Posterior prediction for hierarchical models

Stat 340, Fall 2021

We'll build off our ELS math score example here to explore prediction.

Recap: Fitting the hierarchical model in JAGS

First, write the model in a way JAGS understands:

```
modelString <-"
model {

## sampling
for (i in 1:N){
   y[i] ~ dnorm(mu_j[school[i]], invsigma2)
}

## priors
for (j in 1:J){
   mu_j[j] ~ dnorm(mu, invtau2)
}

invsigma2 ~ dgamma(a_s, b_s)
sigma <- sqrt(pow(invsigma2, -1))

## hyperpriors
mu ~ dnorm(mu0, g0)
invtau2 ~ dgamma(a_t, b_t)
tau <- sqrt(pow(invtau2, -1))
}
"
```

]

Next, define the data and any initial values for your parameters:

```
y <- sub_school$mathscore
school <- sub_school$school
N <- length(y)
J <- length(unique(school))
the_data <- list(y = y, school = school,
                 N = N, J = J,
                 mu0 = 50, g0 = .04,  # prior parameters
                 a_t = 1, b_t = .01,  # hyperparameters
                 a_s = 1, b_s = .01)  # hyperparameters
```

Now, we can use `runjags` to fit the model via MCMC:

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

Posterior prediction

Suppose we want to make a prediction for school 13, **a group that we observed**, then we need a posterior predictive distribution

```
# Work with a data frame
post_df <- as.data.frame(posterior_full$mcmc[[1]])

# Create the posterior predictive via simulation
pp_school13 <- post_df %>%
  select(mu_j = "mu_j[13]", sigma) %>%               # select relevant cols
  mutate(y_pred = rnorm(nrow(post_df), mu_j, sigma)) # generate y
```

Next, suppose we want to make a **prediction for a school we didn't observe**, let's call it school 101

```
# Create the posterior predictive via simulation
pp_school101 <- post_df %>%
  select(mu, tau, sigma) %>%                   # select global params
  mutate(
    mu_j = rnorm(nrow(post_df), mu, tau),      # generate mu_j
    y_pred = rnorm(nrow(post_df), mu_j, sigma) # generate y
  )
```