Econometrics II TA Session #4

Hiroki Kato

1 Empirical Application of Ordered Probit and Logit Model: Housing as Status Goods

1.1 Background and Data

A desire to signal high income or wealth may cause consumers to purchase status goods such as luxury cars. In this application, we explore whether housing serves as status goods, using the case of apartment building. We investigate the relationship between living in a high floor and income, controlling the quality of housing. Our hypothesis is that high-earners are more likely to live on the upper floor.

We use the housing data originally coming from the American Housing Survey conducted in 2013 [1]. This dataset (hereafter `housing`) contains the following variables:

- `Level`: ordered value of a story of respondent's living (1:Low - 4:High)
- `lnPrice`: logged price of housing (proxy for quality of house)
- `Top25`: a dummy variable taking one if household income is in the top 25 percentile in sample.

We split data into two subsets: the *training* data and the *test* data. The training data, which is used for estimation and model fitness, is randoly drawn from the original data. The sample size of this subset is two thirds of total observations of the original one ($N = 1,074$). The test data, which is used for model prediction, consists of observations which the training data does not include ($N = 538$).

```
dt <- read.csv(file = "./data/housing.csv", header = TRUE,  sep = ",")
dt <- dt[,c("Level", "lnPrice", "Top25")]

set.seed(120511)
train_id <- sample(1:nrow(dt), size = (2/3)*nrow(dt), replace = FALSE)
train_dt <- dt[train_id,]; test_dt <- dt[-train_id,]

head(train_dt)
```

[1] https://www.census.gov/programs-surveys/ahs.html. This is a repeated cross-section survey. We use the data at one time.

```
##      Level   lnPrice Top25
## 1099     4  9.903538     0
## 2        4 11.512935     1
## 1398     4 11.775297     0
## 405      2 12.429220     0
## 579      1 11.289794     0
## 1157     1 10.596660     0
```

1.2 Model

The outcome variable is `Level` taking $\{1, 2, 3, 4\}$. Consider the following regression equation of a latent variable:

$$y_i^* = \mathbf{x}_i\beta + u_i,$$

where a vector of explanatory variables are `lnPrice` and `Top25`, and u_i is an error term. The relationship between the latent variable y_i^* and the observed outcome variable is

$$Level = \begin{cases} 1 & \text{if} \quad -\infty < y_i^* \leq a_1 \\ 2 & \text{if} \quad a_1 < y_i^* \leq a_2 \\ 3 & \text{if} \quad a_2 < y_i^* \leq a_3 \\ 4 & \text{if} \quad a_3 < y_i^* < +\infty \end{cases}.$$

Consider the probability of realization of y_i, that is,

$$\begin{aligned} \mathbb{P}(y_i = k|\mathbf{x}_i) &= \mathbb{P}(a_{k-1} - \mathbf{x}_i\beta < u_i \leq a_k - \mathbf{x}_i\beta|\mathbf{x}_i) \\ &= G(a_k - \mathbf{x}_i\beta) - G(a_{k-1} - \mathbf{x}_i\beta), \end{aligned}$$

where $a_4 = +\infty$ and $a_0 = -\infty$. Then, the likelihood function is defined by

$$p((y_i|\mathbf{x}_i), i = 1, \dots, n; \beta, a_1, \dots, a_3) = \prod_{i=1}^{n}\prod_{k=1}^{4}(G(a_k - \mathbf{x}_i\beta) - G(a_{k-1} - \mathbf{x}_i\beta))^{I_{ik}}.$$

where I_{ik} is a indicator variable taking 1 if $y_i = k$. Finally, the log-likelihood function is

$$M(\beta, a_1, a_2, a_3) = \sum_{i=1}^{n}\sum_{k=1}^{4} I_{ik}\log(G(a_k - \mathbf{x}_i\beta) - G(a_{k-1} - \mathbf{x}_i\beta)).$$

Usually, $G(a)$ assumes the standard normal distribution, $\Phi(a)$, or the logistic distribution, $1/(1 + \exp(-a))$.

In `R`, the library (package) `MASS` provides the `polr` function which estimates the ordered probit and logit model. Although we can use the `nlm` function when we define the log-likelihood function, we do not report this method.

```r
library(MASS)

model <- factor(Level) ~ lnPrice + Top25
oprobit <- polr(model, data = train_dt, method = "probit")
ologit <- polr(model, data = train_dt, method = "logistic")

a_oprobit <- round(oprobit$zeta, 3)
a_ologit <- round(ologit$zeta, 3)
```

1.3 Interepretation and Model Fitness

Table 1 shows results. In both models, the latent variable y_i^* is increasing in `Top25`. This means that high-earners have higer value of latent variable y_i^*. Since the cutoff values are increasing in the observed y_i, we can conclude that high-earners are more likely to live on the upper floor.

To evaluate model fitness, we use the *percent correctly predicted*, which is the percentage of unit whose predicted y_i matches the actual y_i. First, we calculate $\mathbf{x}_i\hat{\beta}$. If this value is in $(-\infty, \hat{a}_1]$, $(\hat{a}_1, a_2]$, $(\hat{a}_2, \hat{a}_3]$, and $(\hat{a}_3, +\infty)$, then we take $\hat{y}_i = 1$, $\hat{y}_i = 2$, $\hat{y}_i = 3$ and $\hat{y}_i = 4$, respectively. Using the training data (in-sample) and the test data (out-of-sample), we calculate this index.

```r
library(tidyverse) #use case_when()
# coefficients
bp <- matrix(coef(oprobit), nrow = 2); bl <- matrix(coef(ologit), nrow = 2)
# cutoff value
ap <- oprobit$zeta; al <- ologit$zeta
# in-sample prediction
indt <- as.matrix(train_dt[,c("lnPrice", "Top25")])
in_xbp <- indt %*% bp; in_xbl <- indt %*% bl

in_hatYp <- case_when(
  in_xbp <= ap[1] ~ 1,
  in_xbp <= ap[2] ~ 2,
  in_xbp <= ap[3] ~ 3,
  TRUE ~ 4
)

in_hatYl <- case_when(
  in_xbl <= al[1] ~ 1,
  in_xbl <= al[2] ~ 2,
  in_xbl <= al[3] ~ 3,
  TRUE ~ 4
)
```

```r
inpred_p <- round(sum(train_dt$Level == in_hatYp)/nrow(train_dt), 3)
inpred_l <- round(sum(train_dt$Level == in_hatYl)/nrow(train_dt), 3)

# out-of-sample prediction
outdt <- as.matrix(test_dt[,c("lnPrice", "Top25")])
out_xbp <- outdt %*% bp; out_xbl <- outdt %*% bl

out_hatYp <- case_when(
  out_xbp <= ap[1] ~ 1,
  out_xbp <= ap[2] ~ 2,
  out_xbp <= ap[3] ~ 3,
  TRUE ~ 4
)

out_hatYl <- case_when(
  out_xbl <= al[1] ~ 1,
  out_xbl <= al[2] ~ 2,
  out_xbl <= al[3] ~ 3,
  TRUE ~ 4
)

outpred_p <- round(sum(test_dt$Level == out_hatYp)/nrow(test_dt), 3)
outpred_l <- round(sum(test_dt$Level == out_hatYl)/nrow(test_dt), 3)
```

As a result, the percent correctly predicted is almost 16% when we use the in-sample data. When we use the test data, this index slightly increases. Overall, out model seems not to be good because the percent correctly predicted is low.

```r
library(stargazer)
stargazer(
  oprobit, ologit,
  report = "vcs", keep.stat = c("n"),
  omit = c("Constant"),
  add.lines = list(
    c("Cutoff value at 1|2", a_oprobit[1], a_ologit[1]),
    c("Cutoff value at 2|3", a_oprobit[2], a_ologit[2]),
    c("Cutoff value at 3|4", a_oprobit[3], a_ologit[3]),
    c("Percent correctly predicted (in-sample)", inpred_p, inpred_l),
    c("Percent correctly predicted (out-of-sample)", outpred_p, outpred_l)
  ),
  omit.table.layout = "n", table.placement = "t",
  title = "Floor Level of House: Ordered Probit and Logit Model",
  label = "housing",
  type = "latex", header = FALSE
)
```

Table 1: Floor Level of House: Ordered Probit and Logit Model

	Dependent variable:	
	Level	
	ordered probit	*ordered logistic*
	(1)	(2)
lnPrice	−0.007	−0.013
	(0.019)	(0.031)
Top25	0.133	0.202
	(0.080)	(0.132)
Cutoff value at 1|2	-0.371	-0.611
Cutoff value at 2|3	0.02	0.014
Cutoff value at 3|4	0.719	1.163
Percent correctly predicted (in-sample)	0.161	0.161
Percent correctly predicted (out-of-sample)	0.175	0.175
Observations	1,074	1,074

2 Empirical Application of Multinomial Model: Gender Discremination in Job Position

2.1 Background and Data

Recently, many developed countries move toward women's social advancement, for example, an increase of number of board member. In this application, we explore whether the gender discremination existed in the U.S. bank industry. Our hypothesis is that women are less likely to be given a higher position than male.

We use a built-in dataset called `BankWages` in the library `AER`. This datase contains the following variables:

- `job`: three job position. The rank of position is `custodial < admin < manage`.
- `education`: years of education
- `gender`: a dummy variable of female

Again, we split data into two subsets: the *training* data and the *test* data. The training data, which is used for estimation and model fitness, is randoly drawn from the original data. The sample size of this subset is two thirds of total observations of the original one ($N = 316$). The test data, which is used for model prediction, consists of observations which the training data does not include ($N = 158$).

To use the multinomial logit model in `R`, we need to transform outcome variable into the form `factor`, which is special variable form in `R`. The variable form `factor` is similar to dummy variables. For example, `factor(dt$job, levels = c("admin", "custodial", "manage"))` transforms the variable form `job` from the form `character` into the form `factor`. Moreover, when we use `job` as explanatory variables, `R` automatically makes two dummy variables of `custodial` and `manage`.

```
library(AER)
data(BankWages)
dt <- BankWages
dt$job <- as.character(dt$job)
dt$job <- factor(dt$job, levels = c("admin", "custodial", "manage"))
dt <- dt[,c("job", "education", "gender")]

set.seed(120511)
train_id <- sample(1:nrow(dt), size = (2/3)*nrow(dt), replace = FALSE)
train_dt <- dt[train_id,]; test_dt <- dt[-train_id,]

head(train_dt)
```

```
##        job education gender
## 75   admin        15 female
## 2    admin        16   male
## 374  admin        15   male
## 405  admin        12 female
## 67  manage        16   male
## 92   admin         8 female
```

2.2 Model

The outcome variable y_i takes three values $\{0, 1, 2\}$. Note that there is no meaning in order. Then, the multinomial logit model has the following response probabilities

$$P_{ij} = \mathbb{P}(y_i = j|\mathbf{x}_i) = \begin{cases} \frac{\exp(\mathbf{x}_i\beta_j)}{1+\sum_{k=1}^{2}\exp(\mathbf{x}_i\beta_k)} & \text{if} \quad j = 1, 2 \\ \frac{1}{1+\sum_{k=1}^{2}\exp(\mathbf{x}_i\beta_k)} & \text{if} \quad j = 0 \end{cases}.$$

The log-likelihood function is

$$M_n(\beta_1, \beta_2) = \sum_{i=1}^{n}\sum_{j=0}^{3} d_{ij}\log(P_{ij}),$$

where d_{ij} is a dummy variable taking 1 if $y_i = j$.

In `R`, some packages provide the multinomial logit model. In this application, we use the `multinom` function in the library `nnet`.

```
library(nnet)
est_mlogit <- multinom(job ~ education + gender, data = train_dt)
```

2.3 Interpretations and Model Fitness

Table 2 summarizes the result of multinomial logit model. The coefficient represents the change of $\log(P_{ij}/P_{i0})$ in corresponding covariate beucase the response probabilities yields

$$\frac{P_{ij}}{P_{i0}} = \exp(\mathbf{x}_i \beta_j) \Leftrightarrow \log\left(\frac{P_{ij}}{P_{i0}}\right) = \mathbf{x}_i \beta_j.$$

For example, eduction decreases the log-odds between `custodial` and `admin` by -0.562. This implies that those who received higher education are more likely to obtain the position `admin`. Highly-educated workers are also more likely to obtain the position `manage`. Moreover, a female dummy decrease the log-odds between `manage` and `admin` by -0.748, which implies that females are less likely to obtain higher position `manage`. From this result, we conclude that the U.S. bank disencouraged females to assign higher job position.

To evalue model fitness and prediction, we use two indices: the *pseudo R-squared* and *percent correctly predicted*. The *preudo R-sqaured* is calculated by $1 - L_1/L_0$ where L_1 is the value of log-likelihood for estimated model and L_0 is the value of log-likelihood in the model with only an intercept. `R` snippet for calculation of pseudo R-sqaured is as follows: Note that `nnet:::logLik.multinom()` returns the value of log-likelihood.

```
loglik1 <- as.numeric(nnet:::logLik.multinom(est_mlogit))
est_mlogit0 <- multinom(job ~ 1, data = train_dt)
loglik0 <- as.numeric(nnet:::logLik.multinom(est_mlogit0))
pr2 <- round(1 - loglik1/loglik0, 3)
```

The second index is the *precent correctly predicted*. The predicted outcome is the outcome with the highest estimated probability. Using the training data (in-sample) and the test data (out-of-sample), we calculate this index. `R` snippet for calculation of this index is as follows.

```
# in-sample prediction
inpred <- predict(est_mlogit, newdata = train_dt, "probs")
inpred <- colnames(inpred)[apply(inpred, 1, which.max)]
inpcp <- round(sum(inpred == train_dt$job)/length(inpred), 3)
# out-of-sample prediction
outpred <- predict(est_mlogit, newdata = test_dt, "probs")
outpred <- colnames(outpred)[apply(outpred, 1, which.max)]
outpcp <- round(sum(outpred == test_dt$job)/length(outpred), 3)
```

As a result, our model is good in terms of fitness and prediction because the percent correctly predicted is high (83.9% of in-sample data and 88.0% of out-of-sample data), and the pseudo R-sqaured is 0.523.

Table 2: Multinomial Logit Model of Job Position

	Dependent variable:	
	custodial	manage
	(1)	(2)
Education	−0.547	1.322
	(0.116)	(0.229)
Female = 1	−10.507	−0.891
	(31.352)	(0.524)
Constant	4.634	−21.448
	(1.269)	(3.605)
Observations	948	
Percent correctly predicted (in-sample)	0.839	
Percent correctly predicted (out-of-sample)	0.88	
Log-likelihood	-102.964	
Pseudo R-sq	0.523	

```r
stargazer(
  est_mlogit,
  covariate.labels = c("Education", "Female = 1"),
  report = "vcs", omit.stat = c("aic"),
  add.lines = list(
    c("Observations", n, ""),
    c("Percent correctly predicted (in-sample)", inpcp, ""),
    c("Percent correctly predicted (out-of-sample)", outpcp, ""),
    c("Log-likelihood", round(loglik1, 3), ""),
    c("Pseudo R-sq", pr2, "")
  ),
  omit.table.layout = "n", table.placement = "t",
  title = "Multinomial Logit Model of Job Position",
  label = "job",
  type = "latex", header = FALSE
)
```