Applied Statistical Methods - Solution 4

Peter von Rohr

2020-03-16

Problem 1: Traditional Predicted Breeding Values

Given the following data set with observations and a pedigree for a group of animals.

Table 1: Phenotypic Observations

Animal	Observation
1	100.430
2	103.396
3	114.458
4	100.068
5	104.144
6	117.524
7	97.744
8	111.926
9	103.486
10	97.914
11	104.651
12	115.714
13	86.900
14	101.097
15	102.795
16	112.182
17	109.295
18	105.271
19	91.744
20	101.132
21	107.385

The observations in Table 1 can be read from

https://charlotte-ngs.github.io/GELASMSS2020/ex/w05/data_ex04_phe.csv.

The pedigree showing the ancestral relationships is shown below

Table 2: Pedigree

Animal	Sire	Dam
1	NA	NA

2	NA	NA
3	NA	NA
4	NA	NA
5	NA	NA
6	2	3
7	1	3
8	2	5
9	1	5
10	7	8
11	7	8
12	6	9
13	7	8
14	7	9
15	6	8
16	6	9
17	6	8
18	6	8
19	7	8
20	6	9
21	7	8

The pedigree can be read from

https://charlotte-ngs.github.io/GELASMSS2020/ex/w05/data_ex04_ped.csv

Your Task

Predict breeding values for the animals given in the dataset and in the pedigree without using any genotypic information using a BLUP animal model. Set up the mixed model equations for the BLUP animal model and use the package `pedigreemm` to get the inverse of the relationship matrix.

Hints

- Use a mixed linear model with a constant intercept as a fixed effect and the breeding values of all animals as random effects. Hence the following model can be assumed

$$y = Xb + Za + e$$

where y is the vector of all observations, b has just one element and X has one column with all ones. The vector a contains the breeding values for all animals. The matrix Z links the breeding values to the phenotypic observations. The random errors are represented by the vector e.

- Then residual variance σ_e^2 can be assumed to be $\sigma_e^2 = 75$. The genetic additive variance σ_a^2 is $\sigma_a^2 = 25$

Solution

The phenotypic data is read using the following statements

```
s_course_url <- "https://charlotte-ngs.github.io/GELASMSS2020"
s_phe_path <- file.path(s_course_url, "ex/w05/data_ex04_phe.csv")
tbl_phe <- readr::read_csv(file = s_phe_path)
```

Similarily the pedigree is read using

```
s_ped_path <- file.path(s_course_url, "ex/w05/data_ex04_ped.csv")
tbl_ped <- readr::read_csv(file = s_ped_path)
```

The mixed model equations for the traditional BLUP animal model has the following structure

$$\left[\begin{array}{cc} X^TX & X^TZ \\ Z^TX & Z^TZ + \lambda * A^{-1} \end{array}\right] \left[\begin{array}{c} \hat{b} \\ \hat{a} \end{array}\right] = \left[\begin{array}{c} X^Ty \\ Z^Ty \end{array}\right]$$

The matrix X has just one column which contains all ones. Because all animals have an observation, the matrix Z is an identity matrix. The matrix A is the numerator relationship matrix. The variable λ is the ratio between σ_e^2 and σ_a^2.

The single components of the mixed model equations have the following structure.

- X^TX is a single number and corresponds to the number of observations.
- X^TZ is a matrix with one row with all ones
- Z^TX is a matrix with one column with all ones
- Z^TZ is an identity matrix
- $\lambda = \sigma_e^2/\sigma_a^2$
- A^{-1} is the inverse numerator relationship matrix
- X^Ty is the sum of all observations
- Z^Ty is the matrix with one colum with all observations

The above points are now implemented with the following computations

```
### # number of observations and matrix X
n_nr_obs <- nrow(tbl_phe)
matX <- matrix(1, nrow = n_nr_obs, ncol = 1)
### # number of animals in pedigee and matrix Z
n_nr_ani <- nrow(tbl_ped)
matZ <- diag(nrow = n_nr_ani)
### # observations
vecY <- tbl_phe$Observation
### # numerator relationship matrix
ped <- pedigreemm::pedigree(sire = tbl_ped$Sire,
                            dam = tbl_ped$Dam,
                            label = as.character(tbl_ped$Animal))
matAinv <- as.matrix(pedigreemm::getAInv(ped))
lambda <- resvar/genvar
### # left hand side of mme
matxtx <- crossprod(matX)
matxtz <- crossprod(matX,matZ)
matztzlainv <- crossprod(matZ) + lambda * matAinv
matlhs <- rbind(cbind(matxtx,matxtz),cbind(t(matxtz), matztzlainv))
matrhs <- rbind(crossprod(matX,vecY), crossprod(matZ,vecY))
matSol <- solve(matlhs, matrhs)
```

The matrix `matSol` contains the solutions of the mixed model equations. It has just one column. The first element is the estimate of the intercept. All other elements are the estimated breeding values of all animals. We can now show the soltutions in tabular form.

Table 3: Estimate of fixed Effect (b)

Effect	Estimate
General Mean (b)	104.1966

The results for the predicted breeding values are

Table 4: Predicted Breeding Values for all Animals

Animal	Predicted Breeding Value
1	-2.3843907
2	1.0534653
3	2.3390534
4	-1.0321492
5	0.0240212
6	3.9797662
7	-2.6079054
8	-0.0732777
9	-0.5186034
10	-2.0465923
11	-1.0841637
12	3.1286988
13	-3.6200208
14	-1.7827319
15	1.4739813
16	2.6241274
17	2.4025527
18	1.8276956
19	-2.9280208
20	1.0455560
21	-0.6935923

Problem 2: Prediction of Genomic Breeding Values Using GBLUP

Use the same phenotypic observations as in Problem 1. In addition to that we use genomic information available in

https://charlotte-ngs.github.io/GELASMSS2020/ex/w05/data_ex04_gen.csv

Your Tasks

Predict the genomic breeding values using the GBLUP approach.

Hints

- Use an analogous mixed linear effect model as was used in Problem 1. Instead of the vector of breeding values use the vector g of genomic breeding values as random effects of the model. Hence the following model can be assumed

$$y = Xb + Zg + e$$

where y is the vector of all observations, b has just one element and X has one column with all ones. The vector g contains the genomic breeding values for all animals. The matrix Z links the breeding values to the phenotypic observations. The random errors are represented by the vector e.

- Use the genomic relationship matrix in the mixed model equations
- The ratio λ of between the variances is assumed to be the same as in Problem 1.
- If the inverse of the genomic relationship matrix cannot be computed, adjust the genomic relationship matrix with the numerator relationship matrix A according to the following formula

$$G^* = 0.95 * G + 0.05 * A$$

where G is the matrix determined based on th given data and the nummerator relationship matrix A can be computed with the function `pedigreemm::getA()` from package `pedigreemm`.

Solution

The genomic information is read using the following statements.

```r
s_gen_path <- file.path(s_course_url, "ex/w05/data_ex04_gen.csv")
tbl_gen <- readr::read_csv(file = s_gen_path)
```

The phenotypic observations are read the same way as in Problem 1.

```r
s_course_url <- "https://charlotte-ngs.github.io/GELASMSS2020"
s_phe_path <- file.path(s_course_url, "ex/w05/data_ex04_phe.csv")
tbl_phe <- readr::read_csv(file = s_phe_path)
```

The mixed model equations for the GBLUP model has the following structure

$$\left[\begin{array}{cc} X^TX & X^TZ \\ Z^TX & Z^TZ + \lambda * G^{-1} \end{array}\right] \left[\begin{array}{c} \hat{b} \\ \hat{g} \end{array}\right] = \left[\begin{array}{c} X^Ty \\ Z^Ty \end{array}\right]$$

where G is the genomic relationship matrix. The vector $\hat{g}$ is the vector of predicted genomic breeding values.

The genomic relationship matrix G is computed using the function proposed in the solution of Exercise 3. This function is shown here once again.

```r
computeMatGrm <- function(pmatData) {
  matData <- pmatData
  # check the coding, if matData is -1, 0, 1 coded, then add 1 to get to 0, 1, 2 coding
  if (min(matData) < 0) matData <- matData + 1
  # Allele frequencies, column vector of P and sum of frequency products
  freq <- apply(matData, 2, mean) / 2
  P <- 2 * (freq - 0.5)
  sumpq <- sum(freq*(1-freq))
  # Changing the coding from (0,1,2) to (-1,0,1) and subtract matrix P
  Z <- matData - 1 - matrix(P, nrow = nrow(matData),
                             ncol = ncol(matData),
                             byrow = TRUE)
  # Z%*%Zt is replaced by tcrossprod(Z)
  return(tcrossprod(Z)/(2*sumpq))
}
```

The first step is to convert the the genomic information read from the file into a matrix where only genotypes are present. The file with the genotypic information contains the animal IDs in the first column. These IDs must be removed before we can compute the genomic relationship matrix.

```
### # convert data_frame into matrix and remove animal IDs
matGeno <- as.matrix(tbl_gen[,2:ncol(tbl_gen)])
### # compute genotypic relationship matrix
matGrm <- computeMatGrm(pmatData = matGeno)
### # correction with A because matGrm is singular
matA <- as.matrix(pedigreemm::getA(ped = ped))
matGrmPD <- 0.95 * matGrm + 0.05 * matA
matGrmInv <- solve(matGrmPD)
```

In the coefficient matrix, we have to replace A^{-1} by G^{-1}. Everything else can be taken from the solution of Problem 1.

```
matztzlginv <- crossprod(matZ) + lambda * matGrmInv
matlhsgblup <- rbind(cbind(matxtx,matxtz),cbind(t(matxtz), matztzlginv))
matSolgblup <- solve(matlhsgblup, matrhs)
```

The results are presented the same way as in Problem 1.

Table 5: Estimate of fixed Effect (b)

Effect	Estimate
General Mean (b)	104.237

The results for the predicted breeding values are

Table 6: Predicted Genomic Breeding Values for all Animals

Animal	Predicted Genomic Breeding Value
1	-0.9319013
2	2.1859151
3	3.6129503
4	-0.5363591
5	-0.5213743
6	3.3573633
7	-1.8396597
8	-0.2592185

9	-0.1797162
10	-2.5724641
11	-3.5615369
12	3.6365593
13	-2.2756949
14	-3.5437158
15	-0.9981064
16	3.4208643
17	3.5752874
18	1.7472823
19	-4.2113891
20	-0.4345360
21	0.6094621

Comparing the ranking according to Problem 1 and according to Problem 2 shows the following result

```r
order(matSol[2:nrow(matSol),1], decreasing = TRUE)
```

```
## [1]  6 12 16 17  3 18 15  2 20  5  8  9 21  4 11 14 10  1  7 19 13
```

The same ranking for the genomic breeding values

```r
order(matSolgblup[2:nrow(matSolgblup),1], decreasing = TRUE)
```

```
## [1] 12  3 17 16  6  2 18 21  9  8 20  5  4  1 15  7 13 10 14 11 19
```