




approxposterior: Approximate Posterior Distributions in Python

David P. Fleming¹ and Jake VanderPlas¹

¹ University of Washington

DOI: [10.21105/joss.00704](https://doi.org/10.21105/joss.00704)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Submitted: 24 April 2018

Published: 14 May 2018

Licence

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

This package is a Python implementation of “Bayesian Active Learning for Posterior Estimation” by (Kandasamy, Schneider, and Poczos 2015) and “Adaptive Gaussian process approximation for Bayesian inference with expensive likelihood functions” (Wang and Li 2017). These algorithms allow the user to compute approximate posterior probability distributions using computationally expensive forward models by training a Gaussian Process (GP) surrogate for the likelihood evaluation. The algorithms leverage the inherent uncertainty in the GP’s predictions to identify high-likelihood regions in parameter space where the GP is uncertain. The algorithms then run the forward model at these points to compute their likelihood and re-trains the GP to maximize the GP’s predictive ability while minimizing the number of forward model evaluations. Check out (Kandasamy, Schneider, and Poczos 2015) and (Wang and Li 2017) for in-depth descriptions of the respective algorithms. *approxposterior* is under active development on GitHub and community participation is encouraged. The code is available here (Fleming 2018).

The following is a simple demonstration of *approxposterior* produced using an example Jupyter Notebook provided with the code on GitHub:

Left: Joint posterior probability distribution of the two model parameters from the (Wang and Li 2017) example. The black density map denotes the true distribution while the red contours denote the approximate distribution derived using *approxposterior*. The two distributions are in excellent agreement. Right: Total computational time required to compute the posterior probability distribution of the model parameters from the (Wang and Li 2017) example as a function of forward model evaluation time. The MCMC method (blue) runs the forward model for each MCMC iteration, while the orange curve was derived using the *approxposterior* BAPE implementation.

Acknowledgements

DPF was supported by NASA Headquarters under the NASA Earth and Space Science Fellowship Program - Grant 80NSSC17K0482. This work is supported in part by an NSF IGERT grant DGE-1258485.

References

Fleming, David P. 2018. “Approxposterior on Github.” 2018. <https://github.com/dflemin3/approxposterior>.

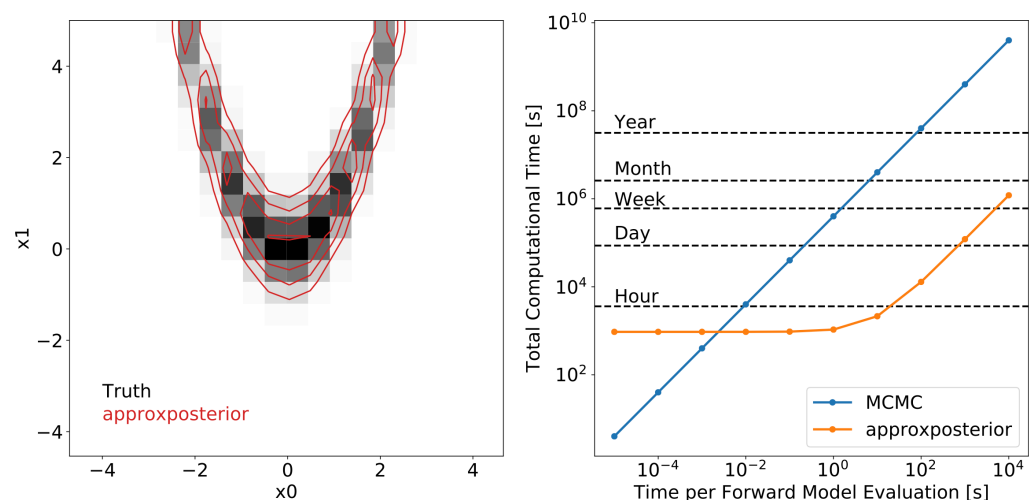


Figure 1: Left: Joint posterior probability distribution of the two model parameters from the Wang and Li (2017) example. The black density map denotes the true distribution while the red contours denote the approximate distribution derived using *approxposterior*. The two distributions are in excellent agreement. Right: Total computational time required to compute the posterior probability distribution of the model parameters from the Wang and Li (2017) example as a function of forward model evaluation time. The MCMC method (blue) runs the forward model for each MCMC iteration, while the orange curve was derived using the *approxposterior* BAPE implementation.

Kandasamy, Kirthivasan, Jeff Schneider, and Barnabas Poczos. 2015. “Bayesian Active Learning for Posterior Estimation.”

Wang, H., and J. Li. 2017. “Adaptive Gaussian process approximation for Bayesian inference with expensive likelihood functions.” *ArXiv E-Prints*, March.