ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Bachelor Thesis International Econometrics and Operational Research

Monte Carlo estimation of the mixed logit model using low-discrepancy sequences

Name Student: Tim de Jonge van Ellemeet
Student ID number: 425288

Supervisor: A. Castelein
Second assessor: C.D. van Oosterom

Date final version: July 7, 2019

Abstract

Monte Carlo estimation evaluates a function at different points to compute an approximated average. Traditionally this is done at random points. Quasi-random sequences are strategically constructed deterministic points that aim to be more efficient than random points. The golden ratio has unique properties to construct low-discrepancy sequences that are compared to traditional quasi-random methods. The methods are compared by estimating the mixed multinomial logit model for panel data.

Contents

1 Introduction

In probability theory, a statistical model is specified and tested. The correctly specified model gives insights in the dynamics of the data and allows the researcher to classify new data or predict future observations. The parameters of a specification are often estimated using the maximization of the log-likelihood, the maximum likelihood method.

Some specifications, however, contain elements which cannot be computed directly or deduced analytically. Thus, approximation methods are needed to estimate the value of the integral. Monte Carlo simulation evaluates the integral at many different points and computes the average value. This method converges to the true value of the integral.

The question arises at which points to evaluate the integrand. Traditionally, this is done at random points, knows as the pseudo-random Monte Carlo (PMC) method. Quasi-random sequences are deterministic sequences of "strategic" points that aim to provide a better coverage of the dimensions and thus, provide more accurate estimation while evaluating the integrand at fewer points.

The quasi-random sequences should have low-discrepancy, meaning that they do not clutter and are evenly spaced. The Halton and Sobol sequences are compared, and a new sequence based on the golden ratio is introduced. The golden ratio has a unique property that it is the "most irrational" number and therefore a simple and effective way to construct well spaced sequences.

The Mixed Multinational logit model is a generalization of the multinomial logit model that allows for coefficients to be randomly distributed. The individual choice probabilities have to be estimated using a Monte Carlo method, to obtain the simulated log-likelihood needed for estimation.

The primary research question is:

How does the golden ratio sequence compare to pseudo-random and other quasi-random sequences at the estimation of the mixed multinomial logit model?

The secondary research questions are:

How can Monte Carlo methods estimate the mixed multinomial model?

How to construct a low-discrepancy sequence using the golden ratio, and what are the underlying reasons why it has such low discrepancy?

Does the Monte Carlo method have to use random points and what are the advantages of low-discrepancy sequences?

A simulation study is done to be able to compare methods in a controlled environment. The data is based on field data on Catsup but the choices made by consumers are determined by a controlled DGP. 1000 pseudo-random datasets are generated to evaluate the different estimation methods.

Monte Carlo methods are increasingly accurate as the number of points at which an integrand is evaluated increases. However, depending on the application, it may be infeasible to evaluate at many points as computation time increases and the estimation becomes impractical. Thus, the motivation for finding smarter sequences is being able to achieve high accuracy estimations while evaluating integrands at less points to be more efficient.

The golden ratio sequence is a low-discrepancy sequence with promising properties. However, for a low amount of draws, it is outperformed by the pseudo-random sequence. As the draws increase, the benefits of the golden ratio sequence become more apparent. The sequence does not outperform the traditional quasi-random sequence but does spark interest for further investigation due to the low-discrepancy behaviour and the underlying reasons why it does not perform as well.

2 Literature

The mixed multinomial logit model and the estimation using quasi-random methods for the simulated log-likelihood has been first been described in Train (1999) and Bhat (2001). The mixed multinomial logit model and the dataset of interest are described in Jain et al. (1994).

The quasi-random methods rely on low-discrepancy sequences. These sequences have

been described for a long time. The Halton sequence is a generalization of the van der Corput sequence that was published in 1923. The Halton sequence was proposed by Halton (1960). The Sobol sequence is a sequence similar to the Halton sequence and exactly equal for certain parameters, was introduced by Sobol' (1967).

This research proposes the use of the golden ratio to construct a low-discrepancy sequence. Interest in the golden ratio dates back until ancient Greece. An overview of the multifold applications of the golden ratio is given in Akhtaruzzaman (2011). There are many ways to generalize the golden ratio, which is necessary for multi-dimensional sequences. Schretter et al. (2012) proposed a method to construct a low-discrepancy sequence using the golden ratio properties. Roberts (2018) gives an introduction to some low-discrepancy sequences and proposes another method to construct a sequence based on the golden ratio, which outperforms the traditional sequences.

A generalization of many low-discrepancy sequences as (t, m, s)-nets is first given by Niederreiter (2005). Applications of these nets and lattice points in the MMNL model for panel data is introduced in Sandor (2013).

Further improving low-discrepancy sequences by scrambling the deterministic sequence is demonstrated in Tan and Boyle (2019). Combining scrambled data with (t, m, s)-nets is introduced in Sivakumar and Bhat (2019).

Low-discrepancy sequences are know to perform well in lower dimension but may become problematic in higher dimensions. In Wang and Sloan (2008) consquences of high dimensional low-discrepancy sequences are presented.

3 Data

The mixed multinomial logit model allows for individual preferences to be drawn from a distribution rather than be fixed coefficients. A common application is the use of panel data to analyze the sensitivity of consumers based on variables such as whether a product is being advertised, or fluctuations in price. Panel data consists of multiple observations. These observations consist of the relevant data on the choices available at time of purchase, as well as the choice made by the household.

To evaluate the accuracy of the estimation methods used, a controlled experiment is necessary. We will therefore define a Data Generating Process (DGP) and compare this to the estimated parameters of the different estimation methods. In order to maintain the applicability of these methods, actual field data is used as independent variables and plausible parameters for the DGP. The experimental design and DGP will be further discussed in section 5.

The dataset of interest is the panel data from a collection of households in Springfield, Missouri and is collected by A.C. Nielsen. It was first introduced by Jain et al. (1994). For this research, the data was retrieved from the Ecdat package and extracted as CSV.

The panel data concerns the purchase of Catsup. Each observation the consumer makes a choice between four products: heinz41, heinz32, heinz28 and hunts32. At the time of purchase the following information is collected:

- Household ID
- Brands on display
- Brands featured
- Brand Prices

The dataset contains information of 300 households that made a total of 2,798 purchases.

4 Methodology

Logistic estimation is often done using the method of maximum likelihood. First the model of interest is specified, which is used to define the log-likelihood function. This function takes the assumed parameters as variables, and calculates the log-likelihood of the observed data under the current set of parameters. The aim is to maximize this log-likelihood, resulting in the optimal set of parameters for the model of interest to explain the observed data. A good estimation has a set of parameters close to the parameters of the DGP.

The calculation of the log-likelihood has be estimated using a sequence of standard normal draws, as will be discuss in section 4.1. There are pseudo-random and quasi-random methods to determine these draws. Increasing the number of draws improves accuracy but decreases performance. The aim of this paper is to construct draws that give high accuracy without requiring a large amount of draws so that the estimation of the MMNL model becomes practical.

The maximization process is an optimization problem. The field of unconstrained nonlinear optimization is complex and many algorithms are available. The estimation is done using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm. It is a quesi-newton hill-climbing technique. BFGS does not guarantee performance to a global minimum, yet is an often used method due to the relative reliability and reasonable computational load.

4.1 Mixed multinomial logit model

The mixed multinomial logit model is a generalization of the multinomial logit model. It allows for a parameter β to be drawn from a distribution. Thus, one can interpret a consumers preference as a distribution rather than a fixed number. A parameter β is drawn out of a density function $f(\beta|\theta)$ where θ represents the parameters specifying the distribution of $f(\beta|\theta)$. In this experiment it is assumed $\beta \sim N(\mu, \sigma)$, although any distribtution is possible. Another frequently used distribution, typically for one sided data, is the log-normal distribution. The unconditional probability P_{qtb} that household q chooses brand b at time t is as follows:

$$\begin{aligned} L_{qtb}(\beta_q) &= \frac{\epsilon^{\beta_q' x_{qtb}}}{\sum_j \epsilon^{\beta_q' x_{qtj}}}, \\ P_{qtb}(\theta) &= \int_{-\infty}^{+\infty} L_{qtb}(\beta) f(\beta|\theta) d(\beta), \end{aligned} \tag{1}$$

where β_q is a vector of the coefficients of household q and x_{qtb} a vector of the independent variables for household q and brand b at time t. In panel data an individual makes a sequences of choices. In the MMNL model the random parameters are drawn per individual. Therefore in panel data, the probability on interest is $P_{q\boldsymbol{t}}$ where $\boldsymbol{t}$ is the observed sequence for an individual q. This is denoted as follows:

$$\begin{aligned} L_{q\boldsymbol{t}}(\beta_q) &= \prod_t \frac{\epsilon^{\beta_q' x_{qtb_t}}}{\sum_j \epsilon^{\beta_q' x_{qtj}}}, \\ P_{q\boldsymbol{t}}(\theta) &= \int_{-\infty}^{+\infty} L_{q\boldsymbol{t}}(\beta) f(\beta|\theta) d(\beta), \end{aligned} \tag{2}$$

where x_{qtb_t} is a vector of the independent variables for household q for the brand b that is chosen at time t.

The MMNL model can be represented in the form of random coefficients. For each choice, the utility is defined, and the option with the highest utility is chosen. This allows for the simple deconstruction of the probability of a choice. The utility of household q selecting brand b at time t is:

$$U_{qtb} = \beta_q' x_{qtb} + \epsilon_{qtb} \tag{3}$$

This structure separates the utility into two elements: the non-identical and non-independent across alternatives part $\beta_q' x_{qtb}$, and the independent and identically type-1 extreme-value distributed part ϵ_{qtb}.

This can be rewritten with the vectors in scalar form as:

$$U_{qtb} = \sum_{k=1}^{K} \beta_{qk} x_{qtbk} + \epsilon_{qtb}, \tag{4}$$

where k indicates the independent variables *display*, *featured* and *price*. As $\beta_{qk} \sim N(\mu_{qk}, \sigma_{qk})$ this can be decomposed as:

$$\beta_{qk} = \mu_{qk} + \sigma_{qk} s_{qk}, \tag{5}$$

where s_{qk} is a standard normal variate. The assumption of normally distributed coefficients allows for this decomposition and the estimation of μ_{qk} and σ_{qk} using equation 5.

The MMNL model does not require all coefficients to be randomly distributed. In the specification for the problem of panel data for brands of Catsup, we define the fixed coefficient α_b, which represents the base preference for a brand b and is fixed across households. This coefficient is added to clean the randomly distributed coefficients of the effect of brand

preference. Arguably, one can make this coefficient varying across households to model the preference of a specific household. However this would add 300 parameters to estimate and does not add to the comparison of estimation methods. As this is a controlled experiment as described in section 5, this restriction should not interfere with the results of this research. Furthermore α_4, the base preference for Hunts32, is set to 0 for the identification of all other parameters.

The utility function for this problem is given as:

$$\begin{aligned} U_{qtb} &= \alpha_b + \sum_{k=1}^{K} (\mu_{qk} + \sigma_{qk} s_{qk}) x_{qtbk} + \epsilon_{qtb} \\ &= V_{qtb} + \sum_{k=1}^{K} \sigma_{qk} s_{qk} x_{qtbk} + \epsilon_{qtb}, \end{aligned} \tag{6}$$

where $V_{qtb} = \alpha_b + \sum_{k=1}^{K} \mu_{qk} x_{qtbk}$

Lastly the indicator function $y_{q\boldsymbol{t}}$ is defined as:

$$y_{q\boldsymbol{t}} = \begin{cases} 1 & \text{if sequence } \boldsymbol{t} \text{ occurs for household } q \\ 0 & \text{otherwise} \end{cases}$$

The log-likelihood gives the likelihood of the observations under the assumed model and set of values for the parameters. The above mentioned equations are used to arrive at:

$$
\begin{aligned}
\mathcal{L} &= \sum_{q}\sum_{\boldsymbol{t}} y_{q\boldsymbol{t}} \log P_{q\boldsymbol{t}} \\
&= \sum_{q}\sum_{\boldsymbol{t}} y_{q\boldsymbol{t}} \log \left(\int_{-\infty}^{+\infty} L_{q\boldsymbol{t}}(\beta) f(\beta|\theta) d(\beta) \right) \\
&= \sum_{q}\sum_{\boldsymbol{t}} y_{q\boldsymbol{t}} \log \left(\prod_{\boldsymbol{t}} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \frac{e^{V_{qtb_t} + \sum_{k=1}^{K} \sigma_{qk} s_{qk} x_{qtb_t k}}}{\sum_j e^{V_{qtj} + \sum_{k=1}^{K} \sigma_{qk} s_{qk} x_{qtjk}}} \quad \mathrm{d}\Phi(s_q 1)\mathrm{d}\Phi(s_q 2)\cdots\mathrm{d}\Phi(s_q k) \right)
\end{aligned}
\tag{7}
$$

As the log-likelihood contains unbounded integrals estimation methods are necessary to compute the likelihood. The next section covers the different methods for approximating the log-likelihood.

Computationally, this implementation of the log-likelihood might lead to error due to small numbers and precision loss in computers. The likelihood function can be rewritten to prevent this. The transformation is as follows:

$$
\prod_t Pr[y_{it} = j]^{I[y_{it}=j]} = \epsilon^{\sum_t I[y_{it}=j] \log Pr[y_{it}=j]}
$$

The used optimization algorithm BFGS, is an unconstrained algorithm. As standard deviations σ_{qk} are non-negative, a transformation is done to bound the values without the need of a more complex constrained optimization algorithm. The transformation is as follows:

$$
\sigma_{qk} = \epsilon^{\gamma_{qk}}, \tag{8}
$$

where γ_{qk} is the parameter estimated by the algorithm. In the log-likelihood function standard deviations are replaced by the equation above and the original parameters are retrieved in the same manner.

4.2 Estimation using low-discrepancy sequences

The estimation of the log-likelihood function is done by approximating the integrand using Monte Carlo simulation. By evaluating the integrand at R different points and taking the average an approximation is obtained. Thus, one needs to generate a sequence of R standard variates in K dimensions, denoted as s_{qkr}. The approximation of $P_{q\boldsymbol{t}}$ is as follows:

$$
\check{P}_{q\boldsymbol{t}} = \frac{1}{R} \prod_t \frac{\epsilon^{V_{qtb_t} + \sum_{k=1}^{K} s_{qk} x_{qtb_t k}}}{\sum_j \epsilon^{V_{qtj} + \sum_{k=1}^{K} s_{qk} x_{qtjk}}} \tag{9}
$$

$$
\begin{aligned}
\check{\mathcal{L}} &= \sum_{q}\sum_{\boldsymbol{t}} y_{q\boldsymbol{t}} \log \check{P}_{q\boldsymbol{t}} \\
&= \sum_{q}\sum_{\boldsymbol{t}} y_{q\boldsymbol{t}} \log \left(\frac{1}{R} \prod_t \frac{\epsilon^{V_{qtb_t} + \sum_{k=1}^{K} s_{qk} x_{qtb_t k}}}{\sum_j \epsilon^{V_{qtj} + \sum_{k=1}^{K} s_{qk} x_{qtjk}}} \right)
\end{aligned}
\tag{10}
$$

Equation 10 is the simulated log-likelihood. The matrix s can be constructed in many ways. The aim is to implement a that required little draws R for an accurate approximation of the coefficients α_b, μ_{qk} and σ_{qk}. R approximately linearly affects the computation time of the simulated log-likelihood, but increasing R improves the accuracy and thus the ability for an optimizer to estimate good parameters.

When performing a minimization, it is important to use the same matrix s for all iterations of the minimization algorithm.

The pseudo-random Monte Carlo (PMC) method is the simplest way of constructing matrix s, by drawing each value from the standard normal distribution randomly, this will be elaborated on in section 4.2.1 and is the baseline of estimation performance.

The quasi-random Monte Carlo (QMC) method uses quasi-random sequences with the aim to construct a deterministic sequence of cleverly crafted points, rather than random points. With a good sequence QMC, needs less points R to achieve the same accuracy as the PMC method, and thus, yields computational gains. This sequence should be a low-discrepancy sequence. Discrepancy refers to the clustering of points. A low-discrepancy sequence has little clustering and aims to be distribute over the dimensional space uniformly,

which is also know as "Blue Noise." The different types of sequences can be categorized by the method of construction of their parameters as follows:

- Irrational fractions: Kronecker, Richtmyer, Ramshaw
- (Co)prime numbers: Van der Corput, Halton, Faure
- Irreducible Polynomials : Niederreiter
- Primitive polynomials: Sobol

The Halton and Sobol sequences will be compared to the pseudo-random sequence and a new sequence based on the golden ratio introduced.

Quasi-random sequences are typically uniformly distributed. As the desired matrix s contains standard normal variates, the uniform sequences need to be transformed. This is done using the inverse CDF and is denoted a follows:

$$s_{qk} = \Phi^{-1}(q_{qk}), \tag{11}$$

where q_{qk} is the kth dimension of the qth point in a quasi-random sequence and a scalar between 0 and 1. This transformation is done for each element to obtain the desired matrix s of standard normal variates.

4.2.1 Pseudo-random Monte Carlo method

The random coefficients are assumed to be normally distributed but the mean and variance are to be estimated. By rewriting the coefficients as shown in equation 6, the random behaviour is separated from the shape parameters that are to be estimated. Thus, the most straightforward way of approximating the integrand is by drawing numbers from the standard normal distribution. The estimator is unbiased of the individual choice probabilities, the variance decreases as R increases.

As computers cannot generate truly random numbers but only sequences that are seemingly random based on the properties of a given sequence, random numbers from a computer science perspective are "pseudo-random" and reproducible using a user-defined starting seed.

4.2.2 Halton sequence

The Halton sequence introduced by Halton (1960) is a generalized version of the one-dimensional low-discrepancy sequence introduced by van der Corput in 1935. A one-dimensional sequences is constructed by first picking a prime number r which is the base. The gth number in the sequence is defined as:

$$\begin{aligned} g &= \sum_{l=0}^{K} b_l r^l, \\ 0 &\leqslant b_l \leqslant r-1, r^L \leqslant g, \leqslant r^{L+1} \\ \varphi_r(g) &= \sum_{l=0}^{K} b_l r^{-l-1} \end{aligned} \tag{12}$$

A Halton sequence is uniformly and efficiently distributed on the (0,1) domain for any prime , and induces negative correlation over the observations. Extending the Halton sequence to K dimensions is done by simply generating K sequences with different coprimes r. In practice, usually the first K primes are chosen. In the case of the Catsup problem with $K = 3$, the Halton sequence is defined as:

$$\Psi_g = (\varphi_2(g), \varphi_3, (g), \varphi_5(g)) \tag{13}$$

The individual choice probability integrand may be sensitive to the starting point of the Halton sequence, as described in Morokoff and Caflisch (1995). Therefore the first 10 items of the sequence will be discarded. The generated sequence will thus have shape:

$$(Q \cdot R) + 10 \times K,$$

where G = $(Q \cdot R) + 10$. After discarding the first 10 items, the matrix is partitioned in Q sub-matrices to obtain the $R \times K$ draws for each household.

4.2.3 Sobol sequence

The Sobol sequence was first introduced by Sobol' (1967). It is an S-dimensional sequence that aims to divide each dimension in increasingly smaller uniform partitions of the unit interval, and thus, models the uniform distribution. This is done using a complex algorithm with primitive polynomials and XOR operations on base 2. A Sobol sequence can be initialized with many different initial values,

which influence the efficiency. The Sobol sequence can be written in a generalized form as:

$$\begin{aligned} t_s &= \sum_{i=1}^{s} \deg p_i - 1 \text{ where,} \\ p_1 &= x \in \mathbb{Z}_2[x], \\ p_{i+1} &= \ i\text{th polynominal over } \mathbb{Z}_2 \\ &\text{ ordered by degree} \end{aligned} \tag{14}$$

There are many packages available to generate the Sobol sequences. They typically include best practice intial values for the sequences up to a certain amount of dimensions. For this research, the Python package sobol_seq is used which is made available at the PyPI repository and includes initializations for up to 40 dimensions.

4.2.4 Golden ratio sequence

Sunflowers grow seeds in a pattern of spirals. This is done by rotating a certain amount after each seed has grown. They aim to optimize the space available by evenly covering it. If a rotation of a fraction is done, say $\frac{1}{n}$, the coverage is limited to n spokes as depicted below:

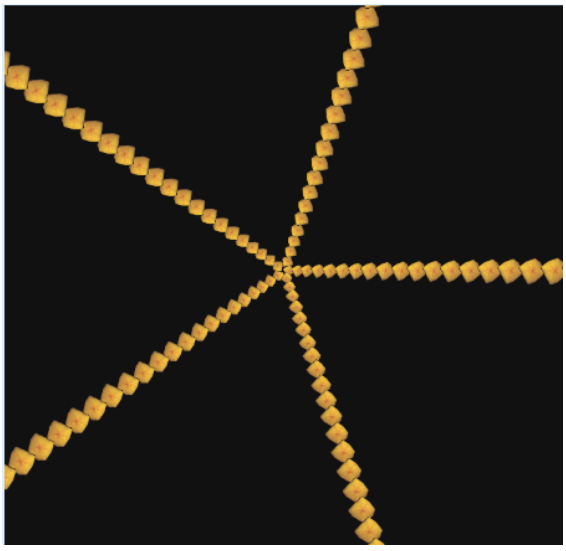

Figure 1: Rotional points with $\frac{1}{5}$

Clearly, a spoke pattern develops as all the steps of a fraction sum up to 1. Irrational numbers cannot be split up in perfect fractions as the decimals are infinitely long. The expected behaviour is thus that no straight spokes should emerge as the steps do not add up exactly to 1. One of the most well known irrational numbers pi or 3.14... gives the following pattern:

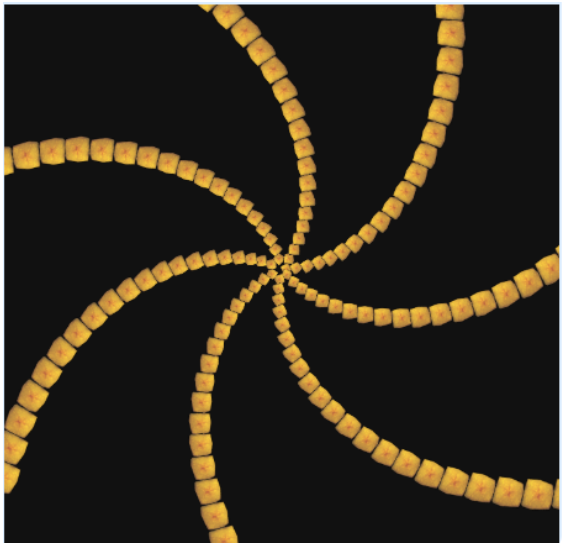

Figure 2: Rotional points with π

The points have a spiral shape as they are no longer straight extended spokes. The points are slightly more evenly distributed but still have clear spokes. This raises the question of which irrational number performs optimally at covering the circle. The golden ratio is observed in many applications in both art and nature as described in Akhtaruzzaman (2011). The golden ratio is an irrational number and the solution to equation:

$$x^2 = x + 1, \tag{15}$$

which has the value:

$$\varphi = \frac{1 + \sqrt{5}}{2} = 1.618...$$

The golden ratio is the most irrational number in the sense that is as far is it can be from any fraction. For example π is very close to $3 + \frac{1}{7} = 3.1428....$ and thus has 7 spokes. The golden ratio has the special property that it can be written as:

$$\varphi = 1 + \frac{1}{\varphi} \tag{16}$$

This leads to the following infinite expansion:

$$\varphi = 1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cdots}}} \tag{17}$$

Therefore the golden ratio is the following pattern:

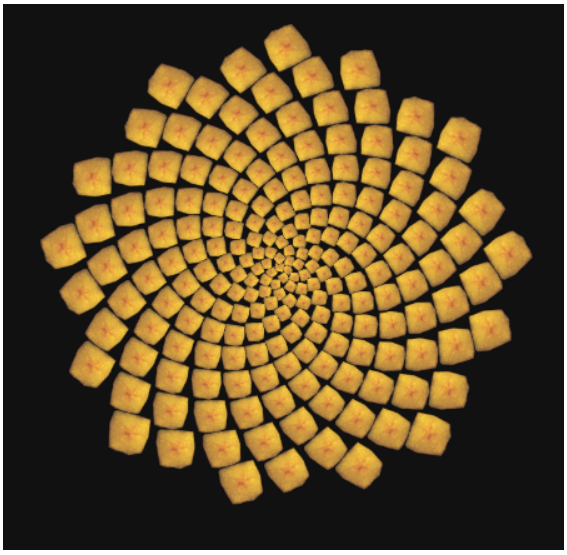

Figure 3: Rotional points with φ

This is the same pattern as seen in sunflowers and optimizes the coverage of the space available as there are no gaps between points.

This result leads to the question if the golden ratio can be used to construct a low-discrepancy sequence. Roberts (2018) proposes an adaption of the golden ratio to construct a low-discrepancy sequence. Since multi-dimensional sequences are required, a generalized form of the golden ratio is needed. There are many ways to achieve this. The following definition is used: φ^d is the unique postive root to $x^{d+1} = x + 1$. φ^2 is also known as the plastic number, named Dutch architect and monk Hans van der Laan in 1928.

The golden ratio sequence is denoted as $R_d(\varphi_d)$ and has the following specification:

$$\begin{aligned} &\boldsymbol{t}_n = s + n\boldsymbol{\alpha}, \;\; n = 1, 2, 3, ... \\ &\text{where } \boldsymbol{\alpha} = \big(\frac{1}{\varphi_d}, \frac{1}{\varphi_d^2} \cdots \frac{1}{\varphi_d^d}\big), \end{aligned} \tag{18}$$

where s is a seed or starting point. This is typically set to 0. however Roberts (2018) found optimal results for seed 0.5.

4.2.5 Illustration of sequences

The different methods, illustrated as two-dimensional , are depicted below:

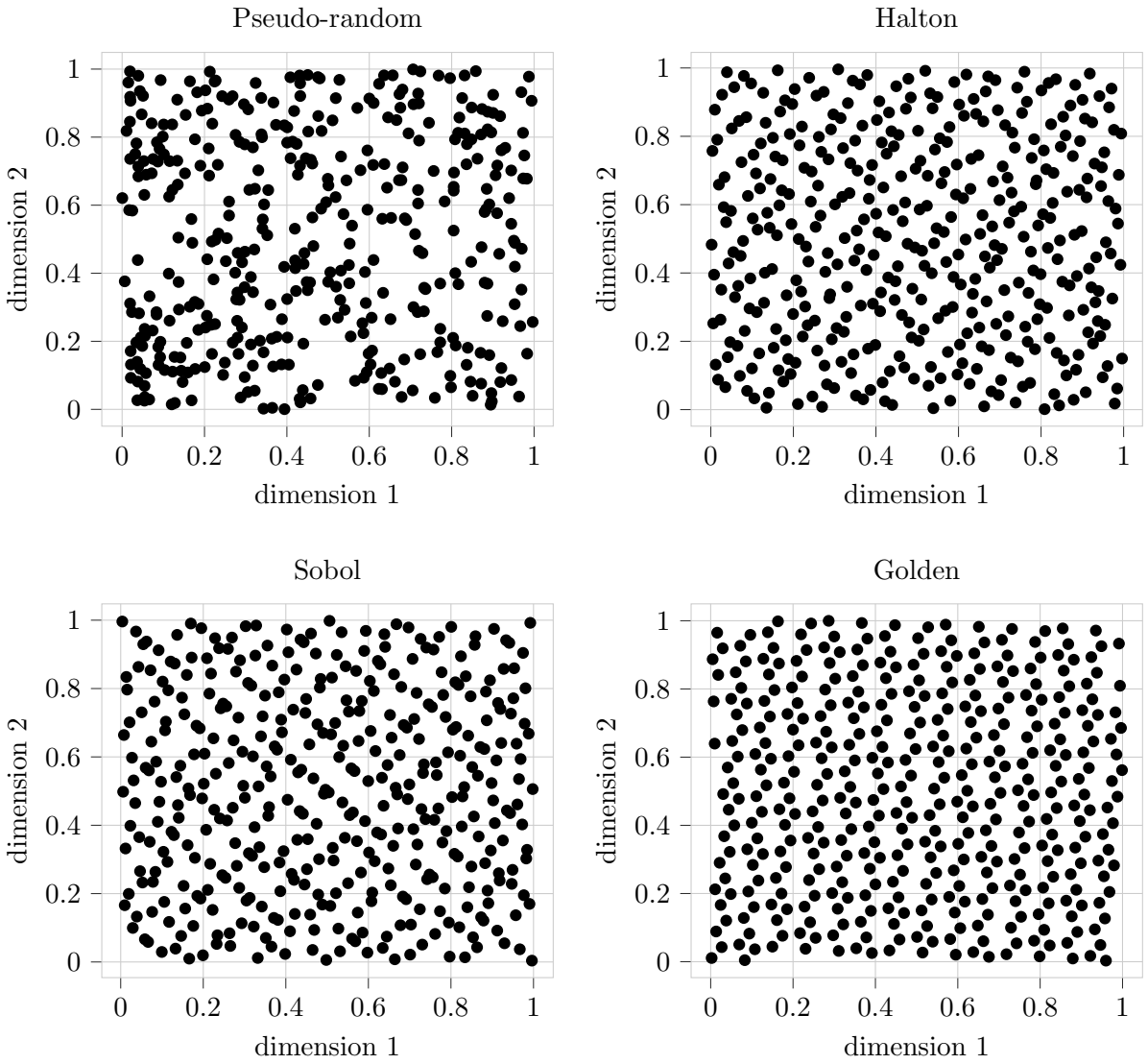

Figure 4: Comparison of 400 points in low-discrepancy sequences

The pseudo-random sequence has no clear patterns. The points are not evenly spaced and a lot of clustering occurs. This sequence is not a low-discrepancy sequence.

The Halton sequence is more evenly spaced but does have clusters of points that are not spaced.

The Sobol sequence is also more evenly spaced than the pseudo-random sequence. However, it does show some string patterns which suggest some correlation and thus, less random behaviour. One should note the straight diagonal line going through the Sobol graph. Such patterns suggest lesser results than of the Halton sequence.

The golden ratio based sequence is the most equally spaced, and well distributed over the 2-dimensional space. However their appear to be vertical line patterns in the graph.

5 Simulation study

To evaluate capabilities of estimation methods one needs to control the experiment by knowing the true values of the estimated parameters. As explained in section 3, field data is

used as basis, with the choice made for an observation being determined by the controlled DGP. This results in what is know as "pseudo-simulated" data.

As this is a simulation study including randomness, it is important to generate multiple datasets to be able to make claims regarding the performance of the various estimation methods for all datasets. This allows to for the evaluation of the mean and standard deviation of the perfomance measures of interest. For this experiment, 1000 datasets have been generated, all of which are available in the code provided. As the number of draws for an estimation increases, estimation slows down. Therefore the amount of datasets used is a trade off between reliability and practicality. In section 4.2.5 the influence on the amount of datasets used will be elaborated on.

To create these datasets one must first define the paramaters of the DGP. The parameters are in the same range as the parameters found in exploratory analysis on the original data, in order to accurately simulate field data. The parameters are set as follows:

	α	
Heinz41	0.8	
Heinz32	0.2	
Heinz28	-0.5	
Hunts32	0	
	μ	σ
Display	0.9	0.1
Featured	1.1	0.5
Price	-0.5	0.25

Table 1: Parameters for DGP

With these parameters, a 3-dimensional array can be constructed containing coefficients for each household for each dataset. In this simulation, the Python package Numpy is used to generate an array of standard normally distributed draws in the desired shape using seed 1234. Next, the coefficient matrix is obtained for each draw computing the appropriate coefficient:

$$\beta_{qkr} = \mu_k + \sigma_k s_{qkr}, \tag{19}$$

Please note that in this equation, r represents the number of the dataset, and is not be confused when using r to represent the number of the draw in estimation.

To simulate the choices made in the datasets, the utility is computed for each of the four choices, and the brand with the highest utility is chosen by the household. The utility of a brand b at time t for household q in dataset r is:

$$U_{qtb} = \alpha_b + \sum_{k=1}^{3} \beta_{qkr} x_{qtbk} + \epsilon_{qtr} \tag{20}$$

The generated independently and identically distributed standard type-1 extreme value error term ϵ_{qtr}, is included to account for the unexplained part of a choice. This allows for the complete the specification of the unconditional choice probabilities given by the mixed logit model. The error term is required to make the utility latent variables that make choices following a logistic distribution. Without this error term, the model will not only be incorrect but estimation will fail as it allows the estimated parameters to explode.

This procedure is used to create 1000 unique datasets with the same underlying DGP. Thus completing a controlled environment to evaluate the performance of the estimation methods.

6 Results

6.1 Background

The estimation process has been implemented using Python. The log-likelihood is evaluated using Numba, a just-in-time compiler for Python that works with a small subset of Python and Numpy. At the first log-likelihood evaluation, Numba has a similar performance to plain python, but after the first iteration the function is compiled to machine code. This results in a highly optimized log-likelihood function and evaluation time is around 100 times faster. As the log-likelihood is evaluated for each iteration of an optimization algorithm this is a very useful performance gain.

For the optimization, the SciPy Minimize package is used, with the log-likelihood function provided and as stopping requirement that

the gradient norm is smaller than 10^{-5}. The initial parameters are random draws from the uniform distribution.

The minimization algorithm is single threaded. As each model is to be estimated 1000 times for multiple methods and amounts of draws, the process has been parallelized using the Joblib package. The class *Mixedlogit* is initialized with all datasets. It implements the four draw methods to perform the estimation on all datasets. If parallelization is enabled, it spans estimations across all available cores.

The results and estimation steps are all stored, and can be analyzed using the class *Analyzer*. As the true parameters are known one can track the evaluation of the performance measures for each iteration of the optimization.

All computing has been done on the AWS Sagemaker platform. The instance type used is ml.c5d.18xlarge. This instances uses multiple Intel Xeon Platinum 8000 series (Skylake-SP) processors with a clockspeed of 3.4GHz. The system has 72 virtual processors and thus is able to perform estimations on 72 datasets simultaneously. The system has 144 GB of RAM storage.

All four methods have been estimated for 5,10,25,50,75 and 100 draws for 1000 datasets. The Halton and pseudo-random sequences have also been estimated at 250 draws on 200 datasets, as it becomes infeasible to estimate all 1000 datasets. The pseudo-random method was also estimated on 200 datasets using 500, 1000 and 2000 draws.

Between 1% and 2% of the estimations failed due to various errors, mainly division by zero failures. This could occur when the optimization algorithm is too far from the optimal solution. In the results table, the amount of successful estimations is included.

6.2 Performance measures and results

To evaluate the performance of the PMC and QMC methods, their ability to recover the true parameters of the DGP are of interest. As this is a controlled experiment and the true parameters are known, the performance is measured using the root mean square error (RMSE). The RMSE measures the proximity of all parameters to the true values. The optimal value for RMSE is 0, in case the estimated parameters are exactly the true parameters. The time to estimate a dataset has been included in the section below RMSE.

For both the RMSE and time, the mean and standard deviation of the estimation of a dataset are given. The standard deviation depicts how stable the performance of a method is. Lastly, the number of successful estimations is included. The results can be found in table 2 below:

Table 2: Results of estimations of the MMNL model using PM and QMC methods

		draws	5	10	25	50	75	100	250	500	1000	2000
		drawtype										
rmse	mean	golden	0.163	0.146	0.139	0.126	0.121	0.122				
		halton	**0.142**	**0.129**	**0.123**	**0.121**	**0.120**	0.121	0.120			
		pseudo	0.160	0.145	0.130	0.126	0.123	0.126	**0.119**	0.120	0.119	0.119
		sobol	0.148	**0.129**	0.124	**0.121**	0.121	**0.120**				
	std	golden	0.043	0.049	0.051	0.051	**0.049**	0.050				
		halton	0.048	0.051	**0.049**	**0.049**	0.050	**0.049**	**0.048**			
		pseudo	**0.042**	**0.048**	0.051	0.050	0.050	0.050	0.049	0.047	0.048	0.048
		sobol	0.046	0.050	0.050	**0.049**	**0.049**	**0.049**				
time (s)	mean	golden	26.773	49.654	176.712	334.954	534.454	682.467				
		halton	26.879	49.098	173.397	344.240	511.580	691.432	1,593.065			
		pseudo	27.731	53.415	179.051	335.515	533.602	683.663	1,783.992	3,277.670	6,762.978	13,541.047
		sobol	27.146	49.521	167.353	346.414	528.350	682.569				
	std	golden	11.386	22.059	76.491	144.567	246.125	297.558				
		halton	10.950	21.593	76.228	151.168	227.466	314.389	662.828			
		pseudo	11.190	24.308	79.247	150.114	257.782	300.388	704.846	1,261.117	2,925.191	5,592.076
		sobol	12.052	21.532	77.012	148.236	229.906	313.125				
	Estimations	golden	987	989	989	996	986	992				
		halton	981	985	983	986	985	988	200			
		pseudo	975	979	986	992	992	991	199	197	200	199
		sobol	986	983	979	989	987	986				

As the amount of draws R increases the accuracy improves for any method. This is in line with the amount of draws improving the approximation of the individual choice probabilities in the log-likelihood function. The only exception is the pseudo-random method at 100 draws. However this appears to be an outlier compared to the other results of the method, and in the range of the standard deviation thus not necessarily significant. The method is random both in draws and the steps of the optimization algorithm. Therefore this is an aberration in a stable trend.

As the DGP includes a random term, estimating the parameters perfectly is highly unlikely. There is a soft lower limit of the best achievable RMSE, which is around 0.119. As the amount of draws increases, all methods converge to this accuracy. The QMC methods converge faster than the PMC method, because of the additional draws are strategically chosen.

All methods have a very stable standard deviation. For 5 draws, the standard deviation varies between 0.042 and 0.048. For all other amounts of draws, for all methods, the standard deviation varies between 0.047 and 0.051. This indicates that the amount of draws does not affect the reliability of the method, as unexpected results are not likely.

The Sobol and Halton sequences give very similar results. They both outperform the PMC method and converge closer to the minimum with the same amount of draws. At 5 draws, Halton performs better with 0.142 compared to 0.148. For all other amounts of draws, the methods differ at most 0.001. They use the same principle to construct a low-discrepancy sequence, and thus have very similar results. The Halton sequence is more simple in construction and therefore often preferred over the Sobol method. The results supports this as they yield similar performance and initializing the Sobol method is tedious.

The golden ratio method at 5 draws has the worst performance of all methods. This is likely because 5 is not enough draws to reach the spread of points which is the value of a golden ratio sequence. As the amount of draws increases, the golden ratio method proves to be the fastest of all methods. At 50 draws it performs comparable to the PMC method and at 75 draws it is comparable to the Halton and Sobol methods. From 75 to 100 draws it follows the same performance as the other QMC methods. As it never outperforms them and performs the worst for lower number of draws, it is not a more accurate method than the Halton and Sobol sequences. However, the performance for higher amounts of draws is similar and thus further research is suggested.

The computation time is approximately linearly related to the amount of draws. The time does not vary much between methods, as the standard deviation is about 40% of the computation time. This makes the difference between computation time insignificant. Thus, all methods have comparable performance.

As the amount of draws , the performance increase of all methods becomes smaller while computational time increases linearly. A trade off is to be made between desired accuracy and computational time for all methods. The Halton method, for example, has the same performance for 25 draws as the PMC method for 75 draws. This implies a 3 times performance increase to estimate with the same accuracy. The Sobol method has an accuracy of 0.120 RMSE for 100 draws while the PMC method has 0.126. Only when increasing the number of draws to 250 and further, does the PMC method have the same performance as the QMC methods at 100 draws, illustrating the benefit of the QMC methods.

7 Conclusion

This paper describes the process of integrand estimation using the Monte Carlo method with different sequences. The application for which the comparison is made, is the MMNL model which allows for the modelling of consumer preference to be randomly distributed. This is done using a pseudo-random sequence and three quasi-random sequences. The Halton and Sobol sequences outperform the random sequence, and the Halton sequence is preferred due to its simplicity.

The golden ratio is used to make a low-discrepancy sequence. It does not perform as well as the other quasi-random sequences for lower draws but as the amount of draws in-

crease it becomes a valid method to estimate the simulated log-likelihood.

Due to the performance of QMC methods, it is concluded that a sequence need not be random, but strategically crafted points offer a faster and more efficient way to estimate integrals accurately.

8 Discussion

The addition of a brand specific term α_b was introduced as level of complexity that may have lead to less large performance differences than found in Bhat (2001). A next step could be discarding this parameter from the estimation and compare results to see the influence it has on performance.

Research has been done on the properties of randomized low-discrepancy sequences by Tan and Boyle (2019). It shows promising results in maintaining the low-discrepancy of the quasi-random sequences, while simulating randomness in what is theoretically a random sequence. As vertical patterns emerged, a further extension could be investigating scrambled golden ratio sequences.

Niederreiter (2005) describes the construction of (t, m, s)-nets, which are a generalization of many low-discrepancy sequences such as Sobol and Halton. Finetuning these nets and scrambling them has shown promising results to further increase the accuracy as shown in Sivakumar and Bhat (2019).

A Code of research

The source code for this paper is made publicly and may be retrieved from https://github.com/timellemeet/mixedlogit

References

Akhtaruzzaman, M. (2011). Geometrical substantiation of phi, the golden ratio and the baroque of nature, architecture, design and engineering. *International Journal of Arts*, 1.

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.

Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1):84–90.

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.

Morokoff, W. J. and Caflisch, R. E. (1995). Quasi-monte carlo integration. *Journal of Computational Physics*, 122(2):218 – 230.

Niederreiter, H. (2005). Constructions of (t,m,s)-nets and (t,s)-sequences. *Finite Fields and Their Applications*, 11(3):578 – 600. Ten Year Anniversary Edition!

Roberts, M. (2018). The unreasonable effectiveness of quasirandom sequences.

Sandor, Z. (2013). Monte carlo simulation in random coefficient logit models involving large sums. *Acta Universitatis Sapientiae, Economics and Business*, 1.

Schretter, C., Kobbelt, L., and Dehaye, P.-O. (2012). Golden ratio sequences for low-discrepancy sampling. *Journal of Graphics Tools*, 16(2):95–104.

Sivakumar, A. and Bhat, C. (2019). Simulation estimation of mixed discrete choice models using randomized quasi-monte carlo sequences: A comparison of alternative sequences, scrambling methods, and uniform-to-normal variate transformation techniques.

Sobol', I. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86 – 112.

Tan, K. and Boyle, P. (2019). Applications of scrambled low discrepancy sequences to exotic options.

Train, K. (1999). Halton sequence for mixed logit.

Wang, X. and Sloan, I. H. (2008). Low discrepancy sequences in high dimensions: How well are their projections distributed? *Journal of Computational and Applied Mathematics*, 213(2):366 – 386.