Appendix A: Model: Choice Probabilities, Relation to MNL and Estimation

A.1. Derivation of the Choice Probabilities for BundleMVL-K

We briefly discuss how the analytical form of the choice probability for the BundleMVL-K model shown in Equation 2 is obtained. If the consumer has made a decision for each of the other products, then they will purchase product i only if the conditional utility defined in Equation 1 exceeds the threshold value 0. The conditional probability of buying product $i \in C$ can be computed as:

$$P\left(X_i = 1 | \{X_j = x_j : j \in C, j \neq i\}\right) = \frac{\exp(\alpha_i + \sum_{j \in C, j \neq i} \beta_{ij} x_j)}{\exp(\alpha_i + \sum_{j \in C, j \neq i} \beta_{ij} x_j) + 1} \mathbb{1}\left\{\sum_{j \in C, j \neq i} x_j < K\right\}.$$

This form of the probability is due to the noise being Gumbel distributed. Also, note that the above probability is non-zero only when the number of products already purchased is strictly less than K. These conditional probabilities can be combined using Besag's characterization theorem (Besag 1974) to get a consistent joint probability distribution of purchase of bundles. Let ϕ be the empty bundle signifying a no-purchase event. As per Besag's theorem, for any $\mathbf{x} = (x_1, \cdots x_n)$ such that $P(\mathbf{x}) > 0$, we have $\frac{P(\mathbf{x})}{P(\phi)} = \prod_{i \in C} \frac{g^i(x_i)}{g^i(0)}$, where $g^i(1) + g^i(0) = 1$ and:

$$g^i(1) = \frac{\exp(\alpha_i + \sum_{j \in C, j < i} \beta_{ij} x_j)}{\exp(\alpha_i + \sum_{j \in C, j < i} \beta_{ij} x_j) + 1} \mathbb{1}\left\{\sum_{j \in C, j < i} x_j < K\right\}.$$

Thus, $\frac{P(\mathbf{x})}{P(\phi)} = \exp\left(\sum_{i \in C} \alpha_i + \sum_{i \in C} \sum_{j \in C, j < i} \beta_{ij} x_j\right)$ if $\mathbb{1}\{\sum_{j \in C, j < i} x_j < K\} = 1$ and 0 otherwise. Using the fact that the sum of probability of purchase over all bundles is one, we get the probability of purchase of a bundle S given C as: $P(S|C) = \frac{V_S}{1 + \sum_{S' \subseteq C, |S'| \leq K} V_{S'}}$, where $V_S = \exp\left(\sum_{i \in C} \alpha_i x_i + \sum_{i \in C} \sum_{j \in C, j < i} \beta_{ij} x_i x_j\right)$ $\forall$ $S \subseteq C$, $|S| \leq K$, and $x_j = 1$ if $j \in S$ and zero otherwise. Note that one can extend our model representation power by including parameters involving three or more products as well, because as long as we ensure that the conditional probability of purchasing a product does not depend on the order of previous purchases, Besag's theorem can still be used to derive an analogous multi-purchase model. To be consistent with the literature on single-choice models, we introduce another parameter v_0 corresponding to the no-purchase probability by scaling each V_S by $\frac{1}{v_0} V_S$. One can interpret this parameter as the result of comparing the conditional utilities to a non-zero threshold.

A.2. Comparison of the Likelihood Expressions of MNL and BundleMVL-2

The likelihood difference of MNL and BundleMVL-2 is from two sources: (1) β_{ij}s (the interactions between product pairs) and (2) the way a purchase with greater than two products is broken up into multiple observations involving a single purchase (for MNL) and at most two item purchases (for BundleMVL-2). Even in the case when β_{ij}s are all zero, the likelihood of observing a purchase is not the same under MNL and BundleMVL-2 because the denominator of the conditional probabilities will be different. Note that only when the probability of choosing a product pair is the same as the product of the probability of choosing each individually, the corresponding β_{ij} will be 0 (this can be rare). When the probability of choosing the product pair itself is 0 (e.g., completely unrelated products), the corresponding $\beta_{ij} = -\inf$ (this is common).

Next, we show that the likelihood of a purchase under BundleMVL-1 (which is MNL) is not the same as BundleMVL-2 when $\{\beta_{ij}\}$s are all zero via an illustrative example. Let (a,b,c,d) be the recommendation set and (a,b,c) be the bundle purchase. Under MNL, we consider this as three singleton choices/purchases given the same recommendation set. Thus, the likelihood of observing this example is as follows.

$$\begin{aligned}
P^{MNL}_{(a,b,c)|(a,b,c,d)} &= P(a|(a,b,c,d)) \times P(b|(a,b,c,d)) \times P(c|(a,b,c,d)),\\
&= \frac{v_a}{v_0+v_a+v_b+v_c+v_d} \times \frac{v_b}{v_0+v_a+v_b+v_c+v_d} \times \frac{v_c}{v_0+v_a+v_b+v_c+v_d},\\
&= \frac{v_a \times v_b \times v_c}{(v_0+v_a+v_b+v_c+v_d)^3}.
\end{aligned}$$

We next compute the likelihood of the same observation under BundleMVL-2, when $\beta_{i,j}$s are all zeros. Let $\exp(\alpha_i) = v_i$ for $i = \{a,b,c,d\}$ and $\exp(\alpha_i + \alpha_j) = v_{ij}$ for $i,j \in \{a,b,c,d\}$. Further, let

$$D = v_0 + v_a + v_b + v_c + v_d + v_{ab} + v_{ac} + v_{ad} + v_{bc} + v_{bd} + v_{cd}.$$

Since $K = 2$ for BundleMVL-2, we consider four likely generative processes for this observation. Given the recommendation set (a,b,c,d), the customer could have purchased: (i) a and (b,c) separately, or (ii) b and (a,c) separately, or (iii) c and (a,b) separately, or (iv) each of a, b and c separately. Lets represent each of these events using their case numbers (e.g., the first event is represented as (i)). Under each event, the choice probabilities are shown below:

$$\begin{aligned}
&P^{BundleMVL-2}_{(a)(b,c)|(a,b,c,d)} = \frac{v_a}{D} \times \frac{v_{bc}}{D} = \frac{v_a v_b v_c}{D^2}, \quad P^{BundleMVL-2}_{(b)(a,c)|(a,b,c,d)} = \frac{v_b}{D} \times \frac{v_{ac}}{D} = \frac{v_a v_b v_c}{D^2},\\
&P^{BundleMVL-2}_{(c)(a,b)|(a,b,c,d)} = \frac{v_c}{D} \times \frac{v_{ab}}{D} = \frac{v_a v_b v_c}{D^2}, \text{ and } P^{BundleMVL-2}_{(a)(b)(c)|(a,b,c,d)} = \frac{v_a}{D} \times \frac{v_b}{D} \times \frac{v_c}{D} = \frac{v_a v_b v_c}{D^3}.
\end{aligned}$$

They can be added together to get the likelihood of the observation under $BundleMVL$-2 as:

$$\begin{aligned}
P^{BundleMVL-2}_{(a,b,c)|(a,b,c,d)} = P(i) \times P^{BundleMVL-2}_{(a)(b,c)|(a,b,c,d)} + P(ii) \times P^{BundleMVL-2}_{(b)(a,c)|(a,b,c,d)}\\
+ P(iii) \times P^{BundleMVL-2}_{(c)(a,b)|(a,b,c,d)} + P(iv) \times P^{BundleMVL-2}_{(a)(b)(c)|(a,b,c,d)}.
\end{aligned}$$

Clearly, the above expression will not be the same as the likelihood of MNL.

A.3. Data Augmentation before MLE for the BundleMVL-K Model

Alg. 1 Dataset Pre-processing

Require: Purchased bundles $\tilde{S}_1, \cdots \tilde{S}_{\tilde{m}}$.

$\mathcal{S} \leftarrow [\ ]$, weights $\leftarrow [\ ]$, $l \leftarrow 1$.
while $l \le \tilde{m}$ **do**
  if $|\tilde{S}_l| \le K$ **then**
    $\mathcal{S}$.append($\tilde{S}_l$) & weights.append(1).
  else
    Find the set of partitions:
    $Q_l = \{(A_1, \cdots A_t)$ s.t. (1) $\cup_{j=1}^t A_j = \tilde{S}_i$,
    (2) $|A_j| \le K \ \forall\ 1 \le j \le t$; and
    (3) $A_1, A_2, \cdots A_t$ are pairwise disjoint$\}$.
    for each A_i in $(A_1, \cdots A_t)$ in Q_l **do**
      $\mathcal{S}$.append(A_i) & weights.append($\frac{1}{|Q_l|}$).
  $l \leftarrow l + 1$.
return $\mathcal{S}$, weights

Alg. 2 MIP Formulation for recommendation set optimization (BundleMVL-2 model)

$$\max \sum_{i \in W} \sum_{j \in W} \hat{r}_{ij} p_{ij}$$

$$\begin{aligned}
\text{s.t. } & p_{ij} \le x_{ij} \ \forall\ i, j \in W, \\
& p_{ij} \le \frac{V_{\{i,j\}}}{v_0} p_{00} \ \forall\ i, j \in W \\
& p_{ij} + \frac{V_{\{i,j\}}}{v_0}(1 - x_{ij}) \ge \frac{V_{\{i,j\}}}{v_0} p_{00} \ \forall\ i, j \in W \\
& x_{ii} + x_{jj} - 1 \le x_{ij} \le \min(x_{ii}, x_{jj}) \ \forall\ i, j \in W \\
& p_{00} + \sum_{i \in W} \sum_{j \in W} p_{ij} = 1 \\
& x_{ij} \in \{0, 1\} \ \forall\ i, j \ \in W \\
& p_{ij} \ge 0 \ \forall\ i, j \in W \\
& p_{00} \ge 0.
\end{aligned}$$

If we want to estimate a BundleMVL-K model where K is smaller than the size of some purchased bundles in the dataset, then we can pre-process these observations. In particular, we first partition the purchased bundle, which is of size larger than K, into subsets of size at most K, and augment each such subset as an additional observation. We consider all possible such partitions and assign them equal probability weights (see Algorithm 1). The MLE objective is also updated to incorporate these importance weights. In other words, the probability of purchasing bundle $\tilde{S}_l$ from offer set C_l when $\tilde{S}_l > K$ can be written as: $\text{Prob}(\tilde{S}_l | C_l) = \frac{1}{|Q_l|} \sum_{(A_1, \cdots A_t) \in Q_l} \prod_{k=1}^t \text{Prob}(A_k | C_l)$, where each set A_k satisfies $|A_k| \le K$ needed for the model. Note that in the generative process assumed for this pre-processing step, each partitioning of the original $\tilde{S}_l$ is assigned equal likelihood. This is a choice that was made to balance simplicity while addressing the model-dataset incompatibility issue. The choice of weights do not change the nature of tractability of estimation. For instance, when the offered set C_l is the same for each observation l, then the estimation problem remains convex even if the weights are assumed unequal. The estimates themselves would change. Changing the generative process to include latent variables could be another direction that we could take to improve on the current process, but we defer this to future work.

Appendix B: The Revenue Maximization Problem: Hardness and Structural Results

B.1. Hardness of Unconstrained Optimization for BundleMVL-2

The revenue functions $R_2(C)$ and $R_K(C)$ have a form similar to the expected revenue of recommendations under the MNL model. For the MNL model, it is known that the unconstrained revenue optimization problem can be solved in linear time as the optimal set is a revenue-ordered set, i.e., it only contains the l highest priced products for some $l \in \mathbb{Z}_+$. But for the BundleMVL-2 model, we assert that this does not hold.

The proof of Theorem 1 below follows from a reduction of the well-known MAXCUT problem. While the existence of a polynomial time approximation algorithm for problem (5) is an open question, we believe it is inapproximable because it is similar to the quadratic knapsack problem with additional constraints on the coefficients (that can be both positive and negative). And it is known that the quadratic knapsack problem (defined on an edge-series parallel graph) is hard to approximate.

Proof of Theorem 1. Consider the decision version of the unconstrained BundleMVL-2 optimization problem:

$$\max_{C \in 2^W} R_2(C) \geq \kappa \Longleftrightarrow \max_{C \in 2^W} \sum_{i \in W} \sum_{j \in W} \theta_{ij} x_i^C x_j^C (\hat{r}_{ij} - \kappa) \geq \kappa v_0 \qquad \text{(Compare-Step)}$$

We will show that this decision version of the revenue optimization problem under the BundleMVL-2 model is NP-complete by a reduction from Max-Cut to this problem. Without loss of generality, we can assume that revenues of all products is less than κ (if not, then these products will be in the recommendation set corresponding to the solution of the optimization problem). Consider a graph G with nodes $\{1, \cdots m\}$. We obtain a modified graph G' by removing all the self-loops in G. Let the adjacency matrix of G' be A'. Let $\mathbf{d} = (d_1, \cdots d_m)$ denote a m-dimensional vector with the i-th entry being the degree of node i in the graph G'. Consider the following $(m+1) \times (m+1)$ matrix $Q = \left(\begin{smallmatrix} 0 & \mathbf{d/2} \\ \mathbf{d/2} & -A' \end{smallmatrix}\right)$ with a generic entry $q_{i,j}$ (in the i-th row and the j-th column). We index the entries of this matrix starting from 0 and the nodes of the graph starting from 1. Hence, for $i > 0$, the i-th column of the Q matrix corresponds to the i-th node in the graph G'. Consider the optimization problem:

$$\arg\max_{C \in 2^W} \sum_{0 \leq i \leq m} \sum_{0 \leq j \leq m} q_{i,j} x_i^C x_j^C, \tag{7}$$

with solution C^*. This optimization problem is equivalent to the MAX-CUT problem on the graph G as shown below. Note that the only positive values $q_{i,j}$ are either in the first row or the first column, hence $x_0^{C^*} = 1$. Now,
$$\sum_{0\le i\le m}\sum_{0\le j\le m} q_{i,j}x_i^{C^*}x_j^{C^*} = 2\sum_{1\le j\le m} q_{0,j}x_j^{C^*} + 2\sum_{1\le i\le m}\sum_{1\le j\le m,\ j>i} q_{i,j}x_i^{C^*}x_j^{C^*}$$
$$= 2\sum_{j\in C^*}\frac{d_j}{2} - 2E_{G'}\left(\tilde{C}^*,\tilde{C}^*\right) \qquad = E_G\left(\tilde{C}^*,\{1,\cdots m\}\backslash\{\tilde{C}^*\}\right),$$
where $E_{G'}(C,C')$ represents the number of edges between the set of nodes C and C' in the graph G', and $\tilde{C}^* = C^*\backslash\{0\}$. The final expression equals the number of edges across the cut $\left(\tilde{C}^*,\{1,\cdots m\}\backslash\{\tilde{C}^*\}\right)$ in the graph G.

We can also see that problem (7) can be transformed into an equivalent COMPARE-STEP problem as follows: choose numbers $\kappa, r_0, r_1, \cdots r_m$ such that $r_0 > \kappa$ and $\kappa/2 > r_1 > r_2 \cdots r_m > 0$. Let $\theta_{ij} = \frac{q_{ij}}{r_i + r_j - \kappa},\ \ 0 \le i,j \le m$. Thus, the problem of finding the maximum cut on any graph can be transformed to the optimization problem in the COMPARE-STEP of a BundleMVL-2 optimization problem. Moreover, given a solution of the COMPARE-STEP optimization problem, the maximum cut of the corresponding graph is evident. Hence, the decision problem and subsequently the BundleMVL-2 optimization problem are NP-complete. ■

B.2. Hardness of Unconstrained Optimization for MMC

THEOREM 2. *[Hardness Result for MMC] The decision version of the unconstrained revenue optimization problem under the MMC model (with number of allowed purchases ≤ 2) is NP-complete.*

Proof of Theorem 2. The unconstrained revenue optimization under the MMC model is given by the following problem:

$$\arg\max_{C\in 2^W} z_1\sum_{i\in C} r_i\frac{V_{\{i\}}}{V_{\{0\}}+\sum_{k\in C}V_{\{k\}}} + z_2\sum_{i,j\in C, i<j}(r_i+r_j)\frac{V_{\{i,j\}}}{V_{\{0,0\}}+\sum_{k,l\in C}V_{\{k,l\}}}, \qquad \text{(2-PRODUCTS MMC AO)}$$

where z_1, z_2 are the probability of purchasing one and two products respectively such that $z_1 + z_2 = 1$. Although Benson et al. (2018) do not model the no purchase option, one way to incorporate the no-purchase option is to assign a utility to the no-purchase option for each size of the subset that can be chosen. The no-purchase option is then treated as an alternative which is always present irrespective of the recommendation set and is represented with the parameters $V_{\{0\}}$ and $V_{\{0,0\}}$. To establish the NP-completeness of 2-PRODUCTS MMC AO, we use the fact that revenue optimization under the mixture of two multinomial logits (2-CLASS LOGIT AO) is NP-complete (Rusmevichientong et al. 2010b). This optimization problem is:

$$\arg\max_{C\in 2^W} \alpha_1\sum_{i\in C} s_i\frac{v_i^1}{v_0^1+\sum_{j\in C}v_j^1} + \alpha_2\sum_{i\in C} s_i\frac{v_i^2}{v_0^2+\sum_{j\in C}v_j^2}, \qquad \text{(2-CLASS LOGIT AO)}$$

where the revenues of products are given by $s_1, \cdots, s_n$ with $s_i \in \mathbb{Z}_+ \ \forall \ i \in [n]$, the preference weights are $(v_0^g, v_1^g, \cdots v_n^g)$ with $v_i^g \in \mathbb{Z}_+ \forall \ i \in [n], \ g = 1, 2$, and the probability of a customer belonging to each of the mixture compoenents is (α_1, α_2) with $\alpha_g \in \mathbb{Q}_+, \ g = 1, 2$ and $\alpha_1 + \alpha_2 = 1$. We claim that there is a reduction from a 2-CLASS LOGIT AO instance to a 2-PRODUCTS MMC AO instance and prove this in two steps:

1. Transform an instance of 2-CLASS LOGIT AO to an instance of 2-PRODUCTS MMC AO: Given an instance of 2-CLASS LOGIT AO, we define an instance of 2-PRODUCTS MMC AO by including an additional (n+1)-th product, which we refer to as a *snowball*. The revenues of the products in the transformed instance are same as their original revenue and the snowball has zero revenue i.e. $r_i = s_i \ \forall i \in [n]$ and $r_{n+1} = 0$. The probability of purchasing one and two products is equal to the probability of belonging to each group i.e. $z_g = \alpha_g, g = 1, 2$. The preference weights in the transformed instance are given as:
   (a) $V_i = v_i^1 \ \forall \ i \in \{0, 1, \cdots, n\}$ and $V_{n+1} = 0$
   (b) $V_{\{i,j\}} = v_0^2$ if $i = 0, j = 0; v_i^2$ if $j = n+1, i \in [n]; v_j^2$ if $i = n+1, j \in [n]; 0$ else.
2. Given a solution S^* of the above instance of 2-PRODUCTS MMC AO obtain a solution for the original instance of 2-CLASS LOGIT AO: this can be done using Lemma 2.

Thus, any instance of the 2-CLASS LOGIT AO can be reduced to an instance of 2-PRODUCTS MMC AO, proving that 2-PRODUCTS MMC AO is also NP-complete. ■

LEMMA 2. *If S^* is the solution for the above instance of 2-PRODUCTS MMC AO, then $S^* \backslash \{n+1\}$ is optimal for the 2-CLASS LOGIT AO problem.*

Proof. It is easy to see that $R_{MMC}(S) \leq R_{MMC}(S \cup \{n+1\}) \ \forall \ S \in 2^W$. Thus, without loss of generality, $n+1 \in S^*$. We define $R_{MMNL}(S) = \alpha_1 \sum_{i \in S} s_i \frac{v_i^1}{v_0^1 + \sum_{j \in S} v_j^1} + \alpha_2 \sum_{i \in S} s_i \frac{v_i^2}{v_0^2 + \sum_{j \in S} v_j^2}$. Thus, with the parameters specified as above, $R_{MMNL}(S) = R_{MMC}(S \cup \{n+1\}), \ \forall \ S \in 2^W$. Assume $\hat{S} \neq S^* \backslash \{n+1\}$ is the solution for 2-CLASS LOGIT AO. Thus, $R_{MMC}(\hat{S} \cup \{n+1\}) = R_{MMNL}(\hat{S}) > R_{MMNL}(S^*) = R_{MMC}(S^*)$ contradicting the assumption that S^* is the solution to 2-PRODUCTS MMC AO. ■

B.3. Structural Properties of the Optimal Unconstrained Solution for BundleMVL-2

Under the setting $\mathcal{C} = 2^W$, i.e., for the unconstrained optimization problem (5), we prove the following structural properties satisfied by the optimal solution C^*:

LEMMA 3. *For all products $i \in W$ that are not in any optimal recommendation set C^*, $r_i \leq R_2(C^*)$. Equivalently, $C_u^* \subseteq C^*$, where $C_u^* = \{i : r_i > R_2(C^*)\}$.*

LEMMA 4. *Let C^* be optimal. For every $i \in C^*$, $\exists\ j(i) \in C^*$, where $j(i) \neq i$ and $r_i + r_{j(i)} \geq R_2(C^*)$.*

Remark: If C^* is an optimal recommendation set of the smallest cardinality, then $\forall\ i \in C^*$, $\exists\ j(i) \in C^*$, where $j(i) \neq i$ and $r_i + r_{j(i)} > R_2(C^*)$.

LEMMA 5. *Let the i-th revenue-ordered recommendation set be defined as $A_i = \{1, 2, \cdots i\}, i \in W$. 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 $R_2(C^*)$, i.e., $R_2(A_1) \leq R_2(A_2) \cdots \leq R_2(A_k)$ where $r_k > R_2(C^*) \geq r_{k+1}$.*

Lemma 3 says that if a product's revenue is greater than the optimal revenue, then it belongs to the optimal recommendation set. Lemma 4 suggests that a product that is in the optimal recommendation set has a corresponding pair that also belongs to the optimal set such that the sum of their revenues is greater than the expected revenue of the set. Finally, Lemma 5 suggests that the objective function has a partial monotonicity property. Overall, these three properties can help us narrow the search for the optimal recommendations. For instance, if an algorithm keeps an estimate of an upper bound on the optimal revenue, then this can help prune the search space based on Lemma 3. We start by decomposing the revenue function.

LEMMA 6. *For two sets C and C' such that $C \cap C'$ is the empty set, we have:*

$$R_2(C \cup C') = \alpha R_2(C) + (1-\alpha) T(C, C'), \tag{8}$$

where $\alpha = \frac{v_0 + \sum_{i \in C} \sum_{j \in C} \theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}}$ is a value between 0 and 1, and the function $T(C, C')$ is defined as $T(C, C') = \frac{\sum_{i \in C'} \sum_{j \in C'} \hat{r}_{ij}\theta_{ij} + 2\sum_{i \in C} \sum_{j \in C'} \hat{r}_{ij}\theta_{ij}}{\sum_{i \in C'} \sum_{j \in C'} \theta_{ij} + 2\sum_{i \in C} \sum_{j \in C'} \theta_{ij}}$.

Proof.

$$\begin{aligned}
R_2(C \cup C') &= \frac{\sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \hat{r}_{ij}\theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}} \\
&= \left(\frac{\sum_{i \in C} \sum_{j \in C} \hat{r}_{ij}\theta_{ij}}{v_0 + \sum_{i \in C} \sum_{j \in C} \theta_{ij}} \right) \left(\frac{v_0 + \sum_{i \in C} \sum_{j \in C} \theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}} \right) + \\
&\quad \left(\frac{\sum_{i \in C'} \sum_{j \in C'} \hat{r}_{ij}\theta_{ij} + 2\sum_{i \in C} \sum_{j \in C'} \hat{r}_{ij}\theta_{ij}}{\sum_{i \in C'} \sum_{j \in C'} \theta_{ij} + 2\sum_{i \in C} \sum_{j \in C'} \theta_{ij}} \right) \left(\frac{\sum_{i \in C'} \sum_{j \in C'} \theta_{ij} + 2\sum_{i \in C} \sum_{j \in C'} \theta_{ij}}{v_0 + \sum_{i \in C \cup C'} \sum_{j \in C \cup C'} \theta_{ij}} \right) \\
&= \alpha R_2(C) + (1-\alpha) T(C, C') \qquad \blacksquare
\end{aligned}$$

Given the above decomposition, the proofs of the Lemmas 3-5 are provided below.

Proof of Lemma 3. Let $i \notin C^*$ such that $r_i > R_2(C^*)$. We know, $R_2(C^* \cup i) = \alpha R_2(C^*) + (1-\alpha)T(C^*, \{i\})$ for some $0 \le \alpha \le 1$. As $r_i > R_2(C^*)$, we have $T(C^*, \{i\}) \ge r_i > R_2(C^*)$. As $R_2(C^* \cup i)$ is a convex combination of $R_2(C^*)$ and $T(C^*, \{i\})$, it is greater than $R_2(C^*)$, contradicting the optimality of the recommendation set C^*. ■

Proof of Lemma 4. Suppose $\exists\ i \in C^*$ such that $r_i + r_j < R_2(C^*)\ \forall\ j \in C^* \backslash i$. This implies that $T(C^* \backslash i, \{i\}) < R_2(C^*)$. We know that $R_2(C^*)$ is a convex combination of $R_2(C^* \backslash i)$ and $T(C^* \backslash i, \{i\})$. Thus, $R_2(C^* \backslash i) > R_2(C^*)$ contradicting the optimality of the recommendation set C^*. ■

Proof of Lemma 5. Using (8), we can decompose the revenue of the revenue-ordered recommendation set A_m as $R_2(A_m) = \alpha R_2(A_{m-1}) + (1-\alpha)T(A_{m-1}, \{m\})$, for some $0 \le \alpha \le 1$. Also, note that $T(A_{m-1}, \{m\}) \ge r_m$. Further, $r_m > R_2(C^*)$ for $m \le k$. Thus, $T(A_{m-1}, \{m\}) > R_2(C^*) \ge R_2(A_{m-1})$ for $m \le k$. But $R_2(A_m)$ is a convex combination of $T(A_{m-1}, \{m\})$ and $R_2(A_{m-1})$. Thus, $R_2(A_m) \ge R_2(A_{m-1})$ for $m \le k$. ■

Appendix C: Revenue Maximizing Recommendations: Additional Details

C.1. Binary Search with Efficient Comparisons

For any given tolerance $\epsilon > 0$, our initial algorithm (Alg. 3) gives an ϵ-optimal solution, i.e., a solution within ϵ of the optimal value. In each iteration of the search process, we narrow the size of the interval in which $R(C^*)$ lies as outlined in BinarySearchAO (Alg. 3). The upper bound on $R(C^*)$ is initialized as the maximum revenue possible from a bundle of two products i.e. $r_1 + r_2$. The optimal recommendation set is arbitrarily initialized as $\{1\}$ and is of relevance only when the optimal revenue is less than ϵ, in which case all recommendation sets have revenue within ϵ of the optimal recommendation set.

Using Structural Properties of C^*. BinarySearchAO can be made more efficient by using the properties of the optimal recommendation set derived earlier (Lemma 3-5). At the cost of additional pre-processing (which is small), we can start with a lower bound greater than 0 based on Lemma 5. Let l be the maximum index i such that $R(A_1) \le R(A_2) \cdots \le R(A_i)$. Then, from Lemma 5, we know that $l \ge k$ (see the definition of k in the Lemma). Thus, $r_k \ge R(C^*) \ge r_{k+1} \ge r_{l+1}$. Hence, at the beginning of the binary search, the lower bound L can be set as r_{l+1}. Lemmas 3 and 4 can be used to make the comparison step faster. In particular, when we have an upper bound U on $R(C^*)$, then using Lemma 3, we know that all products with revenue greater than U should belong to the optimal recommendation set. Similarly, with a lower bound L on the revenue of the optimal recommendation set, we know that all products i such that $r_i + r_1 < L$, cannot belong to the optimal recommendation set. These observations predetermine the fate of some products,

Alg. 3 BinarySearchAO

Require: Parameters $\{r_i\}_{i=1}^n$, $\{\theta_{ij}\}_{i=1,j=1}^n$, tolerance level $\epsilon > 0$, and feasible sets $\mathcal{C}$.

```
L_1 = 0, U_1 = r_1 + r_2, j = 1, and C* = {1}.
while U_j − L_j > ϵ do
    κ_j = (L_j + U_j)/2.
    if κ_j ≤ max_{C∈𝒞} R(C) then
        L_{j+1} = κ_j, U_{j+1} = U_j.
        Pick any C* ∈ {C ∈ 𝒞 : R(C) ≥ κ_j}.
    else
        L_{j+1} = L_j, U_{j+1} = κ_j.
    Increment j by 1.
return C*
```

Alg. 4 BinarySearchAO(Efficient)

Require: Parameters $\{r_i\}_{i=1}^n$, $\{\theta_{ij}\}_{i=1,j=1}^n$, tolerance level $\epsilon > 0$, and feasible sets $\mathcal{C}$.

```
U_1 = r_1 + r_2, j = 1, i = 1 and C* = {1}.
while r_{i+1} ≥ r_i do Increment i by 1.
L_1 = r_{i+1}.
while U_j − L_j > ϵ do
    κ_j = (L_j + U_j)/2.
    if κ_j ≤ max_{C∈𝒞} R(C) then
        L_{j+1} = κ_j, U_{j+1} = U_j.
        Pick a C* such that R(C*) ≥ κ_j, B̄ ⊂ C*, and B̲ ∩ C* = ϕ; where B̄ = {i : r_i > U}, and B̲ = {i : r_i + r_1 < L}.
    else
        L_{j+1} = L_j, U_{j+1} = κ_j.
    Increment j by 1.
return C*
```

reducing the problem size (sometimes significantly as seen in our experiments) in the comparison step (6). BinarySearchAO(Efficient) incorporates these properties as shown in Algorithm 4.

QUBO Heuristics and Noisy Binary Search. Though the QUBO problem is an NP-hard problem (Pardalos and Jha 1992) as discussed before, there has been ample research in heuristic algorithms that return high quality solutions in extremely reasonable computation times (Dunning et al. 2018). This makes it appealing to use these approximate algorithms in solving the Compare-Step. Nonetheless, solving problem (6) approximately can potentially lead to narrowing down on an incorrect interval in the binary search outer loop. We take two steps to alleviate this issue. Firstly, for each QUBO problem, we run multiple QUBO heuristics in parallel. The binary search interval will have an incorrect update only if all the heuristics lead to an incorrect answer for the COMPARE-STEP.

Secondly, we further robustify the binary search outer loop by using a noisy binary search variant (Burnashev and Zigangirov 1974). Here one maintains a distribution over the unknown $R(C^*)$, and each comparison (with the median of the current distribution) is used to obtain an updated distribution using Bayes rule. This prevents incorrect comparison step outcomes from easily misleading the search process. For any specified tolerance level and a probability value with which the given solution needs to lie within the tolerance level, the number of iterations required for the noisy binary search still stays logarithmic. We refer to this algorithm as NOISYBINARYSEARCHAO and its version which uses the structural properties as NOISYBINARYSEARCHAO(EFFICIENT) (we omit its description due to space constraints).

C.2. Optimization with Linear Constraints

Constraints on the feasible recommendation sets are common in practice. For example, there can be a *cardinality* constraint on the maximum number of products in a recommendation set due to webpage/screen size limits in the online e-commerce setting. Business rules and obligations such as ensuring sufficient representation from various sub-groups of products, or the requirement to maintain a precedence order among products can also be formulated as linear constraints (Davis et al. 2013, Tulabandhula et al. 2021). We can incorporate linear constraints in the following way. For a general linear constraint set $Dx = e$, the COMPARE-STEP becomes

$$\max_{\{x\in\{0,1\}^n : Dx=e\}} \sum_{i\in W}\sum_{j\in W} \theta_{ij} x_i x_j (\hat{r}_{ij} - \kappa) \geq \kappa v_0.$$

This is a quadratic binary optimization problem with constraints that can be relaxed to get

$$\max_{x\in\{0,1\}^n} \sum_{i\in W}\sum_{j\in W} \theta_{ij} x_i x_j (\hat{r}_{ij} - \kappa) + \lambda (Dx - e)'(Dx - e),$$

for a suitably large $\lambda < 0$. In this form, the aforementioned QUBO solvers can be used directly.

If we have an inequality constraint, then as long as the components of D and e are integral, a similar transformation can be done with an appropriate number of slack variables (Glover and Kochenberger 2018). For instance, suppose we want to ensure that the size of the recommendation set is at most $e \in \mathbb{Z}_+$, i.e., we have the constraint $\mathbf{1}'x \leq e$ (here, $\mathbf{1}$ is the vector of all ones). Then, using slack variables $s_1, ..., s_e$, we get the following readily solvable QUBO instance:

$$\max_{x\in\{0,1\}^n} \sum_{i\in W}\sum_{j\in W} \theta_{ij} x_i x_j (\hat{r}_{ij} - \kappa) + \lambda (\mathbf{1}'x + \sum_{i=1}^{e} s_i - e)'(\mathbf{1}'x + \sum_{i=1}^{e} s_i - e). \tag{9}$$

C.3. Guarantees for Benchmark Algorithms

REMARK 1. The MIP formulation builds on an earlier formulation for the *mixture of MNLs* model studied in Blanchet et al. (2016), and is described in Algorithm 2. Linear constraints can be incorporated into the formulation as well.

LEMMA 7. *Let $\widehat{C}$ be the solution returned by* ADXOPT2 *using the BundleMVL-2 model. Then, $\widehat{C} \supset C_u^*$, where $C_u^* = \{i : r_i > R_2(C^*)\}$.*

Proof of Lemma 7. Using the arguments in proof of Lemma 3 and the local optimality of $\widehat{C}$, we have $\widehat{C}_u \subset \widehat{C}$, where $\widehat{C}_u = \{i : r_i > R_2(\widehat{C})\}$. As $R_2(\widehat{C}) \le R_2(C^*)$, $C_u^* \subset \widehat{C}_u \subset \widehat{C}$. ■

REMARK 2. The (worst-case) time complexity of the ADXOPT2 algorithm is $O(n^7)$, which is prohibitive for medium to large instances as observed in our experiments.

ASSUMPTION 1. *Model parameters for value conscious customers, i.e., those who prefer cheaper product(s) but derive higher value (product of price and utility) from higher priced product(s), satisfy:*

(a) $V_{\{i\}} \le V_{\{j\}}$ *and* $V_{\{i,k\}} \le V_{\{j,k\}}$ $\forall i < j, i \ne k, j \ne k$ *and* $i, j, k \in W$

(b) $r_i V_{\{i\}} \ge r_j V_{\{j\}}$ *and* $(r_i + r_k) V_{\{i,k\}} \ge (r_j + r_k) V_{\{j,k\}}$ $\forall i < j, i \ne k, j \ne k$ *and* $i, j, k \in W$.

THEOREM 3. *For value conscious customers, the revenue-ordered heuristic produces an optimal recommendation set for the optimization problem* $\max_{|C| \le d} R_2(C)$ *for any $d > 0$.*

Proof of Theorem 3. Let C^* be an optimal recommendation set for the above problem. We will construct a revenue-ordered recommendation set which has revenue $R_2(C^*)$. If C^* is a revenue-ordered recommendation set, then the theorem is trivially true. Hence, we focus on the case when C^* is not a revenue-ordered recommendation set. Then, there exists products l, m such that $l \in C^*$ and $m \notin C^*$ for some $1 \le m < l \le n$. Let $\tilde{C} = \{m\} \cup C^* \setminus \{l\}$. Thus, we have

$$\begin{aligned} R_2(\tilde{C}) &= \frac{\sum_{i \in C^* \setminus \{l\}} r_i V_{\{i\}} + \sum_{i \in C^* \setminus \{l\}} \sum_{j \in C^* \setminus \{l\}, j > i} (r_i + r_j) V_{\{i,j\}} + r_m V_{\{m\}} + \sum_{i \in C^* \setminus \{l\}, i \ne m} (r_i + r_m) V_{\{i,m\}}}{v_0 + \sum_{i \in \tilde{C}} V_{\{i\}} + \sum_{i \in C^* \setminus \{l\}} \sum_{j \in C^* \setminus \{l\}, j > i} V_{\{i,j\}} + V_{\{m\}} + \sum_{i \in C^* \setminus \{l\}, i \ne m} V_{\{i,m\}}} \\ &\ge R_2(C^*). \end{aligned}$$

As C^* is an optimal recommendation set, $R_2(\tilde{C}) \le R_2(C^*)$. Hence, we conclude $R_2(\tilde{C}) = R_2(C^*)$. We can use the above argument repeatedly to construct a sequence of recommendation sets, each having revenue equal to $R_2(C^*)$ until a revenue-ordered recommendation set is obtained. ■

ASSUMPTION 2. *Model parameters satisfy:* $\max_{i,j\in W, i\neq j} V_{\{i,j\}} \leq \epsilon \min_{k\in W\cup\phi} V_{\{k\}}$.

THEOREM 4. *Under Assumption 2, the revenue-ordered heuristic satisfies:* $R_2(C^*_{revord}) \geq \frac{2-\epsilon|C^*_{MNL}|}{2+4\epsilon|C^*|} R_2(C^*)$, *where* $C^*_{revord} \in \arg\max_{C\in\{A_1,A_2\cdots A_n\}} R_2(C)$, *and* C^*_{MNL} *and* C^* *are the optimal solutions of the unconstrained problem under the MNL and the BundleMVL-2 models respectively.*

Prior to giving proof of Theorem 4, we define some notation and lemmas that will be useful. We define $R_{MNL}(C) = \frac{\sum_{i\in C} V_{\{i\}} r_i}{v_0 + \sum_{i\in C} V_{\{i\}}}$, $C^*_{MNL} \in \arg\max_{C\in\mathcal{C}} R_{MNL}(C)$, $C^*_{revord} \in \arg\max_{C\in\{A_1,A_2\cdots A_n\}} R_2(C)$ and $C^* \in \arg\max_{C\in\mathcal{C}} R(C)$. When $\mathcal{C} = 2^W$, we take C^*_{MNL} to be a revenue-ordered set.

LEMMA 8. *Under Assumption 2,* $R_{MNL}(C) \leq \frac{2}{2-\epsilon|C|} R_2(C)$.

Proof.
$$R_{MNL}(C) - R_2(C) \leq \frac{\left(\sum_{i,j\in C, j>i} V_{\{i,j\}}\right)\sum_{i\in C} V_{\{i\}} r_i}{\left(v_0 + \sum_{i\in C} V_{\{i\}}\right)\left(v_0 + \sum_{i\in C} V_{\{i\}} + \sum_{i,j\in C,j>i} V_{\{i,j\}}\right)}$$
$$\leq \frac{|C|(|C|-1)\,\epsilon\left(\min_{k\in W\cup\phi} V_{\{k\}}\right)\sum_{i\in C} V_{\{i\}} r_i}{2\left(v_0 + \sum_{i\in C} V_{\{i\}}\right)^2} \leq \frac{\epsilon|C|}{2} R_{MNL}(C). \quad \blacksquare$$

LEMMA 9. *Under Assumption 2,* $R_2(C) \leq (1 + 2\epsilon|C|)\, R_{MNL}(C)$.

Proof.
$$R_2(C) - R_{MNL}(C) \leq \frac{\left(v_0 + \sum_{i\in C} V_{\{i\}}\right)\sum_{i,j\in C,j>i} V_{\{i,j\}}(r_i + r_j)}{\left(v_0 + \sum_{i\in C} V_{\{i\}}\right)\left(v_0 + \sum_{i\in C} V_{\{i\}} + \sum_{i,j\in C,j>i} V_{\{i,j\}}\right)}$$
$$\leq \frac{2\epsilon|C|\sum_{i\in C} V_{\{i\}} r_i}{v_0 + \sum_{i\in C} V_{\{i\}}} = 2\epsilon|C| R_{MNL}(C). \quad \blacksquare$$

LEMMA 10. *Under Assumption 2,* $R_2(C^*_{MNL}) \geq \frac{2-\epsilon|C^*_{MNL}|}{2+4\epsilon|C^*|} R_2(C^*)$.

Proof.
$$R_2(C^*) - R_2(C^*_{MNL}) = (R_2(C^*) - R_{MNL}(C^*)) + (R_{MNL}(C^*) - R_{MNL}(C^*_{MNL})) + (R_{MNL}(C^*_{MNL})$$
$$-R_2(C^*_{MNL})) \leq 2\epsilon|C^*| R_{MNL}(C^*) + \epsilon\frac{|C^*_{MNL}|}{2} R_{MNL}(C^*_{MNL})$$
$$\leq \frac{4|C^*| + |C^*_{MNL}|}{2}\epsilon R_{MNL}(C^*_{MNL}) \leq \frac{4|C^*| + |C^*_{MNL}|}{2}\,\frac{2}{2-\epsilon|C^*_{MNL}|}\epsilon R_2(C^*_{MNL}).$$
The inequalities are obtained using Lemmas 8 and 9. $\blacksquare$

Proof of Theorem 4. As C^*_{MNL} is a revenue-ordered recommendation set, using Lemma 10, $R_2(C^*_{revord}) \geq R_2(C^*_{MNL}) \geq \frac{2-\epsilon|C^*_{MNL}|}{2+4\epsilon|C^*|} R_2(C^*) \geq \frac{2-\epsilon n}{2+4\epsilon n} R_2(C^*)$. $\blacksquare$

REMARK 3. Guarantees similar to the above can be obtained for BINARYSEARCHAO(EFFICIENT) if we can obtain an approximation guarantee for the COMPARE-STEP under the same assumptions. We omit this analysis here due to space constraints.

Purchase bundle size	Bakery	Instacart	Kosarak	LastFM Genres	Walmart Items	Yoochoose Items	Ta Feng	UCI Online Retail
1	4.8%	4.9%	15.4%	37.8%	21.3%	0.0%	14.9%	22.6%
2	18.1%	5.8%	20.0%	15.6%	17.8%	2.0%	12.7%	6.2%
3	32.9%	6.4%	17.6%	9.0%	10.8%	7.8%	11.2%	4.2%
4	22.7%	6.9%	12.1%	6.1%	8.1%	13.7%	9.3%	3.1%
5	11.5%	7.1%	7.4%	4.5%	6.1%	22.4%	8.0%	3.0%
6	5.1%	7.1%	4.6%	3.5%	4.8%	0.4%	6.8%	2.6%
7	2.9%	6.8%	3.0%	2.8%	3.9%	1.3%	5.6%	2.5%
8	2.0%	6.3%	2.2%	2.4%	3.2%	2.0%	4.7%	2.4%
9		5.7%	1.8%	2.0%	2.7%	3.7%	3.9%	2.4%
10		5.1%	1.5%	1.7%	2.3%	4.4%	3.3%	2.1%
11		4.6%	1.2%	1.4%	2.0%	11.6%	2.8%	2.2%
12		4.1%	1.0%	1.3%	1.8%	0.4%	2.4%	1.9%
13		3.6%	0.9%	1.1%	1.6%	0.7%	2.0%	2.0%
14		3.2%	0.8%	1.0%	1.3%	1.1%	1.7%	2.0%
15		2.9%	0.7%	0.8%	1.2%	1.8%	1.4%	2.1%
16		2.5%	0.6%	0.8%	1.1%	2.4%	1.2%	2.2%
17		2.2%	0.5%	0.7%	0.9%	7.1%	1.1%	1.8%
18		2.0%	0.5%	0.6%	0.8%	0.3%	0.9%	1.7%
19		1.7%	0.4%	0.5%	0.7%	0.4%	0.8%	1.9%
20		1.5%	0.4%	0.5%	0.7%	0.6%	0.7%	1.7%
>20		9.5%	7.2%	6.0%	6.8%	15.8%	4.7%	29.3%

Table 10 **Summary of transaction sizes across the 8 real-world datasets.**

Heuristic	LAGUNA2009HCE	BURER2002	DUARTE2005	FESTA2002VNS
Number of Times Ranked First	22.3%	20.2%	15.9%	14.8 %

Table 11 **Performance of QUBO heuristics: fraction of times the top heursitics gave the best solution.**

Appendix D: Additional Experimental Details

Additional Data Descriptive Statistics. A summary of the fractions of product bundles purchased across datasets (including transactions of all sizes) is presented in Table 10.

Selecting QUBO heuristics for NoisyBinarySearchAO(Efficient). We consider ∼20 heuristics discussed in Dunning et al. (2018) (available at `https://github.com/MQLib/MQLib`), and solve several QUBO instances at the Compare-Steps from optimizing over multiple synthetically generated BundleMVL-2 models. Our results show that none of the heuristics consistently outperform others in terms of solution quality. Thus, we measure the performance of each heuristic in terms of the proportion of times it gave the best solution among all heuristics. Table 11 summarizes the performance of the top four heuristics ranked in this manner. Thus, for our experiments, we run these four heuristics in parallel at each comparison step of NoisyBinarySearchAO, and select the best solution to resolve the comparison.

Qualitative Results on the UCI Online Retail Dataset from Estimated BundleMVL-2. To complement the detailed quantitative analysis using Instacart dataset in Section 2.1, we summarize some qualitative insights using another dataset here. In particular, we take a brief look at the estimated parameters of BundleMVL-2 for the UCI dataset, which has (only) product names available. Based on our exploratory analysis, product pairs that have high β_{ij} values in the estimated BundleMVL-2 model include: (a) necklaces and earrings, (b) necklaces and bracelets, (c) placemats and coasters, and (d) wall art for gents and wall art for ladies, among others. These pairs of products are clearly complementary. The estimated BundleMVL-2 model thus, is able to learn the complementary relation between these pairs of products, and our optimization schemes would be able to use this information effectively for recommendation set optimization.

In addition to the product pairs which are clearly complementary, we also see other pairs of products having high β_{ij}s. For instance, necklaces in two different colors, key rings with two different letters etched, numbered tiles with consecutive numbers, and candle plates and candle holders, which don't seem complementary at a first glance but still end up having high β_{ij} values. Some possible reasons for simultaneous purchase of two such similar items could be: a) if a customer has strong affinity for a product, they might want to have multiple versions of it (such as in different colors), or b) customers might want to purchase two similar items with the intention of choosing one among the two later on and eventually returning one of the products. In either case, information about these less obvious product pairs is also valuable for the decision maker.

Empirical Fit of BundleMVL-2 on Additional Datasets. In Table 12, we report the likelihood fit of multiple choice models on six datasets, showing that it is extremely competitive with competing models. In the table, the MMNL model was estimated with 5 components, and the MMC model was estimated with varying proportions of correction sets. Noe that the estimation technique for MMNL failed to converge for some datasets due to the number of observations.

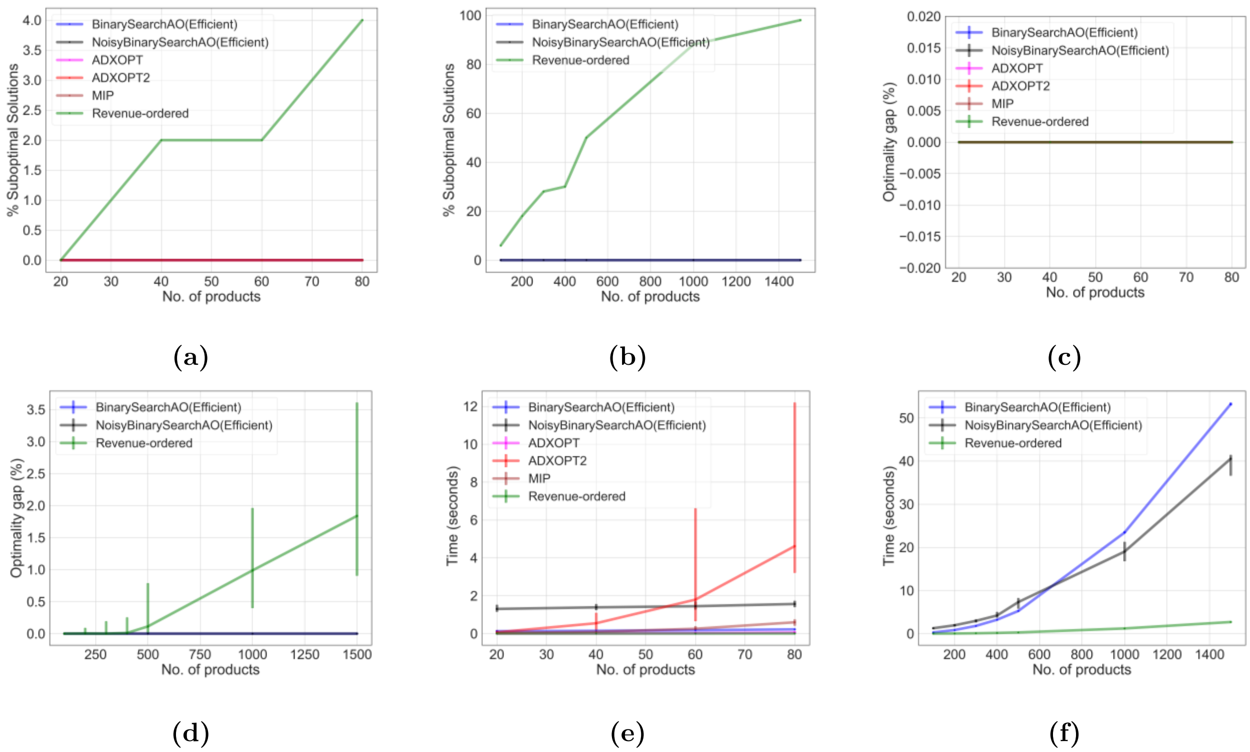


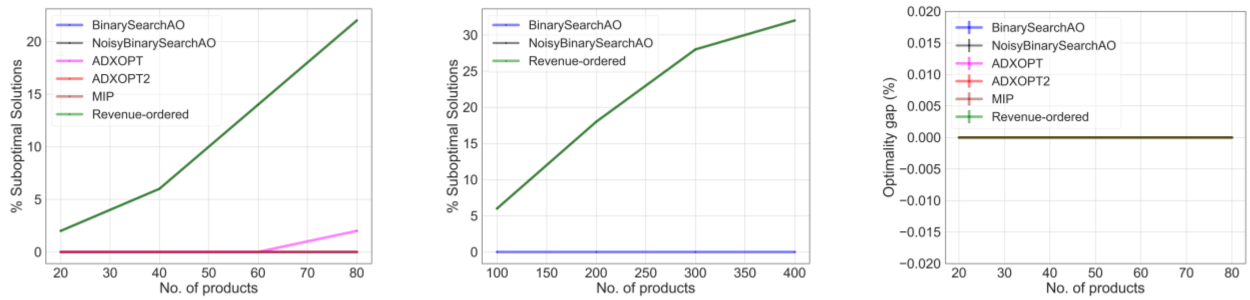
Figure 7 Optimality and run-time plots using the UCI dataset in the unconstrained setting for BundleMVL-2.

Dataset	Bakery Dataset			Kosarak Dataset			Walmart Items Dataset		
Model	**No. Params**	**Train LL**	**Test LL**	**No. Params**	**Train LL**	**Test LL**	**No. Params**	**Train LL**	**Test LL**
MNL	50	-91140	-22736	2621	-1317416	-331857	1075	-160299	-40526
MMNL (5)	255	-91362	-22785	13110	-1337913	-345109	5380	-158568	-40039
MMC (0%)	50	-90977	-22691	2621	-1404888	-353659	1075	-168250	-42494
MMC (1%)	62	-77674	-19289	2812	-1389932	-350060	1098	-142794	-35985
MMC (5%)	110	-77596	-19303	3580	-1386661	-349553	1194	-138402	-35017
MMC (20%)	292	-77451	-19373	6460	-1379610	-349239	1551	-135825	-34871
MMC (50%)	655	-77214	-19431	12220	-1358675	-350582	2265	-131954	-35018
MMC (100%)	1261	-76791	-19571	21819	-1326125	-436080	3456	-125531	-48916
BundleMVL-2	1261	-76791	**-19281**	21733	-1325751	**-294304**	3384	-125410	**-26518**
Dataset	**LastFM Genres Dataset**			**Yoochoose Items Dataset**			**Instacart Dataset**		
Model	**No. Params**	**Train LL**	**Test LL**	**No. Params**	**Train LL**	**Test LL**	**No. Params**	**Train LL**	**Test LL**
MNL	443	-1981969	-495887	52677	-256610	-64795	5981	-2668742	-670192
MMNL (5)	2220	-	-	263390	-	-	29910	-	-
MMC (0%)	443	-2138089	-534898	52679	-303703	-76847.8	5981	-2749875	-690323
MMC (1%)	529	-2128021	-532535	52985	-291047	-74024	6767	-2722321	-684258
MMC (5%)	873	-2112185	-528698	54209	-287511	-73371	9914	-2690143	-679500
MMC (20%)	2166	-2096532	-525428	58801	-286012	-73212	21714	-2624928	-676805
MMC (50%)	4750	-2086600	-524088	67985	-285872	-73462	45313	-2510259	-681712
MMC (100%)	9058	-2076535	-509677	83292	-283825	-73320	84646	-2346723	-1170856
BundleMVL-2	8871	-2009631	**-493188**	34776	-178693	**-38903**	84545	-2346233	**-393099**

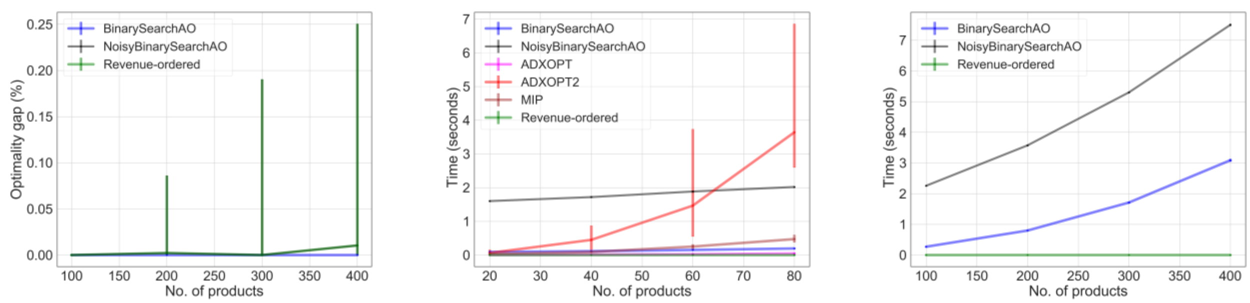
Table 12 Log-likelihood (LL) fits for different models on multiple real datasets.

Run-times for Computing BundleMVL-2 based Recommendation Sets using UCI. We report similar results for UCI, analogous to Section 4.2 in Figures 7 and 8. Similar results for synthetic datasets are omitted due to space constraints.

Computing MMC based Recommendation Sets. We report run-times using a novel mixed integer program (MIP) under the MMC model in the small products regime (Figure 9a), as neither constrained or

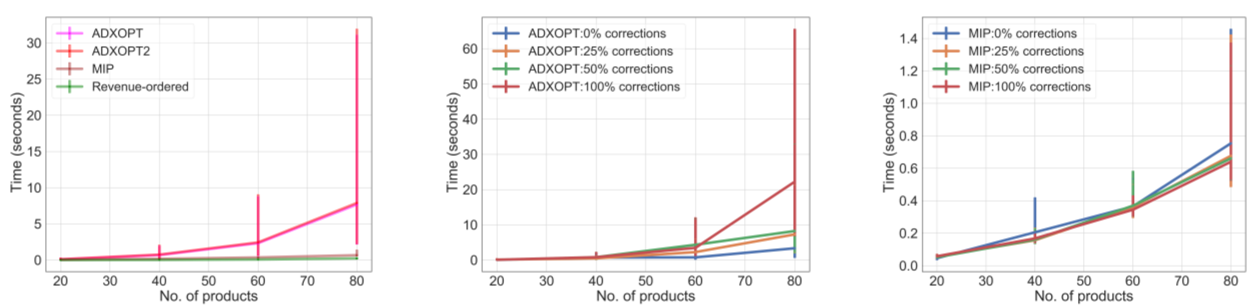


(a) **(b)** **(c)**



(d) **(e)** **(f)**

Figure 8 **Optimality and run-time plots using the UCI dataset in the constrained setting for BundleMVL-2.**



(a) **(b)** **(c)**

Figure 9 **Under the MMC model with synthetic data: (a) Run-times of algorithms, (b) & (c) Sensitivity of run-times to the number of correction sets for Adxopt and MIP.**

unconstrained optimization is addressed in Benson et al. (2018). We observe that Adxopt and Adxopt2 have similar ranges of run-times, thus they significantly overlap with each other. In Figures 9b & 9c, we observe that for a fixed number of products, the time taken by Adxopt increases as the number of correction sets (H-sets) increase, while it remains unaffected for the MIP. Overall, revenue-ordered heuristic is likely the only approach that can scale to much larger instances for this model.