hurdle model

Jieun Shin

2023-05-16

1. zero-inflated model

기본적으로 zero-inflated 모형은 하나의 분포에서 구조적으로 0이 발생할 확률을 추가함므로 만들어진다. $f(y_i)$를 pdf라고 하면 확률변수 Y_i는 다음과 같이 2개의 부분을 따른다:

$$Y_i \sim \begin{cases} 0, & \text{w.p. } \phi_i \\ f(y_i), & \text{w.p. } 1-\phi_i \end{cases}$$

여기서 zero값의 발생은 1) 구조적으로 (필연적으로) 발생하거나 2) 랜덤하게 발생된다. zero값과 non-zero값이 발생할 확률은 다음과 같다:

$$P(Y_i = 0) = \phi_i + (1-\phi_i)f(0), \quad \text{if } y_i = 0$$
$$P(Y_i = y_i) = (1-\phi_i)f(y_i), \quad \text{if } y_i > 0$$

또 다른 zero-inflated 모형화를 위한 모형으로 허들 모형이 사용될 수 있다. 허들 모형은 zero count part와 positive counts가 서로 다른 확률모형으로부터 나온다고 가정한다:

$$P(Y = j) = \begin{cases} f_1(0), & \text{if } j = 0 \\ \frac{1-f_1(0)}{1-f_2(0)} f_2(j), & \text{if } j > 0 \end{cases}$$

여기서 f_1는 zero count의 발생과 관련한 pdf, f_2는 positive count와 관련한 pdf이며 $j > 0$ 부분은 zero-truncated pdf에 해당한다. 허들 모형은 zero-inflated 모형으로 축소 (restricted)될 수 없다.

2. hurdle model

허들 모형의 모수도 regression fomula를 가진다. 앞서 정의했던 두 개의 pdf f_1과 f_2가 포아송 분포이면 포아송 허들모형이 된다:

$$f_{1i}(0) = \exp(-\mu_{1i}), \quad \mu_{1i} = \exp(x_{1i}^T\beta_1) f_{2i}(y_i) = \frac{\mu_{2i}^{y_i}\exp(-\mu_{2i})}{y_i!}, \quad \mu_{2i} = \exp(x_{2i}^T\beta_2)$$

그리고 f_1과 f_2가 음이항 분포이면 음이항 허들모형이 된다:

$$f_{1i}(0) = (1+\tau_1\mu_{1i})^{-1/\tau_1}, \quad \mu_{1i} = \exp(x_{1i}^T\beta_1) f_{2i}(y_i) = NB(\mu_{2i}, \tau_2), \quad \mu_{2i} = \exp(x_{2i}^T\beta_2)$$

만약 $f_1(0) = \phi(1 < \phi < 1)$로 정의하면 sampling zero는 없고 structure zero만 갖는 허들모형이 되며, phi_i는 $\log\frac{\phi_i}{1-\phi_i} = Z_i^T\gamma$와 같이 모형화할 수 있다.

허들 모형의 모수는 최대가능도 추정으로 구할 수 있다. 먼저 데이터를 zero와 non-zero의 두 부분으로 나누기 위해 indicator를 정의하자:

$$d_i = \begin{cases} 1, & \text{if } y_i = 0 \\ 0, & \text{if } y_i > 0 \end{cases}$$

그러면 i번째 관측치에 대한 밀도함수를 다음과 같이 쓸 수 있다:

$$
\begin{aligned}
f(y_i) &= f_1(0|x_i,\theta_1)^{d_i} \times \left[\frac{1-f_1(0|x_i,\theta_1)}{1-f_2(0|x_i,\theta_2)} f_2(y_i|x_i,\theta_2)\right]^{1-d_i} \\
&= \left[f_1(0|x_i,\theta_1)^{d_i}(1-f_1(0|x_i,\theta_1))^{1-d_i}\right] \times \left[\frac{f_2(y_i|x_i,\theta_2)}{1-f_2(0|x_i,\theta_2)}\right]^{1-d_i}
\end{aligned}
$$

그러면 log-likelihood는 다음과 같이 정의된다:

$$
\begin{aligned}
L(\theta_1,\theta_2) = &\sum_{i=1}^{n}\left[d_i \log f_1(0|x_i,\theta_1) + (1-d_i)\log(1-f_1(0|x_i,\theta_1))\right] \\
&+ \sum_{i=1}^{n}(1-d_i)\left[\log f_2(y_i|x_i,\theta_2) - \log(1-f_2(0|x_i,\theta_2))\right] \\
=&L(\theta_1) + L(\theta_2)
\end{aligned}
$$

log-likelihood이 정확하게 두 부분으로 분리가 되는 것을 알 수 있다. $L(\theta_1)$은 zero part와 non-zero part로 나누는 binary process와 관련된 로그가능도 함수이고, $L(\theta_2)$은 non-zero part에 대한 zero truncated count model의 로그가능도 함수이다. 따라서 θ_1과 θ_2에 대한 ML추정량은 한꺼번에 구하지 않고 분리해서 따로 구해도 된다.

3. simulation for hurdle model

추정을 위한 시뮬레이션을 진행한다. 허들 음이항 모형(hurdle negative binomial model)을 고려하기 위해 zero part의 함수는 $f_1 = \phi$으로 truncated pdf는 negative binomial distribution으로 두었다. 먼저 허들 음이항 모형을 따르는 난수를 생성하고 β를 추정한 후 `pscl` 패키지의 추정 결과와 비교하였다.

1. 난수 생성

```
rHNB <- function(n, zp, beta0, beta, tau) {
  bet = c(beta0, beta)
  p = length(beta)

  y <- 0:500
  x = cbind(1, matrix(runif(n*p), nrow = n, ncol = p))

  mu = c(exp(x %*% bet))
  ry = c()

  for(i in 1:n){

    temp <- c()

    for(j in 1:length(y)){
      if(y[j] == 0){
        p = zp
      } else{
        p = (1-zp)/(1-dnbinom(0, mu = mu[i], size = 1/tau)) * dnbinom(y[j], mu = mu[i], size = 1/tau)
      }
      temp[j] <- p
    }

    id = min(which(runif(1) <= cumsum(temp)))
    id = ifelse(id == Inf, max(y), id)
    ry[i] = y[id]
```

```
  }
  return(list(y = ry, x = x, mu = mu))
}

sim_dat = rHNB(200, zp = 0.1, beta0 = 1, beta = c(1, 0.2, 0.5), tau = 0.2)
Y = sim_dat$y
X = sim_dat$x
```

2. 추정

```
d = ifelse(Y > 0, 0, 1)  # indicator

zp_hat = mean(d)  # optim L1

# optim L2
L2_beta = function(beta){
  mu = c(exp(X %*% beta))
  lik = (1-d) * (-log(1-dnbinom(0, mu = mu, size = 1/tau))
                 + dnbinom(Y, mu = mu, size = 1/tau, log = TRUE))
  # print(sum(lik))
  return(-sum(lik))
}

L2_tau = function(tau){
  mu = c(exp(X %*% bet))
  lik = (1-d) * (-log(1-dnbinom(0, mu = mu, size = 1/tau))
                 + dnbinom(Y, mu = mu, size = 1/tau, log = TRUE))
  # print(sum(lik))
  return(-sum(lik))
}

# initialize
bet = runif(ncol(X))
tau = sd(Y)/sqrt(length(Y))

for(i in 1:10){
  bet = optim(par = bet, fn = L2_beta)$par
  tau = optim(par = tau, fn = L2_tau, method = "Brent", lower = 1e-10, upper = 10)$par
  cat("iter =", i, "\n")
  cat("beta_hat =", bet, "\n")
  cat("tau_hat =", tau, "\n")
  # if(norm(bet, "2") < 1e-4) break
}
```

```
## iter = 1
## beta_hat = 1.114013 0.8811326 0.1190585 0.6066946
## tau_hat = 0.2072966
## iter = 2
## beta_hat = 1.158211 0.8614977 0.1133115 0.5811724
## tau_hat = 0.2048383
## iter = 3
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
## tau_hat = 0.2048215
## iter = 4
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
```

```
## tau_hat = 0.2048215
## iter = 5
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
## tau_hat = 0.2048215
## iter = 6
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
## tau_hat = 0.2048215
## iter = 7
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
## tau_hat = 0.2048215
## iter = 8
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
## tau_hat = 0.2048215
## iter = 9
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
## tau_hat = 0.2048215
## iter = 10
## beta_hat = 1.158672 0.8611691 0.1131536 0.5810595
## tau_hat = 0.2048215
```

3. 패키지 결과와 비교

```r
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
dat_X = X[,-1]
hurdle(Y ~ dat_X, dist = "negbin", zero = "negbin")
```

```
##
## Call:
## hurdle(formula = Y ~ dat_X, dist = "negbin", zero.dist = "negbin")
##
## Count model coefficients (truncated negbin with log link):
## (Intercept)       dat_X1       dat_X2       dat_X3
##      1.1587       0.8612       0.1131       0.5811
## Theta = 4.8823
##
## Zero hurdle model coefficients (censored negbin with log link):
## (Intercept)       dat_X1       dat_X2       dat_X3
##     0.72552     -0.36235      0.07149      0.63164
## Theta = 3211.0685
```