Bayesian inference with Biogeme

Michel Bierlaire

November 20, 2025

Report TRANSP-OR xxxxxx
Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

Series on Biogeme

Contents

The package Biogeme (`biogeme.epfl.ch`) is designed to estimate the parameters of various models. It is particularly designed for discrete choice models. Originally designed to use maximum likelihood estimation, it is also possible to use Bayesian inference to estimate the parameters of the model. It is particularly useful for mixtures models, as it allows to avoid the calculation of complex Monte-Carlo integrals.

We assume that the reader is already familiar with discrete choice models, and has successfully installed Biogeme. This document has been written using Biogeme 3.3.2.

It is also highly recommended to review foundational concepts such as simulation methods, Bayesian inference, and Markov chain Monte Carlo. Although these topics are briefly introduced here, a solid understanding of them greatly helps in fully appreciating the power and flexibility of Bayesian estimation for discrete choice models.

1 Bayesian inference

Bayesian inference consists of fitting a probabilistic model to observed data and representing the outcome by a probability distribution over the model parameters.

Bayesian inference differs fundamentally from frequentist inference in the way uncertainty about model parameters is represented and quantified. In the frequentist framework, parameters are treated as fixed but unknown constants, and uncertainty arises solely from the randomness of the data-generating process. In contrast, the Bayesian approach treats parameters as random variables endowed with a prior distribution, which encodes the information available before observing the data. This modeling choice does not imply that parameters are intrinsically random, but rather reflects epistemic uncertainty: the distribution represents our state of knowledge about plausible parameter values given the information at hand. After observing data, Bayes' theorem updates this prior into a posterior distribution, which synthesizes both prior beliefs and empirical evidence. The posterior distribution is therefore the central object of Bayesian inference, providing coherent measures of uncertainty, enabling probabilistic predictions, and allowing for direct probability statements about parameters themselves.

Consider a discrete choice model characterized by a vector of parameters $\boldsymbol{\theta}$ and a likelihood function $\mathrm{L}(\mathcal{D} \mid \boldsymbol{\theta})$, where $\mathcal{D}$ denotes the observed data: for each individual in the sample, it contains the values of the explanatory variables as well as the observed choice. The likelihood function represents the probability that the model, with parameters $\boldsymbol{\theta}$, reproduces exactly all the observations in the sample[1].

In the frequentist framework, estimation consists of finding a point estimate $\hat{\boldsymbol{\theta}}$ that maximizes the likelihood or the log-likelihood. In the Bayesian framework, however, the parameters are treated as unknown quantities described by a prior density $\mathrm{p}(\boldsymbol{\theta})$, reflecting the information available before observing the data.

Once the data are observed, inference is performed through Bayes' theorem, which combines the prior with the likelihood to obtain the posterior distribution of the parameters:

$$\mathrm{p}(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{\mathrm{L}(\mathcal{D} \mid \boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{\theta})}{\int \mathrm{L}(\mathcal{D} \mid \boldsymbol{\theta}')\, \mathrm{p}(\boldsymbol{\theta}')\, \mathrm{d}\boldsymbol{\theta}'}. \tag{1}$$

The denominator ensures that the posterior integrates to one. The bad news is that it is not available in closed form for choice models. The good news is that it is not needed in practice.

This distribution is inherently difficult to work with analytically. For that reason, we rely on simulation to generate realizations — referred to as *draws* — from it. Before explaining how such draws can be produced in practice, we first provide some intuition for why simulation is necessary.

[1] Rigorously, this interpretation holds when the model involves only discrete variables. When the model includes continuous variables, the likelihood is obtained by evaluating the joint probability *density* of the data at the observed values. Although this quantity is not itself a probability, it plays an analogous role.

2 Simulation

The arithmetic of random variables can quickly become intricate. Even in the simple case of two independent random variables X and Y, with respective probability density functions (pdf) f_X and f_Y, the distribution of their sum is not straightforward. If we define $Z = X + Y$, the pdf of Z is obtained through a transformation known as *convolution*:

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x)\, f_Y(z - x)\, dx.$$

For instance, assume that both X and Y follow a uniform distribution:

$$X \sim U(0,1), \qquad Y \sim U(0,1).$$

Then, it can be shown that Z follows a triangular distribution:

$$f_Z(z) = \begin{cases} 0, & z < 0, \\ z, & 0 \leqslant z \leqslant 1, \\ 2 - z, & 1 < z \leqslant 2, \\ 0, & z > 2. \end{cases}$$

However, in practice, the convolution integral rarely has a closed form, making it difficult to handle.

The idea of simulation consists in generating concrete numerical values produced according to the probability law of the random variables of interest. Regular arithmetic can then be applied on those values.

Let X be a random variable with probability density function (pdf) f_X. A *draw from* X is a numerical value obtained from a random mechanism whose outcomes follow exactly the distribution of X.

Formally, consider a sequence of independent draws $X_1, X_2, \ldots, X_R$ from X. For any fixed R, the empirical distribution of these draws can be represented by a histogram. As R becomes large, the histogram provides an increasingly accurate approximation of the true pdf f_X.

More precisely, for any interval $[a, b]$,

$$\frac{1}{R} \sum_{i=1}^{R} \mathbf{1}\{X_i \in [a, b]\} \xrightarrow[R \to \infty]{\text{a.s.}} \int_a^b f_X(x)\, dx, \tag{2}$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function, and "a.s." stands for almost surely, meaning that the convergence holds with probability 1. This property demonstrates that the draws reproduce the probability structure of X: the relative frequency with which the draws fall in any region converges to the probability mass assigned to that region by the pdf f_X.

In this sense, a draw from X is not merely a number, but a realization generated according to f_X, and repeated draws allow us to recover the shape of the density through their empirical distribution.

This is illustrated in Figure 1, which displays histograms of 100'000 independent draws from two uniform random variables $X \sim U(0,1)$ and $Y \sim U(0,1)$, together with the histogram of their sum $Z = X + Y$. The first two panels show that the empirical distributions of X and Y closely match the flat density of the uniform distribution. The third panel presents the resulting distribution of Z, whose histogram approaches the theoretical triangular density obtained by the convolution of the two uniforms. This confirms that, as the number of draws increases, the simulated empirical distributions converge to their corresponding probability density functions.

Most programming languages and numerical libraries provide a built-in function for generating draws from the uniform distribution $U(0,1)$. Although the sequence returned by such a

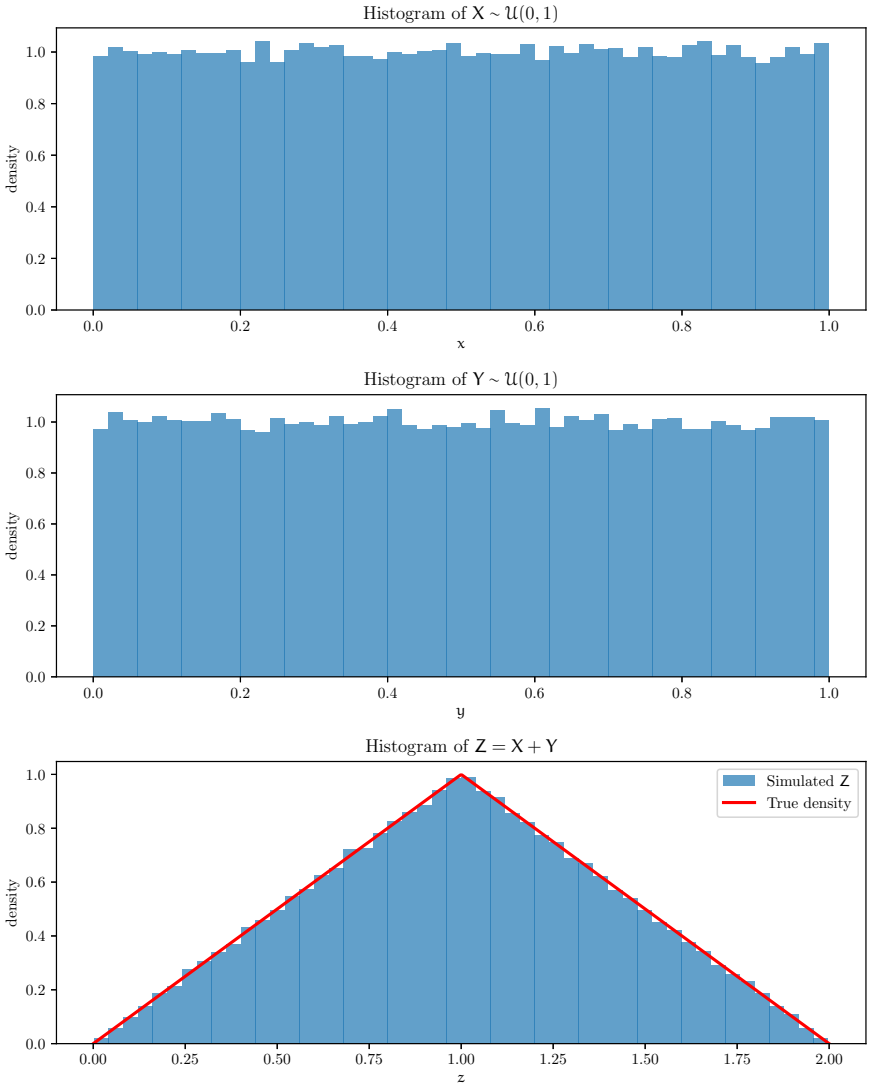

Figure 1: Histograms of X, Y, and $\mathsf{Z} = \mathsf{X} + \mathsf{Y}$ with theoretical density of Z.

function is deterministic (a pseudo-random number generator), it nonetheless exhibits all the statistical properties of a truly random sequence — in particular the convergence property (2). Once we have uniform draws, a variety of simple algorithms exist for transforming them into draws from other distributions (normal, extreme value, gamma, etc.) For a comprehensive treatment of these methods, the reader is referred to the standard text by Ross (2012).

Unfortunately, sampling from the posterior distribution (1) of the parameters of a choice model cannot be achieved through simple transformations of uniform draws. Instead, it requires more advanced simulation techniques, known as Markov chain Monte–Carlo (MCMC) methods. As those methods are the core of Bayesian inference, we provide a brief introduction in the next section. We invite the interested reader to consult the literature for a more comprehensive description (Wang, 2022, **?**).

3 Markov Chain Monte–Carlo methods

The term *Monte-Carlo* refers to the city of Monte-Carlo in the Principality of Monaco, famous for its casino. In mathematics and statistics, the expression "Monte-Carlo" is used whenever randomness is used as a computational tool, typically to approximate integrals, expectations, or probability distributions.

A *Markov chain* is a stochastic process, that is, a sequence of random variables, with specific mathematical properties that make their long-run behavior analytically tractable. Under appropriate conditions (irreducibility, aperiodicity, and positive recurrence), a Markov chain converges to a distribution called its *stationary distribution*. Intuitively, *stationarity* means that, once the chain has run long enough, the distribution of its state no longer changes over time. Rigorously, if X_t denotes the state at iteration t, then stationarity means that there exists a distribution π such that

$$\Pr(X_{t+1} = j) = \Pr(X_t = j) = \pi_j \qquad \text{for all states } j \text{ and all } t \text{ large enough.}$$

Equivalently, if the chain is initiated with $X_0 \sim \pi$, then all future states X_t also follow the same distribution.

The idea behind *Markov Chain Monte-Carlo* (MCMC) methods is to construct a Markov chain whose stationary distribution is precisely the distribution from which we wish to draw samples (for example, the posterior distribution of model parameters). By simulating the chain for a sufficiently large number of iterations, the generated sequence approximates draws from the target distribution.

Formally, a Markov chain $(X_t)_{t \geqslant 0}$ is defined on a state space (which may be discrete or continuous) together with a *transition probability*. For simplicity, we introduce the concept in the discrete case; the continuous version is entirely analogous, with probability density functions replacing probabilities, and integrals replacing sums.

For each pair of states i and j, the transition probability is

$$P_{ij} = \Pr(X_{t+1} = j \mid X_t = i). \tag{3}$$

A key property of Markov chains is that the transition probabilities do not depend on the iteration index t. Moreover, for each state i,

$$\sum_j P_{ij} = 1,$$

so that P_{ij} defines a proper probability distribution over the next state.

A stationary distribution is a vector $\pi = (\pi_j)_j$ satisfying the system

$$\pi_j = \sum_i \pi_i P_{ij} \qquad \text{for all states } j, \tag{4}$$

with the normalization condition

$$\sum_{j} \pi_j = 1. \tag{5}$$

Equation (4) states that if X_t has distribution π, then X_{t+1} also has distribution π. Thus the chain is in equilibrium.

In many MCMC algorithms, the Markov chains used to generate samples satisfy an additional property known as *time reversibility*. A chain is time reversible with respect to a distribution π if

$$\pi_i P_{ij} = \pi_j P_{ji} \qquad \text{for all } i \neq j. \tag{6}$$

This condition is also known as *detailed balance*, and implies that π is stationary. Indeed, summing (6) over all i directly yields (4). Many classical MCMC algorithms, such as the Metropolis–Hastings method, are explicitly designed to satisfy detailed balance with respect to the target distribution.

We illustrate the notion of stationary and time-reversible Markov chains with a simple example involving customer engagement on an online service (e.g., a subscription-based platform).

We consider a single user observed once per day. On any given day, the user is in exactly one of the following three engagement states:

- State 1: low engagement (rarely logs in, uses very few features),
- State 2: medium engagement (uses the service somewhat regularly),
- State 3: high engagement (uses the service intensively and frequently).

We assume that the evolution of the user's engagement from day to day can be modeled as a homogeneous Markov chain $(X_t)_{t \geqslant 0}$ taking values in $\{1, 2, 3\}$, with the following transition matrix:

$$P = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.6 \end{pmatrix}.$$

Each entry P_{ij} denotes the probability that the user moves from state i on day t to state j on day $t + 1$. The entries of P can be read as follows:

- From low engagement (state 1):
  - the user stays low-engagement the next day with probability 0.7,
  - moves up to medium engagement with probability 0.2,
  - jumps directly to high engagement with probability 0.1.
- From medium engagement (state 2):
  - the user drops to low engagement with probability 0.2,
  - remains at medium engagement with probability 0.5,
  - increases to high engagement with probability 0.3.
- From high engagement (state 3):
  - the user cools down to medium engagement with probability 0.3,
  - remains highly engaged with probability 0.6,
  - drops directly to low engagement with probability 0.1.

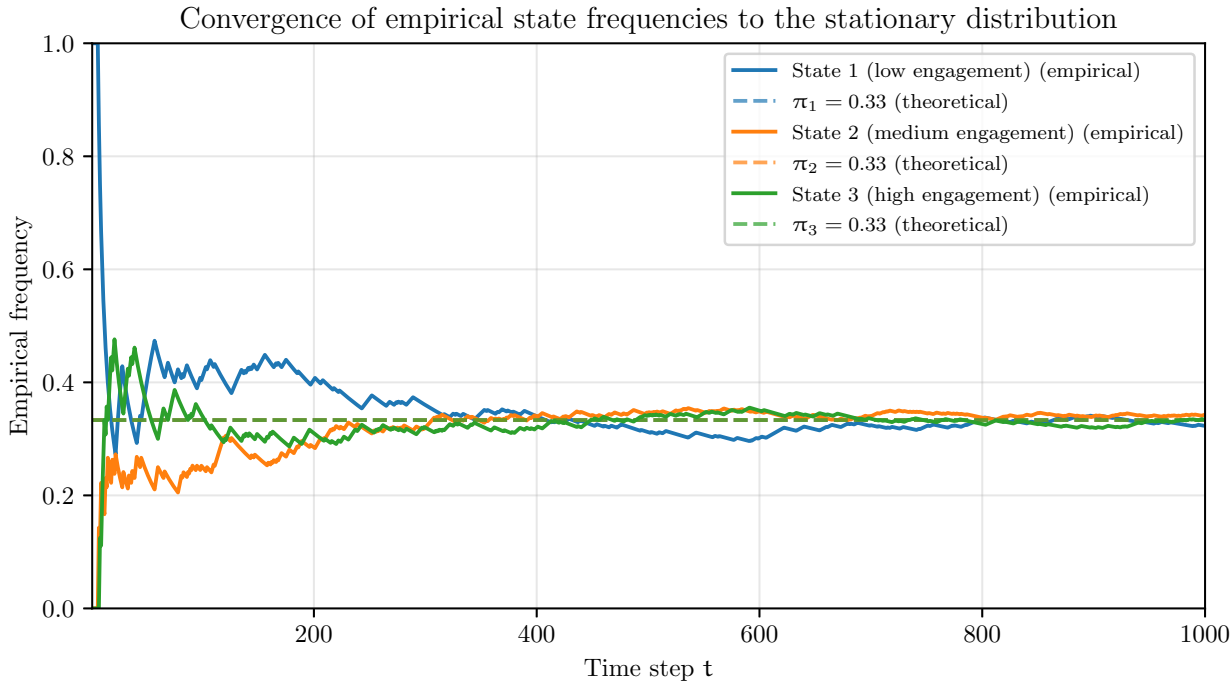

Figure 2: Simulation of the three–state Markov chain (t=0,. . . , 1000)

It is easy to verify that the Markov chain admits the uniform stationary distribution

$$\pi = (\pi_1, \pi_2, \pi_3) = \left(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}\right).$$

The Markov chain is also time-reversible with respect to π. This follows from the fact that $\pi_i = \pi_j = 1/3$ for all i, j, and that P is symmetric

The behavior of the Markov chain introduced above is illustrated in Figure 2. The figure displays the empirical frequency of each state as the simulation evolves over time. At the beginning of the run, these empirical frequencies fluctuate widely and do not yet reflect the target distribution. As the number of iterations increases, however, the proportions stabilize and gradually approach the theoretical stationary distribution $\pi = (1/3, 1/3, 1/3)$ derived earlier. A crucial practical implication is that the draws generated during the early iterations—before the chain has approached stationarity—should not be used as representative samples from the target distribution. Only after the chain has "settled" near equilibrium do the simulated states behave as valid draws from the desired stationary distribution. Typically, in this example, we would simply discard all the 1000 draws displayed in Figure 2, and start using the chain to generate more draws (Figure 3 illustrates the chain from step 1000 to step 2000).

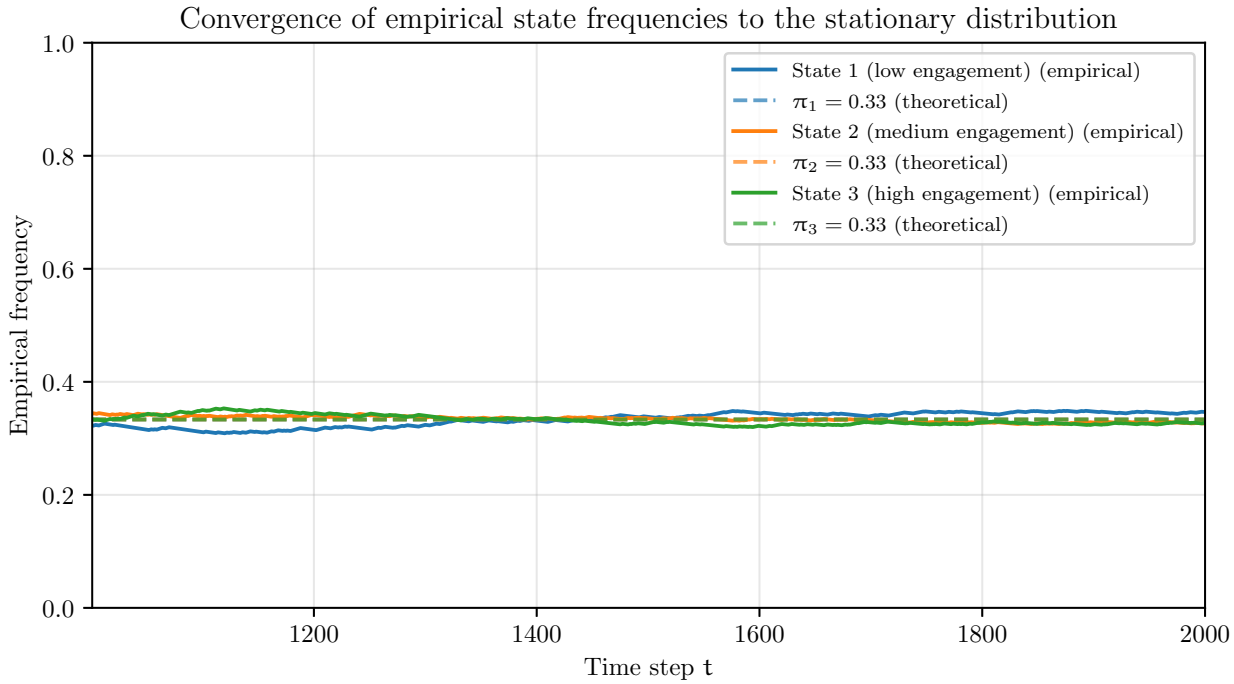

Figure 3: Simulation of the three–state Markov chain (t=1000,. . . , 2000)

References

Ross, S. (2012). *Simulation*, fifth edition edn, Academic Press.

Wang, W. (2022). An introduction to the markov chain Monte Carlo method, *American Journal of Physics* **90**(12): 921–934. DOI: 10.1119/5.0122488.