

# Probabilistic Programming and Bayesian Methods for Hackers

## *Using Python and PyMC*

Bayesian method is the natural approach to inference, yet it is hidden from readers behind chapters of slow, mathematical, analysis. The typical text on Bayesian inference involves two to three chapters on probability theory, then enters what Bayesian inference is. Unfortunately, due to mathematical intractability of most Bayesian models, the reader is only shown simple, artificial examples. This can leave the user with a "so-what" feeling about Bayesian inference. In fact, this was the author's own prior opinion.

After some recent success of Bayesian methods in machine-learning competitions, I decided to investigate the subject again. Even with my mathematical background, it took me three straight-days of reading examples and trying to put the pieces together to understand how the method works so well. That being said, I suffered then so the reader would not have to now. The problem with my misunderstanding was the disconnect between Bayesian mathematics and probabilistic programming. This book attempts to bridge the gap.

If Bayesian inference is the destination, then mathematical analysis is a path to it. On the other hand, computing power is cheap enough that we can afford to take an alternate route via probabilistic programming. The path is much more useful, as it denies the necessity of mathematical intervention at each step, that is, we remove often-intractable mathematical analysis as a prerequisite to Bayesian inference. Simply put, this path proceeds via small intermediate jumps from beginning to end, where as the first path proceeds by enormous leaps, often landing far away from our target. Furthermore, with a tuned-mathematical background, the analysis required by the first path cannot even take place.

*Probabilistic Programming and Bayesian Methods for Hackers* is designed as a introduction to Bayesian methods and inference from a computation/understanding-first, and mathematical-second, point of view. Of course as an introductory book, we can only leave it at that: an introductory book. For the mathematically trained, they may supplement this text with other texts designed with mathematical analysis in mind. For the enthusiast with less mathematical-background, or one who is not interested in the mathematics but simply the practice of Bayesian methods, this text should be sufficient.

The choice of PyMC as the probabilistic programming language is two-fold. As of this writing, there is currently no central resource for examples and explanation in the PyMC universe. The official documentation assumes prior knowledge of Bayesian inference and probabilistic programming. We hope this book encourages users at every level to look at PyMC. Secondly, with recent core developments and popularity of the scientific stack in Python, PyMC is likely to become a core component of the stack.

PyMC does have some dependencies to run, namely NumPy and (optionally) SciPy. To not limit the

user, the examples in this book will rely only on PyMC, NumPy and SciPy only.

## Development

This book has an unusual development design. The content is open-sourced, meaning anyone can be an author. Authors submit content or revisions using the GitHub interface. After a major revision or addition, we collect all the content, compile it to a PDF, and increment the version of *Probabilistic Programming and Bayesian Methods for Hackers*.

## Contributions and Thanks

Thanks to all our contributing authors, including (in chronological order):

- [Cameron Davidson-Pilon](#)
- Andrew Hand
- [Stef Gibson](#)

We would like to thank the Python community for building an amazing architecture. We would like to thank the statistics community for building an amazing architecture.

One final thanks. This book was generated by IPython Notebook, a wonderful tool for developing in Python. We thank the IPython community for developing the Notebook interface. All IPython notebook files are available for download on the GitHub repository.

## How to contribute

### What to contribute?

- The current chapter list is not finalized. If you see something that is missing (MCMC, MAP, Bayesian networks, good prior choices, Potential classes etc.), feel free to start there.
- Cleaning up Python code and making code more PyMC-esque.
- Giving better explanations
- Contributing to the IPython notebook styles.

### Installation and configuration

- IPython 0.14 is a requirement to view the ipynb files. It can be downloaded [here](#)
- For Linux users, you should not have a problem installing Numpy, Scipy and PyMC. For Windows users, check out [pre-compiled versions](#) if you have difficulty.
- In the styles/ directory are a number of files that are customized for the *pdf version of the book*. These are not only designed for the book, but they offer many improvements over the default settings of matplotlib and the IPython notebook. The in notebook style has not been finalized yet.
- Currently the formatting of the style is not set, so try to follow what has been used so far, but inconsistencies are fine.

## Committing

- All commits are welcome, even if they are minor ;)
- If you are unfamiliar with Github, you can email me contributions to the email below.

## Contact

Contact the main author, Cam Davidson-Pilon at [cam.davidson.pilon@gmail.com](mailto:cam.davidson.pilon@gmail.com) or [@cmrndp](#)