Gradient-assisted calibration for financial agent-based models

Joel Dyer
University of Oxford

International Conference on AI in Finance
27th 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

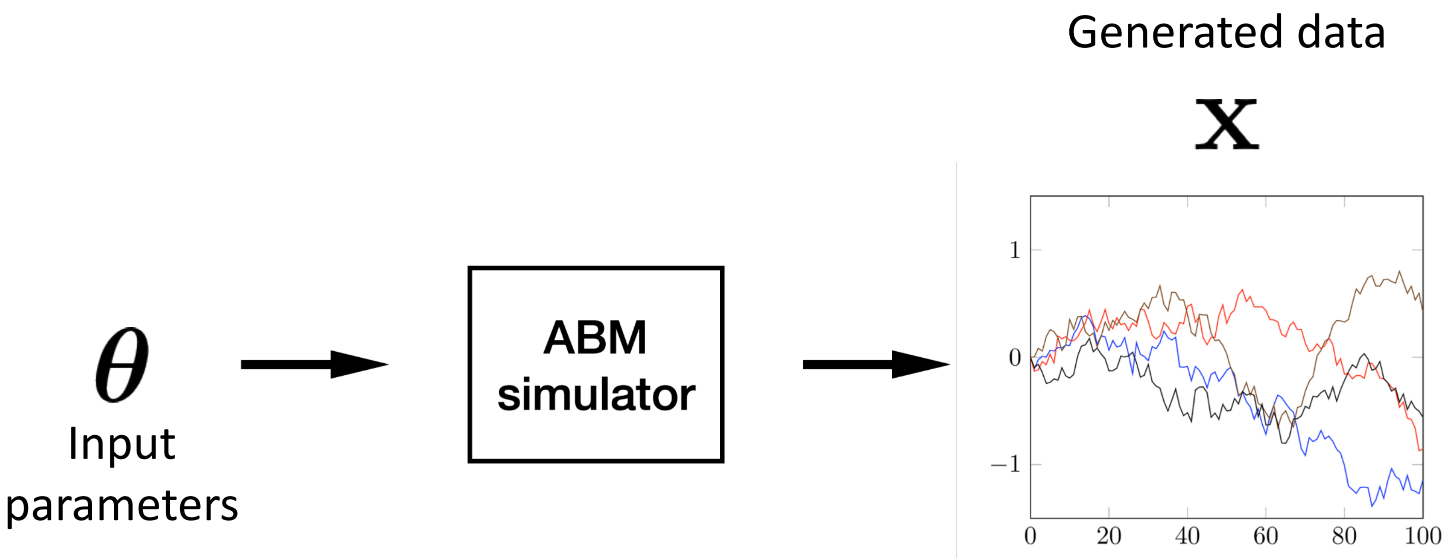
J. Doyne Farmer
University of Oxford
Oxford, United Kingdom

Anisoara Calinescu
University of Oxford
Oxford, United Kingdom

Michael Wooldridge
University of Oxford
Oxford, United Kingdom

GitHub repo: joelnmdyer/gradient_assisted_calibration_abm

Agent-based models



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.

Calibration and optimisation

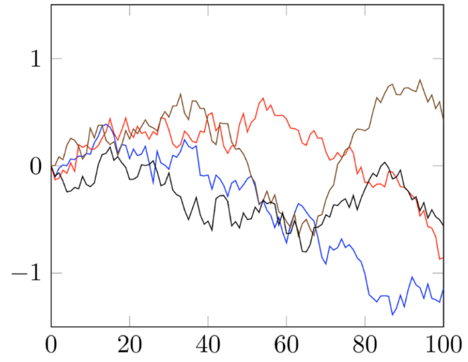
Generated data

$\mathbf{x}$

$\boldsymbol{\theta}$

Input parameters

$\rightarrow$ **ABM simulator** $\rightarrow$



$$\min_{\omega \in \Omega} \mathbb{E}_{z \sim p_\omega} \left[\mathcal{L}(z) \right]$$

Calibration and optimisation

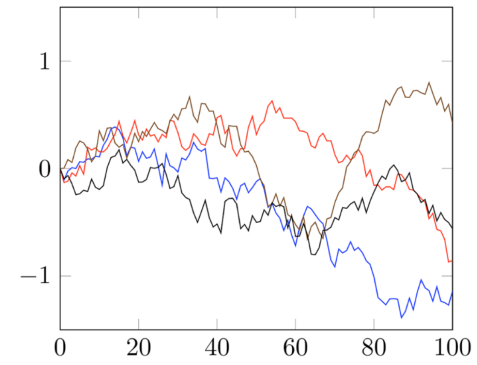
Generated data

$\mathbf{x}$

$\boldsymbol{\theta}$

Input parameters

→ **ABM simulator** →



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

Calibration and optimisation

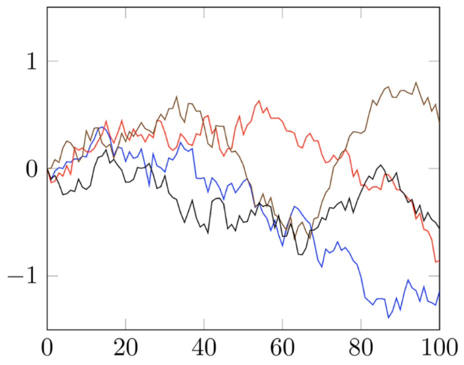
$\boldsymbol{\theta}$

Input parameters

→ **ABM simulator** →

Generated data

$\mathbf{x}$



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

Calibration and optimisation

$\boldsymbol{\theta}$

Input parameters

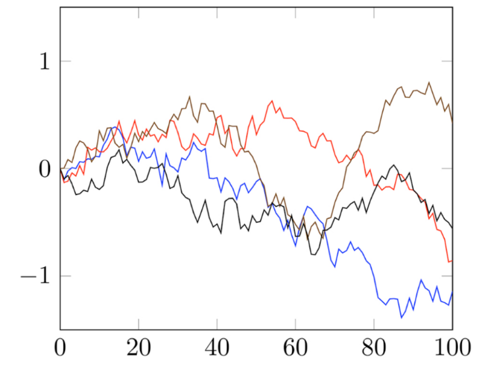
→

ABM simulator

→

Generated data

$\mathbf{x}$



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

Calibration and optimisation

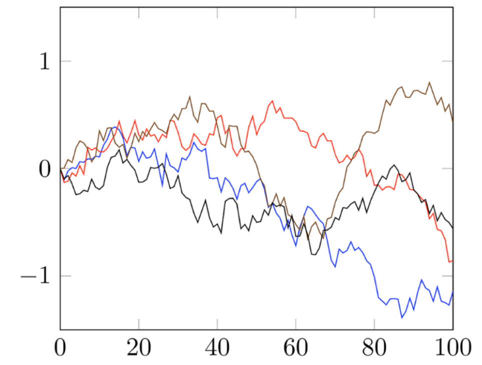
$\boldsymbol{\theta}$

Input parameters

→ **ABM simulator** →

Generated data

$\mathbf{x}$



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

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

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

Two common approaches

- **Score-based** gradient estimator
- **Pathwise** gradient estimator

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

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

Two common approaches

- **Score-based** gradient estimator
  - Generically applicable (e.g. in presence of discrete randomness)
  - Can be high-variance – less reliable gradients
- **Pathwise** gradient estimator

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

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

Two common approaches

- **Score-based** gradient estimator
  - Generically applicable (e.g. in presence of discrete randomness)
  - Can be high-variance – less reliable gradients
- **Pathwise** gradient estimator
  - Requires differentiable loss function $\mathcal{L}$ & "reparameterisable" z
  - Is often (though not always) lower variance – often gives more informative gradient estimates

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:

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

A possible benefit of differentiability

Minimise with gradient-based descent methods:

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

- Differentiability of $\mathcal{L}$ and the agent-based model can (sometimes) provide access to potentially lower-variance pathwise gradients
- Differentiability can be achieved using, e.g.,
  - Approximate model gradients, using smoothed versions of discrete random variables (e.g. Gumbel-Softmax [1])
  - StochasticAD [2]

References

[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 & using 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!)
3. Perform gradient-based descent on expectation of loss function using these pathwise gradients

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

Institute for New Economic Thinking AT THE OXFORD MARTIN SCHOOL
UNIVERSITY OF OXFORD

References

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

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

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

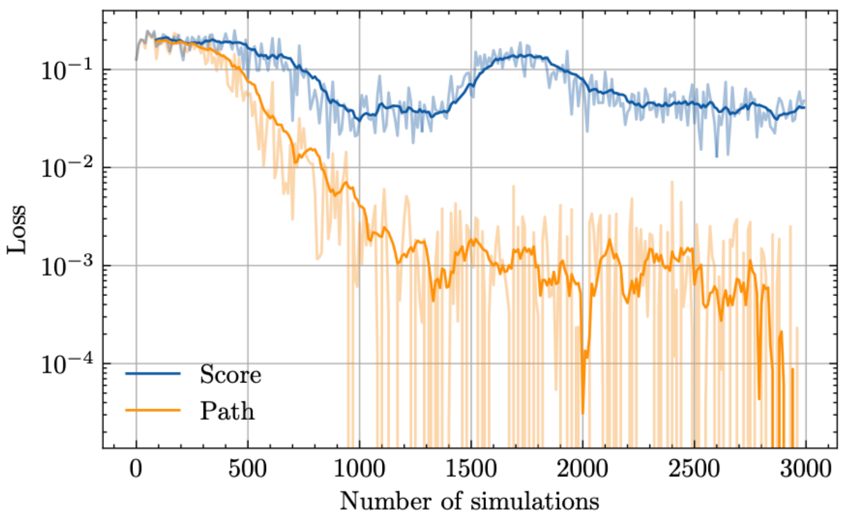


Figure 1: Training loss for the generalised variational inference scheme with score-based (blue) and pathwise (orange) gradient estimators. Dark lines show the moving average loss by averaging over 10 epochs.

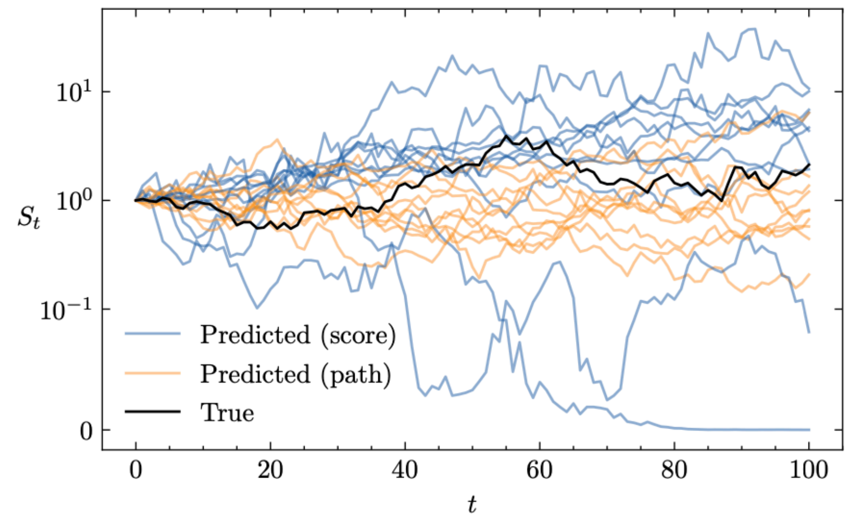


Figure 3: Sample trajectories for the asset price from the posterior predictive distributions obtained from the score-based (blue) and pathwise (orange) gradient estimators. True asset price is shown with the black line.

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

Discussion

- 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)
- Noisier gradients may sometimes be desirable

Discussion

- 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)
- Noisier gradients may sometimes be desirable

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