Exploring probabilistic integration for the estimation of the mixed multinomial logit model

Niels Janssen, 450759
Erasmus University Rotterdam
Erasmus School of Economics*
Bachelor Thesis: Econometrics and Operations Research
Supervisor: A. Castelein, MSc
Second assessor: C.D. van Oosterom, MSc

July 6, 2019

Abstract

The traditional numerical integration method for estimating the mixed multinomial logit model is Monte Carlo Simulation, either based on pseudo-random or quasi-random sequences. The quasi-random sequence used in this paper is the Halton sequence. In recent years, a different approach to numerical integration, called probabilistic integration, has gained more traction. Probabilistic integration uses the uncertainty inherent to numerical integration caused by the impossibility of being able to evaluate the integrand at an infinite amount of points. This paper seeks to explore the possibility of applying one such method called Bayesian cubature to the estimation of the mixed multinomial logit model, and see if it is a viable method for this estimation process. To illustrate Bayesian cubature, Bayesian Quasi-Monte Carlo with a Gaussian kernel is used. The results of this type of Bayesian cubature do not approach the results obtained by the traditional methods for a mixed logit with a normal distribution. However, more advanced Bayesian cubature methods might yield stronger results, especially when more complex distributions are used for the mixed logit, and drawing many states is computationally intensive.

*The views stated in this thesis are those of the author and not necessarily those of Erasmus School of Economics or Erasmus University Rotterdam.

Contents

1 Introduction

Multinomial logistic regression is one of the most common and popular models for multinomial classification problems. According to Google trends in the past ten years, the search term 'Logistic regression' has a worldwide relative average interest score of 42. In comparison with 'Linear regression', which has a score of 65, 'econometrics' a score of 36, and 'Erasmus university Rotterdam' a score of 33. Finalised by McFadden in 1974, logistic regression has since been a standard model in an econometrician's or economist's toolbox as it has a multitude of practical applications. Ranging from recognizing handwritten digits, to what product a consumer would buy; in any categorical problem the multinomial logit has a use. The strong points of the multinomial logit are that it is easy to estimate, allows for in-depth analysis, and clear interpretation. Train (2009) elucidates three possible limitations of the multinomial logit: it can only represent non-random taste variation, independent of irrelevant alternatives, and no correlation in unobserved factors over time.

A general model which mitigates these problems, and can approximate all models based on unobserved utility, is the mixed multinomial logit model (Mcfadden and Train, 2000). The main difference between the multinomial logit model and the mixed logit model is that for the mixed logit the coefficients are drawn from a distribution. Even though the mixed logit mitigate the problems present in the multinomial logit and allows for more flexible structures, estimating a mixed logit model comes with its own challenges. The main challenge the estimation process faces is the computationally intensive integral that is present in the choice probability, and thus the log-likelihood. As the integral is difficult or impossible to solve analytically, methods called numerical integration are developed to approximate these integrals. One of the most common numerical integration methods relies on simulating the integral. The method of choice for the mixed logit is called *Monte Carlo Simulation* (Bhat, 2001).

Monte Carlo simulation relies on drawing random numbers, calculating the function value for these random draws and then simply averaging them. This integration method relies heavily on the sampling distribution of the drawn random numbers, as it is impossible to draw an infinite amount of numbers. On top of that, the estimation returns a single numerical value and ignores the potential error which is inherent to Monte Carlo methods, due the fact that it relies on random numbers. O'Hagan (1987) goes more in depth as to why Monte Carlo is, as O'Hagan even calls it, 'Fundamentally unsound' and states that it violates the *Likelihood Principle*. In essence, O'Hagan concludes that integral estimation is not a numerical, but instead a statistical problem. The estimation process should not return a single numerical value, but a distribution which encompasses the uncertainty in the prediction. This field of thinking is called *Probabilistic numerics*, and is heavily correlated with Bayesian statistics.

Hennig et al. (2015) delivers a 'call to

arms' for *Probabilistic numerics* for numerical analysis, such as numerical integration, which inherently exhibits uncertainty. Not only could probabilistic integration provide more accurate and sound estimations, but also better estimations with fewer iterations, especially when the evaluation of the function over which the integral is calculated is difficult to evaluate. This is what this paper seeks to do; explore the possibility of applying one such technique called *Bayesian cubature* to the estimation process of the mixed logit. To achieve this, a tour d'horizon is presented for the Bayesian cubature method, with the traditional Monte Carlo methods in mind.

This is done by differentiating the different terminologies used in the literature, and discussing how the different kinds of Bayesian cubature methods relate to each other. It is beyond the scope of this paper to examine all Bayesian cubature methods. Therefore, to illustrate Bayesian cubature, the method used in this paper is the method introduced by Rasmussen and Ghahramani (2003). However, effort is made to generalise Bayesian cubature as such that other more advanced types of Bayesian cubature can be understood as well. Subsequently, this method is compared with the more traditional methods which rely on Monte Carlo simulation and figure out which method is more appropriate and/or better. The Bayesian cubature results presented in this paper do not approach the traditional Monte Carlo methods. Nonetheless, other types of Bayesian cubature could still prove to be useful in certain mixed logit models. Thus, this paper contributes to the discussion of when probabilistic numerics is appropriate and when it is not, and 'unsound' methods might yield more practical results.

This thesis will start off with a literature review in which three main subjects are discussed: the (mixed) multinomial logit model, Monte Carlo methods, and Bayesian method. Secondly, the methods are discussed in the methodology. First the model specification, and afterwards the numerical integration methods. After each section the corresponding parameter estimation will be touched upon. Thirdly, the simulation study is discussed and the DGP is specified. Hereafter, the results of this simulation study are presented and subsequently discussed.

2 Literature

2.1 (Mixed) Multinomial Logit

McFadden (1973) first introduced the conditional logit model, which has become synonymous with the multinomial logit. McFadden was not the first to introduce the logistic regression, it first developed by Berkson (1944). After its introduction, it has since been researched extensively. Train (2009) attributes this in chapter 3.1, due to the fact that the function takes a closed form and is readily interpretable. Even though the multinomial logit model is a powerful model with many uses, it does have its constraints. In chapter 6.1, Train (2009) names three constraints that the mixed logit mitigates.

The first constraint is that the multino-

mial logit cannot represent random variation in preference for brands across individuals. In other words, the coefficients are equal for all individuals. For some purposes this assumption might be valid, but for numerous other purposes, it is not. Namely models which concern brand preference, an underlying income difference might heavily influence the outcome. Without knowing exactly which individuals correspond to higher or lower incomes, it is impossible for the multinomial logit to encompass this.

The second constraint is the independence of irrelevant alternatives (IIA) property. For the multinomial logit model, the ratio of two choice probabilities is independent from other choice probabilities. This property fails if a third probability changes. The sum of all probabilities must equal to one, so one of the other two probabilities has to change. Consequentially, the ratio of the two probabilities changes and the IIA property is violated. The famous bus example is given to illustrate the IIA property. Suppose that in order to travel from A to point B, an individual can choose either a car or a blue bus. The probability to choose either the car or bus is equal such that $\mathbb{P}(car) = \mathbb{P}(bus_{blue}) = \frac{1}{2}$. Now imagine that a second identical bus is introduced, except that it is red instead of blue. It would be expected that $\mathbb{P}(car) = \frac{1}{2}$ and $\mathbb{P}(bus_{blue}) = \mathbb{P}(bus_{red}) = \frac{1}{4}$. Except the multinomial logit will predict $\mathbb{P}(car) = \mathbb{P}(bus_{blue}) = \mathbb{P}(bus_{red}) = \frac{1}{3}$. Train (2009) goes in depth as to what the further consequence of the IIA property are, and also discusses the advantages that the IIA property brings to the table.

The third constraint concerns panel data, specifically correlation between unobserved explanatory variables. If the variables are independent from each other, they can be incorporated in the multinomial logit model. An explanatory variable from period $t-1$ is for example allowed to be used as an explanatory variables for the next period, as long as there is not a unobserved variable that influences the dependent variable for both periods $t-1$ and t. Except this assumption is quite strict in the sense that it is likely that some unobserved variable both influences period $t-1$ and t.

The mixed logit relaxes the three constraints inherent to the multinomial logit, and can also approximate all models based on unobserved utility (Mcfadden and Train, 2000). However, the mixed logit is not without its own limitations. First, the integral present in the choice probability function hinders the estimation process. This can mainly be contributed to the fact that for every coefficient which is drawn from a distribution, the dimensionality of the integral increases. Second, the estimation process is also hampered when drawing from the distribution is difficult or computationally complex.

2.2 Monte Carlo Simulation

The most straightforward way of estimating the choice probabilities is with Monte Carlo simulation using either pseudo-random sequences, or quasi-random sequences. Pseudo-random sequences try to emulate 'true' randomness, whereas quasi-random

sequences try to be as evenly distributed as possible. This results in quasi-random sequences being a deterministic method. Section 3.2 discusses both methods in-depth. Bhat (2001) performs an extensive simulation study to compare Monte Carlo simulation using pseudo-random sequences, quasi-random Halton sequences, and a more traditional polynomial-based cubature method. How Halton sequences are precisely generated is discussed in section 3.2.2. He concluded that QMC needs significantly less draws and a fraction of the computation time to obtain comparable root mean square errors and mean absolute percentage errors. The polynomial-based method provided adequate estimates, especially for lower dimensions. The only exception is for higher dimensions as adding draws imposes a significant increase in computation time. Overall, QMC is not only the better, but also the faster method. It takes about 100 quasi-random Halton draws to obtain the same result as 1000 pseudo-random draws in 1-5 dimensions, (Bhat, 2001). The downside of using quasi-random sequences is that they do not allow for statistical analyses of the estimation error as the quasi-random sequences are generated deterministically. Moreover, for higher dimensions Halton sequences also run into trouble as it is more likely that at some point a correlation occurs between two sequences.

Each downside motivates the use of a modified Halton sequences. Bhat (2003) proposes the use of a randomized and scrambled Halton sequence. The randomization tries to combat the statistical analyses problem by combining both quasi and random sequences to induce a small amount of randomness to each quasi-random point. Scrambling combats the correlation in higher dimensions using permutations. Bhat (2003) compares the newly created Halton sequence with the standard Halton sequence and concludes that the strong results presented in Bhat (2001) still hold, but are less significant as first thought when estimating high, in this case 10, dimensional integrals. 150 standard Halton draws is about as accurate as 500 pseudo-random draws, whilst the scrambled Halton draws only needs 100 draws. Additionally, 100 scrambled Halton draws are more accurate than 1000 pseudo-random draws. Bhat did not evaluate more than 150 draws for the standard Halton sequence.

There was still more to improve and Sivakumar et al. (2005) compared all aforementioned, as well as quasi-random (scrambled) Faure sequences for the estimation process of the mixed multinomial logit model. Faure sequences are similar to Halton sequences as in that they are the same when generating one dimensional sequences. However, for multi dimensional sequences, whilst Halton sequences simply pair one dimensional sequences, Faure sequences are generated using values from lower dimensions. Sivakumar et al. (2005) concludes that overall the Faure sequence outperforms the Halton sequence, and scrambling a sequence will result in more accurate estimates for both sequences.

2.3 Bayesian Monte Carlo

So far, the focus has been on Monte Carlo simulation for the estimation of the choice probabilities. However, a different view which has gained more traction lately (Hennig et al., 2015), is the field of probabilistic integration. Probabilistic integration is a sub-field of a bigger field, called probabilistic numerics. Probabilistic numerics applies statistical inference to numerical algorithms that return a value with a certain amount of uncertainty. For probabilistic integration this uncertainty arises from the impossibility of evaluating a function at an infinite amount of draws. In practice, even less draws are evaluated as increasing the amount draws imposes a heavy computational burden. Probabilistic integration utilises this uncertainty instead. This Bayesian approach to numerical problems can be traced as far back as Poincaré (1896), (Diaconis, 1988).

After criticising the use of Monte Carlo methods, O'Hagan developed a Bayesian cubature method which uses statistical inference to estimate integrals, (O'Hagan, 1991). His 'Bayesian-Hermite quadrature' method places a Gaussian prior on the integrand, and derives a posterior distribution for the integrand and the integral itself. Moreover, O'Hagan (1991) also uses a Gaussian kernel to obtain analytical results of the integral of a covariance and selects states the same way as for Gaussian quadrature methods. States are the values for which the integrand has to be evaluated. O'Hagan (1991) focuses on single integral evaluations in one- and multi-dimensional integrals, and obtains promising results. Rasmussen and Ghahramani (2003) followed up with introducing 'Bayesian Monte Carlo' (BMC).

Before this method is discussed, this moment is taken to clarify the term 'Bayesian Monte Carlo'. This term can be quite confusing as it suggests that Monte Carlo simulation is done in a Bayesian way. This is not true. Bayesian Monte Carlo as in Rasmussen and Ghahramani (2003) is essentially the Bayesian-Hermite method, and has hardly anything to do with Monte Carlo Simulation. Rasmussen and Ghahramani (2003) obtains in an example the same specification of the posterior distribution as the Bayes-Hermite Quadrature rule, but introduces Bayesian Monte Carlo as a more general term. The reason that they call it Bayesian Monte Carlo is because the states are generated the same way states are generated for Monte Carlo simulation. This could be based on pseudo-random, or quasi-random sequences. Because of this, the name has stuck. A better name for the general method, without kernel specification, as used in Briol et al. (2019), would be 'Bayesian Cubature'. Which would make Bayesian Monte Carlo, Bayesian cubature with states drawn the same way as for Monte Carlo simulation. Bayesian Quasi-Monte Carlo would for example mean that states are generated according to a quasi-random sequences. Other naming conventions are generated similarly.

Now that the term Bayesian Monte Carlo has been clarified, BMC as in Rasmussen and Ghahramani (2003) will be discussed.

Rasmussen and Ghahramani (2003) obtains promising results for single integral evaluations like O'Hagan (1991). Especially as not as many function evaluations are needed, which could save significant computational time. They also note the possibility of using different kernels or different prior specification, which could result in better estimates.

In recent years, probabilistic numerics, and likewise probabilistic integration, has gotten more attention. Briol et al. (2019) names multiple papers from recent years which further optimize the Bayesian cubature method first introduced by O'Hagan (1991). Briol et al. (2019) discusses in depth how Bayesian cubature in itself could be useful, and specifically provides convergence rates for BMC, BQMC and Bayesian Markov Chain Monte Carlo (BMCMC). BMCMC is a Bayesian cubature method for which the states are generated as they would have been generated for MCMC which is by using Markov chains. Briol et al. (2019) shows that for a random effects regression, BMCMC obtains comparable results as MCMC and the 95% confidence intervals for the BMCMC sketch a more realistic picture. Opposed to all other papers, Xi et al. (2018) discusses the case when one has to evaluate multiple integrals, as opposed to having to evaluate a single integral, and obtains a joint model for evaluating a finite amount of integrals. This would be highly relevant for the mixed logit, as in the log-likelihood multiple integrals are present, and for the maximization of the log-likelihood, the integrals have to be evaluated many times. Sadly, this is out of scope for this paper. This paper instead performs Bayesian cubature as in Rasmussen and Ghahramani (2003), with states generated quasi-randomly instead of pseudo-randomly.

3 Methodology

3.1 Model Specification

A key part of the mixed logit, is that it is very similar to the multinomial logit. For this reason, the multinomial logit is first derived, after which the mixed logit is derived.

3.1.1 Multinomial logit

First, utility U_{ijt} is introduced. Here, every person i can choose a alternative j during purchase occasion t and each alternative has a corresponding utility

$$\begin{aligned} U_{ijt} = X'_{ijt}\beta + \epsilon_{ijt} \text{ for } i &= \{1, \dots, N\}, \\ j &= \{0, \dots, J\}, \\ \text{and } t &= \{1, \dots, T\}. \end{aligned} \quad (1)$$

ϵ_{ijt} is i.i.d. extreme value distributed, with scale parameter set to one for identification and β is a vector of coefficients. β contains coefficient for each choice specific explanatory variable, as there are different values across choice options. X_{ijt} contains the choice specific explanatory variables corresponding to choice j that individual i observes during purchase occasion t. Y_{it} represents the predicted chosen alternatives of i during t. y_{it} is the realised chosen alternative. As ϵ_{ijt} is i.i.d. extreme value distributed, the choice probability for the con-

ditional logit can be derived (McFadden, 1973):

$$S_{ijt}(\beta) = \mathbb{P}[Y_{it} = j \mid X, \beta] = \frac{\exp(X'_{ijt}\beta)}{\sum_{l=0}^{J} \exp(X'_{ilt}\beta)}. \tag{2}$$

3.1.2 Mixed Multinomial Logit

To derive the mixed multinomial logit model, the first constraint of the multinomial logit will be relaxed. In other words, what happens to the model when the coefficient β is allowed to differ across observations? This implies that for all $i = \{1, \ldots, N\}$, β_i differs. Of course, a naive way of achieving this would be by estimating N logit models. However, there are multiple reason why this would not be desirable. For example, when N is large, every individual has few observations, or when correlation between individuals is desired. A different approach to implement this, is to draw β_i from a probability distribution $p(\beta_i \mid \theta)$. $p(\beta_i \mid \theta)$ can be any probability distribution, where θ denotes the parameters of the distribution. In this paper, $p(\beta_i \mid \theta)$ is the normal distribution with mean θ_μ and variance θ_σ.

A repercussion from this is that when calculating the choice probability, β_i is essentially unknown. Consequentially, the integral over β_i has to be taken to weigh S_{ijt} by the distribution of β_i, which is conditional on θ. This is equal to the expected value of S_{ijt} with respect to the random variable β_i. This results in the following choice probability that individual i chooses category j during purchase occasion t, which corresponds to the the mixed multinomial logit model,

$$P_{ijt}[Y_{it} = j \mid X, \theta] = \int S_{ijt}(\beta_i) p(\beta_i \mid \theta) d\beta_i. \tag{3}$$

As β_i is drawn from a distribution, θ is the parameter which is to be estimated, not β as in the case of the multinomial logit model.

For panel data, individual i chooses a sequence of alternatives $\mathbf{j} = \{j_1 \ldots j_T\}$ instead of a single alternative. Because ϵ_{ijt} in equation (3) is independently identically distributed, one can simply take the product of all purchase occasions for individual i to obtain the choice probability that individual i chooses a sequence of alternatives. This probability is defined as follows:

$$S_i(\beta_i) = \prod_{t=1}^{T} \prod_{j=1}^{J} \Big[S_{ijt}(\beta_i) \mathbb{I}[y_{it} = j] \Big],$$

$$P_i[Y_i = \mathbf{j} \mid X, \theta] = \int S_i(\beta_i) p(\beta_i \mid \theta) d\beta_i. \tag{4}$$

3.1.3 Parameter Estimation I

The objective of the mixed logit model is to obtain a model with choice probabilities. To be able to train the model coefficient θ, a loss or likelihood function which can be optimized needs to be defined first. The likelihood function used in this paper, is the log-likelihood function, which is defined as follows:

$$L[\theta \mid X] = \sum_{i=1}^{N} \log P_i. \tag{5}$$

To obtain estimates of θ, $L[\theta \mid X]$ is maximized. However, as P_i is not calculated analytically, simulated log-likelihood has to be used. This will be discussed in section 3.2.4

3.2 Numerical Integration

The purpose of numerical integration is to evaluate integrals that are analytically hard or impossible to compute. The goal of every method is to sample from the function over which the integral is to be calculated, and to weight every evaluated sample. This is also known as a *cubature rule*; Let R denote the total amount of drawn samples, $\{w_r\}_{r=1}^{R}$ the weights, P_i an abbreviation of equation (3), and $\hat{P}_i$ the estimated value of P_i. Then the following equation describes any cubature rule:

$$\hat{P}_i = \sum_{r=1}^{R} w_r S_i(\beta_{ir}). \tag{6}$$

Every numerical integration method tries to cleverly calculate the weights and/or tries to draw β_{ir} in an efficient way. For the mixed logit model, the states are drawn from the distribution $p(\beta_i \mid \theta)$.

3.2.1 Standard Monte Carlo

The first method for integral estimation that will be discussed is Monte Carlo simulation. Standard Monte Carlo (MC) simulation is a non-deterministic method. Monte Carlo estimation draws R states, and to obtain an estimate, the average is taken; $\{w_r\}_{r=1}^{R} = \frac{1}{R}$. The cubature rule for Monte Carlo estimation can be described as follows:

$$\hat{P}_i = \frac{1}{R} \sum_{r=1}^{R} S_i(\beta_r). \tag{7}$$

As 'pure' random draws are only possible theoretically, the standard way of Monte Carlo simulation is based on pseudo-random numbers which tries to emulate pure random draws.

As R approaches infinity, the estimate converges to the true value of the integral because of the law of large numbers. The strong points of SMC are that it is easy to implement and also twice differential and positive definite for any amount of draws, (Bhat, 2001). SMC does however have its downsides. Additionally, because of the pseudo-random sequence, for each SMC estimate a variance can be calculated.

As the states $\{\beta_{ir}\}_{r=1}^{R}$ are drawn randomly, R has to be very large to give a good estimation as it is desirable that the states reflect the function well and this is only consistently obtained with a high R. If a function evaluation is also costly, then SMC can be even slower.

3.2.2 Quasi-Monte Carlo

The second Monte Carlo method is based on quasi-random numbers. As discussed in the section above, drawing pseudo-randomly may not be desirable for integral estimation. Pseudo-random sequences are not designed to reflect the input space as a whole. To the contrary, they are designed to be truly random and thus many states have to be drawn to ensure that the entire space is represented correctly. Quasi-random numbers are instead not random numbers. They are a sequence of numbers which try to be as evenly distributed as possible in a set range. Quasi-random number sequences are also called low-discrepency sequences. The only difference between SMC and QMC is thus how the states are drawn.

The quasi-random sequence used in this paper are Halton sequences. Halton sequences are sequences on the domain $[0, 1)$ and are calculated iteratively where the amount of iterations depends on how long the sequences has to be. Let H_u be the Halton sequence at iteration u, with $H_0 = 0$. Every Halton sequence starts with choosing a prime α that is used to generate each iteration. The only restriction is $\alpha > 1$. The goal of each iteration is to add points to the sequence which evenly 'fill in' the gaps in the domain. In general, the sequence at iteration u can be defined as follows:

$$H_u = \{H_{u-1}, H_{u-1} + \frac{1}{\alpha^u}\} \text{ for } u \in \mathbb{Z} \quad (8)$$

As becomes clear from Figure 1, each subsequent iteration generates values which fill in the gaps created by the previous iteration. Sequences for different primes are created similarly.

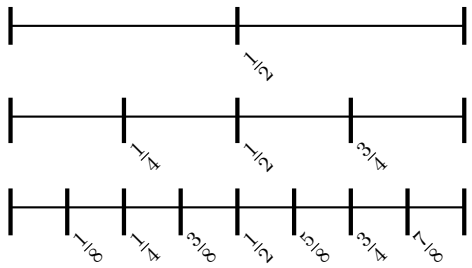

Figure 1: Three iterations for a Halton sequence with $\alpha = 2$

Halton sequences for higher dimensions are generated by defining a prime for each dimension, to ensure that both sequences are independent from each other. Moreover, the first 10 values are deleted as they are potentially correlated. This is the same reason why primes are used in the first place: to minimize potential correlation between sequences created by different primes.

To obtain the necessary Halton draws, one $N \times R + 10$ Halton sequence, with corresponding dimension, is drawn. To obtain individual specific sequence, the first the first 10 results are discarded, then, the first R values correspond to the first individual, the second R to the second, et cetera. This way, no correlation between simulation errors across individuals is present. As the values generated by the Halton sequence are on the domain [0,1), the values are transformed by the inverse CDF of the distribution $p(\beta_i \mid \theta)$.

3.2.3 Bayesian Cubature

To transform the problem of estimating Equation (4) into a Bayesian problem, instead of seeing P_i as being solely random as choice probability, the outcome of the integral itself is also random for a given θ and X. This might not be an intuitive approach, but it is inline with Bayesian thinking. In this case, uncertainty rises from the impossibility of being able to evaluate $S_i(\beta_i)$ at every single point. For ease of notation, let S_i denote $S_i(\beta_i)$. As P_i depends on S_i, a prior is first placed on S_i. Combining this with known samples, a posterior of P_i is obtained. The known samples are the following set: $\Omega_i = \{(\beta_{ir}, S_i(\beta_{ir})) \mid r = 1 \dots R\}$ The most straightforward way of putting priors over functions, is through Gaussian Processes. For a Gaussian prior, the joint distribution of S_i for the draws $\{\beta\}_{r=1}^{R}$ is defined as follows:

$$\mathbf{S}_i = (S_i(\beta_{i1}), \dots, S_i(\beta_{iR}))^T \sim N(0, C), \quad (9)$$

where the mean of the prior is set to zero, without loss of generality (Briol et al., 2019). C can be any covariance function. By putting a prior over S_i, the posterior distribution $\delta(S_i \mid \Omega_i)$ has the following posterior mean and covariance function, (Rasmussen and Ghahramani, 2003).

$$\begin{aligned} m_i(S_i \mid \Omega) &= \mathbb{E}[S_i \mid \Omega_i] \\ &= \int S_i \delta(S_i \mid \Omega_i) dS_i \\ &= c(\beta_i, B_i) C^{-1} S_i(B_i) \\ V_i(S_i \mid \Omega_i) &= c(\beta_i, \beta_i') - c(\beta_i, B_i) C^{-1} c(B_i, \beta_i), \end{aligned}$$

here, $B_i = (\beta_{i1} \dots \beta_{iR})$, $c(B_i, B_i) = C$ with entries $C_{pq} = c(\beta_{ip}, \beta_{iq})$ and $c(\beta_i, B_i) = (c(\beta_i, \beta_{i1}), \dots, c(\beta_i, \beta_{iR}))$. It is important to note that here β_i is an unknown input variable, and β_{ir} is a drawn and known input variable.

To calculate the expected value of P_i, P_i is defined as: $P_i = u(S_i \mid \Omega)$. Where u is a linear projection, (Rasmussen and Ghahramani, 2003). Then, with the help of theorem 5.2.1 in chapter 5.2 of Bain and Engelhardt (1992), the expected value of P_i yields the following result.

$$\begin{aligned} \mathbb{E}[P_i] &= \mathbb{E}[u(S_i \mid \Omega)] \qquad (10) \\ &= \int u(S_i \mid \Omega) \delta(S_i \mid \Omega) dS_i \\ &= \int \int S_i p(\beta_i \mid \theta) d\beta_i \delta(S_i \mid \Omega_i) dS_i \\ &= \int \Big(\int S_i \delta(S_i \mid \Omega_i) dS_i \Big) p(\beta_i \mid \theta) d\beta_i \\ &= \int m_i(S_i \mid \Omega) p(\beta_i \mid \theta) d\beta_i \\ &= \int c(\beta_i, B_i) p(\beta_i \mid \theta) d\beta_i C^{-1} S_i(B_i) \\ &= z C^{-1} S_i(B_i). \qquad (11) \end{aligned}$$

The obtained result can be in closed form if the *kernel mean* ($\int c(\beta_i, B_i) p(\beta_i \mid \theta) d\beta_i$), also called the *representer of integration* can be obtained in closed form and can thus encompass prior knowledge about the covariance, (Xi et al., 2018). To obtain $\hat{P}_i$, the mode of the posterior is used. As the Gaussian prior is a conjugate prior, the posterior is also Gaussian and for a Gaussian the mode equals the mean thus $\hat{P}_i = \mathbb{E}[P_i]$. Using this result, the Bayesian cubature rule is defined as follows:

$$\begin{aligned} \hat{P}_i = \mathbb{E}[P_i] &= z C^{-1} S_i(B_i) \\ &= \sum_{r=1}^{R} w_r S_i(\beta_i), \qquad (12) \end{aligned}$$

where $w = \int c(\beta_i, B_i) p(\beta_i \mid \theta) d\beta_i C^{-1} = z C^{-1}$. This result becomes Bayesian Monte Carlo when the observations or samples or states $\{\beta_i\}_{r=1}^{R}$ are generated using pseudo-random sequences, Bayesian Quasi-Monte Carlo when the states are generated using quasi-random sequences, or Bayesian Markov Chain Monte Carlo when the states are generated using a Markov chain. Xi et al. (2018) suggests the use of the Matérn covariance function as kernel. For the sake of simplicity, this paper follows the methods used in Rasmussen and Ghahramani (2003) with quasi-random sequences, which will now be explained.

Rasmussen and Ghahramani use the following Gaussian kernel: $N(a_r = \beta_{ir}, A = (v_0^2, \dots, v_R^2)$ and the following following covariance function:

$$\begin{aligned} C_{pq} &= \mathrm{Cov}(S_i(\beta_{ip}), S_i(\beta_{iq})) \\ &= v_0 \sum_{d=1}^{D} \frac{(\beta_{ip}^d - \beta_{iq}^d)^2}{v_d}, \qquad (13) \end{aligned}$$

where D is defined as follows; $\beta_{ir} \in \mathbb{R}^D$. v are the hyper parameters of the Gaussian process. The optimization of the hyper parameters are discussed in depth by Williams and Rasmussen (1996), but for this paper $v = 1$. Rasmussen and Ghahramani (2003) obtains the following analytical result for this kernel when $p(\beta_i \mid \theta)$ is also Gaussian. This result is the same as the Bayesian-Hermite quadrature in O'Hagan (1991). Let $p(\beta_i \mid \theta) \sim N(k = \theta_\mu, K = \text{diag}(\theta_\sigma^2)$, then z in Equation (11) is:

$$z_r = v_0 \mid A^{-1}K + I \mid^{-\frac{1}{2}} \times \exp(-0.5(a-k)^{\mathrm{T}}(A+K)^{-1}(a-b)). \quad (14)$$

$\mathrm{V}[P_i]$ can be obtained in a similar way, (Rasmussen and Ghahramani, 2003). In this paper emphasise is placed on the derivation of the expected value as that is used to estimate P_i.

$$V[P_i] = v_0 \mid 2A^{-1}K + I \mid^{-\frac{1}{2}} - z^{\mathrm{T}}C^{-1}z \quad (15)$$

Finally, for the Bayesian cubature method used in this paper, the states $\{\beta_{ir}\}_{r=1}^{R}$ are generated using Halton sequences as, according to section 4.2, they will prove to be superior to pseudo-random sequences.

3.2.4 Parameter Estimation II

In section 3.1.3 the log-likelihood is introduced, now the simulated log-likelihood is introduced. To obtain the simulated log-likelihood, the estimated choice probabilities $\hat{P}_i$ will be inserted into equation (5).

$$\hat{L}[\theta \mid X] = \sum_{i=1}^{N} \log \hat{P}_i \quad (16)$$

To maximize the simulated log-likelihood function, the L-BFGS algortihm will be used. L-BFGS approximates the BFGS algorithm, but uses limited computed memory. L-BFGS, like BFGS, does not requires the specification of a gradient. This method was chosen as the gradient of the Bayesian Monte Carlo method discussed in section 3.2.3 is hard to evaluate. Granted, the fact that a optimizer which uses a derivative cannot be used for BMC has to be taken into account when comparing integration methods.

Everything was programmed in Python, relying mostly on the NumPy package for calculations, and the minimize function from the SciPy package to maximize the simulated the log-likelihood.

4 Simulation study

4.1 DGP

To accurately compare the different numerical integration methods, a data set is simulated. The base dataset comes from Jain et al. (1994). This dataset contains the purchase history of 300 households in Springfield, Missouri, who purchased different kinds of ketchup. The integer after the brand name denotes the bottle size. For every purchase event, three variables were measured: display (if the item was displayed prominently in the store at the time of purchase), feature (if there was a newspaper advertisement) and price of the item. Display and feature are binary variables, where 1 denotes that the event happened, and 0 if it did not. Price is the actual price for the item sold, and the shelf price for the other items. Let N denote the total sample size of 2798, and $J = 4$ the amount of choice specific

variables. Lastly, y_{it} contains the purchased item for each observation.

To accurately compare all estimation methods, the y provided by Jain et al. (1994) is not used; instead y will be generated by quasi-random draws from a data generating process. For each purchase occasion, the corresponding utility and y_{it} is calculated as follows:

$$\begin{aligned} U_{ijt} &= X_{ijt}\beta_i + \epsilon_{ijt} \\ Y_{it} &= \text{argmax}_j(U_{ijt}). \end{aligned} \tag{17}$$

β_i is the choice specific parameter, with $\beta \sim N(\theta)$ to denote the DGP. ϵ_{ijt} is i.i.d. extreme value distributed and models the random effects inherent to the multinomial mixed logit. To obtain estimates for θ, Standard Monte Carlo simulation is used with 200 draws. For the display variable the mean and standard deviation are $1.4, 0.4$, for the feature variable $(1.0, 0.1)$ and for the price $-1.1, 0.6$.

4.2 Results

To evaluate the numerical estimation methods, the methods are measured by their ability to recover the true parameters and choice probabilities for each alternative for every purchase incidence, and computation time. The first two are compared using their root mean squared error (RMSE) and mean absolute percentage error (MAPE). The computation time is the time it takes one method to compute all choice probabilities for every individual. The amount of operations needed would have been less dependent on computational power, except, to speed up the calculations, the Python package 'Numba' was used. This made it difficult to obtain the precise amount of operations needed to optimize one mixed logit model. To combat the dependency on local computational power and obtain more reproducible results, the computation times were obtained using the free online cloud service Google Colab.

This results in four performance measures and additionally the computation time, for every estimation method. To compute the 'true' probability benchmark, a Quasi-Monte Carlo simulation with 20.000 draws was done. There is, of course, still some error present in these probabilities, as it is still an estimation. However, the expected error will be small enough that the results can serve as a benchmark. The results of the simulation study can be found in Table 1.

Firstly, Table 1a illustrates that increasing the amount of draws increases the estimation precision and 2000 draws is significantly better than fewer draws. However, at the same time, increasing the amount of draws also increases the computation time. Both error measures for the ability to retrieve the true parameters and choice probabilities decrease each from 250 to 2000 draws with roughly 20%, and 60% respectively, whilst the computation time increases with 444%.

Turning now to the estimation results of the QMC method in Table 1b, it is evident that that QMC with Halton draws outperforms SMC. As few as 100 Halton draws obtains better parameter estimates and more accurate choice probabilities than 1000 pseudo-random draws. Additionally, 75

Table 1: Maximum simulated log-likelihood estimation results

(a) Estimation results for Standard Monte Carlo (SMC)

Evaluation basis	Performance measure	SMC			
		Number of draws			
		250	500	1000	2000
Parameters	MAPE	15.98	15.55	14.79	12.77
	RMSE ($\times 10^{-2}$)	4.09	3.99	3.62	3.17
Choice probabilities	MAPE	2.35	1.63	1.25	0.84
	RMSE ($\times 10^{-3}$)	5.06	3.36	2.59	1.74
Computation time		0.55	0.80	1.66	2.99

(b) Estimation results for Quasi Monte Carlo (QMC)

Evaluation basis	Performance measure	QMC				
		Number of draws				
		25	50	75	100	125
Parameters	MAPE	13.17	14.45	16.41	13.78	11.49
	RMSE ($\times 10^{-2}$)	4.55	3.92	4.02	3.42	2.85
Choice probabilities	MAPE	2.16	1.55	1.07	0.69	0.58
	RMSE ($\times 10^{-3}$)	3.65	2.73	1.88	1.21	0.95
Computation time		0.19	0.25	0.32	0.36	0.41

(c) Estimation results for Bayesian Quasi Monte Carlo (BQMC)

Evaluation basis	Performance measure	BQMC				
		Number of draws				
		10	15	20	25	30
Parameters	MAPE	54.61	36.72	50.89	49.68	36.22
	RMSE ($\times 10^{-2}$)	41.14	26.79	26.70	25.86	21.89
Choice probabilities	MAPE	48.95	43.32	36.54	32.52	30.95
	RMSE ($\times 10^{-3}$)	176.55	162.99	145.15	132.42	126.23
Computation time		0.40	0.59	0.80	1.09	1.31

This table contains the maximum simulated log-likelihood results for SMC, QMC and BQMC. The performance measures evaluate each model's ability to retrieve the model parameters or choice probabilities. MAPE is the mean absolute percentage error, RMSE the root mean squared error. Computation time is measured in seconds.

Halton draws provide more accurate choice probabilities than 1000 pseudo-random draws, and 100 Halton draws better than both 1000 and 2000 pseudo-random draws. Moreover, QMC is much faster across the board. Adding more draws to increase the the accuracy of QMC is significantly more attractive than for SMC. Performing 125 draws instead of 25 decreases the error rates for the parameters and choice probabilities each with roughly 60% (for the RMSE), and 70% respectively, whilst the computation time increases with 112%. Comparing these results with the above results from SMC, it is clear that QMC is not only more efficient computation wise when adding draws, but also more accurate for fewer draws.

It is also surprising to see that the only parameter MAPE for 25 draws is lower than all draws but 125 draws. At the same time the corresponding RMSE is higher than all other draws, same for both choice probability measures. Furthermore, both parameter measures for 75 draws do not follow the downward trend expected. Bhat (2001) observes a similar result for 100 draws when also estimating three dimensional integrals.

In the final part of Table 1, Table 1c, the results of the Bayesian cubature method are found. The Bayesian cubature method evaluated in this paper is the Bayesian Quasi-Monte Carlo method, which uses Halton sequences. What immediately can be observed is that the results for BQMC do not stack up against both QMC and SMC. All measures for both the ability to retrieve the parameters and choice probabilities are significantly worse. A downward sloping trend of the performance measures by adding more draws is observed. However, when combining the initial error and added computation time, increasing the draws to approach the errors achieved by SMC or QMC is not feasible. The computation time for 50 draws is 4.7 seconds, and for 75 draws 8.13 seconds, which showcases the computational burden of adding more points.

As with QMC with 25 draws, the parameter MAPE for 15 draws is significantly lower than the expected, whilst the other measures are inline with the other draws.

5 Discussion

When comparing the results for SMC and QMC with a similar study, (Bhat, 2001), the same downward trend in error measure when adding draws is observed, as well as the superiority of QMC with respect to SMC. This is as expected as Halton draws are evenly distributed across the sample space, and thus would be a more efficient way of generating draws for Monte Carlo simulation. The only aberrant observations which do not adhere to the statement that more draws equals betters estimations, is the parameter MAPE for 25 QMC draws and 15 BQMC draws, and both parameters measures for 75 draws. The phenomenon observed for 75 draws is also observed for 100 draws in Bhat (2001). As such, this will first be discussed.

This phenomenon can be explained due to the way the Halton sequences for each indi-

vidual are generated. Section 3.2.2 states that to generate the necessary Halton sequence, one D dimensional $N \times R + 10$ sequence is generated, and that the first R values belong to the first individual, et cetera. A consequence of this sampling scheme is that when performing $R + 20$ draws instead of R, each individual does not 'keep' their old R draws, with 20 new draws added. To the contrary, a new set of values is assigned to every individual. This results in the possibility that for a certain amount of draws, less draws could result in a better estimation opposed to more draws with new and different values. However, this is a minor aberration and should not imply a causation between more draws and less accurate estimations, (Bhat, 2001).

The second aberrant observation, the counter intuitive observation that the parameter MAPE for 25 QMC and 15 BQMC draws does not adhere to the general trend of more draws equals more precise estimations whilst the RMSE does, could be attributed to a fault inherent to the MAPE measure. One critique for the MAPE is that is it asymmetric when results are strictly positive, which is the case for the predicted standard errors, (Makridakis, 1993). A consequence of this is that if a prediction is lower than the actual, there is a upper bound for the maximum error, 100%, as the lowest worst prediction is 0. To the contrary, there is no upper bound for the maximum error for prediction higher than the actual. This results in a larger penalty for predictions which are larger than the true value. To give a small example, let $x = 0.5$ and $\bar{x} = 0.1$. When x is the true value, the MAPE equals $100\% \mid \frac{0.5-0.1}{0.5} \mid = 80\%$. Suppose now that $\bar{x}$ is the true value. In this case, the MAPE equals $100\% \mid \frac{0.1-0.5}{0.1} \mid = 400\%$. The same difference results in two different errors. Especially when keeping in mind that the parameter RMSE does follow the general trend, the aberrant MAPE value for both 25 QMC and 15 BQMC draws, does not indicate a different general trend and can be ignored.

In this section, the BQMC results are discussed. Although the method is theoretically sound, it does not deliver adequate results in practice. Besides having large estimation errors, there is also a large computational burden for adding draws to obtain more accurate estimates. The major limitation of this study was that the BQMC method discussed uses a simple Gaussian kernel mean, with no hyper parameter tuning. State of the art results for a random utility model provided by Briol et al. (2019) case study #3 were obtained with a more complex kernel, which utilises prior knowledge about the covariance. Furthermore, S_i itself is not difficult to evaluate, so drawing many samples is not computationally expensive and putting a prior on S_i might over complicate calculations and unnecessarily simplify S_i. Moreover, every single choice probability is estimated, instead of a joint distribution. As such, single errors in choice probabilities accumulate. It would be interesting to see how the method in Xi et al. (2018), which focuses on estimating multiple integrals instead of a single integral and thus might be more suited to the mixed logit model, would perform.

A benefit of employing any Bayesian method is that it allows for statistical inference of the estimation result. However, the Bayesian cubature method employed in this paper estimates every single choice probability separately, instead of $L[\theta \mid X]$ itself. Effort was made to obtain this probability distributions from the separate choice probabilities, but the resulting value proved to be impossible to analyze, or because of the *log*. This resulted in not being able to analyze the estimates given by BQMC.

Does this mean that there is no place for BQMC when estimating the mixed logit model? Not necessarily. Bayesian cubature is a broad method, and the Bayesian cubature method used in this paper is the most simple variant. Bayesian cubature with a kernel more suited for the mixed logit would probably increase the estimation results significantly, and more resemble the results obtained by Briol et al. (2019) in case study #3. But, for the BQMC method used in this paper, estimation results were not competitive with both Monte Carlo methods.

One can raise the question why anyone would even consider applying probabilistic integration and a Bayesian cubature method to a mixed logit model. The argument that Monte Carlo is unsound is more fundamental in nature, but might not gain practical footing. Especially when easier Monte Carlo based methods with quasi-random obtain accurate results efficiently. Moreover, the different quasi-random sequences mentioned in 2.1 perform even better than the Halton sequence used in this paper. The state of the art results obtained by these methods, are hard to approach for a standard mixed logit model. But how would these methods perform when the distribution from which the coefficients are drawn, is not easy to compute. In this case, drawing 50 (computation time of 4.7 seconds)or even 25 draws might be computationally too expensive. In such a case, Bayesian cubature for multiple integrals with a kernel more suited for the mixed logit might outperform the Monte Carlo methods, as it theoretically needs much less draws to obtain sufficient estimates. Perhaps such a Bayesian cubature method can even compete with Monte Carlo methods for a standard mixed logit. This would be a fruitful area for further work.

6 Conclusion

This paper investigated whether probabilistic integration can be applied to the mixed logit model, and how it stacks up against traditional estimation methods. These traditional methods are standard Monte Carlo with pseudo-random draws and Quasi-Monte Carlo with quasi-random Halton sequences. First, a more fundamental argument was made about the elementary limitations of Monte Carlo methods, after which, the practical results are discussed. Of these well known methods, QMC with Halton sequences proves to be the best estimation method. Not only does it need fewer points to obtain more accurate results, computationally wise it is also more efficient. The probabilistic integration method used in this paper is a Bayesian cubature method with

a Gaussian kernel, which is also known as the Bayesian-Hermite quadrature method, but instead of states drawn using pseudo-random sequences, quasi-random Halton sequences are used. This method is also known as Bayesian Quasi Monte Carlo with a Gaussian kernel.

For BQMC, a downward trend for adding draws is observed. Nevertheless, the estimation results can not compete with the results obtained by the traditional estimation methods. Additionally, the added computational burden of adding draws to decrease the estimation error is too large. Does this mean there is no place for Bayesian cubature for estimating the mixed logit model? At least not for the mixed logit model described in this paper. However, for different kind of mixed logit models, more advanced Bayesian cubature might yet prove to be competitive with Monte Carlo methods.

A Code overview

To give insight in how the results are obtained, and to increase reproducibility, a quick over view of the Python code used is given. The github repository can be found by following this link: https://github.com/njajanssen/Thesis.

- To back-end of the code can be found in the mmnl.py file. This contains the main class which is used to perform all estimations. For each estimation method, the MMNL class contains functions to perform the necessary calculations. To start the estimation process, provide the constructor with a set of explanatory, dependent variables, amount of draws, amount of choice specific variables, and the method used. To start the optimization, simply call the solver method.
- The qmc.py file contains the QMC class which generates the necessary Halton draws.
- To calculate the results, the result_calc jupyter notebook is used. Simply load the dgp, set the amount of draws and estimation method, and which Y's from the dgp have to be used by stating a start and end value.
- In order to analyse the results, the result_analyse notebook is used. This contains functions to load the obtained results, and calculate the performance measures.
- The results folders for each method containing all obtained results.
- In the data folder two more notebooks can be found. The dgp notebook, as the name suggest, creates the DGP. The data notebook was used to load the raw data obtained through R and pickle the data as a NumPy array. The folder also contains all data used.

References

Bain, L. and Engelhardt, M. (1992). *Introduction to Probability and Mathematical Statistics*. Brooks/Cole Cengage Learning.

Berkson, J. (1944). Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39(227):357–365.

Bhat, C. R. (2001). Quasi-random maximum simulated likelihood estimation of the mixed multinomial logit model. *Transportation Research Part B: Methodological*, 35(7):677–693.

Bhat, C. R. (2003). Simulation estimation of mixed discrete choice models using randomized and scrambled halton sequences. *Transportation Research Part B: Methodological*, 37(9):837–855.

Briol, F.-X., Oates, C. J., Girolami, M., Osborne, M. A., and Sejdinovic, D. (2019). Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):1–22.

Diaconis, P. (1988). Bayesian numerical analysis. *Statistical decision theory and related topics IV*, 1:163–175.

Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. In *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, volume 471. The Royal Society Publishing.

Jain, D. C., Vilcassim, N. J., and Chintagunta, P. K. (1994). A random-coefficients logit brand-choice model applied to panel data. *Journal of Business Economic Statistics*, 12(3):317–328.

Makridakis, S. (1993). Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*, 9(4):527–529.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behaviour. In *Frontiers in Econometrics*, pages 105–142. Academic Press New York.

Mcfadden, D. and Train, K. (2000). Mixed mnl models for discrete response. *Journal of Applied Econometrics*, 15.

O'Hagan, A. (1987). Monte carlo is fundamentally unsound. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 36(2/3):247–249.

O'Hagan, A. (1991). Bayes–hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260.

Poincaré, H. (1896). *Calcul des probabilitées*. Georges, Carré, Paris.

Rasmussen, C. E. and Ghahramani, Z. (2003). Bayesian monte carlo. In *Advances in Neural Information Processing Systems 15*, pages 505–512. MIT Press.

Sivakumar, A., Bhat, C., and Ökten, G. (2005). Simulation estimation of mixed discrete choice models with the use of randomized quasi-monte carlo sequences: A comparative study. *Transportation Research Record*, 1921:112–122.

Train, K. E. (2009). *Discrete choice methods with simulation*. Cambridge university press.

Williams, C. K. and Rasmussen, C. E. (1996). Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520.

Xi, X., Briol, F.-X., and Girolami, M. (2018). Bayesian quadrature for multiple related integrals. In *Proceedings of the 35th International Conference on Machine Learn-*

ing, volume 80 of *Proceedings of Machine Learning Research*, pages 5373–5382, Stockholmsmässan, Stockholm Sweden. PMLR.