Ordinal Dependent Variables

Week 18

Winnie Xia

Ordered logit and probit regression

- If outcome or dependent variable is binary, then use **logit or probit models**.
- If outcome or dependent variable is categorical but ordered (i.e low to high), then use **ordered logit or ordered probit models**
    - Agree; Neutral; Disagree
    - Low; Middle; High
- If outcome or dependent variable is categorical without any particular order, then use **multinomial logit**.

Ordered logit and probit in R

```r
library(foreign)
dat <- read.dta("https://stats.idre.ucla.edu/stat/data/ologit.dta")
head(dat)
```

```
##             apply pared public  gpa
## 1     very likely     0      0 3.26
## 2 somewhat likely     1      0 3.21
## 3        unlikely     1      1 3.94
## 4 somewhat likely     0      0 2.81
## 5 somewhat likely     0      0 2.53
## 6        unlikely     0      1 2.59
```

Ordered logit and probit in R

Below we use the `polr` function from the MASS package to estimate an ordered logistic regression model.

```
library(MASS)
library(texreg)
m1 <- polr(apply ~ pared + public + gpa, data = dat,
           method = "logistic", Hess=TRUE)

m2 <- polr(apply ~ pared + public + gpa, data = dat,
           method = "probit", Hess=TRUE)

m3 <- glm(as.numeric(apply) ~ pared + public + gpa,
          data = dat)
```

Ordered logit and probit in R

```
##
## =====================================================================
##                                 Model 1      Model 2      Model 3
## ---------------------------------------------------------------------
## pared                              1.05 ***     0.60 ***     0.36 ***
##                                   (0.27)       (0.16)       (0.09)
## public                            -0.06         0.01         0.02
##                                   (0.30)       (0.17)       (0.10)
## gpa                                0.62 *       0.36 *       0.19 *
##                                   (0.26)       (0.16)       (0.09)
## unlikely|somewhat likely           2.20 **      1.30 **
##                                   (0.78)       (0.47)
## somewhat likely|very likely        4.30 ***     2.50 ***
##                                   (0.80)       (0.48)
## (Intercept)                                                  0.91 ***
##                                                             (0.25)
## ---------------------------------------------------------------------
## AIC                              727.02       727.50       798.34
## BIC                              746.98       747.45       818.29
## Log Likelihood                  -358.51      -358.75      -394.17
## Deviance                         717.02       717.50       168.08
## Num. obs.                        400          400          400
## =====================================================================
```

Interpretation

```r
summary(m1)
```

```
## Call:
## polr(formula = apply ~ pared + public + gpa, data = dat, Hess = TRUE,
##     method = "logistic")
##
## Coefficients:
##           Value Std. Error t value
## pared   1.04769     0.2658  3.9418
## public -0.05879     0.2979 -0.1974
## gpa     0.61594     0.2606  2.3632
##
## Intercepts:
##                             Value   Std. Error t value
## unlikely|somewhat likely     2.2039  0.7795     2.8272
## somewhat likely|very likely  4.2994  0.8043     5.3453
##
## Residual Deviance: 717.0249
## AIC: 727.0249
```

Interpretation

We often interpret models by calculating marginal effects for an ordered choice model and their standard errors.

```r
library(erer)
marg <- ocME(m1)
marg
```

```
##        effect.unlikely effect.somewhat likely
## pared           -0.255                  0.137
## public           0.015                 -0.010
## gpa             -0.152                  0.101
##        effect.very likely
## pared               0.117
## public             -0.005
## gpa                 0.051
```

Interpretation

```r
marg$out # get t and p-values
```

```
## $ME.unlikely
##         effect error t.value p.value
## pared   -0.255 0.060  -4.237   0.000
## public   0.015 0.073   0.198   0.843
## gpa     -0.152 0.064  -2.364   0.019
##
## $`ME.somewhat likely`
##         effect error t.value p.value
## pared    0.137 0.029   4.820   0.000
## public  -0.010 0.049  -0.196   0.844
## gpa      0.101 0.044   2.305   0.022
##
## $`ME.very likely`
##         effect error t.value p.value
## pared    0.117 0.039   3.025   0.003
## public  -0.005 0.024  -0.201   0.841
## gpa      0.051 0.022   2.301   0.022
##
## $ME.all
##         effect.unlikely effect.somewhat likely
```

The Parallel Regression Assumption

- An alternative way of conceptualizing the ordered logit/probit model is as the constrained estimation of a system of logit/probit models.
- Parallel regressions describe how the impact of covariates may shift the predicted probability curves to the right ot left, but **does not change the basic slope of these curves across any two categories.**

Assessing the parallel regression assumption

- Fit separate binary logit or probit models for the different cut-off levels and see if the slopes are similar.

- The Brant test

```r
library(brant)
brant(m1)
```

```
## --------------------------------------------
## Test for     X2     df     probability
## --------------------------------------------
## Omnibus        4.34    3    0.23
## pared        0.13    1    0.72
## public        3.44    1    0.06
## gpa        0.18    1    0.67
## --------------------------------------------
##
## H0: Parallel Regression Assumption holds
```

Cumulative Link Model

```
library(ordinal)
m4 <- clm(apply ~ pared + public + gpa, data = dat,
           link = "logit")
```

Cumulative Link Model

```
summary(m4)
```

```
## formula: apply ~ pared + public + gpa
## data:    dat
##
##  link  threshold nobs logLik  AIC    niter max.grad cond.H
##  logit flexible  400  -358.51 727.02 5(0)  1.63e-10 1.3e+03
##
## Coefficients:
##        Estimate Std. Error z value Pr(>|z|)
## pared   1.04766    0.26579   3.942 8.09e-05 ***
## public -0.05868    0.29786  -0.197   0.8438
## gpa     0.61575    0.26063   2.363   0.0182 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                              Estimate Std. Error z value
## unlikely|somewhat likely       2.2033     0.7795   2.826
## somewhat likely|very likely    4.2988     0.8043   5.345
```