Master Project Business Analytics

Customer Choice Modelling

Exploring possibilities with e-commerce tracking data.

PUBLIC VERSION

Author:
Tim Steinkuhler
1853252

Supervisors:
Dr. Alwin Haensel (HAMS)
Prof. Dr. Ger Koole (VU)
Dr. Frans Feldberg (VU)

May 2015

Vrije Universiteit Amsterdam
Faculty of Exact Sciences
De Boelelaan 1081a
1081 HV Amsterdam

Haensel Advanced Mathematical Services
Ritterstrasse 2A
10969 Berlin, Germany

Haensel AMS
Advanced Mathematical Solutions

VU UNIVERSITY AMSTERDAM

“Life is really simple, but we insist on making it complicated.”

Confucius

Preface

This thesis was written during the Master Project Business Analytics I have done at Haensel AMS between October 1st 2014 and March 31st 2015. This project is the final stage of the two-year Master of Science program Business Analytics at the Vrije Universiteit Amsterdam. I have used data that is collected at an airline's website to study choice behaviour by consumers and build a computer program that can be used by Haensel AMS in the future.

I would like to thank Haensel AMS for the opportunity they gave me, in particular Alwin Haensel and Rolf Dilewski. Furthermore, I would like to thank Ger Koole and Frans Feldberg for their valuable feedback from Amsterdam and support during this project. I would also like to thank the airline for providing us with data that was used during this project and in particular Romi Hurts and Linda Schauwen for a fruitful and interesting meeting at the headquarters.

Finally, my thanks go out to my loving family and friends. Thank you Joke Moenis, Ron Steinkühler, Eva Steinkühler and Manon Leeflang for your everlasting love and support. You are the best.

Summary

One ever important goal for a company is to understand their customers' needs. In this research, we aimed to gain an understanding of how customers behave when trying to satisfy these needs. In particular, we've studied how people choose between a number of different flights. To do so, data was gathered from an airline's website that shows examples of the flights customers actually searched for and the flights that were bought after the search. These compared flights were used in choice sets, where the flight that was bought represents the chosen alternative in a set.

We used different properties about the flights in these choice sets and the decision makers that made the choices, to train models that predict choice behaviour. This behaviour is described by estimating the probability with which each of the flights in a set is chosen. First, we've applied the Multinomial Logit (MNL) model to a specific dataset to find those flight properties that give the best prediction. Consequently, we've split up the dataset based on decision maker properties to find differences in choice behaviour between different decision makers. Consequently, we've tried to exploit these differences by implementing the Laten Class MNL (LC-MNL) model.

We've found a set of properties that give the best prediction and have detected a difference in choice behaviour between decision makers. These differences were consequently exploited by improving the prediction with the LC-MNL model. However the final prediction may be improved by further refining the implementation of the LC-MNL model and using data that was currently not available, but should be available to the airline in question. The predictions of the best model described in this document, are significantly better than a random model, but not necessarily very large.

Furthermore, our study has only used information about choice situations in which the decision makers have actually made a choice. This makes it that the resulting choice probabilities rely on the condition that a decision maker will actually choose one from the set of alternatives. In a commercial implementation one might want to know the expected outcome of giving a certain offer to a potential customer. To do so, an extra model should be used to predict the probability with which a customer would choose to buy. The current research could be used to estimate which flight in the offer would be chosen.

Contents

List of Figures

List of Tables

Abbreviations

AIC	**A**kaike **I**nformation **C**riterion
HAMS	**H**aensel **A**dvanced **M**athematical **S**ervices
IIA	**I**ndependence of **I**rrelevant **A**lternatives
LC-MNL	**L**atent **C**lass **MNL**
LOS	**L**ength **O**f **S**tay
LRI	**L**ikelihood **R**atio **I**ndex
MLE	**M**aximum **L**ikelihood **E**stimation
MNL	**M**ulti**n**omial **L**ogit
NPR	**N**egative **P**rediction **R**atio
PR	**P**rediction **R**atio
T50P	**T**op **50**% **P**rediction Ratio

Chapter 1

Introduction

Haensel AMS (HAMS) is a young, Berlin, Germany based consultancy and software development company. HAMS provides Business Analytics and Big-Data Solutions to optimize business decisions and processes through the application of quantitative data analysis, predictive statistics and advanced optimization models.

This Master Project helps HAMS in the development of a new service called Customer Choice, aimed to increase profitability by better anticipating customers' needs. Through this service, HAMS will help clients to better understand what the driving forces are behind their customers' choices. Furthermore, the service includes helping clients use this knowledge to their advantage. HAMS believes there are valuable insights to be gained from the data that is available and is determined to expose these insights.

1.1 Background

In the current time, the trend is to store as much data as we can possibly get our hands on. Often, even before there is a plan on how to use it, data is stored with the idea that some day a possibility will come to extract valuable information from them. Consequently, there is a growing need for expertise on how to make use of data. Furthermore, companies in all sectors have been working towards a customer-centred way of doing business.

During this Master Project, we use a dataset containing information about the traffic at an e-commerce platform to gain a deeper understanding of the customers' choice behaviour. Information about different customers' visits to this website is translated into choice situations and the choice situations that result in a purchasing decision are analysed. We want to better understand why customers choose one alternative (buy one product) after comparing multiple alternatives (products).

To study this choice behaviour, different mathematical models are implemented to find relationships between the properties of different alternatives in a choice situation and the probability with which the alternatives are chosen. Originating from traditional choice modelling and slightly altered to fit the described dataset, each model is trained and tested on a number of different subsets of the website's traffic data. To our knowledge, this implementation has not been previously attempted.

This research can be seen as a step towards a better understanding of customer choice behaviour. This knowledge could ultimately be used to aid in business decisions for marketeers when constructing personalized offers for their customers. By having a better understanding of what is important to a customer, they could make offers that consist of products that are more relevant than a general offer would be. Furthermore, it could help calculate the expected return for re-targeting a customer that has shown certain choice behaviour on the website. If you cannot make a certain customer an offer that is interesting to him / her, than you shouldn't waste money trying to reach them with another, irrelevant offer.

1.2 Research Objectives

The goal of this research is to explore the capability of the given dataset in predicting choice behaviour. In this light, the aim is to answer the following questions:

1. When predicting a choice between several flights, what information best helps predict this choice?
2. Can we detect differences in choice behaviour between decision makers?
3. If so, can we exploit these differences?
4. What can be done to make an even better prediction?

The models that are described are implemented in prototype software and made available to HAMS.

1.3 Structure of the Report

In the next chapter (2), we introduce several choice models from the literature that will be applied. Chapter 3 describes the dataset that was made available to carry out this study. In chapter 4, theory and data are brought together and we explain how they are combined to explore customer choice behaviour. The results of the research are described in chapter 5. Chapter 6 concludes this research by revisiting the research objectives with insights gained and making suggestions for further research.

Chapter 2

Literature Review

This chapter describes the mathematical models from the literature that are used to study customer choice behaviour. We start by introducing general concepts about discrete choice modelling in the first section. The subsequent sections will each describe a more specific implementation of a discrete choice model. This chapter draws heavily on the books by Ben-Akiva and Lerman ([1]), and Train ([2]). Ben-Akiva and Lerman wrote one of the first (very) extensive books about discrete choice models, their origin and derivation. Train extended their research by adding new algorithms and simulation methods.

2.1 Discrete Choice Models

Say there is a number of decision makers making a choice between a number of possible alternatives. Discrete choice models are designed to estimate the probability of being chosen for each of those alternatives, which is what we want to do in this research. This is done by evaluating information about the decision maker and about the properties of the alternatives. The models first estimate the effect of these pieces of information and consequently calculate the probability of being chosen for each of the alternatives.

Discrete Choice models are based on a number of assumptions ([1], page 100). The first assumption to these models is that the analyst can define some choice set C that includes all potential choices for some population, with J being the number of alternatives in C. Then each decision maker in the population has a set C_n (of J_n alternatives) that are logically available to them. Out of this set, each decision maker is assumed to choose one and only one alternative.

Discrete Choice models assume each decision maker (n) aims to maximize utility, by choosing one of J_n alternatives. Each of these would give them a utility of U_{jn} respectively. This utility is known to the decision maker, but not to the analyst. The analyst observes some properties of each of the alternatives as faced by the decision maker, denoted by vectors $\boldsymbol{x}_{jn}, \forall j \in C_n$, and some properties of the decision maker, denoted by vector $\boldsymbol{z}_n$, and can specify a function that relates these observed factors to the decision maker's utility. The function is denoted $V_{in} = V(\boldsymbol{x}_{in}, \boldsymbol{z}_n)$.

Let's say n is choosing between different flights, than $\boldsymbol{x}_{jn}$ may hold information about the price, date(s) and airport locations for each flight $j \in C_n$. Furthermore, $\boldsymbol{z}_n$ may tell us how far in advance n is booking his flight. The vectors $\boldsymbol{x}_{jn} \forall j \in C_n$ and $\boldsymbol{z}_n$ thus hold independent variables.

Function V indicates how the analyst assumes the independent variables are related to n's choice behaviour. V usually depends on parameters that are estimated statistically. The aspects of utility that are not observed, are captured by ϵ_{in}, which is the difference between U_{in} and V_{in}. Because ϵ_{in} is unknown to the analyst, they are treated as random. The probability that decision maker n chooses alternative i, is then given by the probability that the utility of alternative i is greater than or equal to the maximum out of all utilities of the other alternatives.

$$\begin{aligned} P_n(i) &= Pr[U_{in} \geq \max_{j \in C_n}(U_{jn})] \\ &= Pr[V_{in} + \epsilon_{in} \geq \max_{j \in C_n}(V_{jn} + \epsilon_{jn})] \\ &= Pr[\epsilon_{jn} - \epsilon_{in} \leq V_{in} - V_{jn} \forall j \in C_n] \end{aligned} \tag{2.1}$$

Different discrete choice models are defined by the assumptions that are made about how to calculate these probabilities. They differ on the specification of function V and / or on the distribution of random components ϵ_{in}.

2.2 Multinomial Logit

One of the simplest and most widely known discrete choice models is the Multinomial Logit (MNL) model. The model dates back to 1959, when Luce ([3]) derived and implemented the model in the field of mathematical psychology. We use this model for it's analytical properties. It's clean specification allows for fast implementation of the model and direct interpretation of the parameters.

2.2.1 Mathematical Properties

For the MNL model, random components ϵ_{in} are (1) independently distributed, (2) identically distributed and (3) Gumbel-distributed with location parameter γ and scale parameter $\mu > 0$. The Gumbel distribution has a few useful properties:

1. If ϵ is Gumbel distributed with parameters (γ, μ) and V and $\alpha > 0$ are any scalar constants, the $\alpha\epsilon + V$ is Gumbel distributed with parameters $(\alpha\gamma + V, \mu/\alpha)$.

2. If ϵ_1 and ϵ_2 are independent Gumbel-distributed variates with parameters (γ_1, μ) and (γ_2, μ) respectively, then $\epsilon^* = \epsilon_1 - \epsilon_2$ is logistically distributed:

$$F(\epsilon^* = \frac{1}{1 + e^{\mu(\gamma_2 - \gamma_1 - \epsilon^*)}} \tag{2.2}$$

3. If $(\epsilon_1, \epsilon_2, \dots, \epsilon_J)$ are J independent Gumbel-distributed random variables with parameters (γ_1, μ), $(\gamma_2, \mu) \dots (\gamma_J, \mu)$ respectively, then $\max(\epsilon_1, \epsilon_2, \dots, \epsilon_J)$ is Gumbel distributed with parameters

$$\left(\frac{1}{\mu} ln \sum_{j=1}^{J} e^{\mu\gamma_j}, \mu \right)$$

These properties, make it that the probability that alternative i, is chosen in by decision maker n, is straightforwardly calculated. We follow the proof by Ben-Akiva and Lerman ([1]):

Define:

$$U_n^* = \max_{j \in C_n, j \neq i} (V_{jn} + \epsilon_{jn}) \tag{2.3}$$

U_n^* is Gumbel distributed with parameters

$$\left(\frac{1}{\mu} ln \sum_{j \in C_n, j \neq i} e^{\mu V_{jn}}, \mu\right)$$

Which can be written as $U_n^* = V_n^* + \epsilon_n^*$, where

$$V_n^* = \frac{1}{\mu} ln \sum_{j \in C_n, j \neq i} e^{\mu V_{jn}}$$

and ϵ_n^* is Gumbel distributed with parameters $(0, \mu)$. This makes that

$$\begin{aligned} P_n(i) &= Pr[V_{in} + \epsilon_{in} \geq V_n^* + \epsilon_n^*] \\ &= Pr[(V_n^* + \epsilon_n^*) - (V_{in} + \epsilon_{in}) \leq 0] \\ &= \frac{1}{1 + e^{\mu(V_n^* - V_{in})}} \\ &= \frac{e^{\mu V_{in}}}{e^{\mu V_{in}} + e^{\mu(V_n^* - V_{in})}} \\ &= \frac{e^{\mu V_{in}}}{e^{\mu V_{in}} + \exp(ln \sum_{j \in C_n, j \neq i} e^{\mu V_{jn}})} \\ &= \frac{e^{\mu V_{in}}}{\sum_{j \in C_n} e^{\mu V_{jn}}}, \end{aligned} \tag{2.4}$$

where μ is often set to 1, because it cannot be observed and allows for easier calculation. The deterministic part of utility is assumed to be linear-in-parameters $\boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_B)^\top$, a vector of B that are taken into account. Hence, function $V(\boldsymbol{x}_{in})$ is calculated as:

$$V_{in} = \boldsymbol{\beta}^\top * \boldsymbol{x}_{in}, \tag{2.5}$$

where $\boldsymbol{x}_{in}$, holds the information about each of the B aspects of alternative i for decision maker n, that correspond with parameters $\boldsymbol{\beta}$.

To find the MNL models that best fits the data, one can use Maximum Likelihood Estimation (MLE). MLE finds those values for the parameter vector $\boldsymbol{\beta}$, that maximize the likelihood over all choices made by decision makers in a given population. This is equivalent to maximizing the log-likelihood sum:

$$\max_{\boldsymbol{\beta}} LL = \sum_{n=1}^{N} \sum_{i \in C_n} y_{in}(log(P_n(i))), \tag{2.6}$$

where we assume independence between all decision makers. y_{in} indicates which alternative is chosen:

$$y_{in} = \begin{cases} 1 & \text{if alternative } i \text{ is chosen by decision maker } n \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

To determine the MLE estimators, the elements of the gradient vector $\frac{\delta LL}{\delta \hat{\beta}_b}$ for the log likelihood should be set as close as possible to zero. The gradient vector is given by:

$$\frac{\delta LL}{\delta \hat{\beta}_b} = \sum_{n=1}^{N} \sum_{i \in C_n} y_{in} \left(x_{inb} - \frac{\sum_{j \in C_n} e^{\boldsymbol{\beta}^\top \boldsymbol{x}_{jn}} x_{jnb}}{\sum_{j \in C_n} e^{\boldsymbol{\beta}^\top \boldsymbol{x}_{jn}}} \right) = 0, \ \forall b \in \{1, 2, \ldots, B\} \tag{2.8}$$

McFadden (1974) shows that under relatively weak conditions, the log-likelihood function for the MNL model is globally concave, so there will be one unique optimal parameter vector $\boldsymbol{\beta}$.

2.2.2 Implications

The properties of the MNL model that are described above, though making for an easily interpretable model, have two implications that need to be addressed. These will be discussed in the following sections.

2.2.2.1 Taste Variations

In the MNL model, the same parameter vector $\boldsymbol{\beta}$ is used for all decision makers in the population. Let's say two decision makers, m and n, are independently looking to fly to Berlin on the 27th of March. They are both given the same options: depart from Amsterdam and pay €100, or depart from Eindhoven and pay €90. This example might be represented as shown in table 2.1:

	Dep_AMS	Dep_EIN	Price
Alt. 1	1	0	100
Alt. 2	0	1	90

Table 2.1: A Simple Choice Set Example

In other words: $\boldsymbol{x_{1n}} = \boldsymbol{x_{1m}} = (1, 0, 100)$ and $\boldsymbol{x_{2n}} = \boldsymbol{x_{2m}} = (0, 1, 90)$. Now if the analyst has found that the optimal parameter vector is given by:

$$\boldsymbol{\beta} = (\beta_{\text{Dep\_AMS}}, \beta_{\text{Dep\_EIN}}, \beta_{\text{Price}}) = (10, 10, -0.1),$$

then the choice probabilities would be calculated as:

$$\begin{aligned} P_m(1) = P_n(1) &= \frac{e^{(10,10,-0.1)^\top * (1,0,100)}}{e^{(10,10,-0.1)^\top * (1,0,100)} + e^{(10,10,-0.1)^\top * (0,1,90)}} \\ &= \frac{e^0}{e^0 + e^1} = 26.89\% \end{aligned} \tag{2.9}$$

$$\begin{aligned} P_m(1) = P_n(1) &= \frac{e^{(10,10,-0.1)^\top * (0,1,90)}}{e^{(10,10,-0.1)^\top * (1,0,100)} + e^{(10,10,-0.1)^\top * (0,1,90)}} \\ &= \frac{e^1}{e^0 + e^1} = 73.11\% \end{aligned} \tag{2.10}$$

However, the value or importance that decision makers place on each of the properties of the alternatives, varies, in general, over decision makers (Train [2]). Maybe m lives right next to Eindhoven airport and n lives in Amsterdam. This would mean that in fact m and n vary in taste when it comes to this choice and, therefore $P_m(1) \neq P_n(1)$.

The MNL model allows for these taste variations, but only through observed characteristics about the decision maker. Hence, if the analyst knows n lives in Amsterdam, then he could decide to set $\boldsymbol{x_{1n}} = (1.2, 0, 100)$, making $P_n(1) = \frac{e^2}{e^2 + e^1} = 73.11\%$.

2.2.2.2 Substitution Patterns

Another frequently discussed properties of the MNL model, is the *independence of irrelevant alternatives* (IIA) property. By the formulation of the model, for any two alternatives i and h, the ratio between the probabilities of them being chosen is given by:

$$\frac{P_n(i)}{P_n(h)} = \frac{e^{V_{in}} / \sum_{j \in C_n} e^{V_{jn}}}{e^{V_{hn}} / \sum_{j \in C_n} e^{V_{jn}}} = \frac{e^{V_{in}}}{e^{V_{hn}}} = e^{V_{in} - V_{hn}} \tag{2.11}$$

Which is independent of all other alternatives. This means that when a new alternative is introduced in C_n, this ratio will stay the same. Hence, the new alternative will draw proportionally equal amounts of probability away from the existing alternatives. Furthermore, if one of the existing alternatives in the choice set, would be improved, and it's probability of being chosen would increase, this would also draw proportional amounts of probability away from the other alternatives in the set. This can sometimes be hard to defend, one famous counter-argument to defending this property is the red bus - blue bus problem (Train, [2], page 50).

Say a decision maker, n, has the choice to take a blue bus to work, or go by car. Under current circumstances, $V_{\text{blue bus}} = V_{\text{car}}$, thus, by the MNL model,

$$\frac{P_n(\text{blue bus})}{P_n(\text{car})} = e^{V_{\text{blue bus}} - V_{\text{car}}} = 1. \tag{2.12}$$

Now, if a new alternative, say, a red bus was to become available, this would not change $V_{\text{blue bus}}$ or V_{car}. Therefore, by design, this red bus would draw an equal amount of probability from both other the blue bus and the car. Now (if the analyst has no information about the colour preference of the decision maker), $V_{\text{red bus}}$ will be equal to $V_{\text{blue bus}}$, making:

$$V_{\text{red bus}} = V_{\text{blue bus}} = V_{\text{car}} = \frac{1}{3},$$

while in reality,

$$V_{\text{red bus}} = V_{\text{blue bus}} = \frac{1}{4} V_{\text{car}} = \frac{1}{2},$$

would be more likely.

These two implications of the MNL model's mathematical properties are often the reason to develop new models that do not have them.

2.3 Latent Class Multinomial Logit

Not each decision maker will react in the same way, when given the exact same set of choices. In fact, even the same decision maker might not make the same choice when deciding between the same alternatives on different occasions. The Latent Class Multinomial Logit (LC-MNL) model extends the MNL model, allowing for systematic taste variation between decision makers. This is implemented by introducing K latent classes to which each decision maker belongs with some probability (the analyst is assumed not to be able to distinguish between classes exactly). There may be taste variation between different classes, but there is no taste variation within the same class.

This model is chosen as the second model in our research, for it's ability to identify classes of decision makers. These allow us to use information that is known about the decision makers to better predict the choice they will make.

2.3.1 Mathematical Properties

In this research, we will be using an LC-MNL implementation based on the one described by Boxall and Adamowicz ([4]). In contrast to the MNL model, in the LC-MNL model there are K classes, each having a class-specific product preference parameter vector $\boldsymbol{\beta}_k$. The deterministic utility of alternative i for decision maker n, given that decision maker n belongs to class k, is given by equation 2.13:

$$V_{in|v_n=k} = \boldsymbol{\beta}_k^\top * \boldsymbol{x}_{in}. \tag{2.13}$$

We introduce a latent variable v_n, indicating the class of decision maker n. The probability that alternative i is chosen by decision maker n, given that decision maker n belongs to class k is given by:

$$P_n(i|v_n = k) = \pi_{ni|k} = \frac{e^{\mu_k V_{ink}}}{\sum_{j \in C_n} e^{\mu_k V_{jnk}}}, \tag{2.14}$$

$$P_n(i|v_n = k) = \frac{e^{V_{ink}}}{\sum_{j \in C_n} e^{V_{jnk}}}, \tag{2.15}$$

$$P_n(v_n = k) = \frac{e^{\boldsymbol{\lambda}_k^\top \boldsymbol{z}_n}}{\sum_{m=1}^{K} e^{\boldsymbol{\lambda}_m^\top \boldsymbol{z}_n}}, \tag{2.16}$$

$$P_n(i) = \sum_{k=1}^{K} P_n(v_n = k) P_n(i|v_n = k) \tag{2.17}$$

where, the random components of utility for each alternative, given that decision maker n belongs to class k, is assumed to be Gumbel-distributed with scale parameter μ_k. Boxall and Adamowicz ([4], page 426) show that these scale parameters should be set to $\mu_k = 1 \forall k$, to allow for inequality between parameter vectors $\boldsymbol{\beta}_k$. The probability that decision maker n belongs to class k, can be written as:

$$P_n(v_n = k) = \pi_{nk} = \frac{e^{\alpha \boldsymbol{\lambda}_k^\top \boldsymbol{z}_n}}{\sum_{m=1}^{K} e^{\alpha \boldsymbol{\lambda}_m^\top \boldsymbol{z}_n}}, \tag{2.18}$$

where $\boldsymbol{z}_n$ is the vector described above, containing information about decision maker n, and $\boldsymbol{\lambda}_k = (\lambda_{k1}, \ldots, \lambda_{kA})^\top$ a class-specific parameter vector used for calculating the probability of decision maker n belonging to class k. Parameter α indicates the scale parameter of the error term for the class selection process. This parameter is set equal to 1 as required as well (see [4], page 426). The unconditional probability that decision maker n chooses alternative i is then given by:

$$P_n(i) = \pi_{ni} = \sum_{k=1}^{K} P_n(v_n = k) P_n(i|v_n = k) \tag{2.19}$$

The estimation problem contains K product preference parameter vectors $\boldsymbol{\beta}_k$ and K class selection parameter vectors $\boldsymbol{\lambda}_k$. The gradients that are used to find the optimal parameters for the LC-MNL model required some calculations that are described in appendix A. The log likelihood function for the LC-MNL model is not necessarily globally concave. Consequently, good initial values should be used for the parameters, to avoid finding local optima instead of global optima. In this research, we will use as starting values for each class, the optimal parameters that were found running an MNL model on a subset of the data used for the LC-MNL model.

2.3.2 Implications

As stated before, the LC-MNL model allows better for taste variation between decision makers to be modelled. Furthermore, the ratio of probabilities of selecting any two alternatives, contains information about the deterministic utility of the other alternatives:

$$\frac{P_n(i)}{P_n(h)} = \frac{\sum_{k=1}^{K} P_n(v_n = k) P_n(i|v_n = k)}{\sum_{l=1}^{K} P_n(v_n = l) P_n(h|v_n = l)} \tag{2.20}$$

$$= \frac{\sum_{k=1}^{K} \frac{e^{\boldsymbol{\lambda}_k^\top \boldsymbol{z}_n}}{\sum_{m=1}^{K} e^{\boldsymbol{\lambda}_m^\top \boldsymbol{z}_n}} \frac{e^{V_{ink}}}{\sum_{j \in C_n} e^{V_{jnk}}}}{\sum_{l=1}^{K} \frac{e^{\boldsymbol{\lambda}_l^\top \boldsymbol{z}_n}}{\sum_{m=1}^{K} e^{\boldsymbol{\lambda}_m^\top \boldsymbol{z}_n}} \frac{e^{V_{hnl}}}{\sum_{j \in C_n} e^{V_{jnl}}}} \tag{2.21}$$

$$= \frac{\sum_{k=1}^{K} e^{\boldsymbol{\lambda}_k^\top \boldsymbol{z}_n} \frac{e^{V_{ink}}}{\sum_{j \in C_n} e^{V_{jnk}}}}{\sum_{l=1}^{K} e^{\boldsymbol{\lambda}_l^\top \boldsymbol{z}_n} \frac{e^{V_{hnl}}}{\sum_{j \in C_n} e^{V_{jnl}}}}, \tag{2.22}$$

which means the IIA property does not need to be assumed.

Chapter 3

Data Analysis

During this master project, we study data that are created at a commercial airline's e-commerce platform. Information is stored about all visits to this website. Each of these visits is registered as a session, which contains a list of all the pages that are viewed during the session, including information about what was shown on these pages. Cookies[1] are used to link multiple sessions to each other, which means that if the same device is used under the same settings and without removing this website's cookie[2], these sessions are attributed to the same visitor.

From all of the sessions held by all of the visitors, a conversion path database was created as follows. When during a session, a visitor decides to purchase a flight, this is called a conversion. Consequently, all of the sessions by that same visitor that have taken place within the month leading up to the moment of conversion and after a possible previous conversion, are linked with that conversion and are said to belong to the same conversion path.

These conversion paths are quite common in online marketing and are studied to attribute the value gained by each conversion to different marketing channels. This is done by distributing the total conversion value over all of the sessions in the conversion path, determining through which marketing channel (if any) each of the sessions were initiated, and summing this value over all sessions initiated by each marketing channel.

[1] More information on cookies: http://en.wikipedia.org/wiki/HTTP_cookie

[2] Or if a second device is used that copied the cookie from the first

We are studying the same data from a different angle, namely we convert each conversion path into choice sets. We think of the flights in a conversion path as a search history for flights. We extract information about all of the flights that were searched for in a conversion path and assume each of these flights are alternatives in a choice set. Furthermore, we assume the customer will decide between these flights at one point in time at which all of these flights are available. The flight that is bought (conversion flight), represents the alternative that is chosen from the choice set. About each of the flights, we know when it was searched for, departure and arrival airports, departure date, return date (if applicable) and the price.

Please note that the dataset does not contain any other (e.g., socio-economic or geographic) information about the people the tickets were for or who was visiting the website. Neither is it possible to use the data to link multiple conversion paths to the same visitor. The data is made available in such a way that it is impossible to identify single visitors.

The airline in question is a European low-cost airline, flying to holiday destinations throughout Europe and the Mediterranean. They sell their flights through tour operators that have their own tool, and directly to customers through their public website (where our dataset was created).

This chapter continues by giving a few examples of conversion paths that were transformed into choice sets. ently, we describe different pieces of information that can be extracted about each flight and about each decision maker, and conclude with a short data analysis summary.

3.1 Conversion Path Examples

CONFIDENTIAL

3.1.1 Flight Filtering

CONFIDENTIAL

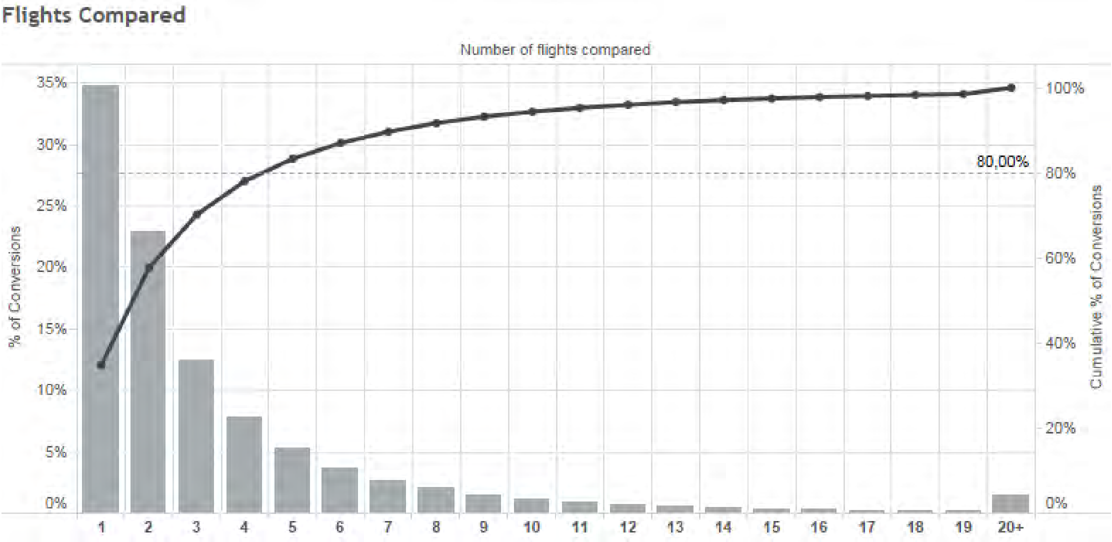


FIGURE 3.1: Histogram: Flights Compared

3.1.2 Dataset Statistics

The dataset contains information about 274,736 conversion paths. To get an idea of how often it happens that these conversion paths actually contain multiple flights from which one was chosen we have included figure 3.1. This figure shows the distribution of how many flights are compared in each conversion path, where a flight in this case is defined by route combined with departure and return dates and the price. This graph tells us that over 65% conversions are made after comparing at least two different flights. Since there at least have to be two flights for someone to be able to make a choice between them, only this 65% can be transformed into a choice set.

3.2 Flight Properties

If a flight is chosen, this means that the visitor to the website decides to purchase transportation by airplane from the departure airport to the arrival airport at the departure date date, and possibly back the other way around at the return date. In the rest of this document, when needed, we will make a distinction between one-way flights (when no return date is specified) and return flights (when the return date is specified).

In the studied dataset, each flight is identified by four pieces of information, namely:

1. A departure airport,
2. an arrival airport,
3. a departure date and
4. a return date (if applicable)

However, the same flight may have different properties when viewed by different decision makers. This section describes properties that are known about the flights in each choice set. We will describe each of the properties and how they can be used by the choice models. These require the properties of the flights to be numerical, hence some of the properties require numerical transformation before they can be put to use.

To allow more properties to be used about each flight, in the calculation in some of these properties, we will compare each flight to the flight that was searched for the first. The choice for the first flight is motivated by two arguments. From the customers' perspective, the first flight you search for is often the flight you are most interested in. From the analyst's perspective, the first flight that is searched for is the first information you receive about a customer's current need for travel. Based on this information, you can try to make an offer out of multiple flights that may best fit their needs. In the following text, we will refer to the flight that was first searched for as the initial flight.

3.2.1 Price

To get where you want to go, and when you want to go there, there is a price to pay. For customers of a low-cost airline, it is likely that the price of the flight plays an important role when deciding between different flights. In this dataset, the price is given in two ways, both in Euro (EUR). The first is the flight-value, which is the flight price per single adult without extras. The second is the revenue (total price for the booking for all people, including extras).

Furthermore, it is common for airlines to display the flight-value for multiple dates and possibly also multiple time slots per day, at the same time. This is also the case for our airline, but not all of these flight-values are found in the data. Only the lowest flight-value for the date that was actually searched for, is found in our data.

There are no different classes of seats when it comes to the flight-value. A choice can be made for a preferred seat, but the extra costs would then be added to the revenue and not the flight-value. Since it is not clear what exactly this revenue is made up of, we will focus on the flight-value. This allows us to compare choice situations for different numbers of people with each other without worrying about the difference in costs.

As such, the *Flight-Value* is one of few properties that can directly be used by the choice models, since it is already numerical. An alternative to the original flight-value could be to use the *Flight-Value Per Kilometer*, to incorporate the possibility that people are more willing to pay for a flight that is over a longer distance. In figure 3.2, histograms are shown of the flight-value (left column) and flight-value per kilometer (right column) for one-way (top row) and return (bottom row) flights.

We do need to keep in mind that the flight-value for a flight changes over time. If a customer looks at a flight today, and comes back one week from now, it is likely that the price has changed. Consequently, if in this example the price has risen during that week, and the customer buys this flight, for the model it appears as though the customer prefers to pay the higher price.

3.2.2 One-way or Return

Our data contains information about both one-way and return flights. Table 3.1 shows the distribution of one-way and return flights for initial flights, over all flights and finally for the conversion flights. In 251,211 of the conversions, the first flight that is searched for, is a return flight and 183,090 of these customers, only look for return flights. However, when also looking at one-way flights, customers often book a one-way

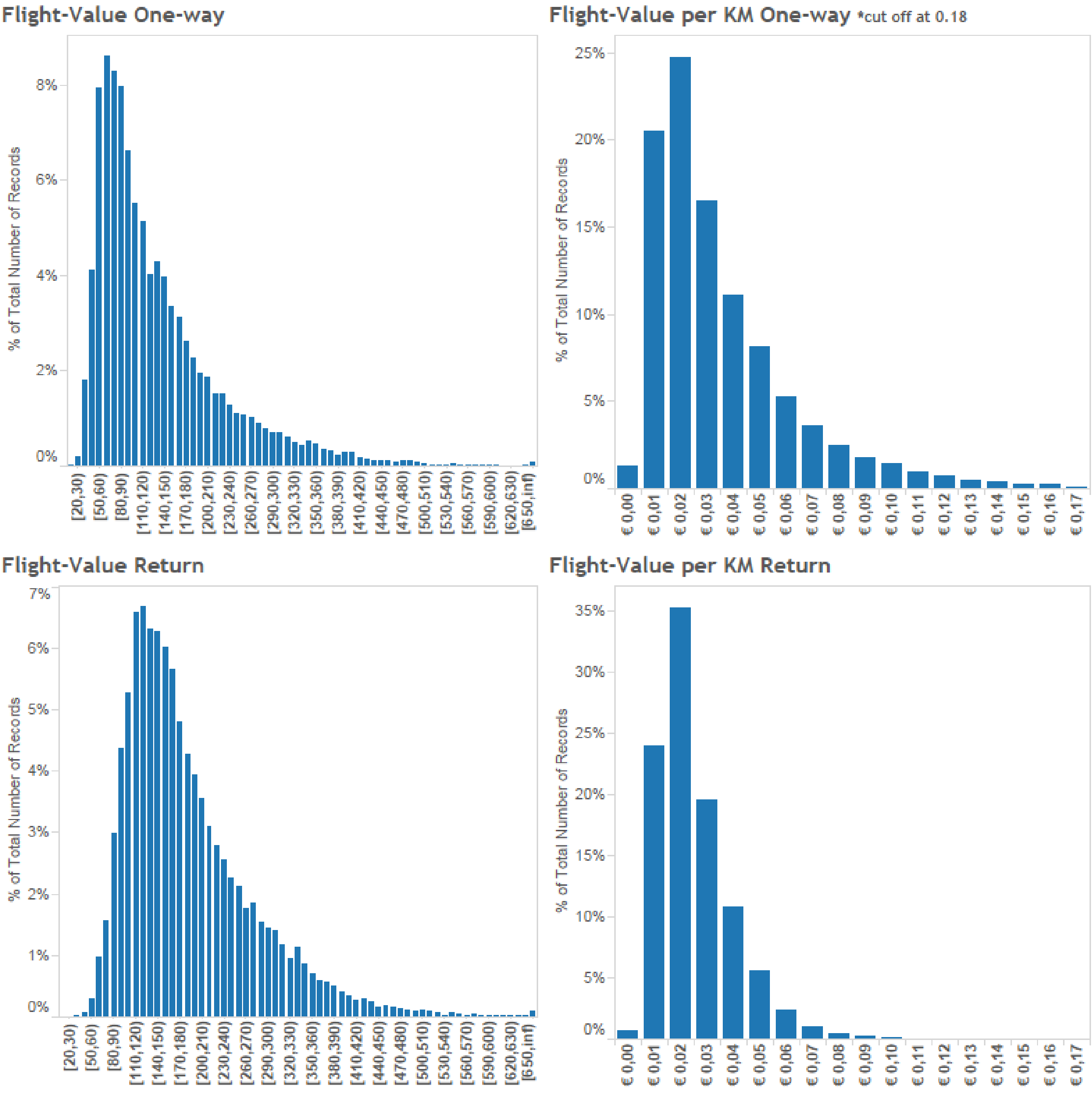


Figure 3.2: Flight-Value Histograms

flight. This behaviour could result from the need for return transportation, but booking only one part of the way at once. The other half of the transportation is then arranged separately, either with the airline in question, or through another channel.

Return flights are factually two one-way flights sold together. However, in the dataset, no distinction is made between the one-way and return flights when it comes to the price (more detailed information about how the price is represented in the dataset will be described in section 3.2.1). This means that there is no way of knowing which proportion of this one price is asked for each of the one-way parts of a return flight. Hence, for the analysis, we need to take this into account and make sure we do not

One-way (initial flight)	**One-way (all flights)**	**One-way (conversion)**	**Conversions**	**%**
0	0 : Return Only	0	183.090	66,64%
	0-1: Both	0	4.765	1,73%
		1	63.356	23,06%
1	0-1: Both	0	1.397	0,51%
		1	3.389	1,23%
	1: One-way Only	1	18.739	6,82%

Table 3.1: Table showing distribution between one-way and return flights

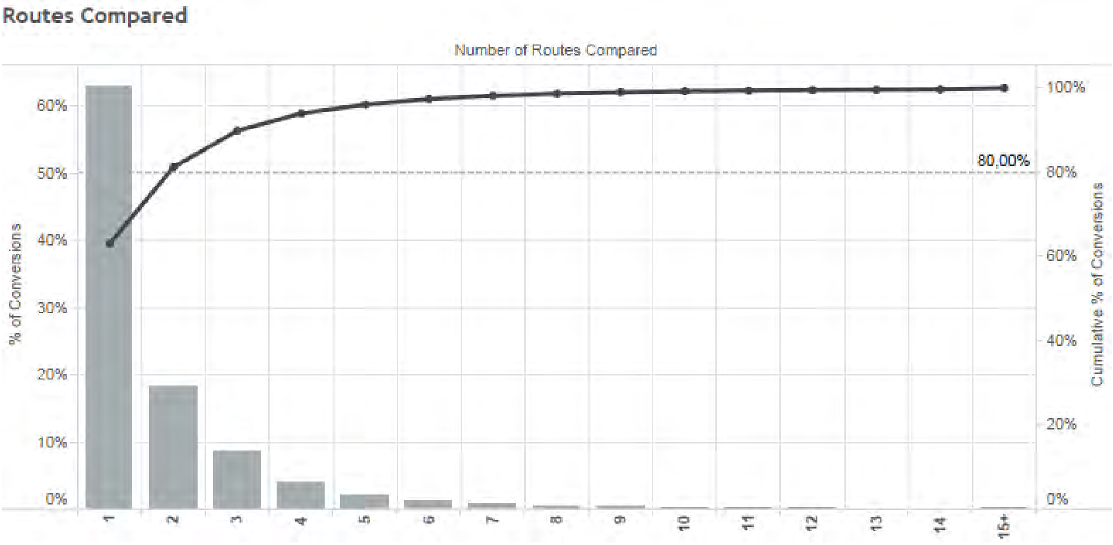


Figure 3.3: Histogram: Routes Compared

unintentionally make comparisons between prices for one-way flights (single tickets) and return flights (double tickets).

Although it might be useful to know why customers would buy one-way flights instead of return flights, this is out of the scope of this research. During the selection process of the data on which the models will be trained (more about the training process can be found in section 4.2.2), a choice will be made for either one-way or return flights. Consequently, choice sets will be used that contain only one-way flights, or only return flights. This means that 26.53% of all conversions will not be taken into account for any of the analyses.

3.2.3 Location

In the dataset, 388 distinct combinations of departure (98 unique values) and arrival (101 unique values) airports are found. These are all indicated by their three letter IATA airport code[3]. As is shown in figure 3.3, in 63.00% (173.076) of the conversion paths in the dataset, only flights on one route are compared. Furthermore, out of the other 37.00% (101.660) conversion paths that included more than one route, 34.49% (35.060) of the times the conversion route matched the initial search route. This indicates that most customers are only interested in flying from one airport to another. Hence, as expected, the departure and arrival airport play an important role when choosing between different flights.

In practice, few customers book a flight from a departure airport to an arrival airport, because those airports are their actual origin and destination locations (except for when transferring between flights). Most customers book a flight because the departure airport is close to their origin and the arrival airport is close to their destination. Whether or not a customer has one specific destination in mind when comparing different flights, may differ between different customers.

3.2.3.1 Distance

In the examples of figure **??** and **??**, it seems clear that the customers had specific destination locations in mind during their search for flights. In these cases, the most logical thing to do would be to find the departure and arrival airports closest to those locations and search for flights on that route. When doing so, the initial flight would be the flight with the preferred departure and arrival airports. The further another flight's airports are away from the initial flight's, the further they would be away from the actual origin and destination locations, and thus the less interesting this other flight would be.

To capture this effect, we introduce the distance between the departure airport of the initial flight and another, as a property of this other flight. This property will be called the *"Distance Between Departure Airports"* and is set to zero for the initial flight. The property *"Distance Between Arrival Airports"* is the name for the arrival airport equivalent.

[3] For more about the IATA airport codes: http://www.iata.org/pages/airports.aspx

In the example of figure **??**, it seems the customer does not have one specific destination in mind. Rather, he or she is comparing different destinations. In this case, the *"Distance Between Arrival Airports"* property would not be a good predictor of which flight he or she would choose. To avoid the situation in which it would seem that the customer wants to book a flight to an arrival airport as far away from the initial flight's arrival airport, the *"Distance Between Arrival Airports"* property may be set to zero for all flights in choice sets that have this distance exceed some logical maximum value. This new version of the same property will be called *"Distance Between Arrival Airports Capped at* $\{max\}$*"*, where *max* is replaced by a numeric threshold.

We believe it is not necessary to introduce a capped version of the *"Distance Between Departure Airports"* property, because a customer is less flexible about choosing a different departure airport. Choosing a departure airport that is far away from the actual origin location, would mean transportation to this far away departure airport would have to be arranged.

The distances described above are all calculated by taking the shortest distance between two points on the surface of the globe in kilometers (orthodromic distance[4]). The airport coordinates were retrieved by matching the IATA codes to an international airport database[5].

3.2.3.2 Airport Specific Properties

There might also be other reasons for which a customer prefers one airport over another (e.g., the presence of shopping areas at the airport). For such preferences, an airport-specific indicator variable may be introduced as a property of a flight. These indicators will be called *"Departure* $\{IATA\}$*"* and *"Arrival* $\{IATA\}$*"* for departure and arrival airports respectively, where *IATA* will be replaced by the IATA for the airport in question. They will get a value of 1 if the flight in question flies from or to this airport and 0 otherwise.

As described above, there are 98 different departure airports and 101 different arrival airports. For computational reasons, it is not possible to use properties for every single one of these. So a choice should be made for a subset of them.

[4] source: http://en.wikipedia.org/wiki/Great-circle_distance

[5] Airport data source: http://openflights.org/data.html

3.2.4 Dates

Another important aspect of a flight is the timing. In fact, booking a flight means arranging transportation at one (or two) specific point(s) in time. It is not possible to consume a flight at another point in time. As stated previously, we know about each of the flights in the choice sets the departure dates and for the return flights also the return date. No information is available about which time the flights would depart or arrive. This section will describe how the dates are used to distinguish between different flights.

3.2.4.1 Departure Date

The date at which the flight departs will logically play a big role in the decision between different flights. There are a lot of different reasons one can think of, why someone would want to fly somewhere at a specific date. Someone might want to attend an event near the destination airport, visit a friend or relative that is available at that time, stay at their vacation house during their favourite season, or go kite-boarding somewhere with the best wind that time of the year. All of these are reasons to pick one date over another, but are not directly found in the data.

With the same logic as in section 3.2.3, when searching for a flight, the logical thing to do would be to first search for flights that depart on the date that you are most interested in. Taking the initial flight departure date as the preferred date, all other flights in a choice set can be given the property *Departure Date Deviation.* This is calculate as the number of days the another flight departs after the preferred date (which is negative if the other flight departs before the preferred date). Incorporating this property in a model allows us to learn if customers tend to choose to depart after or before the initial flight departure date.

3.2.4.2 Length Of Stay

For customers searching for a return flight, there is also the return date to decide on. The number of days between the departure date and the return date, is the resulting length of stay (LOS). In figure 3.4, you can see the distribution of the number of days

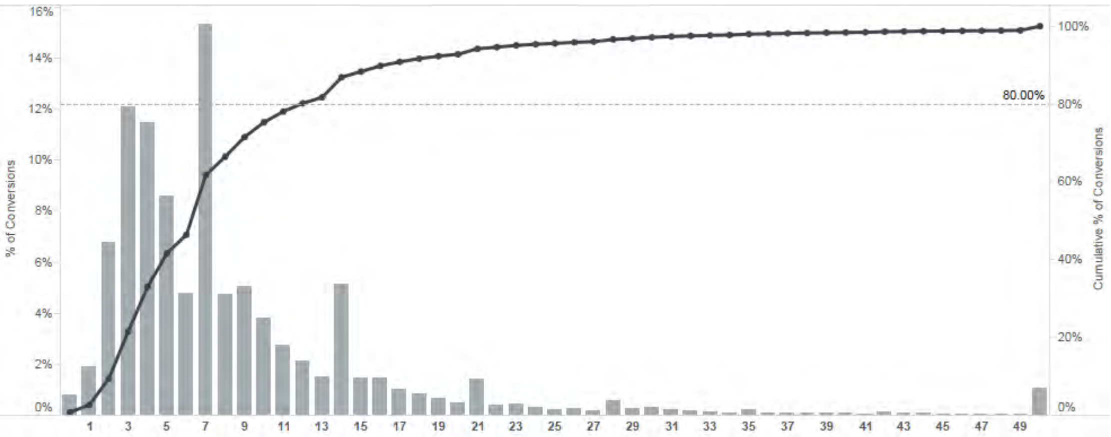


FIGURE 3.4: Length Of Stay Histogram Overall

between departure and return dates over all conversions. The most frequently occurring length of stay is 7 days.

Similarly to the departure date deviation property, we introduce the property *LOS Deviation*. This is calculated as the flight LOS minus the initial flight LOS. To avoid a stronger effect of the property in choice sets with a higher LOS and therefore a higher deviation, the *LOS Relative Deviation* can be used, which is equal to the *LOS Deviation* divided by the initial flight LOS. Both properties describe how many days a customer deviates from the original length of stay.

3.2.5 External Data

So far, location, price and date properties were described that can be directly extracted from the data. In a practical situation, it can be expected that these properties combined do not contain all of the information a customer uses to make a decision between flights. Other factors may indeed play a role, e.g., the costs and availability of hotels, the expected weather conditions at the destination, the fact that the customer has a holiday home at the destination. An attempt was made to enrich the available data by adding publicly available data about the arrival airports' environment. The goal was to use this information to test if it would improve the results of the model.

3.2.5.1 Travel Costs

For people that are trying to decide between different holiday destinations, the decision is not based solely on aspects of the flight. One other important factor can be the expected costs of other expenses during the stay. In attempt to include this factor in our analysis, hotel price index and travel cost per day figures for mid 2014 were retrieved from Numbeo[6]. This is an open data source. In any large-scale implementation of the models, other, paid and more reliable sources should be considered.

For each flight, we now also have a *Numbeo Hotel Price Index* and a *Numbeo Travel Cost Per Day* property. As a third option, the *Numbeo Total Travel Cost* may be used by multiplying the travel cost per day by the LOS. For calculation of these properties, for each flight, the hotel price index or travel cost is taken from the location in the numbeo dataset that is closest to the flight's arrival airport. To convert the travel cost per day from dollars to euros, the average daily midpoint conversion rate over 2014 was used from OANDA[7].

3.3 Decision Maker Properties

While analysing this data and trying to predict the outcome of choices, one must keep in mind that not all decision makers are the same. Not all decision makers would make the same decision, when confronted with the same set of flights to choose from. Hence, it may be useful to try and distinguish between different decision makers and allow them to respond differently to the flight properties.

As stated at the beginning of this chapter, the dataset does not contain any socio-economic or other information directly about any of the decision makers that can help distinguish between them. However, there is some information about their behaviour. As with the derivation of some of the flight properties, for distinguishing between different decision makers, we will study the first flight that was searched for.

[6] http://www.numbeo.com/hotel-prices/rankings.jsp?title=2014-mid

[7] http://www.oanda.com/currency/historical-rates/

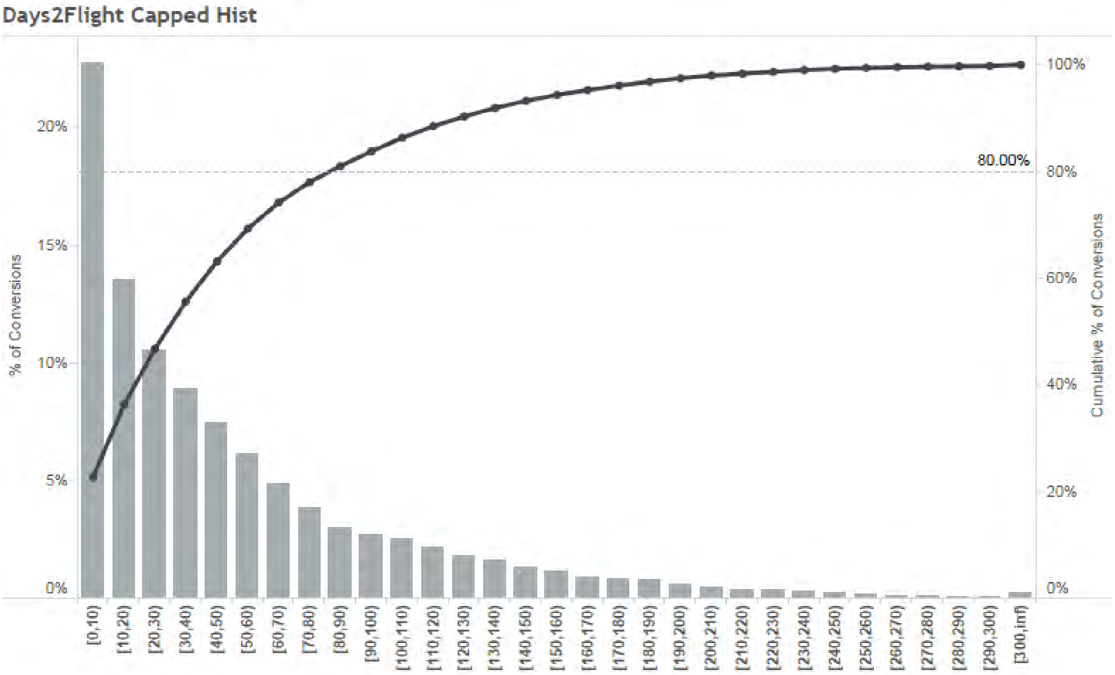


Figure 3.5: Days2Flight Histogram: How far in advance do people book their flights?

3.3.1 Marketing Channels

Each time a potential customer comes to the website, this is achieved by typing in the website address, or clicking on a link somewhere that leads to the website. In marketing terms, the way you come to the website is specified by channels. For this research, we will make a distinction between paid channels and non-paid channels. In our definition, you use a paid channel when you enter the website by clicking on advertisement on another website, on a search engine advertisement, or when you are referred by another website that gets money for this referral (e.g., price comparison websites). We introduce the *Paid Channel* property for decision makers and will test the hypothesis that decision makers that originally come to the website through a paid channel behave different from those that come through a non-paid channel.

3.3.2 Moment of Booking

Another way to distinguish between decision makers is by taking the amount of time they book in advance. An important part of the way prices for flights are calculated, is the amount of time there is still left for other customers to book the same flight. Hence, the price for the same flight can differ depending on the purchasing moment. In figure 3.5, you can find a graph that shows how far in advance customers book their flights. Since most customers know that prices change over time, and are most likely to go up the longer they wait to buy a ticket, this might influence their behaviour. By introducing the *Days to Flight* property for decision makers, we can test if indeed there is a difference in choice behaviour.

3.3.3 Original Interest

We can use different details about the flight that is initially searched for, to distinguish between decision makers. This distinction is based on the type of travel need different decision makers are trying to satisfy. Making use of the initial flight information, allows us to verify if decision makers trying to satisfy different travel needs, respond differently to the flights' properties.

There are many different distinctions one can think of when trying to categorize different flights (or travel needs). In this research, we will use a simple distinction to see if it helps understand choice behaviour. We leave other, perhaps more advanced categorizations to further research. The distinction we will use is based on whether or not the period between the departure date and the return date includes a stay over Saturday night. The corresponding decision maker property will be called *Stay Over Saturday Night*. Note that this property only has meaning for return flights.

3.4 Summary

The dataset that is used, holds short-term search history for flights on one specific airline's website. This search history is used to create choice sets, out of which a decision maker has made a choice. Each flight in each of the choice set has a set of properties that are found in the dataset. The effect of these properties on choice behaviour will be studied to predict choice behaviour. Since not all decision makers will respond in the same way to these properties, a distinction will be made amongst decision makers. We will try to use this distinction to help improve predictive power.

Chapter 4

Research Setup

This chapter begins by revisiting the assumptions that are made by the choice models that are used. Subsequently, a description is given of the way the models are implemented, including performance measures and significance tests. The final section of this chapter describes different settings under which the data and the models were used to find the properties and models with the most predictive power.

4.1 Model and Data Assumptions

As described in chapter 3, we assume there is one moment at which decision makers choose between different flights that were found in their search history. The dates on which certain flights were searched for, is not taken into account and all flights in the choice set are available to choose from at that one moment. Even though the search history spans one month at most, this is not always a realistic assumption. Especially prices vary over time. However, with this assumption, we only use information that may be known to the decision maker. The goal is to use this information to predict choice behaviour.

Before making use of the Discrete Choice Models, described in section 2.1, we need to keep in mind the following list of assumptions:

1. The analyst can define some choice set C that includes all potential choices for some population, with J being the number of elements in C

2. Each member of the population has a subset of $C_n \subseteq C$ (with $J_n < J$ elements) choices that are logically available to them
3. Out of this set, each member is assumed to choose one and only one alternative.

Given these assumptions, there are a number of things to keep in mind, before using this search history data with discrete choice models. These are listed below:

1. The search history does not hold all alternatives for the population
    - 1.1. There may be other airlines offering flights similar to our airline's.
    - 1.2. Consumers could choose another way of travelling.
2. Not each flight in one conversion path necessarily belongs to the same choice set (see section **??**).
    - 2.1. Multiple people could be using the same device.
    - 2.2. One person could be using the same device to satisfy different needs for travel.
3. Not each flight that was compared may be matched to the same choice situation.
    - 3.1. One person could use multiple devices to compare flights for the same need for travel
4. Not all information about each choice situation is available
    - 4.1. Decision makers might base their choice for a flight, based on information that is not found in the dataset. E.g.:
        - 4.1.1. How far the airports are away from actual origin and destination locations,
        - 4.1.2. Weather forecasts (and preferences),
        - 4.1.3. Events at the destination of choice or
        - 4.1.4. Hotel prices and preferences
    - 4.2. The search history doesn't show all information about each event
        - 4.2.1. Only one flight value is given, whilst web pages often show multiple days / flight times with their respective prices

Items 1 and 3 both make that when using the conversion path information to generate choice sets, each choice set C_n may not be complete. Hence, the models we use, only give an indication of how probable it is for each alternative in C_n to be chosen, given that in fact, one of the alternatives in set C_n is chosen.

Item 2 means that a generated choice set C_n might in fact overlap multiple choice sets. To overcome this problem, during the data selection process, we will filter out all alternatives that logically do not match the need for travel that is satisfied by the product that is purchased. This process is described in section 3.1.1.

Item 4 makes that we might miss out on information that could help explain why one alternative is chosen over another. In this research, we will only use the information that is described in chapter 3.

4.2 Implementation

As described in chapter 2, during this research we will use both the MNL and the LC-MNL model to help predict choice behaviour. In the literature (e.g. [2] , [4]), these models are used to predict the choice between one specific set of alternatives (C), out of which some or all are available to each decision maker n as C_n. An example would be transportation to work. C may be: {car alone, car pool, public transportation, bike or walk}, these would appear in the same order for each choice set. Then C_n may be {car pool, transit} for someone that doesn't own a car, and lives too far away from work to either walk or take a bike there. Under such circumstances, there are existing implementations (e.g., in R and Matlab) one could use. These implementations all assume there is this specific set of alternatives.

It would be very impractical to create one set of all possible flights for a certain population with our implementation (there would be up to 388 different routes, for which there are flights on a large number of different dates). Hence, each choice set may include an entirely different set of flights. To facilitate this extra flexibility, a Python program was written and made available to HAMS. This program retrieves a dataset from the database (1), trains a requested choice model (2) and writes the results to a spreadsheet (3). These three modules will be more thoroughly described in sections 4.2.1, 4.2.2 and 4.2.3.

4.2.1 Data Retrieval

4.2.1.1 Choice Set Selection

Out of the total 274,736 conversion paths, a selection needs to be made, upon which the choice models can be built. This selection is made with two motives, namely for the selection to comprehensible by the program (1) and for the choice sets in the selection to be comparable (2).

For the first motive, the selection is comprehensible by the program, when there are not too many choice sets in the selection. An excessively large number of choice sets would cause the program to run for too long. As a rule of thumb, we will use a maximum of 5,000 choice sets at once. On the other hand, the number of choice sets should not be too low. In a small dataset, the influence of each choice set on the results would be too large, which may result in misleading results. Significance tests are done to verify the reliability of the results of the model.

The second motive was added in anticipation of differences in choice behaviour between different decision makers. We preselect certain decision makers based on the flight they initially look for. The results of the model that is trained on this selection is consequently only valid for a selection of decision makers that would also be in the selection.

The selection of choice sets is made by choosing a set of routes, a date range, and either one-way or return flights. All choice sets with the initial flight on one of these routes, searched for within the date range and the same choice of one-way or return, are gathered in a dataset. Consequently, the flights in each choice set are filtered as described in section 3.1.1. Then all choice sets in the dataset that contain both one-way and return flights are removed (see section 3.2.2). Finally, a selection is made on the number of flights in each choice set. Only choice sets that have at least two and no more than ten flights in them, are used in the modelling phase. This last filtering is done, because in any practical implementation of the models, one wouldn't create a choice set with over ten alternatives to choose from and choice sets of larger size have a larger influence on the likelihood function (see section 4.2.3.2).

4.2.1.2 Property Selection

About all of the choice sets and all of the flights in each of these sets, a selection of properties are calculated, a description of these properties is given in section 3.2. This selection should be made such that there is little correlation between them. Taking both *Flight-Value* and *Flight-Value Per Kilometer*, would cause the effect of one of the two to be cancelled by the effect of the other and one of the two would be redundant.

4.2.1.3 Cross-Validation

After a dataset is retrieved, it is split into four subsets for cross-validation. The split is done in such a way that each subset contains approximately the same number of choice sets and on average each subset has approximately the same number of flights per choice set. Consequently, the program performs four runs. In each run, it trains the chosen choice model on three of the four subsets and tests the performance of the model on the fourth subset. This allows us to see how the model performs on predicting choice behaviour by decision makers that the model hasn't seen in the training process.

4.2.2 Model Training

Both choice models make use of a set of parameters for which the optimal values need to be found. For the MNL model, there is one parameter, corresponding to each of the selected flight properties. For the LC-MNL model, there are K parameters for each of the selected flight properties and K parameters for each of the selected decision maker properties. These optimal values for these parameter, are those that best describe how the decision makers in a dataset respond to the flights' properties when making a choice out of their respective choice sets.

The values for the parameters that best describe this response, are those that make the log-likelihood as high as possible. In each training run, an initial value is assumed for each parameter. Consequently, an iterative optimization method is applied to seek the maximum value for the log-likelihood function. We will describe these two in reversed order, to better understand the impact of the initial values on the optimization method.

4.2.2.1 Optimization Method

Given a choice model, an optimization method is used that tries to find the optimal set of parameters. These parameters are those that give the maximum value for the log-likelihood. A maximum is found where the derivative to the log likelihood function is as close as possible to zero.

In practice, the optimization method takes a dataset, a set of initial parameter values, the model's log likelihood function and preferably[1] also the function giving its gradient vector. At each iteration, it calculates the current value for the log-likelihood function and tries to find a new set of parameter values that would improve the log-likelihood. It keeps doing so, until some convergence criterion is met.

To find the optimal set of parameters, we have chosen to use the limited memory Broyden-Fletcher-Goldfard-Shanno (BFGS) algorithm for bound-constrained optimization[2] (L-BFGS-B,[5]). The original BFGS is the default algorithm in the optimization routines of many commercial software packages ([2], page 202). The L-BFGS-B is used, because it uses a smaller amount of computer memory (especially preferable for models with more parameters) and allows us to indicate bounds by which the parameter values are logically constrained.

The L-BFGS-B has two stopping criteria. The first stopping criterion is met when all of the values in the gradient vector are below some chosen value *pgtol*. The second stopping criterion is met when the increase in the log likelihood function between iterations is smaller than some value $f * eps$, where the f can be chosen and eps is the machine's precision (2.2e-16). In our testing, we use $f = 1e6$ and $pgtol = 0.001$

4.2.2.2 Initial Values

After one of the stopping criteria is met, the optimization method is believed to have reached a maximum. However, this is not always the global maximum. It may well be, the optimization method has found a local maximum or an area where the log-likelihood

[1] Providing the optimization method with the gradient function, saves the time that it would otherwise spend to approximate the gradient at each iteration.

[2] Implemented by using the scipy.optimize.fmin_l_bfgs_b Python function (source: `http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.optimize.fmin_l_bfgs_b.html`)

function is almost flat. To avoid these circumstances, initial values should be set that are close to the global maximum.

As stated in [1] (page 119), the log-likelihood for the MNL function, is globally concave, which means it should not matter which initial values we take. The log-likelihood function for the LC-MNL model however, does not have to be globally concave. To find initial values for the LC-MNL model, we will split the training set up into K different subsets, based on decision maker properties. Then we train MNL models on each of these subsets and use the optimal values of these MNL models as initial values for the flight selection parameters of the LC-MNL model. Consequently, we set the initial values for the class selection parameters in such a way that it is most likely that the members of the subsets will fall into that class.

4.2.3 Performance Measures

After the training phase, the model's performance is tested on both the training set and the test set. The performance on the training set will give an impression of how well the model fit to the data on which it was trained. The performance on the test set will give an impression of how well the model fits to data on which it was not trained. The latter is often called out-of-sample performance and is the best indicator of how well a model would do, predicting the outcome of new choices. In this section, different performance measures are described. They are reported on in chapter 5.

4.2.3.1 Akaike Information Criterion

The Akaike Information Criterion (AIC, [6]) is a measure for goodness of fit of a statistical model, correcting for the complexity of that model. It allows for comparison of different models that are built on the same dataset. It is calculated as:

$$AIC = -2 * LL(\boldsymbol{\beta}) + 2 * K, \tag{4.1}$$

Where $\boldsymbol{\beta}$ contains all the model parameters and K is the total number of parameters used. Generally, the model with the lowest AIC score is considered to be the best.

4.2.3.2 Likelihood Ratio

The Likelihood Ratio index (LRI) measure gives an indication of how much a model has improved over a random model to a model that is able to perfectly predict each choice. It is calculated as follows:

$$LRI = 1 - \frac{LL(\beta)}{LL(0)} \tag{4.2}$$

Where LL for a certain choice model is calculated as described in the model's respective section of chapter 2 and quantifies log-likelihood the model attributes to the chosen alternative, summed over all choice situations. $LL(0)$ is the log-likelihood for a model that attributes the same probability to each alternative in a choice situation (you get this model by setting each parameter to 0). $LL(\beta)$ is the log-likelihood for the trained model. Hence, when comparing two models estimated on the same data, it is usually valid to say that the model with the higher LRI fits the data better. Two models estimated on samples that are not identical should not be compared via their likelihood ratio index values ([2], page 72).

4.2.3.3 Prediction Ratio

The prediction ratio (PR) gives an indication of how many times the model would predict the right alternative to be chosen from a set.

$$PR = \frac{\text{\#times model gave highest probability to chosen alternative}}{N} \tag{4.3}$$

This measure should be interpreted with caution. It is based on the idea that the decision maker is predicted by the model to choose the alternative for which the model gives the highest probability, while in fact the model only predicts the decision maker to choose this alternative a certain proportion of times, if the choice was made multiple times ([2], page 73).

4.2.3.4 Negative Prediction Ratio

The prediction ratio (NPR) gives an indication of how many times the model would not predict the chosen alternative to be the least likely alternative to be chosen from a set.

$$NPR = 1 - \frac{\#\text{times model gave lowest probability to chosen alternative}}{N} \tag{4.4}$$

4.2.3.5 Top-50% Prediction Ratio

The top 50% prediction ratio ($T50P$) describes how many times a chosen alternative was amongst the top 50% alternatives based on probability attributed by the model. The measure ranges between 0 and 1. A value of approximately 0.5 would indicate that 50% of the times, the model was correct on a 50/50 coin toss, meaning the model was no better than a random selection.

$$T50P = \frac{\#\text{chosen alternative ranked amongst top 50\%}}{N} \tag{4.5}$$

In case of an uneven number of alternatives, and the chosen alternative is the middle alternative on attributed probability, this counts for 0.5. This success measure betters allows for comparing choice sets of different sizes.

4.2.4 Significance Tests

After training a choice model, different hypotheses can be tested on both the model and its parameters.

4.2.4.1 Model Significance

For a choice model, we will test whether or not a certain model is significantly better than another. This can be tested using a likelihood ratio test ([2], page 74). For this test, two models are built, one restricted model and one without restrictions. These result in two different scores for the log likelihood, namely $LL(\hat{\beta})^H$ for the restricted model and $LL(\hat{\beta})$ for the unrestricted model. The test statistic is $-2(LL(\hat{\beta}^H) - LL(\hat{\beta}))$ and is distributed chi-squared with degrees of freedom equal to the number of restrictions implied by the null hypothesis.

The most common test will be to test if all parameters in the model are in fact equal to zero. To test this, one calculates $-2(LL(0) - LL(\hat{\beta}))$ to perform a chi-squared test with the degrees of freedom equal to the number of parameters used in the model.

4.2.4.2 Parameter Significance

To test whether or not single parameters differ significantly from a certain value, standard t-tests can be used. As a standard, for each of the parameters used in a model, a t-test is performed to see if the parameter value differs significantly from zero. Before calculating the t-test statistics, the variance for each of the parameters needs to be calculated.

To calculate the variance we need to introduce a few terms, this section draws heavily on the book by Train ([2], section 8.3.2 and 8.6). The score of an observation is the derivative of that observation's log likelihood with respect to the parameters: $s_n(\beta_t) = \delta ln P_n(\beta)/\delta\beta$ evaluated at β_t. In our case these observations are choice situations. Then, for any model for which the expected score is zero at the true parameters,

$$\sqrt{N}(\hat{\beta} - \beta^*) \xrightarrow{d} N(0, \mathbf{H^{-1}VH^{-1}}), \tag{4.6}$$

where $\hat{\beta}$ are the parameters found by the model, β^* are the parameters at their true value, $\mathbf{V}$ is the variance of the scores in the population and $\mathbf{H^{-1}}$ is the inverse of the expected Hessian in the population. This means, that $\mathbf{H^{-1}VH^{-1}}/N$ is the covariance matrix of $\hat{\beta}$. This covariance matrix can be estimated by using the so-called "sandwich" estimator of the covariance matrix $H^{-1}BH^{-1}/N$ (often called the robust covariance matrix), where H is the average Hessian in the sample and B is the average outer product over all scores in the sample.

4.3 Settings

To find the answers to the first three of our research questions:

1. When predicting a choice between several flights, what information best helps predict this choice?
2. Can we detect differences in choice behaviour between decision makers?
3. If so, can we exploit these differences?

We have split the implementation into three consecutive phases:

1. Finding the optimal set of flight properties,
2. Finding differences between decision makers and
3. Exploiting the differences between decision makers

Before starting with phase 1, a dataset is retrieved. All choice sets that have as initial flights, return flights departing from Amsterdam, Eindhoven or Rotterdam and arriving at Alicante, Barcelona, Girona or Valencia that were searched for in July 2014 are retrieved from the database. By selecting these airports, we are specifically studying choice behaviour that is displayed by people who, in the month of July, start searching for flights from the Netherlands to the east-coast of Spain.

4.3.1 Finding the optimal set of flight properties

Subsequently, to answer the first research question, the MNL model was trained with different sets of flight properties, to find the set of flight properties that best help predict choice behaviour. The flight properties are grouped into the following groups:

Price *Flight-Value*, *Flight-Value per Kilometer*

Distance Between Departure Airports *Distance Between Departure Airports*

Departure Airports *Departure* {*IATA*}

Distance Between Arrival Airports *Distance Between Arrival Airports* and *Distance Between Arrival Airports Capped at* $\{max\}$

Arrival Airports *Arrival* $\{IATA\}$

Departure Date *Departure Date Deviation*

Length of Stay *LOS Deviation* and *LOS Relative Deviation*

Other Costs *Numbeo Hotel Price Index*, *Numbeo Travel Cost Per Day* and *Numbeo Total Travel Cost*

For the properties in groups *Price*, *Distance Between Departure Airports*, *Distance Between Arrival Airports* and *Other Costs*, by logic, if for any flight the properties gain a higher value, the likelihood of that flight being chosen should decrease. Therefore, these properties' respective parameters in any model should be smaller than zero. This is implemented by setting the upper bound for these parameters at 0.

Maximum one property per group should be used in the models to avoid redundancy, with the exception of the airport specific indicators *Departure* $\{IATA\}$ and *Arrival* $\{IATA\}$. All of the flight properties listed above are candidates to be included in the optimal set. During this phase, we are looking to find the best property in each group or eliminate all of the properties in each group.

In the first step in phase 1, a choice is made for the initial set of properties. This initial set of properties are used to run the first MNL model. After this run, the output of the model is be interpreted. Is the model significantly better than having no model? Are there any parameters that are not adding to the model's predictive power?

In the following step, an attempt is made to improve the best model so far. Different flight properties are used, changing at maximum one group at a time. If the model built on the new set of properties, improves on the best model so far on out-of-sample performance, then the new set of properties is better than the old. The flight property that was in the worst set of the two, is then eliminated as candidate for the optimal set of properties. This step is repeated until there are no more properties left to try.

4.3.2 Finding Differences Between Decision Makers

Each of these properties in the optimal set from phase 1 has a corresponding parameter in the MNL model. These parameters indicate how decision makers respond to the flight properties. The three decision maker properties described in chapter 3 were *Paid Channel*, *Days to Flight* and *Saturday Night Stay*. In phase 2, we split up the dataset based on decision maker properties, to see if different decision makers respond to the flight properties differently.

In three separate runs, the dataset is split in two based on one of the decision maker properties. For the *Days to Flight* property, the split is made at the median, for the other two properties there are only two possible values, which makes for an easier split. In each run, MNL models are trained on both halves, to see if the decision makers in each of the halves respond differently to the flight properties. This difference can be seen by comparing parameter values.

4.3.3 Exploiting the Differences Between Decision Makers

In an attempt to exploit the differences in choice behaviour between decision makers that are found in phase 2, the LC-MNL model is introduced in phase 3. The number of classes K is hereby set to 3. We will use different settings for the initial parameter values, in search of the LC-MNL model with 3 classes, that best fits the dataset.

The out-of-sample performance of the LC-MNL models will than be compared to the performance of the best MNL model from phase 1. If there is an improvement on performance, this means we are able to exploit the differences between decision makers.

Chapter 5

Results

This chapter, describes the outcome of the research as it is described in chapter 4. Each of the three sections cover one of the phases that are described in section 4.3. In each of them, we will answer one of the research questions. We start with the search for the optimal set of flight properties that can be found in our dataset and best help predict choice behaviour, in section 5.1. We then use decision maker properties to see if different decision makers respond differently to certain flight properties, in section 5.2. Finally, we try to exploit these differences, in section 5.3.

5.1 Finding the Optimal Set of Flight Properties

CONFIDENTIAL

5.1.1 Final Selection

CONFIDENTIAL

5.2 Finding Differences Between Decision Makers

In phase 2, we will use the flight properties found in phase 1 to train a new set of models. As described in section 4.3.2, we split up the original dataset (see section 4.3) based on decision maker properties *Paid Channel*, *Days to Flight* and *Stay Over Saturday Night* and train separate MNL models on each of the splits. Consequently we compare the parameter values between the models to see if there is a difference in choice behaviour between different decision makers and answer the second research question: *Can we detect differences in choice behaviour between decision makers?*. To show that there is difference between decision makers, we will focus on the *Flight-Value* property in this section. The rest of the properties are described in appendix B.

For the *Flight-Value* property, parameters were calculated four times in each of a total of six splits. These parameter values, together with the parameter values found by training the models over the entire dataset (described in section 4.3) are shown in figure 5.1. In the graph we have chosen to name the split as follows:

Days to Flight

- *D2F<=40* All 1788 decision makers that searched for their initial flight at most 40 days in advance
- *D2F>40* All 1732 decision makers that searched for their initial flight more than 40 days in advance

Paid Channel

- *PC* All 1004 decision makers that searched for their initial flight after entering the website through a paid channel
- *UPC* All 2516 decision makers that searched for their initial flight after entering the website through an unpaid channel

Saturday Night Stay

- *SNS* All 2936 decision makers that initially searched for a return flight including a Saturday night stay

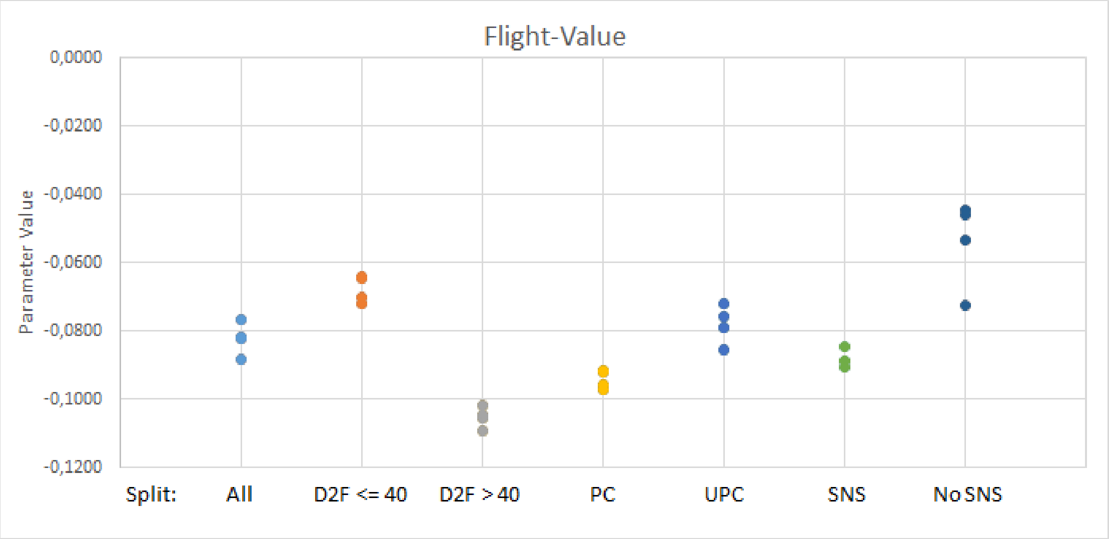


Figure 5.1: Flight-Value Parameter Values

- *No SNS* All 584 decision makers that initially searched for a return flight excluding a Saturday night stay,

where 40 was chosen in the *Days to Flight* split, because it was the median value in the original dataset. What stands out from figure 5.1, is that the parameter values for each of the splits, lie relatively close to each other, where the biggest spread between parameter values within splits, is for the *No SNS* split. This can be explained by noting that this split has the lowest amount of decision makers (584). Furthermore, we can detect differences between the splits.

5.2.1 Days to Flight

When comparing the *D2F<=40* split to the *D2F>40* split, we see that the decision makers that start their search for a flight further ahead of time, have lower parameter values for the *Flight-Value* property. This lower value indicates that these decision makers are more sensitive to changes in the price of a flight.

To confirm that indeed there is a difference in parameter values between the splits *D2F<=40* and *D2F>40*, we have performed one-way ANOVA analysis (see [7]) on the parameter values, the results of which are shown in table 5.1. We may reject the hypothesis that the mean parameter values for both splits are equal (F(1,6) = 224.62, p < .0001). To verify the normality assumption of the ANOVA model, we performed a

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
split	1	0.00	0.00	224.62	0.0000
Residuals	6	0.00	0.00		

Table 5.1: ANOVA on Flight-Value Parameters for split on Days to Flight

Shapiro Wilk test (see [7]) on the residuals. This test did not reject the hypothesis that the residuals are normally distributed (W = 0.8733, p = .1622), hence we can interpret the results of the ANOVA as described above.

5.2.2 Paid Channel

If we look at the *PC* and the *UPC* splits, the difference between parameter values is smaller than in the previous comparison, but there does seem to be a difference. The *PC* split's parameter values are lower than the *UPC* parameter values, indicating that decision makers that start their search for a flight after entering the website through a paid channel are more sensitive to changes in the price of a flight than decision makers that start their search through an unpaid channel. Entering the website through a paid channel, means having clicked on an advertisement, which may be seen as behaviour that is provoked. Entering through an unpaid channel means having either used a search engine or the direct address, which may be seen as more self-initialized. More research should be done to investigate this difference in parameter values, but it seems that the people that initialize their search for a flight by themselves are less sensitive to changes in price. These differences in parameters may also appear, because the flights that people saw through these paid channels, were priced aggressively low. This would mean these flights prices are lower than 'usual' prices. If a decision maker would then compare these flights to a flight with a 'usual' price, there would be a big difference and therefore more reason to pick the cheapest one.

Again, we have performed one-way ANOVA analysis on the two splits' parameter values, the results of the analysis are shown in table 5.2. We may reject the hypothesis that the mean parameter values for both splits are equal (F(1,6) = 25.60, p = .0023). The Shapiro Wilk test on the residuals did not reject the hypothesis that the residuals are normally distributed (W = 0.9628, p = .8364).

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
split	1	0.00	0.00	25.60	0.0023
Residuals	6	0.00	0.00		

Table 5.2: ANOVA on Flight-Value Parameters for split on Paid Channel

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
split	1	0.00	0.00	27.05	0.0020
Residuals	6	0.00	0.00		

Table 5.3: ANOVA on Flight-Value Parameters for split on Paid Channel

5.2.3 Saturday Night Stay

The split that was made on *Saturday Night Stay*, is one that is often used to distinguish between business (*No SNS*) and leisure (*SNS*) travellers, because business travellers don't stay at their destination over the weekend. Based on the parameter values, it seems that people that do want to stay over Saturday night, are more sensitive to price changes than people that do not.

Once again, we have performed one-way ANOVA analysis on the two splits' parameter values, the results of this analysis are shown in table 5.3. We may reject the hypothesis that the mean parameter values for both splits are equal (F(1,6) = 27.05, p = .0020). The Shapiro Wilk test on the residuals did not reject the hypothesis that the residuals are normally distributed (W = 0.8564, p = .1106).

5.2.4 Conclusion

Based on what we saw in figure 5.1 and the analyses that are described above, we can conclude that the *Flight-Value* property parameters were not equal for all decision makers. This means that in our dataset, not all decision makers are equally sensitive to differences in the price for flight tickets. We were able to detect a difference in sensitivity to the *Flight-Value* property amongst decision makers.

In figure 5.2, you can find the average parameter values for each of the properties that were found in phase one. One can see that not just the parameters for the flight property differ per split. As stated above, more information about these differences can be found in appendix B.

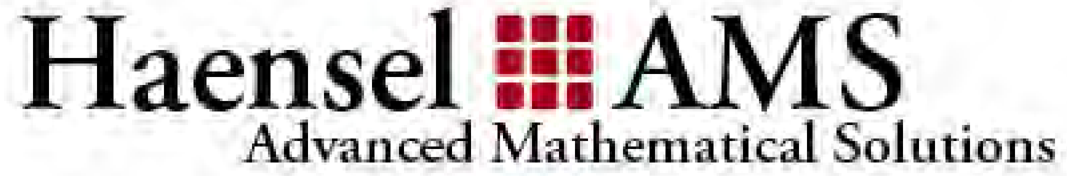


FIGURE 5.2: Figure confidential!

5.3 Exploiting the differences between decision makers

In phase 3, we try to exploit the differences that were found in phase 2, by training choice models that better predict choice behaviour. This attempt is split up into two parts. In the first, we use the hard splits on decision maker properties described in section 5.2 and verify if making these splits and training separate models, improves the prediction. In the second, we introduce classes to which each decision maker belongs with a certain probability and train LC-MNL models over the entire dataset.

5.3.1 Decision Maker Specific Models

In the previous phase, we have split up the dataset based on three decision maker properties. Three separate times, the dataset was divided in two splits (based on different decision maker properties), making it that each decision maker would only be in one of those two splits each time. Consequently, separate MNL models were built on each split, using 4-fold cross-validation as was done in phase one.

The results of the models that were built on the split dataset were aggregated, such that they can be compared to each other and the MNL model on the entire dataset. You can find these aggregated results at the bottom of the table in figure 5.2. The AIC was summed over the two splits, and punished for an extra parameter (the one that was used to make the split). For the other measures, the average was taken, weighted by the number of choice sets (shown at the top of the table).

What stands out, is that the splits that were made on the *Paid Channel* and *Saturday Night Stay* decision maker properties, did not result in more accurate models than the MNL models that were trained on the entire dataset. In fact, making these splits resulted in less accurate models, indicated by the higher *AIC* and lower *LRI* and prediction ratio scores.

On the contrary, by separating the decision makers that start searching early from the ones that start searching late, and training separate models on each split, we were able to lower the *AIC* to 7524.58 (even though we doubled the number of parameters) and increase the LRI to .0600, which indicates that these models were better able to predict choice behaviour better than the models trained over all decision makers were. However, all three prediction ratio measures show a slight decrease, indicating that the models that were trained on the split datasets were less accurate in their prediction.

5.3.2 Latent Class Models

We now introduce the LC-MNL model that is described in section 2.3. This model allows for the analyst to create a number of classes to which decision makers may belong. Each decision maker belongs to each class with a certain probability, which is calculated with the decision maker properties. In contrast to the hard splits that were made in section 5.3.1, now each a decision maker may be part of multiple classes. Each of these classes have a specific parameter for each of the flight properties that are used. We use the selection of flight properties that were found in phase 1. Again, we use 4-fold cross-validation to evaluate the models' performance.

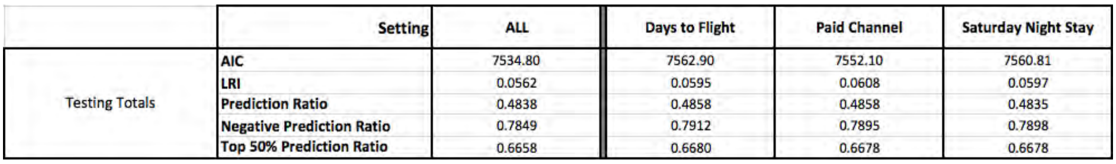
5.3.2.1 Initial Parameter Values

We have chosen to use 3 classes to see if these LC-MNL models perform better than the MNL models. Furthermore, we have tested the LC-MNL model under 3 different settings, defined by the initial values for the parameters (see section 4.2.2.2). We will call these settings *Days to Flight*, *Paid Channel* and *Saturday Night Stay*.

For setting *Days to Flight*, we have set the initial values for the class selection parameters, such that with high probability, the people that start searching early are in class 1, and people that start searching late are in class 3, leaving class 2 as the base class. This is implemented by setting $\boldsymbol{\lambda}_1 = (-0.1, 0, 0)$, $\boldsymbol{\lambda}_2 = (0, 0, 0)$ and $\boldsymbol{\lambda}_1 = (0.1, 0, 0)$.

Consequently, we have set the initial values for the alternative selection parameters, by using the average parameter values from figure 5.2. So $\boldsymbol{\beta}_1$ contains the average parameters for split *D2F<=40*, $\boldsymbol{\beta}_2$ the average parameters over the entire dataset and $\boldsymbol{\beta}3$ the average parameters for split *D2F>40*.

Settings *Paid Channel* and *Saturday Night Stay* have been set up in a similar fashion. After setting these initial values, the models were allowed to alter all parameters in search of the set of parameters that best fit the data. These parameters would result in the highest log-likelihood. Under all three settings, four LC-MNL model were trained on



	Setting	ALL	Days to Flight	Paid Channel	Saturday Night Stay
Testing Totals	AIC	7534.80	7562.90	7552.10	7560.81
	LRI	0.0562	0.0595	0.0608	0.0597
	Prediction Ratio	0.4838	0.4858	0.4858	0.4835
	Negative Prediction Ratio	0.7849	0.7912	0.7895	0.7898
	Top 50% Prediction Ratio	0.6658	0.6680	0.6678	0.6678

Figure 5.3: LC-MNL Results

the exact same dataset and taking into account the same properties. So, if in all of the three settings, the optimal set of parameters would be found, these would be the same under all three settings. However this optimum might not be found due to the LC-MNL model's log-likelihood function not being globally concave (see section 4.2.2.2).

5.3.2.2 Results

In figure 5.3, you can find the results of the three settings. These are aggregated in the same way as was done in phase 1. We can see that the results are different between the settings. Hence, the initial values indeed have an impact on the performance of the LC-MNL models. Based on *AIC* and *LRI* (which are equivalent in this case, because all settings used the same data and number of parameters), we can see that the *Paid Channel* setting (*AIC* 7552.10, LRI .0608), resulted in models that fit the data better than the *Days to Flight* (*AIC* 7562.90, LRI .0595) and the *Saturday Night Stay* (*AIC* 7560.81, LRI .0597) did. Furthermore, the results for this setting, show an increase in LRI compared to the overall MNL model from phase 1 and the decision maker specific MNL models from section 5.3.1. This indicates, we could exploit the differences between decision makers by implementing the LC-MNL models. However, when looking at the *AIC*, this is higher than the values found for the MNL models, indicating that the increase in *LRI* is outweighed by the increase in the number of parameters. Furthermore, the prediction ratio measures show that the LC-MNL models gave better predictions than the overall MNL model did. The three different LC-MNL models showed approximately equal values, with the *Days to Flight* showing a slight advantage over the other two. We assume the results from the *Paid Channel* setting as the best out of the LC-MNL models for the final comparison.

The development of the log-likelihood for the four model that were trained for setting *Paid Channel* is shown in 5.4. In each of the models that were run, the optimization

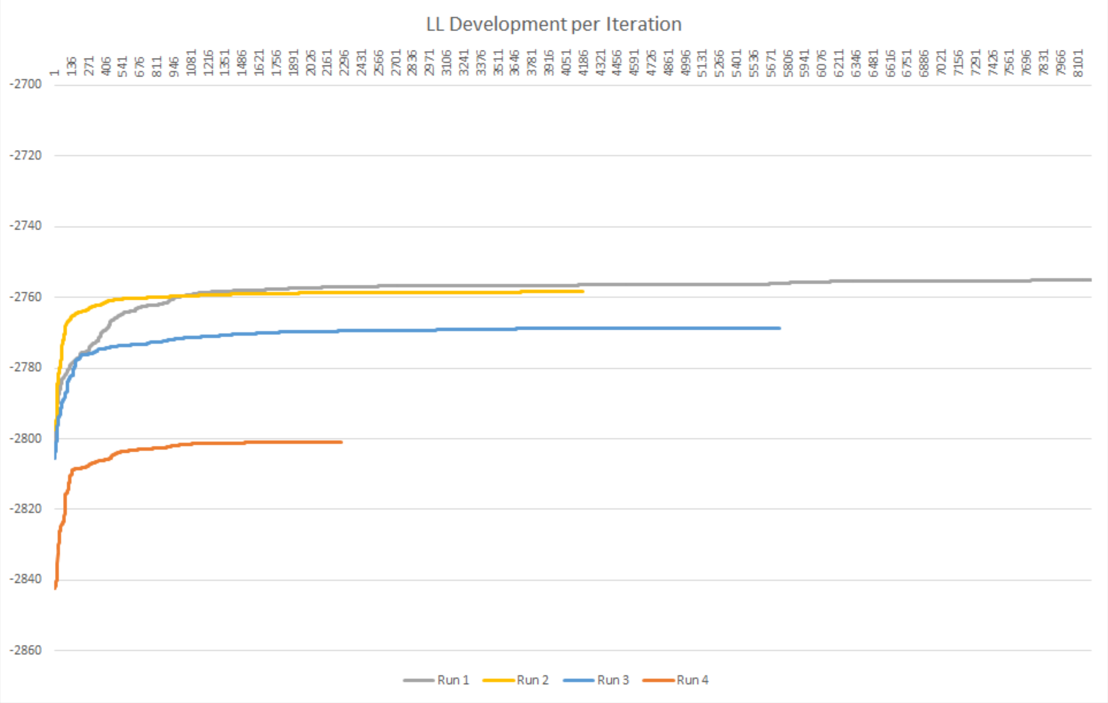
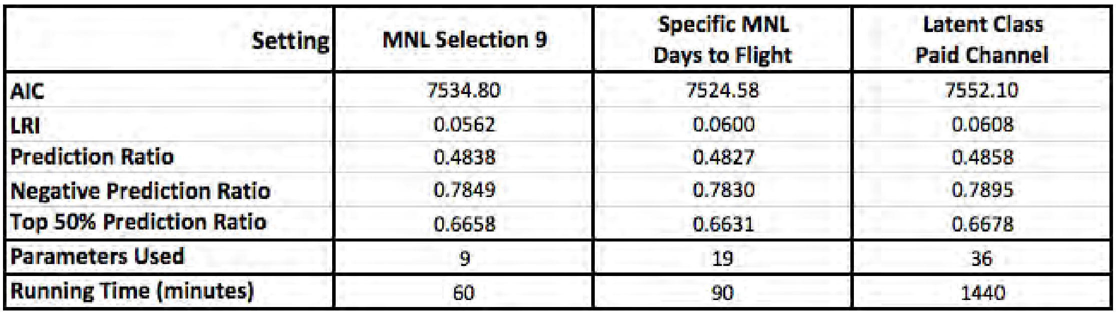


FIGURE 5.4: Log-likelihood Development

method (described in section 4.2.2.1) stopped at a point where the difference in log-likelihood between one iteration and the next was smaller than $2.2e-10$. This was also the case for the other settings. However, at this moment, it was not always the case that the gradients for the parameters were close enough to zero. Hence, the optimization did not find the optimal values for the parameters. This indicates that the initial values we have used did not help the optimization method find the global optimum and other methods may be used to set the initial values such that an even better log-likelihood value can be found.



Setting	MNL Selection 9	Specific MNL Days to Flight	Latent Class Paid Channel
AIC	7534.80	7524.58	7552.10
LRI	0.0562	0.0600	0.0608
Prediction Ratio	0.4838	0.4827	0.4858
Negative Prediction Ratio	0.7849	0.7830	0.7895
Top 50% Prediction Ratio	0.6658	0.6631	0.6678
Parameters Used	9	19	36
Running Time (minutes)	60	90	1440

Figure 5.5: Method Comparison

5.4 Recap

With one specific dataset we have (1) trained MNL models using different properties to find the optimal set of properties, (2) split up the dataset to detect differences between decision makers and (3) tried to exploit these differences. We have put together the testing results for the best models that were trained in these 3 phases. These can be seen in figure 5.5.

We added two extra rows, one that shows the number of parameters that is used and another that shows how much time the computer has run to find the final set of parameter values.

The results show that we were able to train the computer to predict choice behaviour better than a random model would. This is indicated by the LRI scores, they are all greater than zero. Furthermore, we were able to make this prediction better by making use of information about what the decision makers initially searched for, this can be seen by comparing LRI scores for the decision maker specific MNL models (.0600) and the LC-MNL models (.0608) with that of the MNL models for selection 9 (.0562). The LRI indicates that the Latent Class models were able to best predict choice behaviour. However, they have done so using the highest number of parameters. This makes it that the *AIC* score is actually worse for the LC-MNL models and that these took the most computing time.

When looking at the prediction ratio measures, we see that the LC-MNL models gave higher scores than the MNL models did. Overall, we see that the LC-MNL models give the best prediction, although this comes at the price of using more parameters and therefore computing time.

Chapter 6

Conclusions & Further Research

6.1 Conclusions

In this research, we have used website traffic data and extracted different choice sets from this data. Consequently, we have trained different choice models to understand this choice behaviour and make predictions.

To answer the first research question: *"When predicting a choice between several flights, what information best helps predict this choice?"*; we have found that from the original dataset, the flights' price, the location of the airports, the departure dates and the length of stay helped us in predicting this choice behaviour. By adding travel cost data from an open source, we were able to further improve our predictions.

The second research question: *"Can we detect differences in choice behaviour between decision makers?"*; was answered by splitting up the dataset by using information about the first time decision makers searched for a flight. Indeed, we were able to detect differences in choice behaviour between decision makers. The biggest difference was found between the decision makers that start searching for flights far in advance and those that start searching close to the departure date.

Consequently, we have exploited these differences between decision makers to further improve our predictions to answer research question 3: *"If so, can we exploit these differences?"*. We have found that both making a hard split on decision makers based on how far in advance they start searching and building a separate MNL model for both splits, improves the prediction over the one MNL model trained over all decision makers. Introducing latent classes and training an LC-MNL model over all decision makers, improved the prediction even further, however, this model costed more time to train.

We have studied website traffic data to gain a better understanding of customer choice behaviour, thereby exploring possibilities with e-commerce tracking data. Using discrete choice theory, we have implemented a method to predict choice probabilities for alternatives in a choice set, conditional to the fact that the customer will choose exactly one alternative from this set. The airline in question may use our methods to personalize the offers they make to customers, by selecting the most relevant flights and set those prices that would give the maximum expected return. However, more testing should be done before we can make predictions about the expected effect that would have.

Furthermore, the improvement of the choice predictions for the optimal model over the zero model, is significant, but not necessarily very large. This might cause the effects of using the results of the optimal model not to be very large as well. We therefore recommend to further develop the methods, before using them. We will describe suggestions for how to do so in the next section.

6.2 Limitations

With the way our research uses the website traffic data, there are some things that need to be kept in mind. The data does not necessarily reflect the entire choice set for each decision maker. The models we used, can therefore only give an indication of how probable it is for a flight to be chosen, given that in fact one of the flights in the generated set is chosen.

To come to the choice sets that were studied, we have made conditions to leave out irrelevant alternatives from each set. This clearly affects the resulting choice sets and, in particular, the first alternative. This first alternative has consequently played an important role in calculating both the flight and the decision maker properties. Hence, the conditions may be re-evaluated in further applications.

Furthermore, the data does not include all information decision makers use to make a choice, e.g., competitors' flight prices, purchase history, special events, school holidays, weather conditions . Should more data come available, the set of optimal properties found in 5.1 should be re-evaluated.

Moreover, for testing our models, we have used one specific set of decision makers. The optimal set of properties might be different for another set of decision makers. We have also only implemented our models with one specific product, namely flights. For other products, one must keep in mind the assumptions of the choice models and evaluate if using the same methods is a valid approach.

The travel cost data that was used came from an open source and we do not stand in for it's quality. Moreover, the travel costs were calculated based on the city closest to the arrival airport, while in fact, the decision maker might stay elsewhere (and at another cost). Furthermore, the travel costs were only determined for one specific point in time (mid-2014). Since the travel cost information was shown to improve the prediction, it might pay off to find a reliable source for travel cost data that also includes seasonality.

6.3 Further Research

Looking back at the choices that were made during this research and the final results, we now answer the final research question: *"What can be done to make an even better prediction?"* and describe other opportunities for further research.

In phase 3 of our research, we have seen that splitting up the dataset on decision makers, can improve the prediction. Should more information come available about the decision makers, it might pay off to use a clustering algorithm to create clusters of decision makers. Consequently, different choice models may be trained for each cluster.

In the second part of phase 3, we have seen that the outcome of the LC-MNL model depends on the initial values for the parameters. Given more time, we would have like to implement a way to find those initial values that make it that a global optimum can be found. We would suggest to try a number of different sets of initial values and letting the optimization method run for a limited amount of iterations, after starting at each of these sets. We have implemented a way of tracking the development of the log-likelihood, which may help to determine the iteration limit. Consequently, more iterations may then be used for those sets that show the highest log-likelihood.

Furthermore, our study has only used information about choices situations in which the decision makers have actually made a choice. As stated above, this makes it that the resulting choice probabilities rely on the condition that a decision maker will actually choose one from the set of alternatives. In a commercial implementation one might want to know the expected outcome of giving a certain offer to a potential customer. To do so, an extra method should be used to predict the probability with which a customer would choose to buy. The current research could be used to estimate which flight in the offer would be chosen.

In phase two of our research, we have seen that the channel through which a decision maker comes to the website when searching for the first flight allowed us to split up decision makers into two groups. Making this split showed us that the two groups behave in a different way when it comes to making a choice. More research may be done to further investigate the effect of the channel not just on choice behaviour, but on the characteristics of the alternatives in the choice sets.

Appendix A

Gradient Calculations

A.1 LC-MNL

This section describes how the gradient of the Log-Likelihood function is calculated for the LC-MNL model. Remember from section 2.3:

$$P_n(i|v_n = k) = \pi_{ni|k} = \frac{\exp(\boldsymbol{\beta'_k x_{in}})}{\sum_{j \in C_n} \exp(\boldsymbol{\beta'_k x_{jn}})} \tag{A.1}$$

And

$$P_n(v_n = k) = \pi_{nk} = \frac{\exp(\boldsymbol{\lambda'_k z_n})}{\sum_{m=1}^{K} \exp(\boldsymbol{\lambda'_m z_n})} \tag{A.2}$$

The likelihood L of the LC-MNL model is:

$$L = \prod_{n=1}^{N} L_n \tag{A.3}$$

Where L_n is the likelihood per choice situation n, is the average likelihood for choice situation n per class k, weighted by the probability of choice situation n belonging to class k. We assume independence between choice situations.

$$L_n = \sum_{k=1}^{K} \pi_{nk} L_{nk} \tag{A.4}$$

$$L_{n|k} = \prod_{i \in C_n} (\pi_{ni|k})^{y_n i} = \frac{\exp(\boldsymbol{\beta_k' x_{ni*}})}{\sum_{j \in C_n} \exp(\boldsymbol{\beta_k' x_{nj}})}$$

$$= \frac{1}{\sum_{j \in C_n} \exp(\boldsymbol{\beta_k' x_{nj}} - \boldsymbol{\beta_k' x_{ni*}})} \tag{A.5}$$

Where $\boldsymbol{x}_{ni*}$ is the independent variable vector corresponding to the chosen alternative in choice situation n.

We want to know the gradients of the log likelihood function:

$$LL = \sum_{n=1}^{N} ln(L_n) = \sum_{n=1}^{N} LL_n \tag{A.6}$$

$$\begin{aligned} LL_n &= ln\left(\sum_{k=1}^{K} \pi_{nk} \prod_{i \in C_n} (\pi_{ni|k})^{y_{ni}}\right) \\ &= ln\left(\sum_{k=1}^{K} \pi_{nk} L_{n|k}\right) \\ &= ln\left(\sum_{k=1}^{K} \left[\frac{\exp(\boldsymbol{\lambda_k' z_n})}{\sum_{m=1}^{K} \exp(\boldsymbol{\lambda_m' z_n})} L_{n|k}\right]\right) \\ &= ln\left(\sum_{k=1}^{K} \exp(\boldsymbol{\lambda_k' z_n}) L_{n|k}\right) - ln\left(\sum_{m=1}^{K} \exp(\boldsymbol{\lambda_m' z_n})\right) \end{aligned} \tag{A.7}$$

The gradient vectors $\boldsymbol{\nabla LL_n(\lambda_k)}$ with respect to the class selection parameters consist of:

$$\frac{\delta(LL_n)}{\delta(\lambda_{ka})} = \frac{z_{na} \times \exp(\boldsymbol{\lambda_k' z_n}) L_{n|k}}{\sum_{l=1}^{K} \exp(\boldsymbol{\lambda_l' z_n}) L_{n|l}} - \frac{z_{na} \times \exp(\boldsymbol{\lambda_k' z_n})}{\sum_{m=1}^{K} \exp(\boldsymbol{\lambda_m' z_n})} \tag{A.8}$$

$$= z_{na} \left(\frac{L_{n|k}}{\sum_{l=1}^{K} \exp(\boldsymbol{\lambda_l' z_n} - \boldsymbol{\lambda_k' z_n}) L_{n|l}} - \frac{1}{\sum_{m=1}^{K} \exp(\boldsymbol{\lambda_m' z_n} - \boldsymbol{\lambda_k' z_n})}\right) \tag{A.9}$$

$$\forall\ k \in \{1, \ldots, K\},\ a \in \{1, \ldots, A\} \tag{A.10}$$

Where the extra step is taken to avoid unnecessarily high numbers in the exponential (which may cause rounding errors). The gradient vectors $\boldsymbol{\nabla LL_n(\beta_k)}$ with respect to the product preference parameters consist of:

$$\begin{aligned}\frac{\delta(LL_n)}{\delta(\beta_{kb})} &= \frac{\exp(\boldsymbol{\lambda'_k z_n})\frac{\delta(L_{n|k})}{\delta(\beta_{kb})}}{\sum_{l=1}^{K}\exp(\boldsymbol{\lambda'_l z_n})L_{n|l}} \forall\ b \in \{1,\ldots,B\},\ k \in \{1,\ldots,K\} \\ &= \frac{\frac{\delta(L_{n|k})}{\delta(\beta_{kb})}}{\sum_{l=1}^{K}\exp\left(\boldsymbol{\lambda'_l z_n} - \boldsymbol{\lambda'_k z_n}\right) \times L_{n|l}} \end{aligned} \tag{A.11}$$

$$\frac{\delta(LL_{n|k})}{\delta(\beta_{kb})} = -\frac{\sum_{i\in C_n}\exp(\boldsymbol{\beta'_k x_{ni}} - \boldsymbol{\beta'_k x_{ni*}}) * (x_{nib} - x_{ni*b})}{\left(\sum_{j\in C_n}\exp(\boldsymbol{\beta'_k x_{nj}} - \boldsymbol{\beta'_k x_{ni*}})\right)^2} \tag{A.12}$$

Appendix B

More Differences in Choice Behaviour

This appendix describes the differences in choice behaviour on the properties that were not discussed in section 5.2. We use same models of which the *Flight-Value* property was described in section 5.2, but in this appendix describe all of the other properties in less detail.

CONFIDENTIAL

Bibliography

[1] M. Ben-Akiva and S. R. Lerman. *Discrete Choice Analysis.* The MIT Press, 1985.

[2] K. E. Train. *Discrete Choice Methods with Simulation.* Cambridge University Press, 2003.

[3] R Luce. *Individual Choice Behavior: A Theoretical Analysis.* Wiley, New York, 1959.

[4] P. C. Boxall and W. L. Adamowicz. Understanding heterogeneous preferences in random utility models: A latent class approach. *Environmental and Resource Economics*, 23:421–446, 2002.

[5] R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5): 1190–1208, 1995.

[6] H. Bozgodan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, September 1987.

[7] M.C.M. de Gunst. *Statistical Models.* Vrije Universiteit Amsterdam, 2011.