Faster estimation of discrete choice models via dataset reduction

Nicola Ortelli

Matthieu de Lapparent

Michel Bierlaire

STRC conference paper 2022 **May 20, 2022**

STRC | **22nd Swiss Transport Research Conference**
Monte Verità / Ascona, May 18-20, 2022

Faster estimation of discrete choice models via dataset reduction

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

May 20, 2022

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. Specifically, as sample sizes increase, the computational cost of maximum likelihood estimation quickly becomes intractable for anything but trivial model structures. Efforts to tackle this issue have mainly been dedicated to improving the optimization methods used for estimating discrete choice models, but an equally promising approach consists in sampling datasets so as to reduce their size.

This paper proposes a simple dataset reduction method that is specifically designed to preserve the diversity of observations originally present in the dataset. Our approach leverages locality-sensitive hashing to create clusters of similar observations, from which representative observations are then sampled. We demonstrate the efficacy of our approach by applying it on a real-world mode choice dataset; the obtained preliminary results seem to confirm that a carefully selected subsample of observations is capable of providing close-to-identical estimation results while being, by definition, less computationally demanding.

Keywords

discrete choice models, sample size, dataset reduction, locality-sensitive hashing

Suggested Citation

1 Introduction

The technological advancements of the past 20 years have allowed transforming an increasing part of our daily actions and decisions into storable data. Specifically, the rise of digital communication has led to a radical change in the scale and scope of available data in relation to virtually any object of interest. In the field of discrete choice analysis, such abundance of data 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).

Specifically, the use of ever-larger datasets raises two issues: (i) the number of possible model specifications exponentially grows with the number of covariates, implying that analysts must spend more time to find good models; and (ii) the computational cost of maximum likelihood estimation increases with the number of observations and quickly becomes intractable for advanced model structures or for large datasets. While the first issue has spurred great interest,[1] the second has received much less attention: to deal with the increased computational cost associated with large datasets, effort has mainly been dedicated to improving the optimization methods used for estimating DCMs (Lederrey *et al.*, 2021) and to enhancing their implementation (Molloy *et al.*, 2021; Arteaga *et al.*, 2022).

This study explores a less common approach, which consists in reducing the size of large datasets by subsampling their observations. Because the most commonly used algorithms for maximum likelihood estimation compute the log likelihood function and its gradient across the whole dataset *at each iteration*, considering fewer observations effectively reduces their computational burden. Removing observations from a dataset is usually advised against by econometricians and choice modelers, but has nevertheless become common practice when machine learning models need to be trained on large amounts of data. Data reduction techniques such as smart sampling (Pedergnana *et al.*, 2016), instance selection (Arnaiz-González *et al.*, 2016), novelty detection (Pimentel *et al.*, 2014) or curriculum learning (Bengio *et al.*, 2009) all share the same premise that observations within a dataset may have different levels of importance in estimating a specific model; depending on the technique, these observations are either entirely removed from the dataset or set aside and used in later stages of the model training process.

[1] The recent literature is rich in studies that seek to mitigate the need for presumptive structural assumptions. We refer the reader to van Cranenburgh *et al.* (2021) for an extensive review and discussion.

To the best of our knowledge, the only study that explores this same approach in the context of discrete choice modeling is presented in van Cranenburgh and Bliemer (2019): their proposed method scales down any dataset to a predefined fraction of its original size while iteratively minimizing an estimate of the D-error, obtained by means of a simplified version of the model of interest.[2] In doing so, they seek to guarantee that the model parameters are estimated as precisely as possible on a subsample of observations that is much smaller than the full dataset. In reality, this encourages their algorithm to keep only similar observations, which may lead to severely biased parameters, as the model of interest might be estimated on a subsample that is not representative of the original dataset.

In this paper, we propose a simple dataset reduction method that is specifically designed to introduce as little bias as possible in the parameters of models estimated on the obtained subsamples. We diverge from the premise that all datasets contain some fraction of less relevant observations; instead, our method is designed to preserve the diversity of observations originally present in the dataset. Our approach leverages locality-sensitive hashing (LSH) to create clusters of similar observations, from which "representative" observations are then sampled. Observations obtained in such way are then given weights that are proportional to the sizes of the clusters they represent, so as to mimic the original dataset during the model estimation process. As argued in the following sections, we believe that a carefully selected *and weighted* subsample of observations is capable of providing close-to-identical estimation results while being, by definition, less computationally demanding.

The remainder of this document is organized as follows: Section 2 begins by introducing the concept of locality-sensitive hashing and then proceeds to describe our proposed algorithm; Section 3 presents and discusses the preliminary results obtained by applying our method to a real-world mode choice dataset; finally, Section 4 sets out the conclusions of the present study and identifies directions for future research.

[2]The D-error statistic is a measure of efficiency commonly used in experimental design. It is defined as the determinant of the asymptotic variance-covariance matrix of the estimated model parameters.

2 Methodology

2.1 Preliminary

Let us consider a choice dataset containing N observations, each consisting of a vector x_n of explanatory variables associated with individual n, together with the observed choice i_n of that same individual among J alternatives. In its simplest form, a discrete choice model $P(i \mid x_n; \theta)$ calculates the probability that individual n chooses alternative i as a function of x_n and θ, where θ is a vector of model parameters to be estimated from the data.

The values of the model parameters are typically determined through maximum likelihood estimation, which consists in finding values that maximize the joint probability of replicating all observed choices in the dataset. In practice, we usually maximize the logarithm of the likelihood instead, for numerical reasons. The *log likelihood* is therefore defined as

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

Let us now assume that the dataset contains some observations that are identical in all explanatory variables and in the observed choice. By dividing the observations into $G < N$ mutually exclusive groups such that each group exclusively contains identical observations, we may rewrite Eq. (1) as

$$\mathcal{L}(\theta) = \sum_{g=1}^{G} N_g \cdot P\left(i_g \,|\, x_g; \theta\right), \tag{2}$$

where N_g is the size of group g, and i_g and x_g are the observed choice and explanatory variables associated with all observations in group g, respectively. Eq. (1) and Eq. (2) are equivalent; however, since $G < N$, the computational cost associated with evaluating the log likelihood function is smaller for Eq. (2), by a ratio of $\frac{G}{N}$.[3] Models built on few explanatory variables may therefore see their estimation time greatly reduced by this factorization, whereas the same trick is expected to be less effective on models that include a large number of variables, as those will lessen the redundancy in the dataset.

[3] One could argue that multiplying $P\left(i_g \,|\, x_g; \theta\right)$ by N_g adds operations that are not required in Eq. (1). Still, the computational burden of these additional operations is negligible in comparison to the number of operations needed to evaluate $P\left(i_n \,|\, x_n; \theta\right)$ for every n.

The idea behind our dataset reduction method is to extend this "factorization trick" to *nearly* identical observations. In other words, by clustering together not only identical, but also very similar observations, the intent of our method is to further decrease the number of distinct groups and, in doing so, effectively reduce the computational cost associated with evaluating the log likelihood function and its gradient. The clustering technique chosen for this purpose is locality-sensitive hashing (LSH), which we introduce now.

2.2 Locality-sensitive hashing

LSH is an efficient method for finding similar items in data. As opposed to conventional hashing functions, which allocate items to unique encrypted outputs, LSH seeks to gather "similar" items into clusters — or *buckets*. It does so by combining the outcomes of several hashing functions, designed in such way that pairs of items are more likely to be hashed to the same bucket if they are close to each other in their original space than if they are far apart. A considerable advantage of LSH over other clustering techniques is that its computational complexity is linear in the number of items to be hashed.

A *family* of LSH functions $\mathcal{H} = \{h : (M, d) \to \mathbb{Z}\}$ is a collection of functions h that map elements of a metric space (M, d) onto the set of integers $\mathbb{Z}$, each integer representing a different bucket (Leskovec *et al.*, 2020).

Let $d_1 < d_2$ be two distances according to the metric d; family $\mathcal{H}$ is said to be (d_1, d_2, p_1, p_2)-sensitive if, for any pair of points $x_A, x_B \in M$ and for any function $h \in \mathcal{H}$, it fulfills the conditions

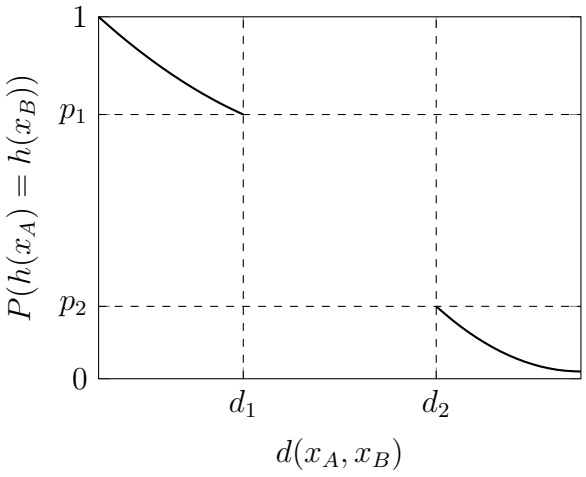
$$P(h(x_A) = h(x_B)) \geq p_1 \quad \text{if } d(x_A, x_B) \leq d_1 \tag{3}$$

and

$$P(h(x_A) = h(x_B)) \leq p_2 \quad \text{if } d(x_A, x_B) \geq d_2. \tag{4}$$

Fig. 1 illustrates the expected behavior of a (d_1, d_2, p_1, p_2)-sensitive hash function. The highest the value of p_1, the lowest the chances of observing false positives, *i.e.*, similar items hashed to different buckets. Likewise, the lowest the value of p_2, the lowest the chances of observing false negatives, *i.e.*, dissimilar items that end up in the same bucket.

Figure 1: Behavior of a (d_1, d_2, p_1, p_2)-sensitive hash function.



By combining several LSH functions together, it is possible to drive p_1 and p_2 apart from each other, hence simultaneously reducing the chances of both false positives and false negatives (Leskovec *et al.*, 2020). Given a (d_1, d_2, p_1, p_2)-sensitive family $\mathcal{H}$ of hash functions, the *AND-construction* and the *OR-construction* are defined as follows.

- Each member h' of a family $\mathcal{H}^{\text{AND}}$ created from $\mathcal{H}$ by the AND-construction combines r randomly chosen functions $\{h_1, \ldots, h_r\} \in \mathcal{H}$ such that

$$h'(x_A) = h'(x_B) \iff h_i(x_A) = h_i(x_B)\ \forall i = 1, \ldots, r. \tag{5}$$

- Similarly, each member h' of a family $\mathcal{H}^{\text{OR}}$ created from $\mathcal{H}$ by the OR-construction combines b randomly chosen functions $\{h_1, \ldots, h_b\} \in \mathcal{H}$ such that

$$h'(x_A) = h'(x_B) \iff \exists\, i \in \{1, \ldots, b\} : h_i(x_A) = h_i(x_B). \tag{6}$$

Because the basic functions $\{h_1, \ldots, h_r\} \in \mathcal{H}$ used to build the members of $\mathcal{H}^{\text{AND}}$ are selected independently, $\mathcal{H}^{\text{AND}}$ is, by construction, a (d_1, d_2, p_1^r, p_2^r)-sensitive family. Likewise, the functions $\{h_1, \ldots, h_b\} \in \mathcal{H}$ used to create members of $\mathcal{H}^{\text{OR}}$ are also chosen independently; $\mathcal{H}^{\text{OR}}$ is therefore $(d_1, d_2, 1-(1-p_1)^b, 1-(1-p_2)^b)$-sensitive by construction. The AND-construction therefore decreases the p_1 and p_2 probabilities of the family it is based on, whereas the OR-construction increases them. A way of combining basic LSH functions that is advocated for in the literature (Arnaiz-González *et al.*, 2016; Leskovec *et al.*, 2020) consists in applying the OR-construction on a family obtained by means of the AND-construction, which results in a $(d_1, d_2, 1-(1-p_1^r)^b, 1-(1-p_2^r)^b)$-sensitive family. We refer to it as an *AND-OR-construction*.

2.3 LSH-based dataset reduction

Our dataset reduction method has two main ingredients, namely: (i) an LSH function or a combination of LSH functions capable of dividing a sample of size N into G buckets that only contain "similar" observations; and (ii) a sampling strategy, applied within each bucket to select an observation that is representative of the bucket. The G observations selected in such way, together with the sizes $N_1, \ldots, N_G$ of the buckets they originate from, constitute the outcome of our dataset reduction method. Any model of interest may then be estimated on the obtained subsample rather than on the whole dataset by using the log likelihood function of Eq. (2), where i_g and x_g now refer to the observed choice and explanatory variables associated with the observation sampled from bucket g, respectively.

We begin by discussing the sampling strategy. The current version of our model selects one observation per bucket *randomly*, but more elaborate strategies may be used instead. For instance, another valid strategy could consist in selecting all bucket medoids.

As regards the LSH-based clustering, the combination of functions employed by our method is obtained by applying the AND-OR-construction — as defined at the end of Section 2.2 — on the family of basic functions given by

$$h_{a,b}(x) = \left\lfloor \frac{a \cdot x + b}{w} \right\rfloor, \tag{7}$$

where a is a vector with entries independently chosen from a normal distribution $\mathcal{N}(0,1)$, b is a real value chosen uniformly over $[0, w]$, w is the bucket width and $\lfloor \cdot \rfloor$ denotes the floor function. The family generated by Eq. (7) is known to be $(\frac{w}{2}, 2w, \frac{1}{2}, \frac{1}{3})$-sensitive (Datar *et al.*, 2004), which means that the AND-OR-construction generates a $(\frac{w}{2}, 2w, 1 - (1 - (\frac{1}{2})^r)^b, 1 - (1 - (\frac{1}{3})^r)^b)$-sensitive family.

The value of w is to be set by the analyst. It plays an important role in the clustering process, as it indirectly controls for the degree of dissimilarity between observations within a bucket: as the width of the buckets is increased, the total number of bucket decreases, causing the average number of observations per bucket to rise. Two other important parameters of the clustering process are r and b, *i.e.*, the number of combined basic functions in each AND-construction and the number of AND-constructions used in the OR-construction, respectively. Finally, it is crucial that all explanatory variables are normalized such that their values are between 0 and 1.

3 Case study

We demonstrate the efficacy of our dataset reduction method by applying it to a real-world mode choice dataset and using the obtained subsamples to estimate two multinomial logit models. The quality of subsamples generated by our method is compared with random subsamples of the same size. The comparison is based on four criteria, namely: (i) the estimation time; (ii) the performance of the estimated models on out-of-sample data in terms of log likelihood; (iii) the efficiency of the estimated model parameters, as measured by the D-error; and (iv) the value of time for one of the alternatives, computed as the ratio of the corresponding parameter estimates. All estimations are performed using the Biogeme package for Python (Bierlaire, 2018, 2020) and are run on a 2.3 GHz 32-core cluster node with 192 GB of RAM.

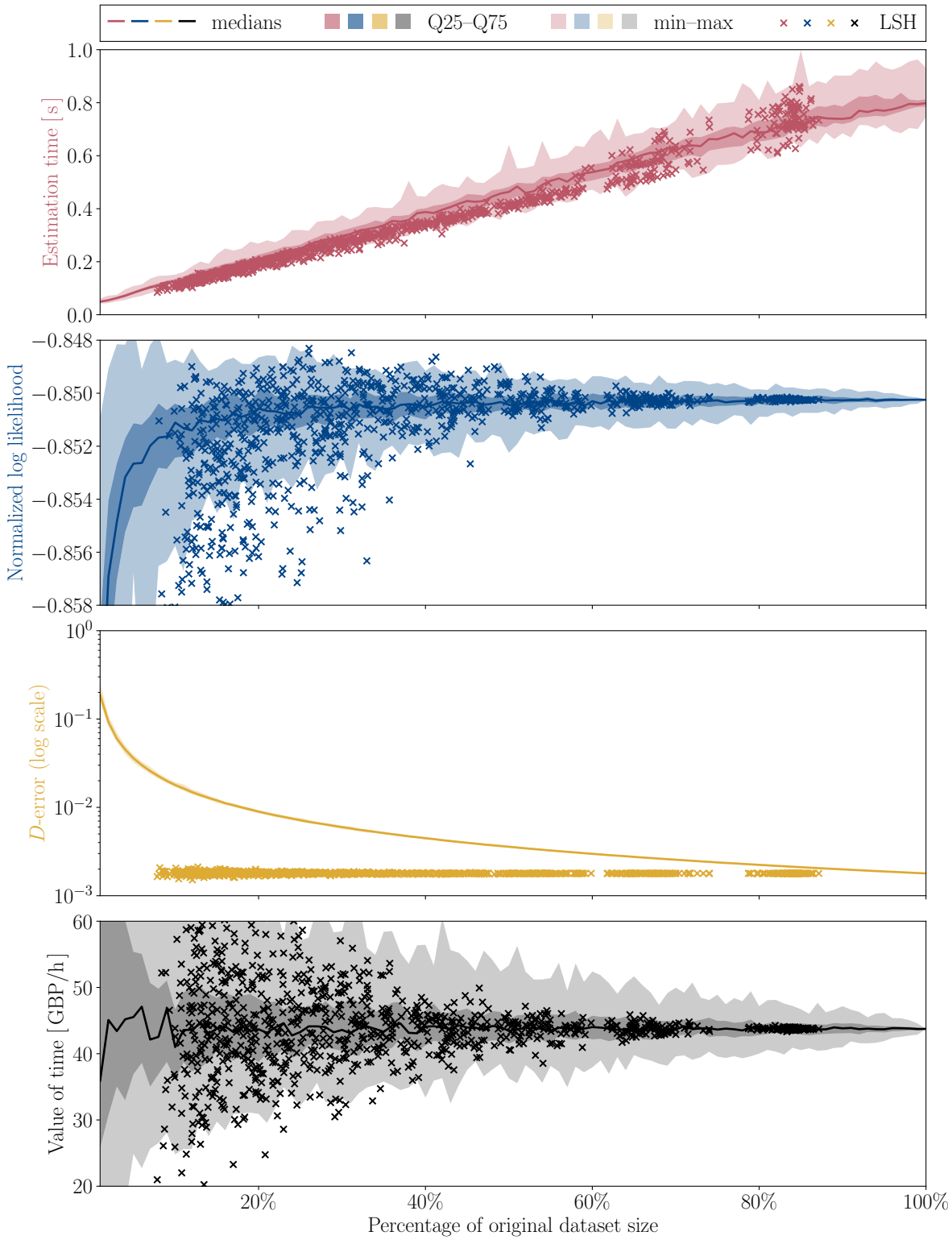
3.1 Dataset and models

All our experiments are based on the London Passenger Mode Choice (LPMC) dataset (Hillel *et al.*, 2018). The LPMC 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 modes are distinguished: walking, cycling, public transport and driving. We divide the dataset into two parts: the first two years of data — 54'766 observations — are used for model estimation whilst the final year of data — 26'320 observations — is set aside for out-of-sample validation.

The two considered multinomial logit models are borrowed from Hillel (2019). We refer to the smallest of the two as Model 1: it includes, as explanatory variables, the travel time and cost of all alternatives and the predicted traffic variability on the driving route. The travel time of the public transport alternative is divided into access and egress time, rail and bus in-vehicle times and interchange time, hence resulting in a total of 10 continuous variables and 13 parameters.

Model 2 is far more complex: in addition to the same 10 continuous variables, the straight-line distance between trip origin and destination is also considered, together with 15 dummy variables that encode socioeconomic characteristics of the individuals and context variables. In total, Model 2 therefore includes 26 explanatory variables and 53 parameters to be estimated.

Figure 2: Results visualization for Model 1.



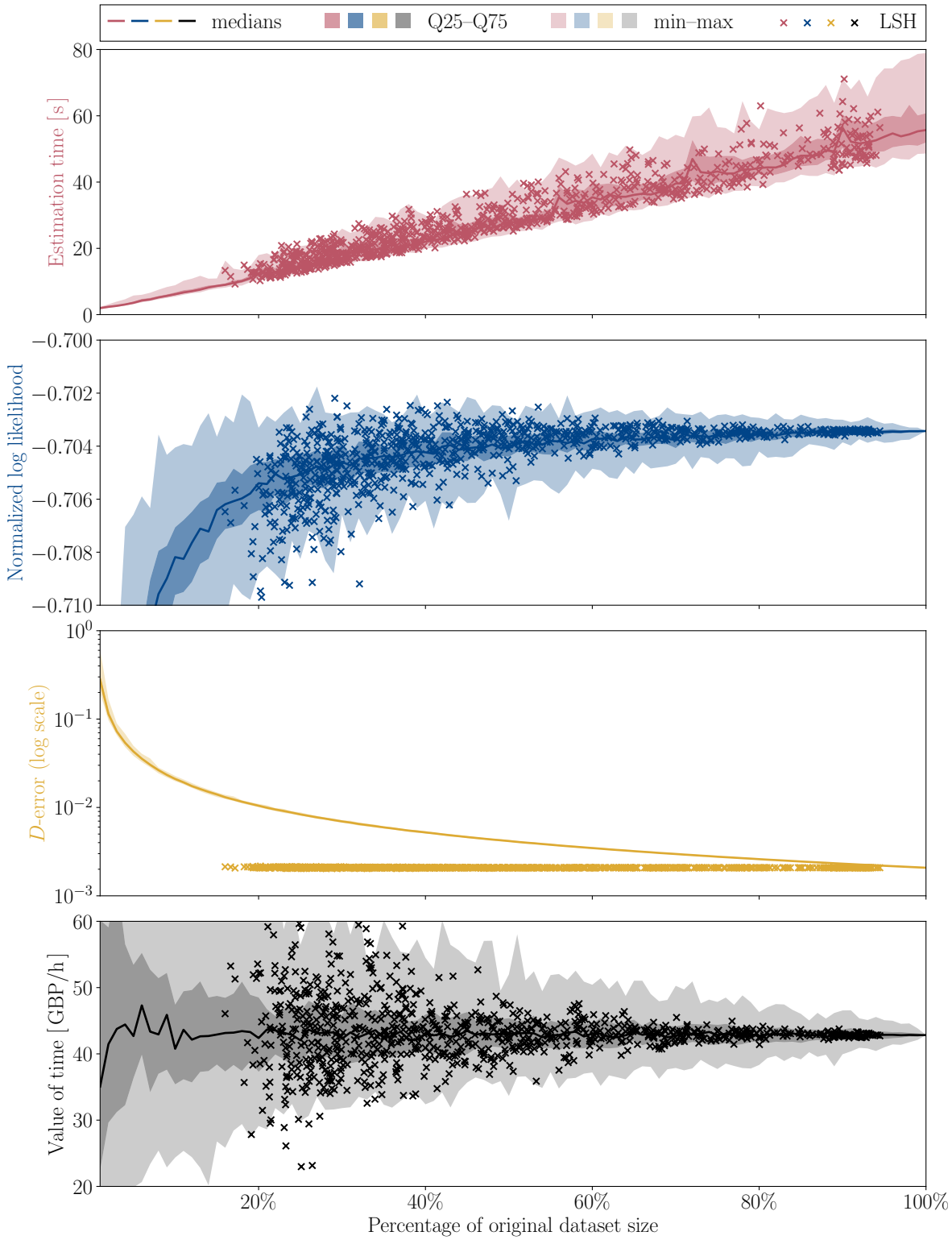
3.2 Preliminary results

Fig. 2 shows the results obtained by applying our method on the LPMC dataset prior to estimating Model 1. In each of the four subfigures, every cross represents a different subsample obtained by means of the LSH-based dataset reduction technique, with $r = 10$, $b = 1$. For each value of $w \in \{0.05, 0.10, 0.15, \ldots, 0.50\}$ the same experiment is repeated 100 times, which is the reason behind the "clusters" of crosses one can see in Fig. 2. For the sake of comparison, the solid-color areas illustrate the results obtained by random sampling. For each percentage of the original dataset size, 100 repetitions are performed.

The first subfigure illustrates the linear dependence between sample size and the resulting computational burden of the model estimation process. This relation is verify both with random subsamples and with subsamples generated by our method. The second subfigure shows the normalized log likelihood yielded on the out-of-sample data by Model 1 when trained on subsamples of the data. For high percentages of the original dataset size, our method reaches values that are nearly identical to the median value obtained on the full dataset. However, as the bucket width w grows, increasingly different observations are hashed to the same buckets and more and more information contained in the original dataset is lost. As a result, the performance of our method progressively deteriorates. Hence, for percentages of the original dataset size lower than 40%, random sampling seems to reach more consistent results than our method. The third subfigure illustrates the relation between sample size and D-error. The latter is known to decrease at a rate of $\frac{1}{\sqrt{N}}$ as the the sample size N increase. Our method produces subsamples that yield a D-error that is comparable to the one obtained on the full dataset because of the weights associated with each observation in the subsample. The last subfigure shows the value of time of the driving alternative, as estimated by models trained on subsamples of the whole dataset. The same trend as with the out-of-sample log likelihood may be observed here: provided the percentage of the original dataset size is above 40%, models estimated on subsamples generated by our method display values of time that are close to the one obtained by estimating Model 1 on the full dataset. However, as soon as the model is estimated on smaller subsamples, the quality of the estimates deteriorates rapidly.

Fig. 3 shows that similar results are obtained for Model 2. Our method is applied in the same conditions as for Model 1, except that $r = 5$ and w takes value in $\{0.1, 0.2, \ldots, 1.0\}$. A notable difference with the previous model is the smaller achieved reduction rate. This is due to the fact that Model 2 includes 15 more variables than Model 1; those contribute to reducing the number of identical and nearly identical observations in the dataset.

Figure 3: Results visualization for Model 2.



4 Conclusion

In this paper, we propose a dataset reduction method that allows for a faster estimation of discrete choice models. The gain in computational time generally comes at the cost of deteriorating the model estimation results; however, our method is specifically designed to mitigate this deterioration by preserving as much diversity as possible among the observations. The preliminary results presented in this paper are encouraging, in that they confirm that a carefully selected and weighted subsample of observations is capable of providing close-to-identical estimation results, while being, by definition, less computationally demanding.

Intended future work includes the development and testing of alternative sampling strategies for selecting observations from buckets. A more elaborate strategy could be used instead, so as to increase the probability of being selected for observations that are truly representative of their bucket. In a similar way, the basic LSH function considered in our method is based on random vectors; additional investigation could therefore consist in in developing more informed LSH functions, so as to make use of the analyst's knowledge of the dataset. Finally, another important direction of research consists in extending our framework to models that include parameter segmentation. In such cases, generating subsamples that capture the diversity among observations without distorting the representativeness of the original dataset is a tedious task.

5 References

Arnaiz-González, Á., J.-F. Díez-Pastor, J. J. Rodríguez and C. García-Osorio (2016) Instance selection of linear complexity for big data, *Knowledge-Based Systems*, **107**, 83–95.

Arteaga, C., J. Park, P. B. Beeramoole and A. Paz (2022) xlogit: An open-source python package for gpu-accelerated estimation of mixed logit models, *Journal of Choice Modelling*, **42**, 100339.

Bengio, Y., J. Louradour, R. Collobert and J. Weston (2009) Curriculum learning, paper presented at the *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Bierlaire, M. (2018) Pandasbiogeme: a short introduction, *Technical Report*, TRANSP-OR 181219. Transport and Mobility Laboratory, ENAC, EPFL.

Bierlaire, M. (2020) A short introduction to pandasbiogeme, *Technical Report*, TRANSP-OR 200605. Transport and Mobility Laboratory, ENAC, EPFL.

Datar, M., N. Immorlica, P. Indyk and V. S. Mirrokni (2004) Locality-sensitive hashing scheme based on p-stable distributions, paper presented at the *Proceedings of the twentieth annual symposium on Computational geometry*, 253–262.

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.

Leskovec, J., A. Rajaraman and J. D. Ullman (2020) *Mining of massive data sets*, Cambridge university press.

Molloy, J., F. Becker, B. Schmid and K. W. Axhausen (2021) mixl: An open-source r package for estimating complex choice models on large datasets, *Journal of choice modelling*, **39**, 100284.

Pedergnana, M., S. G. García *et al.* (2016) Smart sampling and incremental function learning for very large high dimensional data, *Neural Networks*, **78**, 75–87.

Pimentel, M. A., D. A. Clifton, L. Clifton and L. Tarassenko (2014) A review of novelty detection, *Signal processing*, **99**, 215–249.

van Cranenburgh, S. and M. C. Bliemer (2019) Information theoretic-based sampling of observations, *Journal of choice modelling*, **31**, 181–197.

van Cranenburgh, S., S. Wang, A. Vij, F. Pereira and J. Walker (2021) Choice modelling in the age of machine learning-discussion paper, *Journal of Choice Modelling*, 100340.