`pymc3.dp`: Bayesian Stati-sticks for Summer 2021

A Google Summer of Code 2021 Application

by Dong, Larry

Abstract

Bayesian nonparametric (BNP) methods offer more modelling flexibility by relaxing parametric assumptions at the cost of more daunting theoretical underpinnings and challenging implementation. PyMC3 is a Python probabilistic programming library for fitting Bayesian statistical models using Aesara – formerly Theano – as a computational backend for Markov Chain Monte Carlo (MCMC) sampling and variational inference. While PyMC3 provides built-in tools for plotting, model checking and a wide selection of statistical distributions, it currently lacks support for Dirichlet Processes (DP), a BNP method that is garnening much attention from the research community in the last two decades. My Google Summer of Code (GSoC) proposal centers around bridging this gap in the PyMC3 framework by building a submodule for DPs and DP-related methods and contributing notebooks that build upon them. In this proposal, I provide a blueprint for the implementation of DP classes and I also describe my strengths, academic interests and especially how they align well with the project.

About Me

Name: Larry Dong
Current position: PhD student in biostatistics
Academic affiliation: Dalla Lana School of Public Health, University of Toronto
Phone number: +1 (514) 562-8938
Email address: larry.dong@mail.utoronto.ca
GitHub: https://github.com/larryshamalama
LinkedIn: https://www.linkedin.com/in/larry-dong/
Personal website: http://larryshamalama.github.io/

My decision to pursue a PhD in biostatistics was primarily inspired by my passion for mathematics and the time I spent as a former patient and volunteer at the Montreal Children's hospital. Broadly speaking, my PhD goals are threefold: teach other students, advance medical research and promote open science. With my first year of PhD studies almost behind me, I believe that contributing to an open source project would not only complement well my studies, but it can also be the beginning of my journey as an active member of the PyMC community. This summer, I will be studying for my comprehensive theory exam, hence taking up a part-time responsibility of open source development would complement well building up my statistical theory knowledge.

1 Relevant Experiences

Here are some projects that I have worked on previously. All of the projects listed here are publicly available as GitHub repositories.

1.1 Bayesian Programming

The link below is a repository containing my process of exploring different frameworks (NumPyro, Stan, PyMC3) for MCMC-sampling [1,2,17]. While most examples are simple, I attempted to write them from scratch to familiarize myself with different frameworks.

GitHub Link: https://github.com/larryshamalama/bayesian-programming

1.2 Image Denoising

In Green et al. (1995), the reversible jump Markov Chain Monte Carlo was introduced to allow sampling across parameter spaces of varying different dimensions [5]. I implemented the image denoising example provided in the paper in Python for our Bayesian statistics course.

GitHub Link: https://github.com/larryshamalama/Image-Denoising-RJMCMC

1.3 Spatial Analysis

During my graduate course on statistical methods for spatial and spatio-temporal data, our final project consisted of a data analysis in which we used the conditional autoregressive (CAR) prior to model spatial dependency in lattice data. While spatial methods are more popular for geographical data, we attempted to use it on a single MNIST image to see the CAR residuals can be applied for imaging data.

GitHub Link: https://github.com/larryshamalama/Spatial-image-analysis

1.4 Introduction to Neural Networks

During my Master's in Digital Public Health at the University of Bordeaux, I took the initiative to write an hour-long tutorial on using neural networks in Keras. While I was not able to go through all of the material with as much detail as I would have liked, it was a good opportunity for me to learn how to present a complicated quantitative topic to an audience whose background is primarily in public health and epidemiology. After all, disseminating data-driven methods to clinicians and epidemiologists in a simple yet comprehensive manner is an important part of my career as an aspiring biostatistician.

GitHub Link: https://github.com/larryshamalama/phds-intro-to-nn

2 Background

2.1 Bayesian Inference & PyMC3

Traditionally, statistical analyses have been conducted using frequentist inference in which parameters of interest are treated as constants and asymptotic theory is used to obtain estimated confidence intervals and p-values all within a hypothesis testing framework. Some drawbacks of frequentist theory revolve around the lack in interpretability of results and Bayesian methods mitigate this problem by inferring on parameters of interest using probabilistic statements. However, Bayesian methods revolve heavily around algorithms such as Markov Chain Monte Carlo (MCMC) sampling to perform inference which, in turn, the development of frameworks such as Stan, Jags and BUGS in the last two decades. While ongoing research aims to improve efficiently in MCMC-sampling, such as the development of Hamiltonian Monte Carlo methods and notably the No U-Turn Sampler (NUTS), or to further extend variational inference literature, a broader class of approximation methods, the integration of theoretically involved statistical methodology in popular and open source Bayesian frameworks requires ongoing efforts from a development. With PyMC3 transitioning from Theano to Aesara, the former package being no longer supported, I would love to join the growing community by contributing to PyMC3, a package that I have used since I first learned about Bayesian methods [17].

2.2 Dirichlet Processes

Notation and theory are based on Mueller et al. (2015) [13]; further mathematically rigorous and measure theoretic details are available in Phadia (2013) and Ghosal (2017) [4, 15]. Parametric models by assuming that a statistical model G belongs to a parametric family $\mathcal{G} = \{G_\theta : \theta \in \Theta\}$, where θ is often assumed to be finite-dimensional. For instance, if we assume that G to be a normally distribution, inference will then be done on $\theta = (\mu, \sigma^2) \in \Theta \equiv (\mathbb{R}, \mathbb{R}_+)$. However, in cases where we wish to perform inference on G itself or, put differently, relax its parametric assumptions, we can posit prior distributions on spaces of probability measures themselves, hence performing statistical inference on infinite-dimensional spaces.

The Dirichlet Process (DP) denoted by $\text{DP}(\alpha, P_0)$ with centering or base[1] measure P_0 falls under the umbrella of Bayesian nonparametric (BNP) methods in which we attempt to relax assumptions inherent to parametric modelling. In DPs, probability is assigned such that, for any disjoint partition $A_1, \ldots, A_k$ of Ω, its joint distribution follows a Dirichlet distribution as followed:

$$\Big(P(A_1), \ldots, P(A_k)\Big) \sim \text{Dir}\Big(\alpha P_0(A_1), \ldots, \alpha P_0(A_k)\Big)\,.$$

The sensitivity parameter α determines the relative closeness of P with respect to G_0; effectively, for large values of α, we have that DP $\to G_0$. One convenient way to

[1] The product $\alpha P_0(\cdot)$ is sometimes referred as the base measure instead.

construct G is through the stick-breaking procedure first introduced by Sethuraman (1994) [19]. We can also rewrite G as an countably infinite weighted sum for $A \subseteq \Omega$ as such:

$$G(A) = \sum_{h=1}^{\infty} w_h \delta_{m_h}(A)$$
$$w_h = v_h \prod_{\ell :\, \ell < h} (1 - v_\ell), \quad v_h \sim \text{Beta}(1, \alpha)$$

where, for all h, $m_h \sim G_0$, $\delta_{m_h}(A)$ is the Dirac delta function defined as followed:

$$\delta_{m_h}(A) = \begin{cases} 1 & \text{if } A = m_h \\ 0 & \text{otherwise} \end{cases}$$

and weights w_h can be obtained via a "stick-breaking" process demonstrated in the second equation. The analogy is as followed: a stick of length 1 is first split into two sticks of lengths v_1 and $(1 - v_1)$. The remaining $(1 - v_1)$ of the stick is then broken up into two smaller parts of lengths $(1 - v_1)v_2$ and $(1 - v_1)(1 - v_2)$, and this process is repeated ad infinitum in theory, but it can terminate when we have N smaller sticks[2] for some suitably chosen large N (see Ishwaran and James (2001) for more details [9]).

Above all, the DP has a large weak support under mild conditions, that is G does not assign positive probability to events of 0 probability under G_0. In other words, any distribution with the same support as the base measure G_0 can be adequately approximated by a DP. As Peter Mueller once said[3]: "BNP is always right in the sense, no matter the true distribution, our ANOVA DDP prior always puts some probability mass in some neighborhood of the truth so we can learn about it. It has full support." (link to video)

3 Proposed Summer Agenda

Here is a tentative action plan for my potential contributions towards `pymc3` (see Table 3 for more details). Broadly speaking, during the Community Bonding phase (May 17 to June 6), I intend to meet active members either one-on-one or in a small group via Zoom, learn more about the theoretical underpinnings and guarantees of DPs and be more acquainted with the PyMC3 codebase. To do so, I would start working on some small issues as this will also allow me to familiarize myself with the building, testing, deployment and code review processes for PyMC3. Most importantly, I will ensure that the assigned mentors all agree with the project design

[2] This finite estimation is referred as a "Truncated" Dirichlet Process and it should not significantly influence results in practice.

[3] He was talking about ANOVA dependent Dirichlet processes (DDP) for transition times in a dynamic treatment regime [22].

before jumping into the implementation phase (see Weeks 2 – 5 in Table 3).

During the ten weeks of Google Summer of Code (June 7 to August 16), my primary goals will be twofold: implement a submodule for DPs (similar to `pymc3.gp` for Gaussian Processes) and create up-to-date notebooks in which I provide an in-depth description about the methodology and how to use such nonparametric methods in PyMC3. Creating a submodule may be convenient for creating classes and functions in the implementation of DPs. Austin Rochford's notebook available here will serve me well in implementing my proposed GSoC project alongside the NormalMixture distribution implemented here. The second part of my GSoC project's overarching aims would be to provide notebooks for the community on how and when to use the aforementioned BNP estimation methods; this would also be added to the `pymc3-examples` repository.

I anticipate that implementing such classes well and writing well-documented tutorial notebooks can be swift in principle but long in practice. An important skill that I hope to acquire is not only good practices to open source development, but also how to be an active contributor and member of the PyMC community following my GSoC experience. From properly documenting my contributions to ensuring that the methods in my Python classes are well-written yet concise, I truly believe that the main takeaway from a prospective GSoC experience would be the foundation for future open source development, particularly for PyMC3. Given that I intend to develop my thesis around causal inference methods within the Bayesian paradigm, I also believe that it would be nice to promote the use of Python and especially the PyMC3 framework for medical researchers, epidemiologists and statisticians who may find the learning programming daunting.

I also believe that it would be ideal to fulfill secondary goals such as contributing other DP-related methods for PyMC3 and their appropriate tutorial notebooks (e.g. ANOVA DDP [3] or any examples from Mueller et al. (2015) [13]), ensuring that the proposed BNP methods scale well in presence of high-dimensional or large-scale datasets and how to mitigate their use of computational resources and implementing a recently published paper in which researchers use DP-related BNP methods (see Section 4).

Week	Goals and Target Deliverables
Community Bonding phase Monday, May 17 – Sunday, June 6	• Read up on BNP and DP theory [13] • Discuss project design with mentors • Schedule coffee chats with members of the PyMC community! • Address minor issues through pull requests (PR)
Week 1 Monday, June 7 – Sunday, June 13	• Go over Austin's notebook • Replicate one or two data analyses in PyMC3 from Mueller et al. (2015) (R code available here) [13]
Weeks 2 – 5 Monday, June 14 – Sunday, July 11	Implement `pymc3.dp` submodule! This involves creating a class object for the following: • Stick-breaking weights; • DPs; • DP Mixtures (DPM) by leveraging `Mixture` and `NormalMixture`. all while ensuring proper testing and documentation of my code!
Evaluation Week Monday, July 12 – Sunday, July 18	• Peer-mentor evaluations • Adapt schedule depending on progress
Weeks 6 & 7 Monday, July 19 – Sunday, July 25	• Continue testing (unit & integration) and documentation of code • Read up about DP-related methods (e.g. ANOVA DDP, Pólya Trees, etc.)
Week 8 Monday, July 26 – Sunday, August 8	• Extend DP submodule to one or two more DP-related methods such as the ANOVA DDP
Week 9 & 10 Monday, August 9 – Sunday, August 22	• Extra two weeks in the schedule to wrap up any of the aforementioned tasks if not fully completed

Table 1: Detailed Schedule for the proposed 2021 Google Summer of Code Dirichlet Process project.

4 Benefits to Community

Recent statistical methodological extensions of DP-related methods showcase the increasing attention towards BNP methods in the research community [7, 12, 23]. Particularly in causal inference, researchers are shifting towards employing flexible models by reducing parametric assumptions; in the frequentist setting, semi-parametric and doubly-robust methods have been garnering a lot of attention in the last decade, especially in literature surrounding dynamic treatment regimes [11, 16, 18, 20, 21]. From a Bayesian perspective, while literature surrounding causal BNP methods remains relatively scarce, there are many opportunities for methodological extensions in this area [6, 8, 10, 14, 22]. With many researchers developing their own R package for for work, having a foundation for DP methods in PyMC3 would allows statisticians to use this framework for methods-related research surrounding DPs. Likewise, having support for such BNP methods could in turn encourage users to discover the benefits of flexible modelling techniques in a Bayesian framework and indirectly promote the use of DP-related methods in their own data analyses or research.

5 Final Remarks

5.1 Miscellaneous Summer Events

There are two things that I will have to do this summer where I will be unavailable to work on my proposed project for a few days: move to Toronto (I currently live in Montreal) and write my PhD comprehensive exam. I will be moving sometime around May or June and my PhD exam will most likely take place in the end of August (I will find out by the end of April). While it is important for me to let you know about these events, I will be able to adapt my schedule such that I can dedicate on average 18 hours per week to my proposed project.

5.2 Final Final Remarks

There are many things about open source development that I have yet to realize that I don't know. As much as I would like to use the GSoC opportunity to contribute to PyMC3 and to complement my summer study plan for the PhD comprehensive exam, I recognize that there can be many unexpected challenges that I will face. No matter what they are, I am confident that I will excel.

References

[1] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. Szerlip, P. Horsfall, and N. D. Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

[2] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: a probabilistic programming language. *Grantee Submission*, 76(1):1–32, 2017.

[3] M. De Iorio, P. Müller, G. L. Rosner, and S. N. MacEachern. An anova model for dependent random measures. *Journal of the American Statistical Association*, 99(465):205–215, 2004.

[4] S. Ghosal and A. Van der Vaart. *Fundamentals of nonparametric Bayesian inference*, volume 44. Cambridge University Press, 2017.

[5] P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[6] T. H. Guimond. *A Nonparametric Bayesian Approach to Causal Modelling*. PhD thesis, 2018.

[7] B. P. Hejblum, C. Alkhassim, R. Gottardo, F. Caron, R. Thiébaut, et al. Sequential dirichlet process mixtures of multivariate skew t-distributions for model-based clustering of flow cytometry data. *The Annals of Applied Statistics*, 13(1):638–660, 2019.

[8] J. L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

[9] H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

[10] C. Kim, M. J. Daniels, B. H. Marcus, and J. A. Roy. A framework for bayesian nonparametric inference for causal effects of mediation. *Biometrics*, 73(2):401–409, 2017.

[11] M. R. Kosorok and E. E. Moodie. *Adaptive Treatment Strategies in Practice: Planning Trials and Analyzing Data for Personalized Medicine*, volume 21. SIAM: Pennsylvania, USA, 2015.

[12] K. McGregor, A. Labbe, C. M. Greenwood, T. Parsons, and C. Quince. Microbial community modelling and diversity estimation using the hierarchical pitman-yor process. *bioRxiv*, 2020.

[13] P. Müller, F. A. Quintana, A. Jara, and T. Hanson. *Bayesian nonparametric data analysis*. Springer, 2015.

[14] T. A. Murray, Y. Yuan, and P. F. Thall. A bayesian machine learning approach for optimizing dynamic treatment regimes. *Journal of the American Statistical Association*, 113(523):1255–1267, 2018.

[15] E. G. Phadia. *Prior processes and their applications*. Springer, 2015.

[16] J. M. Robins. Optimal structural nested models for optimal sequential decisions. In *Proceedings of the Second Seattle Symposium in Biostatistics*, pages 189–326. Springer, 2004.

[17] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck. Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2:e55, 2016.

[18] J. Schulz and E. E. Moodie. Doubly robust estimation of optimal dosing strategies. *Journal of the American Statistical Association*, pages 1–13, 2020.

[19] J. Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

[20] E. J. T. Tchetgen and I. Shpitser. Semiparametric theory for causal mediation analysis: efficiency bounds, multiple robustness, and sensitivity analysis. *Annals of statistics*, 40(3):1816, 2012.

[21] M. P. Wallace and E. E. Moodie. Doubly-robust dynamic treatment regimen estimation via weighted least squares. *Biometrics*, 71(3):636–644, 2015.

[22] Y. Xu, P. Müller, A. S. Wahed, and P. F. Thall. Bayesian nonparametric estimation for dynamic treatment regimes with sequential transition times. *Journal of the American Statistical Association*, 111(515):921–950, 2016.

[23] S. Zhang, P. Müller, and K.-A. Do. A bayesian semiparametric survival model with longitudinal markers. *Biometrics*, 66(2):435–443, 2010.