RecMax: Exploiting Recommender Systems for Fun and Profit

Amit Goyal
University of British Columbia
Vancouver, BC, Canada
goyal@cs.ubc.ca

Laks V. S. Lakshmanan
University of British Columbia
Vancouver, BC, Canada
laks@cs.ubc.ca

ABSTRACT

In recent times, collaborative filtering based Recommender Systems (RS) have become extremely popular. While research in recommender systems has mostly focused on improving the accuracy of recommendations, in this paper, we look at the "flip" side of a RS. That is, instead of improving existing recommender algorithms, we ask whether we can use an existing operational RS to launch a targeted marketing campaign. To this end, we propose a novel problem called RECMAX that aims to select a set of "seed" users for a marketing campaign for a new product, such that if they endorse the product by providing relatively high ratings, the number of other users to whom the product is recommended by the underlying RS algorithm is maximum. We motivate RECMAX with real world applications. We show that seeding can make a substantial difference, if done carefully. We prove that RECMAX is not only NP-hard to solve optimally, it is NP-hard to even approximate within any reasonable factor. Given this hardness, we explore several natural heuristics on 3 real world datasets – Movielens, Yahoo! Music and Jester Joke and report our findings. We show that even though RECMAX is hard to approximate, simple natural heuristics may provide impressive gains, for targeted marketing using RS.

Categories and Subject Descriptors H.2.8 [Database Management]: Database Applications - *Data Mining*
General Terms: Algorithms, Theory, Experimentation.
Keywords: Recommender Systems, Collaborative Filtering, Targeted Marketing, Seed Set Selection, Maximization.

1. INTRODUCTION

There are many applications like online shopping where the opinions of other users who bought and rated a product are important for a user trying to decide which product to buy. Recommender systems (RS) have emerged as a complementary paradigm for search in order to fulfill users' information needs in such applications, by providing mass personalization [14]. Much of their popularity was spurred by the early success of collaborative filtering techniques [9], which exploit the existing ratings in a system to estimate the expected rating a user might give a product. Today, RS form the backbone of the business of many companies like Amazon and Netflix.

RS research has mostly focused on improving the accuracy with which the algorithm predicts the likely rating a user will provide an item she has not experienced before. The predicted ratings are then used for recommending items to users, e.g., by picking the top-ℓ items with highest predicted rating for the active user. In this paper, we take a look at the "flip" side of a RS. That is, instead of improving existing recommender algorithms, we ask whether we can use an *existing* operational RS to launch a targeted marketing campaign. More concretely, you have a product you want to sell in an online market place powered by RS technology. Suppose you give free samples of your product to a select number of users, called the *seed* users. Based on its quality, the seed users rate the product, preferably with high ratings. E.g., Amazon's Vine program employs a similar scheme. The question is, under the condition that the seed users endorse the product with high ratings, can this lead to the product being recommended by the system to a large number of other users? If so, this represents a huge opportunity for the client companies launching the campaign as it helps effectively advertise their product via the RS. It is a business opportunity for the host company as well since it can offer seeding as a service, whereby it selects effective seeds on behalf of the clients. It is beneficial to the seed users as they get free or discounted samples of a new product. Finally, it is helpful to the rest of the users as they receive recommendations of new products earlier than they otherwise would, a well recognized problem in RS [11].

In this paper, we will show the answer to the above question is "yes" and that the manner in which seeds are selected can make a difference. We focus on the case where the seed users who are targeted like the new product and endorse it with relatively high ratings. As example applications, Warner Brothers can target a select number of Netflix users in order to give free samples of their new movie release. A publisher can target customers of Amazon in a similar way. Motivated by these applications, we study the following problem, called RECMAX: Given a collaborative RS, a number k and a new product ρ, find k users (called *seed set*) to whom ρ should be marketed such that if these users rate the new product with relatively high ratings, the number of users to whom the product is recommended by the system is maximum.

Many ventures like *Increase YouTube Views*[1] and *Mile-*

[1] www.increaseyoutubeviews.com

KDD'12, August 12–16, 2012, Beijing, China.

Stone Internet Marketing[2] have emerged that offer services to increase/improve the ratings and reviews of their clients. Amazon's Vine program operates an interesting ecosystem, whereby reviewers (users) ranked highly by other users based on their past reviews are invited to review new items. These reviewers in turn are offered pre-release versions of products for the purpose of their review. We are not aware of any research works that investigate the mechanism used by the above ventures or by Amazon.

RECMAX has an interesting application for the cold-start items problem [11, 2], for the manufacturer of the item: for a newly introduced item, there may be few ratings provided by users. As a RS based on collaborative filtering relies on existing ratings, the new item will not be recommended to users until it gets enough ratings. Using RECMAX, when a new item is introduced, the item manufacturer has the opportunity to market it to the seed users and have the system recommend it to other users. Further comparison with works dealing with the cold-start problem appears in §2. There has been substantial work on detecting spam in RS and making them robust against spam. Mobasher [12] describes an attack called shilling attack whereby user profiles are injected into the system and reviewers are made to write positive reviews. The goal of RECMAX is *not* to spam a RS. Rather, it is to leverage a normal working RS for purposes of marketing. In particular, the ratings provided by the seed users to the new product are not influenced or interfered with in any way. They are free to rate it as they see fit. We focus on the case where the seed users like the new product and provide a relatively high rating.

The single most important challenge in studying the RECMAX problem is the wide diversity of RS algorithms (see [14], and [1] for comprehensive surveys) that are used in real applications. While User-based [9] and Item-based [15] collaborative filtering methods making use of neighborhood are extensively used, recently model-based approaches such as Matrix Factorization [16] have attracted a lot of attention. All these methods have many variants thus making the problem very difficult to formulate and study. In this work, we focus on User-based and Item-based methods since they are used in many real systems and are representative.

Targeted marketing in RS by selecting seed users has been recognized before [5, 13, 2, 3]. All of them use the notion of influence that a user has might exert in a RS, and based on their definition of influence, they select the seed users. The definition of influence and hence the goal is different in each of these works (as elaborated in §2). While the motivation and overall strategy are similar, that is, to select a set of seed users for marketing, RECMAX focuses on selecting seed users for maximizing the number of (other) users to whom a new item will be recommended by the RS, directly based on a recommender algorithm, as opposed to using an auxiliary notion of influence and resorting to social influence maximization. One of our main contributions is a thorough theoretical analysis of the complexity of RECMAX. As we shall show, RECMAX is not only a hard problem, it's even NP-hard to approximate within any reasonable factor. Given this, we explore various natural heuristics for solving RECMAX. We undertake a detailed analysis of three diverse real data sets, evaluate several natural heuristics for seed selection on those data sets and report our findings. One of the key contributions is showing that RECMAX is a real practical problem where selecting good seeds can yield a big payoff. To our knowledge, *we are the first to study the problem of identifying seed users to market a product to, in order to trigger a large number of recommendations of the product to other users from an existing RS*. We make the following contributions.

- We propose a novel problem RECMAX. It aims to find a set of seed users to whom a new product should be initially marketed such that if they endorse the product, the number of users to whom the product is recommended is maximum. We offer early empirical evidence that seeding does help in boosting the number of recommendations (§4).
- We perform a thorough theoretical analysis of RECMAX. We show that RECMAX is not only NP-hard to solve optimally, it is NP-hard to approximate within a factor of $1/(|V|^{1-\epsilon})$ for any fixed $\epsilon > 0$, both in User-based and Item-based methods (§5), under the settings we study. We also present intuitions behind where the intrinsic hardness of RECMAX comes from.
- Given the hardness, we explore several natural heuristics on 3 real datasets (§6). Some of the key observations we make are: a) If enough budget is allowed, seeding can make a substantial difference in the number of recommendations a new product can get; b) The gains that can be achieved saturates after some point, in both methods, suggesting that the seeding does not help after a certain budget; c) The overall gains that can be achieved in User-based systems are much higher than can be achieved in Item-based systems, though saturation is attained much quicker in Item-based systems; d) Choice of the seed set is critical.

Related work is reviewed in §2 while in §3, we provide a brief background and define RECMAX formally. Finally, we summarize the paper and discuss future research in §7.

2. RELATED WORK

Recommender Systems: The majority of works in RS (see [14] and [1] for detailed surveys) focus on improving the quality of the algorithms w.r.t. prediction accuracy. RS in general can be content-based or based on collaborative filtering. The most popular methodology in RS is based on collaborative filtering, which seeks to predict the expected rating that a user may provide to an item which she hasn't experienced before. Collaborative filtering methods are further classified into memory(or neighborhood)-based (e.g., see Chapter 4 of [14]) or model-based (e.g., matrix factorization [14], Chapter 5). This wide variety in RS makes RECMAX a very challenging problem to define and study. In this paper, we study both User-based and Item-based neighborhood methods, when the predicted ratings are estimated through weighted average.

A by-product of RECMAX is a solution to the cold-start problem [11], for item manufacturers. Various approaches have been proposed to solve the cold start problem including hybrid methods which in addition to existing ratings, utilize the item content data [11]. A recent work by Anand and Griffiths [2] employs an interesting incentive based approach which, given a collection of new items, calls for offering a list of new items for every user in the system. A user gets a fixed payment for every new item she rates. The

[2] `www.milestoneinternet.com/products/social-media/hotel-internet-marketing-reviews.aspx`

payment is determined using the influence of a user in the RS. The authors provide general guidelines on what factors the influence should depend on without explicitly defining it. They infer a social graph and apply existing social influence maximizing heuristics like Degree Discount [4]. As such this is a "proxy" to the original problem motivated by them. Moreover, degree discount is proposed for the independent cascade model where attempts of influence on a user by her neighbors are assumed to be independent, an assumption that, strictly speaking, doesn't hold in their framework. As the authors themselves acknowledge, their framework is vulnerable to fake ratings as the ratings provided by the seed users may be compromised by the free payments they get in return. The key differences from this work are as follows: a) Solving the cold start problem is not our main goal, although it is an interesting by-product; b) We look to pick seed users in a way that maximizes the number of users to whom the item is recommended directly using the underlying algorithms the RS use (e.g., User-based or Item-Based), instead of relying on social influence maximization; c) As mentioned earlier, ratings provided by seed users are not dictated or interfered with in anyway, avoiding fake ratings.

Another related line of work is spam in RS and its detection [12]. As mentioned earlier, the goal of RECMAX is not to spam the RS or to encourage fake ratings for the sake of item promotion. Instead, our goal is to leverage RS for ethical marketing. If seed users do not endorse a new item by providing high ratings, that item would simply not be recommended to other users. In a way, the seed users also play a role of initial customer feedback in our framework.

Influence Maximization: Our study is related in spirit to the problem of influence maximization in social networks as well as to the related problems of adoption and revenue maximization [5, 10, 7, 3]. In fact, the notion of influence from a data mining perspective was first studied in the context of RS [5]. Later Kempe et al. [10] formalized the problem as a discrete optimization problem. To the best of our knowledge, till date, there have been 4 papers which measure users' influence for marketing in RS by selecting seed users [5, 13, 2, 3]. Domingos et al. [5] define influence as the expected lift in profit and aim to maximize it. Anand and Griffiths [2] on the other hand, calculate influence empirically by exploiting social influence maximization heuristics. We provided a detailed account of the differences with our work above. Rashid et al. [13] focus on measuring influence scores of users in a RS and define it as a function of user's ability to change the predicted ratings of other users by δ where δ is the difference between consecutive rating valies (e.g., in a standard 1-5 rating scale, $\delta = 1$). Their main goal is to calculate influence scores of users, not to select a seed set. Bhagat et al. [3] on the other hand, develop a model for product adoption and use it to select seed users to maximize a new product adoption, from an influence maximization perspective.

Several factors set RECMAX apart from influence maximization. A fundamental distinction is that in RECMAX, the goal is to maximize the number of users to whom the system will *recommend* the item. By contrast, in influence maximization, the goal is to maximize the number of users who are influenced by the seed users and *adopt* the item. Second, no explicit social network is assumed as input for RECMAX, whereas for influence/adoption/revenue maximization, the basic backbone is a social network with influence weights. While in principle, one can induce a social network from a RS as [2] does, the resulting problem formulated on the social network is at best a proxy for the original problem of recommendation maximization. Third, unlike in influence maximization, there is no underlying propagation model (such as indepndent cascade or linear threshold – see [10]) in RECMAX.

3. BACKGROUND AND PROBLEM STUDIED

Recommender systems serve the top-ℓ items with the highest predicted ratings for a user as recommendations to that user, where ℓ is the desired length of the recommendation list. While User-based [9] and Item-based [15] are among the most popular collaborative filtering methods, recently hybrid approaches have been proposed [17]. Even more recently, Matrix factorization based methods [16] have gained significant attention. For excellent surveys, see Chapters 4 and 5 of [14]. In this paper, we study RECMAX on User-based and Item-based methods. With $\hat{R}(v,i)$, we denote the expected rating that user v gives an item i. Among User-based methods, perhaps the most popular approach to estimate $\hat{R}(v,i)$ is to use the weighted average:

$$\hat{R}(v,i) = \frac{\sum_{u \in N_i(v)} w(u,v) \cdot R(u,i)}{\sum_{u \in N_i(v)} abs(w(u,v))} \tag{1}$$

where $R(u,i)$ is the rating given to item i by user u, $abs(\cdot)$ is the absolute value function and $N_i(v)$ is the set of nearest neighbors of v, based on the similarity scores $w(\cdot,v)$, who have rated i. The similarity scores are calculated using cosine similarity or Pearson correlation or one of their variations (see [14], Ch. 4). In this paper, we employ the Pearson correlation as the similarity measure for User-based methods. Let $I(u)$ denote the set of items rated by user u and let $\mathcal{I} = I(u) \cap I(v)$. Then the similarity $w(u,v)$ using Pearson correlation is computed as

$$\frac{\sum_{i \in \mathcal{I}} (R(u,i) - \bar{R}_u) \cdot (R(v,i) - \bar{R}_v)}{\sqrt{\sum_{i \in \mathcal{I}} (R(u,i) - \bar{R}_u)^2} \cdot \sqrt{\sum_{i \in \mathcal{I}} (R(v,i) - \bar{R}_v)^2}} \tag{2}$$

where $\bar{R}_u$ is the average of u's ratings over various items.

Item-based methods on the other hand, use the similarities among items to estimate the predicted ratings. For instance, using the weighted average,

$$\hat{R}(v,j) = \frac{\sum_{i \in N_v(j)} w(i,j) \cdot R(v,i)}{\sum_{i \in N_v(j)} abs(w(i,j))} \tag{3}$$

where $N_v(j)$ is the set of nearest neighbors of (i.e., the most similar items to) j that are rated by v. Again, the similarities can be computed based on cosine, Pearson, their variants or other methods (see [14], Ch. 4). In this paper, we focus on adjusted cosine similarity, which is shown to be the most accurate similarity measure for Item-based methods [15]. Let $V(i)$ denote that set of users who rate item i, and let $U = V(i) \cap V(j)$. Then the adjusted cosine similarity between items i and j is

$$\frac{\sum_{u \in U} (R(u,i) - \bar{R}_u) \cdot (R(u,j) - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R(u,i) - \bar{R}_u)^2} \cdot \sqrt{\sum_{u \in U} (R(u,j) - \bar{R}_u)^2}} \tag{4}$$

In both User-based and Item-based methods, the number of nearest neighbors considered is often restricted to a limited

number d, usually between 20 and 50. This has been found to improve not only scalability, but even accuracy [14, 9].

Problem Studied. In addition to the log of past ratings, we assume we know the algorithm used by the RS. In this paper, we focus on both User-based and Item-based methods. For User-based, we employ Pearson correlation similarity measure, while for Item-based, we utilize adjusted cosine similarity measure, both defined above.

With ρ, we denote the new item to be marketed. The number of items that can be recommended to a user v is limited and as a result various items are in competition to enter the recommendation lists of users. Clearly, for ρ to appear in the recommendation list of user v, the expected rating $\hat{R}(v, \rho)$ must be more than the *rating threshold* of user v, which is the predicted rating of v for the last item in its recommendation list. We use θ_v to refer to the rating threshold of user v. We do not necessarily assume that the new item ρ has no prior ratings.

Let V be the set of users in the system. The problem we study in this paper is to select a set $S \subseteq V$ of k users (called *seed set*) such that by convincing them to provide relatively high ratings to the new item ρ, the number of users to whom it is recommended is maximized. We leave the definition of "relatively high rating" open for now and argue that if the new item is not a good item causing the seed users to provide relatively low ratings, then in any case, it should *not* receive many recommendations. The goal of this work is not to promote bad items. In our experiments, we derive a rating $R(u, \rho)$ for a seed user u by taking an average of the top-20% highest ratings provided by u.

We use $H(S)$ to denote the set of users to whom the item is recommended by virtue of the seed set S. Clearly, $H(S) \subseteq V \setminus S$ as the users in S have already adopted the item and there is no value to recommending the item to them. Intuitively, users in the seed set S should have high "influence" where the influence flows indirectly via the recommendations. We define $f(S) := |H(S)|$, the *hit score* of seed set S as:

$$f(S) = \sum_{v \in V - S} I(\hat{R}(v, \rho) > \theta_v) \tag{5}$$

where $I(\cdot)$ is an indicator function which is 1 if $\hat{R}(v, \rho) > \theta_v$ and 0 otherwise. Formally, the problem is defined as follows.

PROBLEM 1 (RECMAX). Given a recommender system and rating thresholds θ_v for each user v in the system, and a number k, find a set S of k users s.t. $f(S)$ is maximized.

4. DOES SEEDING HELP?

Before we study the RECMAX problem of how to select an optimal seed set, we address several natural questions: Does seeding help? What are the gains that may be achieved? How does hit score vary with respect to the seed set size?

To answer these questions, in this section, we show the results from preliminary experiments, using the popular Movielens data set. More detailed experiments are discussed in §6. Predicted ratings are computed according to weighted average, both for User-based and Item-based methods (see Eq. 1 and 3). The similarity measure we use is Pearson correlation for User-based (Eq., 2) and adjusted cosine for Item-based (Eq. 4). See §6 for explicit experiment settings.

Next, we run a RANDOM heuristic to select the seed set and calculate the hit score achieved, both in User-based and Item-based methods. We assume that a seed user $u \in S$ provides a relatively high rating to the new item. In our experimental evaluation, we set this rating as the average of the top-20% highest ratings provided by user u. Given a budget k, a seed set of size k is randomly selected and the hit score is computed. We repeat this process 100 times and take the average. Note that it is important to repeat the process several times as in some cases, the seed set picked by RANDOM may be really good with a large hit score, while in other cases, very poor with a low hit score. Fig. 1 shows the results. As can be seen, the hit score achieved is remarkable, under both methods, with a much higher hit score under User-based. For example, a budget of 500 can get a hit score of 5091 on User-based and a budget of 20 can achieve a hit score of 636 on Item-based.

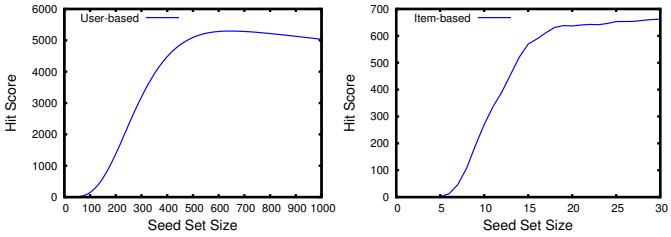


Figure 1: Hit Rate achieved by random seed set on Movielens dataset on User-based (left) and Item-based (right). The plots show that even when the seed sets are selected randomly, seeding does help and exhibits impressive gains in hit score.

This immediately suggests that *seeding does help and establishes the case for* RECMAX. Another interesting observation is that the hit score curve resembles the classic sigmoid curve, implying that if sufficient budget is allowed, then the gains achieved can be substantial. Moreover, if we continue to increase the budget, the hit score plateaus, and sometimes, it may even start decreasing (unlike sigmoid). These interesting observations inspire us to perform a deeper analysis of RECMAX– both theoretical and empirical – and we do so in the next sections.

5. COMPLEXITY OF RECMAX

In this section, we study the complexity of RECMAX. Unfortunately, it turns out to be intractable: in fact, it is hard to approximate within any reasonable factor, unless P=NP. Due to the wide variations in RS methods, it is tedious to study the complexity for everyone of them. Instead we present formal proofs for cases where the underlying methods are either User-based or Item-based, with Pearson and adjusted cosine as similarity functions respectively. We believe these are representative cases and our results give an indication that RECMAX may indeed be a very hard problem for known RS algorithms in general.

We also provide important insights as to why RECMAX is so hard. First, it is easy to show that the function $f(S)$ defining the hit score of a seed set is in general non-monotone and non-submodular. Intuitively, $f(\cdot)$ is based on predicted ratings computed using weighted average, which is neither monotone nor submodular. Indeed, the plot in Fig. 1 (left) shows an example where $f(\cdot)$ is not monotone and both (left) and (right) plots show it is not submodular in general. To help us prove the hardness results, we introduce a helper problem that we call *Maximum Encirclement Problem* (MEP). We show that MEP is as hard to approximate as the classical *Maximum Independent Set Problem* (MIS). Recall that in an undirected graph $G = (V, E)$, a set $S \subseteq V$ is an independent set iff no two nodes in S are adjacent. MIS cannot be approximated within a factor of $1/(|V|^{1-\epsilon})$ for

any fixed $\epsilon > 0$, unless P=NP [8]. We show that RECMAX is as hard to approximate as MEP and our result follows.

Given an undirected graph $G = (V, E)$ and a set $S \subseteq V$, we say a node $v \in V \setminus S$ is *encircled* by S, provided v has at least one neighbor and all its neighbors are in S.

PROBLEM 2 (MEP). Given an undirected graph $G = (V, E)$ and an integer k, find a set $S \subseteq V$ such that $|S| \leq k$ and S encircles the maximum number of nodes.

For the sake of simplicity, we assume there are no isolated (i.e., zero degree) nodes in the graph. As an insight, notice that MEP is very close to the minimum vertex cover problem (MVC) which asks if there is a set $S \subseteq V$ of size $\leq k$, given $k \leq |V|$, such that every edge of the graph is incident on at least one node in S. Notice, S is a vertex cover iff for any node $v \in V \setminus S$, all its neighbors are in S, i.e., iff v is encircled by S (assuming there are no isolated nodes). Thus, MVC is equivalent to asking if there is a set of nodes $S \subseteq V$ of size k such that it encircles rest of the nodes in the graph. From this, it follows that MEP is NP-hard, which is not very surprising. What is interesting is that while MVC enjoys 2-approximation, MEP is not approximable, as we show next.

LEMMA 1. *The Maximum Encirclement Problem (MEP) cannot be approximated within a factor of $\frac{1}{|V|^{1-\epsilon}}$ for any fixed $\epsilon > 0$, unless P=NP.*

PROOF. Consider an instance of MIS, consisting of a graph $G = (V, E)$. Create an instance of MEP by keeping the graph same. Clearly, a set $S \subseteq V$ is an independent set of G if and only if the set $V \setminus S$ encircles each node in S. In particular, S is an optimal solution to MIS iff $V \setminus S$ is an optimal solution to MEP where the budget is set to $|V| - |S|$. Assume there is a β-approximation algorithm for MEP. For $k = 1, ..., |V|$, run the approximation algorithm on the MEP instance with the budget set to k. Let $\hat{S}$ be a set of nodes output by this algorithm for some value of $\hat{k}$: $|\hat{S}| \leq \hat{k}$ and the number of encircled nodes is the maximum among all values of k. Clearly, the number of nodes encircled by $\hat{S}$ is $\geq \beta \cdot (|V| - \hat{k})$, since for some budget, there must exist a set that encircles every node outside the set. Let T be the set of nodes encircled by $\hat{S}$. Clearly, T is an independent set of size $\geq \beta \cdot (|V| - \hat{k})$ which thus is a β-approximate solution to MIS. Therefore, MEP is as hard to approximate as MIS. □

Now, we are ready to prove hardness results for RECMAX.

THEOREM 1. *Under User-based Collaborative Filtering that uses Eq. 1 and 2 to compute predicted ratings,* RECMAX *is NP-hard. Moreover, it cannot be approximated within a factor of $\frac{1}{|V|^{1-\epsilon}}$ for any $\epsilon > 0$, unless P=NP.*

PROOF. We prove the theorem by reducing MEP to RECMAX. From an instance $\mathcal{I}$ of MEP consisting of graph $G = (V, E)$, create an instance $\mathcal{J}$ of RECMAX as follows. For each node $u \in V$, create a user u in instance $\mathcal{J}$. For each edge $(u, v) \in E$, create an item i_{uv} such that users u and v rate the item i_{uv} with rating 2. Add $|V|$ dummy items to the instance $\mathcal{J}$ such that each user rates exactly one of these dummy items, with rating 1. Let d be the maximum number of nearest neighbors that are being considered by the RS, then also add d dummy users (call this set of dummy users X) in instance $\mathcal{J}$ such that each of these dummy users $x \in X$ rates all $|V|$ dummy items with rating 2. Hence, in our construction, we have $|V| + d$ users, $|E| + |V|$ items and $2 \cdot |E| + (d+1) \cdot |V|$ ratings. Let every seed user provide the same rating to the new item, say 2. Next, let each dummy user provide rating 1 to the new item ρ. We will decide the rating threshold θ_v later.

Let $S \subseteq V$ be any seed set. Then, we claim that a node $v \in V \setminus S$ is encircled by S (in instance $\mathcal{I}$) iff the new item is recommended to v, that is, $v \in H(S)$ (in instance $\mathcal{J}$). We prove this claim below. From this claim, it is easy to see that a set S of size k encircles η nodes in instance $\mathcal{I}$ if and only if $|H(S)| = \eta$ in instance $\mathcal{J}$. Therefore, RECMAX is as hard to approximate as MEP, following the same logic used in the proof of Lemma 1, and this proves the theorem.

Consider a user $v \notin S$. According to Eq. 2, the similarity between v and a user $u \in S$ is 1. Likewise, the similarity among v and a dummy user $x \in X$ is -1 With $deg(v)$, we denote the degree of user v in instance $\mathcal{I}$. If $l \leq deg(v)$ (non-dummy) neighbors of v rate the new item ρ, then, according to Eq. 1,

$$\hat{R}(v, \rho) = \frac{\sum_{u \in N_\rho(v)} w(u, v) \cdot R(u, i)}{\sum_{u \in N_\rho(v)} abs(w(u, v))}$$
$$= \frac{l \cdot 1 \cdot 2 + (d - l) \cdot (-1) \cdot 1}{abs(l \cdot 1) + abs((d - l) \cdot (-1))} = \frac{3 \cdot l - d}{d}$$

Set the rating threshold $\theta_v = (3 \cdot deg(v) - d)/d - \delta$ where δ is a small number to ensure that $\hat{R}(v, \rho)$ is $(3 \cdot deg(v) - d)/d$ iff the new item is recommended to v. This rating threshold can be achieved iff all neighbors of v rate the new item. As a result, an item is recommended to a user v iff all of its neighbors are in S. that is, iff v is encircled by S. This was to be shown. □

THEOREM 2. *Under Item-based Collaborative Filtering that uses Eq. 3 and 4 to compute predicted ratings,* RECMAX *is NP-hard. Moreover, it cannot be approximated within a factor of $\frac{1}{|V|^{1-\epsilon}}$ for any fixed $\epsilon > 0$, unless P=NP.*

PROOF. As above, we prove the claim by reducing MEP to RECMAX under the Item-based method that uses weighted average along with adjusted cosine similarity (see Eq. 3 and 4). Consider an instance $\mathcal{I}$ of MEP consisting of an undirected graph $G = (V, E)$. Create an instance $\mathcal{J}$ of RECMAX as follows. Each node $v \in V$ corresponds to a user. Create $|E|$ items in instance $\mathcal{J}$, one for each edge in instance $\mathcal{I}$. An item i_{uv} corresponding to edge $(u, v) \in E$ is rated by users u and v with rating 2. Add a special item j such that each user rates it with rating 1. Thus, in instance $\mathcal{J}$, we have $|V|$ users, $|E| + 1$ items and $2 \cdot |E| + |V|$ ratings. We will decide the values of rating thresholds later.

Let $S \subseteq V$ be any seed set. Like above, we claim that a node $v \in V \setminus S$ will be in $H(S)$ (in instance $\mathcal{J}$) iff v is encircled by S (in instance $\mathcal{I}$). The proof then follows from the same logic used in the proof of Theorem 1.

We now show the above claim. Say, each seed user provides a rating 2 to the new item. Then, for a user $u \in S$, $\bar{R}_u = (2 \cdot deg(u) + 1 + 2)/(deg(u) + 2)$, as it provides rating 2 to all the items that correspond to the edges in $\mathcal{I}$ and 1 to the special item j, in addition to rating 2 to the new item ρ. Considering the adjusted cosine similarity function among items (see Eq. 4), for any item i that corresponds to an edge in instance $\mathcal{I}$, the deviation term is $2 - (2 \cdot deg(u) + 3)/(deg(u) + 2) = 1/(deg(u) + 2)$. Let us

call this x_u. The similarity of i with ρ would then be

$$w(i,\rho) = \frac{\sum_{u\in U} x_u \cdot x_u}{\sqrt{\sum_{u\in U} x_u^2} \cdot \sqrt{\sum_{u\in U} x_u^2}} = 1$$

where $U = V(i) \cap V(\rho) = V(i) \cap S$. Similarly, for the special item j, the deviation term is $y_u = 1 - (2 \cdot deg(u) + 3)/(deg(u) + 2) < 0$. It implies that the similarity between j and ρ is strictly less than 0. Let it be z. Thus, from Eq. 3, the expected rating of a user v on item ρ is

$$\hat{R}(v,\rho) = \frac{\sum_{i\in I(v)\cap I(S)} w(i,\rho) \cdot R(v,i)}{\sum_{i\in I(v)\cap I(S)} abs(w(i,\rho))} \quad (6)$$

where $I(S) = \bigcup_{u\in S} I(u)$ is the set of items rated by any seed user in S. Note that if an item i is not rated by any user in S, then it cannot be among the neighbors of new item ρ because ρ is rated only by users in S. That's why we iterate over all items $i \in I(v) \cap I(S)$ in Eq. 6. Moreover, $|I(v)| = 1 + deg(v)$. Clearly, the special item j is in $I(S)$, as all users rate j. Next, any item $i \neq j$, if added to $I(S)$, cannot decrease $R(v,\rho)$ as all such items are rated with rating 2. Hence, the maximum possible predicted rating that v could give the new item ρ is realized when all the items rated by v are in $I(S)$, i.e., $I(v) \subseteq I(S)$. Following Eq. 6, this maximum rating is

$$R_{max}(v) = \frac{2 \cdot deg(v) - z}{deg(v) + z}$$

Thus, we can set θ_v as $R_{max}(v) - \delta$ where δ is an extremely small number which ensures that the item ρ is recommended to v if and only if the predicted rating $\hat{R}(v,\rho) = R_{max}$. It means that the user $v \in H(S)$ if and only if $I(v) \subseteq I(S)$. However, an item $i_{uv} \in I(v) \setminus \{j\}$ can be in $I(S)$ if and only if the neighbor u of v is in S (because only u and v have rated the item i_{uv}). Thus, $I(v) \subseteq I(S)$ iff all neighbors of v are in S, i.e., iff S encircles v. This was to be shown. □

It should be noted that a slight modification in the reduction shown above shows that the RECMAX under Item-based method (that uses weighted average) is as hard as MEP even when the similarity function is cosine or Pearson. We omit the details for brevity.

Discussion. By now, it is clear that the RECMAX problem cannot be approximated within any reasonable factor, for the specific settings we focus on in this paper. Many real RS are based on the User-based and Item-based methods considered in our settings of RECMAX and thus our results apply to RECMAX over those systems. What can we say about RS that are based on the hybrid of content-based and collaborative filtering, on matrix factorization etc.? Given the complexities of systems based on such methods, it is very likely that RECMAX continues to be hard (to approximate) on these systems. The main reason behind this hardness of RECMAX is its similarity with MEP: in order to push an item to the recommendation list of a user, several other users may need to rate the item. This has a flavor similar to the user being "encircled" by other users. Since the hit score function is neither monotone nor submodular, as noted in the beginning of the section, it is not clear a priori which heuristics will work well for RECMAX. So, the question arises: What can we do with RECMAX? Does it still have a case? As we show in §4, the answer is yes. Even if the problem is too hard to approximate, it still makes a good sense for targeted marketing, as the gains (in terms of hit score) achieved are impressive, even with the RANDOM heuristic. Clearly, we cannot develop an algorithm with theoretical guarantees, but we can try several heuristics and see what works best and under what conditions – which is what we do in the next section.

6. EXPERIMENTS

Algorithms. Given that RECMAX cannot be approximated within any reasonable factor, in this section we explore several natural heuristics, and then gauge their performance on 3 real datasets – Movielens, Yahoo! Music and Jester– and analyze the results. Let k be the budget on seed set size. We explore the following heuristics in our evaluation.

RANDOM: Seed set is selected randomly. The process is repeated 100 times and the average hit score of different choices is taken. We take this as a baseline.

MOST-ACTIVE: Choose the top-k users with most number of ratings. The intuition here is that the users who are most active "control" most of the ratings in the system, and thus can be seen as a good target for marketing.

MOST-POSITIVE: Choose the top-k users with most positive average ratings as seeds. It is another natural way for targeted marketing, as many manufacturers may want to avoid bad ratings/opinions about their product. Thus, targeting positive users is a "safe" marketing choice.

MOST-CRITICAL: Choose the top-k users with most critical average ratings. This is another extreme, where it may be argued that if the most critical users endorse a product, the chances of success (i.e., large hit score) are high.

MOST-CENTRAL: For each user, we compute its similarity with every other user. Then, we compute its aggregate similarity score $agg(u) = \sum_{v\in V} sim(u,v)$. Next, we select top-$k$ users with highest aggregate similarity scores as the seed set. In case of User-based, we compute the similarity according to the Pearson correlation, which is what is used in estimating predicted ratings.

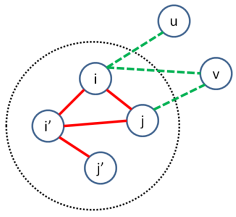


Figure 2: Example.

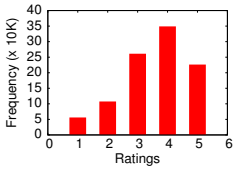
For Item-based, the recommendations are generated using similarities among items, and not users. Hence, it is not clear how to compute similarities among users in case of Item-based. For instance, consider the example shown in Fig. 2. It has 4 items i, j, i', j', and we know the weights on the red edges: they are computed according to Eq. 4. No two users are directly connected in Fig. 2. Hence, to compute the similarity between users u and v, we take the weighted average of ratings provided by the users u and v on every pair of items they rate, as follows.

$$sim(u,v) = \frac{\sum_{i\in I(u), j\in I(v)} w(i,j) \cdot sim(R(u,i), R(v,j))}{\sum_{i\in I(u), j\in I(v)} w(i,j)}$$

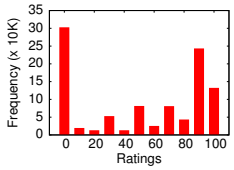
where $sim(R(u,i), R(v,j))$ is the similarity between the ratings provided by user u on item i and user v on item j. If users u and v rate a common item i, i.e., $i = j$, then, $w(i,j) = w(i,i) = 1$. Intuitively, if the ratings are nearly equal, $w(R(u,i), R(v,j))$ should be close to 1 and if the ratings are very different, the similarity score should be close

	Movielens	Yahoo	Jester
#Users	6040	10K	25K
#Items	3706	5069	100
#Ratings	1M	1M	1.8M
Mean	3.58	51.9	10.88
Std. Dev.	1.12	39.9	5.23

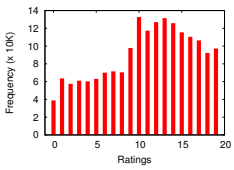
(a) Dataset statistics



(b) Movielens Ratings Distribution

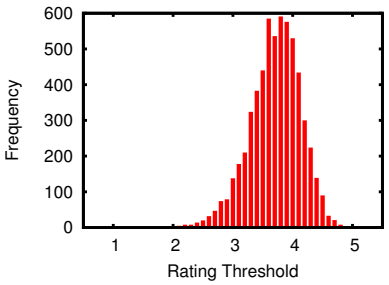


(c) Yahoo! Music Ratings Distribution

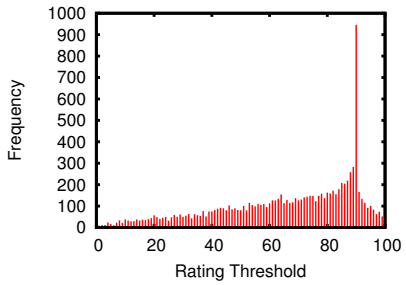


(d) Jester Ratings Distribution

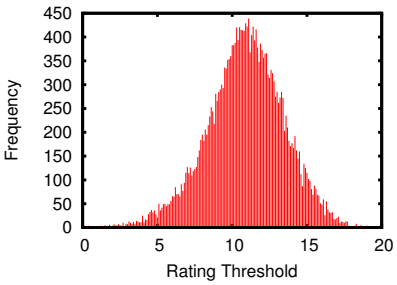
Figure 3: (a) Summary of datasets; (b)-(d) Distributions of ratings.



(a) Movielens users' ratings thresholds



(b) Yahoo! users' ratings thresholds



(c) Jester users' ratings thresholds

Figure 4: Distributions of users's rating thresholds.

to 0. One way to define it is

$$sim(R(u,i),R(v,j)) = 1 - \frac{abs(R(u,i) - R(v,j))}{(R_{max} - R_{min})}$$

where R_{max} and R_{min} are the maximum and minimum ratings in the system and $abs(\cdot)$ is the absolute value function.

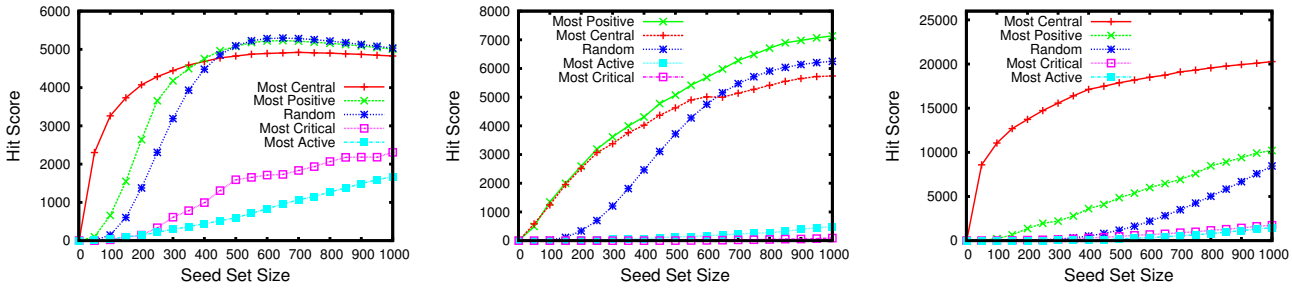
Datasets. We run experiments on three real world datasets, whose statistics are given in Fig. 3(a). The first dataset is Movielens. It consists of 1M ratings given by 6040 users distributed over 3706 movies. The ratings are on a scale of 1-5. Fig. 3(b) shows the distribution of ratings which follows the normal distribution with mean 3.58 and standard deviation 1.12. Next, we use Yahoo! Music[3] data set, provided as part of the Yahoo! Research Alliance Webscope program. It represents a sample of Yahoo! users' ratings of musical artists, gathered over a thirty-day period sometime prior to March 2004. The raw data contains 11.5M ratings over 98K artists given by 1.9M users. The ratings are integers ranging from 0 to 100, except 255 (a special rating meaning "never play again"). From this data set, we randomly sampled 10K users who assigned 20 or more ratings. From the sample, we discarded all the artists who received less than 20 ratings. This pre-processing resulted in 1M ratings over 5069 artists. We ignored the special rating 255 in the pre-processing and kept only explicit ratings. The rating distribution of the resulting data set is presented in Fig. 3(c). Notice that unlike in Movielens, there is a large number of ratings with value 0 in Yahoo!. Finally, we use the Jester Joke collaborative filtering data set released by Ken Goldberg from UC Berkeley[4][6]. It has 100 jokes rated by 25K users. There are a total of 1.8M ratings, on a continuous scale of -10 to 10. Since we use the weighted average to compute expected ratings (according to Eq. 1 and 3) which is meant for positive ratings, we shift the ratings from the $[-10, 10]$ scale to $[0, 20]$, by adding 10 to each rating. The ratings distribution is presented in Fig. 3(d). Again, unlike the movie ratings in Movielens, there is a significant number of negative ratings in Jester. In our experiments, we derive a rating $R(u,\rho)$ for a seed user u by taking an average of the top-20% of the highest ratings provided by u. Finally, while computing $\hat{R}(v,\rho)$, we ignore the small changes in similarity values among users (in case of User-based) and existing items (in case of Item-based) that happen because of ratings provided by seed users to the new item ρ. We evaluate our heuristics for RecMax on both User-based and Item-based collaborative filtering methods.

[3] www.music.yahoo.com

[4] eigentaste.berkeley.edu/user

User-Based collaborative filtering. We first report our findings in User-based systems. Predicted ratings are computed using weighted average along with Pearson correlation (see Eq. 1 and 2). For $\hat{R}(v,i)$ to be non-zero, we require that at least 5 of v's nearest neighbors must rate the item i, i.e., $|N_i(v)| \geq 5$. Similarly, for the similarity between users u and v to be non-zero, we require them to rate at least 5 items in common, i.e., $|I(u) \cap I(v)| \geq 5$. Finally, we consider at most 25 nearest neighbors of v while computing $\hat{R}(v,i)$. These settings are used and recommended by previous works (E.g., see [14, 9]). On Movielens and Yahoo!, we set the length of the recommendation list ℓ as 15, as popular recommendation music/movies systems like Yahoo! Music, Movielens, Youtube usually show 15-20 recommendations on a single page. On Jester, ℓ is set to 1 because only one joke is recommended at a time on the Jester website. Based on these settings, we compute the rating thresholds θ_v of every user v. The distributions of the rating thresholds for the three data sets are shown in Fig. 4.

Fig. 5 compares the various heuristics, described above, on all 3 datasets. The plots reveal many interesting points. (i) On all 3 datasets, we see that Most-Active and Most-Critical perform poorly, and Most-Positive and Most-Central perform relatively better. Thus, simply choosing users who provide lots of ratings or users who are very critical does not seem to help. Surprisingly, even Random exhibits a relatively good performance. For instance, on Yahoo! (Fig. 5(b)), with a budget of 500, Random is able to achieve a hit score of 3719. With the same budget, Most-Central and Most-Positive achieve 4628 and 5073, that is, a 24.4% and a 36.4% improvement over Random, respectively. By contrast, Most-Active and Most-Critical



(a) Movielens (b) Yahoo! (c) Jester

Figure 5: Hit Score achieved with various algorithms on different datasets on User-based RS.

achieve poor hit scores of only 132 and 2 respectively. This clearly establishes that *the choice of seed sets is critical and can make a substantial difference*. Except on Yahoo! where MOST-POSITIVE wins, MOST-CENTRAL is found to be the best. MOST-CENTRAL picks seeds who have the greatest affinity to other users in the aggregate and in the context of User-based collaborative filtering, this does make sense. (ii) In general, the hit score achieved is impressive. For instance, with a budget of only 300, MOST-CENTRAL attains a hit score of 4444, 3377 and 15.6K on Movielens, Yahoo! and Jester respectively. It shows even the simple heuristic strategies for seed selection make RECMAX relevant and useful for targeted marketing, despite the problem being inapproximable within any reasonable factor. (iii) Depending on the data set and the seed set selected, we may encounter a "tipping point", a point at which a slight increase in budget can make a significant difference in the hit score achieved. For example, on Movielens, using MOST-POSITIVE, the hit score with a budget of 50 is only 288, i.e., the average hit score per seed is 5.76. On the other hand, the overall hit score shoots up to 3656 with a the budget of 250, i.e., the average hit score for the last 200 seeds is 16.84. The rate of growth in hit score slows down as the budget is increased beyond a point. E.g., when the budget is extended futher by 200 to 450, the hit score increases to 4962, that is, the average hit score for the last 200 seeds drops to 6.53. When we continue to expand the budget, the hit score plateaus and in some cases it may even decrease. For instance, on Movielens with MOST-POSITIVE, if we increase the budget from 700 to 1000, the hit score falls from 5215 to 5008.

Item-Based collaborative filtering. We now report our findings in Item-based collaborative filtering. Again, to compute the expected rating $\hat{R}(v, i)$, we use the weighted average with similarities among items computed using adjusted cosine (see Eq. 3 and 4). As earlier, we require for $\hat{R}(v, j) > 0$, that there be at least 5 similar items to j that are rated by v, that is, $|N_v(j)| \geq 5$. Similarly, the similarity among items i and j can be more than 0 only if they are rated by at least 5 common users, i.e., $|V(i) \cap V(j)| \geq 5$. Finally, we consider at most 25 most similar items to j to compute $\hat{R}(v, j)$. We keep the length of the recommendation list ℓ the same as before, that is, 15, 15 and 1 for Movielens, Yahoo! and Jester respectively. Based on these settings, the rating thresholds of all users are computed. We found their distributions to be very similar to those for User-based (see Fig. 4) and so suppress their plots for brevity.

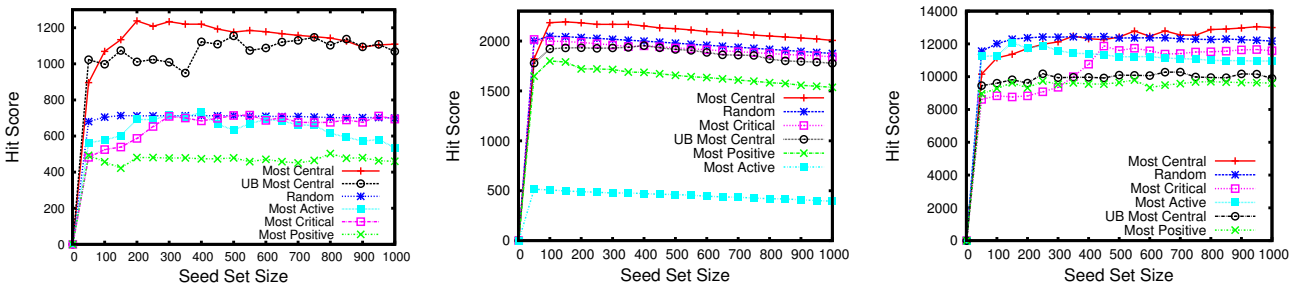
In Fig. 6, we show the hit score achieved by the various heuristics. Note that the MOST-CENTRAL heuristic is fundamentally different in Item-based. Out of curiosity, we also include in our comparison the MOST-CENTRAL heuristic of User-based method. For clarity, we refer to the latter as UB-MOST-CENTRAL in Fig. 6. Again, many interesting observations are made. (i) On all 3 datasets, the overall hit score that is achieved in Item-based is much lower than in User-based. Moreover, on all the datasets, the hit score plateaus much faster in Item-based than in User-based. For example, on Movielens with MOST-CENTRAL heuristics, the hit score saturates at 1238 at the budget of 200 in case of Item-based. In User-based, on the other hand, the hit score saturates at 4825 at the budget of 500 (see Fig. 5(a)). As a result, much less seeding is required to achieve the maximum possible hit score in Item-based. (ii) Except in Jester, MOST-CENTRAL beats every other algorithm. For instance, on Movielens, with a budget of 200, while MOST-CENTRAL gets the hit score of 1238, RANDOM achieves just 712. MOST-ACTIVE, MOST-CRITICAL and MOST-POSITIVE attain the scores 692, 587 and 481 respectively. (iii) Another interesting observation is that while MOST-POSITIVE performs quite well in User-based, it performs poorly in Item-based. This shows strategies that work well in User-based need not work well in Item-based. However, interestingly enough, the intuition underlying MOST-CENTRAL seems to be borne out well by the results for both User-based and Item-based.

Next, we undertake a comparison of User-based with Item-based. Fig. 7 presents a zoomed-in comparison of hit scores achieved in User-based and Item-based methods, with the heuristic MOST-CENTRAL. Except in Movielens, the initial rise of hit score is much steeper in Item-based than in User-based. And as the budget increases, hit score in User-based continues to increase well beyond the peak value of Item-based and it plateaus at a much higher budget than Item-based. Finally, we look at the intersection of seed sets obtained from MOST-CENTRAL heuristic in both cases. Out of 1000 seeds, the number of common seeds are 103, 219 and 62 on Movielens, Yahoo! and Jester, respectively, suggesting that the seed sets are different in both methods.

In summary, the hit scores behave very differently in User-based and Item-based methods. While the overall hit scores that are achieved in User-based are much higher, the saturation in hit scores is achieved much faster in case of Item-based. In both, respective versions of MOST-CENTRAL are found to be most effective, although other heuristics behave differently. For instance, MOST-POSITIVE performs quite well in User-based, while its performance in Item-based is poor. These observations suggest that different approaches are required to select good seed sets on different systems, and no single rule of thumb works well for all systems.

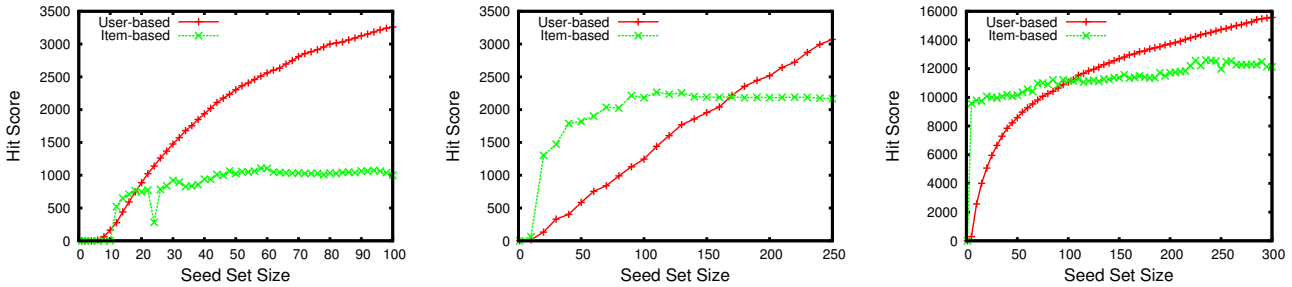
7. CONCLUSIONS AND FUTURE WORK

The main goal of this paper is to propose and study a



(a) Movielens (b) Yahoo! (c) Jester

Figure 6: Hit Score achieved with various algorithms on different datasets on Item-based RS.



(a) Movielens (b) Yahoo! (c) Jester

Figure 7: Comparison of User-based and Item-based RS with the algorithm Most-Central.

novel problem that we call RECMAX. It aims to develop a technology to select seed users in a collborative filtering based RS such that if they endorse a new product with relatively high ratings, the new product is recommended to a large number of users, solely because of the underlying mechanics of the RS. We motivate the problem with real world applications. We focus on the two widely used methods – User-based and Item-based. We perform a thorough theoretical analysis and show that RECMAX is not only NP-hard to solve optimally, it is NP-hard to approximate within any reasonable factor. Given that, we explore various natural heuristics and show that, even though the problem is inapproximable, simple heuristics like MOST-CENTRAL can fetch impressive gains. This makes RECMAX an interesting problem for targeted marketing in RS.

This work opens up a wide array of challenges. First, developing more effective heuristics is important, interesting and challenging, given that the hit score function is neither monotone nor submodular. We also need to calibrate the proposed and new heuristics on a wide array of real data sets. Second, as we saw in our empirical evaluation, the hit score function behaves distinctly on User-based and Item-based systems. It would be interesting to see how it behaves when mixed approaches are employed. Third, studying RECMAX on more recent RS methodologies, for example, Matrix factorization [16] – both theoretically and empirically, would be exciting. Finally, the vision behind RECMAX is marketing based on an operational recommender *system*, which is not just a simple RS algorithm but has a much greater complexity. Formulating and solving RECMAX and launching it on an operational RS is a fascinating challenge with great potential.

Acknowledgments. We thank Suresh Venkatasubramanian and Nick Harvey for helpful discussions. We acknowledge Yun Lou's involvement in the initial phase of the project. We appreciate anonymous reviewer's feedback that helped us in improving the presentation of the paper.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17, 2005.

[2] S. S. Anand and N. Griffiths. A market-based approach to address the new item problem. In *RecSys*, 2011.

[3] S. Bhagat, A. Goyal, and L. V. S. Lakshmanan. Maximizing product adoption in social networks. In *WSDM*, 2012.

[4] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, 2009.

[5] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, 2001.

[6] K. Y. Goldberg et al. Eigentaste: A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2), 2001.

[7] J. D. Hartline et al. Optimal marketing strategies over social networks. In *WWW*, 2008.

[8] J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. In *FOCS*, 1996.

[9] J. L. Herlocker et al. An algorithmic framework for performing collaborative filtering. In *SIGIR*, 1999.

[10] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[11] X. N. Lam et al. Addressing cold-start problem in recommendation systems. In *ICUIMC*, 2008.

[12] B. Mobasher et al. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Techn.*, 7(4), 2007.

[13] A. Rashid, G. Karypis, and J. Riedl. Influence in ratings-based recommender systems: An algorithm-independent approach. In *SDM*, 2005.

[14] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *WWW*, 2001.

[16] G. Takács et al. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *RecSys*, 2008.

[17] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*, 2006.