The Journal of Open Source Software

Bamojax: Bayesian Modelling with JAX

Max Hinne [1*]

1 Radboud University, Nijmegen, The Netherlands * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Kanishka B. Narayan

Reviewers:

- @matt-graham

Submitted: 23 April 2025
Published: unpublished

Summary

Bayesian statistics offers a principled and elegant framework for inferring hidden causes from observed effects. It also provides a rigorous approach to hypothesis testing (model comparison), with advantages such as built-in complexity penalties, and the ability to quantify evidence in favour of the null hypothesis.

However, exact Bayesian inference is computationally intractable in all but the simplest of cases, and requires *approximate inference* techniques, such as Markov chain Monte Carlo and variational inference. Recent advances in the Python JAX (Bradbury et al., 2018) framework have enabled highly efficient implementations of these algorithms, due to features such as automated differentation and GPU acceleration. These developments have the potential to greatly increase the efficiency of statistical modelling pipelines.

`bamojax` ('Bayesian Modelling in Jax') is a probabilistic programming language (PPL) that combines ease-of-use with access to advanced inference algorithms implemented in the Jax ecosystem.

Statement of need

Bamojax is a Bayesian modelling tool based on Python & JAX (Bradbury et al., 2018). It provides an intuitive, intermediate-level interface between defining a Bayesian statistical model conceptually, and performing efficient inference using the Blackjax package (Cabezas et al., 2024).

Existing probabilistic programming languages, such as PyMC (Abril-Pla O, 2023), can export a logdensity function that enables Blackjax-based inference. However, this has two limitations:

1. It does not support Gibbs sampling, where variables are updated individually using their own MCMC kernels. For example, when approximating the posterior over a latent Gaussian process and its hyperparameters, elliptical slice sampling for the GP is often more efficient than applying NUTS to all variables jointly. This becomes even more important when embedding MCMC sampling in Sequential Monte Carlo (Hinne, 2025).
2. It makes it harder to apply tempered Sequential Monte Carlo methods that need separate prior and likelihood densities.

While users can circumvent these issues by manually implementing their models using Blackjax, this is a labor-intensive and error-prone process. **Bamojax** addresses this gap by providing a user-friendly interface for model construction and Gibbs sampling on top of Blackjax.

In **Bamojax**, users can define a probabilistic model by specifying variables as well as their associated distributions and dependencies, structured using a directed acyclic graph (DAG). Under the hood, **Bamojax** translates this DAG and collection of probability distributions to the probability densities used in the approximate inference, leveraging the probability definitions defined in distrax (DeepMind et al., 2020). This abstraction allows users to focus on the conceptual model formulation, rather than the mathematical or inference details, leading to a more intuitive, less error-prone, and more efficient development workflow.

Bamojax is designed for researchers, students, and practitioners that want to make use of the extremely fast approximate inference offered by Blackjax, but want to focus on model development instead of implementation.

Comparison with existing tools

While existing software for probabilistic modelling, such as PyMC (Abril-Pla O, 2023), can also interface with Blackjax for inference, this only supports using a single log-density function that describes the entire probabilistic model. This precludes Gibbs sampling, where individual model parameters are updated in turn, which in practice can greatly increase the efficiency of approximate inference. In contrast, **Bamojax** allows fine-grained control over the inference strategy. This enables users to mix-and-match `Blackjax` MCMC kernels with elements of their probabilistic model, while maintaining the efficiency of JAX-based inference.

Acknowledgements

None at this time.

References

Abril-Pla O, C. C., Andreani V. (2023). PyMC: A modern and comprehensive probabilistic programming framework in Python. *PeerJ Computer Science*, *9*(e1516). https://doi.org/10.7717/peerj-cs.1516

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.4.35). http://github.com/jax-ml/jax

Cabezas, A., Corenflos, A., Lao, J., & Louf, R. (2024). *BlackJAX: Composable Bayesian inference in JAX*. https://arxiv.org/abs/2402.10797

DeepMind, Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., … Viola, F. (2020). *The DeepMind JAX Ecosystem*. http://github.com/deepmind

Hinne, M. (2025). An introduction to Sequential Monte Carlo for Bayesian inference and model comparison—with examples for psychology and behavioral science. *Behavior Research Methods*, *57*(25). https://doi.org/10.3758/s13428-025-02642-1