JointScoreTest Package

Jingchunzi (Jing) Shi, Seunggeun (Shawn) Lee

June 8, 2018

1 Overview

The JointScoreTest package contains functions to implement the score test for jointly testing the fixed and random effects in generalized linear mixed models (GLMMs).

2 The score test for jointly testing the fixed and random effects in a GLMM without nuisance random effects in the null model.

The goal is to test $H_0 : \beta = 0,\ \tau = 0$ in the GLMM

$$g(\mu) = X\beta + S\alpha + Ua, \quad a \sim N(0, \tau\Sigma);$$

where $g(\cdot)$ is a known link function; $\mu = (\mu_{11}, \ldots, \mu_{1n_1}, \ldots, \mu_{m1}, \ldots, \mu_{mn_m})$ is the vectorized conditional mean μ_{ij} of the outcome variable y_{ij} for the jth observation in the ith cluster ($i \in \{1, \ldots, m\}, j \in \{1, \ldots, n_i\}$); S is the design matrices for the fixed effects, to be adjusted for; X and U are the design matrices of interest for the fixed and random effects, respectively, to be tested for; τ is a non-negative scalar and Σ is a pre-specified scaled covariance matrix.

```
> ###########################################################################
> ####### To run this code, first download and unzip example files
> ###########################################################################
>
>
> #############################################################
> ## Joint testing of the fixed and random effects in a GLMM without nuisance
> ## random effects in the null model, the outcome varianble is of the
> ## continuous response type
> ###################################################################
>
> library(JointScoreTest)
> y = as.vector(read.table('CrossSectional_Gaussian_y.txt'))
> X = as.matrix(read.table('CrossSectional_Gaussian_X.txt'))
> S = as.matrix(read.table('CrossSectional_Gaussian_S.txt'))
> U = as.matrix(read.table('CrossSectional_Gaussian_U.txt'))
> SIGMA = as.matrix(read.table('CrossSectional_Gaussian_SIGMA.txt'))
> ##########
>
> CrossSectional_Gaussian = list()
> CrossSectional_Gaussian$y = y[,1]
> CrossSectional_Gaussian$X = X
> CrossSectional_Gaussian$S =S
> CrossSectional_Gaussian$U = U
> CrossSectional_Gaussian$SIGMA = SIGMA
>
>

> JointScoreTest_withoutNuisanceVC(CrossSectional_Gaussian$y, CrossSectional_Gaussian$X,
+         CrossSectional_Gaussian$S, CrossSectional_Gaussian$U,
+         CrossSectional_Gaussian$SIGMA, out_type = "C")

$pvalue
[1] 0.0182905

$opt.rho
[1] 0

$score.each
[1] 3.809544 3.807964 3.804680 3.793937 2.423372

>
>

> #############################################################
> ## Joint testing of the fixed and random effects in a GLMM without nuisance
> ## random effects in the null model, the outcome varianble is of
> ## the binomial response type
> ###################################################################
>
> y = as.vector(read.table('CrossSectional_Binom_y.txt'))
> X = as.matrix(read.table('CrossSectional_Binom_X.txt'))
> S = as.matrix(read.table('CrossSectional_Binom_S.txt'))
> U = as.matrix(read.table('CrossSectional_Binom_U.txt'))
> SIGMA = as.matrix(read.table('CrossSectional_Binom_SIGMA.txt'))
> ############
>
> CrossSectional_Binom = list()
> CrossSectional_Binom$y = y[,1]
> CrossSectional_Binom$X = X
> CrossSectional_Binom$S =S
> CrossSectional_Binom$U = U
> CrossSectional_Binom$SIGMA = SIGMA
>
> JointScoreTest_withoutNuisanceVC(CrossSectional_Binom$y, CrossSectional_Binom$X,
+         CrossSectional_Binom$S, CrossSectional_Binom$U,
+         CrossSectional_Binom$SIGMA, out_type = "D", binom_size = 6)

$pvalue
[1] 0.009117854

$opt.rho
[1] 0

$score.each
[1] 4.616342 4.614419 4.610024 4.593029 3.103386

>
>
```

3 The score test for jointly testing the fixed and random effects in a GLMM with nuisance random effects in the null model.

The goal is to test $H_0 : \beta = 0,\ \tau = 0$ in the GLMM

$$g(\mu) = X\beta + S\alpha + Ua + Zb, \quad a \sim N(0, \tau\Sigma),\ b \sim N(0, D(\theta));$$

where $g(\cdot)$ is a known link function; $\mu = (\mu_{11}, \ldots, \mu_{1n_1}, \ldots, \mu_{m1}, \ldots, \mu_{mn_m})$ is the vectorized conditional mean μ_{ij} of the outcome variable y_{ij} for the jth observation in the ith cluster ($i \in \{1, \ldots, m\}, j \in \{1, \ldots, n_i\}$); S and Z are the design matrices for the fixed and random effects, respectively, to be adjusted for; X and U are the design matrices of interest for the fixed and random effects, respectively, to be tested for; τ is a non-negative scalar and Σ is a pre-specified scaled covariance matrix, and θ is a nuisance variance component vector.

```
> ##############################################################
> ## Joint testing of the fixed and random effects in a GLMM with nuisance
> ## random effects in the null model, the outcome varianble is of the
> ## continuous response type
> ####################################################################
>
> y = as.vector(read.table('Longi_Gaussian_y.txt'))
> X = as.matrix(read.table('Longi_Gaussian_X.txt'))
> S = as.matrix(read.table('Longi_Gaussian_S.txt'))
> U = as.matrix(read.table('Longi_Gaussian_U.txt'))
> Z = as.matrix(read.table('Longi_Gaussian_Z.txt'))
> SIGMA = as.matrix(read.table('Longi_Gaussian_SIGMA.txt'))
> group = as.matrix(read.table('Longi_Gaussian_group.txt'))
> ############
>
> Longi_Gaussian = list()
> Longi_Gaussian$y = y[,1]
> Longi_Gaussian$X = X
> Longi_Gaussian$S =S
> Longi_Gaussian$U = U
> Longi_Gaussian$Z = Z
> Longi_Gaussian$SIGMA = SIGMA
> Longi_Gaussian$group = group
>
> JointScoreTest_withNuisanceVC(Longi_Gaussian$y, Longi_Gaussian$X,
+         Longi_Gaussian$S, Longi_Gaussian$U,
+         Longi_Gaussian$Z, Longi_Gaussian$group,
+         Longi_Gaussian$SIGMA, out_type = "C")

$pvalue
[1] 0.02859903

$opt.rho
[1] 0.75

$score.each
[1] 3.226846 3.231781 3.241346 3.267740 3.110903

> #############################################################
> ## Joint testing of the fixed and random effects in a GLMM with nuisance
> ## random effects in the null model, the outcome varianble is of the
> ## binomial response type
> ###################################################################
>
> y = as.vector(read.table('Longi_Binom_y.txt'))
> X = as.matrix(read.table('Longi_Binom_X.txt'))
> S = as.matrix(read.table('Longi_Binom_S.txt'))
> U = as.matrix(read.table('Longi_Binom_U.txt'))
> Z = as.matrix(read.table('Longi_Binom_Z.txt'))
> SIGMA = as.matrix(read.table('Longi_Binom_SIGMA.txt'))
> group = as.matrix(read.table('Longi_Binom_group.txt'))
> ############
>
> Longi_Binom = list()
> Longi_Binom$y = y[,1]
> Longi_Binom$X = X
> Longi_Binom$S =S
> Longi_Binom$U = U
> Longi_Binom$Z = Z
> Longi_Binom$SIGMA = SIGMA
> Longi_Binom$group = group
>
>

> JointScoreTest_withNuisanceVC(Longi_Binom$y, Longi_Binom$X,
+         Longi_Binom$S, Longi_Binom$U, Longi_Binom$Z,
+         Longi_Binom$group, Longi_Binom$SIGMA,
+         out_type = "D", binom_size = 6)

$pvalue
[1] 0.0009710743

$opt.rho
[1] 0.75

$score.each
[1] 7.186171 7.199028 7.223049 7.282885 6.369138
```