their product

$$\left(\sum_{n=0}^{\infty} a_n z^n\right)\left(\sum_{n=0}^{\infty} b_n z^n\right) = \sum_{n=0}^{\infty} c_n z^n$$

will have coefficients c_n given by

$$c_n = \sum_{k=0}^{n} a_k b_{n-k}. \tag{2.2}$$

As such, (2.1) is 'the same' as (2.2), but consists of a different product.

It is possible to make this correspondence between $T((V))$ and formal power series more precise. Assuming a chosen basis $\{ e_1, \dots, e_d \}$ of V, we can actually view $T((V))$ as a formal power series in the indeterminates $e_{i_1,\dots i_n} := e_{i_1} \otimes \cdots \otimes e_{i_n}$, for $n \geq 0$ and $1 \leq i_1, \dots, i_n \leq d$. Any $a \in T((V))$ can therefore be written in coordinates as

$$a = \sum_{n \geq 0} \sum_{i_1,\dots,i_n} a^{i_1,\dots,i_n} e_{i_1,\dots,i_n}.$$

If b is another element with coefficients $b^{i_1,\dots,i_n}$, and $c = a \otimes b$, then the coefficients of c are given by

$$c^{i_1,\dots,i_n} = \sum_{k=0}^{n} a^{i_1,\dots,i_k} b^{i_{k+1},\dots,i_n}. \tag{2.3}$$

Here, we adopt the convention that the empty index a^0 (at $k = 0$) represents the scalar $a_0 \in V^{\otimes 0} \simeq \mathbf{R}$.

2.2 Signature of paths

The signature of a path is defined as a sequence of iterated integrals over simplices of increasing dimension. The type of paths we work with are functions of type $X : [0, T] \to \mathbf{R}^d$. Furthermore, we assume that these paths are continuous and of bounded variation. The space of such paths will be denoted by $\mathcal{V}([0, T], \mathbf{R}^d)$ and we will write $\|X\|_{1\text{-var}}$ for the total variation of X. The definitions can be found in Appendix A.

Definition 2. Let $X : [0, T] \to \mathbf{R}^d$ be a continuous path of bounded variation and let $\{e_1, \dots, e_d\}$ be the standard basis of $\mathbf{R}^d$, such that $X_t = \sum_{i=1}^{d} X_t^i e_i$. The *signature* $S(X)_{0,T}$ of X is the element

$$S(X)_{0,T} = \sum_{n \geq 0} \sum_{1 \leq i_1,\dots,i_n \leq d} S(X)_{0,T}^{i_1,\dots,i_n} \, e_{i_1} \otimes \cdots \otimes e_{i_n} \in T((\mathbf{R}^d))$$

in which the coefficients are defined recursively as follows:

$$S(X)^0_{0,T} = 1, \quad \text{and} \quad S(X)^{i_1,\dots,i_{n+1}}_{0,T} = \int_0^T S(X)^{i_1,\dots,i_n}_{0,s}\, dX_s^{i_{n+1}}. \tag{2.4}$$

Here the zeroth level $S(X)^0_{0,T} = 1$ (empty index) refers to the scalar component of $S(X)_{0,T}$ in $T((\mathbf{R}^d))$. We also call the corresponding map $S : \mathcal{V}([0,T], \mathbf{R}^d) \to T((\mathbf{R}^d))$ the *signature map*. ⊘

A more direct way to write $S(X)^{i_1,\dots,i_n}_{0,T}$ is by

$$S(X)^{i_1,\dots,i_n}_{0,T} = \int_{0<s_1<\cdots<s_n<T} dX^{i_1}_{s_1} \cdots dX^{i_n}_{s_n}.$$

Of course, we can also break down the signature according to the grading of $T((\mathbf{R}^d))$:

$$S(X)_{0,T} = (1, x_1, x_2, \dots) \in T((\mathbf{R}^d)),$$

where x_n is the corresponding tensor in $(\mathbf{R}^d)^{\otimes n}$. For instance, one can readily see from Definition 2 that x_1 will be precisely the total displacement of the path X, $x_1 = X_T - X_0 \in (\mathbf{R}^d)^{\otimes 1} \simeq \mathbf{R}^d$. The term x_n will be referred to as the nth level of the signature. In the literature, a common way to write x_n is

$$x_n = \int_{0<s_1<\cdots<s_n<T} dX_{s_1} \otimes \cdots \otimes dX_{s_n}$$

Example 3 (Constant path)**.** Consider the constant path $X : [0,T] \to \mathbf{R}^d$ given by $X_t = v$ for all $t \in [0,T]$. As noted before, the first level signature of a path is always the total displacement of the path. Hence, the first level signature of the restricted path $X|_{[0,s]}$ is clearly zero for all $s \in [0,T]$. By induction, any higher level signatures must also be zero. It follows that $S(X)_{0,T} = (1,0,0,\dots)$, which is the unit element of $T((\mathbf{R}^d))$. ⊘

Example 4 (Linear path)**.** Consider a linear planar path $X : [0,1] \to \mathbf{R}^d$ given by $X_t = tv$ for some nonzero vector $v \in \mathbf{R}^d$. We claim that the level n signature of X is given by $v^{\otimes n}/n!$, so that

$$S(X)_{0,1} = \left(1,\ v,\ \frac{1}{2}v^{\otimes 2},\ \frac{1}{6}v^{\otimes 3}, \dots\right).$$

A proof of this fact using coordinates is rather tedious, instead, we can argue as follows due to smoothness of X. Let $x_n(s)$ denote the level n signature of the restricted path $X|_{[0,s]}$ for $s \in [0,1]$. It is clear that $x_1(s) = sv \in \mathbf{R}^d$, we show that $x_n(s) = s^n v^{\otimes n}/n!$ for all s. Indeed, by smoothness of X, the Stieltjes integral for higher level signatures can be written as

$$\begin{aligned} x_{n+1}(t) &= \int_0^t x_n(s) \otimes \dot{X}_s\, ds = \int_0^t \frac{s^n v^{\otimes n}}{n!} \otimes v\, ds \\ &= \frac{v^{\otimes n+1}}{n!} \int_0^t s^n\, ds = \frac{t^{n+1} v^{\otimes n+1}}{(n+1)!}. \end{aligned}$$ ⊘

Example 5 (Closed path)**.** Let $X : [0,1] \to \mathbf{R}^2$ be a closed, simple, piecewise smooth path. Let A denote the area enclosed by X. By Green's theorem,

$$A = \int_{[0,1]} X^1 dX^2 = S^{1,2}(X)_{0,1}.$$

Hence, the second level of the signature contains information about the area enclosed by a path. ⊘

Example 6 (Translated path)**.** Consider a path $X : [0,T] \to \mathbf{R}^d$ and its translation $X + v$ by a vector v. Since the first level of the signature is the total displacement of the path and is independent of the starting point, a simple induction argument similar to Example 3 shows that $S(X+v)_{0,T} = S(X)_{0,T}$. ⊘

2.3 Main Properties

We summarize the key properties of signatures in this subsection. The proofs we give here are elementary and are all based on induction. The first result tells us that the signature is invariant under reparametrizations.

Proposition 7. *Let $X : [T', T''] \to \mathbf{R}^d$ be a continuous path of bounded variation and let $\phi : [0,T] \to [T', T'']$ be a nondecreasing surjection. Then*

$$S(X \circ \phi)_{s,t} = S(X)_{\phi(s),\phi(t)} \quad \textit{for all } s,t \in [0,T].$$

Proof. This follows immediately from the substitution theorem of Riemann-Stieltjes integrals, see Proposition B3. □

As a result of this proposition, it does not matter what type of interval we use for the domain of paths, we can always reparametrize to obtain the interval $[0,T]$ or $[0,1]$.

Next, we show that if we have two paths X and Y and we concatenate them, then the signature of the concatenation is simply the product of the signatures.

Definition 8 (Concatenation)**.** Let $X : [0,T] \to \mathbf{R}^d$ and $Y : [T,2T] \to \mathbf{R}^d$ be paths. The concatenation of X and Y is defined to be the path $X * Y : [0,2T] \to \mathbf{R}^d$ given by

$$(X * Y)_t = \begin{cases} X_t & 0 \le t \le T, \\ X_T + Y_t - Y_T & T < t \le 2T. \end{cases}$$ ⊘

Proposition 9 (Chen)**.** *Let $X : [0,T] \to \mathbf{R}^d$ and $Y : [T,2T] \to \mathbf{R}^d$ be paths of bounded variation. Then the signature satisfies Chen's identity,*

$$S(X * Y)_{0,2T} = S(X)_{0,T} \otimes S(Y)_{T,2T}.$$

Proof. Write $Z = X * Y$, we prove the stronger result

$$S(Z)_{0,r} = S(Z)_{0,T} \otimes S(Z)_{T,r}$$

for any $r \in [T, 2T]$. In coordinates, this means

$$S(Z)^{i_1,\dots,i_n}_{0,r} = \sum_{k=0}^{n} S(Z)^{i_1,\dots,i_k}_{0,T} S(Z)^{i_{k+1},\dots,i_n}_{T,r}, \tag{*}$$

for any multi-index $(i_1, \dots, i_n)$, see (2.3).

We proceed by induction to n. If $n = 0$, both sides of (*) are equal to 1 regardless of r. Suppose the statement holds for $n \geq 0$. For $n + 1$, we have

$$\begin{aligned}
S(Z)^{i_1,\dots,i_{n+1}}_{0,r} &= \int_0^r S(Z)^{i_1,\dots,i_n}_{0,u}\, dZ^{i_{n+1}}_u \\
&= \int_0^T S(Z)^{i_1,\dots,i_n}_{0,u}\, dZ^{i_{n+1}}_u + \int_T^r S(Z)^{i_1,\dots,i_n}_{0,u}\, dZ^{i_{n+1}}_u \\
&= S(Z)^{i_1,\dots,i_{n+1}}_{0,T} + \int_T^r S(Z)^{i_1,\dots,i_n}_{0,u}\, dZ^{i_{n+1}}_u \\
&= S(Z)^{i_1,\dots,i_{n+1}}_{0,T} + \int_T^r \sum_{k=0}^{n} S(Z)^{i_1,\dots,i_k}_{0,T} S(Z)^{i_{k+1},\dots,i_n}_{T,u}\, dZ^{i_{n+1}}_u \\
&= S(Z)^{i_1,\dots,i_{n+1}}_{0,T} + \sum_{k=0}^{n} S(Z)^{i_1,\dots,i_k}_{0,T} \int_T^r S(Z)^{i_{k+1},\dots,i_n}_{T,u}\, dZ^{i_{n+1}}_u \\
&= S(Z)^{i_1,\dots,i_{n+1}}_{0,T} + \sum_{k=0}^{n} S(Z)^{i_1,\dots,i_k}_{0,T} S(Z)^{i_{k+1},\dots,i_{n+1}}_{T,r} \\
&= \sum_{k=0}^{n+1} S(Z)^{i_1,\dots,i_k}_{0,T} S(Z)^{i_{k+1},\dots,i_{n+1}}_{T,r}.
\end{aligned}$$

In the fourth line, we used the induction hypothesis on $S(Z)^{i_1,\dots,i_n}_{0,u}$, as $u \in [T, r] \subset [T, 2T]$. This concludes the proof. □

Example 10 (Piecewise-linear path)**.** Recall that in Example 4, we showed that a linear path X with total displacement $v \in \mathbf{R}^d$ has a signature given by the exponential series,

$$S(X) = \left(1,\ v,\ \frac{1}{2}v^{\otimes 2}, \dots,\ \frac{1}{n}v^{\otimes n!}, \dots\right).$$

Let us define $\exp(v)$ to be the series on the right.

Consider now the piecewise-linear path Y that goes through the points $p_0, \dots, p_m$ in $\mathbf{R}^d$. Chen's identity gives us an useful recipe to compute the signature of Y: break down the path into its m linear segments, compute the signature of each

linear segment, and then take the product of all these signatures. We know that the signature of the linear segment from p_k to p_{k+1} is simply $\exp(p_{k+1} - p_k)$, hence, the signature of Y is

$$S(Y) = \exp(p_1 - p_0) \otimes \cdots \otimes \exp(p_m - p_{m-1}). \qquad \oslash$$

The next proposition tells us that the signature of a path run backwards is the same as the inverse of the signature.

Proposition 11. *Let $X : [0, T] \to \mathbf{R}^d$ be a path of finite variation and let X^{-1} be X run backwards, that is, X^{-1} is the path given by $X_t^{-1} = X_{T-t}$. Then $S(X^{-1}) = S(X)^{-1}$.*

Proof. Using Chen's identity, it is equivalent to show that

$$S(X * X^{-1})_{0,2T} = 1.$$

Write $Z = X * X^{-1} : [0, 2T] \to \mathbf{R}^d$. We prove the slightly stronger result $S(Z)_{0,t} = S(Z)_{0,2T-t}$ for all $t \in [0, 2T]$, which in coordinates means

$$S(Z)_{0,t}^{i_1,\ldots,i_n} = S(Z)_{0,2T-t}^{i_1,\ldots,i_n}. \qquad (*)$$

We proceed by induction to n. The base case $n = 0$ is trivial, assume the statement holds for some $n \geq 0$. If $t \in [0, T]$, we have for $n + 1$,

$$\begin{aligned} S(Z)_{0,2T-t}^{i_1,\ldots,i_{n+1}} &= \int_0^{2T-t} S(Z)_{0,s}^{i_1,\ldots,i_n} \, dZ_s^{i_{n+1}} \\ &= \int_0^t S(Z)_{0,s}^{i_1,\ldots,i_n} \, dZ_s^{i_{n+1}} \\ &\quad + \int_t^T S(Z)_{0,s}^{i_1,\ldots,i_n} \, dZ_s^{i_{n+1}} + \int_T^{2T-t} S(Z)_{0,s}^{i_1,\ldots,i_n} \, dZ_s^{i_{n+1}}. \end{aligned}$$

However, we know that $Z_s = Z_{2T-s}$ and by the induction hypothesis,

$$\begin{aligned} \int_T^{2T-t} S(Z)_{0,s}^{i_1,\ldots,i_n} \, dZ_s^{i_{n+1}} &= \int_T^{2T-t} S(Z)_{0,2T-s}^{i_1,\ldots,i_n} \, dZ_{2T-s}^{i_{n+1}} \\ &= \int_T^t S(Z)_{0,u}^{i_1,\ldots,i_n} \, dZ_u^{i_{n+1}} \\ &= -\int_t^T S(Z)_{0,u}^{i_1,\ldots,i_n} \, dZ_u^{i_{n+1}}. \end{aligned}$$

To obtain the second line, we used the substitution $u = 2T - s$, see Proposition B3. It follows that the last two terms in the equation for $S(Z)_{0,2T-t}^{i_1,\ldots,i_{n+1}}$ cancel, so (*) holds for $n + 1$ and $t \in [0, T]$. If on the other hand $t \in [T, 2T]$ then we can use the same formula for $2T - t \in [0, T]$ to obtain the same result. □

An immediate consequence of the last two propositions is that the range $S \circ \mathcal{V}([0, T], \mathbf{R}^d)$ forms a group under $\otimes$. Furthermore, the signature is a structure-preserving map between $\mathcal{V}([0, T], \mathbf{R}^d)$ and $T((\mathbf{R}^d))$.

2.3.1 One-to-one correspondence

So far we have shown that the signature maps satisfies a few useful algebraic properties. An important question that the reader might consider is to what extent does the signature determine the path? The paper [6] gives a full answer to this question for paths of bounded variation. In this paper, it is proven that the signature of a path is trivial (i.e., $S(X) = (1, 0, 0, \dots)$) if and only if the path is *tree-like*. The definition follows.

Definition 12. A path $X \in \mathcal{V}([0,T], \mathbf{R}^d)$ is said to be *tree-like* if there exists a nonnegative continuous function $h : [0,T] \to \mathbf{R}$ of bounded variation with $h_0 = h_T$, such that

$$\|X_t - X_s\|_{\mathbf{R}^d} \leq h(s) + h(t) - 2 \inf_{u \in [s,t]} h(u)$$

Two paths $X, Y \in \mathcal{V}([0,T], \mathbf{R}^d)$ are said to be tree-like equivalent if $X * Y^{-1}$ is tree-like. ⊘

This definition for a tree-like path may be hard to interpret geometrically. Recall from Proposition 11 that the signature of $X * X^{-1}$ is trivial, for any path $X \in \mathcal{V}([0,T], \mathbf{R}^d)$. Essentially, the above definition attempts to analytically describe just those paths that look like $X * X^{-1}$, i.e., paths that 'reset themselves'. However, not all such paths are of the form $X * X^{-1}$, for instance, $X * Y * Y^{-1} * Z * Z^{-1} * X^{-1}$ is not. There is a very intuitive geometric characterization of tree-like equivalence due to [7, Thm 5.15]: A loop (i.e., a path with same starting and end point) is tree-like if and only if it is contractible within its own range. Here being contractible is in the sense of homotopy: a path is contractible if it can be continuously 'deformed' into a constant path. This equivalence makes it easy to check whether a path is tree-like or not.

We now formulate the main uniqueness result.

Theorem 13 (Uniqueness of signature, [6])**.** *The signature of a path of bounded variation is trivial if and only if the path is tree-like. Hence, two paths have the same signature if and only if they are tree-like equivalent.*

2.4 Dual space and linear functionals

In this section, we study the range of the signature map and linear functionals on this space. As it turns out, there is a natural product on the dual space of the range of the signature map, which gives it the structure of an algebra. The exposition follows that of [8].

Recall that the algebraic dual space of a vector space V, denoted by V^*, is the set of all linear functions from V to $\mathbf{R}$. If V is finite dimensional, then V and V^* have the same dimension. Assume this to be the case, given a basis $\{e_1, \dots e_d\}$ for

V, we write $\{e^1, \ldots, e^d\}$ for the corresponding dual basis of V^*. We know that the set

$$\{\, e_I := e_{i_1} \otimes \cdots \otimes e_{i_n} \mid 1 \leq i_1, \ldots, i_n \leq d \,\}$$

forms a basis of $V^{\otimes n}$, and similarly,

$$\left\{ e^I := e^{i_1} \otimes \cdots \otimes e^{i_n} \;\middle|\; 1 \leq i_1, \ldots, i_n \leq d \right\}$$

forms a basis of $(V^*)^{\otimes n}$. We can now identify $\left(V^{\otimes n}\right)^*$ with $(V^*)^{\otimes n}$ through the pairing

$$\left\langle e^I, e_J \right\rangle = e^I(e_J) = \delta^I_J,$$

that is, we identify $e^I \in (V^*)^{\otimes n}$ with the dual element of e_I in $\left(V^{\otimes n}\right)^*$.

Let us now extend the action of $e^I \in (V^*)^{\otimes n}$ on $V^{\otimes n}$ to $T((V))$ as follows: for $a = (a_0, a_1, \ldots) \in T((V))$,

$$e^I(a) = e^I(a_n) \in \mathbf{R}.$$

This gives a linear map $(V^*)^{\otimes n} \to T((V))^*$. Evaluating e^I on a is essentially picking up the I coefficient of a. For example, using our notation for the signature coefficients, if $I = (i_1, \ldots, i_n)$ then

$$e^I\left(S(X)_{0,T}\right) = S(X)^I_{0,T} = S(X)^{i_1,\ldots,i_n}_{0,T}.$$

The next step is to take finite linear combinations of such linear functionals across tensor products of any length.

Definition 14. Let V be a finite dimensional vector space. We define $T(V^*)$ to be the direct sum

$$T(V^*) := \bigoplus_{n=0}^{\infty} \left(V^*\right)^{\otimes n}.$$

By the above identifications, $T(V^*)$ can be viewed as a strict subspace of the set of all linear functionals $T((V))^*$ of $T((V))$. ⊘

It is important to emphasize that an element of $T(V^*)$ seen as a sequence, only has finitely many terms that are nonzero. This allows elements of $T(V^*)$ to act on $T((V))$ without any issues of convergence.

Next, we introduce a product operation on $T(V^*)$. For this, we recall a special type of permutation.

Definition 15. A permutation σ on the set $\{1, \ldots, n+k\}$ is called an (n,k)-shuffle if the internal order of the first n and last k elements is preserved:

$$\sigma(1) < \cdots < \sigma(n) \quad \text{and} \quad \sigma(n+1) < \cdots < \sigma(n+k).$$

The set of all (n,k)-shuffles is denoted by $\mathrm{Sh}(n,k)$.

If we have two indices $I = (i_1, \dots, i_n)$ and $J = (j_1, \dots, j_n)$ and write $(r_1, \dots, r_{n+k}) = (i_1, \dots, i_n, j_1, \dots, j_k)$, we define $I \sqcup J$ as the set of all shuffles of indices in I and J

$$I \sqcup J = \left\{ \left(r_{\sigma(1)}, \dots, r_{\sigma(n+k)}\right) \;\middle|\; \sigma \in \mathrm{Sh}(n,k) \right\}. \qquad \oslash$$

Definition 16. Let I, J be indices and let $e^I, e^J \in T(V^*)$. We define the *shuffle product* between these two elements as

$$e^I \sqcup e^J = \sum_{K \in I \sqcup J} e^K.$$

The product $\sqcup$ is then defined on all elements of $T(V^*)$ by linear extension. $\oslash$

By construction, $\sqcup : T(V^*) \times T(V^*) \to T(V^*)$ is bilinear and commutative.

Theorem 17. *Let $a \in T((\mathbf{R}^d))$ be the signature of some path $X \in \mathcal{V}([0,T], \mathbf{R}^d)$. Then for any $e^*, f^* \in T(V^*)$,*

$$e^*(a) f^*(a) = (e^* \sqcup f^*)(a).$$

Proof. By linearity of $\sqcup$, it is sufficient to prove this for $e^* = e^I$ and $f^* = e^J$ where $I = (i_1, \dots, i_n)$ and $J = (j_1, \dots, j_k)$. That is, we show that

$$S(X)^I_{0,T} \, S(X)^J_{0,T} = \sum_{K \in I \sqcup J} S(X)^K_{0,T}.$$

We proceed by induction to the combined number of indices $N = |I| + |J|$. Clearly, the statement holds if $n = 0$ or $k = 0$, which covers the case $N = 1$. Suppose the statement is true for $N - 1 \geq 1$, we verify the case for N, assuming that $n \geq 1$ and $k \geq 1$. By the integration by parts formula (Proposition B4),

$$S(X)^I_{0,T} \, S(X)^J_{0,T} = \int_0^T S(X)^I_{0,s} dS(X)^J_{0,s} + \int_0^T S(X)^J_{0,s} dS(X)^I_{0,s}. \qquad (*)$$

Since $S(X)^J_{0,s}$ is an integral defined by equation (2.4), we can apply the associative property of the Riemann-Stieltjes integral (Proposition B5) to the first term of (*),

$$\begin{aligned}
\int_0^T S(X)^I_{0,s} dS(X)^J_{0,s} &= \int_0^T S(X)^{i_1,\dots,i_n}_{0,s} S(X)^{j_1,\dots,j_{k-1}}_{0,s} \, dX^{j_k}_s \\
&= \int_0^T \sum_{\sigma \in \mathrm{Sh}(n,k-1)} S(X)^{r_{\sigma(1)},\dots,r_{\sigma(n+k-1)}}_{0,s} \, dX^{j_k}_s \\
&= \sum_{\sigma \in \mathrm{Sh}(n,k-1)} S(X)^{r_{\sigma(1)},\dots,r_{\sigma(n+k-1)},r_{n+k}}_{0,s}. \qquad (**)
\end{aligned}$$

The second line is obtained by applying the induction hypothesis to $(r_1, \dots, r_{n+k-1}, r_{n+k}) = (i_1, \dots, i_n, j_1, \dots, j_{k-1}, j_k)$. The third line is then a consequence of swapping integration with summation and equation (2.4).

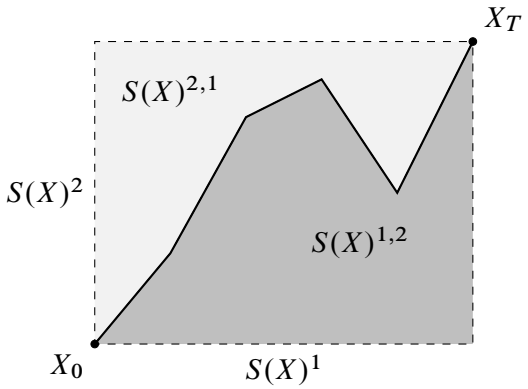


Figure 1: Geometric interpretation of Example 18.

By the same arguments, the second term of (*) is given by:

$$\begin{aligned}\int_0^T S(X)^J_{0,s} dS(X)^I_{0,s} &= \int_0^T S(X)^{i_1,\dots,i_{n-1}}_{0,s} S(X)^{j_1,\dots,j_k}_{0,s}\, dX^{i_n}_s \\ &= \int_0^T \sum_{\sigma\in \mathrm{Sh}(n-1,k)} S(X)^{l_{\sigma(1)},\dots,l_{\sigma(n+k-1)}}_{0,s}\, dX^{i_n}_s \\ &= \sum_{\sigma\in \mathrm{Sh}(n-1,k)} S(X)^{l_{\sigma(1)},\dots,l_{\sigma(n+k-1)},l_{n+k}}_{0,s}, \qquad (***)\end{aligned}$$

in which $(l_1,\dots,l_{n+k-1},l_{n+k}) = (i_1,\dots,i_{n-1},j_1,\dots,j_k,i_n)$. Finally, for $(q_1,\dots,q_{n+k}) = (i_1,\dots,i_n,j_1,\dots,j_k)$ the sum of (**) and (***) is in fact

$$S(X)^I_{0,T}\, S(X)^J_{0,T} = \sum_{\sigma\in \mathrm{Sh}(n,k)} S(X)^{q_{\sigma(1)},\dots,q_{\sigma(n+k)}}_{0,T} = \sum_{K\in I\sqcup J} S(X)^K_{0,T}.$$

This follows from the observation that the set of shuffles $I \sqcup J$ between indices I and J can be partitioned in two sets: one that contains all shuffle permutations in which j_k is in the last position, i.e., (**), and another that contains all shuffles in which i_n is in the last position, (***). This proves that $e^I e^J = (e^I \sqcup e^J)$ on the space of signatures. □

Example 18. For any $X \in \mathcal{V}([0,T],\mathbf{R}^d)$, we have

$$S(X)^i_{0,T}\, S(X)^j_{0,T} = S(X)^{i,j}_{0,T} + S(X)^{j,i}_{0,T}$$

for all $i,j \in \{1,\dots,d\}$. The geometric interpretation of this in the case $d = 2$ is illustrated in Figure 1. ⊘

2.5 Truncated signatures

In applications, it is usually not possible to work with full signatures as elements of the infinite dimensional $T((V))$ – machines can only support finite precision. A simple way to avoid this problem is to truncate the signature to some level N. This is similar to taking a truncated Taylor series when working with polynomials. Another benefit of working with the truncated signatures is that they live in a finite dimensional space. As a result, there is a canonical norm and we can speak of topological notions such as limits and convergence.

Definition 19. Let V be a finite dimensional vector space. Consider the ideal $I_N = \{ a \in T((V)) \mid a_0 = a_1 = \cdots = a_N = 0 \}$ of $T((V))$. The *truncated tensor algebra* of level $N \geq 1$ is defined to be the quotient algebra

$$T^N(V) = T((V))/I_N.$$

⊘

Remark 20. One can straightforwardly show that $T^N(V)$ is canonically isomorphic to $\bigoplus_{n=0}^{N} V^{\otimes n}$ along with truncated multiplication

$$(a_0, \ldots, a_N) \otimes (b_0, \ldots, b_N) = (c_0, \ldots, c_N)$$

where

$$c_n = \sum_{k=0}^{n} a_k \otimes b_{n-k}.$$

The multiplication operation simply ignores all terms of order larger than N. With this view, it is clear that the main properties of full signatures (invariance under reparametrization, Chen's identity, and preservation of inverses) hold for truncated signatures, as these were proven by induction. ⊘

By truncating $T((V))$, we obtain $T^N(V)$ which is a finite dimensional vector space isomorphic to $\mathbf{R}^{1+d+\cdots+d^N}$. This makes it easier to make sense of analytical properties such as convergence. Although we do not need to specify a specific norm to speak of convergence, as all norms are equivalent in finite dimensional vector spaces, in some cases an explicit norm allows for a specific bound.

Our aim next is to relate convergence in the path space $\mathcal{V}([0,T], \mathbf{R}^d)$ with convergence in the truncated tensor algebra $T((\mathbf{R}^d))$.

Theorem 21. *Let $(X_n)_{n \in \mathbf{N}}$ be a sequence of paths in $\mathcal{V}([0,T], \mathbf{R}^d)$ that satisfies the following conditions:*

(i) *there exists $X \in \mathcal{V}([0,T], \mathbf{R}^d)$ such that $X_n \to X$ uniformly;*

(ii) $\sup_{n \in \mathbf{N}} \|X_n\|_{1\text{-var}} < \infty$.

Then we have

$$\lim_{n \to \infty} S^N(X_n)_{0,T} = S^N(X)_{0,T}.$$

Proof. This is a consequence of the general result Proposition B6 for Riemann-Stieltjes integration. Using the proposition, we can show that all individual coefficients converge,

$$S(X_n)_{0,T}^{i_1,\dots,i_k} \longrightarrow S(X)_{0,T}^{i_1,\dots,i_k}.$$

This can be done by an induction to k. The theorem then follows by the fact that $T^N(\mathbf{R}^d) \simeq \mathbf{R}^{1+d+\cdots+d^N}$ is finite dimensional. □

Corollary 22. *Let $X \in \mathcal{V}([0,T], \mathbf{R}^d)$. Then there exists a sequence of piecewise-linear paths $(X_n)_{n\in\mathbf{N}}$ such that $X_n \to X$ uniformly and $S^N(X_n)_{0,T} \to S^N(X)_{0,T}$.*

Proof. It is known that any continuous function on a compact interval can be approximated by piecewise-linear functions. To do this, first pick a sequence of partitions $P_n = \{0 = t_0^n < \cdots < t_{p_n}^n = T\}$ with shrinking mesh size, $|P_n| \to 0$. Then define each X_n to be the path that is linear on the segments $[t_i^n, t_{i+1}^n]$ and for which $X_n(t_i^n) := X(t_i^n)$. It then follows that X_n converges to X uniformly. Furthermore, we have per definition of the total variation (i.e., the supremum over all partitions) that $\|X_n\|_{1\text{-var}} \leq \|X\|_{1\text{-var}} < \infty$ for all n. The result $S^N(X_n)_{0,T} \to S^N(X)_{0,T}$ now follows from the previous theorem. □

2.6 Log signatures

The truncated tensor algebra can still be a rather large space for large values of N. Indeed, as a vector space, $T^N(V)$ is isomorphic to $\mathbf{R}^{1+d+\cdots+d^N}$. Therefore, for practical purposes, a high truncation level comes with the cost of an explosion of the number of coefficients. Even so, one may note certain symmetries in the definition of the signature. For instance, for linear planar paths, $S^{12}(X)_{0,1} = -S^{21}(X)_{0,1}$. Our goal in this section is to present the method to compress the signature to the so-called log signature.

The outline of this section is as follows. First, we apply the theory of Lie groups and Lie algebras to connect two subsets of the truncated tensor algebra. Here we formulate the well-known Campell-Baker-Hausdorff formula, Theorem 26. Then, we construct a special Lie algebra $\mathfrak{g}^N(\mathbf{R}^d)$ and discuss its properties. The main result here is Theorem 30, which states that $\mathfrak{g}^N(\mathbf{R}^d)$ is a special Lie algebra, the free N-step nilpotent Lie algebra over $\mathbf{R}^d$. Finally, we define the log signature and show that it takes value in $\mathfrak{g}^N(\mathbf{R}^d)$ (Theorem 32). This allows us to use the properties of $\mathfrak{g}^N(\mathbf{R}^d)$ to give a powerful representation of the log signature.

Throughout this section, we make use of many results about Lie algebras. These results are summarized in Appendix C.

2.6.1 Preliminary spaces

Briefly recall the definitions:

- a Lie group G is a group that is also a manifold and for which the group multiplication and inverse are smooth;

- a Lie algebra $\mathfrak{g}$ is a vector space endowed with a bracket operation $(v, w) \mapsto [v, w]$ that is bilinear and satisfies the Jacobi identity,

$$[a, [b, c]] + [b, [c, a]] + [c, [a, b]] = 0.$$

Using the Lie bracket operation on vector fields on G, the tangent space $\mathcal{T}_e G$ of G at the identity can be turned into a canonical Lie algebra associated with G. More details can be found in Appendix C.

Now fix a truncation level $N \geq 1$ and consider the following affine-linear subspaces of $T^N(\mathbf{R}^d)$:

$$T_c^N(\mathbf{R}^d) := \left\{ (a_0, \ldots, a_N) \in T^N(\mathbf{R}^d) \;\middle|\; a_0 = c \in \mathbf{R} \right\}.$$

We are only concerned with the two spaces $T_0^N(\mathbf{R}^d)$ and $T_1^N(\mathbf{R}^d)$.

Proposition 23. *The following properties hold for $T_0^N(\mathbf{R}^d)$ and $T_1^N(\mathbf{R}^d)$:*

(i) *$T_1^N(\mathbf{R}^d)$ is a Lie group with multiplication $\otimes$ and unit $1 = (1, 0, \ldots, 0)$;*

(ii) *$T_0^N(\mathbf{R}^d)$ equipped with the bracket operation $[a, b] := a \otimes b - b \otimes a$ is a Lie algebra;*

(iii) *$T_0^N(\mathbf{R}^d)$ is the canonical Lie algebra of the Lie group $T_1^N(\mathbf{R}^d)$.*

Proof. **(i)**. First, we show that $\left(T_1^N(\mathbf{R}^d), \otimes\right)$ is a group. It is clear that $T_1^N(\mathbf{R}^d)$ is closed under $\otimes$. Furthermore, for an element $1 + a \in T_1^N(\mathbf{R}^d)$, a simple computation shows that

$$(1 + a)^{-1} = \sum_{n=0}^{N} (-1)^n a^{\otimes n}$$

is the left and right inverse of $1 + a$ with respect to $\otimes$. Hence, $T_1^N(\mathbf{R}^d)$ is a group. To see that $T_1^N(\mathbf{R}^d)$ is a smooth manifold, simply note that $T_1^N(\mathbf{R}^d)$ is an affine-linear subspace of $T^N(\mathbf{R}^d) \simeq \mathbf{R}^{1+d+\cdots+d^N}$; hence, $T_1^N(\mathbf{R}^d)$ is a submanifold that is diffeomorphic with $\mathbf{R}^{d+\cdots+d^N}$. The charts are induced by a choice of basis on $T^N(\mathbf{R}^d)$ and are global. Moreover, the multiplication map $\otimes$ and inverse map $1 + a \mapsto (1 + a)^{-1}$ are both polynomial functions when written in coordinates, hence they are smooth. This makes $T_1^N(\mathbf{R}^d)$ a Lie group.

(ii). Since $0 \in T_0^N(\mathbf{R}^d)$, it is a vector space. It is also clear that if $a, b \in T_0^N(\mathbf{R}^d)$, then $[a, b] = a \otimes b - b \otimes a$ is again an element of $T_0^N(\mathbf{R}^d)$. Finally, it is straightforward to verify that the bracket operation satisfies the Jacobi equation, this follows from the associativity and distributivity of $\otimes$.

(iii). First, note that $T_1^N(\mathbf{R}^d)$ is affine-linear and its tangent space at any point can be identified with $T_0^N(\mathbf{R}^d) \simeq \mathbf{R}^{d+\cdots+d^N}$. Hence, we only need to show that our custom-defined bracket $[a, b] = a \otimes b - b \otimes a$ is the same as the bracket

obtained from the Lie bracket of the Lie group $T_1^N(\mathbf{R}^d)$. Denote left translation by g by $L_g : h \mapsto g \otimes h$. For $a, b \in T_0^N(\mathbf{R}^d)$, the left-invariant vector field $\vec{a} := g \mapsto (dL_g)_e(v)$ is simply given by

$$\vec{a}_g = g \otimes a \in T_0^N(\mathbf{R}^d) \simeq \mathcal{T}_g\left(T_1^N(\mathbf{R}^d)\right), \quad \text{for all } g \in T_1^N(\mathbf{R}^d).$$

This follows from linearity of the map L_g. Since this vector field is linear, the derivative of $\vec{a}$ is simply itself. Hence, we can easily compute the Lie bracket between the vector fields $\vec{a}$ and $\vec{b}$,

$$\begin{aligned}[\vec{a}, \vec{b}]_1 &= \left(D\vec{b}(\vec{a}) - D\vec{a}(\vec{b})\right)|_1 \\ &= D\vec{b}(a) - D\vec{a}(b) \\ &= a \otimes b - b \otimes a = [a, b].\end{aligned}$$

This concludes the proof. □

There are two very natural maps between $T_0^N(\mathbf{R}^d)$ and $T_1^N(\mathbf{R}^d)$, the exponential and logarithmic map:

$$\exp : T_0^N(\mathbf{R}^d) \to T_1^N(\mathbf{R}^d), \quad a \mapsto \sum_{n=0}^{N} \frac{a^{\otimes n}}{n!},$$

$$\log : T_1^N(\mathbf{R}^d) \to T_0^N(\mathbf{R}^d), \quad 1 + a \mapsto \sum_{n=0}^{N} (-1)^{n+1} \frac{a^{\otimes n}}{n}.$$

Lemma 24. *The maps* exp *and* log *are bijective and each other's inverses. Furthermore, both maps are smooth.*

Proof. It is straightforward algebra to verify that $\exp \circ \log$ and $\log \circ \exp$ are the identity maps in their corresponding spaces. If written in coordinates, both maps are polynomial functions, thus smooth. □

Example 25. Set $N = 2$ and let $a, b \in T_0^N(\mathbf{R}^d)$. Then

$$\begin{aligned}\exp(a) \otimes \exp(b) &= \left(1 + a + \frac{1}{2}a^{\otimes 2}\right) \otimes \left(1 + b + \frac{1}{2}b^{\otimes 2}\right) \\ &= 1 + a + b + a \otimes b + \frac{1}{2}a^{\otimes 2} + \frac{1}{2}b^{\otimes 2} \\ &= 1 + a + b + \frac{1}{2}[a, b] + \frac{1}{2}(a + b)^{\otimes 2} \\ &= \exp\left(a + b + \frac{1}{2}[a, b]\right).\end{aligned}$$

⊘

The above example shows in particular that $\exp(a) \otimes \exp(b) \neq \exp(a + b)$. In general, the expansion of $\log(e^a e^b)$ is given by the Campell-Baker-Hausdorff formula.

Theorem 26 (Campell-Baker-Hausdorff)**.** *For any* $a, b \in T_0^N(\mathbf{R}^d)$, $\log[\exp(a) \otimes \exp(b)]$ *can be written as a finite linear combination of nested right-brackets of* a *and* b.

Proof. This is a consequence of the general Campell-Baker-Hausdorff theorem for Lie algebras, see Theorem C5. This theorem only establishes the result for a, b in a neighborhood U of the zero element $0 \in T_0^N(\mathbf{R}^d)$. Careful inspection of the proof shows that U is chosen as a neighborhood of 0 for which the exponential map exp is a diffeomorphism. Since we already know that exp is a global diffeomorphism (Lemma 24) the statement holds globally for $a, b \in U = T_0^N(\mathbf{R}^d)$. □

2.6.2 The free nilpotent Lie algebra $\mathfrak{g}^N(\mathbf{R}^d)$

Next, we introduce one of the main spaces of interest. As we will see below, this is the space in which the log signature lives.

Definition 27. Define $\mathfrak{g}^N(\mathbf{R}^d) \subset T_0^N(\mathbf{R}^d)$ to be the smallest Lie subalgebra that contains $\mathbf{R}^d$. ⊘

The following lemma gives an useful expression for $\mathfrak{g}^N(\mathbf{R}^d)$.

Lemma 28. *Consider the lower central series of vector spaces*

$$L_1 := \mathbf{R}^d \quad \textit{and} \quad L_{n+1} := [\mathbf{R}^d, L_n],$$

in which $[V, W]$ *means the vector span of the set* $\{[v, w] \mid v \in V,\ w \in W\}$. *Then* $\mathfrak{g}^N(\mathbf{R}^d)$ *is the direct sum of vector subspaces* L_n*,*

$$\mathfrak{g}^N(\mathbf{R}^d) = \bigoplus_{n=0}^{N} L_n.$$

Proof. This is a direct consequence of Proposition C10 for general Lie algebras. In our case, each L_n only has the zero element in common with L_k if $n \neq k$, due to the tensor grading and the bracket being defined in terms of tensor products, so the right-hand side is indeed a direct sum of vector subspaces of $T_0^N(\mathbf{R}^d)$. □

The following important corollary of the Campell-Baker-Hausdorff formula shows that $\mathfrak{g}^N(\mathbf{R}^d)$ is closed under exp and log.

Corollary 29. *If* $a, b \in \mathfrak{g}^N(\mathbf{R}^d)$*, then* $\log[\exp(a) \otimes \exp(b)] \in \mathfrak{g}^N(\mathbf{R}^d)$.

Proof. Theorem 26 tells us that $c = \log[\exp(a) \otimes \exp(b)]$ is a finite linear combination of nested right-brackets in a and b. The previous lemma then shows that $c \in \mathfrak{g}^N(\mathbf{R}^d)$. □

The following theorem establishes that $\mathfrak{g}^N(\mathbf{R}^d)$ is not just any Lie algebra, but in fact a type of 'universal' Lie algebra. Such Lie algebra is called a *free* Lie algebra, which can be interpreted as the 'freest' Lie algebra, in the sense that besides the Jacobi identity, no further restrictions have been imposed on the bracket. We refer to Definition C7 for the definition. Many algebraic structures have a similar notion of a universal free object, e.g., free groups, free vector spaces or free associative algebras. In many cases, such free objects are unique up to their respective morphisms. This is also true for Lie algebras: up to *Lie algebra homomorphisms*, the free Lie algebra over a fixed set is unique.

Theorem 30. *The Lie algebra $\mathfrak{g}^N(\mathbf{R}^d)$ is the free, N-step nilpotent Lie algebra over V.*

Proof. Consider the tensor algebra

$$T(V) := \bigoplus_{n=0}^{\infty} V^{\otimes n},$$

endowed with the product $\otimes$ and bracket $[a, b] = a \otimes b - b \otimes a$. Let $\mathcal{L}(V)$ be the smallest Lie subalgebra of $T(V)$ that contains V. By Theorem C8, $\mathcal{L}(V)$ is the free Lie algebra over V. Similar to the previous lemma, we have by Proposition C10 the expression

$$\mathcal{L}(V) = \bigoplus_{n=0}^{\infty} L_n.$$

Thus, for the ideal $I_N := \{(l_1, l_2, \dots) \in \mathcal{L}(V) \mid l_1 = \cdots = l_N = 0\}$, we have that the quotient $\mathcal{L}(V)/I_{N+1}$ is per definition the free N-step nilpotent Lie algebra. The proof is complete by noting that $\mathcal{L}(V)/I_{N+1}$ is the same as $\mathfrak{g}^N(\mathbf{R}^d)$. □

2.6.3 Connection to signatures

Next, we establish the connection between $\mathfrak{g}^N(\mathbf{R}^d)$ and the range of the truncated signature map S^N.

Theorem 31 (Chow-Rashevskii)**.** *For all $g \in \exp \mathfrak{g}^N(\mathbf{R}^d)$, there exists vectors $v_1, \dots, v_m$ in $\mathbf{R}^d$, such that*

$$g = \exp(v_1) \otimes \cdots \otimes \exp(v_m).$$

In view of Chen's identity, any $g \in \exp \mathfrak{g}^N(\mathbf{R}^d)$ is the signature of some piecewise-linear path.

Proof. We begin with the claim that a nested bracket of a certain degree can be achieved.

Claim 1. Let $1 \leq n \leq N$. For all $l_n \in L_n$, there exists some $R_{n+1} \in \bigoplus_{k=n+1}^{N} L_k$ such that $\exp(l_n + R_{n+1})$ can be written as a product of exponentials of vectors in $\mathbf{R}^d$.

Proof. Let $l_n = [v_1, \dots, v_n] \in L_n$ for vectors $v_1, \dots, v_n \in \mathbf{R}^d$. Define for $1 \le k \le n$ and $t \in \{-1, +1\}$,

$$g_t^1 = \exp(t v_1), \quad g_t^{k+1} = g_t^k \exp(t v_k) g_{-t}^k \exp(-t v_k).$$

A straightforward induction argument shows that

$$g_1^n = \exp([v_1, \dots, v_n] + R_{n+1}),$$

with R_{n+1} consisting of higher order terms. □

Next, for $1 \le n \le N$, we prove that the exponential of any $a \in \bigoplus_{k=n}^N L_k \subset \mathfrak{g}^N(\mathbf{R}^d)$ can be written as a product of exponentials. The idea is to use the claim to iteratively 'correct' brackets of increasing orders. We proceed by backwards induction on n. The base case $n = N$ follows immediately by the claim due to nilpotency: R_{N+1} must be zero. Suppose the statement is true for any $2 \le n \le N$. We show that the result extends to $n - 1$.

Decompose $a = l_{n-1} + R_n$ with $l_{n-1} \in L_{n-1}$ and $R_n \in \bigoplus_{k=n}^N L_k$. By the claim, there is $R_n' \in \bigoplus_{k=n}^N L_k$ such that $\exp(-l_{n-1} + R_n')$ can be written as a product of exponentials. Applying the CBH formula, we find

$$\begin{aligned} \exp\left(R_n''\right) &:= \exp(l_{n-1} + R_n) \otimes \exp(-l_{n-1} + R_n') \\ &= \exp\left(R_n + R_n' + \frac{1}{2}[l_{n-1} + R_n, -l_{n-1} + R_n'] + \dots\right) \end{aligned}$$

Note that $[l_{n-1}, -l_{n-1}] = 0$ so the bracketed terms in the above expression are at least of order $2n - 1 \ge n + 1$. Therefore, by the induction hypothesis, $\exp\left(R_n''\right)$ can be written as a product of exponentials. Thus, by right multiplying the previous equation with $\exp(-l_{n-1} + R_n')^{-1} = \exp(l_{n-1} - R_n')$, we find

$$\exp(a) = \exp(l_{n-1} + R_n) = \exp(R_n'') \otimes \exp(l_{n-1} - R_n'),$$

which is a product of exponentials. This finishes the induction step and the theorem follows by the case $n = 1$. □

Let us denote the range of the truncated signature by $G^N(\mathbf{R}^d)$, $G^N(\mathbf{R}^d) = S^N \circ \mathcal{V}([0, T], \mathbf{R}^d)$. The previous theorem asserts that $\exp \mathfrak{g}^N(\mathbf{R}^d) \subset G^N(\mathbf{R}^d)$. In the next theorem, we prove that this is an equality.

Theorem 32 ([9, Thm 7.30])**.** *We have the equality* $\exp \mathfrak{g}^N(\mathbf{R}^d) = G^N(\mathbf{R}^d)$.

Proof. For the inclusion $\subset$, let $g \in \exp \mathfrak{g}^N(\mathbf{R}^d)$. By Theorem 31, there exists vectors $v_1, \dots, v_m \in \mathbf{R}^d$ such that $g = \exp(v_1) \otimes \cdots \otimes \exp(v_m)$. Take X to be the piecewise-linear path that goes from 0 to v_1 to $v_1 + v_2$ and so on to $v_1 + \cdots + v_m$. Then in view of Example 10, $S^N(X)_{0,T} = g$. This proves that $g \in G^N(\mathbf{R}^d)$.

Conversely, let $X \in \mathcal{V}([0, T], \mathbf{R}^d)$. Then in view of Corollary 22, there exists a sequence of piecewise-linear paths $(X_n)_n$ such that $X_n \to X$ uniformly and

$S^N(X_n)_{0,T} \to S^N(X)_{0,T}$. Because X_n is piecewise-linear, we have $S^N(X_n)_{0,T} \in \exp \mathfrak{g}^N(\mathbf{R}^d)$; hence, the convergence $S^N(X_n)_{0,T} \to S^N(X)_{0,T}$ implies that $S^N(X)_{0,T}$ is in the closure of $\exp \mathfrak{g}^N(\mathbf{R}^d)$ in $T^N(\mathbf{R}^d)$. If we can show that $\exp \mathfrak{g}^N(\mathbf{R}^d)$ is closed in $T^N(\mathbf{R}^d)$, then we are done. For this, note that $\mathfrak{g}^N(\mathbf{R}^d)$ is a linear subspace of $T^N(\mathbf{R}^d)$ which is finite dimensional, so it is closed. Therefore, $\exp \mathfrak{g}^N(\mathbf{R}^d)$ being the image of $\mathfrak{g}^N(\mathbf{R}^d)$ under the smooth embedding $\exp : T_0^N(\mathbf{R}^d) \to T^N(\mathbf{R}^d)$ is also closed. This concludes the proof. □

We are now able to give the proper definition of the log signature.

Definition 33 (Log signature)**.** The level N log signature is defined as the composition map $\log \circ S^N : \mathcal{V}([0,T], \mathbf{R}^d) \to \mathfrak{g}^N(\mathbf{R}^d)$. ⊘

Example 34. Consider for $N = 2$ the linear path $X : [0,1] \to \mathbf{R}^d \ t \mapsto tv$. Then the signature $S(X)_{0,1} = \exp(v)$ so $\log S(X)_{0,1} = v$. ⊘

The power of this definition lies in the space $\mathfrak{g}^N(\mathbf{R}^d)$. As we have shown, $\mathfrak{g}^N(\mathbf{R}^d)$ is the N-step nilpotent free Lie algebra over $\mathbf{R}^d$. While Lemma 28 tells us that the set of all nested right-brackets up to degree N span $\mathfrak{g}^N(\mathbf{R}^d)$, this set is not linearly independent so it does not form a basis. This is because the Jacobi identity introduces dependencies between the brackets. In fact, there is no canonical basis for $\mathfrak{g}^N(\mathbf{R}^d)$, however, there are procedures to construct one. A common choice is the Hall basis, see [10, p. 907].

Example 35 (Hall basis)**.** Let $N = 4$ and $d = 2$, and suppose $\{e_1, e_2\}$ is a basis for $\mathbf{R}^2$. Then the following elements form a basis for $\mathfrak{g}^N(\mathbf{R}^d)$:

$$
\begin{aligned}
&e_1, e_2 \\
&[e_1, e_2] \\
&[e_1, [e_1, e_2]],\ [e_2, [e_1, e_2]] \\
&[e_1, [e_1, [e_1, e_2]]],\ [e_2, [e_1, [e_1, e_2]]],\ [e_2, [e_2, [e_1, e_2]]]
\end{aligned}
$$

These have been computed by the web application on the webpage of [11]. ⊘

The dimension of $\mathfrak{g}^N(\mathbf{R}^d)$ can be calculated using Witt's formula,

$$\dim \mathfrak{g}^N(\mathbf{R}^d) = \frac{1}{N} \sum_{q|N} \mu(q) d^{N/q}$$

Here, the sum is over all divisors q of N and μ is the Möbius function. A proof of this can be found in [12]. Several values of $\dim \mathfrak{g}^N(\mathbf{R}^d)$ are listed in Table 1, see also [13].

2.7 Further reading and comments

Section 2.1 We defined the formal series of tensors $T((V))$ for finite dimensional vector spaces V, similar to [8, 2.2.1]. In [8], the approach is slightly more general, only requiring V to be a Banach space.

Table 1: Computed dimensions of $\mathfrak{g}^N(\mathbf{R}^d)$ for several d and N.

	$d=2$	$d=3$	$d=4$
$N=1$	2	3	4
$N=2$	3	6	10
$N=3$	5	14	30
$N=4$	8	32	90
$N=5$	14	80	294
$N=6$	23	196	964
$N=7$	41	508	3304
$N=8$	71	1318	11464
$N=9$	127	3502	40584
$N=10$	226	9382	145338

Section 2.2 The signature map is defined for continuous paths $X : [0,T] \to \mathbf{R}^d$ of bounded variation. We did this by defining the signature in coordinates; that is, we chose a basis for $\mathbf{R}^d$ and defined each component $S(X)_{0,T}^{i_1,\dots,i_n}$ as an iterated 1-dimensional Riemann-Stieltjes integral.

There are two possible generalizations here.

(i) One can define signatures for Banach space valued paths $X : [0,T] \to V$ by generalizing Riemann-Stieltjes integral to linear maps. By giving a more general definition of Riemann-Stieltjes integration, it is possible to describe multidimensional integrals in a basis independent manner.

(ii) Instead of only considering continuous paths of finite 1-variation, a generalization of the Riemann-Stieltjes integral called the *Young integral* allows one to make sense of integrals $\int_0^T Y_s \, dX_s$ in which X and Y respectively have finite p and q-variation, such that $1/p + 1/q > 1$. Using this type of integral, it is possible to extend the definition of signatures for paths of finite p-variation with $p \in [1,2)$.

Both these generalizations are implemented in [8], which rests on the the existence of the Young integral, see [8, Thm 1.16].

Section 2.3 Proofs in this section (notably Chen's identity and Proposition 11) were simple arguments based on induction, which appear in the original paper by Chen, [3]. Analytical proofs are available through the theory controlled ODEs, see [8, Cor 2.13] and [8, Prop 2.14].

The nontrivial uniqueness result (Theorem 13) for signatures is proven in [6] for continuous paths of bounded variation. More recently, this result has been extended to the setting of weakly geometric rough paths on Banach spaces in [14].

Section 2.4 This section showed that the space of linear functionals on the range of the signature map is an algebra of functions, following the exposition of [8]. The proof of the main result Theorem 17 is found in [15, Lemma 1.7]. The notes [15] also provide an insightful unified view of the range of the signature and its dual in the framework of Hopf algebras.

Section 2.5 The truncated tensor algebra $T^N(V)$ is defined as a quotient of the extended tensor algebra $T((V))$. A more comprehensive overview of analytical properties such as convergence is further elaborated in [9, Chapter 1].

Section 2.6 In this section the log signature was defined on the free nilpotent Lie algebra $\mathfrak{g}^N(\mathbf{R}^d)$. The exposition is mainly based on that of [9, Chapter 7], however, we have tried to include more precise reference material on free Lie algebras. We also gave an alternative proof of Theorem 31 based on an induction argument, rather than an ODE argument. A rough sketch of this can be found in [15, Thm 5.5].

A natural extension of the theory we have left out is its connection to sub-Riemannian geometry. Due to $G^N(\mathbf{R}^d) = \exp \mathfrak{g}^N(\mathbf{R}^d)$ being the range of the signature map, one can define a so-called Carnot-Carathéodory metric on the manifold $G^N(\mathbf{R}^d)$. Hence, Theorem 31 is a special case of the more general Chow-Rashevskii theorem for sub-Riemannian manifolds. Rich results from this point of view are further explored in [9, Chapter 7].

3 Signatures in Machine Learning

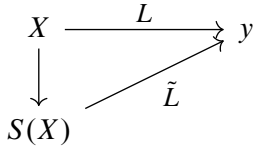
In machine learning, one is typically interested in the modelling of an unknown function based on various known input-output pairs. In supervised learning, the goal is to build a mathematical model using training data: we give the algorithm a set of inputs (also known as features) with corresponding desired outputs and let it 'learn' how inputs should match with outputs. Once the training stage is finished, the model can be used on out-of-sample data. A typical example of this is training a model to classify the object in an image. The training data in this case consists of a set of images (input) and the corresponding labels of the image (output), e.g., dog or cat. The training data is usually crafted by hand or for certain problems, historical data can be used.

Within supervised learning, there are two types of tasks, classification and regression. The difference is simple: classification tasks are problems in which the output variables only take a discrete number of values, while regression tasks have output values that can span an entire interval. For instance, a typical regression problem would be to predict the value of a house given several known features, e.g., living space, age, neighborhood popularity etc.. Problems like these have been well-studied in a statistical context. As such, machine learning algorithms typically employ statistical techniques such as ordinary least squares, Bayesian methods, as well as nonparametric methods.

The focus of the current section is to present how signatures can be applied in supervised learning problems. The main idea is that the signature is able to give a powerful representation of data. The workflow is simple: in a typical machine learning problem, the aim is to find a function L that maps each input X to the correct output y,

$$X \xrightarrow{\quad L \quad} y$$

With the signature method, we first transform the inputs into paths, then we compute the (truncated) signatures, and finally we do learning on the space of signatures. In other words, we learn the function $\tilde{L}$ in the following diagram:



The reason that this approach is effective is due to the following properties of the signature:

(i) (almost) one-to-one correspondence between paths and signatures;

(ii) uniform approximation;

(iii) insensitivity to sampling rate.

The first property is the result of Theorem 13. The theorem asserts that up to certain null paths, a path will be uniquely determined by its signature. This means that if we find a suitable model $\tilde{L}$ on the signature space $S \circ \mathcal{V}([0,T], \mathbf{R}^d)$, then this should be a good substitute for the original model L. Most paths one deals with in practice are not tree-like, and therefore have unique signatures. If uniqueness must be guaranteed, one can simply embed the path in a higher dimension by including the time parameter. Other embeddings are discussed in the next section below.

The second property is derived from Theorem 17, which states that the product of two linear functions on the range of the full signature is again a linear function. Hence, if we were to take the truncation level sufficiently high, any polynomial function on $S^N \circ \mathcal{V}([0,T], \mathbf{R}^d)$ becomes linear. The intuitive implication of this result for machine learning is as follows: if there is a nonlinear relationship between the input X and the output y, that is, $y = L(X)$ for some nonlinear function L, then we expect that the corresponding map $\tilde{L}$ that sends $S(X)$ to y to be a linear map. Theoretically, this would mean that when doing regression on the signature space, it is sufficient to consider only linear algorithms.

The third property is the result of Theorem 21 and Corollary 22, which state that the (truncated) signature of piecewise-linear approximations of a path will converge to the signature of the path when decreasing the mesh size. Therefore, if our discrete data stream is obtained by sampling some path, then the sampling frequency will not have a large impact on the signature we compute. For instance, if we are doing character recognition and our data stream is a handwritten digit represented as a path in $\mathbf{R}^2$ indexed by time which is obtained by recording the location of the pen every n seconds, then the signature should be (to a certain degree) insensitive to n. This enables the signature to be used for dimension reduction, since the number of coefficients in the signature is only dependent on the dimension of the path and the truncation level, and not on the number of points or length of a path.

3.1 Paths from discrete data

In applications, the sequential data one obtains is often discrete. For instance, in financial applications, the price of a stock or bond is often sampled on a fixed interval, say daily or every minute. We discuss three common ways [16] to construct a continuous path from discrete data, with particular consideration for the signature.

Piecewise-linear interpolation

The easiest method to obtain continuous paths from discrete data points is through piecewise-linear interpolation. Let $(X_{t_i})_{i=0}^m$ with $0 = t_0 < t_1 < \cdots < t_m = T$ be a stream of data in $\mathbf{R}^d$. Then the the continuous path $\tilde{X} : [0,T] \to \mathbf{R}^d$ obtained by piecewise-linear interpolation is simply the path that is linear on the segments $[t_i, t_{i+1}]$ and satisfies $\tilde{X}_{t_i} = X_{t_i}$, see Figure 2. Piecewise-linear interpolation serves as the basis to convert discrete paths into continuous paths. The next two methods we discuss both rely on piecewise-linear interpolation as a final step to

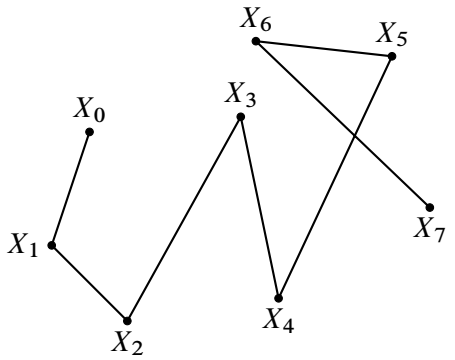


Figure 2: Piecewise-linear interpolation of 7 (sequentially ordered) data points in $\mathbf{R}^2$.

obtain a continuous path. Therefore, whenever we speak of the signature of a discrete path, we mean the signature of its piecewise-linear interpolation.

Time-based transformations

Recall that the signature of $\mathbf{R}$-valued paths is simply a countable sequence of real numbers, because any tensor power of $\mathbf{R}$ is isomorphic to $\mathbf{R}$ itself. Furthermore, since an $\mathbf{R}$-valued path can only go in two directions, Proposition 11 tells us that the signature only depends on the total displacement of the path. As such, signatures of $\mathbf{R}$-valued paths are trivial objects that do not carry much information. Hence, when dealing with univariate discrete data, it is beneficial to consider transformations that embed the data in a second dimension.

The easiest way to embed a path in a higher dimension is to include the time parametrization, which we will call the *time-embedded* or *time-indexed* transformation. For a continuous path X, this simply means the path $t \mapsto (t, X_t)$. Similarly, for discrete $(X_{t_i})_{i=0}^m$ series, the time-indexed version is the path $(t_i, X_{t_i})_{i=0}^m$. Sometimes there is no information about the time parametrization, for instance if we have an ordered array of numbers. In such cases we use the index as time, i.e., $(i, X_i)_{i=0}^m$.

A similar transformation is the *time-joined* transformation, which adds extra intermediate points.

Definition 36. Let $(X_{t_i})_{i=0}^m$ be a discrete univariate series in $\mathbf{R}$. We define the time-joined transformation of X as the stream $\left(X_j^{\text{time-joined}}\right)_{j=0}^{2m+1}$ by

$$X_j^{\text{time-joined}} = \begin{cases} (t_0, 0) & \text{if } j = 0, \\ (t_i, X_{t_{i-1}}) & \text{if } j = 2i, \\ (t_i, X_{t_i}) & \text{if } j = 2i + 1. \end{cases}$$

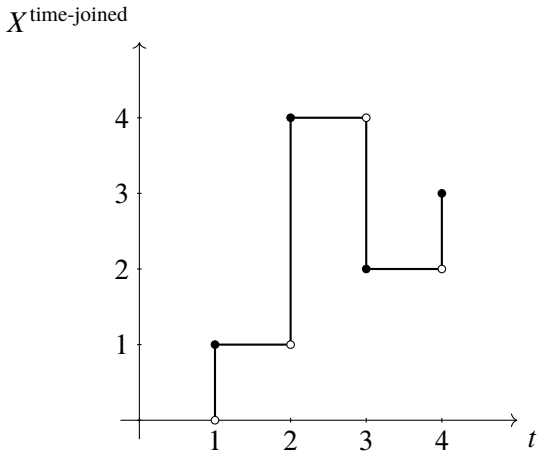


Figure 3: Time-joined transformation of $\{(t_i, X_{t_i})\}_{i=0}^{3} = \{(1,1),(2,4),(3,2),(4,3)\}$. The closed circles are points in the original path; the open circles indicate added points.

By the signature of $X^{\text{time-joined}}$, we mean the signature of the piecewise-linear interpolation of $X^{\text{time-joined}}$. ⊘

Example 37. Suppose $X = (X_{t_0}, \ldots, X_{t_4})$, then

$$X^{\text{time-joined}} = \{(t_0, 0), (t_0, X_{t_0}), (t_1, X_{t_0}), (t_1, X_{t_1}), \ldots, (t_4, X_{t_4})\}.$$

This is illustrated in Figure 3. ⊘

The time-joined transformation has been proposed in [17] for studying classical time series. This is due to two main properties of the signature of the transformation:

- the signature of a time-joined path uniquely determines the original (discrete) time series;
- the values $(X_{t_i})_{i=0}^{m}$ can be represented as a linear functional on the signature of $X^{\text{time-joined}}$.

Lead-lag transformation

The lead-lag transformation is a method to embed an $\mathbf{R}^d$-valued path into an $\mathbf{R}^{2d}$-valued path. We give the following definition for discrete paths.

Definition 38. For a stream of data $(X_{t_i})_{i=0}^{m}$ in $\mathbf{R}^d$, we define the streams $\left(X_j^{\text{lead}}\right)_{j=0}^{2m}$ and $\left(X_j^{\text{lag}}\right)_{j=0}^{2m}$ in $\mathbf{R}^d$ by

$$X_j^{\text{lead}} = \begin{cases} X_{t_i} & \text{if } j = 2i, \\ X_{t_i} & \text{if } j = 2i-1, \end{cases} \quad \text{and} \quad X_j^{\text{lag}} = \begin{cases} X_{t_i} & \text{if } j = 2i, \\ X_{t_i} & \text{if } j = 2i+1. \end{cases}$$

The lead-lag transformation of X is the $2d$-dimensional stream

$$\left(X_j^{\text{lead-lag}}\right)_{j=0}^{2m} = \left(X_j^{\text{lead}}, X_j^{\text{lag}}\right)_{j=0}^{2m}.$$

The signature of $X^{\text{lead-lag}}$ is defined as the signature of the piecewise-linear interpolation of $X^{\text{lead-lag}}$. ⊘

Remark 39. The information about the time parametrization of the discrete path $(X_{t_i})_{i=0}^m$ is lost after taking the lead-lag transformation, as $X^{\text{lead-lag}}$ is generically indexed by j from 0 to $2m$. Only the ordering of the data is preserved. If the time parametrization is important for the analysis at hand, a different transformation such as the time-joined transformation can be used. ⊘

Example 40. Consider a univariate series $X = \{a, b, c, d\}$ (time parametrization does not matter per the above remark). Then

$$X^{\text{lead}} = \{a, b, b, c, c, d, d\} \quad \text{and} \quad X^{\text{lag}} = \{a, a, b, b, c, c, d\},$$

so

$$X^{\text{lead-lag}} = \{(a, a), (b, a), (b, b), (c, b), (c, c), (d, c), (d, d)\}.$$

In Figure 4 the piecewise-linear interpolation of a lead-lag transformation is plotted. ⊘

The lead-lag transformation was originally proposed in [18] for analyzing time series obtained by sampling from continuous semi-martingales. This is due to the following key property: the signature of the lead-lag transformation of a path contains direct information about the quadratic variation of the path.

Proposition 41. *We have*

$$S\left(X^{\textit{lead-lag}}\right)_{0,T}^{[1,2]} = S\left(X^{\textit{lead-lag}}\right)_{0,T}^{1,2} - S\left(X^{\textit{lead-lag}}\right)_{0,T}^{2,1} = \sum_{i=0}^{m-1} \left(X_{t_{i+1}} - X_{t_i}\right)^2.$$

Proof. This can be proven by induction to m, the length of the data stream. For $m = 0$ both sides are zero. Assuming the statement holds for $m \geq 0$, the result for $m + 1$ can be obtained by applying the induction step to X without the last element, and appealing to Chen's identity. Indeed, if $X = Y * \{X_{t_{m+1}}\}$, then

$$X^{\text{lead-lag}} = Y^{\text{lead-lag}} * \{(X_{t_{m+1}}, X_{t_m}), (X_{t_{m+1}}, X_{t_{m+1}})\}.$$

Applying Chen's identity to the piecewise-linear path $X^{\text{lead-lag}}$ yields

$$S\left(X^{\text{lead-lag}}\right)_{0,T} = S\left(Y^{\text{lead-lag}}\right)_{0,T} \otimes \exp((x, 0)) \otimes \exp((0, x)),$$

with $x = X_{t_{m+1}} - X_{t_m}$. Working this out we find

$$S\left(X^{\text{lead-lag}}\right)_{0,T}^{[1,2]} = S\left(Y^{\text{lead-lag}}\right)_{0,T}^{[1,2]} + x^2,$$

which completes the induction step. □

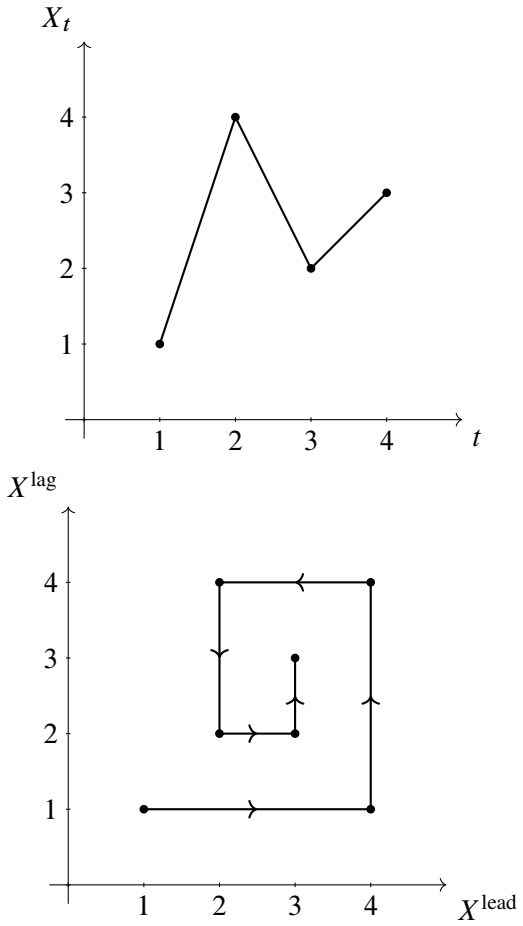


Figure 4: Lead-lag transformation of $(X_i)_{i=1}^4 = \{1, 4, 2, 3\}$.

Although it is possible to use the lead-lag transformation for all components in a multidimensional time series, it is in practice more efficient to only perform lead-lag transformations for those individual components for which the quadratic variation is relevant. For instance, [19] studied high-frequency order book data which included univariate series for trade volume, bid-ask prices, trade prices, and a few other derived quantities. Since the quadratic variation of the price process is an important notion for models in mathematical finance, the authors only performed a lead-lag transformation on the price component of the multivariate series.[2] A partial lead-lag transformation therefore avoids unnecessary 'blowups' of features when computing signatures, as the number of components in signatures scales exponentially with the dimension of the path.

Summary of various transformations

We briefly summarize the advantages of each transformation with respect to the signature.

- The (vanilla) piecewise-linear interpolation is applicable when dealing with discrete data of dimension 2 or higher and when the time parametrization of the path is irrelevant.
- For univariate discrete data, it is necessary to embed the path in a higher dimension before using piecewise-linear interpolation – the signature is trivial for $\mathbf{R}$-valued paths that are piecewise-linear (see Proposition 11). The time-joined, time-embedded and lead-lag transformations can be used for this purpose.
- The time-joined and time-embedded transformations are suitable when preservation of the time parametrization is required. Time-joined has been used for analysis of classical time series due to a (linear) relationship between the signature and the lagged values of the time series [17].
- The lead-lag transformation does not preserve the time parametrization. Signatures of lead-lag transformed paths contain direct information about the quadratic variation of the path. This transformation has been studied in the context of discrete approximations to continuous processes [18].

3.2 Computational aspects

In this section we give brief sketches on the algorithms to compute truncated signatures and log signatures of piecewise-linear interpolations of discrete paths.

[2] To be more precise, the authors used the 'lead' part of each component of the time series and only included the 'lag' part for the price process; this is done in order to ensure that all series are of the same length.

3.2.1 Signatures

Let $(X_{t_i})_{i=0}^{m}$ be a discrete path in $\mathbf{R}^d$. In Example 10, we computed the full signature of such path. Under the current notation, the formula reads,

$$S(X)_{0,T} = \bigotimes_{i=0}^{m-1} \exp\left(X_{t_{i+1}} - X_{t_i}\right). \tag{3.1}$$

The number of operations required to compute the truncated signature of level N using the above formula is around the order $O(md^N)$. Of course, there are certain symmetries in the signature that allows one to reduce this. For instance, computing the signature of a line segment involves computing tensor powers of vectors in succession for the exponential. One could reduce the number of multiplications required by noting that $v^{\otimes n}$ is symmetric and only has $\binom{d+n-1}{n}$ distinct values.

There are currently two open-source software libraries that supply signature computations. The first one is part of the *CoRoPa* (Computational Rough Paths) [11] research program, which contains the Python package `ESig`. A newer implementation is the `iisignature` Python package [20]. Both libraries provide functions for computing log signatures and regular signatures and are implemented in C++. For our numerical experiments in Section 3.3, we use the `iisignature` package due to its superior benchmarks reported in [20].

It should be noted that the only two available open-source implementations are for the Python programming language. If one is constrained to use a different language, a custom solution must be supplied. However, the algorithm to compute signatures from linearly interpolated paths is quite straightforward to implement using formula (3.1). Many programming languages already have high-performance linear algebra libraries that contain functions for tensor manipulation. Specifically, most libraries include a routine that allows one to compute the tensor product of two arbitrary rank k and l tensors, i.e., $v \otimes w$ for $v \in (\mathbf{R}^d)^{\otimes k}$ and $w \in (\mathbf{R}^d)^{\otimes l}$. As such, the task of implementing the signature mainly consists of

- designing a data structure to represent elements of the truncated tensor algebra $T^N(\mathbf{R}^d)$;
- writing a routine that extends the usual tensor product to the tensor product of the truncated tensor algebra $T^N(\mathbf{R}^d)$;
- writing a routine that implements the (truncated) exponential function.

Our experience with this process comes from building a novel interactive JavaScript web application to visualize the signature of a drawn path.[3] The approach taken here is to simply represent the truncated tensor algebra as a list of tensors, e.g.,

```
[1, [1,2], [[3,4],[5,6]], [[[7,8],[9,10]],[[11,12],[13,14]]]]
```

[3] `https://zhy0.com/signature-visualizer`

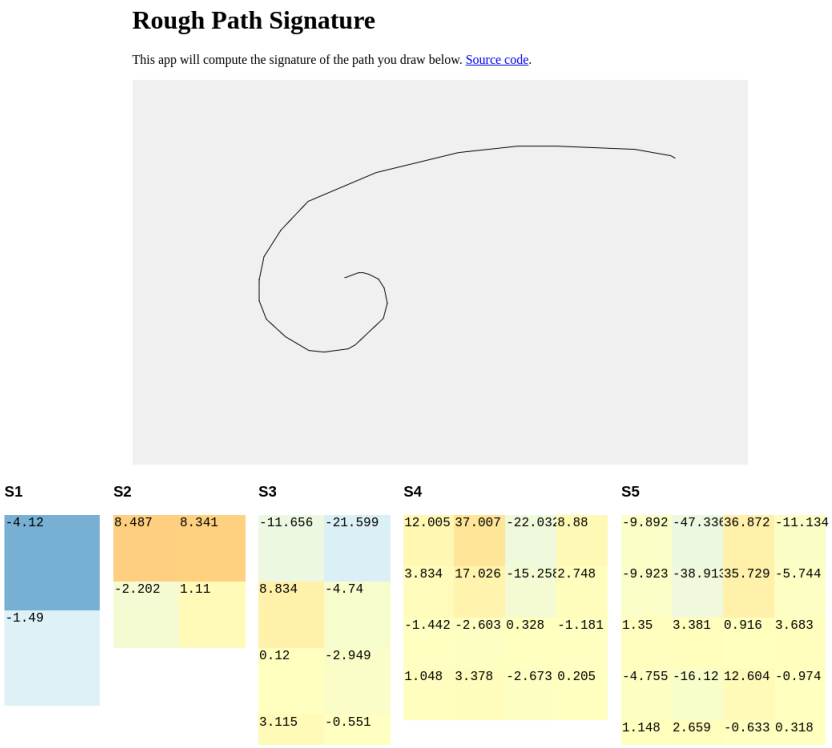


Figure 5: Screenshot of the `signature-visualizer` web app.

Sometimes having the signature as a flat one-dimensional array is more convenient to work with, in which case we can easily flatten the list of tensors. Since JavaScript does not not have a full-featured tensor manipulation library, we use the popular `mathjs` library [21] and supply some of the tensor manipulation functionality. Our implementation does not attempt to optimize performance in any way. The function that computes the signature in the web app uses a naive, nonvectorized implementation of (3.1). Nevertheless, this implementation is fast enough to serve the purposes of visualization in most modern web browsers. The source code of the web application is publicly available at `https://github.com/zhy0/signature-visualizer`.

3.2.2 Log signatures

The log signature is a compressed representation of the signature and lives in the special space $\mathfrak{g}^N(\mathbf{R}^d)$. We present brief sketches of two methods of computing log signatures. For full details, we refer to [20].

Projection of signatures Recall from Section 2.6 that the logarithm of an element of the truncated tensor algebra $T^N(\mathbf{R}^d)$ whose scalar component is 1 is defined as

$$\log(1+a) = \sum_{n=0}^{N} \frac{(-1)^{n+1}}{n} a^{\otimes n}. \tag{3.2}$$

The result is an element in the tensor space $T^N(\mathbf{R}^d)$ and it therefore has the same number of coefficients as $1+a$. In Section 2.6, we showed that signatures of paths can be embedded in the lower-dimensional vector space $\mathfrak{g}^N(\mathbf{R}^d)$, which is the free N-step nilpotent Lie algebra over d elements. Therefore, after computing the logarithm via the above formula, one can obtain the log signature (as an element of $\mathfrak{g}^N(\mathbf{R}^d)$) by projecting/rewriting the result in a basis of $\mathfrak{g}^N(\mathbf{R}^d)$. In other words, a possible procedure to computing the log signature is as follows:

1. compute the regular signature of the path as described in the previous subsection;
2. calculate the logarithm of the signature in the tensor space $T^N(\mathbf{R}^d)$, using (3.2);
3. project the result in a basis of $\mathfrak{g}^N(\mathbf{R}^d)$ (this is always possible by Theorem 32).

The core of this procedure is the third point. Essentially, this comes down to solving a linear equation. If we have an element $a \in \mathfrak{g}^N(\mathbf{R}^d)$ under a particular basis, we can find the corresponding representation of a by expanding the brackets. Let us illustrate this by the following example.

Example 42. Consider $d = 2$ and $N = 2$ and let $\{e_1, e_2\}$ be the standard basis of $\mathbf{R}^2$. Then $B := \{e_1, e_2, e_{11}, e_{12}, e_{21}, e_{22}\}$ is a basis of $T_0^N(\mathbf{R}^d) \simeq T_1^N(\mathbf{R}^d)$, with $e_{ij} := e_i \otimes e_j$, and $A := \{e_1, e_2, [e_1, e_2]\}$ is a basis for $\mathfrak{g}^N(\mathbf{R}^d)$. If under this latter basis $a = (0,0,1) \in \mathfrak{g}^N(\mathbf{R}^d)$, then we find that since $[e_1, e_2] = e_{12} - e_{21}$, it is given by $(0,0,0,1,-1,0)$ in the former basis. We therefore have the identifications of basis elements,

$$\begin{aligned}
(1,0,0) &\to (1,0,0,0,0,0),\\
(0,1,0) &\to (0,1,0,0,0,0),\\
(0,0,1) &\to (0,0,0,1,-1,0).
\end{aligned}$$

Therefore, the transformation matrix from the basis A of $\mathfrak{g}^N(\mathbf{R}^d)$ to the basis B of $T^N(\mathbf{R}^d)$ is given by

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix},$$

such that $a = (a_1, a_2, a_3)$ under basis A has representation La under B.

To go the opposite direction is slightly more difficult. Suppose we have $b \in T^N(\mathbf{R}^d)$ under basis B. Then to find its representation under basis A, we must find $a = (a_1, a_2, a_3)$ such that $La = b$. In general, this system is overdetermined and it may or may not have a solution. However, by Theorem 32, if b is the signature of some path, then we know that it has a unique representation in $\mathfrak{g}^N(\mathbf{R}^d)$; as such, $La = b$ must have a unique solution in this case. Thus, computing the log signature given the regular signature boils down to solving an overdetermined system of linear equations. This is a classical problem in linear algebra; one can for instance use methods such as ordinary least squares. ⊘

To apply this method, we must know what the basis elements of $\mathfrak{g}^N(\mathbf{R}^d)$ are, see for instance [10, p. 907]. Since these are only dependent on d and N (as $\mathfrak{g}^N(\mathbf{R}^d)$ is a universal object), it is possible compute these in advance. Further optimizations are possible, for these we refer to [20].

Direct computation of log signatures Instead of first computing the signature, there is a method to directly compute the log signature. This method is analogous to computing the regular signature, which uses Chen's identity to combine the signatures of individual line segments; here, instead of using Chen's identity, we use the Campell-Baker-Hausdorff formula given in Theorem C5.

Suppose $a, b \in \mathfrak{g}^N(\mathbf{R}^d)$ are the respective log signatures of the paths X and Y. The log signature of the concatenation $X * Y$ is then given by

$$\log S(X * Y) = \log[S(X) \otimes S(Y)] = \log[\exp(a) \otimes \exp(b)],$$

which is precisely the Campell-Baker-Hausdorff 'product' of a and b. Furthermore, recall that the log signature of a line segment is simply its total displacement, i.e., if $X : t \mapsto tv$ on $[0, 1]$, then $\log S(X) = v \in \mathfrak{g}^N(\mathbf{R}^d)$. Hence, we can compute the log signature of piecewise-linear paths by iteratively applying the CBH formula to each linear segment of the path. This method is summarized in the following steps:

1. determine and fix a basis of $\mathfrak{g}^N(\mathbf{R}^d)$;
2. compute the log signature of each linear segment of the path (this is trivial);
3. use the CBH formula to 'patch together' the log signatures to find the total log signature of the path.

We illustrate this procedure in the following example.

Example 43. Let $d = 2$ and $N = 2$ and let $\{e_1, e_2\}$ be the standard basis of $\mathbf{R}^2$, we fix the basis $\{e_1, e_2, [e_1, e_2]\}$ for $\mathfrak{g}^N(\mathbf{R}^d)$. Now consider the piecewise-linear path X that goes from 0 to v to $v + w$ in $\mathbf{R}^2$. We compute the log signature of this path.

The log signature of the linear segment 0 to v is simply $v = (v_1, v_2, 0)$ in the basis of $\mathfrak{g}^N(\mathbf{R}^d)$, and similarly, the log signature of the segment from v to $v + w$

is $w = (w_1, w_2, 0) \in \mathfrak{g}^N(\mathbf{R}^d)$. The CBH formula for the truncation level $N = 2$ is given by

$$\mathrm{cbh}(a, b) := \log[\exp(a) \otimes \exp(b)] = a + b + \frac{1}{2}[a, b].$$

We have that in the given basis of $\mathfrak{g}^N(\mathbf{R}^d)$,

$$[v, w] = [v_1 e_1 + v_2 e_2,\ w_1 e_1 + w_2 e_2] = (v_1 w_2 - v_2 w_1)[e_1, e_2].$$

Hence, the log signature of X is given by

$$\begin{aligned}\mathrm{cbh}(v, w) &= (v_1 + w_1)e_1 + (v_2 + w_2)e_2 + \frac{1}{2}(v_1 w_2 - v_2 w_1)[e_1, e_2] \\ &= \left(v_1 + w_1,\ v_2 + w_2,\ \frac{1}{2}(v_1 w_2 - v_2 w_1)\right). \qquad \oslash\end{aligned}$$

To implement the above procedure for truncation level N, one needs to know all the coefficients and terms of the CBH formula up to order N. These can be computed in advance and stored in a look-up table. See [20] for details.

3.3 Machine learning examples

3.3.1 Regression of stochastic differential equation

In this first example, we attempt to predict the value of a diffusion process driven by a Brownian motion. This example is borrowed from [17]. Let $(B_t)_{t \geq 0}$ be a one-dimensional standard Brownian motion with initial value $B_0 = 0$. We consider a process $(Y_t)_{t \geq 0}$ given by the Stratonovich SDE

$$dY_t = a(1 - Y_t)dt + bY_t^2 \circ dB_t, \quad Y_0 = 0, \tag{3.3}$$

where $a = 1$ and $b = 2$ and $T = 0.25$ are fixed. Our goal is to estimate the terminal value Y_T using the driving signal $(B_t)_{t \in [0,T]}$.

We generate 800 samples of discrete Brownian motion paths with uniform step size, that is, each sample is of the form $(B_{iT/K})_{i=0}^K$, with K the sampling frequency. The generated 800 samples are split 50-50 for training and testing. For each sample Brownian motion path, we use a stochastic Runge-Kutta method [22] implemented by the `sdeint` Python package [23] to approximate its corresponding solution $(Y_{iT/K})_{i=0}^K$ of (3.3), see Figure 6. The terminal value Y_T of each sample will then be the output variable for the machine learning problem. For the input variable, we will use the signature of either the lead-lag, time-joined or time-embedded transformation of the discrete driving signal $\left(B_{iT/K}\right)_{i=0}^K$.

In [17], the authors compared the performance of the signature feature set against the raw increment feature set for different sampling frequencies K. Our approach is to compare the performance of the signature and log signature feature set and the different embedding methods (lead-lag, time-joined and time-embedded).

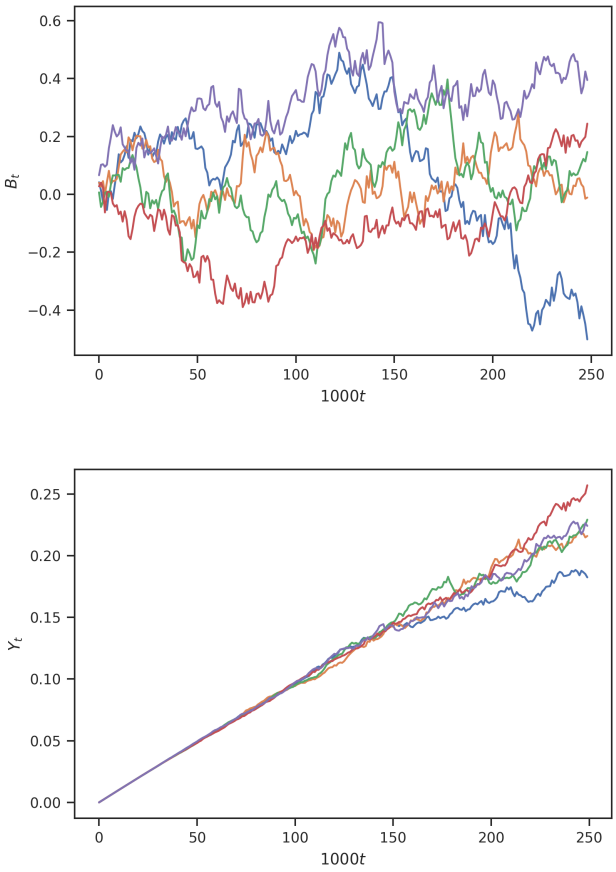


Figure 6: Sample paths of B and corresponding sample paths for Y generated with $K = 250$.

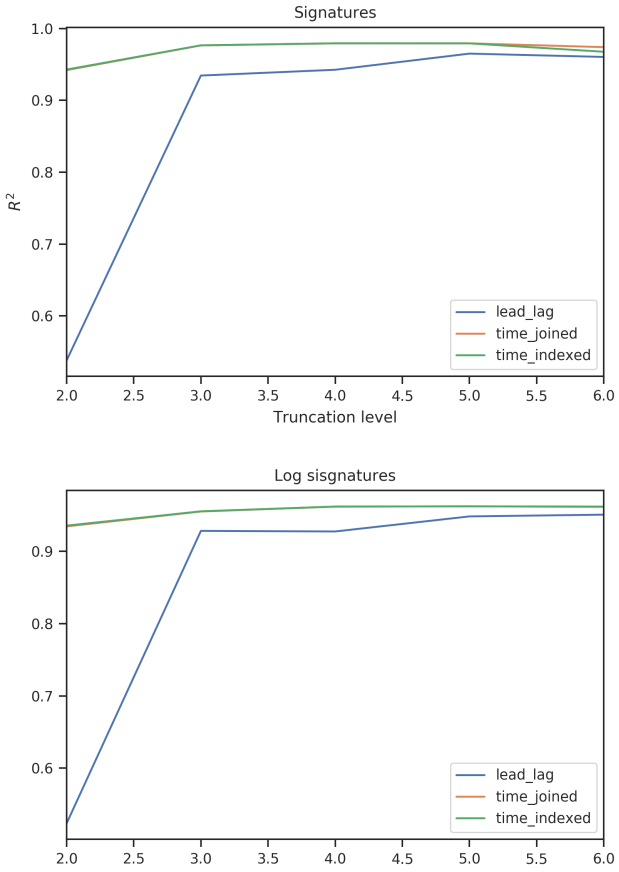


Figure 7: Linear regression results using regular signatures and log signatures for various transformations and truncation levels.

We therefore only consider a fixed sampling frequency of $K = 250$. Furthermore, we use a simple ordinary least squares (OLS) linear regression algorithm without any regularization. As a simple benchmark, the R^2 statistic of the trained model on the testing set is used. The results for different truncation levels and transformations are shown in Figure 7.

We see in the results that signatures and log signatures perform roughly the same. This can be explained by the fact that coefficients in the signature are linear combinations of the coefficients in the log signature. Also, the results for the time-joined and time-embedded transformations are almost indistinguishable and happen to be better than the results for the lead-lag transformation.

3.3.2 Classification of PenDigits

In this classical example, we consider the classification of PenDigits from the paper [24] and the UCI repository [25]. The data set has a total of 10992 samples of handwritten digits which are split into 7494 samples for training and 3498 for testing. Each handwritten digit is encoded as a piecewise-linear path in $\mathbf{R}^2$ with 16 points (Figure 8), this allows us to easily compute the signature without additional transformations.

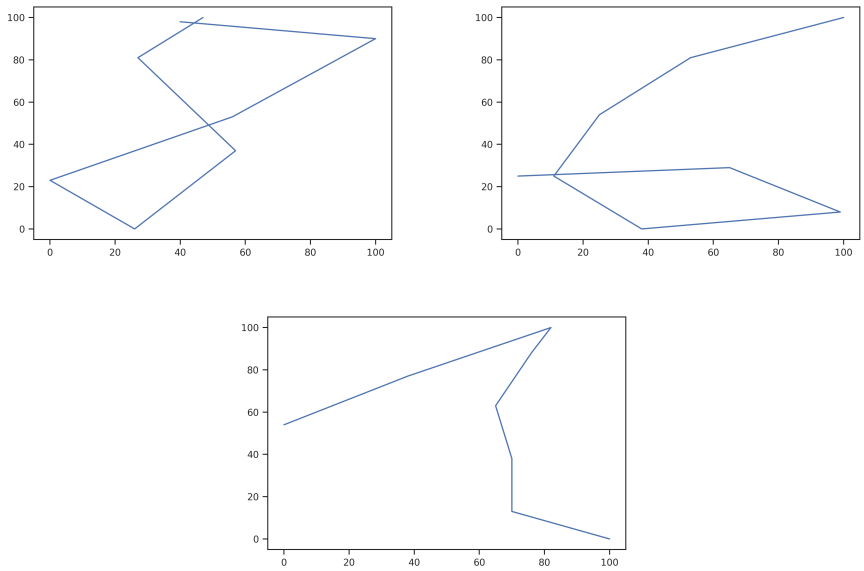


Figure 8: Handwritten digits corresponding to 8, 6 and 1 respectively.

In our numerical experiment, we compare the use of signature features of different truncation levels. For classification, we use logistic regression with L^2 regularization. The percentage of correctly classified digits in the testing set will be used as a simple benchmark. These scores are shown graphically in Figure 9 for several signature truncation levels. Table 2 contains this data along with the training times.

One can observe a peculiar spike in training time for the sixth truncation level. This is likely due to specific implementation details (e.g., cache misses) of the logistic regression algorithm we used.

The results show that we are able to achieve an error rate of less than 5% by using truncation levels above 7. To put this into context, if the same logistic regression algorithm is used without signatures, an error rate of 10% is achieved, similar to level 5 signatures. While signatures with high truncation levels perform better, it should be noted that the training time for vanilla logistic regression is about ten times shorter than its level 5 signature counterpart. Nevertheless, we see that signatures are effective features for this data set.

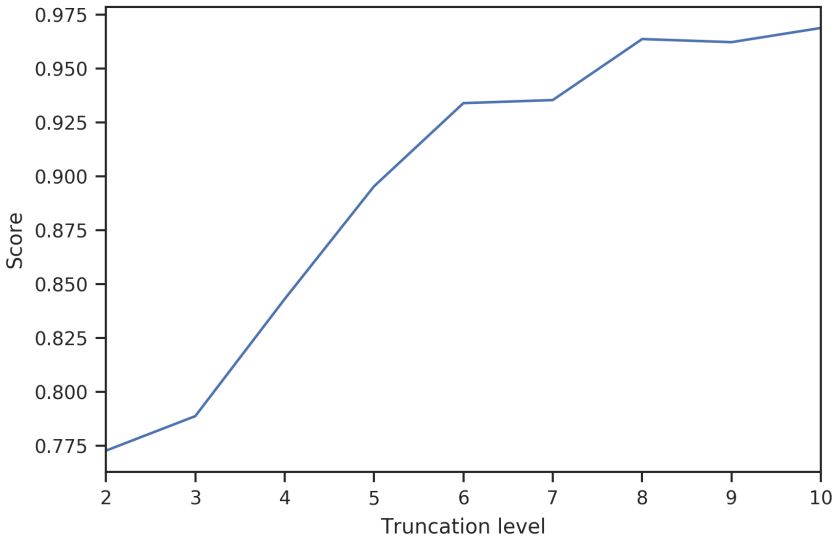


Figure 9: Plot of pendigits classification scores by signature truncation level.

Table 2: Pendigits classification scores and training times.

Truncation level	Score	Training time (s)	Coefficients in signature
2	0.7727	0.6492	3
3	0.7887	1.6003	7
4	0.8431	16.3615	15
5	0.8954	59.3687	31
6	0.9340	130.4544	63
7	0.9354	90.2182	127
8	0.9637	92.2106	255
9	0.9623	149.5867	511
10	0.9688	267.1747	1023

In the literature of signatures, the same pendigits data set has appeared before in [26] and [27]. In the latter text, the same numerical experiment is conducted with a different classification algorithm (random forest ensemble method) and achieves slightly lower scores than logistic regression, but with faster training times. In [26], the authors considered kernel learning using signatures. In their benchmarks on the pendigits data set, an error rate of 3% has been achieved at level 4 truncation using support vector machine classification with a signature-modified Gaussian kernel. The most successful application of signatures in character recognition is perhaps [28]. This paper tackled (online) classification of handwritten Chinese characters. Using a combination of convolutional neural networks and signature features, the author was able to improve the state-of-the-art by a significant 2% in reduced error rate. For a general overview of signatures in machine learning, we refer to [16].

3.4 Benchmarks on UCR TSC archive

In this section, we present several benchmarks of the signature method on the (univariate) UCR Time Series Classification repository [4]. This repository currently contains 128 classification datasets. The problems span many different domains, including chemistry, astronomy and anthropology.

For our numerical experiment, we selected the 114 datasets for which the time series do not have any missing values. The approach is similar to the two previous examples. For each data set, we train a classifier on signature features using the supplied training set. These features are computed by taking the signature of either the lead-lag, time-joined or time-indexed transformation of the univariate series. (We cannot use signatures on the untransformed univariate series, because the signature for one-dimensional paths is trivial, see Section 3.1.) Before classification, the signature features are normalized by subtracting the mean and dividing the standard deviation column-wise. This prevents numerical issues during classification. We then compute the accuracy score (i.e., percentage correctly classified labels) on the supplied testing set. This is repeated for all signature truncation levels between 2 and 10. Three different classifiers have been used:

- logistic regression with L^2 regularization,
- support vector machine classifier (SVC) with linear kernel and L^2 regularization,
- k-nearest neighbors classifier with $k = 5$ and Euclidean distance.

The classifications have been done using Python's `scikit-learn` library [29].

Difference in classifiers We compare the results between the classifiers using their respective maximum scores. That is, for each dataset, we take the maximum achieved score on the testing set over all truncation levels and all transformations. With this data, we can compare classifiers pairwise using a scatter plot, see Figure 10. For instance, if we have a point (x, y) in the left figure, then x is the highest

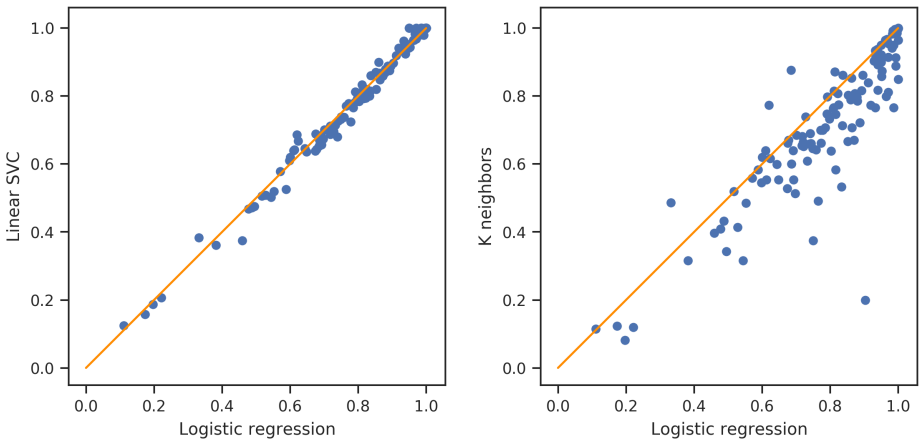


Figure 10: Comparison between max scores for the three classifiers.

achieved score on this dataset using logistic regression and y is the highest achieved score using linear SVC.

From this figure we see that SVC and logistic regression perform on par (with the latter slightly better), while k-neighbors generally performs worse than both.

Difference in transformations In Figure 11, we compare the difference in performance between the three transformations for the linear support vector classifier. Similar to the previous figure, each point in the scatter plot corresponds to the score on a single dataset. This number for a particular transformation is the maximum score over all truncation levels.

In the figure we see that the time-indexed and time-joined transformation perform roughly equally well and outperform the lead-lag transformation most of the time. For the other two classifiers, the results are similar.

Comparison between truncation levels Generally, when the truncation level is increased, there are more coefficients, so the classifier has more available information to work with. In practice, when there are too many features, classifiers can be prone to overfitting. Although regularization partly addresses this problem by introducing a penalty, it is not the case that a higher truncation level always leads to better results.

In Figure 12, the frequency for which a particular truncation level achieves the maximum score is plotted. (If two the maximum is achieved at two different levels, it will count for the lowest level.) We see the results for the lead-lag transformation are fairly uniform, which seems to imply that increasing the truncation level for this transformation does not improve the performance. However, as shown in Figure 11, the lead-lag transformation is a poor choice for most datasets, so the uniformity we observe in the histogram could be caused by this incompatibility in the first place.

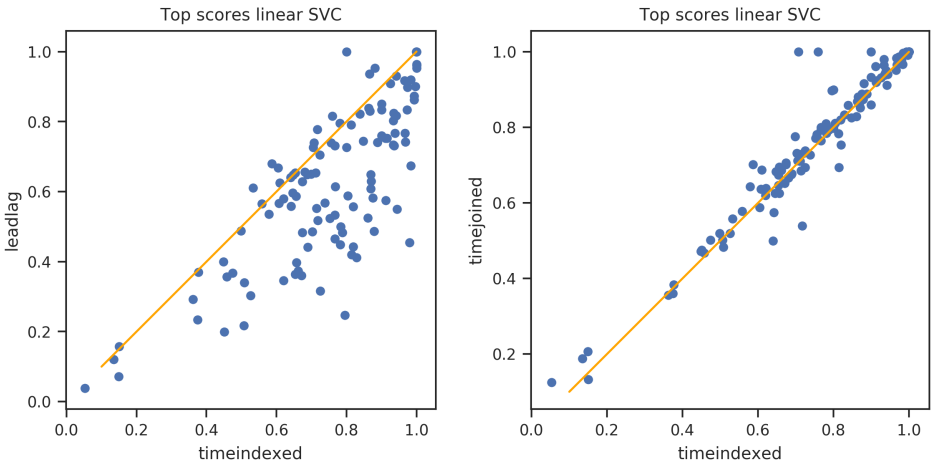


Figure 11: Comparison of max scores between different transformations.

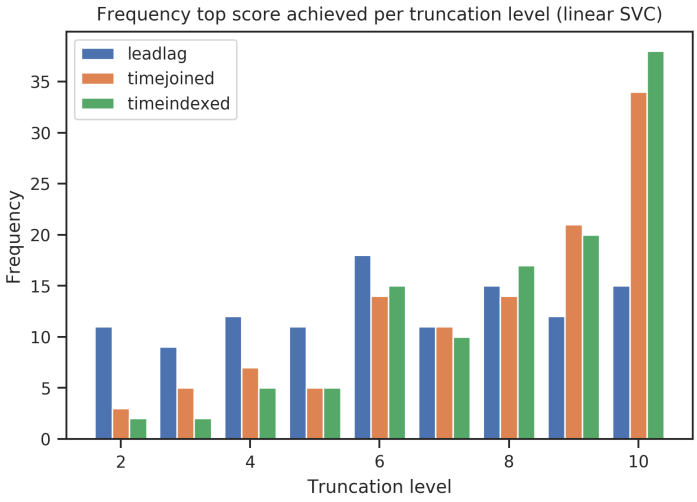


Figure 12: This histogram shows how often the highest score is achieved at a certain truncation level.

For the time-joined and time-indexed transformations, the pattern seems that higher truncation levels lead to better performance, as the trend is generally increasing and the highest level contains the most maximum observed scores. Even so, the 'optimal' truncation level still highly depends on the problem set at hand, and is best uncovered by experimentally verifying multiple truncation levels. Of course, a higher truncation level also comes at an (often significant) increase in training time. In our benchmarks, the higher levels consume the substantial majority of the total training time. For most datasets, the progression of training times does not look different from those in Table 2 for the digit classification problem.

Comparison with other known benchmarks Our results are compared against existing benchmarks supplied with the datasets. Currently, benchmarks are available for 85 datasets (a subset of our 114 selected datasets), and contain results for 37 different classification algorithms. Among these 37 classifiers are basic and well-known algorithms such as vanilla k-nearest neighbors, as well as algorithms specifically designed for time series classification, e.g., shapelet transforms [31]; but also *ensembles* of classifiers, such as Flat-COTE [30]. These ensembles train a collection of classifiers and then select which classifier to use based on a pre-defined voting scheme. The best-performing classifier in the dataset is the HIVE-COTE ensemble [32]. It achieves the highest score in 21 of the 85 datasets.

In Figure 13, we compare the best results for signatures against the best results in the benchmarks. That is, for each dataset, we take the maximum achieved score over all transformation and truncation levels, and compare this against the maximum score over all 37 classifiers in the benchmarks. We see that the existing benchmark maximum outperforms signatures on almost every dataset. There are 6 datasets in which the signature achieves a higher score than the benchmarks, and 5 more in which the signature performs on par with the benchmarks. The 6 datasets for which the signature performs better are highlighted in the table of full results in Appendix D.

In Figure 14, logistic regression is compared against two of the best-performing classifiers. The results seem to be in favor of the other two classifiers, although there are a handful of datasets in which the signature performs better. This may possibly suggest that there is an opportunity for signatures to improve current ensemble classifiers. An interesting future project would therefore be to investigate whether integrating signature features in the COTE ensemble would enhance its performance.

Conclusion and future work In our numerical experiments on the UCR repository, we have attempted to shed light on a few questions regarding the use of signatures in classification problems. Among other things, we have

- tested the use of different classifiers on signature features;

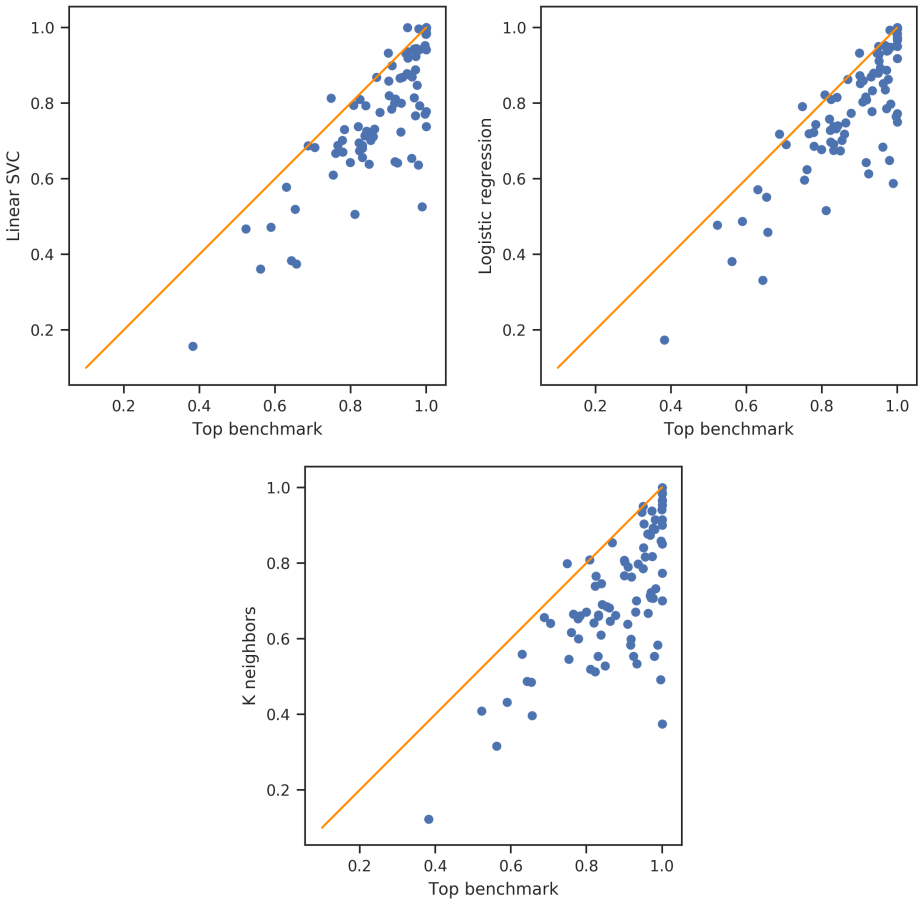


Figure 13: Comparison of signatures against the top benchmark.

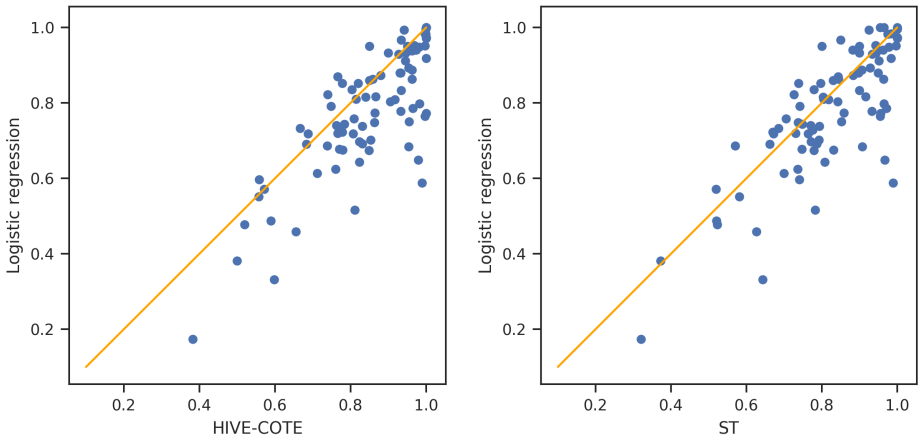


Figure 14: Comparison of signatures with logistic regression against HIVE-COTE [32] and ST [31].

- compared the effectiveness of the various transformation steps introduced in Section 3.1, and
- analyzed the performance of different truncation levels.

Furthermore, a simple comparison with existing benchmarks is provided. These show that signatures generally do not perform better than the current best classifiers. Nevertheless, there are plenty datasets on which signatures achieve good results.

The analysis we presented here is far from comprehensive, however. There are many aspects that can be improved. First, we note that we have only performed each classification experiment once. For more robust results, each classification should be done multiple times with different random seed values, so that we can give a confidence interval for the scores we obtain. Using multiple runs also allows us to use statistical tests to compare the effectiveness between different classifiers and transformations, which is recommended by the creators of the UCR archive [4].

Besides methodological issues, there are also a few practical omissions in our current analysis. For instance, we have completely omitted the log signature in our numerical experiments. Part of our reasoning for this lies in our previous (mostly undocumented) experience with log signatures, which showed that the performance of log signatures is often identical to that of regular signatures. The first problem in Section 3.3 is an example of this, see Figure 7. Nevertheless, an experimental verification of this is needed in a future work. Another omission is that we have not performed any analysis on multivariate datasets. We have disregarded the multivariate UEA archive [33] in our experiments partly due to a lack of computational resources. By investing more in computational resources in a future project, we hope to address these two shortcomings together with the robustness issue described in the first point.

Finally, there are a few fundamental questions which we have not explored in the current work. For instance, we have not attempted to investigate what types of problems are best suited for signature features. It would be beneficial for practitioners to have a set of heuristics that could describe which problems or patterns would be compatible with signatures. A more careful analysis of the individual datasets in a future project would help improve our understanding in this area.

A Paths of Bounded Variation

An important class of functions $[0,T] \to \mathbf{R}^d$ is the set of continuous functions of bounded variation. Roughly speaking, these functions are paths that have finite length. We devote this appendix section to summarize the most important properties of these functions.

Definition A1. Consider the interval $[0,T]$. A *partition* P of $[0,T]$ is a set of points

$$\{\, 0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T \,\}.$$

The *mesh size* of P, denoted by $|P|$, is the largest difference between subsequent points in the partition:

$$|P| := \max_{0 \le i \le n-1} |t_{i+1} - t_i|.$$

The set of all partitions of $[0,T]$ is denoted by $\mathcal{P}([0,T])$. ⊘

Definition A2. Let $X : [0,T] \to \mathbf{R}^d$ be a function. The *total variation* or *1-variation* of X is defined as the following supremum over all partitions of $[0,T]$:

$$\|X\|_{1\text{-var}} := \sup_{P \in \mathcal{P}([0,T])} \sum_{i=0}^{p_n - 1} \left| X_{t_{i+1}} - X_{t_i} \right|.$$

We say that X is of *bounded variation* if X has finite total variation. The space of all continuous functions $[0,T] \to \mathbf{R}^d$ with bounded variation will be denoted by $\mathcal{V}([0,T], \mathbf{R}^d)$. ⊘

One can easily show that the total variation is nonnegative, subadditive and homogeneous; hence a semi-norm on $\mathcal{V}([0,T], \mathbf{R}^d)$.

Example A3. Suppose $X : [0,T] \to \mathbf{R}^d$ is piecewise-linear, then the total variation of X is the sum of the length of each linear piece. ⊘

Proposition A4. *Suppose $X \in \mathcal{V}([0,T], \mathbf{R}^d)$ is also continuously differentiable. Then*

$$\|X\|_{1\text{-var}} = \int_0^T |\dot{X}_s| \, ds,$$

in which $|\cdot|$ is the Euclidean norm on $\mathbf{R}^d$.

B Riemann-Stieltjes Integral

In Riemann integration, one integrates a function $f : \mathbf{R} \to \mathbf{R}$ with respect to the the independent variable,

$$\int_a^b f(t)\, dt.$$

The Riemann-Stieltjes integral is a generalization of the Riemann integral. It allows one to integrate functions with respect to other functions,

$$\int_a^b f(t)\, d\alpha(t).$$

Riemann integration corresponds to the special case when α is the identity map.

In this section we present a brief overview of this type of integration. We aim to highlight the important properties that are needed in our main exposition. As such, proofs of these results are omitted. The approach taken here is for one-dimensional integrands and integrators. The formulations are similar to [9, Chapter 2]. Additionally, we have included references to proofs that may be hard to find in the literature.

Definition B1. Let $X, Y : [0, T] \to \mathbf{R}$ be two one-dimensional paths. Let $(P_n)_{n\in\mathbf{N}}$ be a sequence of partitions

$$P_n = \left\{ 0 = t_0^n < \cdots < t_{p_n}^n = T \right\}$$

with the property that the mesh size $|P_n| \to 0$ as $n \to \infty$. We say that the Riemann-Stieltjes integral of Y against X exists if there is some $I \in \mathbf{R}$ such that

$$I = \lim_{n\to\infty} \sum_{i=0}^{p_n-1} Y_{\xi_i^n} \left(X_{t_{i+1}} - X_{t_i} \right),$$

for any choice of $\xi_i^n \in \left[t_i^n, t_{i+1}^n\right]$. We call I the *Riemann-Stieltjes integral* of Y against X and write

$$\int_0^T Y_s dX_s := I. \qquad \oslash$$

Theorem B2 ([9, Prop 2.2])**.** *For any two elements of $X, Y \in \mathcal{V}([0, T], \mathbf{R})$ and $t \in [0, T]$, the Riemann-Stieltjes integral*

$$\int_0^t Y_s\, dX_s$$

exists. Moreover, we have the following properties,

(i)
$$\int_0^T Y_s\, dX_s = \int_0^t Y_s\, dX_s + \int_t^T Y_s\, dX_s;$$

(ii) *the integral pairing* $\mathcal{V}([0,T],\mathbf{R}) \times \mathcal{V}([0,T],\mathbf{R}) \to \mathbf{R}$

$$(X,Y) \mapsto \int_0^t Y_s \, dX_s$$

is bilinear;

(iii) *if X is continuously differentiable, then*

$$\int_0^t Y_s \, dX_s = \int_0^t Y_s \dot{X}_s \, ds.$$

It is not strictly necessary for the integrand Y to be in $\mathcal{V}([0,T],\mathbf{R})$, being only continuous is sufficient. For the sake of simplicity, we formulate these results for $Y \in \mathcal{V}([0,T],\mathbf{R})$ since this covers all cases we need.

Proposition B3 (Substitution, [34, Thm 12.11])**.** *Let $X, Y : [a,b] \to \mathbf{R}$ be continuous paths of bounded variation and let $\phi : [c,d] \to [a,b]$ be a nondecreasing or nonincreasing surjection. Then*

$$\int_a^b Y_{\phi(s)} \, dX_{\phi(s)} = \int_c^d Y_s \, dX_s.$$

Proposition B4 (Integration by parts, [34, Thm 12.12])**.** *Let $X, Y : [0,T] \to \mathbf{R}$ be continuous paths of bounded variation. Then*

$$X_t Y_t = X_0 Y_0 + \int_0^t X_s \, dY_s + \int_0^t Y_s \, dX_s.$$

Proposition B5 (Associativity, [34, p. 328])**.** *Let $X, Y, Z \in \mathcal{V}([0,T],\mathbf{R})$, then the integral map $[0,T] \to \mathbf{R}$*

$$\int_0^{\cdot} Y_s \, dX_s := t \mapsto \int_0^t Y_s \, dX_s$$

is again continuous and of bounded variation, hence an element of $\mathcal{V}([0,T],\mathbf{R})$. Furthermore,

$$\int_0^t Z_s \, d\left(\int_0^s Y_u \, dX_u \right) = \int_0^t Y_s Z_s \, dX_s.$$

Proposition B6 (Uniform convergence, [9, Prop 2.7])**.** *Let $(X_n)_{n\in\mathbf{N}}$ and $(Y_n)_{n\in\mathbf{N}}$ be two sequences in $\mathcal{V}([0,T],\mathbf{R}^d)$. Assume that both sequences converge uniformly to $X, Y \in \mathcal{V}([0,T],\mathbf{R}^d)$ respectively, and that $\sup_{n\in\mathbf{N}} \|X_n\|_{1\text{-var}} < \infty$. Then*

$$\int_0^{\cdot} Y_n(s) \, dX_n(s) \longrightarrow \int_0^{\cdot} Y_s \, dX_s \quad \textit{uniformly on } [0,T] \textit{ as } n \to \infty.$$

C Lie Groups and Lie Algebras

The theory of Lie groups and Lie algebras is heavily used in Section 2.6 to study a compact representation of signatures. Although most of this theory is elementary and widely known, we found that most of the relevant pieces are rather scattered across the literature. The goal of this section is therefore to present a concise overview of the theory that is needed for Section 2.6.

Definition C1 (Lie group)**.** A Lie group $(G, \cdot)$ is a smooth manifold that is also a group (in the algebraic sense), for which the multiplication $\cdot$ and inversion $^{-1}$ are smooth maps. ⊘

Definition C2 (Lie algebra)**.** A Lie algebra $\mathfrak{g}$ is a vector space equipped with a bracket operation $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ that is bilinear, antisymmetric and satisfies the Jacobi identity:

$$[a,[b,c]] + [b,[c,a]] + [c,[a,b]] = 0,$$

for all $a, b, c \in \mathfrak{g}$. ⊘

Let $\mathfrak{X}(G)$ be the set of all smooth vector fields on G. On this space there is a natural bracket operation that associates to any two vector fields $X, Y \in \mathfrak{X}(G)$ a third vector field $[X, Y] \in \mathfrak{X}(G)$, called the *Lie bracket* of X and Y.

Any element $g \in G$ defines a map

$$L_g : G \to G, \quad h \mapsto gh,$$

which is called *left translation*. Using the left translation, one can turn any vector v in the tangent space at the identity $\mathcal{T}_e G$ into a vector field on G,

$$\mathcal{T}_e G \to \mathfrak{X}(G), \quad v \mapsto \vec{v}, \quad \vec{v}_g = (dL_g)_e(v) \in \mathcal{T}_g G.$$

There is the following relationship between Lie groups and Lie algebras.

Proposition C3. *Let G be a Lie group and let $\mathfrak{g}$ be the tangent space of G at the identity. Then $\mathfrak{g}$ endowed with the bracket operation*

$$[v, w] := [\vec{v}, \vec{w}]$$

becomes a finite dimensional Lie algebra, called the canonical Lie algebra of the Lie group G.

An important map from the Lie algebra to the Lie group is the exponential map.

Definition C4. Let G be a Lie group and let $\mathfrak{g}$ be its Lie algebra. The *exponential map* is defined as

$$\exp : \mathfrak{g} \to G, \quad v \mapsto \gamma(1)$$

where $\gamma : \mathbf{R} \to G$ is the unique integral curve that satisfies

$$\dot{\gamma}(t) = \vec{v}_{\gamma(t)}, \quad \gamma(0) = e.$$

⊘

Theorem C5 (Campell-Baker-Hausdorff, [35, Prop 9.2.32])**.** *Let G be a Lie group and $\mathfrak{g}$ its canonical Lie algebra. Then there is a neighborhood $U \subset \mathfrak{g}$ of the identity element e, such that for any $a, b \in U$, there exists an element $c \in \mathfrak{g}$ such that* $\exp(a)\exp(b) = \exp(c)$*. Furthermore, c is given by the following converging series of right-brackets:*

$$c = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} \sum_{p_i+q_i>0} \frac{[a^{p_1}, b^{q_1}, \ldots, a^{p_n}, b^{q_n}]}{\left(\sum_{i=1}^{n} p_i + q_i\right) \prod_{i=1}^{n} p_i! q_i!},$$

in which a^p means having a appear p times in succession, e.g., $[a^2, b] = [a, a, b] = [a, [a, b]]$, see (C.1)*.*

C.1 Structure theory of Lie algebras

We turn to an algebraic view of Lie algebras.

Definition C6. A Lie algebra homomorphism f between two Lie algebras $\mathfrak{g}$ and $\mathfrak{h}$ is a linear map that preserves brackets,

$$f([a, b]) = [f(a), f(b)],$$

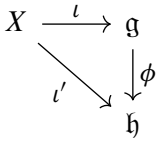
for all $a, b \in \mathfrak{g}$. ⊘

An important object is the free Lie algebra, which one can think of as a Lie algebra that has not been imposed any restrictions, besides for its bracket to satisfy the Jacobi identity.

Definition C7. Let X be a set. A Lie algebra $\mathfrak{g}$ is said to be *free over X* if

(i) there is an inclusion map $\iota : X \to \mathfrak{g}$; and

(ii) for every Lie algebra $\mathfrak{h}$ and $\iota' : X \to \mathfrak{h}$, there exists exactly one Lie algebra homomorphism $\phi : \mathfrak{g} \to \mathfrak{h}$ such that the diagram below commutes.



⊘

One can quickly verify that, up to Lie algebra homomorphisms, there can only be at most one free Lie algebra over a certain set X.

The free Lie algebra can also be viewed as the set of all formal brackets between elements of X. For instance, if $X = \{a, b\}$, then the free Lie algebra over X is the

linear span of elements

$$
\begin{aligned}
&a, b \\
&[a, b] \\
&[a, [a, b]], [b, [a, b]] \\
&[a, [a, [a, b]]], [a, [b, [a, b]]], [[a, b], [a, b]], \ldots \\
&\vdots
\end{aligned}
$$

The list continues indefinitely, spanning nested brackets of all degrees. One can show that free Lie algebras are necessarily infinite dimensional vector spaces. Note also the appearance of 'mixed' brackets such as $[[a, b], [a, b]]$. In Proposition C10 below, it is shown that it is sufficient to only consider 'right brackets' such as $[a, [a, b]]$ and $[a, [a, [a, b]]]$.

The following theorem gives a concrete construction of a free Lie algebra over a set.

Theorem C8 (Construction free Lie algebra)**.** *Let X be a set and let V be the free vector space over X (i.e., V is a vector space which has basis X). Let $T(V)$ be the tensor algebra over V,*

$$T(V) = \bigoplus_{n=0}^{\infty} V^{\otimes n},$$

endowed with bracket

$$[a, b] := a \otimes b - b \otimes a.$$

Then the smallest Lie subalgebra $\mathfrak{L}(V)$ of $T(V)$ that contains V is a free Lie algebra over X.

Proof. The theorem is usually derived from the Poincaré-Birkhoff-Witt theorem, see [36]. □

Remark C9. The reader may note the similarity between the definition of the tensor algebra $T(V)$ and $T((V))$, the *extended* tensor algebra over V (Definition 1). The difference is the following: if $a = (a_0, a_1, \ldots) \in T(V)$ then only *finitely many* a_n are nonzero; while $T((V))$ does not impose this condition. In a certain sense, one can view $T((V))$ as a completion of $T(V)$ under a suitable metric. Analogously, $T(V)$ is to $T((V))$ as the polynomial ring $\mathbf{R}[X]$ is to the formal power series $\mathbf{R}[[X]]$. ⊘

The next proposition gives a useful expression for the smallest Lie subalgebra that contains a set. Before this, we introduce additional notation. We will write $[x_1, \ldots, x_n]$, for the iterated right-bracketing of n elements, i.e., we define recursively

$$[x_1] := x_1, \quad [x_1, \ldots, x_{n+1}] := [x_1, [x_2, \ldots, x_n]] \tag{C.1}$$

If V and W are subsets of $\mathfrak{g}$, then $[V, W]$ denotes the linear subspace

$$[V, W] = \operatorname{span}\{\, [v, w] \mid v \in V,\ w \in W \,\}.$$

Proposition C10. *Let $\mathfrak{g}$ be a Lie algebra and $V \subset \mathfrak{g}$ a linear subspace. Then the smallest Lie subalgebra $\mathfrak{h}$ of $\mathfrak{g}$ that contains V can be written as the sum of vector subspaces*

$$\mathfrak{h} = V + [V, V] + [V, V, V] + \cdots = L_1 + L_2 + \ldots.$$

Proof. The inclusion $\supset$ is clear. Indeed, any element on the right is a finite linear combination of nested brackets of vectors in V, which must lie in $\mathfrak{h}$ because this is a Lie algebra that contains V.

For the reverse inclusion $\subset$, we show that the right-hand side is a Lie subalgebra of $\mathfrak{g}$ that contains V. Then $\mathfrak{h}$, being the smallest such Lie subalgebra, must be contained in the right-hand side. We only need to show that the right-hand side is closed under the the bracket operation. For this, it is sufficient to prove that

$$[L_n, L_k] \subset L_{n+k}, \quad \text{for all } n, k \geq 0.$$

We prove by induction to n with fixed k. The base case $n = 0$ is immediate by definition, for all k. Suppose that the statement holds for all k and a particular n, then for $n + 1$ we have

$$\begin{aligned}
[L_{n+1}, L_k] &= [[L_1, L_n], L_k] = [L_k, [L_1, L_n]] \\
&= -[L_1, [L_n, L_k]] - [L_n, [L_k, L_1]] \\
&\subset [L_1, L_{n+k}] + [L_n, L_{k+1}] \\
&\subset L_{n+k+1}.
\end{aligned}$$

The second line follows from Jacobi's identity and the induction step is applied in the third. This completes the proof. □

Definition C11 (Nilpotency)**.** A Lie algebra $\mathfrak{g}$ is called nilpotent if there exists an integer $n \geq 1$ such that the nth iterated right brackets of $\mathfrak{g}$ is zero:

$$\underbrace{[\mathfrak{g}, \mathfrak{g}, \ldots, \mathfrak{g}]}_{n+1 \text{ times}} = 0.$$

The smallest integer n such that this holds is called the *nilpotency degree* of $\mathfrak{g}$. ⊘

Definition C12. Let $\mathfrak{L}(V)$ be the free Lie algebra over a vector space V. For any $n \geq 1$, we define the ideal $I_n = \{\, l = (l_1, l_2, \ldots) \in \mathfrak{L}(V) \mid l_1 = \cdots = l_n = 0 \,\}$. The quotient $\mathfrak{L}^n(V) = \mathfrak{L}(V)/I_n$ is called the free n-step nilpotent Lie algebra over V. ⊘

D Full Benchmark Results

The following table contains the results of our numerical experiments in Section 3.4. In the experiments, we performed classifcation on nine different truncation levels for each of the transformations introduced in Section 3.1. In this table we do not show these results, but instead show the maximum over the truncation levels for each transformation.

	Logistic regression			Linear SVC			k-neighbors			Benchmarks	
	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Score	Classifier
Dataset											
ACSF1	0.6000	0.4600	0.6100	0.6400	0.5000	0.6400	0.6400	0.4700	0.6400	-	-
Adiac	0.2302	0.5038	0.5166	0.2174	0.5013	0.5064	0.1714	0.5192	0.5038	0.8107	HIVE-COTE
ArrowHead	0.4743	0.7486	0.7314	0.4857	0.7314	0.7029	0.6171	0.6000	0.6457	0.8629	HIVE-COTE
BME	0.8800	0.9867	0.9733	0.8733	1.0000	0.9933	0.6533	0.7667	0.7467	-	-
Beef	0.5000	0.8333	0.7667	0.5333	0.8000	0.7667	0.5000	0.5000	0.5333	0.9333	SVMQ
BeetleFly	0.9000	0.9500	0.9500	0.8500	**1.0000**	0.9000	0.9500	0.9000	0.9000	0.9500	HIVE-COTE
BirdChicken	0.9500	0.8000	0.8000	1.0000	0.9000	0.8000	0.9000	0.7500	0.7500	1.0000	SAXVSM
CBF	0.8989	0.9822	0.9678	0.8978	0.9822	0.9744	0.7267	0.9411	0.8733	0.9989	LPS
Car	0.5000	0.8167	0.8167	0.5000	0.8000	0.7833	0.5000	0.5500	0.5833	0.9167	TWE_1NN
Chinatown	0.7739	0.9623	0.9391	0.8029	0.9652	0.9333	0.8261	0.9594	0.9652	-	-
ChlorineConcentration	0.5674	0.6089	0.6130	0.5581	0.5750	0.6419	0.5094	0.5534	0.5385	0.9242	SVMQ
CinCECGTorso	0.5616	0.7652	0.7464	0.5232	0.7710	0.7522	0.4913	0.4761	0.4500	0.9964	HIVE-COTE
Coffee	0.9643	1.0000	0.9643	0.9643	1.0000	1.0000	0.9643	0.7857	0.8929	1.0000	SVML
Computers	0.6240	0.5960	0.6160	0.6680	0.5880	0.6040	0.6160	0.5840	0.5840	0.7600	HIVE-COTE
CricketX	0.3256	0.6846	0.6974	0.3641	0.6744	0.6538	0.3231	0.4974	0.5128	0.8231	HIVE-COTE
CricketY	0.3154	0.6667	0.6744	0.3462	0.6385	0.6205	0.2846	0.5282	0.5282	0.8487	HIVE-COTE
CricketZ	0.3154	0.6795	0.6923	0.3974	0.6256	0.6564	0.3000	0.5513	0.5538	0.8308	HIVE-COTE
Crop	0.4574	0.7169	0.6843	0.4826	0.7068	0.6742	0.4470	0.6542	0.6166	-	-
DiatomSizeReduction	0.8595	0.9935	0.9902	0.8627	**0.9967**	0.9935	0.8039	0.8889	0.8889	0.9804	LS
DistalPhalanxOutlineAgeGroup	0.7266	0.7410	0.7122	0.7266	0.7122	0.7050	0.6835	0.6475	0.6906	0.8417	SAXVSM
DistalPhalanxOutlineCorrect	0.6775	0.7283	0.6848	0.5870	0.6957	0.6558	0.7391	0.6848	0.6957	0.8225	MLP
DistalPhalanxTW	0.6475	0.6906	0.6547	0.6475	0.6835	0.6475	0.6403	0.6403	0.6403	0.7050	SVML
ECG200	0.7500	0.8600	0.8600	0.7600	0.8600	0.9000	0.7100	0.7900	0.7800	0.9100	MSM_1NN
ECG5000	0.9140	0.9327	0.9331	0.9089	0.9313	0.9258	0.8960	0.9342	0.9309	0.9462	HIVE-COTE
ECGFiveDays	0.7305	0.9187	0.9048	0.7317	0.9419	0.9361	0.6667	0.7108	0.7735	1.0000	LS
EOGHorizontalSignal	0.3343	0.5193	0.5276	0.3398	0.4834	0.5083	0.2376	0.4144	0.4033	-	-
EOGVerticalSignal	0.1906	0.4945	0.4227	0.1989	0.4751	0.4503	0.1713	0.3425	0.3287	-	-
Earthquakes	0.7914	0.7770	0.7914	0.7914	0.7842	**0.8129**	0.7842	0.7770	0.7986	0.7482	RandF
ElectricDevices	0.5300	0.6688	0.6772	0.5359	0.6436	0.5787	0.4765	0.6644	0.6707	0.7992	BOSS
EthanolLevel	0.3760	0.5440	0.5040	0.3680	0.5020	0.4740	0.2920	0.3160	0.3140	-	-
FaceAll	0.5065	0.7337	0.8355	0.4201	0.6941	0.8148	0.4207	0.5077	0.7130	0.9680	SAXVSM

Continued on next page

	Logistic regression			Linear SVC			k-neighbors			Benchmarks	
	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Score	Classifier
Dataset											
FaceFour	0.6023	0.7273	0.7500	0.5682	0.7273	0.7386	0.3182	0.3750	0.3636	1.0000	BOSS
FacesUCR	0.5644	0.8878	0.8766	0.5820	0.8888	0.8756	0.4385	0.7224	0.6941	0.9712	MSM_1NN
FiftyWords	0.3033	0.7582	0.7495	0.3165	0.7385	0.7253	0.2791	0.6154	0.6418	0.8198	EE
Fish	0.2800	0.5600	0.5886	0.3029	0.5200	0.5257	0.2629	0.5429	0.5829	0.9886	ST
FordA	0.7303	0.7773	0.7856	0.7311	0.7652	0.7667	0.7076	0.6318	0.6545	0.9712	ST
FordB	0.5988	0.6358	0.6432	0.5963	0.6259	0.6457	0.5728	0.5889	0.5988	0.9173	LS
FreezerRegularTrain	0.7333	0.9867	0.9832	0.4551	0.9782	0.9807	0.5744	0.9460	0.9498	-	-
FreezerSmallTrain	0.6853	0.9653	0.9705	0.7421	0.9656	0.9684	0.7046	0.7877	0.8109	-	-
Fungi	0.2581	0.9032	0.8172	0.2473	0.8978	0.7957	0.1774	0.1667	0.1989	-	-
GunPoint	0.7533	0.9733	0.9733	0.8333	0.9867	0.9733	0.6067	0.9533	0.9467	1.0000	DDTW_Rn_1NN
GunPointAgeSpan	0.7437	0.9335	0.9051	0.7532	0.9620	0.9114	0.7785	0.9146	0.8829	-	-
GunPointMaleVersusFemale	0.6804	0.9905	0.9905	0.6741	0.9968	0.9842	0.7247	0.9968	0.9937	-	-
GunPointOldVersusYoung	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	-	-
Ham	0.5429	0.7143	0.7333	0.5524	0.6857	0.7143	0.5143	0.6095	0.5714	0.8381	MLP
HandOutlines	0.7784	0.7189	0.7378	0.7054	0.6946	0.7243	0.5919	0.7000	0.6568	0.9324	ST
Haptics	0.3734	0.4773	0.4643	0.3571	0.4675	0.4578	0.3247	0.4091	0.4091	0.5227	ST
Herring	0.6250	**0.7188**	0.6406	0.6250	0.6875	0.6094	0.5469	0.6562	0.6250	0.6875	HIVE-COTE
HouseTwenty	0.7227	0.7899	0.7815	0.7395	0.7815	0.7563	0.7143	0.7311	0.7479	-	-
InlineSkate	0.3000	0.3818	0.3745	0.2927	0.3564	0.3618	0.2636	0.3164	0.3055	0.5618	DD_DTW
InsectEPGRegularTrain	0.8233	0.9960	0.7309	0.8153	1.0000	0.7590	0.8112	0.9920	0.7430	-	-
InsectEPGSmallTrain	0.7671	0.9839	0.7711	0.7390	1.0000	0.7068	0.6586	0.9920	0.6667	-	-
InsectWingbeatSound	0.2409	0.4475	0.4591	0.2338	0.3606	0.3742	0.1510	0.3955	0.3970	0.6561	RandF
ItalyPowerDemand	0.7707	0.9388	0.9300	0.7532	0.9242	0.9155	0.7318	0.9281	0.9378	0.9728	RotF
LargeKitchenAppliances	0.6080	0.7733	0.7547	0.6507	0.7760	0.6987	0.5893	0.6587	0.6613	0.8773	SAXVSM
Lightning2	0.5246	0.8033	0.8525	0.5574	0.7541	0.8197	0.6885	0.7869	0.8033	0.9016	WDTW_1NN
Lightning7	0.4795	**0.8219**	0.7945	0.4658	0.7945	0.7671	0.3836	0.8082	0.7808	0.8082	Flat-COTE
Mallat	0.6908	0.8478	0.8635	0.7441	0.8256	0.8478	0.6682	0.7070	0.6733	0.9761	FS
Meat	0.8167	0.9667	0.9667	0.7667	0.9833	0.9667	0.6333	0.9500	0.9667	1.0000	MLP
MedicalImages	0.6250	0.7211	0.7224	0.6289	0.7013	0.6737	0.5697	0.6408	0.6526	0.7776	HIVE-COTE
MelbournePedestrian	0.5518	0.8261	0.8376	0.5245	0.8294	0.8608	0.5669	0.8273	0.8612	-	-
MiddlePhalanxOutlineAgeGroup	0.3117	0.3312	0.2662	0.3701	0.3831	0.3766	0.4545	0.4870	0.4805	0.6429	ST
MiddlePhalanxOutlineCorrect	0.7388	0.7045	0.6357	0.6564	0.6667	0.6804	0.6598	0.5739	0.5498	0.8316	HIVE-COTE

Continued on next page

	Logistic regression			Linear SVC			k-neighbors			Benchmarks	
	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Score	Classifier
Dataset											
MiddlePhalanxTW	0.5584	0.5714	0.5649	0.5649	0.5779	0.5584	0.5260	0.5260	0.5584	0.6299	RotF
MixedShapesRegularTrain	0.7175	0.8915	0.8961	0.7406	0.8891	0.8882	0.7109	0.8532	0.8614	-	-
MixedShapesSmallTrain	0.6520	0.8099	0.8243	0.4425	0.8198	0.8186	0.5386	0.7753	0.7629	-	-
MoteStrain	0.6701	0.8698	0.8794	0.6494	0.8610	0.8690	0.7228	0.7971	0.7660	0.9369	Flat-COTE
NonInvasiveFatalECGThorax1	0.5394	0.8784	0.8799	0.4865	0.8748	0.8784	0.3028	0.7852	0.7847	0.9496	ST
NonInvasiveFatalECGThorax2	0.5654	0.9120	0.9028	0.5751	0.9191	0.9125	0.3985	0.8377	0.8402	0.9511	ST
OSULeaf	0.5331	0.6488	0.6488	0.5661	0.6364	0.6074	0.5207	0.5537	0.5331	0.9793	HIVE-COTE
OliveOil	0.8000	0.9000	**0.9333**	0.8333	0.9333	0.9000	0.6333	0.7667	0.7333	0.9000	NB
PhalangesOutlinesCorrect	0.6573	0.7179	0.6993	0.6538	0.7110	0.7121	0.6818	0.6469	0.6643	0.8601	RotF
Phoneme	0.1725	0.1577	0.1741	0.1572	0.1324	0.1498	0.0891	0.1145	0.1229	0.3824	HIVE-COTE
PigAirwayPressure	0.1154	0.1971	0.1346	0.1202	0.1875	0.1346	0.0769	0.0817	0.0769	-	-
PigArtPressure	0.0577	0.2212	0.1394	0.0721	0.2067	0.1490	0.0721	0.1202	0.0769	-	-
PigCVP	0.0288	0.1106	0.0769	0.0385	0.1250	0.0529	0.0096	0.1154	0.0481	-	-
Plane	0.8667	0.9619	0.9714	0.9524	1.0000	1.0000	0.8095	0.8762	0.9143	1.0000	DTW_R1_1NN
PowerCons	0.7667	0.9500	0.9500	0.7667	0.9500	0.9389	0.7611	0.9222	0.8778	-	-
ProximalPhalanxOutlineAgeGroup	0.8439	0.8634	0.8585	0.8390	0.8683	0.8634	0.8488	0.8537	0.8439	0.8683	RandF
ProximalPhalanxOutlineCorrect	0.8179	0.8729	0.8316	0.8213	0.8591	0.8385	0.8076	0.7835	0.7801	0.9003	SVMQ
ProximalPhalanxTW	0.7902	0.8098	0.8000	0.7951	0.8098	0.7805	0.7415	0.7610	0.7659	0.8244	SVML
RefrigerationDevices	0.4933	0.5520	0.5333	0.4880	0.5200	0.4987	0.4107	0.4853	0.4560	0.6533	SAXVSM
Rock	0.5400	0.6000	0.6000	0.5800	0.6200	0.6200	0.5600	0.6200	0.5000	-	-
ScreenType	0.3520	0.4693	0.4880	0.4000	0.4720	0.4480	0.4320	0.4293	0.3840	0.5893	HIVE-COTE
SemgHandGenderCh2	0.7017	0.8083	0.8133	0.7267	0.7983	0.8000	0.6550	0.8717	0.8600	-	-
SemgHandMovementCh2	0.3644	0.6044	0.6200	0.3733	0.6867	0.6622	0.3600	0.7733	0.7467	-	-
SemgHandSubjectCh2	0.3911	0.8111	0.8067	0.4111	0.8333	0.8289	0.3644	0.8156	0.7978	-	-
ShapeletSim	0.7722	0.5333	0.7278	0.7778	0.5389	0.7167	0.7000	0.5278	0.5889	1.0000	FS
ShapesAll	0.4333	0.7950	0.8033	0.4483	0.7850	0.7817	0.4117	0.6383	0.6267	0.9083	BOSS
SmallKitchenAppliances	0.6933	0.7013	0.6853	0.6800	0.7013	0.5867	0.6320	0.6853	0.6720	0.8533	HIVE-COTE
SmoothSubspace	0.7600	0.9933	0.9400	0.7333	0.9800	0.9333	0.6800	0.9133	0.8200	-	-
SonyAIBORobotSurface1	0.8186	0.8702	0.8669	0.8303	0.8619	0.8669	0.6705	0.6023	0.5308	0.9301	NB
SonyAIBORobotSurface2	0.9297	0.8814	0.8741	0.9360	0.8814	0.8657	0.9035	0.7754	0.8048	0.9517	Flat-COTE
StarLightCurves	0.9161	0.9213	0.9484	0.9302	0.9117	0.9415	0.9137	0.9040	0.9150	0.9815	HIVE-COTE
Strawberry	0.8459	0.9351	0.9405	0.8243	0.9378	0.9351	0.7919	0.8919	0.8919	0.9757	BOSS

Continued on next page

	Logistic regression			Linear SVC			k-neighbors			Benchmarks	
	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Lead-lag	Time-joined	Time-indexed	Score	Classifier
Dataset											
SwedishLeaf	0.5264	0.8928	0.8848	0.6080	0.8752	0.8688	0.4080	0.8096	0.8160	0.9552	Flat-COTE
Symbols	0.7829	0.9387	0.9407	0.8171	0.9447	0.9417	0.6955	0.8070	0.8171	0.9739	WDDTW_1NN
SyntheticControl	0.9333	0.9667	0.9833	0.9200	0.9667	0.9833	0.8700	0.9033	0.9833	1.0000	Flat-COTE
ToeSegmentation1	0.6667	0.7982	0.7939	0.6140	0.7939	0.7675	0.6184	0.7325	0.6974	0.9825	HIVE-COTE
ToeSegmentation2	0.5462	0.6615	0.6846	0.6538	0.6462	0.6538	0.6923	0.7692	0.8769	0.9615	BOSS
Trace	0.9400	1.0000	1.0000	0.9600	1.0000	1.0000	0.8500	0.8100	0.7800	1.0000	DTW_R1_1NN
TwoLeadECG	0.9517	0.9210	0.8982	0.9526	0.9166	0.8806	0.8586	0.7989	0.8051	0.9974	ST
TwoPatterns	0.9995	1.0000	1.0000	0.9992	1.0000	0.9995	0.9770	0.9992	0.9892	1.0000	DTW_R1_1NN
UMD	0.9236	0.9583	0.9653	0.9167	0.9514	0.9653	0.7986	0.7361	0.7569	-	-
UWaveGestureLibraryAll	0.6011	0.9525	0.9528	0.5497	0.9408	0.9442	0.5251	0.8738	0.8716	0.9685	EE
UWaveGestureLibraryX	0.4844	0.8141	0.8157	0.4832	0.7934	0.7878	0.4634	0.7460	0.7437	0.8398	HIVE-COTE
UWaveGestureLibraryY	0.4894	0.7194	0.7200	0.4419	0.6778	0.6890	0.4654	0.6644	0.6619	0.7655	HIVE-COTE
UWaveGestureLibraryZ	0.5137	0.7426	0.7440	0.5181	0.7306	0.7178	0.4913	0.6602	0.6597	0.7831	HIVE-COTE
Wafer	0.9080	0.9911	0.9961	0.9010	0.9914	0.9968	0.9661	0.9833	0.9862	1.0000	ST
Wine	0.5556	0.8519	0.8333	0.6296	0.8519	0.8704	0.5556	0.6667	0.5370	0.9630	SAXVSM
WordSynonyms	0.3511	0.6818	0.6865	0.3605	0.6520	0.6708	0.2539	0.6003	0.5956	0.7790	EE
Worms	0.5974	0.5455	0.5325	0.6104	0.5584	0.5325	0.5455	0.5455	0.4805	0.7532	ACF
WormsTwoClass	0.6623	0.6753	0.6753	0.6494	0.6753	0.6883	0.6623	0.6104	0.6623	0.8312	ST
Yoga	0.5907	0.8087	0.8057	0.5877	0.8110	0.8040	0.6633	0.7607	0.7633	0.9183	BOSS

References

[1] T. J. Lyons, "Differential equations driven by rough signals," *Rev. Mat. Iberoamericana*, vol. 14, no. 2, pp. 215–310, 1998.

[2] P. K. Friz and M. Hairer, *A course on rough paths*. Universitext, Springer, Cham, 2014. With an introduction to regularity structures.

[3] K.-T. Chen, "Iterated integrals and exponential homomorphisms," *Proc. London Math. Soc. (3)*, vol. 4, pp. 502–512, 1954.

[4] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The UCR Time Series Archive," *arXiv e-prints*, p. arXiv:1810.07758, Oct 2018.

[5] J. M. Lee, *Introduction to smooth manifolds*, vol. 218 of *Graduate Texts in Mathematics*. Springer, New York, second ed., 2013.

[6] B. Hambly and T. Lyons, "Uniqueness for the signature of a path of bounded variation and the reduced path group," *Ann. of Math. (2)*, vol. 171, no. 1, pp. 109–167, 2010.

[7] T. Lévy, "The master field on the plane," *arXiv e-prints*, p. arXiv:1112.2452, Dec 2011.

[8] T. J. Lyons, M. Caruana, and T. Lévy, *Differential equations driven by rough paths*, vol. 1908 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007.

[9] P. K. Friz and N. B. Victoir, *Multidimensional stochastic processes as rough paths*, vol. 120 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2010. Theory and applications.

[10] S. M. LaValle, *Planning algorithms*. Cambridge University Press, Cambridge, 2006.

[11] T. Lyons *et al.*, "CoRoPa Computational Rough Paths (software library)." `https://coropa.sourceforge.io/`, 2010.

[12] M. R. Bremner, I. R. Hentzel, and L. A. Peresi, "Dimension formulas for the free nonassociative algebra," *Resenhas*, vol. 6, no. 2-3, pp. 141–151, 2004.

[13] J. Reizenstein, "Calculation of Iterated-Integral Signatures and Log Signatures," *arXiv e-prints*, p. arXiv:1712.02757, Dec 2017.

[14] H. Boedihardjo, X. Geng, T. Lyons, and D. Yang, "The signature of a rough path: uniqueness," *Adv. Math.*, vol. 293, pp. 720–737, 2016.

[15] J. Diehl, "The signature of iterated integrals: algebra, analysis and machine learning," 2018.

[16] I. Chevyrev and A. Kormilitzin, "A Primer on the Signature Method in Machine Learning," *arXiv e-prints*, p. arXiv:1603.03788, Mar 2016.

[17] D. Levin, T. Lyons, and H. Ni, "Learning from the past, predicting the statistics for the future, learning an evolving system," *arXiv e-prints*, p. arXiv:1309.0260, Sep 2013.

[18] G. Flint, B. Hambly, and T. Lyons, "Discretely sampled signals and the rough Hoff process," *arXiv e-prints*, p. arXiv:1310.4054, Oct 2013.

[19] L. Gergely Gyurkó, T. Lyons, M. Kontkowski, and J. Field, "Extracting information from the signature of a financial data stream," *arXiv e-prints*, p. arXiv:1307.7244, Jul 2013.

[20] J. Reizenstein and B. Graham, "The iisignature library: efficient calculation of iterated-integral signatures and log signatures," *arXiv e-prints*, p. arXiv:1802.08252, Feb 2018.

[21] J. de Jong and E. Mansfield, "Math.js: An advanced mathematics library for JavaScript," *Computing in Science and Engg.*, vol. 20, pp. 20–32, Feb. 2018.

[22] A. Röß ler, "Runge-Kutta methods for the strong approximation of solutions of stochastic differential equations," *SIAM J. Numer. Anal.*, vol. 48, no. 3, pp. 922–952, 2010.

[23] M. J. Aburn, "sdeint, numerical integration of Ito or Stratonovich SDEs (software library)." `https://pypi.org/project/sdeint/`, 2017.

[24] F. Alimoglu, "Combining multiple classifiers for pen-based handwritten digit recognition," Master's thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University, 1996.

[25] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[26] F. J. Király and H. Oberhauser, "Kernels for sequentially ordered data," *arXiv e-prints*, p. arXiv:1601.08169, Jan 2016.

[27] I. Pérez, "Rough path theory and signatures applied to quantitative finance." `https://www.quantstart.com/articles/rough-path-theory-and-signatures-applied-to-quantitative-finance-part-3`, 2017. Retrieved: 2019-06-10.

[28] B. Graham, "Sparse arrays of signatures for online character recognition," *arXiv e-prints*, p. arXiv:1308.0371, Aug 2013.

[29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn:

Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[30] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: The collective of transformation-based ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, pp. 1–1, 09 2015.

[31] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, (New York, NY, USA), pp. 289–297, ACM, 2012.

[32] J. Lines, S. Taylor, and A. Bagnall, "Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles," *ACM Trans. Knowl. Discov. Data*, vol. 12, pp. 52:1–52:35, July 2018.

[33] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The UEA multivariate time series classification archive, 2018," *arXiv e-prints*, p. arXiv:1811.00075, Oct 2018.

[34] M. H. Protter and C. B. Morrey, Jr., *A first course in real analysis*. Undergraduate Texts in Mathematics, Springer-Verlag, New York, second ed., 1991.

[35] J. Hilgert and K.-H. Neeb, *Structure and geometry of Lie groups*. Springer Monographs in Mathematics, Springer, New York, 2012.

[36] A. Bonfiglioli and R. Fulci, "A new proof of the existence of free lie algebras and an application," *ISRN Algebra*, vol. 2011, pp. 1–11, 2011.