Gibbs Sampling

Ty Bruckner and Franco Salinas

Introduction

Using Bayesian statistics we can incorporate prior knowledge into the estimation of our unknown parameters. We incorporate our prior knowledge of the unknown parameter through a prior distribution. Then, we will update our beliefs about θ with observed data and we will end up with what we call a posterior distribution. We will use the posterior to estimate the parameters of interest. Gibbs sampling is a type of Markov Chain Monte Carlo sampling that approximates posterior distributions when we can't sample from the posterior distributions directly. This method is an alternative to Metropolis-Hastings which we have studied in Math Stats and in Bayesian Statistics and that will be explained in more detail later in the report. In a nutshell, Gibbs sampling is a method that samples from separate conditional distributions and is most useful when the joint posterior distribution is unknown or hard to sample from. Each event is dependent on the last event; and it is only dependent on the last event (like in a standard Markov Chain).

Motivation

This is an important topic to expand our knowledge and tool box on Bayesian statistics and estimation. We think the idea of incorporating prior knowledge (priors) in the estimation of a parameter is an interesting tool that can be useful in different careers including quantitative finance and bio statistics. Some people might be familiar with the Metropolis-Hastings algorithm to implement a simulation and to sample our posterior distributions. Gibbs is a suitable alternative to Metropolis-Hastings depending on the information one has available. The authors of this paper were interested in the applications of Monte Carlo simulations in finance and also wanted to explore topics from previous Bayesian Statistics classesn more deeply.

Background Knowledge

Monte Carlo https://towardsdatascience.com/an-overview-of-monte-carlo-methods-675384eb1694

Monte Carlo simulations randomly sample points within a region to approximate a distribution. The example above is a simple illusion of a uniform distribution for an estimate for π. This samples the proportion of points within the square region that fall within the circle's bounds. The proportion would be equal to $\frac{\pi}{4}$ since we are only interested in one fourth of the circle. As we sample more, our estimation for π gets closer to the actual distribution. This is due to the Central Limit Theorem. Monte Carlo simulations work well when the posterior distribution is easy to sample from. However, it is not always possible to sample from the posterior distribution, nor is it always efficient.

Markov Chains

Markov Chains are an example of a random walk. Random walks are a series of random moves through space in succession. Random walks use a combination of past events in the probability to determine the

next step. Markov Chains are a special case in which only the previous step/location is used to determine the probability distribution of the next step. This can be shown as $P(X_{n+1} = x|X_n = x_n)$ meaning the probability distribution of on move n+1 is only conditioned on the result of the previous move n. It is important to talk about the fact that Markov Chains are dependent on the previous move and are not an independent event.

[NEED A PICTURE]

Gibbs Sampling Overview

Gibbs Sampling is a specific type of MCMC sampling that is used when it is hard to sample from the joint pdf or pmf or when it is unknown. To perform Gibbs sampling you must know the conditional distributions of both variables.

Markov Chains Monte Carlo (MCMC)

MCMC is the application of Markov chains to simulate probability models. Two important characteristics are that MCMC samples aren't taken from the posterior pdf and that the samples aren't independent. The fact that the samples aren't independent reflects the "chain" fiture of the algorithm. For example in the $N - length$ MCMC sample (Markov chain) $\{\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}\}$, when constructing the chain $\theta^{(2)}$ is drawn from some model that depends upon $\theta^{(1)}$, $\theta^{(3)}$ is drawn from some model that depends on $\theta^{(2)}$ and so on.

We can say that the (i+1)st chain value $\theta^{(i+1)}$ has a conditional pdf $f(\theta^{(i+1)}|\theta^{(i)}, y)$ is drawn from a model that depends on data y and the previous chain value $\theta^{(i)}$. It's important to note that by the Markov property, $\theta^{(i+1)}$ depends on the preceding chain values only through $\theta^{(i)}$, the most recent value. The only information we need to simulate $\theta^{(i+1)}$ is the value of $\theta^{(i)}$. Therefore, each value can be sampled from a different model, and none of these models are the target posterior. The pdf from which a Markov Chain value is simulated is not equivalent to the posterior pdf.

$$f(\theta^{(i+1)}|\theta^{(i)}, y) \neq f(\theta^{(i+1)}|y)$$

We will conduct the MCMC simulation using the rstan package (Guo and Weber 2020). There are two essential steps to all rstan analyses, first we define the Bayesian model structure and then simulate the posterior.We will use a Beta-Binomial example.

STEP 1: DEFINE the model

Data: Y is the observed number of success trials. We specify that Y is between 10 and 0. Parameters: The model depends on π, therefore we must specify that π can be any real number from 0 to Model: We need to specify the model for the data and the model for the prior.

```
# STEP 1: DEFINE the model
bb_model <- "
  data {
    int<lower = 0, upper = 10> Y;
  }
  parameters {
    real<lower = 0, upper = 1> pi;
  }
  model {
    Y ~ binomial(10, pi);
```

```
    pi ~ beta(2, 2);
  }
"
```

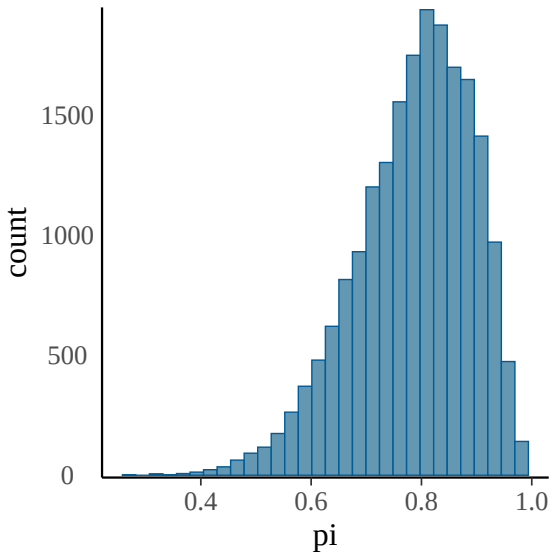
STEP 2: Simulate the posterior

We simulate the posterior using the stan() function. This function designs and runs an MCMC algorithm to produce an approximate sample from the Beta-Binomial posterior. The model code argument requires a string that defines the model. The data argument requires a list of observed data. The chains argument specifies how many parallel Markov Chains we are running. Since we are running four chains we will have four π values. The iter argument specifies the number of iterations or length for each chain. The first half of this iterations are thrown out as "burn in" samples. To keep our random results constant we utilize the seed argument within the stan() function.

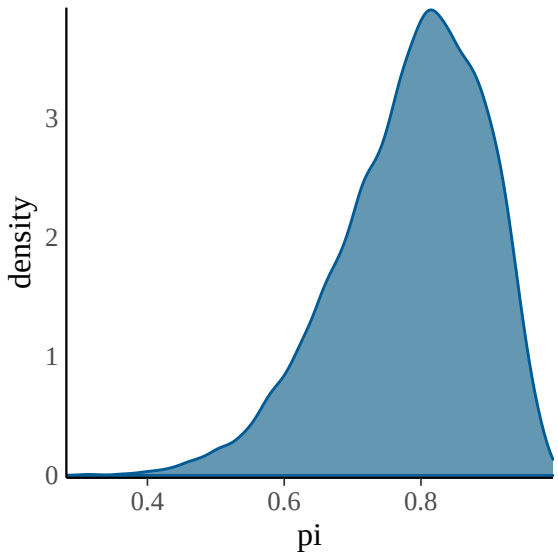
As you can see in Figure 1, when observing the distribution of the sampled π values we approximate the target Beta(11,3) posterior model of π. The target pdf is superimposed in black. (Alicia A. Johnson 2022)

```
# Histogram of the Markov chain values
mcmc_hist(bb_sim, pars = "pi") +
  yaxis_text(TRUE) +
  ylab("count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Density plot of the Markov chain values
mcmc_dens(bb_sim, pars = "pi") +
  yaxis_text(TRUE) +
  ylab("density")
```



Metropolis-Hastings algorithm

If we weren't able to recognize the posterior model of μ in a Normal-Normal model, we could approximate it using the MCMC simulation. Metropolis-Hastings algorithm helps automate the decision of what values of μ to sample and with what frequency. This algorithm iterates through a two step process. If we are in the location $\mu^{(i)} = \mu$ we select the next value to sample first by proposing a random location μ' and then we decide whether to stay at the current location or to stay at the current location $\mu^{(i+1)} = \mu$.

There are special cases of the Metropolis-Hastings that involve a different sampling decision criteria such as the Gibbs sampling, the Monte Carlo and the Metropolis algorithms. In this report we will be focusing on the Gibbs Sampling algorithm.

Gibbs Sampling

Example 1: Bernoulli Distribution:

Example from Youtube

We start with an example of two random variables A, B with a Bernoulli distribution. We now find their conditional distributions

$$P(A|B=0) \in P(A=1) = \frac{4}{5}, P(A=0) = \frac{1}{5}$$

$$P(A|B=1) \in P(A=1) = \frac{2}{5}, P(A=0) = \frac{3}{5}$$

$$P(B|A=0) \in P(B=1) = \frac{3}{4}, P(B=0) = \frac{1}{4}$$

$$P(B|A=1) \in P(B=1) = \frac{1}{3}, P(B=0) = \frac{2}{3}$$

1. Pick specific starting value of (x_0, y_0) Here we pick $(A_0), (B_0)$

2. Condition on B_0
3. Your distribution is now $P(A = 0) = 1/5$ and \$ P(A= 1) = 4/5\$
4. Randomly sample
5. Your random sample leads to $A = 1$
6. Condition on A_1
7. Your distribution is now $P(B = 0) = 2/3$ and \$ P(B= 1) = 1/3\$
8. Randomly sample
9. Your random sample leads to \$B=1 \$
10. Condition on B_1 Now the process repeats thousands of times until and each move is recorded. This algorithm then approximates well the true probability distribution after thousand of trials. More trials will lead to a better approximation.

Example 2: Normal Distribution

Now suppose we have data from a normal distribution where both the mean **and** variance are unknown. For convenience, we'll parameterize this model in terms of the *precision* $\gamma = \frac{1}{\sigma^2}$ instead of the variance σ^2.

$$Y \mid \mu, \gamma \sim N\left(\mu, \frac{1}{\gamma}\right)$$

Suppose we put the following *independent* priors on the mean μ and precision γ:

$$\mu \sim N(m, v)$$

$$\gamma \sim \text{Gamma}(a, b)$$

1. Write down the joint posterior distribution for μ, γ. Does this look like a recognizable probability distribution?

ANSWER: No, this is not a recognizable posterior:

$$\begin{aligned}
g(\mu, \gamma \mid y) &\propto f(y \mid \mu, \gamma) f(\mu, \gamma) \\
&= f(y \mid \mu, \gamma) f(\mu) f(\gamma), \text{ since } \mu, \gamma \text{ independent} \\
&= \left[(2\pi)^{-\frac{1}{2}} \gamma^{\frac{1}{2}} e^{-\frac{1}{2}\gamma(y-\mu)^2}\right] \left[(2\pi v)^{-\frac{1}{2}} e^{-\frac{1}{2v}(\mu-m)^2}\right] \left[\frac{b^a}{\Gamma(a)} \gamma^{a-1} e^{-b\gamma}\right] \\
&\propto \gamma^{\frac{1}{2}} e^{-\frac{1}{2}\gamma(y-\mu)^2} e^{-\frac{1}{2v}(\mu-m)^2} \gamma^{a-1} e^{-b\gamma} \\
&= \gamma^{\frac{1}{2}+a-1} e^{-\frac{1}{2}\gamma(y-\mu)^2 + -\frac{1}{2v}(\mu-m)^2 - b\gamma} \\
&= \gamma^{\frac{1}{2}+a-1} e^{-\frac{1}{2}\left[\gamma(y-\mu)^2 + \frac{1}{v}(\mu-m)^2 + 2b\gamma\right]} \\
&= \gamma^{\frac{1}{2}+a-1} e^{-\frac{1}{2}\left[\gamma y^2 - 2\mu y\gamma + \gamma\mu^2 + \mu^2/v - 2m\mu/v + m^2/v + 2b\gamma\right]} \\
&= \gamma^{\frac{1}{2}+a-1} e^{-\frac{1}{2}\left[\gamma(y^2+2b) - 2\mu(y\gamma + m/v) + \mu^2(\gamma + 1/v) + m^2/v\right]} \\
&\propto \gamma^{\frac{1}{2}+a-1} e^{-\frac{1}{2}\left[\gamma(y^2+2b) - 2\mu(y\gamma + \frac{m}{v}) + \mu^2(\frac{1}{v} + \gamma)\right]}
\end{aligned}$$

You should have answered "no" to Question 1, meaning that we can't use our usual techniques here to find Bayes estimators for μ or γ since we don't have a recognizable posterior distribution. Instead, we'll use a computational technique known as *Gibbs Sampling* to generate samples from this posterior distribution. Gibbs Sampling is particularly useful when we have more than one parameter, and the basic idea involves

reducing our problem to a series of calculations involving one parameter at a time. In order to perform Gibbs Sampling, we need to find the conditional distributions

$$g(\mu \mid y, \gamma) \propto f(y \mid \mu, \gamma) f(\mu)$$

$$g(\gamma \mid y, \mu) \propto f(y \mid \mu, \gamma) f(\gamma)$$

We will use these conditional distributions to sample from the joint posterior $g(\mu, \gamma \mid y)$ according to the following algorithm:

(1) Start with initial values $\mu^{(0)}, \gamma^{(0)}$.

(2) Sample $\mu^{(t+1)} \sim g(\mu \mid y, \gamma = \gamma^{(t)})$.

(3) Sample $\gamma^{(t+1)} \sim g(\gamma \mid y, \mu = \mu^{(t+1)})$.

(4) Repeat many times.

It turns out that the resulting $\mu^{(0)}, \mu^{(1)}, \ldots, \mu^{(N)}$ and $\gamma^{(0)}, \gamma^{(1)}, \ldots, \gamma^{(N)}$ are samples from the joint posterior distribution $g(\mu, \gamma \mid Y)$, and we can use these sampled values to estimate quantities such as the posterior mean of each parameter $\hat{E}(\mu \mid y) = \frac{1}{N} \sum_{i=1}^{N} \mu^{(i)}$, $\hat{E}(\gamma \mid y) = \frac{1}{N} \sum_{i=1}^{N} \gamma^{(i)}$. Note that in practice we typically remove the initial iterations, known as the "burn-in" period: e.g., $\hat{E}(\mu \mid y) = \frac{1}{N-B} \sum_{i=B}^{N} \mu^{(i)}$.

2. Show that the conditional distributions $g(\mu \mid y, \gamma), g(\gamma \mid y, \mu)$ are proportional to $f(y \mid \mu, \gamma) f(\mu), f(y \mid \mu, \gamma) f(\gamma)$, respectively, as stated above.

ANSWER:

$$\begin{aligned} g(\mu \mid y, \gamma) &= \frac{f(\mu, y, \gamma)}{f(y, \gamma)} \\ &\propto f(\mu, y, \gamma), \text{ since } f(y, \gamma) \text{ doesn't depend on } \mu \\ &= f(y \mid \mu, \gamma) f(\mu, \gamma) \\ &= f(y \mid \mu, \gamma) f(\mu) f(\gamma), \text{ since } \mu, \gamma \text{ independent} \\ &\propto f(y \mid \mu, \gamma) f(\mu), \text{ since} f(\gamma) \text{ doesn't depend on } \mu \end{aligned}$$

A similar argument can be used to show $g(\gamma \mid y, \mu) \propto f(y|\mu, \gamma) f(\gamma)$.

3. Use this result to show that $\mu \mid y, \gamma \sim N\left(\frac{y\gamma + \frac{m}{v}}{\gamma + \frac{1}{v}}, \left[\gamma + \frac{1}{v}\right]^{-1}\right)$ and $\gamma \mid y, \mu \sim \text{Gamma}\left(\frac{1}{2} + a, \frac{1}{2}(y - \mu)^2 + b\right)$.

ANSWER:

$$
\begin{aligned}
g(\mu \mid y, \gamma) &\propto f(y \mid \mu, \gamma) f(\mu) \\
&= \left[(2\pi)^{-\frac{1}{2}} \gamma^{\frac{1}{2}} e^{-\frac{1}{2}\gamma(y-\mu)^2}\right] \left[(2\pi v)^{-\frac{1}{2}} e^{-\frac{1}{2v}(\mu-m)^2}\right] \\
&\propto e^{-\frac{1}{2}\gamma(y-\mu)^2 - \frac{1}{2v}(\mu-m)^2} \\
&= e^{-\frac{1}{2}\gamma(y^2-2\mu y+\mu^2) - \frac{1}{2v}(\mu^2-2\mu m+m^2)} \\
&\propto e^{-\frac{1}{2}\gamma(-2\mu y+\mu^2) - \frac{1}{2v}(\mu^2-2\mu m)} \\
&= e^{-\frac{1}{2}\left[\mu^2(\gamma+\frac{1}{v}) - 2\mu(y\gamma+\frac{m}{v})\right]} \\
&= e^{-\frac{1}{2}(\gamma+\frac{1}{v})\left[\mu^2 - 2\mu\left(\frac{y\gamma+\frac{m}{v}}{\gamma+\frac{1}{v}}\right)\right]} \\
&\propto e^{-\frac{1}{2}(\gamma+\frac{1}{v})\left[\mu^2 - 2\mu\left(\frac{y\gamma+\frac{m}{v}}{\gamma+\frac{1}{v}}\right) + \left(\frac{y\gamma+\frac{m}{v}}{\gamma+\frac{1}{v}}\right)^2\right]} \\
&=\propto e^{-\frac{1}{2\left(\gamma+\frac{1}{v}\right)^{-1}}\left[\mu - \left(\frac{y\gamma+\frac{m}{v}}{\gamma+\frac{1}{v}}\right)\right]^2}
\end{aligned}
$$

$$
\implies \mu \mid y, \gamma \sim N\left(\frac{y\gamma + \frac{m}{v}}{\gamma + \frac{1}{v}}, \left[\gamma + \frac{1}{v}\right]^{-1}\right)
$$

$$
\begin{aligned}
g(\gamma \mid y, \mu) &\propto f(y \mid \mu, \gamma) f(\gamma) \\
&= \left[(2\pi)^{-\frac{1}{2}} \gamma^{\frac{1}{2}} e^{-\frac{1}{2}\gamma(y-\mu)^2}\right] \left[\frac{b^a}{\Gamma(a)} \gamma^{a-1} e^{-b\gamma}\right] \\
&\propto \gamma^{\frac{1}{2}} \gamma^{a-1} e^{-\frac{1}{2}\gamma(y-\mu)^2} e^{-b\gamma} \\
&= \gamma^{\frac{1}{2}+a-1} e^{-\frac{1}{2}\gamma(y-\mu)^2 - b\gamma} \\
&= \gamma^{\frac{1}{2}+a-1} e^{-\gamma\left(\frac{1}{2}(y-\mu)^2 + b\right)}
\end{aligned}
$$

$$
\implies \gamma \mid y, \mu \sim \text{Gamma}\left(\frac{1}{2} + a, \frac{1}{2}(y-\mu)^2 + b\right)
$$

4. Suppose that we choose the following hyperparameters for our prior distributions—$m = 0, v = 1, a = 1, b = 1$—and that we observe $y = 2$. Write code to implement this Gibbs Sampler.

ANSWER:

```
# set up priors
m <- 0
v <- 1
a <- 1
b <- 1

# set up data
y <- 2

# choose starting values by randomly sampling from our priors
# (this is just one possible way to choose starting values)
# (it's also useful to try out a few different starting values)
set.seed(1)
mu <- rnorm(1, mean = m, sd = sqrt(v))
gam <- rgamma(1, shape = a, rate = b)
```

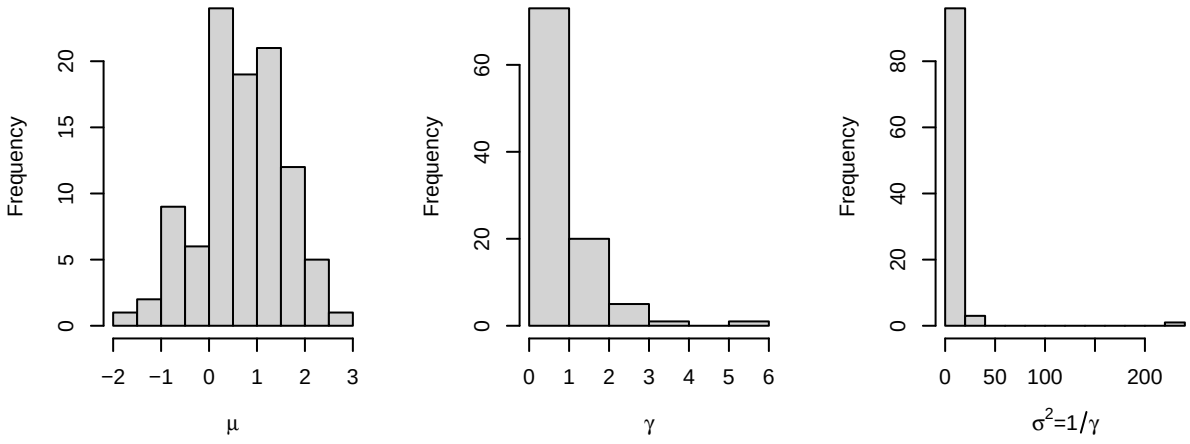
```r
# set up empty vectors to store samples
mus <- c()
gams <- c()

# store starting values in vectors of samples
mus[1] <- mu
gams[1] <- gam
```

```r
# choose number of iterations
# (we'll start with 100, but in practice you'd choose something much bigger)
N <- 100

# run through Gibbs Sampling for a total of N iterations
for(i in 2:N){
  # update mu
  m1 <- y*gam + m/v
  m2 <- gam + 1/v
  mu <- rnorm(n = 1, mean = (m1)/(m2), sd = sqrt(1/m2))

  # update gamma
  g1 <- 0.5 + a
  g2 <- 0.5*(y-mu)^2 + b
  gam <- rgamma(n = 1, shape = g1, rate = g2)

  # store new samples
  mus[i] <- mu
  gams[i] <- gam
}
```

5. Look at a histogram of your posterior samples for μ, γ and $\sigma^2 = \frac{1}{\gamma}$.

ANSWER:

```r
par(mfrow=c(1,3))
hist(mus, xlab = expression(mu), main = '')
hist(gams, xlab = expression(gamma), main = '')
hist(1/gams, xlab = expression(paste(sigma^2,'=',1/gamma)), main = '')
```



6. Estimate the posterior mean and median of μ.

ANSWER:

```
# posterior mean
mean(mus)
```

```
## [1] 0.6840056
```

```
# posterior median
median(mus)
```

```
## [1] 0.6525553
```

7. Find a 90% credible interval for μ, and estimate the probability that $\mu > 2$.

ANSWER:

```
# 90% credible interval
quantile(mus, probs = c(0.05, 0.95))
```

```
##         5%        95%
## -0.8862182  2.0094145
```

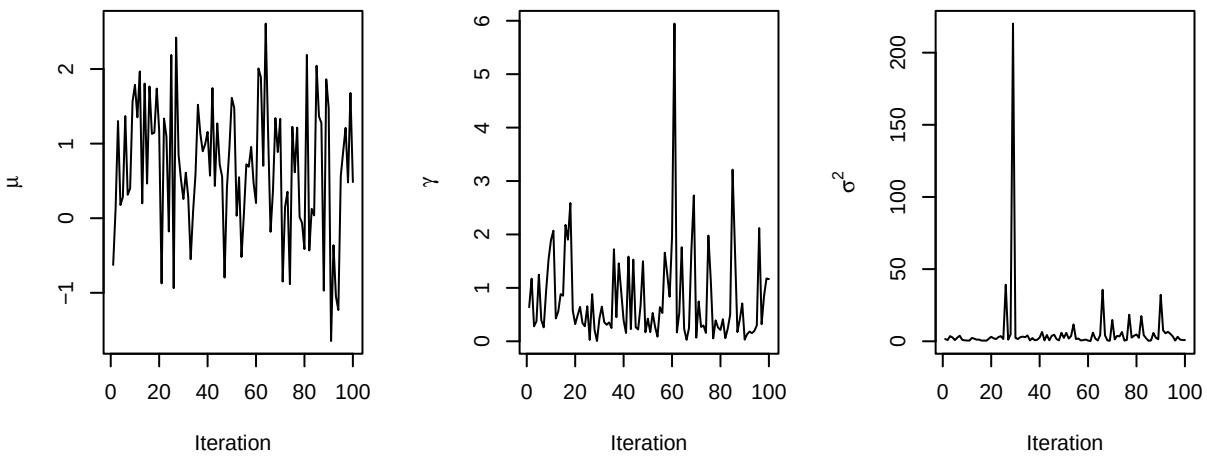
```
# P(mu > 2 | y)
mean(mus > 2)
```

```
## [1] 0.06
```

8. Create a *trace plot* showing the behavior of the samples over the N iterations.

ANSWER:

```
iterations <- 1:N

par(mfrow=c(1,3))
plot(mus ~ iterations, xlab = 'Iteration', ylab = expression(mu), type = 'l')
plot(gams ~ iterations, xlab = 'Iteration', ylab = expression(gamma), type = 'l')
plot(1/gams ~ iterations, xlab = 'Iteration', ylab = expression(sigma^2), type = 'l')
```



9. As mentioned above, in practice we usually pick a burn-in period of initial iterations to remove. This decision is often motivated by the fact that, depending on your choice of starting value, it may take awhile for your chain of samples to look like it is "mixing" well. Play around with your choice of starting value above to see if you can find situations in which a burn-in period might be helpful.

Real World Application

Gibbs sampling has been used in inference of population structure using multilocus genotype data (Jonathan K. Pritchard and Donnelly 2000). In other words, to infer the population of an individual using their genetic information. I will start defining some of the most important terms in the paper. A locus is the specific physical location of a gene or other DNA sequence on a chromosome, like a genetic street address. Genotype is the pair of alleles inherited for a particular gene, where gene is the functional unit of heredity.

Jonathan K. Pritchard, et. al used a Dirichlet distribution

$$D \sim Dir(\alpha) = \frac{1}{Beta(\alpha)} \prod_{i=1}^{J} \theta_i^{\alpha_i - 1}, where Beta(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{J} \alpha_i)} \alpha = (\alpha_1, ..., \alpha_2)$$

where D is a vector of J dimensions of the form $D = (\lambda_1, \lambda_2, ..., \lambda_J), \alpha_i > 0$ and θ belongs to the probability simplex where vectors are positive and the sum of the sum of their probability mass functions are always one. We use this distribution to model the allele frequencies $p = (p_1, p_2, ..., p_J)$ knowing that these frequencies sum to 1.

The authors use a Dirichlet distribution given that it is a commonly used conjugate prior. Conjugate priors make the process of estimating a posterior easier given that the posterior will be in the same probability distribution family.

I will introduce some of their model notation. The authors assumed that each population is modeled by a characteristic set of allele frequencies. X denotes the genotypes of the sampled individuals, Z denotes the individual's unknown populations of origin, and P denotes the unknown allele frequency in all populations. These vectors contain,

$(x_l^{(i,1)}, x_l^{(i,2)})$ = genotype of the ith individuals at the lth locus, where i= 1,2,...,N and l= 1,2,...,L;

$Z^{(i)}$ = population from which individual i originated

p_{klj} = frequency of allele j at locus l in population k, where k=1,2,...,K and $j = 1, 2, ..., J_l$

where J_l is the numner of distinct alleles observed at locus l, and these alleles are labeled 1,2,. . . ,J_l.

The authors used a Bayesian approach to decide how to perform inference for the quantities of interest. The authors specified model priors $Pr(Z)$ and $Pr(P)$ for both Z and P.

Having observed the genotypes, X, our knowledge of Z and P is given by the posterior distribution

$$Pr(Z, P \mid X) \propto Pr(Z)Pr(P)Pr(X \mid Z, P)$$

(1)

We can't compute this distribution exactly but we can obtain an approximate sample $(Z^{(1)}, P^{(1)}), (Z^{(2)}, P^{(2)}), ..., (Z^{(M)}, P^{(M)})$ from $Pr(Z, P \mid X)$ from $Pr(Z, P \mid X)$ using Gibbs Sampling. Inference for Z and P may be based on summary statistics obtained from this sample. This example will focus on a simpler model where each person is assumed to have originated in a single population.

Suppose we genotype N diploid individuals at L loci. Each individual is assumed to originate in one of K populations, each with it's own set set of allele frequencies.

We use the Dirichlet distribution to specify the probability of a particular set of allele frequencies p_{kl} for population k at locus l.

$$Pr(P)p_{kl} \sim D = (\lambda_1, \lambda_2, ..., \lambda_J)$$

(2)

at each locus within a population.

The authors assume that each genotype is an independent draw from the appropriate frequeny distribution and this specifies the probability distribution $Pr(X \mid Z, P)$. Given the population of origin of each individual, the genotypes are assumed to be generated by drawing alleles $x_l^{(i,a)}$ independently from the frequency distribution

$$P(X \mid Z, P) = Pr(x_l^{(i,a)} = j \mid Z, P) = p_z(i)_{lj}$$

(3) Where $p_z(i)_{lj}$ is the frequency of allele j at locus l in the population of origin of individual i.

Asuming tha tbefore observing th egenotypes we have no information about the population of origin of each individual and that the probability that individual i originated in population k is the same for all k,

$$P(Z) = Pr(z^{(i)} = k) = \frac{1}{K}$$

(4)

independently for all individuals.

Then the authors proceed to apply the Gibbs Sampling algorithm which can be described as follows.

Setting $\theta = (\theta_1, \theta_2) = (Z, P)$ and letting $\pi(Z, P) = Pr(Z, P \mid X)$ we can construct a Markov cchain with stationary distribution $Pr(Z, P \mid X)$ as follows:

Starting with initial values Zˆ(0) for Z (choosen randomly) we iterate over the following steps for m=1,2,. . . .

Step 1. Sample $P^{(m)}$ from $Pr(P \mid X, Z^{(m-1)})$ Step 1. Sample $Z^{(m)}$ from $Pr(Z \mid X, P^{(m)})$

In step 1 we are estimating allele frequencies for each population assuming that the population of origin of each individual is known. In step 2 we estiamte the population of origin of each individual, assuming that the population allele frequencies are known. For sufficiently large m and c, $(Z^{(m)}, P^{(m)}), (Z^{(m+c)}, P^{(m+c)}), (Z^{(m+2c)}, P^{(m+2c)}), ...$ will be approximately independent random samples from $Pr(Z, P \mid X)$

To be more specific, starting with initial values $\theta^{(0)} = (\theta_1^{(0)}, ..., \theta_r^{(0)})$, and we iterate the following steps for m= 1,2,. . .

Step 1. Sample $\theta_1^{(m)}$ from $\pi(\theta_1 \mid \theta_2^{(m-1)}, \theta_3^{(m-1)}, ..., \theta_r^{(m-1)})$

Step 2. Sample $\theta_2^{(m)}$ from $\pi(\theta_2 \mid \theta_1^{(m)}, \theta_3^{(m-1)}, ..., \theta_r^{(m-1)})$

Step r. Sample $\theta_r^{(m)}$ from $\pi(\theta_r \mid \theta_1^{(m)}, \theta_2^{(m)}, ..., \theta_r^{(m)})$

We can show that if $\theta^{(m-1)} \sim \pi(\theta)$, then, $\theta^{(m)} \sim \pi(\theta)$, and so $\pi(\theta)$ is the stationary distribution of this Markov chain.

References

Alicia A. Johnson, Mine Dogucu, Miles Q. Ott. 2022. *Bayes Rules!:an Introduction to Applied Bayesian Modeling.* Chapman; Hall/CRC.

Guo, Jonah Gabry, Jiqiang, and Sebastian Weber. 2020. "Rstan: R Interface to Stan." *J Aging Health.*

Jonathan K. Pritchard, Matthew Stephens, and Peter Donnelly. 2000. "Inference of Population Structure Using Multilocus Genotype Data."