amer: Some application examples

Fabian Scheipl*
LMU München

April 6, 2011

Abstract

The following gives some examples of additive mixed models and compares them to linear mixed models on well-known datasets, hopefully demonstrating the utility of penalized spline smoothing for this type of problems.

*fabian.scheipl@stat.uni-muenchen.de

1 `Contraception` data: A Generalized Additive Mixed Model

Conventional GLMM fits, which assume that age has a linear influence on the log-odds of contraceptive use:

Random intercept model:

```
> print(contra1 <- lmer(use ~ urban + age + livch +
+     (1 | district), Contraception, family = binomial),
+     cor = F)

Generalized linear mixed model fit by the Laplace approximation
Formula: use ~ urban + age + livch + (1 | district)
   Data: Contraception
  AIC  BIC logLik deviance
 2428 2467  -1207     2414
Random effects:
 Groups   Name        Variance Std.Dev.
 district (Intercept) 0.212    0.461
Number of obs: 1934, groups: district, 60

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.68971    0.14550  -11.61  < 2e-16 ***
urbanY       0.73299    0.11842    6.19  6.0e-10 ***
age         -0.02660    0.00783   -3.40  0.00068 ***
livch1       1.10918    0.15682    7.07  1.5e-12 ***
livch2       1.37640    0.17331    7.94  2.0e-15 ***
livch3+      1.34518    0.17777    7.57  3.8e-14 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Random slope model:

```
> print(contra2 <- lmer(use ~ urban + age + livch +
+     (urban | district), Contraception, family = binomial),
+     cor = F)

Generalized linear mixed model fit by the Laplace approximation
Formula: use ~ urban + age + livch + (urban | district)
```

```
   Data: Contraception
  AIC  BIC logLik deviance
 2417 2467  -1200     2399
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 district (Intercept) 0.381    0.617
          urbanY      0.642    0.801    -0.798
Number of obs: 1934, groups: district, 60

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.71185    0.15743  -10.87  < 2e-16 ***
urbanY       0.81529    0.16641    4.90  9.6e-07 ***
age         -0.02652    0.00792   -3.35  0.00081 ***
livch1       1.12570    0.15838    7.11  1.2e-12 ***
livch2       1.36833    0.17501    7.82  5.3e-15 ***
livch3+      1.35473    0.18007    7.52  5.3e-14 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Let's try a nonlinear effect for age:

```
> print(contra3 <- amer(use ~ urban + bsp(age) +
+     livch + (urban | district), Contraception,
+     family = binomial), cor = F)
```

```
Generalized additive mixed model fit by the Laplace approximation
Formula: use ~ urban + livch + (urban | district) + bsp(x = age, k = 15,      spline
   Data: Contraception
  AIC  BIC logLik deviance
 2389 2445  -1184     2369
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 district (Intercept) 0.3845   0.620
          urbanY      0.5489   0.741    -0.792
 f.age    bsp         0.0154   0.124
Number of obs: 1934, groups: district, 60; f.age, 13

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.79241    0.18446   -9.72  < 2e-16 ***
urbanY       0.77600    0.16120    4.81  1.5e-06 ***
age.fx1     -0.02550    0.00971   -2.63   0.0087 **
livch1       0.86917    0.16554    5.25  1.5e-07 ***
```

```
livch2       0.95726    0.18870    5.07  3.9e-07 ***
livch3+      0.95983    0.18785    5.11  3.2e-07 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

The estimated variance for the spline coefficients indicates some nonlinearity.

Finally, let's allow the effect of age to be different for urban and rural areas and compare the 4 models:

```
> print(contra4 <- amer(use ~ urban + bsp(age, by = urban) +
+     livch + (urban | district), Contraception,
+     family = binomial), cor = F)

Generalized additive mixed model fit by the Laplace approximation
Formula: use ~ urban + livch + (urban | district) + bsp(x = age, by = urban,      k
   Data: Contraception
  AIC  BIC logLik deviance
 2395 2462  -1186     2371
Random effects:
 Groups       Name        Variance Std.Dev. Corr
 district     (Intercept) 0.3856   0.621
              urbanY      0.5441   0.738    -0.793
 f.age.urbanY bsp         0.0191   0.138
 f.age.urbanN bsp         0.0162   0.127
Number of obs: 1934, groups: district, 60; f.age.urbanY, 13; f.age.urbanN, 13

Fixed effects:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.7801     0.1867   -9.53  < 2e-16 ***
urbanY           0.7117     0.2054    3.46  0.00053 ***
age.urbanN.fx1  -0.0229     0.0108   -2.13  0.03313 *
age.urbanY.fx1  -0.0317     0.0143   -2.21  0.02725 *
livch1           0.8930     0.1662    5.37  7.7e-08 ***
livch2           0.9893     0.1893    5.23  1.7e-07 ***
livch3+          0.9874     0.1889    5.23  1.7e-07 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

> print(anova(contra1, contra2, contra3, contra4))

Data: Contraception
Models:
```

```
contra1: use ~ urban + age + livch + (1 | district)
contra2: use ~ urban + age + livch + (urban | district)
contra3: use ~ urban + livch + (urban | district) + bsp(x = age, k = 15,
contra3:     spline.degree = 3, diff.ord = 2, knots = c(-22.305, -19.5,
contra3:         -16.695, -13.89, -11.085, -8.28, -5.475, -2.67, 0.134999999999998,
contra3:         2.94, 5.745, 8.55, 11.355, 14.16, 16.965, 19.77, 22.575,
contra3:         25.38, 28.185), by = NULL, allPen = FALSE, varying = NULL,
contra3:     diag = FALSE)
contra4: use ~ urban + livch + (urban | district) + bsp(x = age, by = urban,
contra4:     k = 15, spline.degree = 3, diff.ord = 2, knots = c(-22.305,
contra4:         -19.5, -16.695, -13.89, -11.085, -8.28, -5.475, -2.67,
contra4:         0.134999999999998, 2.94, 5.745, 8.55, 11.355, 14.16,
contra4:         16.965, 19.77, 22.575, 25.38, 28.185), allPen = FALSE,
contra4:     varying = NULL, diag = FALSE)
        Df  AIC  BIC logLik Chisq Chi Df Pr(>Chisq)
contra1  7 2428 2467  -1207
contra2  9 2417 2467  -1200  14.6      2    0.00068 ***
contra3 10 2389 2445  -1184  30.1      1      4e-08 ***
contra4 12 2395 2462  -1186   0.0      2    1.00000
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Note the large improvement for model `contra3` when we allow a nonlinear influence of age.

Let's look at the estimated functions:

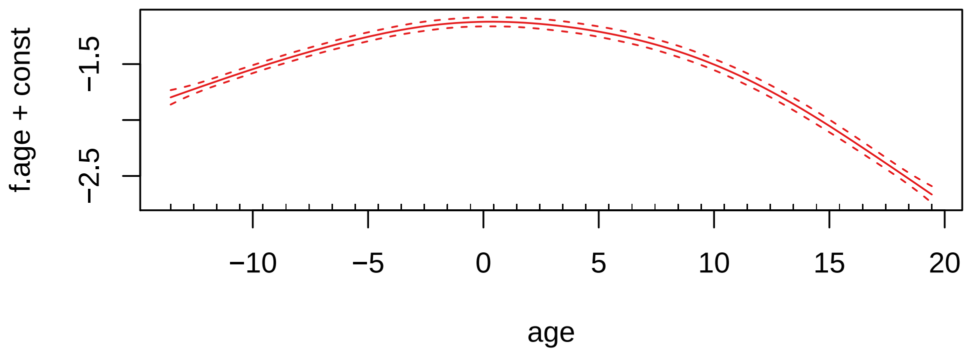

Figure 1: Estimated influence of age on contraception use from `contra3`.

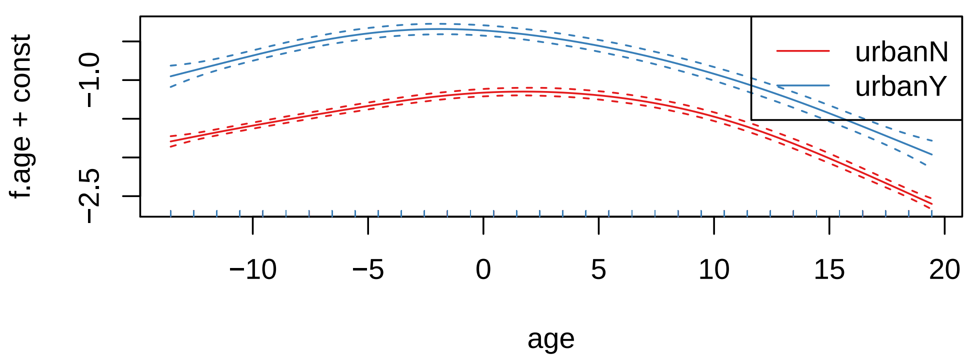

Figure 2: Estimated influence of age on contraception use by rural vs. urban from `contra4`. The difference seems to be captured mostly by the dummy for `urbanY`, the shape of the effect is about the same.

2 `Chem97` data: An AMM for large data

```
> print(chem1 <- lmer(score ~ gcsecnt + (1 | school) +
+     (1 | lea), Chem97), cor = F)

Linear mixed model fit by REML
Formula: score ~ gcsecnt + (1 | school) + (1 | lea)
   Data: Chem97
    AIC    BIC logLik deviance REMLdev
 141707 141749 -70848   141686  141697
Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept) 1.1662   1.080
 lea      (Intercept) 0.0148   0.122
 Residual             5.1542   2.270
Number of obs: 31022, groups: school, 2410; lea, 131

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5.6354     0.0312     180
gcsecnt       2.4726     0.0169     146
```

Maybe there's no *linear* relationship between GCSE score and Chemistry A-levels? We can use a spline to find out:

```
> print(chem2 <- amer(score ~ bsp(gcsecnt) + (1 |
+     school) + (1 | lea), Chem97), cor = F)

Additive mixed model fit by REML
Formula: score ~ (1 | school) + (1 | lea) + bsp(x = gcsecnt, k = 15, spline.degree =
   Data: Chem97
    AIC    BIC logLik deviance REMLdev
 140488 140538 -70238   140472  140476
Random effects:
 Groups    Name        Variance Std.Dev.
 school    (Intercept) 1.1670   1.080
 lea       (Intercept) 0.0148   0.122
 f.gcsecnt bsp         0.6230   0.789
 Residual              4.9412   2.223
Number of obs: 31022, groups: school, 2410; lea, 131; f.gcsecnt, 13

Fixed effects:
            Estimate Std. Error t value
```

```
(Intercept)    6.736      0.330   20.39
gcsecnt.fx1    0.581      0.195    2.98

> print(anova(chem1, chem2))

Data: Chem97
Models:
chem1: score ~ gcsecnt + (1 | school) + (1 | lea)
chem2: score ~ (1 | school) + (1 | lea) + bsp(x = gcsecnt, k = 15, spline.degree = 3
chem2:     diff.ord = 2, knots = c(-8.40568428856967, -7.72568428856967,
chem2:         -7.04568428856967, -6.36568428856967, -5.68568428856967,
chem2:         -5.00568428856967, -4.32568428856967, -3.64568428856967,
chem2:         -2.96568428856967, -2.28568428856967, -1.60568428856967,
chem2:         -0.92568428856967, -0.24568428856967, 0.43431571143033,
chem2:         1.11431571143033, 1.79431571143033, 2.47431571143033,
chem2:         3.15431571143033, 3.83431571143033), by = NULL, allPen = FALSE,
chem2:     varying = NULL, diag = FALSE)
      Df    AIC    BIC logLik Chisq Chi Df Pr(>Chisq)
chem1  5 141696 141737 -70843
chem2  6 140484 140534 -70236  1213      1     <2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

The improvement in the fit is pretty big!

What does the relationship between GCSE score and Chemistry A-levels look like?

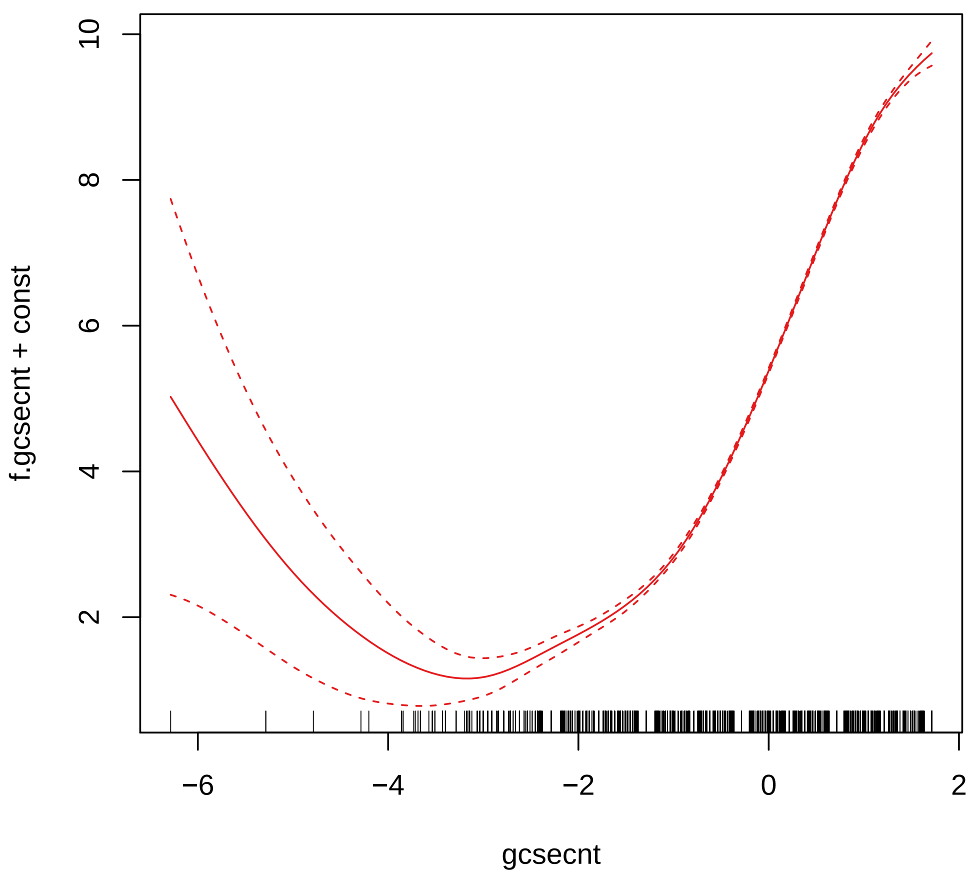

Figure 3: That large rise on the lower end of the GCSE scale is weird and shouldn't be interpreted (consider the width of the pointwise CI's!), but what does make a lot of sense is the saturation effect we see: The slope flattens for below average GCSEs and also, a little, for very high GCSEs.

3 `Oxboys`: An AMM with subject-wise smooth trends

The LMM framework struggles with growth data like this: We have to include fairly arbitrary polynomial terms for both the global trend and the subject-wise trends to fit the data well:

```
> print(oxboys1 <- lmer(height ~ poly(age, 4) +
+     (poly(age, 2) | Subject), data = Oxboys),
+     cor = F)

Linear mixed model fit by REML
Formula: height ~ poly(age, 4) + (poly(age, 2) | Subject)
   Data: Oxboys
 AIC BIC logLik deviance REMLdev
 641 682   -308      625     617
Random effects:
 Groups   Name          Variance Std.Dev. Corr
 Subject  (Intercept)    65.732   8.108
          poly(age, 2)1 282.290  16.801   0.638
          poly(age, 2)2  21.590   4.646   0.258 0.661
 Residual                 0.217   0.466
Number of obs: 234, groups: Subject, 26

Fixed effects:
              Estimate Std. Error t value
(Intercept)    149.520      1.590    94.0
poly(age, 4)1   64.541      3.328    19.4
poly(age, 4)2    4.203      1.024     4.1
poly(age, 4)3    1.291      0.466     2.8
poly(age, 4)4   -0.585      0.466    -1.3
```

In an AMM, we simply include a global smooth term for `age` and subject-wise smooth deviations from it:

```
> print(oxboys2 <- amer(height ~ tp(age, k = 12) +
+     tp(age, k = 4, by = Subject, allPen = T),
+     data = Oxboys), cor = F)

Additive mixed model fit by REML
Formula: height ~ 1 + tp(x = age, k = 12, degree = 1L, by = NULL, allPen = FALSE,
```

```
   Data: Oxboys
 AIC BIC logLik deviance REMLdev
 638 665   -311      625     622
Random effects:
 Groups        Name             Variance Std.Dev. Corr
 f.age.Subject tp                0.952   0.976
 u.age.Subject (Intercept)      62.576   7.911
               age.Subject.fx1  0.412   0.642    0.739
 f.age         tp                0.352   0.593
 Residual                        0.176   0.419
Number of obs: 234, groups: f.age.Subject, 78; u.age.Subject, 26; f.age, 11

Fixed effects:
            Estimate Std. Error t value
(Intercept)  149.483      1.844    81.1
age.fx1        4.007      0.655     6.1
```

```
> print(anova(oxboys1, oxboys2))
```

```
Data: Oxboys
Models:
oxboys2: height ~ 1 + tp(x = age, k = 12, degree = 1L, by = NULL, allPen = FALSE,
oxboys2:     varying = NULL, diag = FALSE, knots = c(-1.57824083172682,
oxboys2:         -1.18917272500261, -0.749484095071128, -0.435112126250342,
oxboys2:         -0.288497805985365, -0.0390991306714729, 0.345647711876505,
oxboys2:         0.504917162943288, 0.823456065076881, 1.16591625104317,
oxboys2:         1.49988823952044), centerscale = c(0.0226346153846154,
oxboys2:         0.647958533847909), scaledknots = TRUE) + tp(x = age,
oxboys2:     k = 4, by = Subject, allPen = T, degree = 1L, varying = NULL,
oxboys2:     diag = FALSE, knots = c(-0.749484095071128, -0.0390991306714729,
oxboys2:         0.823456065076881), centerscale = c(0.0226346153846154,
oxboys2:         0.647958533847909), scaledknots = TRUE)
oxboys1: height ~ poly(age, 4) + (poly(age, 2) | Subject)
        Df AIC BIC logLik Chisq Chi Df Pr(>Chisq)
oxboys2  8 641 669   -313
oxboys1 12 649 691   -313     0      4          1
```

This yields a slightly better fit with a more parsimonious model (Well, depending on how you count the degrees of freedom. Let's agree to not go there...).

4 ScotsSec: An AMM with a nice interpretation

```
> ScotsSec$social <- factor(ScotsSec$social)
> print(scots1 <- lmer(attain ~ sex + (1 | primary) +
+     (1 | second), ScotsSec), cor = F)

Linear mixed model fit by REML
Formula: attain ~ sex + (1 | primary) + (1 | second)
   Data: ScotsSec
   AIC   BIC logLik deviance REMLdev
 17138 17169  -8564    17123   17128
Random effects:
 Groups   Name        Variance Std.Dev.
 primary  (Intercept) 1.11     1.053
 second   (Intercept) 0.37     0.608
 Residual             8.06     2.838
Number of obs: 3435, groups: primary, 148; second, 19

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5.2552     0.1843   28.51
sexF          0.4985     0.0983    5.07

> print(scots2 <- lmer(attain ~ sex + verbal + (1 |
+     primary) + (1 | second), ScotsSec), cor = F)

Linear mixed model fit by REML
Formula: attain ~ sex + verbal + (1 | primary) + (1 | second)
   Data: ScotsSec
   AIC   BIC logLik deviance REMLdev
 14872 14909  -7430    14843   14860
Random effects:
 Groups   Name        Variance Std.Dev.
 primary  (Intercept) 0.2763   0.526
 second   (Intercept) 0.0145   0.120
 Residual             4.2519   2.062
Number of obs: 3435, groups: primary, 148; second, 19

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.91927    0.07615    77.7
sexF         0.11597    0.07146     1.6
verbal       0.15959    0.00278    57.5
```

```
> print(scots3 <- lmer(attain ~ sex + social + verbal +
+     (1 | primary) + (1 | second), ScotsSec), cor = F)

Linear mixed model fit by REML
Formula: attain ~ sex + social + verbal + (1 | primary) + (1 | second)
   Data: ScotsSec
   AIC   BIC logLik deviance REMLdev
 14710 14765  -7346    14667   14692
Random effects:
 Groups   Name        Variance Std.Dev.
 primary  (Intercept) 1.41e-01 3.76e-01
 second   (Intercept) 5.82e-13 7.63e-07
 Residual             4.10e+00 2.02e+00
Number of obs: 3435, groups: primary, 148; second, 19

Fixed effects:
            Estimate Std. Error t value
(Intercept)  5.56128    0.06809    81.7
sexF         0.13786    0.07005     2.0
social1      1.33977    0.16241     8.2
social20     1.12658    0.09147    12.3
social31     0.50970    0.12825     4.0
verbal       0.15194    0.00279    54.5
```

Ok, so the `verbal` score has huge predictive value for this standardized test – is its effect really linear, though?

```
> print(scots4 <- amer(attain ~ sex + social + bsp(verbal) +
+     (1 | primary) + (1 | second), ScotsSec), cor = F)

Additive mixed model fit by REML
Formula: attain ~ sex + social + (1 | primary) + (1 | second) + bsp(x = verbal,
   Data: ScotsSec
   AIC   BIC logLik deviance REMLdev
 14575 14636  -7277    14533   14555
Random effects:
 Groups   Name        Variance Std.Dev.
 primary  (Intercept) 0.13792  0.3714
 second   (Intercept) 0.00288  0.0537
 f.verbal bsp         0.16550  0.4068
 Residual             3.91835  1.9795
Number of obs: 3435, groups: primary, 148; second, 19; f.verbal, 13
```

```
Fixed effects:
            Estimate Std. Error t value
(Intercept)  6.66360    0.18204    36.6
sexF         0.12389    0.06859     1.8
social1      1.35805    0.15960     8.5
social20     1.10605    0.08960    12.3
social31     0.54868    0.12554     4.4
verbal.fx1   0.10915    0.00751    14.5

> print(scots5 <- amer(attain ~ sex + social + bsp(verbal,
+     by = social) + (1 | primary) + (1 | second),
+     ScotsSec), cor = F)

Additive mixed model fit by REML
Formula: attain ~ sex + social + (1 | primary) + (1 | second) + bsp(x = verbal,
   Data: ScotsSec
   AIC   BIC logLik deviance REMLdev
 14609 14707  -7289    14542   14577
Random effects:
 Groups            Name        Variance Std.Dev.
 primary           (Intercept) 0.13435  0.3665
 second            (Intercept) 0.00197  0.0443
 f.verbal.social31 bsp         0.26447  0.5143
 f.verbal.social20 bsp         0.20667  0.4546
 f.verbal.social1  bsp         0.12594  0.3549
 f.verbal.social0  bsp         0.15033  0.3877
 Residual                      3.90559  1.9763
Number of obs: 3435, groups: primary, 148; second, 19; f.verbal.social31, 13; f.verb

Fixed effects:
                    Estimate Std. Error t value
(Intercept)           7.0495     0.2565   27.49
sexF                  0.1295     0.0686    1.89
social1               0.7075     0.3900    1.81
social20              0.5187     0.4107    1.26
social31              0.1093     0.4465    0.24
verbal.social0.fx1    0.1227     0.0104   11.76
verbal.social1.fx1    0.1153     0.0158    7.31
verbal.social20.fx1   0.1138     0.0147    7.73
verbal.social31.fx1   0.1239     0.0178    6.95

> print(anova(scots1, scots2, scots3, scots4, scots5))

Data: ScotsSec
Models:
```

```
scots1: attain ~ sex + (1 | primary) + (1 | second)
scots2: attain ~ sex + verbal + (1 | primary) + (1 | second)
scots3: attain ~ sex + social + verbal + (1 | primary) + (1 | second)
scots4: attain ~ sex + social + (1 | primary) + (1 | second) + bsp(x = verbal,
scots4:     k = 15, spline.degree = 3, diff.ord = 2, knots = c(-48.55,
scots4:         -42.6, -36.65, -30.7, -24.75, -18.8, -12.85, -6.9, -0.94999999999999
scots4:         5.00000000000001, 10.95, 16.9, 22.85, 28.8, 34.75, 40.7,
scots4:         46.65, 52.6, 58.55), by = NULL, allPen = FALSE, varying = NULL,
scots4:     diag = FALSE)
scots5: attain ~ sex + social + (1 | primary) + (1 | second) + bsp(x = verbal,
scots5:     by = social, k = 15, spline.degree = 3, diff.ord = 2, knots = c(-48.55,
scots5:         -42.6, -36.65, -30.7, -24.75, -18.8, -12.85, -6.9, -0.94999999999999
scots5:         5.00000000000001, 10.95, 16.9, 22.85, 28.8, 34.75, 40.7,
scots5:         46.65, 52.6, 58.55), allPen = FALSE, varying = NULL,
scots5:     diag = FALSE)
       Df   AIC   BIC logLik Chisq Chi Df Pr(>Chisq)
scots1  5 17133 17164  -8562
scots2  6 14855 14892  -7421  2280      1     <2e-16 ***
scots3  9 14685 14740  -7333   176      3     <2e-16 ***
scots4 10 14553 14615  -7267   134      1     <2e-16 ***
scots5 16 14574 14672  -7271     0      6          1
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Doesn't seem so, the AMM fits much better, since it's able to model the saturation effect of above-average `attain` scores, as shown in the following figure. The improvement of the fit by letting the effect of `attain` vary by `social` class (i.e. `scots5`) is small.

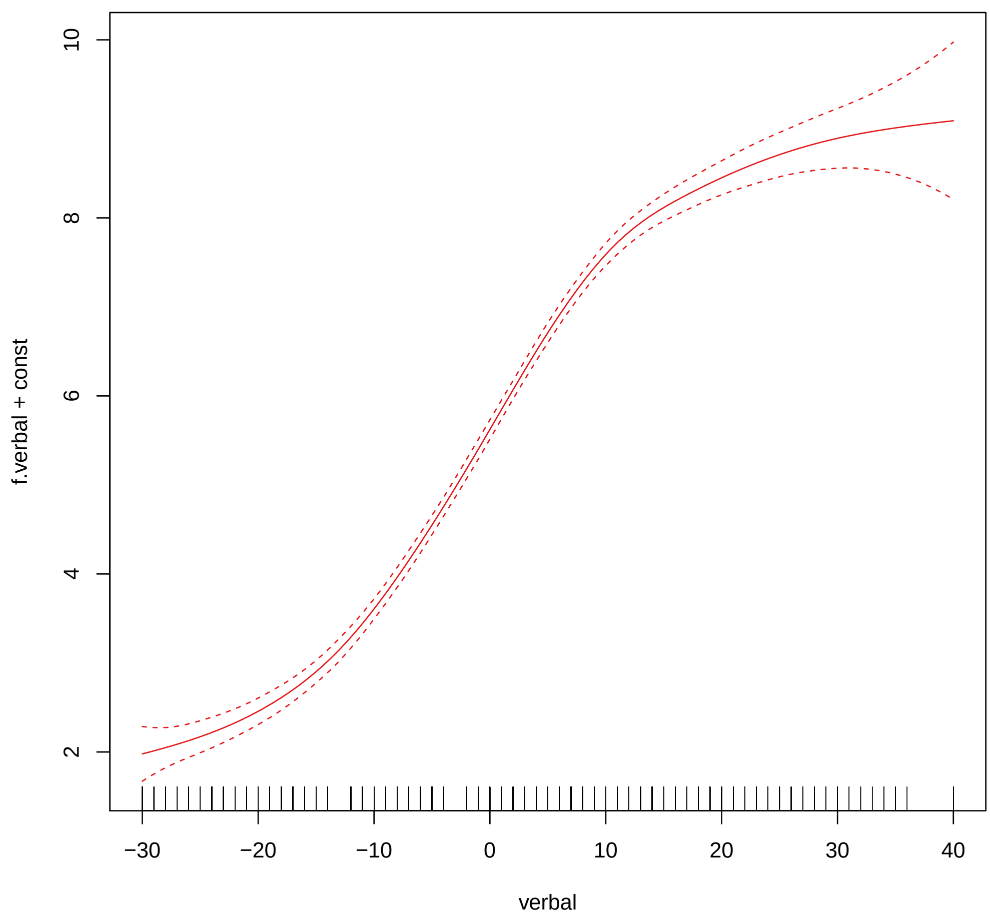

Figure 4: Effect of `verbal` on `attain` as estimated in model `scots4`: If you're really good verbally, it doesn't seem to make much of a difference whether you are in the top 5 % (above 20 points) or in the top 1 % (above 30) - your expected `attain` score will be about the same. Differences in the verbal test scores have a much larger impact for average students.