UJOA#1652123, VOL 0, ISS 0

An Algorithm for Allocating Sponsored Recommendations and Content: Unifying Programmatic Advertising and Recommender Systems

Edward C. Malthouse, Yasaman Kamyab Hessary, Khadija Ali Vakeel, Robin Burke, and Morana Fudurić

QUERY SHEET

This page lists questions we have about your paper. The numbers displayed at left are hyperlinked to the location of the query in your paper.

The title and author names are listed on this sheet as they will be published, both on your paper and on the Table of Contents. Please review and ensure the information is correct and advise us if any changes need to be made. In addition, please review your paper as a whole for typographical and essential corrections.

Your PDF proof has been enabled so that you can comment on the proof directly using Adobe Acrobat. For further information on marking corrections using Acrobat, please visit http://journalauthors.tandf.co.uk/production/acrobat.asp; https://authorservices.taylorandfrancis.com/how-to-correct-proofs-with-adobe/

The CrossRef database (www.crossref.org/) has been used to validate the references.

AUTHOR QUERIES

Q1 Please verify corresponding author's postal address and e-mail.

Q2 Please provide author's missing academic position for Fudurić

Q3 CPM = cost per thousand impressions, correct? Is this equivalent to cost per view? Please verify.

Q4 Please confirm title of book in the Ben-Orat and Tenneholtz 2018 reference and provide the names of the book's editors, as well as the name and city of the publisher.

Journal of Advertising, 0: 1–14

ISSN: 0091-3367 print / 1557-7805 online
DOI: 10.1080/00913367.2019.1652123

An Algorithm for Allocating Sponsored Recommendations and Content: Unifying Programmatic Advertising and Recommender Systems

Edward C. Malthouse
Northwestern University, Evanston, Illinois, USA

Yasaman Kamyab Hessary
Northwestern University, Evanston, Illinois, USA

Khadija Ali Vakeel
Northwestern University, Evanston, Illinois, USA

Robin Burke
University of Colorado Boulder, Boulder, Colorado, USA

Morana Fudurić
University of Zagreb, Zagreb, Croatia

Retailing and media platforms recommend two types of items to their users: sponsored items that generate ad revenue and nonsponsored ones that do not. The platform selects sponsored items to maximize ad revenue, often through programmatic auctions, and nonsponsored items to maximize user utility with a recommender system (RS). We develop a binary integer programming model to allocate sponsored recommendations considering dual objectives of maximizing ad revenue and user utility. We propose an algorithm to solve it in a computationally efficient way. Our method is a form of postfiltering to a traditional RS, making it widely applicable in two-sided markets. We apply the algorithm 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 unifies programmatic advertising and RS and opens a new frontier for advertising and RS research. We provide an extended discussion of future research topics.

Two-sided markets are defined as intermediary platforms that enable direct interactions between two distinct stakeholder groups, generically labeled *consumers* and *providers* (Hagiu and Wright 2015). 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. Just as advertisers pay media companies for access to their audiences, they pay platforms for access to consumers. The two stakeholder groups here 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*. Platforms can, therefore, be viewed as a new form of media. For example, retail-based shopping platforms like Taobao in China and Amazon Marketplace in the United States match consumers with

Address correspondence to Edward C. Malthouse, Medill School of Journalism, Media, Integrated Marketing Communications, Northwestern University, 1845 Sheridan Road, 304D Fisk Hall Evanston, IL 60208, USA. E-mail: ecm@northwestern.edu

Edward C. Malthouse (PhD, Northwestern University) is the Erastus Otis Haven professor and research director of the Spiegel Research Center, Medill School of Journalism, Media, Integrated Marketing Communications, Northwestern University.

Yasaman Kamyab Hessary (PhD, University of North Carolina in Charlotte) is a postdoctoral fellow at the Spiegel Research Center, Medill School of Journalism, Media, Integrated Marketing Communications, Northwestern University.

Khadija Ali Vakeel (PhD, Indian Institute of Management, Indore) is a postdoctoral fellow at the Spiegel Research Center, Medill School of Journalism, Media, Integrated Marketing Communications, Northwestern University.

Robin Burke (PhD, Northwestern University) is a professor in the Department of Information Science, College of Media, Communication, and Information, University of Colorado Boulder.

Morana Fudurić (PhD, Università della Svizzera italiana) is in the Faculty of Economics and Business, University of Zagreb.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/ujoa.

retailers. The platform attracts consumers and matches them with retail providers through recommendations. Instead of consumers, the terms *consumer* and *user* are often used; we employ *user* to refer to the first stakeholder group.

Platforms can show two types of recommendations, *sponsored* and *nonsponsored*, where sponsored providers (i.e., advertisers) pay the platform for access to the user and nonsponsored providers do not. Sponsored recommendations are often sold on programmatic auctions where the platform can adopt various strategies to sell slots (Sayedi 2018; Malthouse, Maslowska, and Franks 2018) to the highest bidder. Because nonsponsored recommendations do not generate ad revenue, we assume that platforms select nonsponsored items to be relevant to the user, which is often done using a class of machine learning and artificial intelligence (AI) algorithms known as *recommender systems* (RS), defined as "[s]oftware tools and techniques providing suggestions for items to be of use to a user" (Ricci, Rokach, and Shapira 2015, p. 1). One notable feature of this RS definition is its exclusive focus on a single stakeholder: the user. The sole purpose of a traditional RS is to recommend items that maximize the user's utility.

While most traditional RS generate recommendations to maximize the utility for the end user, this article develops algorithms for more complex systems where there is another stakeholder—the advertiser—and another objective created by the possibility of generating ad revenue through sponsored recommendations. For example, consider a user looking for a pasta sauce on an online grocer's website. A traditional RS would recommend pasta sauces (items) to maximize the user's utility, but what if some advertisers were willing to pay the retailer to be recommended? In this case the retailer has a second, competing objective of earning revenue from the advertiser. The situation we described is an example of a multiobjective, multistakeholder RS, abbreviated as MRS (Burke and Abdollahpouri 2017). This is a new and important problem in both advertising and RS that has received little attention in the literature. The MRS must decide which advertisers to recommend while striking a balance between sponsored and nonsponsored recommendations. We develop the MRS specifically for the sponsored recommendation problem, where a platform must decide which personalized set of providers to recommend to each user. We express the combined objectives in a single equation:

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

where γ is a parameter that determines how much weight to give to each of the two objectives and Q is the objective function value to be maximized. A platform that allocates sponsored recommendations with an auction has $\gamma = 0$ and is maximizing ad revenue. An RS that considers only the user's utility, such as those used to allocate nonsponsored recommendations, has $\gamma = 1$. Rather than focusing on one or the other, this equation allows for a continuum between the two competing objectives, providing the platform manager a way to manage the trade-off between the two objectives.

We formulate a postprocessing 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 and the number of sponsored recommendations to the user has not exceeded an analyst-specified threshold. We show how to solve the optimization problem in a computationally efficient way and illustrate the approach using real data from an online grocery retailer. Our approach improves user utility substantially while decreasing ad revenue by a small amount. This work follows a design-oriented research approach that has commonly been used in technical and computer sciences (Verschuren and Hartog 2005) and has, due to the nature of the field, recently been gaining traction in advertising research (see Henseler 2017). With this article, we advance research on programmatic advertising and RS by developing algorithms for more complex systems where there is another stakeholder (i.e., the advertiser), and another objective (i.e., generating ad revenue through sponsored recommendations).

LITERATURE REVIEW

Our approach postulates that there are two types of recommendations, sponsored and nonsponsored, which are selected with different objectives. This section begins with a review of how ad space for sponsored content and display ads is sold, and then gives an overview of the RS used to recommend nonsponsored content. We then discuss existing literature on how to manage competing objectives.

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. Typically, the auctions are second price, where the winner pays the second-highest bid amount for the space. The second-price auctions have an optimal strategy for an advertiser of bidding the expected value of the impression (see Edelman, Ostrovsky, and Schwarz 2007). There is a rich literature within computer science and marketing on bidding strategies. For example, Perlich et al. (2012) 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. Wu et al. (2015) proposed a way to leverage machine learning to predict the winning amount from the bidding history. It is important to note that in these stratergies 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 exposure to an advertiser.

Our formulation of the MRS problem for sponsored recommendations is related to the AdWords problem (Leskovec, Rajaraman, and Ullman 2014, section 8.4.2), 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 (CTRs) 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 have both bid on the search term and have budget to pay if the ad is clicked on.

Recommendation Systems Maximizing User Utility

RS is a well-established branch of machine learning and AI. Ekstrand, Riedl, and Konstan (2011) give an overview of their history and survey different collaborative filtering approaches; Adomavicius and Tuzhilin (2005) is a highly cited survey of the field; and Ricci, Rokach, and Shapira (2015) is an extensive collection from contributors covering the different state-of-the-art areas of RS. An RS incorporates a set of users and a set of items. The core task is to generate personalized matches between users with items that are of interest to them. We denote the set of users by $\mathcal{U} = \{1, ..., m\}$ and the set of items by $\mathcal{J} = \{1, ..., n\}$. We reserve the variable u to index users and i to index items.

The starting point of data for a RS is the *$m \times n$ rating matrix* $\mathbf{R}$, where element r_{ui} is the "rating" of item i by user u. The term *rating* denotes preference or utility. A rating could be *explicit*, such as a rating on a five-point scale of how much the user liked an item (e.g., a movie, song, restaurant). In our situation we are more likely to have what the RS literature calls *implicit ratings*, which are given by whether or not the user has purchased an item, or the number of times an item is purchased or consumed.

RS usually have two possible goals: *prediction* and *recommendation*. The prediction goal is to estimate the missing ratings (denoted by $\hat{r}_{ui}$). The recommendation goal is to recommend k items to some user, which is called a top-k list. For example, a retailer might send a personalized e-mail with a 4×5 grid of item suggestions, which is called a "top-20" list. If the goal is to recommend a list of k items to maximize the utility of end users, we can choose the k items with the highest estimated utility $\hat{r}_{ui}$ for each user, although there could be other considerations, such as recommending items from a diverse set of product categories. Many algorithms have been proposed to compute estimates $\hat{r}_{ui}$. We briefly summarize one specific RS algorithm, user-based collaborative filtering (UBCF). Suppose that the RS must make a prediction $\hat{r}_{ui}$ for target item i by some active user u, where u has not rated i. UBCF first finds other users whose ratings are similar to the active user and who have rated i. These other users are called *nearest neighbors*. Prediction $\hat{r}_{ui}$ is then the average of the nearest neighbors' ratings of item i. Additional refinements will improve the estimate, such as giving more weight to the ratings of item i of neighbors who are more similar to active user u. While validating our proposed algorithm on online grocery data, we use UBCF to calculate user utility.

Multistakeholder Recommender System

MRS refers to a design that incorporates the interests of other parties (Burke et al. 2016; Abdollahpouri, Burke, and Mobasher 2017; Abdollahpouri et al. 2019). The objectives of different stakeholders, however, may conflict (Ribeiro et al. 2012; Sürer, Burke, and Malthouse 2018). For example, in the case of a two-sided retail platform, the objective for users is to find items that maximize their utility, while the objective for a specific retail provider is to maximize its orders, which are stimulated by being recommended to many users. The platform also wants to generate ad revenue. Managing recommendations for a platform is therefore a complex task. However, previous studies have shown that RS that consider profits can have a positive effect on purchasing without diminishing trust (Panniello, Hill, and Gorgoglione 2016). Similarly, Azaria et al. (2013) also demonstrate that an RS can increase revenue without a significant drop in user satisfaction.

While Burke et al. (2016) conceptualized the MRS problem, to the best of our knowledge there is no known general solution. As MRSs have a wide variety of forms (see Burke and Abdollahpouri 2017), they require a variety of solution approaches. In our situation the two main stakeholders are users and advertisers. Users expect recommendations to match their preferences and to maximize their utility. Advertisers want to sell their items. The platform wants users to return while also earning ad revenue.

TABLE 1
Selected Literature on Multiobjective, Multistakeholder Recommender Systems (MRS)

Literature	Context	MRS	Algorithm	Objective
Rodriguez, Posse, and Zhang (2012)	Job posting	Multiobjective	RS using semantic matching and extraneous factors	User utility with extraneous factors
Ribeiro et al. (2012)	Movies and music	Multiobjective	Strength Pareto approach in hybrid RS	Accuracy, novelty, and diversity
Burke et al. (2016)	Movies	Yes	Greedy approach	Multidimensional utility
Sürer, Burke, and Malthouse (2018)	Movies and retailing	Yes	Constraint-based integer programming optimization model	Utility of buyers and providers
Ben-Porat and Tennenholtz (2018)	Conceptual	Yes	Game theory and Shapley mediator	Fairness and stability
This article	Sponsored recommendations in programmatic advertising	Yes	Postfiltering optimization RS	User utility and ad revenue

Note. RS = recommender systems.

Previous studies have also explored multiobjective (see Rodriguez, Posse, and Zhang 2012; Ribeiro et al. 2012) and multicriteria (see Adomavicius and Tuzhilin 2011; Zheng 2017) RS. However, they do not address the specific problems with sponsored ads. For example, Ribeiro et al. (2012) use the strength Pareto approach to optimize three different objectives of accuracy, novelty, and diversity. Further, Zheng (2017) maximizes performance in multicriteria RS. Relatedly, other authors have explored the greedy approach (Burke et al. 2016), probabilistic topic models (Zihayat et al. 2019), and the Shapley mediator using game theory (Ben-Porat and Tennenholtz 2018) to optimize user utility functions. From the limited literature on MRS (Table 1), one of the first attempts to propose a model for the MRS problem is Sürer, Burke, and Malthouse (2018), which focused specifically on retail platforms and the problem of promoting retailer retention by guaranteeing a minimum level of exposure to each but did not consider advertising. This study proposes a postprocessing step for use after a traditional RS has estimated the utilities of each user for each item.

THE MULTISTAKEHOLDER SPONSORED ADVERTISING RECOMMENDATION ALGORITHM

We assume that the platform must generate k recommendations for each user, which is called a top-k list in the RS literature. For example, the platform may have to generate e-mail newsletters or promotions for its user, each containing k recommended items. Or a retailer may send out shipping list e-mails to users placing an order today, and at the bottom of the e-mail it recommends k items, attempting to stimulate another order. Some items are sponsored, where each provider has some fixed ad budget for the period, and other items are not sponsored. To avoid showing too many sponsored items to any single user, we also allow for the platform to specify a constraint restricting the maximum number of sponsored items.

We assume that a platform has n items that can be recommended, some of which are sponsored. Our task is to identify personalized top-k lists for each of m users. Among the k items, at most $s_{\max} \leq k$ can be sponsored, which allows for the platform to set an upper limit on the amount of advertising to any single user if it wishes. For now, we assume that the m users are known in advance. Let a_{ui} be the ad revenue (or expected revenue, as is discussed next) associated with displaying item i to user u. If item i is not sponsored, then $a_{ui} = 0$. Let β_i be the total advertising budget associated with item i, where nonsponsored items have $\beta_i = 0$.

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 shown. Since each user receives k recommendations, exactly k values of x_{ui} will equal 1 for a given user. We propose the following optimization model, where $\forall$ means "for all" and $\in$ means "element of":

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

subject to:

$$\sum_{i \in \mathcal{I}} x_{ui} = k \;\; \forall u \in \mathcal{U} \tag{3}$$

$$x_{ui} \in \{0, 1\} \;\; \forall i \in \mathcal{I}, u \in \mathcal{U} \tag{4}$$

$$\sum_{i \in \mathcal{I}} 1(a_{ui} x_{ui} > 0) \le s_{\max} \;\; \forall u \in \mathcal{U} \tag{5}$$

$$\sum_{u \in \mathcal{U}} a_{ui} x_{ui} \le \beta_i \;\; \forall i \in \mathcal{I} \tag{6}$$

We now discuss each part of the model, starting with the objective function in Equation (2). The optimization problem is to select from all possible $\mathbf{X}$ values, in other words, 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 $\hat{r}_{ui} x_{ui}$ 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 nonrecommended items and a_{ui} for recommended ones. The analyst-specified tuning parameter γ $(0 < \gamma \le 1)$ determines how much weight to assign to the two parts. When $\gamma = 1$ the objective gives all weight to user utility and produces the same solution as a traditional RS. When $\gamma \approx 0$ the algorithm places nearly all the weight[1] on ad revenue and mostly ignores user utility, which is how programmatic advertising auctions operate. The constraints are interpreted as follows:

- Equation (3): The number of recommended items is k for each user u.
- Equation (4): x_{ui} equals 0 or 1, making this a binary integer programming problem.
- Equation (5): The number[2] of sponsored recommendations shown to user u must not exceed $s_{\max}$, which is specified by the analyst. Note that $1(\cdot)$ is the indicator function, which equals 1 when the Boolean condition in the argument is satisfied and 0 otherwise.
- Equation (6): Total ad revenue from item i does not exceed the budget for that item, β_i.

Discussion of the Model

We discuss the two parts of the objective function (2): total user utility and total ad revenue. Figure 1 shows expected 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 amounts 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 user utility is at its smallest value, although sponsored recommendations should produce positive user 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. The best results for the proposed algorithm will be achieved when both the the user utility (left) and ad revenue (right) curves are concave, shown by black, solid lines. In this case, a small increase of γ from 0 produces a large increase in user utility (left) with very little loss in ad revenue (right). 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 (right) with little improvement to user utility (left).

Starting at the right side of the plot at the red triangle points, where $\gamma = 1$, indicates that the algorithm should focus exclusively on user utility, as with a traditional RS. At this point, 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 if the curves are concave, then utility decreases by a small amount (left) while ad revenue increases substantially (right). The curvature will be investigated in our empirical examples for real data.

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

Solving the Optimization Problem

To solve the optimization problem, we rewrite the objective function (2):

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

where optimization criterion $c_{ui} = \gamma \hat{r}_{ui} + (1-\gamma) a_{ui}$. Clearly this expression is maximized by selecting the largest c_{ui} values, subject to the constraints. This leads to the algorithm in Table 2. The most computationally

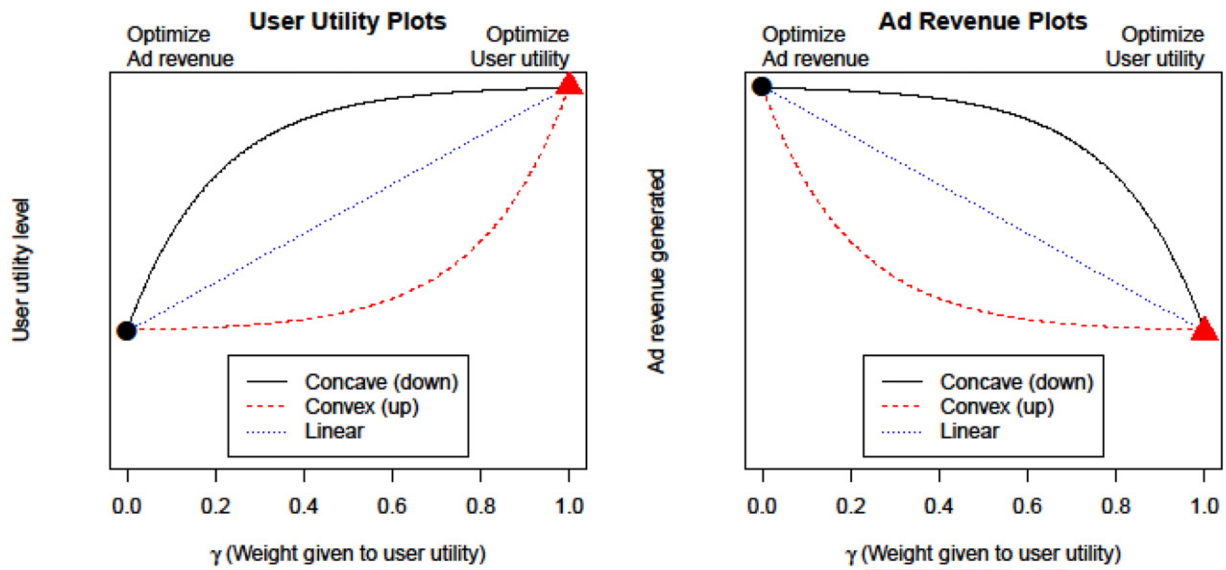

FIG. 1. Hypothetical user utility and ad revenue curves.

TABLE 2
Algorithm 1: The Multistakeholder Advertising Recommendaiton Algorithm

Require: (1) Analyst sets (1) γ, the weight for user utility, and (2) $s_{\max}$, the max number sponsored recommendations shown to a user. Advertiser sets budget β_i.

$c_{ui} \leftarrow \gamma \hat{r}_{ui} + (1-\gamma) a_{ui}, \quad \forall u, i$

Sort c_{ui} in descending order with index function mapping counting numbers $h \in \{1, \ldots, mn\}$ to user-item pairs $(u(h), i(h))$ such that $c_{u(1),i(1)} \geq c_{u(2),i(2)} \geq \cdots \geq c_{u(mn),i(mn)}$.

$s_u \leftarrow 0, \quad \forall u$ — Initialize counter of sponsored ads to user u

$b_i \leftarrow 0, \quad \forall i$ — Initialize spent ad i budget item i

$x_{ui} \leftarrow 0, \ \forall u, i$ — Initialize item to be not selected

$k_u \leftarrow 0, \quad \forall u$ — Initialize number recommended to user u

for $h \leftarrow 1$ until all users are assigned exactly k items **do**

    if $k_{u(h)} < k$ **then**

        $x_{u(h),i(h)} \leftarrow 1$ — Add item to top-k list

        $k_{u(h)} \leftarrow k_{u(h)} + 1$ — Increment recommendation count

        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**

            Label item sponsored

            $s_{u(h)} \leftarrow s_{u(h)} + 1$ — Increment sponsored count

            $b_{i(h)} \leftarrow b_{i(h)} + a_{u(h),i(h)}$ — Tally ad spend

        end if

    end if

end for

expensive part of the algorithm is sorting the mn elements of c_{ui} in descending order. This sorting has superlinear 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, for example, no other c value is larger than $c_{u(1),i(1)}$, and $c_{u(h),i(h)}$ 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 has received k recommendations.

We close this section by discussing situations that are potentially challenging and how we address them in the algorithm. First, there could be situations during the execution of the algorithm in line 11 where a sponsored item ($a_i > 0$) has the highest user utility (i.e., higher than remaining nonsponsored 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 nonsponsored because it has the highest user utility. To have transparency, there should not be a sponsored designation displayed next to the item, and the retailer should not charge for the recommendation.

Second, 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 when it is sponsored than when it is not, in other words $\gamma\hat{r}_{ui} + (1-\gamma)a_{ui} > \gamma\hat{r}_{ui}$. An easy way to handle this is to include all sponsored items twice in the list, once as sponsored and once as nonsponsored ($a_{ui} = 0$), and then do the sort. Therefore, if an item has already been selected to be shown as sponsored, it cannot also be selected as nonsponsored. Last, when allocating the last (k^{th}) slot, there could be multiple items with the same value of criterion c_{ui}. When the criterion values are tied it does not matter which one is selected, and the algorithm could select one at random or with another tie-breaking rule (see Bryce and Colbourn 2007).

Third, one of the most common RS problems is the item cold-start, where there are new items. Because a new item has no ratings, UBCF will never display it and older items will dominate the recommendations. Our proposed algorithm is a postprocessing procedure to the aforementioned step, and a retailer should have an existing method for dealing with cold-start, such as a hybrid RS combining content-based RS (which use meta data about items instead of ratings from other users) for new items with UBCF. Our algorithm could then be deployed separately on each of the component RS.

Commensurate Units

The analyst should be aware that the two parts of the objective function have different units: a_{ui} is likely measured in a currency such as dollars or euros, while the units for $\hat{r}_{ui}$ depend on how user 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 from 1 to 5. One way to address this problem is to use "unitless," standardized versions of a_{ui} and $\hat{r}_{ui}$ in the objective function.

A more desirable common unit would be to express $\hat{r}_{ui}$ in the same monetary unit as a_{ui}, for example, dollars. Estimating the monetary value, however, is a complex 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 user with better recency-frequency-monetary (RFM) status: Recency is now zero days, frequency value is one larger, and monetary value is greater (Malthouse, Maslowska, and Franks 2010). All three variables tend to be associated with higher purchase probabilities in the future (Blattberg, Malthouse, and Neslin 2009). Failing to account for the long-term effects produces evaluations of the value of the recommendation that are biased downward. As we discuss 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.

Ad Revenue

Our proposed algorithm 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 section discusses the source of the a_{ui} values. We intend for our algorithm to be general and to work for many current RS as well as future ones. We assume that advertisers can set an ad budget and that 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; for example, the advertiser could increase the bid for users in a certain demographic segment.

There are currently different revenue models for programmatic advertising. For example, advertisers can pay for a view, click, lead, conversion, or engagement, such as signing up for an e-mail list or newsletter. These models have corresponding metrics such as cost per view (CPV, also known as CPM), cost per click (CPC), etc. The simplest case is when the 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 machine learning 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. However, we focus on the algorithm and take ad revenue as a given.

EMPIRICAL EVALUATION

We first show an idealized example of our empirical evaluation to clarify the two competing objectives of our study: user utility and ad revenue. Next, we use data from an online grocery retailer to study the trade-off between the two.

Motivating Example

Suppose that advertiser A places the highest bid for a sponsored recommendation to a user. Advertiser B places a bid that is only slightly less than A's but would provide the user substantially more utility than A. We argue that the platform should consider giving the slot to B even though it would generate less ad revenue. A possible financial justification for this decision is that B may generate higher lifetime value, where the user buys B, ultimately is satisfied with B, and buys again from the platform in the future. If the platform had shown A, then either the user would not buy it or would not be satisfied and would be less likely to return to the platform in the future. 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 scenario the platform should display A. In this section, we evaluate these issues empirically. We demonstrate the feasibility of our algorithm and investigate the sensitivity of the ad revenue and user utility curves as a function of γ using data from an online grocery retailer.

Data and Preparation

The data are from an online grocery retailer in the United States operating in 12 states and Washington DC, with approximately 25,550 unique stock-keeping units (SKUs), 99,556 unique users, and 2,286,831 orders. The first step in building an RS is to define what an "item" is by aggregating the SKUs, as each size, flavor, and so on 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 subbrands such as Little Movers, OverNites, and Pull-Ups, 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. For the purpose of this example, we minimize rare items by relabeling any subcategory–brand combination with fewer than 2,500 orders as having brand "other," which gives a total of $n = 1,021$ combinations, defining an "item" for the algorithm.

We examine only 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. For the purpose of this example, we draw a simple random sample of size $m = 2,000$ users and build our $\mathbf{R}$ matrix with roughly 2 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 the current data set, items were not sponsored. 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, for example, "other cottage cheese," because they would be composed of multiple small brands. We assign a \$1,000 budget to each branded item for sponsored recommendations and assume advertising 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 checks by varying the fraction of sponsored items.

Preprocessing Step: Generating User Utility Using Traditional RS

Our proposed algorithm post processes ratings from a traditional RS to address the MRS problem. Hence, an RS must first generate ratings for products each user has not rated yet. For this example, we compare item- and user-based collaborative filtering (IBCF and UBCF). For UBCF, we use cosine similarity[3] 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,

$$\mathrm{RMSE} = \sqrt{\frac{1}{|T|}\sum_{(u,i)\in T}\left(\hat{r}_{ui} - r_{ui}\right)^2},$$

where T is the test set and $|T|$ is its cardinality. We use fivefold cross validation to evaluate traditional RS algorithms and follow the established model-building process outlined in Ekstrand, Riedl, and Konstan (2011, section

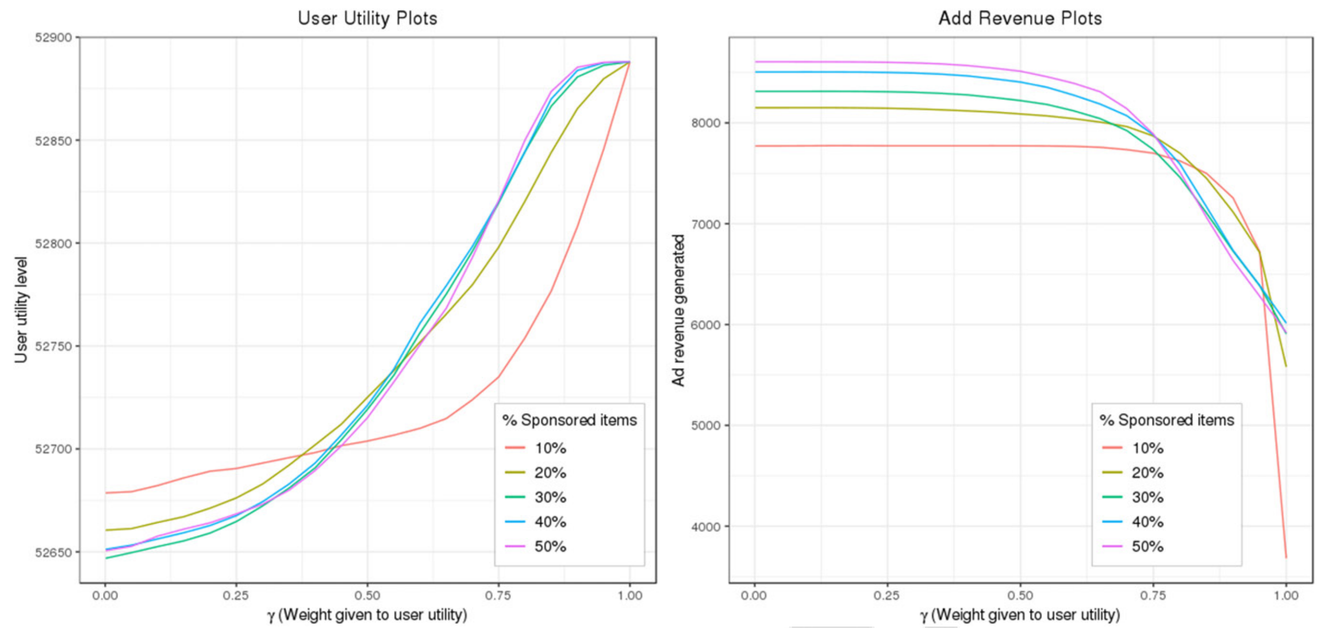

FIG. 2. User utility and ad revenue curves estimated for real data from an online grocery retailer.

TABLE 3
Results from Traditional RS: Item-Based and User-Based Collaborative Filtering (IBCF and UBCF)

Algorithm	NNs	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

Note. Two traditional RS (IBCF and UBCF) are used to generate user ratings and are evaluated using root measure square error (RMSE). UBCF was run for multiple nearest neighbors (NNs). Output matrix of traditional recommender systems is input to our proposed algorithm.

3.2) and Breese, Heckerman, and Kadie (1998). Table 3 summarizes the results of evaluation for traditional RS. The UBCF model performs better on our test data set than IBCF. There is very little difference in RMSE between different numbers of nearest neighbors, and we use 30.

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}. We expect that a small increase of γ from 0 will produce a large increase in user utility with very little loss in ad revenue (this is the best condition for our proposed algorithm, as mentioned before). 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 0 to 1, user utility increases, and ad revenue decreases monotonically. The ad revenue plot (right) is mostly concave, as we had hoped. The utility curve (left), however, is convex for small values of γ, but mostly concave for large values.

In the case of $\gamma \approx 0$, all weight is given to optimizing ad revenue, which maximizes ad revenue and minimizes user utility. This can be considered the status quo for sponsored ads. As shown in ad revenue plot (Figure 2), 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 higher 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

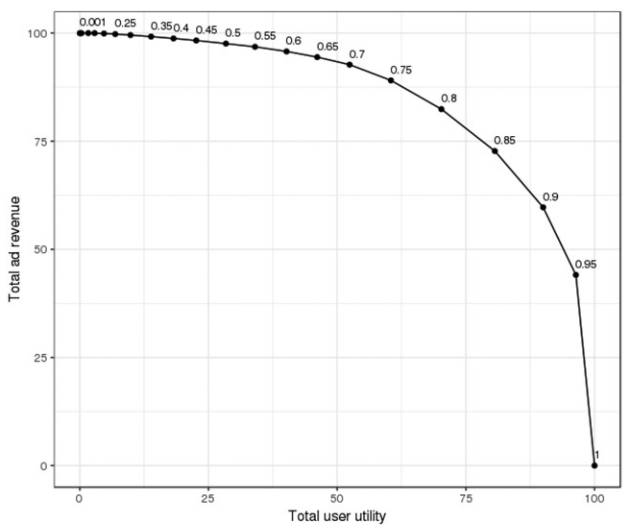

FIG. 3. Plot showing total ad revenue against the total user utility for different values of γ.

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.

Comparison with Traditional RS (UBCF)

We compare our proposed algorithm for MRS with traditional RS, in other words, 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 (Burke et al. 2016; Rodriguez, Posse, and Zhang 2012). To evaluate the performance of our algorithm in comparison with 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% (see Figure 2).

Evaluation of Proposed Algorithm

We evaluate our algorithm by plotting the total ad revenue against the 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 curve resembles a receiver operator curve (ROC; e.g., Fawcett 2006), 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 28 points' increase in user utility while ad revenue decreases by only 2 points.

We have also measured the ranking performance of our proposed algorithm by calculating the normalized discounted cumulative gain (*nDCG*) for all the values of γ with different percentages of sponsored items. It is one of the most popular rank-based metrics used in RS. This metric aims to capture the quality of a particular ranked list by assuming that an RS is more efficient if the higher rated items appear early in the recommendation list. In this case, the evaluated ranked list is the top-*k* recommended items to each user. *nDCG* varies between zero and one, and is defined as

$$nDCG = \frac{DCG}{IDCG}$$

$$DCG = \frac{1}{m}\sum_{u=1}^{m}\left(\hat{r}_{u1} + \sum_{i=2}^{k}\frac{\hat{r}_{ui}}{\log_2(i+1)}\right)$$

nDCG evaluates an ideal ranked list (*IDCG*) against the ranked list generated by our proposed algorithm. Here, *IDCG* is the *DCG* of top-*k* items for each user, chosen solely based on user utility, which is the list generated by the traditional RS, UBCF. The perfect ranking would have *nDCG* of value 1.

Table 4 shows the evaluation results of our proposed algorithm in terms of *nDCG* for different percentages of sponsored items. When $\gamma = 1$, the algorithm chooses the top-20 items exclusively based on user's utility, so *nDCG* is at its maximum. As γ decreases, more weight is given to improving the ad revenue, but as we can observe from the results, it does not hurt the overall quality of the recommendations by much.

DISCUSSION

This study shows how platform managers can consider both ad revenue and user utility while recommending sponsored items to the users. This represents a paradigm shift from extant programmatic advertising approaches that focus on maximizing ad revenue and consider user utility only through click probabilities. The approach described in this article treats the problem of MRS in platform settings as a postprocessing optimization step to be applied after user utilities have been estimated by a traditional RS. Thus, our approach is not dependent on any particular RS

TABLE 4
Evaluation of Recommendation Ranking Produced by the Proposed Algorithm for Different Percentages of Sponsored Items, Measured in *n*DCG

γ	*n*DCG 10%	*n*DCG 20%	*n*DCG 30%	*n*DCG 40%	*n*DCG 50%
0.001	0.977	0.975	0.975	0.975	0.975
0.1	0.977	0.975	0.975	0.975	0.975
0.2	0.978	0.975	0.975	0.976	0.976
0.3	0.978	0.975	0.976	0.977	0.976
0.4	0.978	0.976	0.978	0.978	0.978
0.5	0.979	0.977	0.982	0.982	0.982
0.6	0.981	0.980	0.986	0.986	0.986
0.7	0.984	0.984	0.990	0.990	0.990
0.8	0.987	0.989	0.993	0.994	0.994
0.9	0.992	0.996	0.997	0.998	0.998
1	1.000	1.000	1.000	1.000	1.000

Note. *n*DCG = normalized discounted cumulative gain.

and can be widely applied. Our empirical example suggests that shifting some weight to user utility will increase user utility substantially while decreasing ad revenue only slightly. Thus, we attempt to make programmatic advertising more user-centric by explictly trading off ad revenue to improve user utility while recommending sponsored items.

Recommendations for Future Research

Although our model and algorithm make important contributions to the fields of advertising and RS, we believe this work is only the beginning of a much larger area for future research. We therefore provide an extended discussion of open research questions. First, our model should be tested on data sets from different industries and with field tests. We need to develop a better understanding of how often platforms can increase user utility substantially while reducing ad revenue by a small amount and where to set the value of γ to manage the trade-off.

Second, we need to understand the relationship between user utility and consumer 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). This would mean that γ could potentially become a variable that is selected by the optimization algorithm instead of being set by the manager. Game theory (Sayedi, Jerath, and Srinivasan 2014) could also provide relevant insights as it is often used to model the stability of a system.

Third, we need a better understanding of how to handle situations where a user has high user 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 one hand, showing such a label can decrease the probability of purchase (Beel, Langer, and Genzmehr 2013), and the user might select a nonsponsored 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.

Fourth, note that a_{ui} is really a function of other factors. Advertising 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 by bid order. The ranking function is Bid $\times$ $P(\text{click}|Ad) =$ expected revenue. For example, A is a spammer $P(\text{click}|A) = 0.001$; B is a real company $P(\text{click}|B) = 0.1$. A would have 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 becomes very difficult to model under these circumstances.

Fifth, one could develop algorithms for when items have different "margins." For example, a grocery retailer might make more profit from selling a private-label product and thus recommend it as a nonsponsored item over other items. In this case we 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:

TABLE 5
Example Showing Difficulty of Online Problem

User	Item	c_{ui}
1	A	3
1	B	2
2	A	5
2	B	2

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

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

Seventh, we need to study the 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, each of which is sponsored and with enough budget for exactly one exposure today (see Table 5 for criterion values). Suppose that user 1 visits the platform first, and so showing item A is optimal because $c_{1A} = 3 > 2 = c_{1B}$. Next, user 2 visits, but because 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 offline 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 (Leskovec, Rajaraman, and Ullman 2014), and methods such as the Balance algorithm have been proposed (Kalyanasundaram and Pruhs 1996). Perhaps Balance could be used in the sponsored recommendation situation after substituting c_{ui} for click-through rates, but the details should be carefully thought out.

Eighth, work could be done to improve the efficiency of the algorithm. The example from Table 5 also illustrates why the sorting in the algorithm 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 suboptimal solution. While sorting has superlinear 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 (mn) of user–item c_{ui} values. If the inventory is, say, 10% of mn, one could drop all values less than, say, the 85th 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, say, chi-squared with 3 *df*. The mean, variance, and other moments could be found with a single pass through the list, for example, while computing c_{ui}.Using quantiles from the chi-squared distribution would give the quantiles without sorting, which could then be used to drop user–item pairs with small c_{ui} prior to sorting.

Implications for Advertising

Over the past decade many platforms (e.g., Google, Facebook) have grown quickly to become very large companies, dominant in their industries. Advertising budgets are shifting from traditional media to platforms, which are growing quickly. We point out two features of platforms that have broader implications for the future of advertising. First, platforms are data-rich environments that enable precise targeting and personalization. Consequently, ads will become more like RS because they are moving into these platforms. Second, platforms are successful because they become the first destination for users by providing value: There is an overabundance of choice of digital "items" (information, goods, services, etc.), and the platform helps users find the items that are most relevant, across providers. They receive financial payments from providers for referrals, which could be sponsored. While these revenue streams are attractive, platforms should remember that the heart of their value proposition is being the first destination for users (Malthouse et al. forthcoming), which requires offering something of value to them. Our MRS can help manage this trade-off.

The fields of advertising and RS have developed independently of each other, 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 (see also Jones 1990). The ultimate goal—selling a product or, more generally, increasing CLV—may be the same, but the approaches are completely different. 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.

NOTES

1. We do not allow $\gamma = 0$ because then the algorithm would have no way of allocating nonsponsored 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 nonsponsored 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 $ms_{\max}$ slots.

2. 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.
3. Cosine similarity is one of the most commonly used measures of the similarity between two users. If we think of each user's ratings as n-vector, it is the cosine of the angle between them. When user ratings are centered to have mean 0 for each user, cosine similarity is the Pearson correlation between the row vectors. Values near 1 indicate highly similar users, while values near 0 mean low similarity.

REFERENCES

Abdollahpouri, Himan, Gediminas Adomavicius, Robin Burke, Ido Guy, Dietmar Jannach, Toshihiro Kamishima, Jan Krasnodebski, and Luiz Pizzato (2019), "Beyond Personalization: Research Directions in Multistakeholder Recommendation," preprint arXiv: 1905.01986.

Abdollahpouri, Himan, Robin Burke, and Bamshad Mobasher (2017), "Recommender Systems As Multistakeholder Environments," in *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, New York: ACM, 347–348.

Adomavicius, Gediminas, and Alexander Tuzhilin (2005), "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 734–749.

Adomavicius, Gediminas, and Alexander Tuzhilin (2011), "Context-Aware Recommender Systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. Kantor, eds., Boston, MA: Springer, 217–253.

Azaria, Amos, 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*, New York: ACM, 121–128.

Beel, Joeran, Stefan Langer, and Marcel Genzmehr (2013), "Sponsored vs. Organic Recommendations and the Impact of Labeling," in *International Conference on Theory and Practice of Digital Libraries*, Berlin: Springer, 391–395.

Ben-Porat, Omer, and Moshe Tennenholtz (2018), "A Game-Theoretic Approach to Recommendation Systems with Strategic Content Providers," in *Advances in Neural Information Processing Systems*, 1110–1120. Q4

Blattberg, Robert C., Edward C. Malthouse, and Scott A. Neslin (2009), "Customer Lifetime Value: Empirical Generalizations and Some Conceptual Questions," *Journal of Interactive Marketing*, 23 (2), 157–168.

Breese, John S., David Heckerman, and Carl Kadie (1998), "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, 43–52.

Bryce, Renée C., and Charles J. Colbourn (2007), "The Density Algorithm for Pairwise Interaction Testing," *Software Testing, Verification, and Reliability*, 17 (3), 159–182.

Burke, Robin D., and Himan Abdollahpouri (2017), "Patterns of Multistakeholder Recommendation," in *Value-Aware and Multistakeholder Recommendation Workshop 2017*, http://arxiv.org/abs/1707.09258

———, Himan Abdollahpouri, Bamshad Mobasher, and Trinadh Gupta (2016), "Towards Multi-Stakeholder Utility Evaluation of Recommender Systems," in *CEUR Workshop Proceedings: UMAP 2016 Extended Proceedings*, 1618, http://ceur-ws.org/Vol-1618/SOAP_paper2.pdf.

Edelman, Benjamin, Michael Ostrovsky, and Michael Schwarz (2007), "Internet Advertising and the Generalized Second-price Auction: Selling Billions of Dollars Worth of Keywords," *American Economic Review*, 97 (1), 242–259.

Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan (2011), "Collaborative Filtering Recommender Systems," *Foundations and Trends in Human–Computer Interaction*, 4 (2), 81–173.

Fawcett, Tom (2006), "An Introduction to ROC Analysis," *Pattern Recognition Letters*, 27 (8), 861–874.

Hagiu, Andrei, and Julian Wright (2015), "Multi-Sided Platforms," *International Journal of Industrial Organization*, 43 (C), 162–174.

Henseler, Jörg (2017), "Bridging Design and Behavioral Research with Variance-Based *Structural* Equation Modeling," *Journal of Advertising*, 46 (1), 178–192.

Jones, John P. (1990), "Advertising: Strong Force or Weak Force? Two Views an Ocean Apart," *International Journal of Advertising*, 9 (3), 233–246.

Kalyanasundaram, Bala, and Kirk R. Pruhs (1996), "An Optimal Deterministic Algorithm for Online B-Matching," in *International Conference on Foundations of Software Technology and Theoretical Computer Science*, Berlin: Springer, 193–199.

Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman (2014), *Mining of Massive Datasets*, Cambridge, United Kingdom: Cambridge University Press.

Malthouse, Edward, Alex Buoye, N. Line, D. El-Manstrly, T. Dogru, and J. Kandampully (forthcoming), "Beyond Reciprocal: The Role of Platforms in Diffusing Data Value across Multiple Stakeholders," *Journal of Service Management*.

-- -- --, Ewa Maslowska, and Judy U. Franks (2010), "Accounting for the Long-Term Effects of a Marketing Contact," *Expert Systems with Applications*, 37 (7), 4935–4940.

-- -- --, -- -- --, and -- -- -- (2018), "Understanding Programmatic TV Advertising," *International Journal of Advertising*, 37 (5), 769–784.

Panniello, Umberto, Shawndra Hill, and Michele Gorgoglione (2016), "The Impact of Profit Incentives on the Relevance of Online Recommendations," *Electronic Commerce Research and Applications*, 20 (C), 87–104.

Perlich, Claudia, 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*, New York: ACM, 804–812.

Ribeiro, Marco T., Anisio Lacerda, Adriano Veloso, and Nivio Ziviani (2012), "Pareto-Efficient Hybridization for Multi-Objective Recommender Systems," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, New York: ACM, 19–26.

Ricci, Francesco, Lior Rokach, and Bracha Shapira (2015), "Recommender Systems: Introduction and Challenges," in *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira, eds., Boston, MA: Springer, 1–34.

Rodriguez, Mario, Christian Posse, and Ethan Zhang (2012), "Multiple Objective Optimization in Recommender Systems," in *Proceedings of the Sixth ACM Conference on Recommender Systems*, New York: ACM, 11–18.

Sayedi, Amin (2018), "Real-Time Bidding in Online Display Advertising," *Marketing Science*, 37 (4), 553–568.

———, Kinshuk Jerath, and Kannan Srinivasan (2014), "Competitive Poaching in Sponsored Search Advertising and Its Strategic Impact on Traditional Advertising," *Marketing Science*, 33 (4), 586–608.

Sürer, Özge, Robin Burke, and Edward C. Malthouse (2018), "Multistakeholder Recommendation with Provider Constraints," in *Proceedings of the 12th ACM Conference on Recommender Systems*, New York: ACM, 54–62.

Verschuren, Piet, and Rob Hartog (2005), "Evaluation in Design-Oriented Research," *Quality and Quantity*, 39 (6), 733–762.

Wu, Wush, Chi-Hsuan, 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*, New York: ACM, 1305–1314.

Zheng, Yong (2017), "Criteria Chains: A Novel Multi-Criteria Recommendation Approach," in *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, New York: ACM, 29–33.

Zihayat, Morteza, Anteneh Ayanso, Xing Zhao, Heidar Davoudi, and Aijun An (2019), "A Utility-Based News Recommendation System," *Decision Support Systems*, 117, 14–27.