Parameter estimation of stochastic processes with neural networks

Dávid Kovács

Supervisor: András Lukács

Introduction

Lack of estimators

There are several stochastic processes that are important in theoretical finance, but barely usable in practice, due to the lack of methods to estimate their parameters.

Abundance of generators

However, often it is possible to generate (discretized) trajectories from the processes in question. If we can generate trajectories of a stochastic process, then we can also train neural networks on the generated trajectories for the parameter estimation of the process.

Bayesian estimation with deep learning

Let Θ be the set of the possible parameters and let $P : \mathcal{B}(\Theta) \to [0,1]$ be the prior distribution. Then $\forall \vartheta \in \Theta$ let $Q_\vartheta : \mathcal{B}(\mathcal{X}) \to [0,1]$ be the distribution indexed by ϑ . Moreover let $L : \Theta^2 \to \mathbb{R}_+$ be a loss function. $\forall \vartheta \in \Theta$ let $G_\vartheta \sim Q_\vartheta$ be a generator. We can then sample $\vartheta_1, \vartheta_2, \dots, \vartheta_n$ from distribution P. Then we can create the learning dataset $(G_{\vartheta_1}, \vartheta_1), (G_{\vartheta_2}, \vartheta_2), \dots, (G_{\vartheta_n}, \vartheta_n)$ where n can be arbitrarily large.

The goal of training a neural network on this set with loss function L is trying to find the estimator S that minimizes the Bayesian risk

$$R_P(S) = \int_\vartheta \int_{\mathcal{X}} L(t, S(x)) Q_t(dx)\, P(dt).$$

Ideal learning dataset

Perfect labeling

The data is labeled perfectly, assuming that the generators are functioning correctly.

Infinite unique training data

If the generation is fast enough then we can generate enough data to eliminate the need for reusing any (G_ϑ, ϑ) pairs. In practice this means that during the training process every batch is generated on the fly and one epoch simply means a certain number of training pairs. Using unique data means that overfitting is not possible, hence training losses can be treated as validation losses.

Estimation of shape, location and scale with deep learning

Let Θ be the set of shape parameters, now let $\Theta^* = \Theta \times \mathbb{R} \times \mathbb{R}_+$ be the location and scale extended parameter set. For any $(\vartheta, \nu, \lambda) = \vartheta^* \in \Theta^*$, Q_{ϑ^*} is defined as the distribution of $\lambda X + \nu \cdot v_0$, where $X \sim Q_\vartheta$ and $v_0 \in \mathcal{X}$ is the predefined location basis. We want to estimate ϑ, ν and λ.

We will solve the problem defined by Θ^*, while actually working on just Θ. We achieve this by defining neural modules, that combined appropriately result in neural networks, that can extrapolate to Θ^* after being taught on Θ.

Extrapolation from Θ to Θ^*

Let $X_0 \sim Q_\vartheta$ for some $\vartheta \in \Theta$. And let $X = \lambda X_0 + \nu v_0$.

$$\mathcal{M}^\vartheta(X) = \mathcal{M}^\vartheta(X_0)$$
$$(\mathcal{M}^\vartheta(X) - \vartheta)^2 = (\mathcal{M}^\vartheta(X_0) - \vartheta)^2$$

$$\mathcal{M}^\lambda(X) = \lambda \mathcal{M}^\lambda(X_0)$$
$$(\mathcal{M}^\lambda(X) - \lambda)^2 = \lambda^2(\mathcal{M}^\lambda(X_0) - 1)^2$$

$$\mathcal{M}^\nu(X) = \lambda \mathcal{M}^\nu(X_0) + \nu$$
$$(\mathcal{M}^\nu(X) - \nu)^2 = \lambda^2(\mathcal{M}^\nu(X_0) - 0)^2$$

Homogeneous modules ($^\lambda$)

Homogeneity

$M(\lambda x) = \lambda M(x)$

$E^\lambda : seq \rightarrow vec\_seq$ (embedding)

Let E^λ be a multilayer 1D convolution module with no bias and PReLU activations between the layers.

$A^\lambda : vec\_seq \rightarrow vec$ (average)

Let A^λ be an adaptive average pooling layer.

$P^\lambda : vec \rightarrow scal$ (projection)

Let P^λ be an MLP with no bias and PReLU activation.

Scale-invariant modules (λ)

Scale invariance

$M(\lambda x) = M(x)$

$N_\lambda : vec \to vec$ (normalization)

$$N_\lambda(x) = \frac{x}{P^\lambda(x)}$$

Location-additive modules ($^{\nu}$)

Location additivity

$$M(x + \nu v_0) = M(x) + \nu$$

$M^{\nu,\lambda} : seq \to scal$

$$M^{\nu,\lambda}(x) = \frac{B(x)}{B(v_0)},$$

where $B : seq \to scal$ is the composition of a single 1D convolution layer, an adaptive average layer and a single fully connected linear layer. And by linearity, we mean having no bias and no activation function.

Location-invariant modules ($_\nu$)

Location invariance

$$M(x + \nu v_0) = M(x)$$

$C_\nu^\lambda : seq \to seq$ (centering)

$$C_\nu^\lambda(x) = x - M^{\nu,\sigma}(x) \cdot v_0$$

Putting the modules together

Estimator for ϑ

$$\mathcal{M}^{\vartheta} := P \circ N_{\lambda} \circ A^{\lambda} \circ E^{\lambda} \circ C_{\nu}^{\lambda},$$

where $P : vec \to scal$ is an MLP.

Estimator for λ

$$\mathcal{M}^{\lambda} = P^{\lambda} \circ A^{\lambda} \circ E^{\lambda} \circ C_{\nu}^{\lambda}$$

Estimator for ν

$$\mathcal{M}^{\nu} = P^{\lambda} \circ A^{\lambda} \circ E^{\lambda} \circ C_{\nu}^{\lambda} + M^{\nu,\lambda}$$

Fractional Brownian motion

We want to estimate the parameters of the process $\left(\sigma B_t^H + \mu t\right)_{t\in[0,T]}$, where $H \in (0,1)$, $\mu \in \mathbb{R}$, $\sigma \in \mathbb{R}_+$ and B^H is a fractional Brownian motion with Hurst exponent H.

We have shape parameter $\vartheta = H$ parametrizing $(B_t^H)_{t\in[0,1]}$. And adding scale parameter $\lambda = \sigma T^H$, and location parameter $\nu = \mu T$ with basis $v_0 = (t)_{t\in[0,1]}$ yields every process parametrized above.

$$\lambda \cdot \left(B_t^H\right)_{t\in[0,1]} + \nu \cdot (t)_{t\in[0,1]} = \left(\sigma T^H B_t^H + \mu T t\right)_{t\in[0,1]} \stackrel{d}{=} \left(\sigma B_t^H + \mu t\right)_{t\in[0,T]}$$

Fractional Brownian motion

$\Theta = (0,1)$ and we extend Θ into Θ^*. We only need a prior distribution on Θ, for which a reasonable choice is $U(0,1)$. And obviously we cannot work with $\left(\sigma B_t^H\right)_{t\in[0,1]}$ and $(t)_{t\in[0,1]}$, we need to discretize them in $\{0, 1/N, 2/N, \ldots, 1-1/N, 1\}$, assuming we have N equidistant observations.

We have neural networks $\mathcal{M}^\vartheta$, $\mathcal{M}^\lambda$ and $\mathcal{M}^\nu$ readily available for the estimation of $\vartheta = H$, $\lambda = \sigma T^H$ and $\nu = \mu T$.

Data generation and training

$\forall H \in (0,1)$ we need to generate unscaled and undrifted fBm sequences on $[0,1]$ with Hurst exponent H. For this purpose, we used a Python implementation of the method described in [2]. The Python version was implemented by I. Ivkovic and D. J. Boros.

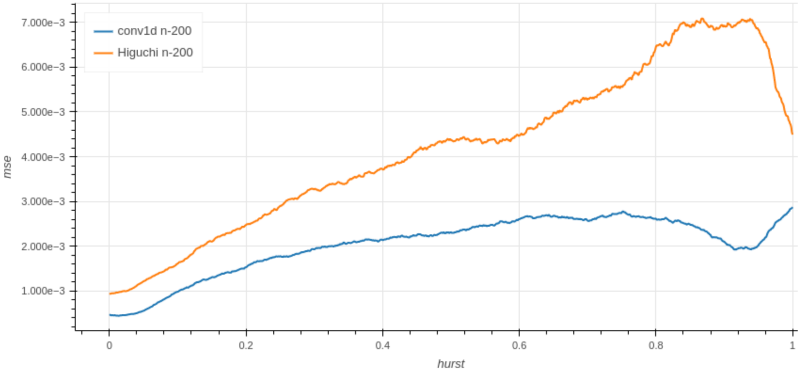
Having a fast generator is highly beneficical, because to achieve the results documented in the next subsection, we needed to train the model for 300 epochs. Here one epoch means 100000 unique input sequences (and of course every epoch is unique, there is no reusage of data).

Results

We pit $\mathcal{M}^\vartheta$ against the Higuchi method [1], which is a statistical estimator for H, that works on scaled (but not drifted) fBm sequences. $\mathcal{M}^\vartheta_N$ denotes $\mathcal{M}^\vartheta$ trained on series of length N.

Squared Bayesian risk of $\hat{H}$	Higuchi	$\mathcal{M}^\vartheta_N$	$\mathcal{M}^\vartheta_{12800}$
$N = 200$	0.00416	0.00200	0.00214
$N = 400$	0.00197	0.00097	0.00106
$N = 800$	0.00105	0.000475	0.000526
$N = 1600$	0.00058	0.000237	0.000249
$N = 3200$	0.000353	0.000125	0.000125
$N = 6400$	0.000231	0.000064	0.000064
$N = 12800$	0.000151	0.0000326	0.0000326

Results (uniform superiority even for N = 200)



Thank you for your attention!

- Higuchi, T. (1990). Relationship between the fractal dimension and the power law index for a time series: A numerical investigation. Physica D: Nonlinear Phenomena.
- Kroese, D. P. & Botev, Z. I. (2014). Spatial Process Simulation. Stochastic Geometry, Spatial Statistics and Random Fields(pp. 369-404). Springer International Publishing.
- Yaozhong Hu, David Nualart & Hongjuan Zhou (2017). Parameter estimation for fractional Ornstein–Uhlenbeck processes of general Hurst parameter. https://arxiv.org/abs/1703.09372.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). Deep learning. MIT Press.