Recommendations under Multi-Product Purchase Behavior

Theja Tulabandhula
tt@theja.org
University of Illinois Chicago

Deeksha Sinha
deeksha@mit.edu
Massachusetts Institute of Technology

Prasoon Patidar
prasoonpatidar@gmail.com
Indian Institute of Technology, Delhi

ABSTRACT

We model consumer behavior wherein users select multiple items from the offered recommendations. We then formulate and solve an optimization problem that computes the best recommendation sets that the e-commerce platform should display to maximize its expected revenues. Numerical simulations show the scalability of our approach in real-world settings.

CCS CONCEPTS

• **Applied computing** → **Online shopping**; *Consumer products.*

KEYWORDS

choice models, combinatorial optimization, e-commerce

ACM Reference Format:
Theja Tulabandhula, Deeksha Sinha, and Prasoon Patidar. 2020. Recommendations under Multi-Product Purchase Behavior. In *Companion Proceedings of the Web Conference 2020 (WWW '20 Companion), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3366424.3382714

1 INTRODUCTION

In both online and offline shopping, consumers typically purchase multiple products. Given a sufficiently expressive choice model that captures this behavior, it is a priori unclear how to optimize for the recommendations (i.e., product assortments) that the platform should show/carry. The ability to both capture a rich enough choice behavior of each consumer as well as be able to display real-time optimized recommendations to them can have a tremendous impact on the user experience and hence the bottomline of the online shopping platforms. In this work, we extend the popular multinomial choice model (MNL) to the setting where the customer purchases one or two items (or none) when they are shown a collection of recommendations. We arrive at a parsimonious multi-product choice model, called the Bundle multivariate logit model (BundleMVL), inspired by variations proposed in the marketing literature [3]. We validate this model on two real world retail datasets. Next, we develop exact and iterative optimization schemes for computing revenue optimal recommendations given the BundleMVL behavior, while balancing their computational time and memory requirements.

2 THE BUNDLEMVL CHOICE MODEL

A bundle of products consists of one or more different products but with at most one unit of each product. Let $W = \{1, 2, \cdots n\}$ denote the set of products that a platform can offer. Product i has revenue $r_i > 0$ and the products are ordered such that $r_1 \geq r_2 \cdots \geq r_n$. We will refer to the collection of products recommended/offered to the customer by C, and the bundle of unique products chosen by the customer by S. If the customer does not purchase anything, then $S = \phi$, the empty set.

Under BundleMVL, the utility of purchasing a bundle of products S (including $S = \phi$) is given by $U_S = \bar{U}_S + \epsilon_S$, where $\bar{U}_S$ is the mean utility and ϵ_s is an additive customer-specific error that is Gumbel distributed. The customer chooses that bundle S which provides the maximum utility to her. Thus, the probability of purchasing a bundle S when recommendation set C is offered is P[select $S|C$ is offered] $\propto V_S$, where $V_S = e^{\bar{U}_S}$. The expected revenue that the platform can obtain if it offers recommendations C is given by: $R(C) = \sum_{S\subset C} V_S r_S / \left(V_\phi + \sum_{S\subset C} V_S\right)$, where $r_S = \sum_{i\in S} r_i$. To be consistent with prior work (especially on single-choice models), we will denote V_ϕ as v_0 in the rest of the paper.

While the size of S can be as large as C, empirically we observe a non-negligible decrease in the proportion of users who purchase three product bundles or higher. Taking this into account, we restrict our attention to modeling purchase behavior assuming at most two products can be purchased, i.e., $|S| \leq 2$. In this case, the expected revenue function can be written more explicitly as:

$$R(C) = \frac{\sum_{i\in W}\sum_{j\in W} \hat{r}_{ij}\theta_{ij}x_i^C x_j^C}{v_0 + \sum_{i\in W}\sum_{j\in W}\theta_{ij}x_i^C x_j^C},$$

where $x_i^C = 1$ if $i \in C$, $\theta_{ij} = \begin{cases} \frac{V_{\{ij\}}}{2} & i \neq j \\ V_{\{i\}} & i = j \end{cases}$, and $\hat{r}_{ij} = \begin{cases} r_i + r_j & i \neq j \\ r_i & i = j \end{cases}$. A detailed evaluation of limiting the bundle size to two will be considered in future work.

3 ALGORITHMS FOR OPTIMIZATION

We aim to find a recommendation set S^* that has the maximum expected revenue among all feasible recommendation sets $\mathcal{S} = 2^W$. This problem can be formulated as:

$$\max_{C\in\mathcal{S}} \frac{\sum_{i\in C}\sum_{j\in C}\hat{r}_{ij}\theta_{ij}x_i^C x_j^C}{v_0 + \sum_{i\in C}\sum_{j\in C}\theta_{ij}x_i^C x_j^C}, \text{ such that } x_i^C \in \{0,1\}\ \forall i \in C. \tag{1}$$

We derive some interesting structural properties of the optimal solution S^*, that guides the iterative algorithms that we design:

Proposition 1. *1. For all products i which are not in the optimal recommendation set S^*, $r_i \leq R(S^*)$. Equivalently, $S_u^* \subseteq S^*$, where $S_u^* = \{i : r_i > R(S^*)\}$.*
2. Let S^* be an optimal recommendation set of the smallest cardinality. For every $i \in S^*$, $\exists\, j(i) \in S^*$, where $j(i) \neq i$ and $r_i + r_{j(i)} > R(S^*)$.
3. Let the i-th revenue-ordered recommendation set be defined as

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW '20 Companion, April 20–24, 2020, Taipei, Taiwan
© 2020 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-7024-0/20/04.
https://doi.org/10.1145/3366424.3382714

$A_i = \{1, 2, \cdots i\}, 1 \leq i \leq n$. Then, the revenue of revenue-ordered recommendation sets increases monotonically as long as the price of all the products in the revenue-ordered recommendation set is greater than $R(S^*)$. That is, $R(A_1) \geq R(A_2) \cdots \geq R(A_k)$ where $r_k > R(S^*) \geq r_{k+1}$.

We propose three classes of algorithms for solving problem (1): a) mixed integer program formulation, b) binary search approach, and c) greedy algorithm. The iterative nature of the second and third algorithms helps in terminating the search for the optimal recommendations gracefully, especially under stringent timing requirements in online applications. Due to space constraints we focus on describing the one which shows the best time performance - the binary search based algorithm.

A binary search based efficient algorithm for a single-choice model was first described in [4]. Similar to that work, we also devise a binary search outer loop and focus on improving the computational speed of the comparison steps. In each iteration of the search process, we narrow the size of the interval in which $R(S^*)$ lies as outlined in Alg. 1. The computationally expensive comparison step (line 7) checks if the optimal revenue is greater than the specified threshold K_j. This calculation can be transformed as follows: $\max_{C \in \mathcal{S}} R(C) \geq K \iff \max_{C \in \mathcal{S}} \sum_{i \in W} \sum_{j \in W} \theta_{ij} x_i^C x_j^C (\hat{r}_{ij} - K) \geq K v_0$.

In particular, the optimization problem on the left hand side of the transformed comparison is a Quadratic Unconstrained Binary Optimization (QUBO) problem. Though the QUBO problem is NP-Hard, there has been ample research in heuristic approaches that return high quality solutions in reasonable computation times [2]. For each QUBO instance, we run multiple heuristics in parallel in our experiments to reduce the chance of getting an incorrect answer while using heuristic approaches. We also make use of all the *structural properties* of the optimal solution stated above to make the binary search more efficient.

Algorithm 1 BinarySearchQUBO

Require: Revenues $\{r_i\}_{i=1}^n$, BundleMVL parameters $\{\theta_{ij}\}_{i=1,j=1}^n$, tolerance level $\epsilon > 0$

1: $U_1 = r_1 + r_2, j = 1, i = 1$ and $S^* = \{1\}$.
2: **while** $r_{i+1} \geq r_i$ **do**
3: Increment i by 1.
4: $L_1 = r_{i+1}$.
5: **while** $U_j - L_j > \epsilon$ **do**
6: $K_j = (L_j + U_j)/2$.
7: **if** $K_j \leq \max_{C \in \mathcal{S}} R(C)$ **then**
8: $L_{j+1} = K, U_{j+1} = U_j$.
9: $\overline{B} = \{i : r_i > U\}, \underline{B} = \{i : r_i + r_1 < L\}$
10: Pick any $S^* \in \{C \in 2^W : R(C) \geq K_j, \overline{B} \subset C, \underline{B} \cap C = \phi\}$.
11: **else**
12: $L_{j+1} = L_j, U_{j+1} = K$.
13: Increment j by 1.
14: **return** S^*

4 EXPERIMENTS

We perform two sets of experiments: a) validation of the empirical fit of BundleMVL compared to the MMC model [1] (a closely related multi-choice model) on real data, and b) benchmarking the solution quality and computational times of the various optimization approaches and infer their suitability in web-scale recommendation settings.

Dataset	Bakery Dataset			Kosarak Dataset		
Model	**#param**	**train_ll**	**test_ll**	**#param**	**train_ll**	**test_ll**
MNL model	50	-91140	-22736	2621	-1317416	-331857
MMC model (0 corrections)	50	-90977	-22691	2621	-1404888	-353659
MMC model (1% corrections)	62	-77674	-19289	2812	-1389932	-350060
MMC model (5% corrections)	110	-77596	-19303	3580	-1386661	-349553
MMC model (20% corrections)	292	-77451	-19373	6460	-1379610	-349239
MMC model (50% corrections)	655	-77214	-19431	12220	-1358675	-350582
MMC model (100% corrections)	1261	-76791	-19571	21819	-1326125	-436080
BundleMVL model	1261	**-76791**	**-19281**	21733	**-1325751**	**-294304**

Table 1: Log likelihood values (BundleMVL has better fits) under different models of the Bakery and Kosarak datasets.

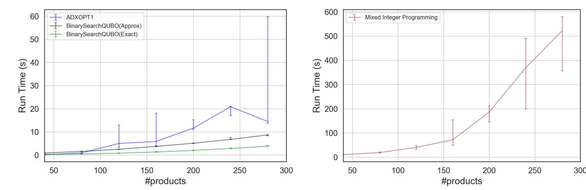

Figure 1: Runtime analysis of optimization algorithms

We compare the out of sample performance of BundleMVL against MNL and MMC models in the setting where the recommendations are the same across multiple customer interactions. The BundleMVL model provides the best fit model (highest log likelihood value) when compared to MMC models with different levels of corrections. The gap between the BundleMVL and MMC models when compared to the single choice model (MNL) is quite large ($1.5 - 2\times$ lower log likelihoods for the latter), validating the necessity for such multi-choice models. Likelihood values are in Table 1. The columns 'train_ll', 'test_ll' and '#parameters' contain the train and test log likelihood values and the number of non-zero parameters in each model.

To benchmark computation times of different algorithms, we synthetically generate BundleMVL model instances. The iterative algorithms perform optimally in terms of revenue i.e. are able to find an optimal assortment (Figure 1, we already know that the integer programming formulation gives the optimal solution). Most importantly, the performance of binary search approach shows that it can be used for web-scale personalized recommendations in near-real-time (<2 secs for 100 products, 10^4 parameters, which can be further reduced via reducing the number of iterations of the binary search at the expense of being slightly sub-optimal).

REFERENCES

[1] Austin R Benson, Ravi Kumar, and Andrew Tomkins. 2018. A discrete choice model for subset selection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 37–45.
[2] Iain Dunning, Swati Gupta, and John Silberholz. 2018. What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO. *INFORMS Journal on Computing* 30, 3 (2018), 608–624.
[3] Vishal P Singh, Karsten T Hansen, and Sachin Gupta. 2005. Modeling preferences for common attributes in multicategory brand choice. *Journal of Marketing Research* 42, 2 (2005), 195–209.
[4] Deeksha Sinha and Theja Tulabandhula. 2017. Optimizing revenue over data-driven assortments. *arXiv preprint arXiv:1708.05510* (2017).