uravu: Making Bayesian modelling easy(er)

Andrew R. McCluskey[1, 2] and Tim Snow[1, 3]

1 Diamond Light Source, Rutherford Appleton Laboratory, Harwell Science and Innovation Campus, Didcot, OX11 0DE, UK **2** Department of Chemistry, University of Bath, Claverton Down, Bath, BA2 7AY, UK **3** School of Chemistry, University of Bristol, Bristol, BS8 1TS, UK

Summary

uravu offers an easy to use interface for data analysis using Bayesian modelling in the Python programming language, aiming to make Bayesian modelling as easy to use as the `scipy.optimize.curve_fit()` method. This software acts to lower the barrier of entry to the use of packages such as:

- `scipy`: for maximum likelihood estimation (Virtanen et al., 2020)
- `emcee`: for Markov chain Monte Carlo investigation of posterior probabilities (Foreman-Mackey et al., 2019)
- `dynesty`: for nested sampling (Skilling, 2006) and dynamic nested sampling (Higson, Handley, Hobson, & Lasenby, 2019) of posterior probabilities and estimation of the Bayesian evidence (Speagle, 2020).

In addition to standard normal ordinate uncertainties, uravu also adds the ability to investigate non-normal ordinate value distributions, with the `Distribution` class and its integration into the `Relationship` class. Furthermore, uravu provides the functionality to create simple, publication-quality plots of relationships, data, and distributions with `matplotlib` (Hunter, 2007) and `corner` (Foreman-Mackey, 2016).

Alongside API information, the uravu documentation offers brief tutorials covering many aspects of the package. This allows those unfamiliar with Bayesian modelling to get to grips with these important tools for data analysis. While Bayesian data modelling is possible with packages such as `emcee` (Foreman-Mackey et al., 2019), `dynesty` (Speagle, 2020), and `PyMC3` (Salvatier, Wiecki, & Fonnesbeck, 2016), to the authors' knowledge uravu is unique in offering a simple to use interface, where it is not necessary to explicitly define functions for the likelihood or prior probabilities (though custom priors are possible). uravu is being actively applied to scientific problems, such as data reduction at large scale scientific facilities and the modelling of diffusion in battery materials.

Statement of Need

The Python language has a large number of powerful packages related to the application of Bayesian modelling. However, to apply these methods to their problems, scientific users need a straightforward environment. For maximum-likelihood modelling, this is achieved using the `scipy.optimize.curve_fit()` method for many users, but, to the best of the authors' knowledge, there is no equivalent method for Bayesian modelling. uravu fills this gap by offering easy access to powerful Python packages to perform Markov chain Monte Carlo and nested sampling, that is capable of handling uncertainties with any distribution. Furthermore, the tutorials, available as documentation online, allow users to become more comfortable with the use of Bayesian methods for data modelling.

Acknowledgements

References

Foreman-Mackey, D. (2016). Corner.py: Scatterplot matrices in python. *Journal of Open Source Software*, *1*(2), 24. doi:10.21105/joss.00024

Foreman-Mackey, D., Farr, W., Sinha, M., Archibald, A., Hogg, D., Sanders, J., Zuntz, J., et al. (2019). Emcee v3: A Python ensemble sampling toolkit for affine-invariant MCMC. *Journal of Open Source Software*, *4*(43), 1864. doi:10.21105/joss.01864

Higson, E., Handley, W., Hobson, M., & Lasenby, A. (2019). Dynamic nested sampling: An improved algorithm for parameter estimation and evidence calculation. *Stat. Comput.*, *29*, 891. doi:10.1007/s11222-018-9844-0

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, *9*(3), 90. doi:10.1109/MCSE.2007.55

Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, *2*, e55. doi:10.7717/peerj-cs.55

Skilling, J. (2006). Nested Sampling. *AIP Conference Proceedings*, *735*(1), 395. doi:10.1063/1.1835238

Speagle, J. S. (2020). dynesty: A Dynamic Nested Sampling Package for Estimating Bayesian Posteriors and Evidences. *Monthly Notices of the Royal Astronomical Society*. doi:10.1093/mnras/staa278

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. doi:10.1038/s41592-019-0686-2