

### Aufgabe 1

In dieser Aufgabe sollen die Ergebnisse der verschiedenen Schätzmethoden für GLMMs verglichen werden.

- (a) Simulieren Sie einen Datensatz in R für folgendes Modell:

$$y_{ij}|x_{ij}, b_i \sim \text{Ber}(\pi_{ij}), \text{ unabhängig verteilt}$$

$$\pi_{ij} = P(y_{ij} = 1|x_{ij}, b_i) = g^{-1}(x_{ij}\beta + b_i), \quad i = 1, \dots, m; \quad j = 1, \dots, n_i$$

mit

- $g^{-1}(\eta) = \frac{\exp(\eta)}{1+\exp(\eta)}$
- $x_{ij} \stackrel{\text{iid}}{\sim} U(-1, 1)$ , d.h.  $x_{ij}$  ist gleichverteilt auf dem Intervall  $[-1, 1]$
- $\beta = 2$
- $b_i \stackrel{\text{iid}}{\sim} N(0, 1)$
- $m = 50$  Subjekte
- $n_i = 5$  Beobachtungen pro Subjekt

Dabei soll der Datensatz so erzeugt werden, dass Sie Parameter, wie Anzahl der Subjekte oder Größe von  $\beta$ , einfach verändern können.

- (b) Schätzen Sie das Modell auf den simulierten Daten mit den 3 Schätzmethoden, die Sie aus der Vorlesung kennen, also mit
- (i) Penalized Quasi-Likelihood (PQL), Funktion `glimmPQL()` aus R-Paket `MASS`
  - (ii) Laplace-Approximation, Funktion `glmer()` aus R-Paket `lme4`
  - (iii) (Adaptive) Gaußsche Quadratur, Funktion `glmer()` aus R-Paket `lme4`

Vergleichen Sie die Ergebnisse aus den verschiedenen Schätzmethoden, insbesondere die Schätzungen der festen Effekte, sowie die Schätzung der Varianz der zufälligen Effekte.

- (c) Erhöhen Sie den Stichprobenumfang und/oder die Zahl der Beobachtungen pro Subjekt. Schätzen Sie das Modell erneut mit den verschiedenen Methoden und betrachten Sie, wie sich die Schätzungen verändern.

### Lösung 1

- (a) 

```
## function to simulate data and responses for a random intercept logit model
## or log-linear poisson model
simRI <- function(
  m = 50, # (approx.) no. of subjects
  ni = 4, #(avg.) no. of obs. per subject
  balanced = TRUE, # balanced design?
```

```

sdInt = 1, # sd(Random Intercept)
beta =c(1), # fixed effects (multiple effects possible)
# simulate binary("binomial") or count ("poisson") data?
family=c("binomial", "poisson")
){

family <- match.arg(family) # defaults to "binomial"
p <- length(beta)

#design matrices:  $x_i \sim U[-1,1]$ , simulate p variables
X <- matrix(runif(m*ni*p, -1, 1), nrow=m*ni, ncol=p)
id <- if(balanced){
  factor(rep(1:m, each=ni)) # balanced
}else{
  factor(sample(1:m, m*ni, rep=TRUE)) # unbalanced
}

# U = incidence matrix for id-factor
U <- model.matrix(~ 0 + id)
# draw random intercepts  $\sim N(0, sdInt^2)$ 
b <- rnorm(nlevels(id), sd = sdInt)

# generate y
if(family == "binomial"){
  mu <- plogis(X %*% beta + U %*% b) # plogis(x) is inverse logit function
  y <- rbinom(m*ni, 1, mu)
}
if(family == "poisson"){
  mu <- exp(X %*% beta + U %*% b)
  y <- rpois(m*ni, mu)
}

return(data.frame(y, X=X, id))
}

# simulate data
data <- simRI(m = 50, ni = 4, balanced = TRUE,
             sdInt = 1, beta = c(2), family="binomial")

```

(b) *#### estimate the models and look at coefficients*

```
# fit the GLMM via PQL
require(MASS)
require(nlme)
fitPQL <- try(glmmPQL(y ~ X, random = ~ 1|id, family = "binomial",
                    data = data))
fixef(fitPQL)

(Intercept)          X
-0.06693632  2.02550358

VarCorr(fitPQL)

id = pdLogChol(1)
          Variance StdDev
(Intercept) 0.2354429 0.4852246
Residual    0.9189242 0.9586054

# fit the GLMM via Laplace
require(lme4)
fitLaplace <- try(glmer(y ~ X + (1|id), family = "binomial",
                      data = data))
fixef(fitLaplace)

(Intercept)          X
-0.07022781  2.07854809

VarCorr(fitLaplace)

Groups Name          Std.Dev.
id      (Intercept) 0.39435

# fit the GLMM via Adaptive Gauss Quadratur
require(lme4)
fitAGQ <- try(glmer(y ~ X + (1|id), family = "binomial",
                  data = data, nAGQ = 11))
fixef(fitAGQ)

(Intercept)          X
-0.07013794  2.08657019
```

```

VarCorr(fitAGQ)

Groups Name      Std.Dev.
id      (Intercept) 0.42472

# true values for fixed effects are 0 and 2
# true variance of random effect is 1

```

(c) ##### example to change  $m$ ,  $n_i$

```

data <- simRI(m = 100, ni = 5, balanced = TRUE,
             sdInt = 1, beta = c(2), family="binomial")

# fit the GLMM via PQL
require(MASS)
require(nlme)
fitPQL <- try(glmmPQL(y ~ X, random = ~ 1|id, family = "binomial",
                    data = data, verbose = F))

fixef(fitPQL)

(Intercept)      X
0.2448828      1.7412453

VarCorr(fitPQL)

id = pdLogChol(1)
          Variance StdDev
(Intercept) 1.1817317 1.0870749
Residual    0.7955728 0.8919489

# fit the GLMM via Laplace
require(lme4)
fitLaplace <- try(glmer(y ~ X + (1|id), family = "binomial",
                      data = data))

fixef(fitLaplace)

(Intercept)      X
0.2649139      1.8977440

VarCorr(fitLaplace)

```

```
Groups Name      Std.Dev.
id      (Intercept) 1.0768
```

```
# fit the GLMM via Adaptive Gauss Quadratur
```

```
require(lme4)
```

```
fitAGQ <- try(glmer(y ~ X + (1|id), family = "binomial",
                   data = data, nAGQ = 11))
```

```
fixef(fitAGQ)
```

```
(Intercept)      X
0.2677241    1.9034867
```

```
VarCorr(fitAGQ)
```

```
Groups Name      Std.Dev.
id      (Intercept) 1.1339
```

```
# true values for fixed effects are 0 and 2
```

```
# true variance of random effect is 1
```