Inventory-based Recommendation

Alibaba Group

Du Chen (chendu2017@sjtu.edu.cn)

Ying Rong

Xun Zhang

Huan Zheng

1. Introduction

2. Model

In our setting, we sequentially sell n products to customer with each product has its own revenue r_i. We assume there are NS segments in the whole selling horizon, and at the begining of segment s, we can observe inventory c_i^s for product i. Segment s contains T_s periods where each period has only one customer. For customer in the period t in the segment $s-1$, we could estimate cvr_{it} and ctr_{it} for customer to purchase product i, and then recommend no more than K products to he or she. After that, customer can purchase some products in the recommendation product set or leave with no-purchase. our objective is try to maximize total revenue in the whole selling horizon.

In this paper, we propose a inventory-based recommendation algorithm based on the dual method. Here, dual method means that we use real time data or historical data to build a deterministic model at the begining of segment. We solve its dual problem and obtain shadow price of each product i. Then, we use this shadow price to help us to recommend products in the next whole segment.

However, dual method does not consider the real time change in inventory, which plays an important role in this problem. In order to overcome this disadvantage, we add a inventory-based penalty function into the objective function, exerting influence of real time inventory on the recommendation.

We present two models and the difference between them is whether salvage value is taken into account. For the sake of presentation, our models are constructed based on real time data(last segment information), historical data also can be incorporate in the similar way. We first give model $[P1]$ without salvage value and its dual model $[D1]$.

$$[P1]\ \max \sum_{i=1}^{n}\sum_{t}^{T_{s-1}} cvr_{it}^{s-1} ctr_{it}^{s-1} r_i x_{it} \tag{1}$$

$$\text{s.t} \sum_{t=1}^{Ts} cvr_{it}^{s-1} ctr_{it}^{s-1} x_{it} \leq b_i, \forall i \tag{2}$$

$$\sum_{i=1}^{n} x_{it} \leq K, \forall t \tag{3}$$

$$0 \leq x_{it} \leq 1, \forall i, t$$

$$\begin{aligned}
[D1]\ \min\ & \sum_{i=1}^{n} b_i \alpha_i + \sum_{t=1}^{T_{s-1}} K\beta_t + \sum_{i=1}^{n}\sum_{t=1}^{T_{s-1}} \eta_{it} \\
\text{s.t}\ & \alpha_i cvr_{it}^{s-1} ctr_{it}^{s-1} + \beta_t + \eta_{it} \geq r_i cvr_{it}^{s-1} ctr_{it}^{s-1}, \forall i, t \\
& \alpha_i \geq 0, \forall i \\
& \beta_t \geq 0, \forall t \\
& \eta_{it} \geq 0, \forall i, t
\end{aligned}$$

Where decision variable x_{it} is the probability of providing product i in the period t, b_i is the proportion of inventory assigned to segment $s-1$. If we use real time information, $b_i = c_i^s/(NS - s+1)$; if We use historical data, for example, the customers in the same segment of the last selling horizon, we define c_i^{-s}, N_i^{-s} as the inventory of product i at the begining of segment s in the last selling horizon and the number of customers in the same segment in the last selling horizon, respectively, then $b_i = c_i^{-s} N_i^{-s}/(N_i^{-s} + ... + N_i^{-NS})$.

In the $[P1]$, the objective (1) is to maximize the total revenue in the last segment, constraint (2) means the sales in the last segment cannot exceed the assigned inventory, and constraint (3) forces us to provide no more than K products in each period. If we solve its dual model $[D1]$, then we obtain the dual solution and α_i is the shadow price for product i.

In the following, we build model $[P2]$ with salvage value and its dual model $[D2]$.

$$[P2]\ \max \sum_{i=1}^{n}\sum_{t}^{T_{s-1}} cvr_{it}^{s-1} ctr_{it}^{s-1} r_i x_{it} + \sum_{i=1}^{n} w_i I_i^0 \tag{4}$$

$$\text{s.t} \sum_{t=1}^{Ts} cvr_{it}^{s-1} ctr_{it}^{s-1} x_{it} + I_i^0 = b_i, \forall i \tag{5}$$

$$\sum_{i=1}^{n} x_{it} \leq K, \forall t \tag{6}$$

$$0 \leq x_{it} \leq 1, \forall i, t$$

$$s_i^0 \geq 0, \forall i$$

$$
\begin{aligned}
[D2]\ \min\ & \sum_{i=1}^{n} b_i\alpha_i + \sum_{t=1}^{T_{s-1}} K\beta_t + \sum_{i=1}^{n}\sum_{t=1}^{T_{s-1}} \eta_{it} \\
\text{s.t}\ & \alpha_i cvr_{it}^{s-1} ctr_{it}^{s-1} + \beta_t + \eta_{it} \geq r_i cvr_{it}^{s-1} ctr_{it}^{s-1}, \forall i, t \\
& \alpha_i \geq w_i, \forall i \\
& \beta_t \geq 0, \forall t \\
& \eta_{it} \geq 0, \forall i, t
\end{aligned}
$$

where I_i^0 is the remaining inventory of product i in the segment $s-1$ and w_i is the salvage value of product i.

In the $[P2]$, objective function (4) includes two parts, the first part is the sales reveneue, and the second part is the salvage revenue. Constraint (5) means that the total of sales quantity and remaining inventory equal to the assigned inventory, Constraint (6) forces us to provide no more than K products in each period. If we solve $[D2]$, we also can get the shadow price α_i for product i.

3. Algorithm

In this section, we propose several algorithms to approximately solve the aforementioned problem. The difficulty of getting a global optimum is three-fold. Firstly, *non-anticipatory*. Namely, only historical data and the information on the arriving customer are known to the decision maker. *Personalized*. Customers with distinctly different preference should be displayed dissimilar assortments. *Real-time*. All calculations are processed in real time, burdening the decision system practically. We propose three simple and straightforward but efficient and easy-to-implement algorithms for real-time personalized assortment optimization problems. These algorithms can be easily applied to problems with salvage values by slightly modefying the nominal value, and we omit details over this case here for the sake of simplicity. The first algorithm is a Customized **Inventory Balancing (IB)** algorithm, in which the nominal revenue of each product is balanced with the remaining inventory. The second algorithm is based on the linear programming (LP) problem, i.e. Problem $[P1]$. We further design bid price policy based on the shadow price obtained by solving $[P1]$. Since this algorithm heavily relies on the dual problem $[D1]$, we call the second algorithm as **Dual Approach**. The last algorithm skillfully combines two former algorithms, called **Primal-Dual Approach**.

3.1. Customized Inventory Balancing

Before diving into our proposed algorithm, we first give a brief introduction to **IB** algorithm. The **Inventory Blancing** algorithm is firstly introduced by Golrezaei et al. (2014). It takes into account both the revenue that would be obtained from the customer and the current inventory levels to decide which assortment to offer. An increasing penalty function $\Psi : [0,1] \to [0,1]$ with $\Psi(0) = 0, \Psi(1) = 1$ is defined to penalize lower inventory levels. Namely, the nominal revenue for each product is modified by the penalty funcion. And based on adjusted unit revenues, the personalized assortment is the one that maximizes the expected revenue. Mathematically, it can be described as (Golrezaei et al. (2014)):

upon the arrival of the customer in period t, offer an assortment S^t:

$$S^t = \arg\max_S \sum_{i \in S} \Psi(I_i^{t-1}/c_{i0}) r_i \phi_i^t(S) \tag{7}$$

where I_i^t denotes the remaining inventory of product i at the end of period t, c_{i0} represents the inventory level of product i at time $t = 0$, and $\phi_i^t(S)$ is the probability of purchasing product i for customer arrives at t. Note that, the purchase probability is based on offer set S. The independent variable of the penalty function in Eq.(7) can be explained as the ratio of real-time remaining inventory level to the initial inventory level. The value of penalty function decreases as the ratio decreases, which means a product with lower inventory level would be penalized more heavily, thus it is more unlikely to be offered.

The **IB** algorithm is easy-to-implement since before we start searching the personalized optimal assortment, it is only necessary to modify the revenue for each product. Practically, only the inventory level of sold product in the last period need to be updated. Suprisingly, the **IB** algorithm theoretically guarantees a satisfying competitive ratio (see Golrezaei et al. (2014)). For example, when $\Psi(x) = \frac{e}{e-1}(1 - e^{-x})$, the revenue obtained by **IB** is greater than $1 - \frac{1}{e}$ percentage of revenue obtained by a clairvoyant linear relaxation programming problem even in the worst case, no matter the arrival of customers is stationary or not.

Based on **IB** algorithm, we propose a customized version for our problem. Specifically, for each arriving customer in peroid t, after observe his or her prefrence (i.e. cvr and ctr), we provide the customer with an assortment $S^t := \{i | x_{it}^* = 1, i = 1, 2, \ldots, n\}$, where x_{it}^* is the optimal solution to the following optimization problem:

$$\max_{x_{it}: x_{it} \in \{0,1\}, \sum_i x_{it} \le K} \sum_{i=1}^{n} \Psi(I_i^{t-1}/c_{i0}) r_i cvr_{it} ctr_{it} x_{it} \tag{8}$$

The notable differences between Eq.(7) and Eq.(8) are purchase probabilities and the cardinality constraint. Instead of cosidering an assortment-based purchase probability like Eq.(7), we simplify customer choice model to an independent choice model, which is widely adopted on recommendation system by industrial community for the sake of computational efficiency (Zhu et al. (2018),Zhou et al. (2018)). As for the cardinality constraint, it makes our problem more appropriate, and this concern is also consisten with commercial scenarios (Feldman et al. (2018)).

Algorithm 1 Customized **IB**

for Segment s=1,2,. . . ,S **do**
  for Customer t in Period T_s **do**
    $score_{it} \leftarrow \Psi(I_i^{t-1}/c_{i0})r_i cvr_{it} ctr_{it}$ ▷ Expected Revenue
    Sort $\{score_{it}\}_{i=1}^{n}$ in descending order, obtain sorted rank $[i]$
    $S^t = \{i : [i] \leq K, I_i^{t-1} > 0\}$
  end for
end for

Formally, the customized **IB** algorithm is as following:

From Algorithm 1, it is clear that to find out the optimal assortment for arriving customer t, we only need to calculate the adjusted expected revenue $\Psi(I_i^{t-1}/c_{i0})r_i cvr_{it} ctr_{it}$ for each product, and select top K products. Although there is no solid theoretical proof verifying the superiority of model (8), our numerical study and large-scale field experiment show positive results.

3.2. Dual Approach

In the second algorithm, the problem in each time segment is formulated as an linear programming (LP) problem, i.e. model $[P1]$, which maximizes the revenue obtained in a single segment. Based on classical Duality Theory, we can easily get the shadow price for each constraint, and further these shadow prices can be interpreted as the marginal value of corrensponding resource. Comparing the value of resource to the expected revenue obtained by offering the product, decision makers can determine whether offer this product or not. This kind of analysis is widely utilized to allocate resource dynamically. However, to build a model like $[P1]$ requires much time to record customers' preference, which is usually referred as exploration. And then, in the following exploitation period, the shadow price is leveraged. Rusmevichientong et al. (2010) and Agrawal et al. (2014) both proposed exploration-then-exploitation algorithms to figure out assortment for each arriving customer. Based on their research, we further improve the LP-based dual approach by a sliding-window or prediction-based technique for our problem. We first formally describe our algorithm as Algorithm 2, and then explain the details.

It is clear that for each segment, a set of updated parameters $(cvr_{it}^s ctr_{it}^s, b_i^s)$ is incorporated to solve Model $[P1]$. The set of $cvr_{it}^s ctr_{it}^s$ is the potential preference of coming customers in the following segment s. Although the exact preferences are not known to the decision maker, they can be approximated by some methods. For example, under the assumption that customers in different segments are drawn independently from an identical (possiblely unknown) distribution, $cvr_{it}^s ctr_{it}^s$ can be replaced with the preference of customers who arrive during the last segment. Without the assupmtion, we could predict $cvr_{it}^s ctr_{it}^s$ based on historical customers' preferences who visit

Algorithm 2 Dual Approach

```
1: for s=1,2,...,S do
2:     Prepare parameters (cvr^s_it ctr^s_it, b^s_i)          ▷ Segment-specified parameters
3:     Solve [P1] with above parameters, or equivalently solve [D1]
4:     Get the shadow price α^s_i of constraint (2)
5:     for Customer t in Period T_s  do
6:         score_it ← (r_i − α^s_i) cvr^s_it ctr^s_it          ▷ Bid Price
7:         Sort {score_it}^n_{i=1} in descending order, obtain sorted rank [i]
8:         S^t = {i : [i] ≤ K, score_it ≥ 0, I^{t−1}_i > 0}
9:     end for
10: end for
```

the recommendation page in the same segment in former days. As for b_i^s, the inventory capacity for segment s, its value also depends on the assupmtion we make. Under the IID assupmtion, $b_i^s = \frac{c_i^s}{(NS-s+1)}$, otherwise, $b_i^s = \frac{c_i^{-s} N_i^{-s}}{(N_i^{-s}+....+N_i^{-NS})}$. In other words, we evenly distribute the remaining inventory into all remaining segments for the former case, while for the latter case, the number of allocated inventory is proportional to the history-based ratio of the number of customers who arrive in the same segment to the total remaining future customers.

The differences between Algorithm 2 and Algorithm 1 are readily discernible. Firstly, Algorithm 2 requires an additional LP problem to calculate Shadow Prices for each segment. The shadow prices will be updated at the beginning of every segment, on which the dynamics of inventory level reflects. However, shadow prices stay unchanged within the current segment, which means the $score_{it}$ would not be correspondingly modified in real time. Therefore, to some extent, whether the delayed updating in shadow price can capture most revenue highly depends on updating frequency. The second difference is the definition of $score_{it}$. In revenue management, Line 6 to Line 8 in Algorithm 2 are referred as dynamics bid-price control if there is no cardinality control. And the basic idea is that if the revenue earned exceeds the value of the resources consumed as measured by bid prices, then selling the product is worthy. Typically, the bid prices are computed as optimal dual prices, such as α_i^s. Therefore, only products with positive bid prices deserve to be sold, i.e. the selection rule in Algorithm 2 Line 8.

In short, the **Dual Approach** provides a new and unique perspective on dynamic recommendation, even though the updating only occurs at the beginning of segments. From a practical view, Algorithm 2 is efficient since commercial solvers, even open-source solvers, have the capability of solving LP quickly and accurately. From a financial view, it also provides a satisfying GMV or Conversion Rate as shown in the field experiment.

3.3. Primal-Dual Approach

The third algorithm combines two algorithms mentioned above. The fact that Algorithm 2 cannot conduct real-time optimization motivites the third method. We directly show the third algorithm as follows:

Algorithm 3 Primal-Dual Approach

1: **for** s=1,2,...,S **do**
2: Prepare parameters $(cvr_{it}^{s}ctr_{it}^{s}, b_i^{s})$ ▷ Segment-specified parameters
3: Solve **[P1]** with above parameters, or equivalently solve **[D1]**
4: Get the shadow price α_i^s of constraint (2)
5: **for** Customer t in Period T_s **do**
6: $score_{it} \leftarrow \Psi(\frac{I_i^{t-1}}{c_i^s}) \times (r_i - \alpha_i^s) cvr_{it}^s ctr_{it}^s$ ▷ Adjusted Bid Price
7: Sort $\{score_{it}\}_{i=1}^{n}$ in descending order, obtain sorted rank $[i]$
8: $S^t = \{i : [i] \leq K, score_{it} \geq 0, I_i^{t-1} > 0\}$
9: **end for**
10: **end for**

Algorithm 3 is almost the same as Algorithm 2, and the only change to Algorithm 2 lies in Line 6, where $score_{it}$ now depends also on $\Psi(I_i^{t-1}/c_{i0})$, the penalty function. Other procedures remain the same. Note that, the independent variable of the penalty function is $\frac{I_i^{t-1}}{c_i^s}$, whose denominator is the remaining inventory level at the start of segment s, instead of $\frac{I_i^{t-1}}{c_{i0}}$, whose denominator is the inventory level when $t = 0$. The reason for this modification is that since the LP model in each segment has taken inventory levels into consideration, incorporating $\frac{I_i^{t-1}}{c_{i0}}$, would penalize the consumed inventory twice. Regarding each segment as a brand-new stage and iteratively updating the initial inventory as c_i^s would eliminate the double-counted mistake. On the other hand, by incorporating real-time information on remaining inventory levels, now Algorithm 3 gain the ability to dynamically modify $score_{it}$ for each customer, thus, personalized recommendation becomes possible.

4.

References

Negin Golrezaei, Hamid Nazerzadeh, and Paat Rusmevichientong. Real-time optimization of personalized assortments. *Management Science*, 60(6):1532–1551, 2014.

Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1079–1088, 2018.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068, 2018.

Jacob Feldman, Dennis Zhang, Xiaofei Liu, and Nannan Zhang. Customer choice models versus machine learning: Finding optimal product displays on alibaba. 2018. Available at SSRN: http://dx.doi.org/10.2139/ssrn.3232059.

Paat Rusmevichientong, Zuo-Jun Max Shen, and David B Shmoys. Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations research*, 58(6):1666–1680, 2010.

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.