STAT 535: Chapter 16: Normal Hierarchical Models

David B. Hitchcock
E-Mail: hitchcock@stat.sc.edu

Spring 2024

An Example of Hierarchical Data

- In this chapter we will focus on the `spotify` data set in the `bayesrules` package.
- This is a subset of a huge dataset of Spotify songs collected by Kaylin Pavlik in 2019.
- The response variable is the popularity of 350 songs. Note that artists (bands or singers) have **multiple songs** in this data set, so the data are grouped (clustered).
- Popularity values for songs coming from the same artist are likely to be correlated.

Complete Pooled Approach

- We will first try pooling all the data together and ignoring the grouping structure.
- Notation for our grouped data: Y_{ij} is the popularity of the i-th song for artist j.
- And n_j is the number of songs in the data set for artist j.
- Note that the first artist, Mia X, has 4 songs, so $n_1 = 4$.
- The overall sample size is

$$n = \sum_{j=1}^{44} n_j = n_1 + n_2 + \cdots + n_{44} = 350.$$

Complete Pooled Data Model

- If we ignore the grouping structure, we can assume the popularity values follow a normal distribution with some mean μ and variance σ^2.
- Let's check an estimated density for the popularity variable in order to see whether the data look normal.
- Here is a formal Bayesian Normal-normal model:

$$\begin{aligned} Y_{ij}|\mu,\sigma &\sim N(\mu,\sigma^2) \\ \mu &\sim N(50, 52^2) \\ \sigma &\sim \text{Exp}(0.048) \end{aligned}$$

- This assumes the most likely value for the overall mean μ is 50 (sensible since popularity values are between 0 and 100).
- Also, it is a weakly informative prior on σ.

Meaning of Model Parameters

- In this model, the parameters μ and σ are global parameters: They do not vary across artists.
- μ = global mean popularity and σ = global standard deviation in popularity from song to song.
- Note that this model is equivalent to a normal regression model with no predictors:

$$Y_{ij} = \beta_0 + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma^2)$$

- We can estimate this using `stan_glm` with a formula `popularity ~ 1`

Drawback of Complete Pooling Model

- We can estimate our mean μ, based on its posterior, using this approach.
- Drawback: If we want to predict the popularity of new songs from several new artists, the predictions will be all the same.
- For any artist, the predicted popularity of a new song is the posterior mean $E(\mu|y) = 58.39$.
- In R, we can plot the posterior predictive means for each artist based on this model (light blue dots) against the sample means for each artist (dark blue).
- We see the posterior predictive means do not reflect the actual differences in artist popularity at all: Bad model!

No Pooled Model

- Now let's use a "no pooling" approach: Allow each artist to have his/her own mean popularity μ_j:

$$Y_{ij}|\mu_j, \sigma \sim N(\mu_j, \sigma^2)$$

- Here, μ_j = mean song popularity for artist j and σ = the standard deviation in popularity from song to song within each artist.
- We **are** assuming the standard deviation is the same for artist to artist, however. Does that match reality? (see R plot)
- Not really, but let's go with this model for now, since assuming a common σ keeps the model simpler.

Formal No Pooling Model

- The data model has a lot of parameters: $44 + 1 = 45$, to be exact:

$$\begin{aligned} Y_{ij}|\mu_j, \sigma_j &\sim N(\mu_j, \sigma^2) \\ \mu_j &\sim N(50, s_j^2) \\ \sigma &\sim \text{Exp}(0.048). \end{aligned}$$

- We can estimate this with a regression model with a separate coefficient for each level of "artist" and no intercept, using the formula: `popularity ~ artist - 1`
- The priors on the μ_j's are each given means of 50, but they are weakly informative, so the data overwhelmed the weak prior information.
- The result is that the posterior predictive distribution of popularity for each artist is centered right at his/her sample mean popularity (see R plot).

Drawbacks of the No-Pooling Model

- This model assumes that the data on one artist cannot be used to estimate the popularity of another artist.
- When we have small sample sizes for a group (artist, here), then our inference about that artist's mean popularity is not precise.
- This model is also not generalizable: If you wanted to predict the mean popularity of a song by an artist who is **not in the sample** (for example, Taylor Swift), then we could not do it with this model.
- This model only tells us about the artists in the sample, not others in the wider population.

A Better Approach: A Hierarchical Model

- To better handle this data set, we now will propose a hierarchical model with three layers, describing:
  1. how song popularity varies within artist j
  2. how the artist-specific mean song popularity μ_j varies across artists
  3. prior models for the **global parameters** μ, σ_y, and σ_μ

Within-Group Normal Model

- We will assume that data values within group j (i.e., artist j here) follow a normal distribution:

$$Y_{ij}|\mu_j, \sigma_y \sim N(\mu_j, \sigma_y^2)$$

- We see that each artist is allowed to have his/her own mean song popularity μ_j, as with the no-pooling model.
- σ_y measures the within-group variability, i.e., the standard deviation in popularity from song to song within each artist.
- This within-group variability is assumed to be the same for each artist (may or may not be true in reality; it's always a good idea to check model assumptions through plots of the data).

Between-Group Layer

- Now, unlike in the no-pooling model, we include a layer that recognizes that all our sampled artists are drawn from a single population.
- We model the variation in mean popularity between (among) artists by assuming a normal model for the μ_j's:

$$\mu_j | \mu, \sigma_\mu \sim N(\mu, \sigma_\mu^2)$$

- The parameter μ (**without** a subscript) is the global average of mean song popularity $\mu_j$across all artists.
- The parameter σ_μ is the between-group variability, i.e., the standard deviation in mean popularity values μ_j among artists.
- Is the normality assumption for the μ_j's appropriate?
- We can't observe the μ_j's themselves, but we can observe the sample mean song popularity for each artist, which are estimates of the μ_j's.
- An estimated density plot of the artist sample means (see R code) shows the normality assumption looks reasonable.

Priors on the Global Parameters

- For this to be a Bayesian model, we need to specify priors on the global parameters μ, σ_y, and σ_μ.
- We will follow the textbook's recommendations for these.
- We will let the prior on μ be Normal, specifically $N(50, 52^2)$: The overall mean of 50 is sensible, and the large variance implies prior uncertainty.
- We will let the prior on σ_y be Exponential with rate 0.048. Any distribution with support on $(0, \infty)$, like the gamma, inverse-gamma, etc., would be reasonable.
- We will let the prior on σ_μ be Exponential with rate 1.

Analysis of Variance

- This is a Bayesian version of the One-Way Analysis of Variance (ANOVA) model.
- The goal of ANOVA is to compare the means of several populations (groups) by comparing the between-group variability and within-group variability.
- In our example, the artists are the groups, and we want to estimate the 44 artist-level means $\mu_1, \ldots, \mu_{44}$.
- Using ANOVA, we can break up the total variance in our Y_{ij}'s into within-group variance σ_y^2 and between-group variance σ_μ^2:

$$\text{Var}(Y_{ij}) = \sigma_y^2 + \sigma_\mu^2$$

Proportion of Variance Explained

- The textbook notes that

$$\frac{\sigma_y^2}{\sigma_\mu^2 + \sigma_y^2} = \text{proportion of Var}(Y_{ij}) \text{ that can be explained by differences in the observations within each group}$$

$$\frac{\sigma_\mu^2}{\sigma_\mu^2 + \sigma_y^2} = \text{proportion of Var}(Y_{ij}) \text{ that can be explained by differences between groups}$$

- Note that $\frac{\sigma_\mu^2}{\sigma_\mu^2 + \sigma_y^2}$ measures the **within-group correlation**, e.g., the correlation between popularities of songs by the same artist.
- Clearly, the model forces this correlation to be positive, which makes sense.

Fitting the Bayesian Model

- The posterior analysis can be done with the `stan_glmer` function.
- The syntax is similar to `stan_glm`, with some slight differences: Since "artist" is a grouping variable in the model rather than a predictor, the formula is `popularity ~ (1 | artist)`
- The `pp_check` function compares the posterior predictive density to the observed data's density in order to check model fit (see R code).
- The Normal hierarchical model fits reasonably well, albeit not perfectly.

Posterior Inference about Model Parameters

- Posterior inference (point estimates, credible intervals) about the global parameters is straightforward in R.
- A posterior point estimate for μ is 52.5, while an 80% credible interval for μ is $(49.3, 55.7)$.
- Posterior estimates of σ_μ and σ_y are 15.1 and 14.0, respectively.
- Thus the estimate of the correlation in song popularity values for **songs from the same artist** is

$$\frac{15.1^2}{15.1^2 + 14.0^2} = 0.54$$

- This is a moderate positive (linear) association.

Posterior Inference about Group-Specific Parameters

- We can also get point and interval estimates of the μ_j's (see R code).
- Our point estimate for Beyoncé's mean popularity is 69.1, and with 80% posterior probability, her mean popularity is between 65.6 and 72.7.
- Our point estimate for Vampire Weekend's mean popularity is 61.6, and with 80% posterior probability, their mean popularity is between 54.8 and 68.5.
- Note the credible intervals' widths vary from artist to artist.
- The wider intervals correspond to artists who have smaller sample sizes (see Frank Ocean vs. Lil Skies).

Posterior Prediction for an Artist in the Sample

- Suppose we wanted to predict the popularity of a new song by an artist in the sample, say, Vampire Weekend.
- An 80% prediction interval for the popularity of a new song by Vampire Weekend is (see R code) $(42.5, 80.8)$.
- Note this is much wider than the 80% credible interval for Vampire Weekend's mean popularity μ_j.
- Does it make sense that we can predict their mean popularity with more precision than we can predict the popularity of one of their songs? Yes.

Posterior Prediction for an Artist Not in the Sample

- Suppose we wanted to predict the popularity of a new song by an artist in the broad population of artists, but **not** in the sample, for example Taylor Swift.
- Recall that we could not do this with the no-pooling model.
- With the hierarchical model, we can use our knowledge about the broader population to make such a prediction.
- We would (1) simulate a set of μ_j values for Swift from our $N(\mu, \sigma_\mu)$ distribution (while varying μ and σ_μ) according to their own posterior distributions; and (2) simulate Y values from the resulting $N(\mu_j, \sigma_y)$ distribution (varying σ_y according to its posterior).
- An 80% prediction interval for the popularity of a new song by Taylor Swift is (see R code) $(25.9, 78.9)$.

Is this Prediction Accurate?

- In real life, do we really believe this prediction for the popularity of a new song by Taylor Swift? Probably not.
- If our "new artist" were someone that we really had no knowledge about, then the values in this prediction interval would be sensible.
- But Taylor Swift is one of the most popular recording artists in the world, so we would in reality expect her new song's popularity to be in the high range.
- To better reflect this, perhaps a more realistic model could include a variable that measures the fact that Taylor is like, totally awesome.
- Seriously, though, our model's prediction of popularity of an artist's song would be better if we included one or more artist-level variables like number of past Grammy nominations, past radio airplay, etc.
- This is something explored in Chapter 17, in which the hierarchical model includes one or more predictor variables.

Shrinkage

- We can plot point and interval predictions (in light blue) of new song popularities for all 44 artists in the sample.
- On the plot, we will also overlay the sample mean popularity (in dark blue) for each artist (see plot).
- This shows the phenomenon called **shrinkage**: Our hierarchical model's predictions **shrink** (or pull) the artists' sample means toward the **global sample mean**.
- Recall that the complete-pooling model would predict song popularity using the global mean, and the no-pooling model would predict song popularity using the artist's own mean.
- So our hierarchical model is a balance of those models.

How much Shrinkage?

- The artists whose own means are **shrunk most** toward the global mean are the ones with the **smallest sample size** (note these have the wide credible intervals for their μ_j's).
- This makes sense: If we have less data on an artist, we want to borrow information from the other artists in the population to help our prediction for that artist.
- If we have lots of data on an artist (Frank Ocean), then for that person's prediction, we don't need to rely as much on the data from other artists.
- Intuitive free throw example: Player A has made 98 of 100 free throws. Player B has made 3 of 3 free throws. Which player do you believe has a higher probability of making her next free throw?

Grouping Variable or Predictor?

- We treated "artist" as a grouping variable in our model. Why did we not treat it as a categorical predictor in an ordinary linear model?
- If the levels of the variable in our sample are all the levels that we care about, then we would include it as a predictor in the model, for example like we did for our "track" variable in our Poisson model with the academic awards.
- In that case, "academic", "vocational", and "general" were all the levels of that variable and NOT a random sample from a larger population of levels.
- In the Spotify example, the artists in the sample are a sample from a **large population of artists**.
- Including "artist" as a grouping variable allows us to make conclusions about the whole population of artistics, including artists not in the sample.
- This is basically the same as the distinction between fixed effects and random effects in classical ANOVA models.