Workshop Speech Prosody. Part II: GLMM and Spline Trees for Longitudinal Data

Marjolein Fokkema

In principle, all model-specification adjustments of Part I of the workshop (e.g., more complex random-effects specifications, correcting for the effects of predictors of a-priori known relevance) can also be applied in the examples below.

```
library("glmertree")
library("gamtree")
```

Example dataset 2: Speech trajectories

We use a dataset from Wieling (2018) on articulatory trajectories. We load and inspect the data as follows:

```
load("full2.rda")
summary(full)
```

```
##        Speaker        Lang           Word       Sound        Loc
##  VENI_EN_10:  4302   EN:71152   thighs : 7199   T :62428   Final:62052
##  VENI_EN_19:  3867   NL:55025   tongs  : 7127   TH:63749   Init :64125
##  VENI_EN_17:  3860              fort   : 7000
##  VENI_EN_21:  3739              ties   : 6840
##  VENI_EN_2 :  3717              forth  : 6727
##  VENI_EN_11:  3714              thongs : 6640
##  (Other)   :102978              (Other):84644
##      Trial             Time             Pos              Pos01
##  355    :   922   Min.   :0.0000   Min.   :-6.7536   Min.   :-0.9791
##  317    :   887   1st Qu.:0.2466   1st Qu.:-0.3441   1st Qu.: 0.4235
##  300    :   883   Median :0.5000   Median : 0.3214   Median : 0.5582
##  305    :   864   Mean   :0.5000   Mean   : 0.3029   Mean   : 0.5555
##  301    :   843   3rd Qu.:0.7534   3rd Qu.: 1.0126   3rd Qu.: 0.6983
##  137    :   816   Max.   :1.0000   Max.   : 5.5685   Max.   : 1.6425
##  (Other):120962
##        xs                  xt
##  Min.   :-2.656455   Min.   :-2.99309
##  1st Qu.:-0.564698   1st Qu.:-0.72922
##  Median :-0.094659   Median :-0.02276
##  Mean   : 0.009832   Mean   :-0.03114
##  3rd Qu.: 0.704837   3rd Qu.: 0.65035
##  Max.   : 2.286645   Max.   : 2.96587
##
```

```r
sapply(full, class)
```

```
##   Speaker      Lang      Word     Sound       Loc     Trial      Time       Pos
##  "factor"  "factor"  "factor"  "factor"  "factor"  "factor" "numeric" "numeric"
##     Pos01        xs        xt
## "numeric" "numeric" "numeric"
```

`Speaker` is an indicator for respondent. `Lang` is an indicator for the native language of the respondent. `Word` is an indicator for the word being pronounced. `Sound` is an indicator for whether the word contains “th” or “t”. `Loc` is an indicator for whether the “t” or “th” is at the beginning or end of the word, `Trial` is an indicator for trial, `Time` is a normalized indicator for time. Finally, `Pos` is our response variable of interest: The anterior-posterior position of the a T1 sensor, positioned about 0.5-1 cm behind the tongue tip) during pronounciation.

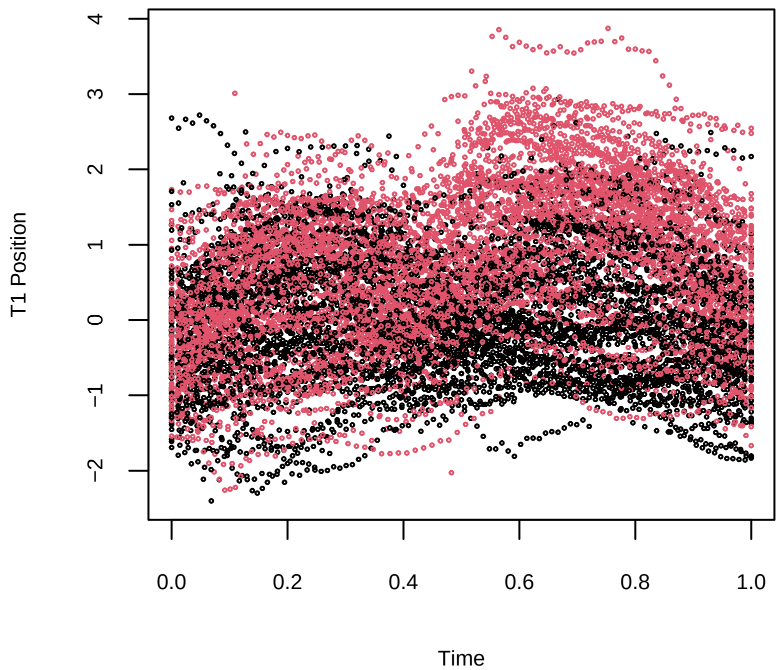
These data are from a carefully structured experiment, where the predictors or relevance are actually already known. For the sake of the current example, which aims to illustrate exploratory analyses, and whether we can detect relevant subgroups, we ignore that we already know which are the predictors of relevance. Furthermore, I added two artificial noise variables to the dataset, to illustrate how the ‘true’ predictors are likely to be picked up by the tree algorithms: `xs`, which is a noise variable that varies at the Speaker level, and `xt` which is a noise variable which varies at the Trial level.

We select a subset of trajectories, where the words “tent” and “tenth” were pronounced:

```r
words <- c("tent","tenth")
dat <- droplevels(full[full$Word %in% words,])
```

We can inspect the dataset, for example, by creating a scatter plot, with the word represented by color:

```r
plot(dat$Time, dat$Pos, cex = .25, xlab = "Time", ylab = "T1 Position",
     col = dat$Word, cex.lab = .7, cex.axis = .7)
```



Fitting trees for linear trajectories

Following Wieling (2018), who first fitted linear models, we also start with subgroup detection in a linear model. Although the scatterplot above already suggests that a non-linear approach should probably be preferred. It does, however, allow us to illustrate how GLMM trees can be fitted to linear trajectories.

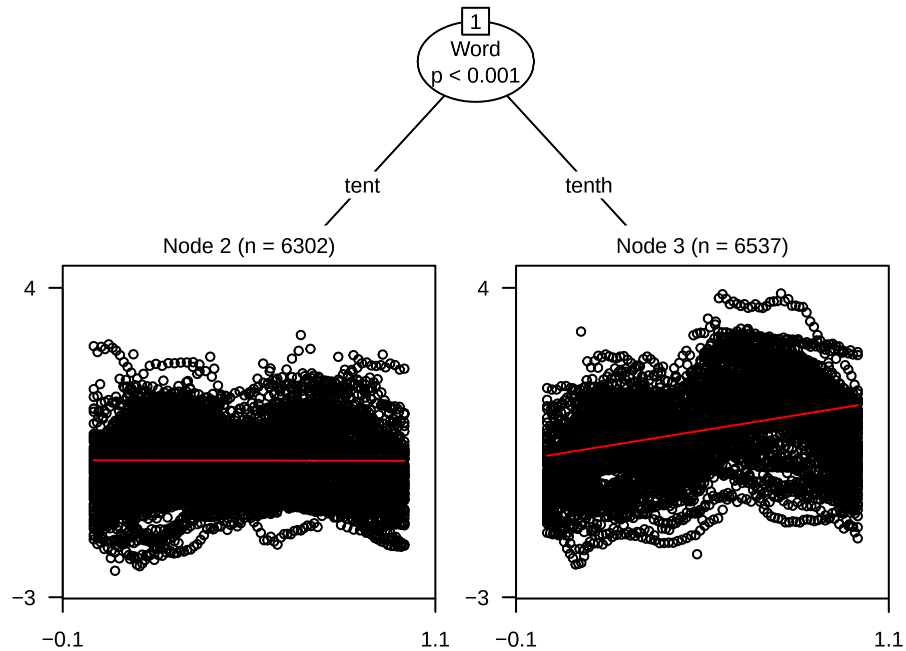
In Fokkema et al. (in press), we found that when fitting GLMM trees to longitudinal trajectories, it is critical to adjust the parameter stability tests. By default, the tests assume that the partitioning variables are measured at the lowest level of observations; in most cases this would mean that they are time-varying. Power likely becomes inflated if the partitioning variables are measured at a higher (or time-constant) level such as trial (such as `Word` and `xt` in the current example) or respondent characteristics (such as `Lang` and `xs` in the current example).

We can account for the measurement level of the partitioning variables using the `cluster` argument. However, only 1 level can be specified, so we immediately have to make a difficult choice, because we want to investigate the effect of variables that were measured at both trial and speaker level. Here, I use the `Trial` indicator.

With longitudinal data, we may want to increase the minimum of observations per node, which by default is set at a minimum of 10. This may not work well for situations with time-intensive data such as in acoustic trajectories. A minimum node size of 10 could theoretically split up a single trial into multiple parts, which should likely be avoided. To avoid isolating individual (or parts of) trials, it is good practice to multiply the default minimum number of observations in a terminal node by the (average) number of observations in a trajectory. Wieling (2018) reports this average to be 78 for these data, so we set `minsize = 780`. This setting may often be inconsequential for the resulting tree, but it may counter overfitting, especially when we fit more flexible spline-based models.

In line with the analyses of Wieling (2018), we estimate a random intercept with respect to subject (but add no random slope for `Word`, as here we aim to capture this effect with the tree structure instead):

```
lt <- lmertree(Pos ~ Time | (1|Speaker) | Word + Lang + xs + xt,
               data = dat, cluster = Trial, minsize = 780)
plot(lt, which = "tree", fitted = "marginal", gp = gpar(cex = .7))
```



Our results largely correspond with the result of Wieling (2018), who found a higher average value (reflecting a more anterior position) for "tenth" compared to "tent". We find the strongest difference in curves for the word being pronounced, the native language may have a too small effect to be picked up in our (mixed-effects) linear model tree. The noise variables were rightly ignored by the tree.

Furthermore, the model indicates substantial random-intercept variance (e.g., when compared to the residual variance), reflecting systematic differences in tongue position between speakers:

```
VarCorr(lt)
```

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

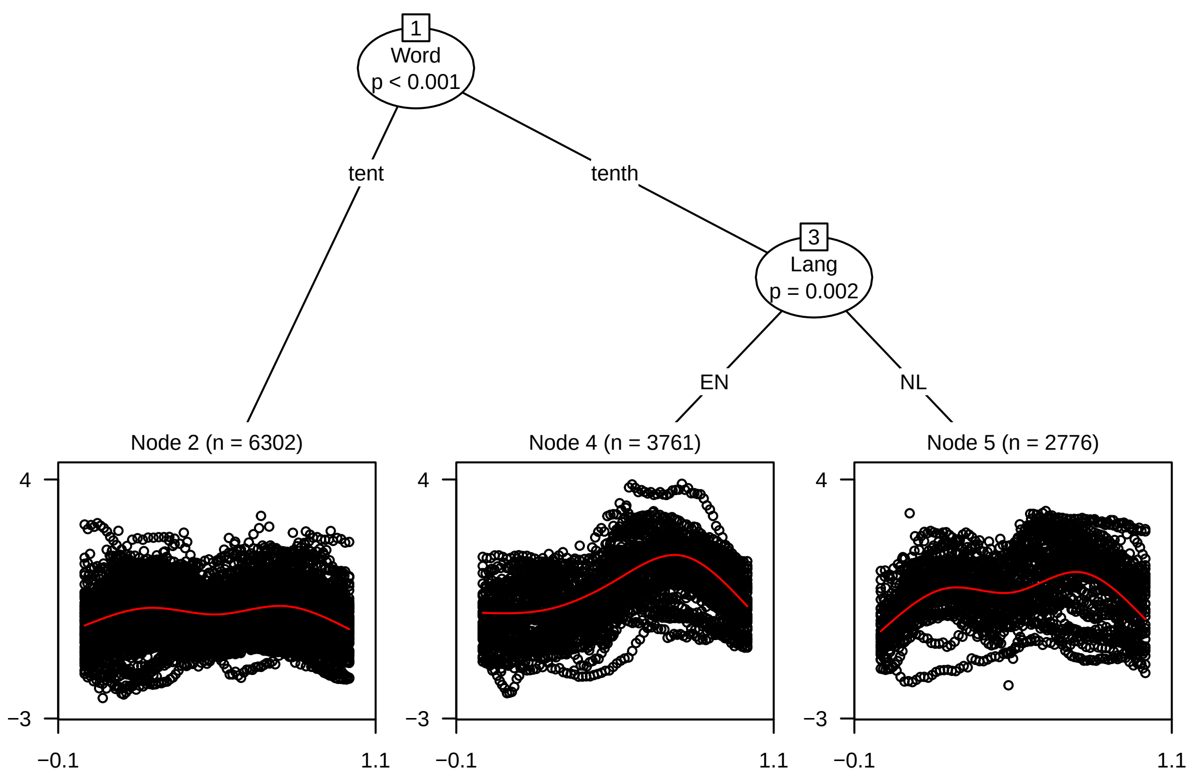
Fitting trees for non-linear trajectories: Parametric spline trees

The linear trajectories obviously do not capture the trajectory shapes well. We therefore incorporate parametric splines in the LMM trees. The most straightforward way to do this is through using function `splinetree` from package **`gamtree`**. It currently allows for specifying natural (using function `ns`) and B-splines (using function `bs`). The latter are often referred to as cubic splines and in accordance, the `bs` function sets up a cubic spline, by default. Similarly, **ns** sets up a natural cubic spline, which is a slightly more restricted version of a cubic spline. For readers unfamiliar with naturaral and B-splines, chapter 7 from James et al. (2013) provides an accessible introduction.

To apply the `ns` or `bs` functions, a user generally only needs to specify the degrees of freedom to be used by the spline. Unlike with penalized, or semi-parametric splines as fitted with package **mgcv**, it is best to keep the degrees of freedom of a parametric spline low, because there is no automatic smoothness selection. For many situations, splines of at most 6 degrees (but often less) suffice to adequately capture non-linearities in the data. One could start from a lower-df spline and increase the df if visual inspection suggests the fit is not flexible enough. In general, the higher the df, the higher the likelihood of overfitting or spurious findings, so high df values should be avoided, or results verified by replicating with a lower value for the df.

Function `splinetree` is basically a wrapper around functions `lmertree` and `glmertree`, that automatically sets up the spline basis so it can be used in partitioning, and so that the fitted trees can later be used for prediction. The only difference is that the function expects a spline specification using `ns` or `bs` before the first vertical bar of the model formula, e.g.:

```
st <- splinetree(Pos ~ ns(Time, df = 4) | (1|Speaker) | Word + Lang + xs + xt,
                 data = dat, cluster = Trial, minsize = 780)
plot(st, which = "tree", gp = gpar(cex = .7))
```




```
VarCorr(st)
```

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

The results correspond with the results of Wieling (2018), who found an effect of native language in addition to an effect of word. With the word “tent”, there is no difference picked up between native EN and NL speakers. With the word “tenth”, this difference has been picked up. The noise variables were rightly ignored by the tree.

References

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Moving Beyond Linearity (chapter 7). In: An Introduction to Statistical Learning. New York: Springer. Freely available from: https://www.statlearning.com/

Fokkema, M., & Zeileis, A. (in press). Subgroup detection in linear growth curve models with generalized linear mixed model (GLMM) trees. *Behavior Research Methods*. https://doi.org/10.3758/s13428-024-02389-1

Wieling, M. (2018). Analyzing dynamic phonetic data using generalized additive mixed modeling: A tutorial focusing on articulatory differences between L1 and L2 speakers of English. *Journal of Phonetics, 70*, 86-116. https://doi.org/10.1016/j.wocn.2018.03.002