Analyzing EEG data using GAMs

Martijn Wieling

Department of Humanities Computing, University of Groningen
Seminar für Sprachwissenschaft, University of Tübingen

Seminar in Methodology and Statistics, April 30, 2014
Overview

- Introduction
  - Some words about linear regression
  - Generalized additive mixed-effects regression modeling
  - ERPs to study gender violations
  - Research question

- Design

- Methods: \texttt{R} code

- Results

- Discussion
A linear regression model

- **linear model**: linear relationship between predictors and dependent variable: \( y = a_1 x_1 + ... + a_n x_n \)
  - Non-linearities via explicit parametrization: \( y = a_1 x_1^2 + a_2 x_1 + ... \)
  - Interactions not very flexible
A generalized additive model (1)

- **generalized additive model (GAM):** relationship between individual predictors and (possibly transformed) dependent variable is estimated by a non-linear smooth function: \( g(y) = s(x_1) + s(x_2, x_3) + a_4 x_4 + \ldots \)
- multiple predictors can be combined in a (hyper)surface smooth
A generalized additive model (2)

- Advantage of GAM over manual specification of non-linearities: the optimal shape of the non-linearity is determined automatically
  - appropriate degree of smoothness is automatically determined on the basis of cross validation to prevent overfitting

- Choosing a smoothing basis
  - Single predictor or isotropic predictors: thin plate regression spline
    - Efficient approximation of the optimal (thin plate) spline
  - Combining non-isotropic predictors: tensor product spline

- Generalized Additive Mixed Modeling:
  - Random effects can be treated as smooths as well (Wood, 2008)
  - R: `gam` and `bam` (package `mgcv`)

- For more (mathematical) details, see Wood (2006)
ERPs to study grammatical gender violations

- A P600 (a positivity ‘around’ 600 ms. after stimulus onset) is sensitive to grammatical violations
- An N400 (a negativity ‘around’ 400 ms. after stimulus onset) is modulated by semantic context and lexical properties of a word
- The P600/N400 are found by comparing the incorrect sentences to the correct sentences

- Native speakers appear to show a P600 for gender violations
  - But analyzed by averaging over items and over subjects!
- Here we are interested in how non-native speakers respond to gender violations (joint work with Nienke Meulman)

- Gender is very hard to learn for L2 learners
- Even though behaviorally L2 learners might show correct responses, the brain may reveal differences in processing gender
Research question

- Is the P600 for gender violations dependent on age of arrival for the L2 learners of German?
ERP data

- Today: analysis of single region of interest (ROI 8)
  - GAMs allow for spatial distribution analyses
Design

- 67 L2 speakers of German (Slavic L1)
- Auditory presentation of correct sentences or sentences with a gender violation (incorrect determiner; no determiners in L1)
- 48 items in each condition: 96 trials per participant (minus artifacts)

Example:

Nach der Schlägerei ist das/*der Auge des Angestellten von der Krankenschwester versorgt worden.

After the fight the_{neut}/*the_{masc} eye of the worker was treated by the nurse.
Data overview

```r
> load('dat.rda')

# data needs to be sorted to test for autocorrelation problems
> dat = dat[order(dat$Subject, dat$TrialNr, dat$Time, decreasing=F),]

> head(dat)

mV  Time  Subject  Group  Word  TrialNr  Roi
44947  8.94400  505  GL102  GenEarly  Wald  2  post.mid
28121 15.56267  515  GL102  GenEarly  Wald  2  post.mid
40909 21.30807  525  GL102  GenEarly  Wald  2  post.mid
1294  13.31573  535  GL102  GenEarly  Wald  2  post.mid
9446  19.10700  545  GL102  GenEarly  Wald  2  post.mid
81324 17.95607  555  GL102  GenEarly  Wald  2  post.mid

Structure  Correctness  L1  AoArr  LoR  Age  Edu  SeqStart
44947        DN  incor  PL   8  24  32  3       TRUE
28121        DN  incor  PL   8  24  32  3       FALSE
40909        DN  incor  PL   8  24  32  3       FALSE
1294         DN  incor  PL   8  24  32  3       FALSE
9446         DN  incor  PL   8  24  32  3       FALSE
81324        DN  incor  PL   8  24  32  3       FALSE

> dim(dat)
[1] 442160     15
```
A lot of individual variation
(the signal has been downsampled to 100 Hz)
General patterns exist (note the arbitrary age split, however)
A first model: a general effect of time
(to prevent too wiggly curves split up data in 2 partially overlapping time spans: here 500-1300 ms)

```r
> m0 = bam(mV ~ s(Time), data=dat, gc.level=2, method='ML')
> summary(m0)

Family: gaussian
Link function: identity

Formula:
mV ~ s(Time)

Parametric coefficients:
  Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.06120  0.02774  -2.206  0.0274 *

Approximate significance of smooth terms:
  edf Ref.df  F p-value
s(Time) 3.917 4.84 25.95 <2e-16 ***

R-sq.(adj) =  0.000284  Deviance explained =  0.0293%
ML score = 1.9162e+06  Scale est. = 340.13   n = 442160
```
Visualizing the time pattern

```r
> par(mfrow=c(1,2)) # 2 plots in one window
> plot(m0, rug=F, shade=T, ylim=c(0.5,-1), seWithMean=T,
     ylab='mV', main='m0: partial effect')
> abline(h = 0) # horizontal line at y = 0
> ic = m0$coef[("(Intercept)")] # intercept (-0.06) from model
> plot(m0, rug=F, shade=T, ylim=c(0.5-ic,-1-ic), seWithMean=T, shift = ic,
     ylab='mV', main='m0: shifted by intercept')
> abline(h = 0)
```

![m0: partial effect](image1)

![m0: shifted by intercept](image2)
Check for additional wigglyness

(if p-value is low and edf close to k'; default for s: 10^{\dim} - 1, default for te: 5^{\dim} - 1)

```r
> gam.check(m0)

Method: ML   Optimizer: outer newton
full convergence after 2 iterations.
Gradient range [-0.6306798,0.1689593]
(score 1916153 & scale 340.1335).
Hessian positive definite, eigenvalue range [1.948326,221080.6].

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

<table>
<thead>
<tr>
<th>k'</th>
<th>edf</th>
<th>k-index</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(\text{Time})</td>
<td>9.000</td>
<td>3.917</td>
<td>0.993</td>
</tr>
</tbody>
</table>
```
Increasing the wigglyness of a smooth with $k$
(double $k$ if higher $k$ is needed, but do not set it too high, i.e. max $\frac{1}{2} \times$ unique time points)

```r
> m0b = bam(mV ~ s(Time, k=20), data=dat, gc.level=2, method='ML')
> summary(m0b)
```

Family: gaussian
Link function: identity

Formula:
mV ~ s(Time, k = 20)

Parametric coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) -0.06119 | 0.02774 | -2.205 | 0.0274 * |

Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Time) 4.038</td>
<td>5.04</td>
<td>24.94</td>
<td>&lt;2e-16 ***</td>
</tr>
</tbody>
</table>

R-sq.(adj) = 0.000284  Deviance explained = 0.0293%
ML score = 1.9162e+06  Scale est. = 340.13  n = 442160
Assessing correct versus incorrect
(smooths are centered, so the factorial predictor also needs to be included in the fixed effects)

```r
> m1 = bam(mV ~ s(Time, by=Correctness) + Correctness, data=dat, gc.level=2, method='ML')
> summary(m1)

Family: gaussian
Link function: identity

Formula:
mV ~ s(Time, by = Correctness) + Correctness

Parametric coefficients:

|                | Estimate | Std. Error | t value | Pr(>|t|) |
|----------------|----------|------------|---------|----------|
| (Intercept)    | -0.50692 | 0.03920    | -12.93  | <2e-16 *** |
| Correctnessincor | 0.89174  | 0.05546    | 16.08   | <2e-16 *** |

Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th></th>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Time):Correctnesscor</td>
<td>1.043</td>
<td>1.085</td>
<td>13.35</td>
<td>0.000194 ***</td>
</tr>
<tr>
<td>s(Time):Correctnessincor</td>
<td>4.370</td>
<td>5.375</td>
<td>46.70</td>
<td>&lt; 2e-16 ***</td>
</tr>
</tbody>
</table>

R-sq.(adj) = 0.00117 Deviance explained = 0.118%
ML score = 1.916e+06 Scale est. = 339.83 n = 442160
Visualizing both patterns: partial effects

```r
> par(mfrow=c(1,2))
> plot(m1, select=1, shade=T, rug=F, ylim=c(2,-2), shade.col='green',
      seWithMean=T, main='m1: correct'); abline(h = 0)
> plot(m1, select=2, shade=T, rug=F, ylim=c(2,-2), shade.col='red',
      seWithMean=T, main='m1: incorrect'); abline(h = 0)
```
Visualizing both patterns: complete model context

> source('plotting.R')  # custom plotting functions
> plotSmooths(m1, "Time", "Correctness", colors=c('green','red'))
# Note that if there are additional variables in the model
# plotSmooths sets these to the specified value (via the 'cond' parameter)
# or to their mean / most frequent value.
Visualizing the differences

> plotDiff(m1, "Time", "Correctness")

Difference between cor and incor
Model comparison

- For comparing differences in the fixed effects (constant random effects) of two models use `method="ML"`
- For comparing differences in the random effects (constant fixed effects) of two models use `method="fREML"` (bam only) or `method="REML"`

- There are three possibilities for comparing GAM models:
  - Look at the $p$-values of the model summary (recommended)
  - Compare AIC values (at least reduction of 2): $AIC(m_0) - AIC(m_1)$
  - Compare (fRE)ML values: `compareML(m0,m1)`

- The table below shows when which method can be applied:
  (N.B. rho is used to correct for autocorrelation in the residuals and explained later)

<table>
<thead>
<tr>
<th></th>
<th>Gaussian without rho</th>
<th>Gaussian with rho</th>
<th>Non-Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>gam</code></td>
<td>(RE)ML / AIC / summary</td>
<td>-</td>
<td>(RE)ML / AIC / summary</td>
</tr>
<tr>
<td><code>bam</code></td>
<td>(fRE)ML / AIC / summary</td>
<td>(fRE)ML / summary</td>
<td>AIC / summary</td>
</tr>
</tbody>
</table>
Model comparison and model fitting

```r
> source('compareML.R') # custom function for comparing ML/(f)REML scores
> compareML(m0,m1)

m0: mV ~ s(Time)
m1: mV ~ s(Time, by = Correctness) + Correctness

Chi-square test of ML scores
-----
<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
<th>Edf</th>
<th>Chisq</th>
<th>Df</th>
<th>p.value</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>m0</td>
<td>1916153</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m1</td>
<td>1915957</td>
<td>6</td>
<td>196.169</td>
<td>3.000</td>
<td>&lt; 2e-16</td>
<td>***</td>
</tr>
</tbody>
</table>

AIC difference: 389.52, model m1 has lower AIC.

▶ Different fitting methods for GAMs (method="...":
  ▶ ML: conservative, but biased variance components (oversmoothing)
  ▶ fREML/REML: better smooths, but less conservative p-values
  ▶ GCV.Cp: good for prediction, but not robust to autocorrelation problems

▶ I generally use method="ML" for my final model (conservative)
Explicit test for significant difference between levels
(smooths for a binary by-variable are not centered, so no change to the fixed effects)

```r
> dat$IsIncorrect = (dat$Correctness == 'incor')*1
> m1b = bam(mV ~ s(Time) + s(Time, by=IsIncorrect), data=dat, gc.level=2, method='ML')
> summary(m1b)

Family: gaussian
Link function: identity

Formula:
mV ~ s(Time) + s(Time, by = IsIncorrect)

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.5069     0.0392  -12.93 <2e-16 ***

Approximate significance of smooth terms:

                  edf Ref.df   F  p-value
s(Time)           1.022   1.042 13.11 0.000257 ***
s(Time):IsIncorrect 5.239   6.221 81.03 < 2e-16 ***

R-sq.(adj) =  0.00117   Deviance explained = 0.118%
ML score = 1.916e+06  Scale est. = 339.83     n = 442160
```
Visualizing correct and incorrect difference
(identical shape as the difference curve of model m1)

> plot(m1b, shade=T, rug=F, ylim=c(1.8,-1.2), select=2,
     main='Difference m1b', ylab='mV', seWithMean=T)
> abline(h = 0)
Changing the reference level
(note that the model becomes more complex in this case)

```r
> dat$IsCorrect = (dat$Correctness == 'cor')*1
> m1c = bam(mV ~ s(Time) + s(Time, by=IsCorrect), data=dat, gc.level=2, method='ML')
> summary(m1c)

... Parametric coefficients:
   Estimate Std. Error t value  Pr(>|t|)
(Intercept)   0.38389    0.03922   9.788 <2e-16 ***

Approximate significance of smooth terms:
   edf   Ref.df   F  p-value
s(Time)     4.698     5.731 43.10 <2e-16 ***
s(Time):IsCorrect 4.778     5.656 69.32 <2e-16 ***

R-sq.(adj) =  0.00118  Deviance explained = 0.12%
ML score = 1.916e+06  Scale est. = 339.83  n = 4421600
```
Visualization with different reference level

```r
> par(mfrow=c(1,2))
> plotSmooths(m1c, "Time", "IsCorrect", colors=c('red','green'))
> plot(m1c, shade=T, rug=F, select=2, main='Difference',
>      ylab='mV', ylim=c(1.2,-1.8), seWithMean=T)
> abline(h = 0)
```

![Model: m1c](image1)

![Difference](image2)
Comparing models with a different reference level

- m1c is more complex than m1b and also results in a worse fit
  - Thus: the reference level matters with binary contrasts!

> compareML(m1b, m1c)

m1b: mV ~ s(Time) + s(Time, by = IsIncorrect)
m1c: mV ~ s(Time) + s(Time, by = IsCorrect)

Chi-square test of ML scores

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
<th>Edf</th>
<th>Chisq</th>
<th>Df</th>
<th>p.value</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 m1c</td>
<td>1915964</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 m1b</td>
<td>1915959</td>
<td>6</td>
<td>5.102</td>
<td>0.000</td>
<td>&lt; 2e-16</td>
<td>***</td>
</tr>
</tbody>
</table>

AIC difference: -0.64, model m1b has lower AIC.
Separating intercept and smooth difference
(the binary by-variable combines the intercept and smooth difference)

> dat$CorrectnessO = as.ordered(dat$Correctness)
> contrasts(dat$CorrectnessO) = 'contr.treatment' # contrast treatment
> m1d = bam(mV ~ s(Time) + s(Time,by=CorrectnessO) + CorrectnessO, data=dat,
          gc.level=2, method='ML')
> summary(m1d)
... 
Formula:
mV ~ s(Time) + s(Time, by = CorrectnessO) + CorrectnessO

Parametric coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -0.50671    | 0.03920  | -12.93  | <2e-16 *** |
| CorrectnessOincor | 0.89148    | 0.05546  | 16.07   | <2e-16 *** |

Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Time)</td>
<td>1.510</td>
<td>1.851</td>
<td>6.994</td>
</tr>
<tr>
<td>s(Time):CorrectnessOincor</td>
<td>4.258</td>
<td>5.241</td>
<td>36.513</td>
</tr>
</tbody>
</table>

R-sq.(adj) = 0.00117  Deviance explained = 0.119%
ML score = 1.916e+06  Scale est. = 339.83  n = 442160
Visualizing correct and incorrect difference

```r
> plot(m1b, shade=T, rug=F, ylim=c(1.8,-2.0), select=2,
    main='Difference m1b', ylab='mV', seWithMean=T)
> abline(h = 0)
> plot(m1d, shade=T, rug=F, ylim=c(1.8,-2.0), select=2,
    main='Difference m1d (without intercept)', ylab='mV', seWithMean=T)
> abline(h = 0)
```
Taking into account individual variation

- The effect of time is non-linear and is variable per subject
- We need a random intercept/slope which is non-linear
  - (instead of random intercepts and slopes of time per subject)

```r
> summary(m2 <- bam(mV ~ s(Time,by=Correctness) + Correctness +
  s(Time,Subject,bs='fs',m=1), data=dat,
  gc.level=2, method='ML'))
...
```

**Parametric coefficients:**

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -0.55611 | 0.29553 | -1.882 | 0.0599 . |
| Correctnessincor | 0.88609 | 0.05493 | 16.132 | <2e-16 *** |

**Approximate significance of smooth terms:**

<table>
<thead>
<tr>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Time):Correctnesscor</td>
<td>1.295</td>
<td>1.458</td>
<td>2.901</td>
</tr>
<tr>
<td>s(Time):Correctnessincor</td>
<td>4.220</td>
<td>5.145</td>
<td>26.494</td>
</tr>
<tr>
<td>s(Time,Subject)</td>
<td>229.770</td>
<td>602.000</td>
<td>14.490</td>
</tr>
</tbody>
</table>

**R-sq.(adj) =  0.0205  Deviance explained =  2.1%**

**ML score = 1.9119e+06  Scale est. = 333.26  n = 442160**
Lots of individual variation

> plot(m2, select=3)
Smoothes become more uncertain

```r
> par(mfrow=c(1,2))
> plotSmooths(m2, "Time", "Correctness", colors=c('green','red'),
               dropRanef="Subject")
> plotDiff(m2, "Time", "Correctness")
```

Model: m2

Difference between cor and incor

![Graph showing the comparison between correct and incorrect responses over time.](image-url)
Autocorrelation in the data is a huge problem!
(residuals should be independent, otherwise the standard errors and p-values are wrong)

# It is essential that the data used in m2 is sorted per individual EEG signal sequence (per subject, per trial, per ROI, etc.). The best rho value can be found by trying different values and comparing models. The autocorrelation at lag 1 for the same model w/o rho seems to work well, however. A good practice is to use the lowest value possible which has a good acf plot and is not significantly worse than the best rho model.
> m2acf = acf(resid(m2)) # show autocorrelation
> rho = as.vector(m2acf[1]$acf)
> rho
[1] 0.9123131 # correlation of residuals at time t with those at time t-1
Correcting for autocorrelation

(rho can only be used with bam, not with gam)

```r
> m3 = bam(mV ~ s(Time,by=Correctness) + Correctness +
          s(Time,Subject,bs='fs',m=1), data=dat,
          gc.level=2, method='ML', rho=rho, AR.start=SeqStart)
> summary(m3)

... Parametric coefficients:

| Estimate  | Std. Error | t value | Pr(>|t|) |
|------------|------------|---------|----------|
| (Intercept)| -0.5601    | 0.3169  | -1.768   | 0.0771   |
| Correctnessincor | 0.9218    | 0.2177  | 4.235    | 2.29e-05 *** |

Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Time):Correctnesscor</td>
<td>1.080</td>
<td>1.156</td>
<td>0.667</td>
</tr>
<tr>
<td>s(Time):Correctnessincor</td>
<td>3.088</td>
<td>4.016</td>
<td>1.87e-05 ***</td>
</tr>
<tr>
<td>s(Time,Subject)</td>
<td>57.862</td>
<td>602.000</td>
<td>0.736</td>
</tr>
</tbody>
</table>

R-sq.(adj) = 0.0186   Deviance explained = 1.87%
ML score = 1.4923e+06  Scale est. = 291.56   n = 442160
Autocorrelation has been removed

> acf.new(m2, m3, "Time") # custom plotting function

Original acf

New acf with rho: 0.91
> compareML(m2,m3)

m2: mV ~ s(Time, by = Correctness) + Correctness +
    s(Time, Subject, bs = "fs", m = 1)
m3: mV ~ s(Time, by = Correctness) + Correctness +
    s(Time, Subject, bs = "fs", m = 1)

Chi-square test of ML scores
-----
  Model  Score  Edf  Chisq    Df  p.value  Sig.
1  m2   1911925 8       -
2  m3   1492310 8 419615.303 0.000 < 2e-16  ***

Warning message:
In compareML(m2, m3):
  AIC is not reliable, because an AR1 model is included
  (rho1 = 0.000000, rho2 = 0.912313).

> AIC(m2) - AIC(m3) # AIC indeed does not work when using rho
  [1] -701.3066
Here: random wiggly curves ≈ random intercepts
(computation could be made faster by using `s(Subject, bs="re")` instead)

```r
> plot(m3, select=3)
```
Resulting smooths: much larger confidence intervals

```r
> par(mfrow=c(1,2))
> plotSmooths(m3, "Time", "Correctness", colors=c('green','red'),
              dropRanef="Subject")
> plotDiff(m3, "Time", "Correctness")
```

Model: m3

Difference between cor and incor
Our research question: the effect of age of arrival

# in this lecture, rho is always set to the acf at lag 1 for the
# same model without rho
> m4 = bam(mV ~ te(Time,AoArr,by=Correctness) + Correctness +
  s(Time,Subject,bs='fs',m=1), data=dat,
  gc.level=2, method='ML', rho=rho, AR.start=SeqStart)

> summary(m4)

...  
Parametric coefficients:

                     Estimate Std. Error   t value  Pr(>|t|)  
(Intercept)       -0.54760   0.31733  -1.72600  0.08443  .
Correctnessincor  0.90383   0.21769   4.15110  3.31e-05 ***

Approximate significance of smooth terms:

                      edf Ref.df    F  p-value     
te(Time,AoArr):Correctnesscor 3.423  3.809 1.585 0.177  
te(Time,AoArr):Correctnessincor 5.975  7.145 4.679 2.62e-05 ***
s(Time,Subject)    56.540 601.000 0.738 < 2e-16 ***

R-sq.(adj) =  0.02  Deviance explained = 2.02%
ML score = 1.4923e+06  Scale est. = 291.11   n = 442160
Visualization of the 2-dimensional smooths

# Custom plotting functions are used to plot the partial effects
# (i.e. the general pattern) and difference surfaces.
# N.B. a surface including all effects (at pre-specified levels) can be
# visualized using vis.gam(...) - see additional slides at the end.

> par(mfrow=c(1,3))
> pvis.gam(m4, plot.type='contour', view=c('Time','AoArr'), select=1,
  color='topo', main='m4: cor (not significant)')
> pvis.gam(m4, plot.type='contour', view=c('Time','AoArr'), select=2,
  color='topo', main='m4: incor')
> plotDiff2D(m4, "Time", "AoArr", "Correctness", plotCI=T)
Interpreting 2-dimensional smooths

Difference between cor and incor

Difference between cor and incor

Difference between cor and incor

Difference between cor and incor (AoArr: 10)

Difference between cor and incor (AoArr: 20)

Difference between cor and incor (AoArr: 30)
Significance testing using a binary variable
(an ordered factor may be used to separate the intercept difference from the difference surface)

```r
> m4b = bam(mV ~ te(Time,AoArr) + te(Time,AoArr,by=IsIncorrect) +
  s(Time,Subject,bs='fs',m=1), data=dat,
  gc.level=2, method='ML', rho=rho, AR.start=SeqStart)

... Parametric coefficients:
    Estimate Std. Error t value Pr(>|t|)
  (Intercept)   -0.5224   0.3216  -1.624   0.104

Approximate significance of smooth terms:
                                edf Ref.df   F  p-value
  te(Time,AoArr)                2.109  2.831 1.032 0.370
  s(Time,Subject)              57.598 602.000 0.749 < 2e-16 ***

R-sq.(adj) =  0.02  Deviance explained = 2.01%
ML score = 1.4923e+06  Scale est. = 291.11    n = 442160
```
Visualizing the difference surface
(note that m4b is a different model than m4, so the difference surfaces do not need to be identical)

> pvis.gam(m4b, plot.type='contour', view=c('Time','AoArr'),
  contour.col='black', select=2, color='topo',
  main='m4b: difference surface')
Assessing if $k$ needs to be increased
(note that $k = 10$ has the same effect as $k = c(10,10)$)

```r
> m4c = bam(mV ~ te(Time,AoArr,by=Correctness,k=c(10,10)) + Correctness +
  s(Time,Subject,bs='fs',m=1), data=dat,
  gc.level=2, method='ML', rho=rho, AR.start=SeqStart)
> compareML(m4,m4c) # the additional complexity is not needed
...
Chi-square test of ML scores
-----
  Model | Score | Edf | Chisq | Df | p.value | Sig.
1 | m4c | 1492296 | 14 | 0.856 | 0.000 | < 2e-16 ***
2 | m4 | 1492295 | 14 | 0.856 | 0.000 | < 2e-16 ***
...
> summary(m4c) # max edf: 99 ($10^2 - 1$)
...
edf | Ref.df | F | p-value
te(Time,AoArr):Correctnesscor | 4.450 | 5.678 | 1.194 | 0.305
te(Time,AoArr):Correctnessincor | 6.576 | 8.394 | 3.948 | 8.58e-05 ***
...
Adding random wiggly curves per word

```r
> require(parallel)
> cl <- makeCluster(4) # use 4 processors for computation
> summary(m5 <- bam(mV ~ te(Time,AoArr,by=Correctness) +
                   Correctness + s(Time,Subject,bs='fs',m=1) +
                   s(Time,Word,bs='fs',m=1), data=dat, gc.level=2,
                   method='ML', rho=rho, AR.start=SeqStart, cluster=cl))
```

Parametric coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | -0.5760    | 0.3581  | -1.609   | 0.108    |
| Correctness | 0.8934     | 0.2185  | 4.088    | 4.35e-05 *** |

Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>te(Time,AoArr):Correctness</td>
<td>3.935</td>
<td>4.719</td>
<td>1.377</td>
</tr>
<tr>
<td>te(Time,AoArr):Correctness</td>
<td>6.012</td>
<td>7.206</td>
<td>4.668</td>
</tr>
<tr>
<td>s(Time,Subject)</td>
<td>56.955</td>
<td>601.000</td>
<td>0.741</td>
</tr>
<tr>
<td>s(Time,Word)</td>
<td>65.427</td>
<td>855.000</td>
<td>0.261</td>
</tr>
</tbody>
</table>

R-sq.(adj) = 0.0295  Deviance explained = 2.98%
ML score = 1.4922e+06  Scale est. = 290.96  n = 442160
Model comp. not essential for random wiggly curves

- Even though model comparison may suggest that random wiggly curves do not improve the model, it is useful to keep them in (if they are significant in the summary) to control for item and subject effects

```r
> compareML(m4r, m5r) # here, however the model does improve

# m4r and m5r are the same as m4 and m5, but with method='REML'
# (required when comparing random effects)
m4r: mV ~ te(Time, AoArr, by = Correctness) + Correctness +
    s(Time, Subject, bs = "fs", m = 1)
m5r: mV ~ te(Time, AoArr, by = Correctness) + Correctness +
    s(Time, Subject, bs = "fs", m = 1) +
    s(Time, Word, bs = "fs", m = 1)

Chi-square test of REML scores
-----
           Model Score Edf Chisq  Df p.value Sig.
1       m4r 1492290 14             
2       m5r 1492235 16  55.014 2.000 < 2e-16 ***
```
Decomposition: the pure effect of age of arrival
(note that the 2-dimensional interactions are not significant)

```r
> summary(m5b <- bam(mV ~ s(Time,by=Correctness) + s(AoArr,by=Correctness) +
        ti(Time,AoArr,by=Correctness) + Correctness +
        s(Time,Subject,bs='fs',m=1) +
        s(Time,Word,bs='fs',m=1), data=dat, gc.level=2,
        method='ML', rho=rho, AR.start=SeqStart, cluster=cl))

... Estimate  Std. Error   t value   Pr(>|t|)
(Intercept)  -0.5794   0.3571  -1.623 0.105
Correctnessincor  0.8970   0.2180   4.115 3.87e-05 ***

Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th>Term</th>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Time):Correctnesscor</td>
<td>1.192</td>
<td>1.361</td>
<td>0.380</td>
<td>0.6025</td>
</tr>
<tr>
<td>s(Time):Correctnessincor</td>
<td>3.088</td>
<td>4.015</td>
<td>6.769</td>
<td>1.88e-05 ***</td>
</tr>
<tr>
<td>s(AoArr):Correctnesscor</td>
<td>2.111</td>
<td>2.592</td>
<td>1.420</td>
<td>0.2327</td>
</tr>
<tr>
<td>s(AoArr):Correctnessincor</td>
<td>1.011</td>
<td>1.017</td>
<td>4.976</td>
<td>0.0252 *</td>
</tr>
<tr>
<td>ti(Time,AoArr):Correctnesscor</td>
<td>1.893</td>
<td>2.633</td>
<td>1.060</td>
<td>0.3516</td>
</tr>
<tr>
<td>ti(Time,AoArr):Correctnessincor</td>
<td>2.198</td>
<td>3.133</td>
<td>0.438</td>
<td>0.7347</td>
</tr>
<tr>
<td>s(Time,Subject)</td>
<td>56.515</td>
<td>601.000</td>
<td>0.722</td>
<td>&lt; 2e-16 ***</td>
</tr>
<tr>
<td>s(Time,Word)</td>
<td>65.570</td>
<td>855.000</td>
<td>0.261</td>
<td>&lt; 2e-16 ***</td>
</tr>
</tbody>
</table>
```
A simpler model

```r
> m5c = bam(mV ~ s(Time,by=Correctness) + s(AoArr,by=Correctness) +
  Correctness + s(Time,Subject,bs='fs',m=1) +
  s(Time,Word,bs='fs',m=1), data=dat, gc.level=2,
  method='ML', rho=rho, AR.start=SeqStart, cluster=cl)

> summary(m5c)

Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)  -0.5829     0.3562   -1.636     0.102
Correctnessincor  0.9024     0.2175    4.149  3.34e-05 ***

Approximate significance of smooth terms:

<table>
<thead>
<tr>
<th>Term</th>
<th>edf</th>
<th>Ref.df</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>s(Time):Correctnesscor</td>
<td>1.027</td>
<td>1.054</td>
<td>0.344</td>
<td>0.5693</td>
</tr>
<tr>
<td>s(Time):Correctnessincor</td>
<td>3.091</td>
<td>4.019</td>
<td>6.792</td>
<td>1.79e-05 ***</td>
</tr>
<tr>
<td>s(AoArr):Correctnesscor</td>
<td>3.404</td>
<td>4.151</td>
<td>1.159</td>
<td>0.3255</td>
</tr>
<tr>
<td>s(AoArr):Correctnessincor</td>
<td>1.003</td>
<td>1.004</td>
<td>4.859</td>
<td>0.0274  *</td>
</tr>
<tr>
<td>s(Time,Subject)</td>
<td>56.249</td>
<td>601.000</td>
<td>0.709</td>
<td>&lt; 2e-16 ***</td>
</tr>
<tr>
<td>s(Time,Word)</td>
<td>65.453</td>
<td>855.000</td>
<td>0.261</td>
<td>&lt; 2e-16 ***</td>
</tr>
</tbody>
</table>

R-sq.(adj) =  0.0296  Deviance explained = 2.99%
ML score = 1.4922e+06  Scale est. = 290.96  n = 442160
```
Model comparison: the simpler model is sufficient

\> \texttt{compareML(m5, m5c)}

\begin{verbatim}
m5: mV ~ te(Time, AoArr, by = Correctness) + Correctness + s(Time, Subject, bs = "fs", m = 1) + s(Time, Word, bs = "fs", m = 1)
m5c: mV ~ s(Time, by = Correctness) + s(AoArr, by = Correctness) + Correctness + s(Time, Subject, bs = "fs", m = 1) + s(Time, Word, bs = "fs", m = 1)
\end{verbatim}

Chi-square test of ML scores
-----

<table>
<thead>
<tr>
<th>Model</th>
<th>Score</th>
<th>Edf</th>
<th>Chisq</th>
<th>Df</th>
<th>p.value</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>m5c</td>
<td>1492237</td>
<td>14</td>
<td>2.564</td>
<td>2.000</td>
<td>0.077</td>
</tr>
<tr>
<td>2</td>
<td>m5</td>
<td>1492234</td>
<td>16</td>
<td>0.000</td>
<td>0.077</td>
<td></td>
</tr>
</tbody>
</table>

Warning messages:
1: \texttt{In compareML(m5, m5c)}: AIC is not reliable, because an AR1 model is included
2: \texttt{In compareML(m5, m5c)}: Only small difference in ML...
Testing for significant smooth differences
(a binary variable may only occur once in a model, so ordered factors are essential here)

```r
> m6 <- bam(mV ~ s(Time) + s(Time,by=CorrectnessO) + s(AoArr) + s(AoArr,by=CorrectnessO) + CorrectnessO + ... )
> summary(m6)
...
Parametric coefficients:

        Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.8574    0.3532  -2.428  0.0152 *
CorrectnessOincor    0.8900    0.2182   4.079 4.52e-05 ***

Approximate significance of smooth terms:

        edf Ref.df  F p-value
s(Time) 1.324  1.568 0.596 0.496843
s(Time):CorrectnessOincor 2.945  3.823 5.231 0.000465 ***
s(AoArr) 1.049  1.057 0.201 0.667579
s(AoArr):CorrectnessOincor 1.011  1.021 33.584 5.72e-09 ***
s(Time,Subject) 56.591 600.000 0.708 < 2e-16 ***
s(Time,Word)   65.434  855.000 0.258 < 2e-16 ***
...
```

▶ Thus: the data of Nienke clearly shows a highly significant decreasing effect of age of arrival on the P600
Difference curves

```r
> par(mfrow=c(1,2))
> plot(m6, shade=T, rug=F, select=2, ylim=c(1.2,-2.5),
     main='Effect of Incorrectness', ylab='mV', seWithMean=T)
> abline(h = 0)
> plot(m6, shade=T, rug=F, select=4, ylim=c(3,-4.5),
     main='Effect of Age of Arrival', ylab='mV', seWithMean=T)
> abline(h = 0)
```
Finally: model criticism

```r
> library(car)
> par(mfrow=c(1,2))
> qqplot.rho(m6) # custom qqplot of residuals taking rho into account
> hist.rho(m6) # custom histogram of residuals taking rho into account
```

- Common pattern for EEG data, hard to correct (work in progress...)
- For now, use a more conservative $p$-value such as 0.001
Discussion

- Still much to do: e.g., testing the significance of other possibly important variables (proficiency, age, etc.)
- But don’t make it too complex!
  - There is much variation present in EEG data and adding very complex surfaces will almost certainly improve your model significantly
  - K.I.S.S., otherwise you won’t be able to interpret the results
Conclusion

- GAMs are very useful to analyze EEG and other time-series data
  - The method is very suitable to detect non-linear patterns, while taking into account individual variation and correcting for autocorrelation

- If you are interested in GAMs and mixed-effects regression and would like to learn more about this in a hands-on setting, you are welcome to participate in my 5-day crash course (June 23 - June 27):
  [http://goo.gl/tZRTr3](http://goo.gl/tZRTr3)
Thank you for your attention!

Any questions?
Visualization: vis.gam vs. pvis.gam (1)

```r
> m4b = bam(mV ~ te(Time,AoArr) + te(Time,AoArr,by=IsIncorrect) +
          s(Time,Subject,bs='fs',m=1), data=dat,
          gc.level=2, method='ML', rho=rho, AR.start=SeqStart)

> par(mfrow=c(2,2))
> vis.gam(m4b, plot.type='contour', view=c('Time','AoArr'),
          contour.col='black', cond=list(IsIncorrect=0,Subject='GL107'),
          color='topo', main='vis.gam - m4b: cor (Subject: GL107)')
> vis.gam(m4b, plot.type='contour', view=c('Time','AoArr'),
          contour.col='black', cond=list(IsIncorrect=1,Subject='GL107'),
          color='topo', main='vis.gam - m4b: incor (Subject: GL107)')
> pvis.gam(m4b, plot.type='contour', view=c('Time','AoArr'),
          contour.col='black', select=1, color='topo',
          main='pvis.gam - m4b: cor (independent of subject)')
> pvis.gam(m4b, plot.type='contour', view=c('Time','AoArr'),
          contour.col='black', select=2, color='topo',
          main='pvis.gam - m4b: difference surface')
```
Visualization: vis.gam vs. pvis.gam (2)
(vis.gam shows the resulting surface of the conditions, pvis.gam shows the smooths in your model)
Differences between ML, REML and GCV

- **ML** is conservative, but the variance component (i.e. smooth) is biased (oversmoothed)
- **(f)REML** is less conservative, but robust to moderate autocorrelation, and the smooth is not biased
- **GCV . Cp** is better for prediction, but not robust to autocorrelation
- Strongest approach is to compare results across different methods (i.e. using AIC)