Choice models - under the hood

CHOICE MODELING FOR MARKETING IN R

Elea McDonnell Feit

Assistant Professor of Marketing, Drexel University

Multinomial logit model

```
v1 <- alpha * 4 + beta * 100
v2 <- alpha * 5 + beta * 150
v3 <- alpha * 2 + beta * 175
```

```
u1 <- v1 + error1
u2 <- v2 + error2
u3 <- v3 + error3
```

```
choice <- which.max(c(u1, u2, u3))
```

```
p1 <- exp(v1) / ( exp(v1) + exp(v2) + exp(v3) )
p2 <- exp(v2) / ( exp(v1) + exp(v2) + exp(v3) )
p3 <- exp(v3) / ( exp(v1) + exp(v2) + exp(v3) )
```

Estimating a multinomial logit model with mlogit()

```
m1 <- mlogit(choice ~ 0 + seat + price, data = sportscar, print.level = 3)
```

```
Initial value of the function : 2197.22457733622
iteration 1, step = 1, lnL = 1918.11271719, chi2 = 534.02624615
            seat   price
param      0.100  -0.151
gradient -24.028 296.443
 ---------------------------------------
iteration 2, step = 1, lnL = 1915.29601553, chi2 = 5.51592264
           seat  price
param     0.114 -0.168
gradient -0.846  9.422
 ---------------------------------------
iteration 3, step = 1, lnL = 1915.29299604, chi2 = 0.00603424
           seat  price
param     0.114 -0.169
gradient -0.001  0.011
```

```
summary(m1)
```

```
Call:
mlogit(formula = choice ~ 0 + seat + price, data = sportscar,
    print.level = 3, method = "nr")
Frequencies of alternatives:
    1     2     3
0.328 0.327 0.345
nr method
4 iterations, 0h:0m:0s
g'(-H)^-1g = 0.00603
successive function values within tolerance limits
Coefficients :
        Estimate Std. Error  z-value  Pr(>|z|)
seat   0.1143487  0.0234195   4.8826 1.047e-06 ***
price -0.1687046  0.0079224 -21.2947 < 2.2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Log-Likelihood: -1915.3
```

mlogit.data objects

```
sportscar <- mlogit.data(sportscar.df, shape = "long",
                         choice = "choice",
                         varying = 5:8, alt.var = "alt")
```

Let's fit some choice models to the chocolate data!

CHOICE MODELING FOR MARKETING IN R

Interpreting choice model parameters

CHOICE MODELING FOR MARKETING IN R

Elea McDonnel Feit

Assistant Professor of Marketing, Drexel University

Multinomial logit model parameters

```
m1 <- mlogit(choice ~ 0 + seat + price, data=sportscar)
summary(m1)
```

```
...

Coefficients :
        Estimate Std. Error  z-value  Pr(>|z|)
seat   0.1143487  0.0234195   4.8826 1.047e-06 ***
price -0.1687046  0.0079224 -21.2947 < 2.2e-16 ***

...
```

Multinomial logit model

```
alpha <- 0.1143487
beta <- -0.1687046
v1 <- alpha * seat1 + beta * price1
v2 <- alpha * seat2 + beta * price2
v3 <- alpha * seat3 + beta * price3

p1 <- exp(v1) / ( exp(v1) + exp(v2) + exp(v3) )
p2 <- exp(v2) / ( exp(v1) + exp(v2) + exp(v3) )
p3 <- exp(v3) / ( exp(v1) + exp(v2) + exp(v3) )
```

Using factors as predictors

```
m2 <- mlogit(choice ~ 0 + seat + price + trans + convert,
             data = sportscar)
summary(m2)
```

```
...

Coefficients :
              Estimate Std. Error  z-value  Pr(>|z|)
seat         0.1201748  0.0248393   4.8381 1.311e-06 ***
price       -0.1895992  0.0086355 -21.9559 < 2.2e-16 ***
transmanual -1.2122404  0.0662971 -18.2850 < 2.2e-16 ***
convertyes   0.1932630  0.0618676   3.1238  0.001785 **
...
```

```
head(model.matrix(m2))
```

```
    seat price transmanual convertyes
1.1    2    35           1          1
1.2    5    40           0          0
1.3    5    30           0          0
2.1    5    35           1          0
2.2    2    30           1          0
2.3    4    35           0          0
```

```
head(sportscar)
```

```
    resp_id ques alt segment seat  trans convert price choice
1.1       1    1   1   basic    2 manual     yes    35  FALSE
1.2       1    1   2   basic    5   auto      no    40  FALSE
1.3       1    1   3   basic    5   auto      no    30   TRUE
2.1       1    2   1   basic    5 manual      no    35  FALSE
2.2       1    2   2   basic    2 manual      no    30   TRUE
2.3       1    2   3   basic    4   auto      no    35  FALSE
```

Interpreting coefficients for factors

```
m2 <- mlogit(choice ~ 0 + seat + price + trans + convert,
             data = sportscar)
summary(m2)
```

```
...
Coefficients :
              Estimate Std. Error  z-value  Pr(>|z|)
seat         0.1201748  0.0248393   4.8381 1.311e-06 ***
price       -0.1895992  0.0086355 -21.9559 < 2.2e-16 ***
transmanual -1.2122404  0.0662971 -18.2850 < 2.2e-16 ***
convertyes   0.1932630  0.0618676   3.1238  0.001785 **
...
```

Treating numeric predictors as factors

```
sportscar$seat <- as.factor(sportscar$seat)
m3 <- mlogit(choice ~ 0 + seat + trans + convert + price,
             data = sportscar)
summary(m3)
```

```
Coefficients :
              Estimate Std. Error  z-value  Pr(>|z|)
seat4       -0.0193861  0.0759029  -0.2554  0.798409
seat5        0.4245449  0.0752808   5.6395 1.706e-08 ***
transmanual -1.2178833  0.0665276 -18.3064 < 2.2e-16 ***
convertyes   0.2008115  0.0620854   3.2344  0.001219 **
price       -0.1907023  0.0086739 -21.9859 < 2.2e-16 ***
```

Willingness to pay (WTP)

```
coef(m3)
```

```
seat4       seat5 transmanual  convertyes       price
-0.01938614  0.42454491 -1.21788327  0.20081149 -0.19070229
```

```
coef(m3) / -coef(m3)[5]
```

```
seat4       seat5 transmanual  convertyes       price
 -0.1016566   2.2262181  -6.3863063   1.0530103  -1.0000000
```

Let's interpret the parameters of the chocolate model!

CHOICE MODELING FOR MARKETING IN R

Intercepts and interactions

CHOICE MODELING FOR MARKETING IN R

Elea McDonnell Feit

Assistant Professor of Marketing, Drexel University

Why no intercepts?

```
v1 <- alpha * seat1 + beta * price1
v2 <- alpha * seat2 + beta * price2
v3 <- alpha * seat3 + beta * price3
p1 <- exp(v1) / ( exp(v1) + exp(v2) + exp(v3) )
p2 <- exp(v2) / ( exp(v1) + exp(v2) + exp(v3) )
p3 <- exp(v3) / ( exp(v1) + exp(v2) + exp(v3) )
```

Example with an intercept

```
intercept <- 5
u1 <- intercept + 1.5
u2 <- intercept +  0
u3 <- intercept + -1.5
p1 <- exp(u1) / ( exp(u1) + exp(u2) + exp(u3) )
p2 <- exp(u2) / ( exp(u1) + exp(u2) + exp(u3) )
p3 <- exp(u3) / ( exp(u1) + exp(u2) + exp(u3) )
c(p1, p2, p3)
```

```
[1] 0.78559703 0.17529039 0.03911257
```

Example with an intercept

```
intercept <- 0
u1 <- intercept + 1.5
u2 <- intercept +  0
u3 <- intercept + -1.5
p1 <- exp(u1) / ( exp(u1) + exp(u2) + exp(u3) )
p2 <- exp(u2) / ( exp(u1) + exp(u2) + exp(u3) )
p3 <- exp(u3) / ( exp(u1) + exp(u2) + exp(u3) )
c(p1, p2, p3)
```

```
0.78559703 0.17529039 0.03911257
```

Interactions

```
u1 <- alpha * seat1 + beta * price1 + gamma * seat1 * price1
u2 <- alpha * seat2 + beta * price2 + gamma * seat2 * price2
u3 <- alpha * seat3 + beta * price3 + gamma * seat3 * price3
```

Putting interactions into formulas

```
m4 <- mlogit(choice ~ 0 + seat + trans + convert + price + trans:price,
             data = sportscar)
```

```
m4 <- mlogit(choice ~ 0 + seat + convert + trans * price,
             data = sportscar)
```

Interpreting interactions

```
summary(m4)
```

```
...
Coefficients :
                          Estimate Std. Error  z-value  Pr(>|z|)
seat4                   -0.0202794  0.0759464  -0.2670  0.789452
seat5                    0.4255033  0.0752777   5.6524 1.582e-08 ***
transmanual             -1.1437766  0.0926133 -12.3500 < 2.2e-16 ***
convertyes               0.2621652  0.0821798   3.1901  0.001422 **
price                   -0.1908102  0.0086784 -21.9869 < 2.2e-16 ***
transmanual:convertyes -0.1444570  0.1265649  -1.1414  0.253717
...
```

```
summary(sportscar$segment)
```

```
basic   fun racer
 3840  1530   630
```

```
m5 <- mlogit(choice ~ 0 + seat + convert + trans + price:segment,
                     data=sportscar)
summary(m5)
```

```
...
Coefficients :
                   Estimate Std. Error  z-value  Pr(>|z|)
seat4             -0.016206   0.076170  -0.2128  0.831511
seat5              0.426851   0.075682   5.6401 1.700e-08 ***
convertyes         0.200792   0.062343   3.2207  0.001279 **
transmanual       -1.228724   0.066893 -18.3686 < 2.2e-16 ***
price:segmentbasic -0.228245   0.011483 -19.8771 < 2.2e-16 ***
price:segmentfun   -0.133885   0.015677  -8.5405 < 2.2e-16 ***
price:segmentracer -0.132417   0.023398  -5.6594 1.519e-08 ***
...
```

Let's try some interactions in the chocolate data.

CHOICE MODELING FOR MARKETING IN R

Predicting shares

CHOICE MODELING FOR MARKETING IN R

Elea McDonnell Feit

Assistant Professor of Marketing, Drexel University

Multinomial logit model

```
v1 <- alpha * seat1 + beta * price1
v2 <- alpha * seat2 + beta * price2
v3 <- alpha * seat3 + beta * price3
p1 <- exp(v1) / ( exp(v1) + exp(v2) + exp(v3) )
p2 <- exp(v2) / ( exp(v1) + exp(v2) + exp(v3) )
p3 <- exp(v3) / ( exp(v1) + exp(v2) + exp(v3) )
```

Data frame for new products

```
price <- c(35, 30)
seat <- factor(c(2, 2), levels = c(2,4,5))
trans <- factor(c("manual", "auto"), levels = c("auto", "manual"))
convert <- factor(c("no", "no"), levels = c("no", "yes"))
segment <- factor(c("basic", "basic"), levels = c("basic",
                                                  "fun",
                                                  "racer"))
prod <- data.frame(seat, trans, convert, price, segment)
prod
```

```
  seat  trans convert price segment
1    2 manual      no    35     basic
2    2   auto      no    30     basic
```

```
m5 <- mlogit(choice ~ 0 + seat + convert + trans + price:segment,
             data = sportscar)
prod.coded <- model.matrix(update(m5$formula, 0 ~ .), data = prod)[, -1]
prod.coded
```

```
  seat4 seat5 convertyes transmanual price:segmentbasic
1     0     0          0           1                  35
2     0     0          0           0                  30
  price:segmentfun price:segmentracer
1                0                  0
2                0                  0
```

```
v <- prod.coded %*% m5$coef
v
```

```
       [,1]
1 -9.217304
2 -6.847354
```

Code for predicting shares

```
p <- exp(v) / sum(exp(v))
cbind(p, prod)
```

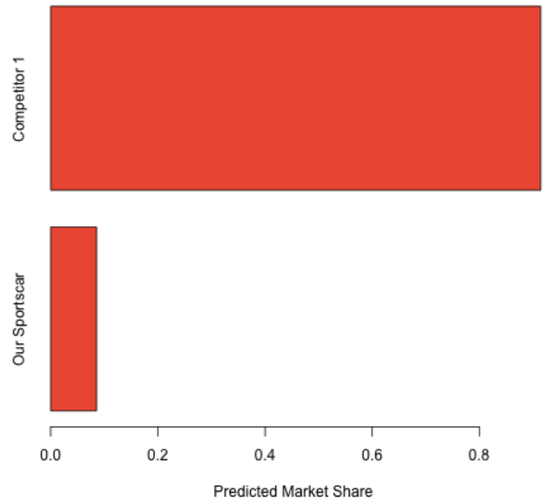
```
           p seat  trans convert price segment
1 0.08549309    2 manual      no    35   basic
2 0.91450691    2   auto      no    30   basic
```

A function for share prediction

```
predict_mnl <- function(model, products) {
  data.model <- model.matrix(update(model$formula, 0 ~ .),
                             data = products)[, -1]
  utility <- data.model %*% model$coef
  share <- exp(utility) / sum(exp(utility))
  cbind(share, products)
}
```

Plotting shares

```
shares <- predict_mnl(m5, products)
barplot(shares$share, horiz = TRUE, col = "tomato2",
        xlab = "Predicted Market Share",
        names.arg = c("Our Sportscar", "Competitor 1"))
```



Let's predict some shares for sports cars and chocolate bars!

CHOICE MODELING FOR MARKETING IN R