Class 7: Multi-layer hierarchical models

Andrew Parnell
andrew.parnell@mu.ie

Learning outcomes:

- Understand how to add in multiple layers to a hierarchical model
- Follow a detailed example of building a model
- Be able to work with missing data in JAGS

Some new terminology

- Most of the models we have covered so far contain only one hidden or *latent* set of parameters
- For example, the data y may depend on a parameter β, which itself depends on a parameter θ. θ is given a prior distribution
- We say that the data are at the 'top level', the parameter β is a *latent parameter* at the second level, and the hyper-parameter θ is also a latent parameter at the third level
- We say that the prior distribution on β is *conditional* on θ, whilst the prior distribution (if it just involves numbers) is a *marginal prior distribution*

What is a multi-layer model?

- A multi-layer model is one where have many (usually more than 2 or 3) layers of parameters conditional on each other
- It's very straightforward to add in these extra layers in JAGS/Stan
- The question is whether they are necessary or not, and how much the data can tell us about them

Writing the same model in different ways

- The model on the previous slides has a different intercept and slope for each ethnicity group, with the information about them tied together through the prior distributions on them
- The likelihood was written as:

```
y[i] ~ dnorm(alpha[eth[i]] +
               beta[eth[i]]*(x[i] - mean(x)),
                 sigma^-2)
```

which in maths can be written as:

$$y_i \sim N(\alpha_{\text{eth}_i} + \beta_{\text{eth}_i} x_i, \sigma^2)$$

where eth_i takes the values 1, 2, 3, or 4

- Remember, y_i is the log-earnings of individual i where $i = 1, \ldots, N$

Re-writing the model

- Commonly you'll see y here re-defined as y_{ij} where $j = 1, .., 4$ represents ethnicity, and $i = 1, \ldots, N_j$ is the number of individuals with ethnicity j
- The likelihood can then be written as:

$$y_{ij} \sim N(\alpha_j + \beta_j x_{ij}, \sigma^2)$$

- Note that this is exactly the same model, just re-written slightly differently. In fact, this latter model is much harder to write out in JAGS/Stan code

Fixed vs random effect models

- Thinking about this model in more detail

$$y_{ij} \sim N(\alpha_j + \beta_j x_{ij}, \sigma^2)$$

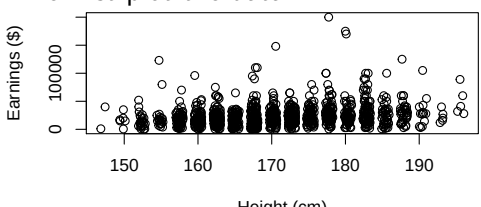
- If the α_j and β_j parameters are all given independent prior distributions, e.g. $\alpha_j \sim N(0, 100)$ or similar, then this is considered a *fixed effects* model
- If the α_j and β_j are given prior distributions that tie the values together, e.g. $\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2)$, then this is often called a *random effects* model
- (In fact, nobody can agree on what a fixed or random effects model actually is)

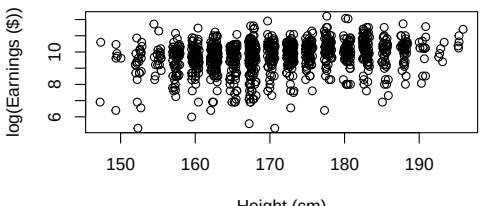
Mixed effects vs hierarchical models

- The hierarchical models we have been studying all use the *random effects* approach wherever possible
- The big advantage of using this approach is that we get to *borrow* strength between the different groups (here `eth`, but it could be anything)
- Whenever we have a categorical covariate we should always be putting a constraining/tying prior distribution on them, and looking at how the effects vary between the groups
- Mathematically you can write out the hierarchicaly estimated intercepts of a group (α_j) as a weighted average of the group intercept means from the data ($\bar{\alpha}_j$) and the overall mean of the entire data set (μ) where the weights are dependent on the group and overall variance and sample sizes.
- Because of the weighted nature of the estimate this is often called *partial pooling* or *shrinkage*

Example: multi-layer earnings data

- We will now go through and build a much more complicated model for the earnings data, taken from the Gelman and Hill book, using only weak priors
- We can generate data from these models (either using the prior or the posterior)
- Our goal is to explore the factors which explain earnings. We have variables on height, age, and ethnicity.
- If we first plot the data





Transformations

- From the left-hand plot there seem to be quite a few extreme observations, and there's a possibility that the relationship between height and earnings is non-linear
- The right-hand plot seems to have stabilised most of the extreme observations, and perhaps linearity is more appropriate
- Notice that a linear model implies:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

whilst the log-linear model implies:

$$y_i = \exp(\alpha + \beta x_i + \epsilon_i) = e^{\alpha} \times e^{\beta x_i} \times e_i^{\epsilon}$$

so the coefficients, once exponentiated, have multiplicative effects that are relatively easy to interpret

Fitting the first model

- If we fit a model with just height (mean centered) we get the following JAGS output

```
## Inference for Bugs model at "4", fit using jags,
##  3 chains, each with 2000 iterations (first 1000 discarded)
##  n.sims = 3000 iterations saved
##              mu.vect sd.vect     2.5%      25%      50%      75%    97.5%  Rhat
## alpha          9.737   0.028    9.683    9.718    9.737    9.756    9.793 1.001
## beta_height    0.023   0.003    0.017    0.021    0.022    0.024    0.028 1.001
## sigma          0.908   0.020    0.870    0.894    0.908    0.921    0.949 1.002
## deviance    2798.573   2.533 2795.735 2796.686 2797.909 2799.654 2805.416 1.002
##             n.eff
## alpha        3000
## beta_height  3000
## sigma        1800
## deviance     3000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 3.2 and DIC = 2801.8
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Interpreting the parameters

- These parameters are directly interpretable:
  - The mean of the log earnings at the mean height is about 9.737, which is about 17k on the original scale
  - We can also get e.g. a 95% confidence interval using the JAGS output. From 16000 to 18000
  - For every extra cm so you gain 0.0226 on the log scale, i.e. an 2.28% gain in income
  - From the posterior of σ, we can guess that about 68% of predictions will be within 0.908 on the log scale or within a factor of about 2.48 of the prediction
- Interpretation for the intercept would have been harder had we not mean-centered the height variable
- The DIC is 2801.78 with 3.21 effective parameters

Improving the model

- Now suppose we fit a model with a random intercept for ethnicity

```
jags_code = '
model{
  # Likelihood
  for(i in 1:N) {
    log_earn[i] ~ dnorm(alpha_eth[eth[i]] +
                        beta_height*(height[i] - mean(height)),
                        sigma^-2)
  }
  # Priors
  for(j in 1:N_eth) {
    alpha_eth[j] ~ dnorm(mu_eth, sigma_eth^-2)
  }
  beta_height ~ dnorm(0, 0.1^-2)
  mu_eth ~ dnorm(11, 2^-2)
  sigma_eth ~ dt(0, 5^-2, 1)T(0,)
  sigma ~ dt(0, 5^-2, 1)T(0,)
}
'
```

Improving the model 2

```
## Inference for Bugs model at "5", fit using jags,
##  3 chains, each with 2000 iterations (first 1000 discarded)
##  n.sims = 3000 iterations saved
##              mu.vect sd.vect     2.5%      25%      50%      75%    97.5%
## alpha_eth[1]   9.683   0.075    9.525    9.635    9.691    9.733    9.817
## alpha_eth[2]   9.660   0.094    9.451    9.603    9.677    9.725    9.818
## alpha_eth[3]   9.748   0.030    9.688    9.728    9.748    9.768    9.809
## alpha_eth[4]   9.749   0.122    9.520    9.683    9.736    9.807   10.036
## beta_height    0.022   0.003    0.017    0.020    0.022    0.024    0.028
## mu_eth         9.717   0.155    9.430    9.667    9.718    9.759   10.036
## sigma          0.907   0.020    0.870    0.894    0.907    0.920    0.948
## deviance    2797.856   3.059 2793.661 2795.749 2797.230 2799.427 2805.381
##              Rhat n.eff
## alpha_eth[1] 1.001  3000
## alpha_eth[2] 1.002  1900
## alpha_eth[3] 1.001  3000
## alpha_eth[4] 1.007   380
## beta_height  1.001  3000
## mu_eth       1.094  1100
## sigma        1.002  1500
## deviance     1.001  3000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 4.7 and DIC = 2802.5
## DIC is an estimate of expected predictive error (lower deviance is better).
```

Interpreting the output

- The parameters α and β_{height} haven't changed much in the mean
- The 95% confidence interval for α has increased: 12000 to 23000
- The DIC is 2802.54 with 4.68 effective parameters. Pretty much the same as above
- We also have estimates for each ethnicity, none of these have a strong effect away from zero

Now an interaction model

```
jags_code = '
model{
  # Likelihood
  for(i in 1:N) {
    log_earn[i] ~ dnorm(alpha_eth[eth[i]] +
                        beta_height[eth[i]]*(height[i] - mean(height)),
                        sigma^-2)
  }
  # Priors
  for(j in 1:N_eth) {
    alpha_eth[j] ~ dnorm(mu_eth, sigma_eth^-2)
    beta_height[j] ~ dnorm(mu_beta_height, sigma_height^-2)
  }
  mu_beta_height ~ dnorm(0, 0.1^-2)
  mu_eth ~ dnorm(11, 2^-2)
  sigma_eth ~ dt(0, 5^-2 ,1)T(0,)
  sigma_height ~ dt(0, 1, 1)T(0,)
  sigma ~ dt(0, 5^-2, 1)T(0,)
}
'
```

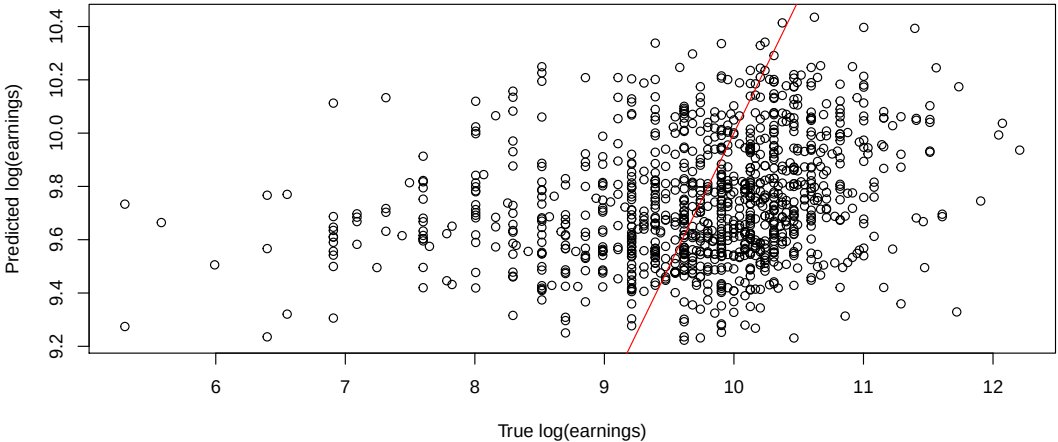
Interaction model results

```
## Inference for Bugs model at "6", fit using jags,
##  3 chains, each with 2000 iterations (first 1000 discarded)
##  n.sims = 3000 iterations saved
##                  mu.vect sd.vect     2.5%      25%      50%      75%    97.5%
## alpha_eth[1]       9.676   0.079    9.506    9.629    9.685    9.730    9.802
## alpha_eth[2]       9.644   0.102    9.418    9.579    9.660    9.719    9.796
## alpha_eth[3]       9.748   0.030    9.689    9.727    9.747    9.769    9.807
## alpha_eth[4]       9.730   0.124    9.479    9.659    9.726    9.787   10.017
## beta_height[1]     0.009   0.009   -0.008    0.003    0.009    0.015    0.025
## beta_height[2]     0.012   0.010   -0.008    0.006    0.013    0.019    0.031
## beta_height[3]     0.025   0.003    0.019    0.023    0.025    0.027    0.031
## beta_height[4]     0.008   0.016   -0.030   -0.001    0.010    0.019    0.034
## mu_beta_height     0.014   0.018   -0.024    0.008    0.015    0.021    0.047
## mu_eth             9.705   0.121    9.450    9.658    9.712    9.758    9.941
## sigma              0.905   0.019    0.870    0.891    0.905    0.918    0.944
## sigma_height       0.024   0.027    0.002    0.009    0.015    0.027    0.110
## deviance        2793.247   3.721 2787.598 2790.481 2792.813 2795.410 2801.979
##                   Rhat n.eff
## alpha_eth[1]     1.008   270
## alpha_eth[2]     1.016   130
## alpha_eth[3]     1.005   460
## alpha_eth[4]     1.006   730
## beta_height[1]   1.012   180
## beta_height[2]   1.007   300
## beta_height[3]   1.001  2800
## beta_height[4]   1.019   120
## mu_beta_height   1.041   230
## mu_eth           1.008   790
## sigma            1.002  3000
## sigma_height     1.030    74
## deviance         1.001  3000
##
## For each parameter, n.eff is a crude measure of effective sample size,
```

Interpreting the output

- The model has improved a bit, DIC now 2800.17 with 6.92 effective parameters
- The confidence intervals for the different slopes are highly different, with the whites group (ethnicity = 3) having a much clearer relationship with height, possibly due to the large sample size
- Go back to the previous classes to see plots of these effects

Checking the model - posterior predictive fit



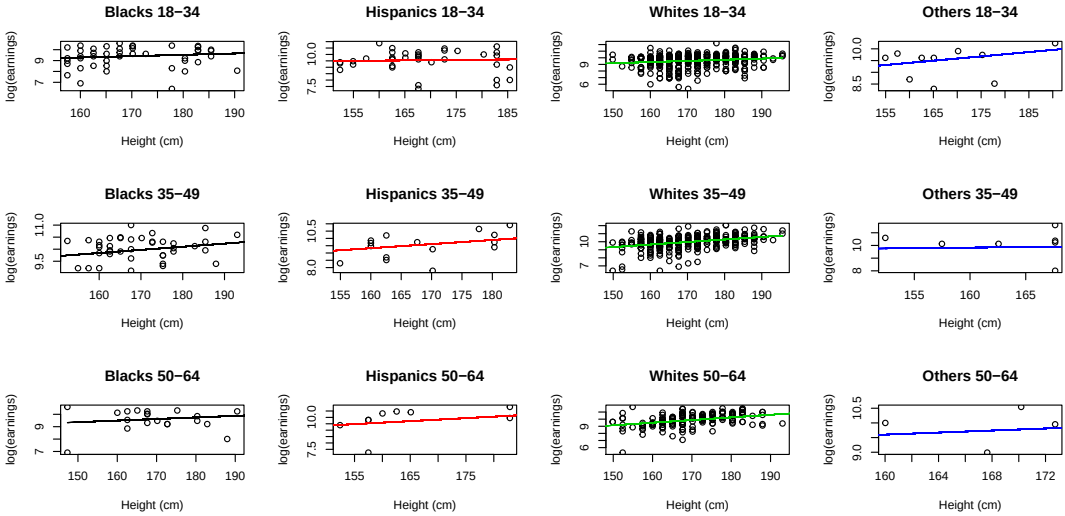
Introducing age

- Let's fit an even more complicated model with intercepts and slopes varying by ethnicity and age group
- Age is divided up into three groups 1: 18-34, 2: 35-49, and 3: 50-64
- We want to know whether the degree to which height affects earnings for different ethnic/age group combinations

JAGS model

```
jags_code = '
model{
  # Likelihood
  for(i in 1:N) {
    log_earn[i] ~ dnorm(alpha[eth[i],age_grp[i]] +
                          beta[eth[i],age_grp[i]]*(height[i] - mean(height)),
                    sigma^-2)
  }
  # Priors
  for(j in 1:N_eth) {
    for(k in 1:N_age_grp) {
      alpha[j,k] ~ dnorm(mu_alpha, sigma_alpha^-2)
      beta[j,k] ~ dnorm(mu_beta, sigma_beta^-2)
    }
  }
  mu_alpha ~ dnorm(11, 2^-2)
  mu_beta ~ dnorm(0, 0.1^-2)
  sigma_alpha ~ dt(0,5^-2,1)T(0,)
  sigma_beta ~ dt(0,1,1)T(0,)
  sigma ~ dt(0,5^-2,1)T(0,)
}
'
```

Model output



More about this model

- So we now have varying effects - we should also plot the uncertainties in these lines (see practical)
- The DIC here is now DIC now 2738.44 with 20.54 effective parameters - a big drop!

Missing and unbalanced data

- There are many definitions of what 'unbalanced' data means in statistics. Usually we mean that there are different numbers of observations in each group. Our format of writing e.g. $y_i \sim N(\alpha_{\text{eth}_i} + \beta_{\text{eth}_i} x_i, \sigma^2)$ allows us to deal with unbalanced data naturally
- Usually the smaller the sample size of the group the more uncertain the posterior distribution will be
- But what if we have some missing data? There are different types, and some need to be more carefully treated than others

Different types of missing data

- There are many different types of missing data:
  - Missing response variables
  - Missing covariates
  - Missingness that occurs completely at random
  - Missingness that occurs as a consequence of the experiment or the data
- The first three are all very easy to deal with in JAGS (less so in Stan). The last one is much harder, and not something we will go into in any detail. It requires building a separate model for the missingness process

The simple way of dealing with missing data in JAGS

- In JAGS it is absolutely trivial to to deal with missingness in the response variable. You simply fill in the missing values with `NA`
- JAGS then treats them as parameters to be estimated. You can 'watch' them in the normal way or just ignore them. You thus have the option of getting a posterior distribution of the missing data points
- Suppose we shoved in some NA values into our data

```
dat2 = dat
dat2$earn[c(177, 763, 771)] = NA
```

Running the model with missingness

```
print(jags_run)
```

```
## Inference for Bugs model at "5", fit using jags,
##  3 chains, each with 2000 iterations (first 1000 discarded)
##  n.sims = 3000 iterations saved
##                mu.vect sd.vect     2.5%      25%      50%      75%    97.5%
## log_earn[177]   10.330   0.872    8.628    9.741   10.337   10.897   12.052
## log_earn[763]    9.487   0.881    7.754    8.879    9.480   10.100   11.180
## log_earn[771]    9.439   0.891    7.669    8.851    9.447   10.018   11.179
## deviance      2704.327   6.482 2693.373 2699.753 2703.778 2708.325 2719.159
##                Rhat n.eff
## log_earn[177] 1.001  3000
## log_earn[763] 1.003   820
## log_earn[771] 1.001  3000
## deviance      1.005   480
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 20.9 and DIC = 2725.3
## DIC is an estimate of expected predictive error (lower deviance is better).
```

More complex varieties of missing data

- If you have missing covariates or joint missing covariates/response variables, you can include these too
- The only extra issue is that you need to give JAGS a prior distribution for the missing covariate values which can make the code a bit fiddlier
- If the response variable (e.g. log earnings) exists but the covariate value is missing, then you are asking JAGS to perform an *inverse regression*
- In Stan missing data is fiddlier to incorporate as you have to separate out the parameters (i.e. missing data) from the observed data

Summary

- We have seen how to create some rich multi-layer models
- We have gone through quite a detailed example
- We have discovered how to deal with missing and unbalanced data sets