Subgroup detection in GLMMs and GAMs

glmertrees, splinetrees and gamtrees

Marjolein Fokkema

Trees

Code and data:
https://github.com/marjoleinF/Speech-prosody-workshop-trees

Or: https://tinyurl.com/mpphwc2h

Short history of trees

Trees recursively partition the observations in a dataset based on the values of covariates, in order to find subgroups that have increasingly similar values on the response variable.

Short history of trees

Early tree methods:

- AID; Automated Interaction Detection (Morgan & Sonquist, 1963)
- CART: Classification and Regression Trees (Breiman et al., 1984)
- ID3 (Quinlan, 1986)
- C4.5 (Quinlan, 1993)

Unbiased recursive partitioning:

- GUIDE: Generalized unbiased interaction detection and estimation (Loh, 2002)
- ctree: Conditional inference trees (Hothorn, Hornik & Zeileis, 2006)
- MOB: Model-based recursive partitioning (Zeileis, Hothorn & Hornik, 2008)

Short history of trees

Model-based recursive partitioning (MOB)

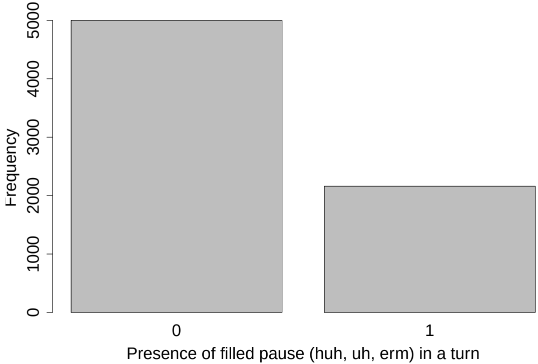
Rationale:

A single global parametric model may not fit all observations well.

- E.g., (G)LM: $y_i = x_i^\top \beta + \epsilon_i$

Example: Filled pauses

Dataset from Gardner et al. (2021), subset of Switchboard Corpus of American English.



Example: Filled pauses

```r
gmod <- glm(NFP.bi ~ 1, data = df, family = "binomial")
coef(gmod)
```

```
## (Intercept)
##  -0.8388668
```

Model-based recursive partitioning (MOB)

Rationale:

A single global parametric model may not fit all observations well.

- E.g., (G)LM: $y_i = x_i^\top \beta + \epsilon_i$

When additional covariates are available, it may be possible to partition the dataset into subgroups, and obtain better-fitting models in each of the subgroups.

- (G)LM tree: $y_i = x_i^\top \beta_j + \epsilon_i$

Example: Filled pauses

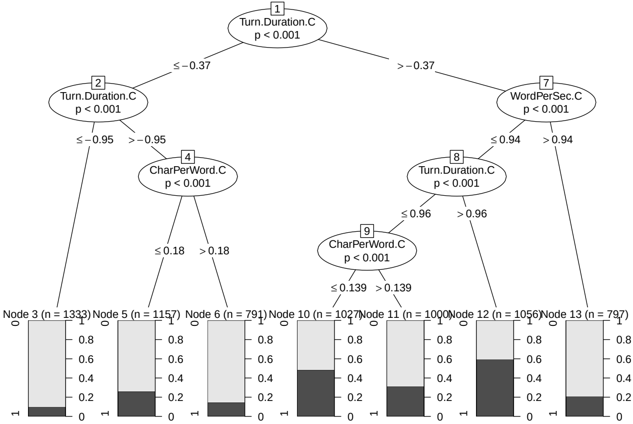
Additional covariates:

- `NVarbs`: Variable context (present or not)
- `Turn.Duration.C`: Standardized turn duration
- `CharPerWord.C`: Mean word length
- `WordPerSec.C`: Speech rate

```
library("partykit") ## Implements, amongst others, MOB
gt <- glmtree(NFP.bi ~ 1 | NVarbs + Turn.Duration.C +
                CharPerWord.C + WordPerSec.C,
              data = df, family = "binomial",
              minsize = 750)
```

Example: Filled pauses

```r
plot(gt)
```



Example: Filled pauses

```r
coef(gt)
```

```
##           3           5           6          10          11          12 
## -2.19140334 -1.04516011 -1.77126095 -0.06038835 -0.78148464  0.39521669 
##          13 
## -1.32016728 
```

MOB algorithm (Zeileis et al., 2008)

GLM-based recursive partitioning:

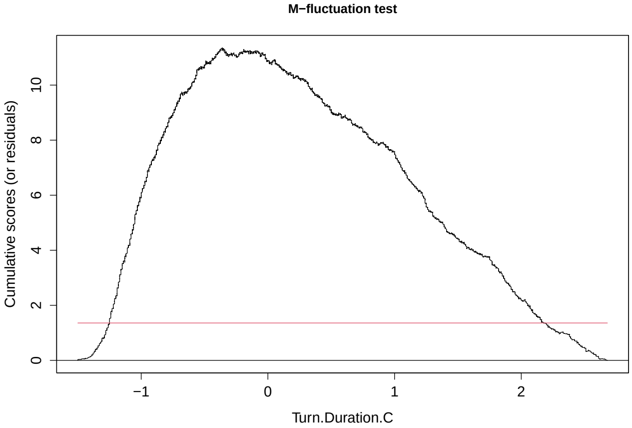
a) Fit a GLM to all observations in the current subgroup.
b) Test for instability of the GLM parameters with respect to each of the partitioning variables.
c) If there is some overall parameter instability, split the subgroup with respect to the partitioning variable associated with the highest instability.
d) Repeat Steps (a) through (c) in each of the resulting subgroups.

Step b): Parameter stability tests

```
##
## $`1`

##           NVarbs Turn.Duration.C CharPerWord.C WordPerSec.C
## statistic 24.926         519.185        60.663      125.472
## p.value    0.000           0.000         0.000        0.000

##
## $`2`

##           NVarbs Turn.Duration.C CharPerWord.C WordPerSec.C
## statistic 12.461          73.227        45.203       26.546
## p.value    0.024           0.000         0.000        0.000
```

Step b): Parameter stability tests

Computation of test statistic for `Turn.Duration.C` in the first node:



Generalized linear mixed-effects model tree (Fokkema et al., 2018; Fokkema & Zeileis, in press)

- LMM trees extend LM trees with random effects, much like the LMM extends the LM:

$$y_i = X_i \beta + Z_i b_i + \epsilon_i$$

- LMM tree:

$$y_i = X_i \beta_j + Z_i b_i + \epsilon_i$$

- Allows to account for and quantify dependence between observations within the same subjects or clusters i.

GLMM trees: Estimation

0. Initialize: Set step $r = 0$ and all random-effect estimates $\hat{b}_{i,(r)} = 0$.
1. Estimate subgroups: Set $r = r + 1$. Fit an LM tree $X_i \hat{\beta}_{j,(r)}$ using $Z_i \hat{b}_{i,(r-1)}$ as an offset. Extract the partition or subgroup memberships $j_{(r)}$.
2. Estimate full mixed-effects model: Fit the mixed-effects model $\mu_i = X_i \beta_{j,(r)} + Z_i b_{i,(r)}$ with the subgroups $j_{(r)}$ from Step 1. Extract the random-effect estimates $\hat{b}_{i,(r)}$ from the fitted model.
3. Repeat Steps 1 and 2 until convergence.

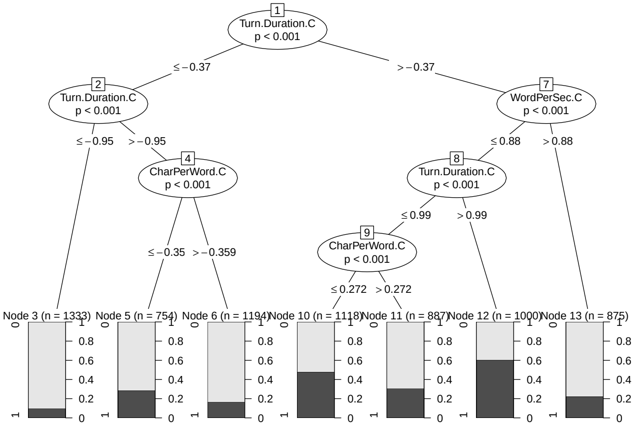
This procedure can easily be generalized to non-Gaussian responses within the GLM(M).

Example: Filled pauses

```r
library("glmertree")
glmmt <- glmertree(NFP.bi ~ 1 | (1 |Speaker_Number) |
                     NVarbs.bi + Turn.Duration.C +
                     CharPerWord.C + WordPerSec.C,
                   minsize = 750, data = df,
                   family = "binomial")
```

Example: Filled pauses

```r
plot(glmmt, which = "tree")
```



Example: Filled pauses

```r
fixef(glmmt)
```

```
##    (Intercept)
## 3   -2.4054977
## 5   -1.1114517
## 6   -1.7355811
## 10  -0.1797613
## 11  -0.8547658
## 12   0.3968831
## 13  -1.1937829
```

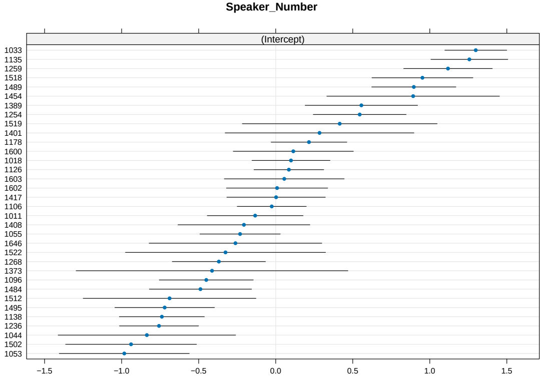
```r
VarCorr(glmmt)
```

```
##  Groups         Name        Std.Dev.
##  Speaker_Number (Intercept) 0.6716
```

Example: Filled pauses

```r
plot(glmmt, which = "ranef")
```

```
## $Speaker_Number
```



Example: Articulatory trajectories

Experimental data from Wieling (2018) with word-specific trajectories:

- 42 speakers, 130 trials.
- Predictor: Time
- Response: Standardized position for each speaker of the T1 sensor in the anterior-posterior direction (higher values, more anterior).
- Covariate: `Word` ("tent" or "tenth"), `Lang` (native language, English or Dutch)
- I added two artificial noise variables, one varying on Trial level (`xt`), one varying on Speaker level (`st`)

Example: Articulatory trajectories

Linear mixed model:

```
## (Intercept)        Time
##   0.1445023   0.5757404
```

```
##  Groups   Name        Std.Dev.
##  Speaker  (Intercept) 0.41595
##  Residual             0.89329
```

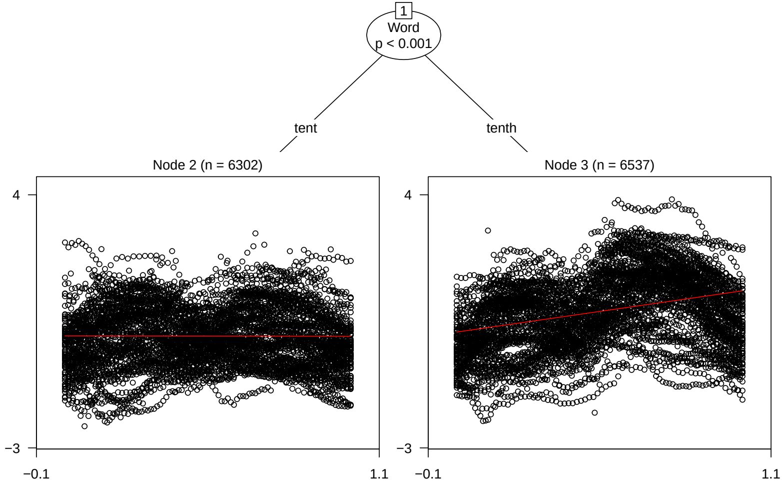
Example: Articulatory trajectories

Mixed-model tree:

```
lmmt <- lmertree(Pos ~ Time |  (1|Speaker) | Word +
                   Lang + xt + xs,
                 cluster = Trial, data = dat)
```

Example: Articulatory trajectories

```
plot(lmmt, which = "tree", fitted = "marginal")
```



Example: Articulatory trajectories

```r
coef(lmmt)
```

```
##   (Intercept)        Time
## 2  0.09672777 -0.01010434
## 3  0.20272788  1.14077046
```

```r
VarCorr(lmmt)
```

```
##  Groups   Name        Std.Dev.
##  Speaker  (Intercept) 0.4351
##  Residual             0.8096
```

Integrating splines

- The (G)LMM (or GLMM tree model) can incorporate parametric splines:

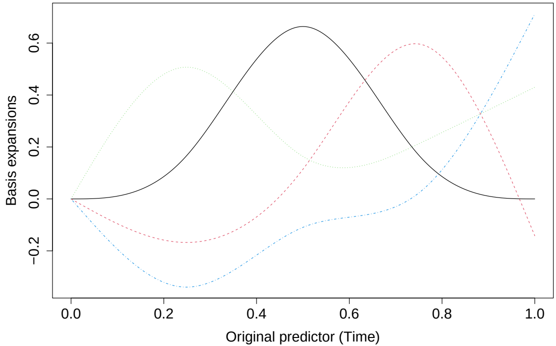
$$y_i = X_i \beta_j + Z_i b_i + \epsilon_i$$

- "Only" need to add non-linear basis functions to the design matrix X_i.

Example: Articulatory trajectories

```
library("splines")
spl_basis <- ns(dat$Time, df = 4)
```

- This sets up a spline basis, which comprises *df* non-linear functions of the original predictor variables:



Example: Articulatory trajectories

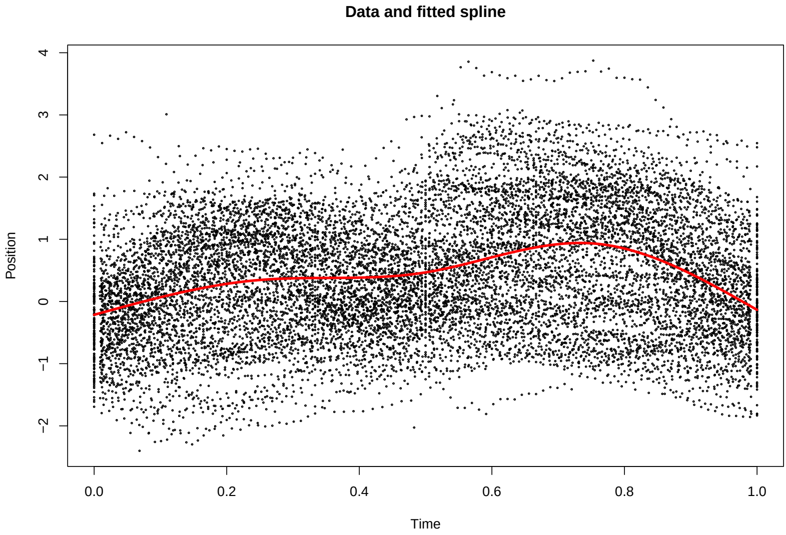
```
lmm_spl <- lmer(Pos ~ spl_basis + (1|Speaker),
                  data = dat)
fixef(lmm_spl)
```

```
## (Intercept)  spl_basis1  spl_basis2  spl_basis3  spl_basis4
##  -0.2138743   0.4389011   1.3899242   1.1945478  -0.3325012
```

```
VarCorr(lmm_spl)
```

```
##  Groups   Name        Std.Dev.
##  Speaker  (Intercept) 0.41667
##  Residual             0.85385
```

Example: Articulatory trajectories

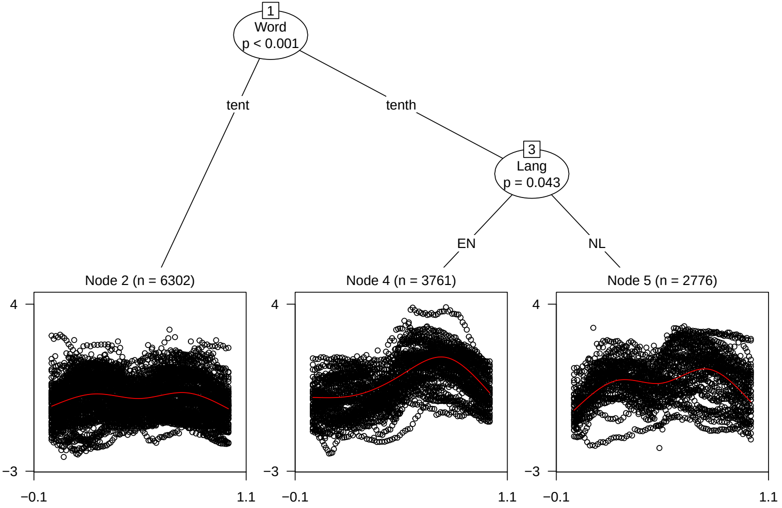


Example: Articulatory trajectories

```
library("gamtree")
sp <- splinetree(Pos ~ ns(Time, df = 4) | (1|Speaker) |
                   Word + Lang + xs + xt,
                 data = dat, cluster = Speaker)
```

Example: Articulatory trajectories

```r
plot(sp, which = "tree", fitted = "marginal")
```



Example: Articulatory trajectories

```r
fixef(sp)
```

```
##   (Intercept) spline.Time1 spline.Time2 spline.Time3 spl
## 2  -0.2768570   0.07866561     0.653180    0.8529127   -
## 4   0.0941362   0.85680121     2.270792    0.8279382
## 5  -0.4447736   0.68939608     1.859992    2.4555487   -
```

```r
VarCorr(sp)
```

```
##  Groups   Name        Std.Dev.
##  Speaker  (Intercept) 0.44710
##  Residual             0.74111
```