Recommender Systems as Multistakeholder Environments

Himan Abdollahpouri, Robin Burke, Bamshad Mobasher
DePaul University
School of Computing
College of Computing and Digital Media
habdolla@depaul.edu, rburke@cs.depaul.edu, mobasher@cs.depaul.edu

Abstract

Recommender systems are typically evaluated on their ability to provide items that satisfy the needs and interests of the end user. However, in many real world applications, users are not the only stakeholders involved. There may be a variety of individuals or organizations that benefit in different ways from the delivery of recommendations. In this paper, we re-define the recommender system as a multistakeholder environment in which different stakeholders are served by delivering recommendations, and we suggest a utility-based approach to evaluating recommendations in such an environment. Our evaluation is capable of distinguishing among the distributions of utility delivered to different stakeholders, and our experiments show that standard algorithms differ in the utility distributions experienced by different classes of stakeholders. Furthermore, we show that using a simple enhancement to baseline algorithms, we can achieve higher utilities for some stakeholders while keeping other stakeholders' utility losses negligible.

1 Introduction

One of the key characteristics of recommender systems is an emphasis on personalization. Recommender systems are evaluated on their ability to provide items that satisfy the needs and interests of the end user, often extrapolated from known ratings or other indicators of interest. Researchers have also examined a variety of metrics (such as accuracy, diversity, novelty and so on) that can be used to measure aspects of the suitability of recommendation results to the target user. However, the end user as the receiver of recommendations is, for the most part, the only consideration for a successful algorithm.

Such focus is entirely appropriate. Users would not make use of recommender systems if they believed such systems were not providing items that matched their interests. Still, it is also clear that, in many recommendation domains, the user for whom recommendations are generated is not the only stakeholder in the recommendation outcome. A number of pertinent examples can be given. Reciprocal recommendation is the term applied to a situation in which a recommendation must be acceptable to both parties in a transaction. For example, in on-line dating, both parties must be interested in order for a match to be successful [14]. Other reciprocal recommendation domains including job seeking, peer-to-peer "sharing economy" recommendation (such as AirBnB, Uber and others), and on-line advertising [8].

It is also important to note that in many e-commerce retail settings, recommendation is viewed as a form of marketing and, as such, the economic considerations of the retailer will also enter into the recommendation function. When recommender systems are evaluated "in the wild", metrics such as engagement, shopping cart size, or other variables more indicative of customer lifetime value are more likely to be used than traditional metrics from recommender systems research. A business may wish to highlight products that are more profitable or that are currently on sale, for example. Commercial recommender systems often use separate "business rules" functionality to integrate such items into the personalized recommendations generated through conventional means. Adding the retailer as a stakeholder allows such considerations to be factored into the recommendation process.

We believe that, far from being special "edge cases", these examples illustrate a more general point about recommendation, namely, that recommender systems serve multiple goals and that a purely user-centered approach does not allow all such goals to enter into the design and evaluation of recommendation algorithms where appropriate. We believe that the scope of recommender system design and evaluation should be broadened to include the perspectives and utilities of multiple stakeholders.

In microeconomics, a similar shift occurred in the early part of the 21st century with the development of the theory of multisided platforms [16, 6]. Prior to that time, the traditional business model focused on a firm's ability to produce products and deliver them to customers at a price that could ensure profitability. This model was inadequate to explain Internet businesses such as search engines, that were giving their products away. Once multisided platform theory was developed, it enabled economists to look back at types of businesses that had existed for many years, such as credit card companies, shopping malls and stock exchanges, and recognize them as examples of multisided platforms as well [5].

The key property of a multisided platform is that it exists to reduce market frictions that may prevent parties from exchanging with each other if left unsupported. A shopping mall, for example, concentrates both retailers and shoppers so that shoppers only have to go to one destination and retailers can benefit from the foot traffic of their neighbors. Today's web economy hosts a profusion of multisided platforms, including such diverse examples as the advertising features of Google's search engine, which brings together searchers and advertisers; OKCupid, which brings together singles looking for dates; Etsy, which brings together shoppers and small-scale artisans; and Kiva.org, which brings together charitably-minded individuals with third-world entrepreneurs in need of capital.

Our contribution in this paper is to explore how recommendation utility can be defined and evaluated for multiple stakeholders. For this purpose, we consider a simplified e-commerce scenario and demonstrate the associated utility calculations. We formulate an algorithm designed to alter the utility distribution among stakeholders and demonstrate its effectiveness.

2 Recommendation scenario

We account for multiple stakeholders by performing explicit, if rough, estimation of the utilities of each party involved in a recommendation transaction. We are not proposing the extraction of personalized utility functions, although this may be worth exploring in future work, but rather simple uniform utility functions for each class of stakeholder: users, suppliers, and the system owner or retailer.

Our experiments below make use of data from the well-known MovieLens 1M dataset [7]. We assume a simplified scenario that differs from the actual MovieLens recommendation context. We treat the recommender as if it were embedded in a pay-per-view video-ondemand site. In this scenario, users are delivered recommendations for movies and can select a movie to watch, paying a fee to do so. The fee paid by the user is split between the studio distributing the movie and the system owner. We will assume for simplicity that all movies are the same price and that the split of revenue is the same for all movies and all distributors. We also assume that such contracts are exclusive – a studio will release all of its movies on a particular site and may, if desired, move them to a competing site.

For example, suppose that a user, v, sits down on her couch to look for a show to watch. On her streaming service S, she is recommended movies i_1, i_2, and i_3 that are distributed respectively by studios d_1, d_2, and d_2: there are two movies from d_2. She chooses to watch movie i_2 and authorizes a payment from her account. From v's point of view, this is a transaction between her and S, but a second transaction is triggered because S pays a per-stream fee to d_2 whenever one of its movies is shown.

In this admittedly-artificial scenario, it is clear that there are three classes of stakeholders, two of which are groups. Users are obviously seeking good recommendations that bring movies of interest to their attention. When they find such movies, they will reward the system by paying to watch. Studios are a second class of stakeholders. Their movies are more likely to be viewed if they are recommended, and an increasing rate of viewing means increased income.

The third class of stakeholder is singular: the system owner. The owner profits directly when movies are watched. However, the owner is also concerned with the long-term health of the business. Studios can take their movies elsewhere and users can subscribe to other services that offer similar viewing experiences. These considerations should be part of the owner's utility function.

A rational system owner seeks to maximize its own utility. However, it may be important to understand how all stakeholders fare under different algorithmic choices. As we will demonstrate in our experiments below, utility distributions for users and studios can be quite different and may be impacted differently by algorithmic changes.

2.1 Defining utility

Off-line evaluation methodology requires that our recommender system be trained on a set of training data and evaluated based on testing data, where both the training and testing data consist of triples associating a user with an item and a corresponding rating. Therefore, we must define the utility of generated recommendations relative to this test data.

The central unit of evaluation in our scenario is the recommendation list. As in our example user above, interaction with the recommender consists of the delivery of recommendation lists. The user obtains utility from such lists in the form of desirable viewing suggestions. For our purposes, we assume that the utility of an item is only a function of its inclusion in the list, not its rank. To make the user utility function personalized, we assume that there is utility for the user when the system recommends items that the user has rated highly.

Studios also obtain utility from each recommendation list. We also assume that studios obtain a fixed utility when their items are recommended, roughly corresponding to the increased probability that a user will select an item from the recommendation list as opposed to seeking it out some other way.

For our purposes, each user v in the test data corresponds to one "visit" to the video streaming site. The number of such visits in the test data is the value N. At each visit, the system generates a recommendation list L of fixed size k. The user obtains utility $u_v(i)$ for each item i: it will be 1 if an item is one that he or she has rated greater than 3 (on a scale from 1 to 5) in the test data. Otherwise, the utility is zero. The total utility of the list is simply the sum of these individual values. We do not intend to compare utilities across different stakeholders, so we can associate an arbitrary fixed value with a recommendation "hit" without loss of generality. Total user utility divided by Nk is the same as the conventional definition of precision@k, so in this sense, aggregate user utility is a familiar metric for recommender system evaluation.

The same recommendation lists also provide utility to the suppliers of the video material. Each supplier d owns a set of items and gains utility u_d from the presence of a liked item i in the recommendation list if they are its distributor. Each supplier receives utility from a recommendation list if an owned video appears in it.

We represent user and supplier utility by two vectors: U_V and U_D:

$$\begin{aligned} U_V &= \langle u_{v_1}, u_{v_2}, ..., v_{|V|} \rangle \\ U_D &= \langle u_{d_1}, u_{d_2}, ..., u_{|D|} \rangle \end{aligned} \tag{1}$$

Users that did not appear in the test data or that were not recommended acceptable items will have zero utility; studios whose movies were never recommended to appropriate users will also have zero utility. In our evaluation, we examine both the total utility and the distribution of utilities.

One could argue that, in a marketing sense, every presentation of an item to the user gains utility for its owner and therefore the user rating of the recommendation is not relevant. However, if utility is calculated in this way, it becomes completely decoupled from user preferences. Because of the sparsity of the rating data, recommendation lists consist mostly of unrated items. It would relatively easy to build a recommendation algorithm that swapped some of these items for random movies from under-performing studios and thus increase studio utility without having a big impact on user utility as we measure it. But a recommendation algorithm that worked in this way would not be perceived as beneficial to users. Therefore, we reward algorithms not for every item included but only those for which we have user ratings.

System utility is proportional to the total studio utility, since the system takes a cut of the video streaming revenue. However, we also wish to represent the system owner's long-term interest in preserving its user base and set of video suppliers. To represent these concerns, we compute two quantities. The first is the fraction of users who received some utility from their recommendation lists:

$$q_V = \frac{|\{v : u_v > 0\}|}{|V|} \tag{2}$$

The second is the fraction of studios receiving some utility:

$$q_D = \frac{|\{d : u_d > 0\}|}{|D|} \tag{3}$$

Finally, the system utility is computed as

$$U_S = q_v q_d \sum_{u_d \in U_D} u_d \quad (4)$$

In other words, we discount the system utility by two factors representing the coverage of users and the coverage of studios. The logic behind this formulation comes from the theory of multisided platforms. The goal of a multisided platform is to maintain communities of buyers and sellers who find the platform useful. If the platform begins to lose buyers or sellers, it can enter a "death spiral" where lower participation by one side triggers less participation by the other side, which lowers its utility and sparks further defections.

Therefore, if good recommendations are delivered only to a small set of users, the system is less useful for other users. Users who dislike the system's recommendations (indicated by zero utility) may migrate to other systems. So, the utility gained over a particular period is discounted by the fraction of users that did not get a satisfactory recommendation list. Similarly, providing recommendations that promote only a small subset of studios may threaten the supplier base. Studios whose movies are not recommended will receive fewer views and may leave to other platforms. So, again we discount, in this case, by the fraction of possible studios gaining utility.

Our formulations of utility above are highly simplified and based on many assumptions about our streaming scenario. In specific applications scenarios, system owners may have much more precise knowledge of the benefits of recommendation and the utility delivered to different stakeholders. In general, the formulation of utility functions is highly-domain dependent and in most cases, dependent also on simplifying assumptions. The aim of this paper is not to present the "right" set of utility calculations for this scenario, but rather to demonstrate that reasonably-formulated utilities can be instructive of the trade-offs between algorithms in ways that conventional metrics might not reveal.

3 CovEn Algorithm

Collaborative recommendation algorithms are designed to optimize solely for users as stakeholders. We would not necessarily expect that such algorithms would maximize system owner utility. For example, the popularity bias in collaborative recommendation is well-known [2, 13]. A recommender with a strong popularity bias might concentrate recommendations on just a few suppliers, resulting in a small q_D fraction and low utility for the system owner.

One way to think of this problem is as a combinatorial joint optimization of all utilities. In theory, there exists some set of recommendation results that optimizes the outcomes of all stakeholders. However, such an algorithm is not practical in a live system. A live system is in the position of compensating at the present time for biases that occurred in prior recommendations. The system does not have the luxury of going back in time to change what recommendation lists were delivered to users in the past. What is called for in a live recommendation setting is an on-line algorithm that tries to achieve balance across suppliers over time [1].

In order to show that our multistakeholder evaluation would give us different insights for different algorithms than standard methods, we propose here a coverage-enhancement algorithm, *CovEn*, that attempts to counter popularity bias at the supplier level. It is a filtering mechanism applied to the recommendation lists returned by the recommender system. Note that our goal here is not to create the best algorithm for increasing coverage, but to demonstrate the value of utility-oriented evaluation and to show how simple algorithmic manipulations may impact the utilities of multiple stakeholder.

To achieve this enhancement of the coverage of the studios, we create a hash map h, keyed by the studio names, counting the number of times a studio's movies have been recommended. After each time period, we update this map so that a running total is maintained in each entry $h(d)$.

When generating recommendation lists for each user, we use the historical coverage information stored in h as follows. First, we generate a recommendation list of length K, where K is larger than k, the desired result set size. (In the experiments below, it is $4k$. So, k is 10 and K is 40.) We expect that this list will be more diverse respect to item popularity than a shorter list. From this list, we remove all items with predicted rating less than or equal to 3. Then, we divide the recommendations into two lists, *seen* and *unseen*: the seen list consists of the items from studios where $h(d) > 0$, and the unseen list has the items from studios where $h(d) = 0$. The popularity bias present in prior recommendations will be reflected in the contents of the *seen* list, so we fill the recommendation list with items from the unseen list first. If there are fewer than k items on the unseen list, then the top items from the seen list are added up to size k. Thus, the algorithm compensates for popularity bias while at the same time including items likely to preferred by the user.

4 Experiments

Our experiments were conducted with three main aims: (1) to apply the multistakeholder approach in our scenario and discover how this approach enhances our understanding of the properties of different algorithms, (2) to demonstrate the application of utility-oriented evaluation metrics across multiple stakeholders, and (3) to see if the simple coverage-oriented algorithm described above can enhance system utility, ideally without significantly harming the other stakeholders.

As an on-line algorithm, CovEn has to be evaluated over time to see how well it is able to compensate for recommendation biases. We therefore adopt a naturalistic time-aware evaluation methodology: as described in [4], community-centered base set, time-dependent rating order, time-based size. The data is divided into time-based epochs, and utility is evaluated on the current epoch, using prior epochs as training data. Thus, CovEn can compensate in one epoch for biases in recommendations given in prior ones.

4.1 Methodology

The Movielens 1M rating dataset contains 1,000,209 ratings of approximately 3,900 movies made by 6,040 MovieLens users who joined MovieLens in 2000 [7]. We trimmed all ratings after December 31, 2000, because the frequency of ratings drops off significantly at this point. We split the remaining data into 50 epochs of approximately 11,000 ratings each. We then removed users that had only ratings in one epoch, as predictions for these users cannot be generated in our training / test paradigm. This resulted in epochs that vary in rating count between about 6,000 and 11,000. User counts across the epochs were fairly similar ranging from 1653 to 1873. The retained data is approximately 32% of the original dataset.[1] The result of this processing was 49 chronological training / test splits, each containing one epoch of test data and a variable number of training epochs.

We gathered the name of the studio releasing each movie from IMDb. In the case of multiple studios being listed, we chose the movie's distributor, which is listed first.

We use Apache Mahout as our recommender system implementation, modified by the inclusion of our CovEn algorithm. For our experiments, we used three baseline recommendation algorithms: user-based KNN, item-based KNN, and SVD. We also augmented each of these with the CovEn algorithm described above.

In each epoch, we record the utility values for all users and all studios. Because the epochs are so short (averaging about a week), it does not make sense to consider the user and studio coverage (and hence the system utility) in each epoch separately. We report instead cumulative system utility at each period.

4.2 Results

As we have indicated, the utility values represent an arbitrary scoring scheme. They represent the interests of the stakeholders in a proportional way, but we make no attempt to relate these abstract utilities to any real-world valuation. The scales are consistent only for each type of stakeholder and therefore cannot be compared with each other.

Figure 1 shows the utility for users and the studios in each epoch over the evaluation period. Note that we use a square-root transformation on the y-axis to be able to visualize results from both the SVD algorithm and the much lower-scoring memory-based alternatives.

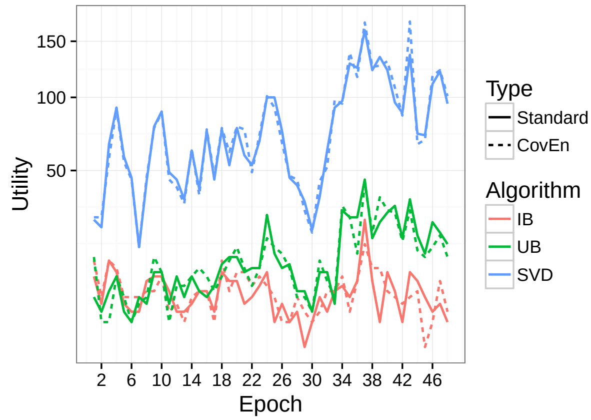
The two figures are quite similar, as might be expected given the similarity in the formulation of the utility measures. Every "hit" for a user is also a hit for the corresponding studio. A studio will also get a hit if a movie is recommended and it is in the test set for that user, but the user's rating is low. The MovieLens data set has a well-known skew towards higher ratings, and the algorithms are designed to avoid recommending disliked items, so these items are relatively rare in the result lists.

The SVD algorithm is the strong winner throughout. One of the key characteristics of the MovieLens data is the relatively short temporal span of rating profiles: very few profiles are found in more than 2 or 3 epochs. Also, the full size of the training set is not achieved until late in the evaluation process. Thus, temporal evaluation presents a classic cold-start recommendation problem and model-based methods such as SVD have well-known advantages in such cases.

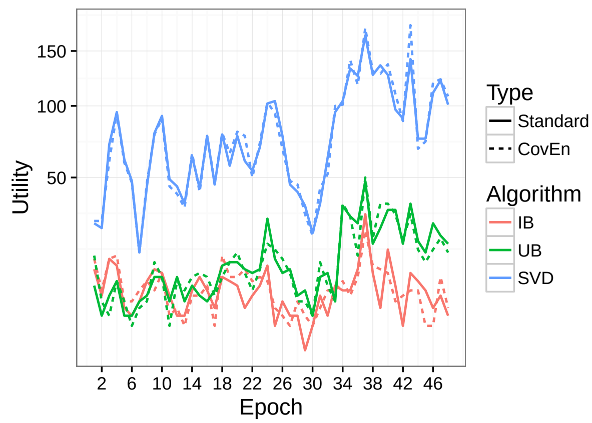
The biggest influence on the shape of these curves is the number of ratings found in each epoch, which can vary by a factor of two. If there are fewer ratings, there will be very small number of possible target items that the system is trying to retrieve. "Difficult" periods can be clearly seen, for example around Epoch 30, where all the algorithms have trouble making good recommendations.

Only very small differences in user and studio utility can be seen between the standard and coverage-enhancement versions of each algorithm. This is not true when we look at the

[1] About half of the MovieLens 1M dataset consists of profiles where all ratings were entered on the same day. These one-shot users greatly reduce the amount of data that can be effectively used for temporal evaluation of this type.



(a)



(b)

Figure 1: Per-epoch utility: (a) user (b) studio

system utility shown in Figure 2, again with a square root transformation on the y axis. (Note: ignore the difference in scale, which, as stated above, is arbitrary and not intended for comparison among the different types of stakeholders.) Particularly in the SVD case, the system utility shows strong gains for the CovEn variants against a high utility baseline. This should not be surprising since the CovEn algorithm was designed specifically to improve system utility.

Table 1 summarizes these utility values. Note that the coverage-enhancement algorithm actually has a positive effect on cumulative utility for all cases, except in the case for users in the user-based algorithm. The item-based algorithm has particularly low utility for all stakeholders, which is somewhat surprising since it has shown greater accuracy than user-based recommendation on the same data in other experiments. We believe this is because the cold-start nature of the temporal evaluation task makes it difficult to put together effective item neighborhoods. We are still exploring the cause of this discrepancy.

A sum over a utility distribution hides possible shifts in the individual values. When there is a gain, it is hard to tell if this is due to a few individuals profiting greatly or a general improvement. Figure 3 (best viewed in color) shows the distribution of utility across all users and all studios. Note that we are using log-log scales here due to the extremely

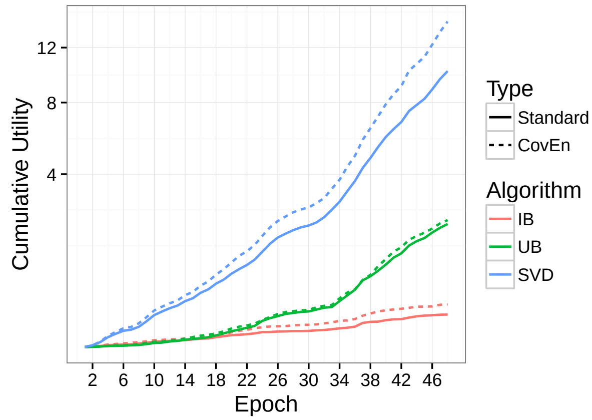


Figure 2: Cumulative system utility

Col1	Col2	Col3	Col4	Col5
Stakeholder	Algorithm	Standard	CovEn	Gain (Loss)
User	UB	632	612	(3.2%)
	IB	242	254	5%
	SVD	3571	3626	1.5%
Studio	UB	671	660	1.6%
	IB	314	322	2.5%
	SVD	3668	3725	1.5%
System	UB	2.02	2.16	6.4%
	IB	0.142	0.24	72.6%
	SVD	10.2	14.2	39.2%

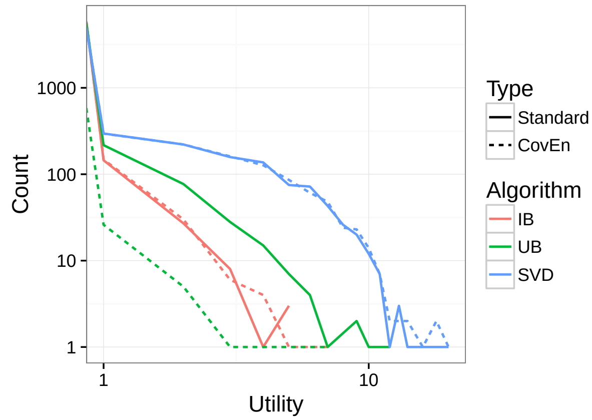
Table 1: Cumulative utility gain (loss) for CovEn algorithm

skewed distributions of utility in these results. The deficiencies of the CovEn variant of the user-based algorithm are clear here, but otherwise the distributions with CovEn are quite similar to the standard baseline. The curves for SVD shows that the coverage-enhancement variant increases very slightly the number of high utility individuals.

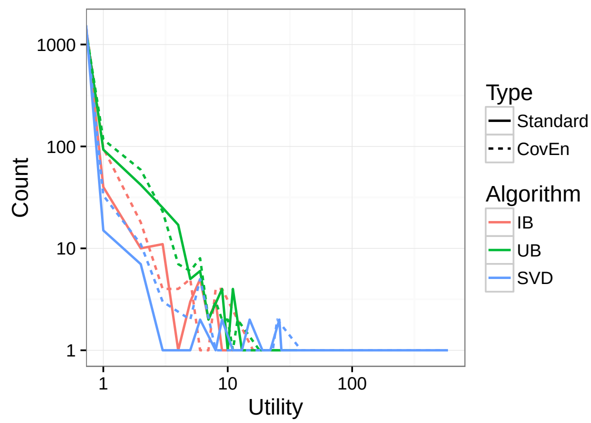
The utility distribution for studios shows an interesting contrast to the users. The popularity bias of SVD is quite evident, with its long tail (high utility for a small number of dominant studios) and its relatively smaller number of mid-range values. This is quite the opposite of what happens with the user distribution. The CovEn variant lifts the distribution slightly across the middle range of utilities, but it remains below the user-based and item-based algorithms.

5 Related work

This research is part of a on-going effort to expand the considerations involved in recommender system evaluation beyond measurements of accuracy. There is a large body of recent work in recommender systems on incorporating diversity, novelty and other metrics into rec-



(a)



(b)

Figure 3: Utility distributions (log-log-scale) for (a) users and (b) studios

ommendation generation and evaluation. See, for example, [18, 22, 19]. Although we do not do so here, our model provides a framework in which such considerations can be explicitly represented as utilities and accounted for in evaluation.

Similarly, multiple utilities may be in play when users' short-term preferences and their long-term well-being may have different associated utility functions. For example, lifestyle recommenders have been developed to encourage users to engage in healthful activities [11, 15]. In such systems, it is important not to recommend items that are too distant from the user's preferences – even if they would maximize health. The goal to be persuasive requires that the user's immediate context and preferences be honored. Although these lifestyle recommenders, to date, have not taken a utility-oriented approach, they can be understood in these terms.

The concept of multiple stakeholders in recommender systems is suggested in a number of prior research works. As discussed above, researchers on reciprocal recommendation have looked at bi-lateral considerations to ensure that a recommendation is acceptable to both parties in the transaction [20]. Similar ideas have appeared in work on group recommender systems where the goal is to find recommendation(s) that can maximize the utility of all users in the group [12].

A more explicit utility-theoretic approach is taken by [17] in which a user's job seeking propensity is combined with their fit for a job description in ranking recruitment candidates in LinkedIn. This paper found that the combined utility yielded higher engagement rates than similarity alone.

There is a substantial literature in real-time targeted advertising in which advertisers' expected revenue and / or available budget are incorporated into the decision to deliver personalized advertising to a user. See [21] for an example. The BALANCE algorithm is designed to achieve balanced budget draw-down in an online advertising setting [10], and served as an inspiration for the CovEn algorithm.

In these advertising settings, the amount of information about each user is generally quite limited and privacy considerations limit how much of it can be shared with advertisers. In addition, the real-time nature of the application and the huge potential user base makes user-level personalization computationally impractical. As a result, ad display is personalized only to a very coarse degree (age and geographic region, for example), if at all.

Finally, Jannach and Adomavicius in their recent work claim that the existing evaluation metrics for recommender systems do not sufficiently measure the quality of recommendations [9]. They proposed some additional evaluation factors such as user engagement, product awareness, etc. as more practical ways for evaluating recommender systems. However, they did not discuss how those metrics should be calculated and they did not conduct experiments.

Our approach is novel in that we explicitly represent the different stakeholders in the recommendation process and formalize their utilities. As discussed in [3], our approach is sufficiently general that a wide variety of recommendation scenarios can be represented including reciprocal recommendation, budget management, and others.

6 Conclusion

There is increasing dissatisfaction with one-dimensional, accuracy-oriented evaluation of recommender systems. In addition, real-world recommendation applications frequently require sensitivity to business needs and context. A utility-oriented approach allows us to represent such concerns explicitly and make clear our modeling assumptions about the relative benefits of different aspects of recommendation outcomes. A multistakeholder approach highlights the multiple actors involved in a given recommender system configuration and allows the concerns of each to be represented and accounted for in evaluation and design.

In this preliminary study, we found that it was possible to formulate a standard recommendation problem as a multistakeholder environment, and to quantify in a rough manner the concerns of the different players. Our experiments, especially the utility distribution results, show very clearly the tension between buyers and sellers relative to the problem of popularity bias. There is almost an exact correlation across algorithms between user utility and the degree of skew in the seller distribution. Our experiments also show that a simple algorithmic change can have a clear benefit to system utility, as we have defined it, with few, if any, negative impacts for other stakeholders.

7 Acknowledgments

This work is supported in part by the National Science Foundation under grant IIS-1423368.

References

[1] Susanne Albers and Stefano Leonardi. "On-line algorithms". In: *ACM Computing Surveys (CSUR)* 31.3es (1999), p. 4.

[2] Erik Brynjolfsson, Yu Jeffrey Hu, and Michael D Smith. "From niches to riches: Anatomy of the long tail". In: *Sloan Management Review* 47.4 (2006), pp. 67–71.

[3] Robin D. Burke et al. "Towards Multi-Stakeholder Utility Evaluation of Recommender Systems". In: *Late-breaking Results, Posters, Demos, Doctoral Consortium and Workshops Proceedings of the 24th ACM Conference on User Modeling, Adaptation and Personalisation (UMAP 2016), Halifax, Canada, July 13-16, 2016.* 2016. URL: `http://ceur-ws.org/Vol-1618/SOAP_paper2.pdf`.

[4] Pedro G Campos, Fernando Díez, and Iván Cantador. "Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols". In: *User Modeling and User-Adapted Interaction* 24.1-2 (2014), pp. 67–119.

[5] David S. Evans and Richard Schmalensee. *Matchmakers: The New Economics of Multisided Platforms*. Harvard Business Review Press, 2016.

[6] David Evans et al. *Platform economics: Essays on multi-sided businesses*. Competition Policy International, 2011. URL: `http://ssrn.com/abstract=1974020`.

[7] F Maxwell Harper and Joseph A Konstan. "The MovieLens Datasets: History and Context". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5.4 (2015), p. 19.

[8] Ganesh Iyer, David Soberman, and J Miguel Villas-Boas. "The targeting of advertising". In: *Marketing Science* 24.3 (2005), pp. 461–476.

[9] Dietmar Jannach and Gediminas Adomavicius. "Recommendations with a Purpose". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM. 2016, pp. 7–10.

[10] Bala Kalyanasundaram and Kirk R Pruhs. "An optimal deterministic algorithm for online b-matching". In: *Theoretical Computer Science* 233.1 (2000), pp. 319–325.

[11] Yuzhong Lin et al. "Motivate: Towards context-aware recommendation mobile system for healthy living". In: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*. IEEE. 2011, pp. 250–253.

[12] Judith Masthoff. "Group recommender systems: Combining individual models". In: *Recommender systems handbook*. Springer, 2011, pp. 677–702.

[13] Yoon-Joo Park and Alexander Tuzhilin. "The long tail of recommender systems and how to leverage it". In: *Proceedings of the 2008 ACM conference on Recommender systems*. ACM. 2008, pp. 11–18.

[14] Luiz Pizzato et al. "RECON: a reciprocal recommender for online dating". In: *Proceedings of the fourth ACM conference on Recommender systems*. ACM. 2010, pp. 207–214.

[15] Victor Ponce et al. “QueFaire: context-aware in-person social activity recommendation system for active aging”. In: *Inclusive Smart Cities and e-Health*. Springer, 2015, pp. 64–75.

[16] Jean-Charles Rochet and Jean Tirole. “Platform competition in two-sided markets”. In: *Journal of the European Economic Association* 1.4 (2003), pp. 990–1029.

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

[18] Barry Smyth and Paul McClave. “Similarity vs. diversity”. In: *Case-Based Reasoning Research and Development*. Springer, 2001, pp. 347–361.

[19] Saúl Vargas and Pablo Castells. “Rank and relevance in novelty and diversity metrics for recommender systems”. In: *Proceedings of the fifth ACM conference on Recommender systems*. ACM. 2011, pp. 109–116.

[20] Peng Xia et al. “Reciprocal Recommendation System for Online Dating”. In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ACM. 2015, pp. 234–241.

[21] Weinan Zhang, Shuai Yuan, and Jun Wang. “Optimal real-time bidding for display advertising”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2014, pp. 1077–1086.

[22] Cai-Nicolas Ziegler et al. “Improving recommendation lists through topic diversification”. In: *Proceedings of the 14th international conference on World Wide Web*. ACM. 2005, pp. 22–32.