Resampling estimation of discrete choice models

Nicola Ortelli [*†] Matthieu de Lapparent [*] Michel Bierlaire [†]

March 30, 2023

Report TRANSP-OR 230330
Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

[*]School of Management and Engineering Vaud, HES-SO // University of Applied Sciences and Arts Western Switzerland, {nicola.ortelli,matthieu.delapparent}@heig-vd.ch

[†]Transport and Mobility Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), {nicola.ortelli,michel.bierlaire}@epfl.ch

Abstract

In the context of discrete choice modeling, the extraction of potential behavioral insights from large datasets is often limited by the poor scalability of maximum likelihood estimation. This paper proposes a simple and fast dataset reduction method that is specifically designed to preserve the richness of observations originally present in a dataset, while reducing the computational complexity of the estimation process. Our approach, called LSH-DR, leverages locality-sensitive hashing to create homogeneous clusters, from which representative observations are then sampled and weighted. We demonstrate the efficacy of our approach by applying it on a real-world mode choice dataset: the obtained results show that the samples generated by LSH-DR allow for substantial savings in estimation time while preserving estimation efficiency at little cost.

Keywords: discrete choice models, maximum likelihood estimation, dataset reduction, sample size, locality-sensitive hashing.

1 Introduction

The technological advancements of the past decades 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).

Indeed, when estimating DCMs, the use of ever-larger datasets raises two issues: (i) the number of possible model specifications exponentially grows with the number of candidate explanatory variables, implying that analysts must spend more time searching for appropriate specifications; and (ii) the computational cost of maximum likelihood estimation increases with the number of observations, quickly becoming intractable for any advanced model structure. While the first issue has spurred great interest,[1] the second has received much less attention: in order to deal with the increased computational cost associated with large datasets, effort has mostly been dedicated to improving the optimization methods used to estimate DCMs (Lederrey et al., 2021; Rodrigues, 2022) or to enhancing their implementation (Molloy et al., 2021; Arteaga et al., 2022).

This study moves beyond traditional techniques by exploring a less common approach, which consists in reducing the size of datasets by sampling. Because the most frequently 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 choice modelers, but has nevertheless become common practice when training machine learning models on large amounts of data: given the iterative nature of model specification, the use of a smaller sample that provides good approximations of the model's quality allows for early modeling decisions to be taken significantly faster (Park et al., 2019). Moreover, in the case of maximum likelihood estimation for DCMs, subsamples may additionally be used to obtain good starting values for estimation on the whole dataset after the definitive model specification is reached.

We propose a fast dataset reduction technique that is designed to introduce as little bias as possible into the parameter estimates of the model of interest. We diverge from the common premise that all datasets contain some fraction of less relevant observations; instead, our method aims at preserving the diversity of observations originally present in the dataset, while reducing its size. Our proposed approach leverages locality-sensitive hashing to create clusters of similar observations, from which "representative" observations are sampled. The 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 full 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 source code is currently being consolidated and integrated into the Biogeme software (Bierlaire, 2018, 2020).

The remainder of this document is organized as follows: Section 2 provides an overview of the pertinent literature; Section 3 introduces the concept of locality-sensitive hashing and then proceeds to describe our proposed algorithm; Section 4 presents and discusses the results obtained by applying our method to a real-world mode choice dataset; finally, Section 5 summarizes the findings of this study and identifies the future steps of this research.

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

2 Literature review

Instance selection and prototype generation are dataset reduction tasks that consist in producing a smaller representative set of data points from a given dataset, respectively by sampling said points or by creating new, artificial ones. While shrinking datasets, instance selection and prototype generation techniques typically aim at minimizing information or performance loss by discarding redundant data points; as such, they have been shown to be particularly beneficial to instance-based methods whose performances rely on specific data points, such as support vector machines or k-nearest neighbors algorithms (Olvera-López et al., 2010; Alexandropoulos et al., 2019).

The recent literature offers a variety of instance selection and prototype generation methods. Among those, a prevalent approach consists in using clustering algorithms to identify groups of similar data points, from which some are either sampled or merged into prototypes. Methods based on k-means and its variations are particularly popular despite becoming computationally heavy when applied to large datasets (Ougiaroglou and Evangelidis, 2016; Ren and Yang, 2019; Castellanos et al., 2021; Chang et al., 2021; Ougiaroglou et al., 2021; Saha et al., 2022). To circumvent this limitation, another stream of research makes use of locality-sensitive hashing (LSH) to cluster similar data points together (Arnaiz-González et al., 2016; Aslani and Seipel, 2020; Zhang and Liu, 2023). While k-means is known to be superior in terms of accuracy and reliability, LSH is intrinsically faster because its complexity is linear in the number of data points to be hashed (Paulevé et al., 2010). This aspect is crucial when instance selection and prototype generation methods are used to reduce the computational burden of model training; we therefore deem techniques based on LSH as particularly promising.

To the best of our knowledge, there have only been two attempts at developing dataset reduction methods in the context of discrete choice modeling. We explain this scarcity by the fact that DCMs are generally used to extract behavioral insights from data and, as such, require datasets to be representative of the actual population of interest. Any dataset reduction technique applied prior to estimating a DCM therefore needs to preserve the characteristics of the full dataset in the smaller sample, as it could otherwise lead to erroneous or biased model estimation results. In contrast, the machine learning paradigm solely focuses on maximizing predictive accuracy — or on minimizing any loss due to dataset reduction — without taking into account representativeness.

The earliest instance selection method for DCMs we could find is proposed by van Cranenburgh and Bliemer (2019): their method scales any dataset down to a predefined fraction of its 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 sample that is much smaller than the full dataset, but in reality, this only encourages their algorithm to keep observations that are similar among them. As a result, the obtained samples may not be representative of the full dataset and therefore lead to biased parameter estimates. The second direct precedent of this study is described by Schmid et al. (2022) as a pre-processing step applied to a very large dataset. The method consists in dividing the dataset into homogeneous clusters using k-means; a single observation is then sampled from each cluster and weighted according to the cluster size. While this approach is capable of preserving the characteristics of the full dataset thanks to the weighting scheme it employs, it still suffers from the fact that k-means is computationally heavy, which severely limits its usage.

[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.

In this paper, we follow an approach similar to the one presented in Schmid et al. (2022), but based on LSH. Our main contribution is a fast dataset reduction technique that relies on clustering to sample and weight observations. Jointly, the sampling strategy and weighting schemes used by our method guarantee that the characteristics of the full dataset are preserved. The generated samples therefore lead to the same behavioral findings as the full dataset, while the computational complexity of model estimation is effectively reduced.

3 Methodology

3.1 Intuition

Consider a choice dataset of N observations (x_n, i_n), 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 any 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 the values of θ that maximize the joint probability of replicating all observed choices in the dataset. In practice, the logarithm of the likelihood is usually maximized instead, for numerical reasons. The *log likelihood* function is therefore defined as

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

Let us now assume that the dataset contains some observations that are identical in all explanatory variables and in the observed choice. By gathering the observations into $G < N$ groups of identical observations, we may rewrite (1) as

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

where N_g denotes the size of group g and i_g and x_g are the observed choice and explanatory variables shared by all observations in group g, respectively. By definition, the G groups are a partition of the full dataset and therefore verify

$$\sum_{g=1}^{G} N_g = N. \tag{3}$$

(1) and (2) are equivalent and, as such, yield the exact same parameter estimates when maximized. However, since $G < N$, the computational cost of evaluating (2) is smaller, by a ratio of approximately $\frac{G}{N}$.[3] This is empirically shown in Section 4, Experiment B.

The idea behind our dataset reduction method is to extend this factorization trick to observations that are nearly identical. In other words, by clustering together not only duplicates, but also "very similar" observations, the intent of our method is to further decrease the

[3] One could argue that multiplying $P\left(i_g \mid x_g; \theta\right)$ by N_g adds arithmetic operations that are not required in (1); still, this additional burden is negligible in comparison to the number of operations needed to evaluate $P\left(i_n \mid x_n; \theta\right)$ for every n, even in trivial model structures.

number of distinct groups and, in doing so, to effectively reduce the computational time associated with evaluating the log likelihood function and its gradient. Of course, this comes at the cost of degrading the estimation results because part of the information contained in the dataset is lost; still, the use of an adequate clustering scheme limits said degradation while granting our method a certain reliability. The clustering technique designed for this purpose is inspired from locality-sensitive hashing (LSH), which we introduce now.

3.2 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 achieves this goal 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. As mentioned earlier, the main 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 $\mathbb{Z}$ (Leskovec et al., 2020). Each integer represents a different bucket, and two data points x_p and x_q belong to the same bucket of function h if and only if $h(x_p) = h(x_q)$. For instance, a well-known family that is suited for Euclidean spaces is based on the function

$$h_{a,b}(x) = \left\lfloor \frac{a \cdot x + b}{w} \right\rfloor, \tag{4}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, a is a vector whose entries are independently drawn from a normal distribution $\mathcal{N}(0, 1)$, b is a real value chosen uniformly from the range $[0, w)$ and w is the bucket width (Datar et al., 2004). One may see (4) as a projection of all data points onto a random line whose direction is given by vector a; an offset equal to b is then added to all projected points before the line is discretized into uniform intervals of size w. All data points that fall in the same interval are therefore deemed "equivalent" — contingent on a — and are assigned to the same bucket.

Parameter w plays a crucial role in the effectiveness of LSH, but its value is context-dependent. By changing the bucket width — or discretization step — one can choose an appropriate degree of similarity between data points within buckets: a sufficiently small w only groups points that are exactly identical, whereas greater values result in fewer buckets that contain larger amounts of increasingly dissimilar points. The effect of increasing or decreasing w is demonstrated in Section 4, Experiment C.

Another way of improving the discriminative power of LSH is to combine several hash functions. In the case of the family defined by (4), this corresponds to simultaneously projecting the data onto multiple random lines. Suppose a and b are drawn R times: now, two data points x_p and x_q belong to the same bucket if and only if they are grouped together by all R random projections, *i.e.*:

$$H_{A,B}(x_p) = H_{A,B}(x_q) \iff h_{a_r,b_r}(x_p) = h_{a_r,b_r}(x_q) \ \ \forall r = 1, \ldots, R, \tag{5}$$

where, for the sake of conciseness, $A = (a_1, \ldots, a_R)$ and $B = (b_1, \ldots, b_R)$ gather the R realizations of a and b, respectively. Increasing R reduces the joint probability that two data points are grouped together by all projections, which results in more buckets containing fewer items. This is also illustrated in Section 4, Experiment C.

3.3 LSH-based dataset reduction (LSH-DR)

Our dataset reduction algorithm has three main ingredients, namely: (i) an LSH function or a combination of LSH functions capable of partitioning a dataset of size N into buckets that only contain "similar" observations; (ii) a sampling strategy, based on which some observations are selected from each bucket; and (iii) a weighting scheme that assigns a weight N_g to each selected observation (x_g, i_g). The G observations obtained in such a way, together with their associated weights $N_1, \ldots, N_G$, 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 (2), with i_g and x_g now referring to the observed choice and explanatory variables associated with the g-th selected observation, respectively.

Clustering Our method uses the family of LSH functions introduced in (4) with, as parameters, the discretization step w and the number of projections R. It is crucial that prior to hashing, all numerical variables are normalized such that their values are between 0 and 1 and that all categorical variables are encoded using binary indicators. The individuals' choices are not taken into account during the clustering, which implies that the buckets might be heterogeneous, *i.e.*, observations with different chosen alternatives might end up in the same bucket.

Sampling The current version of our method randomly selects one observation from each alternative in each bucket. Up to J observations may therefore be sampled from each heterogeneous bucket, J being the number of alternatives in the choice context. Other sampling strategies may be used instead, including some that rely on the content of each bucket to generate synthetic prototypical observations. We leave these for future work.

Weighting Each selected observation (x_g, i_g) is given a weight N_g that is equal to the number of observations that share the same bucket $H_{A,B}(x_g)$ and chosen alternative i_g:

$$N_g = |\{(x_n, i_n) \mid H_{A,B}(x_g) = H_{A,B}(x_n),\, i_g = i_n\}| \tag{6}$$

The weights are of crucial importance in the estimation process, as those ensure that the sample best approximates the dataset: since the observations are not uniformly distributed among the buckets, some particular combinations of variable values might end up underrepresented if the weights are neglected. Jointly, the adopted sampling strategy and weighting scheme guarantee that the sum of all weights is equal to the number of observations in the full dataset, as in (3).

4 Experiments

The efficacy of our method is demonstrated by means of a series of experiments based on the London passenger mode choice (LPMC) 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 modes are distinguished: walk, cycle, ride public transport and drive. 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 — 26'320 observations — is set aside for out-of-sample validation.

In the experiments, the data is used to train two multinomial logit models that we borrow from Hillel (2019). We refer to those as "MNL-S" and "MNL-L" and provide their specifications in the Appendix. The MNL-S includes 10 continuous variables and 13 associated

parameters, whereas the MNL-L considers 11 continuous variables, 8 categorical variables encoded using binary indicators and 53 associated parameters. All model estimations are performed using the Biogeme package for Python (Bierlaire, 2018, 2020) on a 2.3 GHz 32-core cluster node with 192 GB of RAM.

Experiment A: random samples

We begin by illustrating the relation between sample size, estimation time and quality of the estimation results. For this purpose, we estimate the MNL-S on random subsamples of the LPMC dataset and report the following quantities: (i) the execution time, which consists of the sampling and estimation times; (ii) the normalized out-of-sample log likelihood (OSLL), *i.e.*, the log likelihood yielded by the estimated model on the validation data, normalized by the number of observations; (iii) the mean absolute percentage error (MAPE) of the parameter estimates; and (iv) the value of time for the "drive" alternative, computed as the ratio between the estimates of the parameters associated with travel time and cost. The subsamples range from 100% to 1% of the full dataset size — *i.e.*, 54'766 — and 100 repetitions are performed at each sample size. Figure 1 presents the obtained results by means of moving-window boxplots displaying the 5^{th}, 25^{th}, 50^{th}, 75^{th} and 95^{th} percentiles. The parameter estimates are plotted individually on Figure 7 in the Appendix.

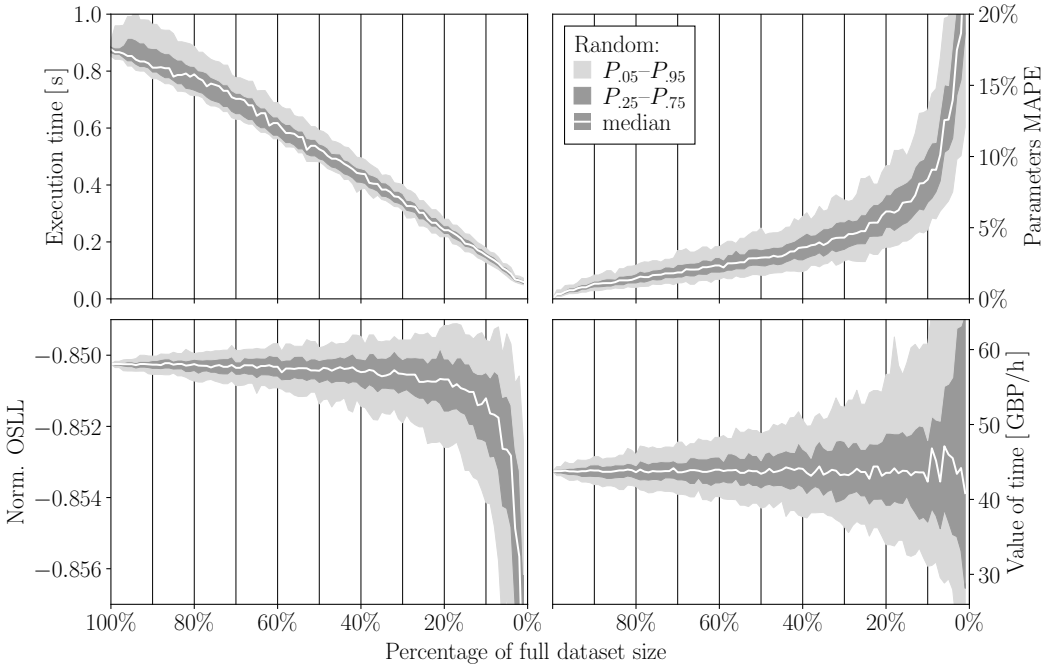

Figure 1: Estimation of the MNL-S on random samples of the LPMC dataset.

The first subfigure illustrates that the model estimation time decreases linearly with the number of observations in the sample, and so does the range of values obtained across repetitions. The normalized OSLL stays reasonably close to its maximum value down to 30% of the full dataset size, but seriously declines for smaller samples. Similarly, the MAPE slowly increases until the sample size reaches approximately 20%, then degrades at a much faster pace. Finally, the median of the value of time appears to be relatively stable, but its accuracy deteriorates considerably as the sample size decreases. Overall, Figure 1 shows that random sampling is a decent strategy when the model of interest is small; still, the next

experiments show that significantly better results can be achieved with a negligible increase in execution time.

Experiment B: LSH-DR

We now move to estimating the MNL-S on samples generated by our proposed method. We apply the LSH-DR algorithm on the LPMC data 10'000 times, with w ranging from 0.02 to 0.2 and $R = 4$. The obtained samples range from 48'206 to 1'361 observations in size, that is, from 88% to 2% of the full dataset. The results are shown in Figure 2 and Figure 3. For comparative purposes, we also report the outcomes of the previous experiment.

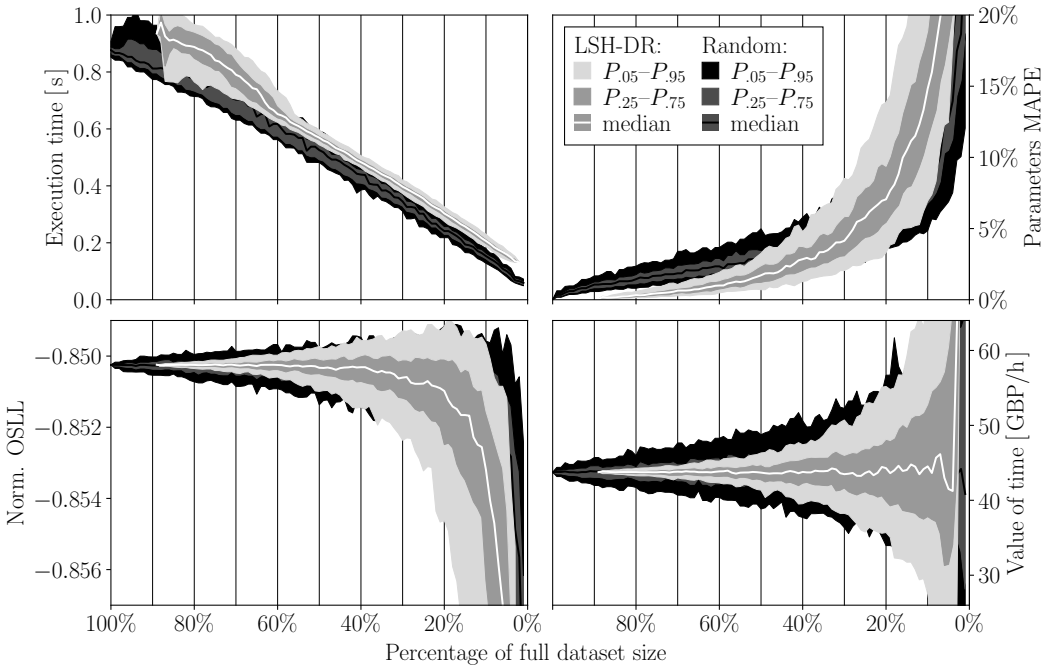

Figure 2: Estimation of the MNL-S on samples generated by LSH-DR. The results obtained on random samples are also reported, for comparative purposes.

Figure 2 demonstrates that LSH-DR is capable of producing substantially better samples than random sampling, for a small increase in execution time: down to approximately 40% of the full dataset size, the samples generated by LSH-DR yield smaller MAPEs of the parameters and more accurate estimates of the value of time. Figure 3 further shows that our method also has a beneficial effect on individual parameter estimates. As the sample sizes decrease, the estimates obtained on random samples deteriorate faster than those obtained on samples obtained via LSH-DR. One should note that the estimate of the parameter associated with rail in-vehicle time degenerates at a much faster pace than the other parameters for both sampling methods, but this is likely due to the fact that the variable is zero in almost 70% of the observations.

Experiment C: sensitivity analysis

This experiment highlights the importance of choosing an appropriate bucket width w and number of projections R for LSH-DR. Figure 4 shows the effect of these parameters on the size of the sample obtained by hashing the first two years of the LPMC data. In this case

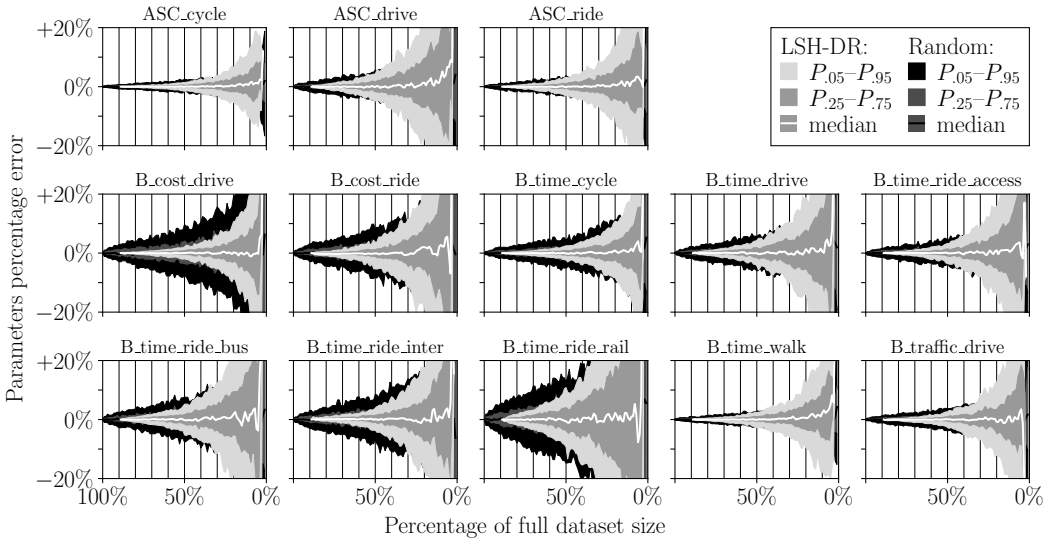

Figure 3: Percentage error of the MNL-S parameter estimates on samples generated by LSH-DR. The results obtained on random samples are also reported, for comparative purposes.

only the 10 explanatory variables of the MNL-S are considered, as if the dataset was being reduced prior to estimating that model.

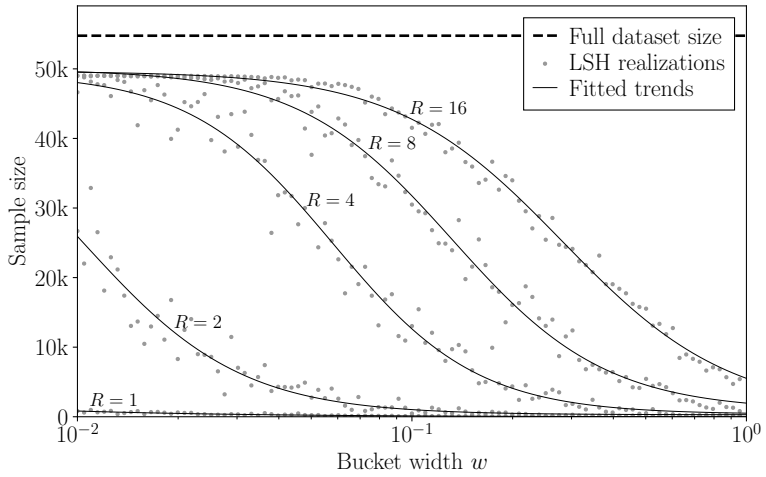

Figure 4: Size of samples generated by LSH-DR as a function of w and R.

Figure 4 shows that depending on R, the behavior of LSH-DR varies greatly as w increases. Still, irrespective of R, a sufficiently small w yields a number of buckets equal to the number of unique observations in the data, which could be seen as lossless dataset reduction.[4] There are 49'020 unique observations in this setting — *i.e.*, when considering only 10 variables — which means that over 5'700 observations are duplicates.

[4]On Figure 4, such a value is too small to be represented for $R = 1$.

Experiment D: comparison with state-of-the-art methods

In this experiment, we compare the performance of our method with three other dataset reduction techniques, namely: (i) random sampling; (ii) k-means clustering, similar to the approach taken in Schmid et al. (2022);[5] and (iii) sampling of observations (SoO), as proposed by van Cranenburgh and Bliemer (2019).[6] We proceed as follows: a certain percentage of the full dataset size is chosen and we retrieve from Experiment–B the 100 samples of size closest to that percentage; the three other dataset reduction techniques are then used to generate samples of those exact sizes, which are finally used to train the MNL-S model. Figure 5 reports the sampling time, normalized OSLL, parameters MAPE and value of time for the "drive" alternative for 75%, 50% and 25% of the full dataset. The size of the samples retrieved from Experiment B ranges from 40'879 to 41'273, from 27'146 to 27'615 and from 13'480 to 13'984, respectively.

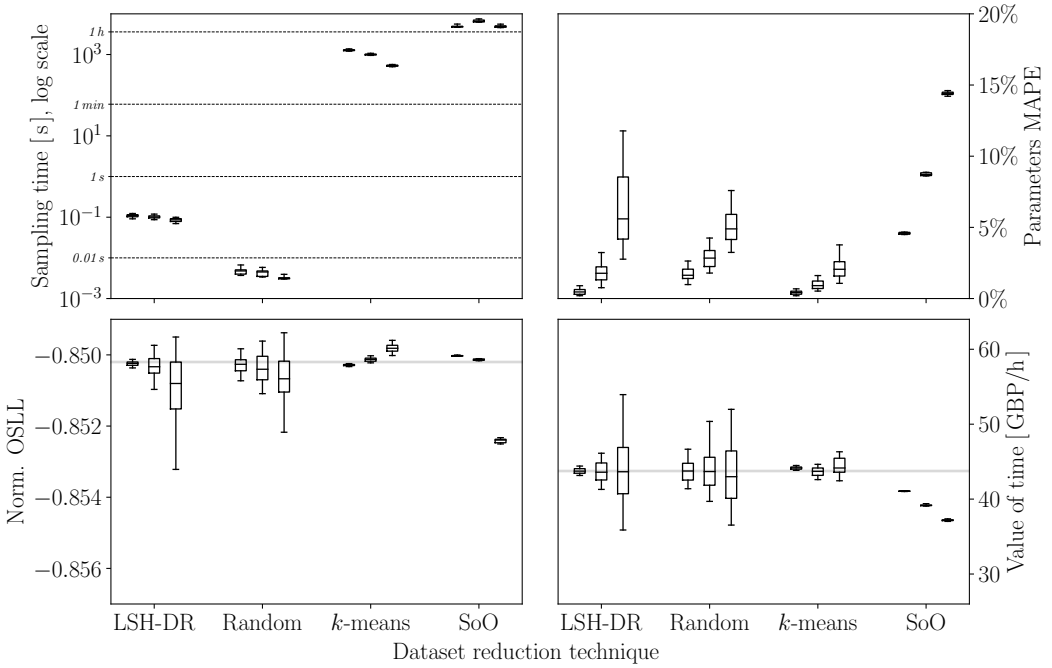

Figure 5: Comparison of dataset reduction techniques at approximately 75%, 50% and 25% of the full dataset size. The boxplot whiskers indicate the 5^{th} and 95^{th} percentiles, whereas the grey horizontal lines represent the normalized OSLL and value of time obtained on the full dataset.

Overall, Figure 5 illustrates that the samples producing the most accurate results are obtained via k-means. For all chosen percentages of the full dataset size, k-means generates the best samples in terms of OSLL, MAPE and value of time; only for the largest size does LSH-DR generate samples of comparable quality. Still, despite its superiority, k-means is practically unusable because of its runtime: it takes from 8 up to 26 minutes to obtain

[5]The technique based on k-means consists in running the clustering algorithm on the observations of each alternative separately, then randomly sampling one observation from each cluster and associating it with a weight that corresponds to the size of the cluster.

[6]In SoO, we use the MNL-S model estimated on the full dataset as the sampling model. All individual Fisher information matrices are pre-computed to speed up the search. The algorithm was implemented from scratch to be compatible with the rest of our code.

a sample from a relatively small dataset. That is between 4'000 and 16'000 times longer than LSH-DR and up to 400'000 times longer than random sampling. As regards SoO, the method is shown to provide the worst results in terms of OSLL, MAPE and value of time, while also displaying the largest runtimes. This is due to the fact that SoO is designed to maximize the efficiency of the parameter estimates rather than their precision or the model's predictive accuracy.

Experiment E: larger model

Finally, we estimate the MNL-L model on samples generated by LSH-DR to demonstrate that our method may also be beneficial to larger models. To this end, we apply the LSH-DR algorithm on the LPMC data 10'000 times, with w ranging from 0.1 to 1 and $R = 4$. Note that the MNL-L model includes several discrete explanatory variables; those are not treated differently by the LSH-DR method, but their inclusion requires the values of w to be adapted to achieve the desired range of sample sizes. The generated samples range from 51'574 to 3'584 observations in size, that is, from 94% to 7% of the full dataset. Figure 6 displays the achieved results. For the sake of comparison, the results obtained on random samples are also shown, both on Figure 6 and on Figure 8 in the Appendix.

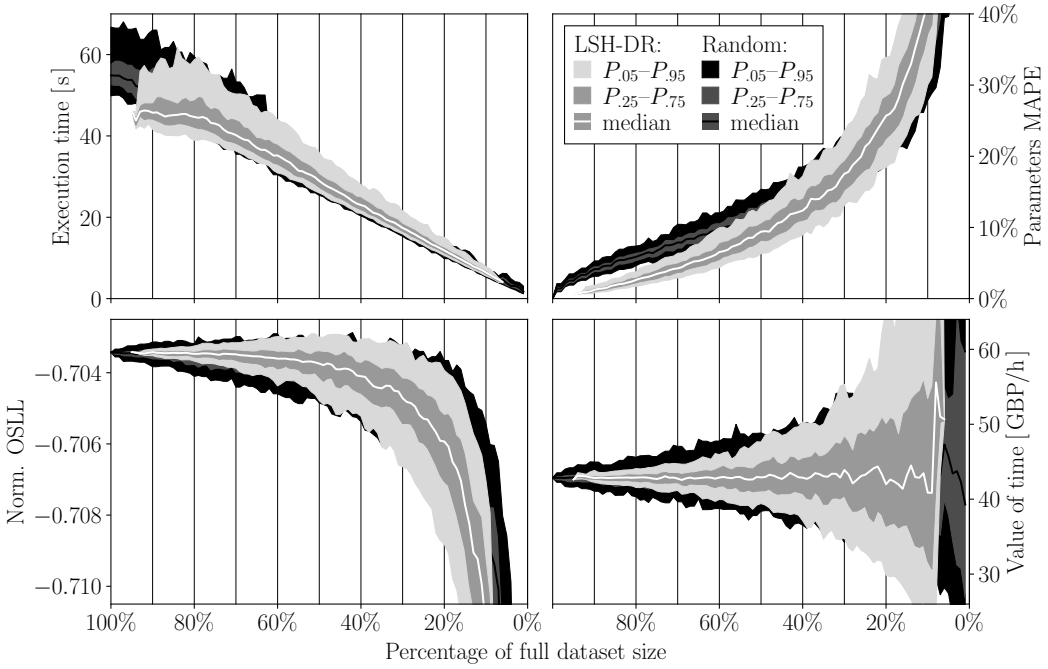

Figure 6: Estimation of the MNL-L on samples generated by LSH-DR. The results obtained on random samples are also reported, for comparative purposes.

Closing remarks

The conducted experiments empirically demonstrate the validity and potential of our proposed resampling method. In particular, they show that the LSH-DR algorithm outperforms random sampling for both model sizes, in exchange for a negligible increase in computational time. The algorithm is also shown to be several orders of magnitude faster than the alternative approaches proposed in the existing literature, while generating samples of comparable

or superior quality. Finally, the reported results confirm that the samples generated by our algorithm yield accurate parameter estimates, which is of crucial importance in discrete choice modeling.

5 Conclusion

In this paper, we propose a simple and fast dataset reduction technique designed to speed up the estimation of discrete choice models. The gain in computational time naturally 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. As a result, the quality of the parameter estimates stays within reasonable ranges even for large reduction rates. The presented results additionally highlight the benefits of our method on the estimation of models of small and medium size.

Intended future work includes the development and testing of more elaborate sampling strategies for selecting observations from buckets. For instance, those could be designed to increase the probability of choosing the most representative observations within each bucket, or to rely on the content of each bucket to generate synthetic prototypical observations. Additional investigation could also consist in developing LSH functions that can accommodate the analyst's knowledge of the dataset or the structure of the model of interest. For instance, one could consider a distinct projection for each alternative in the choice context and, in each projection, include only the variables that are relevant to the corresponding utility function; alternatively, one could associate probability distributions with greater means to specific variables so as to give them more importance in the hashing. Finally, another promising direction of research consists in embedding the LSH-DR method within a stochastic gradient descent algorithm for model estimation, such as the one proposed in Lederrey et al. (2021). The use of carefully selected and weighted batches of data, rather than random ones, could result in significantly better approximates of the gradient and, as a result, speed up convergence.

References

Alexandropoulos, S.-A. N., Kotsiantis, S. B. and Vrahatis, M. N. (2019). Data preprocessing in predictive data mining, *The Knowledge Engineering Review* **34**: e1.

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

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

Aslani, M. and Seipel, S. (2020). A fast instance selection method for support vector machines in building extraction, *Applied Soft Computing* **97**: 106716.

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.

Castellanos, F. J., Valero-Mas, J. J. and Calvo-Zaragoza, J. (2021). Prototype generation in the string space via approximate median for data reduction in nearest neighbor classification, *Soft Computing* **25**(24): 15403–15415.

Chang, E., Shen, X., Yeh, H.-S. and Demberg, V. (2021). On training instance selection for few-shot neural text generation, *arXiv preprint arXiv:2107.03176* .

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

Hillel, T. (2019). *Understanding travel mode choice: A new approach for city scale simulation*, PhD thesis, University of Cambridge.

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

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

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

Olvera-López, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F. and Kittler, J. (2010). A review of instance selection methods, *Artificial Intelligence Review* **34**: 133–143.

Ougiaroglou, S. and Evangelidis, G. (2016). Rhc: a non-parametric cluster-based data reduction for efficient k-nn classification, *Pattern Analysis and Applications* **19**(1): 93–109.

Ougiaroglou, S., Filippakis, P. and Evangelidis, G. (2021). Prototype generation for multi-label nearest neighbours classification, *Hybrid Artificial Intelligent Systems: 16th International Conference, HAIS 2021, Bilbao, Spain, September 22–24, 2021, Proceedings 16*, Springer, pp. 172–183.

Park, Y., Qing, J., Shen, X. and Mozafari, B. (2019). BlinkML: Efficient maximum likelihood estimation with probabilistic guarantees, *Proceedings of the 2019 International Conference on Management of Data*, pp. 1135–1152.

Paulevé, L., Jégou, H. and Amsaleg, L. (2010). Locality sensitive hashing: A comparison of hash function types and querying mechanisms, *Pattern recognition letters* **31**(11): 1348–1358.

Ren, H. and Yang, B. (2019). Clustering-based prototype generation for imbalance classification, *2019 International Conference on Smart Grid and Electrical Automation (ICSGEA)*, IEEE, pp. 422–426.

Rodrigues, F. (2022). Scaling bayesian inference of mixed multinomial logit models to large datasets, *Transportation research part B: methodological* **158**: 1–17.

Saha, S., Sarker, P. S., Al Saud, A., Shatabda, S. and Newton, M. H. (2022). Cluster-oriented instance selection for classification problems, *Information Sciences* **602**: 143–158.

Schmid, B., Becker, F., Molloy, J., Axhausen, K. W., Lüdering, J., Hagen, J. and Blome, A. (2022). Modeling train route decisions during track works, *Journal of Rail Transport Planning & Management* **22**: 100320.

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

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

Zhang, J. and Liu, C. (2023). Fast instance selection method for SVM training based on fuzzy distance metric, *Applied Intelligence* pp. 1–16.

Appendix

Table 1: Specification of the MNL-S and MNL-L logit models (Hillel, 2019). All categorical variables are encoded using binary indicators. All explanatory variables are associated with alternative-specific parameters and all utility functions are linear in parameters. Constants included, the models consider 13 and 53 parameters, respectively.

Variable	Type	MNL-S				MNL-L			
		walk	cycle	ride	drive	walk	cycle	ride	drive
Travel time	cont.	×	×	×*	×	×	×	×*	×
Travel cost	cont.			×	×			×	×
Traffic level	cont.				×				×
Straight-line distance	cont.						×	×	×
Driver's license	bin.						×	×	×
Gender	bin.						×	×	×
Age:	cat.								
child	*ind.*							×	×
pensioner	*ind.*						×	×	×
Car ownership:	cat.								
one car in household	*ind.*						×†	×	×
more than one	*ind.*						×†	×	×
Trip purpose:	cat.								
home-based work	*ind.*						×	×	×
home-based education	*ind.*							×	×
home-based other	*ind.*						×	×	
employers' business	*ind.*						×	×	×
Time of departure:	cat.								
PM peak	*ind.*						×	×	×
inter-peak	*ind.*						×		×
Day of week:	cat.								
weekdays	*ind.*							×	×
Saturday	*ind.*						×	×	
Season:	cat.								
winter	*ind.*						×		×

* The travel time of the "ride" alternative is split into four components, each associated with a distinct parameter: access and egress time, bus in-vehicle time, rail in-vehicle time and interchange time.

† The "cycle" alternative considers a single parameter associated to both car-ownership categories.

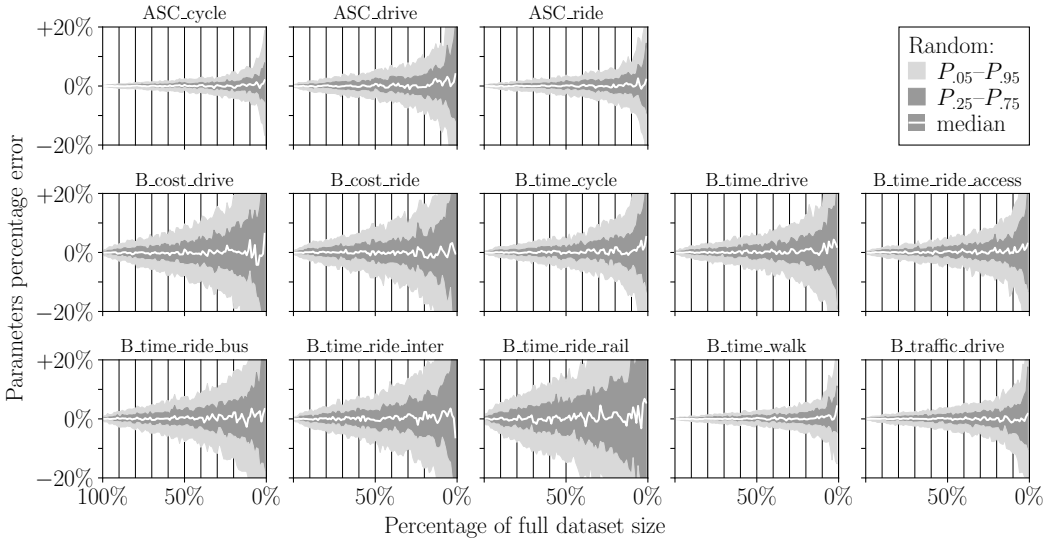

Figure 7: Percentage error of the MNL-S parameter estimates on random samples of the LPMC dataset.

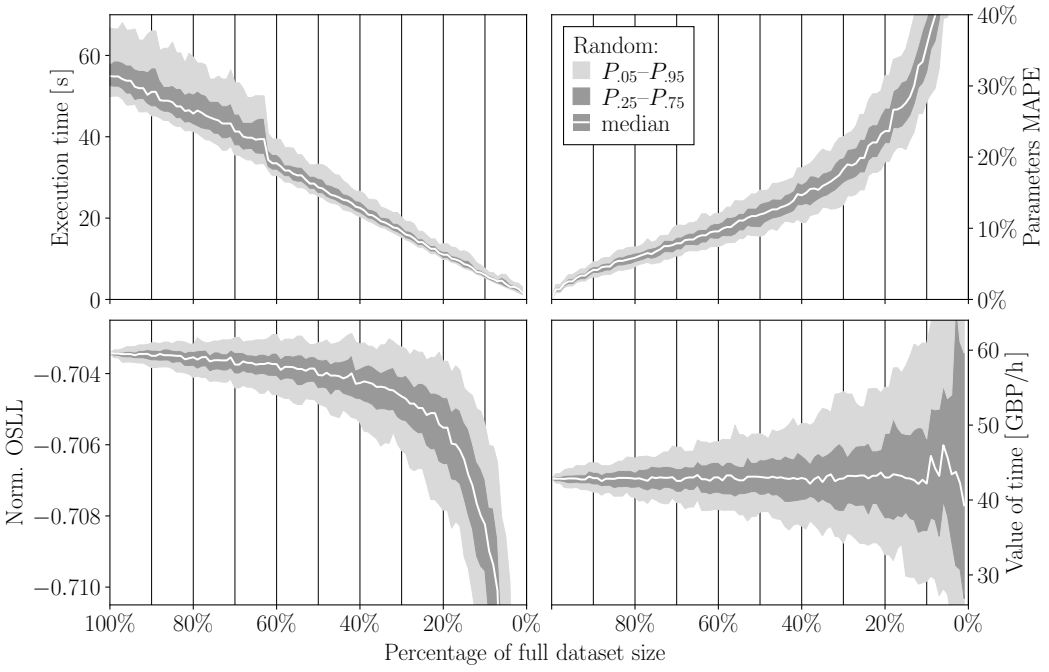

Figure 8: Estimation of the MNL-L on random samples of the LPMC dataset.