

Learning Structure in Nested Logit Models

by

Youssef Medhat Aboutaleb

B.S., Korea Advanced Institute of Science and Technology (2017)

Submitted to the Department of Civil and Environmental Engineering and the Department
of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Science in Transportation

and

Masters of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

© Massachusetts Institute of Technology 2019. All rights reserved.

Author.....
Department of Civil and Environmental Engineering
Department of Electrical Engineering and Computer Science
May 17, 2019

Certified by.....
Moshe Ben-Akiva
Edmund K. Turner Professor of Civil and Environmental Engineering
Thesis Supervisor

Certified by.....
Patrick Jaillet
Dugald C. Jackson Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by.....
Heidi Nepf
Donald and Martha Harleman Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee

Accepted by.....
Leslie A. Kolodziejcki
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Learning Structure in Nested Logit Models

by

Youssef Medhat Aboutaleb

Submitted to the Department of Civil and Environmental Engineering and the Department of
Electrical Engineering and Computer Science
on May 17, 2019, in partial fulfillment of the
requirements for the degree of
Masters of Science in Transportation
and
Masters of Science in Electrical Engineering and Computer Science

Abstract

This work is about developing an estimation procedure for nested logit models that optimizes over the nesting structure in addition to the model parameters. Current estimation practices require an *a priori* specification of a nesting structure. We formulate the problem of learning an optimal nesting structure as a mixed integer nonlinear programming (MINLP) optimization problem and solve it using a variant of the linear outer approximation algorithm. We demonstrate that it is indeed possible to recover the nesting structure directly from the data by applying our method to synthetic and real datasets.

Thesis Supervisor: Moshe Ben-Akiva

Title: Edmund K. Turner Professor of Civil and Environmental Engineering

Thesis Supervisor: Patrick Jaillet

Title: Dugald C. Jackson Professor of Electrical Engineering and Computer Science

Acknowledgments

I start by acknowledging Professor Moshe Ben-Akiva. Thank you for giving me advice and encouragement when I needed it and the freedom to work on my ideas.

A very special thanks to Professor Patrick Jaillet for his insightful comments and encouragement at the start of the project. I can not thank him enough.

This thesis has also greatly benefited from discussions and courses taken with Professors Dimitris Bertsimas and Paul Barton.

I would like to thank Dr. Mazen Danaf who has been very gracious with his time.

I could not have imagined having better advisors and mentors for my graduate study.

I would like to mention my colleagues and friends at the ITS lab Yifei Xie, Siyu Chen, Eytan Gross, and Iveel Tsogsuren. I should also like to mention my dear friends Tim Adams, Hyunjoo Eom, Emil Shelestov, Tom Vasconcelos, Nurzhan Yergozhin and Azamat Kaliyev.

I owe a great amount of debt to KAIST Professors James Morrison, Kil Hyun Kwon, Hwasoo Yeo, Yoon Jin Yoon, and Kyungkuk Kim. I would have never made it to MIT without your mentorship and guidance.

Above all, I am most grateful to my mother and my family to whom this work is dedicated.

Contents

- 1 Introduction 13**
 - 1.1 Motivation 13
 - 1.2 Structure of the thesis 15
 - 1.3 Background and foundational results 16
 - 1.3.1 Random Utility Models 16
 - 1.3.2 Correlations, Partitions, Trees and Closed-form Probabilities 18
 - 1.4 Summary 30

- 2 Nested Logit Structure Learning Problem 33**
 - 2.1 Problem Statement 33
 - 2.2 A Mixed-Integer Nonlinear Program Formulation 34
 - 2.2.1 Objective Function 34
 - 2.2.2 Constraints 36
 - 2.3 Regularization 41
 - 2.4 Summary 44

- 3 Solution by Linear Outer Approximation 45**
 - 3.1 General algorithm overview 46
 - 3.2 Practical Matters 49

3.2.1	Evaluating the likelihood and its gradients	49
3.2.2	Dealing with Exponentially many constraints	52
3.2.3	Implementation Details	53
4	Experiments with real and synthetic datasets	55
4.1	Experiments with synthetic data	55
4.1.1	Data generation	55
4.1.2	Results	56
4.2	Experiments with real datasets	60
4.2.1	Data Description	60
4.2.2	Results	61
4.3	Summary	64
5	Conclusion	65

List of Figures

- 1-1 A tree representation of the nested partition in Example 2. The root node represents the set $S = \{1, 2, 3, 4, 5\}$. Each of the subsets constituting the partition, apart from the set S , are represented by a nest node. 19
- 1-2 The four possible non-degenerate nesting structures for the set $\{1, 2, 3\}$ 22
- 1-3 A nested partition over the set $\mathcal{S} = \{a_1, \dots, a_4\}$ and the equivalent nesting tree and variance-covariance matrix for a multinomial logit model. Notice that in this case the nested partition set \mathcal{B} is simply $\{\mathcal{S}\}$ 23
- 1-4 Multinomial Logit Model: The choice set can be represented by a rooted tree with height 1. 23
- 1-5 Tree representation of the nested partition for example 3 24
- 1-6 A nested partition over the set $\mathcal{S} = \{a_1, \dots, a_4\}$ and the equivalent nesting tree and variance-covariance matrix for a nested logit model 27
- 1-7 Nested Logit Model: For a choice set of size m , a nested logit model can be represented by a tree of height at most $m - 1$ (assuming non-degenerate nests) 29
- 1-8 Nesting structure for Example 4 30
- 2-1 Nesting structure underlying the synthetic data 39
- 2-2 Graph showing all possible edge connections. The edge labels are binary variables indicating their inclusion or exclusion in the estimated model 39
- 2-3 A case where it *is* possible to increase the number of nests without worsening the likelihood 42
- 2-4 Counter example: Increasing the number of nests does not always improve the training likelihood 42

2-5	A case where increasing the number of nests and nesting level each by one does not worsen the likelihood	43
2-6	A counter example: Increasing the number of nests and nesting level can worsen likelihood	43
3-1	Suppose a linearization cut is added around the red point. Since the feasible region is not a convex set, the hyperplane may cut off points of the feasible set. This can be avoided if the cut is translated appropriately.	48
4-1	Nesting structure underlying the synthetic data	55
4-2	A profile of the outer approximation algorithm in action for the nested logit optimization problem with <i>Number of Nests =4</i> and <i>Tree height =2</i>	56
4-3	Negative Log-likelihood values of the (local) optimal nesting structure under all feasible <i>Number of Nests</i> and <i>Tree Height</i> combinations.	57
4-4	The nesting structure with the best training objective function value of 710.68. The true nesting structure, came in second with objective value 711.42	57
4-5	Negative log-likelihood evaluated on the validation dataset	58
4-6	Nesting structure with best objective value of 722.41.	58
4-7	Training and validation negative likelihood profiles for the work travel mode choice logit model	62
4-8	Best performing model on training data (left), and on validation data (right)	62
4-9	A possible interpretation of the learned nesting structure	63

List of Tables

4.1	Estimation results for the synthetic dataset. The scale parameters are shown above each nest	59
-----	--	----

Chapter 1

Introduction

An object can have no value unless it has utility. No one will give anything for an article unless it yield him satisfaction. Doubtless people are sometimes foolish, and buy things, as children do, to please a moment's fancy; but at least they think at the moment that there is a wish to be gratified.

F.M. Taussing
Principles of Economics, 1912

1.1 Motivation

Nested logit is a popular approach in economics and transportation science when one wants to model the choice that an individual makes from a set of mutually exclusive alternatives [16] [3]. Unlike the simpler multinomial logit models, nested logit models provide a flexible modeling structure by allowing certain alternatives in the choice set to be correlated.

Nested logit models are attractive for a number of reasons including:

1. The nesting structure is *interpretable* and provides insight into the choice behavior under study (this is not a black-box machine learning model).
2. Nested logit models are easier to estimate (and consequently to optimize) than models with flexible error distributions (such as probit models) because the choice probabilities have a closed-form expression [6].

3. The elasticities of the choice probabilities with respect to choice attributes can be decomposed by “nest” effects, and substitution and complementary patterns between alternatives can be understood and used in making predictions.

In designing a nested logit model, the researcher hypothesizes a nesting structure over the choice set and proceeds to estimate the model parameters (the coefficients in the utility equations that determine the relative attractiveness of choices to the decision maker). Each nest is associated with a scale parameter (which is also estimated), and quantifies the degree of intra-nest correlation [3]. The nesting structure determines *how* the alternatives are correlated, and the scales determine by *what amount* they are correlated.

The large feasible set of possible nesting structures presents a significant modeling challenge in deciding which nesting structure best reflects the underlying choice behavior of the population. The current *modus operandi* is to use domain knowledge to substantially reduce the feasible set to a small set of candidate structures. This is done at the risk of potentially excluding some ostensibly non-intuitive structures which might actually provide a better description of the choice behaviour of the population under study [13]. This is essentially our core motivation for taking a more *holistic* view of nested logit model estimation, i.e., one that optimizes over structure as well as parameters.

Some of the early proponents of the nested logit model spoke of an “ultimate need” of a method to identify an optimal structure [9]. However, despite the model’s long history, no method has been proposed so far in the literature to learn structure of multi-level nested logit models directly from the data [7]. Heuristics do exist for the single-level nested logit. For example, one could estimate a cross-nested logit and assign the alternatives to the nest corresponding to its highest membership degree. Cross-nested logit models are however difficult to estimate because many more parameters are introduced (leading to a significant loss in estimation efficiency), and furthermore it is not possible to extend this idea to multiple levels of nesting.

The goal of this study is to introduce and model the nested logit structure learning problem as a mixed integer nonlinear programming (MINLP) problem. This entails optimizing not only over the parameters of the model (the traditional approach) but also over all valid nest structures (our proposed approach). In other words, *rather than assuming a nesting structure a priori, the goal is to reveal this structure from the data.*

As we shall see in Chapter 3, this problem turns out to be an *NP-hard* combinatorial problem -which means that, in general, finding a certificate of optimality in a reasonable amount of time can not be guaranteed (unless $P = NP$). Rather than adopting polynomial solvability as a measure of tractability, we instead follow the more pragmatic point of view by [1], and say that a problem is tractable if it can be solved “for sizes and in times that are appropriate for the application”. We also argue that arriving at a *good* solution, even if not globally optimum, greatly enhances the modeling power of nested logit models beyond its current state (which is essentially trial-and-error), and provides the researcher with a useful tool especially when the choice set is large, or the correlation structure between choices is complicated or not understood.

1.2 Structure of the thesis

This thesis is structured as follows:

- The rest of **Chapter 1** introduces nested logit models, their advantages over the simpler multinomial logit models. We also discuss at length the model's assumptions and limitations. This chapter also includes one of the main results of this thesis in the form of a theorem.
- **Chapter 2** presents the formal statement and full representation as an optimization problem of the nested logit structure learning problem.
- **Chapter 3** deals with solving the nested logit structure learning problem and addresses practical aspects of the solution methodology.
- **Chapter 4** presents results on the application of our method to synthetic and real datasets.
- **Chapter 5** summarizes the key messages of this thesis and includes some closing thoughts.

A small note regarding notation: boldfaced lowercase letters denote vectors, boldfaced capital letters denote matrices, ordinary lowercase letters denote scalars, ordinary capital letters denote random variables, and calligraphic type letters denotes sets.

1.3 Background and foundational results

1.3.1 Random Utility Models

Nested logit models belong to a bigger class of models known as random utility models. Random utility models rely on the assumption that the decision maker ranks the alternatives in the choice set in order of preference as represented by a utility function. Each alternative is characterized by a utility and is chosen if and only if its utility exceeds the utility of all other alternatives. Each utility equation includes a random error term, because it is not possible to model every aspect of an alternative or the decision maker in the utility equation. The reader is referred to [3] for a full treatment of discrete choice models.

Let \mathcal{I} be the set of decision-making individuals in a population and \mathcal{C} the choice set.

Definition 1. The *utility*, U_{ij} , of alternative $j \in \mathcal{C}$ to individual $i \in \mathcal{I}$ is a sum of a deterministic component V_{ij} which is a function of the characteristics of individual i and the attributes of alternative j , and a random error term ϵ_{ij} which accounts for aspects not included in the model.

$$U_{ij} = V_{ij} + \epsilon_{ij} \tag{1.1}$$

V_{ij} is represented by an affine function of model parameters β and feature set \mathbf{X}_{ij} . The feature set includes decision maker characteristics and alternative specific attributes. We shall use the terms choice and alternative interchangeably.

Random utility models that assume that the error terms are distributed according to a Gumbel distribution are known as logit models.

Definition 2. A random variable X is *Gumbel distributed* with parameters θ and $\frac{1}{\mu}$ if its density is given by:

$$f(x) = \mu e^{-\mu(x-\theta)} e^{-e^{-\mu(x-\theta)}} \tag{1.2}$$

It can be shown that if $X \sim \text{Gumbel}(\theta, \frac{1}{\mu})$ then $E[X] = \theta + \frac{\gamma}{\mu}$ (where γ is the Euler-Mascheroni constant) and $\text{Var}(X) = \frac{\pi^2}{6\mu^2}$. Furthermore, suppose that $\{X_i\}_{i=0}^n$ are Gumbel distributed random variables with means θ_i and equal scale parameters $\frac{1}{\mu}$. Then, $X_M = \max_i X_i$ is also Gumbel distributed with scale parameter $\frac{1}{\mu}$ and mean given by

$$\theta_M = \mathbb{E}[X_M] = \frac{1}{\mu} \ln \sum_i \exp(\mu\theta_i)$$

This property is known as *stability with respect to maximization*. θ_M is called the *Expected Maximum Utility*, the *inclusive value* or the *inclusive utility*.

According to the random utility maximization theory, an alternative j is chosen by individual i only if its utility to individual i , U_{ij} , exceeds the utilities of all the other available alternatives. Let \mathcal{C}_i denote the choice set available to individual i .

If the error terms are treated as random variables, no certain statements can be made regarding the choice of an individual. Instead, we can only determine the probability that a given alternative will be chosen. Let c_{ij} denote a binary random variable indicating whether person i chooses alternative j . If person i makes their choice so as to maximize their utility, then given individual and alternative features \mathbf{X} , we have that:

$$\mathbb{P}(c_{ij} = 1|\mathbf{X}) = \mathbb{P}(U_{ij} \geq U_{ik} \forall k \in \mathcal{C}_i \setminus \{j\}|\mathbf{X}) \quad (1.3)$$

$$= \mathbb{P}(V_{ij} - V_{ik} \geq \epsilon_{ik} - \epsilon_{ij} \forall k \in \mathcal{C}_i \setminus \{j\}|\mathbf{X}) \quad (1.4)$$

$$= \mathbb{P}(V_{ij} - V_{ik} \geq \max_{k \in \mathcal{C}_i \setminus \{j\}} \epsilon_{ik} - \epsilon_{ij}|\mathbf{X}) \quad (1.5)$$

Depending on the particular assumptions adopted regarding the form of the joint distribution of the random errors, specific mathematical forms of the choice probabilities emerge. In particular, if the error terms are assumed to be Gumbel distributed, and since the maximum of Gumbel distributed random variables is itself a Gumbel distributed random variable, one is able to obtain closed form expressions for the choice probabilities under this assumption.

Further assuming that the error terms are independent and have equal variance (homoscedastic) leads to the celebrated multinomial logit model. This assumption is often problematic and leads to unrealistic choice probabilities [15]. Nested logit provides increased modeling power by allowing the error terms to be correlated while still mostly maintaining tractability.

Under the probit model framework, the error terms are assumed to be normally distributed. This model however is not mathematically tractable.

1.3.2 Correlations, Partitions, Trees and Closed-form Probabilities

In the last section, we saw that multinomial logit models assume independence of the error terms, and that nested logit models allow some correlation between the errors. We would now like to develop two equivalent representations of these correlation structures one that is graph based and another that is partition based. The graph representation is more convenient for optimization. Since the underlying structure is a tree, the entire machinery of graph theory can be used to enforce desired properties such as arborescence. The nested partition representation is convenient when proving certain properties such as counting the total number of possible partitions, or the equation for the correlation between two alternatives.

Recall that a *tree* is an undirected graph in which any two vertices (or nodes) are connected by exactly one path. The following definition provides equivalent (and more convenient) characterizations of a tree [14].

Definition 3. An undirected graph \mathcal{G} is a tree if satisfies *any* of the following conditions:

1. \mathcal{G} is connected and has $n - 1$ edges.
2. \mathcal{G} has no simple cycles and has $n - 1$ edges.

The second condition is the basis of the so-called subtour elimination constraints [4]. These constraints are used in Chapter 2 to guarantee a valid tree structure in our optimization framework.

Next, we introduce the concept of a nested partition:

Definition 4. A *nested partition*, \mathcal{B} , of a set S is a set of nonempty subsets B_m such that:

1. One of the subsets B_m is the set S
2. $\bigcup_m B_m = S$
3. Whenever $B_m \cap B'_m \neq \emptyset$, either $B_m \subseteq B'_m$ or $B'_m \subseteq B_m$

We will refer to the subsets B_m constituting a nesting partition \mathcal{B} as “nests”.

Definition 5. Given a nested partition \mathcal{B} , let $B(j)$ denote the smallest set $B_m \in \mathcal{B}$ such that $j \in B_m$. Formally, $B(j) = \bigcap_{B \in \{\bar{B} \in \mathcal{B} : j \in \bar{B}\}} B$. Similarly, $B(C)$ denotes the smallest subset $B_m \in \mathcal{B}$ such that $C \subseteq B_m$.

A subset $B_m \in \mathcal{B}$ is called *degenerate* if $|B_m| = 1$. A nested partition \mathcal{B} is called degenerate if it contains one or more degenerate subsets.

Example 1. Consider the universal set $S = \{1, 2, 3, 4, 5\}$. The set of sets $\{\{1, 2, 3, 4, 5\}, \{1, 2, 3\}, \{2, 3\}, \{4, 5\}\}$ is a valid nested partition. Whereas $\{\{1, 2, 3, 4, 5\}, \{1, 2, 3\}, \{2, 4, 5\}\}$ is not, because the third condition of Definition 4 is violated: $\{1, 2, 3\} \cap \{2, 4, 5\} \neq \emptyset$, but $\{1, 2, 3\} \not\subseteq \{2, 4, 5\}$, $\{2, 4, 5\} \not\subseteq \{1, 2, 3\}$.

Any nested partition can be represented by a *tree* with $|S|$ leaves (this is straightforward: represent each set in the partition by a nest node). We illustrate this with an example, and show the formal process of building a tree from a nested partition in the coming theorem.

Example 2. The nested partition $\{\{1, 2, 3\}, \{2, 3\}, \{4, 5\}, \{1, 2, 3, 4, 5\}\}$ can be represented by the tree:

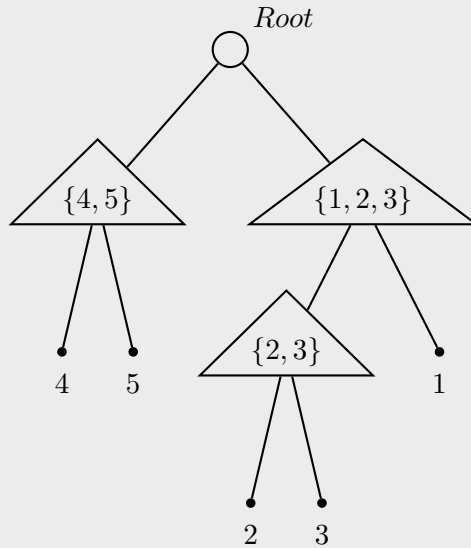


Figure 1-1: A tree representation of the nested partition in Example 2. The root node represents the set $S = \{1, 2, 3, 4, 5\}$. Each of the subsets constituting the partition, apart from the set S , are represented by a nest node.

The following theorem states that if we don't allow degenerate nests, the maximum number of possible nests in a nesting partition is bounded.

Theorem 1. Let S be a finite set with n elements, where $n \geq 2$. Any non-degenerate nested partition of S can be represented by a unique tree with n leaves and at most $n - 2$ nest nodes.

Proof. Let \mathcal{B} denote a nested partition of the set S . We formally show the process of building the graphical representation, \mathcal{G} , of \mathcal{B} :

1. Represent each set in \mathcal{B} and each alternative $j \in S$ by a node.
2. Connect each alternative in the choice set to the smallest set containing it, i.e., for each $j \in S$ find $B(j)$ and connect the node representing $B(j)$ to that representing j .
3. Similarly, connect each set in the nested partition \mathcal{B} to the smallest set in \mathcal{B} containing it, i.e., for each $B_m \in \mathcal{B}$ find $B(B_m)$ and connect the node representing $B(B_m)$ to that representing B_m .

Property 3 of nested partitions guarantees that the minimum is unique (since degeneracy is not allowed). Furthermore by Property 1, there is one node representing the set S with no incoming connections, which we call the root node.

We now show that the number of nest nodes is at most $n - 2$: Let b denote the number of nest nodes. The n leaf nodes are terminal nodes and must have degree one. Since the nesting structure is non-degenerate, all internal nodes must have at least two children. This implies that (i) the root has at least degree two, and (ii) the nest nodes have at least degree three. Therefore

$$\deg(\mathcal{G}) \geq n + 3b + 2$$

Now, by the *Sum of Degrees of Vertices Theorem* we have

$$\deg(\mathcal{G}) = 2|\mathcal{E}|$$

Since \mathcal{G} is a tree, the number of edges is one less than the number of nodes, and we must have $|\mathcal{E}| = (n + b + 1) - 1$, therefore,

$$2(n + b) \geq n + 3b + 2$$

Hence, $b \leq n - 2$ □

This result is significant because it brings tractability to the problem of learning nested logit structures. Namely, in an optimization framework, we can simply include the maximum number of nest nodes and then decide which ones to include.

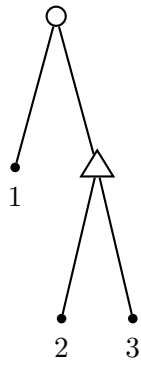
$ S $	1-level	2-level	3-level	...	Total
2	1	0	0	...	1
3	1	3	0	...	4
4	1	13	12	...	26
5	1	50	125	...	236
6	1	201	1040	...	2712

For a set of any size, there is only one 1-level possible partition (corresponding to a rooted tree with no nests). The number of possible 2-level non-degenerate partitions is related to the Bell numbers in combinatorial mathematics. For a set of size n , B_n counts the number of possible partitions of a set. For a set of size n then, the number of 2-level nesting partitions is given by $B_n - 2$ (discounting a degenerate partition, and a level-1 partition). To compute the number of 3-level non-degenerate nesting partitions, one can use recursion. Let $Y_{n,m}$ denote the number of m -level non-degenerate nesting partition for a set of size n . The number of 3-level non-degenerate nesting partitions for a set of size 4, $Y_{4,3}$ is given by the following recursion:

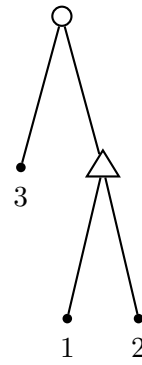
$$Y_{4,3} = \binom{4}{3} Y_{3,2}$$

Similar logic can be applied to obtain the other figures in the table.

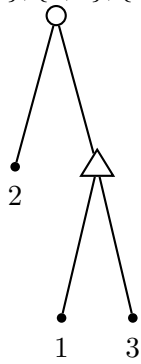
As an illustration, Figure 1-2 shows the four possible nesting structures for the set $\{1, 2, 3\}$.



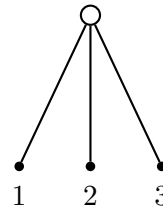
(a) $\{\{1\}, \{2, 3\}, \{1, 2, 3\}\}$



(c) $\{\{3\}, \{1, 2\}, \{1, 2, 3\}\}$



(b) $\{\{2\}, \{1, 3\}, \{1, 2, 3\}\}$



(d) $\{\{1, 2, 3\}\}$

Figure 1-2: The four possible non-degenerate nesting structures for the set $\{1, 2, 3\}$.

Theorem 1 provided the link between the set representation and the graph representation of a nested partition. We now link these two representations to the correlation structure of the error terms for the multinomial and nested logit models. The key idea is that alternatives within the same nest are allowed to be correlated.

Multinomial logit

We start our discussion with multinomial logit models. We have seen that a key assumption here is the independence of the error terms. Figure 1-3 shows three equivalent representations of the *i.i.d* assumption in the logit framework.

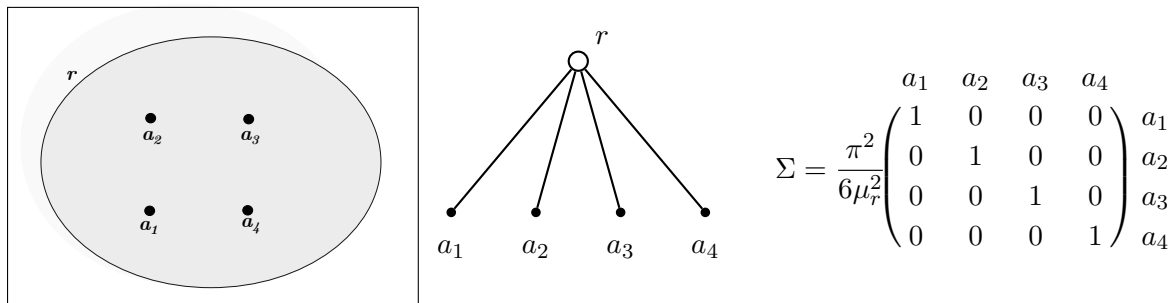


Figure 1-3: A nested partition over the set $\mathcal{S} = \{a_1, \dots, a_4\}$ and the equivalent nesting tree and variance-covariance matrix for a multinomial logit model. Notice that in this case the nested partition set \mathcal{B} is simply $\{\mathcal{S}\}$.

It can be shown that if the error terms, ϵ_{ij} , are independent and identically distributed according to a Gumbel distribution with parameters $\mu = 0$ and $\theta = 1$, then the probability of individual i choosing alternative j is given by

$$P_{ij} = \frac{e^{V_{ij}}}{\sum_{k \in \mathcal{C}} e^{V_{ik}}} \quad (1.6)$$

In general, the choice set in a multinomial logit framework can be represented by a rooted tree with no nests.

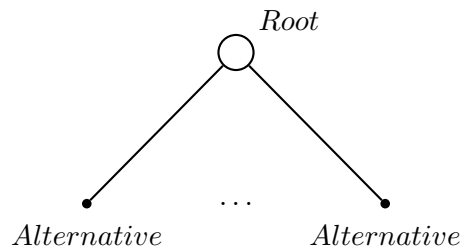


Figure 1-4: Multinomial Logit Model: The choice set can be represented by a rooted tree with height 1.

Example 3 Consider the three alternative choice set $\{1, 2, 3\}$ with utility equations

$$U_1 = V_1 + \epsilon_1$$

$$U_2 = V_2 + \epsilon_2$$

$$U_3 = V_3 + \epsilon_3$$

Where $\epsilon_1, \epsilon_2, \epsilon_3$ are i.i.d $Gumbel(0, 1)$. That is $Var(\epsilon_i) = \frac{\pi^2}{6}$ and $Cov(\epsilon_i, \epsilon_j) = 0 \ i \neq j$. Since the error terms are independent there is no correlation between the alternatives and the corresponding nesting partition is $\{\{1, 2, 3\}\}$.



Figure 1-5: Tree representation of the nested partition for example 3

Multinomial logit, while simple, is unrealistic. To see why, consider two alternatives $j, j' \in \mathcal{C}$. The ratio of the choice probabilities of these two alternatives is independent of the presence or absence of any other alternative $k \in \mathcal{C} \setminus \{j, j'\}$ or their attributes.

$$\frac{P_j}{P_{j'}} = \frac{e^{V_j}}{e^{V_{j'}}$$

This property is known as the *independence of irrelevant alternatives* (IIA) [15]. Now, suppose that a new alternative \tilde{j} , similar to j in every aspect, is added to the choice set \mathcal{C} , then the share of choice \tilde{j} will draw equally from the shares of all alternatives present in the choice set. This is counter-intuitive as one would expect the new alternative \tilde{j} to draw mainly from the share of alternative j . The issue is that alternatives j and \tilde{j} are similar and therefore the error terms in their utility equations are *correlated*. Whereas in applying the multinomial logit framework, the assumption that the error terms are independent is made. Nested logit allows for such correlation which we see next.

Nested logit

Nested logit models generalize multinomial logit models by allowing for correlation between alternatives. The main idea is that similar alternatives share common components in their random error terms. Alternatives are partitioned into *nests*. Intra-nest correlation is quantified by the *scale parameter* of the shared nest error term. Inter-nest correlation is zero, i.e., alternatives in different nests are uncorrelated. We formally state the nested logit modeling assumptions using our partition framework.

Nested Logit Modeling Assumptions

- A1.** The error terms are Gumbel distributed with zero mean.
- A2.** The error terms across different individuals are independent.
- A3.** The total variance of the error terms are equal.
- A4.** The choice set \mathcal{C} is partitioned into a non-degenerate nested partition $\mathcal{B} = \{B_m\}_m$.
- A5.** Each nest $b \in \mathcal{B}$ is associated with a scale parameter $\frac{1}{\mu_b}$.
- A6.** The total variance of an alternative j is a sum of an alternative specific variance and nest specific variances: $\epsilon_j + \sum_{B_m \in \mathcal{B} | j \in B_m} \epsilon_{B_m}$.
- A7.** The alternative specific error term for an alternative j is Gumbel distributed with zero mean and parameter $\frac{1}{\mu_{B(j)}}$ where $B(j)$ is the smallest nest containing alternative j .
- A8.** The nest specific error term for nest B_m is Gumbel distributed with zero mean and parameter $(\frac{1}{\mu_{B(B_m)}^2} - \frac{1}{\mu_{B_m}^2})^{\frac{1}{2}}$ where $B(B_m)$ is the smallest nest containing nest B_m .
- A9.** The alternative specific error terms and nest specific error terms are all independent from each other.
- A10.** The scale parameters satisfy:

$$B_m \subseteq B_{m'} \implies \mu_{m'} \leq \mu_m$$

We make the following remarks on these assumptions:

- The zero mean part of A1 comes at no loss of generality if the systematic part of the utilities, V_{ij} , include an intercept in their specification.
- Regarding A4, strictly speaking the nested partitions need not be non-degenerate, i.e., the nested logit framework still holds. However, degenerate partitions do not change the choice probabilities and are excluded to bring tractability to the problem (cf. Theorem 1).

- A8 is usually stated differently in the literature. The statement is usually that the nest specific error term of a nest B is distributed so that the utility of the maximum over all the alternatives in that nest is Gumbel distributed with scale parameter $\frac{1}{\mu_B}$. We believe our presentation is clearer in the nested partition framework we introduced.
- Note that unless A10 is satisfied, the formula for the variance in A8 is not defined. While this reasoning is purely technical, there is a behavioural interpretation on why the scale should increase with increasing nesting level. Namely the variance of the errors should decrease as the choice set is narrowed going down the choice tree. In fact, [3] show that the choices are consistent with rational utility theory only if this assumption is satisfied.

We are now ready to make a concrete statement regarding the relationship between the nested partition and the correlation structure of the error terms. How does the nested partition shape the correlation structure? The following proposition answers this question.

Proposition 1. Let \mathcal{B} be a nested partition of the choice set. If the nested logit assumptions are satisfied, then the covariance $cov(i, j)$ between any two alternatives $i, j \in \mathcal{S}$ is given by

$$\frac{\pi^2}{6} \left(\frac{1}{\mu_{\mathcal{S}}^2} - \frac{1}{\mu_{B(i,j)}^2} \right)$$

Proof. Let $\tilde{\epsilon}_i$ and $\tilde{\epsilon}_j$ denote the total error component of alternatives i and j respectively, i.e., the error terms that appear in the utility equations for these alternatives. We have:

$$cov(\tilde{\epsilon}_i, \tilde{\epsilon}_j) = cov\left(\epsilon_i + \sum_{B_m \in \mathcal{B} | i \in B_m} \epsilon_{B_m}, \epsilon_j + \sum_{B_m \in \mathcal{B} | j \in B_m} \epsilon_{B_m}\right) \quad (1.7)$$

$$= \sum_{B_m \in \mathcal{B} | i, j \in B_m} var(\epsilon_{B_m}) \quad (1.8)$$

$$= \frac{\pi^2}{6} \sum_{B_m \in \mathcal{B} | i, j \in B_m} \left(\frac{1}{\mu_{B(B_m)}^2} - \frac{1}{\mu_{B_m}^2} \right) \quad (1.9)$$

(1.5) follows directly from A6. (1.6) follows from the independence assumption from A9. (1.7) follows from A8 and the definition of a Gumbel distributed random variable. Now, suppose that the cardinality of the set $\{B_m \in \mathcal{B} | i, j \in B_m\}$ is k . (Note that since $i, j \in \mathcal{S} \in \mathcal{B}$, we always have $k \geq 1$). Define sets $\mathcal{K}^{(i)} = B^i(\{i, j\})$ for $i = 1, \dots, n$ such that:

$$|\mathcal{S}| = |\mathcal{K}^{(k)}| \geq |\mathcal{K}^{(k-1)}| \geq \dots \geq |\mathcal{K}^{(1)}| = |B(i, j)|$$

Where $B^s(\{i, j\})$ is the s fold application of the function $B(\cdot)$ defined earlier (i.e. $\mathcal{K}^{(s)}$ is the s smallest set containing i and j). It is clear that the largest set containing i and j is the set \mathcal{S} , and

the smallest set containing i and j is $B(i, j)$ by definition. Continuing from (1.9),

$$\text{cov}(\tilde{\epsilon}_i, \tilde{\epsilon}_j) = \frac{\pi^2}{6} \sum_{s=1}^{k-1} \left(\frac{1}{\mu_{\mathcal{K}^{(s+1)}}^2} - \frac{1}{\mu_{\mathcal{K}^{(s)}}^2} \right) \quad (1.10)$$

$$= \frac{\pi^2}{6} \left(\frac{1}{\mu_{\mathcal{K}^{(k)}}^2} - \frac{1}{\mu_{\mathcal{K}^{(1)}}^2} \right) \quad (1.11)$$

$$= \frac{\pi^2}{6} \left(\frac{1}{\mu_{\mathcal{S}}^2} - \frac{1}{\mu_{B(i,j)}^2} \right) \quad (1.12)$$

□

Note that by proposition 1, if $B(i, j) = \mathcal{S}$, the covariance between alternatives i and j is zero. Therefore if $B(i, j) = \mathcal{S} \forall i, j$ (i.e., the smallest partition containing any given two alternatives i and j is \mathcal{S}) we are back to the multinomial logit case!).

This presentation makes clear a limitation of nested logit models. Namely, **A10 and Proposition 1 imply that the correlations are always non-negative**. In other words, nested logit models can not handle negative correlations, there is an implicit assumption that any two alternatives are either not correlated or positively correlated.

Proposition 1, enables us to formally link the nested partition representation of the problem to the variance covariance structure. Crucially, we have seen that the scale parameters quantify the correlation between any two alternatives. Figure 1-6 shows an example over a choice set $\mathcal{S} = \{a_1, a_2, a_3, a_4\}$ where the nested partition is given by $\{\{a_1, a_2, a_3, a_4\}, \{a_1\}, \{a_2, a_3, a_4\}, \{a_3, a_4\}\}$. In the graphical representation the set \mathcal{S} is represented by the letter r (root node).

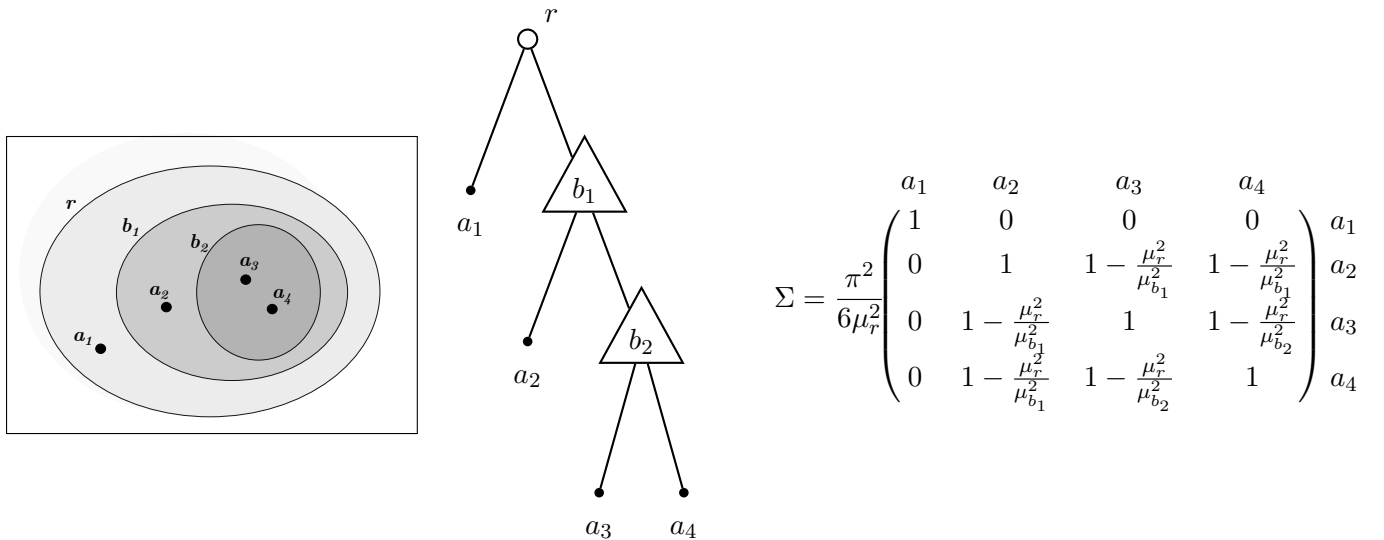


Figure 1-6: A nested partition over the set $\mathcal{S} = \{a_1, \dots, a_4\}$ and the equivalent nesting tree and variance-covariance matrix for a nested logit model

A few remarks on identification issues are in order [8]:

1. Since the overall scale of utility is not defined, only $|\mathcal{S}| - 1$ scale parameters may be identified. We choose to normalize the scale of the root nest, μ_r , to 1.
2. The mean of the Gumbel distribution of the error terms is not identified if V_i contains an intercept. We have therefore assumed, without loss of generality, that the mean of the error terms is zero.
3. If any column of the features matrix \mathbf{X} does not vary over the alternatives, the parameters for one alternative have to be normalized to zero for purposes of identification. This is so because only differences in utilities are relevant for the choice.

The nested logit assumptions (A1-A10) extend the multinomial logit model by allowing alternatives in the same nest to be positively correlated. Furthermore these assumptions lead to tractable forms of the choice probabilities which we discuss next. We first define the concept of an inclusive value which features in the closed-form expressions for the probabilities.

Definition 6. The *Inclusive Value*, Γ_{B_m} , of a nest B_m is its expected maximum utility.

$$\Gamma_{B_m} = \frac{1}{\mu_{B_m}} \ln \left(\sum_{j \in \mathcal{S} | B(j) = B_m} e^{\mu_{B_m} V_j} + \sum_{B_{m'} \in \mathcal{B} | B_{m'} \subseteq B_m} e^{\mu_{B_m} \Gamma_{B_{m'}}} \right) \quad (1.13)$$

This definition was first introduced in [2]. Suppose \mathcal{B} is a nested partition over the set of alternatives \mathcal{S} . By conditioning on the nests containing an alternative $j \in \mathcal{S}$, it can be shown, using the product rule, that the probability of choosing this alternative is given by

$$P_j = P_{j|B(j)} P_{B(j)|B^2(j)} \dots P_{B^{k-1}(j)|B^k(j)} P_{B^k(j)} \quad (1.14)$$

Where $B^s(\cdot)$ is the s -fold application of the function $B(\cdot)$, and k is the number of partitions in \mathcal{B} that contain the alternative j . Note that $B^k(j) = \mathcal{S}$, i.e., the largest set containing the alternative j is the set \mathcal{S} .

Where $P_{j|B(j)}$, the conditional probability of choosing j given that some alternative in $B(j)$ is chosen, is given by the relative attractiveness of the utility of alternative j compared to the maximum utility obtainable from choosing some alternative in $B(j)$, i.e., the inclusive value of $B(j)$, $\Gamma_{B(j)}$:

$$P_{j|B(j)} = \exp(\mu_{B(j)}(V_j - \Gamma_{B(j)})) \quad (1.15)$$

Similarly, the probability of choosing nest $B^s(j)$ conditional on choosing $B^{s+1}(j)$ the smallest nest containing it is given analogously to (1.15)

$$P_{B^s(j)|B^{s+1}(j)} = \exp(\mu_{B^{s+1}(j)}(\Gamma_{B^s(j)} - \Gamma_{B^{s+1}(j)})) \quad (1.16)$$

Finally, since *some* alternative in \mathcal{S} has to be chosen, we have

$$P_{B^k(j)} = 1 \tag{1.17}$$

Using (1.15), (1.16), and (1.17) we can rewrite (1.14) as

$$P_j = \exp \left(\mu_{B(j)} V_j + \sum_{i=1}^{k-1} (\mu_{B^{i+1}(j)} - \mu_{B^i(j)}) \Gamma_{B^i(j)} - \mu_{B^k(j)} \Gamma_{B^k(j)} \right) \tag{1.18}$$

The closed-form (1.18), while mathematically tractable, is a rather complicated function involving nested logs of sums of exponential terms. We discuss the implications of this in Chapter 3.

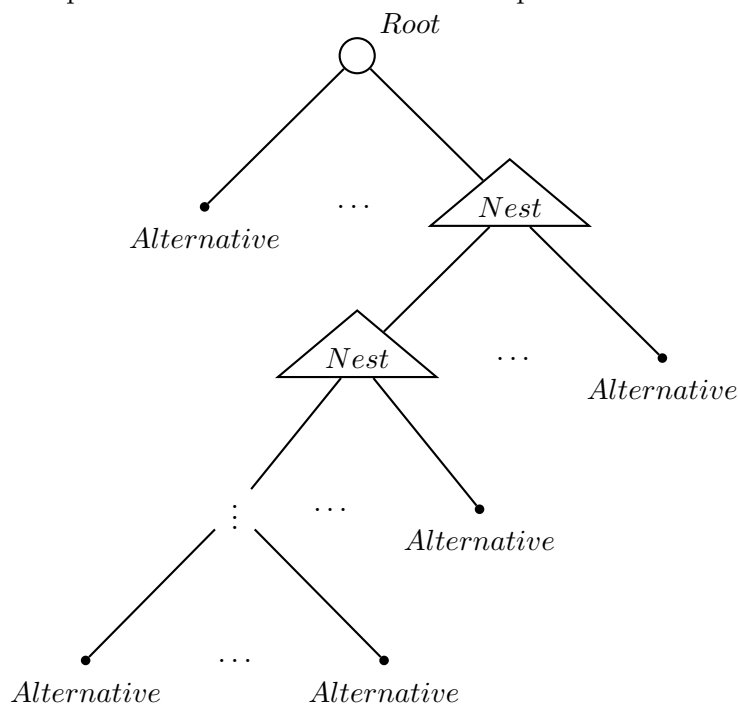


Figure 1-7: Nested Logit Model: For a choice set of size m , a nested logit model can be represented by a tree of height at most $m - 1$ (assuming non-degenerate nests)

Example 4 Consider again the choice set $\{1, 2, 3\}$. With modified utility equations:

$$U_1 = V_1 + \epsilon_a + \epsilon_1$$

$$U_2 = V_2 + \epsilon_a + \epsilon_2$$

$$U_3 = V_3 + \epsilon_3$$

Where $\epsilon_a, \epsilon_1, \epsilon_2, \epsilon_3$ are independent (A2). The *total error* of each of the three alternatives is assumed to be $Gumbel(0, \frac{1}{\mu_r})$ (A3). The variance of these distributions is given by

$$Var(\epsilon_a + \epsilon_1) = Var(\epsilon_a + \epsilon_2) = Var(\epsilon_3) = \frac{\pi^2}{6\mu_r^2}$$

The error terms ϵ_1 and ϵ_2 are assumed to be distributed according to $Gumbel(0, \frac{1}{\mu_a})$ (A7).

$$Var(\epsilon_1) = Var(\epsilon_2) = \frac{\pi^2}{6\mu_a^2}$$

The common error component ϵ_a creates a covariance between the total error terms of alternatives 1 and 2. Note that $Cov(U_1, U_2) = Var(\epsilon_a) + Cov(\epsilon_1, \epsilon_a) + Cov(\epsilon_2, \epsilon_a) + Cov(\epsilon_1, \epsilon_2) = Var(\epsilon_a)$.

The choice structure implied by these equations is depicted by the nesting structure in Figure 6, in which alternatives 1 and 2 are more similar to each other than they are to alternative 3.

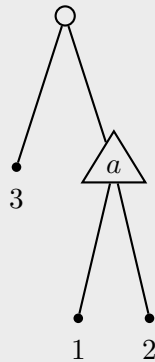


Figure 1-8: Nesting structure for Example 4

Finally since $Var(\epsilon_a + \epsilon_1) \geq Var(\epsilon_1)$, $\frac{\pi^2}{6\mu_r^2} \geq \frac{\pi^2}{6\mu_a^2}$ and we must have $\mu_a \geq \mu_r$ (A10).

1.4 Summary

In this chapter we motivated our desire to take an algorithmic approach to the design of nested logit models. We saw that nested logit models provided some flexibility in the specification of the covariance matrix. At the the same time, tractability in the form of closed form choice probabilities was maintained by making certain assumptions that we clearly listed (A1-A10). The modeling implications of these assumptions are:

1. The total variance of the error terms are equal (homoscedasticity) to some normalized value.
2. The correlation between the error terms is non-negative.
3. In the tree representation, if two alternatives share a nest ancestor, their correlation is a function of the scale of that common ancestor (cf. Proposition 1). Alternatives whose only common ancestor is the root node are not correlated.

4. The error terms are independent across individuals (the data used for estimation can not contain multiple observations for the same individual).

Recall that our core motivation is to algorithmically determine which of the allowable co-variance structures best explains the data. Working with the co-variance matrix directly is not very attractive in the nested logit framework. We therefore introduced notations and equivalences between three different representations of nested logit models namely:

1. A nested partition based representation that was used to formulate the choice probabilities and prove certain properties,
2. A tree based representation that was used to prove Theorem 1, and will be incredibly useful in the optimization representation of the problem,
3. A co-variance based representation which helped us understand the flexibility and limitations of nested logit models.

Chapter 2

Nested Logit Structure Learning Problem

In God we trust, all others must bring data

W. Edwards Deming

2.1 Problem Statement

The general framework we take for finding an optimal tree \mathbb{T} (representing a nesting partition) and its parameters $\theta_{\mathbb{T}}$ (the scale and taste parameters) is a mixed-integer non-linear program. At a high level, the problem is:

$$\begin{aligned} & \max_{\langle \mathbb{T}, \theta_{\mathbb{T}} \rangle} \text{Likelihood} - \text{Complexity Penalty} \\ \mathbf{subject\ to} & \quad (1) \mathbb{T} \text{ is a valid nesting tree} \\ & \quad (2) \text{Scale parameters are consistent with utility maximization (cf. A10)} \end{aligned}$$

In this chapter, we obtain a closed form expression for the likelihood, formulate conditions (1) and (2) using linear constraints, and discuss ways of penalizing model complexity in a regularization framework.

2.2 A Mixed-Integer Nonlinear Program Formulation

Let \mathcal{B} be an abstract nested partition over the set of alternatives \mathcal{S} and consider the graph representation \mathbb{T} of \mathcal{B} . Recall that \mathbb{T} is a directed tree (arborescence) and that in building \mathbb{T} from \mathcal{B} we have:

1. A root node r representing the choice set \mathcal{S} .
2. Internal nest nodes \mathcal{N} each representing a nest in $\mathcal{B} \setminus \mathcal{S}$.
3. Leaf nodes representing each alternative in \mathcal{S} .

There is a directed edge between two nodes u and v in this graph only if node u is the nest node representing the smallest nesting partition containing the nest or alternative represented by v (i.e., $x_{u,v} = 1$ only if $B(u) = v$).

Formally, $\mathbb{T} = (\mathcal{V}, \mathcal{E})$ is a directed graph where $\mathcal{V} = \{r\} \cup \mathcal{N} \cup \mathcal{S}$ is the set of vertices and \mathcal{E} the set of edges.

In general we do not know a priori the structure of the tree, or if nesting is present. The goal is to use optimization to reveal this structure. Theorem 1, provides the following guarantee

$$|\mathcal{N}| \leq |\mathcal{S}| - 2 \tag{2.1}$$

For convenience let $p = |\mathcal{S}| - 2$. The implication of this result is as follows: since *any nested partition can be represented by at most p nest nodes, we start with the nest node set \mathcal{N} containing that maximum number of nest nodes (namely p), and we let an optimization procedure guide the inclusion or exclusion of these nests*. To this end, we define for every nest node $v \in \mathcal{N}$, a variable y_v equal to one if nest v is chosen. Similarly let $x_{u,v} = 1$ if there is a directed edge between nodes u and v .

Over the next few sections we build an optimization problem for finding the optimal tree structure \mathbb{T} (which determines the co-variance structure of the error terms), the utility parameters β (which determine the relative attractiveness of each alternative) and the scale parameters μ which quantify the total variance and correlation between alternatives.

2.2.1 Objective Function

In this section we are concerned with finding a closed form of the log-likelihood function. The probability of choosing alternative $a \in \mathcal{S}$ can be found by conditioning on the path from the root

r to the leaf node a :

$$\mathbb{P}(a) = x_{ra}\mathbb{P}(a|B(a) = B_r)\mathbb{P}(B_r) + \sum_{b \in \mathcal{N}} x_{rb}x_{ba}\mathbb{P}(a|B(a) = B_b)\mathbb{P}(B_b|B_r)\mathbb{P}(B_r) + \dots \quad (2.2)$$

$$+ \sum_{b_1, b_2, \dots, b_p \in \mathcal{N}} (x_{rb_1} \prod_{i=2}^{p-1} x_{b_i b_{i+1}} x_{b_p a}) \mathbb{P}(a|B(a) = B_{b_1}) \mathbb{P}(B_{b_1}|B_{b_2}) \dots \mathbb{P}(B_{b_p}|B_{b_r}) \mathbb{P}(B_r) \quad (2.3)$$

Since \mathbb{T} is a tree (we will look at enforcing this property in the next section), there is a unique path from r to any leaf node a , therefore exactly one of the terms in the summation above will be nonzero. We can exploit this fact to rewrite the probability as a product of terms which will be very convenient since we will be taking logarithms of this quantity:

$$\mathbb{P}(a) = \left(\mathbb{P}(a|B(a) = B_r)\mathbb{P}(B_r) \right)^{x_{ra}} \cdot \prod_{b \in \mathcal{N}} \left(\mathbb{P}(a|B(a) = B_b)\mathbb{P}(B_b|B_r)\mathbb{P}(B_r) \right)^{x_{rb}x_{ba}} \cdot \dots \quad (2.4)$$

$$\prod_{b_1, b_2, \dots, b_p \in \mathcal{N}} \left(\mathbb{P}(a|B(a) = B_{b_1})\mathbb{P}(B_{b_1}|B_{b_2}) \dots \mathbb{P}(B_{b_p}|B_{b_r})\mathbb{P}(B_r) \right)^{x_{rb_1} \prod_{i=2}^{p-1} x_{b_i b_{i+1}} x_{b_p a}} \quad (2.5)$$

Written explicitly in terms of utilities and inclusive values:

$$\mathbb{P}(a) = \left(e^{\mu_r V_a - \mu_r \Gamma_r} \right)^{x_{ra}} \cdot \prod_{b \in \mathcal{N}} \left(e^{\mu_b V_a + (\mu_r - \mu_b) \Gamma_b - \mu_r \Gamma_r} \right)^{x_{rb}x_{ba}} \cdot \dots \quad (2.6)$$

$$\prod_{b_1, b_2, \dots, b_p \in \mathcal{N}} \left(e^{\mu_{b_p} V_a + \sum_{i=1}^{p-1} (\mu_{b_i} - \mu_{b_{i+1}}) \Gamma_{b_{i+1}} + (\mu_r - \mu_{b_1}) \Gamma_{b_1} - \mu_r \Gamma_r} \right)^{x_{rb_1} \prod_{i=2}^{p-1} x_{b_i b_{i+1}} x_{b_p a}} \quad (2.7)$$

Where

$$\Gamma_b = \frac{1}{\mu_b} \ln \left(\sum_{a \in \mathcal{S}} x_{bs} e^{\mu_b V_a} + x_{bb'} \sum_{b' \in \mathcal{N}} e^{\mu_b \Gamma_{b'}} \right), \quad (2.8)$$

and

$$V_a = \mathbf{X}_a \boldsymbol{\beta}. \quad (2.9)$$

Let $c_{na} \in \{0, 1\}$ denote if individual n chooses alternative a . The log likelihood takes a convenient form of a decomposable sum:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \boldsymbol{\beta} | \mathbf{X}_n) = \ln \prod_{n \in \mathcal{I}} \prod_{a \in \mathcal{S}} \mathbb{P}(c_{na})^{c_{na}} \quad (2.10)$$

$$= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} c_{na} \ln \mathbb{P}(c_{na}) \quad (2.11)$$

$$= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} c_{na} \left[x_{ra} (\mu_r V_{na} - \mu_r \Gamma_{nr}) + \sum_{b \in \mathcal{N}} x_{rb} x_{ba} (\mu_b V_{na} + (\mu_r - \mu_b) \Gamma_{nb} - \mu_r \Gamma_{nr}) + \dots \right.$$

$$\left. + \sum_{b_1, b_2, \dots, b_p \in \mathcal{N}} (x_{rb_1} \prod_{i=2}^{p-1} x_{b_i b_{i+1}} x_{b_p a}) (\mu_{b_p} V_{na} + \sum_{i=1}^{p-1} (\mu_{b_i} - \mu_{b_{i+1}}) \Gamma_{nb_{i+1}} + (\mu_r - \mu_{b_1}) \Gamma_{nb_1} - \mu_r \Gamma_{nr}) \right] \quad (2.12)$$

We also note that the likelihood function is non-concave. If the binary variables \mathbf{x} and the scale parameters $\boldsymbol{\mu}$ are fixed, the function is concave in $\boldsymbol{\beta}$.

Furthermore the total number of terms in the summations above is $|\mathcal{I}| \cdot |\mathcal{S}| \cdot \lfloor |\mathcal{S}|!e \rfloor$.

To see why this is so, recall that $p = |\mathcal{N}| = |\mathcal{S} - 2|$ and notice that the number of terms in the square brackets is equal to the number of permutations of the index b . which is $p(1 + p + p(p - 1) + \dots + p!) = p(1 + p! \sum_{k=1}^{p-1} \frac{1}{k!})$. Now $p! \sum_{k=1}^{p-1} \frac{1}{k!} < p! \sum_{k=1}^{\infty} \frac{1}{k!} = p!e$. The difference is given by $p! \sum_{k=n}^{\infty} \frac{1}{k!} = 1 + p! \sum_{k=n+1}^{\infty} \frac{1}{k!} < 2$.

The likelihood function has an exponential number of terms and even for simple cases can take exponential time to evaluate “top-down”. However at valid tree solutions, there is a unique path from the root to each alternative. In such cases to compute the likelihood, for each individual, one starts at the chosen alternative, c_{na} , and follows the alternative’s ancestry adding to the utility of the alternative the inclusive values along the unique path to the root and scaling appropriately. We discuss matters regarding the efficient evaluation of the likelihood in Section 3.2.2.

Generalization to individual specific choice sets

The formulation above assumes that every alternative in the choice set \mathcal{S} is available to each individual in the population. We would like the flexibility to specify the availabilities of each choice to each individual. To this end, let $a_{ij} = 1$ if choice $j \in \mathcal{S}$ is available to individual $i \in \mathcal{I}$.

We modify the definition of the probabilities,

$$P_{ij} = a_{ij} \exp \left(\mu_{B(j)} V_{ij} + \sum_{i=1}^{k-1} (\mu_{B^{i+1}(j)} - \mu_{B^i(j)}) \Gamma_{iB^i(j)} - \mu_{B^k(j)} \Gamma_{iB^k(j)} \right) \quad (2.13)$$

and the inclusive values:

$$\Gamma_{B_{im}} = \frac{1}{\mu_{B_m}} \ln \left(\sum_{j \in \mathcal{S} | B(j)=B_m} a_{ij} e^{\mu_{B_m} V_{ij}} + \sum_{B_{m'} \in \mathcal{B} | B_{m'} \subseteq B_m} e^{\mu_{B_m} \Gamma_{B_{im'}}} \right) \quad (2.14)$$

2.2.2 Constraints

There are two main types of constraints: constraints that guarantee a valid nesting tree, and the rational utility theory constraints. There are also additional structural constraints. We discuss these individually next. Crucially, we show that all the desired properties can be enforced using linear constraints.

Arborescence

Recall that a graph \mathcal{G} with n nodes is a tree (arborescence) if it has no simple cycles and has $n - 1$ edges (see Definition 3). The following constraint guarantees that the total number of edges is one

less than the total number of nodes.

$$\sum_{e \in \mathcal{E}} x_e = \left(\sum_{u \in \mathcal{N}} y_u + |\mathcal{S}| + 1 \right) - 1 \quad (2.15)$$

The set of constraints below takes any potential sub-tour (cycle) and declares it illegal

$$\sum_{\{(u,v) \in \mathcal{E}: u \in A, v \in A\}} x_e \leq |A| - 1, \quad \forall A \subset \mathcal{V} \quad (2.16)$$

Note that the number of subtour elimination constraints is exponential in the number of nodes of the tree. We discuss ways of dealing with this in Chapter 3.

Scale Constraints

In order for the estimated model to be consistent with utility maximization we require the scale parameters μ to increase with nesting level (see A10). The implication is that if $x_{uv} = 1 \implies \mu_u \leq \mu_v$. This can be enforced through the following constraints:

$$\mu_u - \bar{\mu}(1 - x_{uv}) \leq \mu_v \quad \forall \text{ distinct } u, v \in \mathcal{N} \quad (2.17)$$

where $\bar{\mu}$ is an upper bound on the scale parameters.

Structural Constraints

We first define the following sets:

- The set of edges that originate in node u : $\delta_u^{out} = \{(u, v) \in \mathcal{E}\}$
- The set of edges that terminate in node u : $\delta_u^{in} = \{(v, u) \in \mathcal{E}\}$

A choice node can belong to one and only one lower level nest. Therefore the sum of edges incident to choice nodes should sum to unity

$$\sum_{e \in \delta_a^{in}} x_e = 1 \quad \forall a \in \mathcal{S} \quad (2.18)$$

If a nest node is included, it must have exactly one parent

$$\sum_{e \in \delta_v^{in}} x_e = y_v \quad \forall v \in \mathcal{N} \quad (2.19)$$

If a nest node is included it must have a directed edge to at least two nodes (so that it is not degenerate), and if it is not included it cannot make connections.

$$2y_u \leq \sum_{e \in \delta_u^{out}} x_e \leq |\mathcal{S} - 1|y_u \quad \forall u \in \mathcal{N} \quad (2.20)$$

Similarly, the root node can not be degenerate and must connect to at least two nodes in the tree

$$2 \leq \sum_{e \in \delta_r^{out}} x_e \quad (2.21)$$

Furthermore if there are nest nodes, at least one nest node should be connected to the root

$$1 - (1 - \delta) \leq \sum_{u \in \mathcal{N}} x_{ru} \quad (2.22)$$

$$\sum_{u \in \mathcal{N}} y_u \leq |\mathcal{S} - 2|\delta \quad (2.23)$$

Where $\delta \in \{0, 1\}$ is equal to one if there is at least one nest node in the tree.

Choice nodes must be leaf (terminal) nodes:

$$\sum_{e \in \delta_a^{out}} x_e = 0 \quad \forall a \in \mathcal{S} \quad (2.24)$$

The root node r can not have incident edges:

$$\sum_{e \in \delta_r^{out}} x_e = 0 \quad (2.25)$$

Finally, we disallow self-arcs:

$$x_{uu} = 0 \quad \forall u \in \mathcal{V} \quad (2.26)$$

We end this section with an illustrative example.

Example 5 Suppose we have a dataset of sample size $n = 500$. The universal choice set in this test is $\{1, 2, 3\}$. The utilities of each of the alternatives are given by:

$$\begin{aligned} V_{i1} &= \alpha_1 + \beta f_{i1} \\ V_{i2} &= \alpha_2 + \beta f_{i2} \\ V_{i3} &= \beta f_{i3} \\ U_{i1} &= V_{i1} + \epsilon_{ia} + \epsilon_{i1} \\ U_{i2} &= V_{i2} + \epsilon_{ia} + \epsilon_{i2} \\ U_{i3} &= V_{i3} + \epsilon_{i3} \end{aligned}$$

Where f_{ij} , $j = 1, 2, 3$ are alternative specific features. Let $\mathbf{X}_{n \times 3} = [f_{ij}]$ denote the feature matrix. $\epsilon_{ia}, \epsilon_{i1}, \epsilon_{i2}, \epsilon_{i3}$ are individual specific error terms that are independent, $\epsilon_{ia} + \epsilon_{i1}, \epsilon_{ia} + \epsilon_{i2}, \epsilon_{i3} \sim$

$Gumbel(0, 1)$, and ϵ_{i1} and $\epsilon_{i2} \sim Gumbel(0, \frac{1}{\mu_{true}})$, and $\alpha_1 = 0.5$, $\alpha_2 = 1$, $\beta = -0.05$, $\mu_{true} = 1.5$. The resulting true nested logit structure implied by the error terms is shown in the figure overleaf.

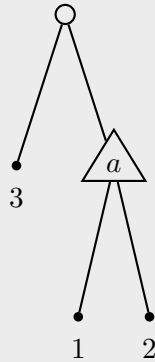


Figure 2-1: Nesting structure underlying the synthetic data

Formulation

Suppose that we did not know this structure a priori and wish to learn it from the data. Note that with three alternatives, Theorem 1 says that we can have at most $3-2=1$ nest nodes. Now consider the graph $\mathcal{G} = (\{r\} \cup \mathcal{B} \cup \mathcal{C}, \mathcal{E})$. Where $\mathcal{B} = \{a\}$ is the set of abstract nest nodes, r is the root node, and $\mathcal{C} = \{1, 2, 3\}$ is the set of alternatives, $\mathcal{E} = \mathbf{x} = \{x_{r1}, x_{r2}, x_{r3}, x_{ra}, x_{a1}, x_{a2}, x_{a3}\}$ is the edge set. Let y_a denote whether nest node a is included.

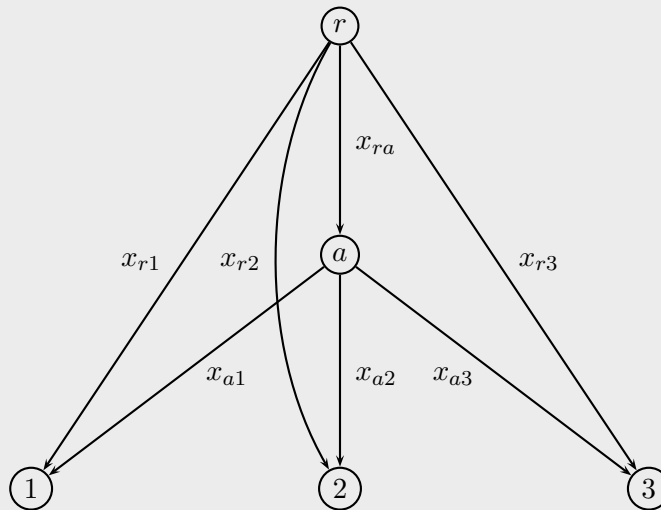


Figure 2-2: Graph showing all possible edge connections. The edge labels are binary variables indicating their inclusion or exclusion in the estimated model

The task is to choose a subset of the edges, \mathcal{E} , and the abstract nest nodes, such that the estimated parameters for that particular choice maximize the likelihood, and there is unique path connecting

the root to each of the alternatives (i.e., the resulting graph is rooted tree).

The likelihood is given by

$$\mathcal{L}(\alpha_1, \alpha_2, \beta, \mathbf{x}) = \sum_{i=1}^{500} \sum_{j=1}^3 \mathbb{1}_{\{c_i=j\}} \ln \mathbb{P}(c_i = j | \mathbf{X}, \mathbf{A})$$

where

$$\ln \mathbb{P}(c_i = j | \mathbf{X}, \mathbf{A}) = a_{ij} [x_{rj}(V_j - \Gamma_r) + x_{ra}x_{aj}(\mu V_{ij} + (1 - \mu)\Gamma_a - \Gamma_r)],$$

$$\Gamma_a = \frac{1}{\mu} \ln \left(\sum_{j=1}^3 a_{ij} x_{aj} e^{\mu V_{ij}} \right), \text{ and}$$

$$\Gamma_r = \ln(x_{ra} e^{\Gamma_a} + \sum_{j=1}^3 a_{ij} x_{rj} e^{V_{ij}})$$

Recall that $\mathbf{A} = [a_{ij}]$ is the availability matrix, where a_{ij} is a binary variable that denotes that availability of alternative j to individual i , $\mathbf{X} = [f_{ij}]$ is a feature matrix that is used to compute the utilities of the alternatives to each individual in the population V_{ij} .

The constraints to the likelihood maximization problem are:

- If nest node a is included its scale parameter should be greater than 1:

$$x_{ra} \leq \mu$$

- If included, nest node a can not be degenerate (i.e., must contain at least two alternatives):

$$2y_a \leq x_{a1} + x_{a2} + x_{a3} \leq 10y_a$$

- Unless included, no connections can be made to nest node a :

$$x_{ra} \leq y_a$$

- The sum of incident edges to each choice node should be one:

$$x_{rj} + x_{aj} = 1 \quad j = 1, 2, 3$$

- The total number of edges must be one less than the total number of nodes

$$x_{r1} + x_{r2} + x_{r3} + x_{ra} + x_{a1} + x_{a2} + x_{a3} \leq (y_a + 3 + 1) - 1$$

In this simple example, cycle elimination constraints were not needed because cycles can not be formed using one nest.

2.3 Regularization

Regularization is a key concept in machine learning techniques. The core idea is to appropriately penalize complexity in the model to avoid fitting to noise (i.e., overfitting). There two questions here: *What* to penalize and by *what amount*. The optimal amount of penalty can be decided by evaluating the model on a hold-out dataset: a model with an optimal amount of penalty (the best model) does best on data the training procedure hasn't seen. This is a common technique in machine learning known as cross-validation. On how to penalize complexity, there are two ways that make a nested logit model more complicated:

1. The number of nests
2. The nesting level (or the tree height¹)

Usually, with traditional machine learning techniques, the training likelihood is a non-decreasing function of complexity. For example, adding regressors to a linear regression model cannot worsen the training likelihood. This is because the training procedure can simply set the coefficients of the added regressors to zero and obtain the same likelihood as a model without the additional regressors. In the nested logit structure learning problem however we do *not* expect any trends on the likelihood of the training dataset. This is somewhat counter-intuitive and is due to the non-degeneracy constraints. We look at three possible cases and provide counter examples that show that the training likelihood does not necessarily improve by increasing complexity as defined by the number of nests and the nesting level:

1. *Increasing the number of nests by 1, while holding the nesting level constant:* Consider the optimal tree with 2 nests and height 2 shown in the left of Figure 2-3. In this particular case, we can increase the number of nests by 1 without increasing the tree height by putting leafs 3 and 4 in a new nest c . The likelihood *cannot* worsen, since in the model on the right, the scale parameter of the new nest c can be set to the value of the root scale parameter to obtain a model with the exact same likelihood as the model on the left.

¹The height of a tree is the depth of its deepest node

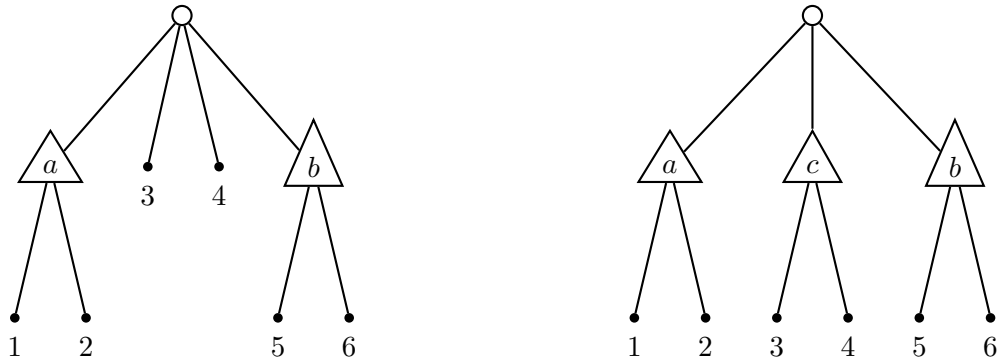


Figure 2-3: A case where it *is* possible to increase the number of nests without worsening the likelihood

However, it is not always possible to increase the number of nests without worsening the training likelihood. As an example, consider the optimal tree with 2 nests and height 2 shown below in Figure 2-4, and consider the problem of adding a third nest c to this tree. It is not possible to add this nest without either increasing the nesting level or running into degeneracy.

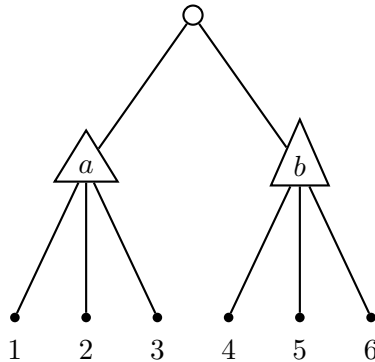


Figure 2-4: Counter example: Increasing the number of nests does not always improve the training likelihood

2. *Increasing the nesting level by 1, while holding the number of nests constant:* Since increasing the nesting level without adding any additional nests would entail changing the nesting structure of a present nest or nests, there is no guarantee that the likelihood cannot worsen.
3. *Increasing the number of nests and the nesting level each by 1:* As in case 1, it is possible to find cases where the trend does hold. For example, consider the optimal tree with 2 nests and height 3 shown in the left of Figure 2-5. Leaf nodes 4 and 5 can be nested together in a new nest c with the same scale parameter as nest b increasing the number of nests and the nesting level without worsening the training likelihood.

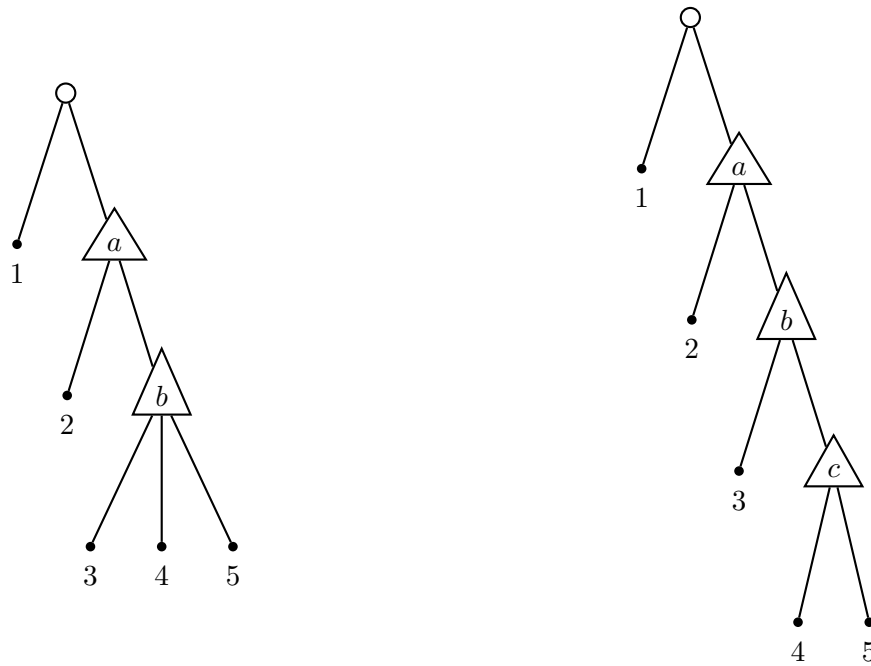


Figure 2-5: A case where increasing the number of nests and nesting level each by one does not worsen the likelihood

In general, there is no guarantee that the likelihood cannot worsen when both the number of nests and the nesting level are increased by one. As an example, consider the tree with 2 nests and height 3 shown below. The only way of increasing the nesting level of this tree is by nesting leaf nodes 4 and 5 together in one new nest c . However nest b will be a degenerate nest in this case.

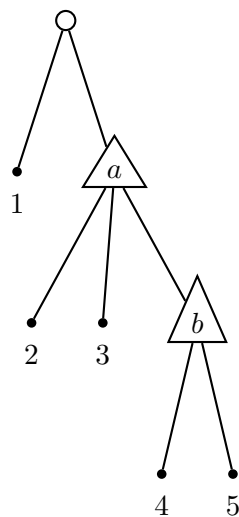


Figure 2-6: A counter example: Increasing the number of nests and nesting level can worsen likelihood

2.4 Summary

Given alternative and feature specific data \mathbf{X} , and observations c_{na} for individuals $n \in \mathcal{I}$ and alternatives $a \in \mathcal{C}$, the problem of determining the optimal tree structure \mathbb{T} and model parameters β and $\mu_{\mathbb{T}}$ was formulated as a maximization problem with a nonlinear non-concave objective function and exponentially many linear constraints. We also saw that the likelihood presents a few challenges beyond its non-concavity. We discuss these at length and study in the next chapter how to exploit the tree structure of \mathbb{T} to make this optimization problem tractable.

Chapter 3

Solution by Linear Outer Approximation

Truth is much too complicated to allow anything but approximations.

John von Neumann

Chapter 2 introduced a formulation of the nested logit structure learning problem (NLSLP) as a mixed integer non-linear program. This chapter deals with finding a practical solution algorithm to this problem. Despite having a closed formulation, NLSLP brings with it several challenges which ultimately shape our approach to finding a solution method:

1. *The likelihood function can only be evaluated at tree solutions.* This is because the inclusive values are defined recursively, and therefore the presence of a cycle will introduce circular references. We will later see that this is also true of the gradients of the likelihood function. Furthermore, since the likelihood function is defined in terms of all possible tree paths from the root to each of the choice leaf nodes, it is not possible to explicitly load the entire function on a computer for problems of practical size. We consider an efficient method later in this chapter for evaluating the likelihood function at a tree solution without having to enumerate all possible paths from the root as the closed form expression for the likelihood would suggest.
2. The inclusive values introduce another complication, namely *the coupling together of all the model parameters*. This precludes the possibility of using local search algorithms that rely on evaluating the effect on the likelihood of the inclusion or exclusion of an edge. In other words, the likelihood can not be decomposed by edge effects. If that were the case, it would suffice to use a maximum spanning tree algorithm to arrive at the optimal structure.
3. *The likelihood function we seek to maximize is jointly non-concave in the discrete and continuous variables.* Once the discrete variables are fixed, however, the problem is reduced to

the usual nested logit model estimation with the addition of the linear scale constraints for which several global optimization techniques already exist and are used in practice such as non-linear branch and bound. Note that if in addition to the discrete structural variables, the scale parameters are also fixed, the likelihood function is concave. However in this case, the estimated values of the betas may be biased.

4. *The number of constraints is exponential in the number of nodes in the tree.* Recall that the cycle elimination constraints are applied to every subset of the nodes of the graph. Furthermore, our regularization framework requires enforcing a specific tree height. This is done by setting the length of a branch to the specified tree height and requiring that all other branches not exceed the length of that branch. This would in general require an exponential number of linear constraints (by limiting the number of edges on every possible path from the root to each alternative). We discuss a way of dealing with this efficiently in Section 3.3.3.
5. *The discrete variables cannot be relaxed.* Methods in the literature of obtaining concave relaxations of functions depend on the ability to evaluate the likelihood at relaxations of the discrete variables [17]. In the NLSLP however, the discrete variables represent structural variables and can not be relaxed: an edge is either present or not.

In lieu of this, we find that the most appropriate solution methodology is through a variation of the linear outer approximation algorithm [12]. We view the discrete structural variables as *complicating variables* and instead of solving the optimization problem in “one go”, we iteratively solve two easier subproblems. The first subproblem deals with estimating the nested logit model parameters for a fixed structure, the second problem finds which nesting logit structure looks most promising at every iteration. We discuss this procedure at length in the next section.

3.1 General algorithm overview

Let m be the number alternatives in an NLSLP. We define $f(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\mu}) = -\mathcal{L}(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\mu} | \mathbf{X})$, and we formulate the NLSLP as the following optimization problem

$$\mathbf{z}^* = \min_{\mathbf{x}, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\mu}} f(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\mu}) \quad (3.1)$$

$$\text{s.t. } \mathbf{x}, \mathbf{y} \in \mathcal{T} \quad (3.2)$$

$$\mathbf{C}\mathbf{x} + \mathbf{B}\boldsymbol{\mu} \leq \mathbf{d} \quad (3.3)$$

$$\boldsymbol{\mu} \in \mathbb{R}^{m-1}, \boldsymbol{\beta} \in \mathbb{R}^p \quad (3.4)$$

$$\mathbf{x} \in \{0, 1\}^{2m-1 \times 2m-1} \quad (3.5)$$

$$\mathbf{y} \in \{0, 1\}^{m-2} \quad (3.6)$$

Where \mathcal{T} is the set of binary vectors (\mathbf{x}, \mathbf{y}) that satisfy the arborescence (2.15-2.16) and structural constraints (2.18-2.26), and for some matrices \mathbf{C} and \mathbf{B} and vector \mathbf{d} that describe the scale constraints (2.17).

Now, for any feasible tree solution $\mathbf{x}^{(k)}$, we define the nonlinear sub-problem $\text{NLP}^{(k)}$ as

$$\mathbf{z}_{\text{NLP}}(\mathbf{x}^{(k)}) = \min_{\boldsymbol{\beta}, \boldsymbol{\mu}} f(\mathbf{x}^{(k)}, \boldsymbol{\beta}, \boldsymbol{\mu}) \quad (3.7)$$

$$\text{s.t. } \mathbf{C}\mathbf{x}^{(k)} + \mathbf{B}\boldsymbol{\mu} \leq \mathbf{d} \quad (3.8)$$

$$\boldsymbol{\mu} \in \mathbb{R}^{m-1}, \boldsymbol{\beta} \in \mathbb{R}^p \quad (3.9)$$

As the feasible set of $\text{NLP}^{(k)}$ is a subset of the feasible set of the original problem, we have that for all $\mathbf{x}^{(k)} \in \mathcal{T}$,

$$\mathbf{z}^* \leq \mathbf{z}_{\text{NLP}}(\mathbf{x}^{(k)}) \quad (3.10)$$

In other words, the solution to any of the non-linear sub-problems $\text{NLP}^{(k)}$ provides a rigorous upper bound on the objective function value of \mathbf{z}^* . We refer to this problem as an upper bounding sub-problem.

Next, we approximate the function f , as the maximum of its linear approximations around a set of feasible points $\mathcal{O}^{(k)} = \{(\mathbf{x}^{(1)}, \boldsymbol{\beta}^{(1)}, \boldsymbol{\mu}^{(1)}), \dots, (\mathbf{x}^{(k)}, \boldsymbol{\beta}^{(k)}, \boldsymbol{\mu}^{(k)})\}$. If f were a convex function then the following problem, called the Master Mixed-Integer Linear Program (MILP), would always provide a lower bound to the original optimization problem, i.e.,

$$\mathbf{z}_{\text{MILP}}^{(k)} \leq \mathbf{z}^* \quad (3.11)$$

The ‘‘lower bounding’’ MILP master problem is given by:

$$\mathbf{z}_{\text{MILP}}^{(k)} = \min_{\eta, \mathbf{x}, \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\mu}} \eta \quad (3.12)$$

$$\text{s.t. } \eta \geq f(\mathbf{x}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\mu}^{(i)}) + \nabla f(\mathbf{x}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\mu}^{(i)})^T \begin{bmatrix} \mathbf{x} - \mathbf{x}^{(i)} \\ \boldsymbol{\beta} - \boldsymbol{\beta}^{(i)} \\ \boldsymbol{\mu} - \boldsymbol{\mu}^{(i)} \end{bmatrix} \quad \forall (\mathbf{x}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\mu}^{(i)}) \in \mathcal{O}^{(k)} \quad (3.13)$$

$$\mathbf{x} \in \mathcal{T} \quad (3.14)$$

$$\mathbf{C}\mathbf{x} + \mathbf{B}\boldsymbol{\mu} \leq \mathbf{d} \quad (3.15)$$

$$\boldsymbol{\mu} \in \mathbb{R}^{m-1}, \boldsymbol{\beta} \in \mathbb{R}^p \quad (3.16)$$

$$\mathbf{x} \in \{0, 1\}^{2m-1 \times 2m-1} \quad (3.17)$$

$$\mathbf{y} \in \{0, 1\}^{m-2} \quad (3.18)$$

The representation above is the so-called epigraph formulation, where the function f is moved out of the objective into the feasible set. If f is convex the linearizations around the set of points $\mathcal{O}^{(k)}$ overestimate the feasible region and we obtain a lower bound on the objective function value as stated in (3.11). Since f is not convex, the tangent hyper-planes are not necessarily global under-estimators and the MILP problem above may cut off regions of the feasible space. An established heuristic to help prevent this is to allow the linearizations to move away from the feasible region. This is done through the use of artificial non-negative variables that are penalized in the objective [18].

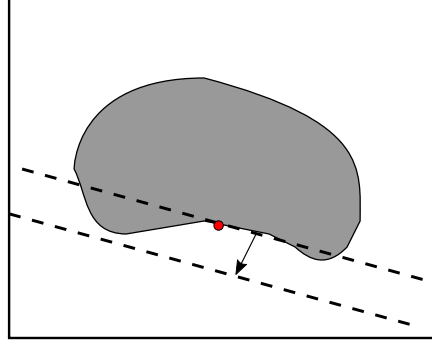


Figure 3-1: Suppose a linearization cut is added around the red point. Since the feasible region is not a convex set, the hyperplane may cut off points of the feasible set. This can be avoided if the cut is translated appropriately.

Linear Outer approximation tackles the original optimization problem by iteratively solving a sequence of two “easier” problems: a Master Mixed-Integer Linear program (MILP) and an NLP subproblem [12]. The algorithm can be described as follows:

Linear Outer Approximation Algorithm

1. Start with a feasible tree solution.
2. Solve the optimization problem with the binary variables (\mathbf{x}, \mathbf{y}) fixed to estimate the taste and scale parameters $(\boldsymbol{\beta}$ and $\boldsymbol{\mu}$).
3. Add linearizing constraints around the found optimal to the linear constraints already present according to (3.15). The new model is referred to as the Master MILP.
4. Solve the Master MILP.
5. The binary part of the resulting optimal solution is then fixed, and the original problem is solved as a nonlinear subproblem.
6. Again, a linearization is carried out around the optimal solution and the new linear constraints are added to the master MILP.
7. Steps 3-5 are repeated until a termination criteria is met (see below).

A few remarks are in order:

1. It is easy to find an initial feasible solution to start the algorithm. For example the multinomial logit tree is always a feasible nesting tree.
2. In step 6, additional constraints are added to cut-off previously found trees and all trees in their equivalence class to guarantee finite convergence and prevent cycling behavior. In our

implementation the nests are labelled, however the nest labels have no effect on the likelihood. When cutting a previously visited tree, one must also cut, from the feasible set, all trees in its equivalence class, i.e., all trees such that when the nest labels are removed, the resulting tree structure is the same. The exact form of the cuts is discussed in Section 3.2.3

3. In practice, linear outer approximation may take a large number of iterations to converge. In fact, there are documented cases where the algorithm visits all feasible points before converging. Since usually, a large portion of the optimally gap (the difference in objective function value between the NLP and MILP) is closed during the first few iterations, a commonly used termination criteria is iteration limit. Another criteria is the worsening of the objective function value of two successive non-linear subproblems [11]. We use an iteration limit in our implementation.

As linearizations are added, the master MILP becomes an improved approximation of the original optimization problem. Convergence to an optimum occurs when the value of the master MILP is worse than the value associated with the NLP subproblem, and the optimum is guaranteed to be a global optimum if the function f is convex. *Since f is not a convex function convergence to a global optimum cannot be guaranteed.*

3.2 Practical Matters

In this section we discuss a few practical implementation details. Crucial to the outer approximation algorithm is the ability to evaluate the value of the function f and its gradients ∇f at points in the feasible set. Since the likelihood function is defined in terms of all possible paths from the root to the leaf nodes, direct evaluation of this function is prohibitive for large choice sets. Fortunately, the tree structure of the problem can be exploited as a work around as we see in Section 3.2.1.

In Section 3.2.2, we discuss a method of dealing with the exponentially many constraints in the Master MILP. We end with a note on the actual code implementation of this problem.

3.2.1 Evaluating the likelihood and its gradients

Efficient evaluation of the likelihood function

Let $\{r \rightarrow a\}_{\mathcal{G}}$ denote the set of all possible paths from r to $a \in \mathcal{S}$ on graph \mathcal{G} . If \mathcal{G} is a tree, the path is unique by definition. Formally, $\{r \rightarrow a\}_{\mathcal{G}}$ is a set of sets of ordered sequences of nodes visited on the path from r to a . For a path $l \in \{r \rightarrow a\}_{\mathcal{G}}$ denote these nodes by $b_l^{(1)}, \dots, b_l^{(s)}$, where $b_l^{(1)} = r$ and $b_l^{(s)} = a$, and s is the length of the path l . The log-likelihood derived in (2.10) can be

rewritten as follows:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \beta | \mathbf{X}_n) = \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \sum_{l \in \{r \rightarrow a\}} x_l \ln \mathbb{P}(a|l) \right), \quad (3.19)$$

where,

$$\ln \mathbb{P}(a|l) = \mu_{b_l^{(s-1)}} V_{na} + \sum_{i=2}^{s-2} (\mu_{b_l^{(s-i)}} - \mu_{b_l^{(s-i+1)}}) \Gamma_{nb_l^{(s-i+1)}} + (\mu_r - \mu_{b_l^{(2)}}) \Gamma_{nb_l^{(2)}} - \mu_r \Gamma_{nr}.$$

Evaluating (3.19), “top-down” would require enumerating all paths $l \in \{r \rightarrow a\}$ for each $a \in \mathcal{S}$ - a prohibitive task even for small choice sets. Instead, consider the following algorithm for efficiently computing the term $\left(c_{na} \sum_{l \in \{r \rightarrow a\}} x_l \ln \mathbb{P}(a|l) \right)$ in (3.19) at tree solutions for a fixed $n \in \mathcal{I}$ and $a \in \mathcal{S}$. At such solutions, there is a unique path $l \in \{r \rightarrow a\}$. Denote this path by the set of nodes $b_l^{(1)}, \dots, b_l^{(s)}$ where $b_l^{(1)} = r$ and $b_l^{(s)} = a$, where s is the length of the path l which we do not necessarily know a priori, we can compute the required contribution as follows:

1. If $c_{na} = 1$ continue to step 2, otherwise the contribution is zero.
2. Start at a leaf node a and propagate to the node’s parent $B(a)$. Add to the likelihood, the quantity $\mu_{B(a)} V_{na}$.
3. If the current node is the root node add the quantity $-\mu_r \Gamma_{nr}$ to the likelihood and stop. Otherwise, propagate to the current node’s parent $B(B(a))$ and add the following quantity to the likelihood $(\mu_{B(B(a))} - \mu_{B(a)}) \Gamma_{B(a)}$.
4. Continue adding contributions as in step 2 until the root node is reached.

Computing gradients

Central to the linear outer approximation algorithm is the availability of gradients of the likelihood function at specified tree solutions. We make a distinction here between the continuous variables β and μ , and the discrete variables \mathbf{x} .

Derivatives of the likelihood function \mathcal{L} with respect to continuous variables are computed through auto-differentiation.

Auto-differentiation however, can not reliably handle derivatives with respect to discrete variables. We resort to analytical differentiation and find that closed form derivatives exist, and can be efficiently evaluated at tree solutions.

The discrete variables \mathbf{x} can be broken down into four distinct types. The derivative of the likelihood function with respect to each of these four types has a different closed form:

1. First order partial derivative of the likelihood function with respect to edges between the root and alternatives x_{ra}

$$\begin{aligned}
\left. \frac{\partial \mathcal{L}}{\partial x_{ra}} \right|_{\mathbb{T}} &= \sum_{n \in \mathcal{I}} \left(c_{na} (\ln \mathbb{P}(a|\{r, a\})) + \sum_{\{r \rightarrow a\}} x_{r \rightarrow a} \frac{\partial \ln \mathbb{P}(a|r \rightarrow a)}{\partial x_{ra}} \right) + \sum_{a' \in \mathcal{S} \setminus \{a\}} c_{na'} \left(\sum_{r \rightarrow a} x_{r \rightarrow a} \frac{\partial \ln \mathbb{P}(a|r \rightarrow a)}{\partial x_{ra}} \right) \\
&= \sum_{n \in \mathcal{I}} \left(c_{na} (\mu_r V_{na} - \mu_r \Gamma_{nr}) + \sum_{a' \in \mathcal{S}} c_{na'} \left(-\mu_r \frac{\partial \Gamma_{nr}}{\partial x_{ra}} \right) \right) \\
&= \sum_{n \in \mathcal{I}} \left(c_{na} (\mu_r V_{na} - \mu_r \Gamma_{nr}) - \sum_{a' \in \mathcal{S}} c_{na'} \exp(\mu_r (V_{na} - \Gamma_{nr})) \right)
\end{aligned}$$

2. First order partial derivative of the likelihood function with respect to edges between the root and nests x_{rb} First note that we can rewrite the log-likelihood as

$$\begin{aligned}
\mathcal{L} &= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{r \rightarrow a} x_{r \rightarrow a} \ln \mathbb{P}(a|r \rightarrow a) \right) \right) \\
&= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{b \in \mathcal{N}} \sum_{b \rightarrow a} x_{rb} x_{b \rightarrow a} \ln \mathbb{P}(a|r, b \rightarrow a) \right) \right)
\end{aligned}$$

Then,

$$\begin{aligned}
\left. \frac{\partial \mathcal{L}}{\partial x_{rb}} \right|_{\mathbb{T}} &= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{b \rightarrow a} (x_{b \rightarrow a} \ln \mathbb{P}(a|r, b \rightarrow a) - x_{rb} x_{b \rightarrow a} \mu_r \frac{\partial \Gamma_{nr}}{\partial x_{ra}}) - \sum_{b' \in \mathcal{N} \setminus \{b\}} \sum_{b' \rightarrow a} x_{rb'} x_{b' \rightarrow a} \mu_r \frac{\partial \Gamma_{nr}}{\partial x_{ra}} \right) \right) \\
&= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{b \rightarrow a} x_{b \rightarrow a} \ln \mathbb{P}(a|r, b \rightarrow a) - \sum_{r \rightarrow a} x_{r \rightarrow a} \mu_r \frac{\partial \Gamma_{nr}}{\partial x_{ra}} \right) \right) \\
&= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{b \rightarrow a} x_{b \rightarrow a} \ln \mathbb{P}(a|r, b \rightarrow a) - \sum_{r \rightarrow a} x_{r \rightarrow a} \exp(\mu_r (\Gamma_{nb} - \Gamma_{nr})) \right) \right)
\end{aligned}$$

Now, at tree solutions, there is a unique path $r \rightarrow a$ such that $x_{r \rightarrow a} = 1$. We can therefore do away with the fourth summation.

$$\left. \frac{\partial \mathcal{L}}{\partial x_{rb}} \right|_{\mathbb{T}} = \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{b \rightarrow a} x_{b \rightarrow a} \ln \mathbb{P}(a|r, b \rightarrow a) - \exp(\mu_r (\Gamma_{nb} - \Gamma_{nr})) \right) \right) \quad (3.20)$$

3. First order partial derivative of the likelihood function with respect to edges between nests and other nests $x_{bb'}$ First note that we can rewrite the log-likelihood as

$$\begin{aligned}
\mathcal{L} &= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{r \rightarrow a} x_{r \rightarrow a} \ln \mathbb{P}(a|r \rightarrow a) \right) \right) \\
&= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{b_1, b_2 \in \mathcal{N}} \sum_{r \rightarrow b_1} \sum_{b_2 \rightarrow a} x_{r \rightarrow b_1} x_{b_1 b_2} x_{b_2 \rightarrow a} \ln \mathbb{P}(a|r \rightarrow b_1, b_2 \rightarrow a) \right) \right)
\end{aligned}$$

Then,

$$\left. \frac{\partial \mathcal{L}}{\partial x_{bb'}} \right|_{\mathbb{T}} = \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{r \rightarrow b} \sum_{b' \rightarrow a} x_{r \rightarrow b} x_{b' \rightarrow a} \ln \mathbb{P}(a|r \rightarrow b, b' \rightarrow a) + \sum_{r \rightarrow a} x_{r \rightarrow a} \frac{\partial \ln \mathbb{P}(a|r \rightarrow a)}{\partial x_{bb'}} \right) \right)$$

4. First order partial derivative of the likelihood function with respect to edges between nests and alternatives x_{ba}

$$\begin{aligned}\mathcal{L} &= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{\{r \rightarrow a\}} x_{\{r \rightarrow a\}} \ln \mathbb{P}(a|\{r \rightarrow a\}) \right) \right) \\ &= \sum_{n \in \mathcal{I}} \sum_{a \in \mathcal{S}} \left(c_{na} \left(\sum_{b \in \mathcal{N}} \sum_{r \rightarrow b} x_{r \rightarrow b} x_{ba} \ln \mathbb{P}(a|r \rightarrow b, a) \right) \right)\end{aligned}$$

Then,

$$\frac{\partial \mathcal{L}}{\partial x_{ba}} = \sum_{n \in \mathcal{I}} \left(c_{na} \left(\sum_{r \rightarrow b} x_{r \rightarrow b} \ln \mathbb{P}(a|r \rightarrow b, a) \right) + \sum_{a' \in \mathcal{S}} c_{na'} \left(\sum_{\{r \rightarrow a'\}} x_{\{r \rightarrow a'\}} \frac{\partial \ln \mathbb{P}(a|\{r \rightarrow a'\})}{\partial x_{bb'}} \right) \right)$$

All four types of derivatives require summations over all possible paths. At tree solutions, a simple variation of the algorithm introduced to efficiently compute the likelihood can be used to also compute the derivatives efficiently.

3.2.2 Dealing with Exponentially many constraints

Cycle elimination constraints

The cycle elimination constraints are required to be enforced in the MILP subproblem to avoid passing over non-tree solutions to the NLP subproblem (which can only be solved at tree solutions). Note that since there is one constraint for every possible subset of nodes of the graph \mathcal{G} , these constraints grow exponentially fast with the size of the graph. These constraints however need not be generated at once and added to the MILP. Instead, they can be generated on the fly using the so-called lazy constraint generation approach. The optimization problem is first solved without these constraints. Then a “separation oracle” is used to determine which constraints are violated and the violated constraints are added and the problem is re-solved. It is important that the separation oracle run time is better than exponential time. A naive exponential time oracle will generate and check each of the constraints. Instead, it suffices to solve the max flow problem with the root as the source node and each alternative as sink nodes which requires at worst polynomial time[14].

MILP anti-cycling cuts

As noted in Section 3.1, certain cuts need to be added to the MILP to prevent revisiting solutions. Furthermore since nest labels have no effect on the likelihood, there is in fact an equivalence class of tree solutions. All trees in the same equivalence class share a common “signature” -namely the ancestry of the leaf nodes. This is used to determine if in fact the current solution belongs to the equivalence class of a tree that has been previously visited, and cut that tree from the current solution accordingly.

Suppose we wish to exclude a particular tree solution \mathbf{x} from the feasible set. Define the index sets

$$O = \{i : x_i = 1\} \tag{3.21}$$

$$Z = \{i : x_i = 0\} \tag{3.22}$$

The integer cut is defined as

$$\sum_{i \in O} x_i - \sum_{i \in Z} x_i \leq |O| - 1 \tag{3.23}$$

Again, these cuts are not generated at once but instead added on the fly as needed.

3.2.3 Implementation Details

The nested logit structure learning problem has been implemented in the Julia programming language [5]. The implementation is flexible enough to handle any form of linear utility specification. Gurboi solver is used to solve the MILP master problems. The JuMP interface allows for user defined cuts, which is critical since we have two types of custom cuts (i) the subtour elimination cuts, and (ii) cuts of previously visited trees and trees in their equivalency class. KNITRO is used to solve the NLP subproblems. KNITRO applies a Sequential Quadratic Programming (SQP) type algorithm which solves a sequence of quadratic programming (QP) subproblems to solve the NLP.

The implementation can be accessed through the following github repository:
github.com/ymedhat95/nested-logit.git

Chapter 4

Experiments with real and synthetic datasets

The proof of the pudding is in the eating.

English proverb

4.1 Experiments with synthetic data

4.1.1 Data generation

A synthetic dataset of $n = 800$ observations was produced for testing purposes. The data was split equally into training and validation datasets. There are eight choices, labelled sequentially, in this dataset. The choices are grouped together into four nests as shown in the figure below.

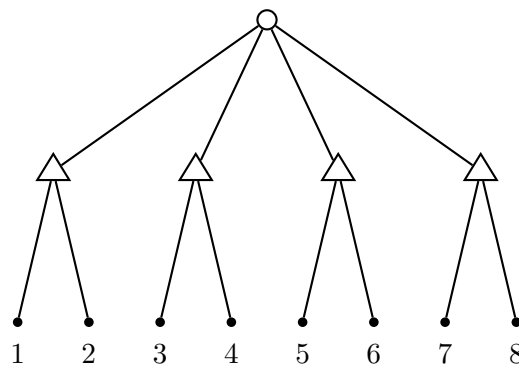


Figure 4-1: Nesting structure underlying the synthetic data

A random matrix $\mathbf{A}_{n \times 8} = [a_{ij}]$ was first generated to simulate varying availabilities of alternatives to individuals in the sample. Utilities were generated for each individual for the available alternatives in the choice set according to the following equations: $V_{ij} = \alpha_j + \beta_j x_{ij}$. The alternative specific and individual specific features x_{ij} were generated according to a normal distribution with mean μ_j and variance σ_j generated uniformly at random.

4.1.2 Results

The optimization problem was solved using linear outer approximation for all feasible settings of the number of nests, and tree height parameter combinations (20 in total). Figure 4-2 shows the outer approximation algorithm iterations for one such feasible combination where the optimal value is known.

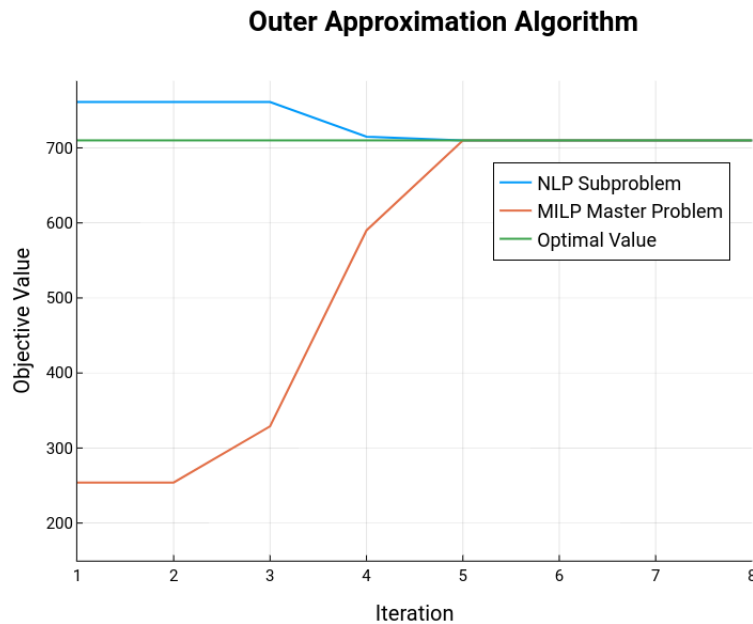


Figure 4-2: A profile of the outer approximation algorithm in action for the nested logit optimization problem with *Number of Nests* = 4 and *Tree height* = 2

Figure 4-3 next page, shows the best negative log-likelihood obtained at each number of nests, and tree height setting. Here, a slightly more complicated model outperforms the true structure on the training set. However, when the negative log-likelihood is evaluated for each of the structures obtained on the validation set, the true tree structure performs best and is correctly distinguished from the other sub-optimal structures as shown in Figure 4-5.

In total, the number of NLP problems solved was 76 - a small fraction of the large feasible set of all possible structures. In practice, the user may, for the sake of expediency and at an increased risk of arriving at a sub-optimal structure, reduce the number of iterations used as a termination

criteria for the outer approximation algorithm or limit the number of nest and tree height parameter combinations tried.

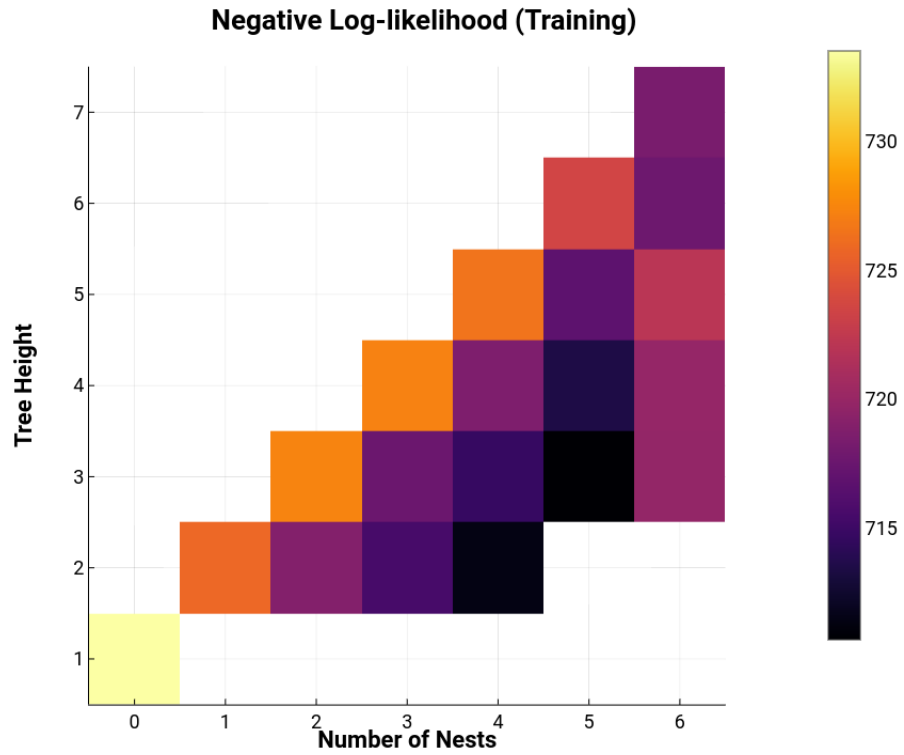


Figure 4-3: Negative Log-likelihood values of the (local) optimal nesting structure under all feasible *Number of Nests* and *Tree Height* combinations.

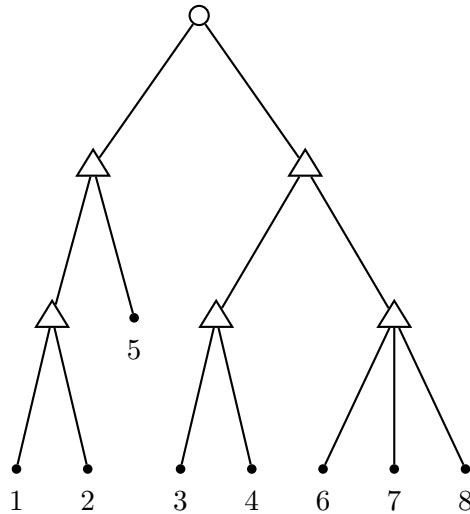


Figure 4-4: The nesting structure with the best training objective function value of 710.68. The true nesting structure, came in second with objective value 711.42

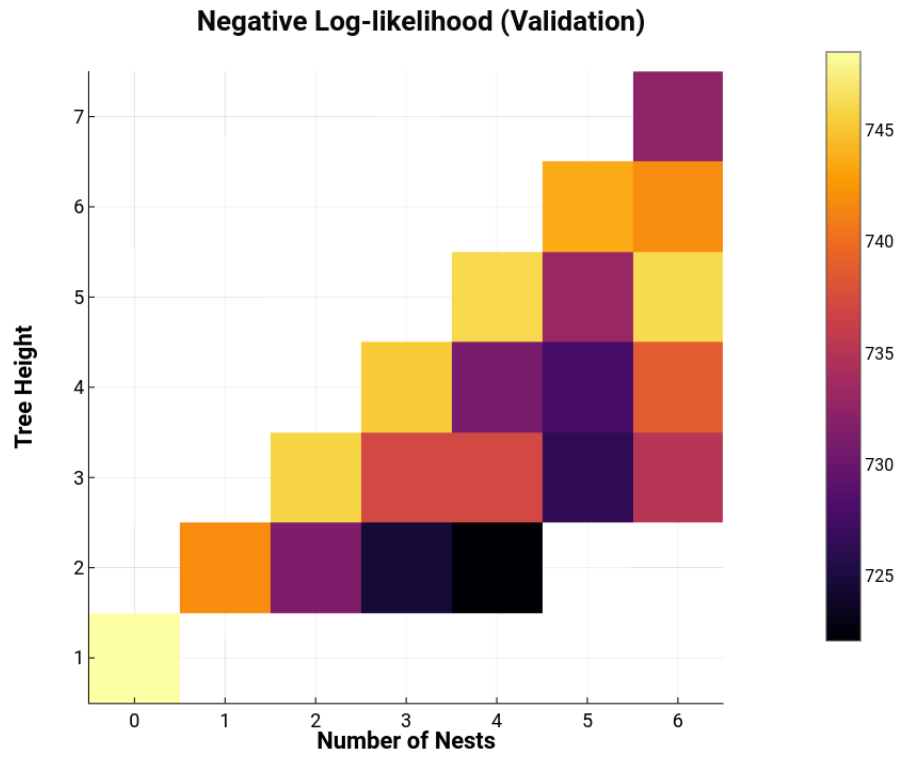


Figure 4-5: Negative log-likelihood evaluated on the validation dataset

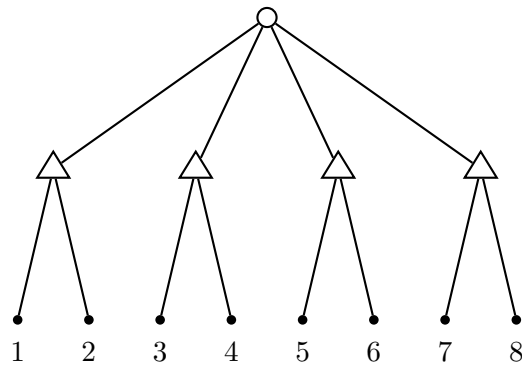


Figure 4-6: Nesting structure with best objective value of 722.41.

	True model	Best Model (Training)	Best Model (Validation)
Structure & scales			
$-\mathcal{L}$	721.54	710.68	722.41
α_1	0.00	0.00	0.00
α_2	1.00	0.82	1.12
α_3	1.00	0.50	1.00
α_4	1.00	0.43	0.91
α_5	1.00	0.62	0.85
α_6	1.00	0.78	0.84
α_7	1.00	0.70	0.70
α_8	1.00	0.77	0.81
β_1	1.00	0.79	1.00
β_2	-0.10	-0.09	-0.13
β_3	-0.10	-0.10	-0.10
β_4	-0.10	-0.14	-0.13
β_5	-0.10	-0.08	-0.06
β_6	-0.10	-0.05	-0.09
β_7	-0.10	-0.10	-0.17
β_8	-0.10	-0.07	-0.12

Table 4.1: Estimation results for the synthetic dataset. The scale parameters are shown above each nest

4.2 Experiments with real datasets

4.2.1 Data Description

Data from the 2010 Massachusetts Travel Survey (MTS) and matrices for car and transit travel times and costs, provided by Boston's Central Transportation Planning Staff (CTPS) [10], are used to estimate a nested logit choice model for the work travel mode. The MTS includes activity diaries for 8,000 individuals belonging to 4,400 households. Individuals were asked to fill out all activities they performed on a designated weekday, and to provide the activity location, the transport mode used to arrive at this locations. The survey also collected individual and household characteristics for participants.

There are six main travel modes reported in the survey:

1. Walk
2. Bike
3. Car
4. Car Pool (2 people)
5. Car Pool (3+ people)
6. Public Transit

Alternative specific attributes included in the systematic specification are:

1. Travel Time
2. Travel Cost

Individual specific characteristics include:

1. Possession of a mass transit pass
2. Income
3. Gender

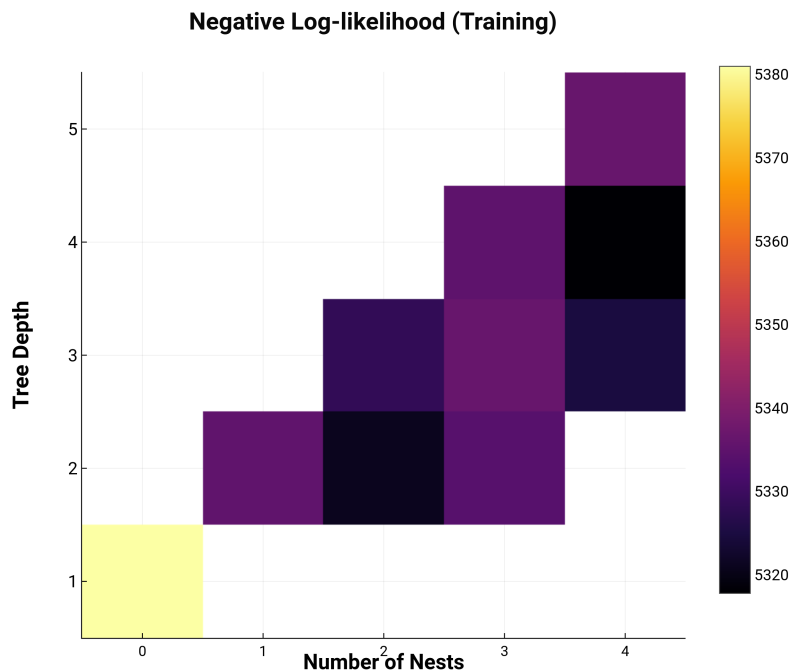
The systematic utilities for each of the travel modes are specified as follows:

$$\begin{aligned}
 V_{walk} &= \beta_{traveltime_{walk}} TT_{walk} + \beta_{transitpass_{walk}} 1_{transitpass} \\
 V_{bike} &= \alpha_{bike} + \beta_{traveltime_{bike}} TT_{bike} + \beta_{transitpass_{bike}} 1_{transitpass} \\
 &\quad + \beta_{income} Income + \beta_{female} 1_{female} \\
 V_{car} &= \alpha_{car} + \beta_{time_{car}} TT_{car} + \beta_{cost_{car}} Cost_{car} + \beta_{income} Income + \beta_{female} 1_{female} \\
 V_{carpool2} &= \alpha_{carpool2} + \beta_{time_{carpool}} TT_{carpool} + \beta_{cost_{carpool}} Cost_{carpool2} \\
 &\quad + \beta_{income} Income + \beta_{female} 1_{female} \\
 V_{carpool3} &= \alpha_{carpool3} + \beta_{time_{carpool}} TT_{carpool} + \beta_{cost_{carpool}} Cost_{carpool3} \\
 &\quad + \beta_{income} Income + \beta_{female} 1_{female} \\
 V_{publictransit} &= \alpha_{publictransit} + \beta_{time_{publictransit}} TT_{publictransit} \\
 &\quad + \beta_{cost_{publictransit}} Cost_{publictransit} + \beta_{transitpass_{publictransit}} 1_{transitpass} \\
 &\quad + \beta_{income} Income + \beta_{female} 1_{female}
 \end{aligned}$$

The goal is to estimate a nested logit model (structure and parameters) from the data.

4.2.2 Results

The figure below shows the negative log-likelihood profile for all feasible combinations for the regularization parameters.



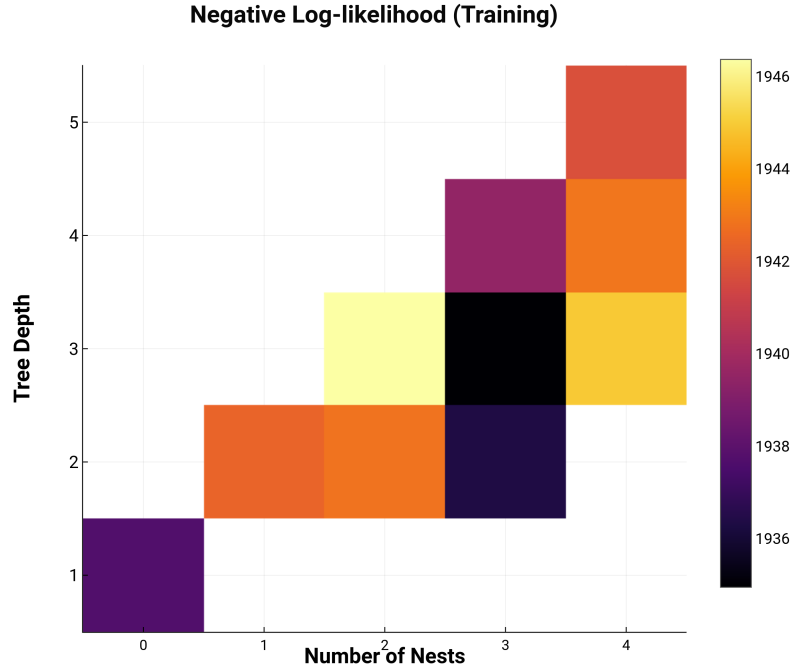


Figure 4-7: Training and validation negative likelihood profiles for the work travel mode choice logit model

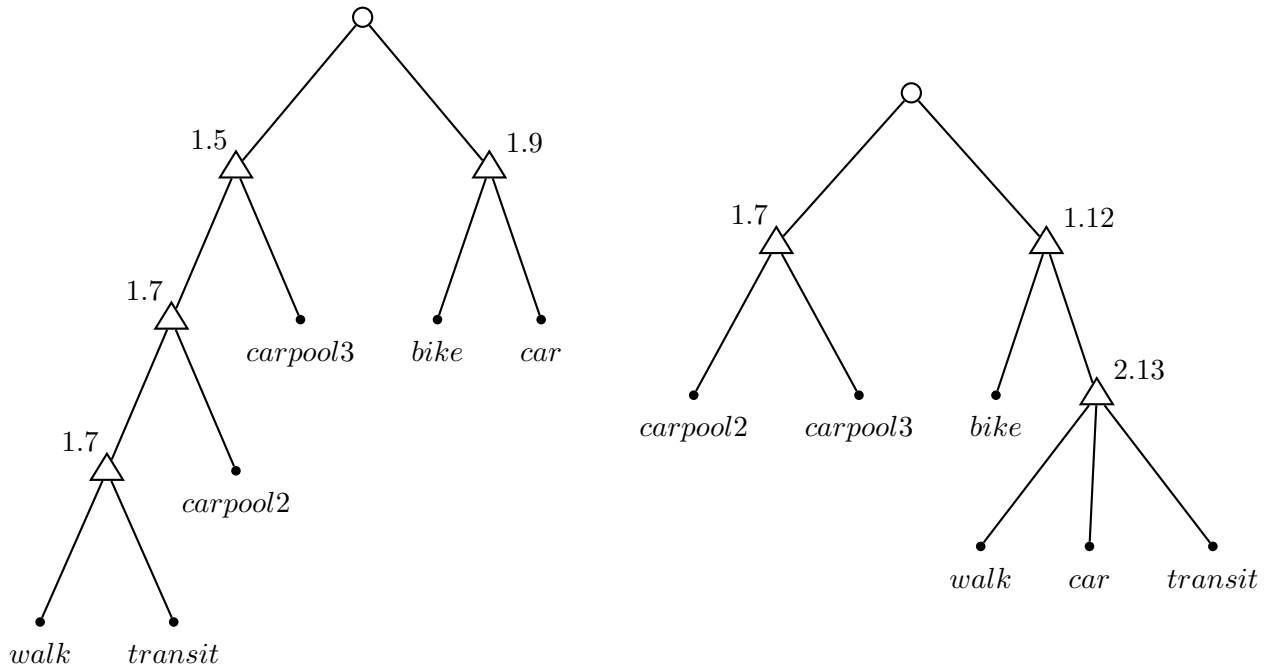


Figure 4-8: Best performing model on training data (left), and on validation data (right)

The best performing model on the validation dataset groups together *walk*, *car*, and *transit* in a lower level nest. The structure implies that the error term for *car* is correlated with *walk* and *transit* and not correlated with *carpool2* and *carpool3*. At a first glance this result is surprising,

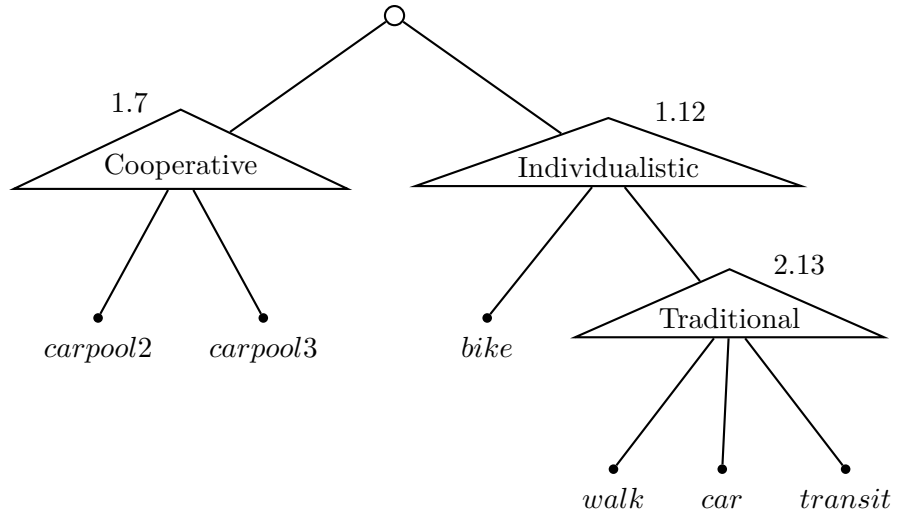
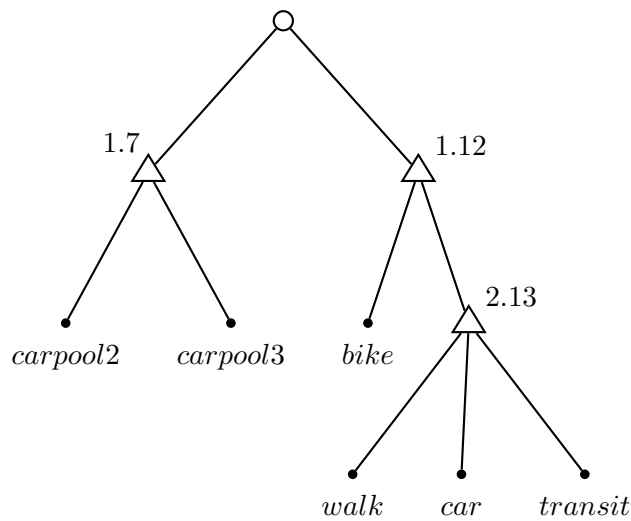


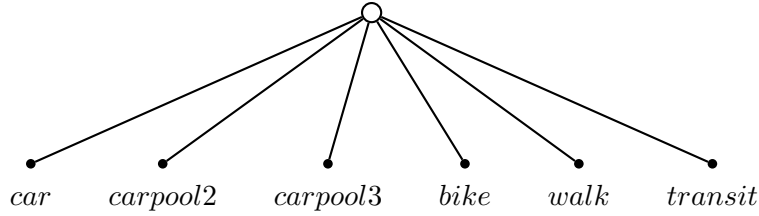
Figure 4-9: A possible interpretation of the learned nesting structure

even counter-intuitive. The learned nesting structure, does provide an interesting insight, and can be interpreted as follows: the decision-maker goes through a thought process in deciding whether to “cooperate” with other people to travel to work not. Car pooling requires cooperating with other decision-makers. Within the non-cooperative “Individualistic” nest, the decision maker chooses between the bike mode and the traditional work travel modes of walk, car, and transit. We look at other alternative nesting specifications (b) and (c) below that the modeler could specify in this case and observe that they perform worse than the learned model (a):

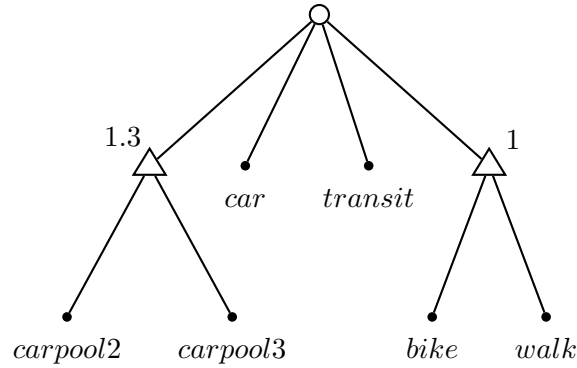
- (a) **Learned Nested Logit Model:** Validation negative log-likelihood $-\mathcal{L} = 1934$



- (b) **Multinomial logit model:** Validation negative log-likelihood $-\mathcal{L} = 1938$



- (c) **Alternative model 1:** Validation negative log-likelihood $-\mathcal{L} = 1936$



Although, the differences in likelihoods are small, each of these models would have vastly different predictions. Suppose for example that the transit fare is decreased. Model (a) would predict that people who walk or drive are more likely to switch to transit than people who bike or car share. Model(b), on the other hand, would predict that all users are equally likely to switch to transit.

4.3 Summary

In this section, we applied the linear outer approximation algorithm presented in Chapter 3 to solve the nested logit structure learning problem introduced in Chapter 2. Using synthetic data we have shown that it is possible to recover the true error structure from the data. Finally we have applied our algorithm to estimate a nested logit model for the work travel mode using Massachusetts travel survey data. The recovered structure, although “unconventional”, did reveal an insight on the behavior of the population under study.

Chapter 5

Conclusion

The object of this thesis is to advance the state-of-the-art in discrete choice models by utilizing the power of optimization in determining the error structure that best explains the choice behavior in the population under study. We have demonstrated that it is possible to recover the correct error structure from the data and we have applied our method to synthetic and real-world datasets.

We associated with a given choice situation, an induced graph with a root node, nest nodes, and leaf nodes representing the choices. The induced graph is always guaranteed to be finite because of the non-degeneracy constraints placed on the nests. We then formulated an optimization problem to determine the best path from the root (through the nest nodes) to the choice leaf nodes, i.e., the best nesting tree structure. The nesting tree needed not be a spanning tree, i.e., it is part of the optimization to choose which of the nest nodes, if any, to include. In addition to finding the best structure (which determines *how* the alternatives are correlated), the optimization problem was also tasked with finding the optimal scale parameters associated with the structure (which determine by *what amount* the alternatives are correlated) and the optimal taste parameters. Complexity was penalized by directly controlling the number of nests and nesting levels of the fitted trees and the optimal penalty was determined through cross-validation.

In advocating for a data-driven approach for specifying a nested logit structure, we are in no way diminishing the role of the modeler or the importance of domain-specific knowledge in specifying and designing good discrete choice models. Recall that the utility of an alternative to an individual is given by a sum of a systematic component and a random component. It is the modeler's purview to correctly specify the systematic part of the utility equation. Specifying the random part, however, is a tricky business and the optimal structure may be counter-intuitive. This part, therefore, is best left to a computer. In fact, the optimal error structure is not independent of the specification of the systematic part. If all aspects of the choice behavior that account for correlation between choices can be fully captured in the systematic part, no nesting is needed.

The major complications faced in the optimization were mainly due to the rather complex forms of the choice probabilities, and the restrictions placed on how the alternatives can be correlated under

the nested logit framework. In particular, the inclusive values coupled together all of the model's parameters, making global optimization very difficult. The nested logit assumptions brought tractability in the form of closed form choice probabilities. This tractability however does not translate to tractability from an optimization point of view.

In logit models, the "simplifying" assumption that the error terms are Gumbel distributed gave us closed-form expressions for the choice probabilities. However, this same assumption is in fact a *complicating* assumption from an optimization point of view since it also gave us the inclusive values that made the optimization problem difficult. Other discrete choice models are even less tractable from an optimization point of view. For example, probit models assume that the errors are normally distributed. There are no closed form probabilities in this case, and trying to optimize over the error structure is a daunting task.

Is it not, therefore, better to not make any assumptions at all regarding the error terms? Instead of working with probabilities we consider working with the utilities directly in a robust optimization framework as a future research direction. Instead of treating the error terms as random variables, we treat them as uncertain parameters belonging to some domain-specific uncertainty set. The framework for treating uncertainties using a robust optimization framework is described here [1]. However, it remains an open question whether such a treatment will lead to valid *econometric* models of choice.

Bibliography

- [1] Chaithanya Bandi and Dimitris Bertsimas. Tractable stochastic analysis in high dimensions via robust optimization. *Mathematical programming*, 134(1):23–70, 2012.
- [2] Moshe E Ben-Akiva. *Structure of passenger travel demand models*. PhD thesis, Massachusetts Institute of Technology, 1973.
- [3] Moshe E Ben-Akiva, Steven R Lerman, and Steven R Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.
- [4] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.
- [5] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.
- [6] Chandra R Bhat. A heteroscedastic extreme value model of intercity travel mode choice. *Transportation Research Part B: Methodological*, 29(6):471–483, 1995.
- [7] Michel Bierlaire. Discrete choice models. In *Operations research and decision aid methodologies in traffic and transportation management*, pages 203–227. Springer, 1998.
- [8] Michel Bierlaire, Tsippy Lotan, and Philippe Toint. On the overspecification of multinomial and nested logit models due to alternative specific constants. *Transportation Science*, 31(4):363–371, 1997.
- [9] Andrew Daly. Estimating “tree” logit models. *Transportation Research Part B: Methodological*, 21(4):251–267, 1987.
- [10] I Viegas de Lima, Mazen Danaf, Arun Akkinapally, CL Azevedo, and Moshe Ben-Akiva. Modelling framework and implementation of activity-and agent-based simulation: Application to the greater boston area. In *Transportation Research Board 97th Ann. Meeting*, 2018.
- [11] Marco A Duran and Ignacio E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339, 1986.
- [12] Roger Fletcher and Sven Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical programming*, 66(1-3):327–349, 1994.
- [13] Frank S Koppelman and Chandra Bhat. A self instructing course in mode choice modeling: multinomial and nested logit models. 2006.

- [14] Bernhard Korte, Jens Vygen, B Korte, and J Vygen. *Combinatorial optimization*, volume 2. Springer, 2012.
- [15] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.
- [16] Daniel McFadden. Econometric models of probabilistic choice. *Structural analysis of discrete data with econometric applications*, 198272, 1981.
- [17] Joseph K Scott, Matthew D Stuber, and Paul I Barton. Generalized mccormick relaxations. *Journal of Global Optimization*, 51(4):569–606, 2011.
- [18] Jagadisan Viswanathan and Ignacio E Grossmann. A combined penalty function and outer-approximation method for minlp optimization. *Computers & Chemical Engineering*, 14(7):769–782, 1990.