PARS: Peers-aware Recommender System

Huiqiang Mao
Alibaba Group
Hangzhou, China
huiqiangm@gmail.com

Yanzhi Li
City University of Hong Kong
Kowloon, Hong Kong
yanzhili@cityu.edu.hk

Chenliang Li
Wuhan University
Wuhan, China
cllee@whu.edu.cn

Di Chen
Alibaba Group
Hangzhou, China
jiuyao.cd@alibaba-inc.com

Xiaoqing Wang
Alibaba Group
Hangzhou, China
robin.wxq@alibaba-inc.com

Yuming Deng
Alibaba Group
Hangzhou, China
yuming.dym@alibaba-inc.com

ABSTRACT

The presence or absence of one item in a recommendation list will affect the demand for other items because customers are often willing to switch to other items if their most preferred items are not available. The cross-item influence, called "peers effect", has been largely ignored in the literature. In this paper, we develop a *peers-aware recommender system*, named PARS. We apply a ranking-based choice model to capture the cross-item influence and solve the resultant MaxMin problem with a decomposition algorithm. The MaxMin model solves for the recommendation decision in the meanwhile of estimating users' preferences towards the items, which yields high-quality recommendations robust to input data variation. Experimental results illustrate that PARS outperforms a few frequently used methods in practice. An online evaluation with a flash sales scenario at Taobao also shows that PARS delivers significant improvements in terms of both conversion rates and user value.

KEYWORDS

E-commerce, Recommender system, Ranking-based model, Demand substitution

ACM Reference Format:
Huiqiang Mao, Yanzhi Li, Chenliang Li, Di Chen, Xiaoqing Wang, and Yuming Deng. 2020. PARS: Peers-aware Recommender System. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3366423.3380013

1 INTRODUCTION

Recommender system is crucial for E-commerce sites to provide personalized services and to increase sales [13]. Current studies about recommender system focus improving the accuracy of individual recommendations with new technologies, such as content based, collaborative based and hybrid based algorithms [1, 2, 8, 12, 15]. However, peers effect, namely, the cross-item influence of items in a recommendation list has largely been ignored in E-commerce: *customers are often willing to purchase other substitutable items if their most preferred items are not available in the presented product assortment.* Put it in other words, the sales of one item may depend on the other items within the recommendation list. Peers effect has long

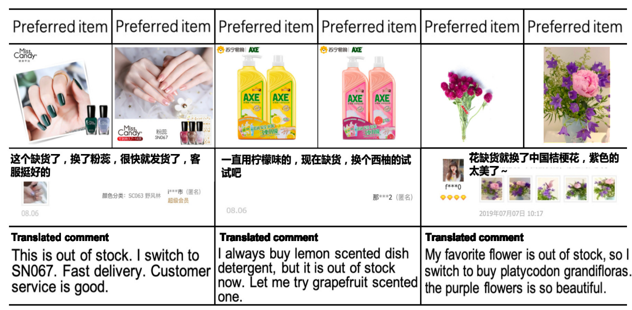


Figure 1: The examples of peers effect in TMALL.

been recognized by retailing practitioners. A survey by [7] reveals that more than 82% of shoppers would be willing to buy another size of the same brand or switch brands for the substitutable items, if their favorite brand-size was not available. A number of relevant studies [5, 18] also support a similar conclusion. A more recent study by Grocery Manufacturers Association (2015)[1] reports that in grocery retailing, about 70% of customers are willing to switch to substitutions if she/he finds a favorite item is not available. Figure 1 provides several examples of cross-item influence in Taobao[2], the largest online shopping platform in China. The first row shows the items that are most preferred by the customers and the corresponding items that were finally purchased. The comments left by these customers (last two rows) indeed confirm that they switch to the purchased item just because their most preferred ones are stockout.

In this paper, we develop a peers-aware recommender system (PARS) by incorporating the cross-item influence within the recommendation list. In a sense, peers effect can be considered as a type of *contextual influence*. Some attempts have been made to incorporate contextual information into the recommendation process, such as [1], [9] and [11]. These studies consider general contextual information, such as time, place and user social network. However, none considers the recommendation list itself as the context and the associated cross-item influence.

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380013

[1] Grocery Manufacturers Association. (2015). Solving the out-of-stock problem: A FMI/GMA trading partner alliance report. Retrieved from http://www.gmaonline.org/filemanager/About/ 15032FMIN_TPA_OutofStock_v51.pdf.
[2] https://www.taobao.com/

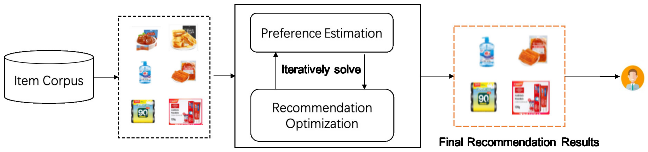


Figure 2: The workflow of peers-aware recommendation architecture.

Incorporating peers effect into recommendation decisions is a challenging issue. It is notoriously difficult to perform optimization over multidimensional problems because of the need to handle the curse of dimensionality. The combinatorial nature of cross-item influence based recommendation problem makes it inevitable to face this difficulty. To address this challenge, we propose to model user behavior with a ranking-based choice model, which is non-parametric, imposes few assumptions, and allows us to infer user's preferences from limited data [6]. Under this model, whether an item will be clicked or purchased by a user depends on whether more preferred items by the user are included in the recommendation list. Therefore, computing an item's click or purchase probability requires full knowledge of the other recommended items. Actually, peers effect is so crucial that we cannot define an item's click probability without knowing the full recommendation list.

We further develop a MaxMin model that estimates user preferences and optimizes recommendation decisions simultaneously. It learns the user's preferences from data with the sparsest fit criterion and search the optimal recommendations based on the learned preference model. To efficiently solve the MaxMin problem, we also design a decomposition algorithm. Such an approach provides a robust recommendation that is resistive to the perturbation of input data, as shown by our experimental studies. Specifically, Figure 2 shows the overall workflow of our proposed PARS. Firstly, a set of items is recalled from item corpus. These item candidates are fed into the model to generate final recommendations. In PARS, a preference estimation model is developed to learn customer preferences from historical user behaviors on displayed products. We describe customer preferences via a distribution over permutations of products. A product permutation (ranking) represents a specific customer segment. We aim to learn this distribution from the observed data and solve an optimization problem to derive recommendation results. The preference estimation procedure and a top N recommendation optimization are iteratively proceeded to yield final recommended products. To the best of our knowledge, this is the first work to incorporate the peers effect into recommender system. Experiments with synthetic and a real-world sales context confirm the necessity and effectiveness of considering the peers effect.

2 RELATED WORK

Majority of existing research on recommender system has been focused on improving the accuracy and diversification of recommendations, such as [8], [16], [12] and [15]. Context-aware recommender system is proposed because relevant contextual information, like time, place and peers within one institute, plays a great role in providing recommendations. [14] establishes the importance of location information on recommendations. [4] includes the user's emotional status. [1] discusses a general contextual information and employed three different algorithmic paradigms to incorporate this information into recommendation decisions. These works connect the user related contextual information while neglect the contextual influence of recommended items to the recommendation process. As an exception, [19] introduces a bundle recommendation problem which includes the dependency relationships between items into recommendation decisions. However, they assume that the mutual dependencies are exogenously given. This is different with what is considered in this paper. As a result, their methods cannot be applied in our context. Different from the above studies, [17] considers item dependencies in recommendations by introducing the decoy effects where the addition of another item increases the attractivity of the item already appearing in the recommendation set. Instead, this paper considers a substitution effects between recommended items.

3 PEERS-AWARE RECOMMENDER SYSTEM

Consider recommendation scenarios in E-commerce, such as personalized product recommendations in Tmall homepages. Customers make their choices from what they are offered and may substitute to a less appealing but still acceptable product if the most preferred is not presented. The recommendation decision in this context is to select a subset of products to offer to the arrived customer from a given list of relevant items. Formally, we define the number of items to recommend at one time as m and the catalog to select from contains N relevant items.

For any given user u, let $P(i|M, H_u)$ be the probability of choosing item i when recommendation list M and purchase history H_u are presented. Define binary variable x_i where $x_i = 1$ means item i is offered to the user and 0 otherwise. $P(i|M, H_u)$ thus can be rewritten as $P(i|x, H_u)$. For simplicity, we ignore the notation H_u and use $P(i|x)$ when no confusion arises. Denote the reward of item i as r_i. This reward can be interpreted as revenue, profit, clicks, etc. Given these notations, the recommendation problem is equivalent to the following optimization problem.

$$\max_{x \in \{0,1\}^N, |x|=m} \sum_i r_i P(i|x). \tag{1}$$

The key to recommendation problem is then to compute probability $P(i|x)$. Given the combinatorial nature of Equation (1), a naive mapping from relevant item sets to choice probabilities by machine learning techniques would suffer the curse of dimensionality. To resolve this issue and to model the cross-item influence, we employ a generic ranking-based model to formulate $P(i|x)$.

3.1 Ranking-based Choice Model

The ranking-based choice model captures the user preference by assuming that there exists a distribution over all permutations (or rankings) of the relevant items. A permutation (or ranking), denoted as σ, provides preference information of a certain customer segment on the item set. A user chooses item i if it has the highest ranking in the set of alternatives, i.e., $i = \arg\min_{j \in M} \sigma(j)$, where $\sigma(j)$ denotes the ranking of item j in permutation σ and M is the set of provided items. To denote the "outside" or "no-purchase" option, we introduce a virtual item (0th product) and label other items from number 1. We formally denote the distribution over rankings as λ which is a vector denoting the probability/percentage of each permutation (customer segment), i.e., $\lambda = (\lambda_k)_{k \in \{1,2,\cdots,N!\}}$. Then the probability $P(i|x)$ can be decoded as the sum of the permutation percentages λ_k which have item i as the highest ranking among the provided alternatives. Mathematically, we write

$$P(i|x) = P(i|M) = \sum_k \lambda_k y_i^k,$$

where y_i^k is a binary decision variable and item i is chosen by permutation k if $y_i^k = 1$ and 0 otherwise.

3.2 Peers-aware Recommendation Problem

The robust approach works as follows. Under all possible ranking models that are consistent with the observed data, we find the model that is the most robust, and we do so by choosing the one that minimizes its associated reward for a given recommendation list; in other words, we search for a conservative model that fits the data. Simultaneously, we find the recommendation that maximizes the conservative reward. Let $\mathcal{M}$ be the set of historical recommendation lists and S be the corresponding choice records associated with the recommendation lists. We formally present *Ranking Model based Recommendation Problem* (RMRP). The mathematical formulation is demonstrated by Equation (2) - (11).

We first explain notations of the inner minimization problem. $\mathbf{A}$ is a matrix that reflects the cross-item influence and models users' substitution behavior. Its columns and rows correspond to possible permutations and historical displayed items respectively. To incorporate the historical recommendation list information, we explicitly write out the recommendation lists. Specifically, $\mathbf{A} = (a_{s_{i,j},k}) \in \{0,1\}^{m \cdot |\mathcal{M}| \times K}, M_j \in \mathcal{M}, 1 \le k \le K, 1 \le j \le |\mathcal{M}|$ and $s_{i,j} = j \cdot m + i$. Here, $a_{s_{i,j},k} = 1$ if i-th item in recommendation list M_j is ranked the highest in permutation k, and $a_{s_{i,j},k} = 0$ otherwise. Then Equation (3) indicates that the plausible permutation distribution should be consistent with the user choice behavior implied by the observed data in terms of the possibilities of different types of permutations as well as the corresponding preference rankings. To further ease the understanding, let us consider a concrete example where two historical recommendation lists are shown as $M_1 = \{item_1, item_2\}$ and $M_2 = \{item_2, item_3\}$ and all permutations of these three items are given as $\{(item_1, item_2, item_3), (item_1, item_3, item_2), (item_2, item_1, item_3), (item_2, item_3, item_1), (item_3, item_2, item_1), (item_3, item_1, item_2)\}$. Matrix $\mathbf{A}$ then can be derived as Figure 3 shows. Recall that $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6)$ in this example. Equation (3) can be interpreted by that the sales of an item i (i = 1,2,3) comes from the customers corresponding to permutations that rank item i the highest within the recommendation list.

Regarding the outer maximization problem, as defined previously, y_i^k is a binary decision variable denoting that item i is chosen by permutation k if $y_i^k = 1$ and 0 otherwise. Constraint (5) denotes that each permutation only chooses one item (including item 0). Constraint (6) ensures that an item is chosen by the user only if it is included in the assortment. Constraints (7)-(8) ensure the definition of permutations; the former means that each permutation only chooses the highest-ranking item as the preference; the latter implies that an item is chosen only when it dominates the outside option. Constraint (9) specifies the size of a recommendation list. The last two constraints define that x_i and y_i^k are binary variables, while λ_k is a nonnegative continuous variable. Specifically, y_i^k is not explicitly shown as a binary variable by constraint (11) because it can be automatically ensured to be binary values by constraints (5)-(8).

Note that the inner minimization can also be understood as to discover the sparest model that is consistent with the observed data. According to [3], the minimization formulation is equivalent to finding a ranking model with the minimal support. This technique is also used by [6]. However, in the MaxMin model, ultimately we are maximizing the profit by optimizing over recommendation decisions. In other words, we are looking for a recommendation that gives us the maximum conservative profit.

$$\max_{\mathbf{x},\mathbf{Y}} \min_{\lambda} \quad \sum_{k=1}^{K} \sum_{i=0}^{N} r_i y_i^k \lambda_k \tag{2}$$

$$\text{s.t.} \quad \mathbf{A}\lambda = S \tag{3}$$

$$\sum_{k=1}^{K} \lambda_k = 1 \tag{4}$$

$$\sum_{i=1}^{N} y_i^k = 1, \forall k \tag{5}$$

$$y_i^k \le x_i, \forall i, k \tag{6}$$

$$\sum\nolimits_{j:\sigma^k(j) > \sigma^k(i)} y_j^k \le 1 - x_i, \forall i, k \tag{7}$$

$$\sum\nolimits_{j:\sigma^k(j) > \sigma^k(0)} y_j^k = 0, \forall k \tag{8}$$

$$\sum_{i=1}^{N} x_i = m, \tag{9}$$

$$x_i \in \{0,1\}, \forall i \tag{10}$$

$$y_i^k \ge 0,\ \lambda_k \ge 0, \forall i, k. \tag{11}$$

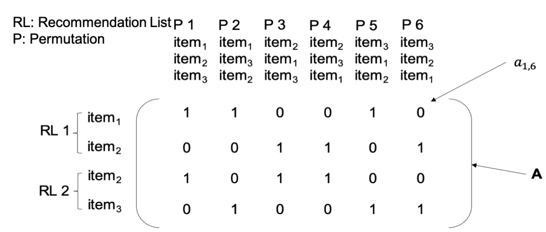


Figure 3: A toy example to show how matrix A is generated.

3.3 Algorithm

We have formulated our peers-aware recommendation problem with a ranking model RMRP. In this section, we discuss the properties of RMRP and develop efficient algorithms to solve it.

RMRP is a large-scale mixed integer program. Solving it faces two major challenges. First, the permutations are exponentially growing with the number of items. As a result, the distribution λ

has a huge number of possible elements, which makes it impossible to model it explicitly. Second, RMRP involves the integer decision variables $\mathbf{x}$ and $\mathbf{Y}$ with the same order of magnitude as λ. To address these challenges, we decompose this mixed integer program into two subproblems and solve the subproblems iteratively. Specifically, RMRP is decomposed into a preference estimation subproblem with continuous decision variable λ and a recommendation optimization subproblem with binary decision variable $\mathbf{x}$ and $\mathbf{Y}$.

3.4 Preference Estimation Subproblem

We first formulate the preference estimation problem as

$$\min_{\lambda} \quad \sum_{k=1}^{K}\sum_{i=0}^{N} r_i y_i^k \lambda_k \tag{12}$$

$$\text{s.t.} \quad A\lambda = S, \tag{13}$$

$$\sum_{k=1}^{K} \lambda_k = 1, \tag{14}$$

$$\lambda_k \geq 0, \quad \forall k \in \{1, 2, \cdots, K\}. \tag{15}$$

We solve this problem with column generation. We start the preference estimation problem with no permutation constraint and derive the corresponding optimal solution. Starting with this solution, we solve the *price problem* discussed below and identify a new permutation σ. If the optimal value of the price problem is negative, we update the constraint set by including the constraints associated with σ and solve the estimation problem under the updated constraints. If the optimal value of the price problem is nonnegative, we stop the iteration.

For efficient computation, the price problem can be approximately solved by a local search procedure. Specifically, we first randomly choose an initial permutation σ. Then, we obtain a neighboring permutation σ' by swapping the rankings of any two distinct products in σ. By evaluating the reduced cost of σ', we decide the next step. If σ' improves the reduced cost of σ, we take the one that gives the most improvement to the reduced cost and repeat this procedure to generate new candidates. Otherwise, we terminate with σ as the local optimal solution. Note that if this local optimal solution has a nonnegative reduced cost, we restart the procedure by randomly choosing a new permutation until a local optimal permutation with a negative reduced cost is found or the maximum number of repetitions is reached.

3.5 Recommendation Optimization Subproblem

Given the estimated permutation distribution, we consider the recommendation decisions in this subsection. We write the recommendation optimization subproblem as below.

Here, we write y_i^k as $\overline{y}_i^k$ to denote that the variable corresponds to the distribution $\widehat{\lambda}_k$. As the permutation distribution λ and the corresponding permutations σ have been provided in the estimation problem, the recommendation optimization problem becomes a standard integer program. In addition, $\overline{\mathbf{Y}}$ is automatically determined once $\mathbf{x}$ and $\widehat{\lambda}$ are derived. Therefore, this integer program only has N decision variables, i.e., $x_i, i = 1, 2, .., N$. The recommendation optimization problem can be efficiently solved via branch-and-bound. Recall that the vector $\mathbf{x}$ specifies the selected items for recommendation. We thus obtain the recommendation results by solving this optimization problem.

$$\max_{w, \mathbf{x}, \overline{\mathbf{Y}}} \quad w \tag{16}$$

$$\text{s.t.} \quad w \leq \sum_{k=1}^{K}\sum_{i=0}^{N} r_i \overline{y}_i^k \widehat{\lambda}_k, \tag{17}$$

$$\sum_{i=1}^{N} \overline{y}_i^k = 1, \forall k$$

$$\overline{y}_i^k \leq x_i, \forall i, k$$

$$\sum\nolimits_{j:\sigma^k(j) > \sigma^k(i)} \overline{y}_j^k \leq 1 - x_i, \forall i, k$$

$$\sum\nolimits_{j:\sigma^k(j) > \sigma^k(0)} \overline{y}_j^k = 0, \forall k$$

$$\sum_{i=1}^{N} x_i = m,$$

$$x_i \in \{0, 1\}, \forall i, \overline{y}_i^k \geq 0, \forall i, k.$$

3.6 Decomposition Algorithm

Now, we are ready to present the overall decomposition algorithm. As Algorithm 1 shows, we first solve the preference estimation problem with a random initial recommendation and obtain the optimal solution. Based on this estimated distribution, we construct a constraint regarding $\overline{y}_i^k$ as a decision variable and incorporate it into the recommendation optimization problem. After solving this optimization problem, we obtain the optimal recommendation for the moment. The algorithm repeats this procedure until convergence, i.e., until the difference between the upper and lower bounds is small enough. It is worth noting that more decision variables are incorporated into the recommendation optimization problem as more constraints are added. This is because the variables $\overline{y}_i^k$ for each constraint are associated with the current estimated distribution $\widehat{\lambda}_l^k$. These constraints reduce the feasible region of the problem and thus accelerate the search for the optimal solution.

4 EXPERIMENTS

In this section, we conduct extensive experiments with synthetic datasets and compare our method with some conventional methods. The results show that PARS delivers very robust performance. Another set of experiments with real-world data from a flash sales context at Taobao shows that PARS consistently outperforms the current technical alternatives. We further present a case study from this real-world Taobao scenario and demonstrate that the proposed PARS can successfully harness the peers effect for recommendation.

4.1 Experimental Setting

Evaluation Protocol. We utilize *revenue* $R(M)$ to evaluate the performance. It is calculated by

$$R(M) = \sum_{i \in M} p_i CVR_i, \; CVR_i = \frac{\#\ of\ purchases\ of\ item\ i}{\#\ of\ recommended\ of\ item\ i},$$

Algorithm 1: Decomposition algorithm

1 **Input**: Price vector P, tolerable error ϵ

2 **Output**: Recommendation M

1: Initialize low bound $LB = -\infty$ and upper bound $UB = +\infty$. Randomly choose an initial recommendation M_0.

2: Let $M = M_0$.

3: **while** $UB - LB > \epsilon$ **do**

4: Solve the preference estimation problem (12) with M and obtain the optimal solution $\widehat{\lambda}$.

5: Add the constraint $w \leq \sum_{k=1}^{K} \sum_{i=0}^{N} p_i \overline{y}_i^k \widehat{\lambda}_k$ into the recommendation optimization problem

6: Derive $\widehat{\mathbf{Y}}$ according to M and $\widehat{\lambda}$ and set $UB = \min\{UB, \sum_{k=1}^{K} \sum_{i=0}^{N} p_i \widehat{y}_i^k \widehat{\lambda}_k\}$

7: Solve the recommendation optimization problem under the added constraints and obtain the optimal recommendation M^* and the maximal revenue $w*$.

8: Let $LB = w^*$ and $M = M^*$.

9: **end while**

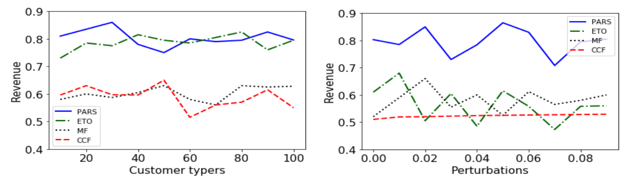
10: **return** The optimal recommendation M^*.

where M denotes the recommended list, p_i is the price of item i and CVR_i is the corresponding conversion rate. In experiments on synthetic dataset, CVR_i can be calculated by generative ranking-based model (customer preference model) because customers make decision according to a presumed manner. As for online experiment, it can be directly obtained from the cache data of customer behaviors. Although CVR_i is a direct index to reflect the relevance of recommended items with users' preferences, we propose the revenue as evaluation metric to show that our approach can satisfy users' preferences and, meanwhile, recommend more high-value products. To make it more clear, in the field experiment, we also report CVR of different approaches as a direct measure index.

Baseline. We compare PARS with the following conventional methods:

Matrix Factorization (MF) MF is a standard recommendation technique. It has been widely employed in recommendation scenarios due to its success in Netflix-prize competition [10]. The main idea of this method is $A_{m\times n} \approx P_{m\times k} \times Q^T_{n\times k}$. Here $P_{m\times k}$ can be interpreted as the extent of interest m users have on k latent factors and $Q_{n\times k}$ is interpreted as the extent to which n items possesses these k factors. They are derived by least squares or stochastic gradient descent algorithms for given data $A_{m\times n}$.

Coupled Collaborative Filtering (CCF) CCF incorporates contextual information into collaborative filtering. It is proposed by [9] to measure the contextual information and improve personal recommendation. The main idea of this approach is to obtain coupled similarity in terms of inter-item, intra-context and inter-context interactions. These similarities are integrated to measure the influence of contextual information on the rating prediction. For user u on item v in context c_i, the rating prediction is obtained as $P_{u,v,c_i} = \frac{\sum_{c_j \in N} r_{u,v,c_j} w_{c_j,c_i,v}}{\sum_{c_j \in N} |w_{c_i,c_j,v}|}$. Here $w_{c_j,c_i,v}$ denotes coupled similarity integrated-weight and r_{u,v,c_j} is observed rates from user u on item v in context c_j.



(a) With respect to $\overline{K}$

(b) With respect to perturbations of input data

Figure 4: The revenue under different methods.

Estimate-Then-Optimize (ETO) This method separately estimates the customer preference with input data and optimizes the recommendation decisions to derive the optimal recommendations.

Wide & Deep Model (WDM) WDM consists of a linear model component and a neural network component. They are jointly trained to combine the benefits of memorization and generation. We employ this model as a baseline in an online A/B test, where item specific features (e.g. GMV, price, brand and category) and user specific features (e.g. gender, demographics) are given to predict the probability of a user action on items. Provided a set of item candidates and recorded users, after training the model, we obtain top n item which have n highest scores as the final recommendation by running a forward inference pass over the WDM.

4.2 Experiments on Synthetic Dataset

A Generative Ranking-based Model. To complete the numerical experiments, we need a simulated environment wherein customers make purchase decisions based on the offered recommendation. We assume that there are K permutations, $\sigma_1, \sigma_2, \cdots, \sigma_K$. They are obtained by randomly generating permutations over N products with replacement. Any permutation k for $k = 1, 2, \cdots, K$ corresponds to a number that is uniformly drawn from the interval $[a, b]$. We normalize these numbers so that they sum up to 1. All other permutations $\sigma_i, i \notin \{1, 2, \cdots, K\}$ are assigned as $\lambda(\sigma_i) = 0$.

Result. Figure 4(a) shows the curves of revenue with respect to $\overline{K}$ under three different models. Here $\overline{K}$ is the number of positive λ_k denoting the degree of user heterogeneity. Generally, it is more difficult to learn user's preference from the limited data and make recommendation decisions when users have a higher degree of heterogeneity. Thus, this result shows the learning ability of the four models. Compared with matrix factorization method and coupled collaborative filtering, our robust approach and estimate-then-optimize model can achieve higher revenues. Table 1 shows the basic statistic of corresponding revenues.

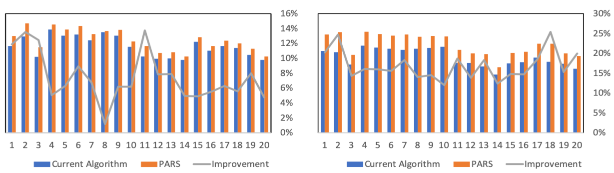
Figure 4(b) also shows the revenue curves of four models with respect to perturbations on input data. Perturbations are imposed by randomly introducing noises on observed data. Obviously, our approach is more robust than the other three methods in terms

Table 1: The revenue statistics under different methods.

Statistic	MF	CCF	ETO	PARS
Mean	0.5745	0.5819	0.8366	0.8395
Standard Deviation	0.015	0.038	0.024	0.021

Table 2: The revenue statistics with the input perturbation.

Statistic	MF	CCF	ETO	PARS
Mean	0.5840	0.5266	0.5821	0.8710
Standard Deviation	0.0354	0.0049	0.0590	0.0603



(a) Conversion Rate (b) User Value

Figure 5: The performance of algorithm in terms of conversion rate and user value.

of revenues. It is worth noting that, although the estimate-then-optimize model can achieve the similar amount of revenues with our robust approach as demonstrated in Figure 4(a), it is more vulnerable to fluctuation of input data. This defect makes it difficult to be employed in practical settings. Table 2 further validates this result.

4.3 Real-data Experiments

To test the necessity of incorporating peers effect and the effectiveness of peers-aware recommendation, we have conducted a set of online A/B test in a flash sales context on Taobao. There are 40 categories for sale with millions of page visits per day. The current solution is derived by recalling items from a WDM and reranking items with a rule-based heuristic. The rules include, for example, the prohibition of displaying items within the same category in a two consecutive days and an upper limit on the number of different categories displayed in a day. The selected items are then presented to customers in a personalized manner. We examine PARS by selecting items at a category-level through RMRP without changing the personalized manner of presentation.

The performance of PARS is evaluated with *Conversion Rate* and *User Value*, which are commonly used measurements in this scenario. Conversion rate is defined by the proportion of customers who have purchased products in a group who have clicked these items. User value is derived as the average Gross Merchandise Volume (GMV) generated by each user visit to this platform, which is positively correlated with revenue metric. These two indicators both measure the effectiveness of items selection for this context.

Figure 5 shows the performance of PARS compared with the current practice in this online platform over 20 days. The histogram respectively shows the conversion rate and user value of PARS and the current method, and the curve depicts the improvements of PARS over the existing method in these two measurements. For commercial confidentiality, we omit the absolute values and report the improvement as a percentage. Compared with the online baseline, our approach achieves average improvements of 7.4% on conversion rate and of 16.9% on user value. It indicates that, similar with the previous results, PARS demonstrates strong capability of learning user preferences and making better recommendation.

4.4 Case Study

We perform a case study in this section to further demonstrate that our approach indeed captures the peers effect in the recommendation, compared with the baseline method. We select two recommendation results of the competing approaches from the online experiment. In Figure 6, the first line shows recommended products of the current algorithm and the second line gives results of our PARS. The current method recommends products across a wide range of categories, while our algorithm focuses on kitchen/household accessories recommendations. It is not hard to see that customers are more likely to have substitutions in PARS recommendations as we make recommendations within a more focused scenario. Also, two models both recommend white crystal sugar to users, but PARS provides the one with high price. That is, the proposed PARS can effectively make more revenue for E-Commerce.

Figure 6: A case study from the online experiment.

5 SUMMARY

The peers effect has been widely recognized by traditional retailers but largely ignored in the online environment. We have proposed PARS, a peers-aware recommender system to capture the peers effect within a recommendation list. In general, the recommendation list itself can also be viewed as contextual information. However, this information is endogenized, which is different from the exogenous contextual information that has been considered in the literature. To implement PARS, we adopt a ranking-based choice model and simultaneously estimate the user preferences and optimize recommendation decisions with a MaxMin model. The MaxMin model leads to the nice robustness property, namely, the produced recommendation is not sensitive to the variations in the input data, which is common when the historical data is limited. This is the first paper that considers peers effect in recommendation. Comparisons with the widely used methods lend great support to the necessity and effectiveness of considering peers effect.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 217–253.

[2] Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12, 4 (2002), 331–370.

[3] Emmanuel J Candes and Terence Tao. 2005. Decoding by linear programming. *IEEE transactions on information theory* 51, 12 (2005), 4203–4215.

[4] Anind K Dey, Gregory D Abowd, and Daniel Salber. 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human–Computer Interaction* 16, 2-4 (2001), 97–166.

[5] Margaret A Emmelhainz, James R Stock, and Larry W Emmelhainz. 1991. Consumer responses to retail stock-outs. *Journal of retailing* 67, 2 (1991), 138–147.

[6] Vivek F Farias, Srikanth Jagabathula, and Devavrat Shah. 2013. A nonparametric approach to modeling choice with limited data. *Management science* 59, 2 (2013), 305–322.

[7] Food Marketing Institute. [n. d.]. Variety or duplication: a process to know where you stand. *The Research Department, Food Marketing Institute, Washington, DC* ([n. d.]).

[8] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[9] Xinxin Jiang, Wei Liu, Longbing Cao, and Guodong Long. 2015. Coupled Collaborative Filtering for Context-aware Recommendation.. In *AAAI*. 4172–4173.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[11] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.

[12] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

[13] J Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*. ACM, 158–166.

[14] Bill N Schilit and Marvin M Theimer. 1994. Disseminating active map information to mobile hosts. *IEEE network* 8, 5 (1994), 22–32.

[15] Shakila Shaikh, Sheetal Rathi, and Prachi Janrao. 2017. Recommendation system in E-commerce websites: A Graph Based Approached. In *Advance Computing Conference (IACC), 2017 IEEE 7th International*. IEEE, 931–934.

[16] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. 2009. Scalable collaborative filtering approaches for large recommender systems. *Journal of machine learning research* 10, Mar (2009), 623–656.

[17] Erich Christian Teppan and Alexander Felfernig. 2012. Minimization of decoy effects in recommender result sets. *Web Intelligence and Agent Systems: An International Journal* 10, 4 (2012), 385–395.

[18] Glen L Urban, Philip L Johnson, and John R Hauser. 1984. Testing competitive market structures. *Marketing Science* 3, 2 (1984), 83–112.

[19] Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. 2014. Bundle recommendation in ecommerce. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 657–666.