Fit-regression

S. Sethi

2018-04-16

Choice of method: 1. Multinomial Logistic Regression is the linear regression analysis to conduct when the dependent variable is nominal with more than two levels

Clean data

```
source("data_wrangling.R")

dat <- t2dsci_data_wrangling("../data/Assessment_Latest.csv")
```

Doing the Chi-squared test

Getting the table:

```
tab <- with(dat, table(dat$X, dat$Y))
tab
```

```
##
##           Easy Somewhat easy Medium Somewhat hard Hard
##   stats      1             0      4             2    2
##   prog       0             1      4             1    0
##   both       8             3      3             3    1
##   neither    2             1      3             3    3
```

```
chisq.test(tab, correct = TRUE)
```

```
## Warning in chisq.test(tab, correct = TRUE): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab
## X-squared = 15.098, df = 12, p-value = 0.2361
```

Thus, the chi-square test does not give us any relevant information about our model.

Ordered Logistic Regression

Now, I will try to fit an ordinal regression model on our data as our Y variable is of the form 1-5.

```
require(foreign)
require(ggplot2)
require(MASS)
require(Hmisc)
require(reshape2)
```

Starting with desciptive statitics:

```
lapply(dat[, c("X", "Y", "C1", "C2", "C3")], table)
```

```
## $X
##
##   stats    prog    both neither
##       9       6      18      12
##
## $Y
##
##          Easy Somewhat easy        Medium Somewhat hard          Hard
##            11              5            14             9             6
##
## $C1
##
##  <1 1-3 3-5  +5
##  15  19   8   3
##
## $C2
##
## FALSE  TRUE
##    21    24
##
## $C3
##
## FALSE  TRUE
##    30    15
```

```
# Three way cross tabs and flattening the table
# With confounding variable 1
ftable(xtabs(~X + C1 + Y, data = dat))
```

```
##             Y Easy Somewhat easy Medium Somewhat hard Hard
## X       C1
## stats   <1       0             0      0             1    2
##         1-3      1             0      2             1    0
##         3-5      0             0      1             0    0
##         +5       0             0      1             0    0
## prog    <1       0             0      0             0    0
##         1-3      0             1      2             1    0
##         3-5      0             0      2             0    0
##         +5       0             0      0             0    0
## both    <1       3             1      1             1    1
##         1-3      4             1      1             1    0
##         3-5      1             1      1             0    0
##         +5       0             0      0             1    0
## neither <1       1             1      1             1    1
##         1-3      0             0      2             1    1
##         3-5      1             0      0             1    0
##         +5       0             0      0             0    1
```

```
# With confounding variable 2
ftable(xtabs(~X + C2 + Y, data = dat))
```

```
##                Y Easy Somewhat easy Medium Somewhat hard Hard
## X       C2
## stats   FALSE      0             0      1             1    1
##         TRUE       1             0      3             1    1
## prog    FALSE      0             0      1             1    0
##         TRUE       0             1      3             0    0
## both    FALSE      4             1      0             3    0
##         TRUE       4             2      3             0    1
## neither FALSE      1             1      2             1    3
##         TRUE       1             0      1             2    0
```

```
# With confounding variable 3
ftable(xtabs(~X + C3 + Y, data = dat))
```

```
##                Y Easy Somewhat easy Medium Somewhat hard Hard
## X       C3
## stats   FALSE      0             0      2             2    1
##         TRUE       1             0      2             0    1
## prog    FALSE      0             1      1             1    0
##         TRUE       0             0      3             0    0
## both    FALSE      5             3      2             1    1
##         TRUE       3             0      1             2    0
## neither FALSE      1             1      3             3    2
##         TRUE       1             0      0             0    1
```

```
o <- polr(Y ~ X, data = dat, Hess = TRUE)
summary(o)
```

```
## Call:
## polr(formula = Y ~ X, data = dat, Hess = TRUE)
##
## Coefficients:
##             Value Std. Error  t value
## Xprog    -0.69233     0.8752 -0.79110
## Xboth    -1.76302     0.7754 -2.27370
## Xneither -0.06364     0.7986 -0.07968
##
## Intercepts:
##                       Value   Std. Error t value
## Easy|Somewhat easy   -2.0901  0.6908    -3.0255
## Somewhat easy|Medium -1.4632  0.6503    -2.2500
## Medium|Somewhat hard  0.0101  0.6028     0.0167
## Somewhat hard|Hard    1.2699  0.6628     1.9161
##
## Residual Deviance: 130.9879
## AIC: 144.9879
```

Now, Will try to calcualte the p-values:

```
coeff_tab <- coef(summary(o))
```

```
# Calculate the p-values
p <- pnorm(abs(coeff_tab[, "t value"]), lower.tail = FALSE) * 2

# combining with the t-values and coeff table
coeff_tab <- cbind(coeff_tab, "p value" = p)
```

```
ci <- confint(o)
```

```
## Waiting for profiling to be done...
```

```
exp(coef(o))
```

```
##     Xprog     Xboth  Xneither
## 0.5004095 0.1715261 0.9383467
```

```
exp(cbind(OR = coef(o), ci))
```

```
##                 OR      2.5 %    97.5 %
## Xprog    0.5004095 0.08725280 2.7880892
## Xboth    0.1715261 0.03568698 0.7632081
## Xneither 0.9383467 0.19309009 4.5395511
```

For a one unit increase in a person having an experience in Programming,

```
df_pred <- data.frame(X = c("stats", "prog", "both", "neither"))
pred <- predict(o, newdata = df_pred, "probs")
```

```
# Probabilties
df_X <- data.frame(X = c("stats", "prog", "both", "neither"), each=45)
pp_X <- cbind(df_X, predict(o, newdata = df_X, "probs", se=TRUE))
by(pp_X[,3:7], pp_X$X, colMeans)
```

```
## pp_X$X: both
##          Easy Somewhat easy        Medium Somewhat hard          Hard
##    0.41894048    0.15546826    0.28043201    0.09920023    0.04595903
## ------------------------------------------------------------
## pp_X$X: neither
##          Easy Somewhat easy        Medium Somewhat hard          Hard
##    0.11644769    0.08144426    0.32052466    0.27301432    0.20856908
## ------------------------------------------------------------
## pp_X$X: prog
##          Easy Somewhat easy        Medium Somewhat hard          Hard
##     0.1981630     0.1181367     0.3524183     0.2080599     0.1232221
## ------------------------------------------------------------
## pp_X$X: stats
##          Easy Somewhat easy        Medium Somewhat hard          Hard
##    0.11005846    0.07792634    0.31453117    0.27821591    0.21926812
```