Data -Driven Market Simulators
and some applications of signature kernel methods
in mathematical finance

Blanka Horvath

Institute Louis Bachelier

FaIR programme: Deep Learning in Finance

Tuesday, 28th September, 2021

The talk is based on

- Mikko Pakkanen and Aitor Muguruza: "Harnessing quantitative finance by data-driven methods"
- Zacharia Issa, Maud Lemercier, Chong, Liu, Cris Salvi, and Terry Lyons (recent+ongoing works)

The talk is based on

- Mikko Pakkanen and Aitor Muguruza: "Harnessing quantitative finance by data-driven methods"
- Zacharia Issa, Maud Lemercier, Chong, Liu, Cris Salvi, and Terry Lyons (recent+ongoing works)
- and the motivation originated from the joint work with

  Hans Bühler, Terry Lyons, Imanol Perez Arribas, Ben Wood

Market Simulation and Market Generation

Transforming horizons of mathematical modelling towards models that can more and more accurately fit market data:

(1) Classical and Neo-classical stochastic market models: $dS_t = rS_t dt + \sigma S_t dW_t \ldots$

(2) DNN-based Generative Modelling in other AI applications: Data driven (no a-priori assumption on distribution of stochastic process) $f \in \mathcal{N}_r(I, d_1, \ldots, d_{r-1}, O; \sigma_1, \ldots \sigma_r)$ "non-parametric" $+$ very flexible

Market Simulation and Market Generation

Transforming horizons of mathematical modelling towards models that can more and more accurately fit market data:

(1) Classical and Neo-classical stochastic market models: $dS_t = rS_t dt + \sigma S_t dW_t \ldots$

(2) DNN-based Generative Modelling in other AI applications: Data driven (no a-priori assumption on distribution of stochastic process) $f \in \mathcal{N}_r(I, d_1, \ldots, d_{r-1}, O; \sigma_1, \ldots \sigma_r)$ "non-parametric" $+$ very flexible

(1.5) Best of both worlds: mixture models neural SDEs, ODEs ...

New challenges arising from DNN-based financial applications:

Training data shapes the DNN:

- **Data Privacy** Can adversaries obtain information about (proprietory) data from the trained engine's network weights if this is trained on real data? (adversarial attacks)
- **Data availability:** Is there enough real data available to train the engine? (Synthetic data!)
- **Model Governance:** Markets are non-stationary $\Rightarrow$ retraining the network may become necessary $\Rightarrow$ Rethinking model governance:
  see Deep Hedging under Rough Volatility (control over non-Markovianity to assess hedging error.)

New challenges arising from DNN-based financial applications:

Training data shapes the DNN:

- **Data Privacy** Can adversaries obtain information about (proprietory) data from the trained engine's network weights if this is trained on real data? (adversarial attacks)
- **Data availability:** Is there enough real data available to train the engine? (Synthetic data!)
- **Model Governance:** Markets are non-stationary $\Rightarrow$ retraining the network may become necessary $\Rightarrow$ Rethinking model governance:
  see Deep Hedging under Rough Volatility (control over non-Markovianity to assess hedging error.)
- **In practice the latter questions can be formulated as ways of quantifying how similar or dissimilar sets of sample paths are one another**

New challenges arising from DNN-based financial applications:

Training data shapes the DNN:

- **Data Privacy** Can adversaries obtain information about (proprietory) data from the trained engine's network weights if this is trained on real data? (adversarial attacks)
- **Data availability:** Is there enough real data available to train the engine? (Synthetic data!)
- **Model Governance:** Markets are non-stationary $\Rightarrow$ retraining the network may become necessary $\Rightarrow$ Rethinking model governance:
  see Deep Hedging under Rough Volatility (control over non-Markovianity to assess hedging error.)
- **In practice the latter questions can be formulated as ways of quantifying how similar or dissimilar sets of sample paths are one another**
- **Evaluating the "quality" of training data** for financial data streams (Application specific approaches? Universal approaches? Adaptation to the sequential and high dimensional nature of data?)

Data Generation for Machine Learning

(2) DNN-based Generative Modelling in other AI applications: Adapt to Finance?

Generative models can be trained to transport **some source distribution** μ (say $\mu \sim \mathcal{N}(0,1)$) to some **target distribution** observed in the data

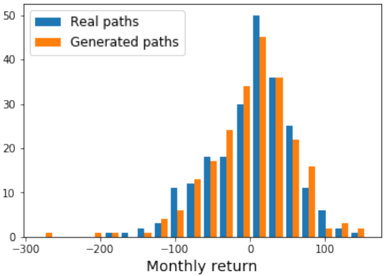
Data Generation for Machine Learning

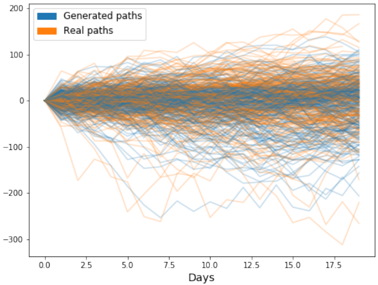
(2) DNN-based Generative Modelling in other AI applications: Adapt to Finance?

Generative models can be trained to transport **some source distribution** μ (say $\mu \sim \mathcal{N}(0,1)$) to some **target distribution** observed in the data and generate samples that are similar/indistinguishable from the ones observed η_X (no assumptions made on the latter) and learn transformation $\mu \to \eta_X$

Market Simulation: Streamed financial data

Generation of Returns vs. Generation of paths





(Rough) Paths-wise Approach to Generative Modelling of Markets

Levin, Lyons, & Ni. (2013) proposed the signature of a path as a basis of functions for a functional on path space.

Definition (Signature of a path)

Let $X : [0, T] \to \mathbb{R}^d$ be a continuous path of bounded variation. The signature of X is then defined by the sequence of iterated integrals given by

$$\mathbb{X}_T^{<\infty} := (1, \mathbb{X}_t^1, \ldots, \mathbb{X}_T^n, \ldots), \quad \text{where}$$

$$\mathbb{X}_T^n := \int_{0<u_1<\ldots<u_k<T} dX_{u_1} \otimes \ldots \otimes dX_{u_k} \in (\mathbb{R}^d)^{\otimes n}$$

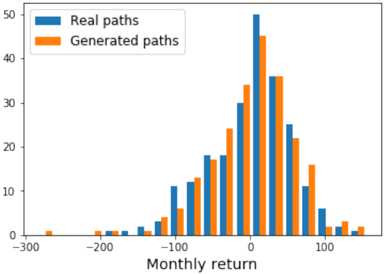
with $\otimes$ the tensor product. Similarly, given $N \in \mathbb{N}$, the truncated signature of order N is defined by

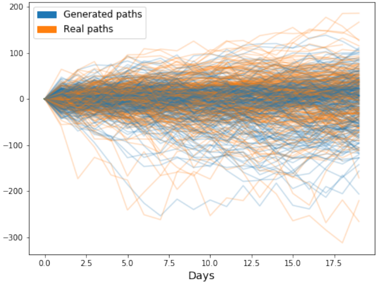
$$\mathbb{X}_T^{\leq N} := (1, \mathbb{X}_T^1, \ldots, \mathbb{X}_T^N).$$

The path X has b.v. (discrete data) $\Rightarrow$ the integrals can be defined i.s.o. Riemann-Stieltjes.

(Rough) Path-wise Approach

Returns-based generation (i) vs. Path-based generation (ii)





Time-series data generation with signatures

(Step 1) **Data extraction from time series** we subdivide original (say an asset price or index) data into partitions of: (1) daily data, (2) weekly path segments, i.e. 5 days, and
(3) monthly path segments, i.e. 20 days.

(Step 2) **Preprocessing the data** transforming data into (i) returns for all (1), (2), (3) and into (ii) log-signatures for (2), (3) of (leadlag) paths.

(Step 3) **Creating and training the VAE and the CVAE network:** VAE, a parsimonious generator model with "bottleneck structure". Conditional Variational Autoencoder is learned to condition VAE on current market conditions (a) current level of the index (b) instantaneous volatility (c) signature of the previous path segment.

(Step 4) **Postprocessing of the outputs of the VAEs**
transforming (i), (ii) back to paths; such as building paths of arbitrary length

(Step 5) **Performance evaluation** similar to the role of the discriminator in GANs, but here without feeding back to the generator.

Time-series data generation with signatures

(Step 1) **Data extraction from time series** we subdivide original (say an asset price or index) data into partitions of: (1) daily data, (2) weekly path segments, i.e. 5 days, and
(3) monthly path segments, i.e. 20 days.

(Step 2) **Preprocessing the data** transforming data into (i) returns for all (1), (2), (3) and into (ii) log-signatures for (2), (3) of (leadlag) paths.

(Step 3) **Creating and training the VAE and the CVAE network:** VAE, a parsimonious generator model with "bottleneck structure". Conditional Variational Autoencoder is learned to condition VAE on current market conditions (a) current level of the index (b) instantaneous volatility (c) signature of the previous path segment.

(Step 4) **Postprocessing of the outputs of the VAEs**
transforming (i), (ii) back to paths; such as building paths of arbitrary length

(Step 5) **Performance evaluation** similar to the role of the discriminator in GANs, but here without feeding back to the generator.

Good performance evaluation metrics?

(Rough) Path-wise Approach

- Signatures provide the right framework for performance evaluation metrics on pathspace, one of the difficulties being that the pathspace $C([0,1], R^d)$ is infinite-dimensional and not locally compact. **Chevyrev, Oberhauser (2018)**
- Signatures provide a basis of functions for a functional on path space in an un-parametrised way (model free, path by path characterisation).
- Robustness to missing data and irregular sampling and towards highly oscillatory data, and invariance under time re-parametrisation. **Liao, Lyons, Ni, Yang (2019)**

(Rough) Path-wise Approach

- Signatures provide the right framework for performance evaluation metrics on pathspace, one of the difficulties being that the pathspace $C([0,1], R^d)$ is infinite-dimensional and not locally compact. **Chevyrev, Oberhauser (2018)**
- Signatures provide a basis of functions for a functional on path space in an un-parametrised way (model free, path by path characterisation).
- Robustness to missing data and irregular sampling and towards highly oscillatory data, and invariance under time re-parametrisation. **Liao, Lyons, Ni, Yang (2019)**

The Signature MMD Two-Sample Test (Base Case)

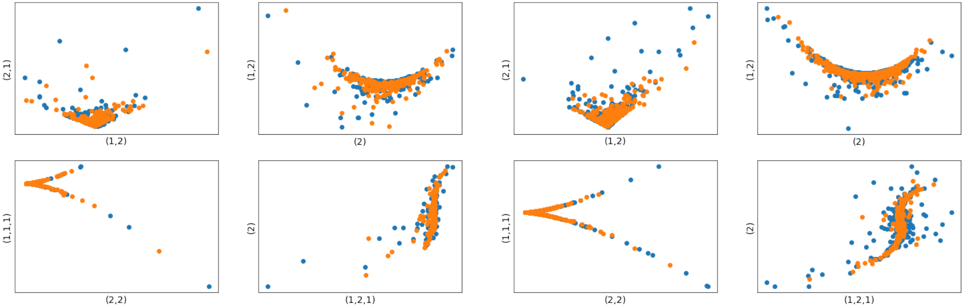
To asses whether a generative model is able to generate paths that are realistic with respect to a sample of real paths $Y_1, \ldots, Y_n$, we sample from the generative model $n \in \mathbb{N}$, paths $X_1, \ldots, X_n$ and we apply the two-sample test proposed by Chevyrev and Oberhauser [CO18]. More specifically, we compute the signature-based MMD test statistic $T(X_1, \ldots, X_n; Y_1, \ldots, Y_n)$

$$T(X_1, \ldots, X_n; Y_1, \ldots, Y_n) := \frac{1}{n(n-1)} \sum_{i,j;i \neq j} k(X_i, X_j) - \frac{2}{n^2} \sum_{i,j} k(X_i, Y_j) + \frac{1}{n(n-1)} \sum_{i,j;i \neq j} k(Y_i, Y_j), \tag{1}$$

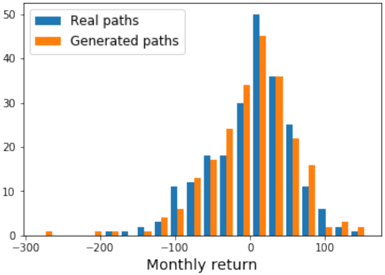
where $k(\cdot, \cdot)$ is the so-called *signature kernel*. Then, given a fixed confidence level $\alpha \in (0, 1)$, we compute the threshold $c_\alpha := 4\sqrt{-n^{-1} \log \alpha}$. The generative model will be said to be realistic with a confidence α if $T_U^2 < c_\alpha$.

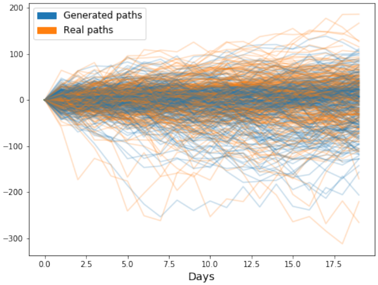
Numerical Results

- Signature-based generative model illustration.

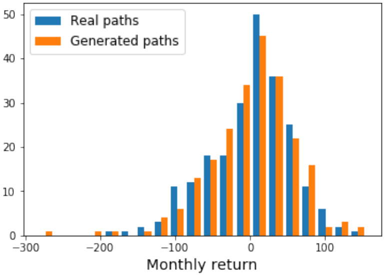


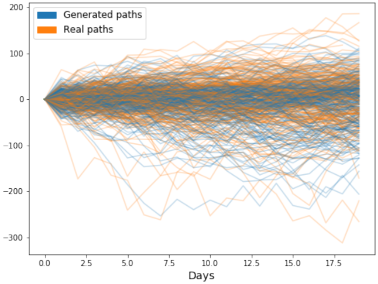
(Rough) Path-wise Approach: Adapted Topologies





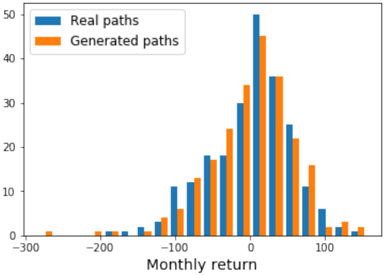
(Rough) Path-wise Approach: Adapted Topologies

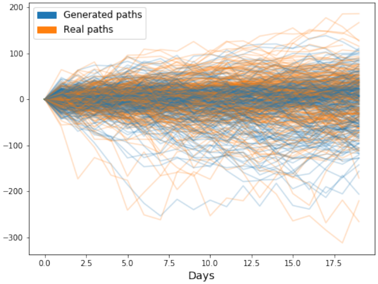




The information structure: Filtration-sensitive approaches: adapted topologies (Wasserstein distance- Adapted Wasserstein distance)
Hoover-Keisler: "Two random variales are alike if they have the same distribution".

(Rough) Path-wise Approach: Adapted Topologies

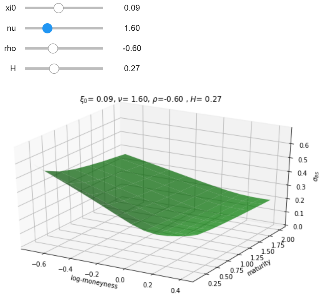




The information structure: Filtration-sensitive approaches: adapted topologies (Wasserstein distance- Adapted Wasserstein distance)
Hoover-Keisler: "Two random variales are alike if they have the same distribution".
On process level "Two Markov processes are alike: Same finite dimensional distributions". Adapted distributions: On first order "Synonymous" (to preserve martingality and conditional expectations).

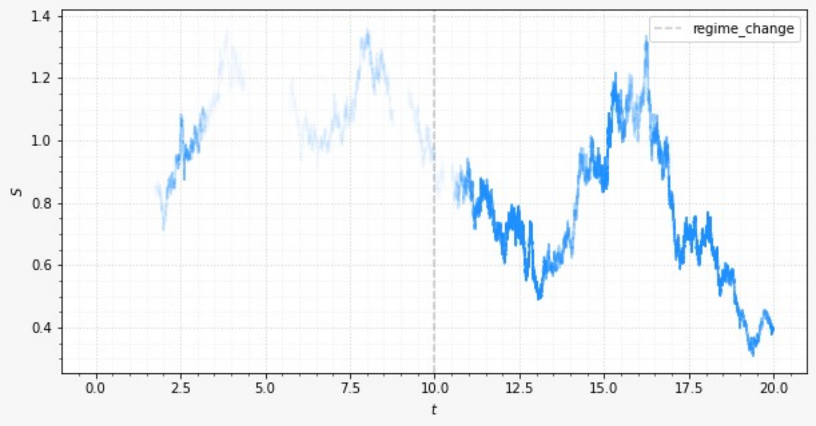
Application I: Regime detection

Mixture models: To simulate a regime change we take a mixture of two (or more) stochastic volatility models and calibrate it to data including the mixture parameter a.
$a \times$ **Heston** $+ (1 - a) \times$ **rBergomi**



(with Christian Bayer, Aitor Muguruza, Mehdi Tomas and Benjamin Stemper)

Application I: Regime Detection



(Zacharia Issa)

Key: Computing similarity metrics

The question of quantifying how similar or dissimilar sets of sample paths are to/from one another is key to several applications (ongoing works with Z. Issa, C. Liu, M. Lemercier, T. Lyons and C. Salvi)

Key: Computing similarity metrics

The question of quantifying how similar or dissimilar sets of sample paths are to/from one another is key to several applications (ongoing works with Z. Issa, C. Liu, M. Lemercier, T. Lyons and C. Salvi)

Applications include:

- Outlier detection: standard regime versus outlier regime and the context of automation (Human-Machine Interface).
  See also: Cochrane, Foster, Lyons, Perez Arribaz (2020)
- Regime detection
  (bull regimes vs. bear regimes or typical regimes vs. atypical regimes)

Key: Computing similarity metrics

The question of quantifying how similar or dissimilar sets of sample paths are to/from one another is key to several applications (ongoing works with Z. Issa, C. Liu, M. Lemercier, T. Lyons and C. Salvi)

Applications include:

- Outlier detection: standard regime versus outlier regime and the context of automation (Human-Machine Interface).
  See also: Cochrane, Foster, Lyons, Perez Arribaz (2020)
- Regime detection
  (bull regimes vs. bear regimes or typical regimes vs. atypical regimes)
  Two sample tests to evaluate similarity of the data directly: Maximum Mean Discrepancy improving the standard two sample tests such as Kolmogorov-Smirnov. Pathwise versions of MMD via Signatures
  Using information from the Implied Volatility Surface

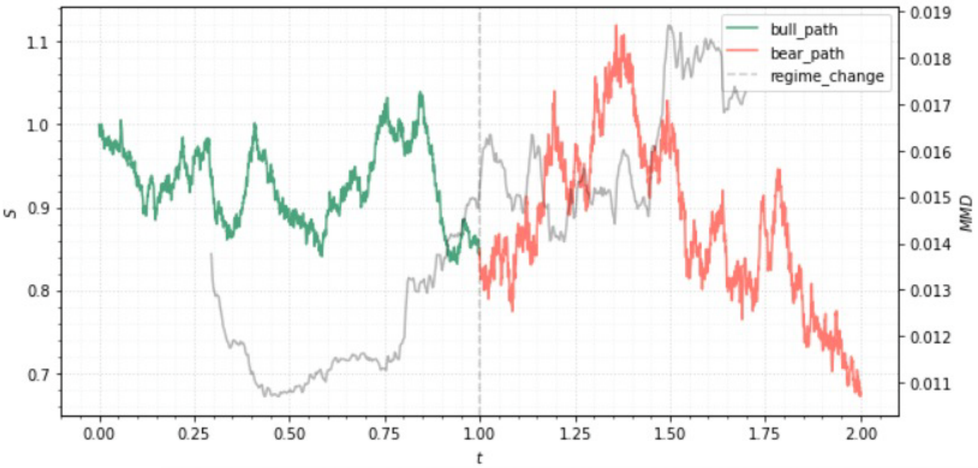
Key: Computing similarity metrics

The question of quantifying how similar or dissimilar sets of sample paths are to/from one another is key to several applications (ongoing works with Z. Issa, C. Liu, M. Lemercier, T. Lyons and C. Salvi)

Applications include:

- Outlier detection: standard regime versus outlier regime and the context of automation (Human-Machine Interface).
  See also: Cochrane, Foster, Lyons, Perez Arribaz (2020)
- Regime detection
  (bull regimes vs. bear regimes or typical regimes vs. atypical regimes)
  Two sample tests to evaluate similarity of the data directly: Maximum Mean Discrepancy improving the standard two sample tests such as Kolmogorov-Smirnov. Pathwise versions of MMD via Signatures
  Using information from the Implied Volatility Surface

Regime Detection using MMD metrics



(Zacharia Issa)

Improvement of the base case I: Kernel-trick

- The goal is to compare two sets of paths (e.g. asset price trajectories).
- This can be achieved by using a kernel function $k(\cdot, \cdot)$ on paths and the corresponding MMD distance

$$T(X_1, \dots, X_n; Y_1, \dots, Y_n) := \frac{1}{n(n-1)} \sum_{i,j;i \neq j} k(X_i, X_j) - \frac{2}{n^2} \sum_{i,j} k(X_i, Y_j) + \frac{1}{n(n-1)} \sum_{i,j;i \neq j} k(Y_i, Y_j), \tag{2}$$

- The signature kernel can be evaluated using state-of-the-art hyperbolic PDE solvers

$$\frac{\partial^2 k}{\partial s \partial t} = \langle \dot{X}_s, \dot{Y}_t \rangle k \tag{3}$$

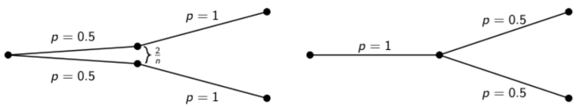
It allows to measure the similarity between high-dimensional paths

Improvement of the base case II: Including filtration information

- However despite the ability of the MMD to distinguish measures on paths, the MMD distance can fail in some situations that are typically encountered in finance
- This motivates a filtration sensitive approach: the MMD distance needs to be extended to a higher order version to capture aspects about the filtration of the underlying process, and not only its moments (expected signature)

Improvement of the base case II: Including filtration information

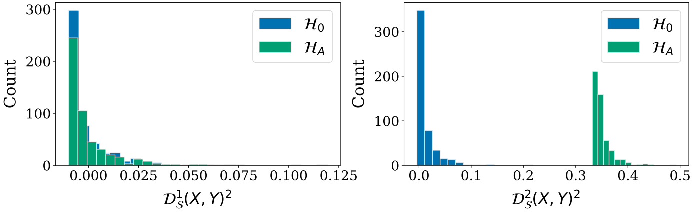
- However despite the ability of the MMD to distinguish measures on paths, the MMD distance can fail in some situations that are typically encountered in finance
- This motivates a filtration sensitive approach: the MMD distance needs to be extended to a higher order version to capture aspects about the filtration of the underlying process, and not only its moments (expected signature)



The supports of $\mathbb{P}_n$ (left) and $\mathbb{P}$ (right).

Improvement of the base case II: Including filtration information

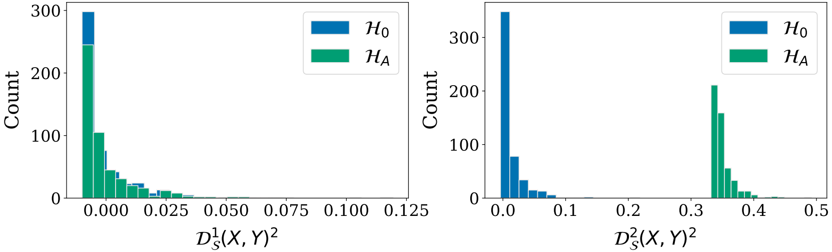
- However despite the ability of the MMD to distinguish measures on paths, the MMD distance can fail in some situations that are typically encountered in finance
- This motivates a filtration sensitive approach: the MMD distance needs to be extended to a higher order version to capture aspects about the filtration of the underlying process, and not only its moments (expected signature)



The supports of $\mathbb{P}_n$ (left) and $\mathbb{P}$ (right).

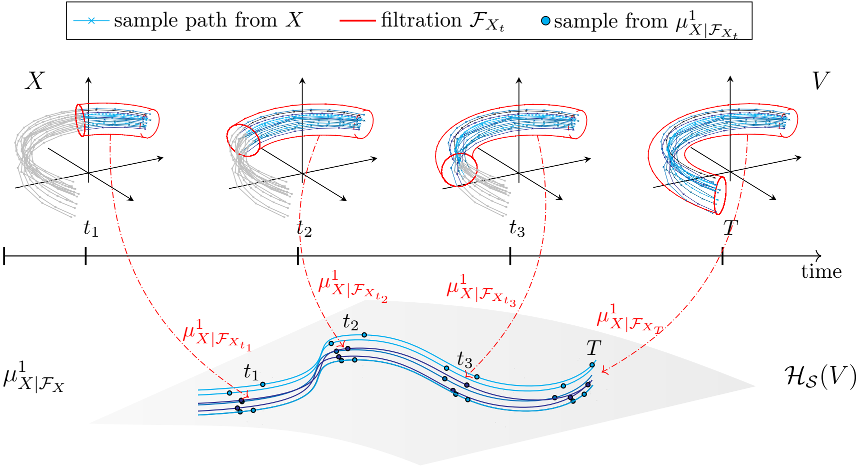
Improvement of the base case II: Including filtration information

- Let $\mathbb{P}_n = \delta_{(0,1/n,1)} + \delta_{(0,-1/n,-1)}$ and $\mathbb{P} = \delta_{(0,0,1)} + \delta_{(0,0,-1)}$



Empirical distribution of the 1st order MMD (left) and the 2nd order MMD (right). Under $\mathcal{H}_0$ the two measures are both equal to $\mathbb{P}$. Under $\mathcal{H}_A$ with $\mathbb{P}$ and $\mathbb{P}_n$ where $n = 10^5$. We use 500 independent instances of the MMD.

Improvement of the base case II: Including filtration information



Application II & III:
Pathwise calibration & Path dependent options

- This methodology allows to formulate classical problems in finance as supervised learning problems that we can solve via kernels leveraging the above generalized distance
- This includes calibration of rough volatility models and path dependent option pricing

Table: Average performances with standard errors in parenthesis.

Kernel	Rough Bergomi model calibration (Acc.)	American option pricing (MSE $\times 10^{-3}$)
RBF	80% (5%)	1.07 (0.75)
Matérn	87% (3%)	2.75 (3.05)
K_S^1	88% (3%)	0.90 (0.34)
K_S^2	**91%** (3%)	**0.52** (0.07)

Thank you for your attention.

Rough Paths Approach to Generative Modelling

In what follows we work with the log-signatures (**Liao, Lyons, Ni, Yang (2019)**)

Definition (Log-signature)

Let $X : [0, T] \to \mathbb{R}^d$ be a path such that its signature $\mathbb{X}_{0,T}^{<\infty}$ is well-defined. The log-signature is then defined by

$$\log \mathbb{X}_T^{<\infty} := -\mathbb{X}_T^{<\infty} + \frac{1}{2}(\mathbb{X}_T^{<\infty})^{\otimes 2} - \frac{1}{3}(\mathbb{X}_T^{<\infty})^{\otimes 3} + \ldots + (-1)^n \frac{1}{n}(\mathbb{X}_T^{<\infty})^{\otimes n} + \ldots,$$

which can be shown to be well-defined.

- There is a one-to-one map between signatures and log-signatures.
- Log-signatures have all positive properties listed above.
- They allow for lower dimensional representation and are better suited to VAE.