Lab work n°3 : On time series generation via Schrödinger Bridge

Alexandre ALOUADI

August 9, 2025

Contents

1 Background

We recall the Schrödinger Bridge problem for time series introduced in [1]. Let μ be the distribution of a time series valued in $\mathbb{R}^d$ of which we can observe samples over a discrete time grid $\mathrm{T} = \{t_1, \cdots, t_N = T\}$. We want to construct a model capable of generating time series samples that follow the distribution $\mu \in \mathcal{P}((\mathbb{R}^d)^N)$ given real observations.

This problem, noted Schrödinger Bridge Time Series (**SBTS**) is formulated as follows:

$$\min_{\alpha} \mathbb{E}_{\mathbb{P}} \left[\int_0^T \|\alpha_t\|^2 \, dt \right] \tag{1.1}$$

such that $dX_t = \alpha_t dt + dW_t^{\mathbb{P}}$ with W a Brownian motion under $\mathbb{P}$, $X_0 = \mathbf{0}, (X_{t_1}, \cdots, X_{t_N}) \overset{\mathbb{P}}{\sim} \mu$.

Theorem 1 *[1] The diffusion process* $X_t = \int_0^t \alpha_s^* ds + W_t, \quad 0 \leq t \leq T$, *with* α^* *defined as*

$$\alpha_t^* = a^*(t, X_t; \boldsymbol{X}_{\eta(t)}), \quad 0 \leq t < T,$$

solves the ***SBTS*** *problem* (1.1), *with* $\eta(t) = \max\{t_i : t_i \leq t\}$, $a^*(t, x; \boldsymbol{x}_i)$, *for* $t \in [t_i, t_{i+1}[$, $\boldsymbol{x}_i = (x_1, \ldots, x_i) \in (\mathbb{R}^d)^i$, $x \in \mathbb{R}^d$, *given by*

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

where

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

To estimate the drift, one can employ a kernel density estimation method using M data samples $\boldsymbol{X}_{t_N}^m = (X_{t_1}^m, \cdots, X_{t_N}^m), m = 1, \cdots, M$:

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

with $t \in [t_i, t_{i+1}[$, $\boldsymbol{x}_i \in (\mathbb{R}^d)^i$, $i \in \{1, \cdots N-1\}$ and K_h a kernel.

2 First numerical experiments

2.1 What kernel can be used in the expression of $\hat{a}$?

In this lab, we will use the *quartic kernel*, which assigns higher weights to points closer to the target and smoothly decreases to zero at the boundary. It is defined as:

$$K_h(x) = \frac{1}{h^d} \left(1 - \left\| \frac{x}{h} \right\|^2 \right)^2 \mathbb{1}_{\{\|x\| < h\}},$$

for $x \in \mathbb{R}^d$, where $h > 0$ is the bandwidth parameter that controls the size of the neighborhood.

2.2 Using the algorithm provided below, implement the SBTS method.

Algorithm 1 Diffusion algorithm between $[t_0, t_N]$ pseudo-code

Require: Samples $X^m_{t_0:t_N}$, $m = 1, \ldots, M$; N^π; Δt_i; $h > 0$
  Initialization: initial state $x_0 = 0$
  for $i = 0, \cdots, N-1$ **do**
    Initialize state $y_0 = x_i$
    for $k = 0, \ldots, N^\pi - 1$ **do**
      Compute $\hat{a}(t^\pi_{k,i}, y_k, \boldsymbol{x}_i)$ using (1)
      Sample $\varepsilon_k \sim \mathcal{N}(0, I_d)$ and update
    $y_{k+1} = y_k + \frac{\Delta t_i}{N^\pi}\hat{a}(t^\pi_{k,i}, y_k, \boldsymbol{x}_i) + \sqrt{\frac{\Delta t_i}{N^\pi}}\varepsilon_k$
    end for
    Set $x_{i+1} = y_{N^\pi}$
  end for
  Return $x_0, \cdots, x_N$

One may notice that certain terms in the $\hat{a}$ function, such as the product of kernels, can be precomputed prior to the loop over $k = 0, \ldots, N^\pi - 1$, as they depend solely on past values and remain constant within the loop.

2.3 Generate GARCH process data

Consider the following GARCH process:

$$\begin{cases} X_{t_{i+1}} = \sigma_{t_{i+1}}\, \varepsilon_{t_{i+1}} \\ \sigma_{t_{i+1}} = \alpha_0 + \alpha_1 X^2_{t_i} + \alpha_2 X^2_{t_{i-1}}, \quad i = 1, \ldots, N \end{cases}$$

with parameters $\alpha_0 = 5$, $\alpha_1 = 0.4$, $\alpha_2 = 0.1$, and $\varepsilon_{t_{i+1}} \sim \mathcal{N}(0, 0.1)$ i.i.d. Gaussian noise.

Starting from 1000 realizations of this GARCH process, each of length $N = 60$, we generate 1000 synthetic samples of the same length using a diffusion-based model with $\Delta t_i = \frac{1}{252}$, $h = 0.2$, and $N^\pi = 100$.

How long does it take to generate 1000 synthetic samples?

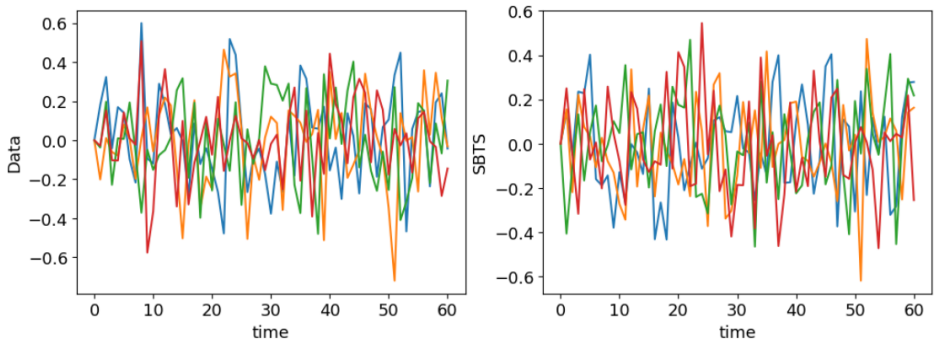

Figure 1: GARCH process expected result. We plot 4 randomly selected samples from real and SBTS data.

2.4 Evaluate the Quality of Your Generated Samples

To assess the quality of the samples you have generated, you may:

1. Compute usuals statistics such as the mean, standard deviation, minimum and maximum values, as well as the 1% and 99% percentiles.
2. Compare the covariance matrices computed from the real data and the SBTS-generated data.
3. Compare the distributions of the quadratic variation over the real and SBTS samples, where the quadratic variation is defined as

$$Q(X) = \sum_i |X_{t_{i+1}} - X_{t_i}|^2.$$

1% Data	1% SBTS	99% Data	99% SBTS	Mean Data	Mean SBTS	Std Data	Std SBTS	Min Data	Min SBTS	Max Data	Max SBTS
-0.52	-0.519	0.522	0.523	0.002	0.002	0.224	0.224	-0.995	-0.931	0.908	0.898

Figure 2: Expected statistics of real vs SBTS samples

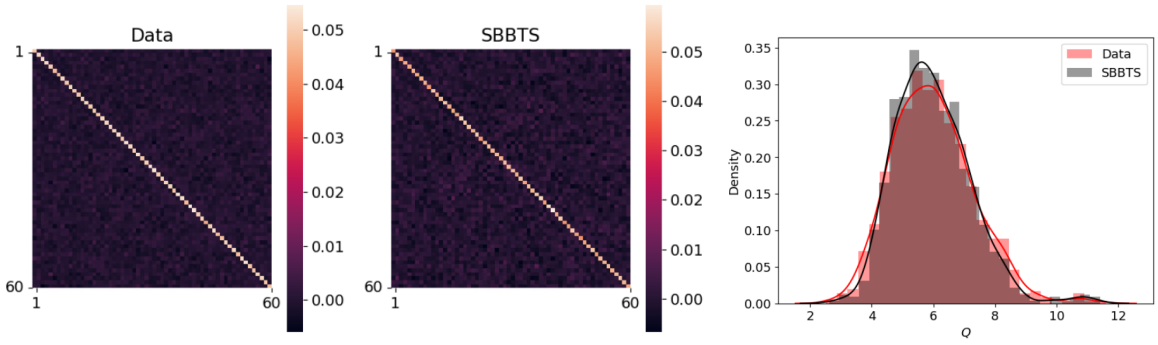

Figure 3: Expected covariance matrix and quadratic variation distribution

2.5 Discuss about the impact oh h and Δt_i.

2.6 Discuss the Impact of h and Δt_i

Experiment with different values of the bandwidth parameter h and the time increments Δt_i for various series lengths N.

What effects do you observe on the model's performance or the generated data? How could you improve the model based on these observations?

3 Ornstein-Uhlenbeck Process (OU)

3.1 Some reminders on the OU

We recall that an Ornstein-Uhlenbeck process is defined as :

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t$$

where $\theta > 0, \mu, \sigma > 0$ are the parameters and W_t a brownian motion. Give the solution of this SDE. What discretization can we use to simulate an OU ?

3.2 Parameter Estimation: Theoretical Framework

How can you estimate the parameters of an OU process using the Maximum Likelihood Estimation method? Derive the corresponding formulas.

3.3 Generating SBTS Samples

Consider the parameters $\theta = 1.5$, $\mu = 1$, and $\sigma = 0.2$. Given 1000 real samples, generate 1000 SBTS samples using, for example, $h = 0.2$ and $\Delta t_i = \frac{1}{252}$. Feel free to experiment with other

values of h and Δt_i to improve the results.

Hint: Convert the original time series into log-returns, then use SBTS to generate log-returns, and finally invert them back to the original price scale. Why is this transformation necessary? What would happen if you generated the raw prices directly? Provide a brief explanation.

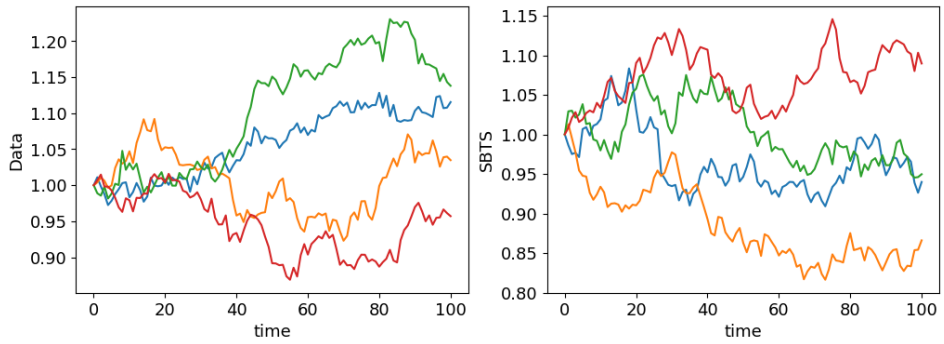

Figure 4: OU data expected result. We plot 4 randomly selected samples from real and SBTS data.

3.4 Parameters estimations: practical test

Estimate for each real and SBTS samples the parameters using 3.2) and plot the distribution for each parameters.

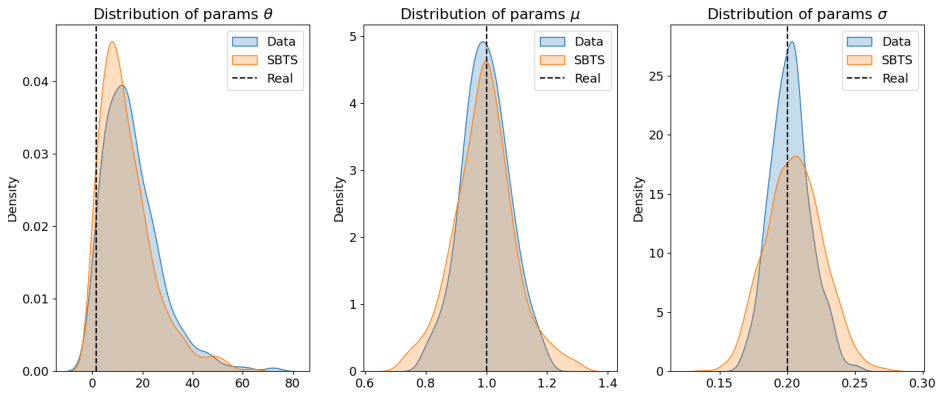

Figure 5: Distribution of estimated Ornstein-Uhlenbeck parameters using MLE for fixed parameters expected result. The orange and blue densities correspond to the SBTS samples and data samples, respectively, while the black line represents the true parameter.

Discuss the results.

4 Open-Ended Project: Using Synthetic Time Series Data

In this part of the lab, you have complete freedom to choose a time series dataset and design your own mini-project focused on data augmentation using SBTS. The primary goal is to demonstrate how SBTS-generated samples can help improve model performance on a task of your choice.

You have full freedom to select:

- Any time series dataset (univariate or multivariate) from any domain, but be careful with high-dimensional data, which can be challenging.
- Any problem or task where synthetic data can be beneficial. This could include, but is not limited to:
  - Data augmentation to improve model training,
  - Data imputation or gap filling,
  - Scenario simulation or stress testing,
  - Anomaly detection,
  - Any other use case relevant to your interests.

Your project should include:

- A clear description of your chosen dataset and task.
- An explanation of how and why you use synthetic data generation.
- Implementation details, including parameter choices for the SBTS method.
- Model training and evaluation protocols, with comparisons when applicable.
- An analysis discussing the impact and benefits (or limitations) of synthetic data in your context.

Note: Keep in mind that **SBTS assumes stationary time series**, so appropriate preprocessing or transformations may be necessary.

Creativity, thoroughness, and critical analysis will be important criteria for assessment. Feel free to explore and justify your choices freely.

To guide you, here is an example workflow:

1. Choose a stock price dataset, for instance daily prices of BNP Paribas, and convert it into log-returns.
2. Generate new samples using SBTS.
3. Train a model to predict the next log-return given the previous P log-returns. We could train several model, each with different combinaison of data, e.g., real data only vs SBTS only ...
4. Compare the model's performance on a test set. Suitable evaluation metrics include MAE, MSE, etc.

References

[1] Mohamed Hamdouche, Pierre Henry-Labordere, and Huyên Pham. Generative modeling for time series via Schrödinger bridge, 2023.