Stochastic adaptive resampling for the estimation of discrete choice models

Nicola Ortelli

Matthieu de Lapparent

Michel Bierlaire

STRC conference paper 2023 **June 6, 2023**

STRC | **23rd Swiss Transport Research Conference**
Monte Verità / Ascona, May 10-12, 2023

Stochastic adaptive resampling for the estimation of discrete choice models

Nicola Ortelli, Matthieu de Lapparent
School of Management and Engineering Vaud
HES-SO
Yverdon-les-Bains, Switzerland
nicola.ortelli@heig-vd.ch

Nicola Ortelli, Michel Bierlaire
Transport and Mobility Laboratory
EPFL
Lausanne, Switzerland

June 6, 2023

Abstract

In the field of choice modeling, the availability of ever-larger datasets has the potential to significantly expand our understanding of human behavior, but this prospect is limited by the poor scalability of discrete choice models (DCMs): as sample sizes increase, the computational cost of maximum likelihood estimation quickly becomes intractable for anything but trivial model structures. To tackle this issue, this study builds upon the work of Lederrey *et al.* (2021) and the adaptive batch size algorithm they propose for the estimation of DCMs. Specifically, we investigate the use of a dataset reduction technique to generate weighted batches that better represent the whole dataset and, as a result, lead the optimization algorithm to faster convergence. We use a real-world dataset and models of different sizes to compare the performance of our approach with existing methods used in practice.

Keywords

discrete choice models, maximum likelihood estimation, stochastic optimization, dataset reduction

Suggested Citation

Ortelli, N., de Lapparent, M., Bierlaire, M. (2023). Stochastic adaptive resampling for the estimation of discrete choice models, 23rd Swiss Transport Research Conference, Ascona, Switzerland.

1 Introduction

Big data has caused a surge in the amount of data collected on practically any object of study. In the field of discrete choice analysis, the availability of these ever-larger datasets could improve our understanding of human decision-making, but that prospect is limited by the poor scalability of estimation methods for discrete choice models (DCMs).

DCMs are usually estimated via maximum likelihood estimation, which most often relies on optimization algorithms such as Newton's method, BFGS (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970), or one of their variations. These algorithms are extremely effective when estimating simple models on small datasets, but they quickly become computationally expensive as model complexity and dataset sizes grow. To circumvent this limitation, Lederrey *et al.* (2021) recently proposed using stochastic approximations of these methods to estimate DCMs. Similar to the stochastic gradient descent method used to train neural networks, a crucial feature of the algorithms developed by Lederrey *et al.* (2021) is the use of subsets of data — or *batches* — of increasing size throughout the optimization process: at each iteration, a new batch is randomly drawn whose size is determined according to the advancement of the process, until the full dataset is eventually reached and the algorithm converges to the maximum likelihood estimates of the model parameters. Lederrey *et al.* (2021) empirically demonstrate that the use of batches in the earlier stages of the optimization significantly contribute to reducing the total computational time of model estimation.

This study builds upon the idea of using batches and adaptive batch sizes for the estimation of DCMs. Namely, we propose a procedure called *stochastic adaptive resampling* — or *STAR* — that leverages the dataset reduction technique proposed by Ortelli *et al.* (2022, 2023) to generate batches of weighted observations that mimic the full dataset. Our procedure follows an adaptive batch-size updating scheme that relies on the performance of the optimization algorithm to select appropriate batch sizes at each iteration. In doing so, we seek to better guide the optimization algorithm during its earlier stages, while maintaining a low computational cost per iteration. While further confirmatory experiments are still needed, our method could theoretically offer significant time savings irrespective of the iterative optimization algorithm it is used in conjunction with.

The rest of this paper is organized as follows: Section 2 describes how the STAR procedure works; Section 3 presents some preliminary results obtained by using our procedure to estimate three logit models on a relatively large dataset; finally, Section 4 summarizes the findings of this study and identifies directions for future research.

2 Methodology

2.1 Preliminaries

Suppose a dataset $\mathcal{N}$ that contains N observations of choices made by individuals in a context that offers J alternatives. Each observation $(\boldsymbol{x}_n, i_n) \in \mathcal{N}$ consists of a vector $\boldsymbol{x}_n$ of explanatory variables, together with the chosen alternative i_n. In its simplest form, a choice model $P(i \,|\, \boldsymbol{x}_n; \boldsymbol{\theta})$ calculates the probability that the decision-maker associated with observation n chooses alternative i, in the context described by $\boldsymbol{x}_n$. Additionally, $\boldsymbol{\theta} \in \mathbb{R}^K$ is a vector of K parameters to be estimated via maximum likelihood estimation (MLE), which consists in maximizing the joint probability of all observed choices in the dataset $\mathcal{N}$. For numerical reasons, however, the *log likelihood* function is used instead:

$$\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log P\left(i_n \,|\, \boldsymbol{x}_n; \boldsymbol{\theta}\right). \tag{1}$$

The optimization problem of (1) is usually solved using Newton-based or quasi-Newton-based optimization methods.[1] Those rely on the gradient and Hessian — respectively, an approximation of the Hessian — of the log likelihood to iteratively update an initial guess of its maximum, until a certain stopping criterion is met. Inspired by Lederrey *et al.* (2021) and their stochastic adaptive batch size algorithms, we propose using the dataset-reduction method introduced in Ortelli *et al.* (2022, 2023), called LSH-DR, to generate a new batch of *weighted* observations at each iteration of the optimization process.

The LSH-DR method relies on locality-sensitive hashing (LSH) to quickly partition a dataset into groups of "similar" observations — or *buckets* — from which representative observations are drawn and then weighted, so as to better imitate the original dataset. As an example, suppose that LSH-DR is applied to a dataset $\mathcal{N} = \{(\boldsymbol{x}_n, i_n) : n = 1, \ldots, N\}$: we denote by $\tilde{\mathcal{N}} = \{(\boldsymbol{x}_g, i_g, N_g) : g = 1, \ldots, G\}$ a weighted sample built from $\mathcal{N}$ by LSH-DR, where all $(\boldsymbol{x}_g, i_g)$ are observations from the original dataset — *i.e.*, $\{(\boldsymbol{x}_g, i_g) : g = 1, \ldots, G\} \subseteq \mathcal{N}$ — and $\{N_1, \ldots, N_G\}$ are their associated weights. The log likelihood obtained on $\tilde{\mathcal{N}}$ is calculated as

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}) = \sum_{g=1}^{G} N_g \cdot \log P\left(i_g \,|\, \boldsymbol{x}_g; \boldsymbol{\theta}\right). \tag{2}$$

[1] We refer the reader to Lederrey *et al.* (2021) for an extensive review of said methods.

The number of selected observations $\mathcal{G}$ indirectly depends on a parameter of the LSH-DR method called the *bucket width*, that we denote as w. By changing the value of w, one can choose an appropriate degree of similarity between observations within buckets: a sufficiently small w only groups observations that are exactly identical, whereas greater values result in fewer buckets that contain larger amounts of increasingly dissimilar observations. Additional details may be found in Ortelli *et al.* (2022, 2023).

2.2 Stochastic adaptive resampling (STAR)

Our algorithm to solve the optimization problem of (1) is organized as follows.

Input The ingredients provided to the algorithm are:

- A dataset $\mathcal{N}$;
- An initial solution θ_0;
- An initial bucket width w_0.

Initialization The iteration number k is set to zero.

Iteration

1. With w_k as an input, LSH-DR is used to create a new, weighted sample $\mathcal{N}_k^*$.
2. A new candidate solution θ_{k+1} is built using an optimization method such as Newton, BFGS, or one of their variations. The log likelihood is computed using $\mathcal{N}_k^*$, as in (2).
3. The new bucket width w_{k+1} is calculated as

$$w_{k+1} = w_k \cdot \min\left(1, \frac{||\nabla_{\text{rel}}\mathcal{L}(\theta_{k+1})||}{||\nabla_{\text{rel}}\mathcal{L}(\theta_k)||}\right), \tag{3}$$

   where $\nabla_{\text{rel}}\mathcal{L}(\theta_k)$ is the relative gradient of $\mathcal{L}(\theta_k)$; each of its components is given by

$$[\nabla_{\text{rel}}\mathcal{L}(\theta)]_j = [\nabla\mathcal{L}(\theta)]_j \cdot \frac{\theta_j}{\mathcal{L}(\theta)}. \tag{4}$$

4. If $||\nabla_{\text{rel}}\mathcal{L}(\theta_{k+1})||$ is smaller than a certain threshold, the algorithm stops. Otherwise, k is incremented by one and the algorithm moves to the next iteration.

3 Experiment

We demonstrate the validity of our approach by estimating three multinomial logit (MNL) models of increasing complexity on the London passenger mode choice data (Hillel *et al.*, 2018). The dataset consists of more than 81'000 trip records collected over three years, combined with systematically matched trip trajectories alongside their corresponding mode alternatives. Four alternatives are distinguished. The three models are borrowed from (Hillel, 2019); we refer to them as "MNL-S", "MNL-M" and "MNL-L". Table 1 reports the number of parameters and explanatory variables they include.

Table 1: Complexity of the MNL-S, MNL-M and MNL-L models.

	MNL-S	MNL-M	MNL-L
Continuous variables	10	11	13
Binary variables	0	15	18
Parameters	13	53	100

We evaluate the benefits of using the STAR procedure in conjunction with the implementation of the Newton trust region (NTR) algorithm available in Biogeme (Bierlaire, 2023); the code is therefore modified to accommodate the resampling of the data at each iteration. Moreover, we test several values for w_0, the initial bucket width: each model is estimated 100 times for each of those and the obtained results are compared with the standard NTR algorithm. All model estimations are performed on two Intel Xeon Platinum 8360Y processors running at 2.4 GHz, with a total of 72 cores and 512 GB of RAM.

Figure 1 shows the obtained execution times for the standard NTR algorithm — in gray — and for NTR-STAR — in blue.[2] As one can see, some values of w_0 allow the NTR-STAR to outperform the standard NTR algorithm by a significant margin, both for the MNL-M and MNL-L. As regards the MNL-S, it seems that the standard NTR performs better. In reality, and as shown in Figure 2, the estimation time is actually lower for NTR-STAR, but the additional time needed to generate the samples outweighs the savings in estimation time. One should also note that the performance of NTR-STAR is dependent on the value of w_0, and the "optimal" value of w_0 seems to be dependent on the model. Indeed, the minimum execution times appear to be reached at $w_0 = 0.2$, $w_0 = 2$ and $w_0 = 1$ for the small, medium and large models, respectively. This constitutes a limitation of our approach: indeed, it would be preferable for the bucket width to be adapted dynamically, so as to mitigate the effects of a poorly chosen w_0.

[2]The execution time is to be understood as the sum of the sampling and estimation times.

Figure 1: Execution times of the NTR-STAR (blue) and standard NTR (gray) algorithms. The models are estimated 100 times for each value of w_0.

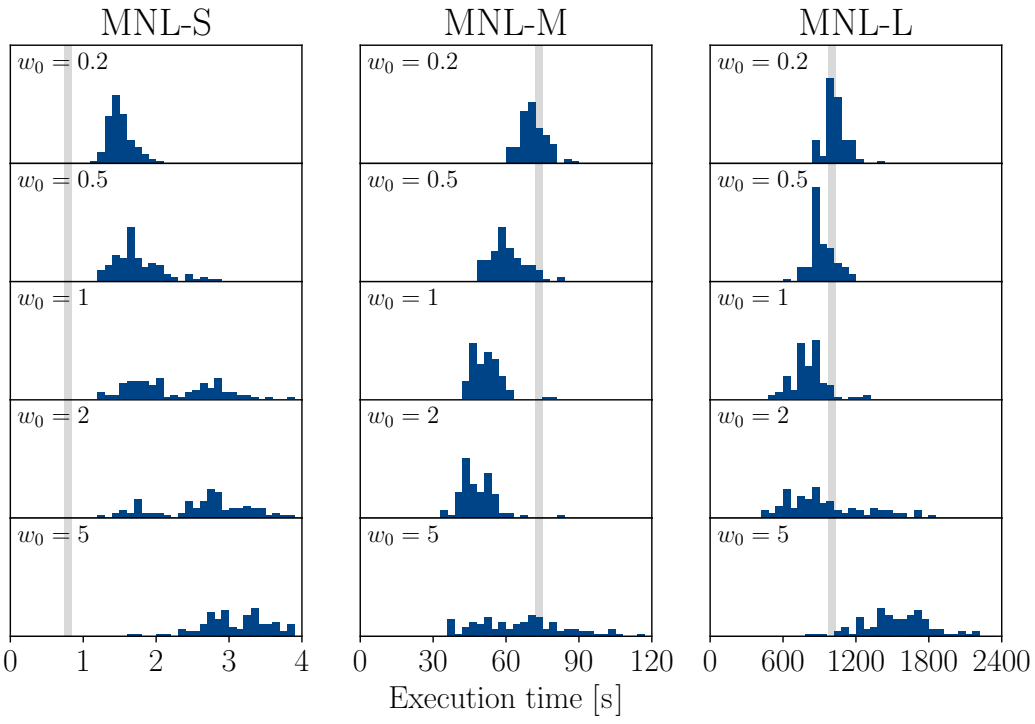
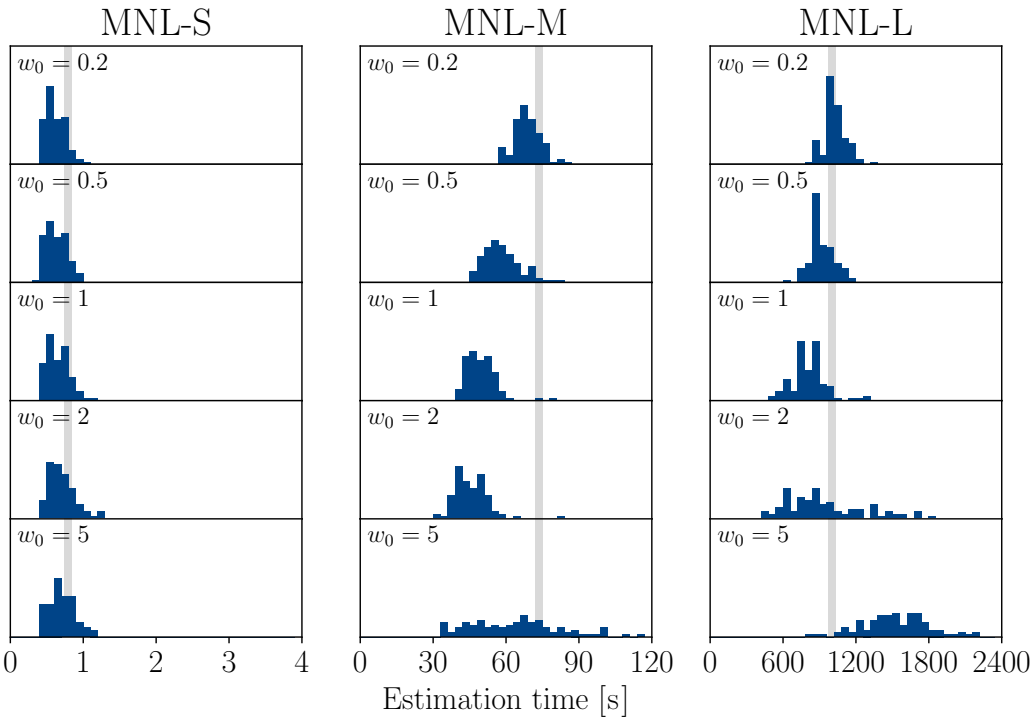


Figure 2: Estimation times of the NTR-STAR (blue) and standard NTR (gray) algorithms. The models are estimated 100 times for each value of w_0.



Finally, tables 2–4 display additional results of our experiments, including the number iterations, the number of epochs and the achieved speedup in comparison to the standard NTR algorithm. It is worth noting that the number of epochs required by NTR-STAR to converge is lower in almost all cases, which implies that our method improves the way data is used within the optimization algorithm.

Table 2: Comparison of the NTR and NTR-STAR algorithms for the MNL-S model.

Method	Iterations	Epochs	Exe. time [s]	Speedup
NTR	9.0 ± 0.0	9.0 ± 0.0	0.80 ± 0.06	—
NTR-STAR:				
$w_0 = 0.2$	8.9 ± 1.1	3.8 ± 1.0	1.51 ± 0.16	0.53
$w_0 = 0.5$	10.5 ± 1.6	3.8 ± 0.9	1.76 ± 0.34	0.45
$w_0 = 1$	12.5 ± 2.8	3.8 ± 1.0	2.27 ± 0.65	0.35
$w_0 = 2$	14.5 ± 3.2	4.3 ± 1.3	2.72 ± 0.69	0.29
$w_0 = 5$	16.5 ± 2.7	4.4 ± 1.3	3.15 ± 0.54	0.25

Table 3: Comparison of the NTR and NTR-STAR algorithms for the MNL-M model.

Method	Iterations	Epochs	Exe. time [s]	Speedup
NTR	7.0 ± 0.0	7.0 ± 0.0	73.6 ± 2.8	—
NTR-STAR:				
$w_0 = 0.2$	7.0 ± 0.3	5.9 ± 0.3	71.2 ± 5.3	1.03
$w_0 = 0.5$	7.2 ± 0.5	4.6 ± 0.5	60.6 ± 7.4	1.21
$w_0 = 1$	7.8 ± 0.5	4.2 ± 0.4	52.0 ± 6.1	1.42
$w_0 = 2$	8.2 ± 0.6	3.6 ± 0.5	48.2 ± 6.8	1.53
$w_0 = 5$	13.7 ± 3.4	5.0 ± 1.6	67.0 ± 18.7	1.10

Table 4: Comparison of the NTR and NTR-STAR algorithms for the MNL-L model.

Method	Iterations	Epochs	Exe. time [s]	Speedup
NTR	9.0 ± 0.0	9.0 ± 0.0	1004 ± 10	—
NTR-STAR:				
$w_0 = 0.2$	9.2 ± 0.7	8.3 ± 0.7	1030 ± 90	0.97
$w_0 = 0.5$	9.2 ± 0.8	7.2 ± 0.8	918 ± 99	1.09
$w_0 = 1$	9.5 ± 1.1	6.2 ± 1.1	812 ± 142	1.24
$w_0 = 2$	13.2 ± 4.0	7.2 ± 2.6	948 ± 322	1.06
$w_0 = 5$	22.6 ± 3.4	11.5 ± 2.1	1531 ± 274	0.66

4 Conclusion

In this study, we propose a simple method that generates batches of observations to be used within a stochastic optimization algorithm, as well as an adaptive batch-size updating scheme that relies on the performance of each iteration to select appropriate batch sizes for the following ones. Our method leverages a dataset reduction technique that generates weighted subsamples, so as to better guide the optimization algorithm, while saving substantial amounts of time at each iteration. The presented preliminary results highlight the potential of this approach on the estimation of multinomial logit models of medium to large sizes.

Intended future work will focus the development of an improved batch-size updating scheme. We believe that more appropriate indicators of the quality of an iteration could be derived, so as to give our method the ability to adapt more dynamically to poorly chosen initial bucket widths.

5 References

Bierlaire, M. (2023) A short introduction to Biogeme, *Technical Report*, TRANSP-OR 230620. Transport and Mobility Laboratory, ENAC, EPFL.

Broyden, C. G. (1970) The convergence of a class of double-rank minimization algorithms 1. general considerations, *IMA Journal of Applied Mathematics*, **6** (1) 76–90.

Fletcher, R. (1970) A new approach to variable metric algorithms, *The computer journal*, **13** (3) 317–322.

Goldfarb, D. (1970) A family of variable-metric methods derived by variational means, *Mathematics of computation*, **24** (109) 23–26.

Hillel, T. (2019) Understanding travel mode choice: A new approach for city scale simulation, Ph.D. Thesis, University of Cambridge.

Hillel, T., M. Z. Elshafie and Y. Jin (2018) Recreating passenger mode choice-sets for transport simulation: A case study of London, UK, *Proceedings of the Institution of Civil Engineers-Smart Infrastructure and Construction*, **171** (1) 29–42.

Lederrey, G., V. Lurkin, T. Hillel and M. Bierlaire (2021) Estimation of discrete choice models with hybrid stochastic adaptive batch size algorithms, *Journal of choice modelling*, **38**, 100226.

Ortelli, N., M. de Lapparent and M. Bierlaire (2022) Faster estimation of discrete choice models via dataset reduction, paper presented at the *Proceedings of the 23rd Swiss Transportation Research Conference*.

Ortelli, N., M. de Lapparent and M. Bierlaire (2023) Resampling estimation of discrete choice models, *Technical Report*, TRANSP-OR 230330. Transport and Mobility Laboratory, ENAC, EPFL.

Shanno, D. F. (1970) Conditioning of quasi-newton methods for function minimization, *Mathematics of computation*, **24** (111) 647–656.