

Graphical Abstract

**Dynamic estimation of multinomial logit choice models**

Max Kapur,

## Highlights

### **Dynamic estimation of multinomial logit choice models**

Max Kapur,

- Research highlight 1
- Research highlight 2

# Dynamic estimation of multinomial logit choice models

Max Kapur,

<sup>a</sup>*Seoul National University, Seoul, South Korea*  
<sup>b</sup>, , , ,

---

## Abstract

Recent literature in choice model approximation has considered the dynamic data setting, in which new observations of consumer choices must be incorporated into the model but computational resources prevent the analyst from storing these observations in memory or solving the . Whereas research in this area has focused on nonparametric choice models, this paper considers the dynamic estimation of the multinomial logit choice (MNL) model and its nested variant. I provide a dynamic approximation scheme for the model parameters that updates the current estimate according to an intuitive rule and a sequence of decreasing step sizes. Stochastic convergence is guaranteed by the Robbins–Monro condition, and numerical results demonstrate the feasibility of the model.

*Keywords:*

---

## 1. Dynamic choice modeling

In a recent paper, Ho-Nguyen and Kilinç-Karzan (2021) articulate the need for dynamic choice-model fitting techniques that can efficiently update the current model in light of new observations of consumer behavior, rather than solving a full optimization problem over the perturbed data. However, that paper concerned nonparametric choice models, which are fundamentally hard to estimate even in a static setting. On the other hand, parametric choice offer a substantial advantage in computational tractability. To establish parity in the larger debate over the relative merits of parametric and nonparametric choice models, it is worth considering whether the tractability of parametric choice models also extends to the dynamic setting.

---

*Email address:* maxkapur@snu.ac.kr (Max Kapur)

The present paper takes a few first steps in this direction by providing a dynamic approximation scheme for the multinomial logit choice model and its nested variant. When a new observation of a consumer choice given a certain consideration set comes in, the technique proposed here updates the current parameter estimate according to a simple rule with a sequence of decreasing step sizes. Stochastic convergence is guaranteed by the Robbins-Monro condition. I demonstrate the feasibility of the technique through numerical results.

## 2. The nested multinomial logit choice model

Consider a set of products  $\mathcal{N} = \{1 \dots n\}$ . A consumer is offered a choice among an assortment of products  $\mathcal{A} \subseteq \mathcal{N}$ . A choice model is a probabilistic model that defines the probability that the consumer will choose product  $i$ .

In the multinomial logit (MNL) choice model, each consumer  $j$ 's preference order over the products in  $\mathcal{A}$  is determined by the sum of a systematic preferability parameter  $\delta_i$  and a noise term  $\epsilon_{ij}$ . The noise terms have an iid extreme value distribution, and the consumer chooses the product in  $\mathcal{A}$  for which  $\delta_i + \epsilon_{ij}$  is greatest. It can be shown that the probability of a given consumer choosing product  $i$  from the assortment  $\mathcal{A}$  is

$$\Pr [ \text{choose } i \mid \text{assortment } \mathcal{A} ] = \frac{\exp \delta_i}{\sum_{h \in \mathcal{A}} \exp \delta_h} \quad (1)$$

The value of this equation does not change when each  $\delta_i$  is perturbed by the same constant; hence, we may assume without loss of generality that  $\sum_i \delta_i = 0$ .

A known issue with the MNL choice model is the assumption that introducing a new item into the assortment does not alter the preferability of the items already in the assortment relative to one another. This assumption, known as the independence of irrelevant alternatives, is violated when some products are near substitutes. The *nested* multinomial logit model (NMNL) addresses this issue by grouping similar products into nests, notated  $k \in \{1 \dots s\}$ . Under NMNL, the customer first chooses a nest  $k$ , then a product  $i$ . Both choices are made using MNL. Each nest's preferability parameter is written  $\sigma_k$ , while the products within the nest have preferability parameters  $\delta_{ki}$ . For convenience, assume each nest contains  $n$  products. Hence  $\mathcal{N} = \{1 \dots s\} \times \{1 \dots n\}$ .

We can represent a given assortment  $\mathcal{A}$  as a list of (nest, product) tuples  $(k, i)$ . The assortment may span the  $s$  nests and  $sn$  products in any arbitrary way. Let  $\mathcal{A}^{\text{nests}}$  denote the set of nests represented in the assortment  $\mathcal{A}$ , and let  $\mathcal{A}_k^{\text{products}}$  denote the set of products that are in both nest  $k$  and the assortment  $\mathcal{A}$ . Then the probability that a given customer chooses the product  $(k, i)$  is

$$\Pr [\text{choose } (k, i) \mid \text{assortment } \mathcal{A}] = \frac{\exp \sigma_k}{\sum_{l \in \mathcal{A}^{\text{nests}}} \exp \sigma_l} \cdot \frac{\exp \delta_i}{\sum_{h \in \mathcal{A}_k^{\text{products}}} \exp \delta_h} \quad (2)$$

where we may assume without loss of generality that  $\sum_k \sigma_k = 0$  and  $\sum_i \delta_{ki} = 0, \forall k$ .

### 3. Static parameter estimation

Consider the static parameter estimation task in the MNL choice model.  $m$  consumers, indexed by  $j$ , were shown assortments  $\mathcal{A}_j \subseteq \mathcal{N}$  and asked to pick their favorite product  $y_j$ . For convenience, abbreviate the preferability parameter associated with consumer  $j$ 's choice from  $\delta_{y_j}$  to  $\delta_j$ .

The likelihood of the  $m$  observations with respect to the parameters  $\delta$  is

$$L(\delta) = \prod_{j=1}^m \frac{\exp \delta_j}{\sum_{h \in \mathcal{A}_j} \exp \delta_h} \quad (3)$$

which is log-concave in  $\delta$ .

Now consider the NMNL choice model. Each consumer  $j$  chooses a product from an assortment  $\mathcal{A}_j$ , and we may write the consumer's choice as  $(x_j, y_j)$ , where  $y_j$  is the product chosen and  $x_j$  is the nest to which  $y_j$  belongs. (Note that this requires that the nests be designated in advance.) Again,  $\mathcal{A}_j$  as a list of (nest, product) tuples  $(k, i)$ . Let  $\mathcal{A}_j^{\text{nests}}$  denote the set of nests represented in the assortment  $\mathcal{A}_j$ , let  $\mathcal{A}_j^{\text{products}}$  denote the set of products that are in both  $x_j$  and the assortment  $\mathcal{A}_j$ , and abbreviate the parameters  $(\sigma_{x_j}, \delta_{y_j})$  to  $(\sigma_j, \delta_j)$ . Then the likelihood function is

$$L(\sigma, \delta) = \prod_{j=1}^m \frac{\exp \sigma_j}{\sum_{l \in \mathcal{A}_j^{\text{nests}}} \exp \sigma_l} \cdot \frac{\exp \delta_j}{\sum_{h \in \mathcal{A}_j^{\text{products}}} \exp \delta_h} \quad (4)$$

Computational tools for static estimation of variants of the MNL choice model are widely available (Croissant, n.d.).

## 4. Parameter estimation in the data stream framework

To frame the discussion of dynamic parameter estimation, I consider an online setting called the data stream. In the data stream, we are only allowed to store our *current* parameter estimate  $\delta$  and a state variable  $t$  representing the number of observations used to form the current estimate. When a new observation  $(\mathcal{A}, y)$  comes in, we can update the stored variables, but we cannot store the observation itself.

### 4.1. The MNL parameter update

First, consider the estimation of MNL choice parameters. At time  $t$ , the parameter estimate is  $\delta^{(t)}$ . A new observation arrives in which a consumer with consideration set  $\mathcal{A}_t$  chose product  $y_t$ .

Let  $\delta_{\mathcal{A}}$  denote the vector of preferability parameters corresponding to the products in  $\mathcal{A}$ . Let  $e_y$  denote the vector of the same length with a one at the index corresponding to  $y$  and zeros elsewhere. Finally, let

$$z^{(t)} = e_{y_t} - \frac{1}{\sum_{i \in \mathcal{A}_t} \exp \delta_i^{(t)}} \exp. \delta_{\mathcal{A}_t}^{(t)} \quad (5)$$

where  $\exp. x = (\exp x_1, \dots, \exp x_n)$ .

I propose the following *MNL parameter update*:

$$\delta_{\mathcal{A}_t}^{(t+1)} = \delta_{\mathcal{A}_t}^{(t)} + \frac{\alpha}{t^r} z^{(t)} \quad \delta_{\mathcal{N} \setminus \mathcal{A}_t}^{(t+1)} = \delta_{\mathcal{N} \setminus \mathcal{A}_t}^{(t)} \quad (6)$$

where  $\alpha > 0$  and  $r \in (0, 1]$  are step parameters.

**Proposition 1.** *If the observations are drawn from an MNL choice model and each consideration set appears with nonzero probability, then the MNL parameter update converges in expectation to the true parameters  $\delta$ .*

The intuition behind this parameter update is as follows. Seeing that  $y_t$  was chosen from  $\mathcal{A}_t$ , we

- Increase the estimated preferability of the item that was picked,
- Decrease the estimated preferability of the items that were rejected, and
- Leave unchanged the preferability of items not in the presented assortment.

Notice that  $\sum_{i=1}^n \delta_i$  remains constant under the parameter update, because  $\sum_i z_i = 0$ .

#### 4.2. A brief example

Suppose the parameter estimate is  $\delta^{(1)} = (-2, 1, 3, -2)$  and we observe a consumer choose item  $y_1 = 4$  from the consideration set  $\mathcal{A}_1 = \{2, 4\}$ . Then when  $\alpha = 1$ ,

$$\delta_{\mathcal{A}_1}^{(1)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad e_{y_1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (7)$$

and

$$\delta_{\mathcal{A}_2}^{(2)} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} + \frac{1}{1} \left( \begin{bmatrix} 0 \\ 1 \end{bmatrix} - \frac{1}{e^1 + e^{-2}} \begin{bmatrix} e^1 \\ e^{-2} \end{bmatrix} \right) = \begin{bmatrix} 0.05 \\ -1.05 \end{bmatrix} \quad (8)$$

$$\implies \delta^{(2)} = (-2, 0.05, 3, -1.05) \quad (9)$$

is the updated parameter estimate.

#### 4.3. Why the parameter update converges

Consider the multivariate root-finding problem

$$\text{find } \theta^* : F(\theta^*) = 0$$

where the Jacobian of  $F : \mathbb{R}^n \mapsto \mathbb{R}^n$  is negative definite (ND), and the iterative scheme

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\alpha}{t^r} F(\theta^{(t)})$$

where  $\alpha > 0$  and  $r \in (0, 1]$  are step parameters. Such an iterative scheme is known to converge to the root in a variety of scenarios, such as finding the maximum of a concave function whose gradient is  $F$  using gradient ascent. However, Robbins and Monro (1951) showed (in the univariate case) that one can replace  $F(\theta)$  with an *experiment*  $\hat{F}(\theta)$  satisfying  $\mathbb{E}[\hat{F}(\theta)] = F(\theta)$ . The same iterative scheme exhibits *stochastic convergence*, meaning that with probability one,

$$\lim_{t \rightarrow \infty} \mathbb{E}[F(\theta^{(t)})] = \theta^*$$

Many convergence rate results also generalize to stochastic interpretations.

A classic example of a Robbins–Monro algorithm is stochastic gradient descent, which replaces the sum of error terms in a loss function with the errors of a randomly chosen subset of samples. In fact, the MNL parameter update can be regarded as a form of stochastic gradient descent for the log

likelihood function, because when the single observation  $(\mathcal{A}_t, y_t)$  constitutes the full dataset, the maximum likelihood estimator for the preferability of the products in  $\mathcal{A}_t$  is precisely  $e_{y_t}$ . I offer a somewhat more targeted argument, however, in the following subsection.

#### 4.4. The MNL parameter update is a Robbins–Monro algorithm

In the MNL parameter estimation task, it is easy to see that for a given assortment  $\mathcal{A}$ , the expected value of the one-hot vector  $e_y$  is the probability weight of each item in the assortment:

$$\mathbb{E}[e_y] = \frac{1}{\sum_{i \in \mathcal{A}} \exp \delta_i} \exp. \delta_{\mathcal{A}} \quad (10)$$

The counterpart to  $F$  is the expected value of the random function  $z$ .

$$\mu(z) = \mathbb{E}[z] = \mathbb{E} \left[ e_y - \frac{1}{\sum_{i \in \mathcal{A}} \exp \delta_i} \exp. \delta_{\mathcal{A}} \right] \quad (11)$$

Assuming that each consideration set  $\mathcal{A} \in \mathcal{N}!$  is presented with nonzero probability,  $\mu(z) = 0$  if and only if  $\delta$  is correct. Observe that the Jacobian of  $\mu(z)$  in  $\delta$  is ND.

The counterpart to the experiment  $\hat{F}$  is the realization  $z^{(t)}$ , which is alike in expected value.

#### 4.5. The NMNL parameter update

The NMNL parameter update is analogous to the MNL parameter update. Each observation may be written as a tuple  $(\mathcal{A}_t, x_t, y_t)$ . Define  $\mathcal{A}_t^{\text{nests}}$  and  $\mathcal{A}_t^{\text{products}}$  as above, and let  $e_x$  and  $e_y$  denote the vectors having the same lengths as  $\mathcal{A}_t^{\text{nests}}$  and  $\mathcal{A}_t^{\text{products}}$ , where  $e_x$  has a one at the index corresponding to  $x$ ,  $e_y$  has a one at the index corresponding to  $y$ , and the other entries are zero. Finally, let

$$w^{(t)} = e_{x_t} - \frac{1}{\sum_{i \in \mathcal{A}_t^{\text{nests}}} \exp \sigma_i} \exp. \sigma_{\mathcal{A}_t^{\text{nests}}}^{(t)} \quad (12)$$

$$z^{(t)} = e_{y_t} - \frac{1}{\sum_{i \in \mathcal{A}_t^{\text{products}}} \exp \delta_i} \exp. \delta_{\mathcal{A}_t^{\text{products}}}^{(t)} \quad (13)$$

The parameters are updated as follows.

$$\sigma_{\mathcal{A}_t^{\text{nests}}}^{(t+1)} = \sigma_{\mathcal{A}_t^{\text{nests}}}^{(t)} + \frac{\beta}{t^r} w^{(t)} \quad \sigma_{\mathcal{N} \setminus \mathcal{A}_t^{\text{nests}}}^{(t+1)} = \sigma_{\mathcal{N} \setminus \mathcal{A}_t^{\text{nests}}}^{(t)} \quad (14)$$

$$\delta_{\mathcal{A}_t^{\text{products}}}^{(t+1)} = \delta_{\mathcal{A}_t^{\text{products}}}^{(t)} + \frac{\alpha}{t^r} z^{(t)} \quad \delta_{\mathcal{N} \setminus \mathcal{A}_t^{\text{products}}}^{(t+1)} = \delta_{\mathcal{N} \setminus \mathcal{A}_t^{\text{products}}}^{(t)} \quad (15)$$

where  $\beta, \alpha > 0$  and  $r \in (0, 1]$  are step parameters.

**Proposition 2.** *If the observations are drawn from an NMNL choice model and each consideration set appears with nonzero probability, then the NMNL parameter update converges to the true parameters  $\sigma$  and  $\delta$ .*

## 5. Computational examples

I demonstrate the tractability of the dynamic parameter update through three simulations. The first simulation, whose results are shown in Figure 1, tests the MNL parameter update, while the second simulation considers the NMNL case. In both simulations, a small number of products was chosen to yield a legible graph; the parameter update formula is computationally tractable for large numbers of products. The third simulation compares maximum likelihood estimation over a full dataset with the average of the final 1000 iterations of a dynamic estimation of the same. While the accuracy is necessarily lower in the latter case, it is acceptable given the savings in computation time. In all cases, it is possible to tune the step parameters  $\alpha$ ,  $\beta$ , and  $r$  to obtain smoother, slower convergence. The Julia code for all three simulations is available on Github at <https://github.com/maxkapur/MultinomialLogit>.

## 6. Conclusion

Recent research in choice models has undergone two shifts in emphasis: one toward nonparametric choice models, and another toward the estimation thereof in a dynamic data setting. This paper has attempted to decouple these by considered the dynamic estimation of *parametric* choice models in the MNL family. Preliminary results suggest that the computational advantage of parametric choice models is also evident in the dynamic data setting. Whereas the algorithms suggested by Ho-Nguyen and Kılınç-Karzan (2021) require solving a combinatorial optimization subproblem at each iteration,

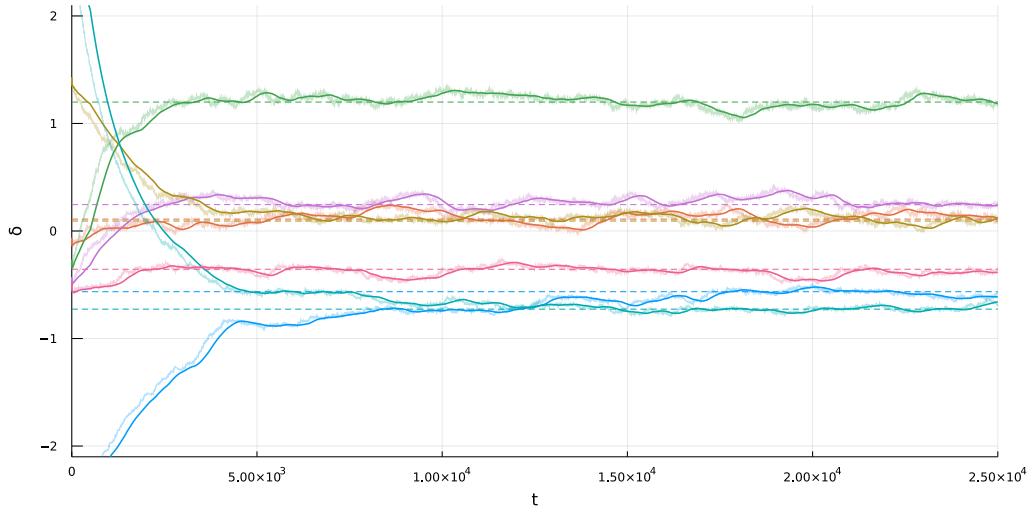


Figure 1: Dynamic estimation of MNL choice parameters for a collection of  $n = 7$  products. True preferability parameters  $\delta$  and the initial estimate  $\delta^{(1)}$  were drawn independently from a normal distribution and centered at zero. At times  $t = 1 \dots 25000$ , a random assortment consisting of between two and five products was drawn, a consumer chose a product using the MNL choice model, and the parameter estimates were updated according to the MNL parameter update (Equation 6). The parameter estimate at each iteration is shown in a light stroke, while the darker lines show a moving average over the previous 500 estimates. The step parameters are  $\alpha = 0.01$  and  $r = 0.05$ .

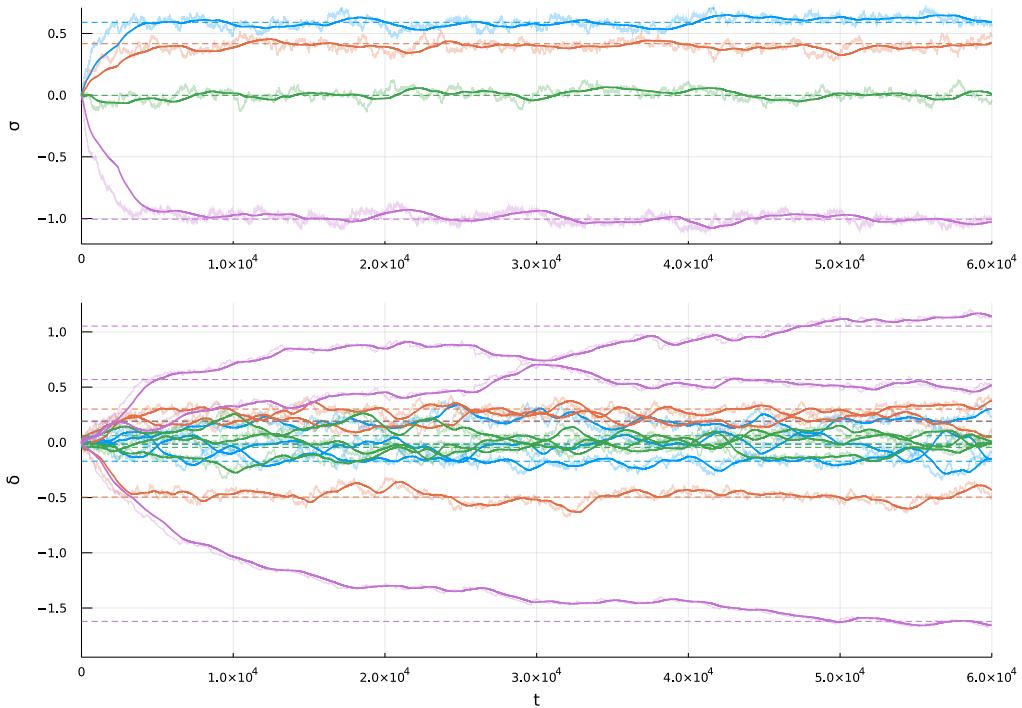


Figure 2: Dynamic estimation of MNL choice parameters for a collection of  $sn = 12$  products divided across  $s = 4$  nests. True preferability parameters  $\sigma$  and  $\delta$  were drawn independently from a normal distribution and centered at zero, while the initial parameter estimates were set to zero. At times  $t = 1 \dots 60000$ , a random assortment consisting of between two and five products was drawn, a consumer chose a product using the MNL choice model, and the parameter estimates were updated according to the NMNL parameter update (Equation 14). The parameter estimate at each iteration is shown in a light stroke, while the darker lines show a moving average over the previous 1200 estimates. The step parameters are  $\beta = 0.01$ ,  $\alpha = 0.03$ , and  $r = 0.05$ . The first pane shows the time series of  $\sigma$  estimates, while the second shows the entries of  $\delta$ , color-coded by nest membership.

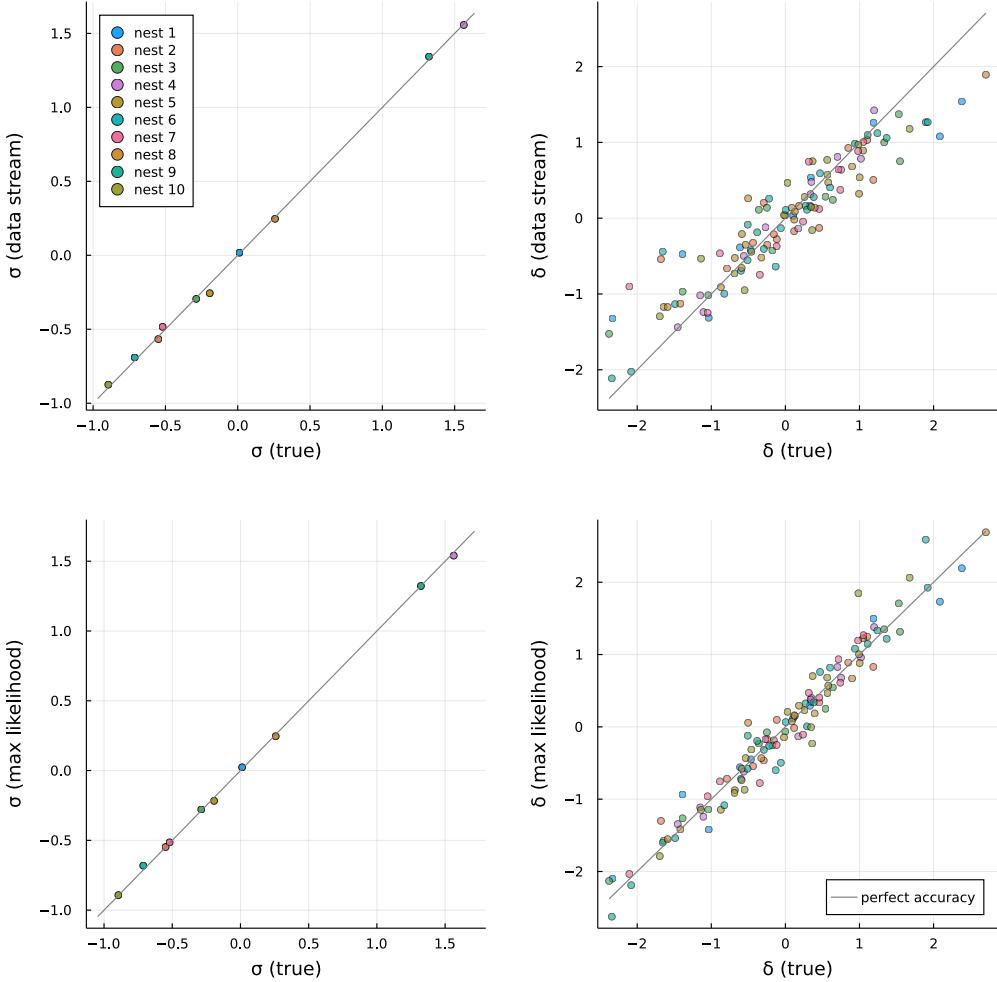


Figure 3: Performance of the dynamic parameter estimate versus maximum likelihood estimation. Preferability parameters for  $sn = 120$  products divided into  $s = 10$  nests were drawn from the normal distribution and centered at zero.  $m = 50000$  assortments and consumer choices were generated according to the NMNL model. In the upper two panes, the observations were treated as a time series, and I report the average of the final 1000 estimates produced by the parameter update with step parameters  $\beta = 0.01$ ,  $\alpha = 0.03$ , and  $r = 0.05$ . In the lower two panes, maximum likelihood estimation was used to estimate the parameters. Dynamic parameter estimation offers acceptable accuracy at substantially lower computational cost.

the complexity of parameter update step proposed here is linear in the number of items and does not require subiterative computation. Of course, all parametric choice models impose assumptions on the underlying distribution of customer preference orders that may not be appropriate in all settings, and the larger debate about the relative merits of parametric and nonparametric models is far from settled.

A natural extension of the ideas proposed here is parameter estimation in other choice models within and beyond the MNL family.

## References

- Croissant, Yves. “Estimation of multinomial logit models in R: The `mlogit` Packages.” [https://www-eio.upc.es/teaching/madt/apunts/mlogit\\_teoria.pdf](https://www-eio.upc.es/teaching/madt/apunts/mlogit_teoria.pdf).
- Ho-Nguyen, Nam and Fatma Kılınç-Karzan. 2021. “Dynamic Data-Driven Estimation of Nonparametric Choice Models.” *Operations Research: Articles in Advance*. <https://doi.org/10.1287/opre.2020.2077>.
- Robbins, Herbert and Sutton Monro. 1951. “A Stochastic Approximation Method.” *The Annals of Mathematical Statistics* 22(3): 400-407. <https://doi.org/10.1214/aoms/1177729586>.