tramME: Mixed Effects Transformation Models Using Template Model Builder

Bálint Tamási

January 8, 2020

Abstract

Linear transformation models constitute a general family of parametric regression models for discrete and continuous responses. In the case of grouped data structures, the model has to take the into account the correlation between observations to make the inference valid. One way of doing this is to use mixed effects models, which capture the conditional distribution of the outcome variable. The tramME package of the R programming language **TODO: one sentence description refer to TMB**.

1 Introduction

TODO: transformation models, new literature, clustered observations, marginally interpretable, as opposed our approach

2 Transformation Mixed Models

TODO: describe the conditional approach Tamási (2019)

3 Applications

In this section several applications of the transformation mixed models, and wherever it is possible, we also compare to other existing implementations.

3.1 Normal Linear Mixed Model

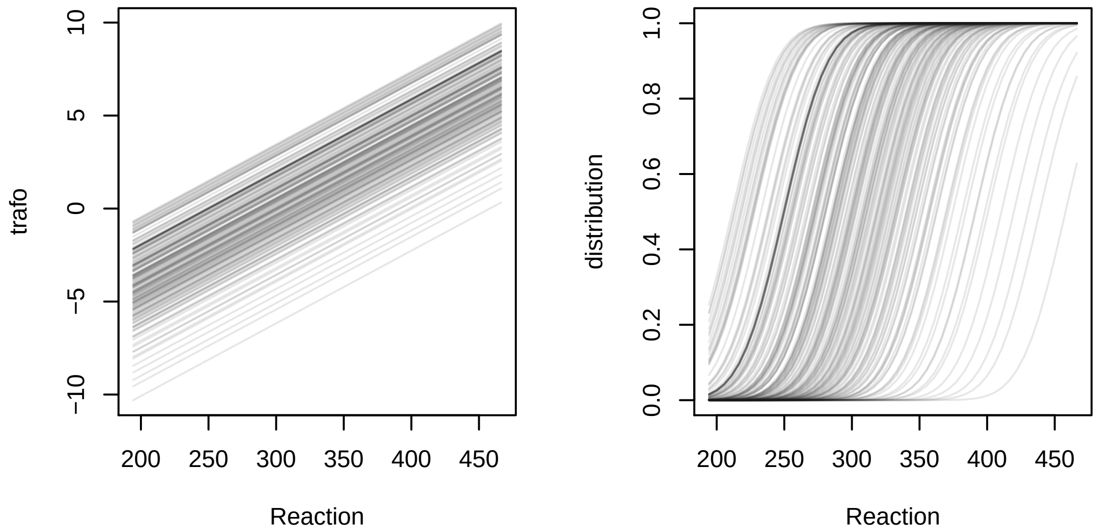
TODO: model description compared to the LMM parametrization

As a first example, we model the reaction times of a sleep deprivation study **TODO: describe in two sentences, plot data, give reference**

```
data("sleepstudy", package = "lme4")
## plot
```

```
library("tramME")
sleep_lm <- LmME(Reaction ~ Days + (Days | Subject), data = sleepstudy)
logLik(sleep_lm)

## 'log Lik.' -875.97 (df=6)
```



TODO: Compare the results to lme4 Bates et al. (2015)

```
library("lme4")
sleep_lmer <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy, REML = FALSE)
logLik(sleep_lmer)

## 'log Lik.' -875.97 (df=6)
```

Note, that we set `REML = FALSE`, as the transformation mixed model implementation only supports the maximum likelihood estimation of the normal linear mixed model specification.

The `as.lm = TRUE` option of various methods of `tramME` facilitates the comparisons between the transformation model parametrization and the results of a linear mixed model parametrization.

Coefficient estimates and their standard errors from the `tramME` estimation are

```
cbind(coef = coef(sleep_lm, as.lm = TRUE),
      se = sqrt(diag(vcov(sleep_lm, as.lm = TRUE, pargroup = "fixef"))))

##                 coef      se
## (Intercept) 251.4051 6.63228
## Days         10.4673 1.50224
```

while the results from `lmer` are

```
summary(sleep_lmer)$coefficients

##             Estimate Std. Error  t value
## (Intercept) 251.4051    6.63212 37.90719
## Days         10.4673    1.50223  6.96783
```

Similarly, the standard deviations and correlations of the random effects, as well as the standard deviations of the error terms are essentially the same

```
VarCorr(sleep_lm, as.lm = TRUE) ## random effects

##
## Grouping factor: Subject (18 levels)
## Standard deviation:
## (Intercept)        Days
##      23.780       5.717
##
## Correlations:
##      (Intercept)
## Days     0.08132
```

```
sigma(sleep_lm) ## residual SD

## [1] 25.5918

VarCorr(sleep_lmer)

##  Groups   Name        Std.Dev. Corr
##  Subject  (Intercept) 23.780
##           Days         5.717   0.08
##  Residual             25.592
```

As the results show, the transformation model approach leads to the same results as the maximum likelihood estimation of the traditional linear mixed model parametrization. **TODO: Introduce the interval-censored example, we can't do this with lme4**

```
library("survival")
ub <- ceiling(sleepstudy$Reaction / 50) * 50
lb <- floor(sleepstudy$Reaction / 50) * 50
lb[ub == 200] <- 0
sleepstudy$Reaction_ic <- Surv(lb, ub, type = "interval2")
head(sleepstudy$Reaction_ic)

## [1] [200, 250] [250, 300] [250, 300] [300, 350] [350, 400] [400, 450]
```

Estimating the normal linear model

```
sleep_lm2 <- LmME(Reaction_ic ~ Days + (Days | Subject), data = sleepstudy)
logLik(sleep_lm2)

## 'log Lik.' -200.535 (df=6)
```

The value of the likelihood is different **TODO: why?**, but the parameter estimates are close to what we got in the exactly observed case

```
cbind(coef = coef(sleep_lm2, as.lm = TRUE),
      se = sqrt(diag(vcov(sleep_lm2, as.lm = TRUE, pargroup = "fixef"))))

##                 coef      se
## (Intercept) 251.4237 6.83426
## Days         10.4886 1.62090

sigma(sleep_lm2)

## [1] 27.9617

VarCorr(sleep_lm2, as.lm = TRUE)

##
## Grouping factor: Subject (18 levels)
## Standard deviation:
## (Intercept)        Days
##      22.260       5.936
##
## Correlations:
##      (Intercept)
## Days     0.05356
```

3.2 Box-Cox-type Mixed Effects Models

Using the transformation model approach, we are able to abandon the conditional normality assumption for the outcome variable. **TODO: switching the linear baseline transformation function to a general non-linear monotone increasing one**

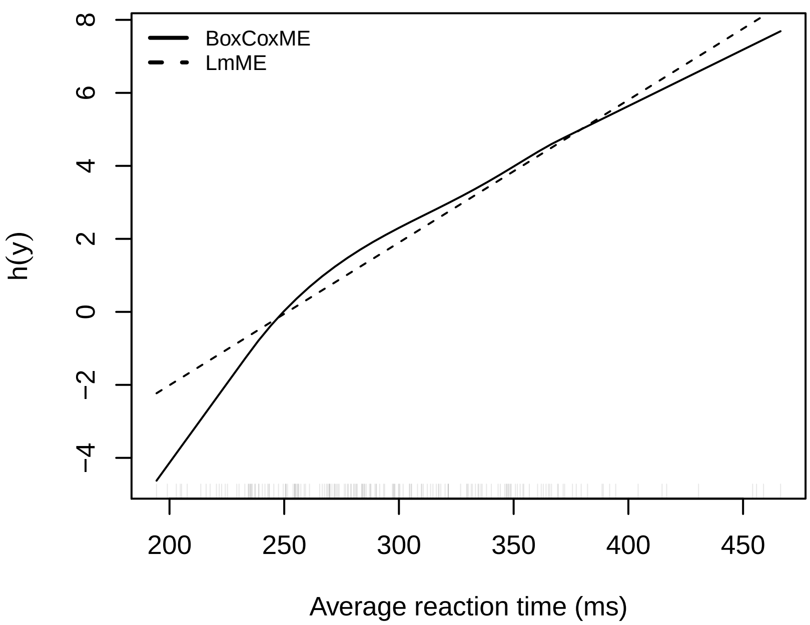
A Box-Cox type transformation mixed model can be estimated using the `BoxCoxME` function of the `tramME` addon package.

```r
sleep_bc <- BoxCoxME(Reaction ~ Days + (Days | Subject), data = sleepstudy)
logLik(sleep_bc)

## 'log Lik.' -859.545 (df=11)
```

Note, that the log-likelihood of this model is higher than in the case of the normal linear model. **TODO: At the expense of some additional parameters, we made it more flexible.**

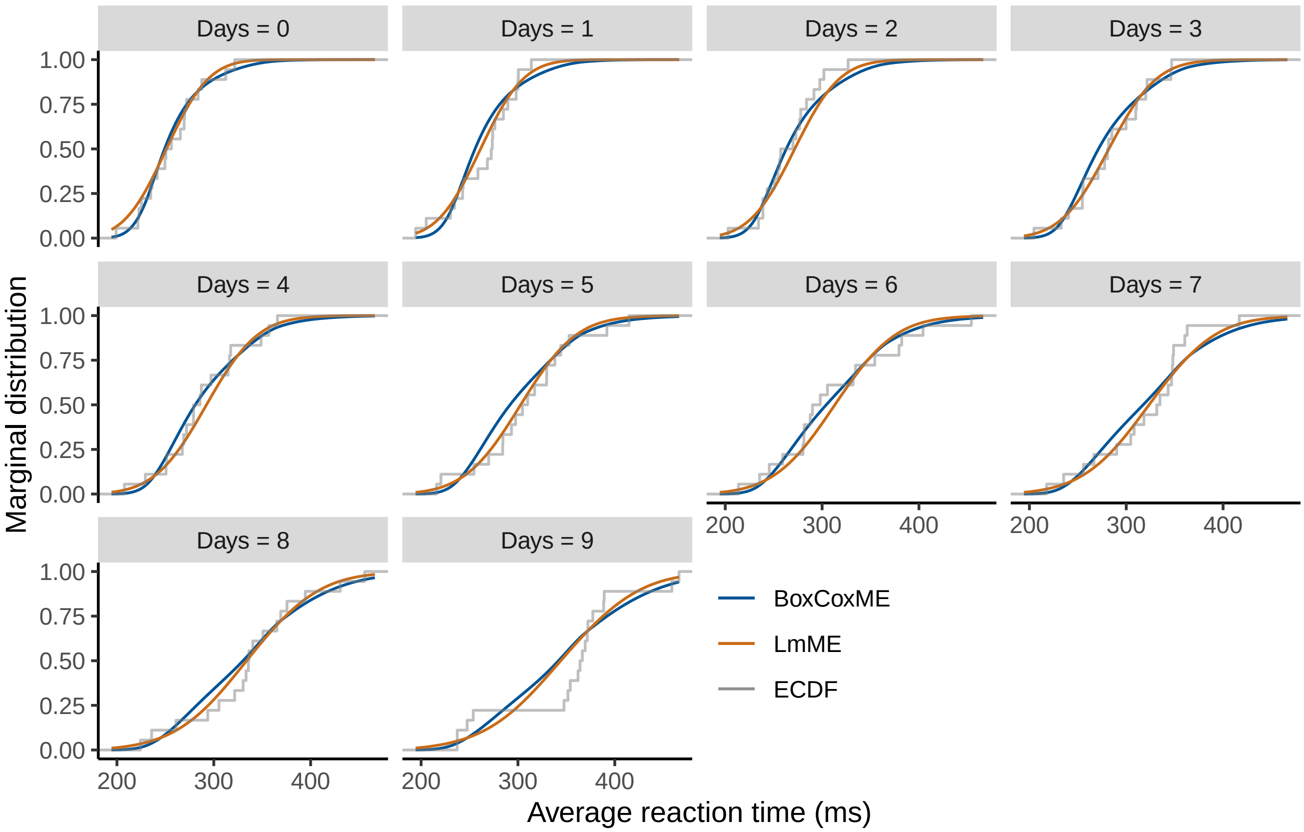
Plotting the baseline transformation $h(y) = \mathbf{a}(y)^\top \boldsymbol{\vartheta}$ **TODO: compare baseline trafo with the LmME. NOTE: kink in the lower part, slightly right skewed distribution**



TODO: integrate out: find analytic for LmME?

```r
ndraws <- 1000
nd <- expand.grid(
  Reaction = seq(min(sleepstudy$Reaction), max(sleepstudy$Reaction), length.out = 100),
  Days = 0:9,
  Subject = 1)

re <- simulate(sleep_bc, newdata = nd, nsim = ndraws, what = "ranef", seed = 100)
cp <- parallel::mclapply(re, function(x) {
  predict(sleep_bc, newdata = nd, ranef = x, type = "distribution")
}, mc.cores = 8)
cp <- array(unlist(cp), dim = c(100, 10, ndraws))
mp_bc <- apply(cp, c(1, 2), mean)
```



3.3 Mixed Effects Continuous Outcome Logistic Regression

3.4 Mixed Effects Transformation Models for Discrete Ordinal Outcomes

3.5 Mixed Effects Transformation Models for Time-to-event Outcomes

4 Discussion

TODO: Laplace approximation

TODO: Marginal vs conditional

References

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48, October 2015. ISSN 1548-7660. doi: 10.18637/jss.v067.i01. URL https://www.jstatsoft.org/index.php/jss/article/view/v067i01.

Bálint Tamási. Transformation models for correlated observations using "Template Model Builder". Master's thesis, University of Zurich, Zurich, June 2019.