Introduction to Bayesian Modeling

Contents

1 Overview

1.1 Goal

The goal of this workshop is to introduce students to the concepts and practice of Bayesian modeling.

1.2 Target Audience

We expect that the typical student will be a graduate student, faculty member, staff member, or researcher in a quantitative field (such as computer science, statistics, engineering, or biology), who would like to learn more about Bayesian modeling.

1.3 Prerequisites

Prerequisites include calculus, some linear algebra, and some familiarity with introductory probability (e.g., we will assume prior familiarity with concepts such as expectation, conditional probability, and commonly used distributions, such as Gaussian and Poisson). The class will use Python as a common language. The workshop will employ student-centered components so **be prepared to**

spend 3-4 hours a day outside of the workshop working on material. This time will be spent on background reading and preparing code/demos/materials for the interactive component of the workshop.

1.4 Textbook

We will use [2], available online at `http://www.stat.columbia.edu/~gelman/book/`.

1.5 Philosophy

Learning in order to create is both more fun and more effective than *learning for some extrinsic purpose*. Hence, the workshop is structured so as to (a) be student-centered and (b) allow self-determination and autonomy in how students engage with the material.

1.6 Format

About half of the workshop will involve lectures via slides.

About half of the workshop will be interactive, including:

- Student “lightning chat" (10 minute) presentations. Something like one per student per workshop, depending on the number of students. To allow for student-centered direction and autonomy, students may choose any of the following:
  - Presentation of Python implementations of models from [1] , [2], or the workshop.
  - Presentation of an exercise from [2] or [1].
  - Reviewing a demo with us from `https://github.com/avehtari/BDA_py_demos`.
  - Presentation of a reading section, blog, etc. of interest.
  - Presentation of a mathematical derivation of something relevant to the course.
  - Presentation of how a concept relates to something from their research area.
- Real-time python applications lab – Google Collab exercises ? Python (rather than R) implementations of [1] and [2].
- Mini reading group discussions – They might not have time to read a whole paper, but we could discuss sections of a text at the very beginning, or perhaps sections of a relevant paper.

2 Topics

Below are topics we plan to cover in the course:

2.1 Introduction to Bayes

We present everything in here using conjugate models with closed-form posteriors. The models are useful in and of themselves, as well as to build intuition for more complicated models.

Primary references here are [1] and [2].

- **Why Bayes?** – See Section 1.3 of [1]. [3] has some nice plots motivating why use Bayesian linear regression over standard linear regression. [4] has some nice plots illustrating the Bayesian approach and how it mitigates overfitting. I can provide a nice example with biometric profiling of human typing dynamics. [5] has a nice simple example of obtaining non-standard functionals from the posterior that can be of interest. [6] presents the case for Bayesian deep learning.
- **Belief functions, Bayes rule** – Sections 2.1, 2.2 of [1]. [4] briefly overviews of the Bayesian framework. *Why most published research findings are false* [7] provides nice motivation. Could perhaps cover exchangeability here.

- **Binomial, Poisson, normal, multivariate normal models** – Sections 3.1, 3.2, 5, and 5 of [1]. Introduce the exponential family formalism [8] for much greater breadth.
- **Bayesian linear regression** – Section 9 of [1]. I have notes on this. There are some nice slides here which also illustrate the use of kernels.[1] Introduce model selection here (Section 9.3 of [1].
- **Bayesian workflow** – Lots of nice resources for Bayesian workflow. For example: [10] or [11]. Section 6 of [2] covers model checking. Some points to make re: model checking
  - *Samples from the posterior predictive should capture important properties of the observed dataset.* For a violation of this, see the normal model for Newcomb's speed of light measurements. (Compare Figures 6.2 and 3.1 of [2].)

We will want to find a way to get students to group up, probably based on domain expertise/interests, so that they can eventually work together on a project.

2.2 Methods

We introduce these methods, which can be used for models without closed-form posteriors. We practice applying them in the next section.

- **MCMC** - Karin will present, including an introduction to pymc3.
- **Variational inference** [9].

2.3 More complicated models

Here are some models which are still fairly standard, but lack conjugate priors, and so inference typically requires VI or MCMC. Karin: Where here, or elsewhere, would you like to illustrate applications of MCMC?

- **Hierarchical models** Hierarchical normal model (e.g. Gelman's 5 schools example). Hierarchical linear regression (Chapter 13 of [2], Secs 11.1-11.3 of [1].). Figure 11.1 and 11.3 (right) of [1] nicely shows the beneficial effect of sharing statistical strength in a hierarchical linear regression, as compared to many separate linear regressions.
- **Regression models for binary and multi-class data** Includes logistic regression, probit regression, binomial, multinomial, etc. Use this to cover additional inference techniques: auxilliary variable trick and Laplace variational inference. Show demo "beating scikit-learn with variational inference." See also pp. 390 of [1] for a useful warm-starting strategy. Could generalize to Bayesian GLMs. Could also cover or mention hierarchical extensions (i.e. Bayesian GLMM's).
- **Mixture models** I will give CAVI for Gaussian mixture models.
- **Time series models** Probably just hidden markov models, although would be nice to also introduce state space models. Could mention embedding of GLM's or GLMM's within them. May give some overview to probabilistic graphical models here.
- **Bayesian Deep Learning** Bayes and neural networks. 20-30 min w/ guest presenter, Kyle Heuton, Ph.D. student, computer science.

References

[1] Peter D Hoff. *A first course in Bayesian statistical methods*, volume 580. Springer, 2009.

[2] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.

[3] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[1] Nice Bayesian linear regression slides: `https://www.cs.toronto.edu/~rgrosse/courses/csc411_f18/slides/lec19-slides.pdf`

[4] Zoubin Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110553, 2013.

[5] Leonhard Held and Chris C Holmes. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian analysis*, 1(1):145–168, 2006.

[6] Andrew Gordon Wilson. The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*, 2020.

[7] John PA Ioannidis. Why most published research findings are false. *PLoS medicine*, 2(8):e124, 2005.

[8] Michael Wojnowicz. *The exponential family*. Available (with permission).

[9] Michael Wojnowicz. *Foundations of variational inference*. Available (with permission) at `https://github.com/mikewojnowicz/vi_foundations`.

[10] Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. Bayesian workflow. *arXiv preprint arXiv:2011.01808*, 2020.

[11] Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in bayesian workflow. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(2):389–402, 2019.

A Resources which may be appropriate for mini-reading group or student presentations

- Intro: *Why most published research findings are false.* [7]
- Model checking: Bayesian workflow.
- Textbook sections TBD.

B Additional problems to work on

B.1 Batting average dataset

The hierarchical normal model for (arcsine-transformed) batting average data on pp. 163 of [2] has some serious deficiencies, as exposed in Table 6.1 in the section on model checking.

Can you construct (and learn) a better model which makes predictions closer to the true final batting average?

Examples:

- Add an extra layer to the hierarchy, so that player p's 1970 batting average inherits from player p's overall batting average which in turn inherits from a population batting average. (Of course, I am speaking of the arcsine-transformed batting averages, so that we can use a hierarchical normal model.)
- Add an autoregressive component, because, as mentioned by Gelman, player batting averages *DO* change over time.

The text also does a poor job of checking the modeling assumption violations that were of concern. Can you do a better job of checking them, and if necessary, address them?

Examples:

- If batting averages are indeed heavy tailed or skewed, move from a normal distribution to something else. For example, could try a t-distribution with Laplace inference to handle the non-conjugacy.
- If the variance is indeed too high for a binomial model, try something that can handle the overdispersion.