SMART WORKING DOCUMENT

Adaptive rank sampling and updating with robust solutions for non-parametric assortment planning

Authors:	*Affiliation:*
Renjun Hu	NUS
Patrick Jaillet	SMART
Tien Mai	SMART, MIT

Abstract

In this paper, we show connections between parametric and rank-based choice models. We propose a new approach to sample ranks and update their distribution from population. We then propose a data-driven robust optimization model, i.e., likelihood robust optimization, for non-parametric assortment planning and we show how to solve the robust model in a tractable way. We provide experimental results using a real-like retail dataset, which show the efficiency of our rank sampling approach and the tractability of our robust method.

November 24, 2019

1 Introduction

A problem of central interest to operations managers is to determine the choices of product offered to the customers. For example, an online social platform need to design their vacancy part of pages to provide advertisement on it. A production enterprise must decide the set of products with different specifications that is offered to their customers (like Apple provides IPhones excluding the 128 GB type). A convenient store manager (like 7-11 shop) must decide the types of good on shelf to maximize profits, provided its limited storage space. In all these examples, an optimized selection of "products" is crucial in operation: It must be a trade off of losing customers purchasing only on low-profits items with gaining from forcing some customers to switch high-profits ones.

The major concern of assortment operations is how to leverage the trade off so as to maximize the total profits. Traditionally, the literature models the assortment problem under the independent demand assumption, that is, customer decides on whether to purchase an item regardless of what he has purchased or what else has been provided. This approach reduce the computational cost since the optimal strategy is simply including all products contributing positive profits. However, such model is cursed for its oversimplification in describing customer behavior. The marginal profit of providing a new item is indeed dynamic: If a cheaper perfect substitute is provided, definitely most customer will switch to it, which results in a loss of sales and overstocking cost of the original item. Even if two items are not correlated in terms of substitute of complement, a customer is usually constrained by budget such that he can only choose one of them.

Customer choice have been introduced and widely applied to model such behaviors. In choice model, customer is assumed to walk in a restaurant and select his most preferred dishes available in menu: He might have some else dishes more preferred, but the chef does not provide them. This purchase behavior can also be interpreted by searches along ones most preferred items and stops when finding something available. To model the complexity of customer, a random utility model (RUM) is introduced assuming that each customer's utility is drawn from some distribution and his "search list" is done by descending utility. Most existing RUM model is parametric: We define the utility function in some mathematical form and fit in the model with data to derive the parameters of functions.

Parametric RUM model have been successful, or popular in modeling customer's behavior. There is vast literature ranged from marketing, economics to sociology and psychology devoted to constructed their parametric model in different settings. On the other hand, the success in fitting the decision to choice model implicitly requires fitting the "correct" parametric model to the problem, which as well requires an in-depth understanding to the problem and a careful check of in what extent the model assumptions matches the context. In all considerations, it is never an easy task. Even if we find the "correct" model to fit our settings, parametric models suffers from over/under-fitting issue: Once the parametric model is fixed, its requirement of data is also fixed such that we may not glean more information from extra data, or need to hypothesize on some missing parts.

To handle the issue of model mismatching, researchers have developed non-parametric method to describe decision behavior by imposing no assumption on the utility functions. In the non-parametric rank based model, each customer is allocated to a type of unique preferences list which is able to capture a general representation of how provided assortment will affect on customer choice. We estimate the population by estimating the proportion of different types of customer from sales data. This type of model is assumption-lax as it only requires the sales information of items. However it is also a curse that it fails to glean more information from extra data.

Apart from the non-parametric approach to deal with model mismatching, researchers have developed literature along the line of data-driven by extending the formulation of utility function to capture more data. However, when it comes to assortment planning, overly complex formulations of choice behavior will make the corresponding optimization problem hard or even intractable.

We hope to take the benefits from both the assumption-lax of non-parametric approach and the data-driven in the well-developed parametric approach. Under this motivation, we developed a tractable and adaptive framework that serves to connect the state-of-art utility-based choice model to the assumption-lax rank-based model. We prove that there is an implicit but strong connection from random utility model (RUM) to the rank based (RB) one. We purpose a rank sampling and update algorithm that convert the non-tractable complex utility function formulation to the standard rank opt formulation. We also consider the uncertainty issue due to the insufficient sales data and handle it with robust approach

1.1 Contributions

We summarize our contribution as follows. We present a new approach of making assortment decision by combining RB model with RUM model. Traditionally, it has been widely believed that RB model is more general in modelling choice behavior as it captures RUM model (**?**). Nevertheless, we prove that RUM model is as general as RB model. This result provides a bridge to connect the two seemingly disjoint We henceforth develop a general framework to estimate population that benefits from both RUM and RB model. This framework is adaptive as it does not impose restriction on the choice of RUM model: one may choose it flexibly depending on the nature of problem even beyond assortment optimization. Besides, it is data-driven since it is capable to process the data from both RUM and RB mode such that our estimation is further refined by connecting them.

We consider the uncertainty of rank distribution from estimation on data. The uncertainty issue of direct rank estimation is especially threatening when the quality of data is questionable, for example, seasonal or insufficient data that leads to bias and distortion. We connect the maximum likelihood estimation (MLE) principle to likelihood robust optimization (LRO) framework. Instead of determining the best assortment at the maximized likelihood point, we consider all the "likely enough" scenarios, as defined in LRO uncertainty set, and design an assortment with performance guarantee even in the worst case. The novelty of this approach is that the estimation part of EM algorithm can directly input as part of LRO formulation.

We demonstrate the effectiveness of our framework in computation. We begin by comparing our ranks sampling & update (SU) approach with non-parametric column generation stated in **?**. The experimental result shows that our approach is significantly successful in predicting the purchase behavior when we choose the correct RUM model to generate the ranks. This approach is comparable with column generation method when the assumption of model is somewhat different from the true population. Note that we benchmark the two approaches in same baseline such that no data other than transaction records is provided. It is demonstrated that the SU approach provides an at least no worse (or better) expression of customer population. Being able to acquire extra data by choosing adaptive RUM model, SU algorithm is highly potential to refine the prediction in today's data abundant circumstances.

We continue with processing a set of real retail data to compare the LRO formulation (**?**) with the original MIP one. We test on the expected profits under the optimized assortment of each framework. We vary the experiment by testing on different size of ranks from SU method. By the computational results, the LRO optimized assortment is especially successful when size of ranks is low, though the improvement diminishes as the size goes up. We provide an intuitive explanation on this phenomenon: A not enough rank size will be likely to sample only few (or only one) rank preferring more on some high profit items. The robust approach reduces the under-fitting issue since it considers the scenario that the distribution on that kind of rank is indeed high.

Lastly, we provide two heuristic methods to solve the LRO formulation. Our simulation shows the optimized assortment derived by linear relaxation on $\{0, 1\}$ is comparable with the old formulation at low ranks and better at high ranks. We also develop a more concise heuristic method that greatly reduces the complexity of conic formulation and the branch & bound algorithm search process implemented in commercial solvers.

1.2 Literature Review

One of the most fundamental work to model discrete choice behavior is the RUM framework by McFadden (1978). The additive RUM framework by Fosgerau et al. (2013b) and McFadden (1978) assumes that a customer's utility is the addition of the deterministic part v_{ni} and random term ϵ_{ni}. Besides, multiplicative RUM framework by Fosgerau and Bierlaire (2009) model the utility by multiplication $\epsilon_{ni} v_{ni}$. The multinomial logit (MNL) model is a special case of additive RUM that assumes the random term ϵ_{ni} are i.i.d distributed extreme value type I. It is built on the assumption of Independence from Irrelevant Alternative (IIA).

In practice, IIA assumption is usually vulnerable such that the random term ϵ_{ni} are indeed correlated. Several extension of the MNL model have been developed that serves to relax the

IIA assumption hence capture a wider range of customer's behavior. One relaxation approach is conducted by making different assumptions on the random term. These models include the nested logit by Ben-Akiva and Lerman (1985) and Ben-Akiva (1973), the cross-nested logit by Vovsha and Bekhor (1998), the generalized nested logit by Wen and Koppelman (2001), the ordered generalized extreme value (GEV) by Small (1987) and specialized compound GEV models by Whelan et al. (2002). Among them, the cross-nested model is referred as the most flexible one as it can approximate any additive RUM model. The other approach is called the mixed MNL (MMNL) that the customer population is assumed to come from a set of populations from MNL model. According to McFadden and Train (2000), this model is fully flexible in the sense that it can approximate any utility based choice model.

As discussed section 1.1, although parametric choice model have been adapted to many settings that prove to be accurate, it is not easy to decide which parametric model to fit the problem context. A simple parametric model like MNL could cause under-fitting issue in predicting choice behavior, while the complicated ones like cross-nested model is likely to overfit. To circumvent the "fitness" issue, researchers have developed non-parametric method that make minimal assumptions on customer choice. Farias et al. (2013) proposed a general rank based choice model that assigns each customer into a rank and estimate population by estimating the proportion of each rank. and proceeds by formulating the assortment optimization under the rank expression. Note that there is an implicit robust idea in their approach as it looks for the sparsest set of distribution (least number of ranks) that is consistent with transaction data. This consideration reduces the computation cost while in same time oversimplifies the problem: Given a fixed transaction data on some fixed assortments, there are many ranks with the corresponding distribution that matches but represent completely different population. In econometrics literature, Sher et al. (2011) consider a partial identification over all ranks (preference relations) to make the estimation tractable.

Along the line of rank based assortment, Bertsimas and Mišić (2015) provide a linear MIP formulation to assortment problem, which further support by showing the exact linear relaxation is the tightest among other formulations (Bertsimas and Mišić, 2018). Their approach does not emphasize on robust approach, but rather set the violation from rank estimation to sales data and employ column generation to generate rank reducing violation the most. van Ryzin and Vulcano (2015) propose a rank generation method by maximum likelihood. Nevertheless, the column generation is faced with the same problem of representing the wrong population.

In perspective of assortment problem under choice model, there is a variety of choice model research works under the setting of assortment optimization. These include the most basic MNL model considered in Talluri and van Ryzin (2014), dynamic MNL in Rusmevichientong et al.(2010), robsut MNL in Rusmevichientong and Topaloglu (2012), NL in Davis et al. (2014), MMNL in Rusmevichientong et al. (2004), robust LCMNL in Bertsimas and Mišić (2016), Markov chain choice in Blanchet et al. (2013).

Computational efficiency is another focus in assortment optimization. Aouad et al. (2018) shows that even the approximation of rank-based assortment optimization is indeed NP-hard. Jagabathula (2014) developed a heuristic ADXOPT method that can applied to a variety of choice model in assortment optimization with theoretical guarantee. Bertsimas and Mišić (2018) design algorithm based on Bender's decomposition and solve the sub-problem by heuristic search.

2 Adaptive rank sampling and updating

We present our framework connecting Rank Based(RB) and Random Utility Model(RUM) to estimate the population. In section 2.1, we prove that the two types of model are equivalently general in terms of representing a population's choice behavior. We then propose the sampling method in section 2.2 and updating algorithm in section 2.3.

In standard RB assortment optimization, one requires to know the types of the customer as well as the corresponding distribution to determine the optimal assortment. However, these information cannot be observed directly from real data. Instead, usually retailers record the sales of different item and the features to imply their relation (for example substitutes or complements) and predict the future sells. However, In this section, our goal is to provide an adaptive algorithm, called rank sampling and updating (SU), which is able to process a variety of retails data and output the most appropriate ranks with the corresponding distribution and counts.

The first step is to specify the ranks of the customers. Clearly it is not possible to enumerate all $(n+1)!$ ranks even for $n \geq 10$. We also notice that under the RB choice model, a customer makes decision by checking through the availability of his most preferred item to least preferred. Suppose each of the items are provided with probability p, then the probability that a random customer will choose his i-th preferred item is p^i. In other words, ranks with difference only in less preferred part will perform almost identically in experiment. It greatly reduces the number of ranks while retains to represent the underlying population if we can smartly choose the "representative" ranks. But how?

One answer to the question is to estimate the model from the purchase data. In the work of **?** and **?**, the data is assumed to be the probability of purchasing item i given the assortment S_m. They first consider the entire sent of $(n+1)!$ ranks on the total set of options $\{0, 1, 2, .., n\}$. We define a matrix $\mathbf{A}$ as the matrix of $A^k_{i,m}$ values, where

$$A^k_{i,m} = \mathbb{I}\{i = \underset{j \in S_m \cup \{0\}}{\arg\min}\ \sigma^k(j)\}$$

the distribution of the ranks $\mathbf{p} = (p^1, ..., p^{(n+1)!})$ and the vector $\mathbf{v}$ as the vector of $v_{i,m}$, the proportion of people purchasing item i given assortment m. Clearly we have

$$v_{i,m} = \mathbb{P}(i|S_m) = \sum_k p^k \cdot \mathbb{I}\{i = \underset{i' \in S_m \cup \{0\}}{\arg\min}\ \sigma^k(i')\}$$

Following the definition of $\mathbf{A}$ and $\mathbf{v}$, we have $\mathbf{Ap} = \mathbf{v}$. However matrix $\mathbf{A}$ is 0-1 matrix in $\mathbb{R}^{(n+1)! \times (n+1)M}$, which implies that the vertex of the corresponding polyhedron is extremely sparse. Hence, we may consider the entries (ranks) with non-zero basic feasible solution in the original formulation to make the problem tractable. **?** purpose a method that put the violation $\|\mathbf{Ap} - \mathbf{v}\|_1$ as an optimization objective and greedily add ranks by column generation methods to reduce the violation.

However, there are some potential issues. On the one hand, the exact column generation is iteratively solving large scale integer programming that is only applicable to toy data set, while the heuristic method is underperforming which we will illustrate in the numerical section later. On the other hand, the column generation estimation is myopic as it learns only from the observed choice probability $\boldsymbol{v}$. Inspired by the success of machine learning in prediction task, we hope to develop an interface that allows access to extra data to refine the prediction of customer choice.

In section 2.1 we prove that the two types of model, RUM and RB, are equivalently general to represent a population's choice bahavior. This serves as a foundation of our framework's establishment. In section 2.2 we gives our rank sampling method, and follows by the updating method in section 2.3.

2.1 Parametric versus non-parametric models

Random utility models have received scholars' attention for decades. It assumes that each product (or in general, alternative) in a choice set is assigned a random utility $u_i \in \mathbb{R}$, $i \in \mathcal{V} = \{1, ..., n\}$. An customer choose a product that maximize this utility. The probability of selecting a product i in a choice set $S \in \mathcal{V}$ is

$$\mathbb{P}^{RUM}(i|S) = \mathbb{P}(u_i \geq u_j, \forall j \in S, j \neq i)$$

The most widely used one is the multinomial logit (MNL), and there are more flexible ones such as the nexted or cross-nested or mixed-logit models (). Under some IIA assumptions, these parametric model can approximate a random utility model (RUM).

In our setting, we employ the RB model, a non-parametric one where the choice probability is described by

$$\mathbb{P}^{RB}(i|S) = \sum_{k \in \mathcal{M}(i,S)} p^k$$

where $\mathcal{M}(i, S) = \{k | \ \sigma^k(i) \leq \sigma^k(j), \ \forall j \in S\}$. In literature, one of the motivation to use RB model is its capacity to describe a more general customers' behavior. Consider a set of random utility $\mathbf{U} = \{u_1, .., u_n\}$ and a distribution of over all ranks $(\mathbf{P}, \Omega)$. Mathematically, RB model is more general if for any RUM with utility $\mathbf{U}$, there exists a RB model $(\mathbf{P}, \Omega)$ that

$$\mathbb{P}^{RUM}(i|\mathcal{V}, \mathbf{U}) = \mathbb{P}^{RB}(i|\mathcal{V}, \mathbf{P}, \Omega),\ \forall i \in \mathcal{V}$$

The question here is that, given a RB model $(\mathbf{P}, \Omega)$, is there any RUM model $\mathbf{U}$ such that for any subset $S \subset \mathcal{V}$, the probability distributions over S given by the RUM and RB model are identical. This issue is particularly relevant to the context of assortment planning. To address the question, we first give the following definition

Definition 1. *Two choice models $\boldsymbol{A}$ and $\boldsymbol{B}$ on set $\mathcal{V} = \{1, \ldots, m\}$ are said "equipvalent", denoted by $\boldsymbol{A} \rightleftharpoons \boldsymbol{B}$ if the probability distributions over any subset of $\mathcal{V}$ given by the two models are identical, i.e.,*

$$P^{\boldsymbol{A}}(i|S) = P^{\boldsymbol{B}}(i|S),\ \forall S \subset \mathcal{V}, \forall i \in S.$$

On the other hand, $\boldsymbol{A}$ and $\boldsymbol{B}$ are said to be "not-equipvalent", denoted by $\boldsymbol{A} \not\rightleftharpoons \boldsymbol{B}$ if there exist $S \subset \mathcal{V}$ and $i \in S$ such that

$$P^{\boldsymbol{A}}(i|S) \neq P^{\boldsymbol{B}}(i|S).$$

Our first theorem says that the RB and RUM models are equivalent in representing probability distributions on every subset of $\mathcal{V}$.

Theorem 1 (RB and RUM are **equivalent** in representing all "in-subset" distributions)**.** *For any* RUM($\boldsymbol{U}$)*, there always exits a RB model $(\boldsymbol{P}, \Omega)$ such that* RUM($\boldsymbol{U}$) $\rightleftharpoons$ RB($\boldsymbol{P}, \Omega$)*, and vice-versa.*

Proof. We first prove that for any RUM($\mathbf{U}$), there always exits a RB model $(\mathbf{P}, \Omega)$ such that RUM($\mathbf{U}$) $\rightleftharpoons$ RB($\mathbf{P}, \Omega$). First, let us define a mapping $\sigma \to \tau(\sigma)$ for any $\sigma \in \Omega$ such that (i) $\tau(\sigma)$ is a permutation of $\{1, \ldots, m\}$ and (ii) $\tau(\sigma)_i = j \in \mathcal{V}$ such that $\sigma_j = i$. In other words, $\tau(\sigma)_i$ is the position of product i in the permutation σ.

We start by considering a permutation $\sigma = \{\sigma_1, ..., \sigma_m\} \in \Omega$ and compute

$$\begin{aligned} P^{\text{RUM}}(\sigma) &= P^{\text{RUM}}(u_{\sigma_1} \geq u_{\sigma_2} \geq, \ldots, \geq u_{\sigma_m}) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{u_{\sigma_1}} \ldots \int_{-\infty}^{\sigma_{m-1}} f(\mathbf{U}) d(u_{\sigma_m}) \ldots d(u_{\sigma_2}) d(u_{\sigma_1}) \end{aligned}$$

where $f(\mathbf{U})$ of the density function of the random vector $\mathbf{U}$. Now, we construct a RB model with probabilities

$$p_\sigma = P^{\text{RUM}}(\sigma), \forall \sigma \in \Omega.$$

Now, given any choice set $S \subset \mathcal{V}$, the choice probability of a product $i \in S$ given by the RB model becomes

$$
\begin{aligned}
P^{\mathrm{RB}}(i|S) &= \sum_{\sigma\in\mathcal{M}(i,S)} p_\sigma = \sum_{\sigma\in\mathcal{M}(i,S)} P^{\mathrm{RUM}}(\sigma) \\
&= \sum_{\sigma|\ \tau(\sigma)_i\geq\tau(\sigma)_j,\ \forall j\in S} P^{\mathrm{RUM}}(\sigma) \\
&= P^{\mathrm{RUM}}(u_i \geq u_j, \forall j\in S, j\neq i) \\
&= P^{\mathrm{RUM}}(i|S,\mathbf{U}).
\end{aligned}
$$

So, we have $\mathrm{RUM}(\mathbf{U}) \rightleftharpoons \mathrm{RB}(\mathbf{P},\Omega)$ as required.

To prove the opposite direction, we consider a RB model $\mathrm{RB}(\mathbf{P},\Omega)$ and show that there is a $\mathrm{RUM}(\mathbf{U})$ that gives identical "in-subset" distributions. To do so, we just take a random vector $\mathbf{U} = (u_1,\ldots,u_m)$ that has a discrete distribution and takes random values in $\{1,\ldots,m\}$ with the following probabilities

$$
P(\mathbf{U}=\sigma) = p_\sigma,\ \forall\sigma\in\Omega.
$$

So, basically, the support set of $\mathbf{U}$ is also Ω and there are $m!$ values that $\mathbf{U}$ can takes. Indeed, with the above RUM model, the probability of a product i in a choice set $S\subset\mathcal{V}$ becomes

$$
\begin{aligned}
P^{\mathrm{RUM}}(i|S,\mathbf{U}) &= P^{\mathrm{RUM}}(u_i\geq u_j,\ \forall j\in S) \\
&= \sum_{\sigma\in\mathcal{M}(i,S)} P(U=\sigma) \\
&= P^{\mathrm{RB}}(i|S,\mathbf{P},\Omega)
\end{aligned}
$$

So, again, with the chosen RUM model, we have $\mathrm{RUM}(\mathbf{U}) \rightleftharpoons \mathrm{RB}(\mathbf{P},\Omega)$, which completes the proof. □

Remark. *Previous studies often argue that the RB is more general than RUM. But the above theorem says that two choice modeling schemes are actually equivalent, in the sense that the "in-subset" distributions given by any RUM can be captured identically by a RB, and vice-versa. In the context of the assortment optimization problem, this means that the use of the two schemes may be equivalent.*

The following proposition tells us how a RB model captures a RUM with i.i.d. utilities. This a direct result from the proof of Theorem 1

Proposition 2 (RB captures RUM models with i.i.d. random utilities)**.** *Given a RUM model* ($\boldsymbol{U}$) *and assume that* u_i, $\forall i\in\mathcal{V}$, *are i.i.d, if we define the following a RB model with the following probability for each permutation* $\sigma=\{\sigma_1,\ldots,\sigma_m\}$

$$
p_\sigma = P^{\mathrm{RB}}(\sigma) = \prod_{t=1}^{m-1}\int_{-\infty}^{0} dF_{\sigma_t\sigma_{t+1}}(u),
$$

where $F_{\sigma_t\sigma_{t+1}}$ *is the cumulative function of* $u_{\sigma_t}-u_{\sigma_{t+1}}$, *then*

$$
\mathrm{RUM}(\boldsymbol{U}) \rightleftharpoons \mathrm{RB}(\boldsymbol{P},\Omega)
$$

In the case of the MNL model, we have

$$\int_{-\infty}^{0} dF_{\sigma_t\sigma_{t+1}}(u) = \frac{\exp(v_{\sigma_t})}{\exp(v_{\sigma_t}) + \exp(v_{\sigma_{t+1}})},$$

which leads to the following result.

Corollary 1 (RB captures MNL)**.** *If the RUM is the MNL model with utilities $u_i = v_i + \epsilon_i$, where ϵ_i, $i \in \mathcal{V}$, are i.i.d. extreme value type I, then the RB model defined as*

$$p_\sigma = \prod_{t=1}^{m} \frac{\exp(v_{\sigma_t})}{\exp(v_{\sigma_t}) + \exp(v_{\sigma_{t+1}})}$$

is "equivalent" to the MNL model, i.e., $\mathrm{RB}(\boldsymbol{P}, \omega) \rightleftharpoons \mathrm{MNL}(\boldsymbol{U})$*, or*

$$P^{\mathrm{RB}}(i|S, \boldsymbol{P}, \Omega) = \frac{\exp(v_i)}{\sum_{j\in S}\exp(v_j)} \;\forall i \in S,\; \forall S \subset \mathcal{V}.$$

A relevant and important question here is whether RUM and RB models can capture any "in-subset" distribution. To address this question, we first introduce a general non-parametric choice model, called universal choice model (UNV), which can capture any distribution on any subset of $\mathcal{V}$

Definition 2. *An UNV choice model is defined by a set $\mathcal{P} = \{p_i^S,\ \forall S \subset \mathcal{V},\ \forall i \in S\}$ such that $p_i^S \geq 0$ and*

$$\sum_{i\in S} p_i^S = 1,\ \forall S \subset \mathcal{V}.$$

Proposition 3. $|\mathcal{P}| = m2^{m-1}$.

Proof. Given any $t = 1, \ldots, m$, the number of subsets of size t is $\begin{pmatrix} m \\ t \end{pmatrix} = m!/(t!(m-t)!)$. So we have $|\mathcal{P}| = \sum_{t=1}^{m} t \begin{pmatrix} m \\ t \end{pmatrix}$. To prove the result, we simply derive $(1+x)^m = \sum_{t=0}^{m} x^m \begin{pmatrix} m \\ t \end{pmatrix}$. Then by taking the first derivative of (**??**) we get $m(1+x)^{m-1} = \sum_{t=0}^{m} tx^{t-1} \begin{pmatrix} m \\ t \end{pmatrix}$. Replacing $x = 1$ we get the desired results. □

The following theorem says that the RUM and RB are not *universal* in the context of Definition 2, i.e., they might be not able to capture the probability distributions given by any UNV model.

Theorem 4 (RB and RUM are not universal)**.** *If $m \geq 3$, then there is an UNV model ($\mathcal{P}$) such that there is **no** RM or RUM model* $(\boldsymbol{P}, \Omega)$ *such that* $\mathrm{RB}(\boldsymbol{P}, \Omega) \rightleftharpoons \mathrm{UNV}(\mathcal{P})$*.*

Proof. Since RB and RUM are equivalent in capturing "in-subset" distributions, it is sufficient to prove the result for the RB model. We first consider the case $m = 3$. There are 6 possible permutations of $\{1,2,3\}$ and we denote the probabilities $p_1, p_2, p_3, p_4, p_5, p_6$ for ranks $(1,2,3)$, $(1,3,2)$, $(2,1,3)$, $(2,3,1)$, $(3,1,2)$, $(3,2,1)$, respectively. If the RB model captures an $UNV(\mathcal{P})$, we have the following equations

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{pmatrix} = \begin{pmatrix} p_1^{\{1,2\}} \\ p_1^{\{1,3\}} \\ p_2^{\{2,3\}} \\ p_1^{\{1,2,3\}} \\ p_2^{\{1,2,3\}} \\ p_3^{\{1,2,3\}} \end{pmatrix}, \tag{1}$$

where p_i^S is the probability of item i in subset $S \subset \{1,2,3\}$. If we choose $p_1^{1,2,3} > p_1^{\{1,2\}}$, then using (1) we have $p_1 + p2 > p_1 + p_2 + p_5$ or $p_5 < 0$. So, in other words, if we choose an UNV model such that $p_1^{1,2,3} > p_1^{\{1,2\}}$, then this UNV mode is not equivalent to any RB model. We validate the results for $m = 3$.

Now, let consider $m > 3$, we just choose an UNV model such that for any $S \subset \mathcal{V}$ and $S \cap \{1,2,3\} = \emptyset$, $p_i^S = 1/|S|$, and if $S \cap \{1,2,3\} \neq \emptyset$ then $p_i^S = 0$ for all $i \notin \{1,2,3\}$. This setting will force any item that is not in {1,2,3} having a probability of zero. If there is a RB model equivalent to the UNV mode, then we easily have $p_\sigma = 0$ if the rank σ has an item ranked higher than any item in {1,2,3}. So, the RB model is reduced to the case of $m = 3$ and we can apply the above results to show that there is no RB model being equivalent to the chosen UNV. We complete the proof. □

Remark. *The number of variables in an UNV model is $m2^{m-1}$ and in a RB model is $m!$. Even though $m2^{m-1} < m!$ if m is large enough, UNV strictly subsumes RB.*

Proposition 5 (RB strictly subsumes RUM of i.i.d utilities)**.** *If $m \geq 3$ then there is a model* $\mathrm{RB}(\boldsymbol{P}, \Omega)$ *such that there is no RUM model with i.i.d utilities that can give the same "in-subset" distributions, i.e.,*

$$\mathrm{RB}(\boldsymbol{P}, \Omega) \not\asymp \mathrm{RUM}(\boldsymbol{U}), \forall \boldsymbol{U}$$

Proof. For the case $m \leq 2$, it is easy to validate that given any RM model, there is an MNL model that can give the same "in-subset" distributions. Now we consider the case $m = 3$ and denote the probabilities of the 6 ranks $(1,2,3)$, $(1,3,2)$, $(2,1,3)$, $(2,3,1)$, $(3,1,2)$, $(3,2,1)$ by $p_1, p_2, p_3, p_4, p_5, p_6$, respectively. We also define $P^{\mathrm{RUM}}(U_1 \geq U_2) = a$, $P^{\mathrm{RUM}}(U_2 \geq U_3) = b$ and $P^{\mathrm{RUM}}(U_1 \geq U_3) = c$. If the RUM is equivalent to the RB, then

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{pmatrix} = \begin{pmatrix} P^{\mathrm{RUM}}(1|\{1,2\}) \\ P^{\mathrm{RUM}}(2|\{2,3\}) \\ P^{\mathrm{RUM}}(1|\{1,3\}) \\ P^{\mathrm{RUM}}(1|\{1,2,3\}) \\ P^{\mathrm{RUM}}(2|\{1,2,3\}) \\ P^{\mathrm{RUM}}(3|\{1,2,3\}) \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ ac \\ (1-a)b \\ (1-c)(1-b) \end{pmatrix}. \tag{2}$$

From this, we have

$$\frac{p_1 + p_2}{p_1 + p_3 + p_4} = p_1 + p_2 + p_5,$$

which cannot be valid for any $p_1, \ldots, p_6 \geq 0$ such that $\sum_{i=1}^{6} p_i = 1$. So. as a result, there is a RB model that can not be represented by any RUM model of i.i.d utilities.

Now we move to the case $m > 3$. Similarly to the proof of Theorem 4, we can just select a RB such that $p_\sigma = 0$ for all ranks σ that has an item ranked higher than any item in $\{1,2,3\}$. This setting will reduce the RB model to the case of $m = 3$ considered above. We complete the proof. □

It is possible to show that, RUM models with i.i.d. utilities cannot approximate RB models arbitrarily well

Theorem 6. *For $m \geq 3$, there is a RM model such that there exists $\delta > 0$ such that*

$$\max_{S \subset \mathcal{V}, i \in S} \left| P^{\text{RB}}(i|S) - P^{\text{RUM}}(i|S) \right| > \delta, \; \forall \text{RUM}(\boldsymbol{U})$$

Proof. To be done, just prove for the case $m = 3$. □

Nevertheless, if we consider choice model with correlated utilities, one can show that such model can approximate arbitrarily well any RB model.

Theorem 7. *Given any RB model and scalar $\delta > 0$, there is an MEV model such that*

$$\max_{S \subset \mathcal{V}, i \in S} \left| P^{\text{RB}}(i|S) - P^{\text{MEV}}(i|S) \right| \leq \delta$$

Proof. Using **?** □

The following theorem says that if there is a RB or RUM model that can represent a UNV model, then there are infinitely many of them can represent that UNV model.

Theorem 8. *Given a model* $\text{UNV}(\mathcal{P})$*, only one of the following cases can occur: (i) There is no RB or RUM that are equivalent to the* $\text{UNV}(\mathcal{P})$*, or (ii) there is an unique RB model that are equivalent to* $\text{UNV}(\mathcal{P})$*.*

Proof. TO BE DONE □

2.2 Rank sampling by pre-training RUM model

Given the equivalence in section 3.1, it is curious to see if we can make use of equivalence property of RUM to RB model to better estimate the underlying population. RUM model has been well-developed during the decades such that there are plentiful of models to fit a variety of problem settings and data input. Some of the models are accurate yet complicated in expression. In consequence, the optimization formulation regarding the population estimation is not tractable. Since this model falls into the category of RUM, there exists a RB expression that is equivalent in choice probability. A natural idea is to convert the estimated RUM to RB model and fit with the standard RB optimization.

Nevertheless, the equivalent RB model usually contains a factorial number of ranks. To make the corresponding optimization tractable, instead of enumerating all possible permutations of items (ranks), we sample a number of utilities from RUM model and sort each of them as a rank.

We further reduce the size of ranks by eliminating the unnecessary ones. Consider the people whose highest utility is non-purchase (we say these people belong to the "bad" ranks). They will not purchase anything under any assortment such that optimizing with respect to other customers will derive a same result. In practice, it is observed that thoe "bad" rank people is quite common or even the majority. We treat all $n!$ kinds of "bad" ranks equally reduce the computational cost. The rank sampling process in summarized in algorithm 1.

Algorithm 1: Rank sampling via RUM pre-training

Input : Input Data $\mathbf{X}$, Number of ranks K, Random utility model $G(\cdot)$
Set $k = 1$, $m_{bad} = 0$;
Pre-train the utility model with data $\mathbf{X}$ to get Utility $\mathbf{U} = (U_0, U_1, , .., U_n) \sim \mathcal{D}$. **while** $k \leq K$ **do**
    Sample $\mathbf{U}^k \sim \mathcal{D}$;
    σ^k = Sort alternatives with descending U_i;
    if $\sigma^k(0) = 1$ **then**
        $m_{bad} = m_{bad} + 1$;
    else
        $k = k + 1$;
Output: Ranks σ^k, $k = 1, ..., K$, Bad ranks count m_{bad}

For a customer in a utility model, its utility can described by $\mathbf{U} \sim \mathcal{D}$ for some distribution $\mathcal{D}$. The set of the sampled ranks are equivalent to the set of utilities from RB model. Hence, we have

Proposition 9. *Sampled ranks is an unbiased and consistent estimator to all choice probability*

Proof. (Unbiasedness) By law of large numbers, we have

$$\begin{aligned}
\mathbb{P}^{RB}_{sample}(i|\mathcal{S}) &= \frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(i|\mathcal{S},\sigma^k) = \frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(\sigma^k(i)\leq\sigma^k(j),\forall j\in\mathcal{S}) \\
&= \frac{1}{K}\sum_{k=1}^{K}\mathbb{I}(u_i^k\geq u_j^k,\forall j\in S) = \frac{1}{K}\mathbb{P}(U_i\geq U_j,\forall j\in S) \\
&\xrightarrow{\text{w.p.1}} \int\cdots\int_{V(i,\mathcal{S})} f(\mathbf{U})\,du_1\ldots du_n \qquad\qquad K\to\infty \\
&= \mathbb{P}^{RUM}(i,\mathcal{S})
\end{aligned}$$

Here $f(\cdot)$ is the probability density function, $V(i,\mathcal{S})=\{(u_1,...,u_n)|u_j\leq u_i,\forall j\in\mathcal{S},j\neq i\}$

(Consistent) We use p to denote $\mathbb{P}^{RUM}(i|\mathcal{S})$ for simplification. Note that

$$Var[\mathbb{P}^{RB}_{sample}(i|\mathcal{S})] = \frac{1}{K}Var[\mathbb{I}(i|\mathcal{S},\sigma^k)] = \frac{p(1-p)}{K}$$

By Chernoff's inequality, $\forall\epsilon$ with $0<\epsilon<p$, we have

$$\begin{aligned}
\mathbb{P}(|\mathbb{P}^{RB}_{sample}(i|\mathcal{S})-p|\geq\epsilon) &= \mathbb{P}(|S-Kp|>Kp\frac{\epsilon}{Kp}) \\
&\leq 2\exp(-Kp(\frac{\epsilon}{p})^2/3) \\
&= 2\exp(-K\epsilon^2/3p)
\end{aligned}$$

□

2.3 Rank update by EM-method

The ranks sampled from RUM model with the same prior is only a crude estimation regarding the utility model. We require a quadratic large number of K for a small enough ϵ since $K\sim O(\epsilon^{-2})$ for a fixed chance constraints bound. A natural idea is that we can update the distribution of those ranks so as to fit the transaction data better. We consider the complete data log likelihood function, given by:

$$\mathcal{L}(\mathbf{p}) = \sum_{t=1}^{T}\sum_{k\in\mathcal{M}(i_t,S_t)}\mathbb{I}\{\sigma_t=\sigma_k\}\log p_k = \sum_{k=1}^{K}m_k\log p_k \tag{3}$$

Our focus is to estimate $m_k=\sum_{t=1}^{T}\mathbb{I}\{\sigma_t=\sigma_k\}$. Since the prior is given as equally distribution, we follow the Bayesian update method purposed by van Ryzin and Vulcano (2017). The probability mass function of the ranks are

$$\mathbb{P}(\sigma^k|i_t,S_t,\hat{\mathbf{p}}) = \frac{\mathbb{P}(i_t|\sigma^k,S_t,\hat{\mathbf{p}})\mathbb{P}(\sigma^k|\hat{\mathbf{p}})}{\mathbb{P}(i_t|S_t,\hat{\mathbf{p}})} = \frac{\mathbb{I}\{i\in\mathcal{M}(i_t,S_t)\}\hat{p}_k}{\sum_{h\in\mathcal{M}(i_t,S_t)}\hat{p}_h} \tag{4}$$

and obtain the estimates

$$\hat{m}_k = \mathbb{E}\left[m_k|\hat{\mathbf{p}}\right] = \sum_{t=1}^{T} \mathbb{P}(\sigma^k |\ i_t, S_t, \hat{\mathbf{p}}) = \sum_{t=1}^{T} \frac{\mathbb{I}\{k \in \mathcal{M}(i_t, S_t)\}\hat{p}_k}{\sum_{h\in\mathcal{M}(i_t,S_t)} \hat{p}_k} \tag{5}$$

We summarize the full algorithm in below.

Algorithm 2: EM algorithm

Input: Prior distribution $\hat{\mathbf{q}} = (\hat{q}_1, .., \hat{q})$,
$\mathcal{M}(i_t, S_t) := \{k : \sigma^k(i_t) \leq \sigma^k(j), \forall j \in S_t\}$
Set $m_k = 0, p_{k,t} = 0\ \forall k \in \{1, .., K\},\ t \in \{1, ..., T\}$
for $t = 1, ..., T$ **do**
  for $k \in \mathcal{M}_t(j_t, A_t)$ **do**
    Set $x_{k,t} := \hat{q}_k / \sum_{h\in\mathcal{M}_t(j_t,A_t)} \hat{q}_h$
for $k = 1, ..., K,\ t = 1, ..., T$ **do**
  $m_k = m_k + x_{k,t}$
for $k = 1, ..., K$ **do**
  $p_k := m_k / \sum_{s=1}^{K} m_s$
Output: Distribution $\mathbf{p}$, Count $\mathbf{m}$

---

3 Robust assortment under non-parametric choice models

We present our robust formulation under non-parametric choice model. In section 3.1 we define the assortment optimization problem and the rank based choice model. Then we introduce the Likelihood Robust framework to model the uncertainty and derive an explicit expression of the robust problem

3.1 Problem formulation

We formulate the assortment optimization problem as a mixed integer programming. In our model, customers are assumed to have a rank-based preference relation for products. That is, each customer has a rank σ, of m products $\{0, 1, \ldots, m\}$. In other words, σ corresponds to a permutation over $m + 1$ products. Note that product 0 refers to the non-purchase option. We further assume that each customer will be exactly one item from the assortment. This assumption is reasonable in a way that for a customer purchase nothing, he is counted as purchasing the vain item 0; and for a customer purchase multiple number of items, he is counted as the corresponding customer. The reader is referred to Farias et al.(2013) for further details

We denote by Ω the set of preference lists and p_σ is the probability of a rank $\sigma \in \Omega$. We also number the references lists in Ω as $\{1, \ldots, K = |\Omega|\}$ and the corresponding probabilities $p_1, \ldots, p_K$. Indeed we require $\sum_{k=1}^{K} p_k = 1$. Given an assortment S, the expected revenue can be computed as

$$R(\mathcal{S}, \mathbf{p}) = \sum_{i \in \mathcal{S}} r_i P(i|\mathcal{S})$$
$$= \sum_{i \in \mathcal{S}} r_i \left(\sum_{\sigma \in \mathcal{M}(i,\mathcal{S})} p_\sigma \right)$$

where $\mathcal{M}(i, S)$ is the set of all permutation $\sigma \in \Omega$ such that the product $i$is ranked highest among all the products in S. That is,

$$\mathcal{M}(i, S) = \{\sigma \in \Omega |\ \sigma(i) \leq \sigma(j),\ \forall j \in S\}.$$

Assortment optimization under the ranking-based choice model can be stated as

$$\max_{S \ni \{0\}} R(S, \mathbf{p}) \tag{6}$$

There are various ways (McBride and Zufryden, 1988; Belloni et al., 2008) to formulate the problem (6) in mixed integr programming (MIP). Bertsimas and Mišić (2017) purposed a formulation, which is further supported by their later work in Bertsimas and Mišić (2018) such that their LP relaxation problem has a tightest constraints as compared to other rank-based choice MIP formulation. We present their formulation below.

$$\begin{aligned}
\max_{x,y} \quad & \sum_{k=1}^{K} \sum_{i=1}^{n} r_i \cdot p_k \cdot y_i^k && (7) \\
\text{subject to} \quad & \sum_{i=0}^{n} y_i^k = 1,\ \forall k \in [K] \\
& y_i^k \leq x_i\ \forall k \in [K],\ i \in [n] \\
& \sum_{j:\sigma^k(j) > \sigma^k(i)} y_j^k \leq 1 - x_i,\ \forall k \in [K],\ i \in [n] \\
& \sum_{j:\sigma^k(j) > \sigma^k(0)} y_j^k = 0,\ \forall k \in [K] \\
& x \in \{0,1\}^n, \quad y \geq 0
\end{aligned}$$

Here, r_i is the marginal profit of selling a unit of item i. The binary variable x_i is 1 if product item i is included in the assortment, and 0 otherwise. y_i^k describe the purchase behavior for customer of rank k, which is 1 if and only if type k customer purchases item i. There is no binary restriction on y_i^k because the constraints will forces it to $\{0, 1\}$. One may refer to Bertsimas and Misic (2017) for the detailed description of constraints.

3.2 Likelihood robust optimization (LRO)

It is interesting to consider p_k the distribution of different types of people, a parameter estimated from data. In real operations, there are many factors that impairs our estimation to the true population, for example, a newly released product (not enough sales sample) or partial/seasonal sales data (bias). The mismatch of estimated and true population will lead to the sub-optimal performance of optimized assortment.

We put our focus to estimation of p_k. In the likelihood estimation approach, the likelihood of purchasing item i_t given assortment $\mathcal{S}_t$ is $\sum_{\sigma\in\mathcal{M}(i,\mathcal{S}_\sqcup)} p_\sigma$. The estimation of p_k is henceforth the maximizer of the likelihood function $\prod_{t=1}^{T}(\sum_{\sigma\in\mathcal{M}(i,\mathcal{S}_\sqcup) p_\sigma})$ and equivalently the maximizer of $\sum_{t=1}^{T}\log(\sum_{\sigma\in\mathcal{M}(i,\mathcal{S}_\sqcup)} p_\sigma)$.

Although the concavity of the log-likelihood function ensures that the maximizer is obtainable by convex programmig, it is costly to do so, as presented by **?**. To circumvent the computational barrier, they consider the complete log-likelihood function with specifying the arriving customer type, given by

$$\sum_{t=1}^{T}\sum_{\sigma_k\in\mathcal{M}(i,\mathcal{S}_t)}\mathbb{I}\{\sigma^t=\sigma_k\}\log p_k=\sum_{k=1}^{K}\hat{m}_k\log p_k \tag{8}$$

where $\hat{m}_k$ is given by $\sum_{t=1}^{T}\mathbb{I}\{\sigma^t=\sigma_k, k\in\mathcal{M}(i_t,\mathcal{S}_t)\}$. Instead of maximizing the log-likelihood function w.r.t p_k, we consider the case that our estimation over $\hat{m}_k$ is not proper due to, for example, partial and biased data. Under this situation, the direct maximizer p_k^* will deviate from true p_k for a considerate amount. Instead, we consider all p_k such that $\sum_{k=1}^{K}\hat{m}_k\log p_k$ is large enough that possibly cover the underlying truth. This observation leads to the motivation of employing Likelihood Uncertainty set, given as

$$\mathbb{D}(\delta)=\left\{\mathbf{p}\;\middle|\;\begin{array}{l}\mathbf{p}\in\Delta\\ \sum_{k=1}^{K}\hat{m}_k\log(p_k)\geq\delta\end{array}\right\} \tag{9}$$

One may wonder how to set the uncertainty parameter δ properly. Indeed, Wang et al. (2014) shows that under the Bayesian setting, for δ satisfying $\mathbb{P}(\mathbf{p}\in\mathbb{D}(\delta))=1-\alpha$

$$\delta-\sum_{k=1}^{K}\hat{m}_k\log\frac{\hat{m}_k}{\sum_{k=1}^{K}\hat{m}_k}\to_p-\frac{1}{2}\chi^2_{K-1,1-\alpha}$$

Under the uncertainty of distribution $\mathbf{p}$, the problem becomes a max-min optimization

$$\max_{\substack{\mathcal{S}\subset\mathcal{V}^+\\ \mathcal{S}\ni\{0\}}}\min_{\mathbf{p}\in\mathbb{D}(\delta)} R(\mathcal{S},\mathbf{p}), \tag{10}$$

To solve (10) we write the Lagrangian of the inner optimization problem

$$L(\mathbf{p}, \eta_1, \eta_2) = \sum_{k=1}^{K}\sum_{i=1}^{n} r_i p_k y_i^k + \eta_1 \left(\delta - \sum_{k=1}^{K} \hat{m}_k \log p_k\right) + \eta_2 \left(1 - \sum_k p_k\right)$$

Taking the derivative of $L(\mathbf{p}, \eta_1, \eta_2)$ and set them to zero, we get

$$\frac{\partial L(\mathbf{p}, \eta_1, \eta_2)}{\partial p_k} = 0 \Rightarrow p_k = \frac{\eta_1 \hat{m}_k}{\sum_{i=1}^{n} r_i y_i^k - \eta_2}.$$

Thus, we can write dual of $\min_{\mathbf{p}\in\mathbb{D}(\delta)} R(\mathcal{S}, \mathbf{p})$ as

$$\begin{aligned} \max_{\eta_1,\eta_2} \quad & \eta_2 + \eta_1(\delta + \hat{m} - \sum_{k=1}^{K} \hat{m}_k \ln \hat{m}_k) + \sum_{k=1}^{K} \hat{m}_k \eta_1 \log(\frac{\sum_{i=1}^{n} r_i y_i^k - \eta_2}{\eta_1}) \\ \text{subject to} \quad & \sum_{i=1}^{n} r_i y_i^k - \eta_2 \geq 0 \\ & \eta_1, \eta_2 \in \mathbb{R},\ \eta_1 \geq 0 \end{aligned} \tag{11}$$

The strong duality holds since the objective function is concave. We add the constraints on $\mathbf{x}, \mathbf{y}$ and derive the explicit optimization problem with concave objective function.

$$\begin{aligned} \max_{x,y,z,\eta_1,\eta_2} \quad & \eta_2 + \eta_1(\delta + \hat{m} - \mathcal{M}) + \sum_{k=1}^{K} \hat{m}_k \eta_1 \log\left(\frac{\sum_{i=1}^{n} r_i y_i^k - \eta_2}{\eta_1}\right) \\ \text{subject to} \quad & y_i^k \leq x_i \ \forall k \in [K],\ i \in [n] \\ & \sum_{i=0}^{n} y_i^k = 1,\ \forall k \in [K] \\ & \sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k \leq 1 - x_i,\ \forall k \in [K],\ i \in [n] \\ & \sum_{j:\sigma^k(j)>\sigma^k(0)} y_j^k = 0,\ \forall k \in [K] \\ & \eta_1 \geq 0,\ \eta_2 \in \mathbb{R},\ \mathbf{x} \in \{0,1\}^n,\ \mathbf{y} \geq 0 \end{aligned} \tag{12}$$

where $\hat{m} = \sum_{k=1}^{K} \hat{m}_k$ and $\mathcal{M} = \sum_{k=1}^{K} \hat{m}_k \ln \hat{m}_k$. To solve problem with nonlinear log function in objective and constraints, we reformulate by epigraph to utilize the computation power of

MIP solver supporting exponential cone constraints. The reformulation of (12) is

$$
\begin{aligned}
\max_{\mathbf{x},\mathbf{y},\mathbf{s},\eta_1,\eta_2} \quad & \eta_2 + \eta_1(\delta + \hat{m} - \mathcal{M}) + \sum_{k=1}^{K} \hat{m}_k t_k \\
\text{subject to} \quad & (s_k, g_k, t_k) \in \mathcal{K}_{exp}, \ \forall k \in [K] \\
& g_k = \eta_1, \ \forall k \in [K] \\
& s_k = \sum_{i=1}^{n} r_i y_i^k - \eta_2 \\
& \sum_{i=0}^{n} y_i^k = 1, \ \forall k \in [K] \\
& y_i^k \le x_i \ \forall k \in [K], \ i \in [n] \\
& \sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k \le 1 - x_i, \ \forall k \in [K], \ i \in [n] \\
& \sum_{j:\sigma^k(j)>\sigma^k(0)} y_j^k = 0, \ \forall k \in [K] \\
& \eta_1 \ge 0, \ \eta_2 \in \mathbb{R}, \ \mathbf{x} \in \{0,1\}^n, \ \mathbf{y}, \mathbf{s} \ge 0
\end{aligned} \tag{13}
$$

Note 1. *Formulation (12) is amendable for large scale optimization.*

By the monotonicity of log function, we can reformulate (12) in epi-graph decomposition.

$$
\begin{aligned}
\max_{\mathbf{x},\eta_1,\eta_2} \quad & \eta_2 + \eta_1(\delta + \hat{m} - \mathcal{M}) - \hat{m}\eta_1 \log \eta_1 + \eta_1 \sum_{k=1}^{K} \hat{m}_k t_k \\
\text{subject to} \quad & t_k \le \log(G_k(\mathbf{x}, \eta_2)) \\
& \eta_1 \ge 0. \ \eta_2 \in \mathbb{R}, \ \mathbf{x} \in \{0,1\}^n
\end{aligned} \tag{14}
$$

where

$$
\begin{aligned}
G_k(\mathbf{x}, \eta_2) := \max_{y^k} \quad & \sum_{i=1}^{n} r_i y_i^k - \eta_2 \\
\text{s.t.} \quad & y_i^k \le x_i, \ \forall i \in [N] \\
& \sum_{i=0}^{n} y_i^k = 1, \ \forall k \in [K] \\
& \sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k \le 1 - x_i, \ \forall k \in [K], \ i \in [n] \\
& \sum_{j:\sigma^k(j)>\sigma^k(0)} y_j^k = 0, \ \forall k \in [K] \\
& \mathbf{y}^k \ge 0
\end{aligned} \tag{15}
$$

with dual

$$\begin{aligned}
\min_{\alpha^k,\beta^k,\gamma^k} \quad & \log\left(\gamma^k + \sum_{j=1}^n \alpha_j^k x_j + \sum_{i=1}^n \beta_i^k (1-x_i) - \eta_2\right) \\
\text{s.t.} \quad & \gamma^k + \alpha_j^k + \sum_{i:\sigma^k(i)<\sigma^k(j)} \beta_i^k \geq r_i, \ \forall j \in [n] \\
& \gamma^k + \sum_{i:\sigma^k(i)<\sigma^k(0)} \beta_i^k \geq 0 \\
& \gamma^k + \sum_{j=1}^n \alpha_j^k x_j + \sum_{i=1}^n \beta_i^k (1-x_i) \geq \eta_2 \\
& \alpha^k, \beta^k \geq 0.
\end{aligned} \tag{16}$$

Hence problem (8) becomes

$$\begin{aligned}
\max_{\mathbf{x},\eta_1,\eta_2} \quad & \eta_2 + \eta_1(\delta + \hat{m} - \mathcal{M}) - \hat{m}\eta_1 \log \eta_1 + \eta_1 \sum_{k=1}^K \hat{m}_k t_k \\
\text{subject to} \quad & t_k \leq \min_{\alpha^k,\beta^k,\gamma^k} \log[\gamma^k + \sum_{j=1}^n \alpha_j^k x_j + \sum_{i=1}^n \beta_i^k (1-x_i) - \eta_2], \ \forall k \in [K] \\
& \gamma^k + \sum_{j=1}^n \alpha_j^k x_j + \sum_{i=1}^n \beta_i^k (1-x_i) \geq \eta_2, \ \forall k \in [K] \\
& \eta_1 \geq 0. \ \eta_2 \in \mathbb{R}, \ \mathbf{x} \in \{0,1\}^n
\end{aligned} \tag{17}$$

Bertsimas and Misic (2018) has shown that the problem (10) has an closed form minimizer $(\alpha_*^k, \beta_*^k, \gamma_*^k)$ such that for a given solution $(\eta_1, \eta_2, \mathbf{x})$, we can check whether the condition of type k is violated by direct computation. This simplifies the constraint generation process for the Bender's decomposition method. Apart from the simplicity of computing the sub-problem, this decomposition doesn't require solving the problem iteratively. In branch & bound algorithm, if a current solution $(\mathbf{x}, \mathbf{t}, \eta_1, \eta_2)$ violating a constraint, we add constraint to all nodes in the B&B tree, discard this solution and the node is resolved. The large scale algorithm to solve the linear relaxation mentioned in Bertsimas and Misic (2018) is also transferable to our formulation for the similarity in structure.

4 Experimental Results

In this section, we apply our sample & update (SU) algorithm in MMNL synthetic data and restrict the pre-train RUM model in simple MNL to demonstrate the effectiveness of SU algorithm even under a slight mismatch of RUM model. We follow by comparing the result robust formulation with the exact formulation on real retail data. We claim that our robust formulation is significant in preventing under-fitting issue at the low rank size case. We provide two heuristic method that are at least as accurate as the exact formulation that we can flexibly leverage the prediction precision and computational cost. The retail data is provided by JDA Labs and we refer reader to Palmer (2016) for more information. All experimental result reported below is implemented by Julia language in Razer Blade 2017 laptop with Intel i7700 CPU. All mathematical programming is modeled in Julia JuMP package with Mosek v9 solver.

4.1 Rank sampling & updating versus column generation

In this section, we test the estimation of true population by SU method with the column generation approach. We conduct experiments on both synthetic and retail data.

Synthetic Data

We first consider the MMNL model with T classes as the underlying true population, whose choice probability is given by

$$\mathbb{P}(i|\mathcal{S}) = \sum_{t=0}^{T} p_t \cdot \frac{\exp(v_{t,i})}{\sum_{i'\in\mathcal{S}} \exp(v_{t,i}) + \exp(v_{t,0})} \tag{18}$$

We generate synthetic data in the following ways: For each item i and customer t we generate values $q_{i,t}$ uniformly on $[0,1]$. We choose 5 of the $N+1$ items and set each chosen item with utility $\log(q_{t,i})$ and the remaining to $log(0.1q_{t,i})$ as utility. The type probability $p_1, ..., p_T$ are uniformly drawn from $(T-1)$-dimensional unit simplex. We test on $T \in \{1, 2, 5, 10\}$, $N \in \{30, 60\}$ and. We generate 2×100 random assortments with each item i has a probability of $p_{i,m} = d \in \{0.2, 0.5\}$ to appear in each assortment $\mathcal{S}_m$. We separate the assortments of equal size as train-test pair.

We restrict our RUM pre-train model to be MNL (a special case of MMNL when T = 1). We impose this restriction in consideration of mismatching issue (i.e. we fail to identity the "correct" RUM model to fit this). So here $T = 1$ is the case we are using the correct RUM model to pre-train the data, and $T \in \{2, 5, 10\}$ corresponds to selecting the wrong ones. We also impose restriction in our input data such that we only know the purchase probability $\mathbb{P}(i|\mathcal{S}_m)$ (which is enough for column generation) for some known $\mathcal{S}_m$, but not the feature information X. We multiply noise $1 + c \cdot \delta_{i,m}$ to $\mathbb{P}(i|\mathcal{S}_m)$ and scale it so that the sum over i is 1. Here $\delta_{i,m}$ is i.i.d uniformly from [0,1] and $c \in \{0.1, 1\}$.

To learn the MNL model without feature information, we use the property $\mathbb{P}(i|\mathcal{S})/\mathbb{P}(j|\mathcal{S}) = \exp(v_i)/\exp(v_j)$. Since the non-purchase option is available in all assortment, we scale $\exp(v_{0,m})$ as one (i.e. $v_{0,m} = 0$) and calculate $v_{i,m} = log[\mathbb{P}(i|\mathcal{S}_m)/\mathbb{P}(j|\mathcal{S}_m])$ respectively. Here $v_{i,m}$ denote the estimation of v_i from assortment $\mathcal{S}_m$, and note that $v_{i,m}$ does not necessary need to equals to v_i due to the noise. We set our estimator of v_m by $\hat{v}_m := \log[\sum_{m:i\in\mathcal{S}_m} \mathbb{P}(i|\mathcal{S}_m)/\mathbb{P}(j|\mathcal{S}_m)]$.

We applied the heuristic column generation method, as described in Bertsimas and Misic (2017). In consideration of tractability issue of the optimization problem, we do comparison by controlling the size of ranks equal. We vary the rank size $K \in \{100, 200, 400\}$ and set the calibration as:

$$G = \frac{1}{M}\sum_{m=1}^{M} \|\mathbf{p}_{predict}^{m} - \mathbf{p}_{true}^{m}\|_2^2 = \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{N}\{\mathbb{P}_{predict}(i|\mathcal{S}_m) - \mathbb{P}_{true}(i|\mathcal{S}_m)\}^2 \tag{19}$$

(T, N, c, D)	SU 100	CG 100	SU 200	CG 200	SU 400	CG 400
(1, 30, 0.1, 0.5)	**5.52**	11.5	**2.97**	4.98	**1.54**	2.43
(1, 30, 0.5, 0.5)	**5.74**	8.23	**2.45**	3.95	**1.69**	3.34
(1, 30, 1.0, 0.5)	**6.22**	13.2	**3.29**	5.06	**1.31**	2.67
(1, 60, 0.5, 0.5)	**6.46**	28.2	**3.42**	10.6	**1.77**	2.08
(1, 30, 0.5, 0.2)	**3.97**	4.76	**1.12**	1.23	**0.64**	1.16
(2, 30, 0.5, 0.5)	**8.64**	10.6	**2.77**	3.81	**1.17**	1.28
(2, 60, 0.1, 0.5)	**8.42**	22.3	**2.63**	8.33	**1.73**	2.82
(2, 60, 0.1, 0.2)	7.93	**7.56**	**1.66**	3.92	**0.68**	0.85
(5, 30, 0.1, 0.5)	**7.55**	9.78	3.50	**3.41**	1.27	**1.02**
(5, 30, 0.5, 0.5)	**6.73**	10.0	**2.88**	3.87	**1.89**	2.54
(5, 60, 0.5, 0.2)	**6.14**	6.97	**1.84**	2.48	**0.72**	1.16
(10,30, 0.1, 0.5)	**6.36**	8.99	**2.66**	3.65	1.70	**1.33**
(10,60, 0.5, 0.2)	**6.93**	8.00	**2.45**	2.55	**0.72**	1.59

Table 1: Purchase Estimation Precision G (unit: 10^{-3})

Table 1 is a summary of experimental results of parameter pairs (N, L, c, d) by different rank generation method. SU 100 denotes the estimation result from sample and update method under K = 100 ranks, CG is for column generation.

(T, N, c, D)	SU 100	CG 100	SU 200	CG 200	SU 400	CG 400
(1, 30, 0.1, 0.5)	**2.74**	6.36	**5.11**	15.3	**9.79**	47.3
(1, 30, 0.5, 0.5)	**2.68**	6.96	**5.27**	15.8	**9.65**	45.1
(1, 30, 1.0, 0.5)	**2.69**	6.45	**5.12**	15.5	**9.98**	44.4
(1, 60, 0.5, 0.5)	**4.86**	19.4	**9.09**	42.3	**16.6**	98.3
(1, 30, 0.5, 0.2)	**3.22**	6.56	**5.81**	15.52	**10.8**	44.0
(2, 30, 0.5, 0.5)	**2.74**	6.24	**5.25**	15.26	**10.0**	49.0
(2, 60, 0.1, 0.5)	**4.82**	18.36	**9.13**	40.20	**17.1**	97.7
(2, 60, 0.1, 0.2)	**5.43**	18.56	**10.3**	40.91	**20.2**	97.1
(5, 30, 0.1, 0.5)	**2.80**	6.20	**5.26**	16.07	**10.0**	44.7
(5, 30, 0.5, 0.5)	**2.83**	6.29	**5.27**	15.18	**10.0**	43.82
(5, 60, 0.5, 0.2)	**5.66**	19.15	**10.89**	41.53	**20.2**	97.12
(10,30, 0.1, 0.5)	**2.80**	6.21	**5.24**	15.25	**10.1**	44.9
(10,60, 0.5, 0.2)	**5.54**	19.28	**10.38**	40.74	**20.2**	98.6

Table 2: Computational cost (unit: s)

Table 2 is a summary of the computational cost. We noticed that the only the choice rank generation method, number of items N and number of ranks are the influencing factor of computational time. We also tried on the exact column generation algorithm in Bertsimas and Misic (2017), which searches for the global optimum by solving MILP with $O(N(N+K))$ number of binary constraint in each of the K iterations. The first iteration of N = 30 case takes 10^3 second to solve and we terminate here claiming exact column generation is not tractable even for toy model.

Retail Data

In the JDA retail data, the seller is offering a total of $N = 375$ items (including the non-purchase item) and they recorded the sales proportion on $M = 3565$ different assortments. We conduct equal train-test split (i.e. each assortment is equally likely to appear in train and test set) on assortments data and compared the prediction accuracy from each choice model on test set. We use kernel MNL to pre-train the RUM model and vary the experiments by different number of samples $T \in \{50000, 100000, 200000\}$. For both SU and CG method, we experiment on different size of ranks $K \in \{250, 500, 1000\}$. Below is the summary of the results

Size K	CG	SU(50000)	SU(100000)	SU(200000)
250	0.0161, 5782.1	0.0162, 2129.9	0.0160, 2273.1	0.0159, 2535.4
500	0.0160, 13365.0	0.0154, 2160.5	0.0151, 2343.9	0.0152, 2632.2
1000	0.0159, 30973.2	0.0149, 2237.7	0.0148, 2455.0	0.0148, 2884.2

Table 3: Estimation Precision (unit: 10^{-2}); Computational Cost(unit: s)

One can see from the table that the Column generation approach is performing comparably same with SU algorithm at low rank case, while in the high rank cases, SU algorithm is substantially better. Besides, SU algorithm consumes computation power majorly in the step of pre-training RUM model such that increasing the number of samples T is computationally feasible.

4.2 LRO formulation versus linear MIP formulation

We generate ranks obtained from the kernel MNL model with size $K_{train} \in \{250, 500, 1000, 2000\}$. In each time t, we sample the pair (i_t, S_t) which is required for the EM update. Since the prior distribution of the assortment is not known, we assume that each assortment m is equally likely to be provided (i.e. $\lambda_m = 1/M$, see Note 5 for the notations). We then sample the purchase item i_t by the probability mass function $\mathbb{P}(i|S_t)$ provided in the transaction data. We set the number of sample iteration to be $T = 20000$ for all K. We use the result from EM update to formulate the linear MIP and LRO MIP respectively to derive assortment solutions $\mathbf{x} \in \{0, 1\}^n$.

Since the true population is never revealed, given a solution $\mathbf{x}$, one can never check the performance of this solution. Hence, it makes sense for us to use another rank model, generated independently of the original one, to test the solution. Here we set test rank size to be $K_{test} = 1000$ for all K_{train} and set the $T = 20000$ for the EM update process. We then compute the expected return of this assortment by $R(S, \hat{\mathbf{p}}) = \sum_{i \in S} r_i \left(\sum_{\sigma \in \mathcal{M}(i,S)} \hat{p}_\sigma \right)$. Below are the boxplots of our experimental result. The x-axis is label of assortment (We use "LRO 250" to denote the optimized assortment regarding the set of 250 ranks in LRO formulation), and the y-axis is expected revenue per customer.

Note 2. *The performance of solution increases as the rank size increases, yet the performance becomes stable when is higher enough*

This observation illustrates that the we need enough ranks to represent the true population.

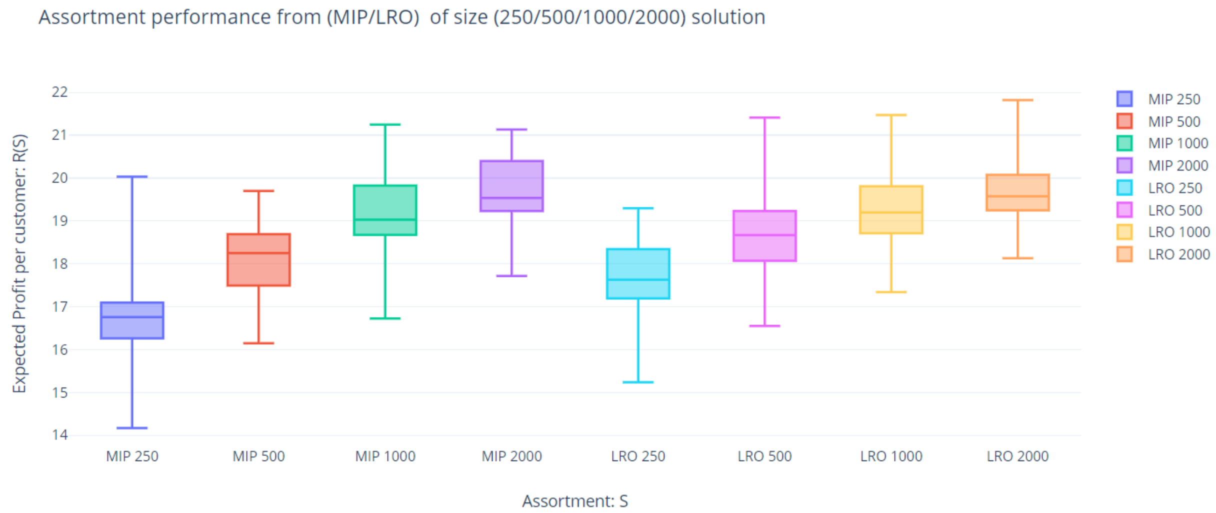

Figure 1: Test result of MIP and LRO formualtion

Intuitively, there will be some of the purchase behavior $\mathbf{U}_i$ that there is only a few or no rank that satisfies when we don't sample enough ranks to adapt it. After the rank size reaches to a considerate level, the marginal improvement regarding the representation power diminishes. Hence, in real practice we may leverage the computational cost and predication accuracy by selecting appropriate rank size (1500-2000 in this case).

Note 3. *LRO dominates MIP formulation at low rank size. The improvement is mere at high rank size.*

From the experimental results, the robust formulation can mitigate the under-representation issue at low rank size. We provide an intuitive explanation here: Consider an item i with relative high profit. When the low rank size is low, it is possible that there are only one rank (denote it as σ_i) that rank this item at the highest or second highest position. In the update step, when a purchase on that item is observed, the EM algorithm will update σ_i with the other many ranks equally, where the other ranks refer to the ones whose highest few preferred item is not available but item i is in the next preferred position. By the equal update under Bayesian setting, this causes an under estimation regarding the popularity of item i. In MIP formulation, optimizer will neglect item i by selecting some less profitable but more popular items. Nevertheless, the robust approach will consider the distribution where p_{σ_i} is high and optimize from this "worst case" which counter the problem of under-representation.

4.3 Heuristic method for the MIP programming

Theoretically, we can approximate the utlity model asymptotically by increasing rank size that the MIP solution will converge to the true population. If it is too computational costly to solve the LRO formulation, one would rather switch solve the MIP formulation of higher ranks instead. This argument does not stand for nothing. Mosek version 9, released on Fall 2019, is the first and the only commercial solver that support modeling exponential cone with MIP constraint. The newly released version is not well optimized in exponential cone MIP as compared to the linear MIP in solvers (CPlex, Gurobi and Mosek itself).

Here we provide two heuristic methods to solve the LRO formulation. The first method is the naïve linear relaxation on $\mathbf{x} \in \{0,1\}^n$ to $\bar{\mathbf{x}} \in [0,1]^n$. The problem becomes a standard convex optimization problem in convex set, which can be solved efficiently by interior point methods. To derive a solution for assortment design, we round $\bar{\mathbf{x}}$ to $\{0,1\}^n$ by setting 0.5 as threshold.

Linear relaxation is a crude heuristic to the problem. To refine the result, we consider taking the solution of η_1 and η_2 and solve the MIP subproblem by fixing the value of η_1 and η_2. The problem becomes

$$
\begin{aligned}
\max_{x,y} \quad & \sum_{k=1}^{K} \hat{m}_k t_k && (20)\\
\text{subject to} \quad & \left(\sum_{i=1}^{n} r_i y_i^k - \eta_2, 1, t_k\right) \in \mathcal{K}_{exp},\ \forall k \in [K] \\
& \sum_{i=0}^{n} y_i^k = 1,\ \forall k \in [K] \\
& y_i^k \le x_i\ \forall k \in [K],\ i \in [n] \\
& \sum_{j:\sigma^k(j)>\sigma^k(i)} y_j^k \le 1 - x_i,\ \forall k \in [K],\ i \in [n] \\
& \sum_{j:\sigma^k(j)>\sigma^k(0)} y_j^k = 0,\ \forall k \in [K] \\
& x \in \{0,1\}^n,\ \mathbf{e}^{\mathrm{T}} x \le c,\ \mathbf{y} \ge 0
\end{aligned}
$$

To see the motivation, kindly check the formulation (13) where we add latent $g_k = \eta_1$ since Mosek does not allow overlapping variable η_1 in different cones. Besides, the objective function is now independent of η_1 and η_2. Fixing the two Lagrangian coefficient simplifies the Branch and Bound procedure implemented inside solver. We use "L1" denotes the naive linear relaxation method, "L2" denotes the method of solving the sub-problem with fixed η_1, η_2

One may notice that the result of MIP and LRO formulation is slightly different from Figure 1. Indeed, in each generation we generate a set of 1000 ranks with updated probability /counts and compute the expected profits under the all different assortment. Below is the computational cost summary.

Formulation	250 ranks	500 ranks	1000 ranks	2000 ranks
MIP	10.72	21.64	45.73	105.92
LRO	134.76	1093.14	4967.89	35621.48
L1	5.05	10.34	20.55	50.89
L2	15.24	334.16	1987.51	9783.20

Table 4: Computational cost (unit: second)

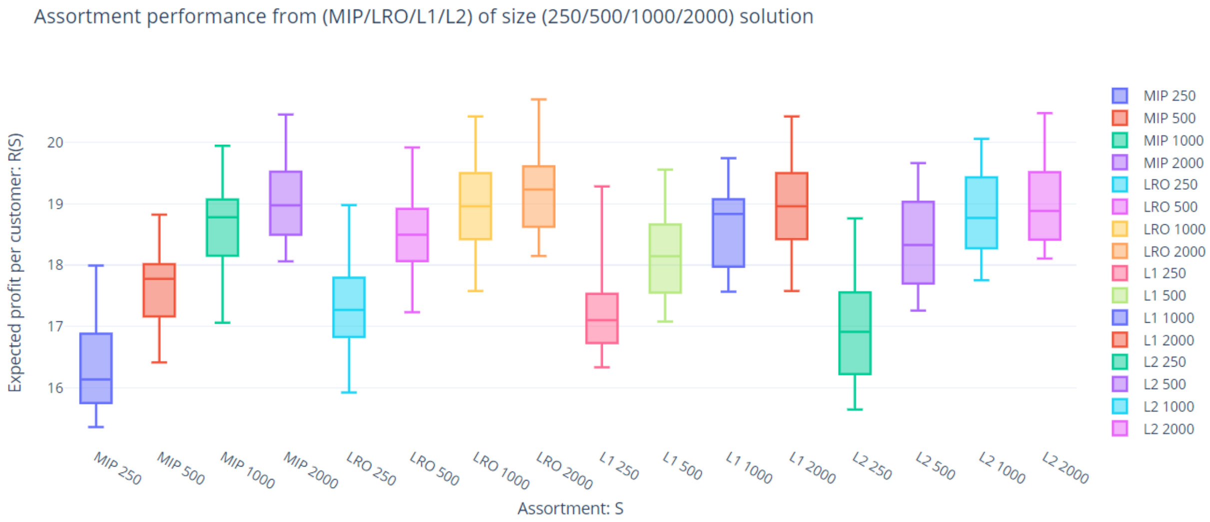

Figure 2: Test results of heuristic methods

Once again, we are making the trade off between computational cost and optimality. When the number of ranks is enough (K = 2000 for this problem), the marginal optimality increment is diminishing to zero such that is better to employ LRO formulation if we wish to improve further. However, if the computation is costly, then L1 heuristic will be the prior choice since it is the fastest and also better than MIP formulation.

5 Conclusion

We study the rank based non-parametric assortment optimization. Specifically, we emphasize on the issues that challenges the performance rank-based optimization framework: (1) Limit on using the features information other than transaction data (2) vulnerable to partial/noisy data. To resolve the first issue, we develop a pre-training, rank sampling and updating algorithm that utilize the feature information and further refine the result in the updating step. We prove that RB model is equivalent to RUM model and bound on sampled ranks to the underlying truth. Our experimental results shows that even when we are using the not matching pre-training RUM model, our SU algorithm gives a more accurate prediction with less computation cost as compared to the column generation method. For the second issue, We introduce the LRO framework to model the uncertainty in estimating the distribution of ranks. We provide exponential cone MILP formulation that is supported by the latest Mosek solver. We also provide a decomposition formulation and demonstrate the formulation is amendable to large scale instances. We conduct experiments on real retail data and show that LRO formulation significant under the low rank case and slightly better in the high rank case.