Modified Henderson's method 3 approach: the R-code

README and R-code: Razaw Al-Sarraj and Lars Rönnegård

R version:

R version 3.5.1 (2018-07-02) – "Feather Spray"

Platform: i386-w64-mingw32/i386 (32-bit)

Packages

nlme 'lme4' was built under R version 3.5.3

1 The README function for the R-code

A a two-way model without interaction is considered,

$$Y = X\beta + Z_1 u_1 + Z_2 u_2 + e, \tag{1}$$

where Y is the vector of observations, X is a known matrix, β is the vector of unknown fixed effects parameters, u_1 and u_2 are unknown random effect parameters, and Z_1 and Z_2 are incidence matrices of known elements. In this model there are three variance components to estimate, i.e. the variances of the two random effects denoted by σ_1^2 and σ_2^2, and the third is the error variance component denoted by σ_e^2. Without loss of generality, the main interest is on the first variance component.

The function `modHenderson_lmfast` calculates the variance component from two different decompositions of Henderson's method 3. They are referred to as Partition I and Partition II. In Partition I, all the variance components are included whereas in Partition II only two variance components are included. In addition, the function computes the corresponding modified variance estimates from the two partitions. For details and the notations, see Al-Sarraj and von Rosen (2009) and the Appendix section in Rönnegård et. al (2008).

Partition I

All the equations and notations are as in Al-Sarraj and von Rosen (2009). The steps of function `modHenderson_lmfast`

- fits the sub-model lm0, $Y = X\beta + e$, Eq.12,

- fits the sub-model lm1, $Y = X\beta + Z_1 u_1 + e$, Eq.13,

- fits the sub-model lm2, $Y = X\beta + Z_1 u_1 + Z_2 u_2 + e$, Eq.14,

- the A, B, C, a, b, c, d, e, f are as defined in Eq.18.

- sigma1, the variance component as in Eq.23.

The modified variance estimator is denoted as modsigma1

- c_1, d_1 and d_2 as in Eq.41, Eq. 42 and Eq. 43, respectively,

- modsigma1 is calculated as in Eq.27.

Partition II

All the equations, notations and pages are as in Rönnegård et. al (2008)

- fits the sub-model lm3, $Y = X\beta + Z_2 u_2$.
- sigma12, the variance component as in Eq.6.

The modified variance component is denoted as modsigma12. For the coefficients involved in computations c_2 and ε_1, we refer to Kelly and Mathew (1994). However the values have been calculated in accordance to our notations and with regard to the set of estimation equations used,

- c_2 and ε_1 are as in page 12. However with correction[1],
- the modified variance component estimator is denoted as modsigma12.

In summary, the function `modHenderson_lmfast` calculates four variance components:

Table 1: Variance estimates

	Original estimate	Modified estimate
Partition I	sigma 1	modsigma1
Partition II	sigma12	modsigma12

[1]The denominator of the equation for c_2 in Rönnegård et. al (2008) should be with '+1'

$$c_2 = \frac{1}{\frac{2}{\{tr(P_{x_{12}} - P_{x2})V_1\}^2}\{tr(P_{x_{12}} - P_{x2})V_1 tr(P_{x_{12}} - P_{x2})V_1 + 1\}}$$

Simple simulation to test the functions

The dimension and size of the different elements of model (1) can be determined:

- the total number of observations,
- the number of clusters for random effect 1
- the number of clusters for random effect 2,
- the incidence matrices for the random effects and the design matrix for the fix effect are determined,
- the observations are simulated by deciding the values for the three variance components sigma1, sigma2 and sigmae of model (1).

From test3, all four variance estimates of Table 1 are obtained. These can be compared with the variance estimate varcomponent1 obtained when using the `lmer` function model (1).

APPENDIX

The R-code

```
####Modified Henderson's method 3############
####Code written by Lars Ronnegard 2008-07-29#####
rm(list=ls())
library(lme4)
library(nlme)
####
modHenderson_lmfast <- function(y, X, Z1, Z2, partitionII=FALSE) {
  #Calculate three linear models
  lm0 <- lm(y~. - 1, data = data.frame(X))
  lm1 <- lm(y~. - 1, data = data.frame(cbind(X, Z1)))
  lm2 <- lm(y~. - 1, data = data.frame(cbind(X, Z1, Z2)))
  #Covariance of parameter estimates
  TT0 <- summary(lm0)$cov.unscaled
  TT1  <- summary(lm1)$cov.unscaled
  TT2  <- summary(lm2)$cov.unscaled
  design.X0 <- model.matrix(lm0)[,!summary(lm0)$aliased]
  design.X <- model.matrix(lm1)[,!summary(lm1)$aliased]
  X1TT1X1 <- design.X%*%TT1%*%t(design.X)
  A <- X1TT1X1 - design.X0%*%TT0%*%t(design.X0)
  design.X <- model.matrix(lm2)[,!summary(lm2)$aliased]
  B <- design.X%*%TT2%*%t(design.X) - X1TT1X1
  rm(TT0); rm(TT1); rm(TT2)
  c <- n-sum(hatvalues(lm2))                #df for residuals
  e <- sum(hatvalues(lm2)-hatvalues(lm1)) #df for vc1
  f <- sum(hatvalues(lm1)-hatvalues(lm0)) #df for vc2
  yAy <- crossprod(y,(fitted(lm1)-fitted(lm0)))
  yBy <- crossprod(y,(fitted(lm2)-fitted(lm1)))
  yCy <- crossprod(y,(y-fitted(lm2)))
  rm(lm0); rm(lm1); rm(design.X0); rm(design.X)
  if (!partitionII) {rm(lm2);  rm(X1TT1X1)}
######################################
  a <- sum(t(Z1)*crossprod(Z1,A))
```

```
  AV1 <- crossprod(Z1,A%*%Z1)
  c1<-1/(2/(a^2)*sum(AV1*AV1)+1)
  rm(AV1)
  d <- sum(t(Z2)*crossprod(Z2,A))
  b <- sum(t(Z2)*crossprod(Z2,B))
  BV2 <-  crossprod(Z2,B%*%Z2)
  d1<-1/(2/(b^2)*sum(BV2*BV2)+1)
  rm(BV2)
########################################
  ##Henderson Partition I
  k<-d*e-f*b
  sigma1<-1/a*(yAy-d/b*yBy+k/(b*c)*yCy)
  ########## modified Henderson Partition I
  d2<-(d/b*d1*e-f)/((k/b)*(2/c+1))
  modsigma1<-c1/a*(yAy-d/b*d1*(yBy)+k/(b*c)*d2*(yCy))
  ###############
  rm(yAy); rm(yBy)
  if (partitionII) {
    ##Henderson Partition II
    lm3 <- lm(y~. - 1, data = data.frame(cbind(X, Z2)))
    TT3  <- summary(lm3)$cov.unscaled
    yEy <- crossprod(y,(fitted(lm2)-fitted(lm3)))
    l <- sum(hatvalues(lm2)-hatvalues(lm3))
    design.X <- model.matrix(lm3)[,!summary(lm3)$aliased]
    X3TT3X3 <- B + X1TT1X1
    rm(B)
    E <-  X3TT3X3 - design.X%*%TT3%*%t(design.X)
    rm(X3TT3X3); rm(design.X)
    EV1 <- crossprod(Z1,E%*%Z1)
    g <- sum(t(Z1)*crossprod(Z1,E))
    sigma12<-yEy/g-l*yCy/(c*g)
    ########## modified Henderson Partition II
    c2<- g^2/(2*sum(EV1*EV1)+g^2)
    e1<-c/(2+c)
    modsigma12<-c2*yEy/g-c2*e1*l*yCy/(c*g)
  }
  if (!partitionII) {
    sigma12=NULL
```

```
    modsigma12=NULL
  }
######################################################################
  vce <- list(HendersonPartitionI=sigma1,HendersonPartitionII=sigma12,m
 }


###Simple simulation to test the functions
set.seed(123)
n=1000 #Total number of observations
q1=100 #Number of clusters for random effect 1
q2=100 #Number of clusters for random effect 2
indx1<-sample(1:q1,n,replace=TRUE)
indx2<-sample(1:q2,n,replace=TRUE)
Z1<-matrix(0,n,q1)    #Incidence matrix for random effect 1
Z2<-matrix(0,n,q2)    #Incidence matrix for random effect 2
for (i in 1:n) {
        Z1[i,indx1[i]]=1
        Z2[i,indx2[i]]=1
}
X <- matrix(1,n,1)  #Design matrix for the fixed effects


###Simulate observations
sigma1=5    ####################variance component 1
sigma2=2        ####################variance component 2
sigmae=10       ####################variance component error

e<-matrix(rnorm(n,0,sqrt(sigmae)), n, 1)    #Residuals
u1<-matrix(rnorm(q1,0,sqrt(sigma1)), q1, 1)  #Random effect 1
u2<-matrix(rnorm(q2,0,sqrt(sigma2)), q2, 1)  #Random effect 2
y <- X + Z1 %*% u1 + Z2 %*% u2 + e
##########################################################################
###Test the functions
test3<-modHenderson_lmfast(y, X, Z1, Z2, partitionII=TRUE)
test3 ################all four estimates of the first variance componen

###########The variance components computed with lmer()
```

```
lmer.results <- lmer(y ~ 1 + (1 | factor(indx1)) + (1 | factor(indx2)))

varcomp<-print(VarCorr(lmer.results),comp="Variance")
varcomponent = as.data.frame(VarCorr(lmer.results))
varcomponent1 =varcomponent[1,'vcov']


varcomponent1######the first variance component according to lmer
```

End of R-code [2]

[2]The R-code is provided also as a separate file (`ModifiedHenderson_thesis.txt`)

References

[1] Al-Sarraj, R. and von Rosen, D. (2009). Improving Henderson's method 3 approach when estimating variance components in a two-way mixed linear model. Peer reviewed book chapter in: *Statistical Inference, Econometric Analysis and Matrix Algebra*. Schipp, B. and Kräer, W. (Eds.). Physica-Verlag, Heidelberg, 125–142.

[2] Rönnegård, L., Al-Sarraj, R. and von Rosen, D. (2008). Non-iterative variance component estimation in QTL analysis. *Journal of Animal Breeding and Genetics*, 126, 110–116.

[3] Kelly, R. J. and Mathew, T. (1994). Improved nonnegative estimation of variance components in some mixed models with unbalanced data. *Technometrics*, 36, 171– 181.