Winter 2012

Analysis of Discrete Choice Probit Models with Structured Correlation Matrices

Bhaskara Ravi
Old Dominion University

ANALYSIS OF DISCRETE CHOICE PROBIT MODELS WITH STRUCTURED CORRELATION MATRICES

by

Bhaskara Ravi
B.Sc. April 1997, Andhra University, India
M.Sc. May 1999, University of Hyderabad, India

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MATHEMATICS AND STATISTICS

OLD DOMINION UNIVERSITY
December 2012

Approved by:

N. Rao Chaganty (Director)

Norou Diawara (Member)

Nak-Kyeong Kim (Member)

Juan Du (Member)

ABSTRACT

ANALYSIS OF DISCRETE CHOICE PROBIT MODELS WITH STRUCTURED CORRELATION MATRICES

Bhaskara Ravi
Old Dominion University, 2012
Director: Dr. N. Rao Chaganty

Discrete choice models are very popular in Economics and the conditional logit model is the most widely used model to analyze consumer choice behavior, which was introduced in a seminal paper by McFadden (1974). This model is based on the assumption that the unobserved factors, which determine the consumer choices, are independent and follow a Gumbel distribution, widely known as the Independence of irrelevant Alternatives (IIA) assumption. Alternate models that relax IIA assumption are the Generalized Extreme Value (GEV) models, which allow dependency between unobserved factors. However, GEV models do not incorporate all dependency patterns, other choice behaviors such as random taste variation and repeated responses over time. The discrete choice probit models are the most flexible in the sense that they model any dependence pattern, random taste variations and repeated responses. But, the probit models require evaluations of multivariate normal distribution function, which are difficult to compute. They were not pursued because of this difficulty, except in a few cases with specific patterns in the covariance structures.

In this dissertation, we study the discrete choice probit models for a couple of correlation structures such as equicorrelation and product correlation. Using stochastic representations, we derive and simplify analytical expressions for the computation of choice probabilities for both of the structures. Further, we illustrate the procedure of obtaining maximum likelihood estimates for the model parameters and analytical expressions for the Fisher information matrix to compute their standard errors. Using simulations, we compare the performance of probit models with logit models in both large sample case as well as small samples. We conclude that the probit models are more asymptotically efficient than logit models as correlation increases. We have provided a sample R-code in the appendix that was used for computations.

Finally, a more general form of choice models are presented using multivariate copulas. We presented a brief introduction of discrete choice copula models using the Gaussian copula and the Extreme value copula. Copula representations are useful in building multivariate distributions with several choices for marginals. The discrete choice probit models are Gaussian Copula models with marginals that are standard normal and the GEV models are Extreme Value Copula models with marginals that are extreme value distributions. This work shows a way of constructing new models using copulas by choosing different marginals within the copula representation. For example, a Gaussian Copula choice model with Gumbel marginals or an Extreme Value Copula choice model with normal marginals is possible. Such models are not yet explored to model consumer choice behavior and this provides a road map for future research.

DEDICATED TO MY PARENTS

R. S. N. Murthy and R. Manikyamba

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my mentor and dissertation advisor Professor N. Rao Chaganty for his continuous support and guidance. I am grateful for the opportunity to work as his student and his interest to work in the research area of discrete choice models. His guidance, continuous support and patience was instrumental in completing this research and dissertation.

I would like to thank Professors Norou Diawara, Nak-kyeong Kim and Juan Du for serving on my dissertation committee. I would like to express my gratefulness to Late Dr. Dayanand Naik for his enormous support during my graduate studies and numerous discussions. Special thanks are due to Dr. Norou Diawara for an in-depth review of my dissertation and numerous suggestions. Thanks are also due to Dr. N. Rao Chaganty for arranging research support at the Eastern Virginia Medical School. Statistical consulting projects at the Eastern Virginia Medical School was a rewarding and learning experience.

Finally, I would also like to thank my fellow students, friends, professors and staff members in the Department of Mathematics and Statistics at Old Dominion University, especially Professor Mark Dorrepaal, for providing a pleasant and stimulating environment. I also wish to acknowledge the staff members Barbara Jeffrey, Miriam Venable and Sheila Hegwood for their help during my graduate studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

DISCRETE CHOICE MODELS

1.1 INTRODUCTION

Almost everyday, consumers encounter several choices or alternatives such as which phone to pick, which mode of transport to use, which car to buy, etc. Interestingly, the decision to pick a particular choice not only depends on the characteristics of the consumer but also heavily relies on the characteristics of the choice scenario. Inherently, consumers attach a utility factor to each alternative and they choose the alternative that has the highest utility. Statistical models that study such consumer choice behavior are known as discrete choice models.

1.1.1 CHOICE SET AND ITS PROPERTIES

In a discrete choice setup, consumers are presented with a set of alternatives known as "Choice Set", that has three important characteristics. First, the choice alternatives are mutually exclusive. This characteristic leads to the fact that only one alternative is picked up as a choice and all others are excluded. Second, the choice set is exhaustive so that all possible alternatives are included. Third, the number of alternatives are countably finite. The first two characteristics are less restrictive in the sense that the alternatives can be modified to satisfy the two characteristics. For example, a choice experiment of travel modes that consists of alternatives such as bus, train, or car can have a possibility of a consumer choosing both the bus and the train as a choice. The choice set can then be modified to accommodate "bus and train" as another option so that the list becomes mutually exclusive and exhaustive. The third characteristic is more restrictive in nature in that it defines the dependent variable to be discrete. Further, these three characteristics lead to the property that the total probability of selection is equal to one among all alternatives.

This dissertation follows the style of *Journal of the American Statistical Association.*

1.1.2 CHOICE PROBABILITIES

In discrete choice models, we are interested in modeling the consumer choice behavior that involves computing the probability of choice. The modeling process is based on the utility maximization theory. A subject assigns a value to each alternative in his mind, known as "utility" that is not observed and chooses the one that has the highest utility value. Further, it is assumed that this latent utility comprises of a deterministic component, "mean", and a random component, "error", that is unobserved. We further make distributional assumptions about the random component, that lead to several discrete choice models. The model with an assumption that the unobserved factors are independent and follow a Gumbel distribution is the McFadden's conditional logit model.

1.1.3 DATA SOURCES

Data for discrete choice models usually come from two sources. The first source is household panels or consumer panels, wherein a set of respondents are selected for a pre-determined period of time and their purchase history is recorded. The purchase history consists of all purchases made by households across several categories and is divided into different categories for analysis. For a particular category such as hair care, each purchase transaction has information about the brand bought, date of purchase, purchasing store, number of units bought, its price, any promotion offered, etc. On a given day, if two households bought two different brands, then the brand bought by first household was available as a choice to the second household and vice versa. Looking at all transactions on the same day and in the same store, a choice set can be constructed that represents major part of the market. The purchase history also has information about price, promotion, etc., that governs choice and this information forms the explanatory variables that determine the consumer choice. The second source is data that comes from discrete choice experiments or conjoint analysis. In market research, a survey is designed to evaluate potential market for a new product or an existing product to understand consumer choice or preferences. A virtual choice set, known as "choice card", is created using design of experiment principles. The choice set consists of a several alternatives characterized by product attributes such as brand, price, promotion etc. A set of choice cards are shown to each respondent and their response is observed one after another. This data is analyzed using discrete choice modeling framework. The data from consumer panels is more robust in the sense that it reveals the true behavior of consumers where as the data coming from experiments tend to be less robust due to sampling errors, survey administration bias, etc.

1.1.4 REAL DATA EXAMPLES

Example 1. Detergent brand choice.
In this example, the data is from a market research study and contains information about the brand and price of the laundry detergent purchased by 2657 consumers originally analyzed by Chintagunta and Prasad (1998). The dataset contains the log prices of six detergent brands Tide, Wisk, EraPlus, Surf, Solo, All, as well as the brand chosen by each household. We are interested to model the brand choice with log-price.

Example 2. Data used to study travel mode choice.
The source of the data is Table 21.2 of Greene (2003). This data contains choices made by 210 individuals traveling between Sydney and Melbourne in Australia. The response has four modes of travel namely Air, Train, Bus or Car. The explanatory variables that are specific to the alternative are waiting time, travel cost, travel time, general cost, party size, and a individual specific variable like household income. There are 840 observations by 210 individuals. We are interested in modeling the travel mode choice using the explanatory variables such as time, cost, waiting time, etc.

1.2 ORGANIZATION OF THIS DISSERTATION

Including the current chapter, this dissertation consists of six chapters. In Chapter 2, we discuss the most widely used discrete choice model known as McFadden's conditional logit model. We introduce the notation needed for discrete choice models and describe the formulation that lead to conditional logit model. The remaining sections consist of identifying difference between a regular multinomial logit and conditional logit model, that lead to variety of models to describe market dynamics. Further, we illustrate the estimation of conditional logit model using maximum likelihood method and analyze the laundry detergent example using this procedure.

Finally, we discuss the pitfalls of this model having the Independence of Irrelevant Alternatives (IIA) assumption and conclude with the alternate models that deal with this limitation.

In Chapter 3, we introduce the multinomial discrete choice probit model that fully relaxes IIA assumption. The probit model requires difficult evaluation of multivariate normal distribution function to calculate choice probabilities. We present a simplification of choice probabilities for equicorrelation structure using stochastic representations. We derive the exact analytical expressions of the choice probabilities and describe the estimation of probit model using ML approach. We also derive the analytical expression of Fisher information to compute the standard errors of parameter estimates. Further, we demonstrate that the probit model is more efficient than the logit model asymptotically as well as in small samples. Finally, we illustrate the probit model using laundry detergent example and compare the results to logit model.

Similar to Chapter 3, Chapter 4 describes the multinomial discrete choice probit model with product correlation structure. We derive analytical expressions for computing the choice probabilities with product correlation structure and describe the procedure of model estimation using ML approach. Further, we compare the performance of the probit model with product correlation structure to the paired combinatorial logit model, that is more appropriate when considering probit with product correlation. We compare the performance of both models using asymptotic relative efficiency. Finally, we illustrate the probit model with product correlation structure using a real data example and compare the results to paired combinatorial logit model.

Chapter 5 describes a unified approach to model the dependency between unobserved factors using copulas. We present the derivations that show that the logit models are special cases of the Extreme Value Discrete Choice Copula Models and the probit models are special cases of the Gaussian Copula Discrete Choice models with normal marginals. This insight lead to the possibility of developing new models to model the consumer choice behavior.

In Chapter 6, we present a brief summary of results obtained in this dissertation. Finally, the Appendix section contains important SAS and R programs we developed for this dissertation.

CHAPTER 2

LOGIT MODELS

2.1 INTRODUCTION

A popular and widely used discrete choice model is the Conditional Logit model. It is popular due to the fact that the choice probabilities in this model have closed form expressions and they are easily interpretable. Under this model, the unobserved utility factors are assumed to be independent and identically distributed as Gumbel, which is an extreme-value distribution. This independence assumption leads to an important property that the ratio of any two choice probabilities depends only on the two alternatives selected and all others become irrelevant. This property is known as "Independence of irrelevant alternatives (IIA)." While the IIA property is realistic in some choice situations, it may not be appropriate in others, see Chipman (1960) and Debreu (1960). Further, statistical tests developed by Hausman and McFadden (1984), McFadden (1987) and Train et al. (1989) are very useful to validate the IIA assumption. When IIA assumption is not tenable, one needs to pursue several alternate models that relax the IIA assumption. These alternative models such as nested logit, heteroscedastic extreme value (HEV) and mixed logit, allow different forms of dependency between alternatives. However, the model that allows most flexible dependence structure is the Multinomial Discrete Choice Probit model.

2.2 CONDITIONAL LOGIT MODEL

Luce (1959) derived the logit formula from assumptions of the characteristics of choice probabilities, namely IIA. As mentioned earlier, discrete choice models are based on utility maximization theory and Marschak (1960) showed that logit model is consistent with utility maximization. Later, Luce and Suppes (1965) showed that the assumption of unobserved utility following an extreme value distribution leads to logit formula. McFadden (1974) completed the proof by showing that the logit formula for the choice probabilities necessarily implies that unobserved utility has extreme value distribution. Hence, it is known as McFadden's conditional logit model.

Suppose we have n subjects and each subject faces c choices, among which one is chosen. Let y_{ij} be the binary response given by,

$$y_{ij} = \begin{cases} 1 & \text{if } i\text{th respondent chooses } j\text{th alternative} \\ 0 & \text{otherwise.} \end{cases}$$

We are interested in computing p_{ij}, the probability of ith subject choosing the jth alternative, $i = 1, \ldots, n$; $j = 1, \ldots, c$. A subject chooses the jth alternative if the latent utility of jth alternative is larger than utilities of all other alternatives. Let u_{ij} denote the latent utility that the ith subject associates with jth alternative and assume that $u_{ij} = \mu_{ij} + z_{ij}$, where μ_{ij} is the mean and z_{ij} is the error component. Further, we assume that $\mu_{ij} = \mathbf{x}'_{\mathbf{ij}}\boldsymbol{\beta}$, where $\mathbf{x}'_{\mathbf{ij}} = (x_{ij1}, \ldots, x_{ijp})$ is a p-variate vector of explanatory variables and $\boldsymbol{\beta}$ is the vector of unknown regression coefficients. Further, z_{ij}'s are independent and identically distributed (iid) as Type I extreme value (Gumbel) distribution with density

$$f(z_{ij}) = e^{-z_{ij}} \exp(-e^{-z_{ij}}), \quad -\infty \le z_{ij} \le \infty \tag{1}$$

and distribution function

$$F(z_{ij}) = \exp(-e^{-z_{ij}}). \tag{2}$$

Conditional on the choices, the model for the probability of selecting jth choice by ith respondent is,

$$p_{ij} = \frac{\exp(\mathbf{x}'_{\mathbf{ij}}\boldsymbol{\beta})}{\sum_{k=1}^{c} \exp(\mathbf{x}'_{\mathbf{ik}}\boldsymbol{\beta})}. \tag{3}$$

To prove Equation (3), we proceed as follows:

$$\begin{aligned}
p_{ij} &= Pr(y_{ij} = 1) \\
&= Pr(u_{ij} > u_{il};\ \forall\, l \ne j) \\
&= Pr(z_{il} < (\mu_{ij} - \mu_{il}) + z_{ij};\ \forall\, l \ne j) \\
&= \int_{-\infty}^{\infty} Pr\left(z_{il} < (\mu_{ij} - \mu_{il}) + z | z_{ij} = z;\ \forall\, l \ne j\right)\ f(z)\, dz \\
&= \int_{-\infty}^{\infty} \left(\prod_{l(\ne j)=1}^{c} \exp(-e^{-(z+\mu_{ij}-\mu_{il})}) \right) e^{-z}\ \exp(-e^{-z})\, dz \\
&= \int_{-\infty}^{\infty} \exp\left(-e^{-z} \sum_{l=1}^{c} e^{-(\mu_{ij}-\mu_{il})} \right) e^{-z}\, dz.
\end{aligned}$$

Let $v = e^{-z}$, then the choice probability p_{ij} given by

$$\begin{aligned} p_{ij} &= \int_0^\infty \exp\left(-v\sum_{l=1}^{c} e^{-(\mu_{ij}-\mu_{il})}\right) dv \\ &= \frac{1}{\sum_{l=1}^{c} e^{-(\mu_{ij}-\mu_{il})}} \\ &= \frac{e^{\mathbf{x}'_{ij}\boldsymbol{\beta}}}{\sum_{k=1}^{c} e^{\mathbf{x}'_{ik}\boldsymbol{\beta}}}. \end{aligned}$$

Note that, the advantage of choosing Gumbel distribution for the error terms results in a closed form expression of the choice probabilities and it is easily interpretable as it has logit form. Conditional on pair of choices j and k, this model can be written as

$$\log\left[p_{ij}/p_{ik}\right] = (\mathbf{x_{ij}} - \mathbf{x_{ik}})'\boldsymbol{\beta}. \tag{4}$$

2.2.1 DIFFERENCE BETWEEN MULTINOMIAL LOGIT AND CONDITIONAL LOGIT

It is worth noting the difference between a multinomial logit and conditional logit model. In a conditional logit model, the explanatory variables are the characteristics of choice alternatives such as price, cost, time, etc. and they vary over alternatives, sometimes also vary over subjects. In a regular multinomial logit model, the explanatory variables such as age, income, are characteristics of subject and remain constant across choices. In fact, the multinomial logit model is a special case of conditional logit model.

Consider a response variable with M nominal categories. The traditional base-line category multinomial logit model has $(M-1)$ logits given by

$$\log\frac{p_{ij}(\mathbf{x_i})}{p_{iM}(\mathbf{x_i})} = \alpha_j + \mathbf{x_i}'\boldsymbol{\beta}_j, \qquad j = 1, \ldots, (M-1),$$

where α_j are constants and $\boldsymbol{\beta}_j$ are vectors of regression coefficients. These $(M-1)$ equations are simplified to compute the response probabilities $p_j(\mathbf{x})$ as

$$p_j(\mathbf{x}) = \frac{\exp(\alpha_j + \mathbf{x}'\boldsymbol{\beta}_j)}{1 + \sum_{k=1}^{M-1}\exp(\alpha_k + \mathbf{x}'\boldsymbol{\beta}_k)},\ j = 1, \ldots, M-1 \tag{5}$$

with $\alpha_M = 0$ and $\beta_M = 0$, being the last category as base-line. A conditional logit model has the probabilities of the form (3) and the multinomial logit model has discrete-choice form after replacing an explanatory variable by M artificial variables; the jth is the product of the explanatory variable with a dummy variable that equals 1 when the response choice is j (Agresti 1990). For example, let x_i denote the value of ith subject, assuming a single explanatory variable, $i = 1\ldots,n$. For $j = 1,\ldots,M$, let δ_{jk} equal 1 when $k = j$ and 0 otherwise. Let $\mathbf{z}_{ij} = (\delta_{j1},\ldots,\delta_{jM},\delta_{j1}x_i,\ldots,\delta_{jM}x_i)'$ and $\boldsymbol{\beta} = (\alpha_1,\ldots,\alpha_M,\beta_1,\ldots,\beta_M)$. Then, $\mathbf{z}'_{ij}\boldsymbol{\beta} = \alpha_j + \beta_j x_i$, the response probabilities for the multinomial logit model (5) are

$$\begin{aligned} p_j(x_i) &= \frac{\exp(\alpha_j + x_i\beta_j)}{\exp(\alpha_1 + x_i\beta_1) + \cdots + \exp(\alpha_M + x_i\beta_J)} \\ &= \frac{\exp(\mathbf{z}'_{ij}\boldsymbol{\beta})}{\exp(\mathbf{z}'_{i1}\boldsymbol{\beta}) + \cdots + \exp(\mathbf{z}'_{iM}\boldsymbol{\beta})}, \end{aligned} \quad (6)$$

which are of the form (3). With this approach, the conditional logit model can contain characteristics of consumer as well as choices and thus multinomial logit model is a special case of conditional logit model.

This difference actually leads to an interesting formulation of "mean" using regression parameters. Three different model formulations are considered for doing market share analysis (Lee 1988) viz., simple effects, differential effects and cross-effects in increasing order of model complexity. Leaving the alternative specific intercepts $\alpha_1,\ldots\alpha_M$, simple effects model assumes same regression coefficient for each covariate across all alternatives. In other words, $\beta_j = \beta\ \forall j = 1,\ldots,c$. Note that a simple effects model requires less number of parameters to be estimated and less complex in nature. A differential effects model assumes regression coefficients to be specific to the alternative for each covariate. A differential effects model requires estimation of a large number of parameters than a simple effects model. A hypothesis test can be performed to test equality of regression coefficients to simplify the model. A more complex model can be obtained by building the cross-effects, which measures the impact of one alternative's covariate (for example, effect of a brand's price change on a competitor) on another alternative's covariate. Such a model requires estimation of a large number of parameters and thus requires a large sample size.

2.2.2 ESTIMATION PROCEDURE

The probabilities for the conditional logit model are in closed form and they can be easily calculated. The estimates of unknown regression coefficients are obtained using maximum likelihood approach. Further, McFadden (1974) demonstrated that the log-likelihood function with these choice probabilities is globally concave in parameters $\boldsymbol{\beta}$ and thus a solution can be obtained by solving score equations with regular optimization routines. The log-likelihood $\ell(\boldsymbol{\beta})$ for n subjects is

$$\ell(\boldsymbol{\beta}) = \log\left[\prod_{i=1}^{n}\prod_{j=1}^{c} p_{ij}^{y_{ij}}\right] = \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\log(p_{ij}).$$

The maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is the solution of score equations $\partial\ell(\beta)/\partial\beta = 0$. The expression for the first order partial derivative of the log-likelihood is

$$\begin{aligned}
\frac{\partial\ell(\beta)}{\partial\boldsymbol{\beta}} &= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\,\frac{\partial}{\partial\boldsymbol{\beta}}\left[\mathbf{x}_{ij}'\boldsymbol{\beta} - \log\left(\sum_{k=1}^{c}\exp(\mathbf{x}_{ik}'\boldsymbol{\beta})\right)\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left[\mathbf{x}_{ij} - \sum_{j=1}^{c}\left(\frac{\exp(\mathbf{x}_{ij}'\boldsymbol{\beta})}{\sum_{k=1}^{c}\exp(\mathbf{x}_{ik}'\boldsymbol{\beta})}\right)\mathbf{x}_{ij}\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\mathbf{x}_{ij} - \sum_{i=1}^{n}\left(\sum_{j=1}^{c} y_{ij}\right)\sum_{j=1}^{c} p_{ij}\mathbf{x}_{ij} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{c}(y_{ij} - p_{ij})\,\mathbf{x}_{ij}. \qquad (7)
\end{aligned}$$

The second order partial derivative of log-likelihood is

$$\begin{aligned}
\frac{\partial^2\ell(\theta)}{\partial\boldsymbol{\beta}^2} &= \frac{\partial}{\partial\boldsymbol{\beta}}\left[\sum_{i=1}^{n}\sum_{j=1}^{c}(y_{ij} - p_{ij})\,\mathbf{x}_{ij}\right] \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{c}\mathbf{x}_{ij}\,\frac{\partial p_{ij}}{\partial\boldsymbol{\beta}} \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{c} p_{ij}\,\mathbf{x}_{ij}\,\frac{\partial\log p_{ij}}{\partial\boldsymbol{\beta}} \\
&= -\sum_{i=1}^{n}\sum_{j=1}^{c} p_{ij}\,\mathbf{x}_{ij}\left(\mathbf{x}_{ij} - \sum_{k=1}^{c} p_{ik}\mathbf{x}_{ik}\right)'. \qquad (8)
\end{aligned}$$

No closed form solution for the score equation (7) is available and a solution is obtained using numerical optimization methods. We illustrate the conditional logit model using the following real data example.

Table 1. Sample data for the laundry detergent example

		Log-price					
Obs	Choice	Tide	Wisk	Era plus	Surf	Solo	All
1	Wisk	0.0606	0.0549	0.0587	0.0389	0.0556	0.0389
2	All	0.0584	0.0450	0.0645	0.0630	0.0645	0.0389
3	Wisk	0.0587	0.0467	0.0645	0.0645	0.0587	0.0389
4	EraPlus	0.0553	0.0488	0.0473	0.0566	0.0645	0.0405
5	Surf	0.0596	0.0498	0.0618	0.0420	0.0655	0.0413
6	Wisk	0.0702	0.0231	0.0702	0.0545	0.0623	0.0436
7	Tide	0.0480	0.0637	0.0559	0.0528	0.0637	0.0405
8	Solo	0.0516	0.0455	0.0606	0.0489	0.0492	0.0352
9	Solo	0.0655	0.0483	0.0567	0.0467	0.0545	0.0436
10	Wisk	0.0637	0.0263	0.0777	0.0693	0.0570	0.0410

2.2.3 ANALYSIS OF LAUNDRY DETERGENT DATA

In this example, the data is from a market research study and contains information about the brand bought, price of the laundry detergent purchased by 2657 consumers, originally analyzed by Chintagunta and Prasad (1998). The dataset contains the log prices of six detergent brands Tide, Wisk, EraPlus, Surf, Solo, and All, as well as the brand chosen by each household. Table 1 display a sample data of first 10 observations from laundry detergent data. Frequency counts of response variable "detergent choice" show the market share owned by each brand and they are given in Table 3. From this, we can observe that Tide and Wisk occupy about 53% of the market and they are the main competitors in the market. Price is one of the key explanatory variables of detergent brand choice, simple descriptive statistics of price are given in Table 2. This gives us basic understanding of the market and the brands price strategy. Further, Figure 1 plots the histogram of log-price for each brand. We can see that all brands are operating at one or two price points and all other price points occurring less frequent. This observation is useful to simulate a continuous covariate for discrete choice model.

We fit the conditional logit model with differential effects to identify the relationship between detergent choice and the log-price. Table 4 provides point estimates,

Table 2. Descriptive Statistics of Price

Brand	Mean	SD	Min.	Max.
Tide	0.0595	0.0074	0.0059	0.1250
Wisk	0.0472	0.0091	0.0007	0.1538
EraPlus	0.0606	0.0067	0.0259	0.1547
Surf	0.0529	0.0098	0.0031	0.1280
Solo	0.0599	0.0078	0.0305	0.1405
All	0.0391	0.0031	0.0216	0.1005

Table 3. Market Shares of laundry detergents

Brand	Frequency	Share (%)
Tide	701	26.4
Wisk	703	26.5
EraPlus	507	19.1
Surf	406	15.3
Solo	253	9.5
All	87	3.3
Total	2657	100.0

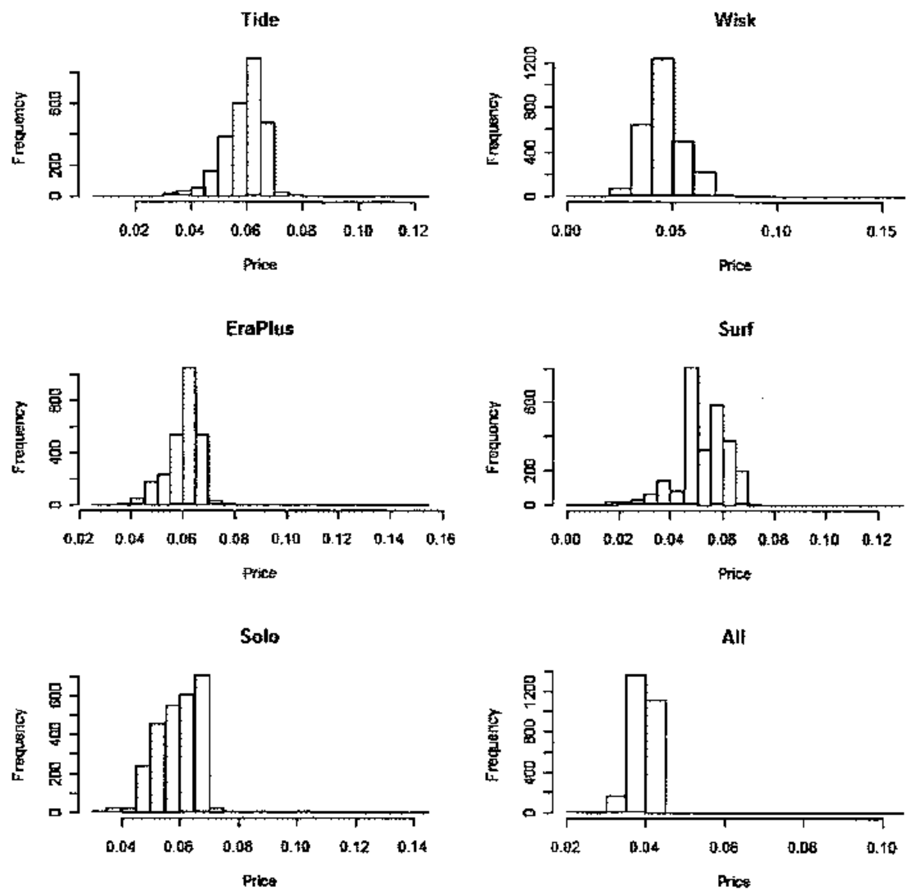

Figure 1. Histogram of log-price

standard errors and p-values for the conditional logit model. The first six coefficients correspond to the intercepts in relative to the last brand "All." They represent the relative preference to the last brand "All." Looking at the price coefficients for all brands, they are all negative intuitively correct signs. Wisk emerges to be a stable brand with lowest sensitivity to price. The last brand "All" is the most sensitive brand to price changes compared to all other brands in the market. Assuming the average prices, we compute the predicted market shares based on model (3) and compared to the actual shares, presented in Table 5. The results show that the conditional logit model fits the data well.

As a next step, this model can be very useful to study how the market reacts to price changes. As an example, suppose we decrease the price of EraPlus by 5% from its average price 0.0606 to 0.0576. This price change not only causes an increase in

Table 4. Conditional Logit ML estimates for the laundry detergents data

Parameter		β EST.	SE	p-value
Intercept	Tide	-3.3248	1.2128	0.0061
	Wisk	-7.5648	1.1574	<0.0001
	EraPlus	-4.2412	1.2114	0.0005
	Surf	-7.0918	1.1733	<0.0001
	Solo	-6.4532	1.2324	<0.0001
	All	0.0000		
log-price	Tide	-159.795	7.9440	<0.0001
	Wisk	-111.102	6.7383	<0.0001
	EraPlus	-146.993	8.0835	<0.0001
	Surf	-122.847	7.0350	<0.0001
	Solo	-124.055	9.2168	<0.0001
	All	-392.411	30.9367	<0.0001

its own market share but a decrease in share of other brands. The increase in market share is about 40.4%, drawn equally from other brands, as shown in Table 6. This is not realistic to the market dynamics that brands tend to draw more shares from their nearest competitors than the rest. This discrepancy is due to the assumption that the unobserved factors are independent and follow Gumbel distribution and this is shown mathematically in the next section.

2.2.4 IIA ASSUMPTION

From Equation (3), we can see that the choice probabilities in a conditional logit model are in a closed form. This is due to the assumption that the unobserved factors are independent and identically distributed as Gumbel. For any two alternatives j and k, the ratio of choice probabilities are of the form

$$\begin{aligned} \frac{p_{ij}}{p_{ik}} &= \frac{\exp(\mathbf{x}'_{ij}\boldsymbol{\beta})}{\exp(\mathbf{x}'_{ik}\boldsymbol{\beta})} \\ &= \exp((\mathbf{x}_{ij} - \mathbf{x}_{ik})'\boldsymbol{\beta}). \end{aligned} \quad (9)$$

Table 5. Actual versus Predicted share

Brand	Actual Share	Predicted Share
Tide	26.4	27.5
Wisk	26.5	28.0
EraPlus	19.1	19.9
Surf	15.3	12.8
Solo	9.5	9.6
All	3.3	2.3

Table 6. Impact of price changes on shares

Brand	Avg. Price	Original Share	New Share	Change(%)
Tide	0.0595	27.5	24.7	-10.0
Wisk	0.0472	28.0	25.2	-10.0
EraPlus	$0.0606 \rightarrow 0.0576$	19.9	27.9	40.4
Surf	0.0529	12.8	11.5	-10.0
Solo	0.0599	9.6	8.6	-10.0
All	0.0390	2.3	2.0	-10.0

Conditional on the choices j and k, a variable's influence only depends on the difference between values for those alternatives and all other alternatives become irrelevant. Luce (1959) called this property as Independence from Irrelevant Alternatives (IIA). It is unrealistic in some applications and hypothesis tests proposed by Hausman and McFadden (1984), McFadden (1987) and Train et al. (1989) are very useful to test IIA assumption. When IIA assumption is no longer valid, several alternate models that relax IIA assumption are applicable to choice situations.

2.2.5 TESTS OF IIA

Tests of IIA were first developed by McFadden (1978). Two types of tests are used to test IIA assumption, choice set partitioning tests and model-based tests. The choice set partitioning tests are based on whether the parameter estimates obtained on a subset of alternatives are significantly different from those obtained from full set. A test of the hypothesis that the parameters estimated on a subset are same as the parameters estimated on the full set constitutes a test of IIA. This was developed by Hausman and McFadden (1984). This test is based on likelihoods comparing the restricted model to the full model. A second test proposed by McFadden (1987) and Train et al. (1989) is based on model performance with inclusion of cross-alternative variables. If the ratio of two alternatives depends on a third alternative, the inclusion of attributes from a third alternative into the utility formulation of initial two alternatives become significant, then IIA does not hold and this constitutes a test of IIA. McFadden (1987) developed a procedure for performing this kind of a test and Train et al. (1989) show how this can be performed within the logit model. Model-based tests are those that test the validity of constraints imposed on a more general model such as nested logit or probit that lead to IIA. The disadvantage of this test is that it requires estimation of both models, often computationally difficult. If IIA assumption is not valid, we need to study alternate models and a review of those models is presented in the next section.

2.3 MODELS RELAXING IIA ASSUMPTION

In discrete choice models, IIA assumption plays an important role in computation of choice probabilities. In fact, the logit models with IIA assumption has proportional substitution pattern across alternatives. In other words, the ratio of any two choice probabilities are proportional to the two alternatives under consideration, as seen in (9). The class of models that exhibit a variety of substitution patterns including proportional substitution pattern are Generalized Extreme Value (GEV) models. In this class of models, we assume the unobserved factors follow a generalized extreme value distribution which allows correlation between alternatives. Thus, it relaxes IIA assumption and when the correlations are zero, the GEV model becomes the standard logit.

2.3.1 GEV MODELS

The most widely used GEV model is Nested logit model, in which alternatives are partitioned into subsets called "nests". For any two alternatives in the same nest, the ratio of choice probabilities is independent of other alternatives, so the IIA assumption holds within the nest. For any two alternatives from two different nests, the ratio of choice probabilities depend on attributes of other alternatives from those two nests, thus IIA does not hold between nests. This model is also consistent with utility maximization theory as shown by Daly and Zachary (1978), McFadden (1978), and Williams (1977).

2.3.2 NESTED LOGIT MODELS

Assume that the set of c alternatives are partitioned into g non-overlapping nests $N_1, \ldots, N_g$. The nested logit model is obtained by assuming that the alternatives within nests are correlated and the alternatives between nests are uncorrelated. Let c_k be the number of alternatives in the nest $N_k, k = 1, \ldots, g$. To impose dependency between alternatives in nest N_k, we assume that the unobserved factors for the ith subject $\mathbf{z_{ik}} = (z_{i1}, \ldots, z_{ic_k})$ follow a multivariate extreme value distribution with distribution function $F(\mathbf{z_{ik}}) = \exp\{-A(e^{-z_{i1}}, \ldots, e^{-z_{ic_k}})\}$, where $A(\mathbf{z}_{ik})$ is known as dependence function that governs the dependency between alternatives within nest $N_k, k = 1, \ldots, g$. McFadden (1978) proposed a dependency function of the form $A(w_1, \ldots, w_m) = (\sum_{r=1}^{m} w_r^{1/\lambda_k})^{\lambda_k}$ that lead to nested logit model, where λ_k denotes the degree of independence between alternatives in nest N_k with $0 < \lambda_k \leq 1, k = 1, \ldots, g$. Due to the independence assumption between nests, the unobserved factors

$\mathbf{z_i} = (\mathbf{z_{i1}}, \ldots, \mathbf{z_{ig}})$ follow an extreme value distribution with distribution function

$$\begin{aligned} F(\mathbf{z}_i) &= \prod_{k=1}^{g} \exp\left\{-\sum_{r=1}^{c_k} e^{-z_{ir}/\lambda_k}\right\}^{\lambda_k} \\ &= \exp\left(-\sum_{k=1}^{g}\left[\left\{\sum_{r=1}^{c_k} e^{-z_{ir}/\lambda_k}\right\}^{\lambda_k}\right]\right). \end{aligned} \tag{10}$$

This distribution function is one type of GEV distribution with the marginals z_{ir}, following univariate extreme value distribution, $r = 1, \ldots, c_k$. The statistic $1 - \lambda_k$ denotes the measure of dependence and a value of $\lambda_k = 1$ indicates no correlation, in which case it reduces to the standard logit model. The choice probability of ith respondent choosing jth alternative in a nested logit model is

$$p_{ij} = \frac{e^{\mu_{ij}/\lambda_k}\left[\sum_{l=1}^{c_k}\exp(\mu_{il}/\lambda_k)\right]^{\lambda_k - 1}}{\sum_{k=1}^{g}\left[\sum_{l=1}^{c_k}\exp(\mu_{il}/\lambda_k)\right]^{\lambda_k}}. \tag{11}$$

Using the expression (11), the ratio of choice probabilities for alternatives j and j' are

$$\frac{p_{ij}}{p_{ij'}} = \frac{e^{\mu_{ij}/\lambda_k}\left[\sum_{l=1}^{c_k}\exp(\mu_{il}/\lambda_k)\right]^{\lambda_k-1}}{e^{\mu_{ij'}/\lambda_{k'}}\left[\sum_{l=1}^{c_{k'}}\exp(\mu_{il}/\lambda_{k'})\right]^{\lambda_{k'}-1}}. \tag{12}$$

If j and j' are from the same nests ($k = k'$), the term in parenthesis cancel out and lead to IIA assumption within nest. If j and j' are from different nests ($k \neq k'$), the term in parenthesis do not cancel out and IIA assumption does not hold between nests. This property often rephrased as "Independence from irrelevant Nests (IIN)" and it is not as restrictive as IIA property.

The value of λ_k must be within a particular range for the model to be consistent with utility-maximizing behavior. If $0 < \lambda_k < 1, \forall k = 1, \ldots, g$, then the model is consistent with utility maximization for all possible values of the explanatory variables (Train 2004). For λ_k greater than one, the model is consistent with utility-maximizing behavior for some range of the explanatory variables but not for all values. Kling and Herriges (1995) and Herriges and Kling (1996) provide tests of consistency of nested logit with utility maximization when $\lambda_k > 1$; and Train et al. (1987a) and Lee (1999) provide examples of models for which $\lambda_k > 1$. A negative value of λ_k is inconsistent with utility maximization. It means that an estimated k outside the (0, 1] bounds suggests a misspecification problem with the model and requires reexamination of the specification. The estimation of a nested logit model is similar to that of a conditional logit model using maximum likelihood estimation. Other estimation methods exist but they are not of relevance in our discussion.

Other dependency functions can be constructed based on multivariate extreme value copulas that must satisfy some conditions such as min-stable multivariate exponential (MSMVE). A detailed discussion of multivariate extreme value (MEV) copulas, properties of MEV distributions and a method to construct MEV distributions with several dependency functions is presented in Chapter 5.

The nested logit models discussed above are known as two-level nested logit models. One can create three or higher level nested logit models by partitioning the set of alternatives into nests and then into subnests. The choice probabilities of these models are generalization of (11) and exhibit the similar variations of IIA assumption within nests and between nests.

So far, we have considered the nests that are non-overlapping and relaxing such an assumption would lead to several types of other GEV models. Vovsha (1997), Bierlaire (1998), and Ben-Akiva and Bierlaire (1999) have proposed models that are called as cross-nested logits (CNLs) which contain multiple overlapping nests. Another model proposed by Chu (1989) is the Paired Combinatorial Logit (PCL), in which each pair of alternatives constitutes a nest. Wen and Koppelman (2001) have developed a generalized nested logit (GNL) model that includes the PCL and other cross-nested models as special cases. A brief discussion of the PCL and GNL models are given in the following sections.

2.3.3 PAIRED COMBINATORIAL LOGIT

As the name suggests, each pair of alternatives are treated as a nest in this model and each alternative is a member of $c-1$ nests. Similar to nested logit model, we assume a parameter λ_{jk} that indicates the degree of independence between alternatives j and k. This model becomes the standard logit model when λ_{jk} equal to 1 for all $1 \leq j,k \leq c$. The choice probability of ith respondent choosing jth alternative is

$$p_{ij} = \frac{\sum_{j \neq k} e^{\mu_{ij}/\lambda_{jk}} \left[e^{\mu_{ij}/\lambda_{jk}} + e^{\mu_{ik}/\lambda_{jk}}\right]^{\lambda_{jk}-1}}{\sum_{l=1}^{c-1} \sum_{l'=l+1}^{c} \left[e^{\mu_{il}/\lambda_{ll'}} + e^{\mu_{il'}/\lambda_{ll'}}\right]^{\lambda_{ll'}}}. \tag{13}$$

The numerator in (13) is evaluation over $c-1$ nests in which the jth alternative paired with other alternatives, similar to the choice probabilities in a nested logit model. If λ_{jk} is between 0 and 1 for all pairs, this model is consistent with utility maximization theory and it is easy to see the model becomes standard logit when $\lambda_{jk}=1$. Koppelman and Wen (2000) found PCL to perform better than nested logit or standard logit. We will return to this model form in Chapter 4 while developing a probit model with product correlation structure.

2.3.4 GENERALIZED NESTED LOGIT (GNL)

A generalized nested logit model is an overlapping nested model with varying levels overlapping of alternatives among nests. In other words, an alternative can be part of several nests with more preference given to some nests than other nests. This is characterized by an allocation parameter $\alpha_{jm}, 1 \le j \le c; 1 \le m \le g$, which is nonnegative and $\sum_{m=1}^{g} \alpha_{jm} = 1$. The parameter α_{jm} represents the portion of alternative allocated to mth nest. Similar to nested logit model, a parameter λ_m is defined to measure the degree of independence within nest m. The choice probability of ith respondent choosing the jth alternative is

$$p_{ij} = \frac{\sum_{m=1}^{g} (\alpha_{jm} e^{\mu_{ij}})^{1/\lambda_m} \left[\sum_{l=1}^{c_m} (\alpha_{lm} e^{\mu_{il}})^{1/\lambda_m}\right]^{\lambda_m - 1}}{\sum_{m=1}^{g} \left[\sum_{l=1}^{c_m} (\alpha_{lm} e^{\mu_{il}})^{1/\lambda_m}\right]^{\lambda_m}}. \tag{14}$$

This formula is similar to (11) except that the numerator is sum over all the nests that contain jth alternative, with respective weights $\alpha_{jm}, m = 1, \ldots, g$. If each alternative belong to only one nest, then the model becomes nested logit. In addition, if the nest independence parameter λ_m is equal to one, then it reduces to standard logit model. Wen and Koppelman (2001) derive various cross-nested models as special cases of the GNL. Including these models, McFadden (1978) developed a process to generate GEV models (see Train 2004), with which new formulations of GEV models can be developed that best fit the specific circumstances of a particular choice situation, discussed in Chapter 4 to generate GEV models.

2.3.5 HETEROSCEDASTIC LOGIT MODEL

Another way to relax IIA assumption is to allow the variance of unobserved factors vary across alternatives. Such a model is known as "Heteroscedastic Extreme Value (HEV)" model, first described by Steckel and Vanhonacker (1988), Bhat (1995), and

Recker (1995) . In this model, we assume z_{ij} follow an extreme value distribution with distribution function

$$F(z_{ij}) = \exp\left[-\exp\left(\frac{z_{ij}}{\theta_j}\right)\right], \tag{15}$$

where θ_j is the scale parameter for the jth alternative. With this formulation, the choice probability of selecting jth alternative by ith respondent (Bhat, 1995) is

$$p_{ij} = \int \left[\prod_{k \neq j} \exp(-e^{-(\mu_{ij}-\mu_{ik}+\theta_j v)/\theta_k})\right] \exp(-e^{-v})e^{-v}dv. \tag{16}$$

This expression does not have a closed form and often evaluated through simulations. Further, Bhat (1995) showed that the heteroscedastic logit probabilities can be calculated effectively with quadrature rather than simulation.

2.4 LIMITATIONS OF LOGIT

The goal of modeling consumer choice behavior is to identify models that are able to incorporate the effects of taste variation, allow different substitution patterns across alternatives and model repeated response over time. The first one, taste variation refers to the differences in response due to differences in respondent tastes and their behaviors. This can come from systematic variation that relates to the observed characteristics of respondent such as age, income, etc. and random taste variation that cannot be linked to consumer characteristics. Second, substitution patterns refers to the way the alternatives are correlated, such as proportional substitution. Third, repeated response refer to the choices made over time or responses to several choice cards by the same respondent.

The conditional logit model based on the assumption that unobserved factors are independent and follow an extreme value distribution is restrictive with IIA assumption, also known as proportional substitution. Even though GEV models alleviate this restriction of proportional substitution by allowing different substitution patterns, they are limited to incorporate random taste variation and correlated response over time. Discrete choice probit models are the most flexible models that incorporate random taste variation, any substitution pattern and include repeated choices.

Probit models are derived under the assumption that the unobserved factors are jointly distributed as multivariate normal with a unknown correlation structure among alternatives. The only limitation of probit models is that the computation of choice probabilities require difficult evaluation of multivariate normal distribution function. But, we can derive the exact analytical expressions for choice probabilities with correlation structures such as equicorrelation, product correlation etc. Chapter 3 and Chapter 4 present a detailed discussion of derivation of choice probabilities for equicorrelation structure, product correlation structure using stochastic representations and compare the performance of probit models with logit models. Chapter 5 presents a unified way of handling the correlated repeated choice data using copulas.

CHAPTER 3

DISCRETE CHOICE PROBIT MODELS

3.1 INTRODUCTION

The multinomial discrete choice probit model is derived under the assumption of multivariate normal unobserved utility components. Thurstone (1927) derived the formula for a binary probit, and Hausman and Wise (1978) and Daganzo (1979) extended the generality of the specification for representing various aspects of choice behavior.

Similar to the assumptions of conditional logit model, let $u_{ij} = \mu_{ij} + z_{ij}$ and $\mu_{ij} = \mathbf{x}'_{ij}\boldsymbol{\beta}$, assuming the same beta coefficients for all alternatives, i.e., $\boldsymbol{\beta}_j = \boldsymbol{\beta}$. Instead of assuming the random components z_{ij}'s are iid Gumbel, let $\mathbf{z}_i = (z_{i1}, \ldots, z_{ic})$ follow a Multivariate Normal (MVN) Distribution with mean $\mathbf{0}$ and correlation structure $\mathbf{R}$. The density of $\mathbf{z}_i$ is given by

$$\phi_c(\mathbf{z}_i; \mathbf{0}, \mathbf{R}) = \frac{1}{(2\pi)^{\frac{c}{2}}|\mathbf{R}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{z}'_i\mathbf{R}^{-1}\mathbf{z}_i\right). \tag{17}$$

Under these assumptions, the model for the probability of selecting jth choice by ith respondent is,

$$\begin{aligned} p_{ij} &= Pr\left(\mu_{ij} + z_{ij} > \mu_{il} + z_{il}\ ;\ \forall\, l \neq j\right) \\ &= \int I(\mu_{ij} + z_{ij} > \mu_{il} + z_{il}\ ;\ \forall\, l \neq j)\ \phi_c(\mathbf{z}_i; \mathbf{0}, \mathbf{R})\ d\mathbf{z}_i, \end{aligned} \tag{18}$$

where $I(.)$ is the indicator function for the condition in parenthesis to hold and the integral is over all values of $\mathbf{z}_i$. This multidimensional integral does not have a closed form and is often evaluated using numerical simulations. This is one of the main restriction in application of probit models in choice situations despite their ability of incorporating various of choice behaviors. Numerous simulators such as "accept-reject", "smoothed accept-reject", and GHK have been proposed for evaluation of probit choice probabilities (Hajivassiliou, McFadden and Ruud 1996). The GHK simulator is the most widely used probit simulator compared to other simulators. Along with simulators, algorithms developed by Genz (1992) based on Cholesky decomposition and a series of transformations compute multivariate normal probabilities to the best possible level of accuracy. The computation of choice probabilities become less difficult in the cases of correlation structures that represent specific substitution patterns between alternatives. For example, Yai, Iwakura and Morichi (1997) estimate a probit model of route choices where the covariance between any two routes depends only on the length of shared route segments; this structure reduces the number of covariance parameters to only one, which captures the relation of the covariance to shared length. Bolduc, Fortin, and Fournier (1996) estimate a model of physicians choice of location where the covariance among locations is a function of their proximity to one another, using a "generalized autoregressive errors" as in Bolduc (1992). Haaijer, Wedel, Vriens and Wansbeek (1998) impose a factor-analytic structure that arises from random coefficients of explanatory variables; Elrod and Keane (1995) impose a factor-analytic structure, that arises from error components.

In this work, we present simplification of probit models for simple structures such as equicorrelation and product correlation using stochastic representations. Later, we present simplification of a much general dependency structure using multivariate copulas and obtain these as special cases. In the following sections, we derive the exact analytical expressions for computation of choice probabilities under equicorrelation structure and present the maximum likelihood method of estimating probit model. We also derive the analytical expressions for the Fisher information matrix to compute standard errors of parameter estimates.

3.1.1 SIMILARITIES TO THE MULTIVARIATE PROBIT MODEL

The Multinomial Discrete Choice Probit (MDCP) model is similar to the multivariate binary probit model with some differences in the ranges of marginals. The response variable in a discrete choice model, even though univariate, can be regarded as multivariate binary random variable and it can be shown that it is similar to a multivariate binary probit model, with choice alternatives treated as repeated measurements.

Suppose we have m variate random variable $\mathbf{Y} = (Y_1, \ldots, Y_m)$ where each Y_j is a repeated response of a binary outcome. In a multivariate probit model, we assume that there exists a latent random variable $\mathbf{U} = (U_1, \ldots, U_m)$ that follows multivariate normal with mean $\boldsymbol{\mu}$ and correlation $\mathbf{R}$ such that

$$Y_j = \begin{cases} 1 & \text{if } U_j \le \mu_j \\ 0 & \text{otherwise,} \end{cases}$$

where μ_j's are constants. Then the probability of $\mathbf{Y} = \mathbf{y}$ can be obtained as

$$Pr(\mathbf{Y} = \mathbf{y}) \;=\; \int_{l_1}^{u_1} \dots \int_{l_m}^{u_m} \phi_m(\mathbf{U}; \mu, \mathbf{R})\, d\mathbf{U} \tag{19}$$

where

$$\begin{cases} l_j = -\infty, u_j = \mu_j & \text{if } y_j = 1 \\ l_j = \mu_j, u_j = \infty & \text{otherwise.} \end{cases}$$

Note that, there are 2^m possible values of the response variable $\mathbf{Y}$ to which the probability adds to 1.

In a discrete choice probit model with m alternatives, the response is a m variate binary response vector with the restriction that only one of them can be equal to 1 and rest are all zero. Therefore the number of possible values of the response variable are m and the total probability adds to 1 of these m possibilities. Further, the choice probabilities can be described in terms of latent variable known as "utility", similar to multivariate probit model. As mentioned before, the discrete choice model is based on utility maximization theory, in which a respondent assigns a utility value to each alternative, that's not observed. The discrete choice probit model is obtained by assuming the latent utility $\mathbf{U}$ follows a multivariate normal with mean $\boldsymbol{\mu}$ and correlation $\mathbf{R}$ such that

$$Y_j = \begin{cases} 1 & \text{if } U_j > U_l; \quad \forall l \ne j \\ 0 & \text{otherwise.} \end{cases}$$

Then the probability of jth alternative picked up by a respondent, $Pr(Y_j = 1)$, is the joint probability $Pr(U_j > U_l; \forall l \ne j)$ and it can be evaluated as

$$Pr(Y_j = 1) = \int_{\mu_1 - \mu_j}^{\infty} \dots \int_{\mu_m - \mu_j}^{\infty} \phi_{m-1}(\mathbf{w}; \boldsymbol{\mu}^*, \mathbf{R}^*)\, d\mathbf{w}, \tag{20}$$

where $\mathbf{W} = (U_1 - U_j, \dots, U_m - U_j)$ is a $m - 1$ multivariate normal with mean $\boldsymbol{\mu}^* = (\mu_1 - \mu_j, \dots, \mu_m - \mu_j)$ and correlation structure $\mathbf{R}^* = \mathbf{CRC}'$ with

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & \dots & -1 & \dots & 0 \\ 0 & 1 & \dots & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 & \dots & 1 \end{pmatrix}.$$

The expressions (19) and (20) are similar except the ranges of integration are different and are of different dimensions. The discrete choice probit model become more complex when we introduce repeated measurements into the model formulation and it requires to accommodate two types of dependencies, one between alternatives and other between repeated measures.

3.1.2 PROBIT MODEL WITH EQUICORRELATION STRUCTURE

Let $\mathbf{R} = (1-\rho)\mathbf{I} + \rho\mathbf{J}$, where ρ is the assumed correlation between any two alternatives. Assuming such a structure eliminates the identification problem (Train 2004) of choice models. However, for R to be positive definite, ρ should satisfy $-1/(c-1) < \rho < 1$. Under the assumption that unobserved factors $\mathbf{z}_i$ follows a multivariate normal with mean $\mathbf{0}$ and correlation structure $\mathbf{R}$, the choice probability of ith subject choosing jth alternative is

$$\begin{aligned} p_{ij} &= Pr\left(\mu_{ij} + z_{ij} > \mu_{il} + z_{il}\right) \\ &= Pr\left(z_{ij} - z_{il} > \mu_{il} - \mu_{ij}\right) \\ &= Pr\left(\frac{z_{ij} - z_{il}}{\sqrt{2(1-\rho)}} > \frac{\mu_{il} - \mu_{ij}}{\sqrt{2(1-\rho)}}\right) \\ &= Pr\left(w_{il} > \mu_{il}^*; \quad \forall\, l(\neq j) = 1 \ldots c\right), \end{aligned}$$

where

$$\begin{aligned} w_{il} &= \frac{(z_{ij} - z_{il})}{\sqrt{2(1-\rho)}} \\ \text{and } \mu_{il}^* &= \frac{(\mu_{il} - \mu_{ij})}{\sqrt{2(1-\rho)}}. \end{aligned}$$

Note that, $\mathrm{E}(w_{il}) = 0; \mathrm{Var}(w_{il}) = 1;$ and $\mathrm{Cov}(w_{il}, w_{il'}) = \frac{1}{2}$. Hence,

$$\begin{aligned} p_{ij} &= Pr\left(w_{il} > \mu_{il}^*\,; \forall l \neq j\right) \\ &= \int_{\mu_{i1}^*}^{\infty} \cdots \int_{\mu_{ij-1}^*}^{\infty} \int_{\mu_{ij+1}^*}^{\infty} \cdots \int_{\mu_{ic}^*}^{\infty} \phi_{c-1}(\mathbf{w}_i; \mathbf{0}, \mathbf{R}^*)\, d\mathbf{w}_i, \end{aligned} \tag{21}$$

where $\mathbf{w}_i = (w_{i1}, \ldots, w_{ij-1}, w_{ij+1}, \ldots w_{ic})'$, $\mathbf{R}^* = \frac{1}{2}\mathbf{I} + (1-\frac{1}{2})\mathbf{J}$ and $\phi_{c-1}(\mathbf{w}_i; \mathbf{0}, \mathbf{R}^*)$ is the probability density function of multivariate normal distribution of dimensionality $c-1$. Though the dimension of integral reduced from c to $c-1$, we still need to evaluate multivariate normal integrals to calculate the choice probabilities. However, a simple transformation known as "stochastic representation" will reduce this task to computation of a univariate integral.

3.1.3 THE STOCHASTIC REPRESENTATIONS

Suppose $X_1, \ldots, X_c$ are jointly distributed as multivariate normal with correlation structure $\mathbf{R}$. Then the random variables $(\sqrt{1-\rho}V_1 + \sqrt{\rho}V_0, \ldots, \sqrt{1-\rho}V_c + \sqrt{\rho}V_0)$ follows multivariate normal with mean $\mathbf{0}$ and correlation structure $\mathbf{R}$, where $V_0, V_1, \ldots, V_c$ are $c+1$ i.i.d $N(0,1)$ random variables. (Tong 1990 Theorem 5.3.9). The representation $X_j = \sqrt{1-\rho}V_j + \sqrt{\rho}V_0$ is known as "stochastic representation" of multivariate normal random variables.

Therefore, for the new correlation structure R^*, let $w_{il} = (v_{i0} + v_{il})/\sqrt{2}$, where $v_0, v_{i1}, \ldots, v_{ic}$ are $c+1$ independent standard normal random variables. Note that, $\mathrm{E}(w_{il}) = 0$, $\mathrm{Var}(w_{il}) = 1$ and $\mathrm{Cov}(w_{il}, w_{il'}) = \frac{1}{2}$. Hence, (21) simplifies to

$$\begin{aligned} p_{ij} &= Pr(w_{il} > \mu^*_{il} \,; \forall l \neq j) \\ &= Pr\left(\frac{1}{\sqrt{2}}(v_{i0} + v_{il}) > \mu^*_{il} \,; \forall l \neq j\right) \\ &= \int_{-\infty}^{\infty} \left[Pr(v_{il} > \sqrt{2}\mu^*_{il} - v | v \,; \forall l \neq j\right] \phi(v)\, dv \\ &= \int_{-\infty}^{\infty} \prod_{l(\neq j)=1}^{c} \left[1 - \Phi\left(\frac{\mu_{il} - \mu_{ij}}{(1-\rho)} - v\right)\right] \phi(v)\, dv \\ &= \int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \prod_{l=1}^{c} \left[\Phi\left(v - \frac{(\mathbf{x_{il}} - \mathbf{x_{ij}})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right)\right] \phi(v)\, dv. \end{aligned} \tag{22}$$

The expression (22) can be computed easily using built-in functions of popular software like SAS and R. After obtaining the choice probabilities for the multinomial discrete choice probit model, we obtain the regression parameter estimates using maximum likelihood approach as outlined in the next section.

3.2 MAXIMUM LIKELIHOOD ESTIMATION

Similar to the logit model, we assume the means $\mu_{ij}, j = 1 \ldots c$ are linear functions of $\mathbf{x}'_{ij}\boldsymbol{\beta}$ and our goal is to estimate the unknown parameter $\boldsymbol{\theta} = (\boldsymbol{\beta}, \rho)$ using the maximum likelihood estimation method. The log-likelihood $\ell(\boldsymbol{\theta})$ for n subjects is

$$\ell(\boldsymbol{\theta}) = \log\left[\prod_{i=1}^{n}\prod_{j=1}^{c} p_{ij}^{y_{ij}}\right] = \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\log(p_{ij}).$$

The maximum likelihood estimate of $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\beta}}, \widehat{\rho})$ of $\boldsymbol{\theta}$ is the solution of likelihood equations $\partial\ell(\theta)/\partial\theta = 0$. The expressions for the first order and second order partial derivatives of the log-likelihood are:

$$\begin{aligned}\frac{\partial\ell(\theta)}{\partial\boldsymbol{\theta}} &= \left[\frac{\partial\ell(\theta)}{\partial\boldsymbol{\beta}}\ \frac{\partial\ell(\theta)}{\partial\rho}\right]\\ &= \left[\frac{\partial\ell(\theta)}{\partial\beta_0},\ldots,\frac{\partial\ell(\theta)}{\partial\beta_p},\ \frac{\partial\ell(\theta)}{\partial\rho}\right],\end{aligned}$$

with the first order partial derivatives are given by

$$\begin{aligned}\frac{\partial\ell(\theta)}{\partial\beta_m} &= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\beta_m}\right)\\ \text{and } \frac{\partial\ell(\theta)}{\partial\rho} &= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\rho}\right).\end{aligned}$$

The Hessian matrix is

$$\frac{\partial^2\ell(\theta)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'} = \begin{pmatrix} \frac{\partial^2\ell(\theta)}{\partial\beta_0^2} & \frac{\partial^2\ell(\theta)}{\partial\beta_0\partial\beta_1} & \cdots & \frac{\partial^2\ell(\theta)}{\partial\beta_0\partial\beta_p} & \frac{\partial^2\ell(\theta)}{\partial\beta_0\partial\rho}\\ \frac{\partial^2\ell(\theta)}{\partial\beta_1\partial\beta_0} & \frac{\partial^2\ell(\theta)}{\partial\beta_1^2} & \cdots & \frac{\partial^2\ell(\theta)}{\partial\beta_1\partial\beta_p} & \frac{\partial^2\ell(\theta)}{\partial\beta_1\partial\rho}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \frac{\partial^2\ell(\theta)}{\partial\beta_p\partial\beta_0} & \frac{\partial^2\ell(\theta)}{\partial\beta_p\partial\beta_1} & \cdots & \frac{\partial^2\ell(\theta)}{\partial\beta_p^2} & \frac{\partial^2\ell(\theta)}{\partial\beta_p\partial\rho}\\ \frac{\partial^2\ell(\theta)}{\partial\rho\partial\beta_0} & \frac{\partial^2\ell(\theta)}{\partial\rho\partial\beta_1} & \cdots & \frac{\partial^2\ell(\theta)}{\partial\rho\partial\beta_p} & \frac{\partial^2\ell(\theta)}{\partial\rho^2}\end{pmatrix},$$

where the second order partial derivatives are given by

$$\begin{aligned}\frac{\partial^2\ell(\theta)}{\partial\beta_m^2} &= \frac{\partial}{\partial\beta_m}\left[\sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\beta_m}\right)\right]\\ &= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial^2 p_{ij}}{\partial\beta_m^2}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\beta_m}\right)^2,\end{aligned}$$

$$\begin{aligned}\frac{\partial^2\ell(\theta)}{\partial\rho^2} &= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial^2 p_{ij}}{\partial\rho^2}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\rho}\right)^2,\\ \frac{\partial^2\ell(\theta)}{\partial\beta_m\partial\rho} &= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial^2 p_{ij}}{\partial\beta_m\partial\rho}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\beta_m}\right)\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\rho}\right),\\ \frac{\partial^2\ell(\theta)}{\partial\beta_{m'}\partial\beta_m} &= \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial^2 p_{ij}}{\partial\beta_m\partial\beta_{m'}}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\beta_m}\right)\left(\frac{1}{p_{ij}}\frac{\partial p_{ij}}{\partial\beta_{m'}}\right),\end{aligned}$$

where $m(\neq m') = 0, 1, \ldots, p$ is the number of covariates. All the first order and second order parital derivatives involve evaluation of the following six terms

$$\frac{\partial \ell(\theta)}{\partial \beta_m}, \frac{\partial \ell(\theta)}{\partial \rho}, \frac{\partial^2 \ell(\theta)}{\partial \beta_m^2}, \frac{\partial^2 \ell(\theta)}{\partial \rho^2}, \frac{\partial^2 \ell(\theta)}{\partial \beta_{m'} \partial \beta_m}, \frac{\partial^2 \ell(\theta)}{\partial \beta_m \partial \rho},$$

and their analytical expressions are derived in the following section.

3.2.1 PARTIAL DERIVATIVES

Let $\boldsymbol{\theta} = (\boldsymbol{\beta}, \rho)$, $A_l(\boldsymbol{\theta}, v) = \Phi\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right)$ and $a_l(\boldsymbol{\theta}, v) = \phi\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right)$. Then the derivatives of $A_l(\boldsymbol{\theta}, v)$, $a_l(\boldsymbol{\theta}, v)$ with respect to $\beta_m, m = 0, \ldots, p; \rho$ are

$$\begin{aligned}
\frac{\partial}{\partial \beta_m} A_l(\boldsymbol{\theta}, v) &= -\phi\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right) \frac{(x_{ilm} - x_{ijm})}{\sqrt{1-\rho}} \\
&= -a_l(\boldsymbol{\theta}, v) \frac{d_{ilm}}{\sqrt{1-\rho}}, \\
\frac{\partial}{\partial \rho} A_l(\boldsymbol{\theta}, v) &= -\phi\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right) \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{2(1-\rho)^{3/2}} \\
&= -a_l(\boldsymbol{\theta}, v) \frac{\mathbf{d}_{il}'\boldsymbol{\beta}}{2(1-\rho)^{3/2}}, \\
\frac{\partial}{\partial \beta_m} a_l(\boldsymbol{\theta}, v) &= \phi\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right)\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right) \frac{(x_{ilm} - x_{ijm})}{\sqrt{1-\rho}} \\
&= a_l(\boldsymbol{\theta}, v)\left(v - \frac{\mathbf{d}_{il}'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right) \frac{d_{ilm}}{\sqrt{1-\rho}}, \\
\frac{\partial}{\partial \rho} a_l(\boldsymbol{\theta}, v) &= \phi\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right)\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right) \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{2(1-\rho)^{3/2}} \\
&= a_l(\boldsymbol{\theta}, v)\left(v - \frac{\mathbf{d}_{il}'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right) \frac{\mathbf{d}_{il}'\boldsymbol{\beta}}{2(1-\rho)^{3/2}},
\end{aligned}$$

where $d_{ilm} = (x_{ilm} - x_{ijm})$ and $\mathbf{d}_{\mathbf{il}}'\boldsymbol{\beta} = (\mathbf{x}_{\mathbf{il}} - \mathbf{x}_{\mathbf{ij}})'\boldsymbol{\beta}$. Hence,

$$\begin{aligned}
\frac{\partial p_{ij}}{\partial \beta_m} &= \frac{\partial}{\partial \beta_m}\left[\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \prod_{l=1}^{c} \Phi\left(v - \frac{(\mathbf{x}_{il} - \mathbf{x}_{ij})'\boldsymbol{\beta}}{\sqrt{(1-\rho)}}\right) \phi(v)\, dv\right] \\
&= \int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left(\prod_{l(\neq k)=1}^{c} A_l(\boldsymbol{\theta}, v) \frac{\partial}{\partial \beta_m}\left(A_k(\boldsymbol{\theta}, v)\right)\right) \phi(v) dv \\
&= -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left(\prod_{l(\neq k)=1}^{c} A_l(\boldsymbol{\theta}, v) a_k(\boldsymbol{\theta}, v) \frac{d_{ikm}}{\sqrt{(1-\rho)}}\right) \phi(v) dv, \qquad (23)
\end{aligned}$$

$$\frac{\partial p_{ij}}{\partial \rho} = -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left(\prod_{l(\neq k)=1}^{c} A_l(\boldsymbol{\theta}, v) a_k(\boldsymbol{\theta}, v) \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}} \right) \phi(v) dv. \quad (24)$$

Further, the second order partial derivative of $\ell(\theta)$ w.r.t β_m is given as

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial \beta_m^2} &= \frac{\partial}{\partial \beta_m} \left[-\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left(\prod_{l(\neq k)=1}^{c} A_l(\boldsymbol{\theta}, v) a_k(\boldsymbol{\theta}, v) \frac{d_{ikm}}{\sqrt{(1-\rho)}} \right) \phi(v) dv \right] \\
&= -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \frac{d_{ikm}}{\sqrt{(1-\rho)}} \frac{\partial}{\partial \beta_m} \left[\prod_{l=1}^{c} A_l(\boldsymbol{\theta}, v) \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \right] \phi(v) dv \\
&= -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \frac{d_{ikm}}{\sqrt{(1-\rho)}} \left[-\frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \sum_{k'=1}^{c} \left(\prod_{l(\neq k')=1}^{c} A_l(\boldsymbol{\theta}, v) \frac{a_k(\boldsymbol{\theta}, v) d_{ik'm}}{\sqrt{(1-\rho)}} \right) \right. \\
&\quad + \prod_{l=1}^{c} A_l(\boldsymbol{\theta}, v) \left\{ \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \left(v - \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{\sqrt{1-\rho}} \right) \frac{d_{ikm}}{\sqrt{(1-\rho)}} \right. \\
&\quad \left. \left. + \left(\frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \right)^2 \frac{d_{ikm}}{\sqrt{(1-\rho)}} \right\} \right] \phi(v) dv.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial \beta_m^2} &= -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left(\prod_{l=1}^{c} A_l(\boldsymbol{\theta}, v) \right) \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \frac{d_{ikm}}{(1-\rho)} \left[-\sum_{k'=1}^{c} \frac{a_{k'}(\boldsymbol{\theta}, v)}{A_{k'}(\boldsymbol{\theta}, v)} d_{ik'm} \right. \\
&\quad \left. + \left(v - \frac{\mathbf{d}'_{\mathrm{ik}}\boldsymbol{\beta}}{\sqrt{1-\rho}} \right) d_{ikm} + \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} d_{ikm} \right] \phi(v) dv.
\end{aligned} \quad (25)$$

Similarly,

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial \beta_{m'} \beta_m} &= -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left(\prod_{l=1}^{c} A_l(\boldsymbol{\theta}, v) \right) \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \frac{d_{ikm}}{(1-\rho)} \left[-\sum_{k'=1}^{c} \frac{a_{k'}(\boldsymbol{\theta}, v)}{A_{k'}(\boldsymbol{\theta}, v)} d_{ik'm'} \right. \\
&\quad \left. + \left(v - \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{\sqrt{1-\rho}} \right) d_{ikm'} + \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} d_{ikm'} \right] \phi(v) dv
\end{aligned} \quad (26)$$

with $m \neq m' = 0, \ldots, p$. Next, the second order partial derivative of $\ell(\theta)$ w.r.t ρ and

β_m is

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\rho\partial\beta_m} &= \frac{\partial}{\partial\rho}\left[-\int_{-\infty}^{\infty}\frac{1}{\Phi(v)}\sum_{k=1}^{c}\left(\prod_{l(\neq k)=1}^{c} A_l(\boldsymbol{\theta},v)a_k(\boldsymbol{\theta},v)\frac{d_{ikm}}{\sqrt{(1-\rho)}}\right)\phi(v)dv\right] \\
&= -\int_{-\infty}^{\infty}\frac{1}{\Phi(v)}\sum_{k=1}^{c}\frac{\partial}{\partial\rho}\left[\prod_{l=1}^{c}A_l(\boldsymbol{\theta},v)\frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\frac{d_{ikm}}{\sqrt{(1-\rho)}}\right]\phi(v)dv \\
&= -\int_{-\infty}^{\infty}\frac{1}{\Phi(v)}\sum_{k=1}^{c}d_{ikm}\left[-\frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\frac{1}{\sqrt{1-\rho}}\sum_{k'=1}^{c}\left(\prod_{l(\neq k')=1}^{c}A_l(\boldsymbol{\theta},v)\right.\right. \\
&\qquad \left.\frac{a_{k'}(\boldsymbol{\theta},v)\mathbf{d}'_{ik'}\boldsymbol{\beta}}{2(1-\rho)^{3/2}}\right) + \frac{\prod_{l=1}^{c}A_l(\boldsymbol{\theta},v)}{\sqrt{1-\rho}}\left\{\frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\left(v-\frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{\sqrt{1-\rho}}\right)\right. \\
&\qquad \left.\frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}} + \left(\frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\right)^2\frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}}\right\} \\
&\qquad \left. -\prod_{l=1}^{c}A_l(\boldsymbol{\theta},v)\frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\frac{1}{2(1-\rho)^{3/2}}\right]\phi(v)dv.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\rho\partial\beta_m} &= -\int_{-\infty}^{\infty}\frac{1}{\Phi(v)}\sum_{k=1}^{c}\left(\prod_{l=1}^{c}A_l(\boldsymbol{\theta},v)\right)\frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\frac{d_{ikm}}{2(1-\rho)^2}\left[-\sum_{k'=1}^{c}\frac{a_{k'}(\boldsymbol{\theta},v)}{A_{k'}(\boldsymbol{\theta},v)}\mathbf{d}'_{ik'}\boldsymbol{\beta}\right. \\
&\qquad \left.+\left(v-\frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{\sqrt{1-\rho}}\right)\mathbf{d}'_{ik}\boldsymbol{\beta} + \frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\mathbf{d}'_{ik}\boldsymbol{\beta} - \sqrt{1-\rho}\right]\phi(v)dv.
\end{aligned} \tag{27}$$

Similarly, the second order partial derivative of $\ell(\theta)$ w.r.t ρ is

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\rho^2} &= \frac{\partial}{\partial\rho}\left[-\int_{-\infty}^{\infty}\frac{1}{\Phi(v)}\sum_{k=1}^{c}\left(\prod_{l(\neq k)=1}^{c}A_l(\boldsymbol{\theta},v)a_k(\boldsymbol{\theta},v)\frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}}\right)\phi(v)dv\right] \\
&= -\int_{-\infty}^{\infty}\frac{1}{\Phi(v)}\sum_{k=1}^{c}\frac{\partial}{\partial\rho}\left[\prod_{l=1}^{c}A_l(\boldsymbol{\theta},v)\frac{a_k(\boldsymbol{\theta},v)}{A_k(\boldsymbol{\theta},v)}\frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}}\right]\phi(v)dv
\end{aligned}$$

Simplifying further,

$$\begin{aligned}\frac{\partial^2 p_{ij}}{\partial \rho^2} &= -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left[-\frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}} \sum_{k'=1}^{c} \left(\prod_{l(\neq k')=1}^{c} A_l(\boldsymbol{\theta}, v) \right.\right. \\ &\quad \left. \frac{a_{k'}(\boldsymbol{\theta}, v)\mathbf{d}'_{ik'}\boldsymbol{\beta}}{2(1-\rho)^{3/2}} \right) + \prod_{l=1}^{c} A_l(\boldsymbol{\theta}, v) \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}} \left\{ \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \right. \\ &\quad \left. \left(v - \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{\sqrt{1-\rho}} \right) \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}} + \left(\frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \right)^2 \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{2(1-\rho)^{3/2}} \right\} \\ &\quad \left. - \prod_{l=1}^{c} A_l(\boldsymbol{\theta}, v) \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \frac{3\, \mathbf{d}'_{ik}\boldsymbol{\beta}}{4(1-\rho)^{5/2}} \right] \phi(v) dv.\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{\partial^2 p_{ij}}{\partial \rho^2} &= -\int_{-\infty}^{\infty} \frac{1}{\Phi(v)} \sum_{k=1}^{c} \left(\prod_{l=1}^{c} A_l(\boldsymbol{\theta}, v) \right) \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{4(1-\rho)^3} \left[-\sum_{k'=1}^{c} \frac{a_{k'}(\boldsymbol{\theta}, v)}{A_{k'}(\boldsymbol{\theta}, v)} \mathbf{d}'_{ik'}\boldsymbol{\beta} \right. \\ &\quad \left. + \left(v - \frac{\mathbf{d}'_{ik}\boldsymbol{\beta}}{\sqrt{1-\rho}} \right) \mathbf{d}'_{ik}\boldsymbol{\beta} + \frac{a_k(\boldsymbol{\theta}, v)}{A_k(\boldsymbol{\theta}, v)} \mathbf{d}'_{ik}\boldsymbol{\beta} - 3\sqrt{1-\rho} \right] \phi(v) dv.\end{aligned} \tag{28}$$

Here m represents the number of covariates.

3.3 ASYMPTOTIC EFFICIENCY COMPARISONS

In this section, we compare the discrete choice probit model with the conditional logit model in large samples and also in small samples. For the large samples case, we compare the asymptotic variance of parameter estimates for both logit and probit models. But, this is not straightforward due to the underlying distributional assumptions of within each model.

3.3.1 NORMALIZATION OF SCALE

In the probit model, the error terms have unit variance by assuming an equicorrelation structure. However, the variance of error terms in conditional logit model are not of unit variance and hence both models are not directly comparable. As the error terms are assumed to follow iid extreme value distribution in a logit model, their variance is $\frac{\pi^2}{6}\sigma^2$. To make the error terms in logit model have unit variance, we scale down the utility u_{ij} by a factor of $\frac{\pi}{\sqrt{6}}\sigma$ as follows:

$$\frac{u_{ij}}{\pi\sigma/\sqrt{6}} = \mathbf{x}'_{ij}\left(\frac{\boldsymbol{\beta}}{\sigma}\right)\frac{1}{\pi/\sqrt{6}} + \frac{z_{ij}}{\pi\sigma/\sqrt{6}}. \tag{29}$$

In a logit model, β and σ are not identified separately but the ratio β/σ is estimated. With scale change in error terms, the new beta coefficients are simply $\frac{1}{\pi/\sqrt{6}}\widehat{\boldsymbol{\beta}}_{CNL}$ and they are now comparable with the probit model.

3.3.2 ASYMPTOTIC RELATIVE EFFICIENCY

From the general theorems for CNL model shown by McFadden (1974), it follows that the maximum likelihood estimator $\widehat{\beta}_{CNL}$ for the conditional logit model has an asymptotically normal distribution with mean β and covariance matrix $\mathcal{I}_{CNL}^{-1}$, where $\mathcal{I}$ is the Fisher information in n subjects given by

$$\mathcal{I}_{CNL} = -\mathrm{E}\left[\frac{\partial^2\ell(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}\right]. \tag{30}$$

Similarly, the maximum likelihood estimator $\widehat{\theta}_{MDCP}$ for the discrete choice probit model with equicorrelation structure is asymptotically normal with mean θ and covariance matrix $\mathcal{I}_{MDCP}^{-1}$ where

$$\mathcal{I}_{MDCP} = -\mathrm{E}\left[\frac{\partial^2\ell(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right]. \tag{31}$$

We computed the asymptotic variances of beta estimates by taking the diagonal elements of the inverses of (30) and (31). The asymptotic relative efficiencies (ARE) are calculated taking the ratio of the variances for the CNL model over the corresponding variances of the Probit model.

$$ARE = \frac{\mathrm{Var}(\frac{\widehat{\boldsymbol{\beta}}_{CNL}}{\pi/\sqrt{6}})}{\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{MDCP})} = \frac{1}{\pi^2/6}\frac{\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{CNL})}{\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{MDCP})}.$$

The expression for second order partial derivatives of conditional logit model is given in (8). This does not involve y_{ij} terms and the expectation of this term is itself. For the multinomial discrete choice probit model, the second order partial derivative matrix consists of expressions (25) through (28) and the expectation of these expressions are as given below.

$$
\begin{aligned}
E\left[\frac{\partial^2 \ell(\theta)}{\partial \beta_m^2}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial \beta_m^2}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial \beta_m}\right)^2, \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial \rho^2}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial \rho^2}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial \rho}\right)^2, \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial \beta_m \partial \rho}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial \beta_m \partial \rho}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial \beta_m}\right)\left(\frac{\partial p_{ij}}{\partial \rho}\right), \text{ and} \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial \beta_{m'} \partial \beta_m}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial \beta_m \partial \beta_{m'}}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial \beta_m}\right)\left(\frac{\partial p_{ij}}{\partial \beta_{m'}}\right).
\end{aligned}
$$

3.3.3 ARE COMPUTATIONS FOR DATA FROM MARKET SCENARIO

For the choice models, data usually comes from two sources namely consumer panels and discrete choice experiments. The models are compared using ARE in both situations. Note that calculation of ARE does not involve estimation of any parameters and it is simply based on a fixed set of covariates with starting parameter values. For computation of AREs, usually the covariates are generated from normal or uniform distributions, which does not work in discrete choice setup. The occurrence of covariates in a discrete choice setup is in such a way that it is competitive in nature between alternatives. To create such a set of covariates, we examined several real time data in literature and generated from multiple normal mixtures so that it reflects true market scenario.

We took a large sample of $n = 1000$ observations with two covariates. The first covariate is a continuous covariate generated from multiple normal mixtures and the second covariate is a discrete covariate with three levels. We assumed the number of choices $c = 4$ and computed ARE for ten different values of ρ ranging from 0.0, …, 0.9. Figure 2 shows histogram of continuous covariate generated from multiple normal mixtures and also a comparison to the real time data.

The respective proportions of discrete covariate with 3 levels for each alternative are given in Table 7. With this setup, the total number of covariates are 6 that include 3 intercepts, 1 continuous covariate and 2 dummy variables for discrete covariate. The mean function is

$$\mu_{ij} = \beta_{01} Int^1 + \beta_{02} Int^2 + \beta_{03} Int^3 + \beta_1 x_{1ij}^{c} + \beta_{21} x_{2ij}^{d1} + \beta_{22} x_{2ij}^{d2}. \tag{32}$$

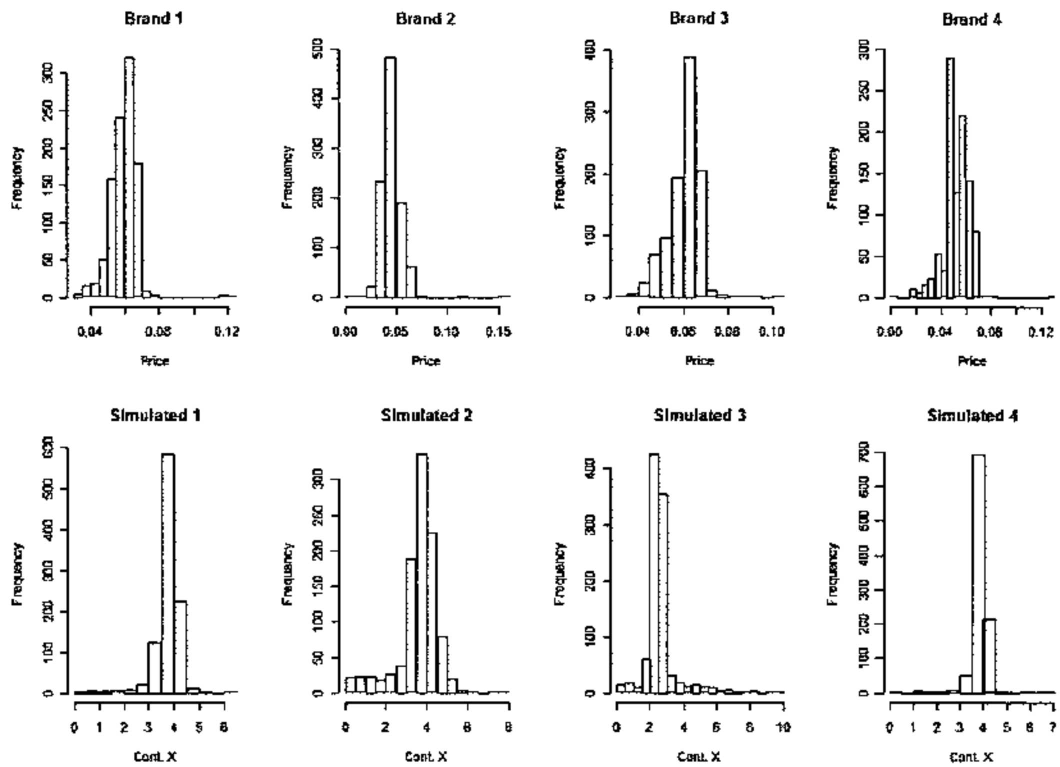

Figure 2. Simulation of continuous covariate that represents true market

The fixed regression coefficients are as follows:
Intercepts: $\widehat{\beta}_{01}$ = -0.479, $\widehat{\beta}_{02}$ = 1.051, $\widehat{\beta}_{03}$ = 0.475,
Continuous covariate: $\widehat{\beta}_1$ = 0.781,
Discrete covariate: $\widehat{\beta}_{21}$ = 0.107, $\widehat{\beta}_{22}$ = -0.525.

We simulated the data with these specifications and for different values of ρ ranging from 0 to 0.9. We obtained the asymptotic variances of both logit and probit models as negative expected value of hessian matrix and computed the variance of parameter estimates as inverse of the Fisher information matrices. Table 8 and Table 9 presents the asymptotic variance and (ARE) for the data simulated from true market scenario.

Table 7. Proportion of levels for discrete covariate

	Level		
Alternative	1	2	3
1	0.15	0.18	0.67
2	0.25	0.23	0.52
3	0.07	0.42	0.51
4	0.39	0.45	0.16

Table 8. Asymptotic variances and ARE* for the intercepts

ρ	$nV(\beta_{01})$	$nV(\beta_{02})$	$nV(\beta_{03})$
0.0	0.0086 (1.072)	0.0053 (0.997)	0.0111 (1.019)
0.1	0.0081 (1.142)	0.0049 (1.079)	0.0104 (1.084)
0.2	0.0075 (1.221)	0.0045 (1.174)	0.0098 (1.156)
0.3	0.0070 (1.311)	0.0041 (1.285)	0.0091 (1.239)
0.4	0.0065 (1.414)	0.0037 (1.414)	0.0085 (1.334)
0.5	0.0060 (1.532)	0.0034 (1.563)	0.0078 (1.444)
0.6	0.0055 (1.667)	0.0031 (1.727)	0.0072 (1.574)
0.7	0.0051 (1.822)	0.0028 (1.878)	0.0065 (1.731)
0.8	0.0046 (2.004)	0.0028 (1.906)	0.0058 (1.939)
0.9	0.0041 (1.254)	0.0035 (1.500)	0.0049 (2.300)
CNL	0.0092	0.0053	0.0113

*AREs are in parenthesis

Table 9. Asymptotic variances and ARE* for the continuous and discrete covariates

ρ	$nV(\beta_1)$	$nV(\beta_{21})$	$nV(\beta_{22})$
0.0	0.0024 (0.924)	0.0076 (0.994)	0.0068 (0.997)
0.1	0.0022 (0.969)	0.0071 (1.067)	0.0064 (1.061)
0.2	0.0021 (1.016)	0.0066 (1.150)	0.0060 (1.134)
0.3	0.0020 (1.067)	0.0060 (1.246)	0.0056 (1.217)
0.4	0.0019 (1.119)	0.0055 (1.358)	0.0052 (1.312)
0.5	0.0019 (1.169)	0.0051 (1.489)	0.0048 (1.420)
0.6	0.0018 (1.210)	0.0046 (1.640)	0.0044 (1.543)
0.7	0.0018 (1.224)	0.0042 (1.816)	0.0041 (1.683)
0.8	0.0019 (1.168)	0.0037 (2.017)	0.0037 (1.833)
0.9	0.0024 (0.924)	0.0032 (2.322)	0.0035 (1.949)
CNL	0.0022	0.0075	0.0068

*AREs are in parenthesis

3.3.4 ARE COMPUTATIONS FOR DATA FROM CHOICE EXPERIMENT

As mentioned in Section 1.1.3, data for choice models come from another source namely designing a choice experiment. This occurs naturally in a market research study in which respondents are shown a choice card consisting of alternatives and asked to pick one. In this setup, an efficient choice design is generated with fixed number of levels for each covariate under consideration. For example, when we would like to evaluate the brand preference of laundry detergent, a choice set is generated to test a fixed number of price levels for each brand. Figure 3 shows a choice set that consists of 18 runs to identify the brand preference of 4 laundry detergent brands. We assume the same setup for ARE computations except that the continuous covariate is replaced with price points from choice design. The results are summarized in Table 10 and Table 11.

3.3.5 DISCUSSION

ARE computations do not involve any parameter estimation and do not require use of optimization routines. The analytical expression for second order partial

Table 10. Asymptotic variances and ARE* for the intercepts in a choice experiment

ρ	$nV(\beta_{01})$	$nV(\beta_{02})$	$nV(\beta_{03})$
0.0	0.0109 (1.100)	0.0061 (1.063)	0.0075 (1.092)
0.1	0.0103 (1.164)	0.0057 (1.176)	0.0070 (1.176)
0.2	0.0097 (1.234)	0.0052 (1.274)	0.0064 (1.274)
0.3	0.0091 (1.311)	0.0048 (1.389)	0.0059 (1.389)
0.4	0.0086 (1.395)	0.0043 (1.526)	0.0054 (1.526)
0.5	0.0081 (1.484)	0.0039 (1.690)	0.0049 (1.690)
0.6	0.0076 (1.571)	0.0035 (1.888)	0.0044 (1.888)
0.7	0.0073 (1.640)	0.0032 (2.131)	0.0039 (2.131)
0.8	0.0073 (1.644)	0.0030 (2.426)	0.0034 (2.426)
0.9	0.0083 (1.440)	0.0033 (2.788)	0.0029 (2.788)
CNL	0.0120	0.0065	0.0082

*AREs are in parenthesis

Table 11. Asymptotic variances and ARE* for the continuous and discrete covariates in a choice experiment

ρ	$nV(\beta_1)$	$nV(\beta_{21})$	$nV(\beta_{22})$
0.0	0.0065 (1.025)	0.0078 (1.013)	0.0062 (1.034)
0.1	0.0061 (1.102)	0.0072 (1.100)	0.0057 (1.117)
0.2	0.0056 (1.191)	0.0066 (1.202)	0.0053 (1.215)
0.3	0.0052 (1.295)	0.0060 (1.326)	0.0048 (1.332)
0.4	0.0047 (1.416)	0.0054 (1.478)	0.0043 (1.473)
0.5	0.0043 (1.559)	0.0047 (1.668)	0.0039 (1.649)
0.6	0.0039 (1.727)	0.0041 (1.915)	0.0034 (1.871)
0.7	0.0035 (1.922)	0.0035 (2.249)	0.0030 (2.160)
0.8	0.0031 (2.125)	0.0029 (2.738)	0.0025 (2.546)
0.9	0.0030 (2.207)	0.0022 (3.616)	0.0021 (3.008)
CNL	0.0067	0.0079	0.0064

*AREs are in parenthesis

Choice of Fabric Softener
Efficient Design

Obs	Sploosh	Plumbbob	Platter	Moosey
1	$1.99	$1.99	$1.99	$2.49
2	$2.49	$1.49	$1.49	$1.99
3	$1.49	$2.49	$2.49	$1.49
4	$2.49	$1.99	$2.49	$1.99
5	$1.49	$1.49	$1.49	$2.49
6	$1.49	$2.49	$1.99	$1.99
7	$2.49	$1.99	$1.99	$1.49
8	$2.49	$2.49	$1.49	$1.49
9	$1.99	$1.49	$2.49	$1.49
10	$1.49	$1.49	$1.99	$1.49
11	$1.99	$2.49	$1.49	$2.49
12	$1.49	$1.99	$1.49	$1.99
13	$1.99	$1.99	$1.49	$1.49
14	$1.49	$1.99	$2.49	$2.49
15	$2.49	$1.49	$2.49	$2.49
16	$1.99	$2.49	$2.49	$1.99
17	$1.99	$1.49	$1.99	$1.99
18	$2.49	$2.49	$1.99	$2.49

Figure 3. Prices from a Choice Experiment

derivatives are derived and then coded directly into SAS and R matrix language software. We computed the expressions (30) and (31) for different values of ρ ranging from 0 to 0.9 by interval of 0.1 and obtained the inverse of Fisher information matrix for probit and logit models. The AREs are calculated for each parameter by taking the ratio of diagonal elements of inverse Fisher information of the two models. The results are displayed in Table 8 for intercepts and in Table 9 for the discrete, continuous covariates in case of data coming from consumer panels. In the case of data coming from designed experiments, the results are displayed in Table 10 for intercepts and in Table 11 for the other covariates. ARE computations for various formulation of mean term (Section 2.2.1) are not performed due to the fact that the results will be similar, irrespective of mean formulation.

From Table 8 and Table 9, the ARE's are about 1 when $\rho = 0$, comparing independent probit model with independent logit. The ARE's increase as the value of ρ increases from 0.0 to 0.9 and the efficiency of probit model is about 2 times to that of logit models for the highest value of $\rho = 0.9$. This table also shows an interesting point relating the coefficients of logit model to the coefficients of probit model in case of nonzero correlation between alternatives. For example, the coefficients in logit model are approximately $\sqrt{1.6}$ times the coefficients of probit model, when $\rho = 0$. The results in Table 8 and Table 9 provides a rough approximation of this relation for $\rho > 0$. Similar conclusions can be drawn in the case of data coming from discrete choice experiments.

3.4 COMPARISONS BASED ON SMALL SAMPLES

In real time applications, sample sizes are usually large for discrete choice models. However, it is of theoretical interest to evaluate the small-sample performance of choice models. To compare the small sample performance, we calculate the mean squared error (MSE) from the true parameter values and compare models. First, we generate the covariates $\mathbf{x}_{ij}$ of sample of size $n = 30$ and fix the regression coefficients $\boldsymbol{\beta}$. Next, we generate the error terms z_{ij} from extreme value distribution for CNL model and from multivariate normal with mean $\mathbf{0}$ and correlation matrix $\mathbf{R}$ for the Probit model. Then the response y_{ij} is generated for both logit and probit models using these inputs so that two datasets are created. For different values of ρ ranging from 0 to 0.9, we simulated 1000 samples and for each sample we estimated the regression parameters using maximum likelihood estimation. The expression for MSE is given as,

$$MSE = \frac{1}{B}\sum_{b=1}^{B}(\widehat{\boldsymbol{\beta}}_b - \boldsymbol{\beta})^2$$

$$\text{and Small Sample Efficiency} = \frac{MSE_L}{MSE_P},$$

where B is the number of simulations. The small sample efficiencies are calculated by taking the ratio of the MSE of the CNL model over the MSE of the MDCP model. Table 12 and Table 13 present the results for small sample efficiencies.

3.4.1 COMPUTATION DETAILS

Small sample efficiency calculations are based on MSE of two models and thus require estimation of parameters. Estimation of parameters involves maximization of log-likelihood function and it requires use of optimization routines. First, we present some of the computational problems involved in obtaining the parameter estimates. For optimization of both logit and probit models, we use a built-in optimization routine in R, called "optim" and it is based on NelderMead, quasi-Newton

Table 12. Small sample variances and efficiency* for intercepts

ρ	$nV(\beta_{01})$	$nV(\beta_{02})$	$nV(\beta_{03})$
0.0	0.2180 (1.559)	0.2046 (2.025)	0.2237 (1.529)
0.1	0.2147 (1.583)	0.2012 (2.060)	0.2097 (1.631)
0.2	0.1790 (1.899)	0.1787 (2.320)	0.2230 (1.534)
0.3	0.1790 (1.899)	0.1670 (2.481)	0.1985 (1.723)
0.4	0.1488 (2.284)	0.1561 (2.656)	0.1683 (2.032)
0.5	0.1438 (2.364)	0.1588 (2.610)	0.1612 (2.122)
0.6	0.1276 (2.664)	0.1596 (2.597)	0.1458 (2.347)
0.7	0.1337 (2.542)	0.1892 (2.191)	0.1481 (2.310)
0.8	0.1135 (2.994)	0.2396 (1.730)	0.1379 (2.480)
0.9	0.1215 (2.797)	0.4093 (1.013)	0.1386 (2.468)

*Efficiency is in parenthesis

Table 13. Small sample variances and efficiency* for continuous and discrete covariates

ρ	$nV(\beta_{1})$	$nV(\beta_{21})$	$nV(\beta_{22})$
0.0	0.0710 (1.455)	0.2219 (1.257)	0.2777 (1.033)
0.1	0.0723 (1.583)	0.2294 (1.216)	0.2459 (1.166)
0.2	0.0769 (1.554)	0.1897 (1.471)	0.2511 (1.142)
0.3	0.0852 (1.463)	0.1771 (1.575)	0.2206 (1.300)
0.4	0.1032 (1.320)	0.1757 (1.587)	0.2144 (1.338)
0.5	0.1119 (1.089)	0.1560 (1.787)	0.2418 (1.186)
0.6	0.1458 (1.004)	0.1522 (1.833)	0.2348 (1.221)
0.7	0.1481 (0.810)	0.1170 (2.383)	0.2097 (1.368)
0.8	0.1379 (0.657)	0.1068 (2.611)	0.2069 (1.386)
0.9	0.1386 (0.392)	0.0914 (3.052)	0.2643 (1.085)

*Efficiency in parenthesis

and conjugate-gradient algorithms. In quasi-Newton methods, two algorithms BFGS and L-BFGS-B are useful for the optimization problem in hand. The first algorithm "BFGS" is useful in the case of estimation of parameters that have no constraints, while the "L-BFGS-B" is Limited memory modified quasi-Newton method with box constraints, most useful when the parameters are constrained. SAS has only limited memory BFGS algorithm as part of PROC OPTMODEL that does not allow box constraints. Please see SAS/OR(R) 9.2 User's Guide: Mathematical Programming: PROC OPTMODEL: NLPU solver for more details. The correlation parameter ρ has constraint $-1/(c-1) < \rho < 1$ and thus require to use constrained optimization. We used R software for optimization.

3.4.2 DISCUSSION

Small sample efficiencies are displayed in Table 12, Table 13 for intercepts and covariates respectively. The results are displayed for different values of ρ from 0.0 to 0.9 by interval 0.1. The results demonstrate the probit model clearly performs better than logit model and this trend increases as ρ increases. Notice that there are few abberations for larger values of $\rho = 0.7, 0.8, 0.9$ for intercepts, partly due to problems in convergence. The convergence rate for both models is well above 95%.

3.5 REAL DATA EXAMPLE

Example 1. Laundry Data:
To illustrate the two models and compare the results, we revisit the laundry detergent example and apply the two models. Here we consider two different formulation of mean as discussed in Section 2.2.1. To recap, the data is from a market research study and contains information about the brand and price of the laundry detergent purchased by 2657 consumers originally analyzed by Chintagunta and Prasad (1998). The dataset contains the log prices of six detergent brands Tide, Wisk, EraPlus, Surf, Solo, and All as well as the brand chosen by each household. We fit both conditional logit model and Multinomial discrete choice probit model to identify the relationship between detergent choice and the price accounting for correlation between alternatives. Table 14 provides point estimates, standard errors and p-values for both the conditional logit and the multivariate discrete choice probit model. It also presents the AIC criterion for comparison of likelihoods of the two models. When comparing two models, the smaller AIC, the better model. Table 14 shows

Table 14. ML estimates for the laundry detergents data

		MDCP Equicorrelation			CNL*		
Parameter		EST.	SE	p-value	EST.	SE	p-value
Intercept	Tide	-1.6982	0.6723	0.0115	-2.6285	1.2128	0.0061
	Wisk	-3.4877	0.6333	< 0.0001	-5.9805	1.1574	<0.0001
	EraPlus	-2.2939	0.6722	0.0006	-3.3530	1.2114	0.0005
	Surf	-3.4071	0.6403	< 0.0001	-5.6066	1.1733	<0.0001
	Solo	-3.1664	0.6758	< 0.0001	-5.1017	1.2324	<0.0001
	All	0.0000	—	—	0.0000	0.0000	—
log-price	Tide	-99.5420	4.8298	< 0.0001	-126.329	7.9440	<0.0001
	Wisk	-66.1671	4.0067	< 0.0001	-87.834	6.7383	<0.0001
	EraPlus	-74.5006	4.9094	< 0.0001	-116.208	8.0835	<0.0001
	Surf	-68.9659	3.9117	< 0.0001	-97.119	7.0350	<0.0001
	Solo	-68.0277	5.2536	< 0.0001	-98.074	9.2168	<0.0001
	All	-202.8864	16.3689	< 0.0001	-310.228	30.9367	<0.0001
ρ		0.1952	0.0086	<0.0001			
AIC		6885.62			7020.79		

*Normalization of scale to have unit variance.

that Probit model performs better than Logit model. The estimated correlation coefficient $\rho = 0.1952$, which is highly significant. The log-price coefficient in probit model has correct intuitive sign and accurately estimated with low standard error compared to the logit model.

Example 2. Travel mode choice:

We illustrate the probit model with equicorrelation structure and the conditional logit model applied to the following travel data example. The data source is Table 21.2 of Greene (2003). This data contains choices made by 210 individuals traveling between Sydney and Melbourne in Australia. The response has four modes of travel namely Air, Train, Bus or Car. The explanatory variables that are specific to alternative are waiting time, travel cost, travel time, general cost, party size and we also have an individual specific variable like household income. There are 840 observations by 210 individuals. We are interested to model the travel mode choice using the explanatory variables such as time, cost, waiting time, etc. We fit both conditional logit model and Multinomial discrete choice probit model with equicorrelation structure and compare the results. Table 15 provides point estimates, standard errors and p-values for both the conditional logit and the multivariate probit model. It also presents the AIC criterion for comparison of likelihoods of the two models.

Table 15. ML estimates for the travel mode data

	MDCP Equicorrelation			**CNL***		
Parameter	EST.	SE	p-value	EST.	SE	p-value
Intercept Air	3.0152	0.5299	< 0.0001	4.0663	0.7857	<0.0001
Train	2.6001	0.2948	< 0.0001	3.4059	0.4314	<0.0001
Bus	2.1068	0.2960	< 0.0001	2.9391	0.4351	<0.0001
Car	0.0000	—	—	—	—	—
Waiting time	-0.0579	0.0061	< 0.0001	-0.0809	0.0091	<0.0001
Travel cost	-0.0563	0.0125	< 0.0001	-0.0663	0.0180	0.0002
Travel time	-0.0086	0.0016	< 0.0001	-0.0104	0.0023	<0.0001
General cost	0.0443	0.0112	< 0.0001	0.0541	0.0162	0.0008
ρ	0.1101	0.0413	0.0077			
AIC	390.813			405.851		

*Normalization of scale to have unit variance.

From Table 15, both models show similar consumer behavior choosing transportation mode. Intercepts show that the relative preference to Air travel is higher compared to other transportation modes. The negative coefficients for waiting time, travel cost and travel time indicate that consumers are choosing the transportation mode that has less waiting time or travel time and cheaper. The estimated correlation is about 0.11, though significant, consumers choose the travel mode alternatives based on factors like time, cost but not switching between them. The AIC criterion shows that probit model performs better than logit model, taking correlation into account.

CHAPTER 4

PROBIT MODEL WITH PRODUCT CORRELATION

Logit models relax IIA assumption by allowing correlation between unobserved factors of choice alternatives. The most widely used GEV models are nested logit models in which all alternatives are partitioned into different nests and relax IIA assumption by assuming a correlation between alternatives within nests. Two variations of nested logit models are prominent, one that allows no overlapping of alternatives between nests and other that allows overlapping of alternatives between nests, known as Generalized Nested Logit (GNL) models. McFadden (1978) developed a process to generate GEV models. Even though the choice probabilities for GEV models can be derived using basic probability rules, this process makes it easier to obtain expression for choice probabilities and development of new GEV models by choosing a different generating function. This process is quite similar to the multivariate extreme value copula models based on properties of MSMVE distributions, discussed in Chapter 5. The process to generate GEV models (McFadden 1978) is outlined in the following section.

4.1 GENERATION OF GEV MODELS

Omitting the subscript i for the subject, consider a function $G(E_1, \ldots, E_c)$ with $E_1, \ldots, E_c \geq 0$ that has the following properties.

1. $G(E_1, \ldots, E_c) \geq 0$ for all positive values of $E_j, \forall j = 1, \ldots, c$.

2. G is homogeneous of degree one, that is $G(\alpha E_1, \ldots, \alpha E_c) = \alpha G(E_1, \ldots, E_c)$ for a constant α

3. $G \to \infty$ as $E_j \to \infty, \forall j = 1, \ldots, c$.

4. The kth order partial derivative of G with respect to E_j are nonnegative for odd k and non-positive for even k.

Any function G that satisfies these properties generates a GEV model and the choice probabilities of this GEV model are of the form

$$p_j = \frac{E_j\, G_j(E_1, \ldots, E_c)}{G(E_1, \ldots, E_c)},$$

where G_j is the first order partial derivative of G with respect to E_j. If we choose $E_j = \exp(\mu_j)$ then E_j is positive for all values of μ_j As an example of this process, we illustrate the derivation of paired combinatorial logit (PCL) model for a specified choice of G that has many potential applications in travel behavior of route choice with overlaps.

4.2 PAIRED COMBINATORIAL LOGIT MODEL

To obtain PCL, let G be of the following form,

$$G(E_{i1}, \ldots, E_{ic}) = \sum_{k=1}^{c-1} \sum_{l=k+1}^{c} \left(E_{ik}^{1/\lambda_{kl}} + E_{il}^{1/\lambda_{kl}} \right)^{\lambda_{kl}}.$$

By choosing $E_{ij} = \exp(\mu_{ij})$, $j = 1, \ldots, c$, the first property of $G \geq 0$ is satisfied. With $0 < \lambda_{kl} \leq 1$, it is easy to see that G is homogeneous of order one and it goes to infinity as E_{ij} goes to infinity. Thus, the corresponding three properties are satisfied. Note that the first order partial derivative of G with respect to E_j is

$$\frac{\partial G(E_{i1}, \ldots, E_{ic})}{\partial E_{ij}} = \sum_{r \neq j} \left(E_{ir}^{1/\lambda_{rj}} + E_{ij}^{1/\lambda_{rj}} \right)^{(\lambda_{rj}-1)} E_{ij}^{(1/\lambda_{rj})-1},$$

and it is nonnegative for $0 < \lambda_{rj} \leq 1$, the second order partial derivative of G with respect to E_{im} is

$$\frac{\partial^2 G(E_{i1}, \ldots, E_{ic})}{\partial E_{im} \partial E_{ij}} = \frac{\lambda_{mj} - 1}{\lambda_{mj}} \left(E_{im}^{1/\lambda_{mj}} + E_{ij}^{1/\lambda_{mj}} \right)^{(\lambda_{mj}-2)} E_{im}^{(1/\lambda_{mj})-1} E_{ij}^{(1/\lambda_{mj})-1}$$

and it is non-positive for $0 < \lambda_{mj} \leq 1$ and so on. Thus all properties are satisfied for chosen G and the expression for choice probability p_{ij} in PCL model is given by

$$\begin{aligned} p_{ij} &= \frac{E_{ij}\, G_j(E_{i1},\ldots,E_{ic})}{G(E_{i1},\ldots,E_{ic})} \\ &= \frac{E_{ij}\, \sum_{r\neq j}\left(E_{ir}^{1/\lambda_{rj}} + E_{ij}^{1/\lambda_{rj}}\right)^{(\lambda_{rj}-1)} E_{ij}^{(1/\lambda_{rj})-1}}{\sum_{k=1}^{c-1}\sum_{l=k+1}^{c}\left(E_{ik}^{1/\lambda_{kl}} + E_{il}^{1/\lambda_{kl}}\right)^{\lambda_{kl}}} \\ &= \frac{\sum_{r\neq j} E_{ij}^{1/\lambda_{rj}}\left(E_{ir}^{1/\lambda_{rj}} + E_{ij}^{1/\lambda_{rj}}\right)^{\lambda_{rj}-1}}{\sum_{k=1}^{c-1}\sum_{l=k+1}^{c}\left(E_{ik}^{1/\lambda_{kl}} + E_{il}^{1/\lambda_{kl}}\right)^{\lambda_{kl}}} \\ &= \frac{\sum_{r\neq j} e^{(\mu_{ij}/\lambda_{rj})}\left(e^{\mu_{ir}/\lambda_{rj}} + e^{\mu_{ij}/\lambda_{rj}}\right)^{\lambda_{rj}-1}}{\sum_{k=1}^{c-1}\sum_{l=k+1}^{c}\left(e^{\mu_{ik}/\lambda_{kl}} + e^{\mu_{il}/\lambda_{kl}}\right)^{\lambda_{kl}}} \end{aligned}$$

which is of the same form as (13). This expression can be rewritten as

$$p_{ij} = \sum_{k\neq j} p_{j/(j,k)} \times p_{(j,k)},$$

where $p_{j/(j,k)}$ is the conditional probability of choosing alternative j given the chosen the pair of alternatives (j,k) and $p_{(j,k)}$ is the marginal probability of selecting the pair (j,k). Given that a pair (j,k) is chosen and the choice of an alternative within this pair follows a binary logit model, the expressions for the conditional probability of choosing jth alternative in the pair (j,k) is

$$p_{j/(j,k)} = \frac{e^{\mu_{ij}/\lambda_{jk}}}{e^{\mu_{ij}/\lambda_{jk}} + e^{\mu_{ik}/\lambda_{jk}}}.$$

Similarly, the marginal probability of choosing the pair (j,k) among the $c(c-1)/2$ possible pairs is given by

$$p_{(j,k)} = \frac{\left(e^{\mu_{ij}/\lambda_{jk}} + e^{\mu_{ik}/\lambda_{jk}}\right)^{\lambda_{jk}}}{\sum_{k=1}^{c-1}\sum_{l=k+1}^{c}\left(e^{\mu_{ik}/\lambda_{kl}} + e^{\mu_{il}/\lambda_{kl}}\right)^{\lambda_{kl}}}.$$

The PCL model has wider application in transportation research for its overlapping nature of choice alternatives. For example, Chu (1989) introduced the PCL model for travel demand analysis and a comparison of conditional logit, nested logit and PCL models is discussed by Koppelman and Wen (2000). Li and Ouyang (2008) presented a modified PCL model that has few computational advantages over original PCL model.

Continuing the performance comparison of probit models over logit models, an equivalent probit model that allows correlation structure similar to PCL is discrete choice probit model with product correlation structure, first considered by Dunnett (1989). The product correlation structure is obtained when $\rho_{jk} = \lambda_j \lambda_k$, where ρ_{jk} is the correlation between alternatives j and k under the restriction $-1 \leq \lambda_j \leq 1$. The correlation coefficient ρ_{jk} is equivalent to the dependency parameter λ_{jk} for the nest containing alternatives j and k. Probit models with simplified structured covariance matrices proposed by Yai, Iwakura and Morichi (1997), Bolduc (1992) to model route choice behavior are some of the alternative models to PCL. In this chapter, we consider probit model with product correlation structure which has more general correlation structure and less parsimonious to PCL. We derive the exact analytical expressions for choice probabilities, Fisher information matrix, ML estimation and compare its performance with PCL model.

4.3 PROBIT MODEL WITH PRODUCT CORRELATION

Assume $\mathbf{R} = [\rho_{jk}]$, where $\rho_{jk} = \lambda_j \lambda_k$ for $-1 \leq \lambda_j \leq 1, j = 1, \ldots, c$. The restrictions on $\lambda_j, j = 1, \ldots, c$ make the correlation structure to be positive definite. Under the assumption of unobserved factors $\mathbf{z}_i$ follows a multivariate normal with mean $\mathbf{0}$ and correlation structure $\mathbf{R}$, the choice probability of ith subject choosing jth alternative is

$$\begin{aligned} p_{ij} &= Pr\left(u_{ij} > u_{ik} \text{ for all } k(\neq j) = 1, \ldots, c\right) \\ &= Pr\left(\mu_{ij} + z_{ij} > \mu_{ik} + z_{ik} \text{ for all } k(\neq j) = 1, \ldots, c\right) \\ &= Pr\left(z_{ij} - z_{ik} > \mu_{ik} - \mu_{ij} \text{ for all } k(\neq j) = 1, \ldots, c\right) \\ &= \int_{\mu_1 - \mu_j}^{\infty} \cdots \int_{\mu_m - \mu_j}^{\infty} \phi_{m-1}(\mathbf{w}; \boldsymbol{\mu}^*, \mathbf{R}^*)\, d\mathbf{w}, \end{aligned}$$

where $\mathbf{W} = (U_1 - U_j, \ldots, U_m - U_j)$ is a $m-1$ multivariate normal with mean $\boldsymbol{\mu}^* = (\mu_1 - \mu_j, \ldots, \mu_m - \mu_j)$ and correlation structure $\mathbf{R}^* = \mathbf{CRC}'$ with

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & \ldots & -1 & \ldots & 0 \\ 0 & 1 & \ldots & -1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -1 & \ldots & 1 \end{pmatrix}$$

4.3.1 STOCHASTIC REPRESENTATIONS

Suppose $X_1, \ldots, X_c$ are has multivariate normal with a product correlation structure $\mathbf{R}$. Then the random variables $(\sqrt{1-\lambda_1^2}V_1 + \lambda_1 V_0, \ldots, \sqrt{1-\lambda_c^2}V_c + \lambda_c V_0)$ follows multivariate normal with mean $\mathbf{0}$ and correlation structure $\mathbf{R}$, where $V_0, V_1, \ldots, V_c$ are i.i.d $N(0,1)$ random variables. (Dunnett 1989). The representation $X_j = \sqrt{1-\lambda_j^2}V_j + \lambda_j V_0$ is known as "stochastic representation" of multivariate normal random variables. In the simplification of choice probabilities with product correlation structure, we apply stochastic representation two times to simplify the $c-1$ variate integral to a bivariate integral.

Using stochastic representation, let $z_{ik} = \sqrt{(1-\lambda_k^2)}v_{ik} + \lambda_k v_{i0}$, where $v_{i0}, v_{i1}, \ldots, v_{ic}$ are independent standard normal variables. Then, $\mathrm{E}(z_{ik}) = 0$, $\mathrm{Var}(z_{ik}) = 1$, and

$$\begin{aligned}
\mathrm{Cov}(z_{ik}, z_{ik'}) &= \sqrt{(1-\lambda_k^2)}\sqrt{(1-\lambda_{k'}^2)}\mathrm{Cov}(v_{ik}, v_{ik'}) \\
&\quad + \sqrt{(1-\lambda_k^2)}\lambda_{k'}\mathrm{Cov}(v_{ik}, v_{i0}) \\
&\quad + \lambda_k\sqrt{(1-\lambda_{k'}^2)}\mathrm{Cov}(v_{i0}, v_{ik'}) \\
&\quad + \lambda_k\lambda_{k'}\mathrm{Cov}(v_{i0}, v_{i0}) \\
&= \lambda_k\lambda_{k'}.
\end{aligned}$$

Hence,

$$\begin{aligned}
p_{ij} &= Pr\left(z_{ij} - z_{ik} > \mu_{ik} - \mu_{ij}\right) \\
&= Pr\left(\sqrt{(1-\lambda_j^2)}v_{ij} - \sqrt{(1-\lambda_k^2)}v_{ik} + (\lambda_j - \lambda_k)v_{i0} > \mu_{ik} - \mu_{ij}\right) \\
&= \int_{-\infty}^{\infty} Pr\left[D_{ik} > (\mu_{ik} - \mu_{ij}) + (\lambda_k - \lambda_j)v | v;\ \text{for all}\ k(\neq j)\right] \phi(v)\, dv \\
&= \int_{-\infty}^{\infty} Pr\left[D_{ik} > C_{ik}(v) | v;\ \text{for all}\ k(\neq j)\right] \phi(v)\, dv, \qquad (33)
\end{aligned}$$

where $C_{ik}(v) = (\mu_{ik} - \mu_{ij}) + (\lambda_k - \lambda_j)v$ and $D_{ik} = \sqrt{(1-\lambda_j^2)}v_{ij} - \sqrt{(1-\lambda_k^2)}v_{ik}$. Note that, $D_{ik}, k(\neq j) = 1, \ldots, c$ are normal with mean 0, variance $(1-\lambda_j^2) + (1-\lambda_k^2)$ and $Cov(D_{ik}, D_{ik'}) = 1 - \lambda_j^2$, so they are not independent. To simplify further, we again use the following stochastic representation.

Let $D_{ik} = \sqrt{(1-\lambda_j^2)}w_{i0} + \sqrt{(1-\lambda_k^2)}w_{ik}, k \neq j$, where $w_{i0}, w_{i1}, \ldots, w_{ic}$ are independent standard normal radon variables. Then,

$$\begin{aligned}
\mathrm{E}(D_{ik}) &= 0; \\
\mathrm{Var}(D_{ik}) &= (1-\lambda_j^2) + (1-\lambda_k^2); \\
\mathrm{Cov}(D_{ik}, D_{ik'}) &= (1-\lambda_j^2)\mathrm{Cov}(w_{i0}, w_{i0}) \\
&= (1-\lambda_j^2).
\end{aligned}$$

Therefore, the choice probability under above special correlation structure becomes,

$$
\begin{aligned}
p_{ij} &= \int_{-\infty}^{\infty} Pr\left[D_{ik} > C_{ik}(v)|v;\ \text{for all}\ k(\neq j)\right] \phi(v)\,dv \\
&= \int_{-\infty}^{\infty} Pr\left[\sqrt{(1-\lambda_j^2)}w_{i0} + \sqrt{(1-\lambda_k^2)}w_{ik} > C_{ik}(v)|v;\ \text{for all}\ k(\neq j)\right] \phi(v)\,dv \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} Pr\left[w_{ik} > \frac{C_{ik}(v) - \sqrt{(1-\lambda_j^2)}w}{\sqrt{(1-\lambda_k^2)}}|v,w;\ \text{for all}\ k(\neq j)\right] \phi(w)\phi(v)dwdv \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \prod_{k(\neq j)=1}^{c} \left[1 - \Phi\left(\frac{C_{ik}(v) - \sqrt{(1-\lambda_j^2)}w}{\sqrt{(1-\lambda_k^2)}}\right)\right] \phi(w)\phi(v)dwdv \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\Phi(w)} \prod_{k=1}^{c} \left[\Phi\left(\frac{(\mu_{ij}-\mu_{ik}) + (\lambda_j - \lambda_k)v + \sqrt{(1-\lambda_j^2)}w}{\sqrt{(1-\lambda_k^2)}}\right)\right] \\
&\qquad \phi(w)\phi(v)dwdv. \qquad (34)
\end{aligned}
$$

4.4 MAXIMUM LIKELIHOOD ESTIMATION

Similar to the probit model with equicorrelation structure, we assume that the means $\mu_{ij}, j = 1 \ldots c$ are linear functions of $\mathrm{x}_{ij}'\boldsymbol{\beta}$ and our goal is to estimate the unknown parameter $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\lambda})$ using the maximum likelihood estimation method. The log-likelihood $\ell(\boldsymbol{\theta})$ for n subjects is

$$
\ell(\boldsymbol{\theta}) = \log\left[\prod_{i=1}^{n}\prod_{j=1}^{c} p_{ij}^{y_{ij}}\right] = \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\log(p_{ij}),
$$

where p_{ij} computed using expression (34). The maximum likelihood estimate of $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\lambda}})$ of $\boldsymbol{\theta}$ is the solution of likelihood equations $\partial\ell(\theta)/\partial\theta = 0$. The expressions for the first order partial derivatives of the log-likelihood are

$$
\begin{aligned}
\frac{\partial\ell(\theta)}{\partial\boldsymbol{\theta}} &= \left[\frac{\partial\ell(\theta)}{\partial\boldsymbol{\beta}}\ \frac{\partial\ell(\theta)}{\partial\boldsymbol{\lambda}}\right] \\
&= \left[\frac{\partial\ell(\theta)}{\partial\beta_0}, \ldots, \frac{\partial\ell(\theta)}{\partial\beta_p}, \frac{\partial\ell(\theta)}{\partial\lambda_1}, \ldots, \frac{\partial\ell(\theta)}{\partial\lambda_c}\right].
\end{aligned}
$$

The first order partial derivatives of $\ell(\boldsymbol{\theta})$ with respect β_m, λ_j, $\lambda_r, (r \neq j)$ are given in the following section.

4.4.1 EXPRESSIONS FOR SCORE EQUATIONS

Let $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\lambda})$ where $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_c)$. For simplicity, let $t_1(v, w, l, j)$, $t_2(v, w, l, j)$ and $t_3(v, w, r, j)$ denote the following quantities.

$$\begin{aligned}
t_1(v,w,l,j) &= \left[\frac{(\mu_{ij} - \mu_{il}) + (\lambda_j - \lambda_l)v + \sqrt{(1-\lambda_j^2)}w}{\sqrt{(1-\lambda_l^2)}}\right], \\
t_2(v,w,l,j) &= \left[\frac{v}{\sqrt{1-\lambda_l^2}} - \frac{\lambda_j w}{\sqrt{1-\lambda_j^2}\sqrt{1-\lambda_l^2}}\right], \\
t_3(v,w,r,j) &= \left[\frac{(\mu_{ij} - \mu_{ir})\lambda_r - v(1-\lambda_r^2) + \lambda_r(\lambda_j - \lambda_r)v + \lambda_r\sqrt{1-\lambda_j^2}w}{(1-\lambda_r^2)^{3/2}}\right].
\end{aligned}$$

Further, let $A_l(\boldsymbol{\theta}, v, w) = \Phi(t_1(v, w, l, j))$ and $a_l(\boldsymbol{\theta}, v, w) = \phi(t_1(v, w, l, j))$ Then the derivatives of $A_l(\boldsymbol{\theta}, v, w)$, $a_l(\boldsymbol{\theta}, v, w)$ with respect to $\beta_m, \lambda_j, \lambda_r (r \neq j)$ for all $m = 0, \ldots, p; r (\neq j) = 1, \ldots, c$ are

$$\begin{aligned}
\frac{\partial}{\partial \beta_m} A_l(\boldsymbol{\theta}, v, w) &= -a_l(\boldsymbol{\theta}, v, w)\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}, \\
\frac{\partial}{\partial \lambda_j} A_l(\boldsymbol{\theta}, v, w) &= a_l(\boldsymbol{\theta}, v, w)\, t_2(v,w,l,j), \\
\frac{\partial}{\partial \lambda_r} A_r(\boldsymbol{\theta}, v, w) &= a_r(\boldsymbol{\theta}, v, w)\, t_3(v,w,r,j), \\
\frac{\partial}{\partial \beta_m} a_l(\boldsymbol{\theta}, v, w) &= a_l(\boldsymbol{\theta}, v, w)\, t_1(v,w,l,j)\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}, \\
\frac{\partial}{\partial \lambda_j} a_l(\boldsymbol{\theta}, v, w) &= -a_l(\boldsymbol{\theta}, v, w)\, t_1(v,w,l,j)\, t_2(v,w,l,j), \\
\frac{\partial}{\partial \lambda_r} a_r(\boldsymbol{\theta}, v, w) &= -a_r(\boldsymbol{\theta}, v, w)\, t_1(v,w,l,j)\, t_3(v,w,r,j),
\end{aligned}$$

where $d_{ilm} = (x_{ilm} - x_{ijm})$. Using the above results, we obtain the first order partial derivatives of log-likelihood function with respect to β_m, λ_j, and $\lambda_r, (r \neq j)$ in the following manner.

$$\begin{aligned}\frac{\partial p_{ij}}{\partial \beta_m} &= \frac{\partial}{\partial \beta_m}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\prod_{l=1}^{c}A_l(\boldsymbol{\theta},v,w)\ \phi(w)\ \phi(v)\ dw\ dv\right]\\ &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left[\sum_{l=1}^{c}\left(\prod_{k(\neq l)=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)a_l(\boldsymbol{\theta},v,w)\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]\\ &\qquad \phi(w)\ \phi(v)\ dw\ dv.\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{\partial p_{ij}}{\partial \beta_m} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]\\ &\qquad \phi(w)\ \phi(v)\ dw\ dv.\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{\partial p_{ij}}{\partial \lambda_j} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left[\sum_{l(\neq j)=1}^{c}\left(\prod_{k(\neq l)=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)\right]\\ &\qquad \phi(w)\ \phi(v)\ dw\ dv\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\ t_2(v,w,l,j)\right]\\ &\qquad \phi(w)\ \phi(v)\ dw\ dv,\end{aligned}$$

and

$$\begin{aligned}\underset{(r\neq j)}{\frac{\partial p_{ij}}{\partial \lambda_r}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k(\neq r)=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)a_r(\boldsymbol{\theta},v,w)\,t_3(v,w,r,j)\\ &\qquad \phi(w)\ \phi(v)\ dw\ dv\\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\ t_3(v,w,r,j)\\ &\qquad \phi(w)\ \phi(v)\ dw\ dv.\end{aligned}$$

4.4.2 EXPRESSIONS FOR HESSIAN MATRIX

The second order partial derivatives consist of evaluating the 7 expressions

$$\frac{\partial^2 p_{ij}}{\partial\beta_{m'}\partial\beta_m},\ \frac{\partial^2 p_{ij}}{\partial\lambda_j\partial\beta_m},\ \frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\beta_m},\ \frac{\partial^2 p_{ij}}{\partial\lambda_j^2},\ \frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\lambda_j},\ \frac{\partial^2 p_{ij}}{\partial\lambda_{r'}\partial\lambda_r},\ \frac{\partial^2 p_{ij}}{\partial\lambda_r^2},$$

and their analytical expressions are derived as given below. The second order partial derivative of log-likelihood with respect to β_m and β'_m, $(m \neq m')$ is given by

$$
\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\beta_{m'}\partial\beta_m} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\frac{\partial}{\partial\beta_{m'}}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right] \\
&\quad + \left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{\partial}{\partial\beta_{m'}}\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]\phi(w)\,\phi(v)\,dw\,dv \\
&= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left\{\left[-\sum_{k'=1}^{c}\left(\prod_{k(\neq k')=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)a_{k'}(\boldsymbol{\theta},v,w)\frac{d_{ik'm'}}{\sqrt{1-\lambda_{k'}^2}}\right]\right. \\
&\quad \left(\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right) + \left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l=1}^{c}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right. \\
&\quad \left.\left.\left\{\frac{a_l(\boldsymbol{\theta},v,w)\,t_1(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm'}}{\sqrt{1-\lambda_l^2}} + \left(\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right)^2\frac{d_{ilm'}}{\sqrt{1-\lambda_l^2}}\right\}\right]\right\} \\
&\quad \phi(w)\,\phi(v)\,dw\,dv.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\beta_{m'}\partial\beta_m} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left\{\left[\sum_{k'=1}^{c}\frac{a_{k'}(\boldsymbol{\theta},v,w)}{A_{k'}(\boldsymbol{\theta},v,w)}\frac{d_{ik'm'}}{\sqrt{1-\lambda_{k'}^2}}\right]\right. \\
&\quad \left(\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right) - \left[\sum_{l=1}^{c}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right. \\
&\quad \left.\left.\frac{d_{ilm'}}{\sqrt{1-\lambda_l^2}}\left\{t_1(v,w,l,j) + \frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right\}\right]\right\}\phi(w)\,\phi(v)dwdv.
\end{aligned} \tag{35}
$$

Similarly the second order partial derivative of log-likelihood with respect to β_m and λ_j is given by

$$
\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\lambda_j\partial\beta_m} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\frac{\partial}{\partial\lambda_j}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right] \\
&\quad + \left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{\partial}{\partial\lambda_j}\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]\phi(w)\,\phi(v)\,dw\,dv.
\end{aligned}
$$

This can be simplified as

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\lambda_j\partial\beta_m} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\Bigg\{\Bigg[\sum_{k'(\neq j)=1}^{c}\left(\prod_{k(\neq k')=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)a_{k'}(\boldsymbol{\theta},v,w)\\
&\qquad t_2(v,w,k',j)\Bigg]\left(\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right)-\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\\
&\qquad\Bigg[\sum_{l=1}^{c}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\Bigg\{t_2(v,w,l,j)\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}t_1(v,w,l,j)\\
&\qquad\qquad+\left(\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right)^2 t_2(v,w,l,j)\Bigg\}\Bigg]\Bigg\}\phi(w)\phi(v)dwdv\\
&= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\Bigg\{\Bigg[\sum_{k'(\neq j)=1}^{c}\frac{a_{k'}(\boldsymbol{\theta},v,w)\,t_2(v,w,k',j)}{A_{k'}(\boldsymbol{\theta},v,w)}\Bigg]\\
&\qquad\left(\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right)-\Bigg[\sum_{l=1}^{c}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}t_2(v,w,l,j)\\
&\qquad\left\{t_1(v,w,l,j)+\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right\}\Bigg]\Bigg\}\phi(w)\,\phi(v)\,dw\,dv. \qquad (36)
\end{aligned}$$

Now, the second order partial derivative of log-likelihood with respect to β_m and $\lambda_r, (r \neq j)$ can be obtained as

$$\begin{aligned}
\underset{(r\neq j)}{\frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\beta_m}} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\frac{\partial}{\partial\lambda_r}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]\\
&\qquad+\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{\partial}{\partial\lambda_r}\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]\phi(w)\,\phi(v)\,dw\,dv\\
&= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\Bigg\{\prod_{k(\neq r)=1}^{c}A_k(\boldsymbol{\theta},v,w)\,a_r(\boldsymbol{\theta},v,w)\,t_3(v,w,r,j)\\
&\qquad\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]+\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\Bigg[-\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\\
&\qquad\frac{d_{irm}\,t_1(v,w,r,j)\,t_3(v,w,r,j)}{\sqrt{1-\lambda_r^2}}+\frac{d_{irm}a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\frac{\lambda_r}{(1-\lambda_r^2)^{3/2}}\\
&\qquad-\frac{d_{irm}\,t_3(v,w,r,j)}{1-\lambda_r^2}\left(\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\right)^2\Bigg]\Bigg\}\phi(w)\phi(v)dwdv.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\underset{(r\neq j)}{\frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\beta_m}} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\left\{\left[\sum_{l=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right.\right.\\
&\quad\left.\frac{d_{ilm}}{\sqrt{1-\lambda_l^2}}\right]t_3(v,w,r,j)-\frac{d_{irm}}{\sqrt{1-\lambda_r^2}}\left[\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\,t_3(v,w,r,j)\right.\\
&\quad\left.\left.+t_1(v,w,r,j)\,t_3(v,w,r,j)-\frac{\lambda_r}{1-\lambda_r^2}\right]\right\}\phi(w)\phi(v)dwdv. \qquad (37)
\end{aligned}$$

Further, the second order partial derivatives of log-likelihood with respect to λ_j^2 is

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\lambda_j^2} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\frac{\partial}{\partial\lambda_j}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\right]\\
&\quad+\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{\partial}{\partial\lambda_j}\left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\right]\phi(w)\,\phi(v)\,dw\,dv\\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left\{\sum_{l(\neq j)=1}^{c}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\right.\\
&\quad\left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\right]+\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l(\neq j)=1}^{c}\left\{\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right.\right.\\
&\quad-t_1(v,w,l,j)\,t_2(v,w,l,j)^2-\left(\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}t_2(v,w,l,j)\right)^2\\
&\quad\left.\left.\left.-\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\frac{w}{\sqrt{1-\lambda_l^2}}\left(\frac{\lambda_j}{\sqrt{1-\lambda_j^2}}+\frac{\lambda_j^2}{(1-\lambda_j^2)^{3/2}}\right)\right\}\right]\right\}\phi(w)\,\phi(v)\,dw\,dv.
\end{aligned}$$

Hence,

$$\begin{aligned}
\frac{\partial^2 p_{ij}}{\partial\lambda_j^2} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left\{\left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\right]^2\right.\\
&\quad-\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\left[t_1(v,w,l,j)\,t_2(v,w,l,j)^2+\frac{w}{\sqrt{1-\lambda_l^2}(1-\lambda_j^2)^{3/2}}\right.\\
&\quad\left.\left.+\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\,t_2(v,w,l,j)^2\right]\right\}\phi(w)\,\phi(v)\,dw\,dv.
\end{aligned}$$

(38)

Also, the second order partial derivatives of log-likelihood with respect to λ_j and $\lambda_r (r \neq j)$ is

$$
\begin{aligned}
\underset{(r\neq j)}{\frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\lambda_j}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\frac{\partial}{\partial\lambda_r}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}t_2(v,w,l,j)\right] \\
&\quad + \left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{\partial}{\partial\lambda_r}\left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\right]\phi(w)\phi(v)\,dwdv \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left\{\prod_{k(\neq r)=1}^{c}A_k(\boldsymbol{\theta},v,w)\,a_r(\boldsymbol{\theta},v,w)t_3(v,w,r,j)\right. \\
&\quad \left[\sum_{l(\neq j)=1}^{c}\frac{a_l(\boldsymbol{\theta},v,w)\,t_2(v,w,l,j)}{A_l(\boldsymbol{\theta},v,w)}\right] + \left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[-\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\right. \\
&\quad t_1(v,w,r,j)\,t_2(v,w,r,j)\,t_3(v,w,r,j) - t_2(v,w,r,j)\,t_3(v,w,r,j) \\
&\quad \left.\left.\left(\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\right)^2 + \frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\left(\frac{\lambda_r v}{(1-\lambda_r^2)^{3/2}} - \frac{\lambda_j w}{\sqrt{1-\lambda_j^2}}\frac{\lambda_r}{(1-\lambda_r^2)^{3/2}}\right)\right]\right\} \\
&\quad \phi(w)\,\phi(v)\,dw\,dv.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
\underset{(r\neq j)}{\frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\lambda_j}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\left\{\left[\sum_{l(\neq j)=1}^{c}t_2(v,w,l,j)\right.\right. \\
&\quad \left.\frac{a_l(\boldsymbol{\theta},v,w)}{A_l(\boldsymbol{\theta},v,w)}\right]t_3(v,w,r,j) - \left[t_1(v,w,r,j)\,t_2(v,w,r,j)\,t_3(v,w,r,j)\right. \\
&\quad \left.\left.+\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}\,t_2(v,w,r,j)\,t_3(v,w,r,j) - \frac{\lambda_r}{(1-\lambda_r^2)^{3/2}}\left(v - \frac{\lambda_j w}{\sqrt{1-\lambda_j^2}}\right)\right]\right\} \\
&\quad \phi(w)\,\phi(v)\,dw\,dv. \qquad (39)
\end{aligned}
$$

Similarly, the second order partial derivatives of log-likelihood with respect to $\lambda_r^2 (r \neq j)$ is

$$
\begin{aligned}
\underset{(r\neq j)}{\frac{\partial^2 p_{ij}}{\partial\lambda_r^2}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{\Phi(w)}\frac{\partial}{\partial\lambda_r}\left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\left[\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}t_3(v,w,r,j)\right] \\
&\quad + \left(\prod_{k=1}^{c}A_k(\boldsymbol{\theta},v,w)\right)\frac{\partial}{\partial\lambda_r}\left[\frac{a_r(\boldsymbol{\theta},v,w)}{A_r(\boldsymbol{\theta},v,w)}t_3(v,w,r,j)\right]\phi(w)\,\phi(v)\,dw\,dv.
\end{aligned}
$$

This can be simplified as

$$\begin{aligned}
\underset{(r\neq j)}{\frac{\partial^2 p_{ij}}{\partial \lambda_r^2}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\Phi(w)} \left\{ \left(\prod_{k=1}^{c} A_k(\boldsymbol{\theta}, v, w)\right) \left[\left(\frac{a_r(\boldsymbol{\theta}, v, w)}{A_r(\boldsymbol{\theta}, v, w)}\right)^2 t_3(v, w, r, j)^2 \right] \right. \\
&\quad - \left(\prod_{k=1}^{c} A_k(\boldsymbol{\theta}, v, w)\right) \frac{a_r(\boldsymbol{\theta}, v, w)}{A_r(\boldsymbol{\theta}, v, w)} t_3(v, w, r, j)^2 t_1(v, w, r, j) \\
&\quad - \left(\prod_{k=1}^{c} A_k(\boldsymbol{\theta}, v, w)\right) \left(\frac{a_r(\boldsymbol{\theta}, v, w)}{A_r(\boldsymbol{\theta}, v, w)} t_3(v, w, r, j)\right)^2 \\
&\quad \left. + \left(\prod_{k=1}^{c} A_k(\boldsymbol{\theta}, v, w)\right) \frac{a_r(\boldsymbol{\theta}, v, w)}{A_r(\boldsymbol{\theta}, v, w)} \frac{\partial t_3(v, w, r, j)}{\partial \lambda_r} \right\} \phi(w)\, \phi(v)\, dw\, dv
\end{aligned}$$

Note that,

$$\frac{\partial t_3(v, w, r, j)}{\partial \lambda_r} = \frac{(\mu_{ij} - \mu_{ir})(1 + 2\lambda_r^2) + (\lambda_j + 2\lambda_j \lambda_r^2 - 3\lambda_r) + \sqrt{1 - \lambda_j^2}(1 + 2\lambda_r^2) w}{(1 - \lambda_r^2)^{5/2}}$$

Therefore,

$$\begin{aligned}
\underset{(r\neq j)}{\frac{\partial^2 p_{ij}}{\partial \lambda_r^2}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\Phi(w)} \left(\prod_{k=1}^{c} A_k(\boldsymbol{\theta}, v, w)\right) \frac{a_r(\boldsymbol{\theta}, v, w)}{A_r(\boldsymbol{\theta}, v, w)} \left\{ \left[-t_1(v, w, r, j) t_3(v, w, r, j)^2 \right.\right. \\
&\quad \left.\left. + \frac{(\mu_{ij} - \mu_{ir})(1 + 2\lambda_r^2) + (\lambda_j + 2\lambda_j \lambda_r^2 - 3\lambda_r) v + \sqrt{1 - \lambda_j^2}(1 + 2\lambda_r^2) w}{(1 - \lambda_r^2)^{5/2}} \right] \right\} \\
&\quad \phi(w)\, \phi(v)\, dw\, dv. \qquad (40)
\end{aligned}$$

Finally, the second order partial derivatives of log-likelihood with respect to $\lambda_r (r \neq j)$ and $\lambda_r' (r \neq r' \neq j)$is

$$\begin{aligned}
\underset{(r\neq r'\neq j)}{\frac{\partial^2 p_{ij}}{\partial \lambda_{r'} \partial \lambda_r}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\Phi(w)} \frac{\partial}{\partial \lambda_{r'}} \left(\prod_{k=1}^{c} A_k(\boldsymbol{\theta}, v, w)\right) \left[\frac{a_r(\boldsymbol{\theta}, v, w)}{A_r(\boldsymbol{\theta}, v, w)} t_3(v, w, r, j) \right] \\
&\quad \phi(w)\, \phi(v)\, dw\, dv \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{\Phi(w)} \left[\frac{a_r(\boldsymbol{\theta}, v, w)}{A_r(\boldsymbol{\theta}, v, w)} t_3(v, w, r, j) \right] \left(\prod_{k=1}^{c} A_k(\boldsymbol{\theta}, v, w)\right) \\
&\quad \frac{a_{r'}(\boldsymbol{\theta}, v, w)}{A_{r'}(\boldsymbol{\theta}, v, w)} t_3(v, w, r', j) \phi(w) \phi(v) dw dv. \qquad (41)
\end{aligned}$$

4.5 ASYMPTOTIC EFFICIENCY COMPARISONS

In this section, we compare the discrete choice probit model with product correlation structure to the paired combinatorial logit model in large samples. As described in Section 3.4, comparisons can be done in small samples using mean square error. However, we do not perform these computations in this dissertation due to time consuming computations that run into several days. As mentioned in Section 3.3.1, normalization of scale is required to compare logit models with probit models. This is even true for comparison of PCL model with discrete choice probit model with product correlation. In order to ensure both models are at same level, we assume a product correlation structure for the PCL model. With normalization of scale, the new beta coefficients in PCL are $\widehat{\beta}_{PCL}/(\pi/\sqrt{6})$. With this the two models are on par with each other and can be compared.

4.5.1 ASYMPTOTIC RELATIVE EFFICIENCY

From the general theorems for logit models shown by McFadden (1974), it follows that the maximum likelihood estimator $\widehat{\theta}_{PCL}$ for the PCL model asymptotically has a normal distribution with mean θ and covariance matrix $\mathcal{I}_{PCL}^{-1}$, where $\mathcal{I}$ is the Fisher information in n subjects given by

$$\mathcal{I}_{PCL} = -\mathrm{E}\left[\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right] \tag{42}$$

Similarly, the maximum likelihood estimator $\widehat{\theta}_{MDCP}$ for the discrete choice probit model with product correlation structure is asymptotically normal with mean θ and covariance matrix $\mathcal{I}_{MDCP}^{-1}$ where

$$\mathcal{I}_{MDCP} = -\mathrm{E}\left[\frac{\partial^2 \ell(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right] \tag{43}$$

We computed the asymptotic variances of beta estimates by taking the diagonal elements of the inverses of (42) and (43). The asymptotic relative efficiencies (ARE) are calculated taking the ratio of the variances for the PCL model over the corresponding variances of the MDCP model with product correlation.

$$ARE = \frac{\mathrm{Var}(\frac{\widehat{\boldsymbol{\theta}}_{PCL}}{\pi/\sqrt{6}})}{\mathrm{Var}(\widehat{\boldsymbol{\theta}}_{MDCP})} = \frac{1}{\pi^2/6}\frac{\mathrm{Var}(\widehat{\boldsymbol{\theta}}_{PCL})}{\mathrm{Var}(\widehat{\boldsymbol{\theta}}_{MDCP})}$$

The expression for second order partial derivatives of PCL model is given in (8). This does not involve y_{ij} terms and the expectation of this term is itself. For the multinomial discrete choice probit model, the second order partial derivative matrix consists of expressions (35) through (41) and the expectation of these expressions are as given below.

$$
\begin{aligned}
E\left[\frac{\partial^2 \ell(\theta)}{\partial\beta_{m'}\partial\beta_m}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial\beta_m\partial\beta_{m'}}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial\beta_m}\right)\left(\frac{\partial p_{ij}}{\partial\beta_{m'}}\right) \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial\lambda_j\partial\beta_m}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial\lambda_j\partial\beta_m}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial\lambda_j}\right)\left(\frac{\partial p_{ij}}{\partial\beta_m}\right) \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial\lambda_r\partial\beta_m}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\beta_m}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial\lambda_r}\right)\left(\frac{\partial p_{ij}}{\partial\beta_m}\right) \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial\lambda_r\partial\lambda_j}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial\lambda_r\partial\lambda_j}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial\lambda_r}\right)\left(\frac{\partial p_{ij}}{\partial\lambda_j}\right) \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial\lambda_{r'}\partial\lambda_r}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial\lambda_{r'}\partial\lambda_r}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial\lambda_{r'}}\right)\left(\frac{\partial p_{ij}}{\partial\lambda_r}\right) \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial\lambda_j^2}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial\lambda_j^2}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial\lambda_j}\right)^2 \\
E\left[\frac{\partial^2 \ell(\theta)}{\partial\lambda_r^2}\right] &= \sum_{i=1}^{n}\sum_{j=1}^{c}\left(\frac{\partial^2 p_{ij}}{\partial\lambda_r^2}\right) - \sum_{i=1}^{n}\sum_{j=1}^{c}\frac{1}{p_{ij}}\left(\frac{\partial p_{ij}}{\partial\lambda_r}\right)^2
\end{aligned}
$$

4.5.2 ARE COMPUTATIONS FOR DATA FROM MARKET SCENARIO

As described in Section 3.3.3, for the choice models data usually comes from two sources namely consumer panels and discrete choice experiments. We perform the efficiency comparisons only in case of data coming from consumer panels. The results are similar in case of data coming from discrete choice experiments. We assume the same setup as in the case of asymptotic efficiency computations for comparing CNL to the probit model with equicorrelation structure. We generate the continuous covariate from multiple normal mixture so as to resemble real market scenario.

Similar to efficiency comparison of CNL to the probit model with equicorrelation structure, we took a large sample of $n = 1000$ observations with two covariates. The first covariate is a continuous covariate generated from multiple normal mixtures (Figure 2) and the second covariate is a discrete covariate with three levels (Table

Table 16. Arbitrarily chosen values of $\boldsymbol{\lambda}$

S. No.	value of $\boldsymbol{\lambda}$
1	(-0.754, -0.681, -0.769, -0.738)
2	(-0.701, -0.516, -0.1686, 0.379)
3	(0.283, -0.075, -0.546, 0.293)
4	(0.676, -0.547, -0.426, -0.810)

7). We assumed the number of choices $c = 4$ and computed ARE for arbitrarily chosen values of correlation parameters $\boldsymbol{\lambda}$. We selected 4 different values of $\boldsymbol{\lambda}$ to see how both models perform relative to each other. Large number of simulations for different values of $\boldsymbol{\lambda}$ can considered, but omitted due to computational burden. These computations are heavier, often run into several days and require optimization of R program we developed. With this setup, the total number of covariates are 6 that include 3 intercepts, 1 continuous covariate and 2 dummy variables for discrete covariate. The mean function is

$$\mu_{ij} = \beta_{01} Int^1 + \beta_{02} Int^2 + \beta_{03} Int^3 + \beta_1 x^c_{1ij} + \beta_{21} x^{d1}_{2ij} + \beta_{22} x^{d2}_{2ij} \tag{44}$$

The fixed regression coefficients are as follows: Intercepts: β_{01} = -0.479, β_{02} = 1.051, β_{03} = 0.475, Continuous covariate: β_1 = 0.781, Discrete covariate: β_{21} = 0.107, β_{22} = -0.525 We simulated the data with these specifications and for 4 different values of $\boldsymbol{\lambda}$. They are given in Table 16. We obtained the asymptotic variances of both PCL and MDCP model with product correlation structure as negative expected value of hessian matrix and computed the variance of parameter estimates as inverse of the Fisher information matrices. Table 17 and Table 18 presents the asymptotic variance and (ARE) for the data simulated from true market scenario.

4.5.3 DISCUSSION

ARE computations does not involve any parameter estimation and doest not require use of optimization routines. The analytical expression for second order partial derivatives are derived and then coded directly into SAS IML and R softwares. We computed the expressions (42) and (43) for arbitrarily chosen values of $\boldsymbol{\lambda}$ and obtained the inverse of Fisher information matrix for MDCP with product correlation

Table 17. Asymptotic variances and ARE for $\boldsymbol{\beta}$ estimates

S.No	Method	β_{01}	β_{02}	β_{03}	β_1	β_{21}	β_{22}
1	MDCP II	0.0475	0.0305	0.0483	0.001	0.0063	0.0043
	PCL	0.0738	0.0536	0.0501	0.0163	0.0101	0.0133
	ARE	1.5537	1.7574	1.0373	23.2857	1.6032	3.0930
2	MDCP II	0.0139	0.0069	0.0145	0.0017	0.0077	0.0063
	PCL	0.0102	0.0086	0.0080	0.0027	0.0069	0.0063
	ARE	0.7338	1.2464	0.5517	1.5882	0.8961	1.0000
3	MDCP II	0.0235	0.0183	0.0189	0.0014	0.0093	0.0066
	PCL	0.0035	0.0120	0.0051	0.0055	0.0026	0.0041
	ARE	0.1489	0.6557	0.2698	3.9286	0.2796	0.6212
4	MDCP II	0.0346	0.0063	0.0089	9.7349	0.0054	0.0045
	PCL	0.2450	0.1156	0.0740	10.9208	0.0068	0.0174
	ARE	7.0809	18.3492	8.3146	1.1218	1.2593	3.8667

Table 18. Asymptotic variances and ARE for $\boldsymbol{\lambda}$ estimates

S.No	Method	λ_1	λ_2	λ_3	λ_4
1	MDCP II	0.0094	0.0028	0.0091	0.0150
	PCL	0.0675	0.0851	0.0997	0.0406
	ARE	7.1809	30.3929	10.9560	2.7067
2	MDCP II	0.0408	0.0028	0.0170	0.0045
	PCL	0.0821	0.0417	0.0093	0.0217
	ARE	2.0123	14.8929	0.5471	4.8222
3	MDCP II	0.0180	0.0033	0.0044	0.0701
	PCL	0.0242	0.0039	0.0729	0.0227
	ARE	1.3444	1.1818	16.5682	0.3238
4	MDCP II	0.0640	0.0018	0.0061	0.0102
	PCL	0.1207	0.1293	0.0745	0.1015
	ARE	1.8859	71.8333	12.2131	9.9510

and PCL models. The AREs are calculated for each parameter by taking the ratio of diagonal elements of inverse Fisher information of the two models. The results are displayed in Table 17 for $\boldsymbol{\beta}$ estimates and in Table 18 for the correlation parameters $\boldsymbol{\lambda}$. ARE computations for various formulation of mean term (Section 2.2.1) are not performed due to the fact that the results will be similar irrespective of mean formulation.

The ARE's are expected to be around 1.64 without normalization of PCL model. Some of the AREs in Table 17 and Table 18 are much higher or much lower than 1.64 due to estimation error. This is especially true in case of the MDCP II model, as the MDCP II models require numerical approximation of a double integral and the built-in "integrate" routine in R sometimes fail. Exploration of other numerical methods to evaluate double integral are required to accurately estimate the variances in the MDCP II model. Note that, the small values of variances indicate that the results are very close to the true values. However, valid conclusions can be drawn only after estimating the variances of MDCP II model to the desired level of accuracy. In general, probit models are preferred to incorporate other phenomenon such as random taste variation or repeated responses. If the data does not contain any of this information, PCL is preferred for its simplicity over probit model.

4.6 REAL DATA EXAMPLE

Example 1. Laundry Data:
To illustrate the two models and compare the results, we revisit the laundry detergent example and apply two models. Here we consider two different formulation of mean as discussed in section 2.2.1. To recap, the data is from a market research study and contains information about the brand and price of the laundry detergent purchased by 2657 consumers originally analyzed by Chintagunta and Prasad (1998). The dataset contains the log prices of six detergent brands Tide, Wisk, EraPlus, Surf, Solo, and All as well as the brand chosen by each household. We fit both PCL model and Multinomial discrete choice probit model with product correlation structure to identify the relationship between detergent choice and the price accounting for correlation between alternatives. Table 19 provides point estimates, standard errors and p-values for both the PCL model and the multinomial discrete choice probit model. It also presents the AIC criterion for comparison of likelihoods of the two

models. Though both models have similar results, we observed that these estimates

Table 19. ML estimates for the laundry detergents data

		MDCP II			PCL*		
Parameter		EST.	SE	p-value	EST.	SE	p-value
Intercept	Tide	3.8442	1.1948	0.0013	3.7509	1.5416	0.0150
	Wisk	2.6804	1.4877	0.0716	3.3861	1.4319	0.0180
	EraPlus	3.4890	1.4492	0.0161	3.7226	1.5096	0.0137
	Surf	4.3297	1.1014	0.0001	3.8608	1.4230	0.0067
	Solo	1.7799	1.7632	0.3128	2.4863	1.5551	0.1099
	All	0.0000	—	—	0.0000	0.0000	—
log-price	Tide	-108.533	12.9253	< 0.0001	-108.489	12.8931	<0.0001
	Wisk	-105.327	12.0531	< 0.0001	-105.926	6.7383	<0.0001
	EraPlus	-105.720	10.6645	< 0.0001	-106.190	10.2916	<0.0001
	Surf	-106.205	11.1658	< 0.0001	-105.474	11.0812	<0.0001
	Solo	-103.499	19.8372	< 0.0001	-104.123	19.7120	<0.0001
	All	-106.629	35.4821	< 0.0001	-106.088	35.9607	0.0032
Correlation	Tide	0.6569	0.4519	0.1460	1.0000	0.5238	0.0562
	Wisk	-0.0419	0.2813	0.8817	-0.6833	0.3559	0.0549
	EraPlus	0.1120	0.2645	0.6719	0.1146	0.1083	0.2900
	Surf	-0.5427	0.8284	0.5124	-0.9868	0.5265	0.0609
	Solo	0.9088	0.9648	0.3462	1.0000	0.7346	0.1735
	All	0.3748	0.1827	0.0403	0.2231	0.3752	0.5522
AIC		7584.25			7610.18		

*Normalization of scale to have unit variance.

are susceptible to starting values. Some more starting values have to be tested before confirming the results of these two models. Due to time consuming computational issues, not all observations were used in estimation. Also, computation of choice probabilities in MDCP model require use of built-in "integrate" routines in R, which does not yield accurate results. Further exploration of numerical methods is required for accurate results. In view of this, we do not interpret the model coefficients and draw any conclusions.

Example 2. Travel mode choice:
We illustrate the probit model with product correlation structure and the PCL model applied to the following travel data example. The data source is Greene (2003) Table 21.2. This data contains choices made by 210 individuals traveling between Sydney and Melbourne in Australia. The response has four modes of travel namely Air, Train, Bus or Car. The explanatory variables that are specific to alternative are waiting time, travel cost, travel time, general cost, party size and we also have individual specific variable like household income. There are 840 observations by 210 individuals. We are interested to model the travel mode choice using the explanatory variables such as time, cost, waiting time, etc. We fit both PCL model and Multinomial discrete choice probit model with product correlation structure and compare the results. Table 20 provides point estimates, standard errors and p-values for both the PCL model and the multinomial discrete choice probit model. It also presents the AIC criterion for comparison of likelihoods of the two models.

Table 20. ML estimates for the travel mode data

	MDCP II			**PCL***		
Parameter	EST.	SE	p-value	EST.	SE	p-value
Intercept Air	4.9645	0.9282	< 0.0001	4.5117	0.9468	<0.0001
Train	4.6968	0.6260	< 0.0001	4.8459	0.5420	<0.0001
Bus	3.0797	0.6631	< 0.0001	4.0197	0.6121	<0.0001
Car	0.0000	——	——	——	——	——
Waiting time	-0.1739	0.0548	0.0015	-0.1150	0.0120	<0.0001
Travel cost	-0.1661	0.0648	0.0104	-0.1095	0.0254	<0.0001
Travel time	-0.0347	0.0526	0.5094	-0.0190	0.0033	<0.0001
General cost	0.0677	0.0717	0.3453	0.0934	0.0241	0.0001
Correlation Air	0.8026	0.6405	0.2102	1.0000	0.6377	0.1168
Train	-0.7291	0.5019	0.1463	-0.5874	0.4505	0.1922
Bus	0.8704	0.8094	0.2822	1.0000	0.7173	0.1633
Car	-0.8954	0.6876	0.1928	-0.9958	0.7512	0.1849
AIC	490.813			465.851		

*Normalization of scale to have unit variance.

From Table 20, Though both models have similar results, we observed that these estimates are susceptible to starting values. Some more starting values have to be tested before confirming the results of these two models. Due to time consuming computational issues, not all observations were used in estimation. Also, computation of choice probabilities in MDCP model require use of built-in "integrate" routines in R, which does not yield accurate results. Further exploration of numerical methods is required for accurate results. In view of this, we do not interpret the model coefficients and draw any conclusions.

CHAPTER 5

DISCRETE CHOICE COPULA MODELS

In the previous two chapters, we developed discrete choice probit models for two correlation structures namely equicorrelation structure and product correlation structure. Further, we compared the efficiency of these probit models to the equivalent specification of logit models and concluded that probit models perform better than logit models. A probit model is obtained by assuming that the unobserved factors have a multivariate normal distribution with a correlation structure in which the diagonal elements are always one. However, logit models do not have unit variance structure across diagonals and require normalization of scale to compare with probit models (see Section 3.3.1). Note that the joint distribution of unobserved factors in a multivariate probit model can be represented using Gaussian copula with standard normal marginals. Without having to normalize the scale, the ideal choice of a logit model to compare with a probit model is the logit model with the joint distribution of unobserved factors modeled using Gaussian copula with extreme value marginals. Further, the logit models can be represented using extreme value copulas that describe the multivariate extreme value distribution with extreme value marginals. Extreme value copulas define a multivariate extreme value distribution with a dependence function that governs the dependence structure between alternatives and choice of several dependence functions lead to several logit models.

In this chapter, we present the theory of copulas, basic definitions, examples and application of copulas in modeling discrete choice behavior. We focus our attention on two copulas, the extreme value copula for logit models and Gaussian copula for probit models. Extreme value copulas are introduced in Section 5.3.1 and Gaussian copulas are given in Example 5.2. Further, we derive previously studied logit and probit models as special cases of these two copulas. We conclude this chapter with some ideas of future research on copula based methods for discrete choice data.

5.1 COPULAS

Copulas are general tools to construct or describe multivariate distributions with specified marginal distributions. By definition, a copula by itself is a multivariate distribution with marginals that are uniform on the unit interval $[0, 1]$. In addition, a copula characterizes the structure of the dependence between the chosen marginals. The simplest way of constructing a multivariate copula function is to "invert" the marginal distribution functions and use them as the elements for the joint distribution function. In the following sections, we define a copula, discuss some well known examples, study their basic properties and related results.

Definition 5.1 A d-dimensional copula is a function $C : [0,1]^d \to [0,1]$ with the following properties. Let $\mathbf{u} = (u_1, \ldots, u_d)$ be in $[0, 1]^d$. Then

1. *$C(\mathbf{u}) = 0$ if at least one element of $\mathbf{u}$ is 0.*
2. *If all elements of $\mathbf{u}$ are 1 except u_k, then $C(\mathbf{u}) = u_k$, for $k = 1, \ldots, d$.*
3. *$C(\mathbf{u})$ is right continuous as a function of $\mathbf{u}$.*
4. *For all $0 < a_{j_1} < a_{j_2} < 1$, $j = 1, \ldots, d$,*

$$\sum_{r_1=1}^{2}\sum_{r_2=1}^{2}\cdots\sum_{r_d=1}^{2}(-1)^{r_1+r_2+\cdots+r_d}C(a_{1r_1}, a_{2r_2}, \ldots, a_{dr_d}) \geq 0$$

It follows that image of $C = [0,1]$, so C is a multivariate uniform distribution function. Below are some examples of copulas that are useful in our context.

5.1.1 EXAMPLES OF COPULAS

Example 5.1 *Independence Copula.* This is also known as *Product Copula.* It is a d-variate function given by

$$C_d(\mathbf{u}) = \prod_{j=1}^{d} u_j. \tag{45}$$

Example 5.2 *Multivariate Gaussian Copula.* Let $\mathbf{R}$ be a symmetric and positive definite correlation matrix. Let $\Phi_c(z_1, \ldots, z_d; \mathbf{0}, \mathbf{R})$ be the d-variate normal distribution function with mean $\mathbf{0}$ and correlation $\mathbf{R}$ given by

$$\Phi_d(z_1, \ldots, z_d; \mathbf{0}, \mathbf{R}) = \int_{-\infty}^{z_d}\cdots\int_{-\infty}^{z_1}\frac{1}{(2\pi)^{\frac{d}{2}}|\mathbf{R}|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\mathbf{z}'\,\mathbf{R}^{-1}\,\mathbf{z}\right)dz$$

The multivariate Gaussian copula with correlation matrix $\mathbf{R}$ is given by

$$C(\mathbf{u}; \mathbf{R}) = \Phi_d(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d); \mathbf{0}, \mathbf{R}), \tag{46}$$

where $\Phi^{-1}(.)$ is the inverse of the cumulative standard normal distribution function. Note that when $\mathbf{R} = \mathbf{I}$, this copula reduces to the Independence Copula.

Example 5.3 *Multivariate Gumbel-Hougaard Copula.* This multivariate copula is given by

$$C(\mathbf{u};\theta) = \exp\{-((-\log u_1)^\theta + \cdots + (-\log u_d)^\theta)^{1/\theta}\}. \quad (47)$$

The range of the parameter θ is $[1,\ \infty)$, and it is an indicator of the degree of dependence. This copula reduces to the independence copula when $\theta = 1$, and as $\theta \to \infty$ it converges to the Fréchet-Hoeffding upper bound C_U given below. See Gudendorf and Segers (2009) for a discussion of this copula.

To establish the relationship between multivariate cumulative distribution functions and their univariate margins via a copula function, the following fundamental theorem due to Sklar (1959) plays an important role.

Theorem 5.1. Let $F(y_1, \ldots, y_d)$ be a joint distribution function of d random variables with marginal distribution functions $F_1(y_1), \ldots, F_d(y_d)$. Then there exists a d-variate copula C such that for real numbers y_i, $1 \leq i \leq d$,

$$F(y_1, \ldots, y_d) = C(F_1(y_1), \ldots, F_d(y_d)). \quad (48)$$

Further, if $F_1, \ldots, F_d$ are continuous, then C is unique. Otherwise, C is uniquely determined on the set $Range(F_1) \times Range(F_2), \times \cdots \times Range(F_d)$. Conversely, if C is a d-variate copula and $F_1(y_1), \ldots, F_d(y_d)$ are univariate distribution functions, then the function $F(y_1, \ldots, y_d)$ defined by (48) is a d-variate distribution function with marginals $F_1(y_1), \ldots, F_d(y_d)$.

The copulas are bounded functions and the bounds are known as Fréchet-Hoeffding bounds which are described in the following theorem.

Theorem 5.2 If C is any d-variate copula, then for every $\mathbf{u} = (u_1, \ldots, u_d)$ in $[0,1]^d$,

$$C_L(\mathbf{u}) \leq C(\mathbf{u}) \leq C_U(\mathbf{u})$$

where the Fréchet-Hoeffding lower bound C_L and upper bound C_U are defined as

$$\begin{aligned} C_L(\mathbf{u}) &= \max(0, u_1 + \cdots + u_d - (d-1)), \\ C_U(\mathbf{u}) &= \min(u_1, \ldots, u_d). \end{aligned}$$

The upper bound C_U is a d-variate copula for any $d \geq 2$. It is known as *comonotonicity copula.* The lower bound C_L is a copula for only $d = 2$, and it is known as *countermonotonicity copula.*

5.1.2 COPULA DENSITIES

By Sklar's theorem, the cumulative distribution function of d-variate random vector $\mathbf{Y} = (Y_1, \ldots, Y_d)$ can be written as

$$F(\mathbf{y}) = C(F_1(y_1), \ldots, F_d(y_d)),$$

where C is a d-dimensional copula. When $\mathbf{Y}$ is a continuous random vector, the joint probability density function of $\mathbf{Y}$ can be obtained as

$$f(\mathbf{y}) = \prod_{j=1}^{d} f_j(y_j) \;\; c(F_1(y_1), \ldots, F_d(y_d)),$$

where $f_j(y_j)$ is the marginal probability density function of Y_j, and the copula density of C given by

$$c(u_1, \ldots, u_d) = \frac{\partial^d C(u_1, \ldots, u_d)}{\partial u_1 \partial u_2 \ldots \partial u_d}.$$

Similarly, when $\mathbf{Y}$ is a discrete random vector, the joint probability mass function of $\mathbf{Y}$ can be written as

$$Pr(\mathbf{y}) = \sum_{j_1=1}^{2} \sum_{j_2=1}^{2} \cdots \sum_{j_d=1}^{2} (-1)^{j_1+j_2+\cdots+j_d} c(u_{1j_1}, u_{2j_2}, \ldots, u_{dj_d}),$$

where $u_{j1}(y_j) = F_j(y_i^-)$ and $u_{j2}(y_j) = F_j(y_i)$. Also, the conditional distribution $F(u_1, \ldots, u_{d-1}|u_d)$ is given by

$$F(u_1, \ldots, u_{d-1}|u_d) = \frac{\partial C(u_1, \ldots, u_d)}{\partial u_d}.$$

5.2 GAUSSIAN COPULA DISCRETE CHOICE MODELS

We now introduce a more general form of discrete choice probit model using the Gaussian copula and show that the probit models in previous chapters are a special case. This generalization allows us to construct discrete choice models with various correlation structures for the unobserved factors. Also, the construction allows us to use other marginal distributions other than standard normal, such as Gumbel and allows comparison without normalization of scale.

Let $\mathbf{Y} = (Y_1, \ldots, Y_c)$ denote the response vector of c choices in a discrete choice experiment. Note that $\mathbf{Y}$ is c-variate binary random vector with the restriction that only one of Y_j is equal to 1 and rest all are zero. Or simply,

$$\mathbf{Y} \sim Multinomial(1, (p_1, \ldots, p_c)) \text{ with } \sum_{j=1}^{c} p_j = 1 \text{ and } \sum_{j=1}^{c} Y_j = 1.$$

Following the utility maximization theory, let U_j be the latent utility of the jth alternative for $j = 1, \ldots, c$. Further, assume that $U_j = \mu_j + Z_j$, where μ_j is the mean and Z_j denotes the unobserved random component. Then the choice probability can be computed as

$$\begin{aligned} p_j &= Pr(U_j > U_k,\ k \neq j) \\ &= Pr(Z_k < Z_j + (\mu_j - \mu_k),\ k \neq j). \end{aligned}$$

Additional assumptions are needed to compute this choice probability. An assumption that the joint distribution of $\mathbf{Z} = (Z_1, \ldots, Z_c)$ is multivariate normal with mean 0 and correlation matrix $\mathbf{R}$ leads to the discrete choice probit model. Replacing the distribution of $\mathbf{Z} = (Z_1, \ldots, Z_c)$ by a copula based distribution would lead to Discrete Choice Copula models. Discrete choice probit model is a Gaussian copula model with marginals as standard normal.

Suppose that the joint distribution of $\mathbf{Z} = (Z_1, \ldots, Z_c)$ can be represented by a Gaussian copula as

$$F(\mathbf{z}) = \Phi_c(F(z_1), \ldots, F(z_c))$$

where $F(z_j)$ is the cumulative distribution function of Z_j. Then the choice probability can be written as

$$\begin{aligned} p_j &= Pr(Z_k < Z_j + (\mu_j - \mu_k),\ k \neq j) \\ &= \int_{-\infty}^{\infty} Pr(Z_k < (\mu_j - \mu_k) + z_j\ k \neq j | Z_j = z_j)\, f(z_j)\, dz_j \\ &= \int_{-\infty}^{\infty} \Phi_{c-1}(F(z_1^*), \ldots, F(z_{j-1}^*), F(z_{j+1}^*), \ldots, F(z_c^*) | Z_j = z_j)\, f(z_j)\, dz_j \quad (49) \end{aligned}$$

where $z_k^* = (\mu_j - \mu_k) + z_j$ and $\Phi_{c-1}(F(z_1), \ldots, F(z_{j-1}), F(z_{j+1}), \ldots, F(z_c)|Z_j = z_j)$ denotes the conditional distribution function. We can simplify the probability (49) further when the marginals are assumed to be standard normal as given below.

Let $\mathbf{R}$ be the correlation matrix parameter in the Gaussian copula. Let $\mathbf{R}$ be partitioned as

$$\mathbf{R} = \begin{pmatrix} 1 & \mathbf{R_{12}} \\ \mathbf{R_{21}} & \mathbf{R_{22}} \end{pmatrix}$$

The conditional distribution of $\mathbf{Z_{-j}} = (Z_1, \ldots, Z_{j-1}, Z_{j+1}, \ldots, Z_c)$ given $Z_j = z_j$ is $MVN(\mathbf{R_{21}}z_j, \mathbf{R_{22}} - \mathbf{R_{21}}\mathbf{R_{12}})$. Hence the choice probability (49) can be written as

$$p_j =$$

$$\int_{-\infty}^{\infty} \Phi_{c-1}(F(z_1^*), \ldots, F(z_{j-1}^*), F(z_{j+1}^*), \ldots, F(z_c^*); \mathbf{R_{21}}z_j, \mathbf{R_{22}} - \mathbf{R_{21}}\mathbf{R_{12}})\, f(z_j)\, dz_j$$

The matrices $\mathbf{R_{21}}$ and $\mathbf{R_{22}} - \mathbf{R_{21}}\mathbf{R_{12}}$ can be easily calculated for equicorrelation and product correlation structures.

5.3 EXTREME VALUE THEORY

Extreme value distributions are limiting distributions of extremes such as minimum or maximum of a sequence of random variables. In the univariate case, the well known "Fisher-Tippett-Gnedenko" three types theorem can be described as follows.

Let $X_1, X_2, \ldots, X_n$ be iid random variables with a common distribution function F. Let $X_{(n)} = \max(X_1, \ldots, X_n)$. For suitably chosen sequences $\{a_n\}$ and $\{b_n\}$, the possible limiting distribution of $(X_{(n)} - a_n)/b_n$ as $n \to \infty$ is one of the following distributions.

1. Gumbel or Extreme value distribution with $F_0(z) = \exp\{-e^{-z}\}, -\infty < z < \infty$
2. Fréchet distribution with $F_1(z, \theta) = \exp\{-z^{-\theta}\}, z > 0, \theta > 0$
3. Weibull distribution with $F_{-1}(z, \theta) = \exp\{-(-z)^{\theta}\}, z < 0, \theta > 0$

where z is of the form $z = (x - \mu)/\sigma$. With location-scale changes, the three distributions can be combined into the Generalized Extreme Value (GEV) family as

$$F(z; \gamma) = \exp\{-(1 + \gamma z)_+^{-1/\gamma}\}, -\infty < z < \infty, -\infty < \gamma < \infty, \tag{50}$$

where $(t)_+ = \max\{0, t\}$. When $\gamma \to 0$, $F(z;\gamma)$ reduces to Gumbel distribution, when $\gamma > 0$, $F(z;\gamma)$ reduces to Fréchet distribution and lastly the condition $\gamma < 0$ yields the Weibull distribution. This theory can be extended to multivariate case yielding multivariate extreme value distributions and the dependence structure via a multivariate copula, known as *Extreme Value Copula.*

5.3.1 EXTREME VALUE COPULAS

Before proceeding onto extreme value copulas, we first describe the characterization of a multivariate extreme value distribution. Let $(X_{i1}, X_{i2}, \ldots, X_{id})$ for $i = 1, \ldots, n$ be d-dimensional iid random vectors with a common joint distribution function F, which is determined by a Copula C_F and marginals $F_1, \ldots, F_d$. Let $(X_{(n1)}, X_{(n2)}, \ldots, X_{(nd)})$ denote the componentwise maxima and $F_{(n1)}, \ldots, F_{(nd)}$ denote their distribution functions. Then the multivariate extreme value (MEV) distribution is a limiting distribution of $((X_{(n1)} - a_{n1})/b_{n1}, \ldots, (X_{(nd)} - a_{nd})/b_{nd})$ as $n \to \infty$ and for some suitable normalizing constants a_{nj} and b_{nj}, $1 \le j \le m$. It can be written in the form $C(H(z_1;\gamma_1), \ldots, H(z_d;\gamma_d))$, where $H(z_j;\gamma_j)$ is a GEV distribution parametrized by γ_j, for $j = 1, \ldots, d$. To construct a MEV distribution and the copula that characterizes this distribution, we need to study the copula related to the maximums. The case of minimums will be similar by symmetry.

Note that the copula of a maximum of n random vectors can be written as $C_{(n)}(\mathbf{u}) = C_F(u_1^{1/n}, \ldots, u_d^{1/n})^n$ for $\mathbf{u}$ in $[0,1]^d$ To see this, observe that $F_{(nj)}(x_j) = Pr(X_{ij} \le x_j \,\forall\, 1 \le i \le n) = [Pr(X_{1j} \le x_j)]^n = [F_j(x_j)]^n$. Now, the joint distribution of componentwise maxima can be obtained as

$$
\begin{aligned}
F_{(n)}(x_1, \ldots, x_d) &= Pr(X_{(n1)} \le x_1, \ldots, X_{(nd)} \le x_d) \\
&= Pr(X_{i1} \le x_1, \ldots, X_{id} \le x_d \,\forall i) \\
&= [F(x_1, \ldots, x_d)]^n \\
&= [C_F(F_1(x_1), \ldots, F_d(x_d))]^n \\
&= [C_F([F_{(n1)}(x_1)]^{1/n}, \ldots, [F_{(nd)}(x_d)]^{1/n})]^n.
\end{aligned}
$$

Therefore, the copula that characterizes the joint distribution of componentwise maxima, denoted by $C_{(n)}(u_1, \ldots, u_d)$ can be written as $C_{(n)}(u_1, \ldots, u_d) = C_F(u_1^{1/n}, \ldots, u_d^{1/n})^n$. This leads to the following definition.

Definition 5.2 A copula C is called an *Extreme Value Copula* if there exists a copula $C_F(\mathbf{u})$ such that

$$C_F(u_1^{1/n}, \ldots, u_d^{1/n})^n \to C(\mathbf{u}) \text{ as } n \to \infty$$

for all $\mathbf{u} = (u_1, \ldots, u_d)$ in $[0,1]^d$. The copula $C_F(\mathbf{u})$ is said to be in the *domain of attraction* of $C(\mathbf{u})$.

Because copula is a multivariate distribution function, we have the following definition.

Definition 5.3 A d-variate copula $C(\mathbf{u})$ is *max-stable* if

$$C(\mathbf{u}) = C(u_1^{1/r}, \ldots, u_d^{1/r})^r$$

holds for every integer $r \geq 1$ and all $\mathbf{u} = (u_1, \ldots, u_d)$ in $[0,1]^d$.

One can show that a copula is max-stable if and only if it is an extreme value copula. See Nelsen (2006) for a proof. We now describe a procedure for constructing extreme value copulas using Pickands (1981) representation.

Let $C(\mathbf{u})$ be a d-variate max-stable copula. Let the distribution of the random vector $X = (X_1, \ldots, X_d)$ be determined by $C(\mathbf{u})$ and standard exponential marginals with mean one. The joint survival function is given by

$$S(x_1, \ldots, x_d) = Pr(X_1 > x_1, \ldots, X_m > x_d) = \overline{C}(e^{-x_1}, \ldots, e^{-x_d}) \tag{51}$$

where $\overline{C}$ is the survival function of the copula C. Let

$$z_j = x_j/(x_1 + \cdots + x_d) \text{ and } r = x_1 + \cdots + x_d.$$

Note that $\sum_{j=1}^d z_j = 1$. Since $C(\mathbf{u})$ is max-stable, we have

$$\begin{aligned} S(x_1, \ldots, x_d) &= S(rz_1, \ldots, rz_d) \\ &= \overline{C}(e^{-rz_1}, \ldots, e^{-rz_d}) \\ &= [\overline{C}(e^{-z_1}, \ldots, e^{-z_d})]^r \\ &= \exp\{-r\, A(z_1, \ldots, z_d)\} \end{aligned}$$

where $A : [0,\infty)^d \to [1/d, 1]$ is the function defined as

$$A(z_1, \ldots, z_d) = -\log \overline{C}(e^{-z_1}, \ldots, e^{-z_d}). \tag{52}$$

The above function $A(z_1, \ldots, z_d)$ is known as *tail dependence function* of the extreme value copula. It is related to the extreme value copula by the equation

$$C(u_1, \ldots, u_d) = \exp\left\{ \log\left(\prod_{j=1}^{d} u_j\right) A\left(\frac{\log\, u_1}{\log\,(\prod_{j=1}^{d} u_j)}, \ldots, \frac{\log\, u_d}{\log\,(\prod_{j=1}^{d} u_j)}\right)\right\}. \tag{53}$$

Therefore, if $C(\mathbf{u})$ is an extreme value copula, then it is of the form (53) for an appropriate choice of $A(z_1, \ldots, z_d)$. For (53) to be a copula, $A(z_1, \ldots, z_d)$ must satisfy the following properties.

1. $A(z_1, \ldots, z_d)$ is convex.

2. $A(z_1, \ldots, z_d)$ is homogeneous of order 1, that is, $A(rz_1, \ldots rz_d) = rA(z_1, \ldots, z_d)$ for $r > 0$.

3. $\max(z_1, \ldots, z_d) \le A(z_1, \ldots, z_d) \le 1$ for all $(z_1, \ldots, z_d)$ in $[0,1]^d$.

The above construction is known as Pickands representation of a min-stable multivariate exponential distribution (MSMVE) using survival function (or max-stable using distribution function). The result is summarized in the following theorem.

Theorem 5.3 A d-variate copula $C(\mathbf{u})$ is an Extreme Value Copula if and only if there exists finite measure H on the unit simplex $D_m = \{(w_1, \ldots, w_d) \in [0,\infty)^d; \sum_{j=1}^{d} w_j = 1\}$, called as *spectral measure*, such that

$$C(\mathbf{u}) = \exp\left\{-A\left(-\log u_1, \ldots, -\log u_d\right)\right\},$$

where the tail dependence function $A : [0,\infty)^d \to [1/d, 1]$ is given by

$$A(z_1, \ldots, z_d) = \int_{D_d} \left[\max_{1\le j\le d} w_j z_j\right] \, dH(w_1, \ldots, w_d), \qquad (z_1, \ldots, z_d) \in [0,\infty)^d$$

For a proof of the above theorem, see Galambos (1987). The above representation of $C(\mathbf{u})$ can be simplified further in bivariate case, that is, when $d = 2$. In bivariate case, Theorem 5.3 reduces to the following result.

Theorem 5.4 A bivariate copula is an Extreme Value Copula if and only if

$$C(u_1, u_2) = \exp\left\{\log\left(u_1 u_2\right) A\left(\frac{\log u_2}{\log\left(u_1 u_2\right)}\right)\right\}$$

where $A : [0,1] \to [0.5, 1]$ is convex and satisfies $\min(z, 1-z) \leq A(z) \leq 1$ for all $z \in [0,1]$.

5.4 EXTREME VALUE COPULA MODELS

Extreme value copulas with a dependency function A of the form $-\log S$, where S is a survival function, result in extreme value distributions that are MSMVE. Joe (1997), Section 6.3, described three dependency functions that are of the form $-\log S$. Two of the three dependent functions are relevant to our discussion. The first one results in Gumbel (1960) family of extreme value copulas and the second results in normal family of extreme value copulas. All GEV models, discussed in Section 2.3.1, can be represented using Gumbel family of extreme value copulas with a variety of dependence patterns generated from the given dependency function. This process first described by McFadden (1978) to generate GEV family is actually based on the properties of MSMVE distributions. Using the normal family of extreme value copulas, we can generate extreme value models with dependency structure similar to that of a multivariate normal distribution. These have not be explored to model choice behavior in the literature. The dependency function that generates normal family of extreme value copulas is derived as an extreme value limit of bivariate or multivariate normal distribution. We exploit the properties of MSMVE distributions to obtain the choice probabilities, which result in a closed form expressions due to the property that the class of MSMVE distributions is closed under margins.

As a way forward, we first explore the case of bivariate families of copula with a single parameter for dependency function and then consider multivariate extensions with multiple parameters that describe the dependency structure between marginals.

5.4.1 GUMBEL-HOUGAARD COPULA MODEL

Consider the dependence function of the form

$$A(z_1, z_2; \lambda) = (z_1^\lambda + z_2^\lambda)^{1/\lambda}, \tag{54}$$

where $\lambda \geq 1$. The related extreme value copula is given by

$$C(u_1, u_2; \lambda) = \exp\{-((-\log u_1)^\lambda + (-\log u_2)^\lambda)^{1/\lambda}\}. \tag{55}$$

This copula family is known as *Gumbel-Hougaard Copula*. It is one of the earliest multivariate extreme value copula models. The copula density of this family is given by

$$\begin{aligned} c(u_1, u_2; \lambda) &= \frac{C(u_1, u_2; \lambda)}{u_1 u_2} \frac{(\log u_1 \ \log u_2)^{\lambda-1}}{[(-\log u_1)^\lambda + (-\log u_2)^\lambda]^{(1-1/\lambda)}} \\ &\quad \times \{[(-\log u_1)^\lambda + (-\log u_2)^\lambda]^{-1/\lambda} + \lambda - 1\}. \end{aligned} \tag{56}$$

A value of $\lambda = 1$ leads to the independence, in which the dependence function becomes $A(z_1, z_2) = z_1 + z_2$. Fréchet upper bound is obtained by letting λ go to ∞, in which case the dependence function becomes $A(z_1, z_2) = \max(z_1, z_2)$. This family can easily be extended to multivariate case with different forms of dependency structure. For example, we can consider the dependency function of the form $A(z_1, \ldots, z_d) = (z_1^\lambda + \cdots + z_d^\lambda)^{1/\lambda}$ that has a single dependency parameter λ. This could be used to generate a copula model with exchangeable correlation between alternatives. We can also consider other dependency functions that allow clustering between alternatives. Using the properties of MSMVE distributions, we obtain the closed form expressions for choice probabilities in the next section.

5.4.2 HUSLER-REISS COPULA MODEL

Consider the dependency function of the form

$$A(z_1, z_2; \lambda) = z_1 \Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(z_1/z_2)\right) + z_2 \Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(z_2/z_1)\right) \tag{57}$$

for $\lambda \geq 0$. The bivariate extreme value copula with dependency function (57) is given by

$$\begin{aligned} C(u_1, u_2; \lambda) &= \exp\left\{-(\log u_1)\, \Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log\left(\frac{\log u_1}{\log u_2}\right)\right)\right. \\ &\quad \left. -(\log u_2)\, \Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\ \log\left(\frac{\log u_2}{\log u_1}\right)\right)\right\}, \end{aligned} \tag{58}$$

where Φ denoted the standard normal distribution function. This copula family was introduced by Husler and Reiss (1989) and it is known as *Husler-Reiss Copula*. This copula is obtained as a limiting form of bivariate Gaussian copula, assuming the dependency correlation ρ converges to 1. The copula density of this family is given by

$$\begin{aligned} c(u_1,u_2;\lambda) &= \frac{C(u_1,u_2;\lambda)}{u_1u_2}\left[\Phi\left(\frac{1}{\lambda}+\frac{\lambda}{2}\log\left(\frac{\log u_1}{\log u_2}\right)\right)\Phi\left(\frac{1}{\lambda}+\frac{\lambda}{2}\log\left(\frac{\log u_2}{\log u_1}\right)\right)\right.\\ &\qquad \left.-\frac{\lambda}{2\log u_2}\phi\left(\frac{1}{\lambda}+\frac{\lambda}{2}\log\left(\frac{\log u_2}{\log u_1}\right)\right)\right] \end{aligned} \tag{59}$$

where ϕ is the standard normal density. When $\lambda = 0$, this copula becomes independence copula and when $\lambda \to \infty$ it attains the Fréchet upper bound. The multivariate extension of this dependency function is closed under margins and dependency parameter for pair (j_1, j_2) is same as for the pair (j_2, j_1). The dependency structure is similar to that of multivariate normal distribution. This model is an extreme value model with normal margins, not yet explored to analyze choice behaviors in the literature. The dependency function of the multivariate case can be written in a recursive form as

$$\begin{aligned} A_{1\ldots m}(\mathbf{z},\lambda_{12},\ldots,\lambda_{1,m}) &= \\ &A_{1\ldots m-1}((z_1,\ldots,z_{m-1}),\lambda_{12},\ldots,\lambda_{m-2,m-1}) + B(z_1,\ldots z_{m-1}) \end{aligned} \tag{60}$$

where

$$B(z_1,\ldots z_{m-1}) = \int_0^{z_m} \bar{\Phi}_{m-1}\left(\frac{1}{\lambda_{j,m}}+\frac{\lambda_{j,m}}{2}\left[\log(\frac{x}{z_j})\right], j \le m-1; [\rho_{mjk}]_{j<k<m-1}\right)dx.$$

See Joe (1997) for details.

5.4.3 COMPUTATION OF CHOICE PROBABILITIES

In this section, we illustrate the computation of choice probabilities for the extreme value copula models. MEV distributions obtained from an extreme value copula have the MSMVE property and hence dependency function A is of the form $-\log S$. Further, the function A is homogeneous of order 1. We exploit these properties to show that the choice probabilities are in a closed form.

Let $\mathbf{U} = (U_1, \ldots, U_c)$ be a random vector of c random variables, where U_j denote the utility associated with jth alternative in a choice model with c choices. Further, assume that $\mathbf{U}$ has MEV distribution obtained from an extreme value copula with dependency function $A(\mathbf{u})$. Since an extreme value copula has dependency function of the form $-\log S$, where S denotes survival function of MEV distribution, we have $S(\mathbf{u}) = \exp(-A(\mathbf{u}))$. Further, let $A_j(\mathbf{U})$ denote the partial derivative of A with respect to u_j. Then we have,

$$\begin{aligned} \frac{\partial S(\mathbf{u})}{\partial u_j} &= -e^{A(\mathbf{u})}\frac{\partial A(\mathbf{u})}{\partial u_j} \\ &= -S(\mathbf{u})A_j(\mathbf{u}) \end{aligned}$$

Now using the property of homogeneity we have for $u_1 > 0, A(u_1, \ldots, u_c) = u_1 A(1, u_2/u_1, \ldots, u_c/u_1)$. Hence,

$$\frac{\partial S(\mathbf{u})}{\partial u_1} = -S(\mathbf{u})\left[A(1, u_2/u_1, \ldots, u_c/u_1) - \sum_{k=2}^{c}(u_k/u_1)A_k(1, u_2/u_1, \ldots, u_c/u_1)\right]$$

For $j = 1$, comparing above two equations yields,

$$A_1(\mathbf{u}) = A(\mathbf{u}/u_1) - \sum_{k=2}^{c}(u_k/u_1)A_k(\mathbf{u}/u_1).$$

Thus $A_1(\mathbf{u})$ only depends on the ratios u_k/u_1. In a similar way, $A_j(\mathbf{u})$ only depends on the ratios $u_k/u_j, k(\neq j) = 1, \ldots, c$. Now consider the conditional survival function

$$\begin{aligned} Pr(U_k > u_k,\ k \neq j | U_j = u_j) &= \frac{Pr(U_k > u_k, k \neq j \text{ and } U_j = u_j)}{Pr(U_j = u_j)} \\ &= -\frac{\partial S(\mathbf{u})}{\partial u_j}\frac{1}{e^{-u_j}} \\ &= e^{u_j}\, S(\mathbf{u})\, A_j, \qquad j = 1, \ldots, c. \end{aligned}$$

Now, the survival function of minimum denoted by $U_{(1)} = \min(U_1, \ldots, U_m)$ such that $U_{(1)} = U_j$ is given by

$$\begin{aligned} Pr(U_{(1)} > t, U_{(1)} = U_j) &= \int_t^\infty e^{-A(x,\ldots,x)}A_j(x, \ldots, x)\, dx \\ &= \int_t^\infty e^{-xA(1,\ldots,1)}A_j(1, \ldots, 1)\, dx \\ &= A_j(1, \ldots, 1)\int_t^\infty e^{-xA(1,\ldots,1)}\, dx \\ &= \frac{A_j(1, \ldots, 1)}{A(1, \ldots, 1)}\, e^{-t\,A(1,\ldots,1)} \qquad (61) \end{aligned}$$

Therefore, $\sum_{j=1}^{c} Pr(U_{(1)} > t, U_{(1)} = U_j) = Pr(U_{(1)} > t) = e^{-t\,A(1,\ldots,1)}$, since $\sum_{j=1}^{c} A_j(1,\ldots,1) = A(1,\ldots,1)$. Thus the survival probability of minimum is in a closed form. We will exploit this property to obtain the choice probabilities in closed form using the survival function.

Let $\mathbf{Z}$ denote the random vector of unobserved utility components that follow MEV distribution $F(\mathbf{z}) = \exp\{-A(e^{-z_1},\ldots,e^{-z_c})\}$, obtained from an extreme value copula with dependency function A of the form $-\log S$. Let U_j be the total utility of jth choice alternative that is sum of mean μ_j and the unobserved component $Z_j, j = 1,\ldots,c$. The jth option is selected if $U_j > U_k$ for all $k \neq j$. Therefore, the choice probability of 1st alternative being chosen is

$$\begin{aligned} p_1 = Pr(U_1 > U_k; k = 2,\ldots,c) &= Pr(Z_1 > Z_k + (\mu_k - \mu_1), k = 2,\ldots,c) \\ &= Pr(Z_k < Z_1 - (\mu_k - \mu_1), k = 2,\ldots,c) \end{aligned}$$

To write this probability in terms of survival function, let $W_k = e^{-Z_k}$ and $w_k = e^{\mu_k}$. Then we have

$$\begin{aligned} p_1 &= Pr(Z_k < Z_1 - (\mu_k - \mu_1), k = 2,\ldots,c) \\ &= Pr\left(e^{-Z_k} > e^{-Z_1} e^{(\mu_k - \mu_1)}, k = 2,\ldots,c\right) \\ &= Pr\left(W_k > \frac{w_k\, W_1}{w_1},\ k = 2,\ldots,c\right) \end{aligned}$$

The range of Z_j is from 0 to ∞. Using the properties of MSMVE distributions,

$$\begin{aligned} p_1 &= \int_0^\infty e^{-A(z, zw_2/w_1,\ldots,zw_c/w_1)} A_1(1, w_2/w_1,\ldots,w_c/w_1)\,dz \\ &= \frac{A_1(1, w_2/w_1,\ldots,w_c/w_1)}{A(1, w_2/w_1,\ldots,w_c/w_1)} \\ &= \frac{A_1(w_1, w_2,\ldots,w_c)}{(1/w_1)A(w_1, w_2,\ldots,w_c)} \\ &= \frac{e^{\mu_1} A_1(e^{\mu_1},\ldots,e^{\mu_c})}{A(e^{\mu_1},\ldots,e^{\mu_c})}. \end{aligned}$$

Similarly, the choice probability for jth option is in a closed form given by

$$p_j = \frac{e^{\mu_j} A_j(e^{\mu_1},\ldots,e^{\mu_c})}{A(e^{\mu_1},\ldots,e^{\mu_c})} \tag{62}$$

Therefore, the models derived from max-stable MEV copulas are convenient in that closed form expressions are obtained for the choice probabilities. This is exactly the procedure McFadden (1978) described to generate GEV models (see Section 4.1). Further, we can assume a regression framework for μ_j as a function of covariates and estimate the regression parameters using likelihood estimation methods. We present few examples in the following section.

5.4.4 EXAMPLES

In this section, first we present examples of Gumbel-Hougaard Copula Model with various dependency functions.

Example 5.4 Complete Independence: Let $A(\mathbf{z}) = z_1 + \cdots + z_c$. Then $A_j(\mathbf{z}) = 1$ and the choice probability becomes

$$p_j = \frac{e^{\mu_j}}{\sum_{j=1}^{m} e^{\mu_j}}.$$

This is the conditional logit model with an assumption that the unobserved factors are independent.

Example 5.5 Equicorrelation: Let $A(\mathbf{z}) = (z_1^\theta + \cdots + z_c^\theta)^{1/\theta}, \theta \geq 1$. Then $A_j(\mathbf{z}) = (z_1^\theta + \cdots + z_c^\theta)^{(1/\theta)-1} z_j^{(\theta-1)}$ and the choice probability becomes

$$p_j = \frac{e^{\theta\mu_j}}{\sum_{j=1}^{c} e^{\theta\mu_j}}.$$

This is the logit model with equicorrelation dependency structure between unobserved factors.

Example 5.6 Nested Structure: Let $A(z_1, z_2, z_3) = ((z_1^\theta + z_2^\theta)^{\delta/\theta} + z_3^\delta)^{1/\delta}, 1 \leq \delta \leq \theta$. Such a dependency function has alternatives $\{1, 2\}$ as one nest and alternative 3 forms a different nest with one alternative. Then the partial derivatives are

$$\begin{aligned} A_j &= [(z_1^\theta + z_2^\theta)^{\delta/\theta} + z_3^\delta]^{(1/\delta)-1} (z_1^\theta + z_2^\theta)^{(\delta/\theta)-1} z_j^{\theta-1} \text{ for } j = 1, 2 \\ A_3 &= [(z_1^\theta + z_2^\theta)^{\delta/\theta} + z_3^\delta]^{(1/\delta)-1} z_3^{\delta-1} \end{aligned}$$

and the choice probabilities are

$$\begin{aligned} p_j &= \frac{w_1^\theta}{(w_1^\theta + w_2^\theta)} \frac{(w_1^\theta + w_2^\theta)^{\delta/\theta}}{((w_1^\theta + z_2^\theta)^{\delta/\theta} + w_3^\delta)} \text{ for } j = 1, 2 \\ p_3 &= \frac{w_1^\theta}{((w_1^\theta + z_2^\theta)^{\delta/\theta} + w_3^\delta)} \end{aligned}$$

where $w_j = e^{\mu_j}, j = 1, 2, 3$. This is a simple example of nested logit model. Complex models can be obtained with different dependency structures by using different dependency functions.

Now, we present few examples of Husler-Reiss copula model in bivariate case with a dependency function of the form (57). This model has dependency structure similar to the bivariate normal distribution with normal margins.

Example 5.7 Bivariate Husler-Reiss model: Let the dependency function is of the form (57). Then the partial derivatives $A_j(z_1, z_2; \lambda), j = 1, 2$ are

$$A_1(z_1, z_2; \lambda) = \Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{z_1}{z_2})\right) + \frac{\lambda z_2}{2z_1}\phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{z_1}{z_2})\right) - \frac{\lambda}{2}\phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{z_2}{z_1})\right),$$

and

$$A_2(z_1, z_2; \lambda) = \Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{z_2}{z_1})\right) + \frac{\lambda z_1}{2z_2}\phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{z_1}{z_2})\right) - \frac{\lambda}{2}\phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{z_1}{z_2})\right).$$

Therefore, the choice probabilities are given by

$$p_1 = \frac{e^{\mu_1}\Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}(\mu_1 - \mu_2)\right)}{e^{\mu_1}\Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}(\mu_1 - \mu_2)\right) + e^{\mu_2}\Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}(\mu_2 - \mu_1)\right)},$$

$$p_2 = \frac{e^{\mu_2}\Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}(\mu_2 - \mu_1)\right)}{e^{\mu_1}\Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}(\mu_1 - \mu_2)\right) + e^{\mu_2}\Phi\left(\frac{1}{\lambda} + \frac{\lambda}{2}(\mu_2 - \mu_1)\right)}.$$

For the multivariate case, we have the recursive relation of the dependency function as in (60). For $m = 3$ with a single dependency parameter λ, the dependency function in recursive form can be written as

$$A(z_1, z_2, z_3; \lambda) = A(z_1, z_2; \lambda) + \int_0^{z_3} \Phi_2\left(\frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{x}{z_1}), \frac{1}{\lambda} + \frac{\lambda}{2}\log(\frac{x}{z_2}); \frac{1}{2}\right) dx$$

and the choice probabilities can be obtained in a similar way. With multivariate extension, this model is a multivariate extreme value model with equicorrelation dependency structure. Other complex models can be obtained using the recursive relation and by imposing a dependency structure to reduce the number of dependency parameters.

5.5 FINAL REMARKS

To summarize, a more general form of choice models are presented using multivariate copulas. We presented a brief introduction of discrete choice copula models using Gaussian copula and Extreme value copulas. Copula representations are useful in building multivariate distributions with several choices for marginals. The multinomial probit models are Gaussian copula models with marginals that are standard normal and the GEV models are extreme value copula models with marginals that are extreme value distributions. This generalization shows a way of constructing new models using copulas by choosing different marginals within the copula representation. For example, a Gaussian copula choice model with Gumbel marginals or an Extreme value copula choice model with normal marginals are possible. Such models are not yet explored to model choice behavior and this provides a road map to future research.

CHAPTER 6

SUMMARY

Discrete choice models are very popular in Economics to model consumer choice behavior and the conditional logit model is the most widely used model. We first introduced this well known conditional logit model with IIA assumption and explained how the failure of such an assumption lead to incorrectly specified models using a numerical example. We presented an overview of existing models in the literature that relax IIA assumption such as GEV models. However, they are limited to handle different phenomenon that occur in consumer choice behavior. To overcome these limitations, we introduced the discrete choice probit models. Though they are flexible, they involve difficult computation of multivariate normal distribution function to compute choice probabilities.

In this dissertation, we presented discrete choice probit models for two correlation structures namely equicorrelation and product correlation. We derived exact analytical expressions for the computation of choice probabilities for both structures using stochastic representations. Further, we described the procedure of obtaining maximum likelihood estimates for the model parameters and derived analytical expressions for Fisher information matrix to compute their standard errors. Using simulations, we compared the performance of probit models with logit models in both large sample case as well as small samples. The results show that the probit models are efficient over logit models in both cases as correlation increases. We provided Sample R-code that performs all computations in the appendix.

Finally, a unified approach combining logit and probit models is presented using multivariate copulas. Copula representations are useful in building multivariate distributions with several choices for marginals. First we introduced discrete choice copula models using Gaussian copula and Extreme value copula. We showed that the discrete choice probit models are Gaussian Copula models with marginals that are standard normal and the GEV models are Extreme Value Copula models with marginals that are extreme value distributions. This insight shows a way of constructing new models using copulas by choosing different marginals within the copula representation. For example, a Gaussian Copula choice model with Gumbel marginals or an Extreme Value Copula choice model with normal marginals are possible. Such models are not yet explored to model consumer choice behavior and it leaves a lot of potential for future research.

REFERENCES

Agresti, A. (1990). *Categorical Data Analysis.* John Wiley and Sons, New York.

Ben-Akiva, M. and Bierlaire, M. (1999), "Discrete choice methods and their applications in short term travel decisions", in *The Handbook of Transportation Science*, eds. R. Hall, Kluwer, Dordrecht, pp. 533.

Bhat, C. (1995), "A heteroscedastic extreme value model of intercity mode choice", *Transportation Research*, B 29, 471-483.

Bierlaire, M. (1998), "Discrete choice models", in *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, eds. M. Labbe, G. Laporte, K. Tanczos, and P. Toint, Springer, Heidelberg, pp. 203-227.

Bolduc, D. (1992), "Generalized autoregressive errors: The multinomial probit model", *Transportation Research*, B 26, 155-170.

Bolduc, D., Fortin, B. and Fournier, M. (1996), "The impact of incentive policies on the practice location of doctors: A multinomial probit analysis", *Journal of Labor Economics*, 14, 703-732.

Chintagunta, P. K., and Prasad, A. R. (1998). "An Empirical Investigation of the Dynamic McFadden Model of Purchase Timing and Brand Choice: Implications for Market Structure" *Journal of Business and Economic Statistics*, 1, pp.2-12.

Chipman, J. (1960). "The foundations of utility". *Econometrica*, 28, 193-224.

Chu, C. (1989), "A paired combinational logit model for travel demand analysis", *Proceedings of Fifth World Conference on Transportation Research*, 4, 295-309.

Daganzo, C. (1979). *Multinomial Probit: The Theory and Its Application to Demand Forecasting*, Academic Press: New York.

Debreu, G. (1960). "Review of R.D. Luce individual choice behavior", *American Economic Review*, 50, 186-188.

Daly, A., and Zachary S. (1978), Improved multiple choice models, in D. Hensher and M. Dalvi, eds., *Determinants of Travel Choice*, Saxon House, Sussex.

Dunnett, C. W. (1989). "Algorithm AS251: Multivariate normal probability integrals with product correlation structure", *Applied Statistics*, 38, 564-579.

Elrod, T., and Keane, M. (1995), "A factor analytic probit model for representing the market structure in panel data", *Journal of Marketing Research*, 32, 116.

Galambos, J. (1987), *The Asymptotic Theory of Extreme Order Statistics*, Second edition. Kreiger Publishing Co., Malabar, FL.

Genz, A. (1992). "Numerical Computation of Multivariate Normal Probabilities", *J. Computational and Graphical Statist.*, 1, 141-149.

Greene, W. H. (2003). *Econometric Analysis*, 5th edition Upper Saddle River NJ: Prentice Hall.

Gumbel, E. J. (1960) "Bivariate exponential distributions", *Journal of American Statistics Association*, 55, 698-707.

Gudendorf, G., and Segers, J. (2009), "Extreme-Value Copulas", *Copula Theory and its Application*. Proceedings of the Workshop Held in Warsaw, Jaworski, et al. (Eds).

Haaijer, M., Wedel M., Vriens M., and Wansbeek T. (1998), "Utility covariances and context effects in conjoint MNP models", *Marketing Science*, 17, 236-252.

Hajivassiliou, V., McFadden D., and Ruud P. (1996), "Simulation of multivariate normal rectangle probabilities and their derivatives: Theoretical and computational results", *Journal of Econometrics*, 72, 85-134.

Hausman, J., and McFadden, D. (1984), "Specification tests for the multinomial logit model", *Econometrica*, 52, 1219-1240.

Hausman, J., and Wise, D. (1978). "A conditional probit model for qualitative choice: Discrete decisions recognizing interdependence and heterogeneous preferences", *Econometrica*, 48, 403-429.

Herriges, J., and Kling, C. (1996), "Testing the consistency of nested logit models with utility maximization", *Economic Letters* 50, 33-39.

Hüsler, J., and Reiss, R. "Maxima of normal random vectors: Between independence and complete dependence", *Statistics and Probability Letters*, 7, 283-286.

Joe, H. (1997) *Multivariate Models and Dependence Concepts*, Monographs on Statistics and Applied Probability, Vol 73. Chapman & Hall, London.

Kling, C., and Herriges, J. (1995), "An empirical investigation of the consistency of nested logit models with utility maximization", *American Journal of Agricultural Economics*, 77, 875-884.

Koppelman, F., and Wen, C. (2000), "The paired combination logit model: Properties, estimation and application", *Transportation Research*, B 34, 75-89.

Lee, B. (1999), "Calling patterns and usage of residential toll service under self-selecting tariffs", *Journal of Regulatory Economics*, 16, 45-82.

Lee, G. C. (1988), *Market-share analysis: evaluating competitive marketing effectiveness*, Kluwer Academic Publishers, Boston.

Li, J., and Ouyanag, J. (2008). *A Modified Paired Combinatorial Logit Route Choice Model with Unified Parameter*, in *Logistics : The Emerging Frontiers of Transportation and Development in China*, Chengdu, China, pp. 3675-3680.

Luce, D. (1959). *Individual Choice Behavior*, John Wiley and Sons, New York.

Luce, D., and Suppes P. (1965). "Preferences, utility and subjective probability" in *Handbook of Mathematical Psychology*, eds. R. Luce, R. Bush, and E. Galanter, John Wiley and Sons, New York, pp. 249-410.

Marschak, J. (1960). "Binary choice constraints on random utility indications" in *Stanford Symposium on Mathematical Methods in the Social Sciences*, eds. K. Arrow, Stanford University Press, Stanford, CA, pp. 312-329.

McFadden, D. (1974). "Conditional logit analysis of qualitative choice behavior", in *Frontiers in Econometrics*, eds. P. Zarembka, Academic Press, New York, pp. 105-142.

McFadden, D. (1978), "Modeling the choice of residential location", in *Spatial Interaction Theory and Planning Models*, eds. A. Karlqvist, L. Lundqvist, F. Snickars, and J. Weibull, North-Holland, Amsterdam, pp. 75-96.

McFadden, D. (1987). "Regression-based specification tests for the multinomial logit model", *Journal of Econometrics* 34, 63-82.

Nelsen, R. B. (2006). *An introduction to Copulas, Second edition.* Springer serise in Statistics. Springer, New York.

Pickands, J. (1981) "Multivariate extreme value distributions", in *Proceedings of the 43rd session of the International Statistical Institute*, Vol. 2 (Buenos Aires, 1981), vol. 49, pp. 859-878, 894-902.

Recker, W. (1995), "Discrete choice with an oddball atlernative", *Transportation Research*, B 29, 207-211.

Sklar, A. (1959), "Fonctions de rpartition n dimensions et leurs marges", *Public Institute of Statisticis*, University of Paris, 8, 229-231.

Steckel, J., and Vanhonacker, W. (1988), "A heterogeneous conditional logit model of choice", *Journal of Business and Economic Statistics*, 6, 391-398.

Thurstone, L. (1927). "A law of comparative judgement", *Psychological Review* 34, 273-286.

Train, K. (2004). *Discrete Choice Modelling, with Simulations*, Cambridge University Press.

Tong Y. L. (1990). *The Multivariate Normal Distribution*, Springer-Verlag, New York.

Train, K., Ben-Akiva, M., and Atherton, T. (1989), "Consumption patterns and selfselecting tariffs", *Review of Economics and Statistics*, 71, 62-73.

Train, K., McFadden, D., and Ben-Akiva, M. (1987a), "The demand for local telephone service: A fully discrete model of residential calling patterns and service choice", *Rand Journal of Economics*, 18, 109-123

Vovsha, P. (1997), "The cross-nested logit model: Application to mode choice in the Tel Aviv metropolitan area", Conference Presentation, *76th Transportation Research Board Meetings*, Washington, DC.

Williams, H. (1977), "On the formation of travel demand models and economic evaluation measures of user benefits", *Environment and Planning*, A 9, 285-344.

Wen, C.-H. and Koppelman, F. (2001), "The generalized nested logit model", *Transportation Research*, B 35, 627-641.

Yai, T., Iwakura S., and Morichi S. (1997), "Multinomial probit with structured covariance for route choice behavior", *Transporation Research*, B 31, 195-207.

APPENDIX

Here we provide two set of R-programs that perform computations of asymptotic efficiency, small sample efficiency and maximum likelihood estimation for Conditional Logit and MDCP with equicorrelation models, Paired Combinatorial Logit and MDCP with Product Correlation models.

Conditional Logit and MDCP Equicorrelation

```
#libraries needed for this program to run
library(MASS)
library(numDeriv)
library(mnormt)
library(mvtnorm)
library(MNP)
library(VGAM)
#*****************************************************************************#
#  Functions needed for computations such as col product, integrands of likelihood,#
#   first derivatives w r t rho, w r t beta etc                               #
#*****************************************************************************#
#function to compute column product of a matrix
colprod.matrix=function(x) {
a=x[,1]
  for(i in 2:dim(x)[2])
    a=a*x[,i]
    a = matrix(a, nrow(x), 1)
  return(a)
}
#function to compute column division of a matrix
coldiv.matrix <- function(x,y) {
z = x
  for(i in 1:dim(x)[2]) {
    z[,i]=x[,i]/y
  }
return(z)
}

#1.1 function to compute integrand for computing probailities
ProbIntegrand <- function(v, MuVec, rho, j) {
  product = 1/pnorm(v)
  for (l in 1:nChoice) {
  product = product*pnorm(v - (MuVec[l] - MuVec[j])/sqrt(1-rho))
  }
ProbInteg = product*exp(-v*v/2)/sqrt(2*pi)
return(ProbInteg)
}

#2.1 function to compute integrand for first derivatives w r t beta
FirstDerBetaIntegrand <- function(v, MuVec, rho, xDiff, j, m) {
  SumBeta = 0
  for (k in 1:nChoice) {
    InnProd = 1/pnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))
```

```
    for (l in 1:nChoice) {
    InnProd = InnProd*pnorm(v - (MuVec[l] - MuVec[j])/sqrt(1-rho))
    }
    SumBeta = SumBeta + InnProd*dnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*
        (xDiff[(m-1)*nChoice+k]-xDiff[(m-1)*nChoice+j])/sqrt(1-rho)
 }
SumBeta = SumBeta/pnorm(v)
FirstDerInteg = SumBeta*exp(-v*v/2)/sqrt(2*pi)
return(FirstDerInteg)
}

#2.2 function to compute integrand for first derivatives w r t Rho
FirstDerRhoIntegrand <- function(v, MuVec, rho, j) {
  SumRho = 0
  for (k in 1:nChoice) {
    InnProd = 1/pnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))
    for (l in 1:nChoice) {
    InnProd = InnProd*pnorm(v - (MuVec[l] - MuVec[j])/sqrt(1-rho))
    }
    SumRho = SumRho + InnProd*dnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*
        (MuVec[k] - MuVec[j])/(2*(1-rho)^(1.5))
  }
SumRho = SumRho/pnorm(v)
FirstDerRhoInteg = SumRho*exp(-v*v/2)/sqrt(2*pi)
return(FirstDerRhoInteg)
}

#3.1 function to compute integrand for second derivatives w r t betaM betaM'
# (when m = m', we get second derivatives for same parameter)
SecondDerBetaMMpmIntegrand <- function(v, MuVec, rho, xDiff, j, m, mpm) {
  InnProd = 1
    for (l in 1:nChoice) {
    InnProd = InnProd*pnorm(v - (MuVec[l] - MuVec[j])/sqrt(1-rho))
    }
  SumBeta = 0
  for (k in 1:nChoice) {
  SumBeta2 = 0
  for (kpm in 1:nChoice) {
    SumBeta2 = SumBeta2 + dnorm(v - (MuVec[kpm] - MuVec[j])/sqrt(1-rho))/pnorm(v - (
        MuVec[kpm] - MuVec[j])/sqrt(1-rho))*(xDiff[(mpm-1)*nChoice+kpm]-xDiff[(mpm
        -1)*nChoice+j])
    }
    SumBeta3 = -SumBeta2 + (v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*(xDiff[(mpm-1)*
        nChoice+k]-xDiff[(mpm-1)*nChoice+j]) + dnorm(v - (MuVec[k] - MuVec[j])/sqrt
        (1-rho))/pnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*(xDiff[(mpm-1)*nChoice
        +k]-xDiff[(mpm-1)*nChoice+j])
    SumBeta = SumBeta + InnProd*SumBeta3*dnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho)
        )*(xDiff[(m-1)*nChoice+k]-xDiff[(m-1)*nChoice+j])/(pnorm(v - (MuVec[k] -
        MuVec[j])/sqrt(1-rho))*(1-rho))
 }
SumBeta = -SumBeta/pnorm(v)
SecondDerIntegBetaMMpm = SumBeta*exp(-v*v/2)/sqrt(2*pi)
```

```
return(SecondDerIntegBetaMMpm)
}
#3.2 function to compute integrand for second derivatives w r t betaM and rho
SecondDerRhoBetaMIntegrand <- function(v, MuVec, rho, xDiff, j, m) {
  InnProd = 1
    for (l in 1:nChoice) {
    InnProd = InnProd*pnorm(v - (MuVec[l] - MuVec[j])/sqrt(1-rho))
    }
  SumBeta = 0
  for (k in 1:nChoice) {
  SumBeta2 = 0
  for (kpm in 1:nChoice) {
    SumBeta2 = SumBeta2 + dnorm(v - (MuVec[kpm] - MuVec[j])/sqrt(1-rho))/pnorm(v - (
        MuVec[kpm] - MuVec[j])/sqrt(1-rho))*(MuVec[kpm] - MuVec[j])
    }
    SumBeta3 = -SumBeta2 + (v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*(MuVec[k] - MuVec
        [j]) + dnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))/pnorm(v - (MuVec[k] -
        MuVec[j])/sqrt(1-rho))*(MuVec[k] - MuVec[j]) + sqrt(1-rho)
    SumBeta = SumBeta + InnProd*SumBeta3*dnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho)
        )*(xDiff[(m-1)*nChoice+k]-xDiff[(m-1)*nChoice+j])/(pnorm(v - (MuVec[k] -
        MuVec[j])/sqrt(1-rho))*2*(1-rho)^2)
 }
SumBeta = -SumBeta/pnorm(v)
SecondDerIntegBetaMRho = SumBeta*exp(-v*v/2)/sqrt(2*pi)
return(SecondDerIntegBetaMRho)
}
#3.3 function to compute integrand for second derivatives w r t rho
SecondDerRho2Integrand <- function(v, MuVec, rho, xDiff, j) {
  InnProd = 1
    for (l in 1:nChoice) {
    InnProd = InnProd*pnorm(v - (MuVec[l] - MuVec[j])/sqrt(1-rho))
    }
  SumRho = 0
  for (k in 1:nChoice) {
  SumBeta2 = 0
  for (kpm in 1:nChoice) {
    SumBeta2 = SumBeta2 + dnorm(v - (MuVec[kpm] - MuVec[j])/sqrt(1-rho))/pnorm(v - (
        MuVec[kpm] - MuVec[j])/sqrt(1-rho))*(MuVec[kpm] - MuVec[j])
    }
    SumBeta3 = -SumBeta2 + (v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*(MuVec[k] - MuVec
        [j]) + dnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))/pnorm(v - (MuVec[k] -
        MuVec[j])/sqrt(1-rho))*(MuVec[k] - MuVec[j]) + 3*sqrt(1-rho)
    SumRho = SumRho + InnProd*SumBeta3*dnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*
        (MuVec[k] - MuVec[j])/(pnorm(v - (MuVec[k] - MuVec[j])/sqrt(1-rho))*4*(1-rho
        )^3)
 }
SumRho = -SumRho/pnorm(v)
SecondDerIntegRho2 = SumRho*exp(-v*v/2)/sqrt(2*pi)
return(SecondDerIntegRho2)
}
```

```
#**************************************************************************#
# Defining Probabilities , Derivatives , Double Derivatives for MDCP I model        #
#**************************************************************************#
#function to compute Equicorrelated Probit Probabilities
ProbMDCP <- function(Data, pars) {
  betainn = pars[1:nCovariates]
  rho = pars[(nCovariates+1)]
  xdata = Data[,2:(nCovariates+1)]
  means =  xdata%*%betainn
  SpMeans = matrix(0,nObs*nChoice, nChoice)
   for (i in 1:nObs) {
            SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice,1)%*
                %t(means[((i-1)*nChoice+1):(i*nChoice),1])
            }
  Prob = matrix(0,nObs*nChoice, 1)
  MuVec = matrix(0,nChoice,1)
  for (i in 1:nObs) {
    for (j in 1:nChoice) {
    MuVec = SpMeans[((i-1)*nChoice+j),1:nChoice]
    q1 = integrate(ProbIntegrand, lower = 0, upper = Inf, MuVec, rho, j)
    q2 = integrate(ProbIntegrand, lower = -20, upper = 0, MuVec, rho, j)
    Prob[((i-1)*nChoice+j),1]= q1$value + q2$value
    }
  }
return(Prob)
}
#function to compute Equicorrelated Probit derivatives
DerMDCP <- function(Data, pars) {
betainn = pars[1:nCovariates]
rho = pars[nCovariates+1]
xdata = Data[,2:(nCovariates+1)]
means =  xdata%*%betainn
SpMeans = matrix(0,nObs*nChoice, nChoice)
SpXs = matrix(0, nObs*nChoice, nChoice*nCovariates)
 for (i in 1:nObs) {
          SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice,1)%*%t
              (means[((i-1)*nChoice+1):(i*nChoice),1])
          rearrange = t(Data[((i-1)*nChoice+1):(i*nChoice),1+1])
          for (m in 2:nCovariates)  {
                rearrange = cbind(rearrange, t(Data[((i-1)*nChoice+1):(i*nChoice),m
                    +1]))
                }
         SpXs[((i-1)*nChoice+1):(i*nChoice),1:(nChoice*nCovariates)] = matrix(1,
             nChoice,1)%*%rearrange
        }
DerProbBeta = matrix(0, nObs*nChoice, nCovariates)
DerProbRho = matrix(0, nObs*nChoice, 1)
MuVec = matrix(0, nChoice,1)
for (i in 1:nObs) {
  for (j in 1:nChoice) {
  MuVec = SpMeans[((i-1)*nChoice+j),1:nChoice]
  xDiff = SpXs[((i-1)*nChoice+j),1:(nChoice*nCovariates)]
```

```
  for (m in 1:nCovariates) {
      drBetaM1 = integrate(FirstDerBetaIntegrand, lower = 0, upper = Inf, MuVec, rho
          , xDiff, j, m)
      drBetaM2 = integrate(FirstDerBetaIntegrand, lower = -10, upper = 0, MuVec, rho
          , xDiff, j, m)
      DerProbBeta[((i-1)*nChoice+j),m] = -drBetaM1$value - drBetaM2$value
    }
  drRho1 = integrate(FirstDerRhoIntegrand, lower = 0, upper = Inf, MuVec, rho, j)
  drRho2 = integrate(FirstDerRhoIntegrand, lower = -10, upper = 0, MuVec, rho, j)
  DerProbRho[((i-1)*nChoice+j),1] = -drRho1$value - drRho2$value
  }
}
return(cbind(DerProbBeta, DerProbRho))
}
#function to compute Equicorrelated Probit Hessian
MDCP.Hessian <- function(Data, pars) {
betain = pars[1:nCovariates]
rho =  pars[nCovariates+1]
xdata = Data[,2:(nCovariates+1)]
means =  xdata%*%betain
SpMeans = matrix(0,nObs*nChoice, nChoice)
SpXs = matrix(0, nObs*nChoice, nChoice*nCovariates)
 for (i in 1:nObs) {
         SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice,1)%*%t
             (means[((i-1)*nChoice+1):(i*nChoice),1])
         rearrange = t(Data[((i-1)*nChoice+1):(i*nChoice),1+1])
         for (m in 2:nCovariates)  {
               rearrange = cbind(rearrange, t(Data[((i-1)*nChoice+1):(i*nChoice),m
                   +1]))
               }
        SpXs[((i-1)*nChoice+1):(i*nChoice),1:(nChoice*nCovariates)] = matrix(1,
            nChoice,1)%*%rearrange
       }
Prob = ProbMDCP(Data, pars)
Der = DerMDCP(Data, pars)
DerProbBeta = Der[,1:nCovariates]
DerProbRho = as.matrix(Der[1:(nObs*nChoice),nPar])
DDerProbBeta = matrix(0, nObs*nChoice, nCovariates^2)
DDerProbRho = matrix(0, nObs*nChoice, nPar)
MuVec = matrix(0, nChoice,1)
for (i in 1:nObs) {
  for (j in 1:nChoice) {
  MuVec = SpMeans[((i-1)*nChoice+j),1:nChoice]
  xDiff = SpXs[((i-1)*nChoice+j),1:(nChoice*nCovariates)]
   for (m in 1:nCovariates) {
     for (mpm in 1:nCovariates) {
        ddrBetaM1 = integrate(SecondDerBetaMMpmIntegrand, lower = 0, upper = Inf,
            MuVec, rho, xDiff, j, m, mpm)
        ddrBetaM2 = integrate(SecondDerBetaMMpmIntegrand, lower = -10, upper = 0,
            MuVec, rho, xDiff, j, m, mpm)
        DDerProbBeta[((i-1)*nChoice+j),((m-1)*nCovariates+mpm)] = ((ddrBetaM1$value
            + ddrBetaM2$value)
```

```
              - (DerProbBeta[((i-1)*nChoice+j),m]*DerProbBeta[((i-1)*nChoice+j),mpm
                  ])/Prob[((i-1)*nChoice+j),1])
      }
    }
   for (m in 1:nCovariates) {
   ddrBetaMRho1 = integrate(SecondDerRhoBetaMIntegrand, lower = 0, upper = Inf,
       MuVec, rho, xDiff, j, m)
   ddrBetaMRho2 = integrate(SecondDerRhoBetaMIntegrand, lower = -10, upper = 0,
       MuVec, rho, xDiff, j, m)
   DDerProbRho[((i-1)*nChoice+j),m] = ((ddrBetaMRho1$value + ddrBetaMRho2$value)
              - (DerProbBeta[((i-1)*nChoice+j),m]/Prob[((i-1)*nChoice+j),1])*(
                  DerProbRho[((i-1)*nChoice+j),1]))
    }
 ddrRho1 = integrate(SecondDerRho2Integrand, lower = 0, upper = Inf, MuVec, rho,
     xDiff, j)
 ddrRho2 = integrate(SecondDerRho2Integrand, lower = -10, upper = 0, MuVec, rho,
     xDiff, j)
 DDerProbRho[((i-1)*nChoice+j),nPar] = ((ddrRho1$value + ddrRho2$value)
              - (DerProbRho[((i-1)*nChoice+j),1]/Prob[((i-1)*nChoice+j),1]))
  }
}
MDCPHess = matrix(0, nPar, nPar)
DDerBeta = apply(DDerProbBeta, 2, sum)
DDerRho = apply(DDerProbRho, 2, sum)
MDCPHess[1:nCovariates, 1:nCovariates] = matrix(DDerBeta, nCovariates, nCovariates)
MDCPHess[nPar,1:nPar] = matrix(DDerRho, 1, nPar)
MDCPHess[1:nPar,nPar] = matrix(DDerRho, nPar, 1)
return(MDCPHess)
}
#*****************************************************************************#
# Defining Probabilities, Derivatives, Double Derivatives for CNL model       #
#*****************************************************************************#
# function to compute Condtional logit Probabilities
ProbCNL <- function(Data, pars) {
betainn = pars[1:nCovariates]
xdata = Data[,2:(nCovariates+1)]
means =  xdata%*%betainn
SpMeans = matrix(0,nObs*nChoice, nChoice)
 for (i in 1:nObs) {
          SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice,1)%*%t
              (means[((i-1)*nChoice+1):(i*nChoice),1])
          }
Prob = exp(means)/apply(exp(SpMeans),1,sum)
return(Prob)
}

# function to compute Condtional logit derivatives
DerCNL <- function(Data, pars) {
Prob = ProbCNL(Data, pars)
xdata = Data[,2:(nCovariates+1)]
DerProbBeta = (Data[,1]-Prob[,1])*xdata
DerProbBeta1 = cbind(DerProbBeta, 0)
```

```
return(DerProbBeta1)
}

#Hessian for Conditional Logit
CNL.Hessian <- function(Data, pars) {
betainn = pars[1:nCovariates]
xdata = Data[,2:(nCovariates+1)]
means =  xdata%*%betainn
SpMeans = matrix(0,nObs*nChoice , nChoice)
 for (i in 1:nObs) {
          SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice ,1)%*%t
              (means[((i-1)*nChoice+1):(i*nChoice),1])
          }
Prob = exp(means)/apply(exp(SpMeans),1,sum)
CNLHess = matrix(0, nCovariates, nCovariates)
for (m in 1:nCovariates) {
  for (mpm in 1:nCovariates) {
  SpXs = matrix(0,nObs*nChoice , nChoice)
    for (i in 1:nObs) {
          SpXs[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice ,1)%*%t(
              xdata[((i-1)*nChoice+1):(i*nChoice),mpm])
          }
  CNLHess[m, mpm] = sum(Prob[,1]*(xdata[,mpm] - apply(SpXs*exp(SpMeans), 1, sum)/
      apply(exp(SpMeans), 1, sum))*xdata[,m])
  }
}
return(CNLHess)
}
#****************************************************************************#
# Defining likelihood, Gradient, Hessian for equicorrelated probit model      #
#****************************************************************************#
#Defining likelihood for equi-correlated probit model.
MDCP.Likelihood <- function(Data, pars) {
Prob = ProbMDCP(Data, pars)
lik = Data[,1]*log(pmax(1e-323,Prob))
loglike = sum(lik)
return(loglike)
}

#Defining gradient for equi-correlated probit model
MDCP.Gradient <- function(Data, pars) {
Prob = ProbMDCP(Data, pars)
Der = DerMDCP(Data, pars)
Grad = matrix(0, nPar,1)
Grd = Data[,1]*(coldiv.matrix(Der,Prob));
Grad = apply(Grd,2,sum)
return(Grad);
}
```

```
#*****************************************************************************#
# Defining likelihood , Gradient , Hessian for Conditional Logit model        #
#*****************************************************************************#
#Defining likelihood for Conditional logit model.
CNL.Likelihood <- function(Data, pars) {
betainn = pars[1:nCovariates]
xdata = Data[,2:(nCovariates+1)]
means =  xdata%*%betainn
SpMeans = matrix(0,nObs*nChoice , nChoice)
 for (i in 1:nObs) {
         SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice ,1)%*%t
             (means[((i-1)*nChoice+1):(i*nChoice),1])
         }
Prob = exp(means)/apply(exp(SpMeans),1,sum)
lik = Data[,1]*log(pmax(1e-323,Prob))
loglike = sum(lik)
return(loglike)
}

#Defining gradient for Conditional logit model.
CNL.Gradient <- function(Data, pars) {
betainn = pars[1:nCovariates]
xdata = Data[,2:(nCovariates+1)]
means =  xdata%*%betainn
SpMeans = matrix(0,nObs*nChoice , nChoice)
 for (i in 1:nObs) {
         SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice ,1)%*%t
             (means[((i-1)*nChoice+1):(i*nChoice),1])
         }
Prob = exp(means)/apply(exp(SpMeans),1,sum)
DerProbBeta = apply((Data[,1]-Prob[,1])*xdata, 2, sum)
DerProbBeta1 = c(DerProbBeta, 0)
return(DerProbBeta1)
}

#*****************************************************************************#
#*  Simulating data for computation of asymptotic efficiency                 *#
#*****************************************************************************#
DataSim <- function(seed, nObs, nChoice , nLevel , StartBeta, rho) {
set.seed(seed)
intmat = rbind(diag(nChoice -1), matrix(0, 1, nChoice -1))
xInt = intmat
  for (i in 1:(nObs-1)) {
      xInt = rbind(xInt, intmat)
  }
  ix = sample(c(3,4), nChoice , prob = c(1/2, 1/2), replace = TRUE)
  xCont = matrix(0, nObs, nChoice)
  xDisc = matrix(0, nObs, nChoice)
  xDiscProb = runif(nChoice*(nLevel-1), min = 0, max = 1/(nLevel-1))
  xDiscProp = matrix(0, nChoice , nLevel)
  xDiscProp[,1:(nLevel-1)] = matrix(xDiscProb, nChoice , nLevel-1)
  xDiscProp[,nLevel] = 1-apply(xDiscProp[,1:(nLevel-1)], 1, sum)
```

```
  for (i in 1:nChoice) {
   xContMean = runif(ix[i], min = 1.5, max = 4.5)
   xContSd    = c(runif(1, min = 0, max = 0.5), runif(ix[i]-1, min = 1, max = 2.5))
   xContBind = matrix(0, nObs, ix[i])
   xContBind[,1] = rnorm(nObs, mean = xContMean[1], sd = xContSd[1])
   for (j in 2:ix[i]) {
   xContBind[,j] = rnorm(nObs, mean = xContMean[j], sd = xContSd[j])
   }
   oneprob = runif(1, min = 0.5, max = 1)
   ixProb = c(oneprob, runif((ix[i]-2), min = 0, max = (1-oneprob)/(ix[i]-2)))
   ixCont = sample(seq(1:ix[i]), nObs, prob = c(ixProb, 1-sum(ixProb)), replace =
       TRUE)
   xCont[, i] =  xContBind[,1]*(ixCont==1)
   for (j in 2:ix[i]) {
     xCont[, i] =  xCont[, i] + xContBind[,j]*(ixCont==j)
   }
  xDisc[,i] = cut(runif(nObs, 0, 1), c(0, cumsum(xDiscProp[i,])), labels = seq(1:
      nLevel))
    }
  xDiscInd = matrix(0, nObs*nChoice, nLevel-1)
  xDisc1 = matrix(t(xDisc), nObs*nChoice, 1)
  for (j in 1:(nLevel-1)) {
    xDiscInd[,j] =  (xDisc1 == j)
  }
  xData = cbind(xInt, matrix(t(abs(xCont)), nObs*nChoice, 1), xDiscInd)
Mean = xData%*%StartBeta
Cov = (1-rho)*diag(nChoice) + rho*matrix(1, nChoice, nChoice)
u = mvrnorm(nObs, matrix(0, nChoice, 1), Cov)
MeanNew = matrix(Mean, nObs, nChoice, byrow = TRUE)
su = MeanNew + u
sumax = matrix(apply(su, 1, max),nObs, 1)
y = matrix(0, nObs, nChoice)
for (j in 1:nChoice) {
y[,j] = (su[,j] == sumax)
}
yData = matrix(t(y), nObs*nChoice, 1)
return(cbind(yData, xData))
}

#Computation of asymptotic efficiency for real market
asympeff <- function(Data, StartBeta, rho) {
  MDCP.Hess = MDCP.Hessian(Data, c(StartBeta, rho))
  CNL.Hess = CNL.Hessian(Data, c(StartBeta, rho))
  InvFishMDCP = solve(-MDCP.Hess)
  InvFishCNL = solve(-CNL.Hess)
  eff = diag(InvFishCNL)/diag(InvFishMDCP[1:nCovariates, 1:nCovariates])
return(eff)
}

#Input parameters for asymptotic efficiencies of real market
seed = 16461
nObs = 1000
```

```
nChoice = 4
nLevel = 3     #Number of levels for discrete covariate
nCovariates = nChoice+nLevel - 1    #Number of covariates such as intercepts,
    parameters
nPar = nCovariates + 1 #Number of parameters
Data = DataSim(seed, nObs, nChoice, nLevel, StartBeta, rho)
efficiency = matrix(0, 10, nCovariates)
for (i in 1:10) {
efficiency[i,] = asympeff(seed = 16461, StartBeta = c(-0.479,  1.051,  0.475,
    0.781,  0.107, -0.525), rho = (i-1)*0.1)
}

#Input parameters for asymptotic efficiencies of choice design
nObs = 900
nChoice = 4
nLevel = 3     #Number of levels for discrete covariate
nCovariates = nChoice+nLevel - 1    #Number of covariates such as intercepts,
    parameters
nPar = nCovariates + 1 #Number of parameters
seed = 16461
fabric <- read.table("C:/Users/bravi/Desktop/Bhaskar@ODU/Class Materials/Research/
    SAS code/Data sets/fabric softner.txt", sep="", header = FALSE)
fabricpric <- as.matrix(fabric[,4:7])
fabricpric1 = matrix(t(fabricpric), nObs*nChoice, 1)
Data = DataSim(seed, nObs=900, nChoice, nLevel, StartBeta, rho)
Data[, 5] = fabricpric1

efficiency = matrix(0, 10, nCovariates)
for (i in 1:10) {
efficiency[i,] = asympeff(seed = 16461, StartBeta = c(-0.479,  1.051,  0.475,
    0.781,  0.107, -0.525), rho = (i-1)*0.1)
}

#Data generation of small-sample efficiencies
xDataGen <- function(seed, nObs, nChoice, nLevel, StartBeta, rho) {
set.seed(seed)
intmat = rbind(diag(nChoice-1), matrix(0, 1, nChoice-1))
xInt = intmat
  for (i in 1:(nObs-1)) {
      xInt = rbind(xInt, intmat)
  }
  ix = sample(c(3,4), nChoice, prob = c(1/2, 1/2), replace = TRUE)
  xCont = matrix(0, nObs, nChoice)
  xDisc = matrix(0, nObs, nChoice)
  xDiscProb = runif(nChoice*(nLevel-1), min = 0, max = 1/(nLevel-1))
  xDiscProp = matrix(0, nChoice, nLevel)
  xDiscProp[,1:(nLevel-1)] = matrix(xDiscProb, nChoice, nLevel-1)
  xDiscProp[,nLevel] = 1-apply(xDiscProp[,1:(nLevel-1)], 1, sum)
  for (i in 1:nChoice) {
   xContMean = runif(ix[i], min = 1.5, max = 4.5)
   xContSd    = c(runif(1, min = 0, max = 0.5), runif(ix[i]-1, min = 1, max = 2.5))
   xContBind = matrix(0, nObs, ix[i])
```

```
    xContBind[,1] = rnorm(nObs, mean = xContMean[1], sd = xContSd[1])
    for (j in 2:ix[i]) {
    xContBind[,j] = rnorm(nObs, mean = xContMean[j], sd = xContSd[j])
    }
    oneprob = runif(1, min = 0.5, max = 1)
    ixProb = c(oneprob, runif((ix[i]-2), min = 0, max = (1-oneprob)/(ix[i]-2)))
    ixCont = sample(seq(1:ix[i]), nObs, prob = c(ixProb, 1-sum(ixProb)), replace =
        TRUE)
    xCont[, i] =  xContBind[,1]*(ixCont==1)
    for (j in 2:ix[i]) {
      xCont[, i] =  xCont[, i] + xContBind[,j]*(ixCont==j)
    }
  xDisc[,i] = cut(runif(nObs, 0, 1), c(0, cumsum(xDiscProp[i,])), labels = seq(1:
      nLevel))
    }
  xDiscInd = matrix(0, nObs*nChoice, nLevel-1)
  xDisc1 = matrix(t(xDisc), nObs*nChoice, 1)
  for (j in 1:(nLevel-1)) {
    xDiscInd[,j] =  (xDisc1 == j)
  }
  xData = cbind(xInt, matrix(t(abs(xCont)), nObs*nChoice, 1), xDiscInd)
return(xData)
}

xData = xDataGen(16461, nObs=1000, nChoice=4, nLevel=3, StartBeta, rho=0.8)
#function for small sample efficiency
smalleff <- function(xData, nObs, nChoice, nLevel, nSim, StartBeta, rho) {
count2 = matrix(0, nSim, 1)
count1 = matrix(0, nSim, 1)
for (i in 1:nSim) {
  Mean = xData%*%StartBeta
  u = rgumbel(nObs*nChoice, location = 0, scale = 1)
  MeanNew = matrix(Mean, nObs, nChoice, byrow = TRUE)
  unew = matrix(u, nObs, nChoice)
  su = MeanNew + unew
  sumax = matrix(apply(su, 1, max),nObs, 1)
  y = matrix(0, nObs, nChoice)
  for (j in 1:nChoice) {
  y[,j] = (su[,j] == sumax)
  }
  apply(y, 2, sum)
  yData = matrix(t(y), nObs*nChoice, 1)
  sampleD = cbind(yData, xData)
  initial = c(StartBeta + runif(nCovariates, min = -0.5, max = 0.5), runif(1, min =
      0, max = min(rho+0.2, 1)))
  sol.CNL = optim(initial, CNL.Likelihood, gr = CNL.Gradient, Data = sampleD, method
      ='L-BFGS-B', lower = c(rep(-50, nCovariates), -1/(nChoice-1)), upper = c(rep
      (50, nCovariates), 0.99), control=list(trace=6, fnscale = -1))
  if (sol.CNL$convergence == 51 | sol.CNL$convergence == 52) {
  count1[i,] = 1
  i = i-1 }
  else {
```

```
  BetaHatCNL[i, ] = sol.CNL$par
  Mean = xData%*%StartBeta
  Cov = (1-rho)*diag(nChoice) + rho*matrix(1, nChoice, nChoice)
  u = mvrnorm(nObs, matrix(0, nChoice, 1), Cov)
  MeanNew = matrix(Mean, nObs, nChoice, byrow = TRUE)
  su = MeanNew + u
  sumax = matrix(apply(su, 1, max),nObs, 1)
  y = matrix(0, nObs, nChoice)
  for (j in 1:nChoice) {
  y[,j] = (su[,j] == sumax)
  }
  yData = matrix(t(y), nObs*nChoice, 1)
  sampleD = cbind(yData, xData)
  sol.MDCP = optim(initial, MDCP.Likelihood, gr = MDCP.Gradient, Data = sampleD,
      method='L-BFGS-B', lower = c(rep(-Inf, nCovariates), -1/(nChoice-1)), upper =
      c(rep(Inf, nCovariates), 0.99), control=list(trace=6, fnscale = -1))
  if (sol.MDCP$message == 51 | sol.MDCP$message == 52) {
  count2[i,] = 1
  i = i-1 }
  else { BetaHatMDCP[i, ] = sol.MDCP$par}
  }
   }
eff = cbind(BetaHatCNL, BetaHatMDCP, count1, count2)
return(eff)
}

nSim = 1000
BetaHatCNL = matrix(0, nSim, nPar)
BetaHatMDCP = matrix(0, nSim, nPar)
eff1 = smalleff(xData, nObs=30, nChoice=4, nLevel=3, nSim=1000, StartBeta = c(0.479,
      1.051,  0.475,  0.781,  0.107, -0.525), rho = 0.5)

#Analysis of Laundry Detergent data
Laundry = read.table("C:/Users/bravi/Desktop/Bhaskar@ODU/Class Materials/Research/
    SAS code/Data sets/Laundry.txt", sep="", header = TRUE)
Laundry = as.matrix(Laundry, 2657, 13)

rho = 0.01
nChoice = 6
nCovariates = 11
nPar = nCovariates+1
nObs = 2657

Price = Laundry[1:nObs,2:7]
Select = Laundry[1:nObs,8:13]
PriceNew = matrix(t(Price), nObs*nChoice, 1)
SelectNew = matrix(t(Select), nObs*nChoice, 1)
intmat = rbind(diag(nChoice-1), matrix(0, 1, nChoice-1))
xInt = intmat
for (i in 1:(nObs-1)) {
xInt = rbind(xInt, intmat)
}
```

```
pInt = diag(nChoice)
priceInt = pInt
for (i in 1:(nObs-1)) {
priceInt = rbind(priceInt, pInt)
}
PriceN = matrix(0, nObs*nChoice, nChoice)
for (i in 1:nChoice) {
PriceN[,i] = PriceNew*priceInt[,i]
}
xData = cbind(xInt, PriceN)
yData = SelectNew
LaundryNew = cbind(yData, xData)
initial = c(2, 1, 1, 2, 1, rep(-120,nChoice) , 0.33)

sol.CNL = optim(initial, CNL.Likelihood, gr = CNL.Gradient, Data = sampleD, method='
    L-BFGS-B', control=list(trace=6, fnscale = -1, maxit = 1000))
sol.MDCP = constrOptim(initial, MDCP.Likelihood, gr=MDCP.Gradient, ui=rbind(c(rep(0,
    nCovariates), 1), c(rep(0,nCovariates), -1)), ci=rbind(-1/(nChoice-1), -1), mu =
     1e-06, control = list(fnscale=-1),
            method = "BFGS", outer.iterations = 100, outer.eps = 1e-05, Data=
                LaundryNew, hessian = FALSE)

MDCP.Hess = MDCP.Hessian(LaundryNew, sol.MDCP$par)
CNL.Hess = PCL.Hessian(LaundryNew, sol.CNL$par)

seMDCP = solve(-MDCP.Hess)
seCNL = solve(-CNL.Hess)

#Travel mode data
Travel = read.table("C:/Users/bravi/Desktop/Bhaskar@ODU/Class Materials/Research/SAS
     code/Data sets/Travel data.txt", sep="", header = TRUE)
nObs = 210
Travel = as.matrix(Travel, nObs, 7)
nChoice = 4
nCovariates = 7
nPar = nCovariates+1
intmat = rbind(diag(nChoice-1), matrix(0, 1, nChoice-1))
xInt = intmat
for (i in 1:(nObs-1)) {
xInt = rbind(xInt, intmat)
}
lower = c(rep(-Inf, nCovariates), -1/(nChoice-1)), upper = c(rep(Inf, nCovariates),
    0.99),
TravelNew = matrix(0, nObs*nChoice, nCovariates+1)
TravelNew[,1] = Travel[,1]
TravelNew[,2:4] = xInt
TravelNew[,5:8] = Travel[,2:5]
initial =  runif(nCovariates+1, min = -1, max = 1)
sol.CNL = optim(initial, CNL.Likelihood, gr = CNL.Gradient, Data = TravelNew, method
    ='L-BFGS-B', control=list(trace=6, fnscale = -1, maxit = 1000) )
```

```
sol.MDCP = constrOptim(initial , MDCP.Likelihood , gr=MDCP.Gradient , ui=rbind(c(rep(0,
    nCovariates), 1), c(rep(0,nCovariates), -1)), ci=rbind(-1/(nChoice-1), -1), mu =
     1e-06, control = list(fnscale=-1),
            method = "BFGS", outer.iterations = 100, outer.eps = 1e-05, Data=
                TravelNew, hessian = FALSE)

MDCP.Hess = MDCP.Hessian(TravelNew, sol.MDCP$par)
CNL.Hess = PCL.Hessian(LaundryNew, sol.CNL$par)

seMDCP = solve(-MDCP.Hess)
seCNL = solve(-CNL.Hess)
```

Paired Combinatorial Logit and MDCP Product Correlation

```
#libraries needed for this program to run#
library(MASS)
library(numDeriv)
library(mnormt)
library(mvtnorm)
library(MNP)
library(VGAM)
#*************************************************************************#
#*  Functions needed for computations such as col product, integrands of     *#
#   likelihood, first derivatives w r t rho, w r t beta etc                  *#
#*************************************************************************#
#function to compute column product of a matrix
colprod.matrix=function(x) {
a=x[,1]
  for(i in 2:dim(x)[2])
    a=a*x[,i]
    a = matrix(a, nrow(x), 1)
  return(a)
}
#function to compute column division of a matrix
coldiv.matrix <- function(x,y) {
z = x
  for(i in 1:dim(x)[2]) {
    z[,i]=x[,i]/y
  }
return(z)
}
#functions needed to compute first and second order partial derivatives.
t1 <- function(v,w,MuVec, lambda, l, j) {return( ((MuVec[j]-MuVec[l]) + (lambda[j]-
    lambda[l])*v + sqrt(1-lambda[j]^2)*w)/sqrt(1-lambda[l]^2)) }
t2 <- function(v,w,MuVec, lambda, l, j) {return( v/sqrt(1-lambda[l]^2) - lambda[j]*w
    /sqrt((1-lambda[l]^2)*(1-lambda[j]^2)))}
t3 <- function(v,w,MuVec, lambda, r, j) {return( ((MuVec[j]-MuVec[r])*lambda[r]-v*
    (1-lambda[r]^2)+lambda[r]*(lambda[j]-lambda[r])*v+lambda[r]*sqrt(1-lambda[j]^2)*
    w)/(1-lambda[r]^2)^(3/2) )}

d_ilm <- function(xDiff,k,j,m){return((xDiff[(m-1)*nChoice+k]-xDiff[(m-1)*nChoice+j
    ]))}
Al_theta_v_w <- function(v, w, MuVec, lambda, l, j) { return(pnorm(t1(v,w,MuVec,
    lambda,l,j))) }
al_theta_v_w <- function(v, w, MuVec, lambda, l, j) { return(dnorm(t1(v,w,MuVec,
    lambda,l,j))) }

ProdAl_theta_v_w <- function(v, w, MuVec, lambda, j) {
  product = 1/pnorm(w)
  for (k in 1:nChoice) { product = product*Al_theta_v_w(v, w, MuVec, lambda, k, j)
      }
  return(product)
}

#1. function to compute integrand for computing probailities
```

```
    ProbIntegrand <- function(v, w, MuVec, lambda, j) {
    ProbInteg = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*exp(-v*v/2)*exp(-w*w/2)/(2*
        pi)
    return(ProbInteg)
    }

#1. function to compute integrand for first derivatives w r t betaM
    FirstDerBetaIntegrand <- function(v, w, MuVec, lambda, xDiff, j, m) {
      SumBeta = 0
      for (k in 1:nChoice) {
        SumBeta = SumBeta + al_theta_v_w(v,w,MuVec,lambda,k,j)*d_ilm(xDiff,k,j,m)/(
            Al_theta_v_w(v,w,MuVec,lambda,k,j)*sqrt(1-lambda[k]^2))
        }
    FirstDerInteg = -ProdAl_theta_v_w(v, w, MuVec, lambda, j)*SumBeta*exp(-v*v/2)*
        exp(-w*w/2)/(2*pi)
    return(FirstDerInteg)
    }
#2. function to compute integrand for first derivatives w r t lambda_j
    FirstDerLambdajIntegrand <- function(v, w, MuVec, lambda, j) {
      SumLambdaj = 0
      for (k in 1:nChoice) {
        SumLambdaj = SumLambdaj + al_theta_v_w(v,w,MuVec,lambda,k,j)*t2(v,w,MuVec,
            lambda,k,j)/Al_theta_v_w(v,w,MuVec,lambda,k,j)
        }
    SumLambdaj = SumLambdaj - (dnorm(w)/pnorm(w))*t2(v,w,MuVec,lambda,j,j)
    FirstDerLamjInteg = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*SumLambdaj*exp(-v*v
        /2)*exp(-w*w/2)/(2*pi)
    return(FirstDerLamjInteg)
    }
#3. function to compute integrand for first derivatives w r t lambda_r
    FirstDerLambdarIntegrand <- function(v, w, MuVec, lambda, r, j) {
    FirstDerLamrInteg = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*al_theta_v_w(v, w,
        MuVec, lambda, r, j)*t3(v,w,MuVec, lambda, r, j)*exp(-v*v/2)*exp(-w*w/2)/(2*
        pi*Al_theta_v_w(v, w, MuVec, lambda, r, j))
    return(FirstDerLamrInteg)
    }

#1. function to compute integrand for second derivatives w r t betaM betaM'
# (when m = m', we get second derivatives with the same parameter)
    SecondDerBetaMMpmIntegrand <- function(v, w, MuVec, lambda, xDiff, j, m, mpm) {
      SumBetaMpm = 0
      for (kpm in 1:nChoice) {
        SumBetaMpm = SumBetaMpm + al_theta_v_w(v,w,MuVec,lambda,kpm,j)*d_ilm(xDiff,
            kpm,j,mpm)/(Al_theta_v_w(v,w,MuVec,lambda,kpm,j)*sqrt(1-lambda[kpm]^2))
        }
      SumBetaM = 0
      for (k in 1:nChoice) {
        SumBetaM = SumBetaM + al_theta_v_w(v,w,MuVec,lambda,k,j)*d_ilm(xDiff,k,j,m)/
            (Al_theta_v_w(v,w,MuVec,lambda,k,j)*sqrt(1-lambda[k]^2))
        }
      SumBetaMBetaMpm = 0
        for (l in 1:nChoice) {
```

```
      SumBetaMBetaMpm = SumBetaMBetaMpm + (d_ilm(xDiff,l,j,m)*al_theta_v_w(v,w,
          MuVec,lambda,l,j)*d_ilm(xDiff,l,j,mpm))/((1-lambda[l]^2)*Al_theta_v_w(v,
          w,MuVec,lambda,l,j))*(t1(v,w,MuVec,lambda,l,j) + al_theta_v_w(v,w,MuVec,
          lambda,l,j)/(Al_theta_v_w(v,w,MuVec,lambda,l,j)))
      }
  SecondDerIntegBetaMMpm = -ProdAl_theta_v_w(v, w, MuVec, lambda, j)*(-SumBetaMpm*
      SumBetaM+SumBetaMBetaMpm)*exp(-v*v/2)*exp(-w*w/2)/(2*pi)
  return(SecondDerIntegBetaMMpm)
  }
#2. function to compute integrand for second derivatives w r t lambdaj, betaM
  SecondDerLambdajBetaMIntegrand <- function(v, w, MuVec, lambda, xDiff, j, m) {
    SumBetaMpm = 0
    for (kpm in 1:nChoice) {
      SumBetaMpm = SumBetaMpm + al_theta_v_w(v,w,MuVec,lambda,kpm,j)*t2(v,w, MuVec
          , lambda, kpm,j)/Al_theta_v_w(v,w,MuVec,lambda,kpm,j)
      }
    SumBetaMpm = SumBetaMpm - (dnorm(w)*t2(v,w, MuVec, lambda,j,j)/pnorm(w))
    SumBetaM = 0
    for (k in 1:nChoice) {
      SumBetaM = SumBetaM + al_theta_v_w(v,w,MuVec,lambda,k,j)*d_ilm(xDiff,k,j,m)/
          (Al_theta_v_w(v,w,MuVec,lambda,k,j)*sqrt(1-lambda[k]^2))
      }
    SumBetaMLambdaj = 0
      for (l in 1:nChoice) {
      SumBetaMLambdaj = SumBetaMLambdaj + (d_ilm(xDiff,l,j,m)*al_theta_v_w(v,w,
          MuVec,lambda,l,j)/(sqrt(1-lambda[l]^2)*Al_theta_v_w(v,w,MuVec,lambda,l,j
          )))*(t1(v,w,MuVec,lambda,l,j)*t2(v,w,MuVec,lambda,l,j)+al_theta_v_w(v,w,
          MuVec,lambda,l,j)*t2(v,w,MuVec,lambda,l,j)/Al_theta_v_w(v,w,MuVec,lambda
          ,l,j))
      }
  SecondDerIntegLambdajBetaM = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*(-
      SumBetaMpm*SumBetaM+SumBetaMLambdaj)*exp(-v*v/2)*exp(-w*w/2)/(2*pi)
  return(SecondDerIntegLambdajBetaM)
  }
#3. function to compute integrand for second derivatives w r t Lambda_r BetaM
  SecondDerLambdarBetaMIntegrand <- function(v, w, MuVec, lambda, xDiff, r, j, m)
      {
    SumBetaM = 0
    for (k in 1:nChoice) {
      SumBetaM = SumBetaM + al_theta_v_w(v,w,MuVec,lambda,k,j)*d_ilm(xDiff,k,j,m)/
          (Al_theta_v_w(v,w,MuVec,lambda,k,j)*sqrt(1-lambda[k]^2))
      }
  product = t3(v,w,MuVec, lambda, r, j)*SumBetaM - (d_ilm(xDiff,r,j,m)/sqrt(1-
      lambda[r]^2))*(t1(v,w,MuVec,lambda,r,j)*t3(v,w,MuVec,lambda,r,j)+(al_theta_v
      _w(v,w,MuVec,lambda,r,j)*t3(v,w,MuVec,lambda,r,j)/Al_theta_v_w(v,w,MuVec,
      lambda,r,j))-lambda[r]/(1-lambda[r]^2))
  SecondDerIntegLambdarBetaM = -ProdAl_theta_v_w(v, w, MuVec, lambda, j)*al_theta_
      v_w(v,w,MuVec,lambda,r,j)*product*exp(-v*v/2)*exp(-w*w/2)/(2*pi*Al_theta_v_w
      (v,w,MuVec,lambda,r,j))
  return(SecondDerIntegLambdarBetaM)
  }
#4. function to compute integrand for second derivatives w r t Lambda_j^2
```

```
    SecondDerLambdaj2Integrand <- function(v, w, MuVec, lambda, j) {
      SumLambdaj = 0
        for (k in 1:nChoice) {
          SumLambdaj = SumLambdaj + al_theta_v_w(v,w,MuVec,lambda,k,j)*t2(v,w,MuVec,
              lambda,k,j)/Al_theta_v_w(v,w,MuVec,lambda,k,j)
          }
      SumLambdaj = SumLambdaj - (dnorm(w)*t2(v,w,MuVec,lambda,j,j)/pnorm(w))
      SumLambdaj2 = 0
      for (l in 1:nChoice) {
            SumLambdaj2 = SumLambdaj2 + (al_theta_v_w(v,w,MuVec,lambda,l,j)/Al_theta
                _v_w(v,w,MuVec,lambda,l,j))*(t2(v,w,MuVec,lambda,l,j)^2*(t1(v,w,
                MuVec,lambda,l,j) + al_theta_v_w(v,w,MuVec,lambda,l,j)/Al_theta_v_w(
                v,w,MuVec,lambda,l,j)) + w/(sqrt(1-lambda[l]^2)*(1-lambda[j]^2)^(3/
                2)))
          }
    SumLambdaj2 = SumLambdaj2 - (dnorm(w)/pnorm(w))*(t2(v,w,MuVec,lambda,j,j)^2*(t1(
        v,w,MuVec,lambda,j,j) + dnorm(w)/pnorm(w)) + w/((1-lambda[j]^2)^(2)))
    SecondDerLamj2Integ = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*(SumLambdaj^2 -
        SumLambdaj2)*exp(-v*v/2)*exp(-w*w/2)/(2*pi)
    return(SecondDerLamj2Integ)
    }
#5. function to compute integrand for second derivatives w r t Lambda_j Lambda_r
    SecondDerLamjLamrIntegrand <- function(v, w, MuVec, lambda, r, j) {
      SumLambdaj = 0
        for (k in 1:nChoice) {
          SumLambdaj = SumLambdaj + al_theta_v_w(v,w,MuVec,lambda,k,j)*t2(v,w,MuVec,
              lambda,k,j)/Al_theta_v_w(v,w,MuVec,lambda,k,j)
          }
      SumLambdaj = SumLambdaj - (dnorm(w)/pnorm(w))*t2(v,w,MuVec,lambda,j,j)
      product = SumLambdaj*t3(v,w,MuVec,lambda,r,j) - t1(v,w,MuVec,lambda,r,j)*t2(v,
          w,MuVec,lambda,r,j)*t3(v,w,MuVec,lambda,r,j) - al_theta_v_w(v,w,MuVec,
          lambda,r,j)*t2(v,w,MuVec,lambda,r,j)*t3(v,w,MuVec,lambda,r,j)/Al_theta_v_w
          (v,w,MuVec,lambda,r,j) + lambda[r]*v/(1-lambda[r]^2)^(3/2) - w*lambda[j]*
          lambda[r]/(sqrt(1-lambda[j]^2)*(1-lambda[r]^2)^(3/2))
    SecondDerLamjLamrInteg = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*product*al_
        theta_v_w(v,w,MuVec,lambda,r,j)*exp(-v*v/2)*exp(-w*w/2)/(2*pi*Al_theta_v_w(v
        ,w,MuVec,lambda,r,j))
    return(SecondDerLamjLamrInteg)
    }
#6. function to compute integrand for second derivatives w r t Lambda_r^2
    SecondDerLamr2Integrand <- function(v, w, MuVec, lambda, r, j) {
      product = -t1(v,w,MuVec,lambda,r,j)*t3(v,w,MuVec,lambda,r,j)^2 + ((1+2*lambda[
          r]^2)*(MuVec[j]-MuVec[r] + w*sqrt(1-lambda[j]^2)) + (lambda[j] + 2*lambda[
          j]*lambda[r]^2 - 3*lambda[r])*v)/(1-lambda[r]^2)^(5/2)
    SecondDerLamr2Integ = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*product*al_theta_
        v_w(v,w,MuVec,lambda,r,j)*exp(-v*v/2)*exp(-w*w/2)/(2*pi*Al_theta_v_w(v,w,
        MuVec,lambda,r,j))
    return(SecondDerLamr2Integ)
    }
#7. function to compute integrand for second derivatives w r t Lambda_r Lambda_rpm
    SecondDerLamrpmLamrIntegrand <- function(v, w, MuVec, lambda, r, rpm, j) {
```

```
      product = al_theta_v_w(v,w,MuVec,lambda,r,j)*t3(v,w,MuVec,lambda,r,j)*al_theta
          _v_w(v,w,MuVec,lambda,rpm,j)*t3(v,w,MuVec,lambda,rpm,j)/(Al_theta_v_w(v,w,
          MuVec,lambda,r,j)*Al_theta_v_w(v,w,MuVec,lambda,rpm,j))
    SecondDerLamrpmLamrInteg = ProdAl_theta_v_w(v, w, MuVec, lambda, j)*product*exp
        (-v*v/2)*exp(-w*w/2)/(2*pi)
    return(SecondDerLamrpmLamrInteg)
    }
Chk <- function(llim, Integrand, ulim) {
 possibleError1 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
     integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
     lower = llim, upper = 0)
                         ,error=function(e) e)
 possibleError2 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
     integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
     lower = 0, upper = ulim)
                         ,error=function(e) e)
 possibleError3 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
     integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
     lower = llim, upper = 0)
                         ,error=function(e) e)
 possibleError4 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
     integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
     lower = 0, upper = ulim)
                         ,error=function(e) e)
return(c((inherits(possibleError1 , "simpleError")), (inherits(possibleError2 , "
    simpleError")), (inherits(possibleError3 , "simpleError")), (inherits(
    possibleError4 , "simpleError"))))
}

#Module to perform integration without interupton
DoubleInteg <- function(llim, Integrand, ulim) {
 ch = Chk(llim, Integrand, ulim)
 chk = (ch[1] | ch[2] | ch[3] | ch[4])
 if (chk == TRUE) {
  llim = -40
  ulim = 40
 possibleError1 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
     integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
     lower = llim, upper = 0)
                         ,error=function(e) e)
 while((inherits(possibleError1 , "simpleError") == TRUE) & (llim <= -10)) {
    llim = llim + 5
   possibleError1 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
        integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
        lower = llim, upper = 0)
                         ,error=function(e) e)
    }
 possibleError4 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
     integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
     lower = 0, upper = ulim)
                         ,error=function(e) e)
 while((inherits(possibleError4 , "simpleError") == TRUE & (ulim >= 10))) {
```

```
   ulim = ulim - 5
possibleError4 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
    integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
    lower = 0, upper = ulim)
                      ,error=function(e) e)
   }
possibleError2 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
    integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
    lower = 0, upper = ulim)
                      ,error=function(e) e)
possibleError3 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
    integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
    lower = llim, upper = 0)
                      ,error=function(e) e)
chk2 = (inherits(possibleError2 , "simpleError")) | (inherits(possibleError3 , "
    simpleError"))
while((chk2 == TRUE) & (llim <= -10 & ulim >= 10)) {
    llim = llim + 5
    ulim = ulim - 5
possibleError2 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
    integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
    lower = 0, upper = ulim)
                      ,error=function(e) e)
possibleError3 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
    integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
    lower = llim, upper = 0)
                      ,error=function(e) e)
chk2 = (inherits(possibleError2 , "simpleError")) | (inherits(possibleError3 , "
    simpleError"))
}
}
if ((llim >= -10) | (ulim <= 10)) {
   a = seq(-10, 10, by=1)
   b = seq(-10, 10, by=1)
   a1 = 0
    for (apm in 1:(length(a)-1)) {
     for (bpm in 1:(length(b)-1)) {
     possibleError <- tryCatch(integrate(function(v) { sapply(v, function(v) {
         integrate(function(w) Integrand(v,w), lower = b[bpm], upper = b[bpm+1])$
         value }) }, lower = a[apm], upper = a[apm+1])
                      ,error=function(e) e)
     if (inherits(possibleError, "simpleError"))
     { add = 0 }
     else {
     add = integrate(function(v) { sapply(v, function(v) { integrate(function(w)
         Integrand(v,w), lower = b[bpm], upper = b[bpm+1])$value }) }, lower = a[
         apm], upper = a[apm+1])$value
     }
     a1 = a1 + add
      }
     }
}
```

```
 else {
    possibleError1 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
        integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
        lower = llim, upper = 0)
                        ,error=function(e) e)
  if (inherits(possibleError1, "simpleError")) { c1 = 0}
  else {c1 = integrate(function(v) { sapply(v, function(v) { integrate(function(w)
      Integrand(v,w), lower = llim, upper = 0)$value }) }, lower = llim, upper = 0)$
      value }

    possibleError2 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
        integrate(function(w) Integrand(v,w), lower = llim, upper = 0)$value }) },
        lower = 0, upper = ulim)
                        ,error=function(e) e)
  if (inherits(possibleError2, "simpleError")) { c2 = 0}
  else {c2 = integrate(function(v) { sapply(v, function(v) { integrate(function(w)
      Integrand(v,w), lower = llim, upper = 0)$value }) }, lower = 0, upper = ulim)$
      value}

    possibleError3 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
        integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
        lower = llim, upper = 0)
                        ,error=function(e) e)
  if (inherits(possibleError3, "simpleError")) { c3 = 0}
  else {c3 = integrate(function(v) { sapply(v, function(v) { integrate(function(w)
      Integrand(v,w), lower = 0, upper = ulim)$value }) }, lower = llim, upper = 0)$
      value }

    possibleError4 <- tryCatch(integrate(function(v) { sapply(v, function(v) {
        integrate(function(w) Integrand(v,w), lower = 0, upper = ulim)$value }) },
        lower = 0, upper = ulim)
                        ,error=function(e) e)
  if (inherits(possibleError4, "simpleError")) { c4 = 0}
  else {c4 = integrate(function(v) { sapply(v, function(v) { integrate(function(w)
      Integrand(v,w), lower = 0, upper = ulim)$value }) }, lower = 0, upper = ulim)$
      value}

    a1 = c1 + c2 + c3 + c4
 }
return(a1)
}

#*****************************************************************************#
# Defining Probabilities, Derivatives, Double Derivatives for MDCP II model    #
#*****************************************************************************#
# function to compute MDCP Probabilities
ProbMDCP <- function(n, Data, pars, nSub, nCovariates, nChoice) {
  betainn = pars[1:nCovariates]
  lambda = pars[(nCovariates+1):(nCovariates+nChoice)]
  xdata = Data[((n-1)*nSub*nChoice+1):(n*nSub*nChoice),2:(nCovariates+1)]
  means =  xdata%*%betainn
  SpMeans = matrix(0,nSub*nChoice, nChoice)
```

```
  for (i in 1:nSub) {
            SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice,1)%*
                %t(means[((i-1)*nChoice+1):(i*nChoice),1])
          }
  Prob = matrix(0, nSub*nChoice, 1)
  for (i in 1:nSub) {
    for (j in 1:nChoice) {
      MuVec = SpMeans[((i-1)*nChoice+j),1:nChoice]
      Integrand <- function(v,w) {
      return(ProbIntegrand(v, w, MuVec, lambda, j))
     }
    Prob[((i-1)*nChoice+j),1] = DoubleInteg(-Inf, Integrand, Inf)
     }
    }
  return(Prob)
}

#wrapper for parellel computation in estimation
ProbMDCP1 <- function(nLoop, Data, pars, nSub, nCovariates, nChoice) {
 xProb <-foreach(m=1:nLoop, .combine=rbind) %dopar% ProbMDCP(m, Data, pars, nSub,
     nCovariates, nChoice)
return(xProb)
}

# function to compute MDCP derivatives
DerMDCP <- function(n, Data, pars, nSub, nCovariates, nChoice) {
  betainn = pars[1:nCovariates]
  lambda = pars[(nCovariates+1):(nCovariates+nChoice)]
  xdata = Data[((n-1)*nSub*nChoice+1):(n*nSub*nChoice),2:(nCovariates+1)]
  means =  xdata%*%betainn
  SpMeans = matrix(0,nSub*nChoice, nChoice)
  SpXs = matrix(0, nSub*nChoice, nChoice*nCovariates)
  for (i in 1:nSub) {
          SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice,1)%*%t
              (means[((i-1)*nChoice+1):(i*nChoice),1])
          rearrange = t(Data[((i-1)*nChoice+1):(i*nChoice),1+1])
          for (m in 2:nCovariates)  {
                rearrange = cbind(rearrange, t(Data[((i-1)*nChoice+1):(i*nChoice),m
                    +1]))
                }
         SpXs[((i-1)*nChoice+1):(i*nChoice),1:(nChoice*nCovariates)] = matrix(1,
             nChoice,1)%*%rearrange
        }
 DerProbBetaMLambda = matrix(0, nSub*nChoice, (nCovariates+nChoice))
 for (i in 1:nSub) {
  for (j in 1:nChoice) {
  MuVec = SpMeans[((i-1)*nChoice+j),1:nChoice]
  xDiff = SpXs[((i-1)*nChoice+j),1:(nChoice*nCovariates)]
  for (m in 1:nCovariates) {
      Integrand <- function(v,w) {
        return(FirstDerBetaIntegrand(v, w, MuVec, lambda, xDiff, j, m))
      }
```

```
    DerProbBetaMLambda[((i-1)*nChoice+j),m] = DoubleInteg(-Inf, Integrand, Inf)
    }
  for (r in 1:nChoice) {
      if (r == j) {
        Integrand <- function(v,w) {
        return(FirstDerLambdajIntegrand(v, w, MuVec, lambda, j))
        }
       a2 = DoubleInteg(-Inf, Integrand, Inf)
      }
      else {
        Integrand <- function(v,w) {
        return(FirstDerLambdarIntegrand(v, w, MuVec, lambda, r, j))
        }
       a2 = DoubleInteg(-Inf, Integrand, Inf)
      }
    DerProbBetaMLambda[((i-1)*nChoice+j),nCovariates+r] = a2
    }
  }
 }
return(DerProbBetaMLambda)
}

# wrapper function to be used in estimation
DerMDCP1 <- function(nLoop, Data, pars, nSub, nCovariates, nChoice, Prob) {
xDer <- foreach(m=1:nLoop, .combine=rbind) %dopar% DerMDCP(m, Data, pars, nSub,
    nCovariates, nChoice)
return(xDer)
}

# function to compute MDCP Double derivatives
DDerMDCP <- function(n, Data, pars, nSub, nCovariates, nChoice) {
  betainn = pars[1:nCovariates]
  lambda = pars[(nCovariates+1):(nCovariates+nChoice)]
  xdata = Data[((n-1)*nSub*nChoice+1):(n*nSub*nChoice),2:(nCovariates+1)]
  means =  xdata%*%betainn
  SpMeans = matrix(0,nSub*nChoice, nChoice)
  SpXs = matrix(0, nSub*nChoice, nChoice*nCovariates)
  for (i in 1:nSub) {
          SpMeans[((i-1)*nChoice+1):(i*nChoice),1:nChoice] = matrix(1,nChoice,1)%*%t
              (means[((i-1)*nChoice+1):(i*nChoice),1])
          rearrange = t(Data[((i-1)*nChoice+1):(i*nChoice),1+1])
          for (m in 2:nCovariates)  {
                rearrange = cbind(rearrange, t(Data[((i-1)*nChoice+1):(i*nChoice),m
                    +1]))
                }
         SpXs[((i-1)*nChoice+1):(i*nChoice),1:(nChoice*nCovariates)] = matrix(1,
             nChoice,1)%*%rearrange
        }
DDerProbBetaMLambda = matrix(0, nSub*nChoice, (nCovariates+nChoice)^2)
for (i in 1:nSub) {
  for (j in 1:nChoice) {
  MuVec = SpMeans[((i-1)*nChoice+j),1:nChoice]
```

```
xDiff = SpXs[((i-1)*nChoice+j),1:(nChoice*nCovariates)]
b1 = matrix(0,(nChoice+nCovariates),(nChoice+nCovariates))
    for (m in 1:nCovariates) {
       for (mpm in m:nCovariates) {
              Integrand <- function(v,w) {
              return(SecondDerBetaMMpmIntegrand(v, w, MuVec, lambda, xDiff, j, m,
                  mpm))
              }
              b1[m,mpm] = b1[m,mpm] + DoubleInteg(-Inf, Integrand, Inf)
           }
      for (r in 1:nChoice) {
              if (r == j) {
                Integrand <- function(v,w) {
                return(SecondDerLambdajBetaMIntegrand(v, w, MuVec, lambda, xDiff,
                    j, m))
                }
              add = DoubleInteg(-Inf, Integrand, Inf)
              }
              else {
                Integrand <- function(v,w) {
                return(SecondDerLambdarBetaMIntegrand(v, w, MuVec, lambda, xDiff,
                    r, j, m))
                }
              add = DoubleInteg(-Inf, Integrand, Inf)
              }
              b1[m,(nCovariates+r)]=b1[m,(nCovariates+r)]+add
           }
         }
    for (r in 1:nChoice) {
      for (rpm in r:nChoice) {
            if ((r==rpm) & (r == j)) {
                  Integrand <- function(v,w) {
                  return(SecondDerLambdaj2Integrand(v, w, MuVec, lambda, j))
                  }
              add = DoubleInteg(-Inf, Integrand, Inf)
            }
          else if ((r!=rpm) & (r == j)) {
                 Integrand <- function(v,w) {
                  return(SecondDerLamjLamrIntegrand(v, w, MuVec, lambda, rpm, j))
                  }
              add = DoubleInteg(-Inf, Integrand, Inf)
           }
          else if ((r==rpm) & (r != j)) {
                 Integrand <- function(v,w) {
                  return(SecondDerLamr2Integrand(v, w, MuVec, lambda, r, j))
                  }
              add = DoubleInteg(-Inf, Integrand, Inf)
           }
          else if ((r!=rpm) & ((r!= j) | (rpm != j))) {
                  Integrand <- function(v,w) {
                  return(SecondDerLamrpmLamrIntegrand(v, w, MuVec, lambda, r, rpm,
                      j))
```

```
                    }
                add = DoubleInteg(-Inf, Integrand, Inf)
                }
            b1[(nCovariates+r),(nCovariates+rpm)]=b1[(nCovariates+r),(nCovariates+
                rpm)]+add
            }
        }
DDerProbBetaMLambda[((i-1)*nChoice+j),] = as.vector(b1+t(b1)-diag(diag(b1)))
  }
 }
return(DDerProbBetaMLambda)
}

#Wrapper to compute DDer MDCP
DDerMDCP1 <- function(nLoop, Data, pars, nSub, nCovariates, nChoice) {
xDDer <- foreach(m=1:nLoop, .combine=rbind) %dopar% DDerMDCP(m, Data, pars, nSub,
    nCovariates, nChoice)
return(xDDer)
}

#******************************************************************************#
# Defining Probabilities, Derivatives, Double Derivatives for PCL model        #
#******************************************************************************#
# function to compute PCL Probabilities
ProbPCL <- function(Data, pars) {
  betainn = pars[1:nCovariates]
  lambda = pars[(nCovariates+1):(nCovariates+nChoice)]
  xdata = Data[,2:(nCovariates+1)]
  means =  xdata%*%betainn
  Prob = matrix(0, nObs*nChoice, 1)
  MuVec = matrix(0, nChoice, 1)
  for (i in 1:nObs) {
    MuVec = means[((i-1)*nChoice+1):(i*nChoice),1]
    ProbMatrixNr = matrix(0, nChoice, nChoice)
    ProbMatrixDr = matrix(0, nChoice, nChoice)
    for (j in 1:nChoice) {
      for (k in 1:nChoice) {
      if (j==k) {
        ProbMatrixNr[j,k]=0
        ProbMatrixDr[j,k]=0
        }
      else {
        ProbMatrixNr[j,k] = exp(MuVec[j]/(lambda[j]*lambda[k]))*(exp(MuVec[j]/(
            lambda[j]*lambda[k])) + exp(MuVec[k]/(lambda[j]*lambda[k])))^(lambda[j]*
            lambda[k]-1)
        ProbMatrixDr[j,k] = (exp(MuVec[j]/(lambda[j]*lambda[k])) + exp(MuVec[k]/(
            lambda[j]*lambda[k])))^(lambda[j]*lambda[k])
    if (is.finite(ProbMatrixNr[j,k])==FALSE) { ProbMatrixNr[j,k] = 0}
    if (is.finite(ProbMatrixDr[j,k])==FALSE) { ProbMatrixDr[j,k] = 0}
      }
   }
  }
```

```
 Prob[((i-1)*nChoice+1):(i*nChoice), 1]= 2*apply(ProbMatrixNr, 1, sum)/sum(
     ProbMatrixDr)
}
return(Prob)
}

# function to compute PCL Probabilities for each i & j
ProbPCL.ij <- function(pars, Data, n, o, p, nSub, nCovariates, nChoice) {
  betainn = pars[1:nCovariates]
  lambda = pars[(nCovariates+1):(nCovariates+nChoice)]
  xdata = Data[((n-1)*nSub*nChoice+1):(n*nSub*nChoice),2:(nCovariates+1)]
  means =  xdata%*%betainn
  Prob = matrix(0, nSub*nChoice, 1)
  MuVec = matrix(0, nChoice, 1)
    i = o
    MuVec = means[((i-1)*nChoice+1):(i*nChoice),1]
    ProbMatrixNr = matrix(0, nChoice, nChoice)
    ProbMatrixDr = matrix(0, nChoice, nChoice)
    for (j in 1:nChoice) {
      for (k in 1:nChoice) {
      if (j==k) {
        ProbMatrixNr[j,k]=0
        ProbMatrixDr[j,k]=0
        }
      else {
        ProbMatrixNr[j,k] = exp(MuVec[j]/(lambda[j]*lambda[k]))*(exp(MuVec[j]/(
             lambda[j]*lambda[k])) + exp(MuVec[k]/(lambda[j]*lambda[k])))^(lambda[j]*
             lambda[k]-1)
        ProbMatrixDr[j,k] = (exp(MuVec[j]/(lambda[j]*lambda[k])) + exp(MuVec[k]/(
             lambda[j]*lambda[k])))^(lambda[j]*lambda[k])
    if (is.finite(ProbMatrixNr[j,k])==FALSE) { ProbMatrixNr[j,k] = 0}
    if (is.finite(ProbMatrixDr[j,k])==FALSE) { ProbMatrixDr[j,k] = 0}
      }
   }
  }
 Prob[((i-1)*nChoice+1):(i*nChoice), 1]= 2*apply(ProbMatrixNr, 1, sum)/sum(
     ProbMatrixDr)
return(Prob[((i-1)*nChoice+p),1])
}

# function to compute PCL derivatives
DerPCL <- function(n, Data, pars, nSub, nCovariates, nChoice) {
library(numDeriv)
PCL.Der = matrix(0, nSub*nChoice, (nCovariates+nChoice))
   for (i in 1:nSub) {
    for (j in 1:nChoice) {
      q3 = grad(ProbPCL.ij, pars, method = "Richardson", Data=Data, n=n, o=i, p=j,
          nSub=nSub, nCovariates=nCovariates, nChoice=nChoice)
              PCL.Der[(i-1)*nChoice+j,] = as.vector(q3)
    }
  }
return(PCL.Der)
```

```
}

DerPCL1 <- function(nLoop, Data, pars, nSub, nCovariates, nChoice) {
PCL.Der1 <- foreach(m=1:nLoop, .combine=rbind) %dopar% DerPCL(m, Data, pars, nSub,
    nCovariates, nChoice)
return(PCL.Der1)
}

# function to compute PCL Double derivatives
DDerPCL <- function(n, Data, pars, nSub, nCovariates, nChoice) {
PCL.DDer = matrix(0, nSub*nChoice, (nCovariates+nChoice)^2)
   for (i in 1:nSub) {
    for (j in 1:nChoice) {
      q3 = hessian(ProbPCL.ij, pars, method = "Richardson", Data=Data, n=n, o=i, p=j
          , nSub=nSub, nCovariates=nCovariates, nChoice=nChoice)
        PCL.DDer[(i-1)*nChoice+j,] = as.vector(q3)
  }
}
return(PCL.DDer)
}

DDerPCL1 <- function(nLoop, Data, pars, nSub, nCovariates, nChoice) {
clusterExport(c2, c("pars"))
PCL.DDer1 <- foreach(m=1:nLoop, .combine=rbind) %dopar% DDerPCL(m, Data, pars, nSub,
     nCovariates, nChoice)
return(PCL.DDer1)
}
#*******************************************************************************#
# Defining likelihood, Gradient, Hessian for product correlated probit model      #
#*******************************************************************************#
#MDCP Likelihood
MDCP.Likelihood <- function(Data, pars, nLoop, nSub, nCovariates, nChoice) {
clusterExport(c2, c("Data", "pars"))
Prob = ProbMDCP1(nLoop, Data, pars, nSub, nCovariates, nChoice)
lik = Data[,1]*log(pmax(1e-323,Prob))
loglike = sum(lik)
return(loglike)
}

#MDCP Gradient
MDCP.Gradient <- function(Data, pars, nLoop, nSub, nCovariates, nChoice) {
clusterExport(c2, c("Data", "pars"))
Prob = ProbMDCP1(nLoop, Data, pars, nSub, nCovariates, nChoice)
Der = DerMDCP1(nLoop, Data, pars, nSub, nCovariates, nChoice)
Grd = Data[,1]*(coldiv.matrix(Der,Prob));
return(apply(Grd,2,sum))
}

#MDCP Hessian
MDCP.Hessian <- function(nLoop, Data, pars, nSub, nCovariates, nChoice) {
Prob = ProbMDCP1(nLoop, Data, pars, nSub, nCovariates, nChoice)
Der = DerMDCP1(nLoop, Data, pars, nSub, nCovariates, nChoice)
```

```
DDer = DDerMDCP1(nLoop, Data, pars, nSub, nCovariates, nChoice)
Product = matrix(0, nObs*nChoice, (nCovariates+nChoice)^2)
for (i in 1:(nCovariates+nChoice)) {
  for (j in 1:(nCovariates+nChoice)) {
  Product[,(i-1)*(nCovariates+nChoice)+j] = Der[,i]*Der[,j]
  }
}
Hess = DDer-coldiv.matrix(Product, Prob)
return(apply(Hess, 2, sum))
}

#*****************************************************************************#
#* Defining likelihood, Gradient, Hessian for PCL Model                       *#
#*****************************************************************************#
#PCL Likelihood
PCL.Likelihood <- function(pars, Data, nLoop, nSub, nCovariates, nChoice) {
clusterExport(c2, c("Data", "pars"))
Prob = ProbPCL(Data, pars)
lik = Data[,1]*log(pmax(1e-323,Prob))
loglike = sum(lik)
return(loglike)
}
#PCL Gradient
PCL.Gradient <- function(pars, Data, nLoop, nSub, nCovariates, nChoice) {
clusterExport(c2, c("Data", "pars"))
Prob = ProbPCL(Data, pars)
Der = DerPCL1(nLoop, Data, pars, nSub, nCovariates, nChoice)
Grd = Data[,1]*(coldiv.matrix(Der,Prob))
return(apply(Grd,2,sum))
}

#PCL Hessian
PCL.Hessian <- function(nLoop, Data, pars, nSub, nCovariates, nChoice) {
Prob = ProbPCL(Data, pars)
Der = DerPCL1(nLoop, Data, pars, nSub, nCovariates, nChoice)
DDer = DDerPCL1(nLoop, Data, pars, nSub, nCovariates, nChoice)
Product = matrix(0, nObs*nChoice, (nCovariates+nChoice)^2)
for (i in 1:(nCovariates+nChoice)) {
  for (j in 1:(nCovariates+nChoice)) {
  Product[,(i-1)*(nCovariates+nChoice)+j] = Der[,i]*Der[,j]
  }
}
Hess = DDer-coldiv.matrix(Product, Prob)
return(apply(Hess, 2, sum))
}

#*****************************************************************************#
#*  Computation of asymptotic efficiency for real market                      *#
#*****************************************************************************#
#Input parameters for asymptotic efficiency
seed = 16461
nObs = 4
```

```
nChoice = 4
nLevel = 3    #Number of levels for discrete covariate
nCovariates = nChoice+nLevel - 1    #Number of covariates such as intercepts,
    parameters
nPar = nCovariates + nChoice #Number of parameters
# Four arbitrairly chosen values of lambda, the correlation parameter
lambda1 = c(-0.7541376, -0.6808193, -0.7693839, -0.7381692)
lambda2 = c(-0.7015052, -0.5163027, -0.1686635,  0.3792168)
lambda3 = c( 0.28316460, -0.07476282,  0.54631999,  0.29311195)
lambda4 = c(0.6755678, -0.5467673, -0.4264408, -0.8104457)
StartBeta = c(-0.479,  1.061,  0.476,  0.781,  0.107, -0.525)
Data = DataSim(seed, nObs, nChoice, nLevel, StartBeta, 0.3)

#Set up for parellel computing
library(doSNOW)
nCores <- 4 # number of CPUs
nSub <- nObs/nCores
nLoop = nCores
c2<-makeCluster(nCores)
clusterExport(c2, c("nSub", "nLoop", "nCovariates", "nChoice"))
clusterExport(c2, c("ProbMDCP", "DerMDCP", "DDerMDCP", "ProbPCL", "DerPCL", "DDerPCL
    ", "ProbPCL.ij", "DoubleInteg", "Chk", "colprod.matrix", "coldiv.matrix", "t1",
    "t2", "t3", "d_ilm", "A1_theta_v_w", "a1_theta_v_w", "ProdA1_theta_v_w", "
    ProbIntegrand"))
clusterExport(c2, c("FirstDerBetaIntegrand", "FirstDerLambdajIntegrand", "
    FirstDerLambdarIntegrand", "SecondDerBetaMMpmIntegrand", "
    SecondDerLambdajBetaMIntegrand", "SecondDerLambdarBetaMIntegrand",
    "SecondDerLambdaj2Integrand", "SecondDerLamjLamrIntegrand", "
        SecondDerLamr2Integrand", "SecondDerLamrpmLamrIntegrand"))
registerDoSNOW(c2)
stopCluster(c2)
#Asymptotic efficiency
asympeff <- function(Data, StartBeta, lambda) {
  pars = c(StartBeta, lambda1)
  clusterExport(c2, "pars")
  MDCP.Hess = MDCP.Hessian(nCores, Data, pars, nSub, nCovariates, nChoice)
  MDCP.Hess = matrix(MDCP.Hess, (nCovariates+nChoice), (nCovariates+nChoice))
  PCL.Hess = PCL.Hessian(Data, pars)
  PCL.Hess = matrix(PCL.Hess, (nCovariates+nChoice), (nCovariates+nChoice))
  InvFishMDCP = solve(-MDCP.Hess)
  InvFishPCL = solve(-PCL.Hess)
  eff = diag(InvFishPCL)/diag(InfFishMDCP)
return(eff)
}

eff1 <- asympeff(Data, StartBeta, lambda1)
eff2 <- asympeff(Data, StartBeta, lambda2)
eff3 <- asympeff(Data, StartBeta, lambda3)
eff4 <- asympeff(Data, StartBeta, lambda4)
```

```
#***************************************************************************#
#*  Application to real time data                                          *#
#***************************************************************************#
#Laundry Detergent data
Laundry = read.table("C:/Users/bravi/Desktop/Laundry.txt", sep="", header = TRUE)
Laundry = as.matrix(Laundry, 2657, 13)

nChoice = 6
nCovariates = 11
nPar = nCovariates+nChoice
nObs = 1000

Price = Laundry[1:(nObs),2:7]
Select = Laundry[1:(nObs),8:13]
PriceNew = matrix(t(Price), nObs*nChoice, 1)
SelectNew = matrix(t(Select), nObs*nChoice, 1)
intmat = rbind(diag(nChoice-1), matrix(0, 1, nChoice-1))
xInt = intmat
for (i in 1:(nObs-1)) {
xInt = rbind(xInt, intmat)
}
pInt = diag(nChoice)
priceInt = pInt
for (i in 1:(nObs-1)) {
priceInt = rbind(priceInt, pInt)
}
PriceN = matrix(0, nObs*nChoice, nChoice)
for (i in 1:nChoice) {
PriceN[,i] = PriceNew*priceInt[,i]
}
xData = cbind(xInt, PriceN)
yData = SelectNew
LaundryNew = cbind(yData, xData)
BetaInit = c(2, 1, 1, 2, 1, rep(-105, nChoice))
LambdaInit = runif(nChoice, -1, 1)
initial = c(BetaInit, LambdaInit)

sol.PCL = constrOptim(initial, PCL.Likelihood, gr=PCL.Gradient, ui=cbind(matrix(0, 2
    *nChoice, nCovariates), rbind(diag(nChoice), -diag(nChoice))), ci=c(rep(-1, 2*
    nChoice)), mu = 1e-06, control = list(fnscale=-1),
            method = "BFGS", outer.iterations = 100, outer.eps = 1e-05, Data=
                LaundryNew, nLoop=nLoop, nSub=nSub, nCovariates=nCovariates, nChoice
                =nChoice, hessian = FALSE)
sol.MDCP = constrOptim(initial, MDCP.Likelihood, gr=MDCP.Gradient, ui=cbind(matrix
    (0, 2*nChoice, nCovariates), rbind(diag(nChoice), -diag(nChoice))), ci=c(rep(-1,
     2*nChoice)), mu = 1e-06, control = list(fnscale=-1),
            method = "BFGS", outer.iterations = 100, outer.eps = 1e-05, Data=
                LaundryNew, nLoop=nLoop, nSub=nSub, nCovariates=nCovariates, nChoice
                =nChoice, hessian = FALSE)

MDCP.Hess = MDCP.Hessian(nCores, LaundryNew, sol.MDCP$par, nSub, nCovariates,
    nChoice)
```

```
PCL.Hess = PCL.Hessian(nLoop, LaundryNew, sol.PCL$par, nSub, nCovariates, nChoice)
MDCP.Hess = matrix(MDCP.Hess, (nCovariates+nChoice), (nCovariates+nChoice))
PCL.Hess = matrix(PCL.Hess, (nCovariates+nChoice), (nCovariates+nChoice))
seMDCP = sqrt(diag(solve(-MDCP.Hess)))
sePCL = sqrt(diag(solve(-PCL.Hess)))

#Travel mode data
Travel = read.table("C:/Users/bravi/Desktop/Travel data.txt", sep="", header = TRUE)
nObs = 210
Travel = as.matrix(Travel, nObs, 7)
nChoice = 4
nCovariates = 7
nPar = nCovariates+nChoice

nObs = 208 #To make use of parellel computing
intmat = rbind(diag(nChoice-1), matrix(0, 1, nChoice-1))
xInt = intmat
for (i in 1:(nObs-1)) {
xInt = rbind(xInt, intmat)
}
TravelNew = matrix(0, nObs*nChoice, nCovariates+1)
TravelNew[1:(nObs*nChoice),1] = Travel[1:(nObs*nChoice),1]
TravelNew[1:(nObs*nChoice),2:4] = xInt
TravelNew[1:(nObs*nChoice),5:8] = Travel[1:(nObs*nChoice),2:5]
initial = c(2.24711280, 3.36113649, 2.90822876, 0.42927333, 0.83295026,
    0.08111699, 0.12676757, 0.54327163, -0.48312499, 0.81286843, -0.91332281)

sol.PCL = constrOptim(initial, PCL.Likelihood, gr=PCL.Gradient, ui=cbind(matrix(0, 2
    *nChoice, nCovariates), rbind(diag(nChoice), -diag(nChoice))), ci=c(rep(-1, 2*
    nChoice)), mu = 1e-06, control = list(fnscale=-1),
            method = "BFGS", outer.iterations = 100, outer.eps = 1e-05, Data=
                TravelNew, nLoop=nLoop, nSub=nSub, nCovariates=nCovariates, nChoice=
                nChoice, hessian = FALSE)

sol.MDCP = constrOptim(initial, MDCP.Likelihood, gr=MDCP.Gradient, ui=cbind(matrix
    (0, 2*nChoice, nCovariates), rbind(diag(nChoice), -diag(nChoice))), ci=c(rep(-1,
     2*nChoice)), mu = 1e-06, control = list(fnscale=-1),
            method = "BFGS", outer.iterations = 100, outer.eps = 1e-05, Data=
                TravelNew, nLoop=nLoop, nSub=nSub, nCovariates=nCovariates, nChoice=
                nChoice, hessian = FALSE)

MDCP.Hess = MDCP.Hessian(nCores, TravelNew, sol.MDCP$par, nSub, nCovariates, nChoice
    )
PCL.Hess = PCL.Hessian(nLoop, TravelNew, sol.PCL$par, nSub, nCovariates, nChoice)
MDCP.Hess = matrix(MDCP.Hess, (nCovariates+nChoice), (nCovariates+nChoice))
PCL.Hess = matrix(PCL.Hess, (nCovariates+nChoice), (nCovariates+nChoice))
seMDCP = sqrt(diag(solve(-MDCP.Hess)))
sePCL = sqrt(diag(solve(-PCL.Hess)))
```

VITA

Bhaskara Ravi
Department of Mathematics and Statistics
Old Dominion University
Norfolk, VA 23529

Education

- Ph.D. in Applied and Computational Mathematics (Statistics), Old Dominion University, Norfolk, VA (December 2012)
- M.Sc. in Statistics and Operations Research, University of Hyderabad, Hyderabad, India (May 1999)
- B.Sc. in Mathematics, Physics and Statistics, Andhra University, India (May 1997)

Experience

- 2010-2012, GRA, Eastern Virginia Medical School, Norfolk, VA.
- 2008-2010, GRA, Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA.
- 2004-2008, Project Manager, TNS India Pvt. Ltd. Mumbai, India.
- 2003-2004, Sr. Executive Market Research, Mensamind Pvt. Ltd, Hyderabad, India.
- 2001-2003, Research Executive, Pragna Research and Consultancy Services, Hyderabad, India.
- 2000-2001, Faculty of Statistics, Vivekananda School of PG Studies, Hyderabad, India.

Typeset using LaTeX.