Hierachical Model

Contents

Motivation

Sometimes we often encounter some grouped data. For example:

- Students belong to different classes;
- Cities belong to different provinces;
- Stores belong to different areas.
- Panel Data

Often times, for these types of data, we have within group correlations: individuals in a group are actually correlated.

Basic framework

suppose the following hierarchical DGP:

$$y_{ik} \sim p_y(\theta_k), \quad \theta_k \sim p_\theta(\beta)$$

How should we estimate this model? Denote $\Theta \equiv (\theta_1, ...\theta_K)$ and $\mathbf{y} \equiv (y_1, , , y_n)$ First notice

$$pr(\Theta, \beta|\mathbf{y}) = \frac{pr(\mathbf{y}|\Theta, \beta)pr(\Theta|\beta)pr(\beta)}{pr(\mathbf{y})}, \quad (1)$$

Hierachical ML

Under this method, we assume β is a determinant, and want to choose β to maximize

$$pr(\beta|\mathbf{y}) \propto \int pr(\mathbf{y}|\Theta, \beta)pr(\Theta|\beta)pr(\beta)d\Theta$$

By maximizing the above equation, we get the estimated parameter $\hat{\beta}$. This method assumes that we do not care about Θ, but we can estimate it using :

$$\hat{\Theta} = argmax_{\Theta} pr(\Theta, \hat{\beta}|\mathbf{y})$$

(question: why don't we directly maximize (1) directly?)

Bayesian Estimation

Under this method, we regard β as following some prior distribution, and do mcmc sampling directly according to (1). In other words, we estimate both Θ and β

Application

Random Intercept Model

Consider the following DGP:

$$y_{ik} = x_{ik}\beta + \alpha_k + \epsilon_{ik}, \quad \epsilon_{ik} \sim N(0, \sigma_\epsilon^2)$$

in which ik means the i individual in group k. α_k are constants. DGP says that we believe that each group has its own intercept.

- We want to estimate $\beta, \alpha_1, ...\alpha_K$. Then we can explicitly estimate by using dummy variables.
- We only care about β. In the case of panel data, we can subtract each individual by the average of the group it belongs to, so α_k disappear.

But we can also regard α_k as unobservable components: different groups have different intercepts, but these intercepts come from a distribution. That is,for example, for all k

$$\alpha_k \sim N(0, \sigma_\alpha^2)$$

One advantage for this setting is that we can know better about the nature of the intercepts by estimating μ and σ.Also, we are estimating less parameters when we assume that α_k is random.

It is natural that we want to estimate σ_α, σ_ϵ, β.

Let's walk through the process of deriving the likelihood function. This is of course applicable for other distribution cases.

$$pr(\sigma_\epsilon, \sigma_\alpha, \beta|\mathbf{y}, \mathbf{x}) \equiv \int pr(\sigma_\epsilon, \beta, \alpha_1, , , , \alpha_K|\mathbf{y}, \mathbf{x}) d(\alpha_1, \alpha_2, ..\alpha_K)$$

$$\propto \int_{\alpha_1, \alpha_2, ..\alpha_K} pr(\mathbf{y}|\mathbf{x}, \sigma_\epsilon, \sigma_\alpha, \beta, \alpha_1, , , , \alpha_K) pr(\alpha_1, ...\alpha_K|\sigma_\alpha) d(\alpha_1, \alpha_2, ..\alpha_K)$$

$$= \int_{\alpha_1, \alpha_2, ..\alpha_K} \left(\prod_{k=1}^{K} pr(\mathbf{y_k}|\mathbf{x_k}, \sigma_\epsilon, \sigma_\alpha, \beta, \alpha_k) \right) \left(\prod_{k=1}^{K} pr(\alpha_k|\sigma_\alpha) \right) d(\alpha_1, \alpha_2, ..\alpha_K)$$

$$= \prod_{k=1}^{K} \int_{\alpha_k} pr(\mathbf{y_k}|\mathbf{x_k}, \sigma_\epsilon, \sigma_\alpha, \beta, \alpha_k) pr(\alpha_k|\sigma_\alpha) d\alpha_k = \prod_{k=1}^{K} pr(\mathbf{y_k}|\mathbf{x_k}, \sigma_\epsilon, \sigma_\alpha, \beta) \quad (2)$$

The third line holds because individuals of different groups are independent. What is the $pr(\mathbf{y_k}|\mathbf{x_k}, \sigma_\epsilon, \sigma_\alpha, \beta)$, then? Suppose there are only two individuals, i and j, in group k. What is the likelihood function for this group? it is

$$pr(\alpha_k + \epsilon_{ik} = y_{ik} - x_{ik}\beta, \alpha_k + \epsilon_{jk} = y_{jk} - x_{jk}\beta)$$

We need to know the joint distribution of $\alpha_i + \epsilon_{ik}$ and $\alpha_k + \epsilon_{jk}$ if we want to know write the expression for this likelihood. It is not difficult. Define $s_{ik} \equiv \alpha_k + \epsilon_{ik}$, $s_{jk} \equiv \alpha_j + \epsilon_{jk}$. We immediately have

$$cov(s_{ik}, s_{jk}) = \sigma_\alpha^2, \quad var(s_{ik}) = var(s_{jk}) = \sigma_\alpha^2 + \sigma_\epsilon^2$$

Therefore, For given group k, the error term of individuals in this group follows a normal distribution, with the covariance matrix as:

- The diagonal elements are all σ_α^2
- The remaining elements are all $\sigma_\alpha^2 + \sigma_\epsilon^2$.

Having this, we can easily write down $pr(\mathbf{y_k}|\mathbf{x_k}, \sigma_\epsilon, \sigma_\alpha, \beta)$ for each k. We then choose the $\sigma_\epsilon, \sigma_\alpha, \beta$ that maximize the whole likelihood function (2).

On the other hand, if we want to do the bayesian estimation, we sample from

$$pr(\sigma_\epsilon, \sigma_\alpha, \beta, \alpha_1, , , \alpha_K|\mathbf{y}, \mathbf{x}) \propto \left(\prod_{k=1}^{K} pr(\mathbf{y_k}|\mathbf{x_k}, \sigma_\epsilon, \sigma_\alpha, \beta, \alpha_k)\right)\left(\prod_{k=1}^{K} pr(\alpha_k|\sigma_\alpha)\right) prior(\beta, \sigma_\alpha, \sigma_\epsilon)$$

in which

$$pr(\mathbf{y_k}|\mathbf{x_k}, \sigma_\epsilon, \sigma_\alpha, \beta, \alpha_k) = \prod_{i=1}^{n_k} pr(\epsilon_{ik} = y_{ik} - x_{ik}\beta - \alpha_k)$$

Random coeffieicnt plus random intercept model

Similar to random intercept model, we also have random coefficient model, i.e., there is difference in the effects of explanatory variable on outcome variables across groups.

$$y_{ik} = x_{ik}(\beta + u_k) + \alpha_k + \epsilon_{ik}$$

in which $\epsilon \sim (0, \sigma_\epsilon)$ and

$$\begin{pmatrix} u_k \\ \alpha_k \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \rho\sigma_u\sigma_\alpha \\ \rho\sigma_u\sigma_\alpha & \sigma_\alpha^2 \end{pmatrix}\right)$$

Following the same reasoning in the previous section, for two individuals i, j in the same group k, the likelihood function is

$$pr(u_k x_{ik} + \alpha_k + \epsilon_{ik} = y_{ik} - \beta x_{ik}, u_k x_{jk} + \alpha_k + \epsilon_{jk} = y_{jk} - \beta x_{jk})$$

Define

$$s_q \equiv u_k x_{qk} + \alpha_k + \epsilon_{qk}, \quad q = i, j$$

We immediately have

$$cov(s_{ik}, s_{jk}) = cov(u_k x_{ik} + \alpha_k + \epsilon_{ik}, u_k x_{jk} + \alpha_k + \epsilon_{jk}) = var(\alpha_k) + x_{ik}x_{jk}var(u_k) + (x_{ik} + x_{jk})cov(u_k, \alpha_k)$$

$$= \sigma_\alpha^2 + x_{ik}x_{jk}\sigma_u^2 + (x_{ik} + x_{jk})\rho\sigma_u\sigma_\alpha$$

and

$$\begin{aligned} var(s_{ik}) = var(u_k x_{ik} + \alpha_k + \epsilon_{ik}) &= var(u_k x_{ik} + \alpha_k) + var(\epsilon_{ik}) \\ &= x_{ik}^2\sigma_u^2 + \sigma_\alpha^2 + x_{ik}\rho\sigma_u\sigma_\alpha + \sigma_\epsilon^2 \end{aligned}$$

Examples:

We now consider a case of poisson data generation

$$r_k \sim N(0, \sigma_r^2), \quad \tau_k \sim N(0, \sigma_\tau^2)$$

$$y_{ik} \sim Poisson(e^{\beta_0 + (\beta_1 + \tau_k)x_{ik} + r_k})$$

## preparation

We use a fishing data as an example. Here, x is temperate, y is fish_num

```
library(brms)
library(rstan)
library(lme4)
library(dplyr)
df_raw =read.csv("./fish.txt", sep=",")%>%
  mutate(weather = ifelse (weather == 'sunny', 1, 0))
head(df_raw)
```

```
##   fish_num weather temperature id
## 1        0       0         5.0  1
## 2        1       0        24.2  2
## 3        6       0        11.5  3
## 4        0       0         9.8  4
## 5        1       0        18.1  5
## 6        1       0        18.1  6
```

Bayeisn Estimation

Using the brms packages, we can easily do estimation. Several things to notice:

- The formula expression may seem not that the intuitive. Look at the formula below. the '1+ temperature' indicates we add the constant and temperature into the model as explanatory variables. $(temperature|weather)$ means that we assume that across the levels of weather, both the coefficients of temperature and and the intercept are different, and both randomly taken out from some distributions. If we only incorporate random intercept, then use $(1|weather)$. (It seems to say that actually we cannot assume ONLY random coefficient?).
- $(temperature|weather)$ and $(temperature||weather)$ are slightly different. The former admits that τ_k and r_k are correlated, while the latter does not.
- For other usages related to brms, check my another notes 'Bayesian Estimation with brms', which talks about other post-estimation analysis

```
bayesian_res = brm (fish_num ~ 1 + temperature + ( temperature | weather),
                    data= df_raw,
                    family = poisson(),# default is gaussian
                    prior   = NULL,
                    seed = 1,
                    chains = 4,
                    iter = 4000,
                    warmup = 2000)
```

```
##
## SAMPLING FOR MODEL '036fce67b0dd4786cec7204e83f0d29a' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 1: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 1: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 1: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 1: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 1: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 1: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 1: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 1: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 1: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 1: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 1: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 8.13 seconds (Warm-up)
## Chain 1:                2.768 seconds (Sampling)
## Chain 1:                10.898 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '036fce67b0dd4786cec7204e83f0d29a' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 2: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 2: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 2: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 2: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 2: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 2: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 2: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 2: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 2: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 2: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 2: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 7.384 seconds (Warm-up)
## Chain 2:                5.811 seconds (Sampling)
## Chain 2:                13.195 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '036fce67b0dd4786cec7204e83f0d29a' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 3: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 3: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 3: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 3: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 3: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 3: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 3: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 3: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 3: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 3: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 3: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 7.265 seconds (Warm-up)
## Chain 3:                6.607 seconds (Sampling)
## Chain 3:                13.872 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '036fce67b0dd4786cec7204e83f0d29a' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 4000 [  0%]  (Warmup)
## Chain 4: Iteration:  400 / 4000 [ 10%]  (Warmup)
## Chain 4: Iteration:  800 / 4000 [ 20%]  (Warmup)
## Chain 4: Iteration: 1200 / 4000 [ 30%]  (Warmup)
## Chain 4: Iteration: 1600 / 4000 [ 40%]  (Warmup)
## Chain 4: Iteration: 2000 / 4000 [ 50%]  (Warmup)
## Chain 4: Iteration: 2001 / 4000 [ 50%]  (Sampling)
## Chain 4: Iteration: 2400 / 4000 [ 60%]  (Sampling)
## Chain 4: Iteration: 2800 / 4000 [ 70%]  (Sampling)
## Chain 4: Iteration: 3200 / 4000 [ 80%]  (Sampling)
## Chain 4: Iteration: 3600 / 4000 [ 90%]  (Sampling)
## Chain 4: Iteration: 4000 / 4000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 8.805 seconds (Warm-up)
## Chain 4:                5.863 seconds (Sampling)
## Chain 4:                14.668 seconds (Total)
## Chain 4:
```

```
summary(bayesian_res)
```

```
##  Family: poisson
##   Links: mu = log
## Formula: fish_num ~ 1 + temperature + (temperature | weather)
##    Data: df_raw (Number of observations: 100)
## Samples: 4 chains, each with iter = 4000; warmup = 2000; thin = 1;
##          total post-warmup samples = 8000
##
## Group-Level Effects:
## ~weather (Number of levels: 2)
##                            Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
## sd(Intercept)                  1.39      1.08     0.11     4.02 1.02      260
## sd(temperature)                0.08      0.10     0.00     0.42 1.07       40
## cor(Intercept,temperature)     0.04      0.59    -0.95     0.96 1.01      531
##                            Tail_ESS
## sd(Intercept)                  1307
## sd(temperature)                  14
## cor(Intercept,temperature)     2215
##
## Population-Level Effects:
##             Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept      -0.13      0.84    -1.89     1.54 1.03      113      487
## temperature     0.06      0.04    -0.02     0.15 1.06       84       80
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

MLE

Use lme4 Package to realize the MLE estimation. The command is super simple. check the the package documentation for more details.

```
mle_res <- lmer(fish_num ~ 1 + temperature + ( 1 | weather),
                data =  df_raw,
                family = poisson(), # how is y generated
                REML      = FALSE,
                na.action = na.omit)
```

Reference

- http://ryotamugiyama.com/wp-content/uploads/2016/01/hierarchicalbeyes.html#three-level-multi-level-model-3
- https://idiom.ucsd.edu/~rlevy/pmsl_textbook/chapters/pmsl_8.pdf
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2601029/#FD5