Big Data and Security

Jeffrey Borowitz, PhD
Lecturer
Sam Nunn School of International Affairs

Bayesian Models and How to Fit Them

Computational Issues

- Fitting parametric models was like:
  - Take a function of data and parameters (like sum of squared residuals)
  - Pick parameters to optimize this function
  - Those are your parameters
- Bayesian methods have an exact computation for the result
  - We have the formula:

$$p(parameters|data) = \frac{p(data|parameters) \cdot p(parameters)}{p(data)}$$

- But what is in $p(data)$?
  - This is all the ways the data could be generated
  - This comes from all the different priors there could possibly be
  - This is an integral, whose difficulty to compute suffers the curse of dimensionality, badly!

Direct Calculation

- Bayes' rule gives a formula for p(parameters | data)
  - You could calculate it!
  - In simple cases, we do this
- But usually this is very hard computationally
  - You are calculating how likely the data is from a range of different values of parameters
  - This is very susceptible to the curse of dimensionality

So How To Compute Bayesian Models?

- Gibbs sampling/Metropolis-Hastings algorithm (a form of “Markov Chain Monte Carlo”)
- I love this - what a wonderful piece of applied math!
- What is the goal of “computing” this model?
  - We want to know about the distribution of the actual parameters
  - Then we would just take the most likely parameters
  - Or we would do inference about how likely parameters are to be in particular ranges
- But it’s really hard to calculate the entire distribution. . .

"Markov Chain Monte Carlo"

- It turns out under specific properties, you can simulate draws from the distribution
  - You start with a draw of parameters.
  - You compute the top of the fraction in Bayes' rule:

$$p(parameters) = p(data|parameters) \cdot p(parameters)$$

- You draw a new value of parameters, called $parameters_2$
  - If f($parameters_2$) > f($parameters$), keep $parameters_2$
  - Otherwise, maybe keep it and maybe throw it out.
- Eventually, you have a whole bunch of values of $parameters$ that you didn't throw out. This is your distribution of $parameters$!

MCMC: Computational Issues

- With a faster processor, we could optimize a function better
- But now, we need lots of randomness!
- And we need to make sure things have converged
- This requires lots of CPU, but little disk or memory

Choosing Priors

- One important issue is how to choose priors
  - One important fact about Bayesian models: if you put exactly 0 probability on a value of a parameter, you will never think that value of the parameter has any posterior probability either
- Want to make sure there's **some** probability on every possible value
- We often use "diffuse" priors
  - These are designed to put fairly equal weight on lots of the distribution
- Another common prior is "empirical" Bayes
  - You have a functional form for your prior (it's normally distributed)
  - Then, you estimate the parameters of this function from your data using a maximization type method.
  - Then, you use that prior and do Bayesian methods

When Are Bayesian Models Good?

- Theoretically they make sense
  - Do you really NOT want to incorporate prior information?
  - Good analysts would always include prior information
- You can express restrictions on classes of parameters which you can estimate with relatively little data
  - Customer preferences from a small set of purchases
  - Student abilities from a few years of test scores
- They provide a computational shortcut around the curse of dimensionality
- (In economics) if you're going to write down very complicated models, it can be easier to fit them.
  - Maximization type models might not converge, but Bayesian models are fit in this whole other way, so those might converge

Georgia Tech

Wages

- Let’s say wages are given by a model like:

$$wage_i = \alpha + \beta educ_i + \gamma_i + \varepsilon_i$$

- Where:
    - γ_i is a meaningful individual factor expressing an individual’s skill
    - ε_i is the regular error term
- And let’s say we have a couple observations per individual
    - Conceptually, we can’t really estimate this without Bayesian methods
    - We only have 2-3 observations per individual, so we estimate γ_i with 3 observations
    - This is no good! Law of Large Numbers can’t apply, and we’ll have garbage

Wages: Bayesian Improvements

- But let's say we assume that the γ_i come from a normal distribution with some mean and some variance.
- Now we can pin down all of the γ_i parameters without requiring exact knowledge of them.

Latent Dirichlet Allocation

- You might hear about Latent Dirichlet Allocation
- This is like the wage example

$$wage_i = \alpha + \beta educ_i + \gamma_i + \varepsilon_i$$

- But γ_i is drawn from a set of normal distributions.
- LDA is very commonly used for topic models
    - Topics are drawn from a particular distribution, and every document is a combination of these topics.
- It's also used when there are multiple groups: this is super likely what Nate Silver uses for his political models
    - Your data is a set of polls - sometimes a similar poll run multiple times, sometimes that same polling company polls in different states, etc.
    - Polls each have bias, but you can't easily estimate the amount of bias from a particular poll with just one result.

Georgia Tech

Bayesian Models: Editorial

- Bayesian models are conceptually appealing
  - While other models deal explicitly with overfitting through penalties (like adjusted R^2, or AIC in least squares, or explicit penalties on the sum of regression coefficients), Bayesian methods bring this into the model directly in the form of priors
- They are fit through simulation rather than maximization, which can have pros and cons
- Most uses of Bayesian methods explicitly use diffuse priors or priors estimated from the data, avoiding their main conceptual advantage
- Bayesian models have conceptual appeal, but in the past have been very difficult to estimate using MCMC
  - So there tended to be a divide between very technically savvy people using Bayesian models and everyone else

Lesson Summary

- Bayes’ rule is often computationally hard to calculate
- To compute Bayesian models, “Markov Chain Monte Carlo” is used
  - Bayesian models have conceptual appeal, but in the past have been very difficult to estimate using MCMC
- Latent Dirichlet Allocation is a specific Bayesian approach to unobserved group effects that is used for topic models