Probabilistic Programming and Bayesian Methods for Hackers

Probabilistic Programming & Bayesian Methods for Hackers

Using Python and PyMC

The 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 the methods. There was simply not enough literature bridging theory to practice. The problem with my misunderstanding was the disconnect between Bayesian mathematics and probabilistic programming. That being said, I suffered then so the reader would not have to now. This book attempts to bridge the gap.

If Bayesian inference is the destination, then mathematical analysis is a particular path to towards it. On the other hand, computing power is cheap enough that we can afford to take an alternate route via probabilistic programming. The latter 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 latter computational 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, without a strong mathematical background, the analysis required by the first path cannot even take place.

Bayesian Methods for Hackers is designed as a introduction to Bayesian inference from a computational/understanding-first, and mathematics-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 cure the curiosity this text generates 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 and entertaining.

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 explanations 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 soon enough.

PyMC does have dependencies to run, namely NumPy and (optionally) SciPy. To not limit the user, the examples in this book will rely only on PyMC, NumPy, SciPy and Matplotlib only.

Contents

(The below chapters are rendered via the *nbviewer* at nbviewer.ipython.org/ (http://nbviewer.ipython.org/), and is read-only and rendered in real-time. Interactive notebooks + examples can be downloaded by cloning!)

- Prologue: (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Prologue/Prologue.ipynb) Why we do it.
- Chapter 1: Introduction to Bayesian Methods (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter1_Introduction/Chapter1_Introduction.ipynb) Introduction to the philosophy and practice of Bayesian methods and answering the question "What is probabilistic programming?" Examples include:
  - Inferring human behaviour changes from text message rates.
- Chapter 2: A little more on PyMC (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter2_MorePyMC/MorePyMC.ipynb) We explore modeling Bayesian problems using Python's PyMC library through examples. How do we create Bayesian models? Examples include:
  - Detecting the frequency of cheating students, while avoiding liars.
  - Calculating probabilities of the Challenger space-shuttle disaster.
- Chapter 3: Opening the Black Box of MCMC (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter3_MCMC/IntroMCMC.ipynb) We discuss how MCMC operates and

diagnostic tools. Examples include:
    - Bayesian clustering with mixture models
- Chapter 4: The Greatest Theorem Never Told (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter4_TheGreatestTheoremNeverTold/LawOfLargeNumbers.ipynb) We explore an incredibly useful, and dangerous, theorem: The Law of Large Numbers. Examples include:
    - Exploring a Kaggle dataset and the pitfalls of naive analysis
    - How to sort Reddit comments from best to worst (not as easy as you think)
- Chapter 5: Would you rather loss an arm or a leg? (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter5_LossFunctions/LossFunctions.ipynb) The introduction of Loss functions and their (awesome) use in Bayesian methods. Examples include:
    - Solving the Price is Right's Showdown
    - Optimizing financial predictions
    - Winning solution to the Kaggle Dark World's competition.
- Chapter 6: Getting our *prior*-ities straight (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter6_Priorities/Priors.ipynb) Probably the most important chapter. We draw on expert opinions to answer questions. Examples include:
    - Multi-Armed Bandits and the Bayesian Bandit solution.
    - what is the relationship between data sample size and prior?
    - estimating financial unknowns using expert priors.

  We explore useful tips to be objective in analysis, and common pitfalls of priors.
- Chapter X1: Bayesian Markov Models
- Chapter X2: Bayesian methods in Machine Learning We explore how to resolve the overfitting problem plus popular ML methods. Also included are probablistic explainations of Ridge Regression and LASSO Regression.
    - Bayesian spam filtering plus *how to defeat Bayesian spam filtering*
    - Tim Saliman's winning solution to Kaggle's *Don't Overfit* problem
- Chapter X3: More PyMC Hackery We explore the gritty details of PyMC. Examples include:
    - Analysis on real-time GitHub repo stars and forks.
- Chapter X4: Troubleshooting and debugging

More questions about PyMC? Please post your modeling, convergence, or any other PyMC question on cross-validated (http://stats.stackexchange.com/), the statistics stack-exchange.

Using the book

The book can be read in three different ways, starting from most recommended to least recommended:

1. The most recommended option is to clone the repository to download the .ipynb files to your local machine. If you have IPython installed, you can view the chapters in your browser *plus* edit and run the code provided (and try some practice questions). This is the preferred option to read this book, though it comes with some dependencies.

    - IPython 0.13 is a requirement to view the ipynb files. It can be downloaded here

    (http://ipython.org/)
    - For Linux users, you should not have a problem installing Numpy, Scipy, Matplotlib and PyMC. For Windows users, check out pre-compiled versions (http://www.lfd.uci.edu/~gohlke/pythonlibs/) if you have difficulty.
    - In the styles/ directory are a number of files (.matplotlirc) that used to make things pretty. These are not only designed for the book, but they offer many improvements over the default settings of matplotlib and the IPython notebook.
    - while technically not required, it may help to run the IPython notebook with `--pylab inline` if you encounter runtime errors.

2. The second, preferred, option is to use the nbviewer.ipython.org site, which display IPython notebooks in the browser (example (http://nbviewer.ipython.org/urls/raw.github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/master/Chapter1_Introduction/Chapter1_Introduction.ipynb)). The contents are updated synchronously as commits are made to the book. You can use the Contents section above to link to the chapters.
3. PDF versions are coming. PDFs are the least-prefered method to read the book, as pdf's are static and non-interactive. If PDFs are desired, they can be created dynamically using Chrome's builtin print-to-pdf feature.

Installation and configuration

If you would like to run the IPython notebooks locally, (option 1. above), you'll need to install the following:

- IPython 0.13 is a requirement to view the ipynb files. It can be downloaded here (http://ipython.org/ipython-doc/dev/install/index.html)
- For Linux users, you should not have a problem installing Numpy, Scipy and PyMC. For Windows users, check out pre-compiled versions (http://www.lfd.uci.edu/~gohlke/pythonlibs/) if you have difficulty.
- also recommended, for data-mining exercises, are PRAW (https://github.com/praw-dev/praw) and requests (https://github.com/kennethreitz/requests).
- In the styles/ directory are a number of files that are customized for the notebook. 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.

Contributions and Thanks

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

- Cameron Davidson-Pilon (http://www.camdp.com)
- Stef Gibson (http://stefgibson.com)
- Vincent Ohprecio (http://bigsnarf.wordpress.com/)
- Lars Buitinck (https://github.com/larsman)
- Paul Magwene (http://github.com/pmagwene)
- Matthias Bussonnier (https://github.com/Carreau)
- Jens Rantil (https://github.com/JensRantil)
- y-p (https://github.com/y-p)
- Ethan Brown (http://www.etano.net/)

- Jonathan Whitmore (http://jonathanwhitmore.com/)
- Mattia Rigotti (https://github.com/matrig)
- Colby Lemon (https://github.com/colibius)
- Gustav W Delius (https://github.com/gustavdelius)
- Matthew Conlen (http://www.mathisonian.com/)
- Jim Radford (https://github.com/radford)
- Vannessa Sabino (http://baniverso.com/)
- Thomas Bratt (https://github.com/thomasbratt)
- Nisan Haramati (https://github.com/nisanharamati)
- Thomas Bratt (https://github.com/thomasbratt)
- Robert Grant (https://github.com/bgrant)
- Yaroslav Halchenko (https://github.com/yarikoptic)
- Alex Garel (https://github.com/alexgarel)

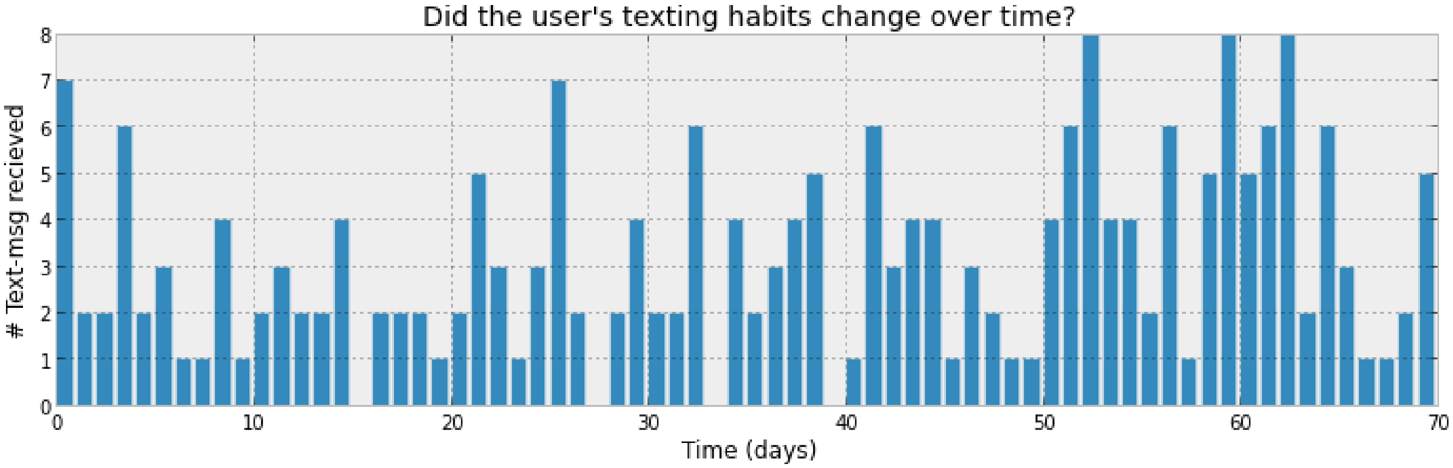
We would like to thank the Python community for building an amazing architecture. We would like to thank the statistics community for building an amazing architecture.

Similarly, the book is only possible because of the PyMC (http://github.com/pymc-devs/pymc) library. A big thanks to the core devs of PyMC: Chris Fonnesbeck, Anand Patil, David Huard and John Salvatier.

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.

Contact

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



Imgur

```
from IPython.core.display import HTML
def css_styling():
    styles = open("../styles/custom.css", "r").read()
    return HTML(styles)
css_styling()
```

Back to top