Generative modeling for time series via Schrödinger bridge

Huyên PHAM

Université Paris Cité (UPC), LPSM

based on joint work with Mohamed HAMDOUCHE (UPC, Qube RT) and Pierre HENRY-LABORDERE (Qube RT)

BIRS Workshop: Applications of stochastic control to finance and economics
April 30-May 5, 2023

What is Generative modeling (for time series)?

• Given the **distribution** μ **of the time series of some process** for which we have access only through data samples

- Sequential audio/video data
- Medical (Intensive Care Unit data) of a patient
- Renewable (wind and solar) energy production
- Finance and insurance: asset price, volatility surface, claim process, ...

▶ The goal is to design algorithms for

- learning μ
- generating real-looking samples of this data distribution:
  - Useful for improving clinical predictions, weather forecast
  - Fnancial industry: market stress test, market risk measurement, deep hedging, reinforcement learning for optimal trading
  - Data-driven approach for risk management

Generative AI

• Generative modeling (GM) has become a classical task in machine learning with several competing methods:

- **Likelihood-based models**: energy-based models (EBM), variational auto-encoders (VAE)
- **Implicit generative models**: generative adversarial network (GAN)
- **Score-based diffusion models**: last generation of generative AI models that outperforms GANs in terms of visual quality.

used notably in image processing with spectacular success (and controversies!), but mostly for static data/image (DALL-E, Midjourney, Stable diffusion, etc).

Challenges of GM for time series

- Temporal setting (sequential data) poses new challenges to GM:
  - capture the potentially complex dynamics of variables across time
  - not enough to learn the time marginals
  - learn the joint distribution without exploiting the sequential structure is also not sufficient

State-of-the-art generative methods for time series

- **Time series GAN** (Yoon et al. 19): combination of an *unsupervised adversarial* loss on real/synthetic data and *supervised* loss for generating sequential data
- **Quant GAN** (Wiese et al. 20): adversarial generator using temporal convolutional networks
- **Causal optimal transport GAN** (Xu et al. 20): adversarial generator using the adapted Wasserstein distance for processes
- **Conditional loss Euler generator** (Remlinger et al. 21): SDE representation of time series and minimizing the conditional distance between transition probabilities of real/synthetic samples
- **Signature** embedding of time series: Fermanian (19), Ni et al. (20), Buehler et al. (20).

State-of-the-art generative methods for time series

• **Time series GAN** (Yoon et al. 19): combination of an *unsupervised adversarial* loss on real/synthetic data and *supervised* loss for generating sequential data

• **Quant GAN** (Wiese et al. 20): adversarial generator using temporal convolutional networks

• **Causal optimal transport GAN** (Xu et al. 20): adversarial generator using the adapted Wasserstein distance for processes

• **Conditional loss Euler generator** (Remlinger et al. 21): SDE representation of time series and minimizing the conditional distance between transition probabilities of real/synthetic samples

• **Signature** embedding of time series: Fermanian (19), Ni et al. (20), Buehler et al. (20).

▶ We propose here a generative model based on Schrödinger bridge, in the spirit of score-based diffusion model, and adapted for time series.

Outline

Entropic interpolation of a time series distribution

Let $\mu \in \mathcal{P}((\mathbb{R}^d)^N)$ be the data time series distribution of some $\mathbb{R}^d$-valued process observed on a time grid $\mathcal{T} = \{t_i : i = 1, \ldots, N\}$. We set $t_0 = 0 < t_1 < \ldots < t_N = T$.

- **Schrödinger bridge time series problem**: Find a diffusion process X on $\mathbb{R}^d$ satisfying

$$\mathrm{d}X_t \quad = \quad \alpha_t \mathrm{d}t + \mathrm{d}W_t, \quad 0 \leq t \leq T, \quad X_0 = 0,$$

with a controlled drift α minimizing

$$\mathbb{E}\Big[\frac{1}{2}\int_0^T |\alpha_t|^2 \mathrm{d}t\Big]$$

and such that $(X_{t_1}, \ldots, X_{t_N}) \sim \mu$ (perfect match of the data distribution).

Assumptions

Assume that μ admits a density w.r.t. Lebesgue measure on $(\mathbb{R}^d)^N$, denoted by misuse of notation: $\mu(x_1, \ldots, x_N)$.

Denote by $\mu_{\mathcal{T}}^W$ the distribution of Brownian motion W on $\mathcal{T}$, i.e. of $(W_{t_1}, \ldots, W_{t_N})$, hence with density:

$$\mu_{\mathcal{T}}^W(x_1, \ldots, x_N) = \prod_{i=0}^{N-1} \frac{1}{\sqrt{2\pi(t_{i+1} - t_i)}} \exp\Big(- \frac{|x_{i+1} - x_i|^2}{2(t_{i+1} - t_i)} \Big).$$

• We assume that the relative entropy of μ w.r.t. $\mu_{\mathcal{T}}^W$ is finite, i.e.

$$\textbf{(H)} \qquad \mathcal{H}(\mu|\mu_{\mathcal{T}}^W) := \int \log \frac{\mu}{\mu_{\mathcal{T}}^W} \mathrm{d}\mu < \infty.$$

Remark: Assumption **(H)** is satisfied whenever μ comes from a process with

- Gaussian noise
- Heavy-tailed distribution but with second moment

Solution to Schrödinger bridge time series (SBTS)

Theorem (Diffusion SBTS)

Under **(H)**, the optimal controlled drift of the SBTS problem is in the **path-dependent** form:

$$\alpha_t^* \quad = \quad \mathrm{a}^*(t, X_t; \boldsymbol{X}_{t_i}), \quad t_i \leq t < t_{i+1}, \quad i = 0, \ldots, N-1,$$

where we set $\boldsymbol{X}_{t_i} := (X_{t_1}, \ldots, X_{t_i})$, and

$$\mathrm{a}^*(t, x; \boldsymbol{x}_i) = \nabla_x \log \mathbb{E}_{\mathbb{W}}\Big[\frac{\mu}{\mu_{\mathcal{T}}^W}(X_{t_1}, \ldots, X_{t_N})\Big|\boldsymbol{X}_{t_i} = \boldsymbol{x}_i, X_t = x\Big],$$

for $\boldsymbol{x}_i = (x_1, \ldots, x_i) \in (\mathbb{R}^d)^i$, $x \in \mathbb{R}^d$. Here $\mathbb{E}_{\mathbb{W}}$ denotes the expectation under which X is a Brownian motion by Girsanov's theorem.

Application: We have then a generative model for the time series with the diffusion

$$\mathrm{d}X_t \quad = \quad \mathrm{a}^*(t, X_t; (X_{t_i})_{t_i \leq t})\mathrm{d}t + \mathrm{d}W_t, \quad X_0 = 0,$$

by simulating e.g. from an Euler scheme $\rightarrow (X_{t_1}, \ldots, X_{t_N}) \sim \mu$.

Schrödinger drift function

Using Bayes formula, we derive the following expression:

$$\mathrm{a}^*(t, x; \boldsymbol{x}_i) = \frac{1}{t_{i+1} - t} \frac{\mathbb{E}_\mu \left[(X_{t_{i+1}} - x) F_i(t, x_i, x, X_{t_{i+1}}) \middle| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i \right]}{\mathbb{E}_\mu \left[F_i(t, x_i, x, X_{t_{i+1}}) \middle| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i \right]}, \tag{1}$$

for $t \in [t_i, t_{i+1})$, $i = 0, \ldots, N-1$, $\boldsymbol{x}_i \in (\mathbb{R}^d)^i$, $x \in \mathbb{R}^d$, where

$$F_i(t, x_i, x, x_{i+1}) = \exp\left(-\frac{(x_{i+1} - x)^2}{2(t_{i+1} - t)} + \frac{(x_{i+1} - x_i)^2}{2(t_{i+1} - t_i)} \right).$$

Here $\mathbb{E}_\mu[\cdot|\cdot]$ is the (conditional) expectation under $\mu \rightarrow$ One can then estimate the drift function by relying directly on **samples of data distribution** μ.

Remark: When μ is the distribution arising from a Markov chain, then the conditional expectations in (1) (and so the drift function) will depend on the past values $\boldsymbol{X}_{t_i} = (X_{t_1}, \ldots, X_{t_i})$ only via the last value X_{t_i}.

Kernel estimation of the drift

• Approximate the conditional expectation under μ by kernel regression methods:

▶ From data samples $\boldsymbol{X}^{(m)} = (X_{t_1}^{(m)}, \ldots, X_{t_N}^{(m)})$, $m = 1, \ldots, M$ from μ, the Nadaraya-Watson estimator of the drift function in (1) is given by

$$\hat{a}(t, x; \boldsymbol{x}_i) \quad = \quad \frac{1}{t_{i+1} - t} \frac{\sum_{m=1}^{M} (X_{t_{i+1}}^{(m)} - x) F_i(t, X_{t_i}^{(m)}, x, X_{t_{i+1}}^{(m)}) \prod_{j=1}^{i} K_h(x_j - X_{t_j}^{(m)})}{\sum_{m=1}^{M} F_i(t, X_{t_i}^{(m)}, x, X_{t_{i+1}}^{(m)}) \prod_{j=1}^{i} K_h(x_j - X_{t_j}^{(m)})},$$

for $\boldsymbol{x}_i = (x_1, \ldots, x_i)$, where K_h is a kernel function on $\mathbb{R}^d$ with bandwith $h > 0$. For lower time complexity reason, we choose the **quartic kernel** $K_h(x) = \frac{1}{h} K(\frac{x}{h})$ with

$$K(x) \quad = \quad (1 - |x|^2) 1_{|x| \leq 1}.$$

Outline

Fractional Brownian motion

Fractional Brownian motion (FBM) with Hurst index $H = 0.2$.

• Parameters: $M = 1000$, $N = 60$, $N^{\pi} = 100$ (number of time steps in Euler scheme), bandwith $h = 0.05$, Runtime for 1000 generated paths = 100s.

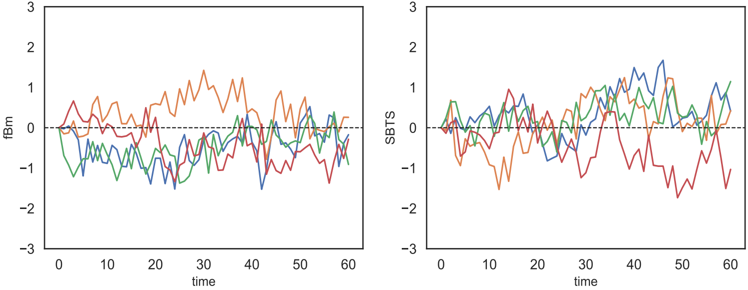

Figure: Four samples path of reference FBM (left) and generator SBTS (right)

Metrics for SBST generator vs FBM

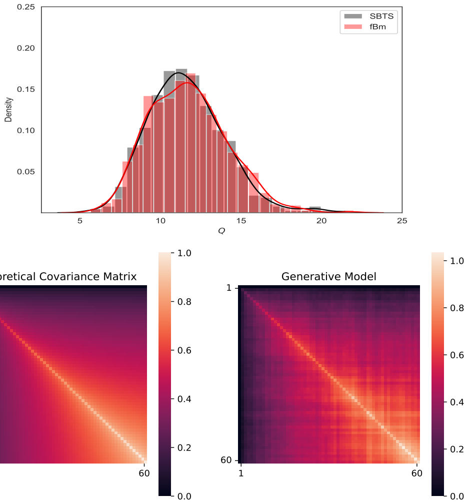

Figure: *Top*: Quadratic variation distribution $\sum_{i=1}^{N} |X_{t_{i+1}} - X_{t_i}|^2$ for $N = 60$.
Bottom: Covariance matrix for reference FBM and SBTS

Estimation of Hurst index

Standard estimator of Hurst index:

$$\hat{H} = \frac{1}{2}\left[1 - \frac{\log\left(\sum_{i=0}^{N-1} |X_{t_{i+1}} - X_{t_i}|^2\right)}{\log N}\right].$$

- From our generated SBTS with $N = 60$, we get:

$$\hat{H} = 0.2016, \quad \text{Std} = 0.004.$$

Real-world data sets on Apple

Data: stock prices of Apple from jan. 1, 2010 to jan. 30, 2020, with sliding window of $N = 60$ days.
$M = 2500$, $N^{\pi} = 100$, bandwith $h = 0.05$, runtime for 500 generated paths $= 100$s.

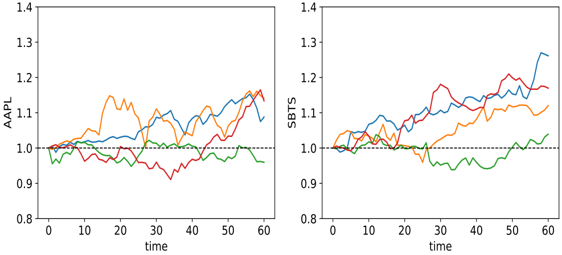

Figure: Four paths generated by SBTS (*right*) vs real ones from Apple (*left*).

Metrics for SBST generator vs real-data Apple

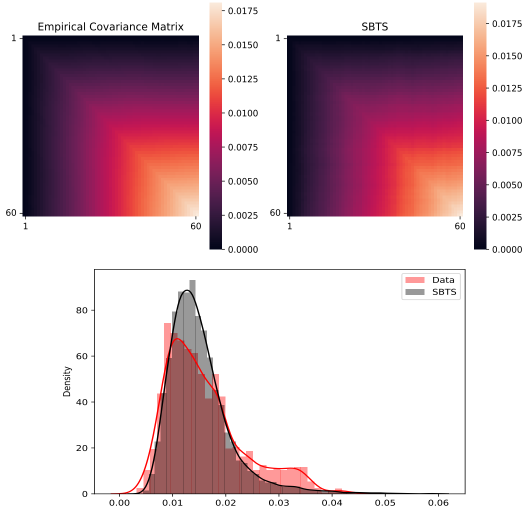

Figure: *Top*: Covariance matrix for real-data and generative SBTS. *Bottom*: Quadratic variation distribution.

Application to deep hedging

- Consider a ATM call option on Apple: $g(S_T) = (S_T - K)_+$, and we search for a price p^* and hedging strategy Δ^* minimizing the quadratic criterion (loss function):

$$(p, \Delta) \quad \mapsto \quad \mathbb{E}\Big| \underbrace{p + \sum_{i=0}^{N-1} \Delta_{t_i}(S_{t_{i+1}} - S_{t_i}) - g(S_T)}_{\text{PnL}} \Big|^2 \;=\; \text{replication} \;\; \text{error}$$

Application to deep hedging

- Consider a ATM call option on Apple: $g(S_T) = (S_T - K)_+$, and we search for a price p^* and hedging strategy Δ^* minimizing the quadratic criterion (loss function):

$$(p, \Delta) \quad \mapsto \quad \mathbb{E}\Big| \underbrace{p + \sum_{i=0}^{N-1} \Delta_{t_i}(S_{t_{i+1}} - S_{t_i}) - g(S_T)}_{\text{PnL}} \Big|^2 \; = \; \text{replication} \;\; \text{error}$$

▶ We parametrize Δ by a LSTM network that is trained from synthetic data sets produced by SBTS, and we compare the results with real-data sets.

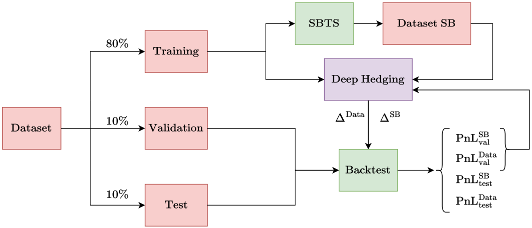

Figure: Procedure of backtest for deep hedging

Comparison of the PnL and replication error with real-data and generative SBTS

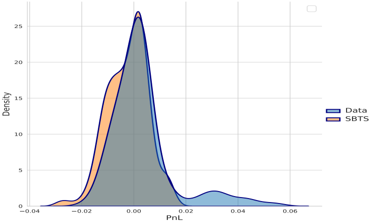

Figure: Deep hedging PnL distribution from test set

		Training Set		Test Set	
	Price	Mean	Std	Mean	Std
Data	0.0415	0.0008	0.0098	0.003	0.012
SBTS	0.0471	0.0004	0.0109	-0.0024	**0.0076**

Table: Mean of PnL and its Std (replication error).

Concluding remarks

- Novel generative model for time series based on Schrödinger bridge (SB) approach:
  - Solution described by a forward stochastic differential equation (SDE) over a finite period, which matchs perfectly the data distribution
  - Path-dependent drift capturing the temporal dynamics of the time series distribution
  - Drift estimated by kernel regression (possibly by vectorization): practical and low-cost computationally

- Compared to GAN type methods, the simulation of synthetic samples from SB is much faster as it does not require training of neural networks.

- Series of numerical experiments, including financial applications with real-data, to illustrate the performance and accuracy of our generative SBTS. Further tests to be developed ...

- Limitations and further developments:
  - Solution obtained under the finiteness of the relative entropy of the time series distribution: may be violated for heavy-tailed distribution (no second-order moment)
  - Numerical instability in very high dimension (e.g. pixels in image)

Reference

M. Hamdouche, P. Henry-Labordère, H. Pham. Generative modeling for time series via Schrödinger bridge. SSRN 4412434, arXiv:2304.05093

THANK YOU FOR YOUR ATTENTION