SMM 641
Revenue Management and Pricing

Week 9:
Price Optimization with Consumer Choice Part 2

MNL Demand Model using WTP/Choice Data
Assortment Optimization

MultiNomial Logit (MNL) model

- As we observed before, the willingness-to-pay differs across individuals, even across individuals within the same segment.
- MNL model captures the randomness in willingness-to-pay and is a popular tool for modeling how consumers choose from several alternatives.
- According to MNL model:

Individual's net utility from alternative j $= u_j - p_j + \varepsilon_j, \; j = 1, \ldots, n$

Gumbel distribution with mean Zero and variance $\mu^2 \pi^2/6$

MultiNomial Logit (MNL) model

- Suppose that we have a population of customers who have the same u_j values.
- However, one customer's random error terms will be different from another's.
- Each customer then chooses the alternative that maximizes her net utility.
- In this setting, the MNL model stipulates that:

Probability that a customer chooses alternative j $= \dfrac{v_j}{1 + \sum_{k=1}^{n} v_k}$ *where* $v_j = \exp\left(\dfrac{u_j - p_j}{\mu}\right), j = 1, \ldots, n$

Example: MultiNomial Logit (MNL) model

- Suppose we offer two products, A and B.
- The gross utilities are 100 and 80 for Products A and B, respectively.
- The prices are 70 and 60 for Products A and B, respectively.
- The variance of gross utilities across all the individuals across all the products is estimated to be 400.
- Question: What is the probability that a randomly chosen customer buys A? How about B? (equivalently, what fraction of customers buy A? B?)

Example: MultiNomial Logit (MNL) model

Using the information about variance of utilities, we first compute μ : 400 = $\mu^2 \pi^2/6$, μ = sqrt(2400)/π =15.6

We can then compute the v_j values for Products A and B.

Product	u_j	p_j	$v_j = \exp[(u_j-p_j)/\mu]$
A	100	70	exp((100-70)/15.6)=7.84
B	80	60	exp((80-60)/15.6)=3.6

Then:

Product	Probability of purchase = $v_j / (1+v_A + v_B)$
A	7.84/(1+7.84+3.6)=0.6
B	3.6/(1+7.84+3.6)=0.31
No Purchase	1/(1+7.84+3.6)=0.09

New York Health Club Part B

MNL Demand Model with WTP data

- Suppose we would like to identify a single price using the MNL demand model
- Step 1: Estimation. We will first estimate the parameters of the MNL demand model using the WTP data. In particular, we will estimate:
  - the gross utilities for each time slot $u_1, u_2, \ldots, u_6$ and
  - The shape (scale) parameter μ
- We can then identify the attraction value for each time slot which will allow us to find the probability that consumers will select each of the time slots.
- Step 2: Optimization. Using these choice probabilities, we can compute the expected revenue and pick the price that maximizes our expected revenue.

New York Health Club Part B

```
# Read the Survey Data
nyhc = read.csv("NYHCSurvey.csv",header=T)

# (1) Uncomment below line to set prices for all client types
#
nyhcnew = nyhc

# (2) To Set prices only for students
# Extract parts of data that correspond to students, i.e., type is 1.
# Uncomment below line to set prices only for students
#
# nyhcnew = subset(nyhc, Type == 1)

# (3) To Set prices for nonstudents (workers)
# Extract parts of data that correspond to Non students, i.e., type is 2
or 3.
# Uncomment below line to set prices only for nonstudents
#
# nyhcnew = subset(nyhc, Type == 2 | Type == 3)
```

New York Health Club Part B

- Step 1: Estimating Parameters ($u_1, u_2, \ldots, u_6$ and μ) of the MNL demand model from WTP data:

```
   Client Type TimeSlot1 TimeSlot2 TimeSlot3 TimeSlot4 TimeSlot5 TimeSlot6
1       1    1        18        50        41        76        69        41
2       2    1        66        14        86        62        71        46
3       3    1        60        43        43        26        91        58
4       4    1        59        48        62        15        91        15
5       5    1        21        24        59        45        81       131
6       6    1        66        69        29        80        37        79
7       7    1       126        41        61        97       111        56
8       8    1        72        71        40        11        78       116
9       9    1        49        64        54        26        49       107
10     10    1        17        21        41        54        76        27
```

- $u_1, u_2, \ldots, u_6$ are found by averaging the WTP of the columns:

```
TimeSlot1 TimeSlot2 TimeSlot3 TimeSlot4 TimeSlot5 TimeSlot6
   61.159    39.156    62.973    42.293    62.812    57.783
```

New York Health Club Part B

- To find μ, we first calculate the variance of all WTP values for all time slots:

```
1583.9056  911.4511  850.3686  563.9131 1452.7214 1400.5224
```

- Then calculate the average variance
  Average Variance = 1127.15

- Hence, $1127.15 = \frac{\mu^2\pi^2}{6}$, which results in $\mu = 26.18$

New York Health Club Part B

- Step 1: Estimating Parameters ($u_1, u_2, \ldots, u_6$ and μ) of the MNL demand model from WTP data:

```
N=nrow(nyhcnew)

# Compute Average Willingness To Pay across Time Slots:
AvgWTPs=colMeans(nyhcnew[3:8])
SdWTPs=colSds(as.matrix(nyhcnew[3:8]))
VarWTPs=SdWTPs^2

AvgVarWTPs=mean(VarWTPs)

# The parameters of the MNL model:
util=AvgWTPs            # u_i
mu=sqrt(6*AvgVarWTPs)/pi  # mu, shape parameter
```

New York Health Club Part B

- Step 2: Finding the Optimal Price

```
eval_f <- function(x){
  price=x

  # An array of prices for all 6 time slots (all equal to the same price)
  prices=rep(price,6)

  # Calculating attraction values
  attractions=exp((util-prices)/mu)

  # Calculating purchase probabilities
  # The +1 corresponds to the no purchase option
  # (no purchase has utility=0, and price=0, so its attraction is exp(0)=1)
  purchaseProbs=attractions/(sum(attractions)+1)

  # Calculating Expected Revenue
  # N: total number of people, for example N=1000
  # Revenue = 1000*( (prob of purchase for timeslot1)*(price of timeslot1)
  #                 +(prob of purchase for timeslot2)*(price of timeslot2)
  #                 + ...
  #                 +(prob of purchase for timeslot6)*(price of timeslot6)

  revenue=N*sum(prices*purchaseProbs)

  objfunction=-revenue
  return(objfunction)
}
```

New York Health Club Part B

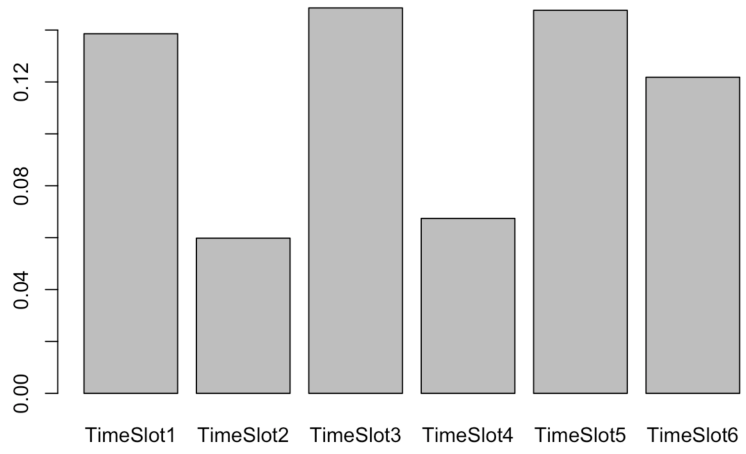
- Step 2: Finding the Optimal Price

```
# initial values
x0 <- 80
# lower and upper bounds
lb <- 25
ub <- 150
opts <- list( "algorithm" = "NLOPT_LN_COBYLA",
              "xtol_rel"  = 1.0e-6,
              "maxeval"   = 1000)

result <- nloptr(x0=x0,eval_f=eval_f,lb=lb,ub=ub,opts=opts)

priceOpt<-round(result$solution,2)

print(paste("Optimal Single Price:",priceOpt))
```

```
## [1] "Optimal Single Price: 82.76"
```

New York Health Club Part B

- Visualizing the Purchase Probabilities at the Optimal Price.



New York Health Club Part B

- Suppose NYHC decided to offer two flat rates, one for student members, and the other for all other members of the gym.
- The flat rates would apply to all time slots. For example, the student flat rate would price equally personal training sessions for students across all 6 time slots.
- Obtain the best price to charge for the students and for the nonstudents using the MNL demand models. (You may assume that the fraction of student members is similar to that of the survey, that is, 29.2% of the members are students.)

Obtaining MNL parameters from survey data on WTP

- For the student segment:

```
# Read the Survey Data
nyhc = read.csv("NYHCSurvey.csv",header=T)

# (1) Uncomment below line to set prices for all client types
#
# nyhcnew = nyhc

# (2) To Set prices only for students
# Extract parts of data that correspond to students, i.e., type is 1.
# Uncomment below line to set prices only for students
#
 nyhcnew = subset(nyhc, Type == 1)

# (3) To Set prices for nonstudents (workers)
# Extract parts of data that correspond to Non students, i.e., type is 2 or 3.
# Uncomment below line to set prices only for nonstudents
#
# nyhcnew = subset(nyhc, Type == 2 | Type == 3)
```

New York Health Club Part B

- Step 1: Estimating Parameters ($u_1, u_2, \ldots, u_6$ and μ) of the MNL demand model from WTP data:

```
N=nrow(nyhcnew)

# Compute Average Willingness To Pay across Time Slots:
AvgWTPs=colMeans(nyhcnew[3:8])
SdWTPs=colSds(as.matrix(nyhcnew[3:8]))
VarWTPs=SdWTPs^2

AvgVarWTPs=mean(VarWTPs)

# The parameters of the MNL model:
util=AvgWTPs           # u_i
mu=sqrt(6*AvgVarWTPs)/pi  # mu, shape parameter
```

New York Health Club Part B

- Step 2: Finding the Optimal Price

```r
eval_f <- function(x){
  price=x

  # An array of prices for all 6 time slots (all equal to the same price)
  prices=rep(price,6)

  # Calculating attraction values
  attractions=exp((util-prices)/mu)

  # Calculating purchase probabilities
  # The +1 corresponds to the no purchase option
  # (no purchase has utility=0, and price=0, so its attraction is exp(0)=1)
  purchaseProbs=attractions/(sum(attractions)+1)

  # Calculating Expected Revenue
  # N: total number of people, for example N=1000
  # Revenue = 1000*( (prob of purchase for timeslot1)*(price of timeslot1)
  #                 +(prob of purchase for timeslot2)*(price of timeslot2)
  #                 + ...
  #                 +(prob of purchase for timeslot6)*(price of timeslot6)

  revenue=N*sum(prices*purchaseProbs)

  objfunction=-revenue
  return(objfunction)
}
```

New York Health Club Part B

- Step 2: Finding the Optimal Price

```r
# initial values
x0 <- 80
# lower and upper bounds
lb <- 25
ub <- 150
opts <- list( "algorithm" = "NLOPT_LN_COBYLA",
              "xtol_rel"  = 1.0e-6,
              "maxeval"   = 1000)

result <- nloptr(x0=x0,eval_f=eval_f,lb=lb,ub=ub,opts=opts)

priceOpt<-round(result$solution,2)

print(paste("Optimal Single Price:",priceOpt))
```

```
## [1] "Optimal Single Price: 69.02"
```

Obtaining MNL parameters from survey data on WTP

- For the student segment:

```
## [1] "The gross utilities are:"
```

```r
util
```

```
## TimeSlot1 TimeSlot2 TimeSlot3 TimeSlot4 TimeSlot5 TimeSlot6
##  57.29795  45.30137  48.81507  52.99315  58.95548  58.35274
```

```r
mu=sqrt(6*AvgVarWTPs)/pi  # mu, shape (scale) parameter
print("The scale parameter mu is:")
```

```
## [1] "The scale parameter mu is:"
```

```r
mu
```

```
## [1] 18.57608
```

Obtaining MNL parameters from Choice Data

- Structure of Choice Data:

```r
# Read Choice Data (for students)
mydata <- read.csv("NYHCstudentChoiceData.csv")
head(mydata)
```

```r
# mlogit expects the price columns be named in the form of price.productname.
# For example, price.timeslot1, price.timeslot2, etc.
# We have introduced DNB to capture no purchase, i.e., Did Not Buy
```

	Individual	price.T1	price.T2	price.T3	price.T4	price.T5	price.T6	price.DNB	choice
1	1	50	50	50	50	50	50	0	T4
2	2	50	50	50	50	50	50	0	T3
3	3	60	60	60	60	60	60	0	T5
4	4	60	60	60	60	60	60	0	T5
5	5	50	50	50	50	50	50	0	T6
6	6	60	60	60	60	60	60	0	T4

Obtaining MNL parameters from Choice Data

- Data Preparation:

```r
# mlogit first needs to transform data into the shape it requires.
# Our original data is in the "wide" format,
# meaning each choice decision is in one row.
# varying means, which columns are varying properties for alternatives
# Here, they are the columns associated with the prices of each alternative.
# Choice means which column contains the overall choice of the customer.
# If the choice is stated as 1s and 0s in each row rather than directly given
# we can do a simple preprocessing to generate a column with the choice name
# Here I prepared the data so that it is ready for the mlogit.data function

mydata2 <- mlogit.data(mydata, shape = "wide", varying=2:8, choice = "choice")

# In this transformed version, The data is now in the long format.
# Each alternative is a row by itself and
# a client has many rows based on the number of alternatives it sees.
```

Obtaining MNL parameters from Choice Data

- Running the (Maximum Likelihood) Estimation:

```r
# We run a Maximum Likelihood Estimation (MLE)
# to obtain the parameters of the MNL demand model
mymodel <- mlogit(choice ~ price, data=mydata2,reflevel="DNB")

# Summarizing the results of the Estimation
stargazer(mymodel, type="text")
```

Obtaining MNL parameters from Choice Data

- MLE Estimation results:

```
##
## ==================================================
##                          Dependent variable:
##                      ----------------------------
##                                choice
## --------------------------------------------------
## timeslot1:(intercept)         10.612**
##                               (4.607)
##
## timeslot2:(intercept)          8.973*
##                               (4.613)
##
## timeslot3:(intercept)          9.704**
##                               (4.609)
##
## timeslot4:(intercept)         10.097**
##                               (4.608)
##
## timeslot5:(intercept)         10.712**
##                               (4.607)
##
## timeslot6:(intercept)         10.502**
##                               (4.607)
##
## price                         -0.155**
##                               (0.079)
##
## --------------------------------------------------
## Observations                    292
## R2                             0.005
## Log Likelihood               -512.976
## LR Test                    4.993 (df = 7)
## ==================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

Obtaining MNL parameters from Choice Data

- Obtaining the parameters of the MNL model:

```r
# Remember from the MNL model, that the expression is in the form (u_i-p_i) / mu
# Hence (u_i/mu) is the intercept and (-1/mu) is the coefficient for price.
# That is, mu=-1/(price coefficient)

mu=-1/summary(mymodel)$coefficients[7]

print(paste("The shape parameter mu is:", round(mu,2)))
```

```
## [1] "The shape parameter mu is: 6.44"
```

```r
# And, u_i= intercept_i * mu

util=summary(mymodel)$coefficients[1:6]*mu
names(util)=paste0("T",c(1:6))

print("The gross utilities, u_i are:")
```

```
## [1] "The gross utilities, u_i are:"
```

```r
print(round(util,2))
```

```
##    T1    T2    T3    T4    T5    T6
## 68.30 57.74 62.45 64.98 68.94 67.59
```

Obtaining MNL parameters from Choice Data

- Price Optimization (similar as before):

```r
# Finding the optimal price

N=nrow(mydata)

eval_f <- function(x){
  price=x

  # An array of prices for all 6 time slots (all equal to the same price)
  prices=rep(price,6)

  # Calculating attraction values
  attractions=exp((util-prices)/mu)

  # Calculating purchase probabilities
  # The +1 corresponds to the no purchase option
  # (no purchase has utility=0, and price=0, so its attraction is exp(0)=1)
  purchaseProbs=attractions/(sum(attractions)+1)

  # Calculating Expected Revenue
  # N: total number of people, for example N=1000
  # Revenue = 1000*( (prob of purchase for timeslot1)*(price of timeslot1)
  #                  +(prob of purchase for timeslot2)*(price of timeslot2)
  #                  + ...
  #                  +(prob of purchase for timeslot6)*(price of timeslot6)

  revenue=N*sum(prices*purchaseProbs)

  objfunction=-revenue
  return(objfunction)
}
```

Obtaining MNL parameters from Choice Data

- Price Optimization (similar as before):

```r
# initial values
x0 <- 80
# lower and upper bounds of control
lb <- 25
ub <- 150
opts <- list( "algorithm" = "NLOPT_LN_COBYLA",
              "xtol_rel"  = 1.0e-6,
              "maxeval"   = 1000)
result <- nloptr(x0=x0,eval_f=eval_f,lb=lb,ub=ub,opts=opts)
#result <- nloptr(x0=x0,eval_f=eval_f,lb=lb,ub=ub,eval_g_ineq=eval_g_ineq,opts=opts)
# print(result)

priceOpt<-round(result$solution,2)

print(paste("Optimal Price:", priceOpt))
```

```
## [1] "Optimal Price: 63.48"
```

Assortment Optimization

- Given a set of items (with given prices) and a choice model,
- What is the best subset of items to offer in order to maximize revenue?
- Note: If there are N potential items, # of subsets of $N = 2^N - 1$. In general, finding the optimal subset may be very difficult.
- For the MNL model with a single class of customers, there is a very efficient algorithm to determine the optimal assortment.

Example: Assortment Optimization under MNL

- Suppose there are 5 possible items to include in the assortment with:
  - prices $(p_1, p_2, \ldots, p_5) = (7, 6, 4, 3, 2)$ (Note: in order of decreasing prices)
  - MNL attraction values $(v_1, v_2, \ldots, v_5) = (3, 5, 6, 4, 5)$ with no purchase option $v_0 = 10$.

Example: Assortment Optimization under MNL

```
# Attractions for the products
v=c(3,5,6,4,5)

# Attraction for the outside option
v0=10

# Prices for the products
p=c(7,6,4,3,2)

# Let i1 denote an indicator funtion for whether product 1 will be
# included in the assortment or not.
# If i1=1, product 1 is included in the assortment
# If i1=0, product 1 is not included in the assortment

# Similarly, defined i2,i3,i4,i5.

# We will consider all possible combinations of assortments.
# For each assortment, we will compute the purchase probabilities
# within that particular assortment and record the revenue
```

Example: Assortment Optimization under MNL

```
# For bookkeeping, we will track each case with an index variable.

index=1

Probpurchase=rep(NA,5)
Revenue=rep(NA,2^5)
assortmenttable=data.frame(matrix(nrow=2^5,ncol = 12))
indicatorLabels=c(paste0("Prod", 1:5))
purchaseProbLabels=c(paste0("pBuy", 1:5))
colnames(assortmenttable)=c("N",indicatorLabels,purchaseProbLabels, "Revenue")

for (i1 in 0:1){
  for (i2 in 0:1){
    for (i3 in 0:1){
      for (i4 in 0:1){
        for (i5 in 0:1){

          # Collecting all assortment indicators in a single array
          i=c(i1,i2,i3,i4,i5)

          # Computing the denominator for the purchase probabilities
          Denominator=sum(v*i)+v0

          #Obtaining the purchase probabilities
          for (j in 1:5){
            Probpurchase[j]=v[j]*i[j]/Denominator
          }

          # Revenue from this assortment
          Revenue[index]=sum(Probpurchase*p)

          assortmenttable[index,1]=sum(i)
          assortmenttable[index,2:6]=i
          assortmenttable[index,7:11]=round(Probpurchase,3)
          assortmenttable[index,12]=round(Revenue[index],3)

          index=index+1
        }
      }
    }
  }
}

assortmenttable2<-assortmenttable[order(assortmenttable$N),]
```

Example: Assortment Optimization under MNL

N	Prod1	Prod2	Prod3	Prod4	Prod5	pBuy1	pBuy2	pBuy3	pBuy4	pBuy5	Revenue
0	0	0	0	0	0	0.000	0.000	0.000	0.000	0.000	0.000
1	0	0	0	0	1	0.000	0.000	0.000	0.000	0.333	0.667
1	0	0	0	1	0	0.000	0.000	0.000	0.286	0.000	0.857
1	0	0	1	0	0	0.000	0.000	0.375	0.000	0.000	1.500
1	0	1	0	0	0	0.000	0.333	0.000	0.000	0.000	2.000
1	1	0	0	0	0	0.231	0.000	0.000	0.000	0.000	1.615
2	0	0	0	1	1	0.000	0.000	0.000	0.211	0.263	1.158
2	0	0	1	0	1	0.000	0.000	0.286	0.000	0.238	1.619
2	0	0	1	1	0	0.000	0.000	0.300	0.200	0.000	1.800
2	0	1	0	0	1	0.000	0.250	0.000	0.000	0.250	2.000
2	0	1	0	1	0	0.000	0.263	0.000	0.211	0.000	2.211
2	0	1	1	0	0	0.000	0.238	0.286	0.000	0.000	2.571
2	1	0	0	0	1	0.167	0.000	0.000	0.000	0.278	1.722
2	1	0	0	1	0	0.176	0.000	0.000	0.235	0.000	1.941
2	1	0	1	0	0	0.158	0.000	0.316	0.000	0.000	2.368
2	1	1	0	0	0	0.167	0.278	0.000	0.000	0.000	2.833
3	0	0	1	1	1	0.000	0.000	0.240	0.160	0.200	1.840
3	0	1	0	1	1	0.000	0.208	0.000	0.167	0.208	2.167
3	0	1	1	0	1	0.000	0.192	0.231	0.000	0.192	2.462
3	0	1	1	1	0	0.000	0.200	0.240	0.160	0.000	2.640
3	1	0	0	1	1	0.136	0.000	0.000	0.182	0.227	1.955
3	1	0	1	0	1	0.125	0.000	0.250	0.000	0.208	2.292
3	1	0	1	1	0	0.130	0.000	0.261	0.174	0.000	2.478
3	1	1	0	0	1	0.130	0.217	0.000	0.000	0.217	2.652
3	1	1	0	1	0	0.136	0.227	0.000	0.182	0.000	2.864
3	1	1	1	0	0	0.125	0.208	0.250	0.000	0.000	3.125
4	0	1	1	1	1	0.000	0.167	0.200	0.133	0.167	2.533
4	1	0	1	1	1	0.107	0.000	0.214	0.143	0.179	2.393
4	1	1	0	1	1	0.111	0.185	0.000	0.148	0.185	2.704
4	1	1	1	0	1	0.103	0.172	0.207	0.000	0.172	2.931
4	1	1	1	1	0	0.107	0.179	0.214	0.143	0.000	3.107
5	1	1	1	1	1	0.091	0.152	0.182	0.121	0.152	2.939

Example: Assortment Optimization under MNL

- Suppose there are 5 possible items to include in the assortment with:
  - prices $(p_1, p_2, \ldots, p_5) = (7, 6, 4, 3, 2)$ (Note: in order of decreasing prices)
  - MNL attraction values $(v_1, v_2, \ldots, v_5) = (3, 5, 6, 4, 5)$ with no purchase option $v_0 = 10$.
- We only need to consider assortments that include the *n* highest priced products in an assortment of size *n*, etc.

Assortment	{1}	{1,2}	{1,2,3}	{1,2,3,4}	{1,2,3,4,5}
Revenue	1.615	2.833	3.125	3.107	2.939

- For highest revenue, include items 1, 2, and 3 in the assortment.