The Journal of Open Source Software

BlackBIRDS: Black-Box Inference foR Differentiable Simulators

Arnau Quera-Bofarull [1,2*¶], **Joel Dyer** [1,2*¶], **Anisoara Calinescu** [1], **J. Doyne Farmer** [2,3,4], **and Michael Wooldridge** [1]

1 Department of Computer Science, University of Oxford, UK **2** Institute for New Economic Thinking, University of Oxford, UK **3** Mathematical Institute, University of Oxford, UK **4** Santa Fe Institute, USA ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.05776

Software

- Review
- Repository
- Archive

Editor: Rachel Kurchin

Reviewers:

- @rajeshrinet
- @marvinschmitt

Submitted: 17 July 2023
Published: 25 September 2023

Summary

`BlackBIRDS` is a Python package consisting of generically applicable, black-box inference methods for differentiable simulation models. It facilitates both (a) the differentiable implementation of simulation models by providing a common object-oriented framework for their implementation in `PyTorch` (Paszke et al., 2019), and (b) the use of a variety of gradient-assisted inference procedures for these simulation models, allowing researchers to easily exploit the differentiable nature of their simulator in parameter estimation tasks. The package consists of both Bayesian and non-Bayesian inference methods, and relies on well-supported software libraries (e.g., `normflows`, Stimper et al., 2023) to provide this broad functionality.

Statement of need

Across scientific disciplines and application domains, simulation is used extensively as a means to studying complex mathematical models of real-world systems. A simulation-based approach to modelling such systems provides the modeller with significant benefits, permitting them to specify their model in the way that they believe most faithfully represents the true data-generating process and relieving them from concerns regarding the mathematical tractability of the resulting model. However, this additional flexibility comes at a price: the resulting model can be too complex to easily perform optimisation and inference tasks on the corresponding simulator, which in many cases necessitates the use of approximate, simulation-based inference and optimisation methods to perform these tasks inexactly.

The complicated and black-box nature of many simulation models can present a significant barrier to the successful deployment of these simulation-based inference and optimisation techniques. Consequently, there has been increasing interest within various scientific communities in constructing *differentiable* simulation models (see e.g., Baydin et al., 2020; Chopra et al., 2023): simulation models for which the gradient of the model output with respect to the model's input parameters can be easily obtained. The primary motivation for this is that access to this additional information, which captures the sensitivity of the output of the simulator to changes in the input, can enable the use of more efficient simulation-based optimisation and inference procedures, helping to reduce the total runtime of such algorithms, their overall consumption of valuable computational resources, and their concomitant financial and environmental costs.

To this end, `BlackBIRDS` was designed to provide researchers with easy access to a set of parameter inference methods that exploit the gradient information provided by differentiable simulators. The package provides support for a variety of approaches to gradient-assisted parameter inference, including:

The Journal of Open Source Software

- Simulated Minimum Distance (Franke, 2009; SMD, see e.g., Gourieroux et al., 1993), in which parameter point estimates $\hat{\theta}$ are obtained as

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \ell(\theta, \mathbf{y}), \tag{1}$$

  where θ are the simulator's parameters, which take values in some set Θ, and ℓ is a loss function capturing the compatibility between the observed data $\mathbf{y}$ and the simulator's behaviour at parameter vector θ;

- Markov chain Monte Carlo (MCMC), in which samples from a parameter posterior

$$\pi(\theta \mid \mathbf{y}) \propto e^{-\ell(\theta, \mathbf{y})} \pi(\theta), \tag{2}$$

  corresponding to a choice of loss function ℓ and a prior density π over Θ are generated by executing a Markov chain on Θ. Currently, support is provided for Metropolis-adjusted Langevin Dynamics (Roberts & Tweedie, 1996), although nothing prevents the inclusion of additional gradient-assisted MCMC algorithms such as Hamiltonian Monte Carlo (Duane et al., 1987);

- Variational Inference (VI), in which a parametric approximation q^* to the intractable posterior is obtained by solving the following optimisation problem over a variational family $\mathcal{Q}$

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathbb{E}_q \left[-\ell(\theta, \mathbf{y})\right] + \mathbb{E}_q \left[\log \frac{q(\theta)}{\pi(\theta)}\right], \tag{3}$$

  where ℓ is defined as above and π is a prior density over Θ.

The package is written such that the user is free to specify their choice of ℓ and π (in the case of Bayesian methods), under the constraint that both choices are differentiable with respect to θ. This allows the user to target a wide variety of parameter point estimators, and both classical and generalised (see e.g., Bissiri et al., 2016; Knoblauch et al., 2022) posteriors. We provide a number of tutorials demonstrating (a) how to implement a simulator in a differentiable framework in PyTorch and (b) how to apply the different parameter inference methods supported by `BlackBIRDS` to these differentiable simulators. Our package provides the user with flexible posterior density estimators with the use of normalising flows, and has already been used in scientific research to calibrate differentiable simulators (Quera-Bofarull, Chopra, et al., 2023; Quera-Bofarull, Dyer, et al., 2023).

Related software

`BlackBIRDS` offers complementary functionality to a number of existing Python packages. `sbi` (Tejero-Cantero et al., 2020) is a package offering PyTorch-based implementations of numerous simulation-based inference algorithms, including those based on the use of MCMC and neural conditional density estimators. Our package differs significantly, however: in contrast to `sbi`, `BlackBIRDS` provides support for both Bayesian and non-Bayesian inference methods, and permits the researcher to exploit gradients of the simulator, loss function, and/or posterior density with respect to parameters $\boldsymbol{\theta}$ during inference tasks. The same comparison applies to the the `BayesFlow` package (Radev et al., 2023). `black-it` (Benedetti et al., 2022) is a further recent Python package that collects some recently developed parameter estimation methods from the agent-based modelling community; the focus of this package is, however, on non-Bayesian methods, and the package does not currently support the exploitation of simulator gradients. `PyVBMC` (Huggins et al., 2023) provides a Python implementation of the Variational Bayesian Monte Carlo algorithm using Gaussian processes, but differs from our package in that it does not exploit simulator gradients and is focused on Bayesian inference alone. Additional older packages (e.g., Dutta et al., 2021; Schälte et al., 2022) also focus on approximate Bayesian inference methods for non-differentiable simulators. Beyond this, we are unaware of other mature software projects in Python that support parameter inference in the specific case of differentiable simulation models.

Features

- User-friendly and flexible API: SMD only requires the loss function ℓ and the optimiser to use, while MCMC (resp. VI) requires only the loss ℓ, the prior density π, and the MCMC method (resp. posterior approximator) to be specified. However, additional arguments can be provided to straightforwardly customise hyperparameters of the different methods.
- Multi-GPU parallelisation support with MPI.
- Support for both forward-mode and reverse-mode auto-differentiation.
- Continuous integration and unit tests.

Acknowledgements

This research was supported by a UKRI AI World Leading Researcher Fellowship awarded to Wooldridge (grant EP/W002949/1). M. Wooldridge and A. Calinescu acknowledge funding from Trustworthy AI - Integrating Learning, Optimisation and Reasoning (TAILOR), a project funded by European Union Horizon2020 research and innovation program under Grant Agreement 952215.

References

Baydin, A. G., NYU, K. C., Feickert, M., Gray, L., Heinrich, L., NYU, A. H., Neubauer, A. M. V. M., Pearkes, J., Simpson, N., Smith, N., & others. (2020). *Differentiable programming in high-energy physics*.

Benedetti, M., Catapano, G., Sclavis, F. D., Favorito, M., Glielmo, A., Magnanimi, D., & Muci, A. (2022). Black-it: A ready-to-use and easy-to-extend calibration kit for agent-based models. *Journal of Open Source Software*, *7*(79), 4622. https://doi.org/10.21105/joss.04622

Bissiri, P. G., Holmes, C., & Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *78*(5), 1103. https://doi.org/10.1111/rssb.12158

Chopra, A., Rodríguez, A., Subramanian, J., Quera-Bofarull, A., Krishnamurthy, B., Prakash, B. A., & Raskar, R. (2023). Differentiable agent-based epidemiology. *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 1848–1857. ISBN: 978-1-4503-9432-1

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, *195*(2), 216–222.

Dutta, R., Schoengens, M., Pacchiardi, L., Ummadisingu, A., Widmer, N., Künzli, P., Onnela, J.-P., & Mira, A. (2021). ABCpy: A high-performance computing perspective to approximate Bayesian computation. *Journal of Statistical Software*, *100*(7), 1–38. https://doi.org/10.18637/jss.v100.i07

Franke, R. (2009). Applying the method of simulated moments to estimate a small agent-based asset pricing model. *Journal of Empirical Finance*, *16*(5), 804–815. https://doi.org/10.1016/j.jempfin.2009.06.006

Gourieroux, C., Monfort, A., & Renault, E. (1993). Indirect inference. *Journal of Applied Econometrics*, *8*(S1), S85–S118.

Huggins, B., Li, C., Tobaben, M., Aarnos, M. J., & Acerbi, L. (2023). PyVBMC: Efficient Bayesian inference in python. *Journal of Open Source Software*, *8*(86), 5428. https://doi.org/10.21105/joss.05428

Knoblauch, J., Jewson, J., & Damoulas, T. (2022). An optimization-centric view on Bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, *23*(132), 1–109.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd international conference on neural information processing systems*. Curran Associates Inc.

Quera-Bofarull, A., Chopra, A., Calinescu, A., Wooldridge, M., & Dyer, J. (2023). Bayesian calibration of differentiable agent-based models. *ICLR Workshop on AI for Agent-Based Modelling*.

Quera-Bofarull, A., Dyer, J., Calinescu, A., & Wooldridge, M. (2023). Some challenges of calibrating differentiable agent-based models. *ICML Differentiable Almost Everything Workshop*.

Radev, S. T., Schmitt, M., Schumacher, L., Elsemüller, L., Pratz, V., Schälte, Y., Köthe, U., & Bürkner, P.-C. (2023). BayesFlow: Amortized Bayesian Workflows With Neural Networks. *Journal of Open Source Software*, *8*(89), 5702. https://doi.org/10.21105/joss.05702

Roberts, G. O., & Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 341–363. https://doi.org/10.2307/3318418

Schälte, Y., Klinger, E., Alamoudi, E., & Hasenauer, J. (2022). pyABC: Efficient and robust easy-to-use approximate Bayesian computation. *Journal of Open Source Software*, *7*(74), 4304. https://doi.org/10.21105/joss.04304

Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., & Hernández-Lobato, J. M. (2023). Normflows: A PyTorch package for normalizing flows. *Journal of Open Source Software*, *8*(86), 5361. https://doi.org/10.21105/joss.05361

Tejero-Cantero, A., Boelts, J., Deistler, M., Lueckmann, J.-M., Durkan, C., Gonçalves, P. J., Greenberg, D. S., & Macke, J. H. (2020). Sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, *5*(52), 2505. https://doi.org/10.21105/joss.02505