Multinomial Logit Model (MNL)

## 1 Introduction

This document elaborates on the steps done to run Multinomial Logistic Regression on the results of the Stated Preference Survey. Multinomial Logistic Regression serves to predict an individual's mode choice (public bus, shared bicycles/PMDs, or Dalphin) using inputs such as the individual's demographics, mode specific variables and non-mode specific variables.

## 2 The Data

First, we read in the data from the stated preference survey which is in the CSV format (Comma Separated Values).

modechoice <- read.csv("ModeChoice.csv")

Looking at the structure of the dataset str(), it contains 1000 observations of 40 variables.

str(modechoice)

## 3 Predicting "NA" Incomes with Linear Regression

Some of the entries in the Income column are NA. To fill this missing information, we used linear regression to predict their values.

We separate the dataset into two: one with stated incomes, and another which has NA for Income.

incomenotNA <- subset(modechoice, modechoice$Income != "NA")  
incomeNA <- subset(modechoice,is.na(modechoice$Income))

There are 752 observations with stated incomes and 248 observations where income is NA.

#### 3.1 Creating Training and Test Sets

We now create training and test sets from the observations where incomes are stated (incomenotNA). The caTools package is used.

library(caTools)

We set the random seed so that the results will be replicable.

set.seed(1)

We use the sample.split() function to obtain the training and test sets with a 65/35 split ratio. Both income training and test sets maintained the same proportion of Age variable as in the known income dataset, incomenotNA.

split <- sample.split(incomenotNA$Age, SplitRatio = 0.65)  
  
incometraining <- subset(incomenotNA,split==TRUE)  
incometest <- subset(incomenotNA,split==FALSE)

#### 3.2 Creating a Linear Regression Model

We now create a linear regression model with the training set. The first model includes all demographic information of the individual:

|  |  |
| --- | --- |
| Variable | Description |
| Age | The individual's age group |
| Gender | The individual's gender |
| HouseholdType | Household type |
| HouseholdSize | Number of people in the household |
| Residential | Residential location of the household |
| Cars | The number of cars in the household |
| Concession | The type of concession pass, if any |

income.model1 <- lm(Income ~ Age + Gender + HouseholdType +   
 HouseholdSize + Residential + Cars +   
 Concession,  
 data = incometraining)  
  
summary(income.model1)

##   
## Call:  
## lm(formula = Income ~ Age + Gender + HouseholdType + HouseholdSize +   
## Residential + Cars + Concession, data = incometraining)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.4952 -0.9929 -0.1726 0.9612 2.9008   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.531995 0.484056 1.099 0.272303   
## Age 0.797357 0.048075 16.586 < 2e-16 \*\*\*  
## Gender -0.360214 0.122404 -2.943 0.003410 \*\*   
## HouseholdType 0.011665 0.045024 0.259 0.795689   
## HouseholdSize -0.293817 0.077210 -3.805 0.000160 \*\*\*  
## Residential 0.003033 0.009273 0.327 0.743766   
## Cars 0.354159 0.101643 3.484 0.000539 \*\*\*  
## Concession 0.145744 0.079579 1.831 0.067655 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.317 on 480 degrees of freedom  
## Multiple R-squared: 0.4055, Adjusted R-squared: 0.3968   
## F-statistic: 46.77 on 7 and 480 DF, p-value: < 2.2e-16

The model has an adjusted value of 0.39682. HouseholdType and Residential are not significant at the 90% significance level.

#### 3.3 Removing HouseholdType and Residential

We remove the insignificant variables from the model.

income.model2 <- lm(Income ~ Age + Gender +HouseholdSize + Cars + Concession,  
 data=incometraining)  
  
summary(income.model2)

##   
## Call:  
## lm(formula = Income ~ Age + Gender + HouseholdSize + Cars + Concession,   
## data = incometraining)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.5561 -0.9873 -0.1662 0.9926 2.9186   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.64785 0.35692 1.815 0.070124 .   
## Age 0.79474 0.04702 16.901 < 2e-16 \*\*\*  
## Gender -0.35916 0.12124 -2.962 0.003205 \*\*   
## HouseholdSize -0.30061 0.07509 -4.004 7.22e-05 \*\*\*  
## Cars 0.36404 0.09577 3.801 0.000163 \*\*\*  
## Concession 0.15273 0.07716 1.979 0.048336 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.315 on 482 degrees of freedom  
## Multiple R-squared: 0.4053, Adjusted R-squared: 0.3991   
## F-statistic: 65.69 on 5 and 482 DF, p-value: < 2.2e-16

This model has an increased adjusted value of 0.39911. Concession and (Intercept) are the least significant variables.

#### 3.4 Variable Selection

Several other combinations of variables were tested. The combination of variables which gave the highest adjusted was chosen.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Age | Gender | HouseholdType | HouseholdSize | Residential | Cars | Concession | Adjusted |
| 1 |  |  |  |  |  |  |  | 0.39682 |
| **2** |  |  |  |  |  |  |  | **0.39911** |
| 3 |  |  |  |  |  |  |  | 0.39548 |
| 4 |  |  |  |  |  |  |  | 0.39794 |
| 5 |  |  |  |  |  |  |  | 0.39799 |

Model 2 has the highest adjusted . We now evaluate this model on the test set by calculating the test . This is given by the formula:

where is the Sum of Squared Errors and is the Sum of Squared Errors (Total).

incomeprediction <- predict(income.model2, newdata=incometest)  
sse <- sum((incomeprediction-incometest$Income)^2)  
sst <- sum((incometest$Income-mean(incometraining$Income))^2)

The test for model 2 is 0.37408. We now use model 2 to predict the NA entries in the dataset.

NAincomeprediction <- predict(income.model2, newdata=incomeNA)   
  
modechoice$Income <- ifelse(is.na(modechoice$Income),  
 NAincomeprediction,  
 modechoice$Income)

## 4 Fitting the MNL model

Now that all entries in the dataset have no NA values, we can proceed to conduct Multinomial Logistic Regression.

#### 4.1 Generating Training and Test Sets

Each row in the dataset corresponds to a choice task of a particular individual. We randomly split the 125 individuals into the training or test set with a 70/30 split ratio, while keeping all 8 choice tasks intact.

set.seed(0)  
mask = rep(x = sample.split(rep(1, 125), SplitRatio = 0.7),  
 each = 8)  
modechoice.training = subset(modechoice, mask==TRUE)  
modechoice.test = subset(modechoice, mask==FALSE)

The table below shows the first 16 rows of the training set. All 8 choice tasks of an individual are preserved. Case refers to the individual, while Task refers to the choice task.

kable(modechoice.training[1:16,2:6])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Case | Task | weather1 | weather2 | weather3 |
| 9 | 2 | 1 | 0 | 0 | 0 |
| 10 | 2 | 2 | 0 | 0 | 0 |
| 11 | 2 | 3 | 0 | 0 | 0 |
| 12 | 2 | 4 | 0 | 0 | 0 |
| 13 | 2 | 5 | 1 | 1 | 1 |
| 14 | 2 | 6 | 1 | 1 | 1 |
| 15 | 2 | 7 | 1 | 1 | 1 |
| 16 | 2 | 8 | 1 | 1 | 1 |
| 17 | 3 | 1 | 0 | 0 | 0 |
| 18 | 3 | 2 | 0 | 0 | 0 |
| 19 | 3 | 3 | 0 | 0 | 0 |
| 20 | 3 | 4 | 0 | 0 | 0 |
| 21 | 3 | 5 | 1 | 1 | 1 |
| 22 | 3 | 6 | 1 | 1 | 1 |
| 23 | 3 | 7 | 1 | 1 | 1 |
| 24 | 3 | 8 | 1 | 1 | 1 |

#### 4.2 Creating an mlogit() Object

We use the mlogit package which will help us conduct Multinomial Logistic Regression.

library(mlogit)

S <- mlogit.data(data = modechoice.training,  
 shape="wide",  
 choice="A",  
 varying=c(4:18),  
 sep="",  
 alt.levels=c("A.bus","A.cyclingPMD","A.dalphin"),  
 id.var="Case")

The mlogit.data() function shapes the data into a suitable form for the use of the mlogit() function. There are several parameters to be defined in this function:

* data specifies the dataset to be used. Here, we use the training set created previously.
* varying refers to the columns that correspond to the attributes of the choice.
* alt.levels specifies a list of the choices available (Public bus, Cycling/PMDs, or Dalphin).

#### 4.3 Calculating AIC

The Akaike Information Criterion (AIC) is a measure of the quality of fit of a model which also takes model complexity into account. A lower AIC value is more desirable. The formula is given by:

where is the Log-likelihood of the estimated model and is the number of coefficients to be estimated (including the intercept).

For convenience, we define a function calc.AIC() that takes in a model and returns its AIC value.

calc.AIC <- function (model) {  
 return(-2\*model$logLik[1] + 2\*length(model$coefficients))  
}

#### 4.4 Variable Selection

There are three broad types of variable selection:

* Forward: Add one variable at a time. Remove this variable if it is not significant. If significant, add another variable.
* Backward: Add all variables into model. Remove least significant variable each time.
* Stepwise: Combination of the forward and backward selection techniques. After each step in which a variable was added, all variables in the model are checked to see if their significance has been reduced below the threshold significance level. If a nonsignificant variable is found, it is removed from the model. A variable can be added or removed from the previous model.

The stepwise variable selection was executed for its thoroughness.

model

weather

fare

wait

ivtt

access

Age

Gender

Income

HouseholdType

HouseholdSize

Residential

Cars

Concession

AIC Value

1

O

1357.116

2

O

O

1333.944

3

O

O

O

1334.169

4

O

O

O

O

1335.254

Model 2 the better model as it has the lowest AIC. We now build upon Model 2 by including individual-specific attributes.

model

weather

fare

wait

ivtt

access

Age

Gender

Income

HouseholdType

HouseholdSize

Residential

Cars

Concession

AIC Value

5

O

O

O

1326.97

6

O

O

O

O

1328.847

7

O

O

O

O

1320.019

8

O

O

O

O

O

1323.093

9

O

O

O

O

1328.914

10

O

O

O

O

O

1324.68

11

O

O

O

O

O

1318.375

12

O

O

O

O

O

O

1319.543

13

O

O

O

O

O

O

1318.999

Now we try adding the other mode-specific variables that we removed just now.

model

weather

fare

wait

ivtt

access

Age

Gender

Income

HouseholdType

HouseholdSize

Residential

Cars

Concession

AIC Value

14

O

O

O

O

O

O

O

1319.172

15

O

O

O

O

O

O

O

O

1320.237

16

O

O

O

O

O

O

O

1320.235

Model 11 is the best as it has the lowest AIC and highest proportion of significant coefficients. Now we build upon Model 4 by including individual-specific attributes.

model

weather

fare

wait

ivtt

access

Age

Gender

Income

HouseholdType

HouseholdSize

Residential

Cars

Concession

AIC Value

17

O

O

O

O

O

1328.258

18

O

O

O

O

O

O

1330.134

19

O

O

O

O

O

O

O

1322.806

20

O

O

O

O

O

O

O

O

1325.969

21

O

O

O

O

O

O

O

1332.142

22

O

O

O

O

O

O

O

O

1311.476

23

O

O

O

O

O

O

O

O

O

1308.912

Overall, Model 23 is the best model with the lowest AIC value.

This is the full table of all the models constructed.

model

weather

fare

wait

ivtt

access

Age

Gender

Income

HouseholdType

HouseholdSize

Residential

Cars

Concession

AIC Value

1

O

1357.116

2

O

O

1333.944

3

O

O

O

1334.169

4

O

O

O

O

1335.254

5

O

O

O

1326.97

6

O

O

O

O

1328.847

7

O

O

O

O

1320.019

8

O

O

O

O

O

1323.093

9

O

O

O

O

1328.914

10

O

O

O

O

O

1324.68

11

O

O

O

O

O

1318.375

12

O

O

O

O

O

O

1319.543

13

O

O

O

O

O

O

1318.999

14

O

O

O

O

O

O

O

1319.172

15

O

O

O

O

O

O

O

O

1320.237

16

O

O

O

O

O

O

O

1320.235

17

O

O

O

O

O

1328.258

18

O

O

O

O

O

O

1330.134

19

O

O

O

O

O

O

O

1322.806

20

O

O

O

O

O

O

O

O

1325.969

21

O

O

O

O

O

O

O

1332.142

22

O

O

O

O

O

O

O

O

1311.476

23

O

O

O

O

O

O

O

O

O

1308.912

#### 4.5 Predicting Mode Choice

We use the predict() function to give the probability of an individual choosing a particular mode.

P <- predict(M23, newdata = S, type = "response")

Looking at the first three rows of the dataset:

Bus

Shared Bicycles/PMDs

Dalphin

0.278

0.614

0.108

0.265

0.584

0.151

0.232

0.511

0.257

For example, the individual corresponding to the first row will choose the bus with a 27.8% chance, Shared Bicycles/PMDs with a 61.4% chance, and the Dalphin with a 10.8% chance.

We let the predicted choice to be the most likely alternative (highest chance of being chosen).

PredictedChoice <- apply(P, 1, which.max)

Using a confusion matrix, we compare the actual choice in the training set with the predicted choice by the model.

ActualChoice <- modechoice.training$A

**Predicted Choice vs. Actual Choice**

Actual Choice

Bus

Shared Bicycles/PMDs

Dalphin

Bus

68

19

46

Shared Bicycles/PMDs

26

42

17

Dalphin

132

64

282

The diagonal entries of the confusion matrix are the instances where the model correctly predicts the actual choice of the individual. Accuracy is calculated by finding the proportion of instances where the model correctly predicted the actual choice. This model has an accuracy of 56.32%.