Bayesian Statistics and Stan

Nathaniel Lane and Dennis Moritz

November 19, 2020

1 Introduction

For our lecture, we presented on Bayesian statistics and how we can use Stan, a programming language, to analyze these models.

2 Bayesian Statistics

Bayesian statistics has two key differences from the more common frequentist statistics; probabilities are not thought of as being a long term tendency and parameters are represented as random variables rather than fixed points. There are several advantages to the Bayesian paradigm one of which is the concept of a credible interval, which is comparable to a confidence interval, but has a more intuitive interpretation due to the addition of parameters as a random variable.

The namesake for Bayesian statistics shared in common with Bayes Theorem and Bayes theorem is a key piece of Bayesian method. These methods center around defining Prior and Posterior distributions which describe the parameters of interest.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$
$$P(A|B) \propto P(B|A) \times P(A)$$
$$Posterior \propto Likelihood \times Prior$$

In general the task of sampling from the posterior distribution can be quite difficult, but is required to describe the resulting distribution and with it the results of the model fit. Monte Carlo simulations can be used to describe posterior distributions by taking random draws from all possibilities. Markov Chain Monte Carlo restricts this idea by chaining together the draws so in a way such that each draw is influenced by the draw directly before it. Hamiltonian Monte Carlo is a specific type of Markov Chain Monte Carlo where the chains are defined by using an analogy to kinetic energy from physics. Stan uses a particular implementation of Hamiltonian Monte Carlo, called NUTS, which dynamically optimizes the algorithm as it is being run.

3 Practical Example in Stan

Stan is a programming language that can be used to analyze Bayesian models. It is used in the context of a larger program written in another language such as R or Python; it cannot be used as an independent program. This larger program compiles the Stan model and passes data into and out of it.

A practical example for which we might want to use Stan is the "Eight schools problem." This is an example that's often used to introduce Bayesian statistics, and was first presented by Gelman et al. in their 1995 textbook "Bayesian Data Analysis." In this problem, eight schools conduct an experiment in which students are first given the PSAT, the practice SAT exam. They are then put through an SAT coaching course to help prepare them for the real SAT. After a while, the students take the real SAT. Each school presents the average improvement, along with a standard error for each measurement.

School	Average Improvement (y)	Standard Error (σ)
A	28	15
B	8	10
C	-3	16
D	7	11
E	-1	9
F	18	10
G	12	18

These are observations that can be used to predict the overall distribution of data. That distribution can in turn be used to predict how much improvement a school can expect from implementing these coaching programs. Two ways of modelling these larger distributions likely come to mind immediately. The first is to model each school independently, i.e., give each school its own distribution where y is the mean and σ is the standard deviation. The problem with this method is that the distributions have a lot of overlap, indicating that they may not be totally independent. The second approach is the exact opposite: suppose that these data points all come from one large distribution. When using this as a model, however, school A becomes an outlier at about 5 standard deviations away from the mean. Having one of eight observations be such an extreme outlier is an indication that the model is a poor one.

A third way to model this is to take a middle approach. We will suppose that each school has its own distribution, but that each one is governed by hidden variables that come from a universal distribution. Put another way, each school will have its own distribution where the means come from one universal distribution.

We use the following program to model this in Stan:

```
data {
  int<lower=0> n; // Number of schools
  real y[n]; // Effect
  real<lower=0> sigma[n]; // Standard errors of effects
}

parameters {
  real mu; // Overall mean
  real<lower=0> tau; // Inverse variance of the effect
  vector[n] eta; // Standardized school-level effects
}

model {
  eta ~ normal(mu, tau); // eta follows a normal distribution
  y ~ normal(eta, sigma); // y follows a normal distribution
}
```

Stan programs are divided into blocks. The first one in this example is the “data" block, which is used to define the input data. The first field, n, is the number of data points we are considering. The second field, y, is the observation for each school. The third field, *sigma*, is the standard error on those observations.

Conventional wisdom when it comes to Stan dictates that we read the rest of the program backwards. The “model" block defines how the data is modeled. The last line says that each y value is normally distributed about some value *eta* with a standard deviation of *eta*. This *eta* value is the one mentioned before that comes from a hidden distribution. We define that distribution in the first line of the “model" block. We say that each *eta* value comes from a normal distribution whose average is *mu* and whose standard deviation is *tau*.

The final block defines the hidden parameters. It is here that we declare *mu*, *tau*, and *eta*. Ultimately, the values listed in this block are the ones we want to predict.

4 Conclusion

Bayesian method can be both powerful and elastic in application. Until recently a major pitfall with Bayesian methods was the computational complexity of sampling from the posterior distribution. With the advent of Stan and the ever increasing availability of computing power, Bayesian methods are becoming more accessible and as such are becoming more widespread.