R-tutorial: A weighted partial likelihood approach for zero-truncated models

Wen-Han Hwang, Dean Heinze and Jakub Stoklosa

05 April, 2019

Example 1: Mt Little Higginbotham mountain pygmy possum data

These data can be obtained from Web Table 1.

```
library(sandwich) # Load the "sandwich" package.

tau <- 3 # No. of capture occasions.
D <- 62 # No. of uniquely caught possums.

# Observed frequencies:

y <- c(2, 3, 1, 3, 2, 2, 1, 3, 2, 2, 2, 1, 2, 2, 3, 1, 1, 1, 1, 1, 1, 1, 2, 1, 3,
     1, 2, 1, 2, 1, 1, 2, 1, 1, 1, 3, 2, 2, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1,
     1, 1, 1, 1, 2, 1, 1, 2, 1, 3, 3, 1)

# First capture times (t_i) for each possum:

t1 <- c(1, 1, 3, 1, 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 2, 2, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 2, 2, 2, 2, 2, 1, 3, 3, 3, 3, 3, 3,
        2, 3, 3, 3, 3, 1, 1, 3, 1, 1, 2, 1, 1 ,1)

# Gender covariate:

x.obs <- c(0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0,
           1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1,
           1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
```

Data summaries:

```
sum(y) # Total no. of captures.
```

```
## [1] 96
```

```
mean(y) # Avergae capture rate.
```

```
## [1] 1.548387
```

```
sum(x.obs) # No. of males.
```

```
## [1] 27
```

```
D - sum(x.obs) # No. of females.
```

```
## [1] 35
```

```r
fs <- table(y)

f1 <- fs[1]
f2 <- fs[2]
f3 <- fs[3]

c(f1, f2, f3) # Frequency of individuals caught exactly x times.
```

```
##  1  2  3
## 37 16  9
```

A function that calculates population size estimates (and standard errors):

```r
# The "m" denotes a glm model object, tau and y are as defined as above

VarNhat.glm <- function(m, tau, y = NULL){
  X <- model.matrix(m)
  beta <- coef(m)
  P <- c(1 / (1 + exp(-X%*%beta)))
  Pi <- 1 - (1 - P)^tau
  Nhat <- sum(Pi^(-1)) # Population size (Horvitz-Thompson) estimator.

  # Standard error estimator using the "sandwich" package. Below we give the
  # standard error estimator for Nhat, see Huggins(1989) for further details.

  var.beta <- sandwich(m) # Variance estimates for model regression coefficients (beta).
  gdash.beta <- t(X)%*%(Pi^(-2)*(1 - P)^tau*tau*P)
  varA<-sum((1 - Pi)/Pi^2)
  varB <- (t(gdash.beta)%*%var.beta)%*%gdash.beta
  varN <- as.vector(varA + varB)
  Se.Nhat <- sqrt(varN)

  return(list(Se.beta = sqrt(diag(var.beta)), Nhat = Nhat, Se.Nhat = Se.Nhat))
}
```

Use full data and fit linear logistic regression (M_0/M_h) models.

Partial likelihood approach:

```r
# Construct PL weights to feed into glm().

R <- y-1
h <- tau-t1
y.p <- R/h
y.p[is.na(y.p)] <- 0

est.PL_0 <- glm(y.p ~ 1, weights = h, family = binomial)
est.PL_const <- VarNhat.glm(est.PL_0, tau, y = y)
```

```r
est.PL_1 <- glm(y.p ~ x.obs, weights = h, family = binomial)
est.PL_Mh <- VarNhat.glm(est.PL_1, tau, y = y)
```

Weighed partial likelihood approach:

```r
# Construct WPL weights to feed into glm.

m.tilde.star <- tau-(tau + 1)/(y + 1)
h.wpl <- m.tilde.star
y.wpl <- R/h.wpl
y.wpl[is.na(y.wpl)] <- 0

est.WPL_0 <- glm(y.wpl ~ 1, weights = h.wpl, family = binomial)
est.WPL_const <- VarNhat.glm(est.WPL_0, tau, y = y)

est.WPL_1 <- glm(y.wpl ~ x.obs, weights = h.wpl, family = binomial)
est.WPL_Mh <- VarNhat.glm(est.WPL_1, tau, y = y)
```

Combine results and display them:

```r
# These should be the same as the first few rows of Table 3.

N_ests <- matrix(round(as.numeric(rbind(c(est.PL_const$Nhat, est.PL_const$Se.Nhat),
                                      c(est.WPL_const$Nhat, est.WPL_const$Se.Nhat),
                                      c(est.PL_Mh$Nhat, est.PL_Mh$Se.Nhat),
                                      c(est.WPL_Mh$Nhat, est.WPL_Mh$Se.Nhat))),
                                      digits = 2), ncol = 2)

rownames(N_ests) <- c("PL", "WPL", "PL-h", "WPL-h")
colnames(N_ests) <- c("N_hat", "S.E.(N_hat)")
round(N_ests, digits = 2)
```

```
##       N_hat S.E.(N_hat)
## PL    76.55        5.88
## WPL   77.39        6.98
## PL-h  76.72        6.06
## WPL-h 81.06        9.28
```

Example 2: Variable selection using GLMNET

The 1987/88 US National Medical Expenditure Survey (NMES) count data were obtained from: https://www.jstatsoft.org/article/view/v027i08

```r
suppressMessages(library(glmnet)) # Load the "glmnet" package.
suppressMessages(library(MASS)) # Load the "MASS" package.

# Load data and extract all variables.

load(file = "DebTrivedi.rda") # Load the data.

dt <- DebTrivedi[,c(1, 5, 6, 8, 9, 11:19)]
dt[, 5] <- as.numeric(dt[, 5])-1
dt[, 7] <- as.numeric(dt[, 7])-1
dt[, 8] <- as.numeric(dt[, 8])-1
dt[, 9] <- as.numeric(dt[, 9])-1
dt[, 12] <- as.numeric(dt[, 12])-1
dt[, 13] <- as.numeric(dt[, 13])-1
dt[, 14] <- as.numeric(dt[, 14])-1
```

Remove all zero counts from data to create artificial zero-truncated data:

```r
dt2 <- dt[-which(dt$ofp==0), ]

y <- dt2$ofp
n <- length(y)

X <- cbind(rep(1, n), dt2[, -1])
colnames(X)[1] <- "(Intercept)"
```

Fit models and apply model selection (AIC and GLMNET):

```r
# Construct WPL weights which feed into glm() and glmnet().

t.tilde.star <- y/(y + 1)
y.tilde <- y-1

# Fit models.

dat2 <- data.frame(cbind(y.tilde, X))

offs <- log(t.tilde.star)

mod2 <- glm(y.tilde ~ emer + hosp + numchron + adldiff + age + black +
              gender + married + school + faminc + employed + privins + medicaid,
            offset = offs, family = poisson, data = dat2)

AIC.glm2 <- stepAIC(mod2, trace = FALSE)$coefficients
```

```r
mod3 <- glmnet(as.matrix(X), y.tilde, family = "poisson",
               offset = log(t.tilde.star))

s <- cv.glmnet(as.matrix(X), y.tilde, family = "poisson",
               offset = log(t.tilde.star))$lambda.min

tmp_coeffs <- coef(mod3, s = s)

mod3.coef <- data.frame(name = tmp_coeffs@Dimnames[[1]][tmp_coeffs@i + 1],
                        coefficient = tmp_coeffs@x)
```

Combine results and display them:

```r
# Should be the same as the second column of Table 4.
AIC.glm2
```

```
##  (Intercept)         emer         hosp     numchron      adldiff
##  2.034776151  0.027342355  0.149985303  0.108174289  0.153897263
##          age      married       school       faminc     employed
## -0.085488130 -0.078854637  0.018825464 -0.004103332  0.045905594
##      privins     medicaid
##  0.144420553  0.185337232
```

```r
# These will be slightly different from the last column of Table 4 because
# glmnet() uses cross-validation to select lambda, thus the data is randomly
# split and will consist of different training/test sets for each fit.
t(mod3.coef)
```

```
##             [,1]           [,2]           [,3]           [,4]
## name        "(Intercept)"  "emer"         "hosp"         "numchron"
## coefficient " 2.013577748" " 0.026494675" " 0.149580192" " 0.106808797"
##             [,5]           [,6]           [,7]           [,8]
## name        "adldiff"      "age"          "gender"       "married"
## coefficient " 0.146736245" "-0.078641418" "-0.007810643" "-0.069292346"
##             [,9]           [,10]          [,11]          [,12]
## name        "school"       "faminc"       "employed"     "privins"
## coefficient " 0.017647390" "-0.002792350" " 0.035659732" " 0.127172568"
##             [,13]
## name        "medicaid"
## coefficient " 0.164513893"
```