The Journal of Open Source Software

pocoMC: A Python package for accelerated Bayesian inference in astronomy and cosmology

Minas Karamanis [1,3¶], **David Nabergoj** [2], **Florian Beutler** [1], **John A. Peacock** [1], **and Uroš Seljak** [3]

1 Institute for Astronomy, University of Edinburgh, Royal Observatory, Blackford Hill, Edinburgh EH9 3HJ, UK **2** Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, 1000 Ljubljana, Slovenia **3** Physics Department, University of California and Lawrence Berkeley National Laboratory Berkeley, CA 94720, USA ¶ Corresponding author

DOI: 10.21105/joss.04634

Software

- Review
- Repository
- Archive

Editor: Dan Foreman-Mackey

Reviewers:

- @kazewong
- @marylou-gabrie

Submitted: 12 July 2022
Published: 09 November 2022

Summary

`pocoMC` is a Python package for accelerated Bayesian inference in astronomy and cosmology. The code is designed to sample efficiently from posterior distributions with non-trivial geometry, including strong multimodality and non-linearity. To this end, `pocoMC` relies on the Preconditioned Monte Carlo algorithm which utilises a Normalising Flow to decorrelate the parameters of the posterior. It facilitates both tasks of parameter estimation and model comparison, focusing especially on computationally expensive applications. It allows fitting arbitrary models defined as a log-likelihood function and a log-prior probability density function in Python. Compared to popular alternatives (e.g. nested sampling) `pocoMC` can speed up the sampling procedure by orders of magnitude, cutting down the computational cost substantially. Finally, parallelisation to computing clusters manifests linear scaling.

Statement of need

Over the past few decades, the volume of astronomical and cosmological data has increased substantially. At the same time, theoretical and phenomenological models in these fields have grown even more complex. As a response to that, a number of methods aiming at efficient Bayesian computation have been developed with the sole task of comparing those models to the available data (Sharma, 2017; Trotta, 2017). In the Bayesian context, scientific inference proceeds through the use of Bayes' theorem:

$$\mathcal{P}(\theta) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}} \tag{1}$$

where the posterior $\mathcal{P}(\theta) \equiv p(\theta|d,\mathcal{M})$ is the probability of the parameters θ given the data d and the model $\mathcal{M}$. The other components of this equation are: the likelihood function $\mathcal{L}(\theta) \equiv p(d|\theta,\mathcal{M})$, the prior $\pi(\theta) \equiv p(\theta|\mathcal{M})$, and the model evidence $\mathcal{Z} = p(d|\mathcal{M})$. The prior and the likelihood are usually provided as input in this equation and one seeks to estimate the posterior and the evidence. Knowledge of the posterior, in the form of samples, is paramount for the task of parameter estimation whereas the ratio of model evidences yields the Bayes factor which is the cornerstone of Bayesian model comparison.

Markov chain Monte Carlo (MCMC) has been established as the standard tool for Bayesian computation in astronomy and cosmology, either as a standalone algorithm or as part of another method (e.g., nested sampling, Skilling, 2006). However, as MCMC relies on the local exploration of the posterior, the presence of a non-linear correlation between parameters and multimodality can at best hinder its performance and at worst violate its theoretical guarantees of convergence (i.e. ergodicity). Usually, those challenges are partially addressed by reparameterising the model using a common change-of-variables parameter transformation. However, guessing the right kind of reparameterisation *a priori* is not trivial as it often requires a deep knowledge of the physical model and its symmetries. These problems are usually complicated further by the substantial computational cost of evaluating astronomical and cosmological models. `pocoMC` is designed to tackle exactly these kinds of difficulties by automatically reparameterising the model such that the parameters of the model are approximately uncorrelated and standard techniques can be applied. As a result, `pocoMC` produces both samples from the posterior distribution and an unbiased estimate of the model evidence thus facilitating both scientific tasks with excellent efficiency and robustness. Compared to popular alternatives such as nested sampling, `pocoMC` can reduce the computational cost, and thus, the total run time of the analysis by orders of magnitude, in both artificial and realistic applications (Karamanis et al., 2022). Finally, the code is well-tested and is currently used for research work in the field of gravitational wave parameter estimation (Vretinaris et al., 2022).

Method

`pocoMC` implements the Preconditioned Monte Carlo (PMC) algorithm. PMC combines the popular Sequential Monte Carlo (SMC, Del Moral et al., 2006) method with a Normalising Flow (NF, Papamakarios et al., 2021). The latter works as a preconditioner for the target distribution of the former. As SMC evolves a population of particles, starting from the prior distribution and gradually approaching the posterior distribution, the NF transforms the parameters of the target distribution such that any correlation between parameters or presence of multimodality is removed. The effect of this bijective transformation is the substantial rise in the sampling efficiency of the algorithm as the particles are allowed to sample freely from the target without being hindered by its locally-curved geometry. The method is explained in detail in the accompanying publication (Karamanis et al., 2022), and we provide only a summary here. NFs have been used extensively to accelerate various sampling algorithms, including Hamiltonian Monte Carlo (Hoffman et al., 2019), Metropolis adjusted Langevin algorithm (Gabrié et al., 2022a), adaptive Independence Metropolis-Hastings (Brofos et al., 2022), adaptive MCMC (Gabrié et al., 2022b), and nested sampling (Moss, 2020).

Sequential Monte Carlo

The basic idea of basic SMC is to sample from the posterior distribution $\mathcal{P}(\theta)$ by first defining a path of intermediate distributions starting from the prior $\pi(\theta)$. In the case of `pocoMC` the path has the form:

$$p_t(\theta) = \pi(\theta)^{1-\beta_t}\mathcal{P}(\theta)^{\beta_t} \tag{2}$$

where $0 = \beta_1 < \beta_2 < \cdots < \beta_T = 1$. Starting from the prior, each distribution with density $p_t(\theta)$ is sampled in turn using a collection of particles propagated by a number of MCMC steps. Before MCMC sampling, the particles are re-weighted using importance sampling and then re-sampled to account for the transition from $p_t(\theta)$ to $p_{t+1}(\theta)$. `pocoMC` utilises the importance weights of this step to define an estimator for the effective sample size (ESS) of the population of particles. Maintaining a fixed value of ESS during the run allows `pocoMC` to adaptively specify the β_t schedule.

Preconditioned Monte Carlo

In vanilla SMC, standard MCMC methods (e.g. Metropolis-Hastings) are used to update the positions of the particles during each iteration. This however can become highly inefficient if the distribution $p_t(\theta)$ is characterised by a non-trivial geometry. `pocoMC`, which is based on PMC, utilises a NF to learn an invertible transformation that simplifies the geometry of the distribution by mapping $p_t(\theta)$ into a zero-mean unit-variance normal distribution. Sampling then proceeds in the latent space in which correlations are substantially reduced. The positions of the particles are transformed back to the original parameter space at the end of each iteration. This way, PMC and `pocoMC` can sample from very challenging posteriors very efficiently using simple Metropolis-Hastings updates in the preconditioned/uncorrelated latent space.

Features

- User-friendly black-box API: only the log-likelihood, log-prior and some prior samples required from the user.
- The default configuration sufficient for most applications: no tuning is required but is possible for experienced users.
- Comprehensive plotting tools: posterior corner, trace, and run plots are all supported.
- Model evidence estimation using Gaussianized Bridge Sampling (Jia & Seljak, 2020).
- Support for both MAF and RealNVP normalising flows with added regularisation (Dinh et al., 2016; Papamakarios et al., 2017).
- Straightforward parallelisation using MPI or multiprocessing.
- Well-tested and documented: continuous integration, unit tests, a wide range of examples, and extensive documentation.

Acknowledgments

MK would like to thank Jamie Donald-McCann and Richard Grumitt for providing constructive comments and George Vretinaris for feedback on an early version of the code. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement 853291), and from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC02-05CH11231 at Lawrence Berkeley National Laboratory to enable research for Data-intensive Machine Learning and Analysis. FB is a University Research Fellow.

References

Brofos, J., Gabrié, M., Brubaker, M. A., & Lederman, R. R. (2022). Adaptation of the independent Metropolis-Hastings sampler with normalizing flow proposals. *International Conference on Artificial Intelligence and Statistics*, 5949–5986.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *J. R. Stat. Soc. B*, *68*(3), 411–436. https://doi.org/10.1111/j.1467-9868.2006.00553.x

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using Real NVP. *arXiv e-Prints*, arXiv:1605.08803. https://arxiv.org/abs/1605.08803

Gabrié, M., Rotskoff, G. M., & Vanden-Eijnden, E. (2022a). Adaptive Monte Carlo augmented with normalizing flows. *Proc. Natl. Acad. Sci.*, *119*(10). https://doi.org/10.1073/pnas.2109420119

Gabrié, M., Rotskoff, G. M., & Vanden-Eijnden, E. (2022b). Adaptive Monte Carlo augmented with normalizing flows. *Proc. Natl. Acad. Sci.*, *119*(10), e2109420119. https://doi.org/10.1073/pnas.2109420119

Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., & Vasudevan, S. (2019). NeuTra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport. *arXiv e-Prints*, arXiv:1903.03704. https://arxiv.org/abs/1903.03704

Jia, H., & Seljak, U. (2020). Normalizing constant estimation with gaussianized bridge sampling. *Symposium on Advances in Approximate Bayesian Inference*, 1–14.

Karamanis, M., Beutler, F., Peacock, J. A., Nabergoj, D., & Seljak, U. (2022). Accelerating astronomical and cosmological inference with preconditioned Monte Carlo. *Mon. Not. R. Astron Soc.*, *516*(2), 1644–1653. https://doi.org/10.1093/mnras/stac2272

Moss, A. (2020). Accelerated Bayesian inference using deep learning. *Mon. Not. R. Astron Soc.*, *496*(1), 328–338. https://doi.org/10.1093/mnras/staa1469

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, *22*(57), 1–64.

Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked autoregressive flow for density estimation. *Adv Neural Inf Process Syst*, *30*.

Sharma, S. (2017). Markov chain Monte Carlo methods for Bayesian data analysis in astronomy. *Annu. Rev. Astron. Astr.*, *55*(1), 213–259. https://doi.org/10.1146/annurev-astro-082214-122339

Skilling, J. (2006). Nested sampling for general Bayesian computation. *Bayesian Anal.*, *1*(4), 833–859. https://doi.org/10.1214/06-ba127

Trotta, R. (2017). Bayesian methods in cosmology. *arXiv e-Prints*, arXiv:1701.01467. https://arxiv.org/abs/1701.01467

Vretinaris, G., Vretinaris, S., Mermigkas, C., Karamanis, M., & Stergioulas, N. (2022). Robust and fast parameter estimation of gravitational waves from neutron star merger remnants. *In Prep*.