

Hierarchical Models in Linear Regression

Basic Ideas

The literature on random effects models (term generally used by frequentists) or hierarchical models (term generally used by Bayesians) is huge, deserving of an entire course (or two) by itself. We will focus on getting the basic idea, and some simple examples.

What do we mean by a hierarchical or random effects model?

These sorts of models refer to when the parameters of models themselves have distributions (hence there is a hierarchy of parameters), or the parameters come from random distributions (hence the term random effects).

For example, consider the basic simple linear regression model:

$$Y = \alpha + \beta X$$

This model assumes that for each individual contributing data, the effects (both intercept and effect of X as given by β) are identical. In other words, all subjects are subject to the identical intercept and slope.

It is easy to imagine that this is often not the case. Consider, for example, giving an individual a drug to increase their bone mineral density (BMD). Now, in a simple model we might assume that β is constant across individuals, but, more realistically, one might consider β as a mean or average effect, and the effects for each individual (owing to body chemistry, other medications they may be taking, etc.) may be different.

Now, if we knew all of these effects, we may begin creating interaction models, which would allow the slope (effect of the drug on BMD) to depend on other characteristics. But if there are one or more *unknown* effects, we can create a hierarchical or random effects model as follows:

$$Y = \alpha + \beta_{fixed}X + \beta_{random}X$$

$$\beta_{random} \sim N(0, \sigma^2)$$

or equivalently

$$Y = \alpha + \beta X$$

$$\beta \sim N(\mu_\beta, \sigma^2)$$

Even simpler, we can have random effects / hierarchical models on the parameters of a normal distribution. Consider the following example:

Suppose that *mean* diastolic blood pressure varies randomly in a population with a mean of 70 mmHg, and standard deviation 10 mmHg. Suppose that in addition, however, each individual's blood pressure also varies throughout the day about their own individual means, with a standard deviation of 5 mmHg.

How could we model this situation?

Suppose we have m measures on each of n subjects. The following model seems to mirror the two sources of variability:

$$x_{ij} \sim N(\mu_i, \sigma_w^2), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m$$

$$\mu_i \sim N(\mu, \sigma_b^2), \quad i = 1, 2, \dots, n$$

Given some data and the above model, we could estimate the parameters from either a frequentist or Bayesian perspective. We will simulate some blood pressure data, and then show how estimation is done in R (frequentist) and WinBUGS (Bayesian).

```
# Simulate data for each individual in two stages:

# First stage: Randomly select means for each of 100 subjects:

> bp.means <- rnorm(100, mean=70, sd=10)

> bp.means
[1] 76.27620 73.28998 81.77357 71.82937 65.39472 ... etc ....

# Second stage, generate 10 individual data points
# for each of 100 subjects, one subject per row of the data set
```

```

# Create an empty vector to store individual data

> bp.data <- rep(NA, 1000)

# Fill in vector with individual values of correct means

> for(i in 1:100) {
  for(j in 1:10) {
bp.data[10*(i-1) + j] <- rnorm(1, mean=bp.means[i], sd=5)
  }
}

# Note how bp.means match each row

> bp.data
 [1] 78.491 81.989 77.345 78.327 75.952 86.086 81.257 75.299 69.783 71.300
[11] 72.918 70.602 65.253 70.061 75.387 69.744 68.688 68.035 67.458 82.689
[21] 82.443 78.124 71.675 77.090 90.796 86.059 81.560 83.870 82.229 76.975
[31] 70.801 79.688 81.731 77.678 73.670 69.109 75.066 68.126 80.726 68.882
[41] 66.379 60.884 69.669 69.959 59.245 61.227 71.388 76.070 64.878 66.273
[51] 63.037 58.527 59.353 54.285 59.518 61.577 55.452 65.067 62.005 65.482
..... etc .....

# We next need an indicator variable to show that the data
# are arranged in groups of 10.

> individual <- rep(NA, 1000)

> for(i in 1:100) {
  for(j in 1:10) {
individual[10*(i-1) + j] <- i
  }
}

# Check that it has worked

> individual
 [1] 1 1 1 1 1 1 1 1 1 1
[11] 2 2 2 2 2 2 2 2 2 2
[21] 3 3 3 3 3 3 3 3 3 3
[31] 4 4 4 4 4 4 4 4 4 4
[41] 5 5 5 5 5 5 5 5 5 5
..... etc .....

```

```

# Now fit a linear model with a random effect
# We use the lme (linear mixed effects model) function in R

# Need to prepare data for this function

# First create a factor variable for the individuals

Subject <- as.factor(individual)

# Then use groupedData function to create data structure for lme
# The nlme package, as it is not included with the base
# functions in R. It can be loaded in same way as BMA.

# A data frame is created called bp.data.frame:

> bp.data.frame <- data.frame(bp = bp.data, Subject)

# Line below shows that data are grouped by subject

> bp.grouped <- groupedData(bp ~ 1 | Subject, data=bp.data.frame)

# Take a look at the resulting data structure

> bp.grouped
Grouped Data: bp ~ 1 | Subject
      bp Subject
1    78.49153     1
2    81.98980     1
3    77.34545     1
4    78.32754     1
5    75.95297     1
.....etc.....

# Same as a usual data frame, but with extra "by subject" structure

# Now run lme function

> output <- lme(bp ~ 1, data=bp.grouped)
> summary(output)
Linear mixed-effects model fit by REML
Data: bp.grouped
      AIC      BIC    logLik
6490.722 6505.442 -3242.361

Random effects:

```

```

Formula: ~1 | Subject
      (Intercept) Residual
StdDev:    9.475058 5.197294

```

```

Fixed effects: bp ~ 1
      Value Std.Error DF t-value p-value
(Intercept) 70.58773 0.9616544 900 73.40238 0

```

```

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-3.17141708 -0.61052249 -0.01657445 0.65436049 2.65165830

```

```

Number of Observations: 1000
Number of Groups: 100

```

```

# Fixed effects were just the overall mean, the intercept.
# Random effects fit a different mean for each subject, around the
# overall mean (intercept) of 70.58773.

```

```

# Note that both standard deviations were well fit, with
# between SD = 9.4 (truth = 10) and within SD = 5.2 (truth = 5)

```

```

# We can also see various other outputs of interest:

```

```

# Get estimated mean for each subject
#   = difference from overall mean (intercept)

```

```

> output$coefficients
$fixed
(Intercept)
 70.58773

```

```

$random
$random$Subject
  (Intercept)
1    6.7913646
2    0.4818299
3   10.1884320
4    3.8446019
5   -3.8735294
6   -9.8602643
...etc...
100 -8.7640744

```

```

# So Subject 1 is about 6.7 points above average bp values, and so on.

```

```
# Can look at residuals:

> output$residuals
      fixed      Subject
1    7.903807955    1.112443319
2   11.402075170    4.610710534
3    6.757719382   -0.033645254
4    7.739813071    0.948448435
5    5.365241734   -1.426122902

# Note two types of residuals, one ignoring individual Subject effect
# the other including it, so always lower.

# See help file for more details on lme.
# In particular, see help on lmeObject for all available outputs
```

We can also program the same example in WinBUGS:

```
# First convert data set from R for use in WinBUGS

> dput(bp.grouped$bp, file="g:\\bp.txt")

# We will cut and paste this file into a WinBUGS analysis window.
```

Write the program for WinBUGS:

```
model {

for( i in 1:100)      # Loop over 100 subjects {

for (j in 1:10)      # Loop over 10 data points per subject
  {
    bp[j,i] ~ dnorm(mu[i] , tau.w) # Note [j,i] not [i,j]
    # Data are normally distributed
    # about individual mean mu[i]
  }
  mu[i] ~ dnorm(mean.bp, tau.b) # Means are normally distributed
  # with overall mean mean.bp
}

mean.bp ~ dnorm(70, .001)      # Prior for mean BP, var = 1/0.001 = 1000
```

```

tau.w <- 1/(sigma.w*sigma.w) # Prior for within SD (precision)
sigma.w ~ dunif(0,100)

tau.b <- 1/(sigma.b*sigma.b) # Prior for between SD (precision)
sigma.b ~ dunif(0,100)

}

# Initial Values

list(sigma.w = 7, sigma.b = 12, mean.bp = 70)

# Data

# Note some changes needed from simple dput file:
# Add .Data word, reverse .Dim

list(bp=structure(.Data=c(78.491534163798, 72.9179967852737,
82.4433694101257, 70.8014851147823, 66.3791467338957,
63.0373238043374, 67.4878552683662, 85.0656743211218,
79.0398796878923, 78.5438769720441, 78.7772194526441,
71.9931192939269, 53.0873697753945, 59.7622998188586,
..... etc .....
77.5633200078565, 58.1785984593208, 67.7575083592776,
78.578700312957, 60.9872446515102), .Dim = c(10, 100)))

# Results

node    mean    sd      MC error  2.5%    median  97.5%  start  sample
mean.bp 70.59  0.9571  0.009583  68.73   70.58   72.47   1001   10000
mu[1]   77.39  1.616   0.0165    74.22   77.4    80.57   1001   10000
mu[2]   71.07  1.615   0.01547   67.95   71.07   74.2    1001   10000
mu[3]   80.74  1.632   0.01429   77.55   80.73   83.95   1001   10000
mu[4]   74.44  1.614   0.01631   71.28   74.42   77.55   1001   10000
mu[5]   66.7   1.644   0.0169    63.49   66.73   69.89   1001   10000
..... etc .....
mu[99]  72.54  1.614   0.01439   69.4    72.53   75.73   1001   10000
mu[100] 61.82  1.606   0.01577   58.68   61.81   64.94   1001   10000
sigma.b  9.592  0.7088  0.007846  8.305   9.553   11.1    1001   10000
sigma.w  5.204  0.1227  0.001459  4.968   5.203   5.453   1001   10000
tau.b    0.01104 0.001616 1.768E-5  0.008123 0.01096 0.0145  1001   10000
tau.w    0.03698 0.001744 2.065E-5  0.03363 0.03693 0.04052 1001   10000

```

The results are more complete than what was available from R, but numerically similar. We easily get CrIs from all parameters, and means are already including the overall mean value (intercept).

Note the overall excellent results for all parameters.

Advantage of hierarchical models

Consider the BP example again. At one extreme, we could consider that each person has their own mean, and estimate the means 10 data points at a time. In other words, no pooling of data.

Advantage: Unbiased individual estimates. Disadvantage: Low accuracy

At the other extreme, we can consider all subjects similar enough, and just estimate a single pooled estimate.

Advantage: High accuracy. Disadvantage: No individual estimates.

Hierarchical estimates are a compromise between these two extremes. One lets the data decide on the degree of pooling. The more similar the mean estimates are, the smaller σ_b will be, and the more pooling takes place. The more different the individual means are, the larger σ_b will be, and less pooling takes place.

Advantages: High accuracy while retaining individual estimates.

Disadvantage: Some slight bias may be introduced to individual estimates, although overall Mean Square Error is optimized.

Example of “bias” in BP example:

Compare mean estimates from “straight average of 10 individual values” to means estimated by hierarchical (random effects) model:

Subject	Mean Ind.	Mean Hier Model
1	77.58343	77.39
2	71.08405	71.07
3	81.0827	80.74
4	74.548	74.44
5	66.59765	66.70

Recall that the overall mean was 70.59, and note that all estimates from the hierarchical model are slightly moved towards that value. This is the *slight* effect of pooling here, only slight because σ_b was rather large. If we repeat the exercise with a smaller

values for σ_b , this “borrowing of strength” or “movement towards the middle would be larger.

The effect will also tend to be larger when there is a smaller number of observations per subject.

Example to linear regression

CRP is a measure of inflammation, of interest to cardiologists for predicting future heart attacks. The course web site has a CRP data set (`crpWinBUGS.txt`), which is already in a format useable by WinBUGS. The data represent six CRP values from each of 15 subjects, followed over three years, with measurements taken every six months. There are a few missing values, but WinBUGS will handle these automatically through imputation, which we will learn about in greater detail later.

We start by plotting these data in R. We can use `read.table` to get the data into R, using the same data set formatted for R (need to erase the `[]`'s that follow each variable name, leaving the number inside the `[]`'s, and take the word “END” away). To plot the data, we would like a single line to represent the data from each subject, so want a graph with 15 lines across it, one for each subject. R code such as the following could be used.

```
> plot(1:6,crp.dat[1,], type="l", ylim=c(0,15), xlab="time", ylab="crp")
> for (i in 2:15){ points(1:6,crp.dat[i,], type="l", lty=i)}
```

Note that in using the `lty=i` option, the data from each patient are given a different line type.

The plot is:


```

for(j in 1:6) {                                     # Loop over 6 points per subject
crp[i,j] ~ dnorm(mu[i], tau.within)                # Within each subject, normal with
}                                                    #   mean = mu[i], individual specific
mu[i] ~ dnorm(overall.mu, tau.between)             # Overall normal mean is overall.mu
}
overall.mu ~ dnorm(0, 1.0E-3)                       # Prior for overall mean has variance 1000
tau.within ~ dgamma(0.001, 0.001)                  # Prior for tau=1/var
tau.between ~ dgamma(0.001, 0.001)                 # Could also use uniform on sigma
sigma.within <- 1 / sqrt(tau.within)                # Convert sigma from tau
sigma.between <- 1 / sqrt(tau.between)
}

```

```
# Initial values
```

```
list(tau.within = 1, tau.between = 1, overall.mu = 0)
```

```
# Data Set
```

crp[,1]	crp[,2]	crp[,3]	crp[,4]	crp[,5]	crp[,6]
3.66	2.09	2.15	1.13	1.90	1.33
3.16	3.82	2.48	NA	2.50	2.35
1.23	14.50	3.37	6.13	NA	NA
6.04	6.56	4.72	4.50	7.25	NA
2.05	1.47	2.42	3.93	2.49	6.61
2.62	2.43	2.35	2.85	2.29	2.65
1.82	1.85	2.57	2.32	1.79	1.59
1.13	0.66	1.97	1.19	1.24	1.06
5.52	6.06	5.68	5.85	4.06	4.71
5.07	5.37	3.98	5.78	2.45	2.05
1.98	1.76	1.62	0.96	1.46	1.41
1.40	2.31	2.23	6.62	2.58	2.21
3.71	2.44	3.75	3.56	2.80	3.04
11.50	9.44	10.20	7.68	7.38	7.42
8.57	7.36	8.51	8.05	8.67	9.74

```
END
```

```
# Results after Running WinBUGS
```

node	mean	sd	2.5%	median	97.5%	start	sample
mu[1]	2.198	0.6438	0.9268	2.198	3.448	1001	10000
mu[2]	2.976	0.685	1.604	2.983	4.33	1001	10000
mu[3]	6.043	0.7865	4.506	6.043	7.611	1001	10000
mu[4]	5.645	0.7007	4.263	5.641	7.016	1001	10000
.....	etc

overall.mu	4.002	0.674	2.636	3.997	5.361	1001	10000
sigma.between	2.452	0.5457	1.639	2.365	3.748	1001	10000
sigma.within	1.633	0.1408	1.385	1.622	1.939	1001	10000
tau.between	0.1899	0.07903	0.07134	0.1787	0.3728	1001	10000
tau.within	0.3832	0.06488	0.2663	0.3802	0.5214	1001	10000

These results correspond to expectations, with overall mean around 4, and between subject variance larger than within subject variance.

Situations where random effects or hierarchical models can be used

Repeated measures models: We have just seen an example with several measurements per subject, so called repeated measurement data. Random effects/hierarchical models are especially useful when subjects are followed over time, where each subject will contribute their own regression over time, so each subject will have their own individual intercept and slope. Putting a hierarchical model on the intercepts and/or slopes will improve individual estimates (again, at the expense of a small amount of individual bias, and *decreased* overall mean square error).

Nested data collection: Suppose patients are nested within doctors treating them, the doctors are nested within hospitals where they work, hospitals are nested within health regions, which are nested within cities, which are nested within provinces, which are nested within the same country. A multiple layers hierarchical model can then be used to take into account the complex data structure.

Unknown interactions: A medication, for example, may act differently on each and every subject (indeed, this is almost always the case, although individual variation may be large or small). If factors which may effect how the drug works for a given individual are unknown or unrecorded, then a hierarchical model can be used (as in our introductory example).

In short, in epidemiology, one can argue that just about *every* model should include a hierarchical component, and, indeed, these models are becoming extremely popular.

Conclusion

Hierarchical or Random Effects models are extremely important to epidemiology. We have just touched on a few basic concepts. Hopefully you will see more in future courses.