SigNaGen: Non-adversarial training of Neural SDEs with signature kernel scores

Bloomberg-Columbia Machine Learning in Finance Conference

Zacharia Issa

September 2024

Table of Contents

Table of Contents

Introduction

- Data-hungry machine learning models are increasingly supplemented with synthetic data
- Such data often is the output of a separate *generative model* $G_\theta : \Theta \times \mathscr{Z} \to \mathscr{X}$, or *generator*
- Objective: Train G_θ such that $\mathbb{P}_{X^\theta} = \mathbb{P}_{X^{\text{true}}}$, where $\mathbb{P}_{X^\theta} = G_{\theta\#}\mathbb{P}_{\mathscr{Z}}$ and $(\mathscr{Z}, \mathbb{P}_{\mathscr{Z}})$ is a latent space
- Particular case: (financial) time series data $X \in \mathscr{X}$, where $\mathscr{X} = C_p([0, T]; \mathbb{R}^d)$, $1 \leq p < 2$
- For theoretical reasons, we always assume $X \in \mathscr{X}$ contains one monotone coordinate, usually taken to be time

Introduction

- Components:
  - A generator $G_\theta : \Theta \times \mathscr{Z} \to \mathscr{X}$
  - A discriminator $D : \mathscr{P}(\mathscr{X}) \times \mathscr{P}(\mathscr{X}) \to \mathbb{R}$, and
  - A training procedure.

- In the case where the discriminator is parametrized, $D = D_\phi$, and training objective is adversarialized, we recover the classic GAN [GPAM+20]

- Goal: a *plug-and-play* pipeline for training generative models on path space, which is
  - *mesh-free*,
  - *stable*,
  - *memory efficient*,
  - easily able to be *conditionalized*, and
  - can handle paths taking values in *infinite-dimensional spaces*

Table of Contents

Neural SDEs

- Let $W : [0, T] \to \mathbb{R}^{d_w}$ be a d_w-dimensional Brownian motion, and $a \sim \mathscr{N}(0, I_{d_a})$

- A *Neural SDE* is a model of the form

$$Y_0 = \xi_\theta(a), \quad dY_t = \mu_\theta(t, Y_t)dt + \sigma_\theta(t, Y_t) \circ dW_t, \quad X_t^\theta = \pi_\theta(Y_t)$$

  where $\xi_\theta : \mathbb{R}^{d_a} \to \mathbb{R}^{d_y}, \mu_\theta : [0, T] \times \mathbb{R}^{d_y} \to \mathbb{R}^{d_y}$, $\sigma_\theta : [0, T] \times \mathbb{R}^{d_y} \to \mathbb{R}^{d_y \times d_w}$ and $\pi_\theta : \mathbb{R}^{d_y} \to \mathbb{R}^{d_x}$ are neural networks

- $\mu_\theta, \sigma_\theta$ Lipschitz and $\mathbb{E}[\xi_\theta(a)^2] < \infty \implies$ strong solution Y exists and is unique

- In general we write $\mathbb{P}_{X^\theta}$ to denote the law of the Neural SDE parametrized by $\theta \in \mathbb{R}^p$

The signature of a path

- Recall the *signature* $S(x)$ of a path $x \in \mathscr{X}$ is given by $S(x) = (1, S^1(x), S^2(x), \ldots,)$, where

$$S^k(x) := \int_{0<t_1<\ldots<t_k<T} dx_{t_1} \otimes dx_{t_2} \otimes \ldots \otimes dx_{t_k}, \quad k \in \mathbb{N}.$$

- $S(x)$ lives in the tensor algebra $T((\mathbb{R}^d)) = \prod_{k=0}^{\infty} (\mathbb{R}^d)^{\otimes k}$
- Can be thought of as a canonical feature map on path space
- Existence is non-trivial; method of integration depends on regularity of x

Important properties of the signature

- Suppose $\mathscr{X}$ is a subset of

$$\mathscr{C}_p([0,T];\mathbb{R}^d) = \{[X] : X \in C_p([0,T];\mathbb{R}^d)\},$$

  where $[X]$ is the equivalence class of paths under the *tree-like* equivalence relation $\sim_\tau$ (equal up to retracings).

- **Characteristicness**: If $X \sim_\tau Y$, then $S(X) = S(Y)$

- **Universality**: Continuous functions on $\mathscr{X}$ can be arbitrarily well-approximated by linear functionals acting on the signature. Given $f \in C(\mathscr{X})$, for every $\varepsilon > 0$ there exists an $L \in T((E))^*$ such that for all $X \in \mathscr{X}$,

$$\left\| f(X) - \langle L, S(X) \rangle_{T((E))} \right\|_\infty < \varepsilon.$$

The signature kernel

- The signature is a characteristic feature map on path space
- The associated kernel $k_{\text{sig}} : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is given by

$$k_{\text{sig}}(x, y) = \sum_{k \geq 0} \langle S^k(x), S^k(y) \rangle_k$$

  where $\langle \cdot, \cdot \rangle_k$ is the inner product on $(\mathbb{R}^d)^{\otimes k}$.

- By characteristicness of S, the mapping $\mathbb{P} \mapsto \mathbb{E}_{\mathbb{P}}[k_{\text{sig}}(X, \cdot)]$ is injective for any $\mathbb{P} \in \mathscr{P}(\mathscr{K})$
- A "kernel trick" exists for k_{sig} [SCF$^+$21] via solving a Goursat PDE

Scoring rules

Definition 1 (Scoring rule, [GR07])

Let $\mathscr{P}$ be a convex class of measures on a probability space $(\mathscr{X}, \mathscr{A})$. A *scoring rule* $s : \mathscr{P} \times \mathscr{X} \to [-\infty, \infty]$ is any function such that $s(\mathbb{P}, \cdot)$ is $\mathscr{P}$*-quasi integrable* for all $\mathbb{P} \in \mathscr{P}$.

Definition 2 (Properness, [GR07])

A scoring rule $s : \mathscr{P} \times \mathscr{X} \to [-\infty, +\infty]$ is called *proper* (relative to the class $\mathscr{P}$) if $s(\mathbb{P}, \mathbb{P}) \leq s(\mathbb{Q}, \mathbb{P})$ for all $\mathbb{P}, \mathbb{Q} \in \mathscr{P}$. It is called *strictly proper* if $\mathbb{Q} = \mathbb{P}$ is the unique minimiser.
Here, $s(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{y \sim \mathbb{Q}}[s(\mathbb{P}, y)]$ denotes the *expected scoring rule*.

Natural way to define a divergence: $\mathscr{D}_s(\mathbb{P}||\mathbb{Q}) = s(\mathbb{Q}.\mathbb{P}) - s(\mathbb{P}, \mathbb{P})$ for strictly proper s

Table of Contents

Related work

- [BHL$^+$20]: VAE-based generative model using the truncated signature MMD. Required signature inversion.
- [KFL$^+$21]: SDE-GAN, adversarialising the training objective via a Neural CDE. Class conditioning examples.
- [NSSV$^+$21, NSW$^+$20]: Training Log-RNN, AR-FNN generator against the Sig-Wasserstein distance.
  - Again uses truncation of the signature
  - Conditioning examples relied on relationship between past truncated signature and future which do not hold in practice
- [WKKK20, WWP$^+$21]: Not mesh-free, and not conditional (mentioned as a future extension)

Signature kernel scores I

- For a given kernel k on $\mathscr{X}$, the associated *kernel scoring rule* s_k is given by

$$s_k(\mathbb{P}, y) = \mathbb{E}_{x,x' \sim \mathbb{P}}[k(x,x')] - 2\mathbb{E}_{x \sim \mathbb{P}}[k(x,y)]$$

- Denote by $\phi_{\text{sig}} : \mathscr{P}(\mathscr{X}) \times \mathscr{X} \to \mathbb{R}$ the kernel scoring rule associated to the signature kernel k_{sig}

Proposition 1 ([IHLS23], Proposition 3.3)

For any compact $\mathscr{K} \subset \mathscr{X}$, ϕ_{sig} is a strictly proper kernel score relative to $\mathscr{P}(\mathscr{K})$, i.e. $\mathbb{E}_{y \sim \mathbb{Q}}[\phi_{\text{sig}}(\mathbb{Q}, y)] \leq \mathbb{E}_{y \sim \mathbb{Q}}[\phi_{\text{sig}}(\mathbb{P}, y)]$ for all $\mathbb{P}, \mathbb{Q} \in \mathscr{P}(\mathscr{K})$, with equality if and only if $\mathbb{P} = \mathbb{Q}$.

Signature kernel scoring rules II

- Given samples $\{x_i\}_{i=1}^m \sim \mathbb{P}$ and $y \in \mathscr{X}$, an unbiased estimator of ϕ_{sig} is given by

$$\hat{\phi}_{\text{sig}}(\mathbb{P}, y) = \frac{1}{m(m-1)} \sum_{i \neq j} k_{\text{sig}}(x_i, x_j) - \frac{2}{m} \sum_{i=1}^{M} k_{\text{sig}}(x_i, y)$$

- Note that

$$\mathscr{D}_{\text{sig}}(\mathbb{P}, \mathbb{Q})^2 = \phi_{\text{sig}}(\mathbb{P}, \mathbb{Q}) + \mathbb{E}_{y, y' \sim \mathbb{Q}}[k_{\text{sig}}(y, y')],$$

that is, we recover the classical (squared) signature maximum mean discrepancy (MMD)

Non-adversarial training of Neural SDEs

- In the unconditional setting, the training objective is

$$\min_{\theta} \mathscr{L}(\theta) \quad \text{where} \quad \mathscr{L}(\theta) = \mathbb{E}_{y \sim \mathbb{P}_{X^{\text{true}}}}[\phi_{\text{sig}}(\mathbb{P}_{X^\theta}, y)] + \lambda \|\theta\|_{L_2}$$

- Strict properness ensures that the training objective is minimised when $\mathbb{P}_{X^\theta} = \mathbb{P}_{X^{\text{true}}}$
- Training procedure can be summarized by

$$\textbf{Generator: } X^\theta \approx \text{SDESolve}(\theta),$$
$$\textbf{Discriminator: } \mathscr{L}(\theta) \approx \text{PDESolve}(X^\theta, X^{\text{true}}).$$

- Both procedures are able to be backpropagated through
  - **Generator:** Through the SDE solver
  - **Discriminator:** Via solving another system of adjoint PDEs [LSC+21]

The conditional case

- The advantage of this approach lies in the ease of adaptability to the conditional generation problem [PD22]
- Let $x \sim \mathbb{Q}$ be any conditioning variable. Then, the training objective becomes

$$\min_{\theta} \mathscr{L}'(\theta) \quad \text{where} \quad \mathscr{L}'(\theta) = \mathbb{E}_{x \sim \mathbb{Q}} \mathbb{E}_{y \sim \mathbb{P}_{X^{\text{true}}}(\cdot|x)} [\phi_{\text{sig}}(\mathbb{P}_{X^\theta}(\cdot|x), y)]$$

- We integrate the conditioning variable(s) by expanding the NNs defining the Neural SDE
- With data sampled as $\{x_i, y_i\}_{i=1}^n$, the (batched) training objective can be written as

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \phi_{\text{sig}}(\mathbb{P}_{X^\theta}(\cdot|x_i), y_i)$$

Spatiotemporal generation

- Up until this point, we have only considered paths evolving in $\mathbb{R}^d$
- It is known that the signature kernel is well-defined for paths evolving in a generic Hilbert space V
- We can thus consider the problem of generating spatiotemporal signals, or paths evolving over a function space $L^2(D)$ with given domain D
- Here the generator is given by a Neural SPDE
- A natural application in financial markets is limit order books (LOB), where $f(x, t)$ denotes the volume available at the price x at time t

Table of Contents

rBergomi model

Goal: Train a Neural SDE to learn the rough stochastic volatility model

$$dy_t = -\frac{1}{2}V_t dt + \sqrt{V_t} dW_t \quad \text{where} \quad d\xi_t^u = \xi_t^u \eta \sqrt{2\alpha+1}(u-t)^\alpha dB_t.$$

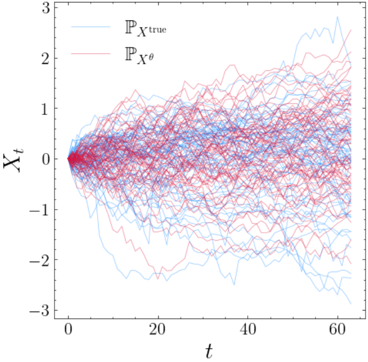
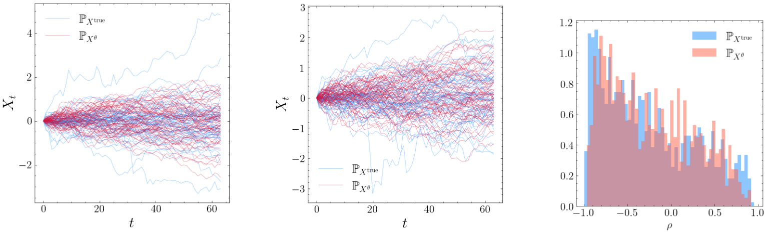


Figure 1: Neural SDE trained with ϕ_{sig} where $X^{\text{true}} \sim \text{rBergomi}(\eta, \rho, H)$.

Currency pairs: EUR/USD and USD/JPY



(a) EUR/USD **(b)** USD/JPY **(c)** Correlation coefficients

Figure 2: Neural SDE trained with ϕ_{sig}, EUR/USD and USD/JPY price pairs

Conditional generation: EURUSD

- Conditioning variables are time-augmented EUR/USD trajectories $\mathbb{Q} \sim x : [t_0 - dt, t_0] \to \mathbb{R}^2$
- Target variables: future trajectories $\mathbb{P}(\cdot|x) \sim X^{\text{true}} : [t_0, t_0 + dt'] \to \mathbb{R}^2$
- Encode conditioning variables via the order 5 log-signature of the input trajectories
- Train to minimise the conditional expected signature kernel score
- Many hyperparameters to consider... in general, path scaling is the most important (same is true for previous case!)

Conditional generation: EURUSD

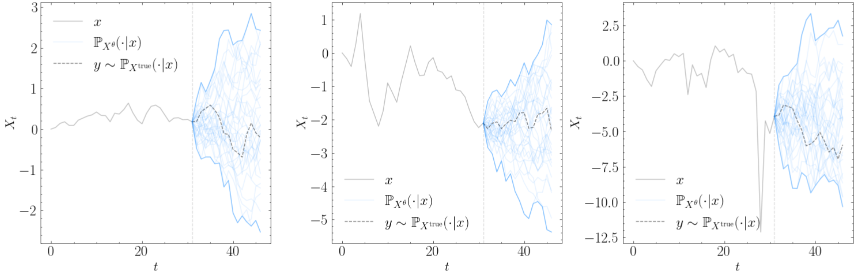


Figure 3: Given a conditioning path $x \sim \mathbb{Q}$, the generator provides (in blue) the conditional distribution $\mathbb{P}_{X^\theta}(\cdot|x)$. The dotted line gives the true path $y \sim \mathbb{P}_{X^{\text{true}}}(\cdot|x)$.

Resultant path is often captured in the envelope of the associated conditional distribution

Limit order books

Train a Neural SPDE model on NASDAQ LOB data, composing the signature kernel with three different SE-T type kernels

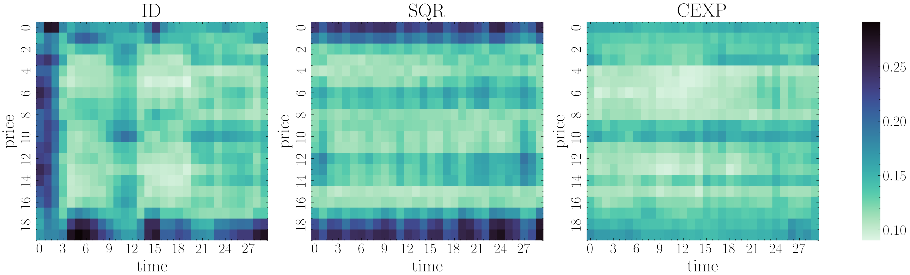


Figure 4: KS test average scores for each spatiotemporal marginal, 100 runs, NASDAQ data.

Table of Contents

Future work

- Enhancing the conditioning model by mixing both class- and non-class conditioning,
- Improving the non-class (continuous) conditioning process by encoding conditioning path via Neural CDE,
- Extending results in spatiotemporal setting to handle implied volatility surfaces,
- Additional penalty terms in the loss function (if necessary),
- Jump processes in driving noise.

Conclusion

- We have outlined a generalized, mesh-free training procedure for Neural SDEs/SPDEs using signature kernel scores
- We can easily extend to a conditional generation 1) without requiring assumptions about the true conditional distribution, and 2) allowing for flexible integration of any conditional variable
- Avoiding adversarializing the loss function means our method is more stable to train
- Computation cost of the signature kernel is linear in the path dimension, meaning memory requirements are not as onerous for high-dimensional paths
- Code is available at $\texttt{https}://\texttt{github.com}/\texttt{issaz}/\texttt{sigker}-\texttt{nsdes}$

Thank you for listening!

References I

Hans Buehler, Blanka Horvath, Terry Lyons, Imanol Perez Arribas, and Ben Wood.
A data-driven market simulator for small data environments.
arXiv preprint arXiv:2006.14498, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.
Generative adversarial networks.
Communications of the ACM, 63(11):139–144, 2020.

Tilmann Gneiting and Adrian E Raftery.
Strictly proper scoring rules, prediction, and estimation.
Journal of the American statistical Association, 102(477):359–378, 2007.

References II

Zacharia Issa, Blanka Horvath, Maud Lemercier, and Cristopher Salvi.
Non-adversarial training of neural sdes with signature kernel scores.
arXiv preprint arXiv:2305.16274, 2023.

Patrick Kidger, James Foster, Xuechen Li, Harald Oberhauser, and Terry Lyons.
Neural sdes as infinite-dimensional gans.
arXiv preprint arXiv:2102.03657, 2021.

Maud Lemercier, Cristopher Salvi, Thomas Cass, Edwin V Bonilla, Theodoros Damoulas, and Terry J Lyons.
Siggpde: Scaling sparse gaussian processes on sequential data.
In *International Conference on Machine Learning*, pages 6233–6242. PMLR, 2021.

References III

Hao Ni, Lukasz Szpruch, Marc Sabate-Vidales, Baoren Xiao, Magnus Wiese, and Shujian Liao.
Sig-wasserstein gans for time series generation.
In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–8, 2021.

Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao.
Conditional sig-wasserstein gans for time series generation.
arXiv preprint arXiv:2006.05421, 2020.

Lorenzo Pacchiardi and Ritabrata Dutta.
Likelihood-free inference with generative neural networks via scoring rule minimization.
arXiv preprint arXiv:2205.15784, 2022.

References IV

Cristopher Salvi, Thomas Cass, James Foster, Terry Lyons, and Weixin Yang.
The signature kernel is the solution of a goursat pde, 2021.

Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer.
Quant gans: deep generation of financial time series.
Quantitative Finance, 20(9):1419–1440, 2020.

Magnus Wiese, Ben Wood, Alexandre Pachoud, Ralf Korn, Hans Buehler, Phillip Murray, and Lianjun Bai.
Multi-asset spot and option market simulation.
arXiv preprint arXiv:2112.06823, 2021.