Livestock Breeding and Genomics - Exercise 10

Peter von Rohr

2022-12-09

Problem 1: Marker Effect Model

We are given the dataset that is shown in the table below. This dataset contains gentyping results of 10 for 2 SNP loci.

Animal	SNP A	SNP B	Observation
1	0	0	156
2	1	0	168
3	0	1	161
4	1	0	164
5	-1	0	128
6	-1	1	124
7	0	-1	143
8	1	1	178
9	1	0	163
10	0	0	151

The above data can be read from:

```
## https://charlotte-ngs.github.io/lbgfs2022/data/geno_data.csv
```

Your Task

- The goal of this problem is to estimate SNP marker effects using a `marker effect model`. Because we have just 2 SNP loci, you can use a fixed effects linear model with the 2 loci as fixed effects. Furthermore you can also include a fixed intercept into the model.
- Specify all the model components including the vector of observations, the design matrix X, the vector of unknowns and the vector of residuals.
- You can use the R-function `lm()` to get the solutions for estimates of the unknown SNP effects.

Problem 2: Breeding Value Model

Use the same data as in Problem 1 to estimate genomic breeding values using a `breeding value model`.

Hints

- The only fixed effect in this model is the mean μ which is the same for all observations.
- You can use the following function to compute the genomic relationship matrix

```r
#' Compute genomic relationship matrix based on data matrix
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))
}
matG <-computeMatGrm(pmatData = t(mat_geno_snp))
matG_star <- matG + 0.01 * diag(nrow = nrow(matG))
n_min_eig_matG_start <- min(eigen(matG_star, only.values = TRUE)$values)
if (n_min_eig_matG_start < sqrt(.Machine$double.eps))
  stop(" *** Genomic relationship matrix singular with smallest eigenvalue: ",
       n_min_eig_matG_start)
```

- The resulting genomic relationship matrix is given by

$$G = \begin{bmatrix} 0.093 & -0.125 & -0.125 & -0.125 & 0.292 & 0.083 & 0.292 & -0.333 & -0.125 & 0.083 \\ -0.125 & 0.718 & -0.333 & 0.708 & -0.958 & -1.167 & 0.083 & 0.5 & 0.708 & -0.125 \\ -0.125 & -0.333 & 0.718 & -0.333 & 0.083 & 0.917 & -0.958 & 0.5 & -0.333 & -0.125 \\ -0.125 & 0.708 & -0.333 & 0.718 & -0.958 & -1.167 & 0.083 & 0.5 & 0.708 & -0.125 \\ 0.292 & -0.958 & 0.083 & -0.958 & 1.552 & 1.333 & 0.5 & -1.167 & -0.958 & 0.292 \\ 0.083 & -1.167 & 0.917 & -1.167 & 1.333 & 2.177 & -0.75 & -0.333 & -1.167 & 0.083 \\ 0.292 & 0.083 & -0.958 & 0.083 & 0.5 & -0.75 & 1.552 & -1.167 & 0.083 & 0.292 \\ -0.333 & 0.5 & 0.5 & 0.5 & -1.167 & -0.333 & -1.167 & 1.343 & 0.5 & -0.333 \\ -0.125 & 0.708 & -0.333 & 0.708 & -0.958 & -1.167 & 0.083 & 0.5 & 0.718 & -0.125 \\ 0.083 & -0.125 & -0.125 & -0.125 & 0.292 & 0.083 & 0.292 & -0.333 & -0.125 & 0.093 \end{bmatrix}$$

Your Tasks

- Specify all model components of the linear mixed model, including the expected values and the variance-covariance matrix of the random effects.