CIRRELT-2020-32

On the Estimation of Discrete Choice Models to Capture Irrational Customer Behaviors

Sanjay Dominik Jena
Andrea Lodi
Claudio Sole

August 2020

Bureau de Montréal

Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec

Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

On the Estimation of Discrete Choice Models to Capture Irrational Customer Behaviors

Sanjay Dominik Jena[1,2], Andrea Lodi[1, 3], Claudio Sole[1,4,*]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] École des Sciences de la Gestion, Université du Québec à Montréal, jena.sanjay-dominik@uqam.ca

[3] Canada Excellence Research Chair in Data-Science for Real-time Decision-Making, Polytechnique Montréal, andrea.lodi@polymtl.ca

[4] Département de mathématiques et de génie industriel, Polytechnique Montréal

Abstract. Many planning problems in Operations Management require a prior estimation of the demand of the products or services involved. While most of the literature has focused on discrete choice models based on the Random Utility Maximization framework, several works on behavioral economics have provided strong empirical evidence of irrational choice behaviors incompatible with such a framework, such as halo effects. Random Utility Maximization models may therefore lead to inaccurate estimates of product demands and, when used to optimize assortment or inventory decisions, to sub-optimal revenues. Hence, more general choice models, overcoming such limitations, have been proposed. However, the estimation of these models remains challenging, as a delicate balance must be struck between the flexibility of the model, its inherent risk of overfitting, and the computational tractability of the estimation procedure. In this work, we address these difficulties by proposing an estimation method for the recently proposed Generalized Stochastic Preference choice model. This choice model subsumes the family of Random Utility Maximization models and is capable of capturing halo effects. Specifically, we show how to use partially-ranked preferences to model irrational customer types, and how to efficiently retrieve them from data. Our estimation procedure is based on column generation, where relevant customer types are discovered in an effective way, by exploiting a tree-like data structure to represent a given set of preferences. An extensive set of experiments assesses the predictive accuracy of the proposed approach, comparing it against rank-based methods with only rational preferences and with a more general benchmark from the literature. Our results show that irrational preferences allow to significantly enhance predictive accuracy on both synthetic and real datasets in the presence of irrational choice behaviors. Finally, our proposed model allows to generalize well on both rational and irrational instances, therefore favorably comparing against existing general choice models.

Keywords: Choice modeling, Halo effects, substitution effects, rank-based model

Acknowledgements. We would like to acknowledge the Wharton Customer Analytics (WCA) research group and the Majid Al Futtaim (MAF) data \& analytics team for the insightful discussions.

---

* Corresponding author: claudio.sole@polymtl.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2020

1. Introduction

Many planning problems in Operations Management require a prior estimation of the demand of the products or services involved. Most often, a predictive model must be learned from historical data representing the choice behavior of an agent faced with a discrete set of alternatives, called the *offer set*. Such a model can then be used as a subroutine in the solution of prescriptive tasks involving, for example, assortment or inventory decisions. A common assumption when dealing with demand estimation is to consider product demands as independent from each other, resulting in the independent demand model (see, e.g., Strauss et al. 2018, Talluri and Van Ryzin 2004). However, it is well known that this assumption does not hold in many real-life scenarios and that product demands interact through substitution and halo effects. In general, we consider product A a substitute of product B if the presence of A in the offer set decreases the probability of B being chosen. On the contrary, we refer to an halo effect if the presence of A in the offer set increases the attractiveness of B, and thus its likelihood of being chosen. Discrete choice models have been widely adopted to model substitution. Among them, the family of choice models that received the most attention in the literature is undoubtedly the one of *Random Utility Maximization* (RUM) models (Thurstone 1927, Block and Marschak 1959, Luce 1959). Choice models belonging to the RUM family assume that a random utility is assigned to every alternative. Utilities are modeled as random variables, and different choices about their distribution lead to different choice models. When faced with an offer set, the decision maker samples a vector of utilities and picks the option with the highest one, so as to maximize her expected payoff. The Multinomial Logit (MNL) model is arguably the most famous RUM choice model. Its popularity stems from the facts that it can be efficiently estimated, it is interpretable and, when used for decision making, it allows to benefit from appealing theoretical and computational properties. The Multinomial Logit model lacks, however, in terms of flexibility. In particular, it obeys the axiom of Independence of Irrelevant Alternatives (IIA) (Arrow 1951), and is therefore incapable of capturing complex substitution behaviors. Thus, many models have been proposed in the last decades to overcome its limitations, such as the Nested Logit model, the Mixed Multinomial Logit model and, more recently, the Markov chain (Blanchet et al. 2016) and the Rank-based (Farias et al. 2013) choice models, each providing different trade-offs between flexibility and tractability. However, these models all belong to the RUM family and therefore obey the so-called *Regularity* assumption, which states that the introduction of an option in the offer set cannot increase the probability of another alternative being chosen. Hence, they cannot be used as they are to capture halo effects. Nevertheless, many studies in the literature of behavioral economics corroborated the reproducibility and robustness of this type of choice behaviors (see, e.g., Simonson 1989, Huber et al. 1982), incompatible with the theory of utility maximization and therefore referred to as *irrational*. For example, in the context of grocery shopping, when two complementary products (e.g., pasta and tomato sauce) are present in the assortment, the perceived attractiveness of both is likely to increase. One may also observe asymmetric, or *decoy* effects (see, e.g., Ariely 2008) when the addition of an option (the decoy) to the offer set increases the choice probability of another alternative perceived as better. This motivated the recent interest in more general choice models, capable of overcoming the limitations of the RUM framework and of capturing more complex choice behaviors. Unfortunately, many of these choice models lack efficient estimation schemes, and their performance on non-RUM instances has not been well understood yet (see, e.g., Jagabathula and Rusmevichientong 2019). Also, the minimal assumptions these models make about the distribution of choice probabilities may increase the risk of capturing spurious patterns from data, i.e., overfit. For example, results from Chen et al. (2019) and Chen and Mišic (2019), where the authors propose an irrational choice model based on decision trees, confirm that such models may well capture irrational behaviors, but may struggle with rational ones. Finding the delicate balance between flexibility and predictive accuracy is therefore of crucial importance for the practical utility of such models. The *Generalized Stochastic Preference* (GSP) choice model, an extension of rank-based choice models introduced by Berbeglia (2018) to capture halo effects, is one of the recently proposed models that fits into this stream of literature. Despite being theoretically attractive, the estimation of the GSP choice model poses significant challenges both from the computational and predictive points of view. The authors suggest that estimation procedures originally developed for rational rank-based choice models (see, e.g., Farias et al. 2013, van Ryzin and Vulcano 2015, Bertsimas and Mišic 2016) may be adapted to their irrational choice model. Nevertheless, no empirical study has been reported in order to assess the estimation efficiency and predictive accuracy of the GSP choice model. Given the flexibility of the model, its estimation avoiding overfitting is a challenging task.

Contributions. In this work, we propose an estimation method for the GSP choice model. Specifically, we show how to use partially-ranked preferences to model irrational customer behaviors, and how to efficiently estimate them from choice data by adapting the column generation approach proposed by Jena et al. (2020). Partially-ranked preferences allow us to circumvent several difficulties regarding the adaption of estimation methods for strictly ranked preferences. In particular, our objective is to train the choice model so as to maximize its predictive accuracy. This is different from Farias et al. (2013), who focus on worst-case revenue prediction for a given assortment of items. Also, our estimation method can easily handle both rational and irrational customer behaviors. In contrast, it is not clear how the Mixed Integer Programming (MIP) formulation of the Market Discovery subproblem from van Ryzin and Vulcano (2015) should be adapted to allow for the discovery of irrational preferences. Finally, the *Growing Preference Tree* (GPT) algorithm of Jena et al. (2020) provides a strong computational advantage in terms of scalability, especially important when dealing with irrational customer behaviors (discussed in the following) and generalizes well to unseen offer sets when tested on RUM instances. The application of partially-ranked preferences for tackling the estimation of generalized stochastic preferences thus looks promising. An appealing property of our approach stems from the fact that the irrationality, and thus the flexibility of the choice model is increased in an adaptive, data-driven way. By increasing the set of possible customer behaviors only when required to better explain the given data, we may limit the risk of overfitting and speed up the estimation procedure. We run an extensive set of experiments to assess the predictive performance of the proposed choice model. Using the methodology delineated by Jagabathula and Rusmevichientong (2019), we characterize the *rationality loss* of both generated and real instances. This allows us to observe that irrational customer types can significantly improve predictive accuracy on instances presenting halo effects among alternatives. We also compare our approach against the Pairwise Choice Markov Chain (PCMC) model proposed by Ragain and Ugander (2016), a general choice model able to capture complex substitution and halo effects. In the experiments reported by the authors, this choice model was shown to outperform both the Multinomial Logit and the Mixed Multinomial Logit in terms of predictive accuracy. Moreover, the PCMC choice model requires only little parameter tuning, thus allowing us to fit it with minimal effort on the large set of instances used in our experiments. Our results show that rank-based approaches generally outperform the PCMC choice model on both rational and irrational instances.

Organization of the paper. In Section 2, we review the literature on irrational choice models. In Section 3, we introduce the GSP choice model from Berbeglia (2018) and our corresponding partially-ranked representation. We show how to estimate the proposed choice model in Section 4. The numerical results of our experiments on both synthetic and real instances are reported in Section 5. Finally, concluding remarks are reported in Section 6.

2. Related work

In order to define the notion of a *rational* agent, most economists rely on a set of consistency principles of rationality, which includes, among others, the aforementioned Regularity assumption and the more famous axiom of Independence of Irrelavant alternatives (IIA). This set of assumptions aims at describing how a rational agent is supposed to make her decisions across different offer sets. However, a vast body of literature has provided strong empirical evidence of choice behaviors incompatible with the theory of rational choice (we refer to Rieskamp et al. (2006) for an excellent overview on the topic). The RUM framework is flexible enough to explain most of these choice behaviors, but cannot account for violations of the Regularity assumption. To overcome such limitation, more general theories of choices have been developed in psychology, such as Decision Field theory (Busemeyer and Townsend 1993, Roe et al. 2001) and the Leaky competing accumulator model (Usher and McClelland 2004). These models belong to the broader class of Sequential Sampling models, which mimic the evolution of the decision-making process over time, and can account for violations of the rationality principles, including the Regularity one. They lack, however, practical estimation algorithms, and are usually adopted from a descriptive point of view more than a predictive one. Other works, such as Tversky and Simonson (1993) and Rooderkerk et al. (2011), embed alternatives into an attribute space, where context-dependent features are computed in order to determine the utility of each of the alternatives. These approaches have usually been applied to small, controlled experiments, and rely on the existence of two metric features, along which customer preferences are supposed to monotonically increase or decrease. This is a key difference with respect to our approach, where no item feature is supposed to be given. Decomposing the utility into two components, item-specific and context-dependent, is also the starting point of Maragheh et al. (2018) and Seshadri et al. (2019), who propose a second-order extension of the MNL model in order to capture positive pairwise product interactions. However, these models do not subsume the RUM framework and thus, as pointed out by Jagabathula and Rusmevichientong (2019), are not guaranteed to provide a better fit than RUM methods, even when applied to irrational instances. The same limitation holds for other models such as the General Attraction Model from Gallego et al. (2014), the Perception-adjusted choice model (Echenique et al. 2018) and the General Luce Model (Echenique and Saito 2019). Feng et al. (2018) propose a welfare-based framework, which subsumes the RUM framework and can be used to obtain choice models able to capture violations of the regularity assumption. The estimation of these choice models, however, is left by the authors as an open research question. Another general approach for which no empirical result has been reported is the Generalized Stochastic Preference choice model (Berbeglia 2018), an extension of rank-based choice models (see, e.g., Farias et al. 2013, van Ryzin and Vulcano 2015) that allows for irrational customer behaviors. This model subsumes the RUM family of models and generalizes the non-RUM approach from Kleinberg et al. (2017) by allowing for heterogeneity in customer preferences. Despite its flexibility, the GSP choice model imposes some structure on the choice probabilities, and some examples are provided by the authors describing choice behaviors that do not belong to the GSP class. Ragain and Ugander (2016) propose the Pairwise Choice Markov Chain model, where each alternative is represented as a node of a continuos time Markov Chain. Given an offer set, the choice probabilities are given by the stationary distribution of the sub-chain consisting of the nodes indexed by the available alternatives. Although the PCMC choice model is able to capture both substitution and halo effects, it obeys the axiom of uniform expansion introduced by Yellott (1977). The authors argue that such property may be desirable in the context of discrete choice modeling.

Some more general choice models have been proposed in the literature, which are able to represent *any* discrete choice function. In particular, Osogami and Otsuka (2014) propose an extension of the MNL model aming at capturing high-order product interactions. They show that the resulting model can be represented as a Restricted Boltzman Machine (RBM), a probabilistic graphical model whose units are divided into two groups, visible and hidden. Visible units are used to encode a binary representation of the offer set and of a given choice, while hidden units learn a latent representation of the input. Given enough hidden units, these models can represent any sort of irrational behavior. An approach based on tree ensembles has recently been proposed by both Chen et al. (2019) and Chen and Mišic (2019), who show that any discrete choice model can be represented as a distribution over decision trees.

As previously mentioned, choice models with rather flexible structures pose some crucial challenges, whose solution greatly impacts the predictive accuracy of the trained choice models. In particular, one needs to balance between flexibility of the choice model, tractability of its estimation procedure, and risk of overfitting when limited amount of data is available. Chen and Mišic (2019) and Chen et al. (2019) tackle these difficulties by proposing regularization methods whose effect is to restrict the search space in a principled way. It may be argued, nevertheless, that less general choice models may be more effective in exploring search spaces that are smaller by definition, and that imposing some structure on the choice probabilities may provide important inductive bias to improve generalization over unseen offer sets when limited amount of data is available. This observation motivates the focus of this paper. In particular, we propose an estimation method for the Generalized Stochastic Preference choice model, which is flexible enough to subsume the RUM family of models and to capture halo effects, but still imposes some structure on the choice probabilities. Moreover, we compare the generalization performances of the resulting model against the PCMC choice model, another approach from the literature making minimal assumptions about the distribution of choice probabilities among alternatives.

We conclude this section by mentioning an interesting line of work from the machine learning community, proposing general approaches based on Neural Networks to approximate the complex, high-order interactions among alternatives (see, e.g., Pfannschmidt et al. 2019, Rosenfeld et al. 2020, Mottini and Acuna-Agost 2017). Despite their flexibility, however, these models have only been applied to settings with product features and large number of training offer sets. Their adaptation to a setting close to ours, where no item featurization is given and the amount of offer sets seen at training time is relatively small, has not been explored yet and does not seem trivial.

3. The choice model

Consider a set of products $\mathcal{N} = \{0, ..., N-1\}$, with label 0 representing the *no-purchase* option. Further, let σ denote both a subset of products in $\mathcal{N}$, and a linear order defined over such products, so that the rank (or position) of product j according to σ is given by $\sigma(j) \geq 0$. A *Generalized Stochastic Preference* (Berbeglia 2018) consists of a ranking $\sigma \subseteq \mathcal{N}$, and an index i, with $0 \leq i < |\sigma|$. When faced with an offer set $S \subseteq \mathcal{N}$, a customer $C(\sigma, i)$ picks the alternative ranked i^{th} in the subsequence of σ that only contains items also available in S. Equivalently, let $\sigma_S \subseteq \sigma$ denote the sequence of products obtained removing from σ every product $j \notin S$. The customer will then choose product j^* so that $\sigma_S(j^*) = i$. If $|\sigma_S| \leq i$, the customer will leave without any purchase. The particular case of $i = 0$ corresponds to customers who always pick their favorite (i.e., highest ranked) product among the available ones. For this reason, we refer to customers $C(\sigma, 0)$ as *rational* behaviors, and to the index i of a generalized stochastic preference as its *irrationality level*. The GSP choice model is then defined by a probability distribution $\lambda \in \mathbb{R}^K$ over K customer types $\{C_k(\sigma_k, i_k)\}_{k=1}^K$. It should be noticed that, since every RUM choice model can be equivalently represented as a distribution over *rational* stochastic preferences (see, e.g., Block and Marschak 1959), the GSP choice model naturally subsumes the RUM family of models. Further, Berbeglia (2018) shows how the GSP choice model can be used to explain the results of several controlled experiments from the literature in behavioral economics providing violations of the regularity assumption. From a modeling perspective, we also note that including the 0 (i.e., no-purchase) option among the ranked alternatives has a useful implication in practice. In particular, contrary to the original formulation in Berbeglia (2018), this allows us to capture violations of the regularity assumption also for the no-purchase option (we refer to Appendix A for more details). Several studies, indeed, have shown that customers' willingness to purchase and overall satisfaction may decrease in the presence of too many alternatives among which a choice has to be made (see, e.g., Iyengar and Lepper 2000, Schwartz 2004).

The estimation of the GSP choice model poses significant computational challenges, given that the space of rational customer types alone is factorially large. Estimation procedures developed for rational, rank-based models such as those from van Ryzin and Vulcano (2015) and Bertsimas and Mišic (2016) cannot be easily adapted to account for learning irrational preferences, nor does their scalability look promising to tackle the even bigger search space implied by the presence of irrational behaviors (see, e.g., Berbeglia et al. 2018, Jena et al. 2020). For these reasons, we decided to adopt the partially-ranked framework from Jena et al. (2020) to represent generalized stochastic preferences. Besides providing a more intuitive, behavioral representation of an agent's decision process, partially-ranked preferences allow for fast estimation schemes and have been shown to generalize well on unseen offer sets. Starting from the observation that, for rational customer behaviors, low-ranked alternatives have a relatively low impact in explaining choice data, Jena et al. (2020) propose to strictly rank only few, relevant alternatives for each preference list, while allowing for ties among the rest of them. Alternatives with the same rank may then be grouped into so-called *indifference sets*.

Building on that work, we thus define a *partially-ranked preference with irrationality* $C(P(\sigma), I(\sigma), i)$, where alternatives belonging to σ are further distinguished into a set $P(\sigma) \subseteq \mathcal{N}$ of strictly ranked alternatives and an indifference set $I(\sigma) \subseteq \mathcal{N} \setminus P(\sigma)$, so that $\sigma(j) = \sigma(j')$ for all $j, j' \in I(\sigma)$, and $\sigma(j) < \sigma(j')$ for all $j \in P(\sigma), j' \in I(\sigma)$. The inclusion of $P(\sigma) \cup I(\sigma) \subseteq \mathcal{N}$ may be strict, in which case customers are assumed to form so-called *consideration sets* (see, e.g., Aouad et al. 2015, Jagabathula and Vulcano 2018). Note that the irrationality level of a partially-ranked preference must be smaller than the number of strictly ranked products, that is, $i < |P(\sigma)|$, since alternatives in the indifference sets all have the same rank. For ease of notation, let $P_S(\sigma) = P(\sigma) \cap S$ and $I_S(\sigma) = I(\sigma) \cap S$ denote the strictly ranked preference list and the indifference set, respectively, obtained after removing from σ every product not available in a given offer set S. A customer $C(P(\sigma), I(\sigma), i)$ will then pick the alternative ranked i^{th} in $P_S(\sigma)$ if $i < |P_S(\sigma)|$, or an alternative chosen uniformly at random in $I_S(\sigma)$ when $|P_S(\sigma)| \leq i < |P_S(\sigma) \cup I_S(\sigma)|$. When $i \geq |P_S(\sigma) \cup I_S(\sigma)|$, the customer leaves without any purchase.

offer set S	$P_S(\sigma)$	$I_S(\sigma)$	Choice$_{C_1}$	Choice$_{C_2}$
$\{2,5,1\}$	(2,5)	$\{1\}$	2	5
$\{2,1,4\}$	(2)	$\{1,4\}$	2	$\sim$Unif$\{1,4\}$
$\{1\}$	()	$\{1\}$	1	0
$\{1,4\}$	()	$\{1,4\}$	$\sim$Unif$\{1,4\}$	$\sim$Unif$\{1,4\}$

Table 1 **Choice behavior of two customers C_1 and C_2 across different offer sets.**

In Table 1, we give an example of the choice behavior of two hypothetical customers $C_1\big((2,3,5), \{1,4\}, 0\big)$ and $C_2\big((2,3,5), \{1,4\}, 1\big)$, who differ only for their irrationality level. As a consequence, for each offer set S, we have that $P_S(\sigma_1) = P_S(\sigma_2)$ and $I_S(\sigma_1) = I_S(\sigma_2)$.

4. Estimation procedure

Jena et al. (2020) have shown that *rational*, partially-ranked preferences can be efficiently learned from data. Although their *Growing Preference Tree* (GPT) algorithm was originally proposed for the estimation of rational preferences, it can be easily adapted to handle partially-ranked preferences with irrationality. In this section, we show how the GPT algorithm can be extended to estimate generalized partially-ranked preferences. We refer to Jena et al. (2020) for a deeper analysis of the algorithm performances on RUM instances, and for more implementation details.

We assume that training data is available in the form of T observations $\mathcal{T} = \{(S_t, c_t)\}_{t=1}^{T}$ with S_t and c_t representing the offer set and the choice, respectively, that have been observed in period t. Let $\mathcal{M}_{train} = \{S_1, ..., S_M\}$ denote the collection of offer sets over which choice data is available. We can further preprocess dataset $\mathcal{T}$ in order to obtain a vector of empirical probabilities $\boldsymbol{v} \in \mathbb{R}^{N \cdot M}$ so that, for each $j \in \mathcal{N}$ and $S \in \mathcal{M}_{train}$, the probability of item j being chosen from offer set S is given by $v_{j,S}$.

The GPT algorithm fits into the general column-generation framework proposed for the estimation of a general class of nonparametric choice models by van Ryzin and Vulcano (2015). In line with this framework, customer behaviors are represented as a choice matrix $\boldsymbol{A} \in \mathbb{R}^{(N \cdot M) \times K}$, encoding K behaviors for M offer sets, whose elements give the probability of customers choosing an item from a given offer set. In particular, based on the choice behaviors of a partially-ranked list with irrationality i defined in Section 3, the elements of the matrix $\boldsymbol{A}$ may be computed as follows:

$$A_{j,m}^{k} = \begin{cases} 1 & \text{if } j \in P_{S_m}(\sigma) \text{ and } j \text{ ranked } i^{th} \text{ in } P_{S_m}(\sigma), \\ \dfrac{1}{|I_{S_m}(\sigma)|} & \text{if } j \in I_{S_m}(\sigma) \text{ and } |P_{S_m}(\sigma)| \leq i < |P_{S_m}(\sigma) \cup I_{S_m}(\sigma)|, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Given a distribution $\boldsymbol{\lambda} \in R^K$ over the customer types, the predicted probability $x_{j,m}$ of a random customer choosing alternative j from the offer set S_m is then given by $x_{j,m} = \sum_k A_{j,m}^k \lambda_k$. One can thus define the *best* distribution $\boldsymbol{\lambda}$, that is, the one for which the predicted probabilities are the closest to the observed ones, and obtain $\boldsymbol{\lambda}$ by solving the following optimization problem:

$$\min_{\boldsymbol{\lambda}} \quad \mathcal{L}(\boldsymbol{x}, \boldsymbol{v}) \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{\lambda} = \boldsymbol{x} \tag{2b}$$

$$\boldsymbol{1}^T \boldsymbol{\lambda} = 1 \tag{2c}$$

$$\boldsymbol{\lambda} \geq 0. \tag{2d}$$

Here, $\mathcal{L}(\boldsymbol{x}, \boldsymbol{v})$ can be any convex loss function measuring the distance between the predicted probabilities $\boldsymbol{x}$ and the observed ones $\boldsymbol{v}$. For example, one may minimize the L_1 error between the two probability distributions, in which case we have

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}) = \sum_{S \in \mathcal{M}_{train}} \sum_{i \in S} |x_{i,S} - v_{i,S}|. \tag{3}$$

Minimizing the L_1 error generally leads to sparse models. Moreover, objective function (3) can be easily linearized (see, e.g., Bertsimas and Mišic 2016), thus leading to computationally effective solution methods.

The Kullback-Leibler divergence is another popular measure of the distance between two probability distributions. It is strictly convex and leads to the same solution as Maximum Likelihood Estimation (see, e.g., Jagabathula and Rusmevichientong 2019). It is computed as

$$\mathcal{L}(\boldsymbol{x},\boldsymbol{v}) = -\frac{1}{T}\sum_{S\in\mathcal{M}_{train}} T_S \sum_{i\in S} v_{i,s}\ log\frac{x_{i,S}}{v_{i,S}}, \tag{4}$$

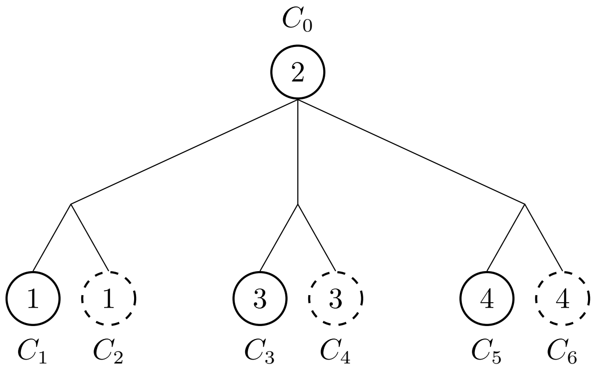
where T_S is the number of samples showing S as offer set.

Discovering new customer types. Again, it should be noticed that solving problem (2) over the factorially large set of possible customer types is not tractable. Hence, in practice, one may proceed by iteratively (i) solving problem (2) over a restricted set of behaviors, and (ii) identifying new, relevant behaviors by solving a subproblem. These steps are then repeated until the loss function achieves a small enough value ϵ_{th}. In particular, let $\boldsymbol{\alpha}\in\mathbb{R}^{N\cdot M}$ and $\nu\in\mathbb{R}$ denote the dual variables associated with constraints (2b) and (2c), respectively. The customers (i.e., preference sequences) worth adding to the model in order to improve its fit of the data are those whose corresponding choice vector $\boldsymbol{a}$, computed as in (1), has a negative reduced cost, i.e., $rc(\boldsymbol{a}) = -\boldsymbol{\alpha}\boldsymbol{a} - \nu < 0$. While finding new preference sequences with negative reduced costs generally tends to be computationally expensive, the GPT algorithm exploits the structure of partially-ranked preferences to efficiently identify such columns. Indeed, partially-ranked preferences allow for using an efficient tree-like data structure, where deeper levels correspond to behaviors with more refined ranked lists. Specifically, consider a given partially-ranked behavior C_k. A behavior C_j is considered a child, or sub-behavior of C_k, if $P(\sigma_j) = (P(\sigma_k), \ell)$ with $\ell \in I(\sigma_k)$. When searching for new customer behaviors, one may thus restrict the search for promising behaviors among the sub-behaviors of $\{C_1, ..., C_K\}$, at any given iteration.

In Figure 1, we give an example of the sub-behaviors obtained from customer $C((2,\{1,3,4\},0)$. The two main advantages of the exploration strategy employed by GPT are the following:

1. The number of strictly ranked objects increases in an adaptive, data-driven way, with more refined preference lists added only when needed. Besides allowing to avoid the computational burden of strictly ranking all the products in a preference list, Jena et al. (2020) show that the explanatory power of indifference sets tends to improve generalization on new offer sets.

2. Since the irrationality level i of a partially-ranked behavior is bounded by the number of strictly ranked products, i.e., $0 \le i < |P(\sigma)|$, the GPT search procedure prioritizes customers with low irrationality levels. This seems to be a behaviorally-plausible inductive bias, which may reduce the risk of overfitting, especially when only a limited amount of data is available.



Customer	P(σ)	I(σ)	i
C_0	(2)	$\{1,3,4\}$	0
C_1	(2,1)	$\{3,4\}$	0
C_2	(2,1)	$\{3,4\}$	1
C_3	(2,3)	$\{1,4\}$	0
C_4	(2,3)	$\{1,4\}$	1
C_5	(2,4)	$\{1,3\}$	0
C_6	(2,4)	$\{1,3\}$	1

Figure 1 **(Left) Tree representation of sub-behavior generation. A path in the tree corresponds to a sequence of strictly ranked products. Dashed nodes correspond to irrational behaviors. (Right) The corresponding explicit behaviors description.**

Observe that customers that differ only in their irrationality level, such as C_1 and C_2 in the example of Figure 1, generate the same sets of sub-behaviors. Hence, only one of them needs to be splitted at generation time. Further, in order to accelerate the generation of sub-behaviors, Jena et al. (2020) suggest to restrict the number of behaviors to split to a fixed amount δ, which may be sampled among $\{C_k : \lambda_k > 0\}$ according to the corresponding probabilities. In this case, we sample only once among the set of behaviors $\mathcal{C}_{P(\sigma),I(\sigma)} = \{C_k(P(\sigma_k), I(\sigma_k), i_k) : P(\sigma_k) = P(\sigma)$ and $I(\sigma_k) = I(\sigma), k = 1, ..., K\}$, with a probability $\tilde{\lambda}_{P(\sigma),I(\sigma)} = \sum_{k:C_k \in \mathcal{C}_{P(\sigma),I(\sigma)}} \lambda_k$.

From a practical standpoint, the decision of whether to incorporate irrational behaviors or not resolves to setting a single, binary hyperparameter. This makes model selection extremely easy in practice, allowing practitioners to understand whether going beyond RUM is actually needed for the given data. Also, other regularization methods based on limiting the irrationality levels of customer types are straightforward to implement. For example, one may decide to avoid generating customer types with an irrationality level above a certain value i_{th}. In our experiments, however, we found that naturally increasing customers' irrationality in a data-driven way during the GPT exploration procedure worked well in practice, confirming the practical utility of indifference sets for avoiding overfitting.

5. Computational results

In this section, we report the results of our experiments on both synthetic and real datasets. The goal is to understand whether irrational, partially-ranked behaviors can improve predictive accuracy on new offer sets. In all our experiments, we compare two variants of the partially-ranked choice model estimated using GPT, namely GPT-rat and GPT-irrat, obtained with and without irrational behaviors, respectively. We further compare the GPT-based approaches with two benchmarks: the enumerative rank-based choice model (RB) with fully-ranked lists obtained by enumerating all the $N!$ possible preferences, and the pairwise choice markov chain (PCMC) proposed by Ragain and Ugander (2016). Section 5.1 focuses on the generalization performances of the various approaches on a set of synthetic instances. In Section 5.2, we test the models on a set of publicly available datasets used in transportation for mode choice analysis.

5.1. Numerical results on synthetic instances

Data Generation. We generate choice data samples according to two ground-truth models, specifically the Halo-MNL model proposed by Maragheh et al. (2018) and the GSP model. Both of them allow us to control the amount of irrationality resulting in the generated instances and to investigate its impacts on the performance of the various approaches. For each ground truth model, instances were generated as follows:

- **Halo-MNL**: this choice model is parametrized by a pairwise interaction matrix U, whose diagonal terms u_{ii} represent the item-specific utilities. Given the offer set S, the overall probability of choosing product i is given by

$$P(i|S) = \frac{\exp(u_{ii} + \sum_{k \notin S} u_{ki})}{\sum_{j \in S} \exp(u_{jj} + \sum_{k \notin S} u_{kj})} \ .$$

It is easy to see that by setting to zero the off-diagonal terms of matrix U, we obtain an MNL model. Following Chen and Mišic (2019), we draw the elements $u_{ii} \sim \text{Unif}[-1, 1]$. We vary the irrationality of the instances by varying the number of pairwise interactions. Specifically, we generate instances where 0%, 10% and 25% of the couples present a positive interaction, obtained by setting the corresponding off-diagonal terms to -1. We simulate both symmetric halo effects, where two products increase each other's attractiveness, and asymmetric halo effects, also known as *decoy* effects, where only one of two products benefits from the presence of the other (the decoy) in the offer set. In order to investigate more complex interaction scenarios, we generalize the Halo-MNL model so as to include multiple customer segments, whose probability is drawn uniformly from the unit simplex. In our experiments, we have used either one or ten customer segments. We note again that when setting the number of pairwise interactions to zero, we end up obtaining rational instances generated under MNL and MMNL ground-truth models, depending on the number of customer segments, 1 and 10, respectively.

- **GSP**: We remind from Section 3 that a generalized stochastic preference is defined as $C(\sigma, i)$, where σ is a ranking over the N alternatives, and i is the irrationality level of the customer type. Instances generated under this ground-truth model contain either 10 or 100 customer types, whose probabilities are randomly drawn from the unit simplex. For each instance, we consider 10%, 20% or 50% of the customer types as irrational, meaning their index i is greater than one. Specifically, the irrationality level i of each of these customer types was randomly chosen in $\{1, 2, ..., i_{max}\}$.

We used $i_{max} = 1, 5,$ and 9 to simulate various levels of irrationality. Rational instances have been obtained by setting the percentage of irrational behaviors to zero.

In all the experiments reported in this section, we used a number of products $N = 10$, one of which represents the no-purchase option. For each ground-truth model, we generate either $3,000$ or $50,000$ transactions, for a total of 10, 20 or 50 training offer sets. This simulates different amount of training data. When using 50,000 transactions, in particular, the goal is to simulate the scenario in which we train the models based on empirical probabilities that are close to the true ones (i.e., those from the ground-truth model), and the effect of any sampling noise becomes negligible. This corresponds to the setting already used, for example, in Bertsimas and Mišic (2016) and Chen and Mišic (2019), where choice models are trained on ground truth probabilities. We further assume that the number of transactions is equally distributed among the training offer sets, which all have dimension $|S| \geq 3$ and contain the no-purchase option.

Estimation of the choice models. Following Jagabathula and Rusmevichientong (2019), we train all the rank-based approaches by minimizing the average Kullback-Leibler (KL) divergence (4) between predicted probability distributions and the empirical ones over training offer sets. As already observed, it is well known that minimizing the KL divergence is equivalent to maximum likelihood estimation in terms of optimal solution retrieved (see, e.g., Jagabathula and Rusmevichientong 2019). For the GPT-based approaches, training stops either when the training objective function reaches a value smaller then $\epsilon_{th} = 0.005$ or when no negative reduced cost column has been found at a given iteration. The PCMC choice model is trained by maximum likelihood estimation. For our experiments, we use the code provided by the authors (code available at `https://github.com/sragain/pcmc-nips`). We refer the reader to Appendix B for more details on the implementation of the PCMC choice model.

Loss of Rationality. We investigate the level of *irrationality* present in the instances we generated. In line with the methodology proposed by Jagabathula and Rusmevichientong (2019), we fit the enumerative rank-based choice model, RB, to all our training instances. It is well known, indeed, that any RUM choice model can be equivalently represented as a probability distribution over rankings of alternatives (Block and Marschak 1959). Thus, by fitting such model to a given instance, the resulting objective function indicates what the authors define as the *Loss of rationality* (LoR) of that instance, which can be interpreted as a measure of the minimum amount of choice data that cannot be explained by using any choice model belonging to the RUM family. Figure 2 reports the LoR value distributions over instances grouped by category (Rational and Irrational), ground-truth models (Halo-MNL and GSP) and number of customer types (in parenthesis). As already mentioned, rational instances for Halo-MNL(1) and Halo-MNL(10) correspond to instances

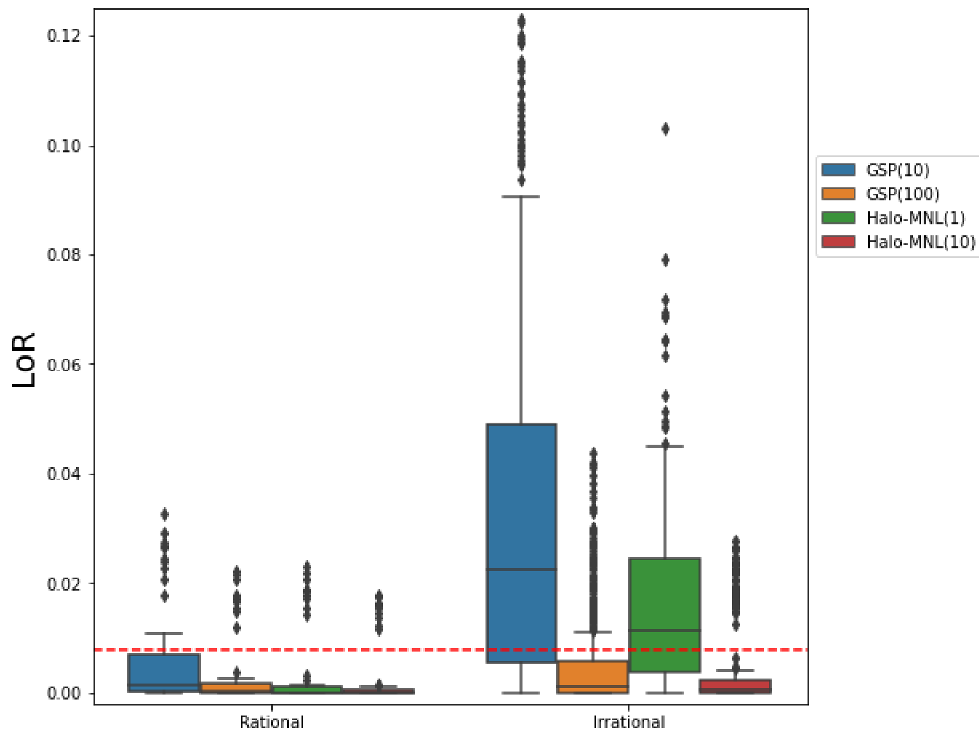


Figure 2 **Distributions of Loss of Rationality for generated instances, grouped by ground-truth models and number of customer behaviors.**

generated under MNL and MMNL ground-truth models, respectively. Also, rational GSP ground-truth models are equivalent to Rank-Based models with the same number of preference lists. It is interesting to note, in particular, that the aggregation of a high number of irrational customer types seems to result into a rational choice behavior at the population level (see Halo-MNL(10) and GSP(100) in Figure 2). We report in Figure 2 a red dashed line, corresponding to a Loss of rationality of 0.008, which visually separates the generated instances based on their irrationality level. Essentially, rationally-generated instances tend to fall below this threshold, but also the irrationally-generated ones in which many customer types are aggregated. In the following, we interpret this value as a threshold to understand whether an instance contains significant amount of irrational choice behaviors. In Appendix C, we further show that the LoR of a given instance can be impacted by other factors as well, such as the number of choice samples and offer sets available for training.

Generalization performances. We now focus on the generalization performance of the various approaches when tested on new offer sets. This has been measured in terms of average L_1 error between the predicted probability distribution $\boldsymbol{x}$ and the ground-truth probability distribution $\boldsymbol{v}$ on new offer sets, and has been computed as

$$L_1(\boldsymbol{x}, \boldsymbol{v}) = \frac{1}{|\mathcal{M}_{test}|} \sum_{S \in \mathcal{M}_{test}} \sum_{i \in S} \left| x_{i,S} - v_{i,S} \right|, \tag{5}$$

where $\mathcal{M}_{test}$ is the collection of *all* possible offer sets that have not been used for training. This means that, once we generate the 512 offer sets of dimensions $3 \leq |S| \leq 10$ (and containing the no-purchase option), $M = 10, 20$ or 50 of them are used for training and $|\mathcal{M}_{test}| = 512 - M$ are used for testing. As noted in Ragain and Ugander (2016), equation (5) can be interpreted as the expected L_1 prediction error given a randomly drawn offer set.

Irrational instances		LoR	RB	GPT			PCMC
	% irrat		rat	rat	irrat	% change	-
Halo-MNL(1)	10	0.0103	0.2185	0.2325	**0.1785**	-23.2	0.2610
	25	0.0244	0.3239	0.3285	**0.2838**	-13.6	0.3804
	avg (all)	0.0173	0.2712	0.2805	**0.2312**	-17.6	0.3207
Halo-MNL(10)	10	0.0035	0.1453	0.1233	**0.1208**	-2.0	0.2195
	25	0.0044	0.1691	0.1650	**0.1551**	-6.0	0.2518
	avg (all)	0.0040	0.1572	0.1442	**0.1379**	-4.3	0.2356
GSP(10)	10	0.0171	0.2402	0.2202	**0.2188**	-0.6	0.4273
	20	0.0286	0.2798	0.2638	**0.2502**	-5.2	0.4625
	50	0.0573	0.3785	0.3712	**0.3352**	-9.7	0.5312
	avg (all)	0.0343	0.2995	0.2851	**0.2681**	-6.0	0.4737
GSP(100)	10	0.0034	0.1683	**0.1483**	0.1539	3.8	0.2890
	20	0.0043	0.1820	**0.1657**	0.1673	1.0	0.3009
	50	0.0082	0.2208	0.2123	**0.2042**	-3.8	0.3394
	avg (all)	0.0053	0.1904	0.1754	**0.1751**	-0.2	0.3098
avg (all)		0.0170	0.2355	0.2247	**0.2102**	-6.5	0.3568
Rational instances							
MNL	-	0.0034	0.1207	**0.1011**	0.1072	6.0	0.1404
MMNL(10)	-	0.0027	0.1280	**0.0843**	0.0888	5.3	0.1529
RB(10)	-	0.0060	0.1560	**0.1449**	0.1583	9.2	0.3841
RB(100)	-	0.0033	0.1542	**0.1351**	0.1429	5.8	0.2793
avg(all)		0.0039	0.1397	**0.1163**	0.1243	6.8	0.2392

Table 2 Average L_1 test errors for each approach under various ground truth models. Each line averages over instances generated with different number of training offer sets (10,20 and 50) and transactions (3,000 and 50,000). For Halo-MNL, each group aggregates instances with both symmetric and asymmetric interactions. For GSP, each group aggregates instances with different irrationality levels.

Table 2 reports the L_1 test errors of each approach on sets of instances grouped by ground-truth models and number of customers types, indicated in parenthesis. The value of column "% irrat" further divides each group based on the characteristics of the ground-truth model generating the corresponding set of instances. For Halo-MNL instances, this column indicates the amount of pairwise interactions among products, while, for GSP instances, it indicates the percentage of irrational behaviors. For GPT-based approaches, we further report in "% change" the percentage change in performance obtained by considering irrational behaviors in the estimation procedure.

We focus first on the set of irrational instances. It is possible to observe how the performance of the rational choice models, RB and GPT-rat, quickly deteriorates as the Loss of Rationality increases. Such trend is significantly less pronounced for GPT-irrat, which can benefit from the generalization power of irrational behaviors. In particular, GPT-irrat seems to behave particularly well in capturing positive interactions on Halo-MNL instances with one customer type, with average test error improvements ranging between 17% and 23%. We remind that the ground-truth model in this set of instances assumes either asymmetric (decoy) or symmetric interactions among products. The latter, in particular, were found to result in slightly higher LoR, for an average value of 0.0229 compared to an average LoR of 0.0136 for asymmetric interactions. The difference in performance between the two methods is also significant on GSP instances with 10 customer types with high numbers of irrational behaviors: up to 9.7% for instances with 50% of irrational behaviors. However, for smaller percentages of irrational behaviors the performance gain of GPT-irrat is less strong. This is probably due to the more complex interactions among alternatives resulting under a GSP ground-truth model. Further analysis has shown that when given enough training data, both in terms of number of training choice samples and number of training offer sets, the performance improvement of GPT-irrat can be as high as 27% on the same set of instances (see Table 10 in Appendix C). As previously observed, Halo-MNL and GSP instances with many irrational behaviors seem to result in more rational interactions at the population level, thus leading to smaller differences in performance between the GPT-rat and GPT-irrat approaches (not necessarily in favor of GPT-irrat). We finally note that GPT-based approaches clearly outperform the two benchmarks, RB and PCMC. The latter, in particular, struggles to generalize well on the GSP(10) set of instances, confirming that the choice behaviors resulting in this case are particularly hard to capture.

We now move our attention to the set of rational instances, where, as one may expect, the flexibility of GPT-irrat may increase the risk of overfitting, thus leading to worse generalization when compared to GPT-rat. This also favorably compares with RB-rat, confirming the results of Jena et al. (2020) on the generalization power of partially-ranked lists. Further, as already mentioned in Jena et al. (2020) and van Ryzin and Vulcano (2015), using all $N!$ fully-ranked preferences for training the RB model increases the risk of overfitting the training set. This supports the hypothesis that adding only relevant types to the estimated choice model is crucial for its generalization performance. Finally, we observe that the PCMC model is clearly outperformed by rank-based methods on this set of instances as well.

Impact of the irrationality level of GSP customer types. To explore the impact of the limited irrationality assumption intrinsic to the GPT algorithm, in Table 3 we further analyze the performance of GPT-based approaches, with and without irrational behaviors respectively, on the set of instances GSP(10). In particular, for each "% irrat" value reported in Table 2, we disaggregate the performance of GPT-rat and GPT-irrat on instances grouped by maximum irrationality level i_{max} of the customer types used to generate the ground-truth models. While particularly high levels of

			LoR	RB	GPT		
	% irrat	i_{max}		rat	rat	irrat	% change
GSP(10)	10	1	0.0189	0.2518	0.2265	**0.2121**	-6.3
		5	0.0186	0.2465	**0.2335**	0.2356	0.9
		9	0.0137	0.2222	**0.2006**	0.2087	4.0
		avg (all)	0.0171	0.2402	0.2202	**0.2188**	-0.6
GSP(10)	20	1	0.0288	0.2885	0.2696	**0.2394**	-11.2
		5	0.0366	0.3177	0.3024	**0.2853**	-5.7
		9	0.0204	0.2333	**0.2196**	0.2259	2.9
		avg (all)	0.0286	0.2798	0.2638	**0.2502**	-5.2
GSP(10)	50	1	0.0519	0.3892	0.3866	**0.3257**	-15.8
		5	0.0762	0.4623	0.4525	**0.4177**	-7.7
		9	0.0440	0.2840	0.2746	**0.2623**	-4.5
		avg (all)	0.0573	0.3785	0.3712	**0.3352**	-9.7
	avg (all)		0.0343	0.2995	0.2851	**0.2681**	-6.0

Table 3 Test errors comparison between RB-rat, GPT-rat and GPT-irrat on GSP instances with 10 customer types, grouped based on the percentage of irrational behaviors in the ground truth model, and their irrationality levels. The metric reported is the average L1 error per offer set

irrationality may not be very common in practice, they allow us to analyze possible limitations of our approach. In column "% change", we thus report the percentage change in performance obtained by GPT-irrat when compared to GPT-rat. It can be noticed that when capturing the behavior of customers with limited levels of irrationality, the average performance gain of GPT-irrat can be as high as 15.8%. It is also interesting to note a decrease in the Loss of Rationality of instances generated for $i_{max} = 9$ and, coherently, an improvement in the predictive accuracy of rational rank-based methods on the same set of instances. Indeed, given an offer set S and a generalized stochastic preference $C(\sigma, 9)$, it is often the case that $|\sigma_S| < i$. We recall from Section 3 that, in such cases, the considered customer type leaves with no purchase. Intuitively, such a behavior can be more easily approximated by a rational choice model imposing a high probability mass on the no-purchase option. This also translates into predictions that are more accurate on average than those obtained for smaller levels of irrationality i_{max}.

We conclude this section by referring the interested reader to Appendix C, where we present an analysis of the impact of data availability on the performance of the various approaches. In summary, we noticed that higher amounts of training data, both in terms of number of training offer sets M and number of samples T available for training, can further increase the performance gap between GPT-rat and GPT-irrat on irrational instances. As one may expect, large numbers of training offer sets also benefit the enumerative approach (RB), since it decreases the risk of overfitting. Finally, we observed that the PCMC approach becomes closer to GPT-based approaches in terms of generalization error when 50 offer sets and 3,000 choice samples are available at training time, but it is always outperformed on average by either GPT-rat or GPT-irrat on rational and irrational instances, respectively.

5.2. Numerical results on mode choice datasets

In this section, we test all approaches on real-world instances, used in transportation for mode choice analysis, and all publicly available. Namely,

- `swissmetro` (Bierlaire et al. 2001): 10,758 choices made by people among car (when available), train and maglev (a type of train exploiting magnetic repulsion in order to alleviate the friction of traditional transportation systems), for traveling among major urban centers in Switzerland. Both train and maglev can operate at three different time intervals, specifically every 30, 60 or 120 minutes the former, and every 10, 20 or 30 minutes the latter. Following Osogami and Otsuka (2014), we consider each case as a separate option, resulting in a total number of 7 alternatives.
- `sfwork` and `sfshop` (Koppelman and Bhat 2006): 3,157 and 5,029 observations, respectively, on the transportation modes chosen by people to commute and to travel to a shopping center in the San Francisco Bay Area. We note that these sets of instances are the same used in the work of Ragain and Ugander (2016).

Basic statistics describing the three sets of instances are summarized in Table 4.

	sfshop	swissmetro	sfwork
nb samples	3157	10758	5029
nb offer sets	10	18	12
nb alternatives	6	7	8
size of offer sets	{4,5,6}	{2,3}	{6,7,8}

Table 4 **Statistics describing the mode choice datasets.**

Since none of these datasets contains the no-purchase option, we slightly modify the behavior of partially-ranked lists with irrationality so that, when faced with an offer set S, customer $C(\sigma,i)$ chooses uniformly at random among the available alternatives whenever $i \geq |P_S(\sigma) \cup I_S(\sigma)|$. In line with other works (see, e.g., Jagabathula et al. 2020, Chen and Mišic 2019, Osogami and Otsuka

2014), in order to assess the predictive accuracy of the various approaches on this set of real instances, we measure the average KL divergence (4) between predicted and empirical probability distributions over test offer sets. Given the limited amount of available data, we decided to report leave-one-out crossvalidated results. Specifically, given M offer sets, we use $M-1$ of them for training and the last one to test the estimated choice model. The average of the M different evaluation metrics obtained in this way is reported in Table 5. In line with our results on synthetic instances (see Section 5.1), we notice that GPT-irrat is able to improve the predictive accuracy on the `sfshop` set of instances that, according to the corresponding LoR, contains a significant amount of irrational choice behaviors. Also, GPT-irrat achieves better generalization on average on the `swissmetro` dataset, which, despite characterized by low loss of rationality, has been shown to contain halo effects (Osogami and Otsuka 2014). This gives some insights on the limitations of using the LoR metric for model selection.

	LoR	RB	GPT		PCMC
		rat	rat	irrat	-
sfshop	0.0124	0.4510	0.4093	0.3561	**0.2745**
sfwork	0.0066	0.0450	**0.0527**	0.0533	0.0612
swissmetro	0.0023	0.0571	0.0590	0.0474	**0.0326**

Table 5 Test error comparison on real instances for mode choice analysis. The metric reported is the average Kullback-Leibler divergence, obtained using leave-one-out cross-validation for each set of instances.

It is important to notice the particularly good performance of PCMC. Contrary to our generated instances, the set of instances used in these experiments contains only a relatively small number of interactions among alternatives to capture. In the `swissmetro` dataset, for example, the options corresponding to the different time intervals for the train are mutually exclusive. Hence, only one of those options can be present in a given offer set. The same is true for the three alternatives relative to the maglev mode of transportation. Using 17 offer sets for training, the PCMC choice model is able to learn good estimates of the pairwise transition matrix parametrizing the model. A similar situation characterizes also the `sfshop` and `sfwork` datasets, where 2 out of 6 and 4 out of 8 alternatives, respectively, are always present in the choice set. Thus, no transition rate needs to be estimated among those options. This hypothesis is further confirmed by the good performance of RB, especially when compared to GPT-rat on the set of more rational instances, i.e., on `sfwork` and `swissmetro`. The dimension of the dataset and the number of samples allow RB to generalize well on the test set, without overfitting. Even if given enough data, however, RB cannot learn the irrational interactions that allow GPT-irrat to better generalize on new offer sets for `sfshop` and `swissmetro` instances. In Appendix D, we report a set of experiments where we reproduce this scenario, by generating instances where 6 out of 10 products are always present in the offer set. Confirming our previous hypothesis, our results show that PCMC benefits from the limited amount of interactions (both in terms of substitutions and halo effects) resulting from this set of instances, obtaining a predictive accuracy competitive with rank-based methods.

6. Conclusion

In this paper, we proposed an estimation method for the *Generalized Stochastic Preference* choice model introduced by Berbeglia (2018). In order to do so, we show how to adapt the partially-ranked representation of rank-based preferences proposed by Jena et al. (2020) to the case of irrational behaviors. This allows us to adapt their column generation approach to efficiently estimate the choice model from choice data. We ran an extensive set of experiments in order to understand whether irrational behaviors can help improving generalization to unseen offer sets. In line with previous works on general choice models overcoming the RUM limitations (see, e.g., Chen and Mišic 2019, Chen et al. 2019), we observed that deciding whether to add irrational behaviors highly depends on the specific dataset at hand, with a significant impact on the predictive accuracy of the estimated choice model. In our experiments, given the limited amount of products, we were able to follow the method of Jagabathula and Rusmevichientong (2019) as a possible model-selection criterion. Although the authors give conditions in which this method can be applied efficiently, it is not tractable in general. Moreover, as we show in Section 5.2 and in Appendix C, the loss of rationality metric can be influenced by sampling noise and number of training offer sets. Hence, it is difficult to assess when the loss of rationality is high enough to justify going beyond the RUM framework. As a consequence, particular attention must be paid when applying these models to a given instance, since they may lead to worse predictive accuracy than RUM methods whenever the presence of irrational behaviors is negligible (see, e.g., Chen et al. 2019, Chen and Mišic 2019). One may thus resort to model selection techniques more in line with the machine learning literature, such as cross-validation, to understand which approach to choose in those cases. An appealing feature of our approach consists in the fact that choosing whether to go beyond the RUM framework or not resolves to setting a single, binary hyperparameter. This greatly simplifies the model selection process, making it easier for practitioners to understand whether going beyond the RUM framework is actually needed. Indeed, within the same framework, it is possible to exploit the explanatory power of both partially-ranked preference lists which, since subsuming fully-ranked preferences (see, e.g., Farias et al. 2013), can theoretically represent any RUM choice model, and irrational behaviors, which, as we have shown, can significantly enhance accuracy on irrational instances. Our approach is thus capable of providing accurate estimates of product demands on both rational instances (see also Jena et al. 2020) and irrational ones. This is not necessarily true for other general choice models that have been proposed in the literature, whose regularization methods have no clear connection to the irrationality level of a given instance, and to the expressive power of the obtained choice model.

References

Aouad, A., V. Farias, R. Levi. 2015. Assortment optimization under consider-then-choose choice models. *Working paper, Massachusetts Institute of Technology* (Available at SSRN 2618823).

Ariely, D. 2008. *Predictably irrational*. HarperCollins, New York, NY, US.

Arrow, K. J. 1951. *Social Choice and Individual Values*. John Wiley & Sons, New York, NY, US.

Berbeglia, G. 2018. The generalized stochastic preference choice model. *arXiv preprint arXiv:1803.04244* .

Berbeglia, G., A. Garassino, G. Vulcano. 2018. A comparative empirical study of discrete choice models in retail operations. *Available at SSRN 3136816* .

Bertsimas, D., V. Mišic. 2016. Data-driven assortment optimization. *Tech. report, Massachusetts Institute of Technology* .

Bierlaire, M., K. Axhausen, G. Abay. 2001. The acceptance of modal innovation: The case of swissmetro. *Proceedings of the 1st Swiss Transportation Research Conference*. Ascona, Switzerland.

Blanchet, J., G. Gallego, V. Goyal. 2016. A Markov Chain Approximation to Choice Modeling. *Operations Research* **64**(4) 886–905.

Block, H.D., J. Marschak. 1959. Random orderings and stochastic theories of response. Tech. Rep. 66, Cowles Foundation for Research in Economics, Yale University.

Busemeyer, J R., J T. Townsend. 1993. Decision field theory: A dynamic-cognitive approach to decision making in an uncertain environment. *Psychological Review* **100**(3) 432–459.

Chen, N., G. Gallego, Z. Tang. 2019. The use of binary choice forests to model and estimate discrete choice models. *Available at SSRN 3430886* .

Chen, Y., V. Mišic. 2019. Decision forest: A nonparametric approach to modeling irrational choice. *arXiv preprint arXiv:1904.11532* .

Echenique, F., K. Saito. 2019. General luce model. *Economic Theory* **68**(4) 811–826.

Echenique, F., K. Saito, G. Tserenjigmid. 2018. The perception-adjusted luce model. *Mathematical Social Sciences* **93** 67–76.

Farias, V. F., S. Jagabathula, D. Shah. 2013. A nonparametric approach to modeling choice with limited data. *Management Science* **59**(2) 305–322.

Feng, G., X. Li, Z. Wang. 2018. On substitutability and complementarity in discrete choice models. *Operations Research Letters* **46**(1) 141 – 146.

Gallego, G., R. Ratliff, S. Shebalov. 2014. A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research* **63**(1) 212–232.

Huber, J., J. W. Payne, C. Puto. 1982. Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. *Journal of consumer research* **9**(1) 90–98.

Iyengar, S. S., M. R. Lepper. 2000. When choice is demotivating: can one desire too much of a good thing? *Journal of personality and social psychology* **79**(6) 995–1006.

Jagabathula, S., P. Rusmevichientong. 2019. The limit of rationality in choice modeling: Formulation, computation, and implications. *Management Science* **65**(5) 2196–2215.

Jagabathula, S., L. Subramanian, A. Venkataraman. 2020. A conditional gradient approach for nonparametric estimation of mixing distributions. *Management Science* .

Jagabathula, S., G. Vulcano. 2018. A partial-order-based model to estimate individual preferences using panel data. *Management Science* **64**(4) 1609–1628.

Jena, S. D., A. Lodi, H. Palmer, C. Sole. 2020. A partially-ranked choice model for large-scale data-driven assortment optimization. *INFORMS Journal on Optimization* (Forthcoming).

Kleinberg, J., S. Mullainathan, J. Ugander. 2017. Comparison-based choices. *Proceedings of the 2017 ACM Conference on Economics and Computation*. 127–144.

Koppelman, F. S., C. Bhat. 2006. A self instructing course in mode choice modeling: multinomial and nested logit models. *US Department of Transportation, Federal Transit Administration* .

Luce, R. D. 1959. *Individual choice behavior*. John Wiley & Sons, Oxford, England.

Maragheh, R. Y., A. Chronopoulou, J. M. Davis. 2018. A customer choice model with halo effect. *arXiv preprint arXiv:1805.01603* .

Mottini, A., R. Acuna-Agost. 2017. Deep choice model using pointer networks for airline itinerary prediction. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1575–1583.

Nocedal, J., S. Wright. 2006. *Numerical optimization*. Springer Series in Operations Research, Springer, New York, NY, US.

Osogami, T., M. Otsuka. 2014. Restricted boltzmann machines modeling human choice. *Advances in Neural Information Processing Systems*. 73–81.

Pfannschmidt, K., P. Gupta, E. H"ullermeier. 2019. Learning choice functions. *arXiv preprint arXiv:1901.10860* .

Ragain, S., J. Ugander. 2016. Pairwise choice markov chains. *Advances in Neural Information Processing Systems*. 3198–3206.

Rieskamp, J., J. R. Busemeyer, B. A. Mellers. 2006. Extending the Bounds of Rationality: Evidence and Theories of Preferential Choice. *Journal of Economic Literature* **44**(3) 631–661.

Roe, R. M., J. R. Busemeyer, J. T. Townsend. 2001. Multialternative decision field theory: a dynamic connectionist model of decision making. *Psychological review* **108**(2) 370–392.

Rooderkerk, R. P., H. J. Van Heerde, T. H.A Bijmolt. 2011. Incorporating Context Effects Into a Choice Model. *Journal of Marketing Research* **48**(4) 767–780.

Rosenfeld, N., K. Oshiba, Y. Singer. 2020. Predicting choice with set-dependent aggregation. *International Conference on Machine Learning*. 2635–2644.

Schwartz, Barry. 2004. *The paradox of choice: Why more is less*. HarperCollins, New York, NY, US.

Seshadri, A., A. Peysakhovich, J. Ugander. 2019. Discovering context effects from raw choice data. *International Conference on Machine Learning*. 5660–5669.

Simonson, I. 1989. Choice based on reasons: The case of attraction and compromise effects. *Journal of consumer research* **16**(2) 158–174.

Strauss, A. K, R. Klein, C. Steinhardt. 2018. A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research* **271**(2) 375–387.

Talluri, K. T., G. J. Van Ryzin. 2004. *The Theory and Practice of Revenue Management*. Springer, Boston, MA, US.

Thurstone, L. L. 1927. A law of comparative judgment. *Psychological review* **34**(4) 273.

Tversky, A., I. Simonson. 1993. Context-Dependent Preferences. *Management Science* **39**(10) 1179–1189.

Usher, M., J. L. McClelland. 2004. Loss aversion and inhibition in dynamical models of multialternative choice. *Psychological review* **111**(3) 757.

van Ryzin, G., G. Vulcano. 2015. A Market Discovery Algorithm to Estimate a General Class of Nonparametric Choice Models. *Management Science* **61**(2) 281–300.

Yellott, J. I. 1977. The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology* **15**(2) 109 – 144.

Appendix

A. Regularity violation of the no-purchase option

The Generalized Stochastic Preference choice model as defined by Berbeglia (2018) does not account for violations of the regularity assumption for the no-purchase option (see Berbeglia 2018, Lemma 1). Given two offer sets $S \subseteq S' \subseteq \mathcal{N}$, in particular, the authors show that every customer choosing the no-purchase option from S', by definition, must choose the no-purchase option from S as well. However, we can circumvent such limitation by allowing a customer type $C(\sigma, i)$ to rank the no-purchase option in σ. Consider, for example, two offer sets $S = \{0, 1, 2\}$ and $S' = \{0, 1, 2, 3\}$, where $S \subset S'$, and a customer type $C\big((3\ 0\ 1\ 2), 1\big)$. Her choice behavior is reported in Table 6. It is easy to see that, by introducing the option 3 in the offer set, we can increase the probability of option 0 being chosen and, thus, of the customer leaving without any purchase.

S	σ_S	Choice
$\{0, 1, 2\}$	$(0\ 1\ 2)$	1
$\{0, 1, 2, 3\}$	$(3\ 0\ 1\ 2)$	0

Table 6 **Choice behavior of customer $C\big((3\ 0\ 1\ 2), 1\big)$ faced with two different offer sets.**

B. Details on the implementation of PCMC

In this section, we elaborate on the implementation details of the PCMC choice model. The model is trained by Maximum Likelihood Estimation, and a Sequential Least SQuares Programming (SLSQP) solver (Nocedal and Wright 2006) is used to optimize the corresponding objective function, which is concave in general. The authors suggest to use additive-smoothing to avoid some numerical issues involved in the training of the model. In particular, given an offer set of size $|S|$ and an additive smoothing parameter α, the probability of choosing alternative j is computed at training time as

$$P(j|S) = \frac{T_{jS} + \alpha}{T_S + \alpha|S|},$$

where T_S is the number of training samples showing offer set S, and T_{jS} the number of times alternative j is chosen from the offer set S. In Table 7, we investigate the change in performance due to different stopping criteria and values of parameter α. In particular, we implemented stopping criteria based on

1. The maximum number of iterations to be performed by the solver: this is set to 25, which is the default value in the code provided by the authors. The corresponding results are reported in column "PCMC-25"
2. The absolute change in the objective function between two consecutive iterations: the algorithm is stopped when this change is smaller than 10^{-6}, and the corresponding results are reported in column "PCMC-∞".

Moreover, for each stopping criterion, we compared the performance obtained by using different amounts of additive-smoothing in the training set. Specifically, column "Crossval" reports the average generalization error obtained using 5-folds crossvalidation to select the best $\alpha \in \{0, 0.01, 0.1, 1, 5, 10\}$. Column "None" corresponds to $\alpha = 0$, for which no additive smoothing was used.

Irrational instances		PCMC -25		PCMC - ∞	
	% irrat	Crossval	None	Crossval	None
Halo-MNL(1)	10	**0.2595**	0.2610	0.2897	0.2941
	25	**0.3756**	0.3804	0.3933	0.3937
	avg (all)	**0.3175**	0.3207	0.3415	0.3439
Halo-MNL(10)	10	**0.1997**	0.2195	0.2348	0.2545
	25	**0.2430**	0.2518	0.2873	0.3010
	avg (all)	**0.2214**	0.2356	0.2610	0.2778
GSP(10)	10	0.4293	**0.4273**	0.4615	0.4625
	20	**0.4622**	0.4625	0.4927	0.4918
	50	**0.5283**	0.5312	0.5668	0.5711
	avg (all)	**0.4733**	0.4737	0.5070	0.5084
GSP(100)	10	**0.2790**	0.2890	0.3223	0.3386
	20	**0.2880**	0.3009	0.3354	0.3478
	50	**0.3283**	0.3394	0.3868	0.3976
	avg (all)	**0.2984**	0.3098	0.3481	0.3613
avg (all)		**0.3500**	0.3568	0.3887	0.3967
Rational instances					
MNL	-	**0.1351**	0.1404	0.1930	0.2012
MMNL	-	**0.1381**	0.1529	0.1970	0.2136
RB(10)	-	**0.3840**	0.3841	0.4028	0.4065
RB(100)	-	**0.2741**	0.2793	0.3167	0.3292
avg (all)		**0.2328**	0.2392	0.2774	0.2876

Table 7 Average L_1 test errors for different PCMC implementations under various ground truth models. Each line averages over instances generated with different number of training offer sets (10,20 and 50) and transactions (3,000 and 50,000).

The average L_1 test error has been reported over instances grouped by ground-truth models and number of customers types, indicated in parenthesis in the first column. The value of column "% irrat" further divides each group based on the characteristics of the ground-truth model generating the corresponding set of instances. For Halo-MNL instances, this column indicates the amount of pairwise interactions among products, while, for GSP instances, it indicates the percentage of irrational behaviors. We observe that limiting the number of iterations seems to be having a major impact on the predictive accuracy of the resulting choice model. Our intuition is that allowing the solver to proceed until convergence is reached may end up in overfitting the training set. We also notice that the gain in performance obtained by using additive-smoothing is not significant in general. To confirm whether the deterioration in performance of PCMC-∞ is actually due to overfitting, in Table 8 we further compare the two variants on a set of instances where $50,000$ training samples have been generated, and we investigate the impact of the number of training offer sets on the resulting choice model. Confirming our previous hypothesis, we notice that when significant amount of training data is available, both in terms of number of training samples and number of training offer sets M, the risk of overfitting decreases and a better fit at training time translates in a significant

Irrational		PCMC - 25		PCMC - ∞	
	M	Crossval	None	Crossval	None
Halo-MNL	10	**0.3091**	0.3097	0.3341	0.3332
	20	0.2824	**0.2801**	0.2827	0.2826
	50	0.2340	0.2329	**0.1546**	0.1570
GSP	10	**0.4859**	0.4873	0.5347	0.5340
	20	**0.4052**	0.4069	0.4300	0.4322
	50	0.3355	0.3383	**0.2817**	0.2827
avg (all)		0.3677	0.3688	**0.3667**	0.3675
Rational					
(M)MNL	10	**0.1212**	0.1216	0.1518	0.1497
	20	**0.0921**	0.0924	0.1231	0.1294
	50	0.0733	0.0704	**0.0659**	0.0662
RB	10	**0.4187**	0.4291	0.4897	0.4997
	20	0.3530	**0.3429**	0.3794	0.3893
	50	0.3072	0.2961	0.2099	**0.2086**
all (avg)		**0.3490**	0.3497	0.3494	0.3506

Table 8 Average L_1 test errors for different PCMC implementations under various ground truth models, on instances with 50,000 choice samples available for training. Instances are further divided based on the number of offer sets M observed during training.

improvement in generalization error. Nevertheless, also for this set of instances, i.e., with $M = 50$ offer sets and $50,000$ samples are available at training time, the performance of the best PCMC variant is worse than the one of the GPT-based approaches (see Table 10). At this point, one may wonder whether more adaptive stopping criteria may be used instead of fixing ahead the maximum number of iterations. However, further experiments revealed that fixing the number of iterations to 25 worked better on average than other stopping criteria based on

- The relative change in the objective function between consecutive iterations ($< 1\%$),
- The maximum absolute change in the predicted probabilities over all training offer sets (< 0.001),
- The maximum absolute change in the value of the parameters of the PCMC choice model,
- The maximum number of iterations set to 100.

We thus avoid reporting the set of results corresponding to such stopping criteria, and use the PCMC-25 variant with no additive smoothing in the rest of our experiments. In particular, this is also the PCMC implementation used for the experiments reported in Section 5.1 and Section 5.2.

C. Impact of the amount of available data

In the following, we analyze how the amount of available training data impacts the predictive accuracy of the trained choice models. We consider the set of instances generated as outlined in Section 5, and group them based on the number of samples available for training. We then report in Table 9 and Table 10 results for instances consisting of $3,000$ and $50,000$ training samples, respectively. Each set of instances is further

Irrational instances		LoR	RB	GPT			PCMC
			rat	rat	irrat	% change	
Halo-MNL(1)	10	0.0073	0.3167	0.3157	**0.2729**	-13.6	0.3971
	20	0.0161	0.2722	0.2805	**0.2580**	-8.0	0.3203
	50	0.0440	0.2471	0.2581	**0.2314**	-10.3	0.2478
	avg (all)	0.0225	0.2787	0.2848	**0.2541**	-10.8	0.3217
Halo-MNL(10)	10	0.0003	0.2153	0.1630	**0.1486**	-8.8	0.2860
	20	0.0023	0.1730	0.1719	**0.1592**	-7.4	0.2477
	50	0.0200	**0.1499**	0.1609	0.2013	25.1	0.1938
	avg (all)	0.0075	0.1794	**0.1652**	0.1697	2.7	0.2425
GSP(10)	10	0.0174	0.3774	**0.3434**	0.3463	0.8	0.5346
	20	0.0371	0.2950	0.2859	**0.2772**	-3.0	0.4432
	50	0.0655	0.2411	0.2381	**0.1953**	-18.0	0.3522
	avg (all)	0.0400	0.3045	0.2891	**0.2729**	-5.6	0.4433
GSP(100)	10	0.0011	0.2441	0.2038	**0.1955**	-4.0	0.3788
	20	0.0035	0.2006	0.1937	**0.1887**	-2.6	0.2984
	50	0.0222	**0.1713**	0.1763	0.2113	19.9	0.2283
	avg (all)	0.0089	0.2054	**0.1912**	0.1985	3.8	0.3018
avg (all)		0.0216	0.2470	0.2355	**0.2284**	-3.0	0.3447
Rational instances							
MNL	10	0.0003	0.1760	0.1178	**0.1176**	-0.2	0.2541
	20	0.0013	0.1446	0.1387	**0.1341**	-3.3	0.1964
	50	0.0186	**0.1223**	0.1340	0.1804	34.7	0.1407
	avg (all)	0.0067	0.1476	**0.1302**	0.1440	10.7	0.1971
MMNL	10	0.0000	0.1850	**0.0994**	0.1061	6.6	0.2571
	20	0.0008	0.1505	0.1403	**0.1291**	-8.0	0.2034
	50	0.0152	**0.1288**	0.1446	0.1841	27.3	0.1390
	avg (all)	0.0053	0.1548	**0.1281**	0.1398	9.1	0.1998
RB(10)	10	0.0024	0.2695	**0.2544**	0.2777	9.1	0.4503
	20	0.0067	**0.1432**	0.1484	0.1682	13.3	0.3516
	50	0.0252	**0.0942**	0.0958	0.1079	12.7	0.2680
	avg (all)	0.0114	0.1689	**0.1662**	0.1846	11.1	0.3566
RB(100)	10	0.0003	0.2162	0.1709	**0.1640**	-4.1	0.3112
	20	0.0017	0.1711	**0.1562**	0.1570	0.5	0.2482
	50	0.0175	**0.1442**	0.1507	0.1988	31.9	0.2146
	avg (all)	0.0065	0.1772	**0.1593**	0.1733	8.8	0.2580
avg (all)		0.0075	0.1621	**0.1460**	0.1604	9.9	0.2529

Table 9 Test error on generated instances with a total number 3,000 training samples. Instances are grouped by ground-truth model, and number of offer sets available for training. For each set of instances, the average L_1 test errors are reported for the various approaches.

Irrational instances		LoR	RB	GPT			PCMC
			rat	rat	irrat	% change	
Halo-MNL(1)	10	0.0052	0.3107	0.3016	**0.2621**	-13.1	0.3590
	20	0.0110	0.2558	0.2688	**0.2056**	-23.5	0.3225
	50	0.0203	0.2245	0.2373	**0.1627**	-31.4	0.2775
	avg (all)	0.0122	0.2637	0.2692	**0.2101**	-22.0	0.3197
Halo-MNL(10)	10	0.0001	0.1950	0.1390	**0.1261**	-9.3	0.2605
	20	0.0002	0.1212	0.1201	**0.0982**	-18.2	0.2377
	50	0.0009	0.0888	0.0960	**0.0781**	-18.7	0.1883
	avg (all)	0.0004	0.1350	0.1184	**0.1008**	-14.8	0.2288
GSP(10)	10	0.0153	0.3924	**0.3414**	0.3515	3.0	0.5931
	20	0.0280	0.2795	0.2708	**0.2568**	-5.2	0.5008
	50	0.0427	0.2115	0.2158	**0.1559**	-27.8	0.4181
	avg (all)	0.0287	0.2945	0.2760	**0.2547**	-7.7	0.5040
GSP(100)	10	0.0004	0.2341	0.1769	**0.1697**	-4.0	0.3815
	20	0.0011	0.1659	0.1548	**0.1472**	-4.9	0.3131
	50	0.0035	0.1262	0.1265	**0.1184**	-6.4	0.2585
	avg (all)	0.0017	0.1754	0.1527	**0.1451**	-5.0	0.3177
avg (all)		0.0124	0.2240	0.2080	**0.1862**	-10.5	0.3688
Rational instances							
MNL	10	0.0000	0.1379	0.0912	**0.0889**	-2.5	0.1054
	20	0.0000	0.0801	**0.0723**	0.0731	1.1	0.0800
	50	0.0002	0.0631	0.0613	**0.0598**	-2.4	0.0658
	avg (all)	0.0001	0.0937	0.0749	**0.0740**	-1.3	0.0837
MMNL	10	0.0000	0.1540	**0.0488**	**0.0488**	0.0	0.1378
	20	0.0000	0.0836	**0.0469**	**0.0469**	0.0	0.1048
	50	0.0001	0.0662	**0.0457**	**0.0457**	0.0	0.0751
	avg (all)	0.0000	0.1013	**0.0471**	**0.0471**	0.0	0.1059
RB(10)	10	0.0001	0.2769	**0.2084**	0.2274	9.1	0.4858
	20	0.0004	**0.1114**	0.1151	0.1330	15.5	0.3979
	50	0.0014	**0.0411**	0.0583	0.0687	17.8	0.3508
	avg (all)	0.0006	0.1431	**0.1273**	0.1430	12.4	0.4115
RB(100)	10	0.0000	0.2037	0.1372	**0.1354**	-1.3	0.3724
	20	0.0000	0.1130	**0.1090**	0.1165	6.9	0.2879
	50	0.0004	**0.0773**	0.0831	0.0920	10.7	0.2413
	avg (all)	0.0001	0.1313	**0.1098**	0.1146	4.4	0.3005
avg (all)		0.0002	0.1174	**0.0898**	0.0947	5.5	0.2254

Table 10 **Test errors on generated instances with a total number of 50,000 training samples. Instances are grouped by ground-truth model, and number of offer sets available for training. For each set of instances, the average L_1 test errors are reported for the various approaches.**

divided based on the ground-truth model used to generate data, with the number of customer types reported in parenthesis, and on the number of offer sets M for which training data is available. In line with the results of Table 2, we observe that GPT-irrat well captures positive pairwise interactions. Specifically, Table 9 shows that irrational behaviors allow to improve predictive accuracy by 8% to 13% for Halo-MNL instances with one customer type and $3,000$ transactions. As one may expect, more complex choice behaviors resulting from GSP(10) instances generally need more training offer sets to be accurately learned from data. With 50 training offer sets, however, GPT-irrat is able to improve predictive accuracy by 18% with respect to GPT with only rational behaviors. Interestingly, PCMC seems to be competitive with GPT-based approaches when $M = 50$ offer sets are observed for training, despite being always outperformed by either GPT-rat or GPT-irrat on rational and irrational instances, respectively. As one may expect, a large number of training offer sets seems to benefit also the fully enumerative, rank-based approach, which otherwise ends up overfitting when M is small.

Results in Table 10 show that, for large numbers of training choice samples, the gap in performance between GPT-rat and GPT-irrat is even more pronounced when dealing with significant presence of irrational behaviors, with GPT-irrat improving predictive accuracy up to 31% and 27% for Halo-MNL(1) and GSP(10) instances, respectively. Interestingly, large numbers of training samples also improve the performance of GPT-irrat on rational instances, with a predictive accuracy deterioration of 5.5% on average, when compared to GPT-rat, against the 9.9% deterioration obtained with $3,000$ training samples. We also notice that in this set of instances, PCMC is not competitive with rank-based approaches, resulting in significantly higher generalization errors. This is also due to the issues related to the training of the PCMC model, on which we elaborated in Appendix B.

We conclude this section by mentioning some key aspects regarding the use of *LoR* as a metric to assess the irrationality level of a given instance. In particular, observing Table 9 and Table 10, we can notice the following:

- LoR tends to increase with the number of training offer sets M, and
- LoR is affected by the sampling noise due to limited number of available data. High levels of LoR may therefore be due to insufficient amount of training data more than to the irrationality level of a given instance.

These two observations shed light on the limitation of the LoR metric as a tool for model selection, and emphasize the importance of more robust model selection techniques based on crossvalidation, in line with the machine learning literature.

D. Experiments on structured instances

In this section, we focus on a set of experiments that aims at investigating the performances of GPT-rat, GPT-irrat, RB-rat and PCMC on a set of instances where only a small number of interactions among alternatives needs to be captured. In order to do so, we generate instances as delineated in Section 5.1, with the only difference that 6 randomly chosen alternatives out of 10 are always present, leading to a total number of 16 offer sets that can be generated for each instance. As a consequence, we also notice that the average dimension of an offer set for these generated instances is of 7.96 alternatives, against an average size of 5.5 alternatives for the set of experiments reported in Table 2. In Table 11, we report the average L_1 test errors of the various approaches on different ground-truth models, with $M = 10$ and $M = 15$ training offer sets. Coherently with our findings in Section 5 on real-world transportation mode data, the PCMC performance is competitive with the rank-based approaches on this set of instances, achieving the best results for Halo-MNL and MNL instances. Also, the difference in performance between RB-rat and GPT-rat is much smaller than the one observed in Table 2. In fact, given the limited amount of interactions that needs to be captured, the amount of training data is enough to avoid the overfitting problems characterizing the results on generic instances. One may also notice that, for rational instances generated using MNL and MMNL as ground-truth models, GPT-rat and GPT-irrat obtain the same test errors on average, which indicates the algorithm ends after the first iteration, without the need of adding additional behaviors. This is explained by the fact that, given the relatively large amount of products always present in the offer set, only limited substitution effects need to be captured.

Irrational instances		LoR	RB	GPT		PCMC
	M		rat	rat	irrat	25
Halo-MNL(1)	10	0.0173	0.1971	0.1995	0.0909	**0.0836**
	15	0.0209	0.1902	0.1908	0.0684	**0.0634**
	avg (all)	0.0191	0.1937	0.1952	0.0797	**0.0735**
Halo-MNL(10)	10	0.0010	0.0720	0.0783	0.0737	**0.0515**
	15	0.0014	0.0591	0.0775	0.0672	**0.0451**
	avg (all)	0.0012	0.0655	0.0779	0.0705	**0.0483**
GSP(10)	10	0.0509	0.2218	0.2289	**0.1966**	0.3047
	15	0.0586	0.2050	0.2041	**0.1247**	0.2722
	avg (all)	0.0547	0.2134	0.2165	**0.1607**	0.2885
GSP(100)	10	0.0047	**0.1227**	0.1277	0.1257	0.1617
	15	0.0063	**0.1110**	0.1201	0.1181	0.1322
	avg (all)	0.0055	**0.1168**	0.1239	0.1219	0.1470
avg (all)		0.0240	0.1542	0.1598	**0.1209**	0.1695
Rational instances						
MNL	10	0.0001	0.0409	0.0500	0.0500	**0.0310**
	15	0.0004	0.0305	0.0671	0.0671	**0.0233**
	avg (all)	0.0003	0.0357	0.0586	0.0586	**0.0272**
MMNL(10)	10	0.0001	0.0453	**0.0286**	**0.0286**	0.0303
	15	0.0003	0.0409	**0.0292**	**0.0292**	0.0326
	avg (all)	0.0002	0.0431	**0.0289**	**0.0289**	0.0314
RB(10)	10	0.0004	0.0284	**0.0237**	0.0375	0.0691
	15	0.0006	**0.0124**	0.0279	0.0317	0.0610
	avg (all)	0.0005	**0.0204**	0.0258	0.0346	0.0651
RB(100)	10	0.0002	**0.0459**	0.0606	0.0759	0.0682
	15	0.0005	**0.0275**	0.0528	0.0649	0.0602
	avg (all)	0.0004	**0.0367**	0.0567	0.0704	0.0642
avg (all)		0.0003	**0.0340**	0.0425	0.0481	0.0470

Table 11 **Test error comparison on structured instances, where 6 out of 10 products are always present in the offer set. The metric reported is the average L1 error per offer set.**