Exercices Misra MNL

Tom BLACHON Meyssa BEDDAR Matthieu SIMOES

09/05/2021

Présentation de l'exercice

Dans ce projet, nous allons apprendre les fondements théoriques des ***Discrete Choice Models*** et en particulier du **Modèle Multinomial Logit**.

Pour celà, nous nous baserons sur l'article de Sanjog Misra suivant :

- MISRA, Sanjog. "Generalized reverse discrete choice models". in *Quantitative Marketing and Economics*, 2005, vol. 3, n°2, p.175-200

Présentation des données

Afin de mieux comprendre le fonctionnement du modèle multinomial logit, nous utiliserons des données de consommateurs, présentant les choix effectués par ces derniers lors de l'achat de yaourts et de crackers. Les consommateurs ont le choix entre 4 marques de yaourts différentes :

- Yoplait
- Dannon
- Weightwatcher
- Hiland

Et 4 marques de crackaers différentes :

- Sunshine
- Keebler
- Nabisco
- Private label

A chaque fois, nous disposons du **prix du produit** (*Price*), de ses **caractéristiques** (*Features*) et de sa **part de marché** (*Market Share*). Nous disposons également du choix effectué par le consommateur entre les quatre produits.

Comme nous le verrons plus tard, notre objectif est donc d'utiliser le modèle multinomial logit (MNL) pour estimer le choix du consommateur à l'aide de ces informations.

Import des données :

Commençons par importer les données, issues des packages "mlogit" et "Ecdat" :

```
library(mlogit)
library(Ecdat)
library(stargazer)
library(dplyr)
library(knitr)
library(kableExtra)
library(tinytex)
library(ggplot2)
```

```
library(ggthemes)

data("Cracker",package="mlogit")
data("Yogurt",package="Ecdat")
```

Pivot des données :

Ensuite, nous faisons **pivoter nos données**. Ainsi, une ligne correspondra au choix effectué par un consommateur face à un produit (choisi/non choisi). Un ensemble de 4 lignes correspond alors au choix du consommateur parmis les 4 marques de produits proposés : 3 lignes correspondront aux produits "non choisis" et une ligne correspondra au produit "choisi" :

```
Yogurtlong<-mlogit.data(Yogurt,shape = "long",varying = 2:9,choice = "choice")
names(Yogurtlong)[3]<-c("Brand")
Yogurtlong<-Yogurtlong[,2:5]
Yogurtlong$choice[Yogurtlong$choice=="FALSE"]=c(0)
Yogurtlong$choice[Yogurtlong$choice=="TRUE"]=c(1)
Yogurtlong$price <- Yogurtlong$price/100
head(Yogurtlong, 4)
```

```
## ~~~~~~~
##  first 4 observations out of 9648
## ~~~~~~~
##   choice   Brand feat price    idx
## 1      0  dannon    0 0.081 1:nnon
## 2      0  hiland    0 0.061 1:land
## 3      1  weight    0 0.079 1:ight
## 4      0 yoplait    0 0.108 1:lait
##
## ~~~ indexes ~~~~
##   chid     alt
## 1    1  dannon
## 2    1  hiland
## 3    1  weight
## 4    1 yoplait
## indexes:  1, 2
```

```
Crackerlong<-mlogit.data(Cracker,shape = "long",varying = 2:13,choice = "choice")
names(Crackerlong)[3]<-c("Brand")
Crackerlong<-Crackerlong[,2:6]
Crackerlong$choice[Crackerlong$choice=="FALSE"]=c(0)
Crackerlong$choice[Crackerlong$choice=="TRUE"]=c(1)
Crackerlong$price <- Crackerlong$price/100
head(Crackerlong, 4)
```

```
## ~~~~~~~
##  first 4 observations out of 13168
## ~~~~~~~
##   choice    Brand disp feat price    idx
## 1      0  keebler    0    0  0.88 1:bler
## 2      1  nabisco    0    0  1.20 1:isco
## 3      0  private    0    0  0.71 1:vate
## 4      0 sunshine    0    0  0.98 1:hine
##
## ~~~ indexes ~~~~
```

```
##   chid      alt
## 1    1  keebler
## 2    1  nabisco
## 3    1  private
## 4    1 sunshine
## indexes:  1, 2
```

Dans nos jeux de données, nous ne concervons donc que les informations concernant la marque, le prix, les caractéristiques et la part du marché d'un produit, ainsi que le choix du consommateur dans la colonne "choice" :

- 0 : Le consommateur n'a pas choisi ce produit
- 1 : Le consommateur a choisi ce produit

Statistiques récapitulatives des données Cracker et Yaourt :

Nous souhaitons ici réaliser un tableau statistique afin de récapituler les données présentes dans nos dataframes :

```
MS_tab=summarise(
  group_by(Yogurtlong,Brand),
           mean=mean(choice),
           sd=sd(choice),
             )
Price_tab=summarise(
  group_by(Yogurtlong,Brand),
           mean=mean(price),
           sd=sd(price),
             )
Feat_tab=summarise(
  group_by(Yogurtlong,Brand),
           mean=mean(feat),
           sd=sd(feat),
               )


MS_tab_bind <- cbind(data.frame(c("Market Shares", "", "", "")), MS_tab)
colnames(MS_tab_bind) <- c("Variable", "Brand", "Mean", "Std. Dev.")
Feat_tab_bind <- cbind(data.frame(c("Feat", "", "", "")), Feat_tab)
colnames(Feat_tab_bind) <- c("Variable", "Brand", "Mean", "Std. Dev.")
Price_tab_bind <- cbind(data.frame(c("Price", "", "", "")), Price_tab)
colnames(Price_tab_bind) <- c("Variable", "Brand", "Mean", "Std. Dev.")

Yog_bind <- rbind(MS_tab_bind, Feat_tab_bind, Price_tab_bind)
```

```
Crack_MS_tab=summarise(
  group_by(Crackerlong,Brand),
           mean=mean(choice),
           sd=sd(choice),
             )
Crack_Price_tab=summarise(
  group_by(Crackerlong,Brand),
           mean=mean(price),
           sd=sd(price),
             )
Crack_Feat_tab=summarise(
```

```
  group_by(Crackerlong,Brand),
          mean=mean(feat),
          sd=sd(feat),
              )


Crack_MS_tab_bind <- cbind(data.frame(c("Market Shares", "", "", "")), Crack_MS_tab)
colnames(Crack_MS_tab_bind) <- c("Variable", "Brand", "Mean", "Std. Dev.")
Crack_Feat_tab_bind <- cbind(data.frame(c("Feat", "", "", "")), Crack_Feat_tab)
colnames(Crack_Feat_tab_bind) <- c("Variable", "Brand", "Mean", "Std. Dev.")
Crack_Price_tab_bind <- cbind(data.frame(c("Price", "", "", "")), Crack_Price_tab)
colnames(Crack_Price_tab_bind) <- c("Variable", "Brand", "Mean", "Std. Dev.")

Crack_bind <- rbind(Crack_MS_tab_bind, Crack_Feat_tab_bind, Crack_Price_tab_bind)
```

```
kable(cbind(Crack_bind,Yog_bind), digits = 5, booktabs = TRUE,
      caption = "Statistiques récapitulatives") %>%
  add_header_above(c("Cracker Data" = 4, "Yogurt Data" = 4)) %>%
  column_spec(c(1,5), italic=TRUE) %>%
  kable_styling(latex_options = "hold_position")
```

Table 1: Statistiques récapitulatives

Cracker Data				Yogurt Data			
Variable	Brand	Mean	Std. Dev.	Variable	Brand	Mean	Std. Dev.
Market Shares	keebler	0.06865	0.25290	*Market Shares*	dannon	0.40216	0.49043
	nabisco	0.54435	0.49810		hiland	0.02944	0.16906
	private	0.31440	0.46435		weight	0.22927	0.42045
	sunshine	0.07260	0.25952		yoplait	0.33914	0.47351
Feat	keebler	0.04253	0.20182	*Feat*	dannon	0.03773	0.19058
	nabisco	0.08657	0.28125		hiland	0.03690	0.18855
	private	0.04708	0.21185		weight	0.03773	0.19058
	sunshine	0.03767	0.19042		yoplait	0.05597	0.22991
Price	keebler	1.12594	0.10638	*Price*	dannon	0.08163	0.01063
	nabisco	1.07923	0.14478		hiland	0.05363	0.00805
	private	0.68073	0.12407		weight	0.07949	0.00774
	sunshine	0.95703	0.13292		yoplait	0.10682	0.01906

Multinomial Logit Model (MNL)

Présentation du modèle :

Notre objectif est de comprendre, à partir des données sur les prix et caractéristiques de chaque marque, comment déterminer la probabilité du choix d'un consommateur pour un Yaourt plutôt qu'un autre. Pour celà, nous allons utiliser le modèle Multinomial Logit (MNL).

Un modèle logit, qu'il soit binomial ou multinomial, vise à estimer la probabilité d'un évènement à partir de variables "explicatives". Dans notre cas, il s'agit d'un modèle multinomial : l'évènement que l'on cherche à estimer est le choix de la marque de Yaourt. Or, il existe 4 marques différentes, donc pour chaque individu, nous pouvons calculer 4 probabilités.

La formule de calcul des probabilités dans ce modèle est donc la suivante :

$$\mathbb{P}_i = \frac{e^{u_i}}{\sum_{k=1}^{n} e^{u_k}},\ i = 1, \cdots, n.$$

où $u_1, \cdots, u_n$ sont les parties déterministes ou systématiques des utilités et n le nombre de choix. Et où $\mathbb{P}_1, \cdots, \mathbb{P}_n$ représente la probabilité des choix.

Dans le modèle MNL, chaque variable explicative est associée à un coefficient β dont on cherche à optimiser la valeur afn de paramètrer correctement le modèle.

Pour connaitre la valeur optimale de β pour une variable, on va calculer la vraissemblance de cette variable. La vraissemblance d'une observation individuelle dans un modèle MNL s'écrit :

$$L = \prod_{i=1}^{n} \mathbb{P}_i^{y_i}$$

où $y_i = 1$ si l'alternative i est choisie et $y_i = 0$ sinon.

La vraissemblance pour tout l'échantillon est égale au produit des vraissemblances des observations individuelles. Mais on préfère généralement utiliser le logarithme de la vraissemblance. Pour une observation individuelle, celui-ci s'écrit de la façon suivante :

$$\log(L) = \sum_{i=1}^{n} y_i \log(\mathbb{P}_i)$$

Nous alons ainsi chercher à maximiser la valeur de ce log vraissemblance afin de déterminer la meilleure valeur de β.

Calcul manuel du β optimal pour la variable "Price" :

Afin de mieux comprendre ce calcul, nous alons le réaliser manuellement pour la variable "Price", bien que de nombreuses fonctions permettent de le calculer automatiquement sur R.

Effectons un premier test pour le premier évènement de nos données :

```
# Initialisation du paramètree beta à 1
beta = 1

# Calcul test de la log vraissemblance sur les 4 premières lignes de la table Yogurt
log(exp(beta*Yogurtlong$price[1:4])/sum(exp(beta*Yogurtlong$price[1:4])))%*%
  Yogurtlong$choice[1:4]
```

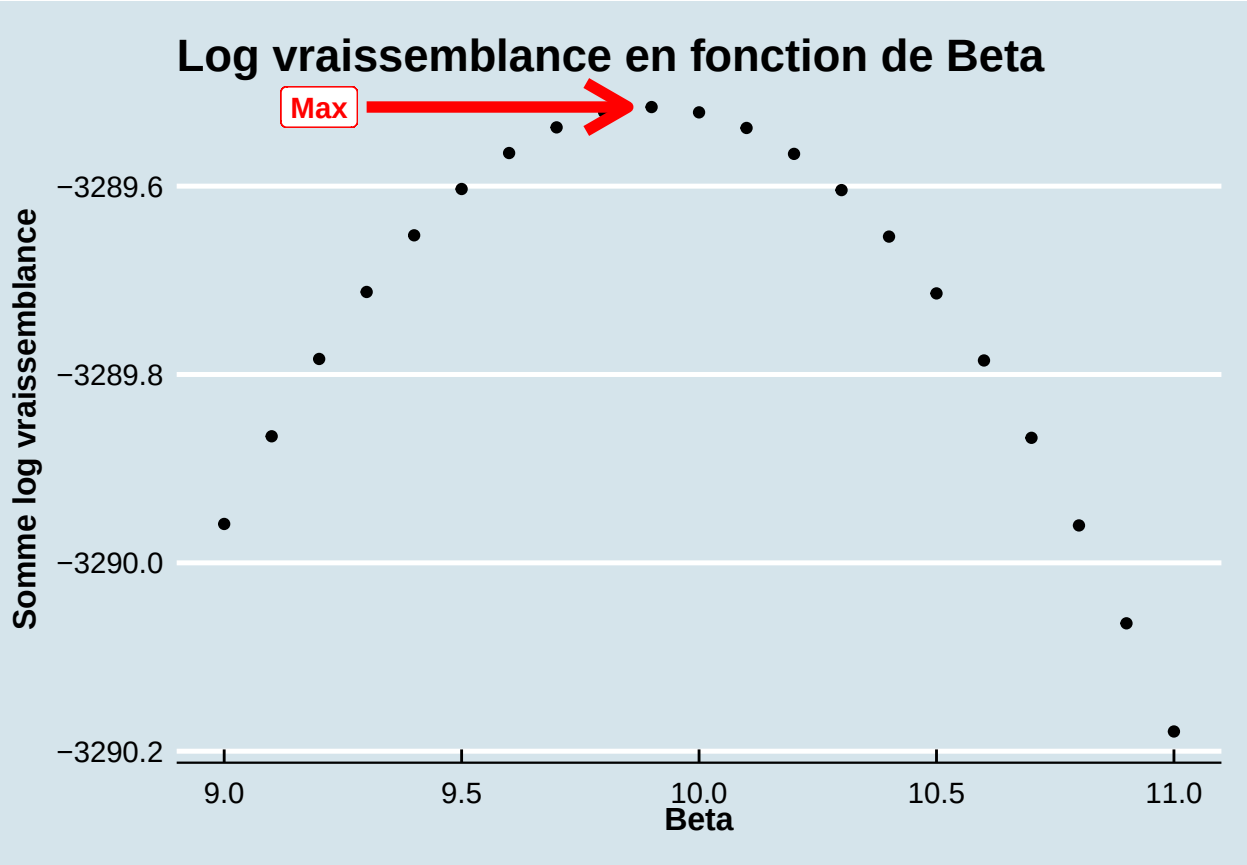
```
##           [,1]
## [1,] -1.389686
```

Maintenant que notre formule semble opérationnelle, créons une fonction nous permetant de calculer la somme des log vraissemblances de notre jeu de données à partir de la valeur de β et de la variable "Price" :

```
LL<- function(beta){
  somme=0
  for(i in seq(0, 9644, 4)){
  loga <- log(exp(beta*Yogurtlong$price[(i+1):(i+4)])/
                sum(exp(beta*Yogurtlong$price[(i+1):(i+4)])))%*%
    Yogurtlong$choice[(i+1):(i+4)]
  somme <- loga + somme
  }
return(somme)
}
```

Nous pouvons maintenant essayer de maximiser cette somme des logs vraissemblance en testant plusieurs valeurs de β. Ici, nous testons les valeurs comprises entre 9 et 11, avec un intervale de 0,1 :

```
resLL = c()
for(beta in seq(9, 11, 0.1)){
  resLL = c(resLL, LL(beta))
}
resLL <- as.data.frame(resLL)
# Visualisation de la somme des log vraissemblances en fonction de beta :
ggplot(resLL, aes(x = seq(9, 11, 0.1), y = resLL)) + geom_point() + theme_economist() +
 ggtitle("Log vraissemblance en fonction de Beta") + xlab("Beta") +
  ylab("Somme log vraissemblance") + theme(
    axis.title.x = element_text(size=12, face="bold"),
    axis.title.y = element_text(size=12, face="bold", vjust = 3)) +
  annotate("segment", x = 9.3, xend = 9.85, y = -3289.516, yend = -3289.516,
           colour = "red", size = 2, arrow = arrow()) +
  geom_label(aes (x= 9.2, y = -3289.516, label = "Max"),
             color = "red", size = 4, fontface = "bold")
```



Ici, la valeur de β maximisant notre somme semble être de 9,8. Afin de vérifier si cette valeur est bonne, nous pouvons utiliser la fonction "optimize" de R, effectuant automatiquement ce calcul :

```
optimize(LL, interval = c(7, 13), maximum = T)
```

```
## $maximum
## [1] 9.898559
##
## $objective
```

```
##           [,1]
## [1,] -3289.516
```

Nous pouvons ainsi voir que la valeur maximale de la somme des log vraissemblance est de **-3289.516**, pour une valeur de β optimale de **9.898554**. Celà correspond bien à ce que nous avons pu observer dans notre graphique. Notre fonction est donc opérationnelle.

Estimation du MNL avec les variables "Price" et "Feat" :

Nous avons à présent compris le fonctionnement technique du modèle Multinomial Logit. Nous pouvons alors appliquer notre modèle à l'aide d'une fonction présente sur R. pour celà, nous entrerons les variables **"Price"** et **"Feat"** en tant que variables explicatives.

Plusieurs fonctions permettent d'appliquer le modèle MNL, comme "glm()" ou "mlogit". Nous allons favoriser "mlogit()" car cette fonction est spécifiquement dédiée aux modèles logit :

```
model <- mlogit(choice ~ price + feat, Yogurtlong)
summary(model)
```

```
##
## Call:
## mlogit(formula = choice ~ price + feat, data = Yogurtlong, method = "nr")
##
## Frequencies of alternatives:choice
##   dannon   hiland   weight  yoplait
## 0.402156 0.029436 0.229270 0.339138
##
## nr method
## 6 iterations, 0h:0m:0s
## g'(-H)^-1g = 0.000763
## successive function values within tolerance limits
##
## Coefficients :
##                       Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):hiland   -3.715595   0.145419 -25.5510 < 2.2e-16 ***
## (Intercept):weight   -0.641184   0.054498 -11.7652 < 2.2e-16 ***
## (Intercept):yoplait   0.734571   0.080644   9.1088 < 2.2e-16 ***
## price               -36.658445   2.436607 -15.0449 < 2.2e-16 ***
## feat                  0.491433   0.120063   4.0931 4.256e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -2656.9
## McFadden R^2:  0.062142
## Likelihood ratio test : chisq = 352.09 (p.value = < 2.22e-16)
```

```
AIC(model)
```

```
## [1] 5323.776
```

Observons les coefficients (c'est à dire la valeur β) de nos deux variables "price" et "feat" :

```
model$coefficients
```

```
##  (Intercept):hiland  (Intercept):weight (Intercept):yoplait               price
##          -3.7155950          -0.6411843           0.7345712         -36.6584451
##                feat
##           0.4914334
```

```
## attr(,"names.sup.coef")
## character(0)
## attr(,"fixed")
##  (Intercept):hiland  (Intercept):weight (Intercept):yoplait               price
##               FALSE               FALSE               FALSE               FALSE
##                feat
##               FALSE
## attr(,"sup")
## character(0)
```

Nous pouvons voir que :

- La valeur du coefficient de "Price" est de -36,7
- La valeur du coefficient de "Feat" est de 0.49

Nous pouvons en déduire que **le prix va donc avoir un impact bien plus fort sur le choix du consommateur** lors de l'achat d'un Yaourt.

Changement de l'alternative de référence :

Enfin, essayons de changer la modalité de référence de notre modèle. Par défaut, il s'agit de du choix d'un yaourt de la marque "Dannon". Mais nous pouvons changer cette modalité, par exemple en la remplaçant par la marque "Hiland":

```
Hiland_model <- mlogit(choice ~ price + feat, Yogurtlong, reflevel = "hiland")
summary(Hiland_model)
```

```
##
## Call:
## mlogit(formula = choice ~ price + feat, data = Yogurtlong, reflevel = "hiland",
##     method = "nr")
##
## Frequencies of alternatives:choice
##   hiland   dannon   weight  yoplait
## 0.029436 0.402156 0.229270 0.339138
##
## nr method
## 6 iterations, 0h:0m:0s
## g'(-H)^-1g = 0.000763
## successive function values within tolerance limits
##
## Coefficients :
##                      Estimate Std. Error  z-value  Pr(>|z|)
## (Intercept):dannon    3.71560    0.14542  25.5510 < 2.2e-16 ***
## (Intercept):weight    3.07441    0.14538  21.1468 < 2.2e-16 ***
## (Intercept):yoplait   4.45017    0.18712  23.7827 < 2.2e-16 ***
## price               -36.65845    2.43661 -15.0449 < 2.2e-16 ***
## feat                  0.49143    0.12006   4.0931 4.256e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log-Likelihood: -2656.9
## McFadden R^2:  0.062142
## Likelihood ratio test : chisq = 352.09 (p.value = < 2.22e-16)
```

Ainsi, nous pouvons constater que la valeur des coefficients ne change pas, mais que notre modèle MNL se base désormais sur le choix de la marque "Hiland".