Efficiency of Hamiltonian Monte Carlo for Hierarchical Bayesian Mixed Logit Estimation

Daan Noordenbos

Efficiency of Hamiltonian Monte Carlo for Hierarchical Bayesian Mixed Logit Estimation

Daan Noordenbos

June 15, 2023

Abstract

This thesis investigates the efficiency of Hamiltonian Monte Carlo (HMC) compared to random walk Metropolis Hastings (RWMH) sampling in the context of hierarchical Bayesian mixed logit (HB MXL) estimation. While HMC has been widely acknowledged as superior to RWMH for general sampling, especially for high-dimensional distributions, its performance for HB MXL has not been explored extensively.

This thesis addresses this gap by comparing the RWMH estimation procedure for HB MXL against a HMC estimation procedure. Specifically, the difference in performance is examined for up to 20 random coefficients using a custom implementation. The analysis focuses on estimating the effect of substituting RWMH with HMC on estimation time, convergence, and overall efficiency.

The results demonstrate that HMC exhibits significantly better estimation speed and convergence compared to RWMH. However, we find that the advantage of HMC diminishes as the number of dimensions increases, contrary to what is to be expected from the existing literature.

Bachelor's Thesis Econometrics and Operations Research
Supervisor: Dr. M.K. Osterhaus
Second assessor: J.W. Ligtenberg, MSc

1. Introduction

Discrete choice models are widely used in behavioral economics to model the decision making processes of economic agents such as investment decisions of firms, purchase decisions of consumers, or travel mode choices of commuters. However, the limited availability of computational resources has posed a challenge for these models in the past. This led to the multinomial logit (MNL) model becoming one of the most widely used discrete choice models, as its estimation is straightforward and offered by various software packages (Train, 2009).

With the advent of increased computational resources, the estimation of more complex discrete choice models has become feasible, allowing researchers to relax MNL's limiting assumptions, such as the homogeneity of agent preferences and the independence of irrelevant alternatives (IIA) assumption. One of these more flexible models is the mixed logit (MXL), which is generalization of the MNL model that allows for heterogeneity in agent preferences. The MXL model is especially attractive because it can approximate almost any discrete choice model derived under utility maximization, a common assumption in economics (McFadden and Train, 2000).

Estimating MXL models is more involved as the associated likelihood does not have a closed form. In the literature, two estimation approaches have emerged to tackle this: Maximum simulated likelihood (MSL) and hierarchical Bayes (HB). While MSL maximizes the log-likelihood function, typically simulated using Monte Carlo integration, to estimate the model parameters specified by the researcher, the Bayesian approach relies on posterior sampling for parameter estimation. Train (2009) found that MSL's performance suffers greatly when the MXL model includes a full covariance, due to the increase in the number of parameters. HB is therefore preferable when dealing with models that have full covariance. Moreover, Train describes a HB estimation procedure for MXL, which uses conjugate priors and Gibbs sampling to greatly simplify the otherwise analytically intractable task of posterior sampling. However, one step still requires sampling from a non-normalized density, which is accomplished with random walk Metropolis-Hastings (RWMH).

This thesis extends the current literature by proposing an alteration to the step involving RWMH in the estimation of MXL models. Specifically, we propose to replace RWMH with Hamiltonian Monte Carlo (HMC). HMC is an alternative method for sampling from complex, non-normalized densities that has superseded RWMH for general sampling (Brooks et al., 2011; Betancourt, 2017). HMC accomplishes this by generating proposals that better exploit the geometry of the target density by simulating a physical system using Hamiltonian dynamics, unlike RWMH which generates proposals randomly (Betancourt, 2017). Other studies, like Monnahan et al. (2017), Chen et al. (2020) and Yamada et al. (2022) have all found that HMC can significantly accelerate the estimation of Bayesian models. But the current literature does not yet contain a comparison between RWMH and HMC for MXL.

Specifically, this thesis investigates the difference in the convergence properties and estimation times between HMC and RWMH for MXL models with varying numbers of random coefficients using extensive Monte Carlo studies. These studies were conducted using generated discrete choice datasets consisting of 500 individuals, who make a choice between six options six times. All the features in the datasets follow a normal distribution, and their variances were chosen in a way that ensures the number of random coefficients does not affect the characteristics of the choice probabilities. This design

was selected to be representative of other discrete choice datasets and allow for a fair comparison across differing amounts of random coefficients. The models were estimated using a custom implementation written in C++, which ensures fast computation and facilitates easy monitoring of the internal processes. The results show that HMC exhibits significantly better estimation speed and convergence compared to RWMH, but the difference diminishes when the amount of random coefficients increases, contrary to what Chen et al. (2020) find for general sampling.
The organization of this paper is as follows. Section 2 provides the necessary background on Bayesian and Markov chain Monte Carlo concepts required for HB MXL estimation. In Section 3, the MXL model and the HB estimation using RWMH are described. To address limitations of RWMH, Section 4 introduces and justifies the use of HMC. We then perform a comprehensive simulation study in Section 5. Finally, Section 6 concludes.

2. Bayesian and MCMC Concepts

Before discussing the estimation of MXL models we give an overview of the Bayesian and Markov chain Monte Carlo (MCMC) concepts relevant for it.

2.1. General Bayesian Approach

Consider a model with parameters $\boldsymbol{\theta}$. Both in the frequentist and the Bayesian approach, we are interested in finding $\boldsymbol{\theta}^*$ the true value of $\boldsymbol{\theta}$. However, unlike in a frequentist approach, we assume that we have an idea of what $\boldsymbol{\theta}$ might be whereby we represent this prior knowledge of $\boldsymbol{\theta}$ through the density $f_{\boldsymbol{\theta}}$. Under the model at hand, the likelihood of observing data y, given $\boldsymbol{\theta} = t$, is $L(y|t)$. So, if we observe y, we can use Bayes' theorem to update our beliefs about $\boldsymbol{\theta}$ as follows,

$$f_{\boldsymbol{\theta}|Y=y}(t) = \frac{f_{Y|\boldsymbol{\theta}=t}(y) f_{\boldsymbol{\theta}}(t)}{f_Y(y)} = \frac{L(y|t) f_{\boldsymbol{\theta}}(t)}{\int L(y|x) f_{\boldsymbol{\theta}}(x) \mathrm{d}x} \propto L(y|t) f_{\boldsymbol{\theta}}(t).$$

This is called the posterior distribution, and determining it is the main goal in the Bayesian approach.

2.2. Bayesian Estimators

Using the posterior distribution, we can obtain so-called Bayesian estimators of the parameters, which minimize the expected posterior loss. The expected posterior loss, also known as the Bayes risk, is defined as follows: Given a loss function $l(\cdot)$ from the state space to the real line, the Bayes risk for an estimate $\tilde{\boldsymbol{\theta}}$ is given by,

$$\mathbb{E}_{\boldsymbol{\theta}} \left[l(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}) | Y = y \right].$$

For particular loss functions, one can express the Bayesian estimator, $\hat{\boldsymbol{\theta}}_{\mathrm{B}}$, in terms of sample statistics. For example, if l is the L^1 norm, then $\hat{\boldsymbol{\theta}}_{\mathrm{B}}$ is the median. If l is the L^2 norm, then $\hat{\boldsymbol{\theta}}_{\mathrm{B}}$ is the mean ($\bar{\boldsymbol{\theta}}$), and if $l(t|h) = \mathbb{1}\{||t|| > h\}$, then $\hat{\boldsymbol{\theta}}_{\mathrm{B}}$ approaches the mode as $h \to 0$. For different loss functions, one can always estimate $\hat{\boldsymbol{\theta}}_{\mathrm{B}}$ by minimizing the sample posterior loss. In our situation, we will use $\bar{\boldsymbol{\theta}}$ as our estimator of choice. This is done because the Bernstein-von Mises Theorem (explained in the next paragraph) informs us that $\bar{\boldsymbol{\theta}}$ has a frequentist interpretation.

2.3. The Bernstein-von Mises Theorem

For the maximum likelihood estimate of $\boldsymbol{\theta}$, $\hat{\boldsymbol{\theta}}_{\text{MLE}}$, we have that $\sqrt{N}(\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_{\text{MLE}}) \xrightarrow{d} \mathcal{N}(0, \mathcal{I}^{-1}(\boldsymbol{\theta}^*))$, where $\mathcal{I}(\boldsymbol{\theta}^*)$ is the Fisher information at the true parameter. With $\mathcal{I}(\boldsymbol{\theta}^*)$, the Bernstein-von Mises Theorem can be stated, which constitutes the following results:

1. $\sqrt{N}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})|Y = y \xrightarrow{d} \mathcal{N}(0, \mathcal{I}^{-1}(\boldsymbol{\theta}^*))$.
2. $\sqrt{N}(\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_{\text{MLE}}) \xrightarrow{p} 0$.
3. $\sqrt{N}(\boldsymbol{\theta}^* - \bar{\boldsymbol{\theta}}) = \sqrt{N}(\boldsymbol{\theta}^* - \hat{\boldsymbol{\theta}}_{\text{MLE}}) - \sqrt{N}(\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_{\text{MLE}}) \xrightarrow{d} \mathcal{N}(0, \mathcal{I}^{-1}(\boldsymbol{\theta}^*))$ (by Slutsky's Theorem).

That is, the posterior distribution converges to a normal distribution, the difference between $\bar{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}_{\text{MLE}}$ diminishes asymptotically, and lastly, $\bar{\boldsymbol{\theta}}$ is asymptotically equivalent to $\hat{\boldsymbol{\theta}}_{\text{MLE}}$. For a formal proof of the Bernstein-von Mises theorem see Van der Vaart (2000). The implication is that we can obtain an estimator with the same asymptotic properties as the maximum likelihood estimator without having to maximize a function. Moreover, to obtain standard errors for our estimator $\hat{\boldsymbol{\theta}}$, we can, according to Bernstein-von Mises, use the standard deviation of the posterior.

2.4. Markov Chain Monte Carlo

When sampling from the posterior is analytically intractable, we have to resort to more general sampling methods to obtain our desired Bayesian estimates. Most sampling methods are part of the Markov chain Monte Carlo (MCMC) family of sampling algorithms. MCMC algorithms sample from a target distribution by constructing a Markov chain which has the target distribution as its stationary distribution (Brooks et al., 2011). Stated more precisely, consider a target distribution ξ with support $\mathcal{X}$ that we want to sample from. Then an MCMC algorithm constructs a Markov transition kernel P such that the associated Markov chain $\{X_n, n \geq 0\}$ has ξ as its unique limiting distribution, whereby a Markov transition kernel is the generalization of a transition probability matrix (i.e., $P(x, A)$ indicates the probability of transitioning from point x to region A). For the chain to have a unique limiting distribution, we want that for all $X_0 \sim \xi_0$ that $X_n \sim \xi_0 P^n \xrightarrow{d} \xi$, that is $\lim_{n\to\infty} ||\xi_0 P^n - \xi|| = 0$. A sufficient condition for this is that the chain is aperiodic (i.e., the chain does not return to its starting condition after a constant amount of steps almost surely) and Harris recurrent (i.e., the chain will visit all subsets of the state space infinitely often). These two conditions are generally satisfied.

As $X_n \xrightarrow{d} \xi$, we have that the sample $\{X_n, X_{n+1}, \dots\}$ will be a representative, although correlated, sample from ξ for a large enough n. The fact that these draws are correlated is not a problem, since by the Ergodic Theorem (law of large numbers) for Markov chains we have that for a sample statistic $h : \mathcal{X} \to \mathbb{R}$ that,

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i) \xrightarrow{p} \int_{\mathcal{X}} h(x)\xi(x)dx.$$

So we can still use the produced sample to calculate meaningful sample statistics.

While the construction of the Markov transition kernel P varies between different MCMC algorithms, many of these constructions are based on the Metropolis-Hastings (MH) algorithm, or derived from it in some way (Metropolis et al., 1953; Hastings, 1970).

Metropolis-Hastings Algorithm

We will now give a derivation for the MH algorithm for the case where $|\mathcal{X}|$ is finite. Note that, since $|\mathcal{X}|$ is finite, the transition kernels become Markov matrices, resulting in a more elementary argument. Let Q be an arbitrary irreducible Markov matrix which acts on $\mathcal{X}$. To construct $\{X_n, n \geq 0\}$ such that its stationary distribution is ξ, that is, $\lim_{n\to\infty} \mathbb{P}(X_n = i) = \xi(i)$, we define it as follows: Given $X_n = i$, generate random variable Y such that $\mathbb{P}(Y = j) = Q_{ij}$, then set

$$X_{n+1} = \begin{cases} Y & \text{with probability } \alpha_{ij} \\ X_n & \text{with probability } 1 - \alpha_{ij} \end{cases}.$$

The transition probabilities of this Markov chain are given by,

$$\begin{aligned} P_{ij} &= Q_{ij}\alpha_{ij}, \text{ if } j \neq i, \\ P_{ii} &= Q_{ii} + \sum_{k\neq i} Q_{ik}(1 - \alpha_{ik}). \end{aligned}$$

A sufficient condition for X_n to have ξ as its stationary distribution is time reversibility. That is

$$\xi(i)P_{ij} = \xi(j)P_{ji} \Rightarrow \xi(i)Q_{ij}\alpha_{ij} = \xi(j)Q_{ji}\alpha_{ji}, \text{ for } i \neq j.$$

It can be verified that

$$\alpha_{ij} = \min\left\{\frac{\xi(j)Q_{ij}}{\xi(i)Q_{ji}}, 1\right\},$$

satisfies the condition for time reversibility. Moreover, observe that if $\xi(i) = C\zeta(i)$ where C is a normalization constant, then α_{ij} simplifies to,

$$\alpha_{ij} = \min\left\{\frac{\zeta(j)Q_{ij}}{\zeta(i)Q_{ji}}, 1\right\}.$$

Therefore, if we have a PMF proportional to ξ, then the MH algorithm (the above procedure) gives us the transition kernel P for a Markov chain with ξ as its stationary distribution (Ross, 2019).

The derivation above uses Markov matrices which only exist when $\mathcal{X}$ is countable, however the result can easily be generalized to the case of general $\mathcal{X}$. Let ζ be proportional to ξ like before and let Q be a Markov transition kernel acting on $\mathcal{X}$. Now to construct $\{X_n, n \geq 0\}$ such that its stationary distribution is ξ, we define it as follows: Given X_n, generate a proposal Y conditional on X_n according to Q, and then set

$$\begin{aligned} X_{n+1} &= \begin{cases} Y & \text{with probability } \alpha(Y, X_n) \\ X_n & \text{with probability } 1 - \alpha(Y, X_n) \end{cases}, \\ \text{with } \alpha(Y, X_n) &= \min\left\{\frac{\zeta(Y)}{\zeta(X_n)}\frac{Q(X_n, \mathrm{d}Y)}{Q(Y, \mathrm{d}X_n)}, 1\right\}. \end{aligned} \tag{1}$$

The stationary distribution of X_n will again be ξ (Roberts and Rosenthal, 2004).

Gibbs Sampling

Lastly, we will discuss Gibbs sampling. Gibbs sampling is an MCMC algorithm which can be employed when a distribution is decomposable into simpler conditional distributions. That is, if the target distribution is $f_{Z_1,Z_2,\ldots,Z_K} = f_Z$ and we can sample from the conditional distributions $f_{Z_k|Z_1,\ldots,Z_{k-1},Z_{k+1},\ldots,Z_K} = f_{Z_k|Z_{-k}}$. It works as follows: We start with $\boldsymbol{x}^{(0)} = \left(x_1^{(0)}, \ldots, x_K^{(0)}\right)$. Next, to obtain $\boldsymbol{x}^{(n+1)}$ we do the following: For each $k \in \{1, \ldots, K\}$ we draw $x_k^{(n+1)} | x_1^{(n+1)}, \ldots, x_{k-1}^{(n+1)}, x_{k+1}^{(n)}, \ldots, x_K^{(n)}$. This procedure will produce a Markov chain with f_Z as its limiting distribution. This follows from the fact that Gibbs sampling is a special case of MH with an acceptence probability of 1 (Brooks et al., 2011). To show this, consider the case where we want to sample from f_Z using MH and we update each element of our Markov chain $\{X^{(n)}, n \geq 0\}$ separately. In each step we generate a proposal y for $X_k^{(n+1)}$ from the conditional distribution $f_{Z_k|Z_{-k}}$ which gives rise to the proposal $Y = (X_1^{(n)}, \ldots, X_{k-1}^{(n)}, y, X_{k+1}^{(n)}, \ldots, X_K^{(n)})$. Then, the probability of accepting this proposal is,

$$\begin{aligned}
\alpha(Y, X^{(n)}) &= \min\left\{ \frac{f_Z(Y)}{f_Z(X^{(n)})} \frac{f_{Z_k,Z_{-k}}(X^{(n)})}{f_{Z_k,Z_{-k}}(Y)}, 1 \right\} \\
&= \min\left\{ \frac{f_Z(Y)}{f_Z(X^{(n)})} \frac{f_{Z_{-k}}(X_1^{(n)}, \ldots, X_{k-1}^{(n)}, X_{k+1}^{(n)}, \ldots, X_K^{(n)})}{f_{Z_{-k}}(X_1^{(n)}, \ldots, X_{k-1}^{(n)}, X_{k+1}^{(n)}, \ldots, X_K^{(n)})} \frac{f_Z(X^{(n)})}{f_Z(Y)}, 1 \right\} = 1.
\end{aligned}$$

So, if a posterior is conditionally conjugate then the sampling procedure can be simplified by using Gibbs sampling.

2.5. MCMC Diagnostics

The effectiveness of MH, and MCMC algorithms in general, is mostly determined by how good the proposal distribution Q is. A poorly chosen proposal distribution can lead to a highly correlated chain that converges slowly to the limiting distribution. To distinguish effective and ineffective MCMC algorithms we use diagnostic tools. These tools serve two primary purposes: Assessing convergence and evaluating sample quality.
Assessing convergence involves identifying the moment at which the chain becomes stationary, where we define a Markov chain to be stationary if it is in the critical set (the smallest set containing a certain probability mass). It is crucial to determine this phase change to exclude samples from the non-stationary part of the chain, which can skew the sample statistics. Once the stationary part of the chain is identified, we focus on the effective sample size - an important statistic which indicates the effectiveness of a sampling algorithm.

2.6. MCMC Convergence

Because MCMC algorithms produce a Markov chain which has the target distribution as its limiting distribution, we can expect that the MCMC algorithm will mostly move around in the critical set. Therefore, it is desirable to pick a starting value which is in the critical set, as this starting value will be representative of the stationary behaviour. This is however often not possible as we do not know a priori where the critical set is. So, we must wait for the Markov chain to enter, the critical set and become stationary.

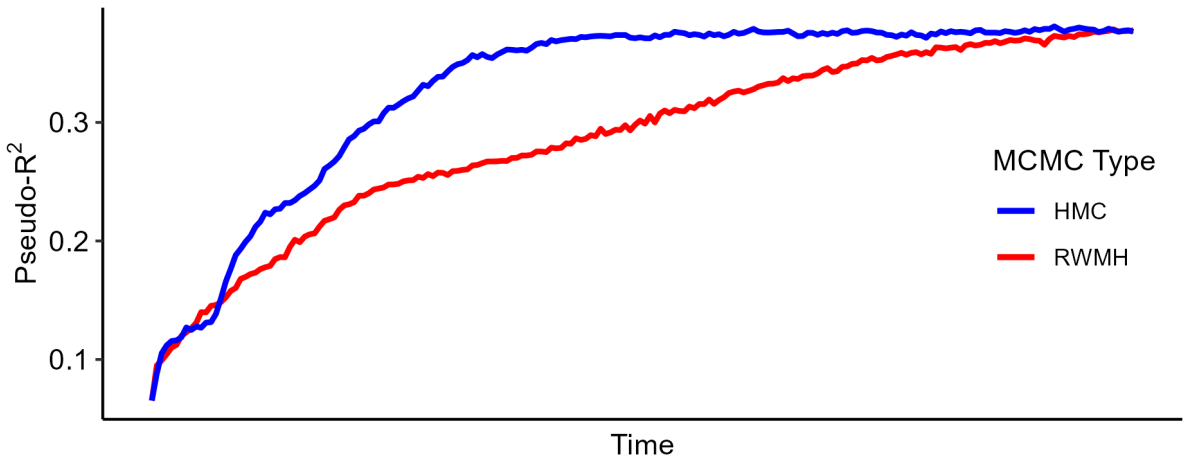

Figure 1: Example pseudo-R^2 trace plot

Cowles and Carlin (1996) point out that determining stationarity usually depends considerably on the situation at hand, so to assess stationarity it is preferable to use multiple quantitative and graphical methods. We will discuss the following such methods: The Gelman-Rubin diagnostic, the pseudo-R^2 trace plot, and a new method we name the quantile consistent diagnostic. Lastly, we will motivate a convergence rate measure.

Gelman-Rubin Diagnostic

The Gelman-Rubin diagnostic first proposed for univariate Markov chains by Gelman and Rubin (1992) and generalized to multivariate Markov chains by Brooks and Gelman (1998), assesses convergence using multiple chains. It is one of the most used diagnostic tools to for determining convergence (Vats and Knudson, 2021). The idea is that when the variance between the chains is similar to the variance within the chains that they have converged. However, as this diagnostic will not be used in the final simulation study, we not discuss it here in detail. For more information on it we refer the reader to Vats and Knudson (2021).

Trace Plots

A trace plot is a time series for a particular variable of interest. Trace plots are the primary graphical method for assessing convergence (Nylander et al., 2008; Brooks et al., 2011). With a trace plot, a researcher can judge whether or not the sample variables and or statistics look stationary. The statistic we are interested in is the pseudo-R^2 given by pseudo-$R^2 = \frac{\text{SLL}(\boldsymbol{\theta}|Y)}{-NT\log J}$ as it often used to quantify the fit of non-linear models. For HB MXL, we have that the pseudo-R^2 should be stationary when the produced samples come from the critical set. An example of such a trace plot is given in Figure 1, from which we can clearly identify convergence and to some extend the convergence rate.

Quantile Consistent Diagnostic

To assess when a chain has become stationary, we propose a diagnostic method that utilizes an estimate of the critical set. Our method addresses two common challenges associated with estimating the critical set. First, a stationary sample is required to estimate the critical set, but we also need an estimate of the critical set to determine the stationary part of the sample. Second, the critical set can exhibit complex shapes making it difficult to estimate. We start with the latter problem, consider a random variable X with support $\mathcal{X} \subseteq \mathbb{R}^K$, the critical set is then defined as follows:

$$\mathcal{C}_\alpha = \underset{V \subset \mathcal{X}}{\operatorname{argmin}} \operatorname{Vol}(V) \text{ such that } \mathbb{P}(X \in V) \geq 1 - \alpha.$$

From this definition, it becomes evident that the critical set can have a pathological shape. In the case of multimodal densities, for instance, the critical set may not be connected. Consequently, when the modes are sufficiently distant from each other, it is likely that we can only identify a portion of the critical set. However, as the densities we will later be using for HB are unimodal, we will focus on the critical set for unimodal densities. Fortunately, the critical set for a unimodal density is convex, making it better behaved. For example, if X follows a $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ distribution, then the critical set is an ellipsoid given by:

$$\mathcal{C}_\alpha = \{\boldsymbol{x} \in \mathbb{R}^K : (\boldsymbol{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \leq \chi^2_{1-\alpha,K}\}.$$

This motivates us to approximate the critical set, in general, by an ellipsoid. Similar to the case of the normal density, we define this ellipsoid in terms of the mean and covariance. The approximation is as follows:

$$\tilde{\mathcal{C}}_\alpha = \{\boldsymbol{x} \in \mathbb{R}^K : (\boldsymbol{x} - \hat{\boldsymbol{\mu}}(s))' \hat{\boldsymbol{\Sigma}}(s)^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}}(s)) \leq q_{1-\alpha}(s)\}$$

where $\hat{\boldsymbol{\mu}}(s) = \frac{1}{R-s} \sum_{r=1+s}^{R} \boldsymbol{x}_r$, $\hat{\boldsymbol{\Sigma}}(s) = \frac{1}{R-s-1} \sum_{r=1+s}^{R} (\boldsymbol{x}_r - \hat{\boldsymbol{\mu}})(\boldsymbol{x}_r - \hat{\boldsymbol{\mu}})'$ and $q_{1-\alpha}(s)$ is the $(1-\alpha)100\%$ quantile estimate of $(\boldsymbol{x}_r - \hat{\boldsymbol{\mu}}(s))' \hat{\boldsymbol{\Sigma}}(s)^{-1} (\boldsymbol{x}_r - \hat{\boldsymbol{\mu}}(s))$ for $r \in \{s+1, \ldots, R\}$. Note that the first s samples have been discarded because the first samples are from the non-stationary part of the sample. As mentioned before, it is challenging to determine when the sample becomes stationary since we require a stationary sample to do so. However, we can overcome this problem with the following observation: s is the first time when the sample becomes stationary, that is, s is the first time that $(\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}(s))' \hat{\boldsymbol{\Sigma}}(s)^{-1} (\boldsymbol{x}_s - \hat{\boldsymbol{\mu}}(s)) \leq q_{1-\alpha}(s)$. We call this property quantile consistency. Thus, if the approximation of the critical set is reasonably accurate, we can identify the stationary part of the sample, by determining the smallest value of s that satisfies the quantile consistency condition.

Rate of Convergence

Most analyses done on the convergence rate of MCMC algorithms are theoretical and focus on upper bounds on the rate of convergence (Roberts and Rosenthal, 2004). However, as upper bounds may not be representative of the average, we will instead quantify convergence empirically. To do this, we will first consider a baseline.

Consider the following Markov chain where $X_{n+1} \sim \mathcal{N}(\rho X_n, \frac{1}{1-\rho^2}\boldsymbol{\Sigma})$. Then by construction $X_n \xrightarrow{d} X \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. We want to know when X_n enters the critical set, which is the same as X_n having a similar distribution as X. We say that X_n and X are similar enough to justify convergence with $(1-\alpha)\cdot 100\%$ confidence if

$$D(X_n, X) \leq \max_{\boldsymbol{\eta} \in \mathcal{C}_{0.05}} D(\boldsymbol{\eta} + X, X),$$

where, D is a statistical distance, and $\mathcal{C}_\alpha$ the critical set of X containing $1-\alpha$ probability. As mentioned previously, the critical set is the smallest set of a certain probability. In this case it can be verified that,

$$\mathcal{C}_\alpha = \{\boldsymbol{x} \in \mathbb{R}^K : \boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \leq \chi^2_{1-\alpha,K}\}.$$

Different choices of D will result in (slightly) different convergence criteria, so we will use the Bhattacharyya distance as it has a closed form for multivariate normals (Fukunaga, 2013). Specifically, when $P \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $Q \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ we have that,

$$D(P,Q) = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \left(\frac{1}{2}\boldsymbol{\Sigma}_1 + \frac{1}{2}\boldsymbol{\Sigma}_2\right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\log\left(\frac{\det\left(\frac{1}{2}\boldsymbol{\Sigma}_1 + \frac{1}{2}\boldsymbol{\Sigma}_2\right)}{\sqrt{\det(\boldsymbol{\Sigma_1})\det(\boldsymbol{\Sigma_2})}}\right).$$

Now, since $X_n \sim \mathcal{N}(\rho^n X_0, (1-\rho^{2n})\boldsymbol{\Sigma})$, it can be verified that,

$$D(X_n, X) = \frac{1}{8}\left(1 - \frac{1}{2}\rho^{2n}\right)^{-1} \rho^{2n} X_0'\boldsymbol{\Sigma}^{-1}X_0 + \frac{1}{2}\log\left(\frac{\left(1-\frac{1}{2}\rho^{2n}\right)^k}{\left(1-\rho^{2n}\right)^{k/2}}\right) \approx \frac{\rho^{2n}}{8} X_0'\boldsymbol{\Sigma}^{-1}X_0.$$

Moreover, as $\max_{\boldsymbol{\eta} \in \mathcal{C}_{0.05}} D(\boldsymbol{\eta}+X, X) = \frac{1}{8}\chi^2_{0.95,K}$, we obtain the following convergence criteria,

$$n \geq \frac{\log\left(\chi^2_{0.95,K}\right) - \log\left(X_0'\boldsymbol{\Sigma}^{-1}X_0\right)}{2\log(\rho)}.$$

Lastly, if we have data on convergence times and the convergence is autoregressive, then we can estimate ρ as,

$$\rho \approx \frac{1}{R}\sum_{r=1}^{R}\left(\frac{\chi^2_{0.95,K}}{X_{r,0}'\boldsymbol{\Sigma}^{-1}X_{r,0}}\right)^{\frac{1}{2n_r}}.$$

2.7. Effective sample size

When the MCMC algorithm becomes stationary, we want it to produce a representative sample quickly. To quantify this notation, the effective sample size (ESS) is used. The ESS of a (correlated) sample is defined to be the amount of i.i.d. samples which contain as much information for inference. This is often less than the actual sample size, due to positive autocorrelations, but may also be greater than the actual sample size when negative autocorrelations are present. This intuitive relation between the ESS and the autocorrelations is backed up by its univariate definition. For a univariate sample $\{x_n\}$, the ESS is given by,

$$R_{\text{eff}} = R\left(\sum_{r=-\infty}^{\infty}\text{corr}(x_0, x_r)\right)^{-1} = R\left(1 + 2\sum_{r=1}^{\infty}\text{corr}(x_0, x_r)\right)^{-1}.$$

Note that for a multivariate sample this is not defined as the autocorrelations are not a scalar. A way to circumvent this is by considering the maximum element wise ESS of the sample. That is, if the sample is $\{\boldsymbol{x}_n\}$, then the ESS can be taken to be,

$$R_{\text{eff}} = \max_{i\in\{1,\dots,K\}} R\left(\sum_{r=-\infty}^{\infty} \text{corr}(x_{i0}, x_{ir})\right)^{-1}.$$

This will however underestimate the ESS as it depends on the slowest mixing element of the sample, moreover it also disregards internal correlations. Vats et al. (2019) consider a different definition. They use that, by the central limit theorem for Markov chains, we have that $\sqrt{n}(\bar{\boldsymbol{x}} - \boldsymbol{x}_n) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi})$ for some $\boldsymbol{\Psi}$. Using this $\boldsymbol{\Psi}$, they take the ESS of a K-dimensional sample to be

$$R_{\text{eff}} = R\left(\frac{\det\left(\text{var}\left(X\right)\right)}{\det\left(\boldsymbol{\Psi}\right)}\right)^{\frac{1}{K}}.$$

$\text{var}\left(X\right)$ can be estimated from the sample, and a strongly consistent estimator (a.s. convergence) of $\boldsymbol{\Psi}$ can be obtained using a batch means estimator. This batch means estimator is computed as follows: Split the sample into a batches of size b and compute the batch means given by

$$\boldsymbol{B}_m = \frac{1}{b}\sum_{k=1}^{b} \boldsymbol{x}_{mb+k}, \text{ for } m \in \{0, \dots, a-1\}.$$

The batch means estimator of $\boldsymbol{\Psi}$ is then,

$$\hat{\boldsymbol{\Psi}} = \frac{b}{a-1}\sum_{k=0}^{a-1}(\boldsymbol{B}_k - \bar{\boldsymbol{B}})(\boldsymbol{B}_k - \bar{\boldsymbol{B}})',$$

where for strong consistency b is required to be increasing in n. Moreover, it is usually chosen to be $b = n^{\alpha}$ where $\alpha \in [\frac{1}{4}, \frac{1}{2}]$.
Lastly, since the speed of a sampling algorithm is also important, we will often look at the amount of time to produce an effective draw. This makes comparing sampling methods where the amount of computation per iteration is different possible.

3. Mixed Logit Model

This section will introduce the mixed logit model and the two estimation approaches that are most commonly used in the literature, namely MSL and HB.

3.1. The Mixed Logit Model

As mentioned earlier, the MXL model is an extension of the MNL model that, as its most important feature, accommodates for heterogeneity in the preferences of agents. Before discussing MXL we will first describe the MNL model in more detail.
According to Train (2009) the MNL model is by far the easiest and most widely used discrete choice model. One reason for this is that the MNL model is a random utility model, a choice model which can be derived under utility maximization. Random utility models are often used in economic situations as economic agents are usually assumed to be utility maximizing. In general, in a random utility model an agent, labeled n, has a choice set of J alternatives and alternative $j \in \{1, \dots, J\}$ has associated utility $U_{nj} = V(\boldsymbol{s}_n, \boldsymbol{x}_{nj}) + \varepsilon_{nj}$ where $\boldsymbol{s}_n$ and $\boldsymbol{x}_{nj}$ are features of the agent and the j-th alternative respectively. The unobserved part of the utility, ε_{nj}, is assumed to be stochastic. By the assumption of utility maximization we have that the probability of agent n choosing alternative j is given by,

$$P_{nj} = \mathbb{P}(U_{nj} > U_{ni} \text{ for all } i \neq j).$$

The MNL model models these choice probabilities as follows: If an agent, labeled n, is faced with a choice between J alternatives, T times, which all have K features, then the likelihood of alternatives $\boldsymbol{y}_n$ being chosen is,

$$L(\boldsymbol{y}_n) = \prod_{t=1}^{T} \frac{\exp(\boldsymbol{\beta}' \boldsymbol{x}_{nty_{nt}})}{\sum_{s=1}^{J} \exp(\boldsymbol{\beta}' \boldsymbol{x}_{nts})} \tag{2}$$

where $\boldsymbol{x}_{ntj}$ are agent n their features for alternative j in time period t and $\boldsymbol{\beta}$ is a K dimensional vector of parameters that assign a weight to each feature. McFadden (1974) showed that one gets the choice probabilities of the MNL model if and only if ε_{ntj} follows a Gumbel distribution and $U_{ntj} = \boldsymbol{\beta}' \boldsymbol{x}_{ntj} + \varepsilon_{ntj}$.
Another reason why MNL is widely used is due to its simplicity. However, if we want to model more complex situations MNL is insufficient. One reason as to why MNL is insufficient is that it requires the assumption of the independence of irrelevant alternatives (IIA), that is, the ratio of the choice probabilities of two options only depends on those two options. When the choices an agent faces are similar, such as choosing between a red or a blue bus, this assumption becomes hard to justify (McFadden, 1974). Another deficiency of the MNL model is that it assumes that the preferences for all agents are identical (i.e., every agent has the same $\boldsymbol{\beta}$)
The MXL deals with these problems by relaxing the homogeneity assumption of agent preferences. In the MXL we allow $\boldsymbol{\beta}$ to be drawn from a distribution with pdf $f_{\boldsymbol{\beta}}$ with parameters $\boldsymbol{\theta}$. In MXL $\boldsymbol{\theta}$ is therefore estimated and not the individual $\boldsymbol{\beta}$'s. Moreover, when agents make multiple choices we need to take this heterogeneity into account. Therefore, the likelihood of agent n choosing alternatives $\boldsymbol{y}_n$ becomes,

$$L(\boldsymbol{y}_n|\boldsymbol{\theta}) = \int_{\mathbb{R}^K} L(\boldsymbol{y}_n|\boldsymbol{t}) f_{\boldsymbol{\beta}}(\boldsymbol{t}|\boldsymbol{\theta}) \mathrm{d}^K \boldsymbol{t}, \text{ where } L(\boldsymbol{y}_n|\boldsymbol{t}) = \prod_{t=1}^{T} \frac{\exp(\boldsymbol{t}' \boldsymbol{x}_{nty_{nt}})}{\sum_s \exp(\boldsymbol{t}' \boldsymbol{x}_{nts})}. \tag{3}$$

The primary advantage of the MXL model over other discrete choice models is that it can approximate (almost) any random utility model to arbitrary precision, even models that do not assume IIA (McFadden and Train, 2000). A disadvantage of the MXL model is that the estimation of the parameters becomes more involved.

3.2. Maximum Simulated Likelihood

The most common method of determining the parameters of a model is by maximizing the associated log-likelihood. By using the choice probabilities given by Equation 3, we obtain the following log-likelihood for the MXL model,

$$\mathrm{LL}(\boldsymbol{\theta}|Y) = \log\left[\prod_{n=1}^{N} L(\boldsymbol{y}_n|\boldsymbol{\theta})\right] = \sum_{n=1}^{N} \log\left(L(\boldsymbol{y}_n|\boldsymbol{\theta})\right).$$

Maximizing LL is not feasible analytically, so numerical optimization algorithms, such as BFGS for example, are used to maximize it. To use an optimization algorithm, $\mathrm{LL}(\boldsymbol{\theta}|Y)$ needs to be evaluated. However, because $L(\boldsymbol{y}_n|\boldsymbol{\theta})$ does not have a closed-form expression, this must also be done numerically. To evaluate $L(\boldsymbol{y}_n|\boldsymbol{\theta})$, many studies employ Monte Carlo integration, which is a numerical integration method that unlike, quadrature-based methods, avoids the issue of error depending on the number of dimensions (Niederreiter, 1992). Monte Carlo integration works by relating the integral to an expectation and computing a sample mean. Specifically, we have that:

$$L(\boldsymbol{y}_n|\boldsymbol{\theta}) = \int_{\mathbb{R}^K} L(\boldsymbol{y}_n|\boldsymbol{t}) f_{\boldsymbol{\beta}}(\boldsymbol{t}|\boldsymbol{\theta}) \mathrm{d}^K\boldsymbol{t} = \mathbb{E}_{\boldsymbol{\beta}}[L(\boldsymbol{y}_n|\boldsymbol{\beta})] \approx \frac{1}{R}\sum_{r=1}^{R} L(\boldsymbol{y}_n|\boldsymbol{\beta}^r)$$

where $\boldsymbol{\beta}^r$ are draws from $f_{\boldsymbol{\beta}}$. Using numerical integration of the integral gives the so-called simulated log-likelihood,

$$\mathrm{SLL}(\boldsymbol{\theta}|Y) = \sum_{n=1}^{N} \log\left[\frac{1}{R}\sum_{r=1}^{R} L(\boldsymbol{y}_n|\boldsymbol{\beta}^r)\right].$$

The simulated log-likelihood can be used to estimate the maximum likelihood estimates. However, MSL does have several deficiencies which do not make it appealing in certain situations. Firstly, as MSL uses optimization we have to deal with problems like failing to converge, local extrema or sensitivity to starting values. For example, when $f_{\boldsymbol{\beta}}$ follows a log normal distribution the maximization of SLL is ill behaved (Train, 2009). Secondly, if we want to allow the elements of $\boldsymbol{\beta}$ to be correlated with each other, then the amount of parameters becomes quadratic instead of linear, resulting in a poor scalability.

3.3. Hierarchical Bayes for Mixed Logit

We will now discuss HB estimation for MXL. An alternative to MSL which works especially well when $\boldsymbol{\beta}$ follows a normal distribution. Moreover, unlike MLS, HB is (much) less sensitive to the initial values and will not get stuck in local maxima. HB does have practical problems, like slow convergence and high autocorrelations, but these problems can be resolved by running the algorithm for longer.

As the name suggests, HB comes from a Bayesian statistics. This is different from the frequentist statistics on which maximum likelihood is based. However, the results from

HB can be interpreted in a frequentist perspective by using the Bernstein-von Mises Theorem. So it is an estimation procedure, and not a different model (Huber and Train, 2001).
We will consider the MXL model where we assume that $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, so the likelihood of agent n choosing $\boldsymbol{y}_n$ then becomes,

$$L(\boldsymbol{y}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \int_{\mathbb{R}^K} L(\boldsymbol{y}_n|\boldsymbol{t})\phi(\boldsymbol{t}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\mathrm{d}^K\boldsymbol{t}.$$

Where $\phi(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The goal is to estimate the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Since we are considering the problem from a Bayesian perspective, we need to pick priors for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to obtain parameter estimates. As we do not have prior information, we pick diffuse priors. The particular diffuse prior is not relevant, so we pick a conjugate prior. This is done because, if a prior is conjugate then the posterior is in the same probability distribution family. This guarantees that we have a closed-form expression for the posterior, eliminating the need for numerical methods. In the standard specification of HB the following two conjugacy relations are used. Consider the distribution, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\eta}, \boldsymbol{\Omega})$ from which samples $\boldsymbol{x}_i$ are observed. If we take $\boldsymbol{\Omega}$ as known, $\boldsymbol{\eta}$ as unknown, and assume the following prior $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\eta}_0, \boldsymbol{\Omega}_0)$, then the posterior is,

$$\boldsymbol{\eta}|\boldsymbol{x}_1 \ldots, \boldsymbol{x}_n \sim \mathcal{N}\left(\left(\boldsymbol{\Omega}_0^{-1} + n\boldsymbol{\Omega}^{-1}\right)^{-1}\left(\boldsymbol{\Omega}_0^{-1}\boldsymbol{\eta}_0 + n\boldsymbol{\Omega}^{-1}\bar{\boldsymbol{x}}\right), \left(\boldsymbol{\Omega}_0^{-1} + n\boldsymbol{\Omega}^{-1}\right)^{-1}\right).$$

Moreover, if we take $\boldsymbol{\Omega}_0 = \mathrm{diag}(s, \ldots, s)$ and let $s \to \infty$, that is we assume an uninformative prior, we have that,

$$\boldsymbol{\eta}|\boldsymbol{x}_1 \ldots, \boldsymbol{x}_n \sim \mathcal{N}\left(\bar{\boldsymbol{x}}, \frac{1}{n}\boldsymbol{\Omega}\right).$$

If we instead take $\boldsymbol{\eta}$ as known, $\boldsymbol{\Omega}$ as unknown, and assume that $\boldsymbol{\Omega} \sim \mathrm{IW}(\boldsymbol{\Psi}, \nu)$, then the posterior is,

$$\boldsymbol{\Omega}|\boldsymbol{x}_1 \ldots, \boldsymbol{x}_n \sim \mathrm{IW}\left(\boldsymbol{\Psi} + \sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{\eta})(\boldsymbol{x}_i - \boldsymbol{\eta})', \nu + n\right)$$

where IW denotes the inverse Wishart distribution which is the multidimensional generalization of the inverse Gamma distribution. The inverse Wishart distribution is defined as follows, if $\boldsymbol{G}_1, \ldots, \boldsymbol{G}_\nu \sim \mathcal{N}(\boldsymbol{0}, \Psi^{-1})$ and $\boldsymbol{S} = [\boldsymbol{G}_1 \ldots \boldsymbol{G}_\nu]$, then $(\boldsymbol{S}\boldsymbol{S}')^{-1} \sim \mathrm{IW}(\boldsymbol{\Psi}, \nu)$. For further information about conjugate priors and proofs of these results see Fink (1997).
The priors for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ will consequently be, $\mu \sim \lim_{s\to\infty} \mathcal{N}(\boldsymbol{0}, sI_K)$ and $\boldsymbol{\Sigma} \sim \mathrm{IW}(\boldsymbol{V}, \nu)$. With these priors the posterior becomes,

$$f_{\boldsymbol{\mu},\boldsymbol{\Sigma}|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N}(\boldsymbol{\eta}, \boldsymbol{\Psi}) \propto f_{\boldsymbol{\mu}}(\boldsymbol{\eta})f_{\boldsymbol{\Sigma}}(\boldsymbol{\Psi})\prod_{n=1}^{N} L(\boldsymbol{y}_n|\boldsymbol{\eta}, \boldsymbol{\Psi}).$$

To sample from this posterior, we will consider the joint density of the posterior and the parameters,

$$f_{\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_N|\boldsymbol{y}_1,\ldots,\boldsymbol{y}_N}(\boldsymbol{\eta}, \boldsymbol{\Psi}, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_N) \propto f_{\boldsymbol{\mu}}(\boldsymbol{\eta})f_{\boldsymbol{\Sigma}}(\boldsymbol{\Psi})\prod_{n=1}^{N} L(\boldsymbol{y}_n|\boldsymbol{t}_n)\phi(\boldsymbol{t}_n|\boldsymbol{\eta}, \boldsymbol{\Psi}).$$

This density has, by conjugacy, simpler conditional distributions making Gibbs sampling effective. These conditional distributions are as follows:

$$f_{\boldsymbol{\beta}_n|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{y}_n}(\boldsymbol{t}) \propto L(\boldsymbol{y}_n|\boldsymbol{t})\phi(\boldsymbol{t}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \text{ for all } n \in \{1,\dots,N\},$$

$$\boldsymbol{\mu}|\boldsymbol{\Sigma},\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_N \sim \mathcal{N}\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\beta}_n, \frac{1}{N}\boldsymbol{\Sigma}\right),$$

$$\boldsymbol{\Sigma}|\boldsymbol{\mu},\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_N \sim \text{IW}\left(\boldsymbol{V}+\sum_{n=1}^{N}(\boldsymbol{\beta}_n-\boldsymbol{\mu})(\boldsymbol{\beta}_n-\boldsymbol{\mu})', \nu+N\right).$$

Using these conditional distributions, we can obtain samples from the posterior by using Gibbs sampling. The baseline specification of this scheme works as follows: Start with $\boldsymbol{\mu}^{(0)}, \boldsymbol{\Sigma}^{(0)}, \boldsymbol{\beta}_1^{(0)}, \dots, \boldsymbol{\beta}_N^{(0)}$ and update them with the subsequent scheme.

1. Draw $\boldsymbol{\mu}^{(r)}$ from $\mathcal{N}\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\beta}_n^{(r-1)}, \frac{1}{N}\boldsymbol{\Sigma}^{(r-1)}\right)$.
2. Draw $\boldsymbol{\Sigma}^{(r)}$ from $\text{IW}\left(\boldsymbol{V}+\sum_{n=1}^{N}\left(\boldsymbol{\beta}_n^{(r-1)}-\boldsymbol{\mu}^{(r)}\right)\left(\boldsymbol{\beta}_n^{(r-1)}-\boldsymbol{\mu}^{(r)}\right)', \nu+N\right)$.
3. For all $n \in \{1,\dots,N\}$, draw $\boldsymbol{\beta}_n^{(r)}$ from a density g which is proportional to $f_{\boldsymbol{\beta}_N|\boldsymbol{\mu}=\boldsymbol{\mu}^{(r)},\boldsymbol{\Sigma}=\boldsymbol{\Sigma}^{(r)},\boldsymbol{y}_n}$ using a single iteration of random walk MH. Specifically, the proposal $\boldsymbol{\beta}_n^*$ is drawn from $\mathcal{N}(\boldsymbol{\beta}_n^{(r-1)}, \rho^2\boldsymbol{\Sigma}^{(r)})$ (ρ determines the size of the proposal jumps), and the proposal is accepted with a probability of

$$\min\left\{\frac{g(\boldsymbol{\beta}_n^*)}{g\left(\boldsymbol{\beta}_n^{(r-1)}\right)}, 1\right\}.$$

Because the Gibbs sampler and the nested MH sampler converge simultaneously, we only need to draw a single iteration of MH for each Gibbs iteration. After the Markov chain has converged, the draws can be used for computing statistics. If we take $\nu = K$ and $\boldsymbol{V} = KI_K$, then we obtain the baseline specification for HB described by Train (2009). This baseline algorithm is a good starting point, but can be have potential drawbacks due its simplicity. Akinc and Vandebroek (2018) point out that the covariance prior used in the baseline specification is predisposed to large correlations, which can result in inaccurate estimation results. Moreover, since we do not have prior knowledge of the problem at hand, we are not able to pick a good starting value for $\boldsymbol{\mu}^{(0)}, \boldsymbol{\Sigma}^{(0)}, \boldsymbol{\beta}_1^{(0)}, \dots, \boldsymbol{\beta}_N^{(0)}$. This is usually remedied by first letting the MCMC algorithm converge to a high probability part of the sample space to only then collect samples for inference. But this time until converge depends on the algorithm used, and RWMH used in the baseline specification converges slowly. Lastly, RWMH is prone to move around slowly, resulting in highly correlated draws. As alluded to previously, we propose to use HMC to remedy the efficiency problems that RWMH has.

4. Hamiltonian Monte Carlo

In this section, we will cover Hamiltonian Monte Carlo (HMC) algorithms. HMC algorithms are MH algorithms which generate their proposals by simulating a physical system using Hamiltonian dynamics. HMC algorithms make more informed proposals than RWMH algorithms, which can result in greater sampling efficiency.
While HMC is a (surprisingly) simple and effective algorithm, it is not evident why it generates a Markov chain with the target distribution as its stationary distribution and why it does this so efficiently. Therefore, we will first discuss the motivation for HMC and then present a derivation of the baseline HMC algorithm.

4.1. HMC Motivation

The main reason why HMC can be better than RWMH is that it makes better use of the geometry of the critical set (Betancourt, 2017). When a density is not unimodal and or symmetric, we have that certain directions are preferred over others for proposals. As an example, consider the density in Figure 2. For this density it is preferential to not move towards the origin. However, as the proposals generated by RWMH are not preferential to any direction, we have that many proposals in the non preferred direction will be rejected. Moreover, since the proposals (trajectories) stay for the most part within the critical set, we have that it can be explored quickly, as can be seen in Figure 3, resulting in a lower autocorrelation.

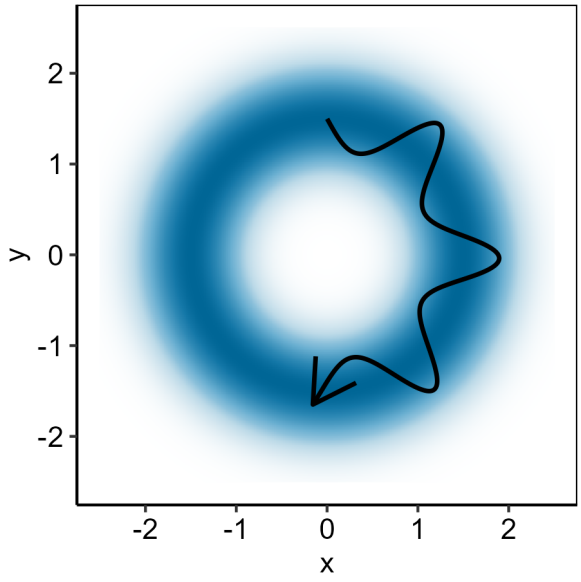

Figure 2: Trajectory on a ring density

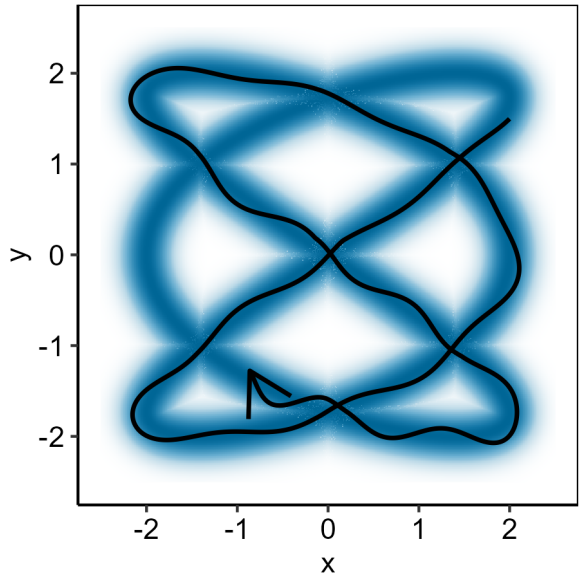

Figure 3: Trajectory on a complex density

4.2. HMC Derivation

To prove the correctness of HMC we present a distilled version of Vats (2023) and Brooks et al. (2011). Start by considering the same situation as in Equation 1, we then propose to generate the proposals as follows: First, draw an auxiliary variable $\boldsymbol{p}$ from a distribution with density $f_{\boldsymbol{p}|\boldsymbol{x}_n}$, then let the proposal be $(\boldsymbol{x}^*, \boldsymbol{p}^*) = g(\boldsymbol{x}_n, \boldsymbol{p})$, where g is its own

inverse (an involution), that is $g(g(\boldsymbol{x}, \boldsymbol{p})) = (\boldsymbol{x}, \boldsymbol{p})$. Then, the probability of accepting the proposal is given by,

$$\alpha(\boldsymbol{x}^*, \boldsymbol{x}_n | \boldsymbol{p}) = \min \left\{ \frac{\zeta(\boldsymbol{x}^*)}{\zeta(\boldsymbol{x}_n)} \frac{f_{\boldsymbol{p}|\boldsymbol{x}_n}(\boldsymbol{p}^*|\boldsymbol{x}^*)}{f_{\boldsymbol{p}|\boldsymbol{x}_n}(\boldsymbol{p}|\boldsymbol{x}_n)} \det\left(\mathrm{D}g(\boldsymbol{x}_n, \boldsymbol{p})\right), 1 \right\}.$$

The Jacobian of $g(\boldsymbol{x}_n, \boldsymbol{p})$, $\mathrm{D}g(\boldsymbol{x}_n, \boldsymbol{p})$, appears due to the fact that we have a change of variables invoked by the deterministic map g. This peculiar proposal is a special case of the Metropolis-Hastings-Green algorithm (MHG), a generalization of the MH algorithm (Green, 1995; Geyer, 2003). We show that HMC is valid by showing that it amounts to a particular choice of g and $f_{\boldsymbol{p}|\boldsymbol{x}_n}$.

The choice of g and $f_{\boldsymbol{p}|\boldsymbol{x}_n}$ is based on Hamiltonian dynamics, so let us cover it briefly. Consider a particle with position vector $\boldsymbol{x}$ and momentum (velocity) vector $\boldsymbol{p}$ moving without friction (loss of energy) through a space with a potential energy field. The evolution of the system can be determined by solving the Hamiltonian equations, a set of partial differential equations. Specifically for HMC, we take the potential energy to be $-\log(\zeta(\boldsymbol{x}))$ and kinetic energy to be $\frac{1}{2}\boldsymbol{p}'\boldsymbol{M}^{-1}\boldsymbol{p}$ where $\boldsymbol{M}$ is a positive definite, usually diagonal, matrix called the 'mass matrix'. The so-called Hamiltonian of this system is the total energy of the system (sum of kinetic and potential energy) and it is given by,

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{p}) = -\log(\zeta(\boldsymbol{x})) + \frac{1}{2}\boldsymbol{p}'\boldsymbol{M}^{-1}\boldsymbol{p}.$$

The Hamiltonian equations, which describe the trajectory of the particle, are the following differential equations,

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial p_i}, \quad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial x_i} \text{ for all } i \in \{1, \ldots, K\}.$$

Given initial conditions $(\boldsymbol{x}_n, \boldsymbol{p})$, we can determine the position and momentum of the particle after s time units. We will denote this as $T_s(\boldsymbol{x}_n, \boldsymbol{p})$. Hamiltonian dynamics have three properties which are relevant for HMC, namely time reversibility, conservation of energy, and volume preservation. Proofs for these properties can be found in the Appendix. Time reversibility entails the following: Let $r(\boldsymbol{x}, \boldsymbol{p}) = (\boldsymbol{x}, -\boldsymbol{p})$ and consider $(r \circ T_s)(\boldsymbol{x}, \boldsymbol{p})$, that is we follow the trajectory for s time units and then negate the momentum vector. Then, due to time reversibility we have that $((r \circ T_s) \circ (r \circ T_s))(\boldsymbol{x}, \boldsymbol{p}) = (\boldsymbol{x}, \boldsymbol{p})$. This means that if we start at $(\boldsymbol{x}, \boldsymbol{p})$, follow the trajectory for s time units, negate the momentum, and again follow the trajectory for s time units, we will end up at $(\boldsymbol{x}, \boldsymbol{p})$ again. So, $r \circ T_s$ is an involution. As, $\mathcal{H}(\boldsymbol{x}, \boldsymbol{p}) = \mathcal{H}(\boldsymbol{x}, -\boldsymbol{p})$ we can take T_s to be our choice of g.

Now if we let $\boldsymbol{p}$ follow $\mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$, we have all we need for our special case of MHG. We find that the acceptance probability is given by

$$\begin{aligned}
\alpha(\boldsymbol{x}^*, \boldsymbol{x}_n | \boldsymbol{p}) &= \min \left\{ \frac{\zeta(\boldsymbol{x}^*)}{\zeta(\boldsymbol{x}_n)} \frac{\phi(\boldsymbol{p}^*|\boldsymbol{0}, M)}{\phi(\boldsymbol{p}|\boldsymbol{0}, M)} \det\left(\mathrm{D}T_s(\boldsymbol{x}_n, \boldsymbol{p})\right), 1 \right\} \\
&= \min \left\{ \frac{\exp\left(\log\left(\zeta(\boldsymbol{x}^*)\right)\right)}{\exp\left(\log\left(\zeta(\boldsymbol{x}_n)\right)\right)} \frac{(2\pi)^{-K/2} \det(M)^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{p}^{*\prime}\boldsymbol{M}^{-1}\boldsymbol{p}^*\right)}{(2\pi)^{-K/2} \det(M)^{-1/2} \exp\left(-\frac{1}{2}\boldsymbol{p}'\boldsymbol{M}^{-1}\boldsymbol{p}\right)}, 1 \right\} \\
&= \min\left\{\exp\left(-\mathcal{H}(\boldsymbol{x}^*, \boldsymbol{p}^*) + \mathcal{H}(\boldsymbol{x}_n, \boldsymbol{p})\right), 1\right\} = 1.
\end{aligned}$$

The Jacobian disappears because of volume preservation, and in the last step we use the fact that the Hamiltonian is conserved.

In practice, because the target distribution is interesting, we cannot solve the Hamiltonian analytically, so we must compute the trajectory numerically. This is typically done with leapfrog integration as it preserves volumes. However, leapfrog integration does not conserve the Hamiltonian, so the acceptance probability does need to be calculated as it might not be 1. The baseline Hamiltonian Monte Carlo algorithm is then as follows: Draw $\boldsymbol{p}$ from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{M})$, generate a proposal $(\boldsymbol{x}^*, \boldsymbol{p}^*) = T_L(\boldsymbol{x}_n, \boldsymbol{p})$ by using L/ε leapfrog integration steps, where L is the time the trajectory is simulated and ε is the step size, then accept the proposal with a probability of $\min\left\{\exp\left(-\mathcal{H}(\boldsymbol{x}^*, \boldsymbol{p}^*) + \mathcal{H}(\boldsymbol{x}_n, \boldsymbol{p}\right), 1\right\}$.

Implementation Details

As mentioned before, leapfrog integration is used to compute the trajectories governed by Hamiltonian dynamics numerically. It works as follows, if we know that $(\boldsymbol{x}_s, \boldsymbol{p}_s) = T_{s\varepsilon}(\boldsymbol{x}, \boldsymbol{p})$, then the leapfrog approximation of $(\boldsymbol{x}_{s+1}, \boldsymbol{p}_{s+1})$ is,

$$\begin{aligned}
\boldsymbol{p}_{s+\frac{1}{2}} &= \boldsymbol{p}_s + \frac{1}{2}\varepsilon \mathrm{D}(\log(\zeta))(\boldsymbol{x}_s)', \\
\boldsymbol{x}_{s+1} &= \boldsymbol{x}_s + \varepsilon \boldsymbol{M}^{-1}\boldsymbol{p}_{s+\frac{1}{2}}, \\
\boldsymbol{p}_{s+1} &= \boldsymbol{p}_{s+\frac{1}{2}} + \frac{1}{2}\varepsilon \mathrm{D}(\log(\zeta))(\boldsymbol{x}_{s+1})'.
\end{aligned}$$

Leapfrog integration is used because it is stable, in the sense that the errors in the trajectories do not compound. Lastly, in MXL the gradient of $\log(\zeta)$, $\log\left(f_{\boldsymbol{\beta}_n|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{y}_n}(\boldsymbol{t})\right)$ for MXL, can be determined analytically. Recall $\log(\zeta)$ is given by,

$$\log(\zeta(\boldsymbol{\beta})) = \sum_{t=1}^{T} \log\left(\frac{\exp(\boldsymbol{\beta}'\boldsymbol{x}_{nty_{nt}})}{\sum_s \exp(\boldsymbol{\beta}'\boldsymbol{x}_{nts})}\right) - \frac{K}{2}\log(2\pi) - \frac{1}{2}\log(\det(\boldsymbol{\Sigma})) - \frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu}).$$

The partial derivatives are then,

$$\frac{\partial \log(\zeta(\boldsymbol{\beta}))}{\partial \beta_k} = \sum_{t=1}^{T}\left(x_{nty_{nt}k} - \frac{\sum_s x_{ntsk}\exp(\boldsymbol{\beta}'\boldsymbol{x}_{nts})}{\sum_s \exp(\boldsymbol{\beta}'\boldsymbol{x}_{nts})}\right) - \left[\boldsymbol{\Sigma}^{-1}(\boldsymbol{\beta}-\boldsymbol{\mu})\right]_k.$$

By evaluating these terms in parallel, we avoid recomputing $\exp(\boldsymbol{\beta}'\boldsymbol{x}_{nts})$, resulting in an order of magnitude fewer computations.

4.3. HMC Hyperparameters

The choice of ε and L is very important as it can make or break the effectiveness of an HMC sampler (Hoffman et al., 2014). For example, if the target distribution is multivariate normal with zero mean and covariance $\boldsymbol{\Sigma}$, then $\mathcal{H}(\boldsymbol{x}, \boldsymbol{p}) = \frac{1}{2}\boldsymbol{x}'\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{p}'\boldsymbol{M}^{-1}\boldsymbol{p}$. The Hamiltonian equations will give that,

$$\begin{pmatrix}\boldsymbol{x}'(t)\\ \boldsymbol{p}'(t)\end{pmatrix} = \begin{pmatrix}\boldsymbol{O} & -\boldsymbol{M}^{-1}\\ \boldsymbol{\Sigma}^{-1} & \boldsymbol{O}\end{pmatrix}\begin{pmatrix}\boldsymbol{x}(t)\\ \boldsymbol{p}(t)\end{pmatrix} \Rightarrow T_s(\boldsymbol{x}_n, \boldsymbol{p}) = \begin{pmatrix}\boldsymbol{x}_n\\ \boldsymbol{p}\end{pmatrix}\exp\left(s\begin{pmatrix}\boldsymbol{O} & -\boldsymbol{M}^{-1}\\ \boldsymbol{\Sigma}^{-1} & \boldsymbol{O}\end{pmatrix}\right).$$

It can be verified that the eigenvalues of $\begin{pmatrix}\boldsymbol{O} & -\boldsymbol{M}^{-1}\\ \boldsymbol{\Sigma}^{-1} & \boldsymbol{O}\end{pmatrix}$ are $\pm i\frac{1}{\sqrt{\lambda_k}}$ where λ_k are the eigenvalues of $\boldsymbol{\Sigma}\boldsymbol{M}$. As these eigenvalues are strictly complex, they indicate that the trajectories will be periodic, so after some time they will come close to the starting point.

Ending up where we started is of course undesirable, so this is to be avoided. This behaviour is not unique to this simple example (we see this behavior in Figures 2 and 3 as well). Hamiltonian trajectories will in fact (almost) always return arbitrarily closely to their starting point (Betancourt, 2016).
Hoffman et al. (2014) introduced the No U-Turn sampler, a HMC sampler which addresses this issue. Most of their work is focused on ensuring that their sampler is time reversible whilst aiming to make no U-turns in its trajectories. The resulting sampling scheme is much more involved than standard HMC, so instead of using their sampling scheme we will tune the hyperparameters of HMC to make few U-turns, emulating their sampler. They define a U-turn to occur when the (squared) distance between $\boldsymbol{x}(t)$ and $\boldsymbol{x}(0)$ goes from increasing to decreasing. That is, when $||\boldsymbol{x}(t) - \boldsymbol{x}(0)||$ goes from increasing to decreasing. In the case of $\boldsymbol{M} = I_K$, this condition is equivalent to

$$\frac{\mathrm{d}||\boldsymbol{x}(t) - \boldsymbol{x}(0)||^2}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\left((\boldsymbol{x}(t) - \boldsymbol{x}(0))'\boldsymbol{x}(t) - \boldsymbol{x}(0)\right) = 2(\boldsymbol{x}(t) - \boldsymbol{x}(0))'\boldsymbol{p}(t),$$

going from positive to negative. With this condition good values for L can be found.
Determining a good choice for ε is less straightforward as it depends strongly on the specific density at hand. When ε is large, the integration error will be large and the acceptance probability, α, will be low. However, when ε is small, each iteration will require more computation. The trade-off between computational time and acceptance probability comes down to determining an optimal average acceptance probability α^*, and finding the ε to match.
To find the ε associated with α^*, we will, just like Hoffman et al. (2014), update ε during the warm-up phase using dual averaging. The principle of dual averaging is as follows: If $\alpha^* - \alpha_s$ is positive then the integration error and ε must be too large, so we decrease ε. If $\alpha^* - \alpha_s$ is negative then our integration error is allowed to be larger (i.e., we increase ε to reduce computational efforts). A simple, but effective updating algorithm is given as follows,

$$\varepsilon_{s+1} = \min\{0.15, \max\{0.001, \varepsilon_s - 0.1 \cdot 0.975^s(\alpha^* - \alpha_s)\}\}.$$

For stability, we enforce that $\varepsilon_s \in [0.001, 0.15]$, as the integration time is otherwise too long, or the integration error too large. The remaining constants were hand-tuned to give good results for MXL.

5. Monte Carlo Study

To assess the effectiveness of estimation the procedures, a simulation study is conducted. What follows is a description of the data generating process and an overview of which sumulations will be run.

5.1. Data Generating Process

As stated previously, we assume that $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, therefore the parameters of the model to be estimated are $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. The magnitude of these parameters is not important, since their magnitude can be negated by letting the features have a different magnitude. It is however desirable for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to have a similar magnitude, since when $\boldsymbol{\Sigma}$ is small compared to $\boldsymbol{\mu}$ there is not enough heterogeneity to justify using an MXL model, and if $\boldsymbol{\Sigma}$ is large compared to $\boldsymbol{\mu}$ it can be difficult to identify $\boldsymbol{\mu}$. So for every MC replicate, we draw $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, the parameters associated with the data set, from $\mathcal{N}(\mathbf{0}, I_K)$ and $\frac{1}{K}\text{IW}(I_K, K)$ respectively, as this will ensure they are of a similar magnitude.
To generate the data sets the following procedure is used: Specify the amount of agents, time periods, alternatives, and features. Then for each agent draw $\boldsymbol{\beta}_n$ from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and draw all x_{ntjk} from a normal distribution to be defined later. Lastly, for each time period draw y_{nt} from a multinomial distribution with $p_j = \frac{\exp(\boldsymbol{\beta}_n' \boldsymbol{x}_{ntj})}{\sum_s \exp(\boldsymbol{\beta}_n' \boldsymbol{x}_{nts})}$.
Letting $\boldsymbol{x}_{ntj}$ follow a normal distribution is a simplifying assumption, since typical datasets used in discrete choice modelling include features which can take on discrete and binary values (Czajkowski and Budziński, 2019). However, since the amount of non-normal features is usually small, and since it is already not straightforward to determine a good normal distribution for $\boldsymbol{x}_{ntj}$ that generates well-behaved datasets, we make this normality assumption. To determine the mean and covariance of this distribution we firstly assume that all features are independent of each other. Making them dependent would reduce the amount of information they contain, but since we will be varying the amount of random coefficients, we do not find it necessary to allow for correlations. Secondly, observe that the choice probabilities do not depend on the mean of the features, because we look at the difference in utilities. So we take x_{ntjk} to have a mean of zero. Thirdly, we assume that the characteristics of choice probabilities should not depend on the number of features, only the number of choices. So, the distribution of $V_j = \boldsymbol{\beta}_n' \boldsymbol{x}_{ntj}$ should not depend on K. However, observe that if we take $x_{ntjk} \sim \mathcal{N}(0, \gamma^2)$, then $\boldsymbol{\beta}_n' \boldsymbol{x}_{ntj} \sim \mathcal{N}(\mathbf{0}, \gamma^2 \boldsymbol{\beta}_n' \boldsymbol{\beta}_n)$. Now, the expected variance is given by,

$$\mathbb{E}\left[\gamma^2 \boldsymbol{\beta}_n' \boldsymbol{\beta}_n\right] = \mathbb{E}\left[\mathbb{E}\left[\gamma^2 \boldsymbol{\beta}_n' \boldsymbol{\beta}_n | \boldsymbol{\mu}, \boldsymbol{\Sigma}\right]\right] = \gamma^2 \mathbb{E}[\text{tr}(\boldsymbol{\Sigma}) + \boldsymbol{\mu}' \boldsymbol{\mu}] = 2K\gamma^2.$$

To remove this dependency on K, and ensure that the expectation of the utility variance is constant we take $x_{ntjk} \sim \mathcal{N}\left(0, \frac{\gamma^2}{K}\right)$. However, the variance of the utilities' variance does still depend on the amount of random coefficients, as when K approaches infinity $\text{var}\left(\frac{\gamma^2}{K} \boldsymbol{\beta}_n' \boldsymbol{\beta}_n\right) \to 0$, so $\frac{1}{K} \boldsymbol{\beta}_n' \boldsymbol{\beta}_n$ degenerates into a constant. As a result the choice probabilities still depend on the amount of random coefficients. A good way to visualize this is by looking at the scaled entropy of the choice probabilities, given by,

$$H(\boldsymbol{p}) = -\frac{1}{\log J} \sum_{j=1}^{J} p_i \log p_i.$$

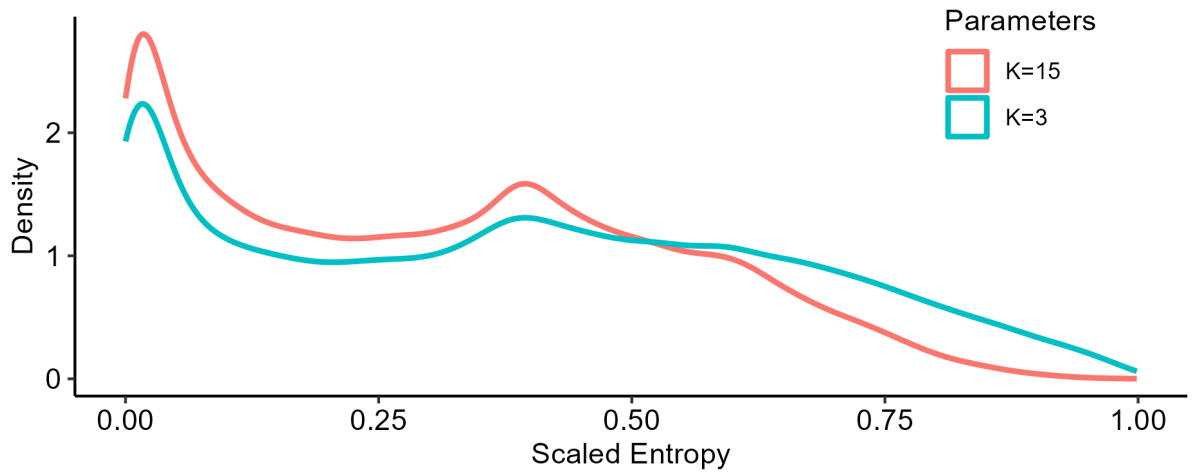

Figure 4: Scaled entropy distributions for $\gamma = 3$.

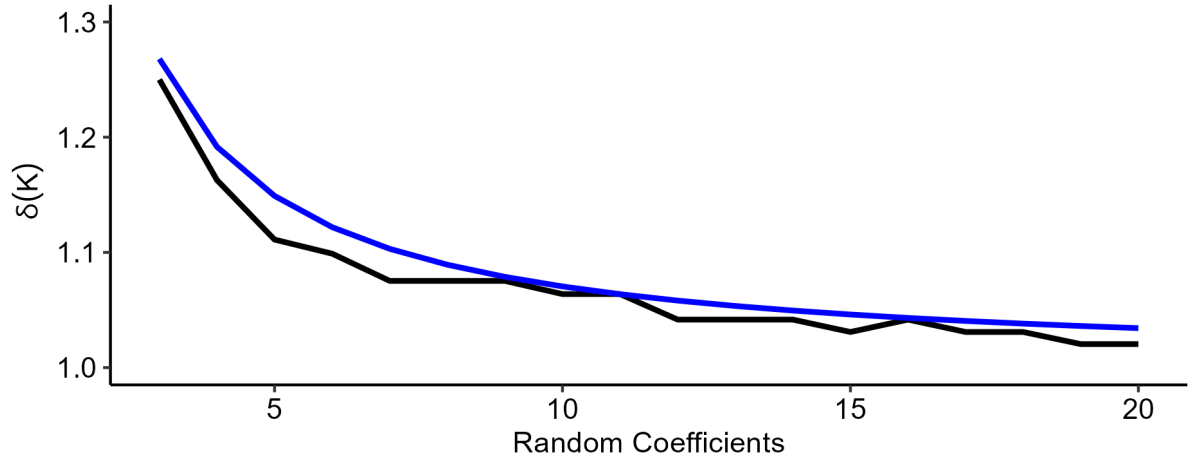

Figure 5: Optimal variance correction factor compared against $\frac{K}{\chi^2_{0.5,K}}$.

The distribution of the scaled entropy is a useful diagnostic tool to assess what the characteristics of the choice probabilities are, and whether they are similar. When the scaled entropy is close to one the choices have uniform probability, and when the scaled entropy is close to zero there is one dominating probability. Figure 4 shows that for smaller K the choice probabilities are more equal. To adjust this we need to increase the variance until the scaled entropy distribution closer resembles the desired distribution. In Figure 5 we have plotted the optimal scaling factor, where Kullback-Leibler divergence was used to determine the distance between distributions. We find that $\delta(K) \approx \frac{K}{\chi^2_{0.5,K}} \approx \left(1 - \frac{2}{9K}\right)^{-3}$. This relation was found by observing that the median might be a better estimator than the mean. Therefore, we take $x_{ntjk} \sim \mathcal{N}\left(0, \frac{\gamma^2}{K}\left(1 - \frac{2}{9K}\right)^{-6}\right)$.

What remains is to pick a good γ. If we pick a small γ then the difference in utilities between the choices will be small. As a result this will make the choice probabilities more uniform, which makes the model less interesting as it is not very different from the null model. If we, however, pick γ to be large, then the difference in utilities between choices will also be large. As a consequence the choice probabilities will be more extreme. This is also not desirable as the model is less sensitive to the parameters. To find a trade of

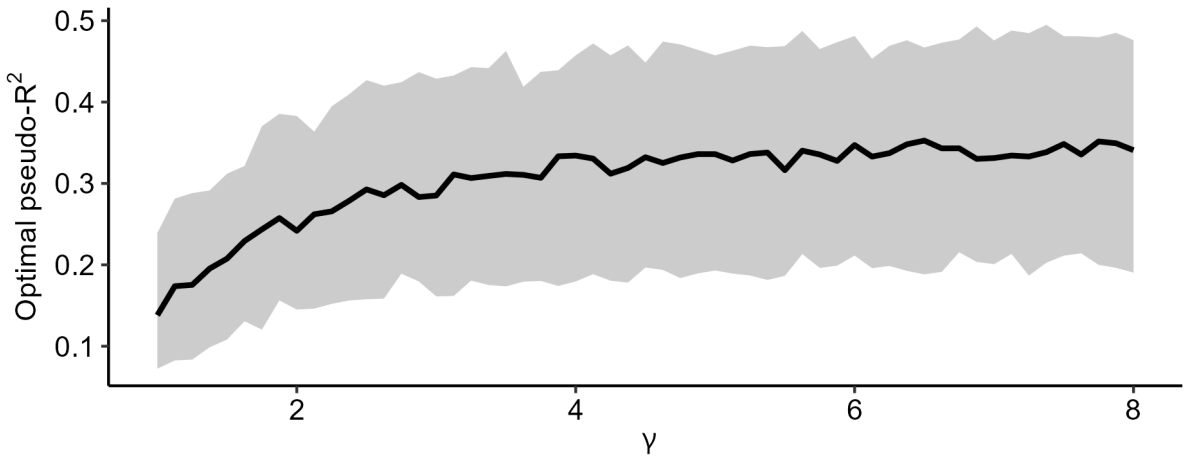

Figure 6: Optimal pseudo-R^2 percentiles.

between these two extremes, we will pick γ such that the pseudo-R^2 of the model is large compared to γ, as this indicates that the model has identifiable information in it. To do this, relatively quickly, for each value of γ we generated 150 datasets and calculated the pseudo-R^2 based on the true parameters. Basing it on the true parameters slightly increases the pseudo-R^2 when compared to the pseudo-R^2 based on the parameters of the estimated model, but this is much faster than using the estimated parameters. From this data we have constructed a 90% confidence interval for the pseudo-R^2, as can be seen in Figure 6. From this we conclude that $\gamma = 3$ is a good value, as the amount of discernible information plateaus at this point. We prefer to take a smaller γ because the model is better behaved in this scenario.
With this data generating process the data sets used in the simulation study were generated. These data sets were chosen to consists of 500 individuals, who make a choice between six options six times, and have up to 20 random coefficients depending on the experiment. This particular size was chosen to balance computational time and size, but it is similar to the data set used by Train (2009) regarding the choice of energy supplier.

5.2. Performance Measures

To assess the performance of HMC we will compare it against RWMH for the entire HB MXL model applied to a generated data set (fully-fledged HB MXL) and the bottle neck density of HB MXL, $\Omega(\boldsymbol{t})$, given by,

$$\Omega(\boldsymbol{t}) = f_{\boldsymbol{\beta}_n|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{y}_n}(\boldsymbol{t}) \propto \phi(\boldsymbol{t}|\boldsymbol{\mu},\boldsymbol{\Sigma})L(\boldsymbol{y}_n|\boldsymbol{t}) = \phi(\boldsymbol{t}|\boldsymbol{\mu},\boldsymbol{\Sigma})\prod_{t=1}^{T}\frac{\exp(\boldsymbol{t}'\boldsymbol{x}_{nty_{nt}})}{\sum_s \exp(\boldsymbol{t}'\boldsymbol{x}_{nts})}.$$

Where $\boldsymbol{\beta}_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ and $\boldsymbol{y}_n$ are associated with the data generation process. To do this comparison, we will first determine good values for the hyperparameters L and α^* as HMC would otherwise not be effective. Moreover, this will only be done for $\Omega(\boldsymbol{t})$ as it would be impractical to do so for the fully-fledged HB MXL.
For good values of L and α^* the time per effective draw is minimal. This is accomplished when the integration error is relatively small and when the Hamiltonian trajectories do not loop back on themselves. As mentioned previously, Hamiltonian trajectories start to loop back on themselves when they make a U-turn. Therefore, to determine good

values for L we will, for various different values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with a small ε, determine the U-turn points. With this data regarding U-turns, values for L can be determined which on average almost have a U-turn.
If α^* is close to one then the effective sample size is large, but the sampler is slow. For small α^* the effective sample size is small, but the sampler is fast. Therefore to determine a good value for α^*, we pick it such that the effective samples per second is maximal. A description of the programs used for this simulation study can be found in the Appendix.

5.3. Results

HMC Hyperparameters

A 90% confidence interval of the time until a U-turn around the average, can be seen in Figure 7. These trajectories are from the stationary phase of the Markov chain and had $T = 6$, $J = 6$, different values of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ to get a good coverage, and a small value of ε for accuracy. As overestimating L is more costly than underestimating it, we will pick $L = \log K$ as integration time for Ω.

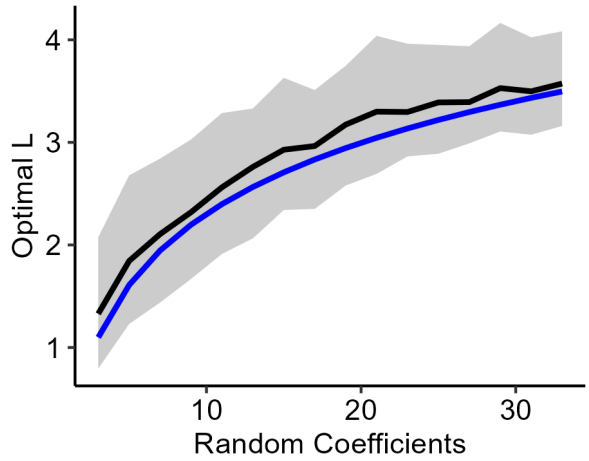

Figure 7: Optimal integration time for Ω compared to $\log K$.

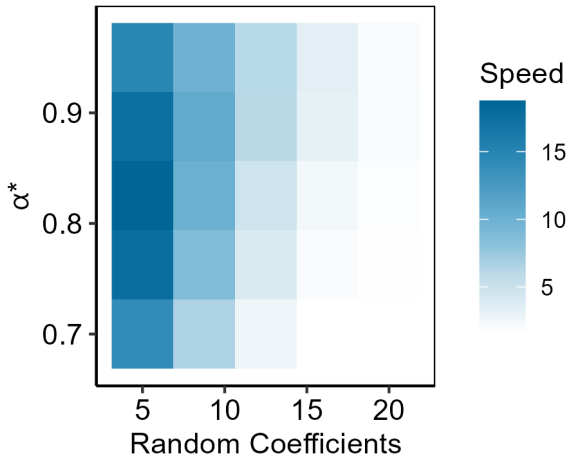

Figure 8: Sampling speed dependence on α^*.

Using $L = \log K$ the optimal target acceptance probability, α^*, can be determined. To do this, we computed the expected amount of effective samples per millisecond (speed) for different values of α^*. As can be seen in Figure 8, $\alpha^* = 0.8$ is better for when K is small and $\alpha^* = 0.9$ is better for larger K. These choices for the hyperparameter result in a well tuned HMC sampler, which emulates the no U-turn sampler.

HMC vs RWMH for Ω

To do a fair comparison between RWMH and HMC the computation time per iteration in will be taken into account. This is because each iteration of RWMH is much faster than an iteration of HMC, but the effective sample size is smaller. The result for this comparison can be found in Figure 9, where a 90% confidence interval for the sampling speed of HMC compared to RWMH around the average is presented. We see that the average is above one, so HMC still produces effective samples more quickly than RWMH.

However, contrary to initial expectation the speed up effect diminishes with increasing dimensions.

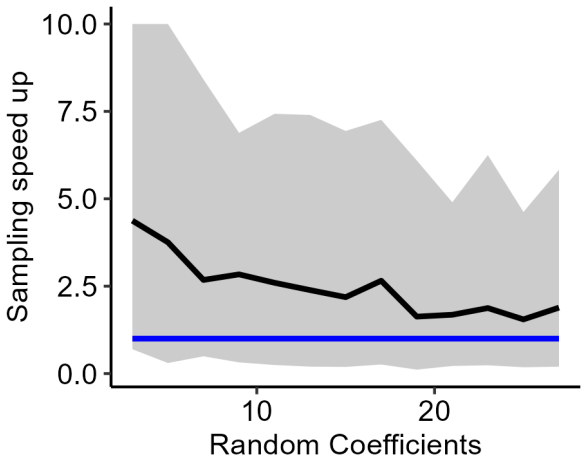

Figure 9: HMC sampling speed up compared to RWMH.

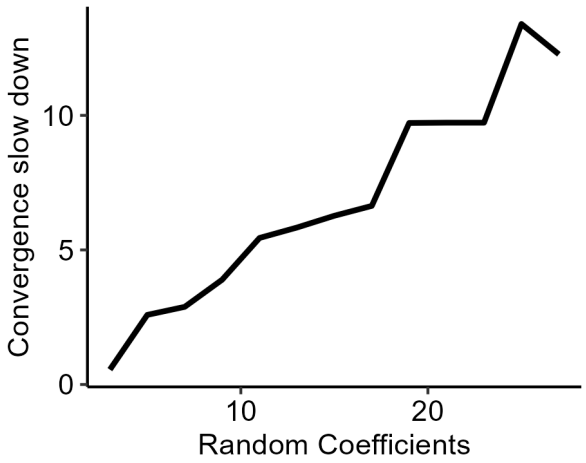

Figure 10: Median time till convergence slowdown of HMC compared to RWMH.

Moreover, according to Figure 10 and 11 RWMH converges more quickly to the critical set than HMC on average, where convergence was assessed using the quantile consistent convergence criteria. This is due to several factors. Firstly, HMC generates samples with negative correlations, this makes the sampler more efficient, but affects the convergence adversely. Secondly, HMC was tuned in the stationary phase, so the choices of L and α^* may not be optimal for the non stationary phase. Thirdly, because the initial value of ε is small we have that the early iterations take longer, making the convergence slower. Lastly, as HMC is a gradient based method, we have that it will perform poorly if the gradient is very large due to limited computer precision. These large gradients are present further away from the critical set, consequently making the convergence rate worse.

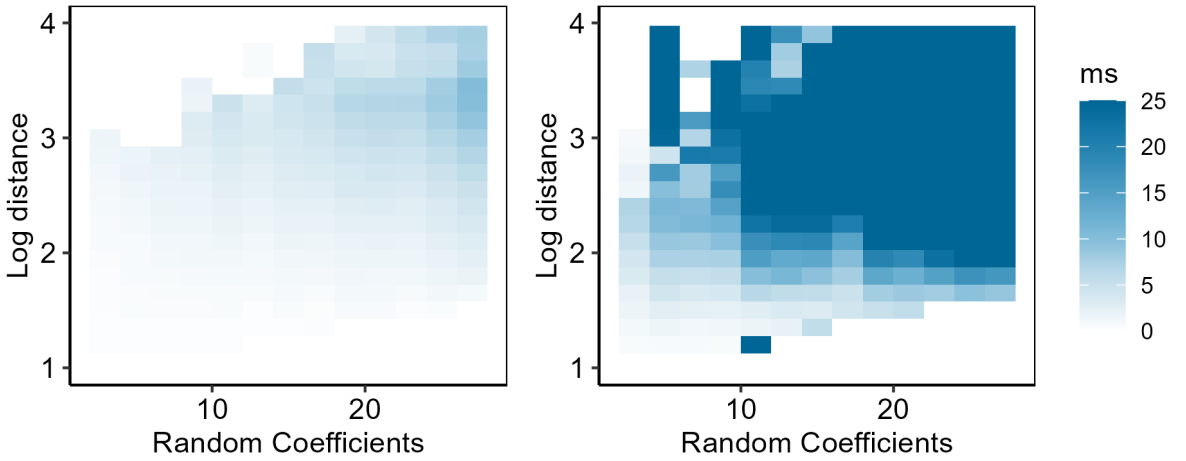

Figure 11: Convergence speed w.r.t log distance and amount of random coefficients

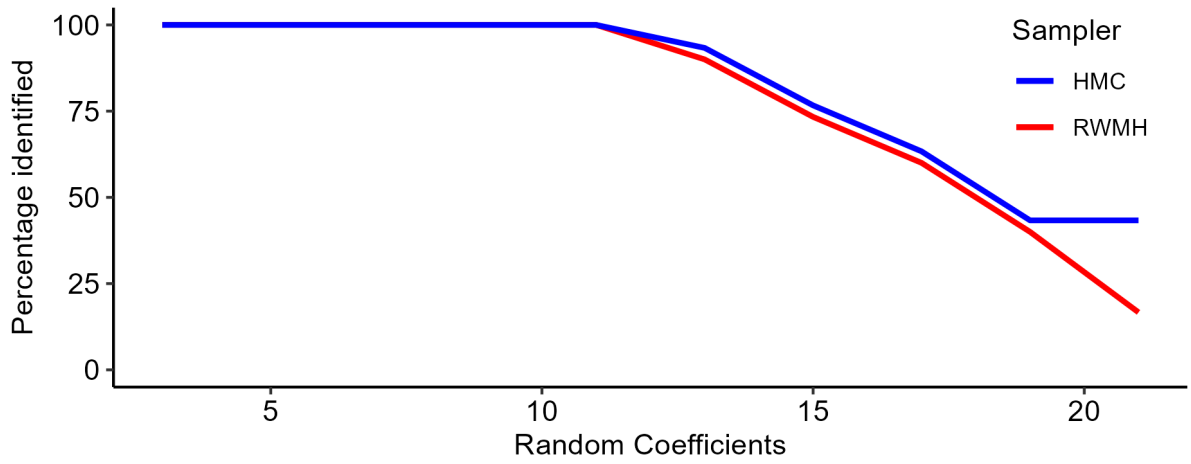

Figure 12: Percentage of identified full HB MXL models

HMC vs RWMH for HB MXL Model

The full HB MXL comparison was done with data sets consisting of 500 individuals who make six choices. This however was sometimes too small for when amount of random coefficients is large, resulting in problems with identifiability. Where identification of the parameters is set to have occurred if the resulting posterior sample has a well-behaved effective sample size (between 0 and 10). In Figure 12, we see that HMC had fewer identifiability issues than than RWMH.

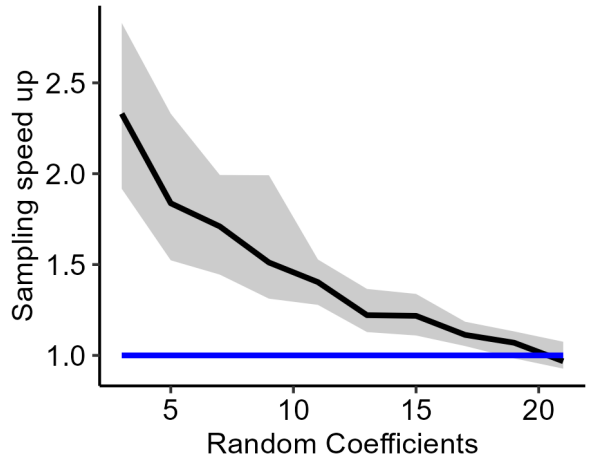

Figure 13: 90% confidence interval, HMC sampling speed up compared to RWMH for full HB MXL model.

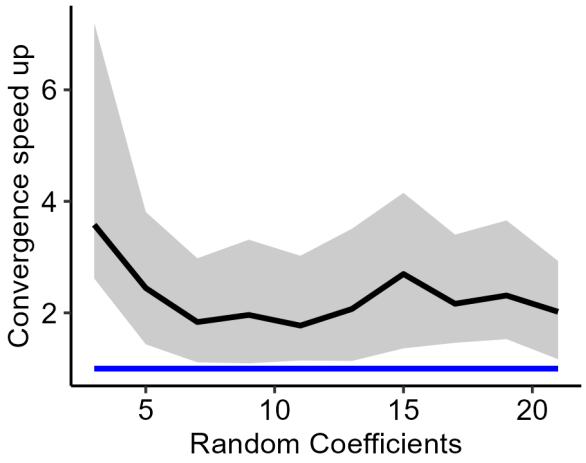

Figure 14: 90% confidence interval, HMC convergence speed up compared to RWMH for full HB MXL model.

For the cases where the model could be identified, the sampling speed and convergence time was compared using a 90% confidence interval. In Figures 13 and 14 we see that HMC is superior to RWMH as it converges more quickly and produces samples more quickly when converged. The sampling speed up is smaller when compared to the speed up for the bottleneck density Ω. This is because both methods also need to execute the other steps of the Gibbs sampler, adding this constant time reduces the speed up factor that HMC introduces.

Convergence time is however shorter for HMC than for RWMH, which is unlike the result for the bottleneck density Ω. This can be attributed to the fact that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ have initially not yet converged to their true values, therefore $\boldsymbol{\beta} = \mathbf{0}$ is within the critical set. In the implementation this was chosen as the starting value for this reason. The result it that the $\boldsymbol{\beta}$'s are already close to the critical set and move along with it as it moves. This is advantageous to HMC as it performs better near to the critical set.

6. Conclusion and Discussion

This thesis investigated the effect of substituting RWMH with HMC on the estimation of HB MXL models with varying amount of random coefficients. With this we aimed to determine the scalability of HMC for MXL models. This is of interest, since the contemporary increase in the availability of data calls for larger MXL models to be estimated efficiently.

The decision to substitute RWMH with HMC was motivated by the fact that HMC has been shown to have superseded RWMH for general sampling, and because previous studies have found that using HMC in Bayesian models can accelerate their estimation.

To investigate the effect of this substitution, extensive Monte Carlo simulations were conducted. The data generating process for these simulations relied on two main assumptions. Firstly, for simplicity, it was assumed that the features that the agents had for each alternative were normally distributed. Secondly, it was assumed that the characteristics of the choice probabilities should be invariant under the amount of random coefficients. This was accomplished by letting the variance of the features depend on the amount random coefficients. This latter assumption was made to ensure that the identifiability of the models in question was not strongly affected by increasing the amount of random coefficients. As when this assumption was not made, the choice probabilities became concentrated in a single choice, resulting in the model becoming less sensitive to the parameters. The generated data sets were chosen to consists of 500 individuals each making six choices. This was chosen as to be representative of other large discrete choice datasets, whilst keeping the estimation times for the models small enough to allow for a large scale Monte Carlo study.

The results of our simulation study demonstrated that HMC outperformed RWMH in terms of estimation speed and convergence for MXL models. However, we observed that as the number of random coefficients increased, the advantage of HMC diminished, which is in contrast to what the existing literature finds for general sampling. It is however uncertain if the diminishing advantage of HMC compared to RWMH is caused by the relatively simplistic implementation of HMC used or is inherent to high-dimensional MXL models. A follow-up study could replicate this thesis using a more sophisticated implementation of HMC to determine the underlying cause. The probabilistic programming language Stan (Stan Development Team, 2018), which has been utilized in other studies involving HMC, could be employed due to its highly performant HMC implementation.

References

Akinc, Deniz and Martina Vandebroek (2018). Bayesian estimation of mixed logit models: Selecting an appropriate prior for the covariance matrix. *Journal of choice modelling 29*, 133–151.

Betancourt, Michael (2016). Identifying the optimal integration time in Hamiltonian Monte Carlo. *arXiv preprint arXiv:1601.00225*.

Betancourt, Michael (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.

Brooks, Steve, Andrew Gelman, Galin Jones, and Xiao-Li Meng (2011). *Handbook of Markov chain Monte Carlo*. CRC press.

Brooks, Stephen P and Andrew Gelman (1998). General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics 7*(4), 434–455.

Chen, Yuansi, Raaz Dwivedi, Martin J Wainwright, and Bin Yu (2020). Fast mixing of Metropolized Hamiltonian Monte Carlo: Benefits of multi-step gradients. *The Journal of Machine Learning Research 21*(1), 3647–3717.

Cowles, Mary Kathryn and Bradley P Carlin (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association 91*(434), 883–904.

Czajkowski, Mikołaj and Wiktor Budziński (2019). Simulation error in maximum likelihood estimation of discrete choice models. *Journal of choice modelling 31*, 73–85.

Fink, Daniel (1997). A compendium of conjugate priors. *See http://www. people. cornell. edu/pages/df36/CONJINTRnew% 20TEX. pdf 46*.

Frank, Jason (2008). Symplectic flows and maps and volume preserving methods.

Fukunaga, Keinosuke (2013). *Introduction to statistical pattern recognition*. Elsevier.

Gelman, Andrew and Donald B Rubin (1992). Inference from iterative simulation using multiple sequences. *Statistical science*, 457–472.

Geyer, Charles J (2003). The Metropolis-Hastings-Green algorithm. Technical report.

Green, Peter J (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*(4), 711–732.

Hastings, W Keith (1970). Monte Carlo sampling methods using Markov chains and their applications.

Hoffman, Matthew D, Andrew Gelman, et al. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res. 15*(1), 1593–1623.

Huber, Joel and Kenneth Train (2001). On the similarity of classical and Bayesian estimates of individual mean partworths. *Marketing Letters 12*, 259–269.

McFadden, Daniel (1974). *Conditional logit analysis of qualitative choice behavior*. New York: Academic press.

McFadden, Daniel and Kenneth Train (2000). Mixed MNL models for discrete response. *Journal of applied Econometrics 15*(5), 447–470.

Metropolis, Nicholas, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics 21*(6), 1087–1092.

Monnahan, Cole C, James T Thorson, and Trevor A Branch (2017). Faster estimation of bayesian models in ecology using Hamiltonian Monte Carlo. *Methods in Ecology and Evolution 8*(3), 339–348.

Niederreiter, Harald (1992). *Random number generation and quasi-Monte Carlo methods*. SIAM.

Nylander, Johan AA, James C Wilgenbusch, Dan L Warren, and David L Swofford (2008). Awty (are we there yet?): a system for graphical exploration of MCMC convergence in Bayesian phylogenetics. *Bioinformatics 24*(4), 581–583.

Roberts, Gareth O and Jeffrey S Rosenthal (2004). General state space Markov chains and MCMC algorithms.

Ross, S.M. (2019). *Introduction to probability models*. Elsevier Science.

Stan Development Team (2018). The Stan Core Library. Version 2.18.0.

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

Van der Vaart, Aad W (2000). *Asymptotic statistics*, Volume 3. Cambridge university press.

Vats, Dootika (2023). Hamiltonian Monte Carlo for (physics) dummies. Technical report.

Vats, Dootika, James M Flegal, and Galin L Jones (2019). Multivariate output analysis for Markov chain Monte Carlo. *Biometrika 106*(2), 321–337.

Vats, Dootika and Christina Knudson (2021). Revisiting the Gelman-Rubin diagnostic. *Statistical Science 36*(4), 518–529.

Yamada, Taisuke, Keitaro Ohno, and Yusaku Ohta (2022). Comparison between the Hamiltonian Monte Carlo method and the Metropolis-Hastings method for coseismic fault model estimation. *Earth, Planets and Space 74*(1), 86.

Appendix

Properties of Hamiltonian dynamics

Proofs of the used properties of Hamiltonian dynamics.

Conservation of the Hamiltonian

The Hamiltonian is conserved if $\mathcal{H}(\boldsymbol{x}, \boldsymbol{p})$ does not change over time, that is, its derivative w.r.t. t is zero. This follow directly from the chain rule and the Hamiltonian equations.

$$\begin{aligned}\frac{\mathrm{d}\mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\mathrm{d}t} &= \sum_{k=1}^{K} \left(\frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial x_k} \frac{\partial x_k}{\partial t} + \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial p_k} \frac{\partial p_k}{\partial t} \right) \\ &= \sum_{k=1}^{K} \left(\frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial x_k} \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial p_k} - \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial p_k} \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial x_k} \right) = 0\end{aligned}$$

Time Reversibility

The Hamiltonian equations are time reversible if when the momentum is negated it follows the path back in time. To show this we consider the mapping $\boldsymbol{p} \mapsto -\boldsymbol{p}$, so we negate the momentum, then the Hamiltonian equations are given by,

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{\partial \mathcal{H}(\boldsymbol{x}, -\boldsymbol{p})}{\partial(-p_i)}, \quad \frac{\mathrm{d}(-p_i)}{\mathrm{d}t} = -\frac{\partial \mathcal{H}(\boldsymbol{x}, -\boldsymbol{p})}{\partial x_i} \text{ for all } i \in \{1, \ldots, K\}.$$

Now observe that in our case,

$$\mathcal{H}(\boldsymbol{x}, -\boldsymbol{p}) = -\log\left(\zeta(\boldsymbol{x})\right) + \frac{1}{2}(-\boldsymbol{p})' M^{-1}(-\boldsymbol{p}) = -\log\left(\zeta(\boldsymbol{x})\right) + \frac{1}{2}\boldsymbol{p}' M^{-1}\boldsymbol{p} = \mathcal{H}(\boldsymbol{x}, \boldsymbol{p}).$$

Therefore the Hamiltonian equations become,

$$\frac{\mathrm{d}x_i}{\mathrm{d}(-t)} = \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial p_i}, \quad \frac{\mathrm{d}p_i}{\mathrm{d}(-t)} = -\frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial x_i} \text{ for all } i \in \{1, \ldots, K\}.$$

Now since these equations are the same as the normal Hamiltonian equation but with respect to $-t$ the system will indeed be time reversible.

Volume Preservation

To show that $\det\left(\mathrm{D}T_s(\boldsymbol{x}, \boldsymbol{p})\right) = 1$ we express Frank (2008) their argument with more elementary mathematics. Let $f : \mathbb{R}^{2K} \to \mathbb{R}^{2K}$ be given by,

$$f(\boldsymbol{x}, \boldsymbol{p}) = \left(\frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial p_1}, \ldots, \frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial p_K}, -\frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial x_1}, \ldots, -\frac{\partial \mathcal{H}(\boldsymbol{x}, \boldsymbol{p})}{\partial x_K} \right).$$

Then because $T_s(\boldsymbol{x}, \boldsymbol{p})$ is governed by Hamiltonian dynamics we have by definition that,

$$\frac{\mathrm{d}}{\mathrm{d}s} T_s(\boldsymbol{x}, \boldsymbol{p}) = f(T_s(\boldsymbol{x}, \boldsymbol{p})).$$

This will imply the following,

$$\begin{aligned}
&\frac{\partial}{\partial(\boldsymbol{x},\boldsymbol{p})}\left(\frac{\mathrm{d}}{\mathrm{d}s}T_s(\boldsymbol{x},\boldsymbol{p})\right) = \frac{\partial}{\partial(\boldsymbol{x},\boldsymbol{p})}\left(f(T_s(\boldsymbol{x},\boldsymbol{p}))\right)\\
\Rightarrow &\frac{\mathrm{d}}{\mathrm{d}s}\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p}) = \mathrm{D}f(T_s(\boldsymbol{x},\boldsymbol{p}))\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})\\
\Rightarrow &\left(\frac{\mathrm{d}}{\mathrm{d}s}\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})\right)\left(\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})\right)^{-1} = \mathrm{D}f(T_s(\boldsymbol{x},\boldsymbol{p}))\\
\Rightarrow &\mathrm{Tr}\left(\left(\frac{\mathrm{d}}{\mathrm{d}s}\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})\right)\left(\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})\right)^{-1}\right) = \mathrm{Tr}\left(\mathrm{D}f(T_s(\boldsymbol{x},\boldsymbol{p}))\right).
\end{aligned}$$

Now observe that,

$$\mathrm{Tr}\left(\mathrm{D}f(T_s(\boldsymbol{x},\boldsymbol{p}))\right) = \sum_{k=1}^{K}\left(\frac{\partial\mathcal{H}(\boldsymbol{x},\boldsymbol{p})}{\partial p_k}\frac{\partial p_k}{\partial t} + \frac{\partial\mathcal{H}(\boldsymbol{x},\boldsymbol{p})}{\partial x_k}\frac{\partial x_k}{\partial t}\right) = \frac{\mathrm{d}\mathcal{H}(\boldsymbol{x},\boldsymbol{p})}{\mathrm{d}t} = 0.$$

Moreover, recall the following identity known as Jacobi's formula,

$$\frac{\mathrm{d}}{\mathrm{d}s}\det(A(s)) = \det(A(s))\cdot\mathrm{Tr}\left(A(s)^{-1}\frac{\mathrm{d}}{\mathrm{d}s}A(s)\right).$$

Lastly, note that,

$$T_s(\boldsymbol{x},\boldsymbol{p}) = \begin{pmatrix}\boldsymbol{x}\\ \boldsymbol{p}\end{pmatrix} + sf(T_s(\boldsymbol{x},\boldsymbol{p})) + O(s^2) \Rightarrow \mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p}) = I_{2K} + s\mathrm{D}f(T_s(\boldsymbol{x},\boldsymbol{p})) + O(s^2).$$

So, $\mathrm{D}T_0(\boldsymbol{x},\boldsymbol{p}) = I_{2K}$. With the previous results the proof can be finished,

$$\begin{aligned}
&\mathrm{Tr}\left(\left(\frac{\mathrm{d}}{\mathrm{d}s}\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})\right)\left(\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})\right)^{-1}\right) = \frac{\frac{\mathrm{d}}{\mathrm{d}s}\det(\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p}))}{\det(\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p}))} = 0\\
\Rightarrow &\frac{\mathrm{d}}{\mathrm{d}s}\det(\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})) = 0 \Rightarrow \det(\mathrm{D}T_s(\boldsymbol{x},\boldsymbol{p})) = \det(\mathrm{D}T_0(\boldsymbol{x},\boldsymbol{p})) = 1.
\end{aligned}$$

For physicists this means that $T_s(\boldsymbol{x},\boldsymbol{p})$ is volume preserving.

Code documentation

The HB MXL estimation was done using a custom C++ program. The code consists of the following source files:

- main.cpp is the control unit. From main all the experiments are set up.
- MSL.cpp contains a deprecated implementation of MSL. It uses the BFGS optimizer, but has rudimentary line search so it is not very effective.
- DSL.cpp is 'Daan's Sampling Library', it contains many fast and vetted pseudo random number generators. It can generate variates from several univariate, categorical and multivariate distributions.
- NUTS.cpp implements sampler functions for the density Ω.
- DGP.cpp contains the data generating process.
- HB.cpp contains the main HB MXL estimation functions, it is based on the specification Train (2009) gives. It also contains a function which can compute the pseudo-R^2 using Halton sequences.

The data analysis was performed in R.