Estimating Market Shares and Optimal Prices with a Heterogenous Multinominal Logit Model

An Empirical Study of TNS Beer Market Data

Term Paper

submitted to

Prof. Dr. Thomas Otter

Goethe University Frankfurt am Main
School of Business and Economics
Chair of Services Marketing

by

Lukas Jürgensmeier
(Mat.-Nr.: 6904281)

in partial fulfillment of the requirements of the lecture

Customer Satisfaction and Consumer Choice
Summer Semester 2019

and for the degree of

Master of Science in Business Administration

July 23, 2019

Contents

List of Figures

List of Tables

1 Contextual Introduction to the Beer Market

asd

2 Motivation of the Analytical Model

3 Exogenous Variables and Expected Relationonships

asd

4 Model Description and Results

asd

5 Market Simulation

asd

6 Managerial and Research Implications

asd

A R Code

```r
################################
### Estimation for Beer #######
################################


## Prepare and load data

rm(list=ls())

library(WriteXLS)
library(Rcpp)
library(devtools)
library(RcppArmadillo)
library(MASS)
library(lattice)
library(Matrix)
library(xtable)
library(bayesm)

set.seed(66)

###Increase memory capacities
memory.limit(size=1000000)

load("Estimation_Data_Beer_20170423.Rdata")

products = c("Amstel Extra Lata 37,5 cl","Amstel Extra Lata 33 cl","Amstel Lata 37,5 cl","Amstel Lata 33 cl","Amstel Cl?sica Lata 33 cl",
             "Cruzcampo Lata 33 cl","Estrella Damm Lata 33 cl","Estrella Galicia Lata 33 cl","Heineken Lata 33 cl","Mahou 5 Estrellas Lata 33 cl",
             "Mahou Cl?sica Lata 33 cl","San Miguel Lata 33 cl","Voll Damm Lata 33 cl","Steinburg (Marca Blanca Mercadona) Lata 33 cl",
             "Marca Blanca Carrefour Lata 33 cl")

N = length(E_Data$lgtdata)

for(i in 1:N){
  colnames(E_Data$lgtdata[[i]]$X) = c(products,"Price")
}

colnames(E_Data$lgtdata[[101]]$X)

save(E_Data,file="Estimation_Data_Beer_20170423.Rdata")


## estimation preparation for bayesm package
Prior = list(ncomp=1)
Mcmc=list(R=6000,keep=2)

out_HB = rhierMnlRwMixture(Data=E_Data,Prior=Prior,Mcmc=Mcmc)
beta_HB = out_HB$betadraw
compdraw_HB = out_HB$nmix$compdraw
probdraw_HB = out_HB$nmix$probdraw

windows()
plot(out_HB$loglike, type="l")

###Get rid of burnin
burnin = 1000
R = dim(beta_HB)[3]

beta_HB = beta_HB[,,(burnin+1):R]
compdraw_HB = compdraw_HB[(burnin+1):R]
probdraw_HB = probdraw_HB[(burnin+1):R]

###EVALUATION
R = dim(beta_HB)[3]
```

```r
N = dim(beta_HB)[1]

l = 100
index = rep(rank(runif(R)),l)

beta_HP <- array(0,dim=c(R*l,dim(beta_HB)[2]))
#simulate from posterior predictive density of beta (hierarchical prior)
#simulate from posterior predictive density of beta (hierarchical prior)
#simulate from posterior predictive density of beta (hierarchical prior)
#  check ?rmixture
for(j in 1:(R*l)){
  beta_HP[j,] = rmixture(1,probdraw_HB[index[j]],compdraw_HB[[index[j]]])$x
}


#######resampling from out_HB$betadraw
beta_HP <- array(aperm(beta_HB,perm=c(1,3,2)),dim=c(dim(beta_HB)[1]*dim(beta_HB)[3],
    dim(beta_HB)[2]))
# beta_HP2 <- NULL
# for(j in 1:R){
#   beta_HP = rbind(beta_HP,beta_HB[,,j])
#   }

#######using posterior means
#######using posterior means
#######using posterior means
beta_HP <- rowMeans(beta_HB,dim=2)


#Illustrate specified distribution graphically
windows()
par(mfrow=c(4,4))
hist(beta_HP[,1], freq = FALSE,breaks=100,xlab="BETA",ylab="DENSITY",
     main=paste("Attribute 1 Level 1:", round(mean(beta_HP[,1]),digits = 2)));grid()
hist(beta_HP[,2], freq = FALSE,breaks=100,xlab="BETA",ylab="DENSITY",
     main=paste("Attribute 1 Level 2:", round(mean(beta_HP[,2]),digits = 2)));grid()
hist(beta_HP[,3], freq = FALSE,breaks=80,xlab="BETA",ylab="DENSITY",
     main=paste("Attribute 2 Level 2:", round(mean(beta_HP[,3]),digits = 2)));grid()

hist(beta_HP[,4], freq = FALSE,breaks=100,xlab="BETA",ylab="DENSITY",
     main=paste("Attribute 1 Level 1:", round(mean(beta_HP[,1]),digits = 2)));grid()
hist(beta_HP[,5], freq = FALSE,breaks=100,xlab="BETA",ylab="DENSITY",
     main=paste("Attribute 1 Level 2:", round(mean(beta_HP[,2]),digits = 2)));grid()
hist(beta_HP[,6], freq = FALSE,breaks=100,xlab="BETA",ylab="DENSITY",
     main=paste("Attribute 2 Level 2:", round(mean(beta_HP[,3]),digits = 2)));grid()

hist(beta_HP[,16], freq = FALSE,breaks=80,xlab="BETA",ylab="DENSITY",
     main=paste("Price:",  round(mean(beta_HP[,16]),digits = 2)));grid()


###Matrix of product combinations (ignoring price)
#comb_m = rbind(c(1,0,1,0),c(0,1,1,0),c(1,0,0,1),c(0,1,0,1))

###Specify grid for price
min_p = 0
max_p = 1
step = 0.01 #.5
price_grid = seq(min_p,max_p,step)
grid_length = length(price_grid)

# create matrix with two columns: First one only 1 for our product, second one price
    grid
comb_m_p <- cbind(rep(1,grid_length), price_grid)

###Product Optimization###

###Approximate expected market share

NUT_agg_esti = beta_HP[,-c(2:15)]%*%t(comb_m_p) # remember identification of the
    choice likelihood?
```

```
#NUT_agg_esti = beta_HP[runif(dim(beta_HP)[1])>.9,]%*%t(comb_m_p[,-3]) # without (
    randomized) subsetting the object will be too big for the workspace

Exp_esti = exp(-NUT_agg_esti)
sc_esti_all = 1/(1+Exp_esti) #Compute share of choice (sc) ~ market share from
    monopolistic perspective
sc_esti = apply(sc_esti_all,2,mean) #Compute mean over draws to approximate the
    integral (expected value)
#Compute profits
costs_A2 = 0.1
profits_esti = array(0,dim=c(length(sc_esti),dim(costs_A2)[1]))
grid_price_minus_cost = price_grid - costs_A2
profits_esti = sc_esti * grid_price_minus_cost

# for(i in 1:dim(costs_A2)[2]){
#   profits_esti[,i] = sc_esti * grid_price_minus_cost[,i]
# }
#Compute optimal product for each possible cost combinations
optimal_product_esti = array(0,dim=c(dim(costs_A2)[1],1))
optimal_product_esti = which(profits_esti == max(profits_esti[]), arr.ind = TRUE)
# for(i in 1:dim(costs_A2)[2]){
#   optimal_product_esti[i] = which(profits_esti[,i] == max(profits_esti[,i]), arr.ind
     = TRUE)
# }

###Plot profit curve for each cost scenarion
plot(profits_esti[],col = "red",type="l",xlab="Price", main="Optimal price of first
    brand in monopolistic market" ,ylab="Profits");grid()
abline(v=optimal_product_esti[],col = "red",lty=3,lwd=3)


windows()
par(mfrow=c(3,1))  # multiple plots are filled by rows!!!
plot(profits_esti[,1],col = "red",type="l",xlab="Product Index", main="Cost Scenario I
    " ,ylab="Profits");grid()
abline(v=optimal_product_esti[1],col = "red",lty=3,lwd=3)
legend("bottomleft",expression("Profits Estimated"),cex=1,lty=c(1),lwd=c(1),col=c("red
    "))
plot(profits_esti[,2],col = "red",type="l",xlab="Product Index", main="Cost Scenario
    II" ,ylab="Profits");grid()
abline(v=optimal_product_esti[2],col = "red",lty=3,lwd=3)
legend("bottomleft",expression("Profits Estimated"),cex=1,lty=c(1),lwd=c(1),col=c("red
    "))
plot(profits_esti[,3],type="l",col = "red",xlab="Product Index", main="Cost Scenario
    III" ,ylab="Profits");grid()
abline(v=optimal_product_esti[3],col = "red",lty=3,lwd=3)
legend("bottomleft",expression("Profits Estimated"),cex=1,lty=c(1),lwd=c(1),col=c("red
    "))

###Summarize results in one matrix
Optimal_Product_ALL = NULL
Optimal_Product_esti <- array(0,dim=c(3,6))
rownames(Optimal_Product_esti) = c("Scenario I","Scenario II","Scenario III")
colnames(Optimal_Product_esti) = c("Attribute 1 Level 1", "Attribute 1 Level 2", "
    Attribute 2 Level 1", "Attribute 2 Level 2", "Price", "Profits")
for(i in 1:dim(costs_A2)[2]){
  Optimal_Product_esti[i,] = c(comb_m_p[optimal_product_esti[i],],profits_esti[optimal
      _product_esti[i],i])
}

###Optimal products from estimation
Optimal_Product_esti
###True Optimal products
Optimal_Product
```

Listing 1: Estimation Code

Statutory Declaration

I herewith declare that I have completed the present term paper independently, without making use of other than the specified literature and aids. Sentences or parts of sentences quoted literally are marked as quotations; identification of other references with regard to the statement and scope of the work is quoted. The thesis in this form or in any other form has not been submitted to an examination body and has not been published. This thesis has not been used, either in whole or part, for another examination achievement.

Frankfurt am Main, July 23, 2019

. . . . . .

Lukas Jürgensmeier