Medical Care - Comparison of different count data models

January 5, 2012

To fit the count data from the medcare data set Poisson Models, Negative Binomial Models, Zero Inflated Models and Hurdle Models will be fitted. To fit a Negative Binomial Generalized Linear Model we will use the function "glm.nb" from the "MASS" package. The package "pscl" will be needed to fit Zero Inflated Models and Hurdle Models.

```
> library(catdata)
> data(medcare)
> library(MASS)
> library(pscl)
```

The data from the medcare data set are reduced to males with responses smaller than 30.

```
> medmale <- medcare[medcare$male == 1, ]
> medmale <- medmale[medmale$ofp <= 30, ]
```

Now we split the data 50 times randomly into a training and a test data set and fit a Poisson Modell, an Negative Binomial Modell, two Zero Inflated Model and two Hurdle Models (each timne one model with only the Intercept for the mixture, one with predictors in the mixture). As a prediction we take the Median of the predicted distributions.

```
> set.seed(5)
> subs <- 1:nrow(medmale)
> reps <- 50
> squerror1 <- squerror3 <- squerror4 <- squerror5 <- squerror6 <- squerror7 <- c()
> abserror1 <- abserror3 <- abserror4 <- abserror5 <- abserror6 <- abserror7 <- c()
> for (i in 1:reps) {
+     learn <- sample(subs, 600)
+     test <- subs[-learn]
+     med = glm(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school, family = poisson,
+         data = medmale[learn, ])
+     l <- predict(med, newdata = medmale[test,
+         ], type = "response")
+     a <- rep(0, length(medmale[test, 1]))
+     for (j in 1:length(medmale[test, 1])) {
+         while (ppois(a[j], lambda = l[j]) < 0.5) {
```

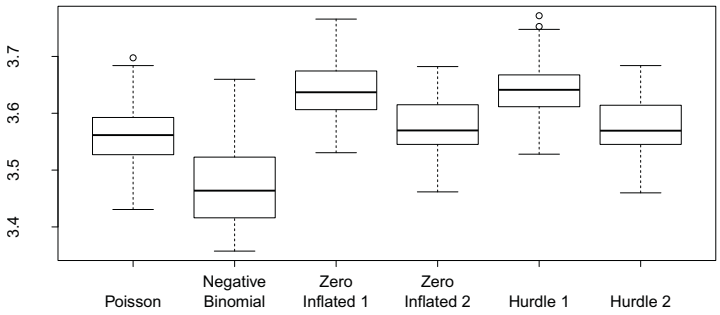
```
+             a[j] <- a[j] + 1
+         }
+     }
+     diffs <- a - medmale[test, 1]
+     squerror1[i] <- mean(diffs^2)
+     abserror1[i] <- mean(abs(diffs))
+     med3 = glm.nb(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school, data = medmale[learn,
+         ])
+     l <- predict(med3, newdata = medmale[test,
+         ], type = "response")
+     a <- rep(0, length(medmale[test, 1]))
+     for (j in 1:length(medmale[test, 1])) {
+         while ((pnbinom(a[j], mu = l[j], size = med3$theta)) <
+             0.5) {
+             a[j] <- a[j] + 1
+         }
+     }
+     diffs <- a - medmale[test, 1]
+     squerror3[i] <- mean(diffs^2)
+     abserror3[i] <- mean(abs(diffs))
+     med4 = zeroinfl(ofp ~ hosp + healthpoor +
+         healthexcellent + numchron + age + married +
+         school | 1, data = medmale[learn, ])
+     pii <- 1 - predict(med4, newdata = medmale[test,
+         ], type = "zero")
+     mui <- predict(med4, newdata = medmale[test,
+         ], type = "count")
+     a <- rep(0, length(medmale[test, 1]))
+     for (j in 1:length(medmale[test, 1])) {
+         cdis <- 0
+         while (cdis < 0.5) {
+             cdis <- cdis + pii[j] * exp(-mui[j]) *
+                 ((mui[j]^a[j])/factorial(a[j])) +
+                 (1 - pii[j]) * I(a[j] == 0)
+             a[j] <- a[j] + 1
+         }
+         a[j] <- a[j] - 1
+     }
+     diffs <- a - medmale[test, 1]
+     squerror4[i] <- mean(diffs^2)
+     abserror4[i] <- mean(abs(diffs))
+     med5 = zeroinfl(ofp ~ hosp + healthpoor +
+         healthexcellent + numchron + age + married +
+         school, data = medmale[learn, ])
+     pii <- 1 - predict(med5, newdata = medmale[test,
+         ], type = "zero")
+     mui <- predict(med5, newdata = medmale[test,
+         ], type = "count")
+     a <- rep(0, length(medmale[test, 1]))
```

```
+     for (j in 1:length(medmale[test, 1])) {
+         cdis <- 0
+         while (cdis < 0.5) {
+             cdis <- cdis + pii[j] * exp(-mui[j]) *
+                 ((mui[j]^a[j])/factorial(a[j])) +
+                 (1 - pii[j]) * I(a[j] == 0)
+             a[j] <- a[j] + 1
+         }
+         a[j] <- a[j] - 1
+     }
+     diffs <- a - medmale[test, 1]
+     squerror5[i] <- mean(diffs^2)
+     abserror5[i] <- mean(abs(diffs))
+     med6 = hurdle(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school | 1,
+         data = medmale[learn, ])
+     mui <- predict(med6, newdata = medmale[test,
+         ], type = "count")
+     gammai <- predict(med6, newdata = medmale[test,
+         ], type = "zero")
+     pii2 <- 1 - gammai * (1 - exp(-mui))
+     fa <- function(z, a) {
+         ((z^a)/factorial(a)) * exp(-z)
+     }
+     a <- rep(0, length(medmale[test, 1]))
+     for (j in 1:length(medmale[test, 1])) {
+         cdis <- pii2[j]
+         if (cdis < 0.5) {
+             while (cdis < 0.5) {
+                 cdis <- cdis + fa(z = mui[j],
+                   a = (a[j] + 1)) * gammai[j]
+                 a[j] <- a[j] + 1
+             }
+         }
+         else {
+             a[j] <- 0
+         }
+     }
+     diffs <- a - medmale[test, 1]
+     squerror6[i] <- mean(diffs^2)
+     abserror6[i] <- mean(abs(diffs))
+     med7 = hurdle(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school, data = medmale[learn,
+         ])
+     mui <- predict(med7, newdata = medmale[test,
+         ], type = "count")
+     gammai <- predict(med7, newdata = medmale[test,
+         ], type = "zero")
+     pii2 <- 1 - gammai * (1 - exp(-mui))
+     fa <- function(z, a) {
```

```
+         ((z^a)/factorial(a)) * exp(-z)
+     }
+     a <- rep(0, length(medmale[test, 1]))
+     for (j in 1:length(medmale[test, 1])) {
+         cdis <- pii2[j]
+         if (cdis < 0.5) {
+             while (cdis < 0.5) {
+                 cdis <- cdis + fa(z = mui[j],
+                   a = (a[j] + 1)) * gammai[j]
+                 a[j] <- a[j] + 1
+             }
+         }
+         else {
+             a[j] <- 0
+         }
+     }
+     diffs <- a - medmale[test, 1]
+     squerror7[i] <- mean(diffs^2)
+     abserror7[i] <- mean(abs(diffs))
+ }
```

Here the absolut 50 prediction errors of the respective models are plotted.

```
> par(mgp = c(0, 3, 0))
> boxplot(abserror1, abserror3, abserror4, abserror5,
+     abserror6, abserror7, names = c("Poisson",
+         "Negative\nBinomial", "Zero\nInflated 1",
+         "Zero\nInflated 2", "Hurdle 1", "Hurdle 2"),
+     cex = 1.3, cex.axis = 1.6)
```



Now the ranked probability score is computed. Here the difference between the predicted and the empirical cumulative distribution function is measured.

```
> library(catdata)
> data(medcare)
> library(MASS)
> library(pscl)
> medmale <- medcare[medcare$male == 1, ]
> medmale <- medmale[medmale$ofp <= 30, ]
```

```
> set.seed(5)
> subs <- 1:nrow(medmale)
> reps <- 50
> eps <- 1e-05
> lrps <- lrps3 <- lrps4 <- lrps5 <- lrps6 <- lrps7 <- rep(0,
+     reps)
> for (i in 1:reps) {
+     learn <- sample(subs, 600)
+     test <- subs[-learn]
+     med = glm(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school, family = poisson,
+         data = medmale[learn, ])
+     l <- predict(med, newdata = medmale[test,
+         ], type = "response")
+     for (j in 1:length(medmale[test, 1])) {
+         for (a in 0:100) {
+             lrps[i] <- lrps[i] + (ppois(a, lambda = l[j]) -
+                 I(medmale[test, 1][j] <= a))^2
+         }
+     }
+     med3 = glm.nb(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school, data = medmale[learn,
+         ])
+     l <- predict(med3, newdata = medmale[test,
+         ], type = "response")
+     for (j in 1:length(medmale[test, 1])) {
+         for (a in 0:100) {
+             lrps3[i] <- lrps3[i] + (pnbinom(a,
+                 mu = l[j], size = med3$theta) -
+                 I(medmale[test, 1][j] <= a))^2
+         }
+     }
+     med4 = zeroinfl(ofp ~ hosp + healthpoor +
+         healthexcellent + numchron + age + married +
+         school | 1, data = medmale[learn, ])
+     pii <- 1 - predict(med4, newdata = medmale[test,
+         ], type = "zero")
+     mui <- predict(med4, newdata = medmale[test,
+         ], type = "count")
+     for (j in 1:length(medmale[test, 1])) {
+         a <- 0
+         cumdist <- (1 - pii[j])
+         for (a in 0:100) {
+             cumdist <- cumdist + pii[j] * exp(-mui[j]) *
+                 (mui[j]^a) * (factorial(a)^(-1))
+             lrps4[i] <- lrps4[i] + as.numeric((cumdist -
+                 I(medmale[test, 1][j] <= a))^2)
+         }
+     }
+     med5 = zeroinfl(ofp ~ hosp + healthpoor +
```

```
+         healthexcellent + numchron + age + married +
+         school, data = medmale[learn, ])
+     pii <- 1 - predict(med5, newdata = medmale[test,
+         ], type = "zero")
+     mui <- predict(med5, newdata = medmale[test,
+         ], type = "count")
+     for (j in 1:length(medmale[test, 1])) {
+         a <- 0
+         cumdist <- (1 - pii[j])
+         for (a in 0:100) {
+             cumdist <- cumdist + pii[j] * exp(-mui[j]) *
+                 (mui[j]^a) * (factorial(a)^(-1))
+             lrps5[i] <- lrps5[i] + as.numeric((cumdist -
+                 I(medmale[test, 1][j] <= a))^2)
+         }
+     }
+     med6 = hurdle(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school | 1,
+         data = medmale[learn, ])
+     mui <- predict(med6, newdata = medmale[test,
+         ], type = "count")
+     gammai <- predict(med6, newdata = medmale[test,
+         ], type = "zero")
+     pii2 <- 1 - gammai * (1 - exp(-mui))
+     fa <- function(z, a) {
+         ((z^a)/factorial(a)) * exp(-z)
+     }
+     for (j in 1:length(medmale[test, 1])) {
+         cumdist <- pii2[j]
+         for (a in 1:100) {
+             cumdist <- cumdist + fa(z = mui[j],
+                 a = a) * gammai[j]
+             lrps6[i] <- lrps6[i] + as.numeric((cumdist -
+                 I(medmale[test, 1][j] <= a))^2)
+         }
+     }
+     med7 = hurdle(ofp ~ hosp + healthpoor + healthexcellent +
+         numchron + age + married + school, data = medmale[learn,
+         ])
+     mui <- predict(med7, newdata = medmale[test,
+         ], type = "count")
+     gammai <- predict(med7, newdata = medmale[test,
+         ], type = "zero")
+     pii2 <- 1 - gammai * (1 - exp(-mui))
+     fa <- function(z, a) {
+         ((z^a)/factorial(a)) * exp(-z)
+     }
+     for (j in 1:length(medmale[test, 1])) {
+         cumdist <- pii2[j]
+         for (a in 1:100) {
```

```
+             cumdist <- cumdist + fa(z = mui[j],
+                 a = a) * gammai[j]
+             lrps7[i] <- lrps7[i] + as.numeric((cumdist -
+                 I(medmale[test, 1][j] <= a))^2)
+         }
+     }
+ }
> lrps <- lrps/length(medmale[test, 1])
> lrps3 <- lrps3/length(medmale[test, 1])
> lrps4 <- lrps4/length(medmale[test, 1])
> lrps5 <- lrps5/length(medmale[test, 1])
> lrps6 <- lrps6/length(medmale[test, 1])
> lrps7 <- lrps7/length(medmale[test, 1])
```

Now the respective ranked probability scores are plotted.

```
> par(mgp = c(0, 3, 0))
> boxplot(lrps, lrps3, lrps4, lrps5, lrps6, lrps7,
+     names = c("Poisson", "Negative\nBinomial",
+         "Zero\nInflated 1", "Zero\nInflated 2",
+         "Hurdle 1", "Hurdle 2"), cex = 1.3, cex.axis = 1.6)
```