Signature Methods for Time Series Data

Sam Morley
DataSig

DataSig

A rough path between
mathematics and data science

University of Oxford
The Alan Turing Institute
Imperial College London
UCL
UKRI Engineering and Physical Sciences Research Council

Who is the I see before me?

- Research software engineer on the DataSig project.
- Python enthusiast. Pure mathematician at heart.
- Author of “Applying Math with Python”.
- Formerly lecturer in mathematics.

Controlled differential equations? That sounds scary!

Basic idea

Given an input signal path X_t, find a path Y_t that satisfies

$$\mathrm{d}Y_t = f(Y_t)\,\mathrm{d}X_t.$$

This a is *controlled differential equation*, the path X_t is the *control path* or *driver path* and Y_t is the response path.

It turns out that an important object in solving such equations is the *signature* of the path X_t.

Signature sounds less scary

A *signature* is an abstract description of a path X_t in the *tensor algebra*

$$\mathrm{T}(\mathbb{R}^d) = \bigoplus_{k=0}^{\infty} (\mathbb{R}^d)^{\otimes k}$$

Think of this as the space of sequences of the form

$$(s^{(0)}, s_1^{(1)}, \dots, s_d^{(1)}, s_{1,1}^{(2)}, \dots, s_{d,d}^{(2)}, s_{1,1,1}^{(3)}, \dots, s_{d,d,d}^{(3)}, \dots, s_{i_1,\dots,i_n}^{(n)}, \dots)$$

The signature of X_t over an interval $a \leq t \leq b$ is the element of $\mathrm{T}(\mathbb{R}^d)$ defined by the following formulae

$$
\begin{aligned}
S^{(0)} &= 1, \\
S_i^{(1)} &= X_b^i - X_a^i && (1 \leq i \leq d), \\
S_{i,j}^{(2)} &= \int_a^b \int_a^u \mathrm{d}X_v^i \, \mathrm{d}X_u^j && (1 \leq i, j \leq d), \\
&\vdots \\
S_{i_1,\ldots,i_n}^{(n)} &= \int_{0<u_1<\cdots<u_n<T} \mathrm{d}X_{u_1}^{i_1} \ldots \mathrm{d}X_{u_n}^{i_n} && (1 \leq i_1, \ldots, i_n \leq d) \\
&\vdots
\end{aligned}
$$

Here X_t^i is the ith component of the path X_t at t.

The signature of X_t over an interval $a \leq t \leq b$ is denoted $\mathrm{S}(X_t)_{a,b}$.

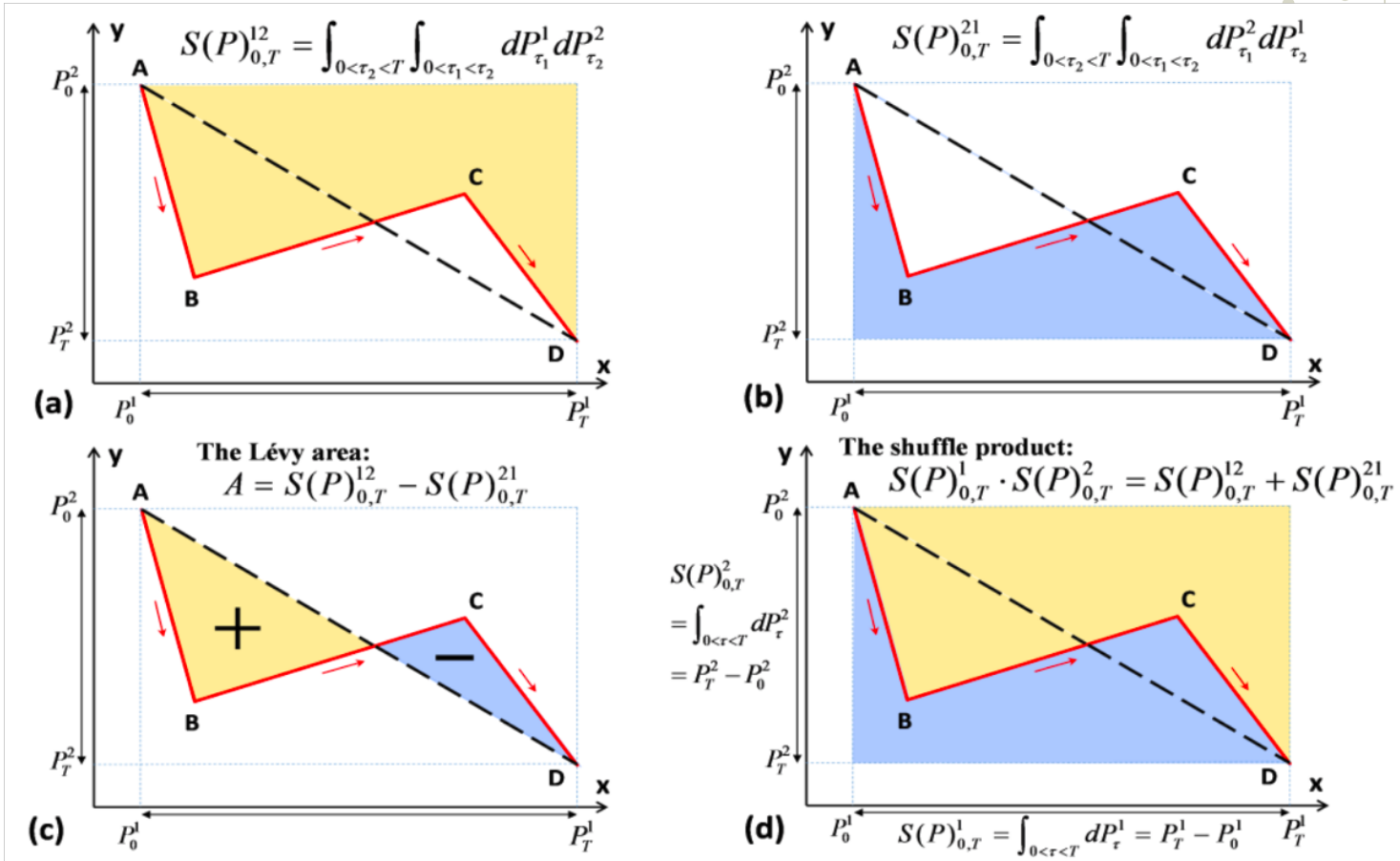
Properties

1. Invariant to reparametrisation.
2. Translation invariant - paths that start at different points but change in the same way have the same signature;
3. Respects concatenation: If $a < u < b$ then

$$\mathrm{S}(X_t)_{a,b} = \mathrm{S}(X_t)_{a,u} \otimes \mathrm{S}(X_t)_{u,b}$$

Since the full signature of a path is not practical for calculations, we usually truncate the signature to some level $n \geq 1$, written $\mathrm{S}^{(n)}(X_t)_{a,b}$.

Ok, but what does all this really mean?



Note that the signature is a property of the (continuous) path, not the data points.

How do I actually compute signatures?

Our path data often arrives as a sequence of increments:

$$\mathbf{x}_r = (x_r^1, x_r^2, \dots, x_r^d) \qquad (r = 1, 2, \dots, N).$$

In this case we can compute the (truncated signature) by embedding these increments in $\mathrm{T}^{(n)}(\mathbb{R}^d)$ as

$$I_r = (0, x_r^1, \dots, x_r^d, 0, \dots) \in \mathrm{T}^{(n)}(\mathbb{R}^d)$$

and computing the product of exponentials

$$\mathrm{S}^{(n)}(X_t)_{t_1,t_N} = \prod_{r=1}^{N} \exp(I_r) \tag{1}$$

where exp is taken in the truncated tensor algebra.

I don't understand, show me an example

Suppose we observed the following increments of a path in $\mathbb{R}^2$:

$$\mathbf{x}_1 = (1, 2) \quad \mathbf{x}_2 = (3, 4)$$

(for example, if we had sampled $(0, 0)$, $(1, 2)$, $(4, 6)$ as points on our path.)
Then the exponential of the increments (in the tensor algebra truncated at level 2) are

$$\exp(I_1) = (1, 1, 2, 0.5, 1, 1, 2)$$
$$\exp(I_2) = (1, 3, 4, 4.5, 6, 6, 8)$$

and the signature is

$$(1, 1 + 3, 2 + 4, 0.5 + 3 + 4.5, 0.5 + 4 + 6, 1 + 4 + 6, 2 + 8 + 8).$$

Great, but what does this all mean?

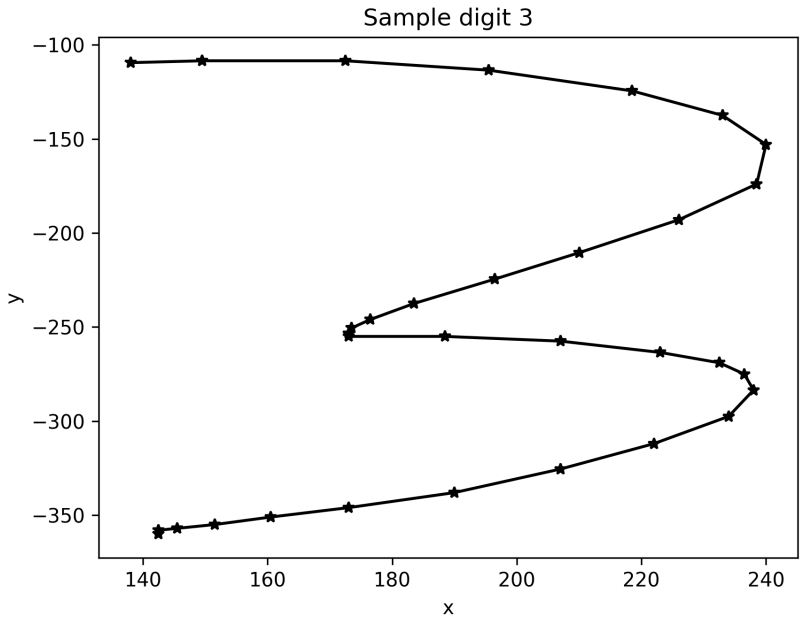
A theorem tells us that (continuous) functions on paths are equivalent to linear functions on signatures, so anything that can be "learn" about paths can be learned from their signatures instead.

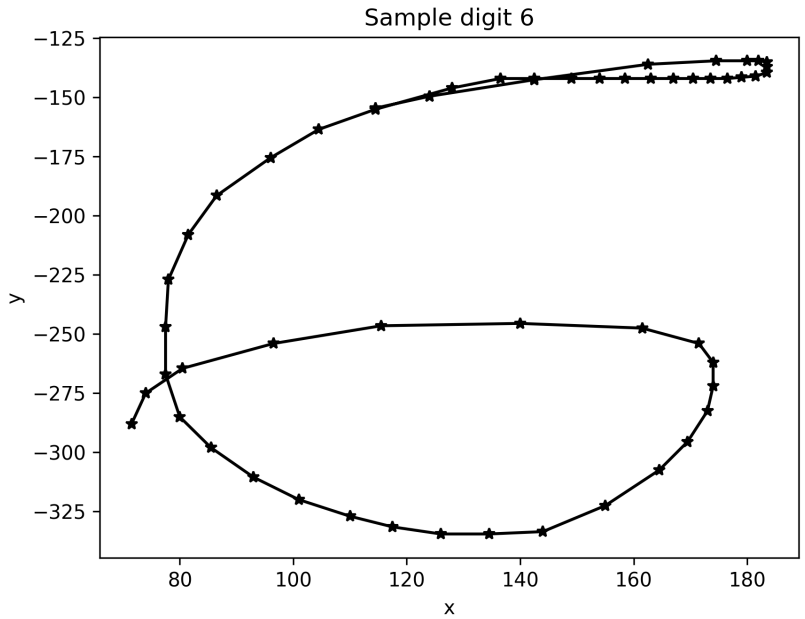
In particular, we can replace "path data" with "signature data" for inference tasks without any real loss of generality.

The dimension of a signature is only depends on the dimension of the underlying space and the level at which the signature is truncated.

The size of the signature does not depend on the number of samples or increments used to compute it.

Can you give me a real example?





We have a dataset with a collection of handwritten digit paths. This is really a (short) time series in two coordinates.
Let's try to identify the digit 3 using signatures.

Note

The approach I'm about to use is not very scientific and probably not effective, it is just a quick demo of how I might use a signature.

Rough procedure:

1. Compute signatures of each of the paths describing a digit 3 from the dataset;
2. Compute the "expected signature" for the digit 3;
3. Compare signatures of new digit paths to this expected signature to decide if it is also a 3.

There are several packages you can use to compute signatures: esig, iisignature, signatory. (I maintain the esig package, so I'm going to use this.)

Use the `stream2sig` function from esig to compute the signature; it takes the stream as a $N \times d$ array where N is the number of samples and d is the dimension of the space (here, 2), and the depth (8 in this case).

The result is a Numpy array with 511 entries (features) for each input sample path.

The first few terms of the expected signature of digit 3 paths is

```
1.00 -0.04  1.88  0.06  2.54 -2.64  1.77  0.00
```

Now we can look at the signature of a new digit. The first few terms are as follows:

```
1.00 -0.77  1.34  0.30 -1.87  0.84  0.89 -0.08
```

These don't look to be the same kind of digit: The maxmimum modulus difference between these two the 5.27, which is relatively large.

In fact this new path is from the 6 figure from the earlier slide.

What can I do with signatures?

We've used signatures in many, very different applications including:

- Handwriting recognition* - example of classification on MNIST dataset
- Sepsis detection* - example using MIMIC-III data set
- Human action recognition*
- Drone identification*
- Natural language processing*
- Analysing energy usage data
- Analysis of radio frequency data

* has example notebook on our website

Where can I find more information?

You can find more information on our website: `https://datasig.ac.uk`.
See the research tab for

- publications and preprints
- examples in self-contained Jupyter notebooks.

The esig package is on PyPI and Github `https://github.com/datasig-ac-uk/esig`.

My contacts are:

- Email: `sam.morley@maths.ox.ac.uk`
- Website: `https://inakleinbottle.com`
- GitHub: @inakleinbottle