Institutt for matematiske fag, NTNU

Project 3

You may work in groups of up to two students. The final report should be be written in R markdown or LaTeX using knitr (preferably) and uploaded as **a single pdf file** to this dropbox folder by ~~October 30~~November 20 (midnight). The first page must include **your email adresse(s) and your candidate number(s)** (not your studentnumber).

All questions should be posted via the piazza forum.

1

In this problem you will implement a function `mylmm` that computes the maximum likelihood and restricted maximum likelihood estimates of the parameters ($\beta_0, \beta_1, \tau_0^2, \tau_1^2, \tau_{01}, \sigma^2$) of the mixed model

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_{0,i} + \gamma_{1,i} x_{ij} + \epsilon_{ij}$$

where $\boldsymbol{\gamma}_i = (\gamma_{0,i}, \gamma_{1,i})^T$ are iid binormally distributed with zero mean and variance matrix

$$\begin{bmatrix} \tau_0^2 & \tau_{01} \\ \tau_{01} & \tau_1^2 \end{bmatrix}$$

and ϵ_{ij} are iid $N(0, \sigma^2)$ for $i = 1, \ldots, m, j = 1, \ldots, n_i$.

a) First briefly explain how the model can be expressed in matrix notation

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U}\boldsymbol{\gamma} + \boldsymbol{\epsilon}.$$

b) Your function `mylmm` should take the vectors `y` (the response), `x` (a numeric covariate) and `group` (the grouping factor) as input and return estimates of the six model parameters as a vector.

Use the formulas on p. 373 for the profile and restricted log likelihoods. Either of these will have to be maximised numerically. You may use function `optim` for this. Your function `mylmm` should take a third logical argument `REML`

such that the additional term $-\frac{1}{2}\ln\det(\mathbf{X}^T\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{X})$ is added when `REML=TRUE`.

The goal is to apply the function to the `sleepstudy` data in `library(lme4)` but when developing your function, it may be wise to create and work on a small artificial data set.

c) Check the estimates against those computed by `lme4::lmer(Reaction ~ 1 + Days + (1+ Days|Subject~~group~~), REML=FALSE)` and with `REML==TRUE`.

d) Briefly comment on the difference between the ML and REML estimates.

Hints: As part of the computations inside `mylmm`, you'll need to construct some blockdiagonal matrices, e.g. $\mathbf{U}$. This is best done using the `bdiag` function in `library(Matrix)`. The following code illustrates roughly how this can be done:

```
# artificial data
m <- 5
n <- rep(3, m)
x <- rnorm(sum(n))
group <- factor((rep(1:m,n)))
# Computing U
library(Matrix)
U <- list() # set up an empty list
for (i in 1:m) {
   U[[i]] <- cbind(1,x[group==i]) # Construct a
}
U <- bdiag(U) # Change the list to a block diag
```

Using the `Matrix` class has the advantage of saving memory and cpu time since the matrix sparseness is exploited for example in computations of determinants and inverses. Also make sure that log determinants are computed correctly using the `Matrix`-method of the `determinant` function (see help page for `Matrix-class`).

To avoid repeating code, make liberal use of local functions computing $\mathbf{V}(\boldsymbol{\theta})$, $\hat{\boldsymbol{\beta}}(\mathbf{V})$, and both $l_P(\boldsymbol{\theta})$ and $l_R(\boldsymbol{\theta})$. Additionally, things that don't change with changing parameter values should preferably be be computed only once.

Write clear R code with proper indentation, following e.g. this style guide.

Note that `optim` by default minimises the function `fn` given. To instead maximise the given function supply the `control=list(fnscale=-1)` argument.

Also note that the computed log likelihood based on matrix classes in the `Matrix` package becomes a `1 x 1 Matrix of class "dgeMatrix"` which will cause `optim` to produce the error `Error in optim(c(-1, 1), f) : no method for coercing this S4 class to a vector`. A workaround is to use `as.vector( )` before returning the computed log likelihood.

Because of the somewhat akward constraints on the parameters in the numerical optimisation of $l_P(\boldsymbol{\theta})$ or $l_R(\boldsymbol{\theta})$ you may run into trouble if optim tries to evaluate the likelihood outside the domain of the parameters. To get around this problem it may be useful to consider one of the following strategies:

- Define $\boldsymbol{\theta}$ as a suitable transformation of $(\tau_0^2, \tau_1^2, \tau_{01}, \sigma^2)$ mapping the parameter space to all of $\mathbb{R}^4$.
- Do "box-constrained" optimization via the `upper` and `lower` arguments in the call to `optim`. In this case, instead of letting one of the entries of `theta` represent the covariance τ_{01} , it may be better to work the correlation which satisfies a box constraint.
- The default optimisation algorithm used by `optim` (`method="Nelder-Mead"`) can maximise functions returning `-Inf` in some parts of the parameter space so one option is to modify your function `l` such that it returns `-Inf` for illegal parameter values.
- Choose starting values close to the optimum and hope for the best.

Also, using built-in debugger in Rstudio is useful during development to verify that your code work as intended.

2

In this problem we will use a generalized linear mixed model to analyse part of the 2018 results from the Norwegian elite football league. Load the data by doing

```
long <- read.csv("https://www.math.ntnu.no/emner
```

Each match is represented by pairs of consecutive rows in the data frame; the first row below contains the number of goals scored by Molde against Sandefjord fotball with Molde playing at their home field, and the second row contains the number of goals by Sandefjord fotball (not playing at their home field) against Molde in the same

match. The covariates `attack` and `defence` are factors both with 16 levels. The variable `home` is similarly encoded as a factor with two levels and represents which team had the home field advantage.

```
> head(long)
                attack            defence home goa
1               Molde Sandefjord_Fotball  yes
2 Sandefjord_Fotball              Molde   no
3      Stroemsgodset            Stabaek  yes
4             Stabaek      Stroemsgodset   no
5                 Odd          Haugesund  yes
6           Haugesund                Odd   no
```

The data is incomplete in that the number of goals for some matches were not available at the time the data were recorded. Our aim is to fit a model to the available data and then simulate either the whole series or the remaining matches using the fitted model. Based on these simulations we will estimate the probabilities of each team winning the whole league.

a) First fit the model we will consider by doing

```
library(glmmTMB)
mod <- glmmTMB(goals ~ home + (1|attack) + (1|de
```

State the precise assumptions of this model in suitable mathematical notation. This should include an interpretation of all model parameters including random effect parameters associated with each team. Explain why the Poisson assumption may be reasonable way to model the inherent randomness of a football game.

Note that the model has crossed random effects so you will perhaps need multiple indices or nested indices defining the grouping structure for the data. Note also that each pair of teams plays against each other twice every season, one time at each others home field.

b) Extract the parameter estimates from the fitted model by doing

```
summary(mod)
ranef(mod)
```

Briefly discuss if the various parameter estimates appear reasonable.

If a team of average attack strength plays at its home field against another team of average defence strength, find estimates of the expectation and variance of number of goals scored by each team.

c) If we instead consider two randomly selected teams with random strength parameters, one of which plays at its home field, what is the expectation and variance of the number of goals scored by each team? Hint: You may find the laws of total expectations and variance useful.

Some of the variance in number of goals scored can be attributed to the inherent randomness of a football game and some of the variance to differences between the strength of the teams playing. Compute an estimate of each of these two proportions.

d) Using likelihood ratio tests, test if the two random effect terms in the model are significant. In addition to the p-values of the also compute the critical value of the test. Hint: See Fahrmeir pp. 381-383.

Also do a likelihood ratio test of the home field advantage. Hint: Are these likelihoods comparable?

e) For each match, if there is a draw both teams gets 1 point each, otherwise the winning team gets 3 points and the loosing team 0 points. Based on the result for the matches played, write a function that takes the above data.frame as input, computes the total number of points given to each team, and returns the ranking of each team. If two teams gets the same number of total points they are ranked based on their respective goal differences. For each team, this goal difference equals to the total number of goals scored in all matches minus goals conceded. If these goal differences are also equal, ranking is based on total number of goals scored. To easily compute these ranks, perhaps use `data.table::frankv`. Use option `ties="random"` as a last resort (thus we will not rank the teams based on the complete regulations of NFF given in § 4.1d in Turneringsbestemmelser for eliteserien).

f) Based on the fitted model, the expected number of goals in given matches can be computed by doing

```
predict(mod, type="response")
```

where `newdata` is either a subset or the whole dataset. Based on these expectations, we will simulate a new

realisations of the missing matches or the whole series using the `rpois` function in R. We want to simulate 1000 such realisations. Using the function in point e), compute the ranking of each team and store this in a 1000×16 matrix. You should perhaps do this inside another function that inserts the simulated goals into the complete data frame which you then pass to your function from point d).

Summarise the simulation results in a suitable way, for example, you may want to compute estimates of the probability that each team is ranked as number 1, number 2, number 3 etc or you may want to compute the expected final ranking of each team.

g) Does there seem to be any simple relationship between the random effects parameters in point b) and the average ranking obtained in point e)?

Hints: The object returned by the function `ranef` is a nested `list` with various attributes. One way to explore it is to do e.g. `re ← ranef(mod); str(re)` but everything is also described in the "Value" section of the help page.

2020-11-05, Jarle Tufto

Logg inn