Session: Choice-Based Conjoint

Contents

Content with * is optional.

You will need the following packages for this session:

```
library("xtable") # processing of regression output
library("knitr") # used for report compilation and table display
library("ggplot2") # very popular plotting library ggplot2
library("mlogit") # multinomial logit
```

```
## Loading required package: dfidx
```

```
##
## Attaching package: 'dfidx'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
library("caret") # ConfusionMatrix
```

```
## Loading required package: lattice
```

Suppose a company is developing a new line of tablets and is trying to determine how large the tablet should be and what type of storage and RAM it should have.

To inform this decision, it would be helpful to understand how customers value those different features.

- Do customers like or dislike big screen sizes?
- If they like them, how much more would they be willing to pay for a bigger size screen?
- Are there segments of customers who like big-size screens more than other customers?

1 Choice-Based Conjoint Analysis Data

```
cbc.df<-read.csv("Data_Conjoint_Choice.csv", stringsAsFactors = TRUE)
str(cbc.df)
```

```
## 'data.frame':    6165 obs. of  10 variables:
##  $ ConsumerId        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ChoiceSetId       : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ AlternativeIdInSet: int  1 2 3 1 2 3 1 2 3 1 ...
##  $ Choice            : int  1 0 0 1 0 0 0 0 1 1 ...
##  $ Brand             : Factor w/ 5 levels "Galaxy","iPad",..: 2 5 3 2 5 4 5 3 1 2 ...
##  $ Size              : Factor w/ 4 levels "sz10inch","sz7inch",..: 2 1 4 3 1 2 3 4 1 4 ...
##  $ Storage           : Factor w/ 4 levels "st128gb","st16gb",..: 3 4 2 3 1 4 4 2 1 2 ...
##  $ Ram               : Factor w/ 3 levels "r1gb","r2gb",..: 3 2 2 1 3 1 1 3 2 3 ...
##  $ Battery           : Factor w/ 3 levels "b7h","b8h","b9h": 1 3 2 2 1 3 1 3 3 1 ...
##  $ Price             : int  499 399 499 399 299 199 199 399 499 299 ...
```

```
head(cbc.df)
```

```
##   ConsumerId ChoiceSetId AlternativeIdInSet Choice   Brand     Size Storage
## 1          1           1                  1      1    iPad  sz7inch  st32gb
## 2          1           1                  2      0 Surface sz10inch  st64gb
## 3          1           1                  3      0  Kindle  sz9inch  st16gb
## 4          1           2                  1      1    iPad  sz8inch  st32gb
## 5          1           2                  2      0 Surface sz10inch st128gb
## 6          1           2                  3      0   Nexus  sz7inch  st64gb
##    Ram Battery Price
## 1 r4gb     b7h   499
## 2 r2gb     b9h   399
## 3 r2gb     b8h   499
## 4 r1gb     b8h   399
## 5 r4gb     b7h   299
## 6 r1gb     b9h   199
```

- The first three rows in *cbc.df* describe the first question that was asked of respondent 1. The *choice* column shows that this respondent chose the first alternative.
- *ConsumerId* indicates which respondent answered this question.
- *ChoiceSetId* indicates that these first three rows were the profiles in the first question. *ChoiceSetId* numbered 1:15 is for respondent 1, then 15:30 for respondent 2, and so forth.
- *AlternativeIdInSet* indicates that the first row was alternative 1, the second was alternative 2, and the third was alternative 3.
- *Choice* indicates which alternative the respondent chose; it takes the value of 1 for the profile in each choice question that was indicated as the preferred alternative.

It is now important to estimate a complete choice model. The first step for any modelling is to get an understanding of the data using basic descriptives. We start with *summary*

```
summary(cbc.df)
```

```
##    ConsumerId   ChoiceSetId   AlternativeIdInSet     Choice
##  Min.   :  1   Min.   :   1   Min.   :1          Min.   :0.0000
##  1st Qu.: 35   1st Qu.: 514   1st Qu.:1          1st Qu.:0.0000
##  Median : 69   Median :1028   Median :2          Median :0.0000
##  Mean   : 69   Mean   :1028   Mean   :2          Mean   :0.3333
##  3rd Qu.:103   3rd Qu.:1542   3rd Qu.:3          3rd Qu.:1.0000
##  Max.   :137   Max.   :2055   Max.   :3          Max.   :1.0000
##      Brand            Size         Storage       Ram       Battery
##  Galaxy :1263   sz10inch:1371   st128gb:1376   r1gb:2192   b7h:1918
##  iPad   :1538   sz7inch :1767   st16gb :1370   r2gb:2055   b8h:2055
##  Kindle :1119   sz8inch :1520   st32gb :1774   r4gb:1918   b9h:2192
##  Nexus  :1104   sz9inch :1507   st64gb :1645
##  Surface:1141
##
##      Price
##  Min.   :169.0
##  1st Qu.:199.0
##  Median :299.0
##  Mean   :307.5
##  3rd Qu.:399.0
##  Max.   :499.0
```

We see how many times each attribute level appeared in the questions. A more informative way to summarize choice data is to compute choice *counts*, which are cross tabs on the number of times respondents chose an alternative at each feature level. We can do this easily using *xtabs()*.

```
xtabs(Choice~Price, data=cbc.df)
```

```
## Price
## 169 199 299 399 499
## 688 472 329 365 201
```

Respondents chose a tablet at the £169 price point much more often than they chose tablets priced at £299 or £499.

If we compute counts for the *Size* attribute, we find that the choices were more balanced among 7inch, 8 inch, and 10inch, and more choices on 9inch.

```
xtabs(Choice~Size, data=cbc.df)
```

```
## Size
## sz10inch  sz7inch  sz8inch  sz9inch
##      475      535      472      573
```

It is always encouraged to compute choice counts for each attribute before estimating a choice model.

2 Prepare the data

We can now estimate our first choice model. By fitting a choice model, we can get a precise measurement of how much each attribute is associated with respondents' choices.

We use the *mlogit* package, which you may need to install with *install.packages()*. *mlogit* estimates the most basic and commonly used choice model, the *multinominal logit model*.

2.1 Define reference levels - used when estimating model

```
cbc.df$Brand <- relevel(cbc.df$Brand, ref = "Nexus")
cbc.df$Size <- relevel(cbc.df$Size, ref = "sz7inch")
cbc.df$Storage <- relevel(cbc.df$Storage, ref = "st16gb")
cbc.df$Ram <- relevel(cbc.df$Ram, ref = "r1gb")
cbc.df$Battery <- relevel(cbc.df$Battery, ref = "b7h")
```

2.2 Define data format

mlogit requires the choice data to be in a specific data format. We use *dfidx* function from the *dfidx* package to shape the format.

- The *choice* parameter indicates which columns contain the response data. In our case, *choice = "Choice"*.

- The *idx* parameter indicates how alternatives are structured. The first index in the list indicates the columns of Choice Sets and Consumers, the second index indicates the column of alternative in each choice set.

```
library(dfidx)  #install if needed
cbc.mlogit <- dfidx(cbc.df, choice="Choice",
                    idx=list(c("ChoiceSetId", "ConsumerId"), "AlternativeIdInSet"))
```

3 Multinomial conjoint model estimation with *mlogit()*

When we run the model, it selects the *reference* level for each *discrete* attribute. The utility of the reference level is normalized to zero. We specified a reference level for each discrete attribute at the data-loading stage. These reference levels are Nexus, 7" screen, 16GB HD, 1GB RAM, 7-hour battery. We treat price as a continuous variable, so we do not need to specify a reference level.

The model assumes the utility of alternative j without an error term is expressed as follows.

$$\begin{aligned} V_j = &\beta_{11}\,[\text{Brand=Galaxy}] + \beta_{12}\,[\text{Brand=iPad}] + \beta_{13}\,[\text{Brand=Kindle}] + \beta_{14}\,[\text{Brand=Surface}] + \\ &\beta_{21}\,[\text{Screen=10inch}] + \beta_{22}\,[\text{Screen=9inch}] + \beta_{23}\,[\text{Screen=8inch}] + \\ &\beta_{31}\,[\text{Storage=128gb}] + \beta_{32}\,[\text{Storage=64gb}] + \beta_{33}\,[\text{Storage=32gb}] + \\ &\beta_{41}\,[\text{RAM=4gb}] + \beta_{42}\,[\text{RAM=2gb}] + \\ &\beta_{51}\,[\text{Battery=9h}] + \beta_{52}\,[\text{Battery=8h}] + \\ &\beta_{6}\text{Price} \end{aligned}$$

where $U_j = V_j + \text{error}$. That is, there are 15 parameters β to estimate.

Assuming independent extreme value error distribution, consumer chooses alternative j from the choice set of three alternatives with probability

$$p_j = \frac{\exp(V_j)}{\exp(V_1) + \exp(V_2) + \exp(V_3)}, \quad j \in \{1, 2, 3\}$$

Clearly, $p_1 + p_2 + p_3 = 1$.

We actually estimate the model.

```
model<-mlogit(Choice ~ 0+Brand+Size+Storage+Ram+Battery+Price, data=cbc.mlogit)
kable(summary(model)$CoefTable)
```

	Estimate	Std. Error	z-value	Pr(>|z|)
BrandGalaxy	0.3378857	0.0925056	3.652596	0.0002596
BrandiPad	0.9780287	0.0937336	10.434136	0.0000000
BrandKindle	0.2630105	0.0996254	2.639995	0.0082907
BrandSurface	0.1450365	0.0938521	1.545373	0.1222560
Sizesz10inch	0.3240632	0.0841953	3.848949	0.0001186
Sizesz8inch	0.1890775	0.0829232	2.280151	0.0225987
Sizesz9inch	0.4355415	0.0808408	5.387644	0.0000001
Storagest128gb	0.5897703	0.0870533	6.774822	0.0000000
Storagest32gb	0.2168719	0.0829213	2.615395	0.0089124

	Estimate	Std. Error	z-value	Pr(>|z|)
Storagest64gb	0.5782183	0.0808259	7.153877	0.0000000
Ramr2gb	0.3189348	0.0672579	4.741970	0.0000021
Ramr4gb	0.6357438	0.0645225	9.853053	0.0000000
Batteryb8h	0.1299599	0.0651501	1.994777	0.0460672
Batteryb9h	0.1253824	0.0650588	1.927216	0.0539528
Price	-0.0050888	0.0002752	-18.488626	0.0000000

3.1 Meaning of parameters

After estimation, we obtain a coefficient estimate for each level (except the reference one) of every discrete attribute. Such a coefficient captures the relative utility or *partworth* of the level of attribute compared to the reference. For example, in case of the brand attribute, *BrandiPad* coefficient gives us an estimate of iPad's brand relative utility compared to Nexus (reference brand).

The positive sign tells us that, on average, customers prefer iPad over Nexus because larger estimates indicate stronger preferences, so we can see that customers strongly like 64 GB storage (relative to the base level, which is 16 GB). These parameter estimates are on the logit scale.

In case of the continuous price, we get a single coefficient, which captures how the utility of the alternative changes when the price goes up by one unit ($1), holding all other characteristics of the alternative fixed.

3.2 Model fit

One may wonder if the preference is only driven by the brand effect. We can estimate a model with only the brand as a predictor.

```
model.constraint <-mlogit(Choice ~ 0+Brand, data = cbc.mlogit)
```

Then we can use *lrtest* to compare the two models.

```
lrtest(model, model.constraint)
```

```
## Likelihood ratio test
##
## Model 1: Choice ~ 0 + Brand + Size + Storage + Ram + Battery + Price
## Model 2: Choice ~ 0 + Brand
##   #Df  LogLik  Df  Chisq Pr(>Chisq)
## 1  15 -1938.9
## 2   4 -2218.0 -11 558.29  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see the p-value (*Pr(>Chisq)*) is smaller than 0.05, which indicates that the two models are significantly different in fitting the data. The larger model (our first model) fits the data better. So, we should keep all the variables.

4 Interpreting Conjoint Analysis Findings

It is normally difficult to interpret the choice model partworth estimates directly. The coefficients are on an unfamiliar scale (i.e., log), and they measure relative preference for the levels, which can make them difficult to understand. So, instead of presenting the coefficients, most choice modellers prefer to focus on using the model to make *choice share predictions* or compute *willingness-to-pay* for each attribute.

4.1 Predicted Market Share

We can also use the estimated parameters to predict the probabilities of the choice for different alternatives in the data. Here, we print the prediction for the first six choice sets in the data.

```
kable(head(predict(model,cbc.mlogit)))
```

1	2	3
0.3717263	0.4405521	0.1877216
0.2367797	0.4718620	0.2913583
0.4760867	0.2974319	0.2264814
0.3730366	0.4456505	0.1813129
0.3984632	0.1618560	0.4396807
0.3791075	0.2506170	0.3702755

And now, we can measure the accuracy of prediction across all data.

```
predicted_alternative <- apply(predict(model,cbc.mlogit),1,which.max)
selected_alternative <- cbc.mlogit$AlternativeIdInSet[cbc.mlogit$Choice>0]
confusionMatrix(table(predicted_alternative,selected_alternative),positive = "1")
```

```
## Confusion Matrix and Statistics
##
##                      selected_alternative
## predicted_alternative   1   2   3
##                     1 362 158 130
##                     2 164 449 149
##                     3 136 160 347
##
## Overall Statistics
##
##                Accuracy : 0.5635
##                  95% CI : (0.5417, 0.5851)
##     No Information Rate : 0.3732
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.343
##
##  Mcnemar's Test P-Value : 0.8875
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
```

```
## Sensitivity            0.5468   0.5854   0.5543
## Specificity            0.7933   0.7570   0.7929
## Pos Pred Value         0.5569   0.5892   0.5397
## Neg Pred Value         0.7865   0.7541   0.8024
## Prevalence             0.3221   0.3732   0.3046
## Detection Rate         0.1762   0.2185   0.1689
## Detection Prevalence   0.3163   0.3708   0.3129
## Balanced Accuracy      0.6700   0.6712   0.6736
```

Note that if the predictions were random, the accuracy would be 33.3% (for three alternatives). Our simple model is doing much better than that – although it is not perfect.

4.2 Conjoint simulator *

Now, let us see how we can use model parameters to predict market shares under hypothetical market scenarios for an arbitrary set of products.

```
predict.share <- function(model, d) {
  temp <- model.matrix(update(model$formula, 0 ~ .), data = d)[,-1] # generate dummy matrix
  u <- temp%*%model$coef[colnames(temp)] # calculate utilities; %*% is matrix multiplication
  probs <- t(exp(u)/sum(exp(u))) # calculate probabilities
  colnames(probs) <- paste("alternative", colnames(probs))
  return(probs)
}

# hypothetical base market structure with 4 alternatives in the market

d.base <- cbc.df[c(44,34,33,40),c("Brand","Size","Storage","Ram", "Battery","Price")]

d.base <- cbind(d.base,as.vector(predict.share(model,d.base)))

colnames(d.base)[7] <- "Predicted.Share"
rownames(d.base) <- c()

kable(d.base)
```

Brand	Size	Storage	Ram	Battery	Price	Predicted.Share
iPad	sz7inch	st64gb	r2gb	b8h	399	0.3423928
Galaxy	sz10inch	st32gb	r2gb	b7h	299	0.2540301
Surface	sz10inch	st64gb	r1gb	b7h	399	0.1313854
Kindle	sz7inch	st32gb	r1gb	b9h	169	0.2721917

```
# hypothetical market structure after Galaxy gets a RAM upgrade

d.new <- d.base
d.new[2,'Ram'] <- "r4gb"

d.new$Predicted.Share <- as.vector(predict.share(model,d.new))
kable(d.new)
```

Brand	Size	Storage	Ram	Battery	Price	Predicted.Share
iPad	sz7inch	st64gb	r2gb	b8h	399	0.3127768
Galaxy	sz10inch	st32gb	r4gb	b7h	299	0.3185544
Surface	sz10inch	st64gb	r1gb	b7h	399	0.1200209
Kindle	sz7inch	st32gb	r1gb	b9h	169	0.2486479

4.3 Willingness to pay

Very importantly, using parameter estimates, we can calculate how much a consumer would be willing to pay for the selected level of an attribute by dividing the coefficient for that level by the coefficient for the price. In other words, we estimate what change in price would cause a shift in utility equivalent to that due to a change in the level of the attribute in question from the reference level.

For example, we see that an average consumer would be indifferent between getting a Galaxy vs. paying $125.8 more and getting an iPad. Phrasing this differently, an average consumer would be willing to pay up to $125.8 to get an iPad instead of a Nexus, holding all other characteristics fixed.

4.3.1 What is the brand value of iPad relative to Galaxy?

Brand equity - The dollar value of an upgrade from Galaxy to iPad

```
(coef(model)["BrandiPad"]-coef(model)["BrandGalaxy"]) / (-coef(model)["Price"])
```

```
## BrandiPad
##  125.7944
```

4.3.2 Willingness to Pay for an Attribute Upgrade

The dollar value of an upgrade from 1gb to 4gb ram (1gb is reference level. Hence its coeff is 0)

```
coef(model)["Ramr4gb"] /(-coef(model)["Price"])
```

```
##  Ramr4gb
## 124.9299
```

Dollar value of an upgrade from 7-inch to 9-inch screen (7-inch is the reference level)

```
coef(model)["Sizesz9inch"] / (-coef(model)["Price"])
```

```
## Sizesz9inch
##     85.5882
```