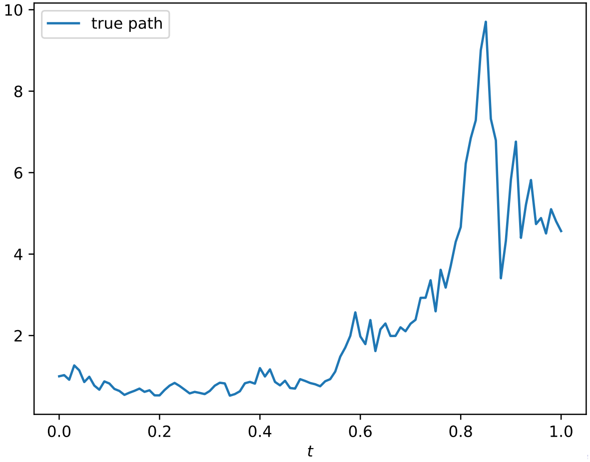
Neural Jump Ordinary Differential Equations: Consistent Continuous-Time Prediction and Filtering

Calypso Herrera   Florian Krach   Josef Teichmann
Department of Mathematics, ETH Zürich

ICLR - May 2021

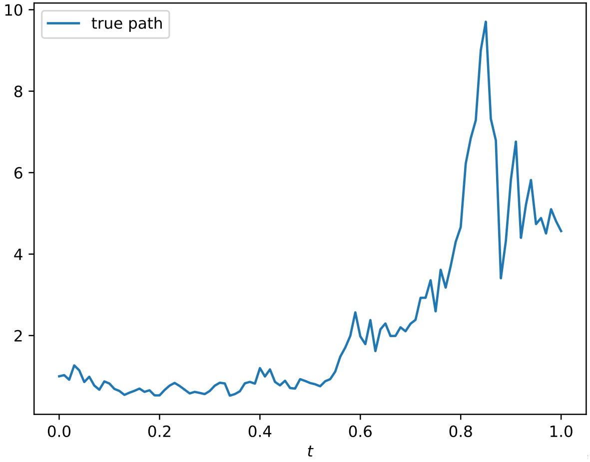
Problem Statement

- Markovian stochastic process $dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t$



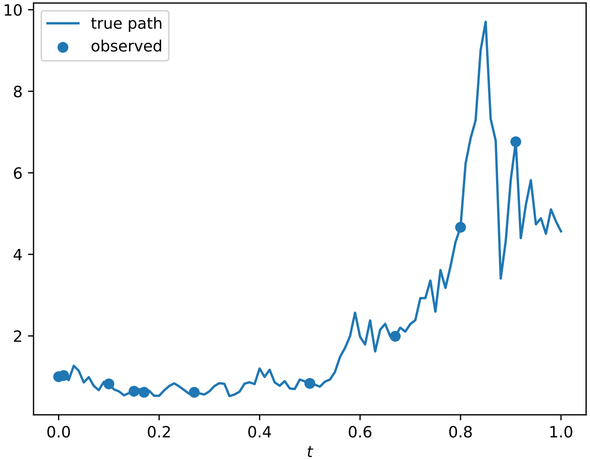
Problem Statement

- Markovian stochastic process $dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t$
- training set of N independent realisations $x_1^{(j)}, \ldots, x_{n_j}^{(j)} \in \mathbb{R}^{d_X}$



Problem Statement

- Markovian stochastic process $dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t$
- training set of N independent realisations $x_1^{(j)}, \ldots, x_{n_j}^{(j)} \in \mathbb{R}^{d_X}$
- observed at n_j random observation times $t_1^{(j)}, \ldots, t_{n_j}^{(j)} \in [0, T]$

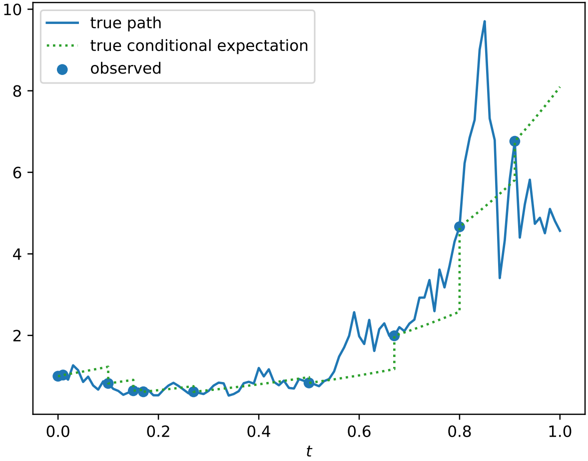


Problem Statement

- Markovian stochastic process $dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t$
- training set of N independent realisations $x_1^{(j)}, \dots, x_{n_j}^{(j)} \in \mathbb{R}^{d_X}$
- observed at n_j random observation times $t_1^{(j)}, \dots, t_{n_j}^{(j)} \in [0, T]$

Goal: learn the L^2-optimal prediction of X given last observation x_t:

$$f(x_t, t, s) := E[X_{t+s} | X_t = x_t]$$



Background

- Recurrent Neural Network

$$h_{i+1} := \mathrm{RNNCell}(h_i, x_{i+1}).$$

Background

- Recurrent Neural Network

$$h_{i+1} := \mathrm{RNNCell}(h_i, x_{i+1}).$$

- Neural Ordinary Differential Equation

$$h_t := h_0 + \int_{t_0}^{t} f_\theta(h_s, s)ds = \mathrm{ODESolve}(f_\theta, h_0, (t_0, t)),$$

Background

- Recurrent Neural Network

$$h_{i+1} := \mathrm{RNNCell}(h_i, x_{i+1}).$$

- Neural Ordinary Differential Equation

$$h_t := h_0 + \int_{t_0}^{t} f_\theta(h_s, s)ds = \mathrm{ODESolve}(f_\theta, h_0, (t_0, t)),$$

- ODE-RNN and GRU-ODE-Bayes

$$\begin{cases} h_i' & := & \mathrm{ODESolve}(f_\theta, h_{i-1}, (t_{i-1}, t_i)) \\ h_i & := & \mathrm{RNNCell}(h_i', x_i)\,. \end{cases}$$

Background

- Recurrent Neural Network

$$h_{i+1} := \mathrm{RNNCell}(h_i, x_{i+1}).$$

- Neural Ordinary Differential Equation

$$h_t := h_0 + \int_{t_0}^{t} f_\theta(h_s, s)ds = \mathrm{ODESolve}(f_\theta, h_0, (t_0, t)),$$

- ODE-RNN and GRU-ODE-Bayes

$$\begin{cases} h_i' & := & \mathrm{ODESolve}(f_\theta, h_{i-1}, (t_{i-1}, t_i)) \\ h_i & := & \mathrm{RNNCell}(h_i', x_i)\,. \end{cases}$$

- no theoretical guarantees exist

Neural Jump ODE

- jumpNN instead of RNN & additional inputs for neural ODE

$$\begin{cases} h_i' & := & \mathrm{ODESolve}(f_\theta, (h_{i-1}, x_{i-1}, t_{i-1}, t - t_{i-1}), (t_{i-1}, t_i)) \\ h_i & := & \mathrm{jumpNN}(x_i) \\ y_{i-} & := & \mathrm{outputNN}(h_i'), \quad y_i := \mathrm{outputNN}(h_i) \end{cases}$$

Neural Jump ODE

- jumpNN instead of RNN & additional inputs for neural ODE

$$\begin{cases} h'_i & := \quad \text{ODESolve}(f_\theta, (h_{i-1}, x_{i-1}, t_{i-1}, t - t_{i-1}), (t_{i-1}, t_i)) \\ h_i & := \quad \text{jumpNN}(x_i) \\ y_{i-} & := \quad \text{outputNN}(h'_i), \quad y_i := \text{outputNN}(h_i) \end{cases}$$

- new objective function

$$\hat{\Phi}_N(\theta) := \underbrace{\frac{1}{N}\sum_{j=1}^{N}}_{\text{paths}} \underbrace{\frac{1}{n^j}\sum_{i=1}^{n^j}}_{\text{dates}} \Bigg(\underbrace{|x_i^{(j)} - y_i^{(j)}|}_{\substack{\text{jump part} \\ \text{at observations}}} + \underbrace{|y_i^{(j)} - y_{i-}^{(j)}|}_{\substack{\text{continuous part} \\ \text{between two observations}}} \Bigg)^2$$

Theoretical Convergence Guarantees

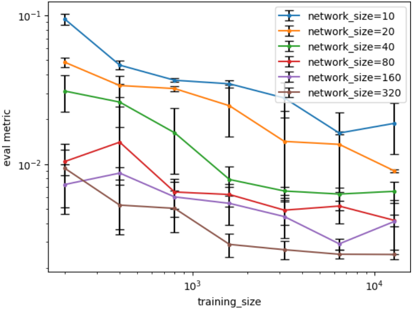
Theorem (informal)

Neural Jump ODE output Y converges to L^2-optimal prediction as the number of paths N and network sizes M increase to ∞.

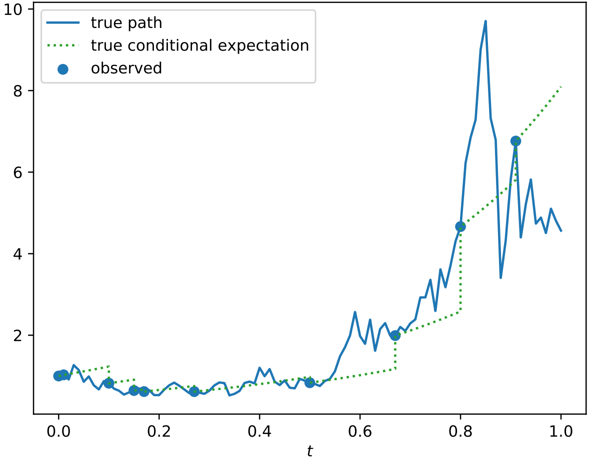
Theoretical Convergence Guarantees

Theorem (informal)

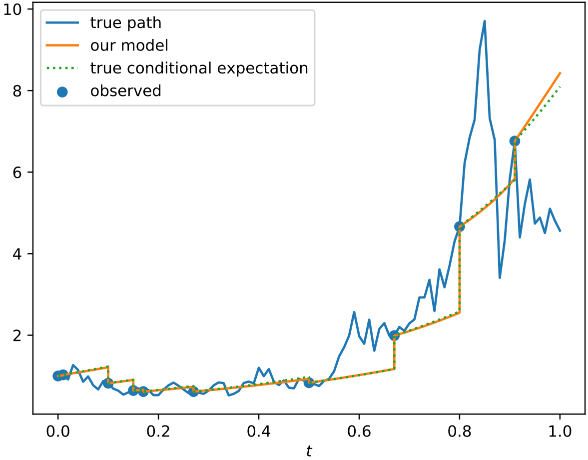
Neural Jump ODE output Y converges to L^2-optimal prediction as the number of paths N and network sizes M increase to ∞.



Experiments on synthetic Heston dataset

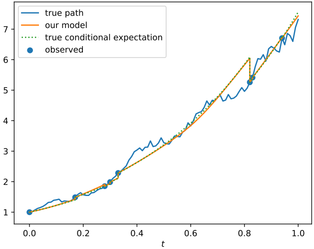


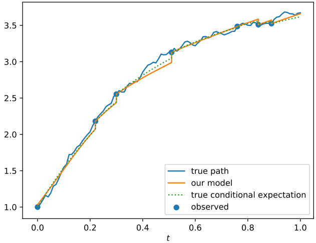
Experiments on synthetic Heston dataset



Further experiments on synthetic datasets

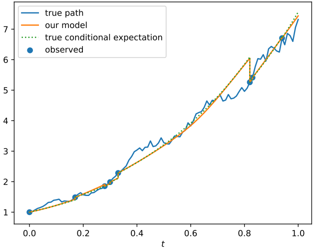
- Black Scholes (left) & Ornstein Uhlenbeck (right) dataset

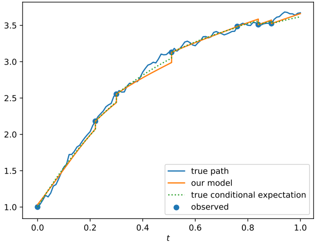




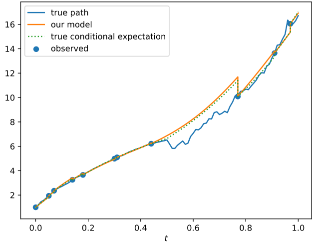
Further experiments on synthetic datasets

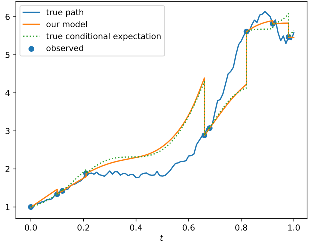
- Black Scholes (left) & Ornstein Uhlenbeck (right) dataset





- switching regime (left) & time-dependent drift μ (right)





Experiments on real world datasets

- climate forecast on USHCN dataset

	USHCN – MSE	# params
neural ODE-VAE	0.96 ± 0.11	-
neural ODE-VAE-MASK	0.83 ± 0.10	-
sequential VAE	0.83 ± 0.07	-
GRU-SIMPLE	0.75 ± 0.12	-
GRU-D	0.53 ± 0.06	-
T-LSTM	0.59 ± 0.11	-
GRU-ODE-Bayes	0.43 ± 0.07	$42'640$
NJ-ODE (S)	0.45 ± 0.06	$10'925$
NJ-ODE (L)	$\mathbf{0.40} \pm \mathbf{0.07}$	$571'305$

Experiments on real world datasets

- climate forecast on USHCN dataset

	USHCN – MSE	# params
neural ODE-VAE	0.96 ± 0.11	-
neural ODE-VAE-MASK	0.83 ± 0.10	-
sequential VAE	0.83 ± 0.07	-
GRU-SIMPLE	0.75 ± 0.12	-
GRU-D	0.53 ± 0.06	-
T-LSTM	0.59 ± 0.11	-
GRU-ODE-Bayes	0.43 ± 0.07	$42'640$
NJ-ODE (S)	0.45 ± 0.06	$10'925$
NJ-ODE (L)	$\mathbf{0.40} \pm \mathbf{0.07}$	$571'305$

- PhysioNet dataset

	Physionet – MSE ($\times 10^{-3}$)	# params
RNN-VAE	3.055 ± 0.145	-
Latent ODE (RNN enc.)	3.162 ± 0.052	-
Latent ODE (ODE enc)	2.231 ± 0.029	$163'972$
Latent ODE + Poisson	2.208 ± 0.050	$181'723$
NJ-ODE	$\mathbf{1.945} \pm \mathbf{0.007}$	$24'423$

Conclusion

- continuous-time prediction and filtering with Neural Jump ODE

Conclusion

- continuous-time prediction and filtering with Neural Jump ODE
- first model with theoretical convergence guarantees

Conclusion

- continuous-time prediction and filtering with Neural Jump ODE
- first model with theoretical convergence guarantees
- outperforming existing models on different empirical examples

Conclusion

- continuous-time prediction and filtering with Neural Jump ODE
- first model with theoretical convergence guarantees
- outperforming existing models on different empirical examples

Thank you for watching!