A Multistakeholder Recommender Systems Algorithm for Allocating Sponsored Recommendations*

Edward C. Malthouse
ecm@northwestern.edu
Northwestern University
Evanston, Illinois

Khadija Ali Vakeel*
khadija@northwestern.edu
Northwestern University
Evanston, Illinois

Yasaman Kamyab Hessary†
yasaman.kamyab@northwestern.edu
Northwestern University
Evanston, Illinois

Robin Burke
robin.burke@colorado.edu
University of Colorado
Boulder, Colorado

Morana Fudurić
mfuduric@efzg.hr
University of Zagreb
Zagreb, Croatia

ABSTRACT

Retailing and social media platforms recommend two types of items to their users: sponsored items that generate ad revenue and non-sponsored ones that do not. The platform selects sponsored items to maximize ad revenue, often through some form of programmatic auction, and non-sponsored items to maximize user utility with a recommender system (RS). We develop a multiobjective binary integer programming model to allocate sponsored recommendations considering a dual objective of maximizing ad revenue and user utility. We propose an algorithm to solve it in a computationally efficient way. Our method can be applied as a form of post processing to an existing RS, making it widely applicable. We apply the model to data from an online grocery retailer and show that user utility for the recommended items can be improved while reducing ad revenue by a small amount. This multiobjective approach, which unifies programmatic advertising and RS, opens a new frontier for advertising and RS research and we therefore provide an extended discussion of future research topics.

CCS CONCEPTS

• **Information systems → Data mining**; **Collaborative filtering**; **Computational advertising**.

KEYWORDS

recommender systems, multistakeholder, platforms, sponsored content, sponsored recommendations

1 INTRODUCTION

A two-sided market is an intermediary platform between two distinct stakeholder groups that provide each other with network benefits. The media have long been held up as an exemplar. Media companies attract an audience of consumers with, for example, news or entertainment content (e.g., a TV program or magazine), and then sell advertisers access to the audience. The two stakeholder groups are consumers and advertisers. In recent years, a wide variety of different types of two-sided markets have been introduced, often under the name *platform* rather than two-sided market [10]. Here are some examples of successful platforms:

- *Shopping platforms.* Taobao in China and companies like Shoprunner and Amazon Marketplace in the US are platforms that match consumers with retailers and merchants. Related shopping examples include Ebay and Etsy.
- *Online travel agencies* (OTAs) such as Orbitz, Expedia and Google match consumers with travel-related services (e.g., hotels, airlines, car rentals). AirBnB, VRBO and Uber are other successful platforms in the travel category.
- *Social media* such as Facebook and LinkedIn match consumers with user-generated content and advertisers.
- *Online retailers* often act as platforms, matching brands (advertisers) with customers.

This article studies recommendations on platforms with two stakeholder groups, *users* and *providers* (e.g., advertisers). The platform attracts users and matches them with providers through recommendations. We assume two types of provider recommendations, *sponsored* and *non-sponsored*, where sponsored providers (i.e., advertisers) pay the platform for access to the user and non-sponsored providers do not. For example, online retailing shopping platforms might show sponsored items from certain manufacturers, as well as items from manufacturers or vendors that the platform thinks the user would like.

Sponsored recommendations are often sold on programmatic auctions where the platform can adopt various protocols to sell slots [16, 20] to the highest bidder, e.g., second-price auctions where the winner pays the second-highest bid amount. Since non-sponsored recommendations, by definition, do not generate ad revenue, we assume that platforms select non-sponsored items to be relevant to the user. Later we discuss other factors that the platform may take into consideration such as one non-sponsored item could have a larger margin than another. Core assumptions of this paper are that sponsored recommendations are sold to maximize ad revenue to the platform, while non-sponsored recommendations are selected to provide utility to the user.

The focus of this article is on deciding which sponsored items, if any, to recommend to each user. Platforms must balance between recommending sponsored items with high ad revenue and high

Presented at the RMSE workshop held in conjunction with the 13th ACM Conference on Recommender Systems (RecSys), 2019, in Copenhagen, Denmark.

†Both authors contributed equally to this research.

user utility. We express the two objectives in a single equation:

$$Q = \gamma(\text{user utility}) + (1 - \gamma)(\text{ad revenue}), \tag{1}$$

where γ is a parameter that determines how much weight to give to each of the two objectives. A platform that allocates sponsored recommendations with an auction has $\gamma = 0$ and is maximizing ad revenue (depending on how much the advertiser knows about the active user and how good its predictive models are, ad bids will also reflect user utility). RS that consider only the user's utility such as those used to allocate non-sponsored recommendations have $\gamma = 1$. Rather than focusing on one or the other, this equation allows for a continuum between the two, competing objectives.

An idealized example clarifies this trade-off, which will be empirically investigated later in the paper. Suppose that advertiser A places the highest bid for a sponsored recommendation to some user. Advertiser B places a bid that is only slightly less than A's, but would provide the user with substantially more utility than A. It would seem reasonable for the platform to give the slot to B even though it would generate less ad revenue. There could be financial justifications for this decision as well, such as B may lead to higher lifetime value, where the user buys B, and ultimately is satisfied with it, and buys again from the platform into the future. If the retailer had shown A then either the user would not buy it, or would not be satisfied, and would be less likely to return in the future. Of course, an alternative scenario is also conceivable, where the ad revenue from B is substantially lower and/or B provides no advantage over A in user utility. In this second scenario the platform should display A.

This is an example of a multiobjective, multistakeholder recommendation system (MRS) [5]. The system (platform) must recommend one set of stakeholders (providers) to another group (users), while balancing the two objectives of generating ad revenue and creating utility for users. We develop the MRS problem specifically for sponsored recommendations, where a platform must select a personalized list of providers to recommend to each user.

We formulate a post-processing binary integer programming model to maximize the dual-objective function in equation (1) over all users, subject to constraints that the advertiser has budget to pay for the recommendation and the number of sponsored recommendations to any user has not exceeded a threshold. We show how to solve the optimization problem optimally in a computationally efficient way. We test it using real data from an online grocery store. With this approach, we make the following contributions. First, we integrate the fields of computational advertising with RS. Second, we show how to handle a situation where a platform is interspersing non-sponsored recommendations with sponsored ones. While sponsored recommendations guarantee the platform a constant stream of revenue, by balancing user utility and ad revenue, managers can generate higher customer lifetime value [12].

2 LITERATURE REVIEW

2.1 Programmatic Auctions Maximizing Ad Revenue to Platforms

Search and display advertising is often sold using programmatic auctions. An auction is held for an impression and different advertisers place bids. Auctions are typically second-price, where the winner pays the second-highest bid amount. Second-price auctions have an optimal strategy for an advertiser of bidding the expected value of the impression [7]. There is a rich literature on bidding strategies. For example, [18] presented a bid-optimization approach that combines several supervised learning algorithms, as well as second-price auction theory, to determine the correct price and ensure that the right message is delivered to the right person, at the right time. [23] proposed a way to leverage machine learning to predict the winning amount from the bidding history. It is important to note that in the above strategies, the platform is accepting the best bid and is therefore maximizing only its ad revenue. The platform does not consider the user's utility when it awards the exposure to an advertiser.

Our formulation of the MRS problem for sponsored recommendations is related to the AdWords problem [14], which considers "online" situations where user queries come in over time to some system and assumes (1) a set of bids from advertisers for a query; (2) known (or predicted) click-through rates (CTR) for each advertiser-query pair; (3) a budget for each advertiser; and (4) an upper bound for the number of ads to be displayed with each query. Algorithms solving the problem return search results that do not exceed the upper bound, and the advertisers must both bid on the search term and have budget to pay if the ad is clicked on.

2.2 Multistakeholder RS

As we mentioned earlier, traditional RS have an exclusive focus on one stakeholder, the user. In other words, traditional RS do not accommodate more complicated systems with multiple objectives and stakeholders. *Multistakeholder recommender system (MRS)* refers to a designs that incorporate the interests of other parties [1, 6]. While [6] conceptualized the MRS problem, there is no known general solution and different authors have started to address MRS subproblems. [17] use an experimental study to show that RS that consider profits can have a positive effect on purchasing without diminishing trust. [2] also demonstrate that RS can increase revenue without a significant drop in user satisfaction. One of the first attempts to propose a model for the MRS problem for MSPs is [22], which focused specifically on promoting retailer retention by guaranteeing a minimum level of exposures to each.

Following [22], we also propose a post-processing optimization problem. Our contribution is to formulate an optimization model specifically for the sponsored recommendation ad problem. Rather than constraining the number of recommendations that each retailer (provider) receives, our model includes two terms in the objective function for the different stakeholders and provides a tuning parameter to determine how much weight to give to each. Our model allows for advertisers to have specified budgets and also allows for the system to restrict the number of sponsored recommendations that any user receives to not exceed some threshold.

3 THE MULTISTAKEHOLDER SPONSORED RECOMMENDATION ALGORITHM

We assume that a platform has item set $\mathcal{I} = \{1, \ldots, n\}$, some of which are sponsored. Our task is to identify top-k lists for each of m users in the set $\mathcal{U} = \{1, \ldots, m\}$. Let $\mathbf{R}$ be the $m \times n$ rating matrix, where r_{ui} is the rating of item i by user u and $\widehat{r}_{ui}$ is the predicted rating from some traditional RS. Among the k items, at most $s_{\max} \leq k$ can be sponsored, which allows the platform to set an upper limit on the number of ads to show any single user. For now, we assume the *offline assignment problem*, where the m users are known in advance. This situation arises when, for example, a system plans to send email promotions to m known users.[1] Let a_{ui} be the ad revenue (or expected revenue, as discussed below) associated with displaying item i to user u. If item i is not sponsored then $a_{ui} \equiv 0$. Let β_i be the total ad budget associated with item i, where non-sponsored items have $\beta_i \equiv 0$.

We model the recommendation output via decision variable x_{ui} where $x_{ui} = 1$ if item i is included in the top-k list for user u and 0 if not. The $m \times n$ matrix of x_{ui} variables, denoted in matrix form as $\mathbf{X}$, determines which recommendations will be delivered in the email promotion. We propose the following optimization model:

$$\max_{\mathbf{X}} \left[\gamma \overbrace{\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \widehat{r}_{ui} x_{ui}}^{\text{User utility}} + (1-\gamma) \overbrace{\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} a_{ui} x_{ui}}^{\text{Ad revenue}} \right] \quad (2)$$

subject to

$$\sum_{i \in \mathcal{I}} x_{ui} = k \quad (\forall u \in \mathcal{U}) \quad (3)$$

$$x_{ui} \in \{0, 1\} \quad (\forall i \in \mathcal{I}, u \in \mathcal{U}) \quad (4)$$

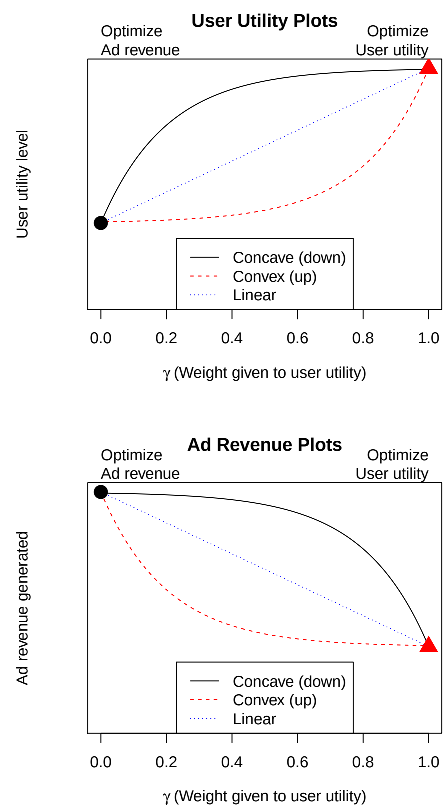
$$\sum_{i \in \mathcal{I}} \mathbb{1}(a_{ui} x_{ui} > 0) \leq s_{\max} \quad (\forall u \in \mathcal{U}) \quad (5)$$

$$\sum_{u \in \mathcal{U}} a_{ui} x_{ui} \leq \beta_i \quad (\forall i \in \mathcal{I}) \quad (6)$$

We now discuss each part of the model, starting with the objective function in (2). The optimization problem is to select over all possible $\mathbf{X}$ values, i.e., the top-k lists for the m users, the values of $\mathbf{X}$ to maximize a two-part objective. The first part, labeled "user utility," measures the total utility of the items recommended to all users. Product $\widehat{r}_{ui} x_{ui}$ equals 0 for all items not selected, and the estimated utility for those selected; terms are summed over all users and items. The second part of the objective function, labeled "Ad revenue," measures the total revenue from recommended sponsored items. Product $a_{ui} x_{ui}$ equals 0 for non-recommended items and a_{ui} for recommended ones. The analyst-specified tuning parameter γ ($0 < \gamma \leq 1$) determines how much weight to assign to the two parts. When $\gamma = 1$ the objective gives all weight to customer utility and produces the same solution as a traditional RS. When $\gamma \approx 0$ the algorithm places nearly all the weight[2] on ad revenue and mostly ignores user utility, which is how programmatic advertising auctions operate. The constraints are interpreted as follows:

- (3): the number of recommended items is k for each user u.
- (4): x_{ui} takes only values 0 or 1.
- (5): the number[3] of sponsored recommendations shown to user u must not exceed analyst-specified $s_{\max}$. Note that $\mathbb{1}(\cdot)$ is the indicator function, which equals 1 when the Boolean condition in the argument is satisfied and 0 otherwise.
- (6): the total amount of ad revenue from item i does not exceed the budget for that item, β_i.

Figure 1: Possible user-utility and ad-revenue curves



We now discuss the two parts of the objective function (2), total user utility and total ad revenue. Figure 1 shows some possible curves for the two parts as a function of γ. First, suppose that $\gamma \approx 0$, indicating that the algorithm should focus on optimizing ad revenue. The amount of user utility and ad revenue are indicated by the black dots on the left side ($\gamma \approx 0$) of the respective plots. Ad revenue is at its maximum value, while utility is at its smallest value, although the ads should produce positive utility. If we start to consider user utility by increasing the value of γ two things will happen: (1) user utility will either increase or stay the same, and (2) ad revenue will either decrease or stay the same. An important question is, how? The best scenario is when both the the user-utility

[1] The "online" situation where we do not know which customers will require recommendations will be discussed in the future research section. This online situation arises when, for example, customers visit the system's website, but the system does not know in advance which ones will show up on a given day.

[2] We do not allow $\gamma = 0$ because then the algorithm would have no way of allocating non-sponsored items when, for example, $s_{\max} < k$. When γ is very small, e.g., $\gamma = 10^{-8}$, then up to $s_{\max}$ *sponsored* items are selected based on the ad revenue, and the remaining non-sponsored items, which have ad revenue equal to 0 causing the ad revenue term in (2) to equal 0, are selected based on user utility. Constraining $\gamma > 0$ also addresses other degenerate cases, such as not having enough sponsored items to fill $ns_{\max}$ slots.

[3] Sometimes it is more convenient to express this constraint as an upper bound on the *fraction* of sponsored items rather than the number. In such cases, one can compute $s_{\max}$ by multiplying the upper-bound fraction by k and using the algorithm as stated.

and ad revenue curves are concave, shown by black, solid lines. In this case a small increase of γ from 0 produces a large increase in utility with very little loss in ad revenue. The worst case is when both curves are convex, shown by red, dashed lines, where a small increase of γ from 0 results in a large decrease in ad revenue with little improvement to utility.

We start at the right sides of the plots at the red triangles ($\gamma = 1$), where the algorithm should focus exclusively on user utility, as with a traditional RS. User utility is maximized, while ad revenue is at its lowest point. The same conclusions hold that the concave functions are the best scenario and convex the worst. If we decrease γ, allocating some weight to ad revenue, and the curves are concave, then utility decreases by a small amount while ad revenue increases substantially. The curvature will be investigated in our empirical examples for a real company.

The curves will be concave when the platform has flexibility. When there are many sponsored and non-sponsored items with roughly the same utility, the platform can substitute one item for another in a top-k list without sacrificing much utility. Likewise, when there are many sponsored items with roughly the same ad revenue, substitutions can be made to improve utility.

To solve the optimization problem, rewrite (2):

$$
\begin{aligned}
Q &= \left[\gamma \overbrace{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}} \widehat{r}_{ui}x_{ui}}^{\text{User utility}} + (1-\gamma)\overbrace{\sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}} a_{ui}x_{ui}}^{\text{Ad revenue}}\right] \\
&= \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}} x_{ui}\underbrace{[\gamma\widehat{r}_{ui} + (1-\gamma)a_{ui}]}_{c_{ui}}
\end{aligned}
$$

where optimization criterion $c_{ui} = \gamma\widehat{r}_{ui} + (1-\gamma)a_{ui}$. This expression is maximized by selecting the largest c_{ui} values, subject to the constraints, leading to Algorithm 1 below. The most computationally expensive part of the algorithm is sorting the mn elements of c_{ui} in descending order. Sorting has super-linear computational complexity, requiring on the order of $mn\log(mn)$ operations, so this algorithm will be computationally feasible for very large problems. Sorting is indicated by an index function that returns a user-item pair, e.g., no other c value is larger than $c_{u(1),i(1)}$; also $c_{u(h),i(h)}$ gives the h^{th} largest c value. The variable s_u counts the number of sponsored recommendations shown to user u, b_i totals the amount of ad budget from item i spent, and k_u counts the number of items recommended to user u. The algorithm walks through c_{ui} one element at a time, starting from largest, until each of the m users have received k recommendations.

We discuss some potentially challenging situations. In line 11 of the algorithm there could be situations where a sponsored item ($a_i > 0$) has the highest utility (i.e., higher than remaining non-sponsored items), but either the ad budget (β_i) for the item has been exhausted or the maximum number of sponsored item slots ($s_{\max}$) has been filled. In these cases, the algorithm should display the item as non-sponsored, since it has the highest utility. In order to have transparency, there should not be a sponsored designation displayed next to the item and the platform should not charge for the recommendation. Note that for $\gamma < 1$ the sort order of c_{ui} changes if ad revenue is set to 0. An item has more value to the platform

Algorithm 1 Multistakeholder advertising recommendation

Require: Analyst sets the weight for user utility (γ), and the max number sponsored recommendations shown to a user ($s_{\max}$). Advertiser sets budget (β_i).

1: $c_{ui} \leftarrow \gamma\widehat{r}_{ui} + (1-\gamma)a_{ui}, \quad \forall u, i$
2: Sort c_{ui} in descending order with index function mapping $h \in \{1,\ldots,mn\}$ to user-item pairs $[u(h), i(h)]$ such that $c_{u(1),i(1)} \geq c_{u(2),i(2)} \geq \cdots \geq c_{u(mn),i(mn)}$.
3: $s_u \leftarrow 0, \quad \forall u$ ▷ Initialize counter of sponsored ads to user u
4: $b_i \leftarrow 0, \quad \forall i$ ▷ Initialize spent ad i budget item i
5: $x_{ui} \leftarrow 0, \quad \forall u, i$ ▷ Initialize item to be not selected
6: $k_u \leftarrow 0, \quad \forall u$ ▷ Initialize number recommended to user u
7: **for** $h \leftarrow 1$ until all users are assigned exactly k items **do**
8:   **if** $k_{u(h)} < k$ **then**
9:     $x_{u(h),i(h)} \leftarrow 1$ ▷ Add item to top-k list
10:     $k_{u(h)} \leftarrow k_{u(h)} + 1$ ▷ Increment recommendation count
11:     **if** $a_{u(h),i(h)} > 0$ and $b_{i(h)} + a_{u(h),i(h)} \leq \beta_{i(h)}$ and $s_{u(h)} < s_{\max}$ **then**
12:       Label item sponsored
13:       $s_{u(h)} \leftarrow s_{u(h)} + 1$ ▷ Increment sponsored count
14:       $b_{i(h)} \leftarrow b_{i(h)} + a_{u(h),i(h)}$ ▷ Tally ad spend
15:     **end if**
16:   **end if**
17: **end for**

when it is sponsored than when it is not: $\gamma\widehat{r}_{ui} + (1-\gamma)a_{ui} > \gamma\widehat{r}_{ui}$. A simple way to handle this is to include all sponsored items twice in the list, once as sponsored and once as non-sponsored ($a_{ui} = 0$), and then do the sort. An additional if-then will be necessary to guarantee that if an item has already been selected to be shown as sponsored, it cannot also be selected as non-sponsored. Second, when allocating the last (k^{th}) slot there could be multiple items with the same value of criterion c_{ui}. Since the criterion values are tied it does not matter which one is selected, and the algorithm could select one at random or some with tie-breaking rule.

3.1 Commensurate units

The analyst should be aware that the two parts of the objective function have different units: a_{ui} is likely measured in some currency such as dollars or euros, while the units for $\widehat{r}_{ui}$ depend on how utility is operationalized. For example, it could be the log number of previous purchases plus one, the probability of purchase, or the estimated rating on a five-point scale. One way to avoid this problem is to use "unitless," standardized versions of a_{ui} and $\widehat{r}_{ui}$ in the objective function.

A more desirable common unit would be to express $\widehat{r}_{ui}$ in the same monetary unit as a_{ui}, e.g., dollars. Assigning a monetary interpretation, however, is a difficult task requiring substantial modeling. In addition to quantifying the probability of a current purchase, one would also have to consider the long-term implications. If a recommended item produces a sale, the retailer receives some short-term profit from selling the item, but also has a customer with better recency-frequency-monetary (RFM) status: recency is now zero days, frequency is one larger and monetary value is greater [15]. All three variables tend to be associated with higher purchase probabilities in the future [4]. Failing to account for the long-term effects produces evaluations of the value of the recommendation that are biased downward. As noted in the future research section, if such common units are available, one could optimize over γ as a decision variable rather than specifying it exogenously.

3.2 Ad revenue

Algorithm 1 requires the analyst to provide values of a_{ui}, the ad revenue associated with item i for user u. Notice that there are two subscripts, allowing different revenue amounts for different users and items. This subsection discusses the source of the a_{ui} values. We intend for our algorithm to be general and work for many current RS as well as systems that have not yet been devised. We assume that advertisers can set an ad budget and there are auctions for exposures. The advertiser can select items on which to place bids, and may have the ability to modify the bid amount based on additional information about the user, e.g., the advertiser could increase the bid for users in a certain demographic segment.

There are currently different revenue models for computational advertising. For example, advertisers can pay for a view, click, lead, conversion or engagement, such as signing up for an email list or newsletter. These models have corresponding metrics such as cost-per-view (CPV, which is often expressed in the cost for 1000 impressions, abbreviated CPM), cost-per-click (CPC), etc. In the simplest case where there is a fixed cost per exposure and the value a_{ui} is a constant. In more complicated cases, a_{ui} is an expected amount, which may be computed from a predictive model or various auction strategies. For example, there could be a model estimating the chance of a conversion and probability of winning the auction based on a certain bid. Or there could be a model where platform chooses from auction strategies like a first-price auction, second-price auction, etc. However, we focus on the algorithm and take ad revenue as a constant.

4 EMPIRICAL EVALUATION

We return to the example in the introduction between advertisers A and B, where A has bid more than B and won the auction. We investigate the sensitivity of the ad revenue and utility curves as a function of γ using data from an online grocery retailer.

4.1 Data and Preparation

The data is from an online grocery retailer in the United States, operating in 12 states and Washington D.C., with approximately 25,550 unique stock-keeping units (SKUs), 99,556 unique users, and 2,286,831 orders. The first step in building a RS is to define what an "item" is by aggregating the SKUs, since each size, flavor, etc. has its own SKU. For example, in the subcategory "soda," the brand "Diet Coke" is separate from "Coke Classic." The retailer provided a taxonomy of 136 subcategories, which, when crossed with brand names, gives 3,966 combinations. We clean the data further by combining different brand lines of the same product into a single brand. For example, within diapers "Pampers" is a single brand, while Huggies has many sub-brands such as "Little movers," "Overnites," "Pull ups," etc., which are collapsed into one Huggies brand. We count the number of orders that contain each subcategory-brand combination. With this approach, the brand "Dean's" in the Ice Cream subcategory is a distinct brand from Dean's milk, Dean's juice and Dean's cottage cheese, but all flavors and sizes of Dean's ice cream are grouped together. Many brand-subcategory combinations have small order counts. We minimize rare items by relabeling any subcategory-brand combination with fewer than 2500 orders as having brand "other," which gives a total of n=1,021 combinations, defining an "item" for the algorithm.

Table 1: Results from Traditional RS: Item based- and user based-collaborative filtering (IBCF and UBCF)

Algorithm	NNs	Test set RMSE
UBCF	5	0.672
	10	0.666
	20	0.661
	30	0.657
	50	0.652
	70	0.648
	90	0.646
	100	0.645
	110	0.644
IBCF	–	0.965

We only examine regular users who have placed 15 or more orders and have purchased within the past six months (i.e., recency $\leq$ 0.5 years), giving a universe of 19,360 users. We draw a simple random sample of size n=2,000 users, and build our $\mathbf{R}$ matrix with roughly two million entries (2,000 users $\times$ 1,021 items). We use implicit ratings by taking the log number of past purchases, which provides gradation between products that are never purchased, purchased occasionally, and purchased often by a user.

In our data items were not sponsored. In order to study sponsored items, we assume that some fraction of the branded items is sponsored. Private-label items are not allowed to be sponsored because they are owned by the retailer. Likewise, we exclude "other" subcategories, e.g., "other cottage cheese" because they would be composed of multiple small brands. We assign a \$1000 budget to each branded item for sponsored recommendations and assume ad revenue (a_i) drawn randomly from the Gaussian distribution $\mathcal{N}(\mu = 1, \sigma = 0.2)$. The algorithm is then executed, and the results are compared for different values of γ. We conduct robustness check by varying the fraction of sponsored items.

4.2 Estimating User Utility

Our algorithm post-processes ratings from a traditional RS to address the MRS problem. Hence, an RS must first predict ratings for items each user has not rated yet. We compare item- and user-based collaborative filtering (IBCF and UBCF). For UBCF, we use cosine similarity and vary the number of nearest neighbors (NNs): 5, 10, 20, 30, 50, 70, 90, 100 and 110. The performance of both traditional RS is measured by the root mean square error (RMSE). We use five-fold cross validation to evaluate traditional RS algorithms and follow the established model-building process outlined in section 3.2 of [8]. Table 1 summarizes the results. UBCF performs better than IBCF. There is little difference in RMSE between different numbers of nearest neighbors, and we use 30.

4.3 Results from Proposed Algorithm

After estimating the ratings with UBCF, we test our algorithm. The goal is to recommend the top 20 items ($k_u = 20$) to each user, among which, at most three ($s_{\max} \leq 3$) can be sponsored. To put both ratings ($\hat{r}_{ui}$) and ad revenue (a_{ui}) values on the same scale, we use their Z-scores to calculate c_{ui}. Figure 2 displays the results of the algorithm for user utility and ad revenue as a function of γ with different percentages (10%, 20%, 30%, 40%, 50%) of sponsored items. In all five cases, we see that as γ (the weight given to user utility) increases from zero to one, user utility increases and ad revenue decreases monotonically. The ad revenue plot (bottom) is mostly concave as we had hoped. The user utility curve (top), however, is convex for small values of γ, but mostly concave for large values.

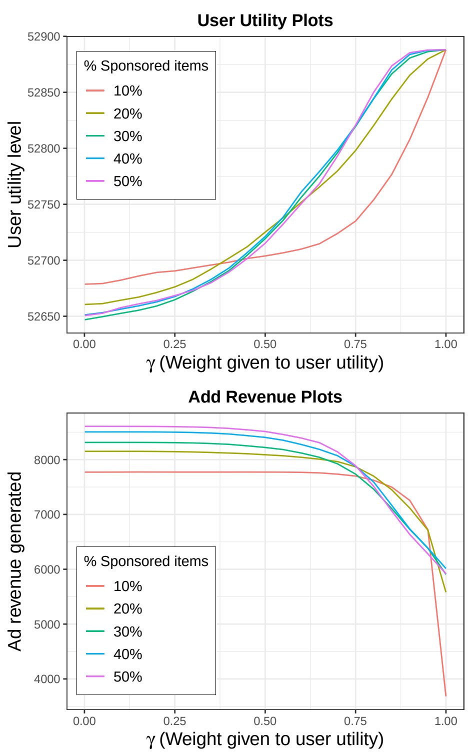


Figure 2: User utility and ad revenue curves estimated for real data from an online grocery retailer

When $\gamma \approx 0$, all weight is given to optimizing ad revenue. This can be considered the status quo for sponsored ads. As shown in ad revenue plot (Figure2, bottom), when the number of sponsored items increase, there are more options to choose from and ad revenue grows. In the user utility plot, having fewer options in sponsored items would lead the algorithm to choose items with higher user utility, generating higher overall user utility in the case of 10%, compared to 50% of sponsored items.

After γ reaches approximately 0.5, the ad revenue plots show a faster decrease in the cases with a larger number of sponsored items. With more sponsored items in the inventory, as γ increases, c_{ui} puts more weight on user utility and the algorithm has more ways to select sponsored items that have lower ad revenue but higher ratings. At the end, the sum of ad revenue has a faster decline in the cases with more sponsored items as to when the number of sponsored items is limited, since those limited items might have had higher ad revenue with higher ratings. In all cases, the ideal scenario with both curves concave is partially realized.

We compare our proposed algorithm for MRS with traditional RS, i.e., UBCF. While evaluating MRS, previous studies have shown that a slight decrease in one objective improves over the previous system if there is a substantial increase in competing objectives [6, 19]. To compare the performance of our algorithm with a traditional RS, we show the change in one objective as we vary the weight given to the competing objective. Recall when $\gamma = 1$, the algorithm chooses the top 20 items solely based on user's utility. At 20% sponsored items, we observe that as γ changes from 1 to 0.75, user utility decreases by only 0.17%, while ad revenue increases substantially by more than 40%.

We can also evaluate our algorithm by plotting total ad revenue against total user utility for different values of γ, as shown in Figure 3. For comparison, the axes have been scaled so that 0 indicates the minimum value of ad revenue or utility, and 100 indicates the maximum value. The values of the points on the curve correspond to different γ values. This resembles a receiver operator curve (ROC) [9], where, in this application, the area under the curve (AUC) measures how much flexibility the platform has. In an ideal situation, the platform would be able to achieve 100% of both the ad revenue and user utility, giving a point in the upper right corner of the graph. In our example, we can see that reducing the ad revenue by a fairly small amount increases user utility substantially. For example, changing γ from 0 to 0.50 achieves roughly a 28 point increase in user utility while ad revenue decreases by only 2 points.

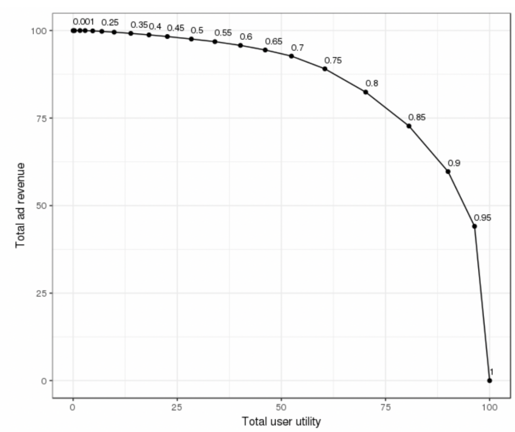


Figure 3: Plot showing total ad revenue against the total user utility for different values of γ

5 FUTURE RESEARCH AGENDA

Although our optimization model and algorithm to solve it make important contributions to the fields of computational advertising and RS, we believe this to be only the beginning of a larger area for future research. We therefore provide an extended discussion of open research questions and extensions. The algorithm should be evaluated with field tests, on data sets from different industries, and with more traditional RS algorithms.

Understand the relationship between customer utility and customer lifetime value (CLV). When a platform recommends items with lower utility than others, how does this affect CLV? If this relationship is understood, then the trade-off can be between ad revenue and CLV, which are both measured in the same units (e.g, dollars). Then γ could potentially become a variable that is selected by the optimization algorithm instead of being set by the manager. Game theory [21] could also provide relevant insights as it is often used to model the stability of a system. It is a trade-off where the gain of one player is a loss of another. As our study also shows, the system manager makes a trade-off of a small portion of ad revenue to generate more consumer-centric recommendations. The stability of the proposed model then lies in the expertise of a system manager on how well she can tune γ.

Understand how to handle situations where a user has high utility for a sponsored item. For example, if the item with the highest utility for user u is sponsored, should the system show a "sponsored" label? On the one hand, showing such a label can decrease the probability of purchase [3], and the user might select a non-sponsored item providing less utility than the sponsored one. On the other hand, if the system receives ad revenue for showing an item then the item is sponsored, and the system should be transparent, or risk losing the trust of its users.

Model components of ad revenue (a_{ui}). Ad auctions tend to be second-price, meaning that the highest bidder A pays the second-highest bid. This is from a game-theoretic result that keeps the auction stable, but it makes modeling revenue more complex. The second-price issue could be addressed by multiplying the highest bid by a constant that adjusts to the second-highest bid. In the proposed model, the rank is determined by user utility as well as the bid order. The ranking function is Bid$\times P(\text{click}|Ad)$ = expected revenue. Example: A is a spammer $P(\text{click}|A) = 0.001$; B is a real company $P(\text{click}|B) = 0.1$. A has to bid $100 \times B$'s bid to win the auction, not just B's bid plus some ϵ, and would have to pay an amount that provides expected revenue = B for each click. The second-price criterion is very difficult to model under these circumstances.

Develop algorithms for when items have different margins. A grocery retailer might make more profit from selling a private-label product, and recommend it as a non-sponsored item over other items [11]. We would need a separate set of constants for margin. Let $\mu(i, r)$ be the expected margin on item i for someone with utility r (which accounts for different click probabilities). Then the objective becomes:

$$Q_2 = \left[\gamma \overbrace{\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} \widehat{r}_{ui} x_{ui}}^{\text{User utility}} + (1-\gamma) \overbrace{\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} [\mu(i, \widehat{r}_{ui}) + a_{ui}] x_{ui}}^{\text{Revenue}} \right].$$

Optimize over $s_{\max}$. Setting $s_{\max}$ to a small value restricts ad inventory, which will likely increase auction bids and thus the ad revenue per slot, whereas larger values increase the supply of inventory and reduce the bids. This raises the possibility of reducing sponsored ad inventory without reducing ad revenue. One would need to know the elasticities of the supply and demand curves, which could be an important study on its own.

Develop algorithms to handle online situation, where users arrive over time and the platform does not know which ones will visit. A simple example will illustrate the problem. Suppose that we are to recommend $k = 1$ item and that there are $n = 2$ items A and B, each of which is sponsored and with enough budget for exactly one exposure today. There are $m = 2$ users, 1 and 2. The criterion values are $c_{1A} = 3$, $c_{1B} = 2$, $c_{2A} = 5$ and $c_{2B} = 2$. Suppose that user 1 visits the platform first, and so showing item A is optimal since $c_{1A} = 3 > 2 = c_{1B}$. Next user 2 visits, but since A has exhausted its ad budget, item B is shown giving $c_{2B} = 2$ and a total criteria of $3 + 2 = 5$. The optimal solution to the off-line problem, however, is to show user 1 item B and user 2 item A, giving a total criteria of $5 + 2 = 7 > 5$. Search advertising faces the same issue [14], and methods such as the Balance algorithm [13] have been proposed. Perhaps the Balance algorithm could be used in the sponsored recommendation situation after substituting c_{ui} for click-through rates, but the details should be carefully examined.

Improve the efficiency of Algorithm 1. The example in the previous paragraph also illustrates why the sorting in Algorithm 1 must be done over all mn items to find an optimal solution. Sorting within a user, as in the example, or within items may lead to a sub-optimal solution. While sorting has super-linear computational complexity and is therefore possible for large data sets, there might be ways improve the efficiency. For example, the slowest part of the algorithm is the sort, which can be sped up by reducing the number of c_{ui} values. If the inventory is, say, 10% of mn, one could drop all values less than, say, the 85$^{\text{th}}$ percentile, reducing the sort cost substantially. Knowing percentiles, however, requires sorting, unless the shape of the distribution is known. The c_{ui} values for the grocery data tend to be moderately right-skewed. The mean, variance and other moments could be found with a single pass, e.g., while computing c_{ui}. This would give information about the quantiles without sorting, which could then be used to drop user-item pairs with small c_{ui} values.

6 DISCUSSION

This study exemplifies how platform managers can consider both ad revenue and user utility while recommending items to the users. This is a paradigm shift from extant computational advertising approaches that focus on maximizing ad revenue and only consider user utility through click probabilities. Our approach treats the problem of MRS in the platform settings as a post-processing optimization step to be applied after user utilities have been estimated by an existing recommendation algorithm. Thus, our approach is not dependent on any particular RS algorithm, and can be widely applied. Our approach allows platform managers to trade-off ad revenue for higher user utility. Our empirical simulation suggests that shifting some weight to user utility will increase utility substantially while decreasing ad revenue only slightly, although this should be confirmed with field tests. Thus, we make computational advertising more user-centric by explicitly trading-off ad revenue to improve user utility while recommending sponsored items.

Advertising and RS have developed independently of one another and the two have starkly different underlying philosophies: the goal of advertising has been to persuade while the goal of RS has been to help users find items of interest to them. The ultimate goal—selling a product or, more generally, increasing CLV—may be the same, but the approaches are completely different. This study attempts to unify the two by balancing ad revenue and user utility in the specific context of sponsored recommendations. A more general question is whether this hybrid approach could be applied to other areas of advertising: can media selection and targeting work better by explicitly considering user utility? We believe this is a profoundly important question, as users find new ways to avoid ads.

ACKNOWLEDGMENTS

We thank Özge Sürer for helpful discussions and Northwestern University's Spiegel Research Center for supporting the second and third authors.

REFERENCES

[1] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. 2017. Recommender Systems as Multistakeholder Environments. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP'17)*. ACM, New York, NY, USA, 347–348.

[2] Amos Azaria, Avinatan Hassidim, Sarit Kraus, Adi Eshkol, Ofer Weintraub, and Irit Netanely. 2013. Movie Recommender System for Profit Maximization. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 121–128. https://doi.org/10.1145/2507157.2507162

[3] Joeran Beel, Stefan Langer, and Marcel Genzmehr. 2013. Sponsored vs. organic (research paper) recommendations and the impact of labeling. In *International Conference on Theory and Practice of Digital Libraries*. Springer, 391–395.

[4] Robert C Blattberg, Edward C Malthouse, and Scott A Neslin. 2009. Customer lifetime value: Empirical generalizations and some conceptual questions. *Journal of Interactive Marketing* 23, 2 (2009), 157–168.

[5] Robin Burke and Himan Abdollahpouri. 2017. Patterns of Multistakeholder Recommendation. In *VAMS Workshop 2017*. arxiv, http://arxiv.org/abs/1707.09258, Article 1, 5 pages. arXiv:1707.09258

[6] Robin D. Burke, Himan Abdollahpouri, Bamshad Mobasher, and Trinadh Gupta. 2016. Towards Multi-stakeholder Utility Evaluation of Recommender Systems. In *UMAP Extended Proceedings*, Vol. 1618. CEUR Workshop Proceedings, Aachen University.

[7] Benjamin Edelman, Sonia Jaffe, and Scott Duke Kominers. 2011. To groupon or not to groupon: The profitability of deep discounts. *Marketing Letters* (2011), 1–15.

[8] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[9] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[10] Andrei Hagiu and Julian Wright. 2015. Multi-sided platforms. *International Journal of Industrial Organization* 43 (2015), 162–174.

[11] Dietmar Jannach and Gediminas Adomavicius. 2017. Price and profit awareness in recommender systems. *arXiv preprint arXiv:1707.08029* (2017).

[12] Bernard J Jansen and Tracy Mullen. 2008. Sponsored search: an overview of the concept, history, and technology. *International Journal of Electronic Business* 6, 2 (2008), 114–131.

[13] Bala Kalyanasundaram and Kirk R Pruhs. 2000. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science* 233, 1-2 (2000), 319–325.

[14] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of massive datasets*. Cambridge university press.

[15] Edward C Malthouse. 2010. Accounting for the long-term effects of a marketing contact. *Expert Systems With Applications* 37, 7 (2010), 4935–4940.

[16] Edward C Malthouse, Ewa Maslowska, and Judy U Franks. 2018. Understanding programmatic TV advertising. *International Journal of Advertising* (2018), 1–16.

[17] Umberto Panniello, Shawndra Hill, and Michele Gorgoglione. 2016. The impact of profit incentives on the relevance of online recommendations. *Electronic Commerce Research and Applications* 20 (2016), 87–104.

[18] Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster Provost. 2012. Bid optimizing and inventory scoring in targeted online advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 804–812.

[19] Mario Rodriguez, Christian Posse, and Ethan Zhang. 2012. Multiple objective optimization in recommender systems. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 11–18.

[20] Amin Sayedi. 2018. Real-Time Bidding in Online Display Advertising. *Marketing Science* (2018).

[21] Amin Sayedi, Kinshuk Jerath, and Kannan Srinivasan. 2014. Competitive poaching in sponsored search advertising and its strategic impact on traditional advertising. *Marketing Science* 33, 4 (2014), 586–608.

[22] Özge Sürer, Robin Burke, and Edward C. Malthouse. 2018. Multistakeholder Recommendation with Provider Constraints. *ACM Recommendation Systems Conference Proceedings* (2018).

[23] Wush Chi-Hsuan Wu, Mi-Yen Yeh, and Ming-Syan Chen. 2015. Predicting winning price in real time bidding with censored data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1305–1314.