STAT 656: Bayesian Data Analysis
Fall 2024
Homework 3

Juanwu Lu[*]

Gibbs Sampling for Normal Hierarchical Models

In Lecture 7, we modeled the NBA data with a normal hierarchical model with known variances as follows:

$$p(\mu, \tau^2) = p(\mu|\tau^2)p(\tau^2) \propto p(\tau^2)$$

$$\theta_j|\mu, \tau^2 \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mu, \tau^2), j = 1, \ldots, J$$

$$y_{ij}|\theta, \sigma^2, \mu, \tau^2 \overset{\perp\!\!\!\perp}{\sim} \mathcal{N}(\theta_j, \sigma_j^2), \quad i = 1, \ldots, n_j.$$

Now we will extend this to allow the variances σ_j^2 to be unknown as well. We will consider two models:

1. **Unknown but identical variances**: $\sigma_1^2 = \ldots = \sigma_J^2 = \sigma^2$, with $\sigma^2 \sim p_v$
2. **Unknown and independent variances**: $\sigma_1^2, \ldots, \sigma_J^2 \overset{\text{i.i.d}}{\sim} p_v$.

For both models, set $p_v(\sigma^2) \propto 1/\sigma$. Your task is to implement a Gibbs sampler to simulate all latent variables given the data from `nba_data.csv`. The Gibbs sampler involves the following steps for each iteration:

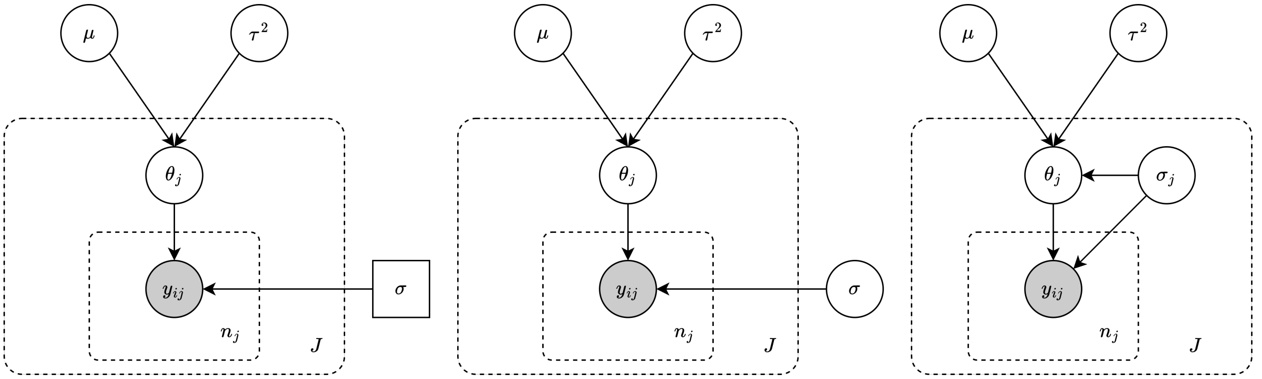
- (*For both models*) Sample from $\mu, \tau^2|\theta_1, \ldots, \theta_J, \sigma_1^2, \ldots, \sigma_J^2, Y$, and then
- (*For model 1*) Sample from $\theta_1, \ldots, \theta_J|\sigma^2, \mu, \tau^2, Y$, and then $\sigma^2|\theta_1, \ldots, \theta_J, \mu, \tau^2, Y$
- (*For model 2*) Sample from $(\theta_1, \sigma_1^2), \ldots, (\theta_J, \sigma_J^2)|\mu, \tau^2, Y$.

For full credit, for each model, you need to:

1. (15 points) Write down the form of the conditional distributions above.
2. (25 points) Write down R code to implement the Gibbs samplers.
3. (30 points) Run the two Gibbs samplers on the NBA dataset and summarize the results. Specifically, for μ, τ^2 and a few θ_j and σ_j, plot the MCMC traceplots, and diagnose mixing. Also plot the corresponding posterior distributions.
4. (30 points) Comment on how the posterior distributions differ from each other. Use each model to make predictions on the NBA games from the rest of the season (following code from Lecture 7), and comment on which model (or the model with known variances from Lecture 7) performs best.

[*] College of Engineering, Purdue University, West Lafayette, IN, USA

Solution:



(a) Model from Lecture 7 (b) Model 1 from Problem (c) Model 2 from Problem

Figure 1: Probabilistic graphical model for the NBA data.

First, we load in the data. Instead of recording the score for each game $y_{ij}, i = 1, \ldots, n_j$, we can see from the data that it only gives the average score $\bar{y}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} y_{ij}$ for each team $j = 1, \ldots, J$.

```r
if (file.exists("data/NBA_data.csv")) {
  df <- read.csv("data/NBA_data.csv")[, -1]
} else {
  stop("FileNotFound: data file not found at 'data/NBA_data.csv'.")
}
head(df)
```

```
##   number_games_played    Y_bar sample_sd number_remaining_games
## 1                  10 108.0000  5.868939                     72
## 2                  10 107.2000  8.841820                     72
## 3                   8 106.8750  6.957781                     74
## 4                  10 104.8000 11.545080                     72
## 5                  10 103.8000 13.870830                     72
## 6                   9 103.4444 14.292580                     73
##   average_remaining_games        team
## 1               110.50556     Phoenix
## 2               100.68556        Utah
## 3               105.18514      Denver
## 4               106.78167 GoldenState
## 5                94.70028      Boston
## 6               104.45041 Washington
```

Following the lecture notes, we have

$$\bar{y}_j|\theta j \sim \mathcal{N}(\theta_j, \frac{1}{n_j}\sigma_j^2),$$

and since $\theta_j \sim \mathcal{N}(\mu, \tau^2)$, we have

$$\bar{y}_j \sim \mathcal{N}(\mu, \tau^2 + \frac{\sigma_j^2}{n_j}).$$

Therefore, the log joint posterior of (μ, τ^2) is given by

$$\log p(\mu, \tau^2|\sigma_1, \dots, \sigma_J, Y) = \log p(\mu, \tau^2) - \frac{1}{2}\sum_{j=1}^{J} \log\left(\tau^2 + \frac{\sigma_j^2}{n_j}\right) - \frac{1}{2}\sum_{j=1}^{J}\left\{\left(\tau^2 + \frac{\sigma_j^2}{n_j}\right)^{-1} (\bar{y}_j - \mu)^2\right\} + \text{const}, \tag{*}$$

where const is constant irrelavant of μ and τ^2. The same notation is used in the following section. If we denote $\sigma_\mu^{-2} = \sum\limits_{j=1}^{J}\left(\tau^2 + \frac{\sigma_j^2}{n_j}\right)^{-1}$, then we can obtain the conditional posterior distribution of μ by separating out the terms involving μ in equation (*), which is given by

$$\log p(\mu|\tau^2, \sigma_1, \dots, \sigma_J, Y) = -\frac{1}{2}\left\{\frac{\mu^2}{\sigma_\mu^2} - 2\sum_{j=1}^{J}\left(\tau^2 + \frac{\sigma_j^2}{n_j}\right)^{-1} \bar{y}_j \cdot \mu + \text{const}\right\}$$

$$\Rightarrow \quad \mu|\tau^2, \sigma_1, \dots, \sigma_J, Y \sim \mathcal{N}\left(\hat{\mu}, \sigma_\mu^2\right), \quad \hat{\mu} = \sum_{j=1}^{J}\left(\tau^2 + \frac{\sigma_j^2}{n_j}\right)^{-1} \bar{y}_j \cdot \sigma_\mu^2.$$

Hence, the log conditional posterior of τ^2 is given by the following equation:

$$\begin{aligned}\log p(\tau^2|\mu, \sigma_1, \dots, \sigma_J, Y) &= \log p(\mu, \tau^2|Y) - \log p(\mu|\tau^2, \sigma_1, \dots, \sigma_J, Y) \\ &= \log p(\mu, \tau^2) - \frac{1}{2}\sum_{j=1}^{J} \log\left(\tau^2 + \frac{\sigma_j^2}{n_j}\right) - \frac{1}{2}\log \sigma_\mu - \frac{1}{2}\sum_{j=1}^{J}\left\{\left(\tau^2 + \frac{\sigma_j^2}{n_j}\right)^{-1} (\bar{y}_j - \hat{\mu})^2\right\}\end{aligned}$$

For fair comparison with the model from the lecture, we pose the same prior $p(\mu, \tau^2) = \tau^{-1}$, the function for evaluating the log conditional posterior distribution of τ is implemented as follows:

```r
log_posterior_tau_sq <- function(mu, tau_sq, sigma_sq, n, y_bar) {
  precision_mu <- sum(1 / (tau_sq + sigma_sq / n))
  mu_hat <- sum(y_bar / (tau_sq + sigma_sq / n)) / precision_mu

  # NOTE: uses prior p(mu, tau^2) = tau^{-1}
  log_prior <- -0.5 * log(tau_sq)
  log_determinant <- -0.5 * (
    sum(log(tau_sq + sigma_sq / n))
    + log(precision_mu)
  )
  log_mahalanobis <- -0.5 * sum(
    (y_bar - mu_hat) ^ 2 / (tau_sq + sigma_sq / n)
  )

  return(log_prior + log_determinant + log_mahalanobis)
}
```

The log conditional posterior distribution of μ is implemented as follows:

```r
log_posterior_mu <- function(mu, tau_sq, sigma_sq, n, y_bar) {
  precision_mu <- sum(1 / (tau_sq + sigma_sq / n))
  mu_hat <- sum(y_bar / (tau_sq + sigma_sq / n)) / precision_mu
  log_prob <- -0.5 * (
    log(2 * pi) - log(precision_mu) + precision_mu * (mu - mu_hat) ^ 2
  )

  return(log_prob)
}
```

For model 1, as shown by Figure 1(b), we need the conditional distribution $p(\theta_1,\dots,\theta_J|\sigma^2,\mu,\tau^2,Y)$ and $p(\sigma^2|\theta_1,\dots,\theta_J,\mu,\tau^2,Y)$. The log joint conditional posterior distribution of $\theta_1,\dots,\theta_J$ is given by

$$\begin{aligned}\log p(\theta_1,\dots,\theta_J|\sigma^2,\mu,\tau^2,Y) &= \log p(\mu,\tau^2) + \log p_v + \sum_{j=1}^{J} \log\left(p(\theta_j|\mu,\tau^2)p(\bar{y}_j|\theta_j,\sigma_j^2)\right)\\ &= -\frac{1}{2}\sum_{j=1}^{J}\left[\left(\tau^{-2}+\left(\frac{n_j}{\sigma_j^2}\right)\right)\theta_j^2 - 2\left(\tau^{-2}\mu+\left(\frac{n_j}{\sigma_j^2}\right)\bar{y}_j\right)\theta_j\right] + \text{const},\end{aligned}$$

which is similar to the formula for conditional posterior distribution of μ. Considering $\theta_1,\dots,\theta_J$ are independent and identically distributed, the conditional posterior distribution of θ_j is given by

$$\theta_j \sim \mathcal{N}\left(\frac{\tau^{-2}\mu+\left(\frac{n_j}{\sigma_j^2}\right)\bar{y}_j}{\tau^{-2}+\left(\frac{n_j}{\sigma_j^2}\right)}, \left(\tau^{-2}+\left(\frac{n_j}{\sigma_j^2}\right)\right)^{-1}\right)$$

Meanwhile, the log conditional posterior distribution of σ^2 is given by

$$\begin{aligned}\log p(\sigma^2|\theta_1,\dots,\theta_J,\mu,\tau^2,Y) &= \log p_v + \sum_{j=1}^{J}\log\left(p(\bar{y}_j|\theta_j,\sigma_j^2)\right) + \text{const}\\ &= -\log(\sigma) - \frac{1}{2}\sum_{j=1}^{J}\left\{\log\left(\frac{\sigma_j^2}{n_j}\right) + \frac{(\bar{y}_j-\theta_j)^2}{\sigma_j^2/n_j}\right\} + \text{const}.\end{aligned}$$

For model 2, as shown by Figure 1(c), the variances σ_j for each team are now independent and identically distributed with $\sigma_1^2,\dots,\sigma_J^2 \overset{\text{i.i.d}}{\sim} p_v$. To leverage the conjugate structure, the conditional distribution of θ_j is then given by $\theta_j \sim \mathcal{N}(\mu,\tau^2\sigma_j^2)$. As a result, the log joint conditional posterior distribution of $\left(\theta_j,\sigma_j^2\right)$ is given by

$$\begin{aligned}\log p(\theta,\sigma^2 \mid \mu,\tau^2,Y) &= \sum_{j=1}^{J}\log\left(p_v(\sigma_j)\cdot p(\theta_j \mid \mu,\tau^2\sigma_j^2)\cdot p(\bar{y}_j \mid \theta_j,\sigma_j^2)\right)\\ &= -\sum_{j=1}^{J}\left(\log(\sigma_j) + \frac{1}{2}\log(\tau^2\sigma^2) + \frac{1}{2}\frac{(\theta_j-\mu)^2}{\tau^2\sigma_j^2} + \frac{1}{2}\log(\sigma_j^2) + \frac{1}{2}\frac{(\bar{y}_j-\theta_j)^2}{\sigma_j^2} + \text{const}\right).\end{aligned}$$

The Stroop Effect

Consider a psychological task where subjects are presented with a word at the center of a computer screen (`'red'`, `'blue'`, or `'green'`). Further, the word is colored either red, blue or green. In some trials, the word matches the color of the text ('congruent' condition); otherwise they do not match ('incongruent' condition, *e.g., the word is `'red'` but it is colored blue*). Subjects are told to focus only on the color that the word is written in, and press 1 if the color is red, 2 if it is blue, and 3 if it is green. In each case, the experimenter measures the reaction time (*i.e., how long it takes them to press the correct button*). The Stroop effect is a robust effect in psychology where the reaction time in the incongruent condition is on average **larger** than in the congruent condition.

Your task is to use the data in `stroop_data.csv` to verify if this is the case. The data measures multiple reactions times of different subjects in congruent and incongruent settings. You will model this with a hierarchical Bayeisan model, with the goal of determining:

- how much longer reaction times are for each color in incongruent vs congruent cases, and whether this difference is significant.
- how different the effect is for each color, and whether these differences are significant.
- how different individuals in the study are from each other.

Your model should account for the fact that

- each response of each individual involves random fluctuations
- reaction times and the manitude of the Stroop effect can be different for different individuals
- reaction times and the magnitude of the Stroop effect can be different for different colors (*e.g., it might be smaller for red where you have to press* 1 *vs others*)

Your hierarchical model should allow statistical sharing among individuals and possibly among different colors. Justify your model and prior choices, implement it in `Stan` and sicusss your findings, being sure to include visualizations and predictive checks of model fit. You must present your final results in a form that can be readily understood by a general audience.

Solution:

```r
# Read the data
if (file.exists("data/stroop_dataset.csv")) {
  data <- read.csv("data/stroop_dataset.csv", header = TRUE, row.names = 1)
} else {
  stop("FileNotFound: data file not found at 'data/stroop_dataset.csv'.")
}
head(data)
```

First we take a look at the distribution of reaction time over all subjects and trials. The following histogram shows that reaction time has a support of all positive real numbers, with a right-skewed long-tailed distribution. Therefore, we can model reaction time with a log-normal distribution.

```r
ggplot(data, aes(x = RT)) +
  geom_histogram(aes(y = after_stat(density)), bins = 100, fill = "lightblue") +
  labs(title = "Histogram of Reaction Time", x = "Reaction Time (ms)", y = "Frequency") +
  geom_density()
```

According to the problem description, we have three colors and three words. Without loss of generality, we can represent them as discrete random variables. Then we obtain a binary indicator to represent the congruence of the word and the color. In this formulation, we introduce means for congruent and incongruent conditions, where both of them follow a log-normal distribution. In the distribution of the reaction time, we activate either one of them based on the binary congruence indicator. Finally, we pose standard Gamma priors on the standard deviations in the log-normal distribution of reaction time.

```
hierarchical_model_code <- "
  data {
      int<lower=0> n;                       // number of responses
      int<lower=0> k;                       // number of subjects
      int<lower=0> c;                       // number of colors
      real<lower=0> pr_std;                 // Prior standard deviations
      array[n] int<lower=1, upper=k> subjs; // Subject ID for each response
      array[n] int<lower=1, upper=c> color; // Color of the word
      array[n] int<lower=1, upper=c> word;  // Word in each trial
      array[n] real y;                      // Reaction time of responses
  }

  parameters {
      vector<lower=0>[k] std;       // Emission standard deviation
      matrix[k, c] mu_congruent;    // Mean of reaction time for congruent
      matrix[k, c] mu_incongruent;  // Mean of reaction time for incongruent
  }

  transformed parameters {
      // Calculate the binary indicator for congruence
      vector[n] congruent;
      for (i in 1:n) {
          congruent[i] = (color[i] == word[i]);
      }

      // Calculate the emission mean as a Bernoulli mixture
      vector[n] mu;
      for (i in 1:n) {
          mu[i] = (
              mu_congruent[subjs[i], color[i]] * congruent[i]
              + mu_incongruent[subjs[i], color[i]] * (1 - congruent[i])
          );
      }
  }

  model {
      // Sample subject parameters from priors
      for (i in 1:k) {
          for (j in 1:c) {
              mu_congruent[i, j] ~ normal(0, pr_std);
              mu_incongruent[i, j] ~ normal(1, pr_std);
          }
      }
      std ~ gamma(1, 1);
      for (i in 1:n) {
          y[i] ~ lognormal(mu[i], std[subjs[i]]);
      }
  }

  generated quantities {
      array[n] real y_hat;
      for (i in 1:n) {
          y_hat[i] = lognormal_rng(mu[i], std[subjs[i]]);
      }
  }
"
hierarchical_model <- stan_model(
  model_name = "stroop_hierarchical",
  model_code = hierarchical_model_code
)
```

We sample 5, 000 samples from the hierarchical model using the data as follows. In the samples, we obtain two matrix representing the logarithm of mean of reaction time for each subject for each color in congruent and incongruent conditions, respectively. The two matrix is used for answering the three questions in the problem description.

```
n <- dim(data)[1]
k <- length(unique(data$subj))
c <- length(unique(data$color))
subjs <- data$subj
color <- as.numeric(factor(data$color))
word <- as.numeric(factor(data$word))
y <- data$RT
hm_data <- list(
  n = n,
  k = k,
  c = c,
  pr_std = 10.0,
  subjs = subjs,
  color = color,
  word = word,
  y = y
)
nfit <- sampling(
  hierarchical_model,
  data = hm_data,
  chains = 1,
  iter = 10000,
  warmup = 5000,
  show_message = FALSE,
  cores = 16,
  seed = 42,
)
samples <- as.data.frame(nfit)
```

The following visualization shows the distribution of difference in the posterior mean of reaction time between incongruent and congruent conditions for each color averaged by subjects. The plot shows that the Stroop effect is significant for all colors. Among the three colors, the Stroop effect is most significant if the words are green, followed by red and blue. Difference in effect between red and blue is not significant.

```
mu_c <- exp(samples[, grep(pattern="^mu_congruent", x = colnames(samples))])
mu_inc <- exp(samples[, grep(pattern="^mu_incongruent", x = colnames(samples))])
diff <- data.matrix(mu_inc) - data.matrix(mu_c)

color_diff <- array(0, dim = c(dim(samples)[1], c))
colnames(color_diff) <- levels(factor(data$color))
for (i in 1:c) {
  color_diff[, i] <- apply(
      diff[, (k * (i-1) + 1):(k * i)],
      c(1),
      mean
  )
}
mcmc_areas(
  x = color_diff,
  par = levels(factor(data$color)),
  prob = 0.68,
) + labs(
  title = "Average Difference in Posterior Mean of Reaction Time",
  subtitle = "with median and 68% intervals for each color"
)
```

The following visualization shows the distribution of difference in the posterior mean of reaction time between incongruent and congruent conditions for each subject averaged by colors. Based on the result, the mean difference varies significantly among subjects, where we observed the Stroop effect in most of their posterior distributions, with a few exceptions (*e.g., subject* 15*, subject* 27*, subject* 41*, etc.*).

```
subjs_diff <- array(0, dim = c(dim(samples)[1], k))
colnames(subjs_diff) <- paste("", unique(data$subj))
for (i in 1:k) {
  subjs_diff[, i] <- apply(
    diff[, seq(1, k * c, by = k) + (i - 1)],
    c(1),
    mean
  )
}
mcmc_intervals(
  x = subjs_diff,
  par = colnames(subjs_diff),
  prob = 0.68,
) + labs(
  title = "Average Difference in Posterior Mean of Reaction Time",
  subtitle = "with median and 68% intervals for each subject"
)
```