Homework Solutions: MNL model

Harsh Tandon

Due on March 2, 2020

First, load the data

```
#install.packages("mlogit") #it needs R version to be 3.5 or newer
#install.packages("data.table")
library(mlogit)
library(data.table)
yogurtdata = fread("yogurt_3brands.csv")

#change the names, so that the software can recognize which column is for which choice altern
ative
#The names of each choice alternative needs to be consistent with those in the Choice variabl
e

setnames(yogurtdata, c("Feature_S", "Feature_D", "Feature_Y", "HH Size", "Pan ID"),
                c("Feature.Stonyfield", "Feature.Dannon", "Feature.Yoplait",
                  "HHSize", "PanID"))
setnames(yogurtdata, c("Price_S", "Price_D", "Price_Y"),
                  c("Price.Stonyfield", "Price.Dannon", "Price.Yoplait"))
```

Second, set up the model

Please get ready the data for estimating the following MNL model, specified using the latent utility functions for each of the three brands

- Stonyfield $U_{is} = \beta_1 Price_s + \beta_2 Feature_s + \epsilon_{is}$
- Yoplait $U_{iy} = \beta_{0y} + \beta_1 Price_y + \beta_2 Feature_y + \beta_3 Income_i + \beta_4 HHsize_i + \epsilon_{iy}$
- Dannon $U_{id} = \beta_{0d} + \beta_1 Price_d + \beta_2 Feature_d + \beta_3 Income_i + \beta_4 HHsize_i + \epsilon_{id}$

You need to create an additional column in yogurtdata, called “Choice”, indicating the choices made by each person, and set this column to be a factor.

```
# Create a Choice variable that lists the choice made
yogurtdata[Stonyfield==1, Choice := "Stonyfield"]
yogurtdata[Yoplait==1, Choice := "Yoplait"]
yogurtdata[Dannon==1, Choice := "Dannon"]

yogurtdata[, Choice := as.factor(Choice)]

yogurtdata = yogurtdata[, -c("Stonyfield","Yoplait","Dannon")]
head(yogurtdata)
```

```
##    Index Feature.Stonyfield Feature.Yoplait Feature.Dannon
## 1:     1                  0               0              0
## 2:     2                  0               0              0
## 3:     3                  0               0              0
## 4:     4                  0               0              0
## 5:     5                  0               0              0
## 6:     6                  0               0              0
##    Price.Stonyfield Price.Yoplait Price.Dannon Income HHSize PanID  Choice
## 1:            0.108         0.081        0.061      9      2     1  Dannon
## 2:            0.108         0.098        0.064      9      2     1 Yoplait
## 3:            0.108         0.098        0.061      9      2     1 Yoplait
## 4:            0.108         0.098        0.061      9      2     1 Yoplait
## 5:            0.125         0.098        0.049      9      2     1 Yoplait
## 6:            0.108         0.092        0.050      9      2     1 Yoplait
```

Then you need to setup the data format that is understandable by the package, using `mlogit.data()`

```
yLogitData = mlogit.data(yogurtdata[,-c("Index" )], shape="wide",
                choice="Choice", id="PanID", varying=1:6)
head(yLogitData)
```

```
##      Income HHSize PanID Choice        alt Feature Price chid
## 1319      9      2     1   TRUE     Dannon       0 0.061    1
## 1         9      2     1  FALSE Stonyfield       0 0.108    1
## 660       9      2     1  FALSE    Yoplait       0 0.081    1
## 1320      9      2     1  FALSE     Dannon       0 0.064    2
## 2         9      2     1  FALSE Stonyfield       0 0.108    2
## 661       9      2     1   TRUE    Yoplait       0 0.098    2
```

Third, now estimate the model

The format for using `mFormula()` is the following

Choice ~ X different, beta same |X same, beta same |X different, beta different

```
featureFormula = mFormula(Choice ~ Feature + Price | Income + HHSize)

model1 = mlogit(featureFormula, yLogitData, reflevel = 'Stonyfield')
summary(model1)
```

```
##
## Call:
## mlogit(formula = Choice ~ Feature + Price | Income + HHSize,
##     data = yLogitData, reflevel = "Stonyfield", method = "nr")
##
## Frequencies of alternatives:
## Stonyfield     Dannon    Yoplait
##    0.33080    0.33687    0.33232
##
## nr method
## 4 iterations, 0h:0m:0s
## g'(-H)^-1g = 8.68E-08
## gradient close to zero
##
## Coefficients :
##                       Estimate Std. Error z-value  Pr(>|z|)
## Dannon:(intercept)   -1.572326   0.369253 -4.2581 2.061e-05 ***
## Yoplait:(intercept)   1.276613   0.287864  4.4348 9.217e-06 ***
## Feature               0.371186   0.206549  1.7971 0.0723230 .
## Price               -23.480763   3.667916 -6.4017 1.537e-10 ***
## Dannon:Income         0.125584   0.030431  4.1268 3.678e-05 ***
## Yoplait:Income       -0.092926   0.028349 -3.2779 0.0010458 **
## Dannon:HHSize        -0.265701   0.116981 -2.2713 0.0231280 *
## Yoplait:HHSize       -0.362255   0.107934 -3.3563 0.0007901 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -632.78
## McFadden R^2:  0.12595
## Likelihood ratio test : chisq = 182.36 (p.value = < 2.22e-16)
```

> Q: Please interpret the model estimation results.

- The intercept for Yoplait is positive and for Dannon is negative, indicating that everything else being equal, Dannon is the least preferred brand.
- The `Feature` parameter for all brands are the same, and it is positive and statistically significant.
- The `Price` parameter for all brands are the same, and it is negative and statistically significant.
- The `Income` parameter for Yoplait is negative and for Dannon is positive, meaning holding everything else the same, the families with higher income tend to prefer Dannon; with not slightly higher income tend to prefer Stonyfield.
- The `HHsize` parameter for both Yoplait and Dannon is negative, meaning holding everything else constant, the larger families tend to prefer Stonyfield over Dannon and Yoplait; with families size not so large, they tend to prefer Dannon over Yoplait.

Change the model

In the above model, all brands are constrained to have the same price parameter. Re-estimate the above model, but instead allow the price parameter to be brand specific, that is different across brands.

```
featureFormula1 = mFormula(Choice ~ Feature | Income + HHSize | Price)

model2 = mlogit(featureFormula1, yLogitData, reflevel = 'Stonyfield')
summary(model2)
```

```
##
## Call:
## mlogit(formula = Choice ~ Feature | Income + HHSize | Price,
##     data = yLogitData, reflevel = "Stonyfield", method = "nr")
##
## Frequencies of alternatives:
## Stonyfield     Dannon    Yoplait
##    0.33080    0.33687    0.33232
##
## nr method
## 4 iterations, 0h:0m:0s
## g'(-H)^-1g = 2.79E-07
## gradient close to zero
##
## Coefficients :
##                       Estimate Std. Error z-value  Pr(>|z|)
## Dannon:(intercept)   -0.359542   0.765243 -0.4698 0.6384689
## Yoplait:(intercept)   1.348014   0.931055  1.4478 0.1476629
## Feature               0.325126   0.211679  1.5359 0.1245541
## Dannon:Income         0.139728   0.031674  4.4114 1.027e-05 ***
## Yoplait:Income       -0.090071   0.028565 -3.1532 0.0016147 **
## Dannon:HHSize        -0.293239   0.118332 -2.4781 0.0132082 *
## Yoplait:HHSize       -0.367275   0.108078 -3.3982 0.0006782 ***
## Stonyfield:Price    -21.209482   4.115792 -5.1532 2.561e-07 ***
## Dannon:Price        -42.594788  11.100031 -3.8374 0.0001244 ***
## Yoplait:Price       -21.622927   9.535203 -2.2677 0.0233478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -631.07
## McFadden R^2:  0.12831
## Likelihood ratio test : chisq = 185.79 (p.value = < 2.22e-16)
```

```
prob = predict(model1,yLogitData)
probnew=predict(model2,yLogitData)
colMeans(prob)
```

```
## Stonyfield     Dannon    Yoplait
##  0.3308042  0.3368741  0.3323217
```

```
colMeans(probnew)
```

```
## Stonyfield     Dannon    Yoplait
##  0.3308042  0.3368741  0.3323217
```

> Q: Compare the above two models

- First model, constrained the price parameters to be the same across brands
- Second model, allow the price parameters to be different across brands

in the following: - First, based on the price parameters, do you think it makes sense to constrain them to be the same across the three brands? - Second, compare the model fits, using the AIC values that we learned before

$$AIC = -2LogLikelihood + 2K$$

K is the number of model parameters.

Constraining coefficient of Price to be same across the three brands would mean that, if price changes by one unit, the impact of this change on all three brands would be the same. However, we want to measure the impact of price on each brand individually, thus constraining them to be the same across all three bands does not make sense.

Model1 constraints the coefficient of Price to be same across the three brands, however, Model2 allows price coefficients to vary individually with the brand. Model2 could give a deeper insight into how one's choice would be affected if the price of one brand changes (say Yoplait gives discount), but price of other brands remain the same.

```
AIC(model1,model2)
```

```
##        df      AIC
## model1  8 1281.567
## model2 10 1282.146
```

Comparing models using AIC: **We see that, although logically we want Model2 to be our model, the AIC score of Model1 is lower than AIC score of Model2. The AIC scores suggest that we should go with Model1, however, AIC is just a statistical tool to corroborate with our intuition. It is possible that with more data at our disposal, AIC score of Model2 could be less than Model1. Thus, a more concrete statistical analysis could be presented if more data was present. So, in this scenario, statistically, we would go with Model1, intuitively we would go with Model2.**