Variational Bayesian inference for agent-based models

Joel Dyer
University of Oxford

INET Seminar
23rd Nov 2023

University of Oxford
Institute for New Economic Thinking at the Oxford Martin School

University of Oxford
Department of Computer Science

Gradient-Assisted Calibration for Financial Agent-Based Models

Joel Dyer*
University of Oxford
Oxford, United Kingdom
joel.dyer@cs.ox.ac.uk

Arnau Quera-Bofarull*
University of Oxford
Oxford, United Kingdom
arnau.quera-bofarull@cs.ox.ac.uk

Ayush Chopra
Massachusetts Institute of Technology
Boston, USA
ayushc@mit.edu

J. Doyne Farmer
University of Oxford
Oxford, United Kingdom

Anisoara Calinescu
University of Oxford
Oxford, United Kingdom

Michael Wooldridge
University of Oxford
Oxford, United Kingdom

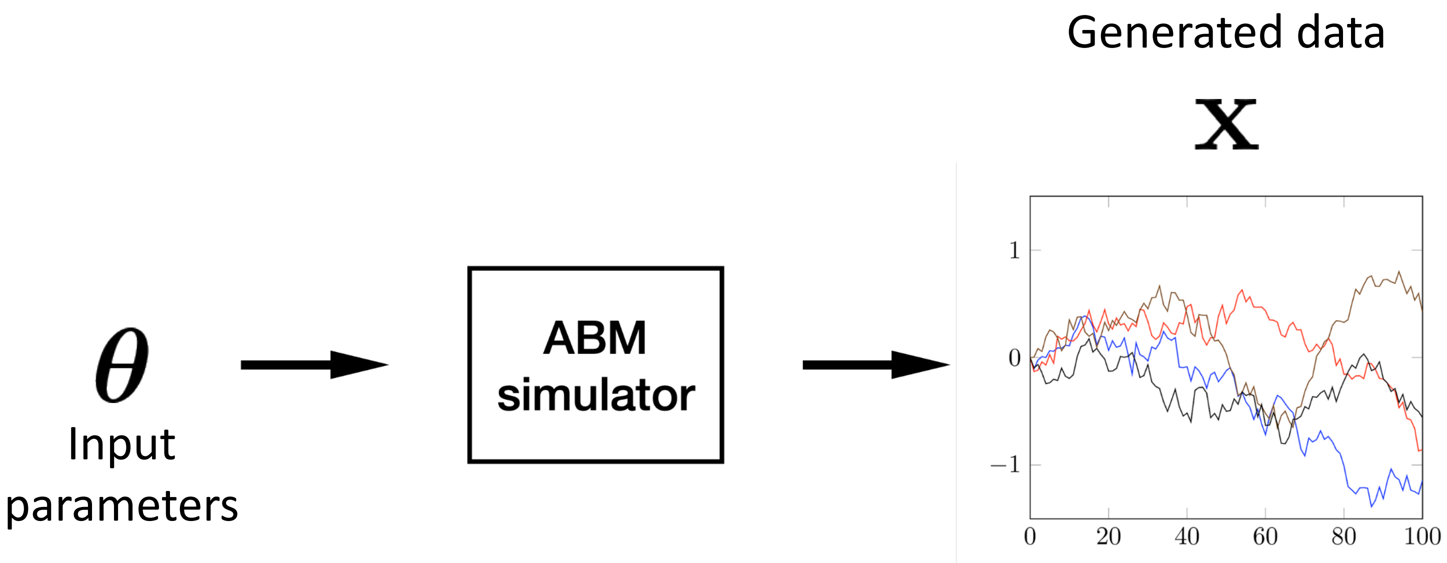
Forthcoming in:
Proceedings of the Fourth International Conference on AI in Finance (ICAIF 2023)

GitHub repo: joelnmdyer/gradient_assisted_calibration_abm

University of Oxford
Institute for New Economic Thinking
AT THE OXFORD MARTIN SCHOOL

University of Oxford
DEPARTMENT OF COMPUTER SCIENCE

Introduction



ABM simulator

- (**Stochastic**) simulation model of many autonomous, interacting agents making (often **discrete**) decisions
- Simulation denoted mathematically as sampling from likelihood function:

$$\mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta})$$

Using agent-based models

- Usually want to calibrate ABMs when applying them in practice, e.g. using Bayesian inference:

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto e^{-\ell(\boldsymbol{\theta},\mathbf{y})}\pi(\boldsymbol{\theta})$$

Observed data — Plausible parameters

$$\mathbf{y} \longrightarrow \boldsymbol{\theta}$$

- Calibration (and other problems) made complicated by complexity of ABMs – likelihood function unavailable, expensive to simulate, discrete randomness prevents immediate construction of useful gradients etc.

This talk

- Performing optimisation-centric calibration procedures with and without model gradients
- How to get around difficulty of differentiating through discrete randomness

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto e^{-\ell(\boldsymbol{\theta}, \mathbf{y})} \pi(\boldsymbol{\theta})$$

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Taking $\mathcal{Q}$ to be the set of all distributions on $\boldsymbol{\Theta}$:

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{\theta} \sim q} \left[\log \frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

References

An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference
J Knoblauch, J Jewson, T Damoulas - The Journal of Machine Learning Research, 2022

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Taking $\mathcal{Q}$ to be the set of all distributions on $\boldsymbol{\Theta}$:

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{\theta} \sim q} \left[\log \frac{q(\boldsymbol{\theta})\, Z(\mathbf{y})}{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})} \pi(\boldsymbol{\theta})} \right]$$

References

An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference
J Knoblauch, J Jewson, T Damoulas - The Journal of Machine Learning Research, 2022

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Taking $\mathcal{Q}$ to be the set of all distributions on $\boldsymbol{\Theta}$:

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{\theta} \sim q} \left[-\log \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} + \log \frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

References

An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference
J Knoblauch, J Jewson, T Damoulas - The Journal of Machine Learning Research, 2022

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Taking $\mathcal{Q}$ to be the set of all distributions on $\boldsymbol{\Theta}$:

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{\theta} \sim q} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

References

An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference
J Knoblauch, J Jewson, T Damoulas - The Journal of Machine Learning Research, 2022

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Taking $\mathcal{Q}$ to be the set of all distributions on $\boldsymbol{\Theta}$:

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_{\boldsymbol{\theta} \sim q} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

$$q^*(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta} \mid \mathbf{y})$$

References

An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference
J Knoblauch, J Jewson, T Damoulas - The Journal of Machine Learning Research, 2022

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Taking $\mathcal{Q}_\Phi$ to be a restricted set $\{q_\phi \mid \phi \in \Phi\}$ of distributions on $\boldsymbol{\Theta}$:

$$\phi^* = \arg\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

$$q_{\phi^*}(\boldsymbol{\theta}) \approx \pi(\boldsymbol{\theta} \mid \mathbf{y})$$

References

An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference
J Knoblauch, J Jewson, T Damoulas - The Journal of Machine Learning Research, 2022

Bayesian inference as optimisation

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) = \frac{e^{-\ell(\boldsymbol{\theta}, \mathbf{y})}}{Z(\mathbf{y})} \pi(\boldsymbol{\theta})$$

Taking $\mathcal{Q}_\Phi$ to be a restricted set $\{q_\phi \mid \phi \in \Phi\}$ of distributions on $\boldsymbol{\Theta}$:

$$\phi^* = \arg\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

$$q_{\phi^*}(\boldsymbol{\theta}) \approx \pi(\boldsymbol{\theta} \mid \mathbf{y})$$

Generalised variational Bayesian inference

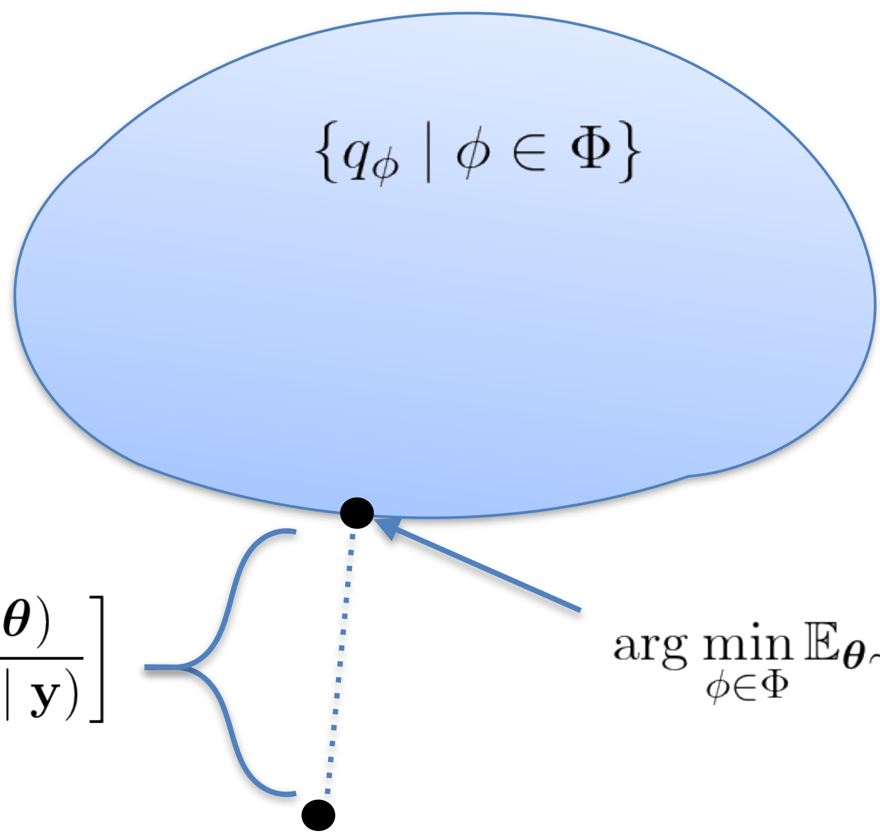
Institute for New Economic Thinking at the Oxford Martin School

References

An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference
J Knoblauch, J Jewson, T Damoulas - The Journal of Machine Learning Research, 2022

Department of Computer Science

Quick schematic of variational Bayes



$$\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

$$\arg\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

$$\pi(\boldsymbol{\theta} \mid \mathbf{y})$$

Optimising

Optimisation problem:

$$\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

Optimising

Optimisation problem:

$$\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

Minimise with gradient-based descent methods:

$$\nabla_\phi \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

Optimising

Optimisation problem:

$$\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

Minimise with gradient-based descent methods:

$$\nabla_\phi \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

Optimising

Optimisation problem:

$$\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

Minimise with gradient-based descent methods:

$$? \quad \nabla_\phi \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right] \quad ?$$

Optimising

Optimisation problem:

$$\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right]$$

Minimise with gradient-based descent methods:

$$\text{Find } G \text{ such that } \mathbb{E}[G] = \nabla_\phi \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

Monte Carlo gradient estimation

$$G \quad \text{such that} \quad \mathbb{E}\left[G\right] = \nabla_\omega \mathbb{E}_{z \sim p_\omega}\left[\mathcal{L}(z)\right]$$

References

Monte carlo gradient estimation in machine learning
S Mohamed, M Rosca, M Figurnov, A Mnih - The Journal of Machine Learning Research, 2020

Monte Carlo gradient estimation

$$G \quad \text{such that} \quad \mathbb{E}\left[G\right] = \nabla_\omega \mathbb{E}_{z \sim p_\omega}\left[\mathcal{L}(z)\right]$$

Score-based estimator

$$G = \mathcal{L}(z) \nabla_\omega \log p_\omega(z)$$

$$\mathbb{E}_{z \sim p_\omega}\left[\mathcal{L}(z) \nabla_\omega \log p_\omega(z)\right] = \nabla_\omega \mathbb{E}_{z \sim p_\omega}\left[\mathcal{L}(z)\right]$$

References

Monte carlo gradient estimation in machine learning
S Mohamed, M Rosca, M Figurnov, A Mnih - The Journal of Machine Learning Research, 2020

Monte Carlo gradient estimation

$$G \quad \text{such that} \quad \mathbb{E}\left[G\right] = \nabla_\omega \mathbb{E}_{z\sim p_\omega}\left[\mathcal{L}(z)\right]$$

Score-based estimator

$$G = \mathcal{L}(z)\nabla_\omega \log p_\omega(z)$$

$$\mathbb{E}_{z\sim p_\omega}\left[\mathcal{L}(z)\nabla_\omega \log p_\omega(z)\right] = \nabla_\omega \mathbb{E}_{z\sim p_\omega}\left[\mathcal{L}(z)\right]$$

- Generically applicable (e.g. in presence of discrete randomness)
- Can be high-variance – less reliable gradients

References

Monte carlo gradient estimation in machine learning
S Mohamed, M Rosca, M Figurnov, A Mnih - The Journal of Machine Learning Research, 2020

Monte Carlo gradient estimation

$$G \quad \text{such that} \quad \mathbb{E}\left[G\right] = \nabla_\omega \mathbb{E}_{z\sim p_\omega}\left[\mathcal{L}(z)\right]$$

Pathwise estimator

$$G = \nabla_\omega \mathcal{L}(g(u,\omega))\big|_{g(u,\omega)=z}$$

$$\mathbb{E}_{u\sim p}\left[\nabla_\omega \mathcal{L}(g(u,\omega))\big|_{g(u,\omega)=z}\right] = \nabla_\omega \mathbb{E}_{z\sim p_\omega}\left[\mathcal{L}(z)\right]$$

References

Monte carlo gradient estimation in machine learning
S Mohamed, M Rosca, M Figurnov, A Mnih - The Journal of Machine Learning Research, 2020

Monte Carlo gradient estimation

$$G \quad \text{such that} \quad \mathbb{E}\left[G\right] = \nabla_\omega \mathbb{E}_{z\sim p_\omega}\left[\mathcal{L}(z)\right]$$

Pathwise estimator

$$G = \nabla_\omega \mathcal{L}(g(u,\omega))\Big|_{g(u,\omega)=z}$$

$$\mathbb{E}_{u\sim p}\left[\nabla_\omega \mathcal{L}(g(u,\omega))\Big|_{g(u,\omega)=z}\right] = \nabla_\omega \mathbb{E}_{z\sim p_\omega}\left[\mathcal{L}(z)\right]$$

- Requires differentiable $\mathcal{L}$ & "reparameterisable" z
- Is often (though not always) *lower-variance* – often gives more informative gradients!

References

[Monte carlo gradient estimation in machine learning]
S Mohamed, M Rosca, M Figurnov, A Mnih - The Journal of Machine Learning Research, 2020

A possible benefit of differentiability

Minimise with gradient-based descent methods:

$$\text{Find } G \text{ such that } \mathbb{E}[G] = \nabla_\phi \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

A possible benefit of differentiability

Minimise with gradient-based descent methods:

$$\text{Find } G \text{ such that } \mathbb{E}[G] = \nabla_\phi \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

If our choice of $\ell(\boldsymbol{\theta}, \mathbf{y})$ is (a) differentiable and (b) estimated using samples from the agent-based model, then we can *potentially* obtain lower-variance estimates of the gradient during optimisation by implementing the agent-based model in a differentiable programming language and using the *pathwise* gradient estimator

Accessing pathwise gradients

Reparameterisable sampling

Accessing pathwise gradients

Reparameterisable sampling

$$z \sim \mathcal{N}(\mu, \sigma)$$

Accessing pathwise gradients

Reparameterisable sampling

$$u \sim \mathcal{N}(0, 1)$$

$$z = g(u, \omega),$$

$$\text{where} \quad \omega = (\mu, \sigma) \quad \text{and}$$

$$z \sim \mathcal{N}(\mu, \sigma)$$

or

$$g(u, \omega) = \mu + \sigma u$$

Accessing pathwise gradients

Reparameterisable sampling

$$u \sim \mathcal{N}(0, 1)$$

$$z = g(u, \omega),$$

$$\text{where} \quad \omega = (\mu, \sigma) \quad \text{and}$$

$$z \sim \mathcal{N}(\mu, \sigma)$$

or

$$\boxed{g(u, \omega) = \mu + \sigma u}$$

Reparameterisation

Accessing pathwise gradients

Reparameterisable sampling

$$u \sim \mathcal{N}(0, 1)$$

$$z = g(u, \omega),$$

$$\text{where} \quad \omega = (\mu, \sigma) \quad \text{and}$$

$$z \sim \mathcal{N}(\mu, \sigma)$$

or

$$\boxed{g(u, \omega) = \mu + \sigma u}$$

Reparameterisation

$$\frac{\partial z}{\partial \mu} = 1; \qquad \frac{\partial z}{\partial \sigma} = u$$

Accessing pathwise gradients

Reparameterisable sampling for discrete random variables

- Harder, but possible:
  - Approximate model gradient using smoothed versions of discrete random variables (e.g. Gumbel-Softmax [1])
    - Imperfect solution/workaround, since the model gradient is still not “correct”. But can work sufficiently well in some settings!
  - StochasticAD [2]

[1] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. arXiv:1611.01144 [cs, stat]

[2] Gaurav Arya, Moritz Schauer, Frank Schäfer, and Chris Rackauckas. 2022. Automatic Differentiation of Programs with Discrete Randomness. arXiv:2210.08572 [cs, math]

Accessing pathwise gradients

Overall strategy

1. Implement ABM in differentiable framework (e.g. PyTorch, Jax)
2. Use aforementioned tricks to obtain an (approximate) model gradient (without changing the forward pass of model!)

Experiments

Experiments

Agent-based model

Implement and calibrate differentiable version of simple ABM of volatility clustering in financial markets (Cont, 2007)

- Discrete choices by agents
- Threshold effects

References

Rama Cont. 2007. Volatility clustering in financial markets: empirical facts and agent-based models. Long memory in economics (2007), 289–309

Experiments

Agent-based model

Implement and calibrate differentiable version of simple ABM of volatility clustering in financial markets (Cont, 2007)

- Discrete choices by agents
- Threshold effects

Simulation loop:

1. Each agent receives a common information signal
2. Each agent processes signal and decides whether to place purchase order
3. Excess demand determines change in price
4. Agents consequently update their signal processing procedure

References

Rama Cont. 2007. Volatility clustering in financial markets: empirical facts and agent-based models. Long memory in economics (2007), 289–309

Experiments

Inference problem

Perform *generalised* Bayesian inference targeting

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto e^{-\ell(\boldsymbol{\theta}, \mathbf{y})} \pi(\boldsymbol{\theta})$$

with $\ell(\boldsymbol{\theta}, \mathbf{y})$ a divergence between distribution of simulated and "real" log-returns, by solving minimisation problem shown before

References

Rama Cont. 2007. Volatility clustering in financial markets: empirical facts and agent-based models. Long memory in economics (2007), 289–309

Experiments

Inference problem

Perform *generalised* Bayesian inference targeting

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}) \propto e^{-\ell(\boldsymbol{\theta},\mathbf{y})}\pi(\boldsymbol{\theta})$$

with $\ell(\boldsymbol{\theta}, \mathbf{y})$ a divergence between distribution of simulated and "real" log-returns, by solving minimisation problem shown before

Approximate (intractable) $\pi(\boldsymbol{\theta} \mid \mathbf{y})$ using variational inference

$$\min_{\phi \in \Phi} \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \left[\ell(\boldsymbol{\theta}, \mathbf{y}) + \log \frac{q_\phi(\boldsymbol{\theta})}{\pi(\boldsymbol{\theta})} \right]$$

taking q_ϕ to be a "normalising flow" (i.e. neural density estimator)

References

Rama Cont. 2007. Volatility clustering in financial markets: empirical facts and agent-based models. Long memory in economics (2007), 289–309

Experiments

Calibrate using our BlackBIRDS* software package

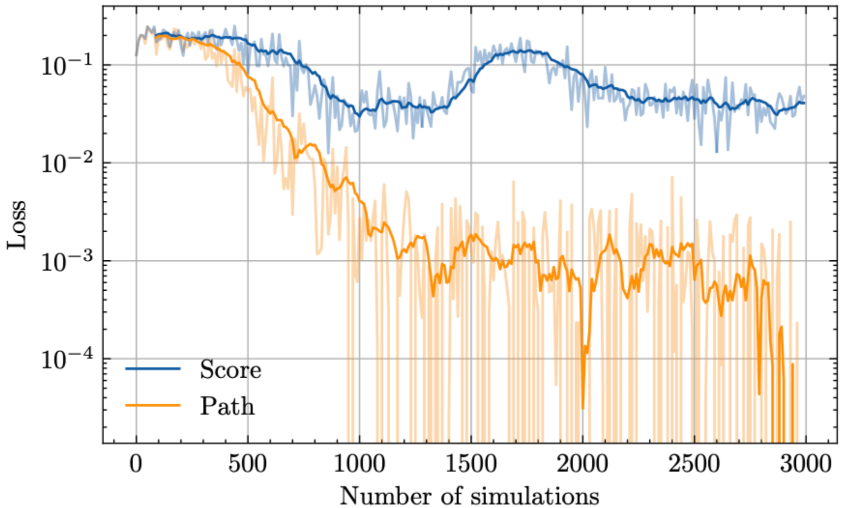


Figure 1: Training loss for the generalised variational inference scheme with score-based (blue) and pathwise (orange) gradient estimators. Dark lines show the moving average loss by averaging over 10 epochs.

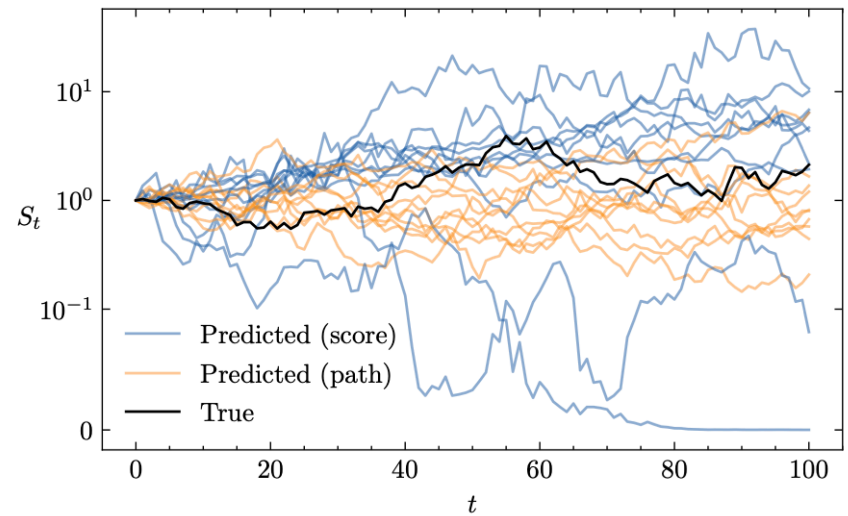


Figure 3: Sample trajectories for the asset price from the posterior predictive distributions obtained from the score-based (blue) and pathwise (orange) gradient estimators. True asset price is shown with the black line.

References

*BlackBIRDS: Black-Box Inference foR Differentiable Simulators, A Quera-Bofarull, J Dyer, et al., *Journal of Open Source Software*, 2023

Code?

joelnmdyer / **gradient_assisted_calibration_abm**

BlackBIRDS

Contributors 2

arnauqb Arnau Quera-Bofarull

joelnmdyer Joel Dyer

Recently published in:
The Journal of Open Source Software

Summary/discussion

- Bayesian inference (and other calibration approaches) for ABMs can be written as an optimisation problem – minimising expectations of random losses
- Differentiability of ABMs can sometimes help to perform this optimisation
- Limitations:
  - Usefulness of gradients depends on bias-variance tradeoff of estimators they are used for
  - Simulator gradients less useful when (derivative of) loss function is intractable (consider e.g. maximum likelihood estimation)

Summary/discussion

- Bayesian inference (and other calibration approaches) for ABMs can be written as an optimisation problem – minimising expectations of random losses
- Differentiability of ABMs can sometimes help to perform this optimisation
- Limitations:
  - Usefulness of gradients depends on bias-variance tradeoff of estimators they are used for
  - Simulator gradients less useful when (derivative of) loss function is intractable (consider e.g. maximum likelihood estimation)

Thank you!

Email: joel.dyer@cs.ox.ac.uk

Website: **joelnmdyer.github.io**

Twitter/X: @joelnmdyer

University of Oxford Institute for New Economic Thinking at the Oxford Martin School

University of Oxford Department of Computer Science