Applied Statistical Methods - Solution 5

Peter von Rohr

2019-03-25

Problem 1: Predicted Traditional Breeding Values For Progeny

According to the results from Exercise 4, animals 6 and 9 are mated to each other. The result of this mating are two male and two female offspring. From the two brothers and the two sisters the better male offspring and the better female offspring should be selected. Figure 1 shows the pedigree of the described matings.

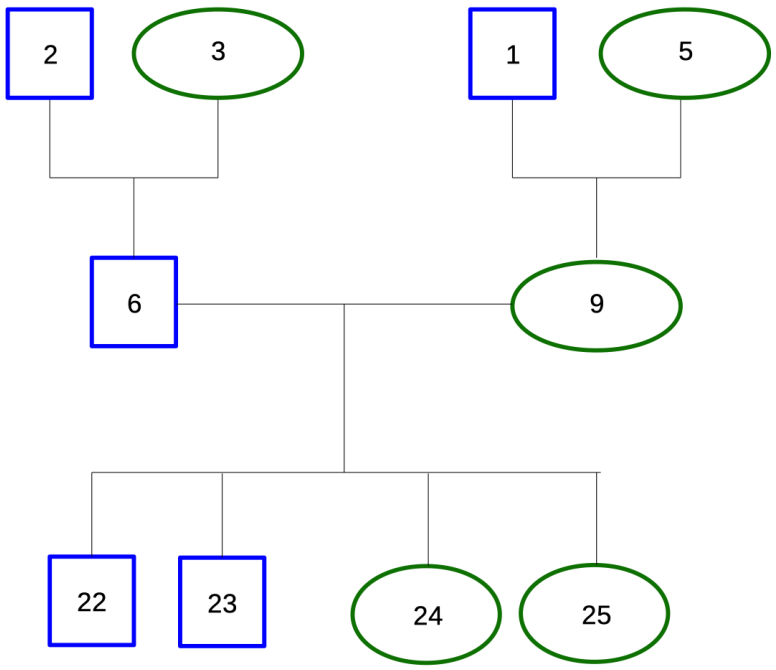

Figure 1: Pedigree Of Mating Between Parent Animals

Your Tasks

- Compute the predicted breeding values for all offspring from mating animals 6 and 9, assuming that the offspring do not have any observations.
- Is it possible to find the better male and female offspring based on the predicted

Hints

- The breeding values of the offspring can either be derived using an argument from quantiative genetics about the relationship of predicted breeding values of parents and progeny or it can be computed by extending the mixed model equation from Problem 1 in Exercise 4.

Solution

The argument from quantitative genetics about predicted breeding values of parents and progeny is that for an animal i with parents s and d the predicted breeding value $\hat{a}_i$ is the average of the predicted breeding values $\hat{a}_s$ and $\hat{a}_d$ of parents s and d, provided that animal i does not have any own performance records. Because the two brothers and sisters have the same parents 6 and 9, they all have the same predicted breeding values. With this it is clear that the task of finding the better male and female of the brothers and sisters cannot be solved with the traditionally predicted breeding values.

We are going to verify the above described argument by extending the computations of Problem 1 from Exercise 4. The same procedure is used to come up with the predicted breeding values for the extended pedigree. We start by specifying the model components such as the design matrices X and Z and the vector y of phenotypic observations. Together with the numerator relationship matrix, we can set up the mixed model equations.

Because the four offspring 22, 23, 24 and 25 do not have any phenotypic observations, the vector y and the design matrix X do not changed compared to the solution of Problem 1 in Exercise 4. Because there are four offspring animals added to the pedigree, the vector a of all breeding values and with that the matrices Z and A have to be changed.

```
### # read the phenotypic data again and extract the vector of observations
s_course_url <- "https://charlotte-ngs.github.io/GELASMSS2019"
### # this part is only needed for local execution, it can be cut out in the final version
bOnline <- FALSE
if (!bOnline) s_course_url <- here::here()
### # end of local execution stuff
```

```
### # define the path to the phenotypic data
s_phe_path <- file.path(s_course_url, "ex/w05/data_ex04_phe.csv")
tbl_phe <- readr::read_csv(file = s_phe_path)
```

```
### vector of phenotypic observations
vec_y <- tbl_phe$Observation
### # the matrix X
n_nr_obs <- nrow(tbl_phe)
mat_X <- matrix(1, nrow = n_nr_obs, ncol = 1)
```

The extended pedigree is read and the inverse of the numerator relationship matrix is constructed

```
s_ped_ext_path <- file.path(s_course_url, "ex/w06/data_ex05_ped.csv")
tbl_ped_ext <- readr::read_csv(file = s_ped_ext_path, col_types = "iii")
ped_ext <- pedigreemm::pedigree(sire = tbl_ped_ext$Sire,
                                dam = tbl_ped_ext$Dam,
                                label = as.character(tbl_ped_ext$Animal))
mat_Ainv_ext <- as.matrix(pedigreemm::getAInv(ped = ped_ext))
```

The matrix Z for the extended pedigree has four columns more, because there are four animals more in the pedigree. The number of rows stays the same as it was in Problem 1 of Exercise 4. The matrix Z can be written as

```
n_nr_animal_ext <- nrow(tbl_ped_ext)
mat_Z_ext <- cbind(diag(nrow = n_nr_obs),
                        matrix(0, nrow = n_nr_obs,
                               ncol= n_nr_animal_ext - n_nr_obs))
```

All compoents of the mixed model equations are ready to be setup

```
genvar <- 25
resvar <- 75
lambda <- resvar / genvar
mat_xtx <- crossprod(mat_X)
mat_xtz <- crossprod(mat_X, mat_Z_ext)
mat_ztzlainv <- crossprod(mat_Z_ext) + lambda * mat_Ainv_ext
mat_M <- rbind(cbind(mat_xtx, mat_xtz), cbind(t(mat_xtz), mat_ztzlainv))
mat_rhs <- rbind(crossprod(mat_X, vec_y), crossprod(mat_Z_ext, vec_y))
mat_sol <- solve(mat_M, mat_rhs)
```

The solutions are shown in the following tables.

Table 1: Estimate of fixed Effect (b)

Effect	Estimate
General Mean (b)	104.3447

The results for the predicted breeding values are

Table 2: Predicted Breeding Values for all Animals

Animal	Predicted Breeding Value
1	-0.9690123
2	-0.3434062
3	3.4925207
4	-0.4899800
5	-1.6901222
6	3.4612342
7	-3.5161795
8	0.3555508
9	0.6606556
10	-2.2732197
11	1.6795290
12	3.3907168
13	-3.7158891
14	-1.6877462
15	1.5451453
16	-0.2349340
17	2.4737168
18	-1.2222197
19	-3.1546483
20	-1.8135054
21	2.0701004
22	2.0609449
23	2.0609449

24	2.0609449
25	2.0609449

From the results shown in Table 2, we can see that the brothers and sisters 22, 23, 24 and 25 all have the same predicted breeding value. This predicted breeding value corresponds to the average of the predicted breeding values of the parents 6 and 9. The predicted breeding values of the parents are

```
require(dplyr)
tbl_pred_bv %>%
  filter(Animal == n_id_sire) -> tbl_sire_bv
tbl_pred_bv %>%
  filter(Animal == n_id_dam) -> tbl_dam_bv
cat("Predicted breeding value of sire ", n_id_sire, ": ", tbl_sire_bv$`Predicted Breeding Value`, "\n",
    "Predicted breeding value of dam  ", n_id_dam,  ": ", tbl_dam_bv$`Predicted Breeding Value`)
```

```
## Predicted breeding value of sire  6 :  3.461234
##  Predicted breeding value of dam   9 :  0.6606556
```

The mean of the predicted breeding values of the parents is computed as

```
mean(c(tbl_sire_bv$`Predicted Breeding Value`, tbl_dam_bv$`Predicted Breeding Value`))
```

```
## [1] 2.060945
```

With the fact that all brothers and sisters from parents 6 and 9 without own performance observations have the same predicted breeding value, it becomes clear, that we cannot select between the four animals 22, 23, 24 and 25. With the traditionally predicted breeding values, selection among brothers and sisters is only possible as soon as they get own performance records or as soon as they get records based on their own progeny. This fact increases the generation interval in species such as cattle.

The solution of Problem 2 illustrates how this can be changed by including genomic information on the brothers and sisters which allows for selection between brothers and sisters as soon as the genomic information is available.

Problem 2: Genomic Breeding Values For Progeny

The four offsprings of parents 6 and 9 also have genotype data. The complete genomic data for all animals can be read from

/Users/peter/Data/Projects/GitHub/charlotte-ngs/GELASMSS2019_gh-root/master/GELASMSS2019/ex/w06/data_ex05_g

Your Tasks

- Predict genomic breeding values as in Problem 2 of Exercise 4 using a GBLUP approach. But this time the four offsprings of parents 6 and 9 are also included in the analysis.
- Try to rank the four offpsring according to their genomic breeding value.

Hints

- Use the same phenotypic information as in Problem 1.
- Use the same procedure to make the genomic relationship matrix G invertible as shown in Problem 2 of Exercise 4.
- Use the same model as in Problem 2 of Exercise 4.

Solution

The genomic information is read using the following statements.

```
s_gen_path <- file.path(s_course_url, "ex/w06/data_ex05_gen.csv")
tbl_gen <- readr::read_csv(file = s_gen_path)
```

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

```
### # define the path to the phenotypic data
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 in which the partitioning between genotyped animals with and without observations is reflected in the MME.

$$\begin{bmatrix} X^TX & X^TZ & 0 \\ Z^TX & Z^TZ+\lambda*G^{(11)} & \lambda*G^{(12)} \\ 0 & \lambda*G^{(21)} & \lambda*G^{(22)} \end{bmatrix} \begin{bmatrix} \hat{b} \\ \hat{g}_1 \\ \hat{g}_2 \end{bmatrix} = \begin{bmatrix} X^Ty \\ Z^Ty \\ 0 \end{bmatrix}$$

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

```
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_ext <- as.matrix(pedigreemm::getA(ped = ped_ext))
matGrmPD <- 0.95 * matGrm + 0.05 * matA_ext
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(mat_Z_ext) + lambda * matGrmInv
matlhsgblup <- rbind(cbind(mat_xtx,mat_xtz),cbind(t(mat_xtz), matztzlginv))
```

```
matSolgblup <- solve(matlhsgblup, mat_rhs)
```

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

Table 3: Estimate of fixed Effect (b)

Effect	Estimate
General Mean (b)	104.6622

The results for the predicted breeding values are

Table 4: Predicted Genomic Breeding Values for all Animals

Animal	Predicted Genomic Breeding Value
1	-0.2923797
2	-1.1731150
3	4.7264854
4	-0.1353683
5	-2.1101362
6	3.8596177
7	-4.5409302
8	0.5320952
9	0.2431149
10	-2.5648008
11	2.5027457
12	5.3734944
13	-5.5815345
14	-4.5590086
15	0.7218607
16	-0.9236166
17	3.8180742
18	-2.6311500
19	-4.3808952
20	-4.1923179
21	2.6579106
22	2.7235210
23	2.8554832
24	0.9845656
25	2.3677558

The predicted genomic breeding values for the four offspring are:

```
tbl_pred_bv_prog <- tbl_pred_bv %>% filter(Animal > 21)
knitr::kable(tbl_pred_bv_prog, booktabs = TRUE, longtable = TRUE,
             caption = "Predicted Genomic Breeding Values for four Offspring")
```

Table 5: Predicted Genomic Breeding Values for four Offspring

Animal	Predicted Genomic Breeding Value
22	2.7235210

23	2.8554832
24	0.9845656
25	2.3677558

From the last line of the mixed model equations shows how the genomic breeding values $\hat{g}_2$ for genotyped animals without observations can be predicted.

$$\hat{g}_2 = -(G^{(22)})^{-1} \cdot G^{(21)} \cdot \hat{g}_1$$

For our example this means

```
(mat_comp_gbv <- -solve(matGrmInv[22:25,22:25]) %*% matGrmInv[22:25,1:21] %*% matSolgblup[2:22,1])
```

```
##         [,1]
## 22 2.7235210
## 23 2.8554832
## 24 0.9845656
## 25 2.3677558
```

The ranking of the four offspring corresponds to

```
order(mat_comp_gbv, decreasing = TRUE)
```

```
## [1] 2 1 4 3
```