The Pennsylvania State University
The Graduate School

DEMAND ESTIMATION WITH MISSINGNESS OF PRODUCT AVAILABILITY

A Thesis in
Statistics
by
Thilini Rasanga Saram

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2021

The thesis of Thilini Rasanga Saram was reviewed and approved by the following:

David Russell Hunter
Professor of Statistics
Thesis Advisor

Prabhani Kuruppumullage Don
Assistant Research Professor of Statistics

Ephraim Mont Hanks
Associate Professor of Statistics
Head of the Graduate Program

Abstract

Discrete choice models predict the choices among two or more discrete alternatives. We discuss some existing models but focus on the Multinomial Choice Model (MNL) and explain Expectation-Maximization (EM) algorithms. We provide evidence that failing to account for product availability leads to bias in demand estimates and use an illustrative example to demonstrate this. We propose a new model accounting for product availability. To accomplish this, we use EM algorithms and direct optimization of observed data log-likelihood for estimating maximum likelihood estimates by introducing product availability as a missing variable. We use a simulation study to compare the models' prediction accuracy and fit the new model to the illustrative example.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my parents for supporting and encouraging me to pursue my dreams. Second, I would like to thank my husband for helping and being my strength through the rough times and keeping me optimistic. I cannot express enough thanks to my advisor, Dr. David Hunter, for advising me throughout the research and not give up on me when my progress is slow. I learned not only about research but regarding many parts of academia because of you. Further, I would like to thank Dr. Aydin Alptekinoglu for providing me with several resources and advice about the topic. I want to thank Dr. Ephraim and Dr. Prabhani for reviewing my thesis. Finally, I would like to thank all the professors in the department, my cohort, and the Sri Lankan community in State College. My heartfelt thanks to everyone.

1 Modeling Product Demand

Demand modeling is an essential aspect of supply chains that involves creating a probability model for the individual choices made when a consumer selects one from a group of related products. Discrete choice models are probability models that predict choices between two or more discrete outcomes. In this case, we are interested in which one to choose.

This thesis addresses a situation known as missing product availability that sometimes occurs in demand modeling contexts. For example, suppose that several candy brands such as Starburst, Skittles, etc., are available in a shop. The shop may be interested in modeling the probability that a customer chooses the k^{th} brand in week t based on past sales data. However, the shop may not have recorded which brands were available to customers every week. This means that in weeks where no sales of brand k were recorded, we do not know whether this was because brand k was unavailable or because it was simply not selected by any of the customers. As we will demonstrate, these two alternatives have different implications for the estimates obtained via discrete choice models.

Some previous work on discrete choice models ignores the fact that product availability is not always observed. For example, Alptekinoglu and Semple (2018) assume that zero units sold imply the product was available, but no one has purchased the product. Some other studies, such as Campo et al. (2003) assume the product is not available when there are zero purchased units.

In this thesis, first, we compare the demand estimates and parameters of the demand model under two scenarios. The first scenario assumes that no one has purchased the brands with zero sales. The second scenario assumes that brands with zero purchase units were not available during the period. Then we build an estimation framework in which a product's presence is considered unobserved data, i.e., that we correctly calculate parameter estimates no matter which scenario we encounter.

This thesis's primary goal is to provide evidence that failing to account for product availability leads to bias in demand estimates and to correct this bias by introducing the brands' availability during a week as a latent/missing variable. To accomplish this, we use an Expectation-Maximization (EM) algorithm and direct optimization of observed data log-likelihood for maximum likelihood estimation.

As mentioned above, Alptekinoglu and Semple (2018) uses discrete choice models such as the multinomial logit (MNL) model, the exponomial choice (EC) model, and the heteroscedastic exponomial choice (HEC) model to estimate demand under observed factors. In this thesis, we focus entirely on the MNL model. However, we briefly introduce the EC model and leave the possibility to extend our missing-data framework.

1.1 Discrete Choice Models

A discrete choice model describes the decision-makers' choice among a set of alternatives. The choice set in the discrete choice framework is the set of alternatives. The choice set must meet three requirements (Train, 2009).

1. The alternatives must be mutually exclusive, which means that choosing one alternative means not choosing another alternative.

2. The choice set must be exhaustive, meaning that all possible alternatives are included and that the decision-maker necessarily chooses one of the alternatives.

3. The number of alternatives must be finite.

The choice set is exhaustive even if the decision-maker is not choosing any options or choosing an outside option because we can add these "no choice" or "none of the above" to the choice set.

Discrete choice models are derived under utility theory which assumes utility-maximizing behavior by the decision-maker (Train, 2009). Utility maximizing related to purchase decisions is the concept of purchasing the goods that give the highest satisfaction. Economists and mathematical psychologists independently developed the theory behind choice modeling. Thurstone (1927) developed this concept in terms of psychological values. The models that can be derived in this manner are called random utility models, and McFadden (1974) formalized them.

The decision-maker labeled as j, indexed $1, ..., m$, selects a choice among n alternatives. Let $1, ..., n$ denote the choice set. U_{jk} is the random utility that the decision-maker j obtains from alternative k. According to random utility models, the decision-maker

j chooses the alternative that provides the highest utility. Thus, the decision-maker chooses alternative k if and only if $U_{jk} > U_{jl}$ for all $k \neq l$.

Let V_{jk} be the function that relates a vector of observed attributes ($\boldsymbol{x}_{jk}$) relating to j^{th} decision-maker and k^{th} alternative. A linear model gives

$$V_{jk} = \alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{jk}, \tag{1.1}$$

where α_k is the intrinsic desirability of product k, $\boldsymbol{x}_{jk}$ is the vector of observed variables of product k by customer j, and $\boldsymbol{\beta}$ is the vector of unknown parameters that reflect customers' preferences. $\boldsymbol{x}_{jk}$ are alternative specific variables with a generic coefficient β. For example, $\boldsymbol{x}_{jk}$ could include sensitivity to price, promotional activities such as feature and display. Since we cannot observe some aspects of utility, let ϵ_{jk} be the random vector of factors that affect utility but not included in V_{jk}. Thus, $U_{jk} = V_{jk} + \epsilon_{jk}$ for all j and k (Train, 2009). Therefore, the choice of the j^{th} decision-maker is given by

$$\underset{1 \leq k \leq n}{\arg\max} \; V_{jk} + \epsilon_{jk}, \tag{1.2}$$

which is same as choosing the alternative k if and only if $U_{jk} > U_{jl}$ for all $k \neq l$.

Denote the probability that the j^{th} decision maker chooses the alternative k as

$$\begin{aligned} P_{jk} &= P\left(U_{jk} > U_{jl} \text{ for all } k \neq l\right) && (1.3)\\ &= P\left(V_{jk} + \epsilon_{jk} > V_{jl} + \epsilon_{jl} \text{ for all } k \neq l\right) && (1.4)\\ &= P\left(\epsilon_{jl} < \epsilon_{jk} + V_{jk} - V_{jl} \text{ for all } k \neq l\right). && (1.5) \end{aligned}$$

1.1.1 The Multinomial Logit Model

The demand model that we consider in this thesis is multinomial logit. The model is multinomial since the probability of choosing a product by a customer during a particular period is constant. In the logit model, probabilities are proportional to the exponential of the covariates' linear function associated with the j^{th} individual.

To obtain the logit model we assume that the ϵ_{jk} are independent and identically distributed Standard Gumbel with cumulative distribution function $F(x) = e^{-e^{(-x)}}$. If ϵ_{jk} is given, equation (1.5) provides the cumulative distribution for each ϵ_{jl} evaluated at $\epsilon_{jk} + V_{jk} - V_{jl}$. Thus

$$P_{jk} | \epsilon_{jk} = \prod_{l \neq k} e^{-e^{-(\epsilon_{jk} + V_{jk} - V_{jl})}}.$$

Since $\epsilon_{jk}$is random, the choice probability is the integral $P_{jk}|\epsilon_{jk}$ over all the values of ϵ_{jk} with respect to its density. Thus

$$P_{jk} = \int_{-\infty}^{\infty} \prod_{l \neq k} e^{-e^{-(\epsilon + V_{jk} - V_{jl})}} e^{-\epsilon} e^{-e^{-\epsilon}} d\epsilon.$$

Since $V_{jk} - V_{jk} = 0$, $e^{-e^{-\epsilon}} = e^{-e^{-(\epsilon + V_{jk} - V_{jk})}}$. Therefore,

$$\begin{aligned} P_{jk} &= \int_{-\infty}^{\infty} \prod_{l \neq k} e^{-e^{-(\epsilon + V_{jk} - V_{jl})}} e^{-\epsilon} e^{-e^{-(\epsilon + V_{jk} - V_{jk})}} d\epsilon \\ &= \int_{-\infty}^{\infty} \exp(-e^{-\epsilon} \sum_{l=1}^{n} e^{-(V_{jk} - V_{jl})}) e^{-\epsilon} d\epsilon. \end{aligned}$$

Let $s = e^{-\epsilon}$. Then $-ds = e^{-\epsilon} d\epsilon$. Thus by integration we obtain

$$\begin{aligned} P_{jk} &= \int_{0}^{\infty} \exp(-s \sum_{l=1}^{n} e^{-(V_{jk} - V_{jl})}) ds && (1.6) \\ &= \frac{e^{V_{jk}}}{\sum_{l=1}^{n} e^{V_{jl}}}. && (1.7) \end{aligned}$$

Since equation (1.7) is precisely the definition of the multinomial logit probability based on $V_{j1}, ..., V_{jn}$, we have shown that if the unobserved utility follows a standard Gumbel distribution, then the probabilities vector follows a multinomial logit model. McFadden (1974) shows the converse that is, if the choice probabilities follow the logit formula, then the unobserved utility is distributed standard Gumbel.

Then applying equation (1.1) for V_{jk} the probability of choosing k^{th} product by customer j can be derived as

$$P_{jk} = \frac{e^{\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{jk}}}{\sum\limits_{l=1}^{n} e^{\alpha_l + \boldsymbol{\beta}^T \boldsymbol{x}_{jl}}}. \tag{1.8}$$

1.1.2 Other Discrete Choice Models

There are other discrete choice models such as the multinomial probit model, the exponomial choice model, and the heteroscedastic exponomial choice model.

The probit model assumes that the errors are distributed jointly normal. This model can accommodate the errors' heteroscedasticity, while the logit model assumes that the errors are uncorrelated over alternatives. The probit model's limitation is that it requires the normal distributions of all the utilities' error components. The logit model is more popular than the probit model since the logit model has a closed form expression of the choice probabilities, whereas the probit model does not have a closed form.

Below, we derive the choice probabilities of the homoscedastic exponomial choice model. The homoscedastic exponomial choice model assumes that the $-\epsilon_{jkt}$ are independent and identically distributed exponential exponentially with rate λ. Let $\Gamma_{jkt} = -\epsilon_{jkt}$. (Alptekinoğlu and Semple, 2016)

Replacing ϵ by $-\Gamma$ in equation (1.4) we obtain

$$\begin{aligned} P_{jkt} &= P\left(V_{jkt} - \Gamma_{jkt} > V_{jlt} - \Gamma_{jlt} \text{ for all } k \neq l\right) \\ &= P\left(\Gamma_{jlt} > V_{jlt} - V_{jkt} + \Gamma_{jkt} \text{ for all } k \neq l\right) \\ &= \prod_{l \neq k} (1 - P\left(\Gamma_{jlt} \leq V_{jlt} - V_{jkt} + \Gamma_{jkt}\right) \\ &= \int_0^\infty \prod_{l \neq k} (1 - F\left(V_{jlt} - V_{jkt} + \gamma\right) f(\gamma) d\gamma \end{aligned} \tag{1.9}$$

where $f(\gamma) = \lambda e^{-\lambda\gamma}$ and $F(\gamma) = 1 - e^{-\lambda\gamma}$ for $\gamma \geq 0$. Since γ is exponentially distributed $f(\gamma) = 0$ and $F(\gamma) = 0$ for $\gamma < 0$. Equation (1.9) is defined for $V_{jlt} - V_{jkt} + \Gamma_{jkt} \geq 0$ since cumulative distribution is zero over the non positive domain.

Thus by integrating we obtain

$$\begin{aligned} P_{jkt} = &\frac{e^{-\lambda \sum_{l=k}^{n} (V_{jlt} - V_{k-1})}}{(n-k+1)} \\ &- \frac{e^{-\lambda \sum_{l=k-1}^{n} (V_{jlt} - V_{j(k-1)t})}}{(n-k+1)(n-k+2)} \\ &- \frac{e^{-\lambda \sum_{l=k-2}^{n} (V_{jlt} - V_{j(k-2)t})}}{(n-k+2)(n-k+3)} \\ &\quad . \\ &\quad . \\ &\quad . \\ &- \frac{e^{-\lambda \sum_{l=1}^{n} (V_{jlt} - V_{j1t})}}{(n-1)n}. \end{aligned} \tag{1.10}$$

We can substitute equation (1.1) into equation (1.10) to obtain the probability in terms of the parameters α and β.

The heteroscedastic exponomial choice model generalizes the exponomial choice model by including choice-specific variances for the utility's error terms (Alptekinoglu and Semple, 2018). In other words, we assume that Γ_{jkt} is exponentially distributed with rate λ_k.

1.2 EM Algorithms

EM is a general approach to find local maxima of a log-likelihood function using the iterative computation of maximum-likelihood estimates when the model depends on incomplete data. The term incomplete data refers to the observed data when a subset of the full data has unobserved or missing data. Complete data refers to the combination of both observed and unobserved data. The general framework for EM algorithms was explained by Dempster et al. (1977). The name EM algorithm arises since each iteration of an EM algorithm alternates between an expectation (E) step and a maximization (M) step.

Let $\mathbf{Y}$ be a random vector corresponding to observed data with probability distribution function $f(\mathbf{y};\boldsymbol{\theta})$ where $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)^T$ is a vector of unknown parameters. Let $\mathbf{X}$ denote the vector of complete data with probability density function $g(\mathbf{x};\boldsymbol{\theta})$. $\mathbf{Y}$ is a vector of incomplete data if there is a many-to-one mapping from $\mathcal{X}$ to $\mathcal{Y}$ where $\mathcal{X}$, $\mathcal{Y}$ are sample spaces. The observed data vector $\mathbf{y}$ is a realization from $\mathcal{Y}$ such that $\mathbf{y} = \mathbf{y}(\mathbf{x})$ and $\mathbf{x}$ in $\mathcal{X}$ is a realization of the complete data.

The marginal likelihood of the observed data (the incomplete data likelihood) is related to the complete data likelihood by

$$L(\boldsymbol{\theta};\mathbf{y}) = f(\mathbf{y};\boldsymbol{\theta}) = \int_{\mathcal{X}(\mathbf{y})} g(\mathbf{x};\boldsymbol{\theta})d\mathbf{x},$$

where $\mathcal{X}(\mathbf{y})$ is the subset of $\mathcal{X}$ determined by the equation $\mathbf{y} = \mathbf{y}(\mathbf{x})$.

The aim is to find the estimates of $\boldsymbol{\theta}$ that maximize $L(\boldsymbol{\theta};\mathbf{y})$. We can use EM algorithms to find these estimates when the maximum likelihood is complicated due to missing or truncated data.

1.2.1 Derivation of EM algorithms

Following Dempster et al. (1977) and McLachlan and Krishnan (2008), let

$$h(\mathbf{x}|\mathbf{y};\boldsymbol{\theta}) = \frac{g(\mathbf{x};\boldsymbol{\theta})}{f(\mathbf{y};\boldsymbol{\theta})}.$$

Then

$$L(\boldsymbol{\theta}; \mathbf{x}) = \frac{g(\mathbf{x}; \boldsymbol{\theta})}{h(\mathbf{x}|\mathbf{y}; \boldsymbol{\theta})}$$

so that

$$\log L(\boldsymbol{\theta}; \mathbf{y}) = \log g(\mathbf{x}; \boldsymbol{\theta}) - \log h(\mathbf{x}|\mathbf{y}; \boldsymbol{\theta})$$

Take expectation of both sides with respect to the conditional distribution of $\mathbf{X}$ given $\mathbf{Y} = \mathbf{y}$ for fixed $\boldsymbol{\theta}^{(w)}$. For a given $\mathbf{Y} = \mathbf{y}$,

$$E_{\boldsymbol{\theta}^{(w)}}[\log L(\boldsymbol{\theta}; \mathbf{Y})|\mathbf{y}] = \log L(\boldsymbol{\theta}; \mathbf{y}).$$

Therefore,

$$\begin{aligned}\log L(\boldsymbol{\theta}; \mathbf{y}) &= E_{\boldsymbol{\theta}^{(w)}}[\log g(\mathbf{X}; \boldsymbol{\theta})|\mathbf{y}] - E_{\boldsymbol{\theta}^{(w)}}[\log h(\mathbf{X}|\mathbf{y}; \boldsymbol{\theta})|\mathbf{y}] \\ &= Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(w)}) - H(\boldsymbol{\theta}; \boldsymbol{\theta}^{(w)})\end{aligned}$$

where

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(w)}) = E_{\boldsymbol{\theta}^{(w)}}[\log g(\mathbf{X}; \boldsymbol{\theta})|\mathbf{y}]$$

and

$$H(\boldsymbol{\theta}; \boldsymbol{\theta}^{(w)}) = E_{\boldsymbol{\theta}^{(w)}}[\log h(\mathbf{X}|\mathbf{y}; \boldsymbol{\theta})|\mathbf{y}].$$

Choose $\boldsymbol{\theta}^{(w+1)}$ such that $Q(\boldsymbol{\theta}^{(w+1)}; \boldsymbol{\theta}^{(w)}) \geq Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(w)})$ for all $\boldsymbol{\theta}$. Therefore,

$$\begin{aligned}\log L(\boldsymbol{\theta}^{(w+1)}; \mathbf{y}) - \log L(\boldsymbol{\theta}^{(w)}; \mathbf{y}) = Q(\boldsymbol{\theta}^{(w+1)}; \boldsymbol{\theta}^{(w)}) - Q(\boldsymbol{\theta}^{(w)}; \boldsymbol{\theta}^{(w)}) - \\ (H(\boldsymbol{\theta}^{(w+1)}; \boldsymbol{\theta}^{(w)}) - H(\boldsymbol{\theta}^{(w)}; \boldsymbol{\theta}^{(w)})).\end{aligned} \tag{1.11}$$

For any $\boldsymbol{\theta}$, consider

$$\begin{aligned}
H(\boldsymbol{\theta};\boldsymbol{\theta}^{(w)}) - H(\boldsymbol{\theta}^{(w)};\boldsymbol{\theta}^{(w)}) &= E_{\boldsymbol{\theta}^{(w)}}[\log h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta})|\mathbf{y}] - E_{\boldsymbol{\theta}^{(w)}}[\log h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}^{(w)})|\mathbf{y}] \\
&= E_{\boldsymbol{\theta}^{(w)}}[\log h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}) - \log h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}^{(w)})|\mathbf{y}] \\
&= E_{\boldsymbol{\theta}^{(w)}}\left[\log \frac{h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta})}{h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}^{(w)})}|\mathbf{y}\right] \\
&\leq \log E_{\boldsymbol{\theta}^{(w)}}\left[\frac{h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta})}{h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}^{(w)})}|\mathbf{y}\right] \\
&= \log \int_{\mathcal{X}(\mathbf{y})} \frac{h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta})}{h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}^{(w)})} * h(\mathbf{X}|\mathbf{y};\boldsymbol{\theta}^{(w)}) d\mathbf{x} \\
&= \log \int_{\mathcal{X}(\mathbf{y})} \frac{g(\mathbf{x};\boldsymbol{\theta})}{f(\mathbf{y};\boldsymbol{\theta})} d\mathbf{x} \\
&= \log \frac{1}{f(\mathbf{y};\boldsymbol{\theta})} \int_{\mathcal{X}(\mathbf{y})} g(\mathbf{x};\boldsymbol{\theta}) d\mathbf{x} \\
&= 0
\end{aligned} \tag{1.12}$$

which follows as a result of the concavity of the logarithm and Jensen's inequality.

By applying result of equation (1.12) in equation (1.11) we obtain,

$$\log L(\boldsymbol{\theta}^{(w+1)};\mathbf{y}) - \log L(\boldsymbol{\theta}^{(w)};\mathbf{y}) \geq Q(\boldsymbol{\theta}^{(w+1)};\boldsymbol{\theta}^{(w)}) - Q(\boldsymbol{\theta}^{(w)};\boldsymbol{\theta}^{(w)}).$$

Since $Q(\boldsymbol{\theta}^{(w+1)};\boldsymbol{\theta}^{(w)}) \geq Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(w)})$ for all $\boldsymbol{\theta}$,

$$\log L(\boldsymbol{\theta}^{(w+1)};\mathbf{y}) \geq \log L(\boldsymbol{\theta}^{(w)};\mathbf{y}).$$

Therefore, the incomplete data likelihood is never decreased after an EM iteration. Therefore, it monotonically converges to some $L^* = L(\boldsymbol{\theta}^*;\mathbf{y})$ if the likelihood is bounded above.

We summarize the E- and M-steps of an EM algorithm in which w denotes the iteration number as follows:

Expectation (E) step: Compute

$$Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(w)}) = E_{\boldsymbol{\theta}^{(w)}}[\log g(\mathbf{X};\boldsymbol{\theta})|\mathbf{y}]$$

Maximization (M) step:

$$\boldsymbol{\theta}^{(w+1)} = \arg\max_{\theta}\ Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(w)})$$

2 Parameter estimation in the presence of product missingness

In this chapter, we apply the multinomial logit model to a grocery purchase dataset. This dataset is of interest because it does not contain information regarding product availability. Our objective is to introduce product availability as a latent variable. We summarize previous research that introduces latent variables to demand estimates. We illustrate via an example that assumptions about zero counts of sales i.e. whether 0 means a brand is available or not, can make a difference. Then we introduce the new model after incorporating product availability into the model and used an EM algorithm to obtain maximum likelihood estimates. We explain the stopping rule used in the EM algorithm and derive the observed data log-likelihood in section 2.5. Further, we use a simulation study to compare the models' prediction accuracy and fit the new model to the grocery purchase dataset.

2.1 Application of MNL Model

We apply the multinomial logit model combined with EM algorithms to the grocery purchase dataset in which we cannot observe the availability of products. The dataset used in this thesis is the grocery purchase dataset used in Alptekinoglu and Semple (2018). They have obtained the panel data set of grocery purchases from Briesch et al. (2013). It consists of aggregated households' purchase records among multiple retail outlets over 104 weeks in Charlotte, South Carolina. Alptekinoglu and Semple (2018) uses the top 30 product categories according to market penetration. Few product categories are candy, carbonated drinks, and vegetables which consist of 13, 47, and 19 brands, respectively. In each category, there is a different number of brands.

The dataset contains the variables: brand name, week number, number of units sold in a week, the brand's price during the week, feature, and display. Feature and display are binary variables related to promotional activities. They denote whether the brand is featured or used as a display product in a week. The number of units sold is standardized because the package size is different within and across brands. The data we use fromAlptekinoglu and Semple (2018) have converted all purchase quantities in a given category to a standardized value. Further, we do not have customer-level data, and the dataset consists of aggregated data per brand over the weeks.

We analyze whether we can obtain information regarding the product availability from the variables feature and display because if there are featured or displayed products, that means they are available during that week. However, there are no brands that have 0 units but is a featured or a displayed product. That means when the feature or display variables have a value of 1, the number of units for those brands is greater than zero.

2.2 Previous Research

For simplicity, most of the demand models rely on observed data. However, these models do not consider the effect to demand estimates when the information on product availability does not exist. This thesis highlights the difference in parameter estimates when a product is considered available, although there are no sales records of the product, and when a product is considered not available when there are zero sales. Due to this difference, we introduce the unobserved variable of product availability to the demand model. Therefore, our work is related to the research that introduces latent variables to demand estimates.

Talluri and van Ryzin (2004) develop an estimation procedure for choice models when we cannot observe 'no purchase' options. In this setting, the no-purchase option is due to no arrivals of customers or arrivals that do not purchase. Therefore, they develop a model based on the expectation-maximization method that estimates arrival rates and choice model parameters. Their work is different from ours because they know the products offered in a period whereas we do not observe this.

Conlon and Mortimer (2013) develop a method for incorporating latent stock-out periods into demand estimates. A stock-out period is a period that there is no inventory of a specific item. Their situation is different from ours because they observe which products are out-of-stock every four hours, but we do not have any information on product availability. However, they do not observe when the products run out. Thus, they have unobserved choice-set regimes and introduce sales before and after the stock-out as latent variables. Therefore, they develop a method that uses the EM algorithm to prove that failing to account for stock-out periods can result in bias in demand estimates.

Musalem et al. (2010) simulate a time-varying set of available alternatives to address the stock-out problem using their posterior distribution. Similar to Conlon and Mortimer (2013), they know which products are not available at the end of the period, but they do not know when they run out of stock. They use the Markov-Chain Monte Carlo approach with sampling using Bayesian methods to simplify the estimation compared with an EM approach. Conlon and Mortimer (2013) and Musalem et al. (2010) have similar demand models applied to different datasets.

Closest to our work is Vulcano et al. (2012), which uses an EM method to estimate the number of customers who would have purchased if all products are in stock. Their demand model combines a multinomial logit choice model with a non-homogeneous Poisson model of customers' arrivals over multiple periods. They view the observed sales as incomplete observations of demand that would have been observed if all the products had been available in all the periods. Their work is different from ours because they assume that the set of products available for sale in a period is known. Their latent variable is whether a product is purchased because it is the customer's preferred product or a substitute for an out-of-stock product.

2.3 Illustrative Example

To identify the importance of introducing product availability in modeling for demand estimates, we illustrate the differences in parameter estimates according to the two scenarios mentioned in chapter 1. For that, we use the same grocery purchase dataset introduced in section 2.1. As an example, we use the category candy, which has 13 brands.

Some brands have zero sales in some weeks. However, there is no information on whether those brands were not available during those weeks, or no one had purchased those brands during those weeks. Thus, we consider the two scenarios and estimate the model parameters, as shown in the table 2.1.

- all available: Assuming all the brands are available in each week
- missing 0: Assuming brands with zero units are not available in that week

In the example we consider here in the thesis, each customer makes decisions independently, but the $\boldsymbol{x}_{jk}$ does not depend on j because we have no customer-level data. This allows us to aggregate by customer. We can use utility maximizing theory as described in Section (1.1) and obtain $P\left(U_k > U_l \text{ for all } k \neq l\right)$. Since the data depend on the week, we expand this as $P\left(U_{kt} > U_{lt} \text{ for all } k \neq l\right)$. Removing the dependency on the customer we can determine the probability that k^{th} product is chosen in week t.

For all available this probability is

$$P_{kt} = \frac{e^{\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}}}{\sum_{l=1}^{n} e^{\alpha_l + \boldsymbol{\beta}^T \boldsymbol{x}_{lt}}}. \tag{2.1}$$

To find the estimates α_k and $\boldsymbol{\beta}$ that maximize the utility of the decision-maker, first we calculate the log-likelihood assuming independent purchases aggregating over T weeks and n products at time t, $1 \leq t \leq T$. Letting C_{kt} denote the number of purchases of product k in week t, the log-likelihood function is

$$\begin{aligned}
\ell(\alpha, \beta) &= \sum_{t=1}^{T} \sum_{k=1}^{n} C_{kt} \ \log P_{kt} \\
&= \sum_{t=1}^{T} \sum_{k=1}^{n} C_{kt} \ [\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}] - \sum_{t=1}^{T} \sum_{k=1}^{n} C_{kt} \ \log \sum_{l=1}^{n} \ e^{\alpha_l + \boldsymbol{\beta}^T \boldsymbol{x}_{lt}}.
\end{aligned} \tag{2.2}$$

For missing 0 scenario, the probability of choosing k^{th} product in week t is

$$P_{kt} = \frac{I\{C_{kt} > 0\} e^{\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}}}{\sum_{l=1}^{n} \ I\{C_{lt} > 0\} e^{\alpha_l + \boldsymbol{\beta}^T \boldsymbol{x}_{lt}}}. \tag{2.3}$$

Then by applying this probability in equation (2.2) we can derive the log-likelihood function as

$$\ell(\alpha, \beta) = \sum_{t=1}^{T} \sum_{k=1}^{n} C_{kt} \ [\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}] - \sum_{t=1}^{T} \sum_{k=1}^{n} C_{kt} \ \log \sum_{l=1}^{n} \ I\{C_{kt} > 0\} e^{\alpha_l + \boldsymbol{\beta}^T \boldsymbol{x}_{lt}}. \tag{2.4}$$

Based on these log-likelihoods, we can obtain the parameter estimates for the two scenarios. According to the table 2.1, we can see that the parameter estimates are different under the two assumptions. The ranks in the table 2.1 are based on the brands' desirability, which is given by α_k parameters. It is visible that the ranks of brands under the two assumptions are different. For example, the brand with the rank 3 according to all available has the rank 5 in the missing 0 scenario. Therefore, according to the assumptions, the decisions we make about products are different. Thus, we can see that failing to account for product availability can lead to inaccurate demand estimates. Therefore, we consider the approach of introducing a latent variable that indicates product availability.

Table 2.1: Maximum likelihood estimates of α_k parameters in equations (2.1), (2.3)

Brand	**all available (S1)** (Rank)	**missing 0 (S2)** (Rank)	**Weeks with zero sales**	**Absolute difference of S1 and S2**
brachs	-4.796 (10)	-4.029 (9)	63	0.767
leaf pay day	-5.177 (12)	-4.295 (12)	60	0.882
sweetarts	-5.09 (11)	-4.229 (11)	58	0.861
skittles	-5.354 (13)	-4.602 (13)	58	0.752
brachs milk maid	-4.557 (9)	-3.809 (6)	55	0.748
altoids	-1.52 (2)	-0.628 (2)	40	0.892
lifesavers creme sav	–3.751 (3)	-3.624(4)	40	0.127
starburst	-4.4 (8)	-4.024 (8)	33	0.376
jolly rancher	-4.376 (7)	-4.032 (10)	32	0.344
van melles mentos	-3.986 (4)	-3.574 (3)	30	0.412
private label	-4.158 (5)	-3.948 (7)	25	0.21
lifesavers	-4.158(5)	-3.632 (5)	12	0.526
osg	0 (1)	0 (1)	0	0

Table 2.2: Maximum likelihood estimates of β parameters in equations (2.1), (2.3)

Product	**all available**	**missing 0**
price	-0.336	-0.372
feature	0.56	0.184
display	1.3	0.924

In addition to comparing the parameter estimates across a category, we compare parameter estimates of price, feature, and display among all the product categories in the dataset, assuming all the brands are available each week. According to the figure 2.1, we can see that the household necessities such as toilet papers, fresh eggs, and vegetables have the most negative impact on utility from price. The figure 2.2 shows that whether the product is feature advertised or not has a higher positive impact on products such as laundry detergent, ice cream, and paper towels. Luncheon meats, candy, and oil have the highest positive impact on utility from the variable display.

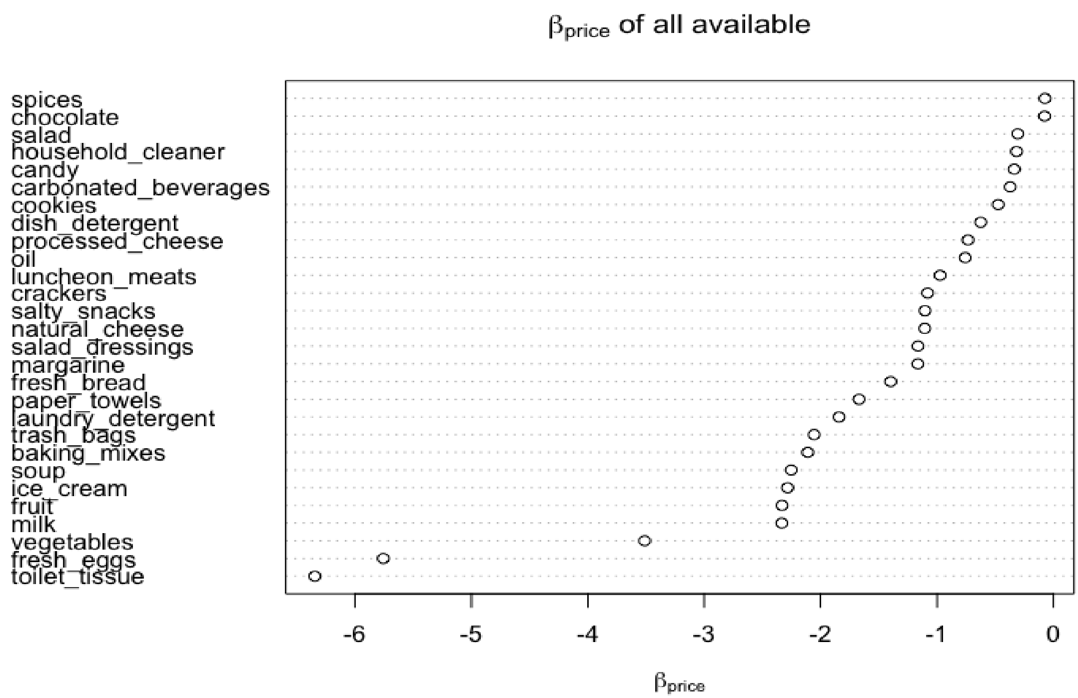

Figure 2.1: β_{price} of all available

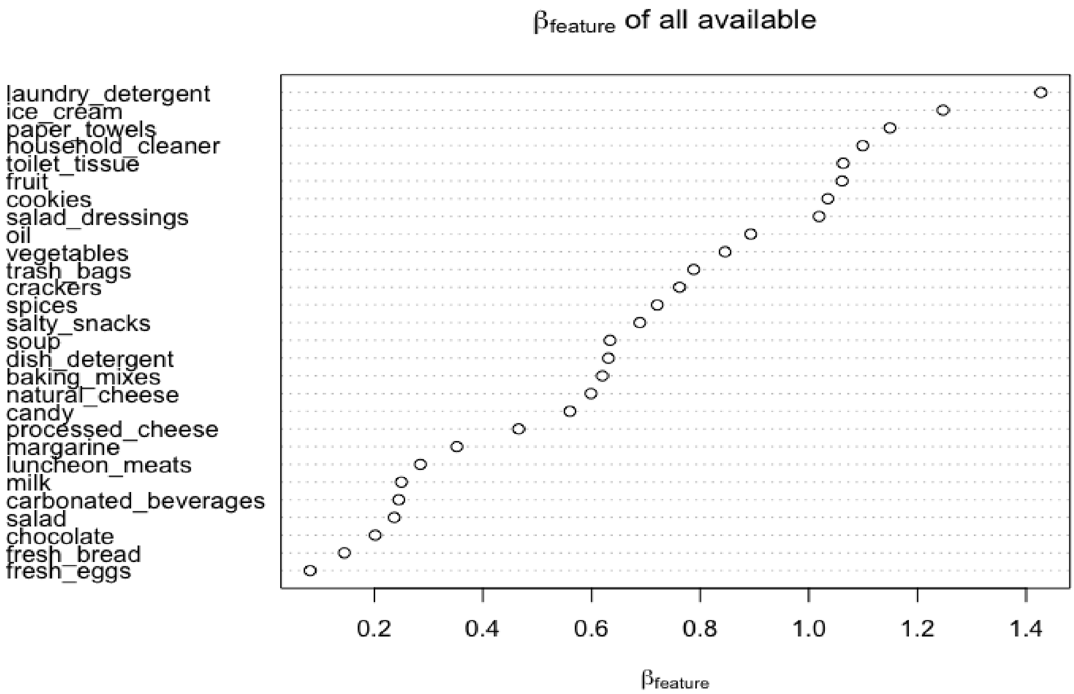

Figure 2.2: $\beta_{feature}$ of all available

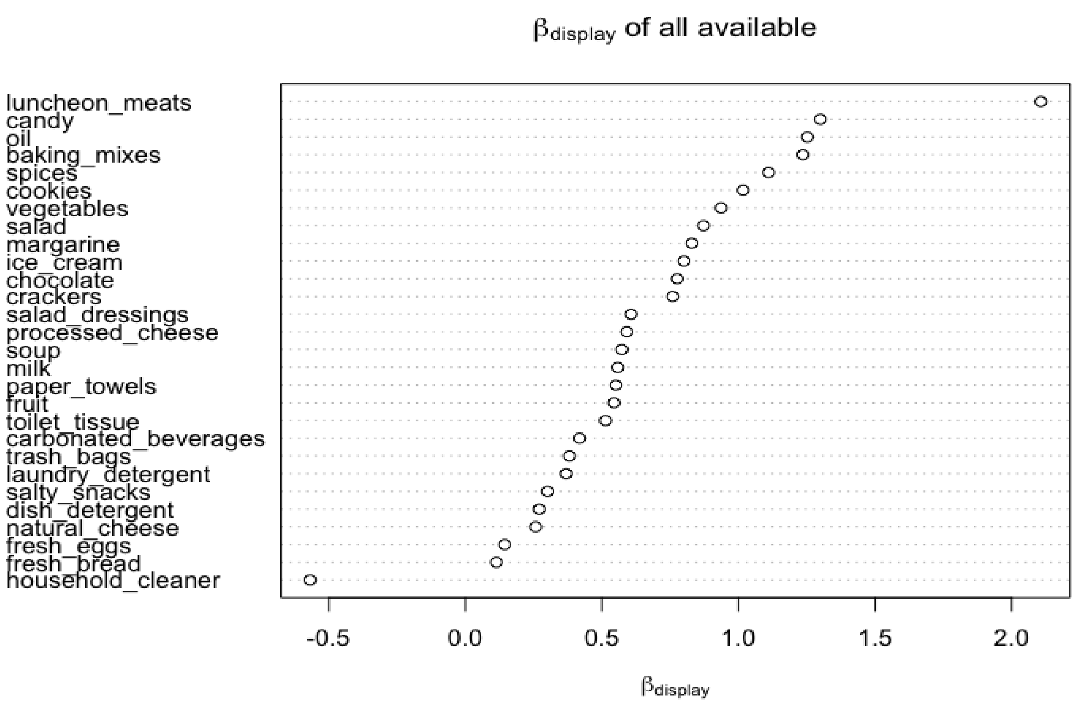

Figure 2.3: $\beta_{display}$ of all available

2.4 Incorporate Product Availability to The Model

To account for product availability, we introduce the unobserved variable Z_{kt}, which indicates whether product k is available or not in the week t. Introducing this variable alter equation (2.1) as

$$P_{kt} = \frac{Z_{kt}\ e^{\alpha_k+\boldsymbol{\beta}^T\boldsymbol{x}_{kt}}}{\sum_{l=1}^{n} Z_{lt}\ e^{\alpha_l+\boldsymbol{\beta}^T\boldsymbol{x}_{lt}}} \tag{2.5}$$

which is the probability that k^{th} product is chosen in week t where Z_{kt}= I$\{k^{th}$ product available in week $t\}$.

To find the estimates α_k and $\boldsymbol{\beta}$ that maximize the utility of the decision-maker, we apply the above probability in equation (2.2) and derive the complete data log-likelihood as

$$\begin{aligned}\ell_c(\alpha,\beta) &= \sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt}\log P_{kt} \\ &= \sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt}\ [\alpha_k + \boldsymbol{\beta}^T\boldsymbol{x}_{kt}] - \sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt}\ \log\sum_{l=1}^{n}\ Z_{lt}e^{\alpha_l}e^{\boldsymbol{\beta}^T\boldsymbol{x}_{lt}},\end{aligned} \tag{2.6}$$

where C_{kt} denotes the number of purchases. If k^{th} product is not available in week t, C_{kt} is zero and if k^{th} product is available in week t, Z_{kt} is one which makes $\log Z_{kt}$ zero. Therefore, $C_{kt}\log Z_{kt}$ is always zero.

Since we introduce an unobserved latent variable to account for product availability, the probability of choosing k^{th} product in week t depends on the introduced variable as shown in equation (2.5). Therefore, the log-likelihood function given in equation (2.6) includes the unobserved variable denoted by Z_{kt} which denotes whether the product is available in week t or not. Expectation-Maximization (EM) is an iterative procedure to find the maximum likelihood of a statistical model that incorporates unobserved latent variables.

For discrete missing data denoted collectively as z, discrete observed dependent variables denoted collectively as y, and parameters $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta})$, the log-likelihood function is $\ell_c(\boldsymbol{\theta}) = \log \sum_z P(y|z,\boldsymbol{\theta})P(z|\boldsymbol{\theta})$, where $P(y|z,\boldsymbol{\theta})$ is the probability of outcomes conditional on z, and $P(z|\boldsymbol{\theta})$ is the density of missing data. This function can be maximized through a recursive method called EM algorithm. Start with initial values of the parameters labeled $\boldsymbol{\theta}^{(w)}$ for $w = 0$ and update the parameters in a given iteration denoted $\boldsymbol{\theta}^{(w)}$ through the function given in equation (2.7).

$$E_{\boldsymbol{\theta}^{(w)}}[\ell_c(\boldsymbol{\theta})|Y = y] = \sum_z h(z|y, \boldsymbol{\theta}^{(w)}) \log P(y, z|\boldsymbol{\theta}) \tag{2.7}$$

$$= \sum_z h(z|y, \boldsymbol{\theta}^{(w)}) \log(P(y|z, \boldsymbol{\theta})P(z|\boldsymbol{\theta})), \tag{2.8}$$

where the conditional density h is calculated using the current initial value of the parameters, $\boldsymbol{\theta}^{(w)}$, and $\log P(y, z|\boldsymbol{\theta})$ is the complete-data log-likelihood function given in equation (2.6). An EM algorithm involves repeatedly maximizing equation (2.7). Starting with initial values, update the parameters in each recursion denoted as $\boldsymbol{\theta}^{(w+1)}$ by the formula

$$\boldsymbol{\theta}^{(w+1)} = \arg\max_{\boldsymbol{\theta}} \ E_{\boldsymbol{\theta}^{(w)}}[\ell_c(\boldsymbol{\theta})|Y = y]. \tag{2.9}$$

The equation (2.6) denotes the completed data log-likelihood where the observed variables is C_{kt}, parameters are α_k and $\boldsymbol{\beta}$ and discrete missing variable is Z_{kt}. Therefore, to apply an EM procedure to maximize the log-likelihood, we need to find

$$E_{\alpha^{(w)},\boldsymbol{\beta}^{(\boldsymbol{w})}}[\sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt} \ [\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}] - \sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt} \ \log \sum_{l=1}^{n} \ Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T \boldsymbol{x}_{lt}} |\boldsymbol{C}]$$

for a fixed t where $\boldsymbol{C} = (C_{1t}, ..., C_{nt})$. Since $Z_{lt} = 1$ whenever $C_{lt} > 0$, letting N denote the set of $l : C_{lt} = 0$, we need to find the joint conditional distribution of $\{Z_{lt} : l \in N\}$ given $C_{1t}, ..., C_{nt}$ and using $\alpha^{(w)}, \ \boldsymbol{\beta}^{(w)}$.

Let $\boldsymbol{Z}_t = \{Z_{1t}, ..., Z_{nt}\}$. Then, for a fixed week t, the complete log-likelihood for $\boldsymbol{C}$ is,

$$\ell(\alpha, \beta) = \sum_{k=1}^{n} C_{kt} \ [\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}] - \sum_{k=1}^{n} C_{kt} \ \log \sum_{l=1}^{n} \ Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T \boldsymbol{x}_{lt}},$$

and complete-likelihood is

$$L(\alpha, \beta) = \prod_{k=1}^{n} \left[\frac{\exp(\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt})}{\sum_{l=1}^{n} z_l \exp(\alpha_l + \boldsymbol{\beta}^T \boldsymbol{x}_{lt})} \right]^{C_{kt}}.$$

Therefore, based on the definition of conditional probability,

$$P(\boldsymbol{Z}_t = \boldsymbol{z}|\boldsymbol{C}) \propto \begin{cases} \prod_{k=1}^{n} \left[\frac{\exp(\alpha_k+\boldsymbol{\beta}^T\boldsymbol{x}_{kt})}{\sum_{l=1}^{n} z_l \exp(\alpha_l+\boldsymbol{\beta}^T\boldsymbol{x}_{lt})}\right]^{C_{kt}} & \text{if } \boldsymbol{z} \text{ is legal given } \boldsymbol{c} \\ 0 & \text{if} \boldsymbol{z} \text{ is not legal given } \boldsymbol{c} \end{cases}$$

where $\boldsymbol{z}$ is not legal given $\boldsymbol{c}$ if $z_{lt} \neq 1$ whenever $C_{lt} > 0$. Thus,

$$\begin{aligned} P_{\alpha^{(w)},\boldsymbol{\beta}^{(w)}}(\boldsymbol{Z}_t = \boldsymbol{z}|\boldsymbol{C}) = p_t^{(w)}(\boldsymbol{z}) &= \kappa_t \prod_{r=1}^{n} \left[\frac{\exp(\alpha_r^{(w)} + \boldsymbol{\beta}^{(w)T}\boldsymbol{x}_{rt})}{\sum_{s=1}^{n} z_s \exp(\alpha_s^{(w)} + \boldsymbol{\beta}^{(w)T}\boldsymbol{x}_{st})}\right]^{C_{rt}} \\ &= \kappa_t \frac{\prod_{r=1}^{n}[\exp(\alpha_r^{(w)} + \boldsymbol{\beta}^{(w)T}\boldsymbol{x}_{rt})]^{C_{rt}}}{\prod_{r=1}^{n}[\sum_{s=1}^{n} z_s \exp(\alpha_s^{(w)} + \boldsymbol{\beta}^{(w)T}\boldsymbol{x}_{st})]^{C_{rt}}}, \end{aligned} \tag{2.10}$$

where κ_t is a constant of proportionality that must be determined by summing the right hand side of equation (2.10) over all possible values of the vector $\boldsymbol{z}$. Once κ is known, the value of $h(z|y, \boldsymbol{\theta})$ from equation (2.7) is given by the right side of equation (2.10). Therefore,

$$\begin{aligned} E_{\alpha^{(w)},\boldsymbol{\beta}^{(w)}}[\ell_c(\alpha, \boldsymbol{\beta})|\boldsymbol{C}] = &\sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt} \left[\alpha_k + \boldsymbol{\beta}^T\boldsymbol{x}_{kt}\right] \\ &- \sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt} \sum_{z\in S_t} \left(p_t^{(w)}(\boldsymbol{z}) \log \sum_{l=1}^{n} z_l \exp(\alpha_l + \boldsymbol{\beta}^T\boldsymbol{x}_{lt})\right), \end{aligned} \tag{2.11}$$

where $S_t = \{\boldsymbol{z} \in (0,1)^n : z_l = 1 \text{ whenever } C_{it} > 0\}$. The values corresponding to equation (2.9) is obtained by maximizing $E_{\alpha^{(w)},\boldsymbol{\beta}^{(w)}}[\ell_c(\alpha, \boldsymbol{\beta})|\boldsymbol{C}]$ with respect to each individual parameter.

Thus, the EM algorithm can be summarized as,

- Initialize $(\alpha^{(0)}, \boldsymbol{\beta}^{(0)})$
- E-Step: Given by equation (2.11)
- M-step: $(\alpha^{(w+1)}, \boldsymbol{\beta}^{(w+1)}) \in \arg\max_{(\alpha,\boldsymbol{\beta})} E_{\alpha^{(w)},\boldsymbol{\beta}^{(w)}}[\ell_c(\alpha, \boldsymbol{\beta})|\boldsymbol{C}]$
- Repeat until convergence.

2.5 EM Algorithms Stopping Rule

Bohning et al. (1994) states that an EM algorithm converges linearly and the rate of convergence is often very slow. Often it is suggested to stop the algorithm when $|\ell^{(w+1)} - \ell^{(w)}| \leq \epsilon$, where $\ell^{(w)}$ is the observed data log-likelihood calculated at w^{th} iteration with small positive constant ϵ. However, this is a lack of progression criterion than a stopping rule. Therefore, they developed a method using the Aitken acceleration on log-likelihood estimates. Suppose $\hat{\ell}$ is the log-likelihood at the maximum likelihood solution that is,

$$\ell^{(w+1)} - \hat{\ell} \simeq c(\ell^{(w)} - \hat{\ell}) \text{ for } 0<c<1. \tag{2.12}$$

$$\begin{aligned}\hat{\ell} &\simeq \frac{\ell^{(w+1)} - c\ell^{(w)}}{1-c} \\ &\simeq \frac{\ell^{(w+1)}(1-c) + \ell^{(w+1)} - \ell^{(w)}}{1-c}.\end{aligned}$$

The Aitken accelerated estimate of $\hat{\ell}$ is

$$\hat{\ell}^{(w)} = \ell^{(w-1)} + \frac{1}{1-c^{(w)}}(\ell^{(w)} - \ell^{(w-1)}), \text{ where}$$

$$c^{(w)} = \frac{(\ell^{(w+1)} - \ell^{(w)})}{(\ell^{(w)} - \ell^{(w-1)})}.$$

Based on equation (2.12) we can see that $c^{(w)}$ is an estimate of

$$\lim_{w\to\infty} \frac{(\hat{\ell} - \ell^{(w)})}{(\hat{\ell} - \ell^{(w-1)})}.$$

Therefore, the stopping rule is

$$\text{Stop EM if } |\hat{\ell}^{(w)} - \ell^{(w)}| < \epsilon.$$

Lindsay (1995) mentions that if parameter estimators' target accuracy has a standard error of 0.1 then $\epsilon = 0.005$ is a meaningful statistical goal.

To apply this stopping rule to our algorithm, we need to derive the observed data log-likelihood from complete data log-likelihood in equation (2.6). The complete data likelihood is as follows,

$$L_c(\alpha,\boldsymbol{\beta}) = \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt}\ [\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}] - \sum_{t=1}^{T}\sum_{k=1}^{n} C_{kt}\ \log \sum_{l=1}^{n}\ Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T \boldsymbol{x}_{lt}}\right)$$

$$= \prod_{t=1}^{T}\prod_{k=1}^{n}\left[\frac{\exp(\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt})}{\sum\limits_{l=1}^{n}\ Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T \boldsymbol{x}_{lt}}}\right]^{C_{kt}}.$$

Therefore, the observed data likelihood is,

$$L(\alpha,\boldsymbol{\beta}) = \sum_{Z}\prod_{t=1}^{T}\prod_{k=1}^{n}\left[\frac{\exp(\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt})}{\sum\limits_{l=1}^{n}\ Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T \boldsymbol{x}_{lt}}}\right]^{C_{kt}}$$

$$= \prod_{t=1}^{T}\sum_{Z_t}\prod_{k=1}^{n}\left[\frac{\exp(\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt})}{\sum\limits_{l=1}^{n}\ Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T \boldsymbol{x}_{lt}}}\right]^{C_{kt}}$$

$$= \prod_{t=1}^{T}\prod_{k=1}^{n}[\exp(\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt})]^{C_{kt}} \sum_{Z_t} \frac{1}{[\sum\limits_{l=1}^{n} Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T}]^{\sum\limits_{k=1}^{n} C_{kt}}}.$$

Then, the observed data log-likelihood is,

$$\ell(\alpha,\boldsymbol{\beta}) = \sum_{t=1}^{T}\log\left[\prod_{k=1}^{n}[\exp(\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt})]^{C_{kt}} \sum_{Z_t} \frac{1}{[\sum\limits_{l=1}^{n} Z_{lt} e^{\alpha_l} e^{\boldsymbol{\beta}^T \boldsymbol{x}_{lt}}]^{\sum\limits_{k=1}^{n} C_{kt}}}\right]$$

$$= \sum_{t=1}^{T}\left[\sum_{k=1}^{n} C_{kt}(\alpha_k + \boldsymbol{\beta}^T \boldsymbol{x}_{kt}) + \log\left(\sum_{Z_t} \frac{1}{[\sum\limits_{l=1}^{n} Z_{lt}\exp(\alpha_l + \boldsymbol{\beta}^T \boldsymbol{x}_{lt})]^{\sum\limits_{k=1}^{n} C_{kt}}}\right)\right].$$

2.6 Simulation Study

We conduct a simulation analysis to determine the three models' prediction accuracy using the measure Root-Mean-Square Error (RMSE). We use three levels of the number of units per week (40, 60, 100). Then we assign two levels of unavailable products for each case (20, 30). The number of products with zero units is the summation of the number of unavailable products and the number of products for which the number of units is zero, based on the randomly generated number of units per brand per week. We record the RMSE after 150 repetitions.

The table (2.3) shows that the model that includes the product availability as a missing variable (RMSE_MLE) achieves the best (lowest) RMSE value in each case. Therefore, it is evident that the model with product availability as a missing variable is better than considering all products are available or removing all the products with zero sales units.

Table 2.3: Predition accuracy of the three models

Expected number of units per week	**40**		**60**		**100**	
Number of unavailable products	**20**	**30**	**20**	**30**	**20**	**30**
Average number of products with 0 units	**91.7**	**93.3**	**73.8**	**87.3**	**54.2**	**59.9**
RMSE_MLE %	0.429	0.418	0.332	0.317	0.233	0.22
RMSE_all available %	0.70	0.792	0.684	0.783	0.663	0.778
RMSE_missing 0 %	0.684	0.651	0.469	0.345	0.285	0.272

2.7 Grocery Purchase Dataset Results

We use an EM algorithm and direct optimization of the observed data log-likelihood to obtain the parameter estimates when introducing the unobserved latent variable to account for product availability. We applied this method to the illustrative example in section 2.3. According to the table (2.4) we can see that the rank of the products based on their intrinsic desirability is according to the model used. The product that has the rank 7 using the MLE-based method has ranks 10 and 9 according to all available and missing 0 scenarios, respectively. Further, we can see that consumer sensitivity to price, feature, and display are also different based on the model.

Table 2.4: Maximum likelihood estimates of α_k parameters in equations (2.5) of Candy - *Directly calculating the MLE required 1.397 secs and the EM algorithm required 42.940 secs (31 times as long)*

Products	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
altoids	-1.41 (2)	-1.52 (2)	-0.628 (2)
lifesavers	-3.66 (3)	-3.751 (3)	-3.624 (4)
van melles mentos	-3.769 (4)	-3.986 (4)	-3.574 (3)
lifesavers creme sav	-3.828 (5)	-4.158 (5)	-3.632 (5)
private label	-3.964 (6)	-4.158 (5)	-3.948 (7)
brachs milk maid	-4.015 (7)	-4.557 (9)	-3.809 (6)
starburst	-4.122 (8)	-4.4 (8)	-4.024 (8)
jolly rancher	-4.126 (9)	-4.376 (7)	-4.032 (10)
brachs	-4.259 (10)	-4.796 (10)	-4.029 (9)
sweetarts	-4.585 (11)	-5.09 (11)	-4.229 (11)
leaf pay day	-4.738 (12)	-5.177 (12)	-4.295 (12)
skittles	-4.968 (13)	-5.354 (13)	-4.602 (13)

Table 2.5: Maximum likelihood estimates of β parameters in equations (2.5)

Product	**MLE**	**all available**	**missing 0**
price	-0.348	-0.336	-0.372
feature	0.281	0.56	0.184
display	1.023	1.3	0.924

There are 30 product categories in the grocery purchase dataset. However, we could obtain the results using the EM algorithm in section 2.4 or the direct optimization method for 28 product categories out of 30. This is because, for each week, the latent variable ($\mathbf{z}$) is a matrix with a column size of the number of products and row size of the exponentiated number of 0 counts of brands in that week. The methods to obtain MLE involve looping over all the rows of the matrix $\mathbf{z}$ for each week. The two brands cold cereal and frozen dinner have a maximum of 23 zero counts in a week. We were not able to obtain the results for these two categories.

3 Discussion

This thesis has not calculated the standard errors of the estimates obtained using the EM algorithm. We can use numerical methods to obtain them. We will be able to calculate or approximate the Hessian matrix by taking the 2^{nd} derivative of the expectation or log-likelihood function in the E-step.

For direct optimization of the observed data log-likelihood derived in section 2.5, we use the optim function in R. We compare the EM algorithm showed in section 2.4 and the direct optimization of observed data log-likelihood to obtain maximum likelihood estimates when we introduce product availability as a latent variable. Unlike in many cases, directly calculating the MLE provide faster results than the EM algorithm. For example, the category of trash bags took 2.2 seconds using the EM algorithm. In comparison, the direct optimizing method took 0.34 seconds. The category carbonated beverages used 5.3 hours using the EM algorithm while the other form took only 13.21 minutes. We provided the time taken for each category in the tables in Appendix B. Both methods provided almost the same solution. Further, the EM algorithm converged within a few iterations, which is not the usual case.

Appendix A
R Code

Functions involved with the EM algorithm is as follows:

```
#Here z is the all the possible z vectors for a period
#k = number of products with 0 units
# beta=beta0
# betaw=beta0
# t=1
# periods=numperiods

pwt<-function(beta,X,choicevec,numproducts,periods,z,count){
  U=X%*%beta
  #cat("Within pwt, count =", count, ".\n")
  SetofU<-U[count:(count+numproducts-1)] #(numproducts-1+3)*1
  Setchoice<-choicevec[count:(count+numproducts-1)]
  denom <-z%*%exp(SetofU) #(2^k*numproducts)(numproducts*1)->2^k*1
  logP=rep(NA,nrow(z))
  for(i in 1:nrow(z)){
    logP[i]=sum(Setchoice*(SetofU-log(denom[i])))
  }
  logP <- logP -max(logP)
  P <- exp(logP)
  P <- P/sum(P)
  return(P)
}

Estep<-function(beta,betaw,X,choicevec,numproducts,periods,z_list){
```

```
  Exp=0
  count=1
  U=X%*%beta
  for(t in 1:periods){
    z <- z_list[[t]]
    SetofU=U[count:(count+numproducts-1)]
    Setchoice=choicevec[count:(count+numproducts-1)]
    Exp1 = rep(NA,nrow(z))
    # cat("count should be ", count, "...  ")
    P= pwt(betaw,X,choicevec,numproducts,periods,z,count)
    count = count + numproducts
    Exp2 = z%*%exp(SetofU)
    Exp1= log(Exp2)
    Exp=Exp+sum(Setchoice*SetofU)-sum(Setchoice)*sum(P*Exp1)
    Exp
    t=t+1
  }
  return(Exp)
}

#observed log-likelihood
ObservedL<-function(beta,X,choicevec,numproducts,periods,z_list){
  logL=0
  # L=1
  count=1
  U=X%*%beta
  for(t in 1:periods){
    z <- z_list[[t]] #z: 2^k*numproducts
    SetofU<-U[count:(count+numproducts-1)] #13*1
    Setchoice<-choicevec[count:(count+numproducts-1)]
    count <- count + numproducts
    term1 <- sum(Setchoice*SetofU)
    minuslogterm2 <- -sum(Setchoice)*log(z%*%exp(SetofU)) #2^k*1
    shift <- max(minuslogterm2)
    altTerm2 <- log(sum(exp(minuslogterm2 - shift)))  + shift
```

```
    #adjusted to avoid log0
    logL=logL+term1+altTerm2
  }
  return(logL)
}


Binaryize <- function(x, ndigits = ceiling(log(x)/log(2))) {
  ans <- rep(0, ndigits)
  for (i in 1:ndigits) {
    ans[i] <- x %% 2
    x <- x %/% 2
  }
  rev(ans)
}
```

Functions used to calculate parameter estimates for scenario 1 are as follows:

```
 LogLikeMNL1<-function(beta,X,choicevec,numproducts,periods){
  #X=X;choicevec=choicevec;numproducts=numproducts;periods=fitperiods

  #  Build U=X*beta, which is the vector of utilities (U)
  #  for every product in every period.
  #  Periods are numbered 1...periods.
  #  Numset is the array containing the number of choices in
  #  different periods
  #  We break the utility vector into sets for each period.
  #  Note that the set may change size because product offerings
  #  may change each period.
  #  SetofU is the array of utilities for each period in the dataset.
  #  SetofU is redefined for each pass through the loop;
  #  this is because
  #  each pass through the loop is a separate period,
  #  and the set of utilities may change.
  #  This is the big outer loop (indexed using "i") where we build
  #  the loglikelihood
```

```
  #  function ONE PERIOD AT A TIME!

  U=X%*%beta# X*(beta)^T
  tot=0.0
  count=1
  for(i in 1:periods){
    SetofU=matrix(NA,nrow=numproducts,ncol=1)
    Setchoice=matrix(NA,nrow=numproducts,ncol=1)
    SetofU[1:numproducts]=U[count:(count+numproducts-1)]
    Setchoice[1:numproducts]=choicevec[count:(count+numproducts-1)]
    count=count+numproducts

    #
    #  Build the choice probabilities for MNL;
    #create denominator first,
    #then the individual probabilities.
    #
    denom=0.0
    denom=denom+sum(exp(SetofU))

    Q=NULL
    Q= exp(SetofU)/denom

    #
    #  Now we can build the loglikelihood function
    #using the choice probabilities and choice frequencies.
    #
    tot=tot+sum(Setchoice*log(Q))

  }
  y=tot
}
```

Functions used to calculate parameter estimates for scenario 2 are as follows:

```
 LogLikeMNL2<-function(beta,X,choicevec,numset,periods){
```

```
#   Build U=X*beta, which is the vector of utilities (U) for
#    every product in every period.
#   Periods are numbered 1...periods.
#   Numset is the array containing the number of choices in
#  different periods
#   We break the utility vector into sets for each period.
#   Note that the set may change size because product
#  offerings may change each period.

#   SetofU is the array of utilities for each period in the dataset.
#   SetofU is redefined for each pass through the loop; this is because
#   each pass through the loop is a separate period,
# and the set of utilities may change.
#   This is the big outer loop (indexed using "i") where we build the
#   loglikelihood function ONE PERIOD AT A TIME!

tot=0.0
count=1
for(i in 1:periods){
  productscolumn2 = productscolumn[count:(count+numset[i]-1)]
  beta2=as.matrix(beta[c(productscolumn2-1,(n-6):(n-4))])
  #remove columns corresponding to 0 units
  X2=X[count:(count+numset[i]-1),c(productscolumn2-1,(n-6):(n-4))]
  U2=X2%*%beta2
  SetofU =U2
  Setchoice=matrix(NA,nrow=numset[i],ncol=1)
  Setchoice[1:numset[i]]=choicevec[count:(count+numset[i]-1)]
  count=count+numset[i]


  #Build the choice probabilities for MNL;
  #create denominator first,
  #then the individual probabilities.
  denom=0.0
```

```
    denom=denom+sum(exp(SetofU))

    Q=NULL
    Q= exp(SetofU)/denom

    #
    #   Now we can build the loglikelihood function
    #   using the choice probabilities and choice frequencies.
    #
    tot=tot+sum(Setchoice*log(Q))

  }
  y=tot
}
```

Code used to run these algorithms for all three cases is as follows:

```
 library(tidyverse)
setwd("/Users/ts/Desktop/Penn State_3_3/Research/Thesis/Coding/Masters/")
source("EM_Functions_Final.R")
source("LogLikelihood_Scenario1.R")
source("LogLikelihood_Scenario2.R")
category = c('BAKING_MIXES','CANDY','CARBONATED_BEVERAGES','CHOCOLATE',
            'COLD_CEREAL','COOKIES','CRACKERS','DISH_DETERGENT',
            'FRESH_BREAD','FRESH_EGGS','FRUIT','FZ_DINNER',
             'HOUSEHOLD_CLEANER','ICE_CREAM','LAUNDRY_DETERGENT',
             'LUNCHEON_MEATS','MARGARINE','MILK','NATURAL_CHEESE',
             'OIL','PAPER_TOWELS','PROCESSED_CHEESE','SALAD',
             'SALAD_DRESSINGS','SALTY_SNACKS','SOUP','SPICES',
             'TOILET_TISSUE','TRASH_BAGS','VEGETABLES')

numcategories = length(category)
datafile = paste0("Orgdata/",category,'.csv')
outputfile = paste0("NewScenario_withnumperiods_V2/",category,'_out.csv')
#fitperiods = 80 #to allocate weeks 81 to 104 for out of sample estimation
numstarts = 10
```

```
for (cat in 1:numcategories){
#cat=13
#for (cat in 7:11){
  print(cat)
  start=Sys.time()
  A = read_csv(datafile[cat],col_names = F)
  A  = as.matrix(A)
  m= nrow(A)
  n= ncol(A)
  productscolumn = A[,3] #stores the brand number
  choicevec = A[,4] #choicevec stores the number of units
  #Indicator of the product, and price, display, featured
  X  =  A[,5:n]
  productlist = unique(productscolumn)
  numproducts = length(productlist)

  # Determine the number of periods (=numperiods)
  numperiods = length(unique(A[,2]))


  #determine number of products with zero purchase in each period
  missing =rep(0,numperiods)
  missing_count=rep(0,numperiods)
  missing_list <-as.list(rep(0,numperiods))
  count=1
  for(i in 1:numperiods){
    choice=matrix(NA,nrow=numproducts,ncol=1)
    choice[1:numproducts]=choicevec[count:(count+numproducts-1)]
    df = as.data.frame(choice[1:numproducts])
    colnames(df)<-"units"
    #obtain number of missing items per period
    missing[i]=rowSums(t(choice[1:numproducts])==0)
    missing_count[i]=length(missing[i])
    #identify the missing items per period and store in a  list
    missing_list[[i]] =c(which(df$units==0))
```

```
  count=count+numproducts
}
max_missing=max(missing_count)

#create the matrix for latent variable
z_list = as.list(rep(NA,numperiods))
for(i in 1:numperiods){
  missing= missing_list[[i]]
  z=matrix(1,ncol=numproducts,nrow=2^length(missing))
  z[, missing] <- t(sapply(0:(2^length(missing)-1),
  Binaryize, length(missing)))
  z_list[[i]] <-z
}

####optimization_EM#######
count=1
numstarts=10
bestLL = -Inf
options.MaxFunEvals = n*100000
options.TolX = 1.00e-100
options.MaxIter = n*10000

OL = 0
c=0
OLhat =c(0,0)
set.seed(1234)
beta0 = as.matrix(runif(n=n-4,min=-2,max=2))
OL[2]=ObservedL(beta=beta0,X=X,choicevec=choicevec,
      numproducts=numproducts,periods=numperiods,z=z_list)

#maximize E-step
result=optim(par=beta0,Estep,betaw=beta0,X=X,choicevec=choicevec,
      numproducts=numproducts,periods=numperiods,z=z_list,
      control = list(maxit=options.MaxIter,reltol=options.TolX,fnscale=-1),
      method="BFGS")
```

```
bestbeta=result$par
fval=result$value
print(paste0("fval=",fval))
#bestbeta = beta

k=2
repeat{
  #maximize E-step
  result=optim(par=bestbeta,Estep,betaw=bestbeta,X=X,choicevec=choicevec,
         numproducts=numproducts,periods=numperiods,z=z_list,
         control = list(maxit=options.MaxIter,reltol=options.TolX,
         fnscale=-1),method="BFGS")
  fval=result$value
  print(paste0("fval=",fval))
  bestbeta=result$par
  bestLL = fval
  k=k+1
  OL[k]=ObservedL(beta=bestbeta,X=X,choicevec=choicevec,
        numproducts=numproducts,periods=numperiods,z=z_list)
  c[k-1]= (OL[k]-OL[k-1])/(OL[k-1]-OL[k-2])
  #AIC estimate
  OLhat[k-1] = OL[k-2]+1/(1-c[k-1])*(OL[k-1]-OL[k-2])
  if(abs(OLhat[k-1]-OL[k-1])<0.005){
    break
  }
}
print(OL)
LogLikelihood_EM = round(bestLL,3)
IdealUtilities_EM = round(bestbeta,3)
output1 = c(numproducts,numperiods,LogLikelihood_EM,IdealUtilities_EM)

##optimization Scenario1###
bestLL = -Inf
for(iter in 1:numstarts){
  set.seed(iter)
```

```
  beta0 = as.matrix(runif(n=n-4,min=-2,max=2))
  # %X has columns 5:15 of A,  choicevec=number of units,
  #numproducts=number
  # %of choices in each period
  #objfun = LogLikeMNL(beta,X,choicevec,numproducts,periods)
  result=optim(par=beta0,LogLikeMNL1,X=X,choicevec=choicevec,
          numproducts=numproducts,periods=numperiods,
          control = list(maxit=options.MaxIter,reltol=options.TolX,
          fnscale=-1),method="BFGS")
  fval=result$value
  beta=result$par
  if (fval > bestLL){
    bestLL = fval
    bestbeta = beta
  }
}
LogLikelihood_S1 = round(bestLL,3)
IdealUtilities_S1 = round(bestbeta,3)
output2 = c(numproducts,numperiods,LogLikelihood_S1,IdealUtilities_S1)

A = A[A[,4]!=0,]
m= nrow(A)
n= ncol(A)
productscolumn = A[,3] #stores the brand number
choicevec = A[,4] #choicevec stores the number of units
X  =  A[,5:n]
productlist = unique(productscolumn)
numproducts = length(productlist)
numperiods = length(unique(A[,2]))
numset = as.numeric(table(A[,2]))

bestLL = -Inf
for(iter in 1:numstarts){
  set.seed(iter)
  beta0 = as.matrix(runif(n=n-4,min=-2,max=2))
```

```
    # %X has columns 5:15 of A,  choicevec=number of units,
    #numproducts=number
    # %of choices in each period
    #objfun = LogLikeMNL(beta,X,choicevec,numproducts,periods)
    result=optim(par=beta0,LogLikeMNL2,X=X,choicevec=choicevec,
            numset=numset,periods=numperiods,
            control = list(maxit=options.MaxIter,reltol=options.TolX,
            fnscale=-1),method="BFGS")
    fval=result$value
    beta=result$par
    if (fval > bestLL){
      bestLL = fval
      bestbeta = beta
    }
  }
  LogLikelihood_S2 = round(bestLL,3)
  IdealUtilities_S2 = round(bestbeta,3)
  #output3 = c(1IdealUtilities_S2)

  output =cbind(IdealUtilities_EM,IdealUtilities_S1,IdealUtilities_S2)
  alpha  = output[1:(numproducts-1),]
  alpha = cbind(2:(numproducts),alpha)
  alpha = alpha[order(alpha[,2],decreasing = T),]
  beta= output[numproducts:(numproducts+2),]
  beta = cbind(c("PRICE","FEATURE","DISPLAY"),beta)
  LL =cbind("Log-Likelihood",LogLikelihood_EM,LogLikelihood_S1,
        LogLikelihood_S2)
  output =as.data.frame(rbind(alpha,beta,LL),quote=FALSE)
  colnames(output) <-c("Variables","New","Scenario1","Scenario2")
  write_csv(output,outputfile[cat])
  time=Sys.time()-start
  print(time)
}
```

The code used for the simulation is as follows:

```
library(tidyverse)
```

```
library(DescTools)
setwd("/Users/ts/Desktop/Penn State_3_3/Research/Thesis/Coding/Masters/")
#source("EM_Functions.R")
source("EM_Functions_Final.R")
source("LogLikelihood_Scenario1.R")
source("LogLikelihood_Scenario2.R")
#simulate the dataset
numproducts=13
numperiods=15
set.seed(123)
totchoices =rpois(numperiods,60)
set.seed(123)
price <-   abs(rnorm(numproducts*numperiods,mean=1.5,sd=2))
set.seed(123)
feature <- as.integer(rbernoulli(numproducts*numperiods,p=0.1))
set.seed(123)
display <- as.integer(rbernoulli(numproducts*numperiods,p=0.05))

#allocate NA for some rows (alocate the products which are not available)
# the number of random values to replace
N <-10

#do not replace the 1st product (this is the outside option) every week
a<-c(1:(numproducts*numperiods))
remove<- c(a[seq(1,length(a),numproducts)])
z<-a [!a %in% remove]
#replace the following indexes
set.seed(123)
inds <-  sample(a,size=N,replace=F)


filenum=rep(1,numproducts*numperiods)
week <-rep(1:numperiods,each=numproducts)
brand <- rep(c(1:numproducts),numperiods)
dat=as.matrix(cbind(price,feature,display))
```

```
mat =rbind(rep(0,numproducts-1),diag(1,nrow=numproducts-1,
    ncol=numproducts-1))
mat= do.call(rbind, replicate(numperiods, mat, simplify=FALSE))
X=as.matrix(cbind(mat,price,feature,display))

numstarts=10
count=1
numstarts=10
bestLL = -Inf
zero <-0
samples =100

IdealUtilities_EM = matrix(NA,nrow=numproducts+2 ,ncol=samples)
IdealUtilities_U = matrix(NA,nrow=numproducts+2,ncol=samples)
IdealUtilities_S1= matrix(NA,nrow=numproducts+2 ,ncol=samples)
IdealUtilities_S2= matrix(NA,nrow=numproducts+2 ,ncol=samples)
betaA<-c(-2.546,-3.671,-3.695,-3.751,-3.824,-3.884,-3.973,-4.019,
        -4.059,-4.387,-4.738,-4.882,-0.229,0.14,0.904)

U=X%*%betaA
U[inds] <- NA
expU=rep(NA,nrow(U))
for(i in 1:nrow(U)){
  if(!is.na(U[i])){
    expU[i]=exp(U[i])
  }else{
    expU[i]=0
  }
}
for(steps in 1:samples){
  pkt_list=as.list(rep(NA,numperiods))
  units =as.list(rep(NA,numperiods))
  s= rep(NA,numperiods)
  count=1
  for(i in 1:numperiods){
```

```
  set.seed(123+i+steps)
  SetofexpU=matrix(NA,nrow=numproducts,ncol=1)
  SetofexpU[1:numproducts]=expU[count:(count+numproducts-1)]
  denomsum=sum(SetofexpU)
  pkt_list[[i]]=SetofexpU/denomsum
  #pkt_list[[i]][is.na(pkt_list[[i]])]<-0
  s[i] = sum(pkt_list[[i]])
  units[[i]] = rmultinom(1,totchoices[i],as.vector(pkt_list[[i]]))
  count=count+numproducts
}
units=unlist(units)


A =as.matrix(cbind(filenum,week,brand,units,mat,price,feature,display))
# A <- as.tibble(A)
# A %>% group_by(brand) %>% summarise(max(units))
m= nrow(A)
n= ncol(A)
productscolumn = A[,3] #stores the brand number
choicevec = A[,4] #choicevec stores the number of units
#Indicator of the product, and price, display, featured
X  =  A[,5:n]
productlist = unique(productscolumn)
numperiods = length(unique(A[,2]))

#determine number of products with zero purchase in each period
missing =rep(0,numperiods)
missing_list <-as.list(rep(0,numperiods))
count=1
for(i in 1:numperiods){
  choice=matrix(NA,nrow=numproducts,ncol=1)
  choice[1:numproducts]=choicevec[count:(count+numproducts-1)]
  df = as.data.frame(choice[1:numproducts])
  colnames(df)<-"units"
  #obtain number of missing items per period
```

```
  missing[i]=rowSums(t(choice[1:numproducts])==0)
  #identify the missing items per period and store in a  list
  missing_list[[i]] =c(which(df$units==0))
  count=count+numproducts
}

#create the matrix for latent variable
z_list = as.list(rep(NA,numperiods))
for(i in 1:numperiods){
  missing= missing_list[[i]]
  z=matrix(1,ncol=numproducts,nrow=2^length(missing))
  z[, missing] <- t(sapply(0:(2^length(missing)-1),
                  Binaryize, length(missing)))
  z_list[[i]] <-z
}

options.MaxFunEvals = n*100000
options.TolX = 1.00e-100
options.MaxIter = n*10000

####optimization_EM#######
OL = 0
c=0
OLhat =c(0,0)
set.seed(123+steps)
beta0 = as.matrix(runif(n=n-4,min=-2,max=2))
OL[2]=ObservedL(beta=beta0,X=X,choicevec=choicevec,
      numproducts=numproducts,periods=numperiods,z=z_list)
Start <- Sys.time()
#maximize E-step
result=optim(par=beta0,Estep,betaw=beta0,X=X,choicevec=choicevec,
      numproducts=numproducts,periods=numperiods,z=z_list,
      control = list(maxit=options.MaxIter,reltol=options.TolX,
      fnscale=-1),method="BFGS")
bestbeta=result$par
```

```
fval=result$value

k=2
repeat{
  #maximize E-step
  result=optim(par=bestbeta,Estep,betaw=bestbeta,X=X,choicevec=choicevec,
          numproducts=numproducts,periods=numperiods,z=z_list,
          control = list(maxit=options.MaxIter,reltol=options.TolX,
          fnscale=-1),method="BFGS")
  fval=result$value
  bestLL = fval
  bestbeta=result$par
  k=k+1
  OL[k]=ObservedL(beta=bestbeta,X=X,choicevec=choicevec,
          numproducts=numproducts,periods=numperiods,z=z_list)
  c[k-1]= (OL[k]-OL[k-1])/(OL[k-1]-OL[k-2])
  #AIC estimate
  OLhat[k-1] = OL[k-2]+1/(1-c[k-1])*(OL[k-1]-OL[k-2])
  if(abs(OLhat[k-1]-OL[k-1])<0.005){
    break
  }
}
LogLikelihood_EM = round(bestLL,digits=3)
IdealUtilities_EM[,steps] = bestbeta

OLhat =c(0,0)
OL2 =0
OL2[2]=ObservedL(beta=beta0,X=X,choicevec=choicevec,
      numproducts=numproducts,periods=numperiods,z=z_list)
#Start2 <- Sys.time()
#maximize E-step
result=optim(par=beta0,ObservedL,X=X,choicevec=choicevec,
          numproducts=numproducts,periods=numperiods,z=z_list,
             control = list(maxit=options.MaxIter,reltol=options.TolX,
             fnscale=-1),method="BFGS")
```

```
fval=result$value
#print(fval)
bestbeta=result$par

j=2
repeat{
  #maximize E-step
  result=optim(par=bestbeta,ObservedL,X=X,choicevec=choicevec,
          numproducts=numproducts,periods=numperiods,z=z_list,
               control = list(maxit=options.MaxIter,reltol=options.TolX,
               fnscale=-1),method="BFGS")
  fval=result$value
  #print(paste0("fval=",fval))
  bestLL = fval
  bestbeta=result$par
  j=j+1
  #OL[j]=fval
  OL2[j]=ObservedL(beta=bestbeta,X=X,choicevec=choicevec,
          numproducts=numproducts,periods=numperiods,z=z_list)
  c[j-1]= (OL2[j]-OL2[j-1])/(OL2[j-1]-OL2[j-2])
  #AIC estimate
  OLhat[j-1] = OL2[j-2]+1/(1-c[j-1])*(OL2[j-1]-OL2[j-2])
  if(abs(OLhat[j-1]-OL2[j-1])<0.005){
    break
  }
}
LogLikelihood_U = round(bestLL,digits=3)
#IdealUtilities_U = round(bestbeta,3)
IdealUtilities_U[,steps] = bestbeta

##optimization Scenario1###
bestLL = -Inf
for(iter in 1:numstarts){
  set.seed(iter)
  beta0 = as.matrix(runif(n=n-4,min=-2,max=2))
```

```
  result=optim(par=beta0,LogLikeMNL1,X=X,choicevec=choicevec,
          numproducts=numproducts,periods=numperiods,
               control = list(maxit=options.MaxIter,reltol=options.TolX,
               fnscale=-1),method="BFGS")
  fval=result$value
  beta=result$par
  if (fval > bestLL){
    bestLL = fval
    bestbeta = beta
  }
}
LogLikelihood_S1 = bestLL
IdealUtilities_S1[,steps] = bestbeta
#EstU_S1[,steps]=X%*%IdealUtilities_S1

#Scenario2
A = A[A[,4]!=0,]
m= nrow(A)
n= ncol(A)
productscolumn = A[,3] #stores the brand number
choicevec = A[,4] #choicevec stores the number of units
X2  =  A[,5:n]
productlist = unique(productscolumn)
numproducts = length(productlist)
numperiods = length(unique(A[,2]))
numset = as.numeric(table(A[,2]))

bestLL = -Inf
for(iter in 1:numstarts){
  set.seed(iter)
  beta0 = as.matrix(runif(n=n-4,min=-2,max=2))
  result=optim(par=beta0,LogLikeMNL2,X=X2,choicevec=choicevec,
      numset=numset,periods=numperiods,
               control = list(maxit=options.MaxIter,reltol=options.TolX,
               fnscale=-1),method="BFGS")
```

```
    fval=result$value
    beta=result$par
    if (fval > bestLL){
      bestLL = fval
      bestbeta = beta
    }
  }
  LogLikelihood_S2 = bestLL
  IdealUtilities_S2[,steps] = bestbeta

  #cat(paste0("missing=",length(units[units==0])))
  zero <- zero+length(units[units==0])
}

IdealUtilities_EM <- as.vector(IdealUtilities_EM)
IdealUtilities_U <- as.vector(IdealUtilities_U)
IdealUtilities_S1 <- as.vector(IdealUtilities_S1)
IdealUtilities_S2 <- as.vector(IdealUtilities_S2)

RMSE(IdealUtilities_EM,ref=betaA)
RMSE(IdealUtilities_U,ref=betaA)
RMSE(IdealUtilities_S1,ref=betaA)
RMSE(IdealUtilities_S2,ref=betaA)
zero/samples
```

Appendix B

Tables of Results

Table B.1: Maximum likelihood estimates of α_k parameters in equations (2.5) of Baking Mixes - *Directly calculating the MLE required 0.572 secs and the EM algorithm required 20.099 sec (35 times as long).*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
jello	-0.825 (2)	-1.244 (2)	-0.606 (2)
betty crocker	-1.401 (3)	-1.393 (4)	-1.403 (3)
martha white	-1.406 (4)	-1.375 (3)	-1.411 (4)
duncan hines	-2.026 (5)	-2.129 (5)	-2.021 (5)
pillsbury	-2.3 (6)	-2.695 (7)	-2.258 (6)
betty crocker superm	-2.453 (7)	-2.626 (6)	-2.446 (7)
bisquick	-2.692 (8)	-2.78 (8)	-2.688 (8)
private label	-2.806 (9)	-2.967 (9)	-2.794 (9)
pillsbury moist supr	-3.339 (10)	-3.947 (10)	-3.304 (10)
martha white cotton	-3.684 (11)	-4.004 (11)	-3.425 (11)
house autry	-3.831 (12)	-4.586 (13)	-3.59 (12)
jiffy	-4.142 (13)	-4.296 (12)	-4.13 (13)

Table B.2: Maximum likelihood estimates of α_k parameters in equations (2.5) of Carbonated beverages - *Directly calculating the MLE required 13.212 mins and the EM algorithm required 5.375 hours (24 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
pepsi	-2.404 (2)	-2.427 (2)	-2.404 (2)
coke classic	-2.441 (3)	-2.463 (3)	-2.439 (3)
sun drop	-2.554 (4)	-2.562 (4)	-2.554 (4)
private label	-2.796 (5)	-2.789 (5)	-2.786 (5)
mountain dew	-3.18 (6)	-3.198 (6)	-3.177 (6)
caffeine free diet c	-3.416 (7)	-3.447 (8)	-3.415 (7)
diet pepsi	-3.423 (8)	-3.439 (7)	-3.421 (8)
diet coke	-3.46 (9)	-3.474 (9)	-3.458 (9)
dr pepper	-3.501 (10)	-3.507 (10)	-3.501 (10)
cheerwine	-3.572 (11)	-3.574 (11)	-3.57 (11)
diet mountain dew	-3.607 (12)	-3.615 (12)	-3.606 (12)
sprite	-3.669 (13)	-3.688 (13)	-3.669 (13)
caffeine free diet p	-3.71 (14)	-3.757 (14)	-3.707 (14)
sierra mist	-3.768 (15)	-3.987 (17)	-3.761 (15)
caffeine free pepsi	-3.81 (16)	-3.839 (15)	-3.806 (16)
caffeine free coke c	-3.818 (17)	-3.882 (16)	-3.815 (17)
diet dr pepper	-4.097 (18)	-4.14 (18)	-4.093 (18)
wild cherry pepsi	-4.224 (19)	-4.234 (19)	-4.222 (20)
cherry coke	-4.281 (20)	-4.569 (22)	-4.149 (19)
diet sierra mist	-4.286 (21)	-4.466 (21)	-4.264 (21)
mello yello	-4.326 (22)	-4.433 (20)	-4.308 (22)
diet sprite	-4.448 (23)	-4.579 (23)	-4.416 (23)
diet cherry coke	-4.536 (24)	-4.751 (25)	-4.486 (24)
7 up	-4.609 (25)	-4.695 (24)	-4.589 (27)
diet rite	-4.629 (26)	-4.807 (27)	-4.587 (26)
sunkist	-4.669 (27)	-4.782 (26)	-4.638 (28)
pepsi one	-4.726 (28)	-4.94 (30)	-4.669 (29)
mountain dew code re	-4.74 (29)	-4.893 (29)	-4.699 (31)
canada dry	-4.744 (30)	-4.853 (28)	-4.711 (32)
diet mountain dew ca	-4.853 (31)	-5.08 (33)	-4.775 (34)
schweppes	-4.856 (32)	-5.095 (34)	-4.735 (33)
vanilla coke	-4.872 (33)	-5.015 (31)	-4.819 (35)
fresca	-4.875 (34)	-5.374 (36)	-4.546 (25)
mountain dew caffein	-4.994 (35)	-5.048 (32)	-4.973 (37)
ibc	-5.017 (36)	-5.364 (35)	-4.685 (30)
fanta	-5.167 (37)	-5.576 (41)	-4.866 (36)
a & w	-5.193 (38)	-5.397 (37)	-5.094 (42)
cherry 7 up	-5.202 (39)	-5.546 (39)	-5.001 (38)
r c	-5.207 (40)	-5.561 (40)	-5.019 (40)
diet mountain dew co	-5.268 (41)	-5.545 (38)	-5.132 (43)
diet 7 up	-5.348 (42)	-5.657 (42)	-5.165 (44)
dr pepper red fusion	-5.355 (43)	-5.906 (44)	-5.004 (39)
crush	-5.427 (44)	-5.867 (43)	-5.058 (41)
seagrams	-5.481 (45)	-5.908 (45)	-5.166 (45)
caffeine free diet d	-5.718 (46)	-6.144 (46)	-5.343 (47)
pepsi vanilla	-5.76 (47)	-6.273 (47)	-5.328 (46)

Table B.3: Maximum likelihood estimates of α_k parameters in equations (2.5) of Chocolate - *Directly calculating the MLE required 9.010 secs and the EM algorithm required 4.9886 secs (33 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
snickers	-3.388 (2)	-3.429 (2)	-3.383 (2)
hersheys	-3.464 (3)	-3.509 (3)	-3.454 (3)
m & ms	-3.497 (4)	-3.594 (4)	-3.483 (4)
hersheys kisses	-3.566 (5)	-3.698 (5)	-3.537 (5)
reeses	-3.694 (6)	-3.723 (6)	-3.685 (6)
nestle butterfinger	-3.969 (7)	-4.208 (7)	-3.913 (7)
hersheys nuggets	-4.171 (8)	-4.74 (10)	-3.969 (8)
nestle baby ruth	-4.364 (9)	-4.641 (9)	-4.255 (9)
nestle crunch	-4.42 (10)	-4.572 (8)	-4.342 (10)
york peppermint patt	-4.59 (11)	-4.92 (12)	-4.418 (11)
kit kat	-4.668 (12)	-4.806 (11)	-4.601 (13)
reeses fast break	-4.765 (13)	-5.049 (14)	-4.596 (12)
three musketeers	-4.807 (14)	-5.041 (13)	-4.681 (15)
peter paul almond jo	-4.846 (15)	-5.165 (15)	-4.643 (14)
milky way	-4.891 (16)	-5.196 (16)	-4.707 (16)
hersheys mr goodbar	-5.22 (17)	-5.675 (17)	-4.814 (17)
peter paul mounds	-5.688 (18)	-6.055 (18)	-5.308 (18)
reeses sticks	-5.937 (19)	-6.358 (19)	-5.443 (19)
hersheys cookies n c	-6.097 (20)	-6.491 (20)	-5.6 (20)

Table B.4: Maximum likelihood estimates of α_k parameters in equations (2.5) of Cookies - *Directly calculating the MLE required 9.010 secs and the EM algorithm required 4.9886 secs (33 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
private label	-2.74 (2)	-2.849 (2)	-2.684 (2)
oreo	-2.851 (3)	-3.043 (4)	-2.827 (3)
nabisco nilla	-2.906 (4)	-2.869 (3)	-2.964 (5)
little debbie	-2.938 (5)	-3.062 (5)	-2.879 (4)
keebler chips deluxe	-3.037 (6)	-3.253 (7)	-3.002 (6)
murray	-3.138 (7)	-3.182 (6)	-3.118 (7)
nabisco oreo double	-3.345 (8)	-3.67 (11)	-3.289 (8)
archway	-3.355 (9)	-3.518 (9)	-3.358 (10)
chips ahoy	-3.392 (10)	-3.696 (12)	-3.349 (9)
nabisco snackwells	-3.431 (11)	-3.464 (8)	-3.428 (11)
oreo double delight	-3.568 (12)	-4.062 (17)	-3.465 (12)
murray sugar free	-3.596 (13)	-3.66 (10)	-3.552 (14)
keebler fudge shoppe	-3.619 (14)	-3.859 (13)	-3.553 (15)
nabisco nutter butte	-3.671 (15)	-3.925 (15)	-3.616 (16)
fig newton	-3.698 (16)	-3.926 (16)	-3.65 (17)
pepperidge farm mila	-3.785 (17)	-3.888 (14)	-3.51 (13)
nabisco teddy graham	-3.873 (18)	-4.069 (18)	-3.783 (19)
little debbie nutty	-3.882 (19)	-4.22 (19)	-3.803 (20)
lance nekot	-3.898 (20)	-4.237 (20)	-3.812 (21)
chewy chips ahoy	-3.965 (21)	-4.472 (23)	-3.777 (18)
keebler rainbow chip	-3.968 (22)	-4.441 (22)	-3.815 (22)
moon pie	-4.051 (23)	-4.478 (24)	-3.93 (26)
keebler pecan sandie	-4.086 (24)	-4.531 (26)	-3.905 (24)
keebler chocolat lov	-4.115 (25)	-4.599 (27)	-3.91 (25)
keebler e l fudge	-4.181 (26)	-4.744 (28)	-3.874 (23)
jacks	-4.291 (27)	-4.526 (25)	-4.21 (30)
famous amos	-4.306 (28)	-4.822 (29)	-4.004 (27)
barnums animals	-4.34 (29)	-4.331 (21)	-4.194 (29)
lance	-4.373 (30)	-4.933 (31)	-4.041 (28)
little debbie fudge	-4.463 (31)	-4.879 (30)	-4.312 (31)

Table B.5: Maximum likelihood estimates of α_k parameters in equations (2.5) of Crackers - *Directly calculating the MLE required 1.859 secs and the EM algorithm required 1.885 mins (61 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
kraft handi snacks	-1.916 (2)	-1.885 (2)	-1.84 (2)
nabisco triscuit	-2.18 (3)	-2.11 (3)	-2.211 (4)
sunshine cheez it	-2.183 (4)	-2.167 (4)	-2.197 (3)
nabisco wheat thins	-2.343 (5)	-2.291 (5)	-2.375 (6)
nabisco ritz bits	-2.344 (6)	-2.35 (6)	-2.306 (5)
pepperidge farm gold	-2.391 (7)	-2.358 (7)	-2.42 (7)
lance	-2.514 (8)	-2.515 (8)	-2.529 (8)
nabisco ritz	-2.6 (9)	-2.657 (9)	-2.594 (9)
nabisco honey maid	-2.875 (10)	-2.868 (10)	-2.889 (11)
lance toast chee	-3.008 (11)	-3.054 (11)	-3.002 (13)
private label	-3.046 (12)	-3.192 (12)	-2.996 (12)
nabisco cheese nips	-3.081 (13)	-3.22 (13)	-3.064 (15)
keebler townhouse	-3.113 (14)	-3.22 (13)	-3.098 (17)
keebler zesta	-3.125 (15)	-3.254 (15)	-3.087 (16)
keebler wheatables	-3.143 (16)	-3.521 (16)	-2.826 (10)
nabisco honey maid s	-3.305 (17)	-3.753 (20)	-3.037 (14)
keebler club	-3.498 (18)	-3.731 (18)	-3.436 (19)
nabisco premium	-3.507 (19)	-3.628 (17)	-3.472 (20)
keebler grahams	-3.51 (20)	-3.859 (21)	-3.345 (18)
lance captains wafer	-3.643 (21)	-3.75 (19)	-3.613 (21)
lance nip chee	-3.867 (22)	-4.166 (22)	-3.734 (22)

Table B.6: Maximum likelihood estimates of α_k parameters in equations (2.5) of Dish detergent - *Directly calculating the MLE required 0.518 secs and the EM algorithm required 9.565 secs (18 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
electrasol	-2.276 (2)	-2.506 (4)	-2.275 (2)
cascade	-2.378 (3)	-2.456 (3)	-2.372 (3)
private label	-2.447 (4)	-2.416 (2)	-2.442 (4)
dawn	-2.464 (5)	-2.55 (5)	-2.463 (5)
cascade complete	-2.958 (6)	-3.778 (9)	-2.885 (6)
joy	-2.989 (7)	-3.103 (6)	-2.978 (7)
dawn fresh escapes	-3.102 (8)	-3.562 (8)	-3.028 (8)
palmolive	-3.117 (9)	-3.548 (7)	-3.077 (9)
ajax	-3.481 (10)	-3.81 (10)	-3.428 (10)

Table B.7: Maximum likelihood estimates of α_k parameters in equations (2.5) of Fresh bread - *Directly calculating the MLE required 2.934 secs and the EM algorithm required 1.756 mins (36 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
private label	-1.532 (2)	-1.583 (2)	-1.52 (2)
merita	-1.704 (3)	-1.695 (3)	-1.707 (3)
natures own	-2.168 (4)	-2.159 (4)	-2.17 (4)
kings hawaiian	-2.343 (5)	-2.552 (6)	-2.283 (5)
sunbeam	-2.509 (6)	-2.479 (5)	-2.516 (6)
bunny	-2.624 (7)	-2.634 (7)	-2.621 (7)
natures own whitewhe	-2.83 (8)	-2.832 (8)	-2.83 (8)
sara lee	-2.876 (9)	-2.928 (9)	-2.88 (9)
thomas	-2.969 (10)	-3.171 (12)	-2.966 (10)
merita country	-3 (11)	-3.011 (10)	-3 (11)
cobblestone mill	-3.043 (12)	-3.028 (11)	-3.047 (12)
arnold	-3.419 (13)	-3.512 (13)	-3.421 (13)
iron kids	-3.543 (14)	-3.879 (15)	-3.499 (14)
pepperidge farm	-3.581 (15)	-3.746 (14)	-3.55 (15)
d italiano	-4.007 (16)	-4.221 (16)	-3.944 (17)
pepperidge farm farm	-4.112 (17)	-4.79 (20)	-3.949 (18)
toufayan	-4.207 (18)	-4.797 (21)	-3.897 (16)
merita harvest ridge	-4.215 (19)	-4.716 (18)	-4.026 (19)
roman meal	-4.321 (20)	-4.609 (17)	-4.198 (20)
brownberry dutch cou	-4.389 (21)	-4.744 (19)	-4.364 (21)

Table B.8: Maximum likelihood estimates of α_k parameters in equations (2.5) of Fresh egg - *Directly calculating the MLE required 0.212 secs and the EM algorithm required 1.646 secs (8 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (3)	0 (3)	0 (3)
egg beaters	3.219 (1)	2.987 (1)	3.222 (1)
private label	0.028 (2)	0.027 (2)	0.028 (2)
eggland best	-3.171 (4)	-3.635 (4)	-3.172 (4)

Table B.9: Maximum likelihood estimates of α_k parameters in equations (2.5) of Fruits - *Directly calculating the MLE required 0.83 secs and the EM algorithm required 19.334 secs (23 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
del monte fruit to g	-0.986 (2)	-0.626 (2)	-1.052 (2)
dole fruit bowls	-1.001 (3)	-0.745 (3)	-1.06 (3)
private label	-1.478 (4)	-1.608 (4)	-1.461 (4)
del monte	-2.091 (5)	-2.128 (5)	-2.086 (5)
white house	-2.368 (6)	-2.459 (6)	-2.358 (6)
dole	-2.512 (7)	-2.819 (7)	-2.497 (7)
ocean spray	-2.592 (8)	-2.903 (9)	-2.549 (8)
motts	-2.745 (9)	-2.888 (8)	-2.738 (9)
del monte lite	-2.885 (10)	-3.091 (10)	-2.871 (11)
motts fruitsations	-2.895 (11)	-3.195 (11)	-2.851 (10)
del monte fruit natu	-3.411 (12)	-3.911 (12)	-3.277 (12)
del monte fresh cut	-3.481 (13)	-3.955 (13)	-3.424 (13)

Table B.10: Maximum likelihood estimates of α_k parameters in equations (2.5) of Household cleaner - *Directly calculating the MLE required 1.408 secs and the EM algorithm required 39.051 secs (28 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
private label	-2.922 (2)	-3.034 (2)	-2.91 (2)
lysol	-2.948 (3)	-3.125 (3)	-2.954 (3)
greased lightning	-3.081 (4)	-3.773 (5)	-3.041 (4)
windex	-3.191 (5)	-3.814 (6)	-3.139 (5)
liquid plumr	-3.342 (6)	-4.146 (9)	-3.246 (6)
pine sol	-3.385 (7)	-3.762 (4)	-3.342 (7)
mr clean	-3.467 (8)	-4.215 (10)	-3.366 (8)
clorox	-3.542 (9)	-3.876 (7)	-3.474 (9)
clorox clean up	-3.716 (10)	-4.497 (13)	-3.541 (10)
soft scrub	-3.789 (11)	-4.467 (11)	-3.574 (11)
comet	-4.028 (12)	-4.477 (12)	-3.906 (12)
lysol cling	-4.101 (13)	-3.916 (8)	-3.963 (13)

Table B.11: Maximum likelihood estimates of α_k parameters in equations (2.5) of Ice cream - *Directly calculating the MLE required 1.429 secs and the EM algorithm required 43.164 secs (3 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (3)	0 (2)	0 (3)
ben & jerrys	1.675 (1)	1.267 (1)	1.701 (1)
haagen dazs	0.635 (2)	-0.093 (3)	0.853 (2)
breyers	-0.901 (4)	-1.052 (4)	-0.899 (4)
private label	-1.234 (5)	-1.294 (5)	-1.232 (5)
dreyers edys grand	-1.597 (6)	-2.053 (6)	-1.595 (6)
healthy choice	-1.89 (7)	-2.206 (7)	-1.89 (7)
pet	-2.035 (8)	-2.417 (8)	-2.031 (8)
breyers ice cream pa	-2.198 (9)	-2.576 (9)	-2.196 (9)
dreyers edys grand l	-2.269 (10)	-2.867 (10)	-2.265 (10)
pet signature collec	-2.459 (11)	-2.902 (11)	-2.455 (11)
dreyers edys	-2.848 (12)	-3.542 (13)	-2.835 (12)
pet homemade	-2.97 (13)	-3.54 (12)	-2.961 (13)
breyers take two	-3.675 (14)	-4.282 (14)	-3.636 (14)

Table B.12: Maximum likelihood estimates of α_k parameters in equations (2.5) of Laundry detergent - *Directly calculating the MLE required 0.574 secs and the EM algorithm required 6.878 secs (12 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
tide	-1.36 (2)	-1.435 (2)	-1.359 (2)
gain	-1.968 (3)	-2.151 (3)	-1.968 (3)
wisk	-2.384 (4)	-2.548 (4)	-2.384 (4)
purex	-2.523 (5)	-2.872 (5)	-2.522 (5)
cheer	-2.739 (6)	-3.125 (7)	-2.736 (6)
xtra	-3.02 (7)	-3.122 (6)	-3.02 (7)
all	-3.162 (8)	-3.617 (8)	-3.16 (8)
private label	-3.362 (9)	-3.809 (9)	-3.36 (9)
surf	-3.456 (10)	-4.009 (10)	-3.44 (10)

Table B.13: Maximum likelihood estimates of α_k parameters in equations (2.5) of Luncheon meats - *Directly calculating the MLE required 5.158 secs and the EM algorithm required 3.102 mins (36 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (2)	0 (2)	0 (2)
armour	0.219 (1)	0.591 (1)	0.081 (1)
oscar mayer	-0.526 (3)	-0.499 (3)	-0.541 (4)
sara lee	-0.645 (4)	-0.681 (4)	-0.518 (3)
hillshire farm deli	-0.927 (5)	-0.725 (5)	-1.045 (6)
private label	-1.04 (6)	-1.039 (6)	-1.041 (5)
hormel	-1.325 (7)	-1.403 (8)	-1.249 (7)
buddig	-1.445 (8)	-1.367 (7)	-1.497 (8)
armour premium	-1.749 (9)	-1.785 (9)	-1.591 (9)
butterball	-1.862 (10)	-1.823 (10)	-1.896 (11)
smithfield	-1.886 (11)	-2.096 (11)	-1.888 (10)
land o frost premium	-2.116 (12)	-2.395 (13)	-2.13 (12)
plumrose	-2.222 (13)	-2.246 (12)	-2.248 (14)
healthy choice deli	-2.336 (14)	-2.804 (14)	-2.225 (13)
neeses	-3.243 (15)	-3.399 (15)	-3.165 (15)
dinner bell	-3.467 (16)	-3.823 (17)	-3.18 (16)
curtis	-3.54 (17)	-3.678 (16)	-3.498 (17)
kahns	-3.795 (18)	-4.311 (18)	-3.574 (18)
frank corriher	-4.359 (19)	-4.629 (19)	-4.26 (19)
gwaltney	-4.457 (20)	-4.795 (20)	-4.345 (20)
jenkins	-4.798 (21)	-5.352 (21)	-4.627 (21)
valleydale	-5.235 (22)	-5.774 (22)	-4.81 (22)

Table B.14: Maximum likelihood estimates of α_k parameters in equations (2.5) of Oil - *Directly calculating the MLE required 0.405 secs and the EM algorithm required 5.288 secs (13 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (2)	0 (2)	0 (2)
pam	0.58 (1)	0.677 (1)	0.601 (1)
private label	-1.159 (3)	-1.169 (3)	-1.158 (3)
crisco	-1.424 (4)	-1.458 (4)	-1.42 (4)
wesson	-2.113 (5)	-2.272 (5)	-2.106 (5)
mazola	-3.018 (6)	-3.724 (6)	-2.866 (6)
wesson best blend	-3.099 (7)	-3.773 (7)	-3.009 (7)

Table B.15: Maximum likelihood estimates of α_k parameters in equations (2.5) of Margarine - *Directly calculating the MLE required 0.502 secs and the EM algorithm required 15.155 secs (30 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
shedds country crock	-1.227 (2)	-1.242 (2)	-1.224 (2)
i cant believe its n	-1.341 (3)	-1.339 (3)	-1.346 (3)
parkay	-1.553 (4)	-1.551 (4)	-1.555 (4)
land o lakes	-1.718 (5)	-1.763 (5)	-1.72 (5)
blue bonnet	-1.832 (6)	-1.862 (6)	-1.827 (6)
smart balance	-1.997 (7)	-2.235 (8)	-1.95 (7)
private label	-2.129 (8)	-2.163 (7)	-2.124 (8)
brummel & brown	-2.612 (9)	-3.01 (9)	-2.475 (9)
fleischmanns	-2.948 (10)	-3.363 (11)	-2.801 (11)
nucoa smart beat	-2.963 (11)	-3.496 (13)	-2.682 (10)
shedds cntry crck sp	-2.972 (12)	-3.442 (12)	-2.901 (12)
mrs filberts	-3.008 (13)	-3.142 (10)	-2.993 (13)

Table B.16: Maximum likelihood estimates of α_k parameters in equations (2.5) of Milk - *Directly calculating the MLE required 0.483 secs and the EM algorithm required 5.4112 secs (11 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (2)	0 (2)	0 (2)
private label	0.245 (1)	0.244 (1)	0.244 (1)
pet	-2.424 (3)	-2.417 (3)	-2.421 (3)
silk	-2.879 (4)	-2.897 (4)	-2.875 (4)
nestle nesquik	-3.176 (5)	-3.221 (5)	-3.171 (5)
hersheys-morningstar	-3.589 (6)	-4.175 (6)	-3.48 (6)
horizon organic	-3.859 (7)	-4.702 (8)	-3.81 (7)
maola	-3.959 (8)	-4.187 (7)	-3.955 (8)

Table B.17: Maximum likelihood estimates of α_k parameters in equations (2.5) of Natural cheese - *Directly calculating the MLE required 0.815 secs and the EM algorithm required 17.097 secs (21 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
kraft	-0.794 (2)	-0.781 (2)	-0.796 (3)
private label	-0.983 (3)	-1.018 (3)	-0.976 (5)
frigo	-1.001 (4)	-1.113 (5)	-0.619 (2)
athenos	-1.024 (5)	-1.05 (4)	-0.873 (4)
sargento	-1.682 (6)	-1.692 (6)	-1.688 (6)
borden	-1.881 (7)	-2.029 (7)	-1.873 (7)
kraft cracker barrel	-2.253 (8)	-2.428 (8)	-2.225 (8)
frigo cheese heads	-3.555 (9)	-4.067 (11)	-3.277 (9)
kraft classic melts	-3.557 (10)	-3.753 (9)	-3.467 (11)
frigo cheese heads s	-3.624 (11)	-4.035 (10)	-3.371 (10)
sorrento	-4.903 (12)	-5.568 (12)	-4.793 (12)

Table B.18: Maximum likelihood estimates of α_k parameters in equations (2.5) of Paper towels - *Directly calculating the MLE required 0.441 secs and the EM algorithm required 5.931 secs (13 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
bounty	-1.04 (2)	-1.083 (2)	-1.044 (2)
kleenex viva	-1.326 (3)	-1.577 (3)	-1.305 (3)
scott	-1.858 (4)	-1.957 (4)	-1.857 (4)
brawny	-2.668 (5)	-2.928 (7)	-2.663 (5)
private label	-2.703 (6)	-2.734 (5)	-2.697 (6)
sparkle	-2.749 (7)	-2.897 (6)	-2.744 (7)
mardi gras	-3.624 (8)	-4.131 (8)	-3.224 (8)

Table B.19: Maximum likelihood estimates of α_k parameters in equations (2.5) of Processed cheese - *Directly calculating the MLE required 0.713 secs and the EM algorithm required 14.961 secs (21 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
nabisco easy cheese	-1.275 (2)	-1.573 (4)	-1.139 (2)
kraft singles	-1.48 (3)	-1.482 (2)	-1.475 (3)
private label	-1.486 (4)	-1.493 (3)	-1.476 (4)
kraft deli deluxe	-1.738 (5)	-1.784 (6)	-1.748 (5)
borden	-1.778 (6)	-1.781 (5)	-1.772 (6)
kraft velveeta	-1.91 (7)	-1.939 (7)	-1.911 (7)
ruths	-2.151 (8)	-2.181 (8)	-2.154 (8)
kraft 2 percent milk	-2.889 (9)	-3.383 (9)	-2.528 (9)
borden fat free	-3.052 (10)	-3.568 (11)	-2.68 (10)
sedgefield	-3.112 (11)	-3.446 (10)	-2.902 (11)
kinsers	-3.723 (12)	-4.004 (12)	-3.584 (12)

Table B.20: Maximum likelihood estimates of α_k parameters in equations (2.5) of Salad - *Directly calculating the MLE required 0.469 secs and the EM algorithm required 12.363 secs (26 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
fresh express	-0.166 (2)	-0.136 (2)	-0.158 (2)
nobrand	-2.211 (3)	-2.671 (4)	-2.218 (4)
private label	-2.218 (4)	-2.229 (3)	-2.215 (3)
earthbound farm	-2.853 (5)	-2.982 (5)	-2.262 (5)
mrs kinsers	-3.381 (6)	-4.024 (6)	-3.249 (6)
stony gap	-4.043 (7)	-4.49 (8)	-3.639 (7)
ruths	-4.125 (8)	-4.486 (7)	-3.91 (8)

Table B.21: Maximum likelihood estimates of α_k parameters in equations (2.5) of Salad dressing - *Directly calculating the MLE required 0.452 secs and the EM algorithm required 7.082 secs (16 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
kraft	-1.166 (2)	-1.22 (2)	-1.158 (2)
kens steak house	-1.48 (3)	-1.483 (3)	-1.485 (3)
wishbone	-1.749 (4)	-1.848 (4)	-1.753 (4)
hidden valley ranch	-1.918 (5)	-2.067 (5)	-1.906 (5)
hidden valley	-2.038 (6)	-2.141 (6)	-2.031 (6)
private label	-2.149 (7)	-2.261 (7)	-2.136 (7)
kraft free	-2.345 (8)	-2.685 (8)	-2.294 (8)
kraft light done rig	-2.524 (9)	-2.846 (9)	-2.465 (9)
kraft special collec	-2.815 (10)	-3.17 (10)	-2.72 (11)
wishbone just 2 good	-2.99 (11)	-3.566 (11)	-2.696 (10)
good season	-3.301 (12)	-3.759 (12)	-3.008 (12)

Table B.22: Maximum likelihood estimates of α_k parameters in equations (2.5) of Soup - *Directly calculating the MLE required 1.659 secs and the EM algorithm required 42.8289 secs (26 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (2)	0 (2)	0 (2)
lipton recipe secret	12.661 (1)	13.851 (1)	12.292 (1)
campbells soup at ha	-1.346 (3)	-1.48 (3)	-1.364 (3)
nissin cup noodles	-1.572 (4)	-1.566 (4)	-1.525 (4)
campbells	-1.588 (5)	-1.647 (5)	-1.575 (5)
campbells chunky sou	-1.678 (6)	-1.726 (6)	-1.679 (6)
progresso	-2.077 (7)	-2.199 (7)	-2.076 (7)
campbells select	-2.118 (8)	-2.245 (8)	-2.124 (8)
private label	-2.319 (9)	-2.377 (9)	-2.301 (9)
maruchan instant lun	-2.728 (10)	-2.937 (10)	-2.512 (10)
healthy choice	-2.863 (11)	-3.399 (12)	-2.773 (11)
campbells healthy re	-3.227 (12)	-3.286 (11)	-3.217 (12)
swanson	-3.706 (13)	-3.856 (13)	-3.676 (13)
nissin top ramen	-4.351 (14)	-4.503 (14)	-4.316 (14)
swanson natural good	-4.454 (15)	-5.006 (16)	-4.371 (15)
maruchan	-4.668 (16)	-4.837 (15)	-4.629 (16)

Table B.23: Maximum likelihood estimates of α_k parameters in equations (2.5) of Salty snacks - *Directly calculating the MLE required 4.320 mins and the EM algorithm required 1.73 hours (24 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
baked lays	-1.952 (2)	-2.038 (2)	-1.947 (2)
lays	-2.437 (3)	-2.464 (3)	-2.43 (3)
frito lay	-2.579 (4)	-2.823 (6)	-2.538 (5)
cape cod	-2.667 (5)	-2.973 (9)	-2.467 (4)
doritos	-2.7 (6)	-2.723 (4)	-2.696 (6)
tostitos	-2.755 (7)	-2.778 (5)	-2.768 (7)
funyuns	-2.927 (8)	-2.904 (7)	-2.975 (10)
pringles	-2.951 (9)	-2.971 (8)	-2.966 (9)
wise	-3.02 (10)	-3.013 (10)	-3.036 (11)
tostitos scoops	-3.077 (11)	-3.345 (15)	-3.043 (12)
chee tos	-3.113 (12)	-3.125 (11)	-3.119 (16)
wavy lays	-3.113 (12)	-3.156 (12)	-3.103 (14)
private label	-3.128 (14)	-3.18 (13)	-3.078 (13)
general mills chex m	-3.187 (15)	-3.268 (14)	-3.196 (17)
baked doritos	-3.267 (16)	-3.63 (18)	-2.962 (8)
ruffles	-3.308 (17)	-3.466 (17)	-3.303 (18)
fritos	-3.323 (18)	-3.373 (16)	-3.317 (20)
jays krunchers	-3.372 (19)	-3.664 (19)	-3.104 (15)
bugles	-3.476 (20)	-3.747 (22)	-3.313 (19)
fritos scoops	-3.614 (21)	-3.73 (21)	-3.591 (22)
snyders of hanover	-3.649 (22)	-3.694 (20)	-3.613 (23)
wise cheez doodles	-3.769 (23)	-3.821 (23)	-3.737 (25)
sunchips	-3.786 (24)	-4.145 (25)	-3.441 (21)
rold gold	-3.894 (25)	-3.935 (24)	-3.858 (26)
pringles fat free	-3.924 (26)	-4.265 (27)	-3.624 (24)
wise ridgies	-3.936 (27)	-4.18 (26)	-3.897 (28)
moores	-3.981 (28)	-4.375 (29)	-3.889 (27)
pringles right crisp	-4.154 (29)	-4.348 (28)	-4.06 (29)
munchies	-4.356 (30)	-4.743 (30)	-4.101 (30)
lance thunder	-4.503 (31)	-4.794 (32)	-4.355 (32)
frenchs	-4.53 (32)	-4.793 (31)	-4.309 (31)
fritos flavor twists	-4.954 (33)	-5.208 (33)	-4.791 (33)
pringles cheezums	-5.216 (34)	-5.543 (34)	-4.939 (34)
wise bravos	-5.472 (35)	-6.022 (35)	-5.257 (35)

Table B.24: Maximum likelihood estimates of α_k parameters in equations (2.5) of Spices - *Directly calculating the MLE required 0.627 secs and the EM algorithm required 11.077 secs (18 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
mccormick	-1.395 (2)	-1.862 (4)	-1.14 (2)
durkee	-1.489 (3)	-1.944 (5)	-1.245 (3)
private label	-1.58 (4)	-1.568 (2)	-1.584 (4)
morton	-1.59 (5)	-1.661 (3)	-1.595 (5)
diamond salt sense	-3.064 (6)	-3.604 (6)	-2.697 (6)
a & a spice world	-3.32 (7)	-3.758 (8)	-2.93 (7)
lawrys	-3.405 (8)	-3.73 (7)	-3.143 (8)
mrs dash	-3.85 (9)	-4.249 (9)	-3.384 (9)

Table B.25: Maximum likelihood estimates of α_k parameters in equations (2.5) of Toilet tissue - *Directly calculating the MLE required 0.318 secs and the EM algorithm required 4.341 secs (14 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
scott	-1.249 (2)	-1.272 (2)	-1.25 (2)
kleenex cottonelle	-1.493 (3)	-1.522 (3)	-1.494 (3)
quilted northern	-2.064 (4)	-2.222 (5)	-2.064 (4)
quilted northern ult	-2.084 (5)	-2.653 (8)	-2.079 (5)
angel soft	-2.143 (6)	-2.189 (4)	-2.143 (6)
charmin ultra	-2.378 (7)	-2.517 (6)	-2.378 (7)
private label	-2.586 (8)	-2.598 (7)	-2.586 (8)
charmin	-2.655 (9)	-2.766 (9)	-2.655 (9)

Table B.26: Maximum likelihood estimates of α_k parameters in equations (2.5) of Trash bags - *Directly calculating the MLE required 0.348 secs and the EM algorithm required 2.201 secs (6 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (1)	0 (1)	0 (1)
private label	-1.108 (2)	-1.103 (2)	-1.108 (2)
ziploc	-2.406 (3)	-2.468 (3)	-2.406 (3)
glad	-2.954 (4)	-3.38 (4)	-2.954 (4)
glad lock	-3.055 (5)	-3.551 (5)	-3.055 (5)
hefty cinch sak	-3.369 (6)	-4.1 (7)	-3.369 (6)
hefty one zip	-3.422 (7)	-3.764 (6)	-3.422 (7)

Table B.27: Maximum likelihood estimates of α_k parameters in equations (2.5) of Vegetables - *Directly calculating the MLE required 2.379 secs and the EM algorithm required 1.010 mins (25 times as long)*

Brands	**MLE** (Rank)	**all available** (Rank)	**missing 0** (Rank)
osg	0 (2)	0 (2)	0 (2)
giorgio	4.537 (1)	3.963 (1)	4.794 (1)
private label	-0.988 (3)	-0.987 (3)	-0.988 (3)
del monte fresh cut	-1.371 (4)	-1.386 (4)	-1.371 (4)
green giant	-1.729 (5)	-1.755 (5)	-1.729 (5)
greenwood	-1.988 (6)	-2.316 (8)	-1.931 (6)
glory foods	-1.99 (7)	-2.082 (6)	-1.993 (7)
bush brothers bushs	-2.172 (8)	-2.165 (7)	-2.171 (8)
green giant le sueur	-2.784 (9)	-2.831 (9)	-2.785 (10)
margaret holmes	-2.785 (10)	-2.956 (11)	-2.76 (9)
lucks	-2.848 (11)	-2.837 (10)	-2.846 (11)
hanover	-3.12 (12)	-3.122 (12)	-3.118 (13)
green giant mexicorn	-3.191 (13)	-3.683 (14)	-2.87 (12)
green giant kitchen	-3.347 (14)	-3.53 (13)	-3.336 (14)
bruces	-3.459 (15)	-3.845 (16)	-3.42 (15)
bush brothers chili	-3.652 (16)	-4.159 (18)	-3.468 (16)
veg all	-3.7 (17)	-3.78 (15)	-3.693 (17)
del monte	-3.721 (18)	-3.852 (17)	-3.708 (18)
allens	-4.254 (19)	-4.789 (19)	-4.086 (19)

Bibliography

Alptekinoglu, A. and Semple, J. (2018). Heteroscedastic exponomial choice. *SSRN Electronic Journal*.

Alptekinoğlu, A. and Semple, J. H. (2016). The exponomial choice model: A new alternative for assortment and price optimization. *Operations Research*, 64(1):79–93.

Bohning, D., Dietz, E., Schaub, R., Schlattmann, P., and G, L. B. (1994). The distribution of the likelihood ratio for mixtures of densities from the one-parameter exponential family. *Annals of the Institute of Statistical Mathematics*, 46(2):373–388.

Briesch, R., Dillon, W., and Fox, E. (2013). Category positioning and store choice: The role of destination categories. *Marketing Science*, 32:488–509.

Campo, K., Gijsbrechts, E., and Nisol, P. (2003). The impact of retailer stockouts on whether, what and how much to buy. *International Journal of Research in Marketing*, 20(3):273–286. Pagination: 13.

Conlon, C. T. and Mortimer, J. H. (2013). Demand estimation under incomplete product availability. *American Economic Journal: Microeconomics*, 5(4):1–30.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Lindsay, B. G. (1995). Mixture models: Theory, geometry and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics*, 5:i–163.

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In Zarembka, P., editor, *Frontiers in Econometrics*, page 105–142. Academic Press, New York.

McLachlan, G. and Krishnan, T. (2008). *The EM algorithm and extensions*. Wiley series in probability and statistics. Wiley, Hoboken, NJ, 2. ed edition.

Musalem, A., Olivares, M., Bradlow, E. T., Terwiesch, C., and Corsten, D. (2010). Structural estimation of the effect of out-of-stocks. *Management Science*, 56(7):1180–1197.

Talluri, K. and van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33.

Thurstone, L. L. (1927). A law of comparative judgment. *Psychological Review*, 34(4):273–286.

Train, K. E. (2009). *Discrete Choice Methods with Simulation*. Cambridge University Press, 2 edition.

Vulcano, G., van Ryzin, G., and Ratliff, R. (2012). Estimating primary demand for substitutable products from sales transaction data. *Operations Research*, 60(2):313–334.