

Mixed Effects GLM in fMRI

Han Bossier

28/2/2019

Introduction

Generate fMRI time series using a full mixed effects GLM model. We estimate the fixed- and random-effects parameters using *lmer* to see whether we have unbiased estimates.

Some general parameters:

```
# Number of batches run  
nbatch <- 100
```

Data

Consider the following linear model for a subject $i = 1, \dots, K$:

$$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{u}_i + \boldsymbol{\varepsilon}_i \quad (1)$$

$$\mathbf{u}_i \sim N(\mathbf{0}, \mathbf{D}) \quad (2)$$

$$\boldsymbol{\varepsilon}_i \sim N(\mathbf{0}, \sigma_w^2 I), \quad (3)$$

where:

- \mathbf{Y}_i is the response vector of dimension $T_i \times 1$ where T_i is the number of time points for subject i
- \mathbf{X}_i is the $T_i \times p$ model matrix for the fixed effects with p the number of predictors including the intercept
- $\boldsymbol{\beta}$ is the $p \times 1$ vector of fixed effects coefficients
- \mathbf{Z}_i is the $T_i \times q$ model matrix for the random effects where q equals the number of random effect parameters
- \mathbf{u}_i is the $q \times 1$ vector of random effect coefficients
- $\boldsymbol{\varepsilon}_i$ is the $T_i \times 1$ vector of error terms

One can combine all subjects into:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\varepsilon} \quad (4)$$

We will generate data using one continuous predictor (i.e. an ON/OFF block design) and two random effects (one for the intercept and one for the effect of X on Y, $q = 2$). Hence, the $q \times q$ variance-covariance matrix of the random effects (\mathbf{D}) equals:

$$\mathbf{D} = \begin{bmatrix} \sigma_{b0}^2 & 0 \\ 0 & \sigma_{b1}^2 \end{bmatrix} \quad (5)$$

In this report, we set $\sigma_{b0}^2 = 0$ and $\sigma_{b1}^2 = 4$. Note that we generate data with homogeneous error variances where we set $\sigma_w^2 = 16$. We will have 40 subjects ($K = 40$) and 100 scans for each subject ($n_i = 50$ for all $i = 1, \dots, K$). Furthermore, we set $\beta_0 = 100$ and $\beta_1 = 3$ in the $\boldsymbol{\beta}$ vector.

For the Monte-Carlo simulations, we will generate the entire \mathbf{u} matrix beforehand. Hence in **R**, we have the following true (parameter) values and \mathbf{u} :

```
# subject
nsub <- 40

# fMRI paradigm: block design 10s ON/OFF + 100 scans
nscans <- 100
tr <- 2
total.time <- nscans * tr
dur <- 10
onsets <- seq(1, total.time, dur * 2)

# Fixed effects parameters
beta0 <- 100
beta1 <- 3

# Random effects parameters
sigma_e <- 4
sigmab0 <- 0
sigmab1 <- 2

# Generate the variance covariance matrix of the random effects
var_cov_U <- rbind(c(sigmab0**2, 0), c(0, sigmab1**2))
# Generate the values for b0 and b1
B_matrix <- MASS::mvrnorm(nsub, mu = c(0,0), Sigma = var_cov_U)

# General X list with the design matrix specification
Xgen <- simprepTemporal(totaltime = total.time,
                        regions = 1,
                        onsets = onsets,
                        effectsize = beta1,
                        durations=dur,
                        TR = tr,
                        acc=0.1, hrf="double-gamma")

# Predicted signal
pred <- simTSfmri(design = Xgen,
                  base = 0,
                  SNR = 1,
                  noise = "none", verbose = FALSE)
```

Variance of Fixed Effects Parameters

To estimate the variances of the fixed effect parameters (β), we first define the variance-covariance matrix of the observed responses for subject i , $\text{Var}(\mathbf{Y}_i)$:

$$\text{Var}(\mathbf{Y}_i) = \mathbf{V}_i = \mathbf{Z}_i \mathbf{D} \mathbf{Z}_i' + \sigma_w^2 \mathbf{I} \quad (6)$$

Then the variance-covariance matrix of the fixed effect parameters is given by:

$$\text{Var}(\beta) = \left(\sum_{i=1}^K \mathbf{X}_i \mathbf{V}^{-1} \mathbf{X}_i \right)^{-1}. \quad (7)$$

In order to calculate the true variance-covariance matrix of our fixed effect parameters, we hence need to generate all responses in each Monte-Carlo simulation run beforehand. In **R**, this is:

```
# First we create empty vectors for X and V, as well as an empty matrix
# for the variance-covariance matrices of the fixed effects.
ComplX <- ComplV <- matrix(NA, nrow = 1, ncol = 1)
VarCovBeta_raw <- matrix(0, ncol = 2, nrow = 2)
Xlist <- Zlist <- list()

# Pre-define the true variance-covariance matrix for fixed effects parameters
for(i in 1:nsub){
  # Predictor for this subject
  X <- cbind(1, pred)

  # Z-matrix for this subject
  Z <- X

  # V-matrix
  V <- Z %*% var_cov_U %*% t(Z) +
    diag(sigma_e**2, nscans)

  # Part of var-covar-beta matrix
  VarCovBeta_raw <- VarCovBeta_raw + t(X) %*% solve(V) %*% X

  # Save X and Z
  Xlist[[i]] <- X
  Zlist[[i]] <- Z
}
```

While summing in each class, we get the final true variance covariance matrix for the fixed effects parameters:

```
# Now calculate true variance-covariance matrix
VarCovBeta <- solve(VarCovBeta_raw)
VarCovBeta

              pred
      0.005825895 -0.001807453
pred -0.001807453  0.101789199

# Standard error of beta = sqrt(var(beta))
SEBeta <- data.frame('term' = c('(Intercept)', 'X'),
                     'TrueSE' = sqrt(diag(VarCovBeta)),
                     stringsAsFactors = FALSE)
SEBeta

      term    TrueSE
(Intercept) 0.07632755
pred        X 0.31904420
```

Monte-Carlo

We run the following code in the HPC, and save the estimates for the fixed and random effects parameters. Just to show one simulation, I paste the code here for just one loop.

```
# Just one loop to show the code
startIndex <- endIndex <- 1

# Empty data frame with simulation results
FitTotDat <- data.frame() %>% as_tibble()

# For loop over the simulations
for(r in startIndex:endIndex){
  # Set starting seed
  starting.seed <- pi*r
  set.seed(starting.seed)

  # Empty data frame
  TotDat <- data.frame()

  # Loop over the subjects
  for(i in 1:nsub){
    # Generate data using:  $X\beta + Z\mu + e$ 
    dat <- Xlist[[i]] %*% matrix(c(beta0, beta1), ncol = 1) +
      Zlist[[i]] %*% matrix(c(B_matrix[i,1], B_matrix[i,2]), ncol = 1) +
      rnorm(n = nscans, mean = 0, sd = sigma_e)

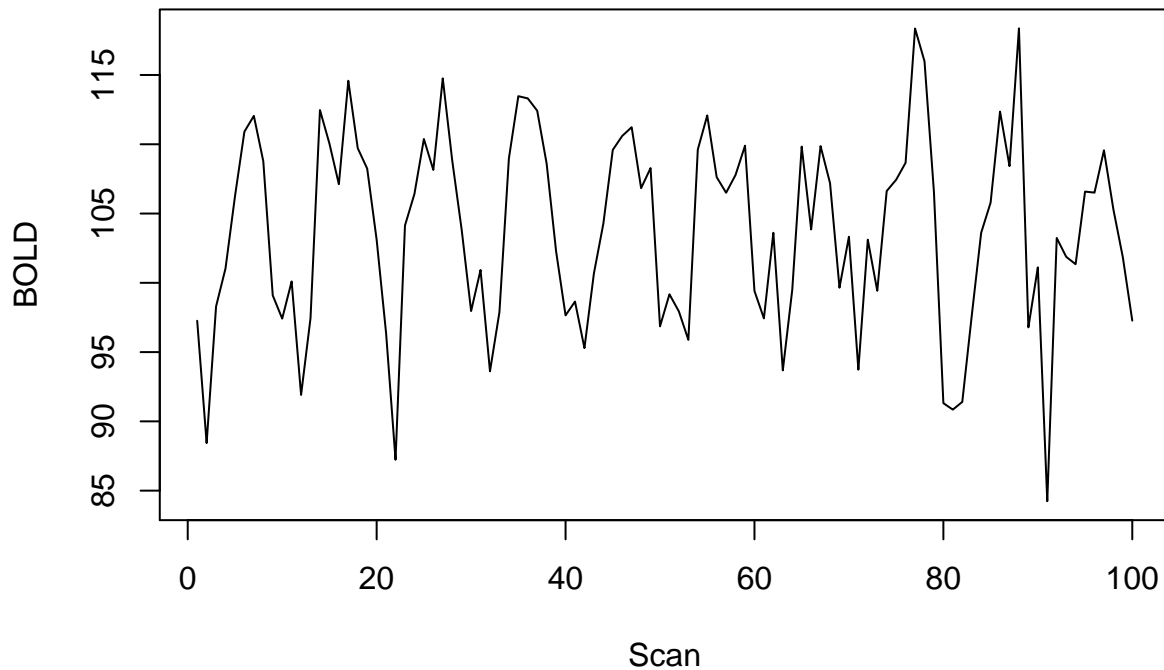
    # Add to data frame
    TotDat <- data.frame(Y = dat, X = Xlist[[i]][,2], subj = i) %>% as_tibble() %>%
      bind_rows(TotDat,.)
  }

  # Analysis
  fit <- lmer(Y ~ 1 + X + (1 + X|subj), data = TotDat, REML = TRUE)
  FitTotDat <- broom::tidy(fit) %>%
    # Add true SE
    left_join(.,SEBeta, by = 'term') %>%
    mutate(sim = r) %>%
    bind_rows(FitTotDat,.)
}
```

This is an example of one time series for one subject:

```
plot(dat, type = 'l', main = 'Example BOLD response for one subject',
      ylab = 'BOLD', xlab = 'Scan')
```

Example BOLD response for one subject



And the estimated GLM parameters:

FitTotDat

```
# A tibble: 6 x 7
  term                estimate std.error statistic group   TrueSE   sim
  <chr>              <dbl>    <dbl>    <dbl> <chr>   <dbl> <int>
1 (Intercept)        99.9      0.0767   1303. fixed    0.0763     1
2 X                   2.91      0.282     10.3 fixed    0.319     1
3 sd_(Intercept).subj 0         NA        NA    subj    NA       1
4 sd_X.subj           1.77      NA        NA    subj    NA       1
5 cor_(Intercept).X.su~ NaN       NA        NA    subj    NA       1
6 sd_Observation.Residu~ 4.02     NA        NA    Residu~ NA       1
```

Analysis

Here we read in the saved data.

```
# Empty data frame with simulation results
FitTotDat <- data.frame() %>% as_tibble()

# For loop over the batches (containing multiple simulation runs)
for(i in 1:nbatch){
  FitTotDat <-
    readRDS(file = paste(locDat, '/fMRMixEffglm_', i, '.rda', sep = '')) %>%
    bind_rows(FitTotDat, .)
}
```

Fixed effects

First let us check the fixed effects parameters.

```
FitTotDat %>%
  filter(term %in% c('(Intercept)', 'X')) %>%
  # Add true estimate
  left_join(., data.frame(term = c('(Intercept)', 'X'),
                        TrueEst = c(beta0, beta1), stringsAsFactors = FALSE),
            by = 'term') %>%
  group_by(term) %>%
  summarise(AvgEst = mean(estimate),
            TrueEst = mean(TrueEst),
            AvgSE = mean(std.error),
            TrueSE = mean(TrueSE))
```

```
# A tibble: 2 x 5
  term      AvgEst TrueEst AvgSE TrueSE
<chr>      <dbl>   <dbl> <dbl> <dbl>
1 (Intercept) 100.     100 0.0793 0.0763
2 X           2.95      3 0.314 0.319
```

Random effects

And then the random effects.

```
FitTotDat %>%
  filter(term %in% c('sd_Observation.Residual',
                    'sd_(Intercept).subj',
                    'sd_X.subj')) %>%
  dplyr::select(term, estimate, sim) %>%
  left_join(., data.frame('term' = c('sd_Observation.Residual',
                                    'sd_(Intercept).subj',
                                    'sd_X.subj'),
                        'TrueSD_ran' = c(sigma_e, sigmab0, sigmab1),
                        stringsAsFactors = FALSE),
            by = 'term') %>%
  group_by(term) %>%
  summarise(AvgEst = mean(estimate),
            AvgTrueSD = mean(TrueSD_ran))
```

```
# A tibble: 3 x 3
  term      AvgEst AvgTrueSD
<chr>      <dbl>   <dbl>
1 sd_(Intercept).subj 0.0930      0
2 sd_Observation.Residual 4.00      4
3 sd_X.subj          1.96      2
```