Multistakeholder Recommendation with Provider Constraints

Özge Sürer
Northwestern University
Evanston, Illinois
ozgesurer2019@u.northwestern.edu

Robin Burke
DePaul University
Chicago, Illinois
rburke@cs.depaul.edu

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

ABSTRACT

Recommender systems are typically designed to optimize the utility of the end user. In many settings, however, the end user is not the only stakeholder and this exclusive focus may produce unsatisfactory results for other stakeholders. One such setting is found in multisided platforms, which bring together buyers and sellers. In such platforms, it may be necessary to jointly optimize the value for both buyers and sellers. This paper proposes a constraint-based integer programming optimization model, in which different sets of constraints are used to reflect the goals of the different stakeholders. This model is applied as a post-processing step, so it can easily be added onto an existing recommendation system to make it multistakeholder aware. For computational tractability with larger data sets, we reformulate the integer problem using the Lagrangian dual and use subgradient optimization. In experiments with two data sets, we evaluate empirically the interaction between the utilities of buyers and sellers and show that our approximation can achieve good upper and lower bounds in practical situations.

KEYWORDS

Multistakeholder Recommendation; Constraint-based Recommendation; Multisided Platforms

ACM Reference format:
Özge Sürer, Robin Burke, and Edward C. Malthouse. 2018. Multistakeholder Recommendation with Provider Constraints. In *Proceedings of Twelfth ACM Conference on Recommender Systems, Vancouver, BC, Canada, October 2–7, 2018 (RecSys '18),* 9 pages.
https://doi.org/10.1145/3240323.3240350

1 INTRODUCTION

Most recommendation systems generate recommendations to maximize the utility for the end user. The system knows previous purchases or ratings of some items and uses this information to estimate the utility of other items that the user has not already seen. Items with large utilities are then recommended in short recommendation lists that help the user find the most useful ones.

A *multisided platform (MSP)* is a business concern that helps parties in a transaction find and do business with each other, reducing transaction costs. Examples of MSP systems have been proliferating rapidly, including:

- Buyers / sellers: Amazon Marketplace, JD.com, Etsy, eBay, Tienmao, Taobao, Rakuten, ShopRunner.
- Viewers / venues / events: Fandango, Ticketmaster.
- Riders / drivers: Uber, Lyft.
- Hosts / guests: AirBnB, VRBO
- Users / advertisers / content developers: Facebook, Tumblr, and other social media applications.

Multisided platforms (MSPs) require new kinds of recommendation systems, because a purely user-centered approach does not recognize the needs and concerns of all parties [7].

The term *multistakeholder recommender system (MRS)* refers to recommendation designs that incorporate the interests of parties in addition to end-users [1, 6]. We can divide the stakeholders of a given recommendation system into three categories: consumers, providers and platform or system [5]. Consumers expect recommendations to match their preferences; providers supply items and gain from consumers' choices. The platform brings consumers and providers together and maximizes its utility if the matching is done successfully. While the MRS problem has been conceptualized, to the best of our knowledge there is no known general solution.

A multistakeholder design allows a recommender system for an MSP to consider the utility of multiple stakeholders and manage related objectives. Note that the objectives of different stakeholders in an MSP may conflict. In the case of a multisided retail platform, the objective for consumers is to find items that maximize their utility, while the objective for a specific seller is to maximize its orders, which are stimulated by being recommended to many consumers. The best result for the seller (many recommendations of its products to many users) may result in low utility for consumers, who may stop visiting the site if too many irrelevant recommendations are received. If only popular sellers are recommended, lower profile ones may leave the MSP network, due to receiving too few new customer referrals. Managing recommendations for an MSP is therefore a complex task that requires recommending items that have high consumer utility and, at the same time, recommending sellers sufficiently often so that they see value in being part of the MSP network.

The approach described in this paper treats the problem of multistakeholder recommendation in MSP 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 algorithmic approach to recommendation generation, and can be widely applied.

We formulate the problem as a constrained optimization. We want to determine the top-k items for each consumer that also satisfy certain provider constraints. This requires solving a large integer programming problem that would be computationally too difficult to solve for realistically large recommendation data sets. We avoid the computational complexity of integer programming

RecSys '18, October 2–7, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5901-6/18/10…$15.00
https://doi.org/10.1145/3240323.3240350

by proposing Lagrangian relaxations of the original optimization problem, which can then be solved with subgradient methods. The approach is tested empirically on the well-known MovieLens data set and on a proprietary data set from the multisided e-commerce platform ShopRunner.

2 PRIOR WORK

Multistakeholder recommendation can be traced back to research into reciprocal recommender systems [9]. In a reciprocal recommender, the success of a particular recommendation depends on both parties' satisfaction with it. We can consider these systems as two-sided platforms where both sides are looking for each other. Online dating sites or buyers and sellers on eBay are examples of such systems. However, reciprocal recommenders differ from MSPs in that they are generally designed to support the consummation of a single matching transaction in which each individual finds the ideal counterparty. We see this constraint in the case of online dating sites, where each person has limited availability and one person should not be recommended to too many others. The objectives of the parties in a reciprocal system are similar, if not identical. In an MSP, the patterns of successful interaction may be quite varied and the objectives of the parties can differ greatly.

One way to include goals of different parties into an MRS is to use multi-objective optimization. In the literature, most studies focus on multiple user-oriented objectives such as accuracy, diversity and novelty. Ribeiro et al. [11] consider combining different recommendation algorithms to deal with the compromise between accuracy, novelty and diversity. They use a genetic algorithm to find the best possible weights. In content recommendation, Agarwal et al. [2, 3] consider the total number of clicks and total time spent as multiple objectives. They try to find the best trade-off between losing clicks and gaining engagement by using constrained optimization. Jambor and Wang propose a constrained linear optimization model for each user separately to promote the long tail items along with user preference scores [8]. In their paper, they consider including external factors into the system as constraints as well. However, they do not consider the interaction between consumers and providers.

There are other studies incorporating different perspectives into MRS. Rodriguez et al. [12] consider both user and recruiter sides to increase the utility of talent recommendation systems. They want a recommended candidate to be a good match for the job and open to a new job at the same time. They modify the semantic score indicating the relevance of a job and a candidate by incorporating job-seeking intent into their objective function. Akoglu and Faloutsos [4] develop an integer programming model for each user using proximity and gain as a score to consider the user and vendor side separately. However, in MSPs, since consumers and providers are interdependent parties, separating recommendation problems for each customer may result in infeasibility of provider constraints.

An MSP should consider multiple goals from different parties simultaneously since there is a network effect among the various parties in the system [7]. Our approach uses constraints for customers and providers at the same time to maximize the utility of the system. To our knowledge, this is the first paper integrating provider constraints into a mathematical model to increase the utility of MSPs.

3 OPTIMIZATION MODELS

There are many applications of recommendation systems in e-commerce platforms. This paper focuses on a personalized mass promotion, for example, via email. The platform sends a message to consumers with a short personalized list of recommended items or of recommended sellers. For example, the e-commerce platform might be sending shipping lists to all customers who purchased today, or be sending a routine email promotion to its database of customers. This setting has the conceptual benefit that all recommendations are generated simultaneously. We will assume, without loss of generality, that the MSP exists to match consumers with different retailers (a specific example of a provider) within its network, and the recommendations of specific retailers to users. Such systems should consider multiple goals from different parties.

One way to solve multi-objective problems is to write multiple objectives as a single-objective optimization problem by taking linear combinations of different objectives. However, when multiple stakeholders exist with conflicting objectives, we may want to manage their goals via restricting and prioritizing them through constraints. A single-objective optimization problem does not help in managing the conflicts, so we use a constrained optimization framework with multiple objectives.

Let $U = \{1, \dots, m\}$ be a set of users and $I = \{1, \dots, n\}$ be a set of items. We have a set of retailers $R = \{1, \dots, t\}$ such that, for each $i \in I$, $i \in I_r$ means that item i belongs to the catalog of retailer r. We assume that the user-side utility for each item i that can potentially be recommended to user j has already been estimated as $\hat{u}_{ij}$. If we want to recommend a list of k items to maximize the utility of end-users, we can choose the k items with the highest utility $\hat{u}_{ij}$ for each user. This is a single-sided recommendation solution, looking only at the benefit to the user.

We will assume for the purposes of this research that the system is also trying to enhance the distribution of recommendations across retailers. That is, a set of recommendations with greater coverage of the retailer base is preferred over a low-coverage set that focuses only on a small subset of the retailers. The low-coverage case will be undesirable for most retailers because promotional recommendations have the chance to increase their sales and to acquire new customers. We explore different formulations of this coverage-based objective in the models that follow.

We model the recommendation output via a decision variable x_{ij} where $x_{ij} = 1$ if item i is included in the top k list for user j. The matrix of x_{ij} variables for all users and items determines which recommendations will be delivered in the email promotion. The following mathematical model (M1) includes a constraint that we can use to specify a minimum fraction of recommended items from each retailer across the recommendation set:

$$\max_{x} \quad \sum_{j \in U} \sum_{r \in R} \sum_{i \in I_r} \hat{u}_{ij} x_{ij},$$

subject to

$$\sum_{r \in R} \sum_{i \in I_r} x_{ij} = k \qquad \forall j \in U, \qquad (1)$$

$$\frac{\sum_{j \in U} \sum_{i \in I_r} x_{ij}}{\sum_{j \in U} \sum_{r \in R} \sum_{i \in I_r} x_{ij}} \geq \alpha_r p_r \qquad \forall r \in R, \qquad (2)$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in I, j \in U, \qquad (3)$$

where p_r is the lower bound of the percentage of items recommended from retailer r and α_r is a nonnegative tuning parameter that determines how precisely this lower bound must be met.

Constraint (1) ensures that each user receives exactly k recommendations. Constraint (2) guarantees that at least a predefined percentage of the total number of recommended items belongs to the catalog of retailer r. Since each user receives k recommendations, the total number of recommended products equals $\sum_{j \in U} \sum_{r \in R} \sum_{i \in I_r} x_{ij} = mk$. Therefore, for each retailer $r \in R$, we can rewrite Constraint (2) as $\sum_{j \in U} \sum_{i \in I_r} x_{ij} \geq \alpha_r c_r$, where $c_r = p_r mk$, to provide a lower bound for the number of recommended products from each retailer r. Constraint (3) specifies that this is a binary integer programming problem — each item is either recommended to a given user or not.

Delivering more balanced recommendation sets comes at the cost of lower overall utility. Given that retailers have different sized inventories, it may be impractical to set a uniform limit of recommended items across all retailers. For example, if there are 100 retailers, it might not make sense to require that each retailer gets 1% of the total recommendation volume. Some of them may have niche markets, where the expected fraction of recommendations would be smaller than retailers with broader inventories. Thus, the model has retailer-specific p_r values. The higher these values are set, the more the system may have to deviate from the optimum end-user utility. The tuning parameter α_r interacts with the retailer-specific lower bound value to control how strictly the optimization adheres to the recommendation volume limits. The α_r values can be adjusted to promote different types of retailers, for example, niche retailers or recent platform entrants. (See experiments below.)

M1 sets constraints on the aggregate utility of a retailer over all recommendation lists. A given retailer r will have its items recommended at least $\alpha_r c_r$ times. However, it does not make any guarantees about the distribution of these recommendations. They may be concentrated on a few users and the constraint would still be met. This might not be acceptable to retailers who are interested in reaching a diverse group of consumers. Our next model M2 therefore adds the constraint that a minimum number of retailers should be included in each recommendation list.

The list-wise model (M2) is below. It includes a new binary variable y_{rj} indicating whether the top-k items for customer j include any items of retailer r or not. Establishing a lower bound for the sum of y_{rj} values across all users ensures a minimum degree of retailer diversity in each list:

$$\max_x \quad \sum_{j \in U} \sum_{r \in R} \sum_{i \in I_r} \hat{u}_{ij} x_{ij},$$

subject to

$$\sum_{r \in R} \sum_{i \in I_r} x_{ij} = k \qquad \forall j \in U, \qquad (4)$$

$$\sum_{j \in U} \sum_{i \in I_r} x_{ij} \geq \alpha_r c_r \qquad \forall r \in R, \qquad (5)$$

$$\sum_{i \in I_r} x_{ij} \leq K y_{rj} \qquad \forall j \in U, \forall r \in R, \qquad (6)$$

$$\sum_{i \in I_r} x_{ij} \geq 1 - K(1 - y_{rj}) \qquad \forall j \in U, \forall r \in R, \qquad (7)$$

$$\sum_{r \in R} y_{rj} \geq w \qquad \forall j \in U, \qquad (8)$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in I, j \in U, \qquad (9)$$

$$y_{rj} \in \{0, 1\} \qquad \forall r \in R, j \in U, \qquad (10)$$

where K is a large constant and w is the retailer diversity parameter. Here, in addition to constraints existing in M1, Constraints (6–7) ensure that when any item of retailer r is included in the top-k items of customer j, the value of y_{rj} will be equal to one. These constraints determine the relations between the values of y_{rj} and x_{ij}. Constraint (8) guarantees that the number of retailers existing in the recommendations of each customer j should be at least w. For a given customer's top-k list, the diversity of retailers increases with increasing w.

4 SOLUTION APPROXIMATION

M1 and M2 implement possible goals that an MSP might have for its providers and retailers. In M1, we assert a minimum degree of exposure for each retailer via the recommendation algorithm. In M2, we augment that requirement with one to ensure retailer diversity in the lists, so that the recommendations are at least minimally diverse with respect to retailers.

These models are zero-one integer programs. Smaller scale instances of these problems can be solved exactly via integer programming using various off-the-shelf tools. However, because the computational effort to solve them becomes exponentially larger with an increasing number of users and items, approximations may be necessary for practical applications. In this section, we provide Lagrangian relaxations of the models and generate a lower and upper bound for the optimal solution of the original problem using subgradient optimization.

4.1 Lagrangian Relaxation

We define Lagrangian relaxations of problems M1 and M2 with respect to Constraints (2) and (5) respectively by introducing a Lagrangian multiplier for each retailer $r \in R$. These constraints are complicating constraints and Lagrangian relaxations of M1 and M2 are constructed by multiplying each of those constraints with λ_r and then bringing them into the objective function. For fixed multipliers $\lambda_r \geq 0 \quad \forall r \in R$, the optimal value of the Lagrangian upper bound problem (LUBP) provides an upper bound to the original optimal objective.

The Lagrangian upper bound problem (LUBP1) for M1 after re-organizing the objective function is:

$$\max_x \quad \sum_{j \in U} \sum_{r \in R} \sum_{i \in I_r} (\hat{u}_{ij} + \lambda_r) x_{ij} - \sum_{r \in R} \lambda_r \alpha_r c_r,$$

subject to

Constraints (1) and (3). (11)

We can perform a similar relaxation of M2 over Constraint (5). The objective is the same as LUBP1, giving us the Lagrangian upper bound problem (LUBP2) below:

$$\max_{x} \sum_{j \in U} \sum_{r \in R} \sum_{i \in I_r} (\hat{u}_{ij} + \lambda_r) x_{ij} - \sum_{r \in R} \lambda_r \alpha_r c_r,$$

subject to

$$\text{Constraints (4), (6)} - \text{(10)}. \tag{12}$$

Both LUBPs can be decomposed allowing for us to solve the problem separately for each user $j \in U$. This is an attractive property for integer programming because, for fixed multipliers, the relaxed problem decomposes into m independent subproblems, each of which can be solved easily. Once we solve each subproblem, the sum of objectives provides an upper bound for the original problem.

Derivation of upper and lower bounds as close as possible to the optimum solution is important for combinatorial optimization problems. Each time we solve LUBP1 and LUBP2 we have an opportunity to transform the solution into a feasible solution for the original problem, which provides a lower bound for M1 and M2, respectively. In this paper, we apply a simple greedy heuristic to get a feasible solution to the original problem based on the Lagrangian solution. In order to tighten the lower and upper bounds, we need to update the multipliers in each iteration. We use subgradient optimization to iteratively, and systematically, generate the values of the Lagrangian multipliers [10].

4.2 Subgradient Optimization

We define π as a user-defined parameter satisfying $0 < \pi \leq 2$, and define Z_{UB} and Z_{LB} to be the upper and lower bounds of the optimal solution. The basic procedure is as follows for M1 (M2):

(1) Find an initial feasible solution to M1 (M2) and update Z_{LB}. Choose an initial set of multipliers $\lambda_r \geq 0 \quad \forall r \in R$.
(2) Solve LUBP1 (LUBP2) with the current set of multipliers. Update Z_{UB}.
(3) Define subgradients G_r for the relaxed constraints such that $G_r = \sum_{j \in U} \sum_{i \in I_r} x_{ij} - \alpha_r c_r \quad \forall r \in R$.
(4) Define step size $T = \dfrac{\pi(Z_{UB} - Z_{LB})}{\sum_{r \in R} G_r^2}$.
(5) Modify the solution of LUBP1 (LUBP2) to obtain a feasible solution. Update Z_{LB}.
(6) Update $\lambda_r \quad \forall r \in R$. Go to step (2) and resolve the LUBP1 (LUBP2) with the new set of multipliers.

At each iteration, step size T is updated using upper and lower bounds. Then, multipliers are updated based on the subgradient value of the relaxed constraints for each retailer $r \in R$.

In Step (5), we must modify the LUBP solution to make it feasible, matching the constraints of the original problem. We use a greedy approach to do this, modifying the recommendation lists to produce a feasible solution.

Once we solve LUBP1, Constraint (2) is satisfied for some retailers. For those retailers, we first drop the lowest rated $\sum_{j \in U} \sum_{i \in I_r} x_{ij} - \alpha_r c_r$ items from the recommendations lists to balance the item distribution for the rest of the retailers. While we are removing items of some retailers, we delete some of the recommended items from each customer's list so that they have fewer than K items. Then, for each retailer whose constraint is not satisfied, we randomly pick a customer with fewer than K items and recommend that retailer's item until a customer reaches K items. We continue picking customers until each retailer's constraint is satisfied.

After passing through all the retailers, Constraint (2) is satisfied for each retailer. However, there might be customers with fewer than K items. For these customers, we recommend new items with the highest predicted ratings until they have K items in their lists. At each step, by using this approach, we can obtain a feasible solution for M1 and if the objective value of the new feasible solution is better than the current lower bound value, we can use it for the remaining iterations as a lower bound.

If we apply the same greedy approach to LUBP2, the minimal retailer diversity constraint might not be satisfied. To generate feasible solutions to M2 using the solutions of LUBP2, we use the greedy approach explained above. Then, we identify the customers whose retailer diversity constraint is not satisfied. For those customers, we remove the item of the most frequent retailer from the list and add an item from a retailer not in the list. We continue passing through the retailers until the diversity constraint is satisfied for the given customer, creating a feasible solution to M2.

5 RESULTS

We tested our approach empirically on the MovieLens 100K data set and a proprietary data set from the multisided e-commerce platform ShopRunner. We solve all optimization models optimally using Gurobi Optimizer 7.5.2.

5.1 MovieLens Evaluation

The section has the following aims: (1) to explore the effect of different provider-item distributions, (2) to understand the effect of global and provider-specific α_r values, (3) to observe the effect of milder and stricter α_r values, (4) to understand the effect of diversity parameter w, and (5) to compare the Lagrangian relaxation solutions with the optimal ones.

This section reports our empirical results using the MovieLens 100K data set. We use this database because it is publicly available and well-known in the recommendation system research community. Following the example of [6], we envision a recommendation scenario in which there is a platform that recommends movies to consumers. Different movies are shown at different "providers," which are analogous to retailers. While our example is fictitious, one could think of providers as different websites that stream movies for a fee. Because we are just interested here in understanding the effects of constraints on overall utility and comparing the Lagrangian relaxation solution with the optimal, we assign movies randomly among 50 providers. We assume that each provider has at least 20 movies in its catalog. First, we assign 20 movies randomly to each provider. Then, we assign the remaining movies randomly via a normal and separately, power-law distribution. For the sake of clarity, we index the providers so that $r = 1$ corresponds to the provider at the leftmost tail of the distribution, whereas $r = 50$ indicates the provider at the rightmost tail of the distribution. Because our model is added as a post-processing step to an existing recommendation system and we assume the estimated utilities $\hat{u}_{ij}$ are provided, we

Model	S	α_r design		α_r experimental values		w
		Global/provider-specific?	Level	Normal	Power-law	
M0	$S01$	-	-	-	-	-
M1	$S11$	Global	Mild	$\alpha_1, \ldots, \alpha_{50} = 0.5$	$\alpha_1, \ldots, \alpha_{50} = 0.5$	-
	$S12$	Global	Strong	$\alpha_1, \ldots, \alpha_{50} = 1$	$\alpha_1, \ldots, \alpha_{50} = 1$	-
	$S13$	Provider-specific	Mild	$\alpha_1, \ldots, \alpha_{10} = \alpha_{41}, \ldots, \alpha_{50} = 0.5$ $\alpha_{11}, \ldots, \alpha_{20} = \alpha_{31}, \ldots, \alpha_{40} = 1$ $\alpha_{21}, \ldots, \alpha_{30} = 0$	$\alpha_1, \ldots, \alpha_3 = 0; \alpha_4, \ldots, \alpha_{10} = 0.75$ $\alpha_{11}, \ldots, \alpha_{50} = 1$	-
	$S14$	Provider-specific	Strong	$\alpha_1, \ldots, \alpha_{10} = \alpha_{41}, \ldots, \alpha_{50} = 1$ $\alpha_{11}, \ldots, \alpha_{20} = \alpha_{31}, \ldots, \alpha_{40} = 1.5$ $\alpha_{21}, \ldots, \alpha_{30} = 0$	$\alpha_1, \ldots, \alpha_3 = 0; \alpha_4, \ldots, \alpha_{10} = 0.75$ $\alpha_{11}, \ldots, \alpha_{50} = 1.25$	-
M2	$S21$	Global	Mild	Same as $S11$	Same as $S11$	12
	$S22$	Global	Strong	Same as $S12$	Same as $S12$	12
	$S23$	Provider-specific	Mild	Same as $S13$	Same as $S13$	12
	$S24$	Provider-specific	Strong	Same as $S14$	Same as $S14$	12
M3	$S31$	-	-	-	-	12

Table 1: Design of experiments with MovieLens data set.

do not compare methods of estimating utilities and simply use the original rating matrix so that the predictions are perfect. In order to calculate a lower bound for the number of recommended movies from each provider, we need to calculate p_r values for each $r \in R$ in M1 and M2. In our experiment, we set p_r as the percentage of movies that provider r has in its catalog of movies. We generate a top-20 list for each customer.

As an unconstrained baseline, we define algorithm M0 to assign the highest rated movies to users without considering any provider constraints. This would be the output of the original recommender system unmodified by post-processing, and also corresponds to the optimal solution of M1 without Constraint (2). As defined above, M1 adds the constraint that the number of recommended items from each provider should match the fraction of items that they offer in the catalog as a whole. M1 also defines α_r values, which can be used to selectively relax these constraints for different providers. We experiment with different ways of setting these values. In addition to the settings of α_r values as in M1, we test M2 by setting a lower bound w for the number of providers for each recommendation list. To create an additional comparative model, we modify M2 by excluding Constraint (5). We call this M3. In the optimal solution of M3, at least w movie providers are included in each recommendation list. Because this list-wise diversity constraint is not paired with the participation constraint, the balanced distribution of movies among all providers is not guaranteed.

We consider different scenarios (S) for our models in the experiments, summarized in Table 1. S11 and S12 set a global α value, that is $\alpha_r = \alpha \quad \forall r \in R$. With smaller α values, the effect of a lower bound for each provider on the distribution of recommended movies is milder compared to larger α values. We test two different levels of α values to observe the change in the optimal solution. M1 and M2 allow provider-specific α_r values as well. With global α values, the distribution of recommended movies might become similar to the catalog distribution of movies. It is also possible that MSPs might want to prioritize some of the providers based on business requirements. For example, in the case of a power-law distribution, most movies are associated with a small number of providers. In order to give priority to the long-tail providers, we

S	Normal		Power-law	
	NDCG	$\text{NDCG}_{\text{subgr}}$	NDCG	$\text{NDCG}_{\text{subgr}}$
$S01$	1.000	-	1.000	-
$S11$	0.993	0.986	0.981	0.959
$S12$	0.924	0.909	0.859	0.846
$S13$	0.930	0.904	0.861	0.843
$S14$	0.811	0.805	0.772	0.747
$S21$	0.992	0.986	0.980	0.960
$S22$	0.923	0.561	0.860	0.571
$S23$	0.932	0.777	0.860	0.823
$S24$	0.819	0.566	0.765	0.741
$S31$	1.000	-	0.996	-

Table 2: NDCG values with MovieLens data set.

can increase α_r values of those providers. By tuning the value of α_r for each provider, we can decrease/increase its priority by decreasing/increasing its lower bound, which can be done by setting lower/higher α_r values.

To experiment with promoting lower-volume providers, we create S13 and S14 with two sets of provider-specific α_r values to promote providers in the two tails for normal and the long tail for power-law distribution. As in S11 and S12, we include both mild and strict versions of the constraints. For M2, we use the same settings for α_r as in M1. M2 also has the parameter w used to constrain the number of different providers included in each list. In these experiments with lists of length 20, we set the parameter $w = 12$, ensuring that no more than 40% of the list can come from a single provider. Since the average number of providers is larger than 10 for the optimal solution of the unconstrained case, setting w to smaller values does not help us to explore the effect of this parameter.

Since the MovieLens 100K data set is relatively small, it is possible to compute the optimal integer programming solution to all of the scenarios in Table 1, obtaining NDCG values reflecting the ranking accuracy of the multi-objective optimization. We also apply our subgradient-based approximation, and show the $\text{NDCG}_{\text{subgr}}$ values obtained by this algorithm.

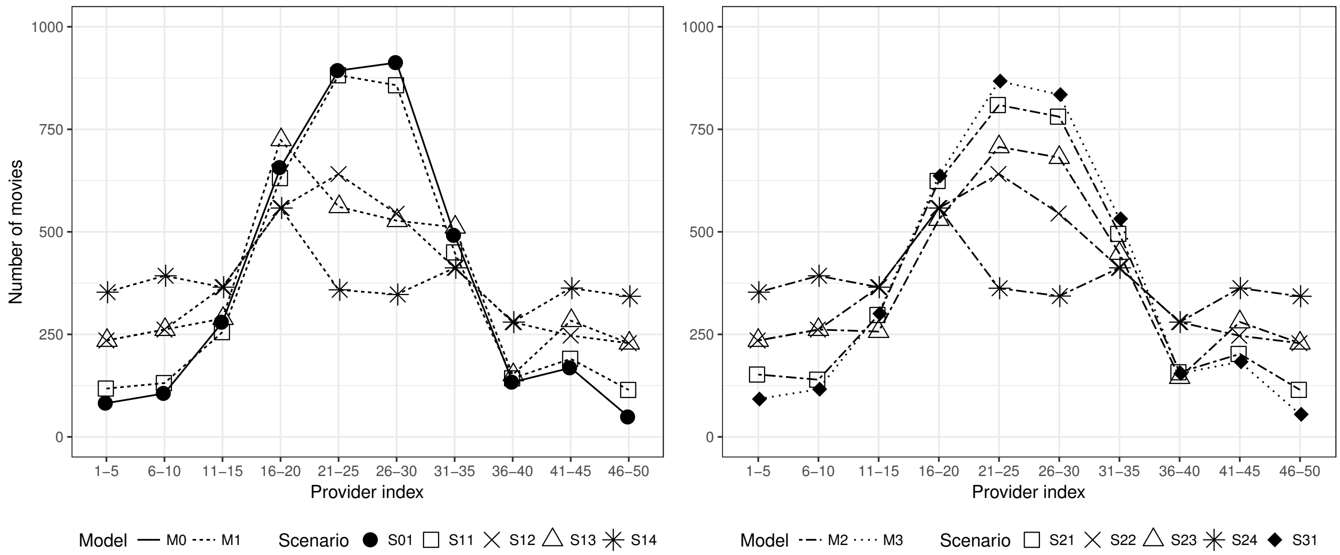

Figure 1: Provider-recommended movie assignment for normal distribution with MovieLens data set.

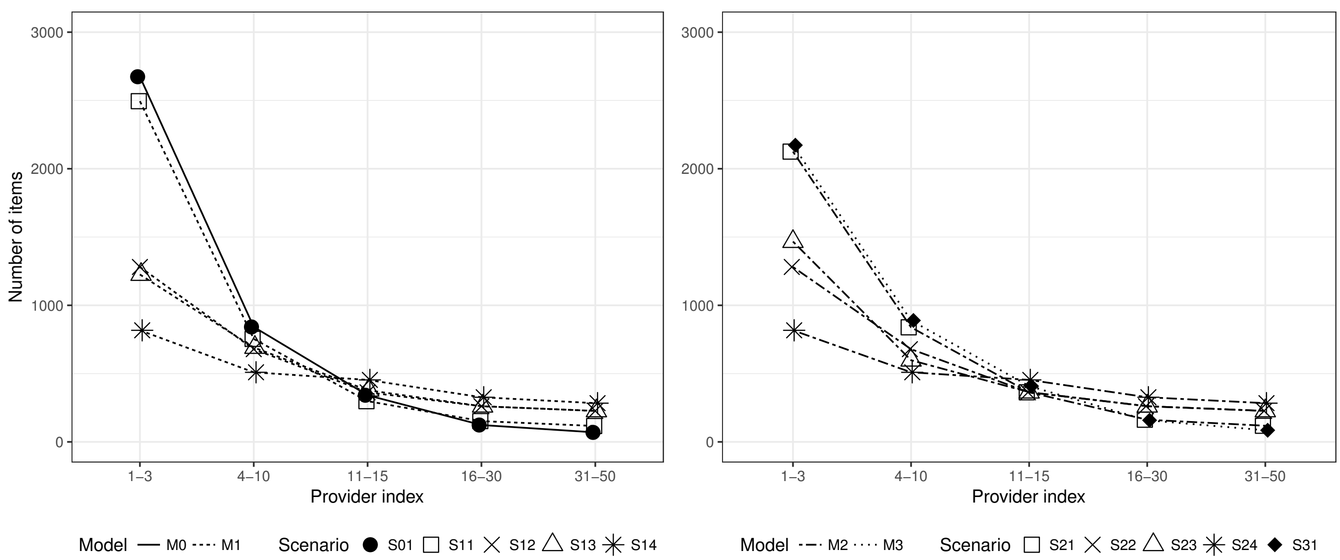

Figure 2: Provider-recommended movie assignment for power-law distribution with MovieLens data set.

As might be expected, the NDCG values are better for the normal distribution compared to the power-law distribution. It gets harder to find higher utility solutions satisfying provider constraints with the highly-skewed power-law distribution. Applying stricter constraints as in S12 and S14 also results in greater loss in ranking accuracy, as the algorithm is forced to include more movies from the less-popular providers.

We also note that the results of M1 and M2 under the same settings of α_r are similar to each other. This implies that M2 with the setting $w = 12$ helps us to eliminate the alternative solutions of M1 that are close to optimal but do not satisfy the minimum provider requirement of M2.

Except in S21, there is a substantial difference in the quality of solutions obtained by the subgradient algorithm across the two models, a loss for trying to satisfy the additional w constraint. Table 3 gives a closer look at these conditions by examining the optimal utility obtained by the exact integer solution (Z^*) and the percentage difference for the lower and upper bounds computed in the approximation. We see that the bounds are quite tight (1–2%) for many of the conditions indicating that the approximation is

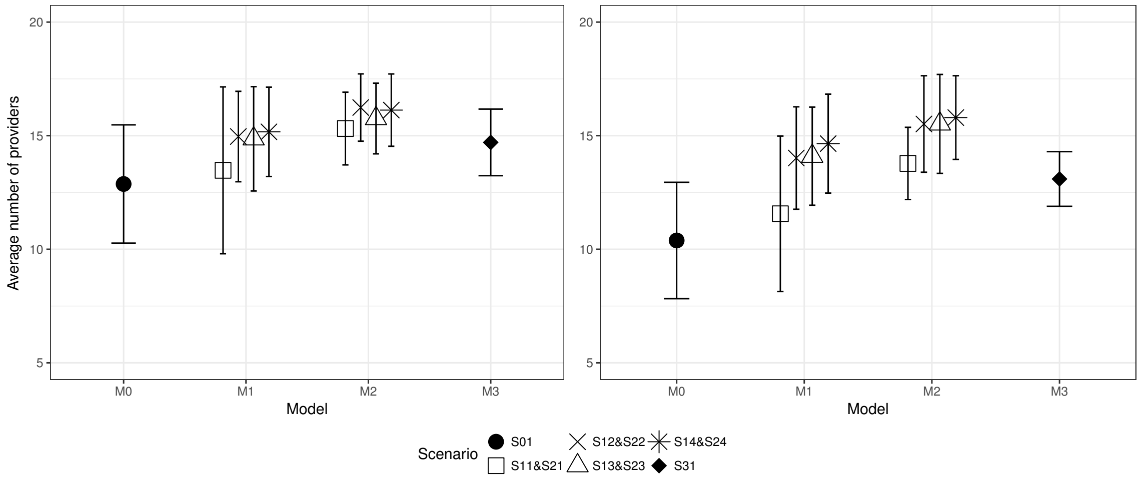

Figure 3: Number of providers for normal and power-law distribution with MovieLens data set.

S	Normal			Power-law		
	Z^*	LB (%)	UB (%)	Z^*	LB (%)	UB (%)
$S01$	85352	-	-	85352	-	-
$S11$	85101	99.9	100.2	84797	99.5	100.4
$S12$	83007	96.8	100.6	81135	97.2	100.1
$S13$	83228	98.8	100.1	81135	98.0	100.0
$S14$	78877	98.2	100.0	77025	97.7	100.1
$S21$	85094	99.8	100.2	84640	98.1	100.1
$S22$	83003	65.8	100.2	81064	66.1	100.2
$S23$	83220	80.0	100.0	81064	97.1	100.1
$S24$	78872	66.5	100.0	77003	97.8	100.3
$S31$	85345			85123		

Table 3: Utility values with MovieLens data set.

achieving a solution quite close to the best possible, but the more difficult cases such as S22 show larger gaps.

Figures 1–2 show how recommended movies are balanced among providers under various scenarios for normal and power-law distributions, respectively. Using the optimal solution in each scenario, we calculate the total number of recommended movies from each provider r. Because of how the labels are assigned to providers neighboring each other in the distribution, we expect that the consecutive providers' assignments to be similar. In these figures, we smooth the distribution by calculating the average of the total number of recommended movies from consecutive providers. The x-axis indicates the provider index range used to calculate the average and the y-axis shows the average of the total number of recommended movies. For example, in Figure 1, in every five consecutive providers, the average number of recommended movies is calculated for the normal distribution. The figures on the left show the results for M0 and M1 under different cases, whereas those on the right show the results for M2 and M3.

For both distributions, the assignments of S11 are similar to S01, the unconstrained case, indicating that milder global α values do not have a strong effect on the distribution of recommendations: the assignments stay approximately proportional to the percentage of movies in the catalog for each provider. Once we increase the value of α in S12, the constraints become tighter compared to S11 and a relatively balanced distribution of movies are obtained. By promoting the providers at the left and right tails for normal and right tail for power-law with milder provider-specific α_r values in S13, we can obtain the similar results with S12. If we want to treat each provider equally, we can use stricter provider-specific α_r values, which flatten the movie distribution for both S14 and S24. S31 of M3 satisfies the minimum number of providers requirement. NDCG and the optimal utility value are nearly the same with the unconstrained case. This set of recommendations perfectly matches with consumers' preference, however, there still exist many providers that are not recommended very often.

Figure 3 displays the average number of different providers in each list under different scenarios with respect to the normal and power-law distribution. Error bars represent one standard deviation from the mean. All cases enable providers to reach a more diverse group of users compared to the unconstrained case S01 of M0. The average number of providers is smaller for the power-law distribution compared to the normal distribution for S01. In general, the average number of providers is larger for scenarios of M2 compared to M1. Setting a diversity threshold parameter w for each customer in M2 helps to increase the average and decrease the standard deviation of the number of providers.

S31 of M3 provides a lower bound for all scenarios created for M2 in terms of the average number of providers. This shows that the provider constraints enabling the balanced distribution of items in M2 help to increase the provider diversity in the lists compared to M3. For M1 and M2, using stricter α_r values increases the average number of providers in the list significantly. Especially, if we use global and relatively milder α values, the average number of providers is lower and the variation is higher compared to the other cases. This implies that low α values do not help too much to ensure heterogeneity in the lists. From Tables 2–3, we can see that NDCG and optimal utility do not change from M1 to M2 for the similar scenario settings. This implies that there are many alternative

α	0	0.2	0.4	0.6	0.8	1
loss (%)	-	1.68	7.70	18.48	38.02	46.34
median	173	146.5	250	374	499	623
min	7	125	250	374	499	623
max	5067	5055	5030	4864	4286	647

Table 4: Comparison of results for different α values for SR.

recommendation sets that provide similar utility for the scenarios of M1. The list-wise constraints in M2 help us to eliminate those solutions that do not satisfy the minimum provider requirement constraints. We can use M2 to prevent an unbalanced distribution of providers among different customers.

5.2 Experiments with Real Data

In this section, we report results using data from the ShopRunner (SR) MSP. SR connects retailers and brands to online shoppers. SR's aim is to help retail partners acquire high-value customers and increase existing customers' purchase frequency. Our data set includes customer purchase history through the website of SR from 2011–2018. They use a recommendation system to recommend retailers, brands and offers to their customers. SR creates value for consumers by giving them seamless access to items across hundreds of retailers and categories, and value to retailers by giving them access to prospective customers. Value to either stakeholder can be enhanced through recommendation.

In this study, we generate a set of $k = 20$ retailer recommendations (picture a 4×5 grid) with a balanced coverage of the retailers. Since we recommend k retailers to each customer, the diversity constraint set in M2 is redundant. We therefore only use M1 to construct the set of recommendations. In M1, I_r usually records the catalog of items for retailer r, but since we are recommending retailers, each I_r contains only the retailer itself. We consider customers who have purchased from at least 20 different retailers to construct an estimated utility matrix. Our data set consists of a random sample of 5982 customers and 192 retailers. We use the SVD decompositional method to construct the estimated utility matrix, with utility calculated as the log number of times each customer purchased from a specific retailer.

For this experiment, we specify that each retailer is equally likely to be recommended. We use the same lower bound of the percentage of recommended retailers, that is $p_r = 1/192 \quad \forall r \in R$. We define a global α value and set $\alpha_r = \alpha \quad \forall r \in R$. We test $\alpha = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and show the results in Table 4. When $\alpha = 0$, it corresponds to the case where we do not consider the provider constraints. The constraints become tighter as α increases.

Table 4 provides the loss in optimal utility value with respect to the case where $\alpha = 0$. Once we obtain the optimal solution for some α, we calculate the total number of customers to whom retailer r is recommended. Then, we obtain the median, minimum and maximum of these values among all retailers. For smaller α values, we can obtain a relatively balanced distribution of items with a small loss in utility. However, when $\alpha = 0.8$ or 1, the loss in utility is larger. This is because the model has more freedom with smaller α values. While the minimum number of times each retailer appears in the lists gets larger with an increasing α, the maximum gets smaller. When $\alpha = 0$, the maximum is 5067, which means that there is a retailer recommended to nearly all of the customers. For $\alpha > 0.2$ the median and minimum values are equal to each other. This indicates that most of the retailers are recommended to satisfy the lower bound defined by the constraints in M1. Once the lower bound for the number of recommended retailers is satisfied, the rest of the recommended retailers are the popular ones, which appear in most of the customers' list. When $\alpha = 1$, the constraints are the tightest. In this case, nearly all of the retailers are recommended equally often, but the loss in utility approaches 50%.

6 DISCUSSION

There are many opportunities for future research on this topic. Further study will be required to improve the bounds in some of the more difficult cases. We also plan to experiment with larger data sets and examine closely the computational tradeoffs of approximation. The proposed method allows for an MSP to set the values of α_r, which determine how much to favor one retailer over another, but there are open questions about how to set these values. How should the α_r values be set to maximize the long-term profits of the MSP? Beyond MSPs, there are other situations involving multiple stakeholders where our constrained-optimization approach may be useful. For example, government funded news sites may have diversity-of-viewpoint requirements, which could be modeled as constraints. Then stories would be recommended to maximize the consumer's utility subject to meeting the diversity regulations.

We are also interested in conditions in which the internal dynamics of the MSP can be used as input to the retailer promotion aspect of our models. We can view the providers in the MSP as a network, where the strength of ties between nodes is determined by the number of customers cross-sold from one to another. As a further study, the provider network structure can be integrated into our mathematical models.

7 CONCLUSION

MSPs are becoming common in many industries to serve the needs of different stakeholders. In the case of retailers, recommendations are designed to appeal to multiple sides by giving consumers access to products across retailers and categories. This study defines a new way of optimizing the dynamics between different sides in MSPs. We propose two mathematical models to recommend items while compromising between customers' and providers' objectives. Our model enables MSPs to adjust each provider's goal individually. As an example, in some of the scenarios we set parameter values to promote the items of long-tail providers; in others, we consider each provider to be equally important. MSPs can adjust parameter values based on their system's requirement, giving them flexibility. To address the tractability of the constraint-based formulation, we propose a method to solve the Lagrangian relaxations of the mathematical models. We relax the constraints that make the problem harder and the remaining problem is easily solvable. We show that under many scenarios a simple greedy heuristic can obtain tight upper and lower bounds, especially for M1.

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] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. 2011. Click Shaping to Optimize Multiple Objectives. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*. ACM, New York, NY, USA, 132–140.

[3] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. 2012. Personalized Click Shaping Through Lagrangian Duality for Online Recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*. ACM, New York, NY, USA, 485–494.

[4] Leman Akoglu and Christos Faloutsos. 2010. Valuepick: Towards a Value-oriented Dual-goal Recommender System. In *2010 IEEE International Conference on Data Mining Workshops*. IEEE, 1151–1158.

[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] Andrei Hagiu. 2014. Strategic Decisions for Multisided Platforms. *MIT Sloan Management Review* 55, 2 (2014), 71.

[8] Tamas Jambor and Jun Wang. 2010. Optimizing Multiple Objectives in Collaborative Filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 55–62.

[9] Luiz Pizzato, Tomek Rej, Thomas Chung, Irena Koprinska, and Judy Kay. 2010. RECON: A Reciprocal Recommender for Online Dating. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, New York, NY, USA, 207–214.

[10] Colin R. Reeves (Ed.). 1993. *Modern Heuristic Techniques for Combinatorial Problems.* John Wiley & Sons, Inc., New York, NY, USA. 243–303 pages. http://dl.acm.org/citation.cfm?id=166648.166660

[11] Marco Tulio Ribeiro, 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 (RecSys '12)*. ACM, New York, NY, USA, 19–26.

[12] 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, New York, NY, USA, 11–18.