Analysis of Caries Experience (dmfs)

Supplementary material of the manuscript "Predicting caries experience: Advantages of the use of the hurdle model"

Introduction

This document contains the R-code that was used for the analyses described in the manuscript `Predicting caries experience: Advantages of the use of the hurdle model`. The dataset that was used is `OralhealthNL.txt`. It contains observations of 440 nine-year-old children in the Netherlands. The outcome variable of interest is `dmfs`, which is a measure of caries experience in the primary teeth. Furthermore, the dataset has three demographic risk factors, three life style risk factors (dichotomized at Dutch norms) and one psychological risk factor (also dichotomized using the clinical cut-off value). You may use this data set for educational purposes only. If you use the data set, refer to:

Hofstetter, H., Dusseldorp, E., Zeileis, A., and Schuller A.A.. Predicting caries experience: Advantages of the use of the hurdle model. *Manuscript submitted for publication.*

The analyses are performed in R, a software environment that is freely available at CRAN.

Preparation for the analysis in R

Data

Load the dataset in R, inspect the data, and omit outlier and NAs:

```
OralHealthNL <- read.table("OralHealthNL.txt", header = TRUE)
head(OralHealthNL)
```

```
##   dmfs education gender ethnicity brushing breakfast fooddrink corah
## 1    1       low   male    native     >= 2         7      <= 7  <NA>
## 2    0      high   male    native     >= 2         7      <= 7  <NA>
## 3    6      high female    native     >= 2         7      <= 7  < 13
## 4    9       low female    native     >= 2         7       > 7  < 13
## 5    0      high female    native      < 2         7      <= 7  <NA>
## 6    0       low   male    native     >= 2         7      <= 7  < 13
```

```
OralHealthNL <- na.omit(subset(OralHealthNL, dmfs < 40))
```

Packages in R

Various packages are needed for the actual count data regressions (`countreg`), the model comparisons (`lmtest` and `memisc`), the Vuong test (`nonnest2`), and a suggestion of illustrating a bias-reduced zero hurdle regression (`brglm`), respectively.

```
library("memisc")
library("countreg")
library("lmtest")
library("nonnest2")
library("brglm")
```

The `countreg` package contains slightly enhanced implemementations of `zeroinfl()` and `hurdle()` along with various other useful tools for count data regression (e.g., the model-based visualizations shown below). It is not on CRAN, yet, but it is planned to be released soon. The package can be easily installed via `install.packages("countreg", repos = "http://R-Forge.R-project.org")` Note that it is important to load `countreg` after `memisc` because it also improves some of the methods from that package.

The `nonnest2` package contains a more general implementation of the Vuong test that is also applicable to other model classes.

Start the analysis

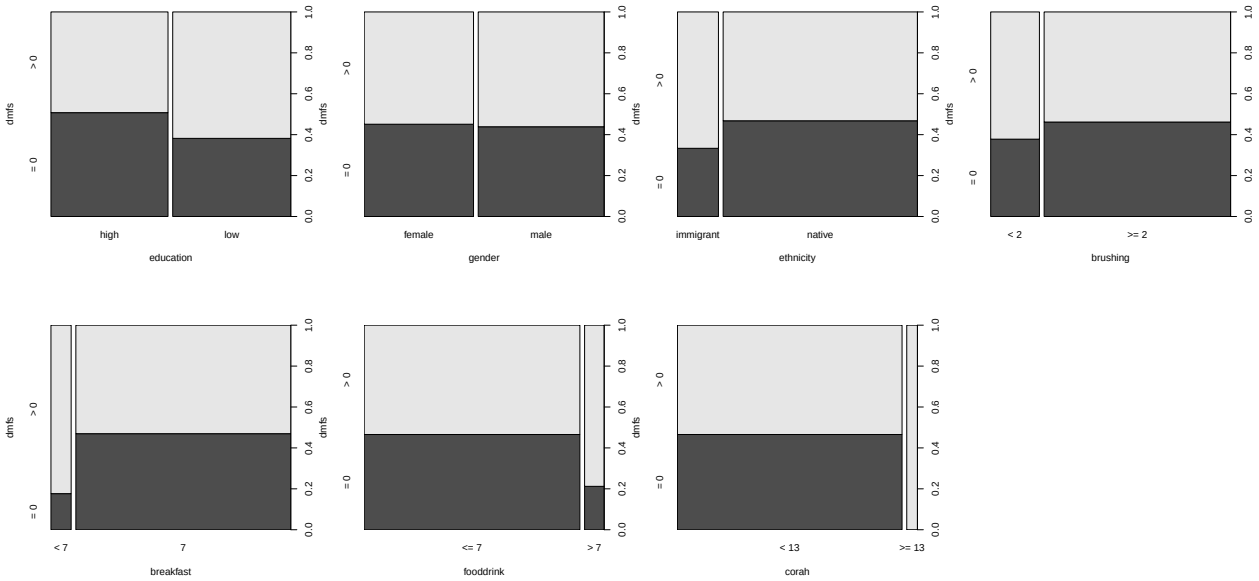
Exploratory visualization

For a first look at the data, we can employ marginal displays of the outcome given each regressor. Of course, this is just the marginal association ignoring the correlation with other regressors; it gives a first insight in the underlying relationships.

Is dmfs > 0?

Create a binary outcome factor for `dmfs` and plot that against each regressor.

```
par(mfrow = c(2, 4))
plot(factor(dmfs > 0, levels = c(TRUE, FALSE), labels = c("> 0", "= 0")) ~ .,
  data = OralHealthNL, ylevels = 2:1, ylab = "dmfs")
```

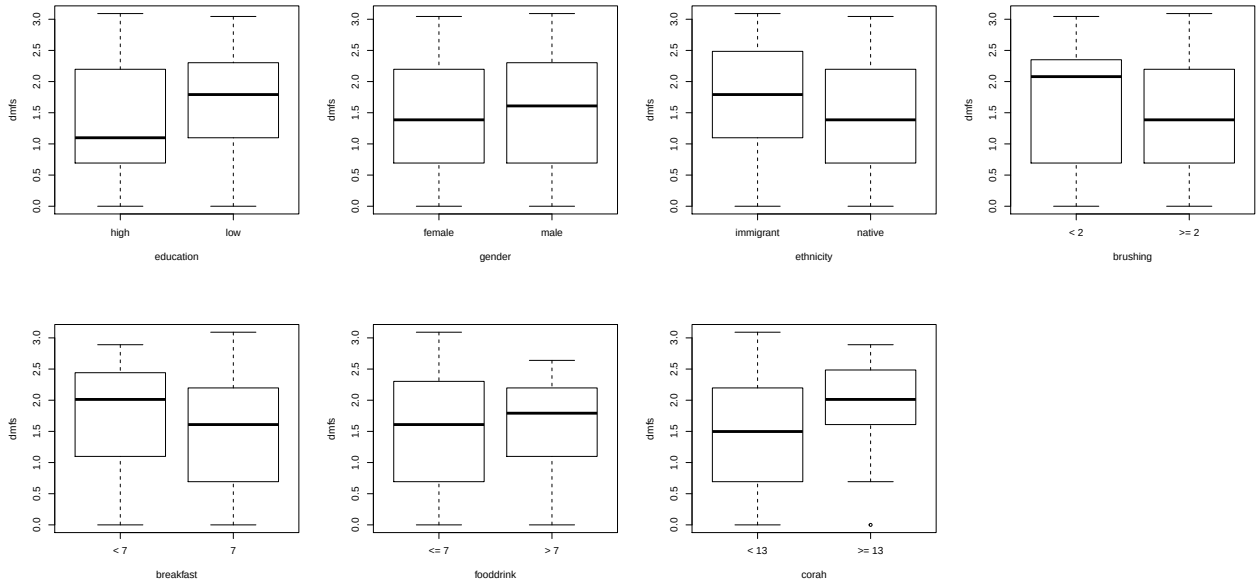


The black areas indicate the probability of `dmfs = 0`. The plots suggest that `breakfast`, `fooddrink`, and `corah` are most important. Note that the cutpoint of 13 for the Corah-score leads to quasi-complete separation: All children with `corah >= 13` also have `dmfs > 0`. Quasi-complete separation is easy to deal with in the hurdle model because the zero and the count model can be estimated separately. For details see below.

Count: How large is log(dmfs) given dmfs > 0?

For the subset of observations with `dmfs > 0`, we inspect the log-transformed outcome plotted against each regressor.

```
par(mfrow = c(2, 4))
plot(log(dmfs) ~ ., data = OralHealthNL, subset = dmfs > 0, ylab = "dmfs")
```



Main analyses: Count data regression models

The following competing models for count data are fitted: zero-inflated Poisson (ZIP), zero-inflated negative binomial (ZINB), hurdle Poisson (HP), and hurdle negative binomial (HNB).

```
#Relevel the factor variables in such a way that non-risk group is the reference
OralHealthNL <- within(OralHealthNL, ethnicity <- relevel(ethnicity, ref = "native"))
OralHealthNL <- within(OralHealthNL, brushing <- relevel(brushing, ref = ">= 2"))
OralHealthNL <- within(OralHealthNL, breakfast <- relevel(breakfast, ref = "7"))

zinb <- zeroinfl(dmfs ~ ., data = OralHealthNL, dist = "negbin")
zip <- zeroinfl(dmfs ~ ., data = OralHealthNL, dist = "poisson")
hnb <- hurdle(dmfs ~ ., data = OralHealthNL, dist = "negbin")
hp <- hurdle(dmfs ~ ., data = OralHealthNL, dist = "poisson")
```

It is quite clear that the NB generally performs better than Poisson. The information criteria (AIC and BIC) of the ZINB and HNB are very similar, with a slightly better performance of HNB.

```
cbind(AIC(hnb, zinb, hp, zip), BIC = BIC(hnb, zinb, hp, zip)[, 2])
```

```
##      df      AIC      BIC
## hnb  17 1710.477 1778.161
## zinb 17 1712.954 1780.638
## hp   16 1969.529 2033.232
## zip  16 1969.571 2033.274
```

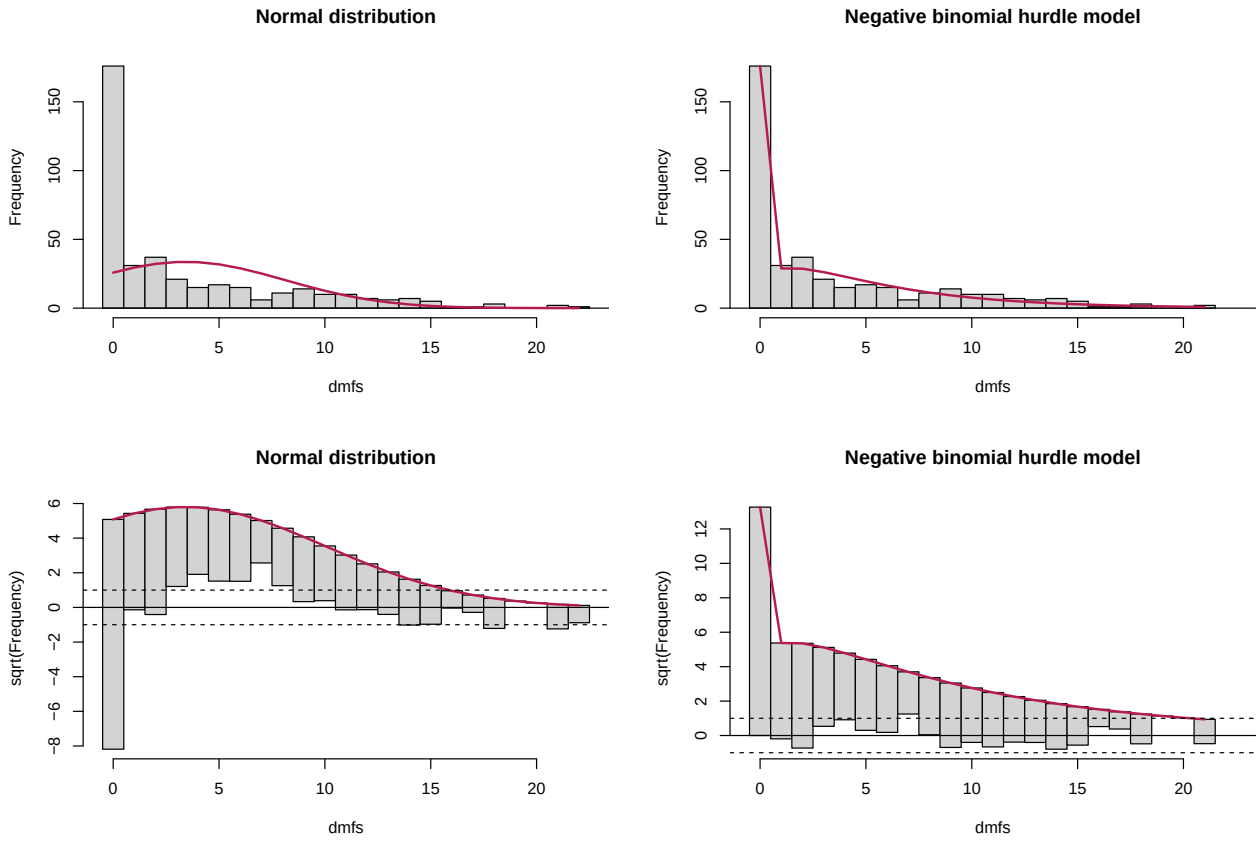
Rootograms

To show that the HNB model is really necessary and fits better than a normal model, histograms of the outcome with the fitted model can be used. In this example, the normal model has no regressors but the histograms would be similar if regressors were included.

The histograms can be created with the `rootogram()` function from the `countreg` package. By default, this function does not create a standard histogram but instead shows the square-root of the frequencies to create approximately equal variances (that allow to better judge model deviations for small frequencies). Furthermore, the deviations are by default "hanging" from the fitted frequencies in order to align the deviations along the x-axis.

In the following display, the first row uses "standing histograms" and the second "hanging rootograms".

```
par(mfrow = c(2, 2))
rootogram(OralHealthNL$dmfs, "normal",
  style = "standing", scale = "raw",
  breaks = 0:23 - 0.5, xlim = c(-0.5, 22.5),
  xlab = "dmfs", main = "Normal distribution")
rootogram(hnb,
  style = "standing", scale = "raw",
  width = 1, xlim = c(-0.5, 22.5),
  xlab = "dmfs", main = "Negative binomial hurdle model")
rootogram(OralHealthNL$dmfs, "normal",
  breaks = 0:23 - 0.5, xlim = c(-0.5, 22.5),
  xlab = "dmfs", main = "Normal distribution")
abline(h = c(-1, 1), lty = 2)
rootogram(hnb,
  width = 1, xlim = c(-0.5, 22.5),
  xlab = "dmfs", main = "Negative binomial hurdle model")
abline(h = c(-1, 1), lty = 2)
```



Model comparisons

The comparison of information criteria above already tells essentially the whole story: (1) NB fits better than Poisson, (2) the information criteria of HNB and ZINB are comparable with very slight advantages for HNB.

Here, we also look at a formal likelihood ratio test for HP vs. HNB:

```
lrtest(hnb, hp)
```

```
## Likelihood ratio test
##
## Model 1: dmfs ~ .
## Model 2: dmfs ~ .
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  17 -838.24
## 2  16 -968.76 -1 261.05  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value would have to be halved as it corresponds to testing whether the parameter `theta` is on the boundary of its parameter space. However, it is already virtually zero.

Essentially the same result is obtained for `lrtest(zinb, zip)`.

A Vuong test for comparing the ZINB and HNB also prefers the latter. This is not surprising given that they have the same number of parameters but the HNB has a better log-likelihood. However, the null hypothesis of indistinguishability cannot be rejected.

```
vuongtest(zinb, hnb)
```

```
##
## Model 1
##  Class: zeroinfl
##  Call: zeroinfl(formula = dmfs ~ ., data = OralHealthNL, dist = "negbin")
##
## Model 2
##  Class: hurdle
##  Call: hurdle(formula = dmfs ~ ., data = OralHealthNL, dist = "negbin")
##
##
## Variance test
##   H0: Model 1 and Model 2 are indistinguishable
##   H1: Model 1 and Model 2 are distinguishable
##   w2 = 0.001,   p = 0.382
##
## Non-nested likelihood ratio test
##   H0: Model fits are equal for the focal population
##   H1A: Model 1 fits better than Model 2
##   z = -2.511,   p = 0.994
##   H1B: Model 2 fits better than Model 1
##   z = -2.511,   p = 0.006015
```

Both models predict the number of zeros very well.

```
c(dmfs = sum(OralHealthNL$dmfs == 0),
  ZINB = sum(predict(zinb, type = "prob")[,1]),
  Hurdle = sum(predict(hnb, type = "prob")[,1]))
```

```
##     dmfs     ZINB   Hurdle
## 176.0000 177.1231 176.0000
```

The fitted means from both models are very closely correlated and moderately correlated with the outcome.

```
cor(cbind(dmfs = OralHealthNL$dmfs, ZINB = fitted(zinb), HNB = fitted(hnb)))
```

```
##           dmfs     ZINB       HNB
## dmfs 1.0000000 0.348679 0.3490063
## ZINB 0.3486790 1.000000 0.9997870
## HNB  0.3490063 0.999787 1.0000000
```

Coefficient tables

Finally, the estimated coefficients from both models can be compared. As zeros and non-zeros in `dmfs` are apparently well-separated, the ZINB and HNB model are very close.

```
mtable(ZINB = zinb, HNB = hnb, digits = 2)
```

```
##
## Calls:
## ZINB: zeroinfl(formula = dmfs ~ ., data = OralHealthNL, dist = "negbin")
## HNB: hurdle(formula = dmfs ~ ., data = OralHealthNL, dist = "negbin")
##
## ========================================================================
##                                    ZINB                 HNB
##                                  count     zero      count     zero
## ------------------------------------------------------------------------
## (Intercept)                     1.31***     0.06    1.29***    -0.33
##                                (0.13)      (0.24)  (0.13)      (0.20)
## education: low/high             0.30*      -0.34    0.30*       0.40
##                                (0.13)      (0.25)  (0.13)      (0.21)
## gender: male/female             0.05        0.07    0.05       -0.04
##                                (0.12)      (0.25)  (0.12)      (0.21)
## ethnicity: immigrant/native     0.33*      -0.46    0.34*       0.49
##                                (0.15)      (0.34)  (0.15)      (0.29)
## brushing: < 2/>= 2              0.35*      -0.24    0.38**      0.26
##                                (0.14)      (0.31)  (0.14)      (0.27)
## breakfast: < 7                  0.15       -1.41*   0.14        1.26**
##                                (0.18)      (0.65)  (0.19)      (0.48)
## fooddrink: > 7/<= 7            -0.07       -1.21   -0.08        0.98*
##                                (0.19)      (0.65)  (0.20)      (0.46)
## corah: >= 13/< 13               0.43*     -16.15    0.38       16.15
##                                (0.21)    (974.75)  (0.22)    (865.50)
## Log(theta)                      0.57**              0.51**
##                                (0.18)              (0.19)
## ------------------------------------------------------------------------
## Log-likelihood                       -839.48             -838.24
## AIC                                  1712.95             1710.48
## BIC                                  1780.64             1778.16
## N                                     396                 396
## ========================================================================
```

Notice the effect of `corah`: All children with `corah >= 13` also had `dmfs > 0`. Hence, the corresponding coefficient in the zero components of the models is huge. The value of 16 on the logit scale is essentially infinite. The corresponding standard errors cannot be estimated (are also infinite) causing a lack of significance stars. But, of course, the `corah` variable is highly relevant in the zero components!

Odds ratios and rate ratios are obtained by taken the exponential of the coefficients. In addition, 95% confidence intervals around these estimates can be obtained.

```
exp(confint(hnb))
```

```
##                                2.5 %   97.5 %
## count_(Intercept)          2.8176305 4.710609
## count_educationlow         1.0516938 1.747675
## count_gendermale           0.8232799 1.342851
## count_ethnicityimmigrant   1.0410117 1.879962
## count_brushing< 2          1.1019136 1.931618
## count_breakfast< 7         0.7983455 1.649344
## count_fooddrink> 7         0.6261915 1.354050
## count_corah>= 13           0.9530647 2.263966
## zero_(Intercept)           0.4890990 1.062205
```

```
## zero_educationlow        0.9851189 2.278641
## zero_gendermale          0.6298696 1.454358
## zero_ethnicityimmigrant  0.9206978 2.882191
## zero_brushing< 2         0.7686977 2.198867
## zero_breakfast< 7        1.3831085 8.937661
## zero_fooddrink> 7        1.0928891 6.542907
## zero_corah>= 13          0.0000000      Inf
```

```
exp(confint(zinb))
```

```
##                               2.5 %    97.5 %
## count_(Intercept)        2.87728799 4.7399839
## count_educationlow       1.05358981 1.7408558
## count_gendermale         0.83032458 1.3392020
## count_ethnicityimmigrant 1.03700891 1.8491811
## count_brushing< 2        1.08222465 1.8760869
## count_breakfast< 7       0.81383157 1.6536956
## count_fooddrink> 7       0.63596745 1.3590801
## count_corah>= 13         1.01247274 2.3333949
## zero_(Intercept)         0.67229117 1.6908065
## zero_educationlow        0.43459181 1.1554917
## zero_gendermale          0.66228485 1.7408712
## zero_ethnicityimmigrant  0.32595659 1.2220538
## zero_brushing< 2         0.42567249 1.4411974
## zero_breakfast< 7        0.06835275 0.8785863
## zero_fooddrink> 7        0.08307571 1.0765754
## zero_corah>= 13          0.00000000       Inf
```

Alternative model for the zero component: bias-reduced logistic regression

While the infinite coefficients in the maximum likelihood estimation clearly signal a high influence of the relevant regressor, they do prevent an interpretation of the absolute size (e.g., in terms of odds ratios). To overcome this issue one may try to estimate a regularized zero hurdle. One possibility to do so is to employ a bias-reduced logistic regression (as provided in `brglm`). This yields a large but finite coefficient on the binary `corah` regressor (along with finite standard errors) while shrinking the remaining variables only very moderately.

```
br <- brglm(factor(dmfs == 0, levels = c(TRUE, FALSE), labels = c("= 0", "> 0")) ~ .,
  data = OralHealthNL)
print(coeftest(br), digits = 1)
```

```
##
## z test of coefficients:
##
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.32       0.20    -1.6     0.10
## educationlow           0.40       0.21     1.9     0.06 .
## gendermale            -0.04       0.21    -0.2     0.84
## ethnicityimmigrant     0.47       0.29     1.6     0.10
## brushing< 2            0.26       0.27     1.0     0.34
## breakfast< 7           1.19       0.46     2.6     0.01 *
## fooddrink> 7           0.93       0.45     2.1     0.04 *
```

```
## corah>= 13             3.33       1.48     2.3     0.02 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```