Random intercept + slope: estimating variance of standardized effects

Han Bossier

7-2-2018

Contents

1 Introduction

In this report, we simulate 3-level data using a linear mixed effects model with a random slope for each study. We try to calculate the true value of the between-study variability of the standardized effect size. We then compare these with observed values using a Monte-Carlo simulation study. We generate data for just one voxel, at subject level (i.e. a time series), for each study in a meta-analysis. Data does not contain any smoothing and noise is random (i.e. white noise). We use 5000 Monte-Carlo simulations.

Some general parameters for this report:

- 50 subjects per study
- 80 studies per MA
- No between-subject heterogeneity
- With between-study heterogeneity

2 Theory

First we discuss the underlying model. Then how we generate data.

2.1 Statistical model

Consider the following linear model:

$$Y_{ij} = \beta_0 + b_{1j} + (\beta_1 + b_{2j})X + \varepsilon_{ij},$$

where Y_{ij} is a vector of length T (number of time points) containing the BOLD signal for subject i in study j. Furthermore, we have $\varepsilon \sim N(0, \sigma^2)$, a random intercept $b_{1j} \sim N(0, 1)$ and the random slope $b_{2j} \sim N(0, \eta^2)$. We set $\beta_0 = 100$, $\beta_1 = 3$, $\sigma^2 = 100^2$, $\eta^2 = 50^2$ and X is a design matrix obtained by convoluting an ON/OFF blocked design with a canonical HRF. There is no correlation between the random intercepts and slopes.

2.2 Data generation

First we set some global variables.

```
# number of simulations
nsim <- 5000
# number of subjects and studies
nsub <- 50
# nstud <- 50
nstud <- 80

# Value for sigma in the model
sigma_m <- 100

# Value for eta in the model
# eta_m <- 10
# eta_m <- 50
eta_m <- 25
```

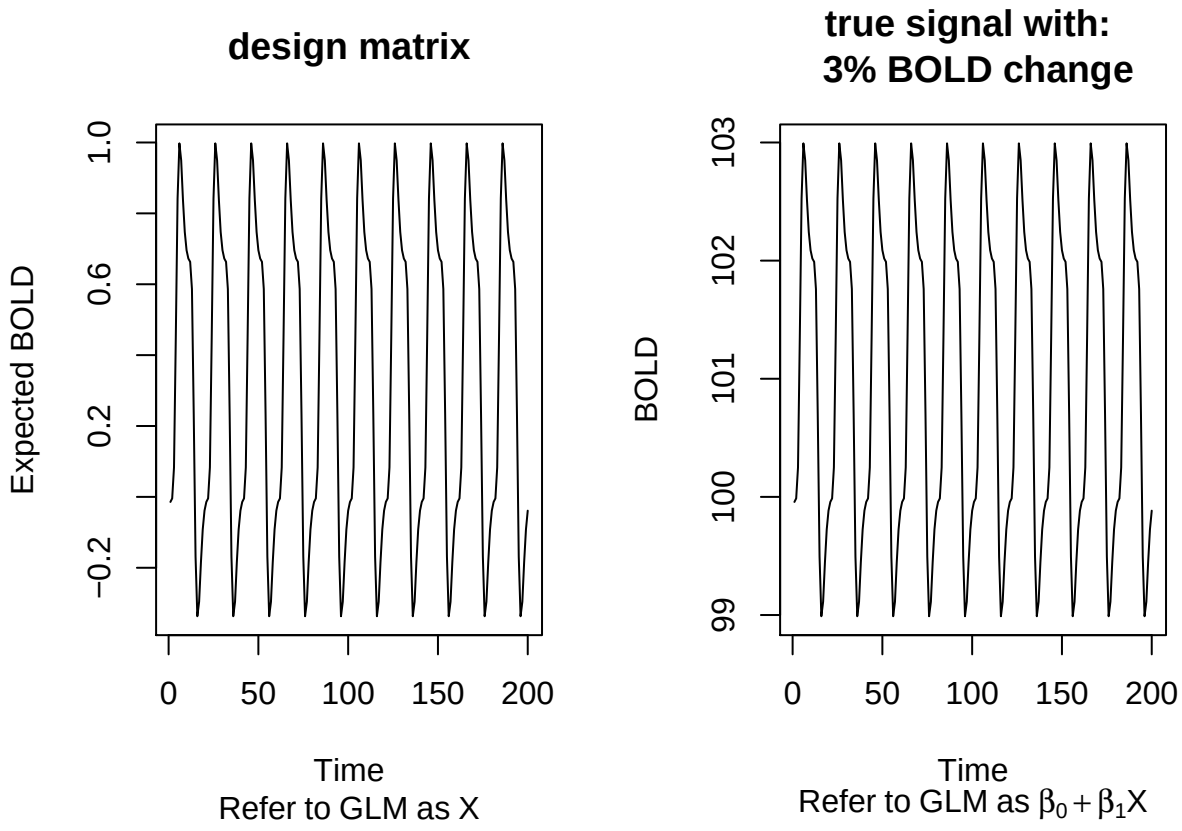
2.2.1 Design matrix

To obtain X, we use *neuRosim*. In the section/code below, β_0 is denoted as *base*. We begin by setting the following parameters:

- TR
- number of scans
- onsets of blocks ON
- duration of blocks (in sec)

Then we use the functions `neuRosim::simprepTemporal` to generate the parameters and `neuRosim::simTSfmri` to generate the time series of the design matrix. This results in a vector X of length T.

```
# Signal characteristics
TR <- 2
nscan <- 200
total <- TR*nscan
on1 <- seq(1,total,40)
onsets <- list(on1)
duration <- list(20)

# true %BOLD change
BOLDC <- 3

# Base/intercept of signal
base <- 100

# Generating a design matrix: convolution of block design with double-gamma HRF
X <- neuRosim::simprepTemporal(total,1,onsets = onsets,
                               effectsize = 1, durations = duration,
                               TR = TR, acc = 0.1, hrf = "double-gamma")

# X vector for one subject = predicted signal
X_s <- neuRosim::simTSfmri(design=X, base=0, SNR=1, noise="none", verbose=FALSE)

## Design parameters
# Extend the design matrix with the intercept
xIN <- cbind(base,X_s)

# Contrast: not interested in intercept
CONTRAST <- matrix(c(0,1),nrow=1)

# Calculate (X'X)^(-1) with contrast
design_factor <- CONTRAST %*% (solve(t(xIN) %*% xIN )) %*% t(CONTRAST)

# Plot
par(mfrow = c(1,2))
plot(X_s, type = 'l', main = 'design matrix',
     sub = expression(Refer ~ to ~ GLM ~ as ~ X),
     ylab = 'Expected BOLD',
     xlab = 'Time')
plot(base + BOLDC * X_s, type = 'l', main =
       paste0('true signal with: \n ', BOLDC,'% BOLD change'),
     sub = expression(Refer ~ to ~ GLM ~ as ~ beta[0] + beta[1] * X),
    ylab = 'BOLD', xlab = 'Time')
```



2.2.2 Data generation

We are now ready to simulate data. First we create an empty vector Y, then generate a value for b_j, loop over all subjects in this study, add to the data frame and finally loop again to the next study. As an example in **R**:

```
# Empty vector
Y <- data.frame() %>% as_tibble()

# Generate D matrix: variance-covariance matrix of random intercept + slope
# Variance of slope = eta_m**2
var_cov_D <- rbind(c(1.0**2, 0), c(0, eta_m**2))
# Generate values using this D-matrix for intercept and slope per study
B_matrix <- MASS::mvrnorm(nstud, mu=c(0,0), Sigma=var_cov_D)

# For loop over studies
for(t in 1:nstud){
  # Take values for b_slope and b_int
  b_int <- B_matrix[t,1]
  b_sl <- B_matrix[t,2]

  # For loop over all subjects
  for(i in 1:nsub){
  # Generate nscan values, corresponding to time series of one subject
    # within a study
  Y_s <- base + b_int + ((BOLDC + b_sl) * X_s) + rnorm(n = nscan, mean = 0, sd = sigma_m)
```

```
  # Add to data frame
  Y <- data.frame(Y = Y_s, X = X_s, sub = i, stud = t) %>% as_tibble() %>%
    bind_rows(Y, .)
  }
}

# Make factors of subjects and studies
Y$stud <- as.factor(Y$stud)
Y$sub <- as.factor(Y$sub)
```

2.3 Statistical analysis

2.3.1 Random intercept + slope model

First we fit a linear model on Y with a random intercept for subject, a random intercept for study and a random slope for the effect of X on Y from each study. This corresponds to:

```
summary(lmer(Y ~ 1 + X + (1 | sub) + (1 + X|stud), data = Y))
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: Y ~ 1 + X + (1 | sub) + (1 + X | stud)
   Data: Y

REML criterion at convergence: 6022290

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.4900 -0.6743  0.0015  0.6744  5.9585

Random effects:
 Groups   Name        Variance  Std.Dev.  Corr
 sub      (Intercept) 3.984e-10 1.996e-05
 stud     (Intercept) 1.686e-03 4.106e-02
          X           8.972e+01 9.472e+00 1.00
 Residual             9.960e+03 9.980e+01
Number of obs: 500000, groups:  sub, 50; stud, 50

Fixed effects:
            Estimate Std. Error t value
(Intercept)  99.8367     0.1763   566.4
X            -0.2329     1.3769    -0.2

Correlation of Fixed Effects:
  (Intr)
X -0.106
convergence code: 0
unable to evaluate scaled gradient
Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

2.3.2 Standardized effect sizes

Using a random effects meta-analysis, our goal is to summarise the standardized effect sizes from each study.

Standardized effect sizes (such as Cohen's d) at the second level (study level) are generally calculated by dividing the mean effect with the standard deviation. In our case, this corresponds to:

$$\mathrm{d} = \frac{\hat{\beta}^*}{\sigma\sqrt{(X'X)^{-1}}}. \tag{1}$$

We are mainly interested in Hedges' g. This is obtained by multiplying d with a correction factor J:

```
NeuRRoStat::corrJ
```

```
function(N){
  1-(3/((4*(N-1))-1))
}
<environment: namespace:NeuRRoStat>
```

The expected value of Hedges' g over all simulations is equal to:

$$\frac{\sum_{m=1}^{M} \mathrm{g}_m}{M} = \frac{3}{\sigma\sqrt{(X'X)^{-1}}} \times J. \tag{2}$$

2.4 Third level

The estimated value at the third level (meta-analysis) corresponds with the weighted average of all standardized effect sizes at the second level. We use a random-effects model with the method of moments estimator for between-study heterogeneity. The weights correspond to the inverse of the sum of within- and between study variability. As we use the same amount of subjects for each study, the expected value of the within-study variability will be equal for each study. Hence asymptotically, all weights are the same and the weighted average is equal to an unweighted average. Hence the true value for μ, the population effect is equal to:

$$\begin{aligned} \mu &= E(g) \\ &= g, \end{aligned} \tag{3}$$

as $E(g)$ is a constant in our set-up.

3 Simulation Results

We saved the following objects in each simulation:

- the estimated β parameter of the linear model with random effects (LME).
- the estimated parameter for $\mathrm{var}(b_s)$ using REML and the LME.
- the standardized weighted effect size of the random effects meta-analysis.
- the method of moments estimator for between-study variability (τ^2) of the standardized effect size.

First we read in the **R** objects.

```
LocBase <- '/Volumes/2_TB_WD_Elements_10B8_Han/PhD/Simulation/Results'
# Location of data (comments are other versions)
# LocDat <- paste0(LocBase, '/VectorMA_RanSlope/Eta_10')
# LocDat <- paste0(LocBase, '/VectorMA_RanSlope/Eta_50_nsim_5000')
LocDat <- paste0(LocBase, '/VectorMA_RanSlope/Eta_25_nstud_80')
```

```r
# Empty data frames
comb_res <- data.frame() %>% as_tibble()

# For loop over the split-up simulations (every .rda file contains 10 simulations)
for(i in 1:c(nsim/10)){
  # Read in data
  comb_res <- readRDS(paste0(LocDat, '/LMER_', i, '.rda')) %>%
    bind_rows(comb_res,. )
}
```

Now let us summarise the values over all simulations.

```r
# MAX <- matrix(1, nrow = nstud)
# MACt <- solve(t(MAX) %*% MAX)
# sd_beta <- (sigma_m * sqrt(MACt))

sd_beta <- (sigma_m * sqrt(design_factor))

options(scipen = 2)
comb_res %>%
  mutate(EstTau2 = EstTau**2) %>%
  group_by(parameter, model) %>%
  summarise(AvgEst = mean(estimate),
            AvgEta2 = mean(EstTau2)) %>%
  ungroup() %>%
  mutate(TrueEst = c(BOLDC, BOLDC/sd_beta  * corrJ(N = nsub))) %>%
  mutate(TrueEta2 = c(eta_m**2, eta_m**2 * sd_beta^(-2))) %>%
  dplyr::select(parameter, model, AvgEst, TrueEst, AvgEta2, TrueEta2) %>%
  knitr::kable(., digits = 4, longtable = TRUE, booktabs = TRUE)
```

parameter	model	AvgEst	TrueEst	AvgEta2	TrueEta2
BETA	LMER	3.1224	3.0000	620.5558	625.0000
STAN_ES	MA	0.1848	0.1852	1.1477	2.4569

4 Conclusion

My intuition is that the measure of heterogeneity estimated by the random effects meta-analysis is scale invariant. While the estimated variance of the random effects is not. At the moment, I do not know how to relate these two. I will try to simulate using a different approach.