

732A73: Bayesian Learning

Computer Lab 3

Oriol Garrobé, Dávid Hrabovszki

06 May 2020

Question 1. Normal model, mixture of normal model with semi-conjugate prior

The data `rainfall.dat` consist of daily records, from the beginning of 1948 to the end of 1983, of precipitation (rain or snow in units of 1/100 inch, and records of zero precipitation are excluded) at Snoqualmie Falls, Washington.

(a)

First we analyze the data using a normal model. We assume that the daily precipitation y_1, \dots, y_n values are independent normally distributed:

$$y_1, \dots, y_n | \mu, \sigma^2 \sim N(\mu, \sigma^2)$$

where μ, σ^2 are unknown. The prior distributions for these parameters are:

$$\mu \sim N(\mu_0, \tau_0^2)$$

$$\sigma^2 \sim \text{Inv} - \chi^2(\nu_0, \sigma_0^2)$$

(i)

Now we implement a Gibbs sampler that simulates from the joint posterior distribution:

$$p(\mu, \sigma^2 | y_1, \dots, y_n)$$

.

The full conditional posteriors:

$$\mu | \sigma^2, x \sim N(\mu_n, \tau_n^2)$$

$$\sigma^2 | \mu, x \sim \text{Inv} - \chi^2\left(\nu_n, \frac{\nu_0 \sigma_0^2 + \sum_{i=1}^n (x_i - \mu)^2}{n + \nu_0}\right)$$

where

$$\mu_n = w\bar{x} + (1 - w)\mu_0$$

,

$$w = \frac{\frac{n}{\sigma^2}}{\frac{n}{\sigma^2} + \frac{1}{\tau_0^2}}$$

$$\frac{1}{\tau_n^2} = \frac{n}{\sigma^2} + \frac{1}{\tau_0^2}$$

We choose the following prior values: $\mu_0 = 30$, from looking at the data, because we had no prior knowledge in the topic. To express our uncertainty about this, we set the variance of the prior mean to a high value: $\tau_0^2 = 100$. Similarly, we choose $\sigma_0^2 = 1547.103$, which is the variance of the data sample. But then we pick a small prior for the degrees of freedom, because the prior is non-informative: $\nu_0 = 4$.

(ii)

In this part we analyze the daily precipitation using our Gibbs sampler.

Figure 1 shows the trajectory of the Markov chain for sampling the mean and the cumulative mean of the mean statistic as the number of iterations grows. We ran the sampler for 1000 iterations, but it seems clear that it converges after a very short time. We ignore the first 10 draws, as this seems to be a burn-in period, where the draws are not inside the distribution. The observed mean from the data is 32.283, while the mean of the draws is 32.285, so this indicates a successful Gibbs sampling. The Markov chain has a highly oscillating shape and the fact that the cumulative mean also converges to 32.28 means that the chain converges. The daily precipitation then has an average of around 32.28, but there is a possibility that it is between 31 and 33.5.

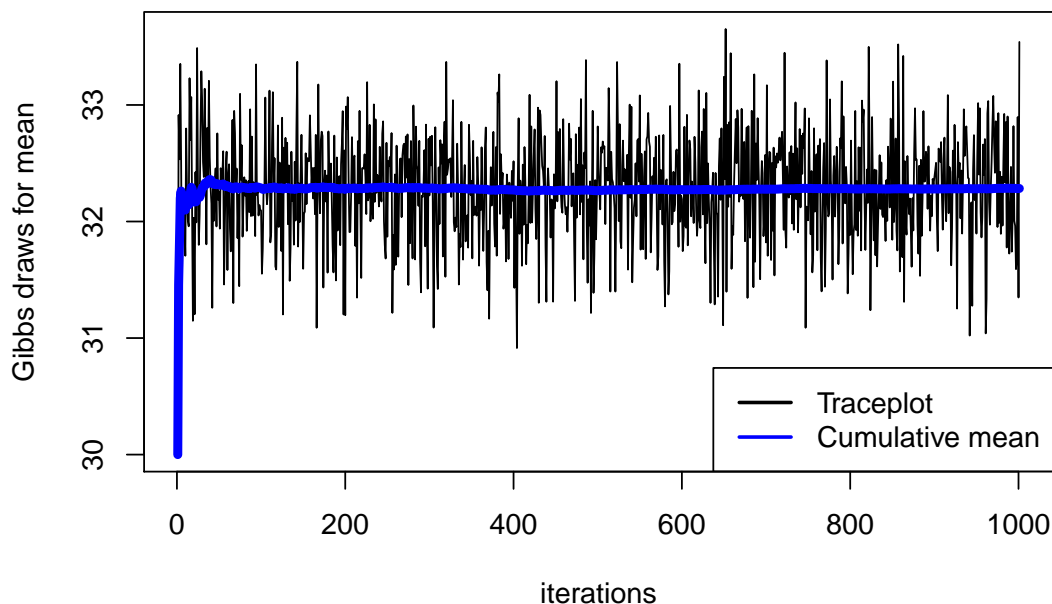


Figure 1: Markov chain of the sampled mean

Figure 2 shows the trajectory of the Markov chain for sampling the variance and the cumulative mean of the variance statistic as the number of iterations grows. We observe that the chain converges a bit more slowly than in the case of the mean, but it does so quite quickly nevertheless. Again, we ignore the burn in period (first 10 draws). The observed variance from the data is 1547.103, while the mean of the draws is 1546.152, so this indicates a successful Gibbs sampling. The Markov chain has a highly oscillating shape and the fact

that the cumulative mean of the draws also converges to 1546.149 means that the chain converges. The daily precipitation then has a variance of around 1546.152, but there is a possibility that it is between 1475 and 1625.

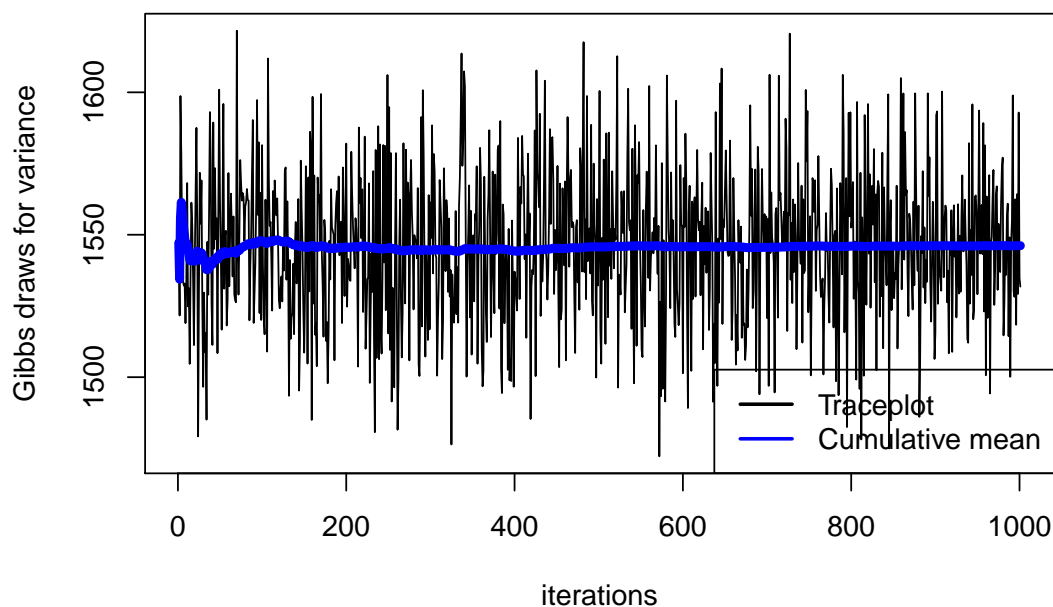


Figure 2: Markov chain of the sampled variance

Figures 3 and 4 show that the Gibbs samples for the mean and variance follow normal distributions, which is no surprise, since the priors were normal.

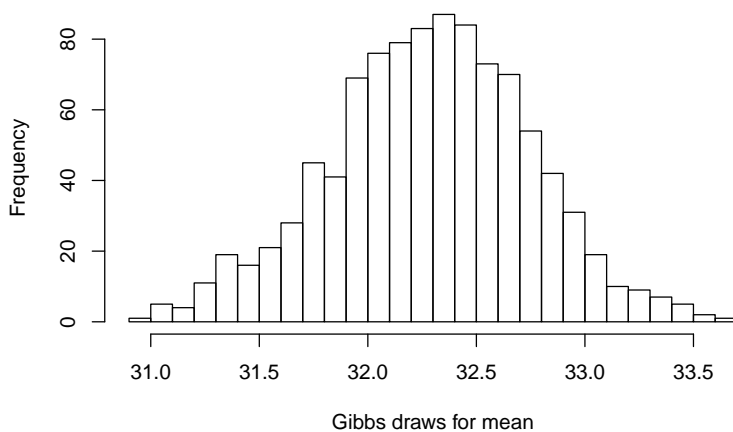


Figure 3: Histogram of the sampled mean

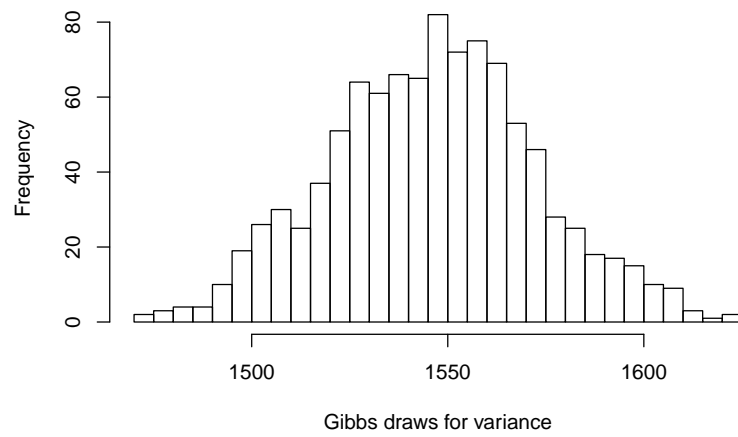


Figure 4: Histogram of the sampled variance

(b)

Slightly modified Mattias Villani's code.

(c)

Question 2. Metropolis Random Walk for Poisson regression

(a)

(b)

(c)

(d)

Appendix

```
knitr::opts_chunk$set(echo=FALSE, eval=TRUE)
# R version
RNGversion('3.5.1')

# libraries

#### 1.a
data = read.table('rainfall.dat', col.names = 'rainfall')

#### (i)

n = length(data$rainfall)
avg = mean(data$rainfall)

#prior setup
mu_0 = 30 #mu prior from prior knowledge, or looking at the data if there is no knowledge
tau2_0 = 100 #var(mu_0) high, because we have no prior knowledge
sigma2_0 = var(data$rainfall) # should use var(data$rainfall) for best guess
nu_0 = 4 #df, small, because we have no prior knowledge

# scaled-inverse chi-square simulator (from lab1 and from NormalMixtureGibbs.R)
rScaledInvChi2 <- function(n, df, scale){
  return((df*scale)/rchisq(n,df=df))
}

# Gibbs sampling
steps = 1000
gibbsDraws <- matrix(0,steps+1,2) #1st column: mu, 2nd column: sigma2
gibbsDraws[1,1] = mu_0
gibbsDraws[1,2] = sigma2_0

# Seed
set.seed(1234567890)
for (i in 1:steps) {
  #simulate mu from conditional posterior (normal)
  w = (n/gibbsDraws[i,2])/(n/gibbsDraws[i,2] + 1/tau2_0)
  mu_n = w * avg + (1-w) * mu_0
  tau2_n = 1 / (n/gibbsDraws[i,2] + 1/tau2_0)

  #update mu
  gibbsDraws[i+1,1] = rnorm(1, mu_n, sqrt(tau2_n))

  #simulate sigma2 from conditional posterior (inv-chisq)
  nu_n = nu_0 + n
  scale = (nu_0 * sigma2_0 + sum((data$rainfall - gibbsDraws[i+1,1])^2)) / (n + nu_0)

  #update sigma2
  gibbsDraws[i+1,2] = rScaledInvChi2(1, nu_n, scale)
}

#### (ii)
#Plot Markov chains and cumulative mean plots
```

```

cum_mean = numeric(steps + 1)
cum_var = numeric(steps + 1)
for (i in 1:(steps+1)) {
  cum_mean[i] = mean(gibbsDraws[1:i,1])
  cum_var[i] = mean(gibbsDraws[1:i,2])
}
plot(1:(steps+1), gibbsDraws[,1], type = 'l', xlab = 'iterations', ylab = 'Gibbs draws for mean')
legend(x='bottomright', legend=c('Traceplot', 'Cumulative mean'), col = c('black', 'blue'), lwd = 2)
lines(1:(steps+1), cum_mean, col = 'blue', lwd = 5)
plot(1:(steps+1), gibbsDraws[,2], type = 'l', xlab = 'iterations', ylab = 'Gibbs draws for variance')
legend(x='bottomright', legend=c('Traceplot', 'Cumulative mean'), col = c('black', 'blue'), lwd = 2)
lines(1:(steps+1), cum_var, col = 'blue', lwd = 5)

# result: mean of all posterior draws
burn_in_mu = 10
burn_in_sigma2 = 10
mu_gibbs = mean(gibbsDraws[burn_in_mu:length(gibbsDraws[,1]),1]) #32.28486
sigma2_gibbs = mean(gibbsDraws[burn_in_sigma2:length(gibbsDraws[,2]),2]) #1546.152

hist(gibbsDraws[burn_in_mu:length(gibbsDraws[,1]),1], breaks = 30, main = '', xlab = 'Gibbs draws for m
hist(gibbsDraws[burn_in_sigma2:length(gibbsDraws[,2]),2], breaks = 40, main = '', xlab = 'Gibbs draws f

```