PS2720: Bayesian Data Analysis for Social Scientists Syllabus

Michael Colaresi
mcolaresi@pitt.edu

Spring 2018

Class meeting:
Wed. 9:30am-12:00pm
4430 Posvar

Office hours:
1:30-2:30pm Wed. and by apt.
4619 Posvar

Being able to learn about social interactions and choices, from civil war to democratic accountability, after viewing new pieces of information, be they new conflict events or a specific tally for a party, is the fundamental purpose of conventional quantitative social science research and one important component of the emerging inter-disciplinary field of computational social science. While Bayes rule supplies an elegant framework for updating prior beliefs through observations, until recently computational limitations either necessitated diversions from this straightforward approach or severely limited the use of Bayesian inference to a small set of analytically tractable representations.

Modern computing power, coupled with advances in stochastic sampling (both theoretical and applied), now allow researchers to transcend these previous limitations. It is currently possible for students, after a one semester graduate course, to estimate the posterior distribution of parameters that comprise a unique domain specific model tightly informed by theory that includes a complicated hierarchical structure and multiple indicators. This flexibility has led to advances in the topic modeling of texts, the measurement of latent concepts across datasets, and more accurate forecasting systems.

Yet, to fully harness these exciting advances, researchers need to rethink the roles that 1) models and information and 2) computers play in their analyses. Conventional frequentist treatments of inference, that are common across the social sciences, are built upon a very different foundation from the Bayesian approach. Where conventional approaches offer a fixed selection of pre-qualified models, Bayesian computation invites you to roll your own formalization or combine and mix multiple model components as you see fit. As frequentist inference sweeps prior information under the rug, Bayesian models explicitly define priors as part of the underlying model. Where Bayesian analyses condition on uncertainty, null-hypothesis significance testing imagines certainty. Perhaps most crucially, while it is conventional to evaluate models based on in-sample statistics (such as p-values), Bayesian-influenced analysis seeks measures of generalization performance to new data and situations. These distinctions are more than nit-picky philosophical sideshows, they define the very inferences that our models are built to inform.

Related to the previous points, because Bayesian inference in non-trivial mathematically – tracking the propagation of usually high dimensional information from one state (your prior) to another (the posterior) after observing new instances involves high dimensional integrals – computers are an applied Bayesian data analyst's best friend. Thus, researchers must be sufficiently comfortable writing programs that unlock a computers potential to run through thousands, if not millions, of calculations to draw useful samples for analysis. Moreover, these calculations need to be made on your model, not just any generic model (in most cases). Thus, researchers need to be able to create programs that express their domain knowledge of a problem in a way that a computer can understand and run with.

Not surprisingly, advances in applied Bayesian analysis in weather forecasting, genetics, natural language process and other domains have proceeded in lockstep with software that allows users to enlist significant computational resources to learn from data. The evolution of Bayesian modeling languages, from BATS to WinBugs then OpenBugs and JAGS to the more recent Stan, track the increasingly expressive syntax readily available to Bayesian modelers.[1] Modern tools like Stan reward researchers that are looking to kick the habit of canned routines and offer programmers both pre-existing building blocks and extensibility.

Note: The syllabus is subject to updates as the semester progresses. Please check courseweb for news and course information.

Goals

The goals of this class are to train students to a) use Bayesian inference to build models that represent their understanding of the social phenomena of interest as well as competing or complementary representations, b) understand how to propagate information from the data, through the model, into a new computed representation of the process, c) interrogate a set of models to illuminate flaws that can be improved upon iteratively, d) use predictions to validate the usefulness of sets of models, and e) clearly understand the limits of Bayesian inference. To accomplish these goals, the class will introduce the Stan modeling language as well as guide students in writing some samplers from scratch.

Performance metrics

Assignments, Midterm, Final Project

There will be 8 assignments and a final project.

- *Assignments*: Each assignment is worth 5 percent of your grade. All of the assignments include computational components and these should be completed in either R or Python. We will be doing class readings related to R, but for those that want to pursue Python, that is no problem. Assignments are due by midnight on the day listed below. Your name and the assignment number should be the title of the files when you email them in to me. I may change these due dates (to give you more time) if our pace is slower than I anticipate.

[1] While BATS had a Readme file that was approximate 50 lines long and a few pages of introduction in a textbook in 1994, and WinBugs had an 60 page manual, including references in 2008, the Stan-ual is over 635 pages, as of version 2.17.0, and growing with each release.

- Assignment 1, from SR pg. 45-47: 2M1, 2M2, 2H1, 2H2, 2H3. Due Friday, January 19.
- Assignment 2, from SR pg. 69-70: 3E2, 3E3, 3E4, 3E6, 3E7, 3H1, 3H2, 3H3. Due Friday, January 26.
- Assignment 3, from SR pg. 115-116: 4E1, 4E2, 4E3, 4M1, 4M2,4H1. Due Friday, February 2.
- Assignment 4, from SR pg. 163-164: 5H1, 5H2, Extra credit: 5H3. Due Friday February 12.
- Assignment 5, from SR pg. 205-207. 6M5, 6M6, 6H1, 6H2, 6H3, 6H4, 6H5. Due February 23.
- Assignment 6, from SR pg. 264. 8E1, 8E2, 8E3, 8E4, 8E5, 8E6, 8H6, then discuss what is different about HMC as compared to the Metropolis algorithm. Due March 2.
- Assignment 7, from SR pg. 329: 10E1, 10E2, 10E3, 10E4, 10M1, 10M2, also: take example 10.1.3 on page 304-310(top) and rerun these analyses using Stan data, parameter, model and generated quantities blocks for the models (10.6, 10.7, 10.8, 10.9). Due March 16
- Assignment 8 from SR pg. 352-353 and 384-385: 11H1 and 11H2, 12E5, 12M3. Due April 6.

- *Final project*: Students will be writing a research report on a topic that they are interested in. The project should be turned in as an R markdown file or jupyter notebook. You can begin a new project using a Bayesian perspective or take a published paper and port it into our Bayesian inferential framework (if the previous project was Bayesian, you much make a substantial change, in this case, talk to me first). This final project will be worth 35 percent of your final grade. The sections should be as follows:
  - *Research Question*: What is the central research question and puzzle? Why is it important?
  - *Formal Process*: What is the formal process you are trying to measure? Write out, in an equation or more likely in a set of equations, what the crucial underlying latent concepts are, how they relate to each other, and how they are likely to lead to observable indicators. How does this process help to illuminate answers to your central research question/puzzle? If you are comparing several competing approaches, do the above for each approach (advice: keep it manageable).
  - *Model*: What set of models will you be estimating? How are they related to the formal process(es) you laid out in the previous step? What priors have you chosen and why? What are you doing about the potential for multi-level structure, measurement contamination, any missing data, and nonlinearities? In this step, you will also define what data you are using and how you plan to compute the parameters of the model (eg analytically, HMC, etc).
  - *Code*: What is the code for the model(s)? How are you estimating the posterior distribution of the parameters? What quantities of interest do you eventually want to compute, provide code for that also.

- *Computation*: Use your code to estimate the parameters and quantities of interest, showing each step including formating your data for the model estimation and post-estimation computation of quantities of interest steps.
- *Interpretation*: Appropriately interpret the posterior distributions and quantities of interest. Did the model converge? What is your evidence for that? Evaluate/compare the predictive performance of the model(s) you ran either using information criteria or held-out data.
- *Criticism*: What is a statistic or visualization you can use to critique the performance of your model(s)? After calculating that statistic or creating that visualization, what are the deficiencies of each model representation? How does the statistic/visualization you chose lead to that conclusion?
- *Improvement*: What improvement(s) could be made to your model(s) based on the results of the model criticism step? How would they be implemented in a new formal process and how would that be modeled?

Attendance and Engagement

The final 25 percent of your grade is based on your attendance in class and your engagement with the material. Each weekly class will be a mix of a lecture, with questions and discussion of the readings, and computational workshop, with hands-on coding practice. Thus, you will need your laptop in each class, at least for the workshop portion.

I expect you to have read for class and be prepared to have a high-level discussion of the material. Your role in the discussion could be asking quality questions about the material or helping to clarify points from the readings and lectures. I would like everyone in the class to participate in discussion. Moreover, another form of engagement is effort during the computational workshops in the class.

Tools for Success

Software

You should have, at a minimum, a bash shell, R, Stan and a useful text editor installed on your computer. While I expect the majority of students will be using R and Stan, I am fine if students replace R with Python (but note the book will not be as helpful directly). In addition, I reserve the right to introduce a few ideas in Python as the semester goes on (because it is a terrific language).

You should be able to do everything we do in class using modern Mac, Linux and even Windows operating systems that allow compilable code. If you are a Windows user, please follow these steps or these steps to install bash or the Windows subsystem for Linux. If you have questions, ask me. In the course, we will be using free, open-source software. As noted above, it will be necessary to have a laptop in class as we begin to code.

Installation instructions are below:

- *R*: The official version of R is open source and available here. This installation gives you base R, the language interpreter with a minimal set of functions. R is a more fiddly language

than Python generally, but has very easy, intuitive add-on packages (also free) especially for cleaning and plotting data. We will mainly be munging data in R, calling Stan and then plotting results. Make sure you install the tidyverse packages, see here. Follow the instructions for the "installation", these commands would be typed in RStudio (see the next step below). Do not forget rtools if you are a windows user.

- *RStudio*: Once you have R, you should install RStudio to run your R code. Many people find it very helpful. You want the free desktop version of RStudio.

- *Stan*: Stan is a programing language that can be run from the command line or called from a variety of other languages. You should install rstan and pystan at a minimum. The Stan-ual is the Wes Anderson of software manuals (it has moments of brilliance, delineates a well planned out world, but it could be more accessible). You can find it here. This page also includes some terrific tutorials.

- *A text editor*: Sublime Text, BBEdit, emacs, vim, TextWranger, atom are all good options for text editors to write code in (Sublime Text is written in Python, btw). You do not want to ever write code in Word or the mis-named Mac TextEdit (which does not save to plain text by default). Really. RStudio can suffice in a pinch for those just using R, but you should get familiar with at least one good text editor sooner rather than later.

- Optional: *Python*: I strongly suggest the free installation by Anaconda. Python is a rather easy to learn computer language that can do quite a bit. It is especially prevalent in the text-as-data and machine learning community. Python has an extremely strong developer community that supplies add-on libraries and Anaconda (using the conda program) provides a useful and convenient repository of many of these. One of Python's advantages is jupyter notebooks, which serve as browser-based development environments. This can be downloaded at the command line with conda install jupyter-notebook. The use of Python or R (see below) is really a matter of taste and strategic choice for this class. It is easy to call Stan from either R (using rstan) or Python (using pystan).

Books

There are many terrific books for Bayesian inference. We will use:

- McElreath, Richard. 2015. Statistical Rethinking: A Bayesian Course with Examples in R and Stan. CRC Press. First Edition. *SR*

Since computation is a key part of this course, we will also be using:

- Wickham, Hadley. R for Data Science. Can be downloaded free here: http://r4ds.had.co.nz. *R4DS*

For those that want to pick up Python, I suggest:

- Muller, Andreas and Sarah Guido. 2017. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly press.

Other terrific books on Bayesian inference include:

- Hoff, Peter. 2009. A First Course in Bayesian Statistical Methods. Springer. First Edition. I will draw parts of a few lectures from this book as I love its introduction to Bayesian computation and MCMC.
- Kruschke, John K. 2014. Doing Bayesian Data Analysis. Academic Press. Second Edition. This book (sometimes known as the dog book) does just about the same thing as *SR*. It has a great chapter on HMC that we will read.
- Jackman, Simon. 2009. Bayesian Analysis for the Social Sciences. Wiley. First Edition. A bonus is that it was written by a political scientist. A problem is that it uses JAGS not Stan. It does go over some extremely useful advanced topics like IRT models.
- Gill, Jeff. 2014. Bayesian Methods: A Social and Behavioral Science Approach. CRC. 3rd Edition. Another good resource from a statistician active in social science.
- Lynch, Scott. 2007. Introduction to Applied Bayesian Statistics and Estimation for Social Scientists. Springer. Another good, but slightly dated intro book.
- Gelman, Andrew, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. 2013. Bayesian Data Analysis. CRC. BDA3 is more of a reference text, more comprehensive and challenging than the others.
- Jim Albert's Bayesian Computation in R is also a good resource.

More Resources

There is a remarkable amount of information out there to learn Bayesian inference. Here are a few interesting places to explore:

- Jayne's notes: http://bayes.wustl.edu/etj/science.pdf.html
- Stan tutorials: http://mc-stan.org/users/documentation/tutorials.html
- Chi Feng's interactive visualizations of MCMC algorithms: http://chi-feng.github.io/mcmc-demo.

Schedule of Topics

Week 1: Wednesday, January 10

Reorganizing information and get restarted. Have software installed.

- SR, Intro and Chapter 1, xi-17
- R4DS, Introduction. Chapters 1, 2 and 3
- Jackman, Simon, Into chapter from Bayesian Analysis for the Social Sciences, provided by instructor.

- Wasserstein, R. & Lazar, N. 2016. The ASAs Statement on p-Values: Context, Process, and Purpose The American Statistician

Week 2: Wednesday, January 17

Reinterpreting models and inferences from a Bayesian point of view

- SR, Chapter 2, pgs. 19-47
- R4DS, Chapter 4, 5 and 6
- Gill, Jeff. 1999. The Insignificance of Null Hypothesis Significance Testing. Political Research Quarterly. 52(3): 647- 674

Week 3: Wednesday, January 24

Still Reinterpreting models and inferences from a Bayesian point of view

- SR, Chapter 3, pg. 49-70
- R4DS, Chapter 7 and 8
- Western, B. and Jackman, S. 1994. Bayesian Inference for Comparative Research. American Political Science Review, 88(2):412-423.

Week 4: Wednesday, January 31

Rethinking simple regression

- SR, Chapter 4 pg. 71-117
- R4DS, Chapters 9, 10, and 11
- Seaman, J. W. I., Seaman, J. W. J., and Stamey, J. D. 2012. Hidden Dangers of Specifying Non-informative Priors The American Statistician, 66(2):77-84.

Week 5: Wednesday, February 7

Rethinking multiple regression

- SR, Chapter 5 pg. 119-164
- R4DS, Chapters 12, 13, 15, 15
- Horiuchi, Y. Imai, K., and Taniguchi, N. 2007. Designing and Analyzing Randomized Experiments: Application to a Japanese Election Survey Experiment American Journal of Political Science. 51(3):669-687.

Week 6: Wednesday, February 14

Relearning how to do applied research

- SR, Chapter 6 and 7 pg. 165-239
- R4DS, Chapters 17, 18, 19
- Montgomery, Jacob, Florian Hollenback, and Michael D. Ward "Improving Predictions Using Ensemble Bayesian Model Averaging" Political Analysis 20(3): pg. 271-291.
- Warren, T. C. (2014). Not by the Sword Alone: Soft Power, Mass Media, and the Production of State Sovereignty. International Organization, 68(1):111-141.

Week 7: Wednesday, February 21

reMCMC

- SR, Chapter 8 pg. 239-265
- Kruschke, HMC/Stan chapter from Doing Bayesian Data Analysis, provided by instructor.
- R4DS, Chapters 20 and 21
- Jackman, S. 2000. Estimation and Inference via Bayesian Simulation: An Introduction to Markov Chain Monte Carlo. American Journal of Political Science 44, 375-404.
- Optional but fun: Von Hilgers, P. & Langville, A. here

Week 8: Wednesday, February 28

Rediscovering where GLMs come from: Maximum Entropy

- SR, Chapter 9 pg. 265-289
- Stone, James V. Intro chapters from Information Theory: A Tutorial Introduction, provided by instructor.
- R4DS, Chapters 22-25

Week 9: Wednesday, March 7

Recouping, No class spring recess

Week 10: Wednesday, March 14

Recounting Binomial and count representations

- SR, Chapter 10 pg. 291-330
- R4DS, Chapters 26-30
- Monroe, Burt, et al "Fightin' Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict" Political Analysis 16: pg. 372-403.

Week 11: Wednesday, March 21

Remixing theories into models

- SR, Chapter 11 pg. 331-353
- Fariss, C. J. (2014). Respect for Human Rights has Improved Over Time: Modeling the Changing Standard of Accountability. American Political Science Review, 108(2):297-318.

Week 12: Wednesday, March 28

Rebuilding hierarchical and multilevel models

- SR, Chapter 12 pg. 355-386
- Danneman, N. and Ritter, E. H. (2014). Contagious Rebellion and Preemptive Repression. Journal of Conflict Resolution, 58(2):254-279.

Week 13: Wednesday, April 4

Rebuilding hierarchical and multilevel models (continued)

- SR, First part of Chapter 13 pg. 387-410(top)
- Stegmueller, D. 2013. How Many Countries for Multilevel Modeling? A Comparison of Frequentist and Bayesian Approaches American Journal of Political Science. 57(3): 748-761.

Week 14: Wednesday, April 11

Reintroducing nonlinearities wth Gaussian processes

- SR, last part of Chapter 13, pg. 410-421
- Stan Gaussian Process tutorial by Michael Betancourt, https://betanalpha.github.io/assets/case_studies/gp_part1/part1.html

Week 15: Wednesday, April 18

weRe did they go?: Missing data and measurement, and other topics we can (over)fit in

- SR, Chapter 14, pg. 423-443
- Treier, S. and Jackman, S. (2008). Democracy as a Latent Variable. American Journal of Political Science, 52(1):201-217.

Exam Week: April 25

Final project presentations and files due