Signatures for modeling drawdown paths: a primer

Emiel Lemahieu

Ghent University, InvestSuite

Abstract

This primer introduces signatures, a *graded summary* of path structured data or streams, in a more introductory way. Signatures arose from the geometry of K.T. Chen and the algebra of Y.C. Young and have played a pivotal role in T. Lyons' theory of *rough paths* [1]. Signatures are non-parametric transformations of paths that preserve nice geometrical features and have properties that allow for a parsimonious encoding or representation of paths in a machine learning context.

Keywords: Signatures, Drawdowns, Machine Learning

1. Context

1.1. Generation of realistic drawdown samples

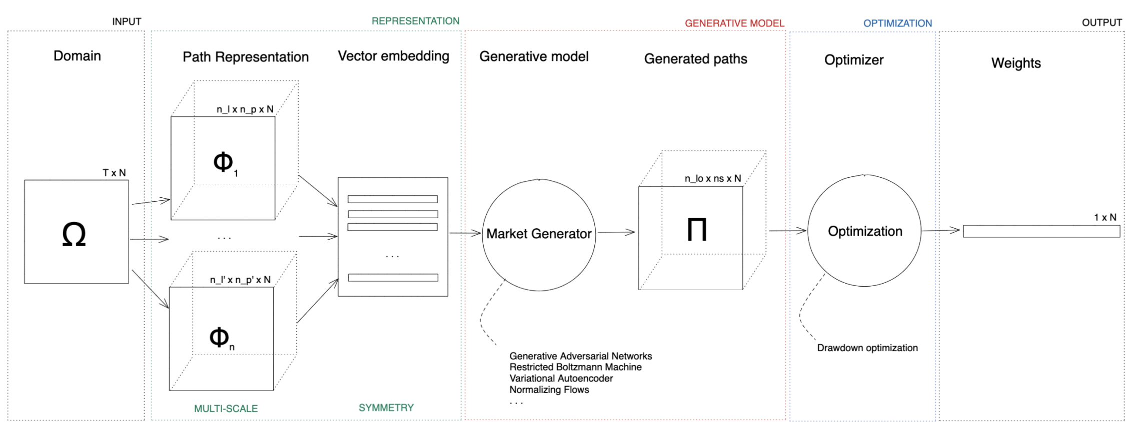

Figure 1: Path embeddings in a broader optimization framework

Figure 1 sketches an overview of the role of path representation in a machine learning-based drawdown optimization context. Before we introduce paths more formally below, let us discuss where paths live in such a problem. Firstly, there is the physical domain Ω in which the input data is structured. Think in a financial context about market data, e.g. the timeseries of the T last historical prices for N financial instruments. These can be seen as paths in $\mathbb{R}^{N+1}$ with N evolutions of asset prices and the evolution of time over [0,T]. However, when we want to optimize for some features of these timeseries (e.g. minimize drawdowns, see below), it is not generally (and typically not) the case that the *signal* lives in the physical domain. We need to represent the path in a domain that allows to find traces of the processes that generate these features, called a signal domain.

In line with the upcoming literature on geometric learning [2], this representation step is in the above figure split up according to two principles: multi-scale representation and symmetry. The first principle implies that representation of different granularities of the input grid (finer and more coarse-grained time intervals) can help the learning model to find traces of these processes. The second principle means that Ω is a rich domain with hierarchical and symmetrical relationships that can be exploited. For instance, when two instruments almost perfectly behave in accordance to a third (whatever your similarity metric), the T x 3 observations can be reduced to a T x 1 series and 2 similarity metrics that provide an inverse mapping. The result of the representation step is a so-called vector *embedding*, which represents the complex input paths as a collection of real-valued vectors that will serve as training samples to the learning pipeline. For this embedding we propose *signatures* below, and discuss the reasons for doing so.

The learning pipeline in our later applications is a (neural) generative model, such as a Generative Adversarial Network (GAN), a Restricted Boltzmann Machine (RBM) or a Variational Autoencoder (VAE). These learning models represent the relationships between the vectors in a parsimonious way according to some learning objective. The resulting (latent) representation can then be used to generate new samples of vectors, which can be inversely transformed into genuinely new paths that are statistically indistinguishable from the original samples, according to some similarity metric. This is the general idea but not the focal point of this primer.

Next, these samples are fed to a portfolio optimizer. Our application will try to find a set of portfolio weights w that minimize the expected drawdown as defined as the expected deviation of the portfolio path $w\mathbf{\Pi}_t$ from a monotonic growing path m_t, or more specifically our optimizer tries to minimize

$$\begin{array}{ll} \min_w & \mathbb{E}(\xi(w)) \\ \text{s.t.} & \xi = m_t - w\mathbf{\Pi}_t \\ & m_t \geq m_{t-1} \\ & w\mathbf{I}^N = 1 \end{array} \tag{1}$$

It is immediately clear that the path structure of $\mathbf{\Pi}$ is critical because the drawdown series $\xi_t,\ t \in [0, T_s]$ is dependent on the local maxima m_t. Compared to static return-based optimizers (e.g. mean-variance quadratic utility frameworks), drawdown optimizers require a dynamic functional ξ_t, first formally introduced by Chekhlov et al. [3] (Definition 3.1):

$$\xi = (\xi_1, \xi_2, ..., \xi_T), \xi_t = \max_{t_k < t}(P_{t_k}) - P_t \tag{2}$$

with P_t in our notation being equivalent to $w\mathbf{\Pi}_{i,t}$, or the timeseries of portfolio values and $i \in n_s$ a particular scenario. The latter is central in our notion of expected drawdown function $\mathbb{E}(\xi(w))$.

1.2. Paths

In section 1 we have introduced the kind of paths we are interested in and why the path structure is especially important in our application. In this section we more formally define what we mean by a path.

A path is one of the most basic elements in financial theory, but it is usually not well thought of as a path, i.e. transformations like calculating returns essentially transform the data from path space to distribution space (i.e. a profit-and-loss distribution or P&L). Although econometrics focuses on sequences of returns and its variation, it models distributions rather than paths. The same holds for risk measures such as VaR, CVaR - both cut-offs of the P&L distribution - and portfolio optimization tools such as classical mean-variance optimization and risk-based methods such as popular min vol, risk parity and maximum diversification approaches. Let us thus first start by defining a path in general and then make it specific for our context explained in 1.

A path γ in $\mathbb{R}^d$ is a continuous map from some interval $[a, b]$ to $\mathbb{R}^d$, written as $\gamma : [a, b] \to \mathbb{R}^d$. We use subscript $\gamma_t = \gamma(t)$ to denote the time as parameter

$t \in [a, b]$, and usually for convenience we take $a = 0, b = T, t \in [0, T]$. In our examples we will assume that paths are piecewise linear, smooth and differentiable, i.e. the path has derivates of all orders over [0,T][1]. A path in d dimensions can be written as

$$\gamma : [0, T] \to \mathbb{R}^d, \gamma_t = \{\gamma_t^1, \gamma_t^2, ..., \gamma_t^d\} \tag{3}$$

A simple example of such a path in $\mathbb{R}^2$ would be the following path $\gamma : [0, 5] \to \mathbb{R}^2$, $\gamma_t = \{t, \gamma_t^2\} = \{\{0,\ 1,\ 2,\ 3,\ 4,\ 5\},\ \{1,\ 2,\ 1,\ 3\ ,2,\ 5\}\}$.

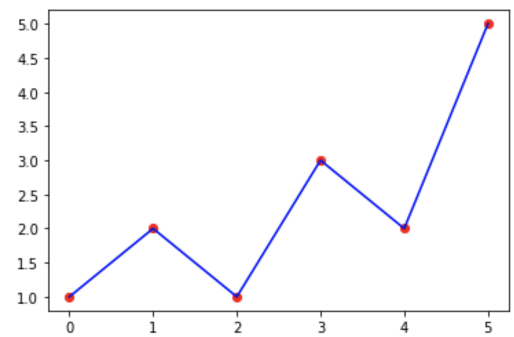

Figure 2: Example of a path in $\mathbb{R}^2$

1.3. Path integrals

For a path $\gamma : [0, T] \to \mathbb{R}$ and a function $f : \mathbb{R} \to \mathbb{R}$, the path integral of γ against f is defined by

$$\int_0^T f(\gamma_t) d\gamma_t = \int_0^T f(\gamma_t) \frac{d\gamma_t}{dt} dt = \int_0^T f(\gamma_t) \dot{\gamma}_t dt \tag{4}$$

in which context f is called a *1-form*. The last integral is the 'usual' Riemann integral. Note that f itself is a real-valued path on $[0, T]$. This is a special case of the Riemann-Stieltjes integral of one path against another [4].

In general, one can integrate any two paths on $[0, T]$, $\phi : [0, T] \to \mathbb{R}, \gamma : [0, T] \to \mathbb{R}$, against one another:

$$\int_0^T \phi_t d\gamma_t = \int_0^T \phi_t \dot{\gamma}_t dt \tag{5}$$

[1] However, the same properties hold for general (rough) paths of bounded variation, see [4].

2. Signatures

2.1. Definition

Now that we have defined a path in $\mathbb{R}^d$ and path integrals, let us consider a particular path integral defined for any *single* index $i \in \{1, 2, ..., d\}$:

$$S(\gamma)^i_{0,T} = \int_0^T d\gamma^i = \gamma^i_T - \gamma^i_0 \tag{6}$$

which is the increment of the path along the dimension i in $\{1, 2, ..., d\}$. Now for any *pair* of indexes $i, j \in \{1, 2, ..., d\}$, let us define:

$$S(\gamma)^{i,j}_{0,T} = \int_0^T \int_0^{t_j} d\gamma^i d\gamma^j \tag{7}$$

and likewise for *triple* indices in $i, j, k \in \{1, 2, ..., d\}$:

$$S(\gamma)^{i,j,k}_{0,T} = \int_0^T \int_{t_k}^{t_j} \int_0^{t_k} d\gamma^i d\gamma^j d\gamma^k \tag{8}$$

and we can continue for the collection of indices $i_1, i_2, ..., i_k \in \{1, 2, ..., d\}$:

$$S(\gamma)^{i_1,i_2,...,i_k}_{0,T} = \int_0^T ... \int_{t_2<t1} \int_0^{t_1} d\gamma^{i_1} d\gamma^{i_2} ... d\gamma^{i_k} \tag{9}$$

which we call the k-fold iterated integral of γ along $\{i_1, i_2, ..., i_k\}$.

Definition 2.1 (Signature). *From [4]: The signature of a path $\gamma : [0, T] \rightarrow \mathbb{R}$ denoted $S(\gamma)_{0,T}$ is the collection (infinite series) of all the iterated integrals of γ. Formally, $S(\gamma)_{0,T}$ is the sequence of real numbers*

$$S(\gamma)_{0,T} = (1, S(X)^1_{0,T}, S(X)^2_{0,T}, ..., S(X)^d_{0,T}, S(X)^{1,1}_{0,T}, S(X)^{1,2}_{0,T}, ...) \tag{10}$$

where the zeroth term is 1 by convention and the superscript runs along the set of multi-indices:

$$W = \{(i_1, i_2, ..., i_k) | k \geq 1; i_1, i_2, ..., i_k \in \{1, 2, ..., d\}\} \tag{11}$$

In other words, the signature is the collection of all the iterated integrals consisting of any combination of indices in d to any length of combination,

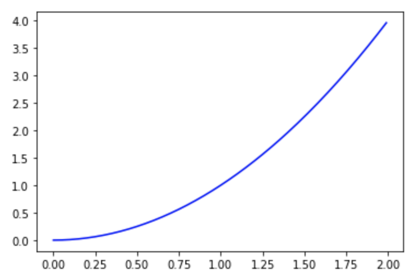

Figure 3: Example of a path in $\mathbb{R}^2$

hence an infinite series. However, it is important to note that these signatures are ordered along this length, which is called the *order* or *level* of the signature.

We often consider the M-th level truncated signature, defined as the finite collection of all terms where the superscript is of max length M:

$$S_M(\gamma) = (1, S^1(\gamma), S^2(\gamma), ..., S^M(\gamma)) \tag{12}$$

where $S^k(\gamma)$ denotes all the signature terms of order k, e.g.

$$S^1(\gamma) = (S(\gamma)^1, S(\gamma)^2, ..., S(\gamma)^d) \tag{13}$$

$$S^2(\gamma) = (S(\gamma)^{1,1}, S(\gamma)^{1,2}, ..., S(\gamma)^{d,d}) \tag{14}$$

A very simple example (Figure 2.1) would be the path $\gamma : [0, 2] \to \mathbb{R}^2$, $\gamma_t = \{\gamma_t^1, \gamma_t^2\} = \{t, f(t)\} = \{t, t^2\}$, which corresponds to a quadratically growing path through the origin. Below are the signature terms numerically up till M = 2.

$$S(\gamma)^1_{0,2} = \int_0^2 d\gamma^1 = \int_0^2 dt = 2 \tag{15}$$

$$S(\gamma)^2_{0,2} = \int_0^2 d\gamma^2 = \int_0^2 2tdt = 4 \tag{16}$$

$$S(\gamma)^{1,1}_{0,2} = \int_0^2 \int_0^{t_2} d\gamma^1 d\gamma^1 = \int_0^2 \int_0^{t_2} dt_1 dt_2 = 2 \tag{17}$$

$$S(\gamma)^{1,2}_{0,2} = \int_0^2 \int_0^{t_2} d\gamma^1 d\gamma^2 = \int_0^2 \int_0^{t_2} 2t_2 dt_1 dt_2 = 16/3 \tag{18}$$

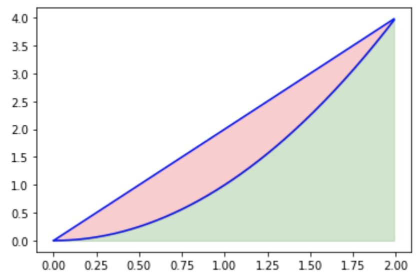

Figure 4: Levy area (red)

$$S(\gamma)^{2,1}_{0,2} = \int_0^2 \int_0^{t_2} d\gamma^2 d\gamma^1 = \int_0^2 \int_0^{t_2} 2t_1 dt_1 dt_2 = 8/3 \tag{19}$$

$$S(\gamma)^{2,2}_{0,2} = \int_0^2 \int_0^{t_2} d\gamma^2 d\gamma^2 = \int_0^2 \int_0^{t_2} 2t_2 dt_2 = 8 \tag{20}$$

2.2. Properties of the signature

2.2.1. Geometric interpretation of the terms

As we discussed in equation (6) the first order terms are the increments of the paths, $S(X)^1 = 2$ over time, and $S(X)^2 = 4$ over the 2^{nd} dimension, e.g. the price axis. In a financial context, this would have the interpretation of *drift*.

The second order terms can be interpreted as a measure for variation (see section 2.4). More specifically, the combination of cross-terms measures the Levy area L, defined as the area between the chord connecting the first and the last point and the path:

$$L = \frac{1}{2}(S(\gamma)^{1,2} - S(\gamma)^{2,1}) \tag{21}$$

In the example:

$$L = \frac{1}{2}(\frac{16}{3} - \frac{8}{3}) = \frac{4}{3} \tag{22}$$

Given the simplicity of the example, this can be verified through calculating the area of the triangle Red + Green in Figure 4 (i.e. half the square of the increments $2 * 4/2 = 4$) and subtracting the Green area $\int_0^2 t^2 dt = 8/3$ resulting in Red - Green $= 4/3$. Clearly the Levy area measures the deviation from the drift over the path.

2.2.2. *Factorial decay*

One key property of signatures is factorial decay, which makes it a *graded summary* of paths.

As an analogue to the distributional setting (cf. 1.2) consider the well-known principal component analysis (PCA). In PCA we use linear combinations of the data to decompose it into its components that maximise the variance of the data set. It is equivalent to the eigendecomposition of the covariance matrix of the data set. A key feature is that we commonly see exponential decay or rate decay, namely that the sorted absolute values of the eigenvalues of the covariance matrix of $\Omega : \mathbb{R}^{T \times N}$ decay fast enough, i.e. the j^{th} largest coefficient $|\beta|_j \leq Aj^{-a}, a \geq 1/2, \forall j$ and constants a and A do not depend on the dimension T. The latter implies that the first N components typically already explain a vast part of the shared variance in the data set.

Informally, this intuition can be applied to paths as well. Lyons [1] shows that for paths of bounded variation[2] the following similar norm can be imposed on the signature terms (with $1 \leq i_1, ..., i_n \leq d$):

$$||\int ... \int d\gamma^{i_1} d\gamma^{i_2} ... d\gamma^{i_n}|| \leq \frac{||\gamma_1^n||}{n!} \tag{23}$$

with

$$||\gamma||_1 = \sup_{t_i \subset [0,T]} \sum_i |\gamma_{t_{i+1}} - \gamma_{t_i}| \tag{24}$$

where we take the supremum over all partitions of [0,T].

This theorem proven in [1] guarantees that higher-order terms of the signature have factorial decay, i.e. that the order of signatures imply a graded summary of the path, from global to more local characteristics of the path. This implies that the truncated signature for increasing orders throws away less and less information, similar to a low-rank approximation in PCA.

2.2.3. *Shuffle product*

Another key property of paths is that it linearizes the complex non-linear dynamics of high-dimensional oscillatory systems, such as financial time-series.

[2] $\gamma : [0,T] \to \mathbb{R}$ is of bounded variation if all changes $\sum_i |\gamma_{t_{i+1}} - \gamma_{t_i}|$ are bounded (finite) for all partitions $0 \leq t_0 \leq t_1 \leq ... \leq T$

One result that makes this more specific is the work of Ree [5] about the algebra of shuffles and Lie elements. One implication of his work is that the product of two terms $S(\gamma)_{0,T}^{i_1,...,i_k}$ and $S(\gamma)_{0,T}^{j_1,...,j_m}$ can be written as a sum of another collection of terms in $S(\gamma)_{0,T}$ which only depend on indices with $(i_1, ..., i_k)$ and $(j_1, ..., j_m)$ called the *shuffles* of i and j indices.

Let us denote all such combinations that preserve the order in i and j as $I \sqcup\!\sqcup J$. For a path $\gamma : [0,T] \to \mathbb{R}^d$ and two multi-indices I and J with all I: $i_1, ..., i_k \in d$ and all J: $j_1, ..., j_m \in d$, it holds that:

$$S(\gamma)_{0,T}^{I} S(\gamma)_{0,T}^{J} = \sum_{K \in I \sqcup\!\sqcup J} S(\gamma)_{0,T}^{K} \tag{25}$$

The proof is based on Fubini's theorem and can be found in [6]. For instance:

$$S(\gamma)^1 S(\gamma)^2 = S(\gamma)^{1,2} + S(\gamma)^{2,1} \tag{26}$$

In our example:

$$2 * 4 = \frac{16}{3} + \frac{8}{3} = 8 \tag{27}$$

In particular, this implies that the product of lower order signatures can be expressed as a linear sum of higher order terms. For instance, two 'interacting' drifts of paths in two dimensions can be expressed as a sum of variation across dimensions.

2.2.4. Other properties

In brief terms:

- **Invariance under time reparametrization**: sampling has no impact on signature values. This makes it robust to discrete paths and irregular sampling.

- **Chen's identity**:

$$S(\gamma * \phi)_{0,T} = S(\gamma)_{0,t1} \otimes S(\phi)_{t1,T} \tag{28}$$

  This identity implies that the signature of a concatenation of two paths $*$ can be expressed as a tensor product of the individual paths' signatures.

- **Time reversal**: the time-reversed path (values in $\mathbb{R}^d$ are put in the reverse order, denoted $\leftarrow$, over [a,b]) gives rise to the following equality:

$$S(X)_{a,b} \otimes S(\overleftarrow{X})_{a,b} = 1 \tag{29}$$

- **Log signatures**: we can take the formal logarithm of the signature (through the algebra of formal power series). This allows us to write a signature in a more concise form as its *log-signature*. This is often a more sparse representation of paths, but is computationally more expensive.

2.3. Cumulative sum of a path and lead-lag transform

Let us now investigate the interesting properties of paths which originate from embedding points using cumulative sums.

For instance $\gamma_t = \{\gamma_t^1, \gamma_t^2, ..., \gamma_t^d\}$ can be transformed (hereafter called CS transform) to the path

$$\begin{aligned}\tilde{\gamma}_t = \{0, \gamma_1^1, \gamma_1^1 + \gamma_2^1, ...,\\ \gamma_1^2, \gamma_1^2 + \gamma_2^2, ..., \gamma_k^d\}\end{aligned} \tag{30}$$

with $\gamma_k^d = \sum_{i=1}^k \gamma_i^d$. Chevyrev and Oberhauser [7] proved that the truncated signature of $\tilde{\gamma}$ at level M determines the statistical moments up to level L of the process that generates the path. The proofs can be found in the referenced paper and are beyond the scope of this introduction. However, let us consider an example.

Before we do, let us consider the lead-lag transform of a path, because if we apply this transform to the CS transformed data, the formula for the mean and variance becomes very straightforward [4].

The lead-lag transform for a path γ_t is defined as:

$$\hat{\gamma} = \begin{cases} \{\gamma_{t_i}, \gamma_{t_{i+1}}\}, & t \in [2i, 2i+1] \\ \{\gamma_{t_i}, \gamma_{t_{i+1}} + 2(t-(2i+1))(\gamma_{t_{i+2}} - \gamma_{t_{i+1}})\}, & t \in [2i+1, 2i+3/2] \\ \{\gamma_{t_i} + 2(t-(2i+3/2))(\gamma_{t_{i+1}} - \gamma_{t_{i+1}}), \gamma_{t_{i+1}}, \gamma_{t_{i+2}}\}, & t \in [2i+3/2, 2i+2] \end{cases} \tag{31}$$

with t running over [0, 2N]. The definition seems a bit convoluted, but the interpretation and visualisation is very intuitive. The lead-lag transform creates two paths, a lead and a lag where values of the initial path are shifted one-ahead (lead) or one-behind (lag). For instance, consider the path in $\mathbb{R}^2$: $\gamma = \{\{0, 1, 2, 3\}, \{1, 8, 5, 6\}\}$ depicted in figure 2.3. Its CS transform is shown in figure 2.3 and its lead-lag transform in figure 2.3. The CS and lead-lag transformed path is $\hat{\tilde{\gamma}} = \{\{0, 1, 1, 9, 9, 14, 14, 20, 20\}\{0, 0, 1, 1, 9, 9, 14, 14, 20\}\}$. It is clear that the first two sample moments of the original path is $Mean(\gamma^1) =$

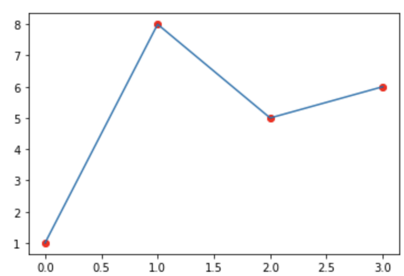

Figure 5: Sample path in $\mathbb{R}^2$

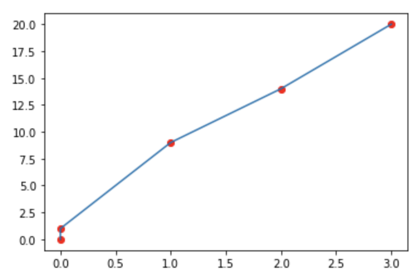

Figure 6: CS transform of sample path in figure 2.3

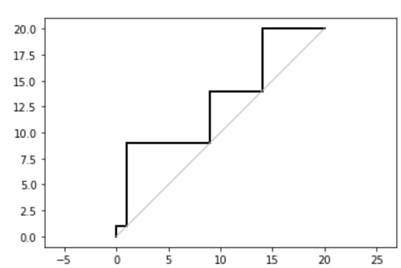

Figure 7: Lead-lag transform of sample path in figure 2.3

$5, Var(\gamma^1) = 6.5$. The signature of $\hat{\hat{\gamma}}$ is $(1, 20, 20, 200, 263, 137, 200)$. Chevyrev and Kormilitzin [4] show that

$$Mean(X) = \frac{1}{T} S(\hat{\hat{\gamma}})^1 \tag{32}$$

and

$$Var(X) = \frac{-(T+1)}{T^2} S(\hat{\hat{\gamma}})^{1,2} + \frac{T-1}{T^2} S(\hat{\hat{\gamma}})^{2,1} \tag{33}$$

When applied to our example we find:

$$Mean(X) = \frac{1}{4} 20 = 5 \tag{34}$$

$$Var(X) = \frac{-(4+1)}{16} 137 + \frac{4-1}{16} 263 = 6.5 \tag{35}$$

The traditional moments of γ^1 are generally determined by the first two orders of the signature of the CS transformed path. The lead-lag transform helps us to write the moments as straightforward functions of the signature. This is already one step towards the rigorous work in [7] which we will (briefly) summarize next.

2.4. The statistical moments of a path: traces of the stochastic law of the path

In the distributional setting (cf. section 1.2), there are well-established metrics to compare two distributions. In machine learning, we often encounter distributional distance metrics from information theory, such as the Kullback-Leibler (KL) and Jensen-Shannon (JS) divergences between two distributions. For stochastic processes that generate vector-valued data, there are well-known statistical tests for determining whether two samples are generated by the same stochastic process, such as the sequence of (normalised) moments and the Fourier transform (complex moments).

For path-valued data, Chevyrev and Oberhauser ([7]) introduce an analogue to normalised moments using the signature. They prove that for suitable normalizations λ, the sequence

$$(\mathbb{E}[\lambda(X)^m \int dX^{\otimes m}])_{m \geq 0} \tag{36}$$

determines the law of X *uniquely*[3]. According to the paper, this leads to efficient algorithms that can be combined with tools from machine learning such as maximum mean distances and kernelizations (e.g. [8]).

Generally, they also argue in favor of signatures as a feature map (i.e. to embed paths generated by a stochastic process in into a linear space) because of its *universality* and *characteristicness*. Universality implies that non-linear functions of the data are approximated by linear functionals in feature space (cf. above). Characteristicness is exactly their merit, i.e. that the expected value of the feature map *characterizes the law of the random variable*. A maximum mean distance[4] based on these moments was argued for in the market generator paper by Buehler et al [9].

3. Summary

5 reasons to use signatures for input representation of path-structured data (such as ξ sequences):

- It serves as a natural *basis* for describing path or stream[5] data that preserves the path structure and exploits symmetries.
- Linearizes the interaction effects between paths (e.g. by means of shuffle products). Previously referred to as *universality*.
- Determines the law of the process (e.g. ξ-generating process) *uniquely*. Previously referred to as *characteristicness*.
- Permits model-free (or data-driven) modelling as we do not impose parametric structure on the paths to summarize them (e.g. Buehler et al. [9])
- Finally, it is easy to implement and many quality packages available, e.g. *iisignature* and *esig* packages.

[3] Up to tree-like equivalance, see [7].

[4] A maximum mean discrepancy (MMD) between two samples is defined as $d(\mu,\nu) = \sup_f |\mathbb{E}_{X\sim\mu}[f(X)] - \mathbb{E}_{Y\sim\nu}[f(Y)]|$, where the samples are typically kernelized (e.g. over Gaussian or Euclidean kernels) for computational reasons. Efficient recursive algorithms then exist to find d.

[5] I.e. high-dimensional paths

4. Example applications to ξ processes

4.1. Relative drawdown prediction of Eurostoxx50 stocks

As a sample application that links drawdowns, machine learning and signatures, we will build a very simple classifier algorithm. Given Eurostoxx50 company data, we want to classify stocks according to their relative drawdown characteristics. Given the N=50 *previous* 6 month performance paths (110-day paths, hence $\gamma : [0, 110] \to \mathbb{R}^{50}$), we want to divide our sample into three classes of the *next* 1 month drawdown performance. We collected data from 1999-12-31 to 2021-09-30.

We do the following:

- Split up historical sample into blocks of 6 months, every such path is a feature γ_i. We test two transformations of the data:
  - The signature of the price paths up to order 5. These are calculated using the *iisignature* package [10].
  - The signature of the ξ-transformed paths up to order 5 according to the definition below (37). Denote by T_{γ_i-s} the start date of each path i and by T_{γ_i-e} the end date. Note that given our discussion, the signature determines the moments of the drawdown generating process up the the order of truncation.

$$\xi_t = \max_{i<t}(P_i) - P_t, t \in [T_{\gamma_i-s}, T_{\gamma_i-e}] \tag{37}$$

- For every γ_i we define the drawdown of the month following the last date of the path T_{γ_i-end} as $\mathbb{E}(\xi(\gamma))$. In the training sample, we can simply calculate the next-month (20-day) average drawdown as follows:

$$Mean(\hat{\xi}) = \frac{1}{20}\sum_{t=0}^{20} \hat{\xi}_t, \hat{\xi}_t = \max_{i<t}(P_i) - P_t, t \in [T_{\gamma_i-e}, T_{\gamma_i-e} + 20] \tag{38}$$

- Define the classes Y (= labels) as the top, middle and bottom tertile of $\mathbb{E}(\xi(\gamma))$. We let the classes correspond to a prediction of the points:

$$Y_j = \begin{cases} (-1, 0), & \mathbb{E}(\xi(\gamma_j)) < q_{1/3}(\mathbb{E}(\xi(\gamma)) \\ (1, 0), & q_{1/3}(\mathbb{E}(\xi(\gamma))) < \mathbb{E}(\xi(\gamma_j)) < q_{2/3}(\mathbb{E}(\xi(\gamma))) \\ (0, 1), & q_{2/3}(\mathbb{E}(\xi(\gamma))) < \mathbb{E}(\xi(\gamma_j)) \end{cases} \tag{39}$$

where j denotes the j^{th} instrument in $d = \{1, ..., 50\}$ and q_x the quantile operator up till the x^{th} quantile.

- Use a Random Forest classifier[6] as the simplest example of a machine learning pipeline that follows our feature representation.

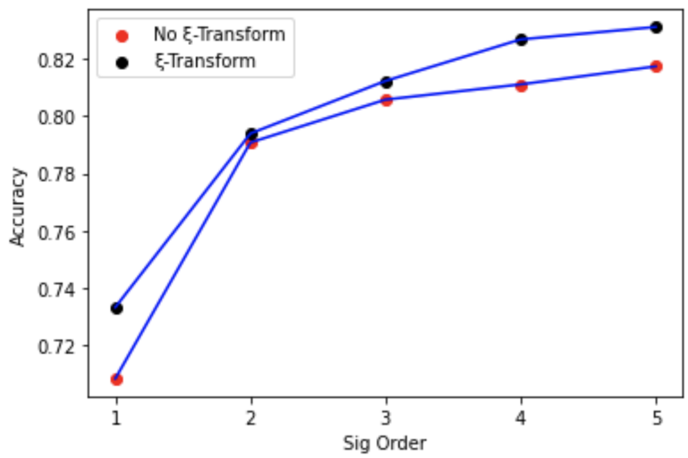

Figure 8: Classification accuracy as a function of the signature order

Sig Order	**Not ξ-transformed**	**ξ-transformed**
1	0.708534	0.733365
2	0.790743	0.793877
3	0.805689	0.812199
4	0.810993	0.826663
5	0.817261	0.831003

Table 1: Classification accuracy

We obtain around 80% out-of-sample accuracy (80-20 train-test split) in this classification task. Moreover, we notice that the ξ-transform is additive to prediction power.

[6]scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier. We used 100 estimators or 'trees in the forest'.

4.2. Detection of large codrawdown timeseries

Finally, we end with a numerical thought experiment and bottom-up simulate correlated drawdown paths, or series with varying 'codrawdowns'.

For the simulation of the ξ series we use the well-known bivariate Cholesky decomposition yielding:

$$\begin{aligned}\xi_1 = \max(dB_1, 0), dB_1 \sim N(0,1)\\ \xi_2 = \max(dB_2, 0)\\ dB_2 = \rho dB_1 + \sqrt{1-\rho^2}dB_3\\ dB_3 \sim N(0,1)\end{aligned} \tag{40}$$

Because of the max truncation required to generate drawdown samples, the correlation between ξ_1 and ξ_2 is not ρ anymore, but this allows us still to simulate highly correlated or decorrelated drawdown series, as pictured in Figure 9.

We sample 1000 such series each for two values of ρ, namely a highly correlated series ($\rho = 0.75$) and a decorrelated series ($\rho = -0.15$). We will first look at the difference in the signatures of the CS-transformed paths between the two regimes in Figure 4.2, where we show the 2-dimensional projections of all signatures up to order 2. The decorrelated paths have blue dots and the correlated ones are indicated with red ones. We immediately notice that the two regimes correspond to differing projections. We now want to test whether a machine learning model can use these projections to classify from which regime a new ξ series was generated.

For this we use a simple regularized linear model, called the LASSO[7]. We find that initially higher order projections substantially contribute to predictive power, with an eventual convergence to roughly 95% out-of-sample accuracy.

[7] Least Absolute Shrinkage and Selection Operator, scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso

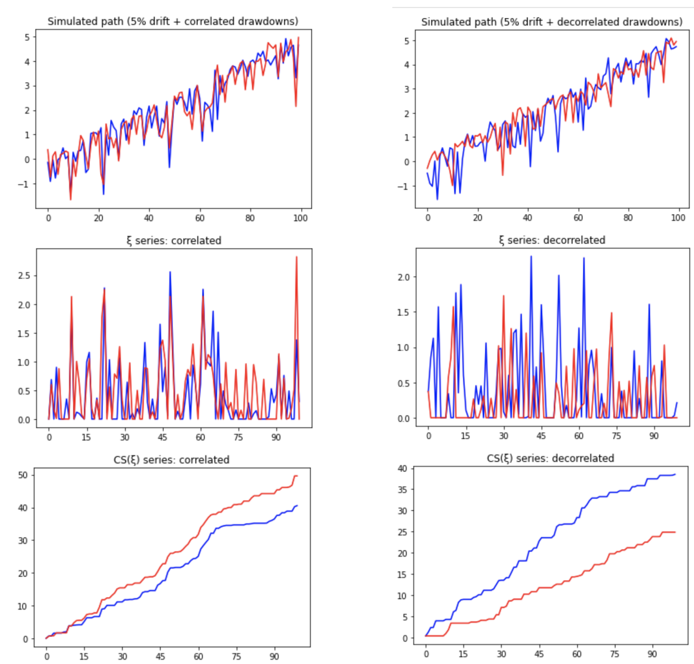

Figure 9: Simulated ξ and CS(ξ) series

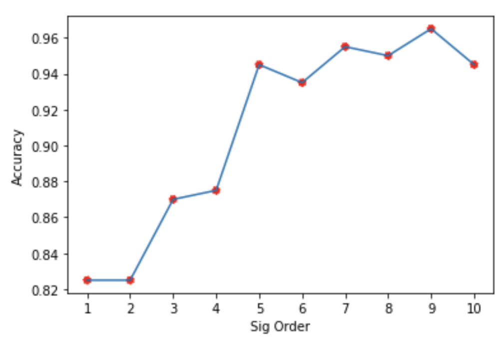

Figure 10: Performance of the LASSO classifier as a function of the signature order

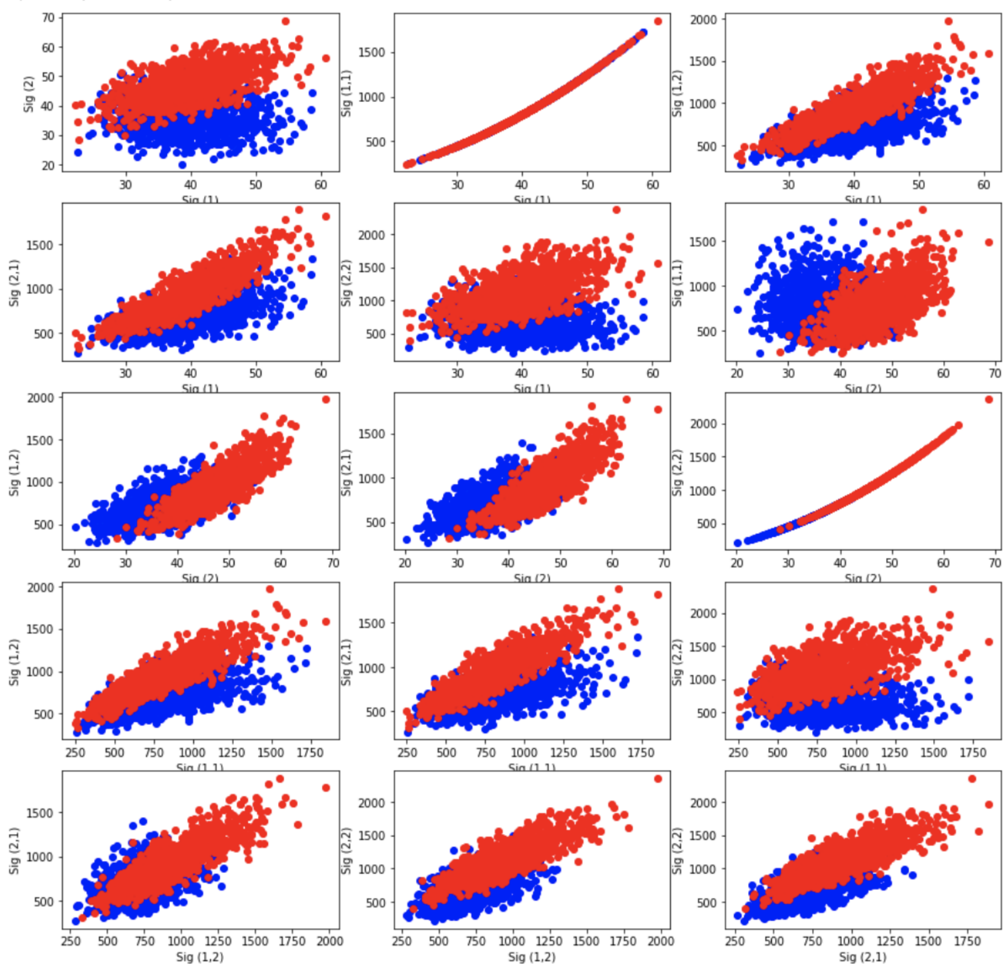

Figure 11: Signature projections up till the second order in two-dimensional space

References

[1] T. Lyons, Rough paths, signatures and the modelling of functions on streams, arXiv preprint arXiv:1405.4537 (2014).

[2] M. M. Bronstein, J. Bruna, T. Cohen, P. Veličković, Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, arXiv preprint arXiv:2104.13478 (2021).

[3] A. Chekhlov, S. Uryasev, M. Zabarankin, Drawdown measure in portfolio optimization, International Journal of Theoretical and Applied Finance 8 (2005) 13–58.

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

[5] R. Ree, Lie elements and an algebra associated with shuffles, Annals of Mathematics (1958) 210–220.

[6] T. J. Lyons, M. Caruana, T. Lévy, Differential equations driven by rough paths, Springer, 2007.

[7] I. Chevyrev, H. Oberhauser, Signature moments to characterize laws of stochastic processes, arXiv preprint arXiv:1810.10971 (2018).

[8] C. Salvi, M. Lemercier, C. Liu, B. Hovarth, T. Damoulas, T. Lyons, Higher order kernel mean embeddings to capture filtrations of stochastic processes, arXiv preprint arXiv:2109.03582 (2021).

[9] H. Buehler, B. Horvath, T. Lyons, I. Perez Arribas, B. Wood, A data-driven market simulator for small data environments, Available at SSRN 3632431 (2020).

[10] J. F. Reizenstein, B. Graham, Algorithm 1004: The iisignature library: Efficient calculation of iterated-integral signatures and log signatures, ACM Transactions on Mathematical Software (TOMS) 46 (2020) 1–21.