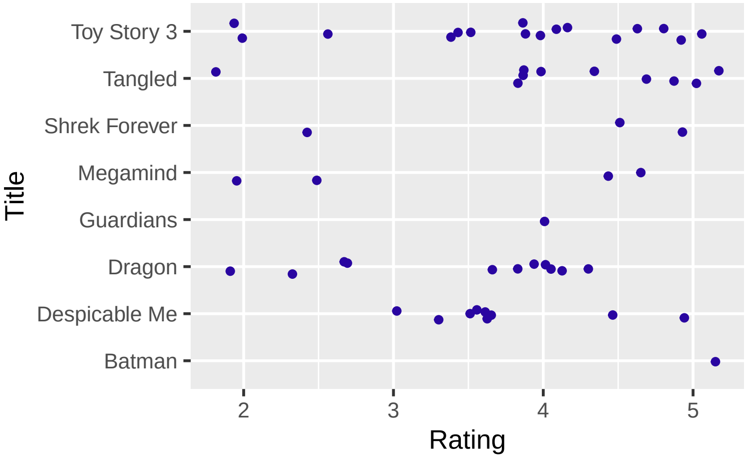
Chapter 10.2 Hierarchical Normal Modeling

Jim Albert and Monika Hu

Chapter 10 Bayesian Hierarchical Modeling

Example: ratings of animation movies

- MovieLens
- A sample: eight different animation movies released in 2010; 55 ratings



Example: ratings of animation movies cont'd

Movie Title	Mean	SD	N
Batman: Under the Red Hood	5.00		1
Despicable Me	3.72	0.62	9
How to Train Your Dragon	3.41	0.86	11
Legend of the Guardians	4.00		1
Megamind	3.38	1.31	4
Shrek Forever After	4.00	1.32	3
Tangled	4.20	0.89	10
Toy Story 3	3.81	0.96	16

- variability in the sample sizes
- to improve the estimate of mean rating by using rating information from similar movies

A hierarchical normal model with random σ

- Y_{ij} denotes the i-th rating for the j-th movie title
- Sampling model: normal
- Assume a movie-specific mean μ_j and a common and random σ

A hierarchical normal model with random σ

- Y_{ij} denotes the i-th rating for the j-th movie title
- Sampling model: normal
- Assume a movie-specific mean μ_j and a common and random σ
- Sampling, for $j = 1, \cdots, 8$ and $i = 1, \cdots, n_j$:

$$Y_{ij} \mid \mu_j, \sigma \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu_j, \sigma) \tag{1}$$

- Prior for μ_j, $j = 1, \cdots, 8$:

$$\mu_j \mid \mu, \tau \sim \mathrm{Normal}(\mu, \tau) \tag{2}$$

Pooling information across movies

$$\mu_j \mid \mu, \tau \sim \mathrm{Normal}(\mu, \tau)$$

- Large value of τ:
  - the μ_j's are very different from each other *a priori*
  - modest pooling of the eight sets of ratings
- Small value of τ:
  - the μ_j's are very similar to each other *a priori*
  - large pooling of the eight sets of ratings

Pooling information across movies

$$\mu_j \mid \mu, \tau \sim \mathrm{Normal}(\mu, \tau)$$

- Large value of τ:
  - the μ_j's are very different from each other *a priori*
  - modest pooling of the eight sets of ratings
- Small value of τ:
  - the μ_j's are very similar to each other *a priori*
  - large pooling of the eight sets of ratings

- Simultaneously estimate:
  - a mean for each movie (the μ_j's)
  - the variationamong the movies by the parameter τ

Hyperparameters

$$\mu_j \mid \mu, \tau \sim \mathrm{Normal}(\mu, \tau)$$

- μ and τ: hyperparameters
- Treat as random (we are unsure about the degree of pooling)
- e.g. weakly informative prior distribution

Complete model specification

- Sampling: for $j = 1, \cdots, 8$ and $i = 1, \cdots, n_j$:

$$Y_{ij} \mid \mu_j, \sigma \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu_j, \sigma) \tag{3}$$

- Prior for μ_j, Stage 1: μ_j, $j = 1, \cdots, 8$:

$$\mu_j \mid \mu, \tau \sim \mathrm{Normal}(\mu, \tau) \tag{4}$$

- Prior for μ_j, Stage 2:

$$\mu, \tau \sim \pi(\mu, \tau) \tag{5}$$

- Prior for σ:

$$1/\sigma^2 \mid a_\sigma, b_\sigma \sim \mathrm{Gamma}(a_\sigma, b_\sigma) \tag{6}$$

Disucssion on sharing

- Two-stage prior for $\{\mu_j\}$ vs shared σ
- Differences?

Graphical representation of the hierarchical model

$$
\begin{aligned}
\mu, \tau &\sim \pi(\mu, \tau) \\
\mu_j &\sim \text{Normal}(\mu, \tau) \\
Y_{ij} &\sim \text{Normal}(\mu_j, \sigma) \\
1/\sigma^2 &\sim \text{Gamma}(a_\sigma, b_\sigma)
\end{aligned}
$$

$$
\begin{tikzcd}
 & \mu, \tau \arrow[dl] \arrow[dr] & \\
\mu_1 \arrow[d] & \cdots & \mu_8 \arrow[d] \\
\{Y_{i1}\} & \cdots & \{Y_{i8}\} \\
 & \sigma \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Second-stage prior

- μ and τ are hyperparameters for the normal prior distribution for $\{\mu_j\}$
- hyperparameters and hyperpriors
- The hyperprior for μ and τ:

$$\begin{aligned} \mu \mid \mu_0, \gamma_0 &\sim \mathrm{Normal}(\mu_0, \gamma_0) \quad (7) \\ 1/\tau^2 \mid a, b &\sim \mathrm{Gamma}(a_\tau, b_\tau) \quad (8) \end{aligned}$$

Second-stage prior

- μ and τ are hyperparameters for the normal prior distribution for $\{\mu_j\}$
- hyperparameters and hyperpriors
- The hyperprior for μ and τ:

$$\begin{aligned} \mu \mid \mu_0, \gamma_0 &\sim \mathrm{Normal}(\mu_0, \gamma_0) && (7) \\ 1/\tau^2 \mid a, b &\sim \mathrm{Gamma}(a_\tau, b_\tau) && (8) \end{aligned}$$

- e.g.$\mu_0 = 3$ and $\gamma_0 = 1$
- e.g. $a_\sigma = b_\sigma = 1$

Inference through MCMC

- Sampling: for $j = 1, \cdots, 8$ and $i = 1, \cdots, n_j$:

$$Y_{ij} \mid \mu_j, \sigma_j \overset{i.i.d.}{\sim} \mathrm{Normal}(\mu_j, \sigma_j) \tag{9}$$

- Prior for μ_j, Stage 1: for $j = 1, \cdots, 8$:

$$\mu_j \mid \mu, \tau \sim \mathrm{Normal}(\mu, \tau) \tag{10}$$

- Prior for μ_j, Stage 2: the hyperpriors:

$$\begin{aligned} \mu &\sim \mathrm{Normal}(3, 1) && (11) \\ 1/\tau^2 &\sim \mathrm{Gamma}(1, 1) && (12) \end{aligned}$$

- Prior for σ:

$$1/\sigma^2 \sim \mathrm{Gamma}(1, 1) \tag{13}$$

JAGS step 1: describe the model by a script

```
modelString <-"
model {
## sampling
for (i in 1:N){
   y[i] ~ dnorm(mu_j[MovieIndex[i]], invsigma2)
}
## priors and hyperpriors
for (j in 1:J){
   mu_j[j] ~ dnorm(mu, invtau2)
}
invsigma2 ~ dgamma(a_s, b_s)
sigma <- sqrt(pow(invsigma2, -1))
mu ~ dnorm(mu0, g0)
invtau2 ~ dgamma(a_t, b_t)
tau <- sqrt(pow(invtau2, -1))
}
"
```

JAGS step 2: define the data and prior parameters

```
y <- MovieRatings$rating
MovieIndex <- MovieRatings$Group_Number
N <- length(y)
J <- length(unique(MovieIndex))
the_data <- list("y" = y, "MovieIndex" = MovieIndex,
                 "N" = N, "J" = J,
                 "mu0" = 3, "g0" = 1,
                 "a_t" = 1, "b_t" = 1,
                 "a_s" = 1, "b_s" = 1)
```

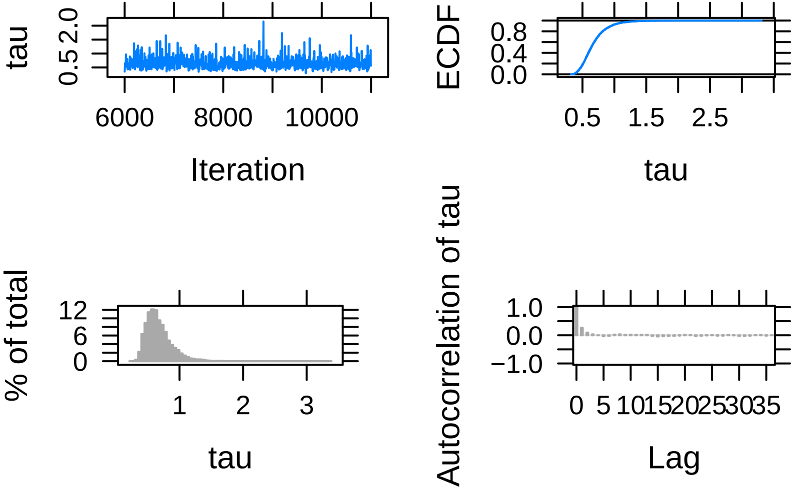
JAGS step 2: define the data and prior parameters cont'd

```
posterior <- run.jags(modelString,
                      n.chains = 1,
                      data = the_data,
                      monitor = c("mu", "tau",
                                  "mu_j", "sigma"),
                      adapt = 1000,
                      burnin = 5000,
                      sample = 5000)
```

MCMC diagnostics and summarization

```r
plot(posterior, vars = "tau")
```

```
## Generating plots...
```



MCMC diagnostics and summarization cont'd

```
##
## JAGS model summary statistics from 5000 samples (ada
##
##          Lower95 Median Upper95  Mean     SD Mode   M
## mu          3.25   3.78     4.4  3.78  0.287   -- 0.0
## tau        0.349  0.637    1.08 0.678  0.213   -- 0.0
## mu_j[1]     2.99   3.47    3.99  3.47  0.259   -- 0.0
## mu_j[2]     3.39   3.82    4.27  3.82  0.219   --   0.
## mu_j[3]     3.05   3.91    4.73  3.91  0.424   -- 0.0
## mu_j[4]     3.16   3.73    4.29  3.73  0.287   -- 0.0
## mu_j[5]     3.06   4.18    5.36   4.2  0.589   --   0.
## mu_j[6]     2.76   3.87       5  3.87   0.57   -- 0.0
## mu_j[7]     2.76   3.51    4.26  3.51  0.389   -- 0.0
## mu_j[8]     3.55   4.12    4.64  4.12  0.275   -- 0.0
## sigma      0.757  0.923    1.11 0.929 0.0928   --   0.
##
##          psrf
```

Inferences

- "How to Train Your Dragon" (corresponding to μ_1) and "Megamind" (corresponding to μ_7) have the lowest average ratings with short 90% credible intervals, (2.96, 3.99) and (2.74, 4.27) respectively
- "Legend of the Guardians: The Owls of Ga'Hoole" (corresponding to μ_6) also has a low average rating but with a wider 90% credible interval (2.70, 4.99)
- "Batman: Under the Red Hood" (corresponding to μ_5): average rating μ_5 has the largest median value among all μ_j's, at 4.15, and also a wide 90% credible interval, (3.09, 5.43)

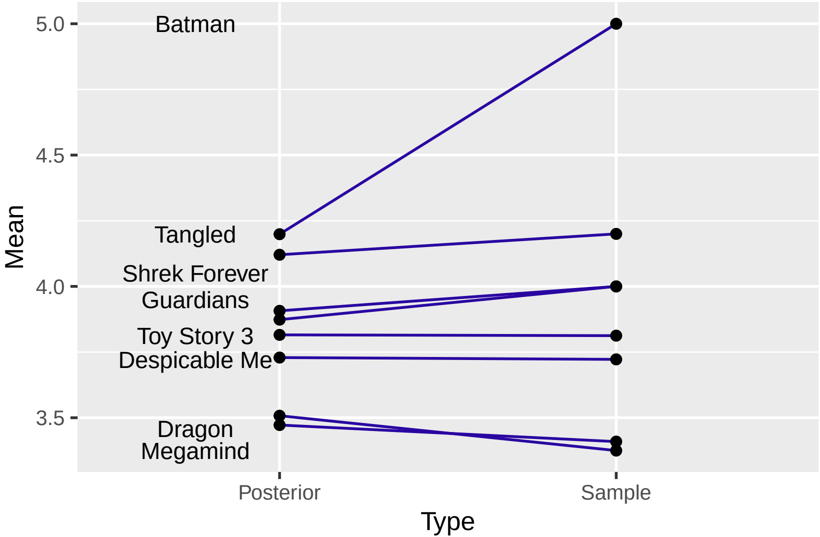
Inferences cont'd

- The differences in the width of the credible intervals stem from the sample sizes:
  - “How to Train Your Dragon” (11)
  - “Megamind” (4)
  - “Legend of the Guardians: The Owls of Ga’Hoole" (1)
  - The smaller the sample size, the larger the variability in the inference, even if one pools information across groups

Shrinkage

- The two-stage prior specifies a shared prior $\mathrm{Normal}(\mu, \tau)$ for all μ_j's
  - estimation of the movie mean ratings (the μ_j's)
  - estimation of the variation among the movie mean ratings through the parameters μ and τ
- The posterior mean of the rating for a particular movie μ_j shrinks the observed mean rating towards an average rating

Shrinkage cont'd



Shrinkage cont'd

- The shrinkage effect is obvious for the movie "Batman: Under the Red Hood"
- A large shrinkage is desirable for a movie with a small number of ratings such as "Batman: Under the Red Hood"
- For a movie with a small sample size, information about other ratings of similar movies helps to produce a more reasonable estimate at the "true" average movie rating.
- By pooling ratings across movies, one is able to estimate the standard deviation σ of the ratings
    - without this pooling, one would be unable to estimate the standard deviation for a movie with only one rating

Sources of variability

- Two sources for the variability among the observed Y_{ij}'s

$$
\begin{aligned}
Y_{ij} &\overset{i.i.d.}{\sim} \mathrm{Normal}(\mu_j, \sigma) \text{ [within-group variability]} && (14)\\
\mu_j \mid \mu, \tau &\sim \mathrm{Normal}(\mu, \tau) \text{ [between-group variability]} && (15)
\end{aligned}
$$

- The Bayesian posterior inference in the hierarchical model is able to compare these two sources of variability, taking into account
  - the prior belief
  - the information from the data

Sources of variability cont'd

- To compare these two sources of variation

$$R = \frac{\tau^2}{\sigma^2 + \tau^2} \tag{16}$$

- Calculate R from the posterior samples of σ and τ
- R represents the fraction of the total variability in the movie ratings due to the differences between groups
  - if R is close to 1, most of the total variability is attributed to the between-group variability
  - if R is close to 0, most of the variation is within groups and there is little significant differences between groups

Sources of variability cont'd

```
tau_draws <- as.mcmc(posterior, vars = "tau")
sigma_draws <- as.mcmc(posterior, vars = "sigma")
R <- tau_draws ^ 2 / (tau_draws ^ 2 + sigma_draws ^ 2)
quantile(R, c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.1469869 0.6329924
```

- A 95% credible interval for R
- The variation between the mean movie rating titles is overall smaller than the variation of the ratings within the movie titles in this example

Sources of variability cont'd

```
df = as.data.frame(R)
ggplot(df, aes(x=R)) +
  geom_density(size = 1, color = crcblue) +
  increasefont()
```

```
## Don't know how to automatically pick scale for objec
```