Mixed effects models in gcreg

Joshua Bon

2017-11-30

A user interface for constrained mixed effects models is still under development. In the meantime we detail how to access the packages internal functions for monotone-constrained mixed effects models. Currently, these have been developed for models with block-diagonal random effects covariance matrices.

Model specification

$$(\mathcal{Y}|\mathcal{U} = \mathbf{U}) \sim \mathcal{N}\left(\mathbf{X}\beta + \mathbf{Z}\mathbf{U}, \sigma^2\mathbf{I}\right) \qquad \mathcal{U} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{G}\right)$$

where $\mathbf{X}$ is a (mean) polynomial design matrix, $\mathbf{Z}$ is the subject-specific polynomial matrix, and $\mathbf{G}$ is a block-diagonal matrix.

Fit a monotone-constrained mixed effects model

We use a sample from the Berkeley growth data set in fda (Tuddenham and Snyder, 1954). The code below details two subsets of models from the above specification

- The subject-specific curves, as defined by the random effects, are *unconstrained*.
- $r <= 2$ and the subject-specific curves are *constrained*.

```r
library(fda)
library(ggplot2)
library(reshape2)
library(dplyr)
library(gcreg)

# format data
boys <- fda::growth$hgtm %>% melt(value.name = "height", varnames = c("age","id"))

# take a random sample of 10 boys
set.seed(2017)
ids <- boys$id %>% as.character() %>% unique() %>% sample(size = 10, replace = F)
boys_s <- boys %>% filter(id %in% ids)

#scale data
b_age_sc <- gen_scale_data_funs(boys_s$age)
b_height_sc <- gen_scale_data_funs(boys_s$height)

boys_s <- boys_s %>% mutate(age2 = b_age_sc$scale(age), height2 = b_height_sc$scale(height))

# set up model we wish to fit
md <- gcreg:::make_em_model_specs(height2~age2, data = boys_s,
                                  p_degree = 8, r_degree = 3,
                                  r_constrained = F, # should random effects be constrained?
                                  mcontr_region = c(-1,1),
                                  group_name = "id"
                                  )
```

```r
# verbose = T shows the steps of the quasi-log-likelihood
boys_em <- gcreg:::constrained_lmm_em(model = md, tol = 1e-02, verbose = T, save_steps = F)
```

```
## it: 1 ql: -130.8768
## it: 2 ql: -741.1243
## it: 3 ql: -1406.33
## it: 4 ql: -1835.069
## it: 5 ql: -1958.223
## it: 6 ql: -1976.879
```

Currently gcreg:::constrained_lmm_em() returns a list with the relevant values. Below are some examples interacting with the fits.

```r
fit_points <- seq(-1,1,length.out = 201)

fitted_vals <- c(list(mean = boys_em$beta_mean),  boys_em$beta_grp) %>%
  lapply(FUN = polynomial) %>%
  sapply(FUN = predict, newdata = fit_points) %>%
  cbind(x = fit_points) %>%
  as_tibble() %>%
  melt(id = "x", variable.name = "grp", value.name = "y")

x_brks <- seq(0,18,by = 2)
y_brks <- seq(60,200,by = 20)

ggplot() +
geom_line(data = fitted_vals, aes(x = x, y = y)) +
geom_point(data = md$dat, aes(y = y,x = x),size = 0.1) +
facet_wrap(~grp) + theme_bw() +
scale_x_continuous("Age (years)", labels = x_brks, breaks = b_age_sc$scale(x_brks)) +
scale_y_continuous("Height (cm)", labels = y_brks, breaks = b_height_sc$scale(y_brks))
```
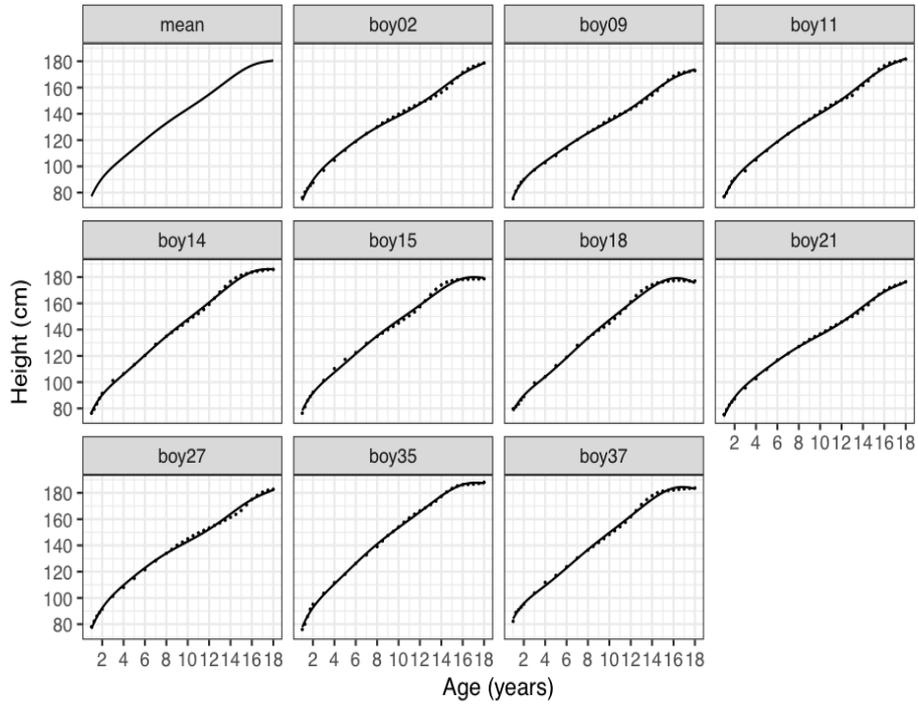
Note that some of the subject-specific curves still have non-monotonic areas around 16-18 years. If we wish to correct for this we should constrain the subject-specific curves to be monotonic. This can be done with `gcreg:::constrained_lmm_em()` if $r <= 2$ or `gcreg:::constrained_lmm_mcem()` otherwise. The latter is uses Monte Carlo Expectation Maximisation algorithm, hence is more computationally expensive.