Marketing Models - Assignment 1

E.C. van Groningen

11/22/2020

Load libraries

```
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Load the data

```
data <- load("Assignment_1.RData")
```

Assignment 1

Inspect the data

```r
head(Conjoint_data)
```

```
##   respondent_id profile        form   noapply disinfect bio    price choice
## 1             1       1 Concentrate 200 times       Yes  No 35 cents      1
## 2             1       2      Powder 200 times       Yes  No 35 cents      0
## 3             1       3      Premix 100 times       Yes Yes 49 cents      1
## 4             1       4      Powder 200 times       Yes Yes 49 cents      0
## 5             1       5      Powder  50 times       Yes  No 79 cents      0
## 6             1       6 Concentrate 200 times        No Yes 79 cents      0
```

Estimate the model

```r
# Set baseline for variables in regression
Conjoint_data$form <- relevel(Conjoint_data$form, ref = "Powder")
Conjoint_data$noapply <- relevel(Conjoint_data$noapply, ref = "200 times")
Conjoint_data$disinfect <- relevel(Conjoint_data$disinfect, ref = "No")
Conjoint_data$price <- relevel(Conjoint_data$price, ref = "35 cents")

# Estimate model
model <-
  glm(choice ~ form + noapply + disinfect + bio + price,
      family = "binomial",
      data = Conjoint_data)

# Show results
results <- summary(model)
results
```

```
##
## Call:
## glm(formula = choice ~ form + noapply + disinfect + bio + price,
##     family = "binomial", data = Conjoint_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1680  -0.7913   0.3276   0.8173   2.1374
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         1.1357     0.4662   2.436 0.014859 *
## formConcentrate     0.6482     0.3293   1.968 0.049035 *
## formPremix         -0.2519     0.3347  -0.753 0.451638
## noapply100 times   -0.2335     0.3457  -0.675 0.499416
## noapply50 times    -0.5904     0.3601  -1.639 0.101111
## disinfectYes        1.1143     0.2998   3.717 0.000202 ***
## bioYes              0.2762     0.2862   0.965 0.334477
## price49 cents      -1.3828     0.3609  -3.831 0.000128 ***
## price79 cents      -3.0606     0.3810  -8.034 9.46e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 449.25  on 329  degrees of freedom
## Residual deviance: 348.95  on 321  degrees of freedom
## AIC: 366.95
##
## Number of Fisher Scoring iterations: 4
```

```
# Tabulate results for report
stargazer(model, type = "text", title = "Logistic Regression Model",
          intercept.bottom = FALSE, no.space = TRUE)
```

```
##
## Logistic Regression Model
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                choice
## ------------------------------------------------
## Constant                       1.136**
##                                (0.466)
## formConcentrate                0.648**
##                                (0.329)
## formPremix                     -0.252
##                                (0.335)
## noapply100 times               -0.234
##                                (0.346)
## noapply50 times                -0.590
##                                (0.360)
## disinfectYes                  1.114***
##                                (0.300)
## bioYes                          0.276
##                                (0.286)
## price49 cents                 -1.383***
##                                (0.361)
## price79 cents                 -3.061***
##                                (0.381)
## ------------------------------------------------
## Observations                     330
## Log Likelihood                -174.477
## Akaike Inf. Crit.              366.955
## ================================================
## Note:              *p<0.1; **p<0.05; ***p<0.01
```

Create empty part-worth

```
# Initialize empty part-worth table (type list)
partworth <-
  list("form" = rep(0, 3), "noapply" = rep(0, 3), "disinfect" = rep(0, 2),
       "bio" = rep(0, 2), "price" = rep(0, 3))
```

```r
# Set levels part-worth
names(partworth$form) <- levels(Conjoint_data$form)
names(partworth$noapply) <- levels(Conjoint_data$noapply)
names(partworth$disinfect) <- levels(Conjoint_data$disinfect)
names(partworth$bio) <- levels(Conjoint_data$bio)
names(partworth$price) <- levels(Conjoint_data$price)

# Set insignificant coefficients to zero
coeffs <- results$coefficients[,1]  * (results$coefficients[,4] < .05)
coeffs[1]
```

```
## (Intercept)
##    1.135682
```

```r
# Fill the part-worth table
partworth$form[2:3] <- coeffs[2:3]
partworth$noapply[2:3] <- coeffs[4:5]
partworth$disinfect[2] <- coeffs[6]
partworth$bio[2] <- coeffs[7]
partworth$price[2:3] <- coeffs[8:9]

# Show results
partworth
```

```
## $form
##      Powder Concentrate      Premix
##   0.0000000   0.6481925   0.0000000
##
## $noapply
## 200 times 100 times  50 times
##         0         0         0
##
## $disinfect
##       No      Yes
## 0.000000 1.114257
##
## $bio
##  No Yes
##   0   0
##
## $price
##  35 cents  49 cents  79 cents
##  0.000000 -1.382753 -3.060619
```

Compute probability

```r
prob <-
  exp(coeffs[[1]] + max(partworth$form) + max(partworth$noapply) +
        max(partworth$disinfect) + max(partworth$bio) + max(partworth$price)) /
  (1 + exp(coeffs[[1]] + max(partworth$form) + max(partworth$noapply) +
```

```
            max(partworth$disinfect) + max(partworth$bio) +
            max(partworth$price)))
prob
```

```
## [1] 0.947754
```

Question 2

Question 2a

```
lik_bgnbd <- function(para) {
  # This is the likelihood function with CLV_data
  # Change CLV_data to the names your datasets; the variables should be the same.

  # unpack 4 parameters:
  # r and alpha are the parameters of the gamma distribution of lambda
  # a and b are the parameters of the beta distribution ofr the "death" rate
  r <- para[1]
  alpha <- para[2]
  a <- para[3]
  b <- para[4]

  # unpack three variables:
  # x - # of transactions (frequency)
  # t_x - the most recent time of transaction (recency)
  # T - the end time for the Poisson process
  x <- CLV_data$x
  t_x <- CLV_data$t_x
  T <- CLV_data$T

  # A1 - A4 corresponds to the log of 4 terms in the likelihood function
  # See p.280 of Fader et al. (2005)
  # They are the log of A1-A4 for the log-likelihood
  A1 <- lgamma(r+x) + r*log(alpha) - lgamma(r)
  A2 <- lgamma(a+b) + lgamma(b+x) - lgamma(b) - lgamma(a+b+x)
  A3 <- -(r+x)*log(alpha+T)

  A4 <- rep(0,length(x))
  idx <- x>0
  if (sum(idx)>0) {
    A4[idx] <- log(a/(b+x[idx]-1)) - (r+x[idx])*log(alpha+t_x[idx])
  }

  lh <- sum(A1+A2+log(exp(A3)+idx*exp(A4)))

  # To return the minus of likelihood for minimization
  # maximize likelihood = minimize negative likelihood
  return(-lh)

}
```

```
pred_Y <- function(para, H, t) {
  # Given estimated parameters, for a customer with history {x,t_x,T}, to
  # predict her # of transactions in t time.
  # Please see Equation (10) of Fader et al. (2005), p.279;
  # This function uses a package "hypergeo" for function evaluation;
  # As evaluation of 2F1(.) is somewhat costly, this function is to accommodate
  # only 1 customer;
  # You can extend the function to calculate for many customers by iterations or
  # use lapply(.);

# The function takes three input:
  # para - a vector of the 4 parameters;
  # H - a data.frame with only 1 row (for 1 customer) and three variables
  # {x,t_x,T};
  # t - how long into the future?

  # unpack 4 parameters:
  # r and alpha are the parameters of the gamma distribution of lambda
  # a and b are the parameters of the beta distribution of the "death" rate
  r <- para[1]
  alpha <- para[2]
  a <- para[3]
  b <- para[4]

  # unpack three variables:
  # x - # of transactions (frequency)
  # t_x - the most recent time of transaction (recency)
  # T - the end time for the Poisson process
  x <- H$x
  t_x <- H$t_x
  T <- H$T

  # the term of the Gaussian hypergeometric function 2F1(.)
  hg <- hypergeo::hypergeo(r+x,b+x,a+b+x-1,t/(alpha+T+t))
  hg <- Re(hg) # hg is a real number in the complex format Re(.) makes it real.

  # the numerator term, given hg
  Y1 <- (a+b+x-1)/(a-1)*(1-(((alpha+T)/(alpha+T+t))^(r+x))*hg)
  Y2 <- 1
  if (x>0) {
    Y2 <- Y2 + a/(b+x-1)*(((alpha+T)/(alpha+t_x))^(r+x))
  }

  return(Y1/Y2)
}
```

Estimating the Parameters of the BG/NBD model

```
# to specify the constraint matrix
ui <- diag(4)
ci <- rep(0,4)
```

```r
# para: the starting values; a vector with 4 elements corresponding to
# (r,alpha,a,b).
# you can change this to other values or test different values.
para <- rep(.01,4)

# to run the constrained optimization
results <- constrOptim(para, lik_bgnbd, NULL, ui, ci)

# to get the estimated parameters in the order of (r,alpha,a,b)
para <- results$par
names(para) <- c("r","alpha","a","b")

stargazer(para, summary=FALSE, title="Parameters GB/NBD model", type = "text")
```

```
##
## Parameters GB/NBD model
## ======================
## r     alpha   a     b
## ----------------------
## 0.232 8.957 1.194 3.191
## ----------------------
```

Predict the No. of Transactions

```r
# suppose we want to predict next year and set t = 52
t <- 30

preds <-  c()
for(i in 1:5) {
  # let's predict the first customer in the data
  H <- CLV_data[i,]

  # to obtain the prediction
  Yhat <- pred_Y(para,H,t)
  preds <- append(preds, as.numeric(Yhat))
}
preds <- round(preds, 3)

res <- tibble("gamer ID" = c(rep(1:5)), "number of pruchase" = preds)

stargazer(res, summary=FALSE, type="text",
          title="Predict # of purchases in the next month (t=30)")
```

```
##
## Predict # of purchases in the next month (t=30)
## ============================
##   gamer ID number of pruchase
## ----------------------------
## 1    1           0.178
## 2    2           0.243
## 3    3           0.087
```

```
## 4    4           0.753
## 5    5           0.089
## ----------------------------
```