data = read.csv(file = "M:/A Master of Science in Marketing Sciences/Mathematical Models in Marketing (Kohli)/latent/data\_ipad.csv",head = TRUE)

attach(data)

summary(data)

## Modelling Step 1: Set up Training Set and Dev Set

random\_factor = sample(1:15,137,replace = TRUE)

determinant\_factor = (1:137)\*15 - 15

## The index for Dev Set is the summation of Random and Deterministic.

index = random\_factor + determinant\_factor

dev = data[which(choice\_set\_id %in% index) ,]

train = data[which(!(choice\_set\_id %in% index)) ,]

## Model 1: Fit an aggregate model without any segment

## In the training set,

X1\_train = subset(train, alternative\_id\_in\_set == 1)

X2\_train = subset(train, alternative\_id\_in\_set == 2)

X3\_train = subset(train, alternative\_id\_in\_set == 3)

## In the test set,``

X1\_test = subset(dev, alternative\_id\_in\_set == 1)

X2\_test = subset(dev, alternative\_id\_in\_set == 2)

X3\_test = subset(dev, alternative\_id\_in\_set == 3)

## The dependent variable

train\_choice = X1\_train$choice

test\_choice = X1\_test$choice

train\_set = cbind(X1\_train[,5:22],X2\_train[,5:22],X3\_train[,5:22])

test\_set = cbind(X1\_test[,5:22],X2\_test[,5:22],X3\_test[,5:22])

## Multi-Nomial Logit Estimation

par = rnorm(18)

N = 1918

ll <- function(beta)

{

res = 0

M1 = as.matrix(train\_set)[,1:18] %\*% beta

M2 = as.matrix(train\_set)[,19:36] %\*% beta

M3 = as.matrix(train\_set)[,37:54] %\*% beta

M = cbind(exp(M1),exp(M2),exp(M3))

M = M / rowSums(M)

MP = M[cbind(seq(1,length(train\_choice)),train\_choice)]

res = res + sum(-log(MP))

return (res)

}

ML = nlm(ll,par,hessian = TRUE)

X1\_predict = as.matrix(X1\_test)[,5:22] %\*% ML$estimate

X2\_predict = as.matrix(X2\_test)[,5:22] %\*% ML$estimate

X3\_predict = as.matrix(X3\_test)[,5:22] %\*% ML$estimate

prediction = max.col(cbind(X1\_predict,X2\_predict,X3\_predict))

## The performance of the model without segments is evaluated by the proportion of correct predictions.

performance\_0 = sum(prediction == test\_choice)/length(test\_choice)

Estimate SE Tvalue minusll

[1,] 0.23 0.10 2.22 3595.66

[2,] 1.01 0.10 10.30 3595.66

[3,] 0.36 0.10 3.80 3595.66

[4,] 0.17 0.10 1.70 3595.66

[5,] 0.20 0.09 2.35 3595.66

[6,] 0.45 0.08 5.32 3595.66

[7,] 0.29 0.09 3.34 3595.66

[8,] 0.22 0.09 2.48 3595.66

[9,] 0.59 0.08 6.99 3595.66

[10,] 0.62 0.09 6.83 3595.66

[11,] 0.32 0.07 4.55 3595.66

[12,] 0.64 0.07 9.48 3595.66

[13,] 0.14 0.07 2.10 3595.66

[14,] 0.14 0.07 2.09 3595.66

[15,] -0.32 0.09 -3.70 3595.66

[16,] -0.75 0.09 -8.02 3595.66

[17,] -1.28 0.10 -12.21 3595.66

[18,] -1.78 0.11 -16.34 3595.66

Performance Level is 0.5182482.

Question: How to use the holdout data to validate model especially with the latent-class models?

In the test set, do we need to calculate the probability for each individual to choose each alternative and then do sampling?

Estimate SE Tvalue minusll

[1,] 0.23 0.08 3.02 3611.46

[2,] 0.99 0.09 11.01 3611.46

[3,] 0.34 0.09 3.84 3611.46

[4,] 0.12 0.09 1.32 3611.46

[5,] 0.21 0.08 2.68 3611.46

[6,] 0.44 0.08 5.66 3611.46

[7,] 0.33 0.07 4.72 3611.46

[8,] 0.19 0.09 2.15 3611.46

[9,] 0.57 0.08 6.72 3611.46

[10,] 0.57 0.09 6.30 3611.46

[11,] 0.33 0.07 4.79 3611.46

[12,] 0.64 0.07 9.52 3611.46

[13,] 0.11 0.06 1.78 3611.46

[14,] 0.13 0.07 1.99 3611.46

[15,] -0.28 0.09 -3.28 3611.46

[16,] -0.72 0.09 -7.65 3611.46

[17,] -1.27 0.10 -12.18 3611.46

[18,] -1.71 0.11 -15.83 3611.46

[19,] 0.56 NaN NaN 3611.46

> performance\_0

[1] 0.5547445

Estimate SE Tvalue minusll

[1,] 0.23 0.00 98323064.26 3611.46

[2,] 0.99 0.08 12.01 3611.46

[3,] 0.34 0.08 4.32 3611.46

[4,] 0.12 0.08 1.45 3611.46

[5,] 0.21 0.09 2.47 3611.46

[6,] 0.44 0.08 5.30 3611.46

[7,] 0.33 0.09 3.76 3611.46

[8,] 0.19 0.09 2.15 3611.46

[9,] 0.57 0.08 6.72 3611.46

[10,] 0.57 0.09 6.29 3611.46

[11,] 0.33 0.07 4.82 3611.46

[12,] 0.64 0.07 9.57 3611.46

[13,] 0.11 0.07 1.62 3611.46

[14,] 0.13 0.07 1.97 3611.46

[15,] -0.28 0.09 -3.30 3611.46

[16,] -0.72 0.09 -7.65 3611.46

[17,] -1.27 0.10 -12.18 3611.46

[18,] -1.71 0.11 -15.85 3611.46

[19,] -0.52 NaN NaN 3611.46

performance\_0

[1] 0.5547445

Estimate SE Tvalue minusll

[1,] 0.23 0.00 98322493.12 3611.46

[2,] 0.99 0.08 12.01 3611.46

[3,] 0.34 0.08 4.32 3611.46

[4,] 0.12 0.08 1.45 3611.46

[5,] 0.21 0.09 2.47 3611.46

[6,] 0.44 0.08 5.30 3611.46

[7,] 0.33 0.09 3.76 3611.46

[8,] 0.19 0.09 2.15 3611.46

[9,] 0.57 0.08 6.72 3611.46

[10,] 0.57 0.09 6.29 3611.46

[11,] 0.33 0.07 4.82 3611.46

[12,] 0.64 0.07 9.57 3611.46

[13,] 0.11 0.07 1.62 3611.46

[14,] 0.13 0.07 1.97 3611.46

[15,] -0.28 0.09 -3.30 3611.46

[16,] -0.72 0.09 -7.65 3611.46

[17,] -1.27 0.10 -12.18 3611.46

[18,] -1.71 0.11 -15.85 3611.46

[19,] 0.93 NaN NaN 3611.46

>

> performance\_0

[1] 0.5547445

Estimate SE Tvalue minusll

[1,] 0.23 0.10 2.24 3611.46

[2,] 0.99 0.10 10.15 3611.46

[3,] 0.34 0.10 3.55 3611.46

[4,] 0.12 0.10 1.25 3611.46

[5,] 0.21 0.09 2.47 3611.46

[6,] 0.44 0.08 5.32 3611.46

[7,] 0.33 0.09 3.78 3611.46

[8,] 0.19 0.09 2.16 3611.46

[9,] 0.57 0.08 6.72 3611.46

[10,] 0.57 0.09 6.29 3611.46

[11,] 0.33 0.07 4.79 3611.46

[12,] 0.64 0.07 9.53 3611.46

[13,] 0.11 0.07 1.62 3611.46

[14,] 0.13 0.07 1.97 3611.46

[15,] -0.28 0.08 -3.69 3611.46

[16,] -0.72 0.08 -8.63 3611.46

[17,] -1.27 0.00 -Inf 3611.46

[18,] -1.71 0.10 -17.67 3611.46

[19,] 1.25 NaN NaN 3611.46

>

> performance\_0

[1] 0.5547445

Estimate SE Tvalue minusll

[1,] 0.23 0.09 2.53 3611.46

[2,] 0.99 0.08 12.48 3611.46

[3,] 0.34 0.09 4.00 3611.46

[4,] 0.12 0.09 1.34 3611.46

[5,] 0.21 0.09 2.47 3611.46

[6,] 0.44 0.08 5.28 3611.46

[7,] 0.33 0.09 3.76 3611.46

[8,] 0.19 0.09 2.16 3611.46

[9,] 0.57 0.08 6.73 3611.46

[10,] 0.57 0.09 6.30 3611.46

[11,] 0.33 0.06 5.24 3611.46

[12,] 0.64 0.07 9.74 3611.46

[13,] 0.11 0.07 1.62 3611.46

[14,] 0.13 0.07 1.97 3611.46

[15,] -0.28 0.08 -3.44 3611.46

[16,] -0.72 0.09 -7.85 3611.46

[17,] -1.27 0.10 -12.61 3611.46

[18,] -1.71 0.09 -18.87 3611.46

[19,] 0.91 NaN NaN 3611.46

performance\_0

[1] 0.5547445