New Recommendation Techniques for Multi-Criteria Rating Systems

Gediminas Adomavicius
gedas@umn.edu

YoungOk Kwon
ykwon@csom.umn.edu

Department of Information and Decision Sciences
Carlson School of Management
University of Minnesota

Abstract. While traditional single-rating recommender systems have been successful in a number of personalization applications, the research area of multi-criteria recommender systems has been largely untouched. In order to take full advantage of multi-criteria ratings in various applications, new recommendation techniques are required. In this paper we propose two new approaches – the similarity-based approach and the aggregation function-based approach – to incorporating and leveraging multi-criteria rating information in recommender systems. We also discuss multiple variations of each proposed approach, and perform empirical analysis of these approaches using a real-world dataset. Our experimental results show that multi-criteria ratings can be successfully leveraged to improve recommendation accuracy, as compared to traditional single-rating recommendation techniques.

Keywords: Personalization, recommender systems, collaborative filtering, multi-criteria ratings, rating estimation.

1. Introduction and Motivation

In order to make good decisions in any situation, it is typically necessary to possess a certain sufficient amount of information. Technologies enable us to easily obtain more information, especially on the Internet. For instance, if an individual wants to rent a movie online, there are numerous choices available. However, too much information can make decision-making inefficient, leading to information overload. Personalization technologies and recommender systems help to overcome this problem by providing personalized suggestions regarding which information is most relevant to users. Most of online shopping sites and many other applications now use recommender systems. The most popular examples include Netflix, which recommends movies, and Amazon.com, which recommends books, CDs, and various other products. If users offer their feedback on purchased or consumed items, the task of recommender systems is to predict user preferences for the yet unseen items based on user's prior feedback and activities and, subsequently, to recommend the item(s) with the highest estimated relevance to the user.

Recommender systems are usually classified into three categories based on their approach to recommendation: *content-based*, *collaborative*, and *hybrid* approaches (Balabanovic & Shoham 1997). Content-based recommender systems recommend items similar to the ones the user preferred in the past. Collaborative (or collaborative filtering) recommender systems recommend items that users with similar preferences have liked in the past. Finally, hybrid approaches combine content-based and collaborative methods, which can be done in many different ways (Adomavicius & Tuzhilin 2005). Furthermore, recommender systems can also be classified based on the nature of their algorithmic technique into *memory-based* and *model-based* approaches (Breese et al. 1998). In particular, memory-based techniques usually represent heuristics that calculate recommendations "on the fly" based directly on the previous user

activities. In contrast, model-based techniques use previous user activities to first learn a predictive model (typically using some statistical or machine-learning methods), which is then used to make recommendations.

While a comprehensive survey of recommender systems research literature is beyond the scope of this paper,[1] it is important to note that vast majority of current recommender systems typically use a single criterion (i.e., a single numerical rating) to represent the utility of an item to a user in the two-dimensional $Users \times Items$ space. The recommendation process starts with the specification of the initial set of ratings that is either explicitly provided by the users or is implicitly inferred by the system. For example, in case of a movie recommender system, user John Doe may assign a rating of 11 (out of 13) for movie "Vertigo," i.e., set R(John_Doe, Vertigo) = 11. Once these initial ratings are specified, a recommender system tries to estimate the rating function R

$$R: Users \times Items \rightarrow R_0 \qquad (1)$$

for the (user, item) pairs that have not been rated yet. R_0 is usually represented by a totally ordered set (e.g., integers or real numbers within a certain range). Once function R is estimated, a recommender system can recommend the highest-rated item (or a set of N highest-rated items) for each user. In summary, one goal of a typical recommender system is to correctly estimate the ratings of unrated items based on the given ratings; another goal is to find items that maximize the user's utility.

While single-rating recommender systems have been successful in a number of applications, multi-criteria rating systems are being more and more commonly employed in many industries. Restaurant guides, such as Zagat's Guide, provide three criteria for restaurant ratings

[1] A recent survey of recommender systems research literature can be found in (Adomavicius & Tuzhilin 2005).

(e.g., food, décor, and service). Online shopping malls, such as Circuitcity.com and Buy.com, use multi-criteria ratings for consumer electronics (e.g., display, performance, battery life, and cost). Note that the aforementioned multi-criteria rating systems are not used in the context of personalization, i.e., the rating on each criterion is the same for all users (for example, the "food" rating for a specific restaurant published by Zagat's Guide) and not personalized to each individual consumer. In order to take full advantage of existing multi-criteria ratings in personalization applications, new recommendation techniques are required. For example, recently Yahoo! Movies launched a movie recommendation service that uses multi-criteria ratings for each movie, which indicates that multi-criteria data provides value both to online content providers and consumers and may become an important component in different personalization applications. Therefore, in this paper we propose several new approaches on how to extend recommendation technologies in order to incorporate and leverage *multi-criteria* rating information.

The rest of this paper is organized as follows. In Section 2, we briefly discuss some of the research related to multi-criteria ratings, including from the recommender systems literature. In Section 3, we provide some background on a traditional single-criterion collaborative filtering algorithm, which is used as an example throughout the paper. We then propose new recommendation techniques for multi-criteria ratings in Section 4. In Section 5, we describe our empirical analysis based on a real-world dataset. Finally, we conclude our paper in Section 6.

2. Related Work

Multi-criteria problems have been studied extensively in operation research and decision science fields. The majority of engineering problems are essentially multi-criteria optimization problems (Statnikov & Matusov 1995). For example, when an airplane is being designed, its reliability,

longevity, efficiency, cost, and the combination of other utilization factors need to be considered. Typical methods to solve the multi-criteria optimization problems include: finding Pareto optimal solutions; optimizing the most important criterion and converting other criteria to constraints; consecutively optimizing one criterion at a time, converting an optimal solution to constraints, and repeating the process for other criteria.

The decision science field treats organizational decision making as a multi-criteria problem, i.e., it considers various points of view, such as financial, human resources-related, and environmental aspects in making a decision (Figueria et al. 2005). The objective of multi-criteria decision analysis is to assist a decision maker in choosing the best alternative when multiple criteria conflict and compete with each other. Most commonly used decision aiding methods, such as outranking methods and the analytical hierarchy process, are based on multi-criteria aggregation procedures. Outranking methods determine which alternatives are preferred to others by systematically comparing possible alternatives on each criterion. The analytical hierarchy process structures multi-criteria into a hierarchy and calculates the score of each criterion as a weighted sum of its sub-criteria.

Similarly, in marketing research literature, buying a product also can be regarded as a multi-criteria decision problem. For example, when we purchase a car, we consider its multiple attributes, such as price, brand, and color. The conjoint model is most commonly used technique for solving multi-criteria problems in this field (Green et al. 2001). This model determines the importance weights of product attributes and the values of the attributes. The customers' preference for the product then can be calculated as a linear combination of weights and values.

Multi-criteria information is also used in certain electronic market mechanisms, such as multi-attribute auctions (Bichler 2000). Multi-attribute auctions are typically used in

procurement settings and enable auction participants negotiate not only on price, but also on other attributes of a deal (e.g., quality level, style, delivery date). It has been demonstrated that multi-attribute auctions have several advantages over their single-attribute (i.e., price-only) counterparts, including the improvements in the overall utility and suitability for various application domains (Bichler 2000).

However, the multi-criteria problems addressed in above-mentioned fields typically are not intended for personalization and recommendation settings. These problems find the solutions or items that are optimal in general (i.e., optimal with respect to all users), and differences in individual user preferences are not explicitly considered. Recently, multi-criteria rating problems have started receiving attention in recommender systems research and are regarded as one of the important issues for the next generation of recommender systems (Adomavicius & Tuzhilin 2005). In recommender systems literature, the roots of multi-criteria ratings could be traced to the approaches that started incorporating content-based features into collaborative recommendation techniques. This allowed the recommender systems to identify favorite content attributes (e.g., "comedy" movies) based on the content analysis of the previously rated items, and then also to recommend items to a user based not only on the ratings of similar users, but also based on these favorite content attributes (Balabanovic & Shoham 1997). However, the users were able to submit just a single rating for each item, and could not specify their individualized feedback about a specific movie component/aspect (such as movie's visual effects).

In addition, Ricci et al. (2002) developed a recommender system for personalizing travel using case-based reasoning techniques. The recommendations are performed by ranking and aggregating elementary items (locations, activities, services) based on the user's preferences and

a repository of past travels. While these techniques do not use multi-criteria ratings per se, the recommendation process does take into account multiple criteria, and the optimization is performed over a multidimensional solution space.

Furthermore, there has been some research on providing recommendation *filtering* capabilities based on item content information. For example, Schafer (2005) implements a meta-recommendation system that allows users to indicate the preference for each content attribute (e.g., movie genre, MPAA rating, or film length) and rate the importance of these attributes. For example, users can indicate that they want only "comedy" movies, and that it is the most important condition for recommendations – the users' requirements will filter the potential recommendations towards what the users really want. Note, however, that this does not represent a multi-criteria rating environment, since the users are specifying general filtering requirements for all movies (such as specifying the preferred value and weight for movie genre attribute). Similarly, Lee et al. (2002) also obtain the importance weights of content attributes directly from the user. They use each attribute's rank to compare the items, but the value or rank of each attribute is assumed to be the same for all users. In contrast to Schafer (2005) and Lee et al. (2002), in multi-criteria rating environment users would be able to specify subjective ratings for various components of *individual* items (e.g., to rate visual effects component for the "Star Wars" movie), which could then be leveraged for prediction and personalization purposes.

In summary, while there have been several different approaches discussed in personalization literature that are somewhat related to the issue of incorporating and leveraging multi-criteria ratings in recommender systems, it would be fair to say that this issue is largely unexplored. For this reason, in this paper we focus on new recommendation techniques for multi-criteria rating systems.

3. Background: Traditional Single-Rating Similarity-Based Collaborative Filtering Approach

Before proceeding with the discussion on new recommendation techniques for multi-criteria rating settings, we briefly describe one of the traditional and commonly used single-rating collaborative recommendation techniques, which we will use as an example throughout the paper.

Specifically, based on the recommender systems classification schemes mentioned in Section 1, let's consider the memory-based collaborative filtering technique that estimates $R(u, i)$ – the rating that user u would give to item i – by computing the weighted average of all known ratings $R(u', i)$, where user u' is "similar" to u. Two popular ways to compute this weighted average are (Breese et al. 1998):

- Weighted sum approach, i.e.,

$$R(u,i) = z \sum_{u' \in N(u)} sim(u,u') \cdot R(u',i); \tag{2}$$

- Adjusted weighted sum approach, i.e.,

$$R(u,i) = \overline{R(u)} + z \sum_{u' \in N(u)} sim(u,u') \cdot \left(R(u',i) - \overline{R(u')} \right). \tag{3}$$

Here the value of rating $R(u', i)$ is weighted by the similarity of user u' to user u – the more similar the two users are, the more weight $R(u', i)$ will have in the computation of rating $R(u, i)$. Furthermore, multiplier z serves as a normalizing factor and is usually set to $z = 1 / \sum_{u' \in N(u)} |sim(u,u')|$, $\overline{R(u)}$ represents the average rating of user u, and $N(u)$ represents the set of users that are similar to user u. The size of set $N(u)$ can range anywhere from 1 to all users in the dataset. Limiting the neighborhood size to some specific number (e.g., 3) will determine how many similar users will be used in the computation of rating $R(u, i)$.

Furthermore, there are several ways to compute similarity $sim(u, u')$ between two users, including *cosine-based* and *correlation-based* computations (Breese et al. 1998). We will use the cosine-based similarity in this paper, since it is arguably the most commonly used technique for determining how similar two users are in memory-based collaborative filtering algorithms. Assuming $I(u, u')$ represents the set of all items rated by both users u and u', the cosine-based similarity can be calculated as follows:

$$sim(u,u') = \left(\sum_{i \in I(u,u')} R(u,i)\ R(u',i) \right) \Bigg/ \left(\sqrt{\sum_{i \in I(u,u')} R(u,i)^2} \sqrt{\sum_{i \in I(u,u')} R(u',i)^2} \right) \qquad (4)$$

In addition, because of the inherent symmetry between users and items in the traditional memory-based collaborative filtering setting, this approach can be either *user-based* or *item-based*, depending on whether we want to calculate the similarity between users or items. Equations (2) and (3) represent the user-based approach, but they can be straightforwardly rewritten for the item-based approach. For example, the item-based adjusted weighted sum can be calculated as follows (Sarwar et al. 2001):

$$R(u,i) = \overline{R(i)} + z \sum_{i' \in N(i)} sim(i,i') \cdot \left(R(u,i') - \overline{R(i')} \right) \qquad (5)$$

and z, $\overline{R(i)}$, $sim(i, i')$, and $N(i)$ are analogous to their user-based counterparts.

In the rest of the paper, unless explicitly stated otherwise, by "traditional collaborative filtering approach" we will refer to the *user-based adjusted weighted sum* approach (3) that uses the *cosine-based* similarity function (4).

Finally, recommender systems typically recommend the items with the highest predicted rating to the user. In other words, recommenders often are not concerned about predicting the ratings of all items as accurately as possible, but rather about accurately predicting the highest-rated items, since users in real-world personalization applications are usually interested in

looking only at few highest-ranked item recommendations. Therefore, it is useful to evaluate the recommender system performance based on items that get the top N highest scores for each user, assuming, of course, that the values of the top N ratings are high enough to merit an actual recommendation. This is the evaluation approach that we adopt in this paper, as will be discussed in Section 5.

4. Extending Recommender Systems to Incorporate Multi-Criteria Ratings

In addition to the overall rating, multi-criteria ratings provide additional information about user preferences regarding several important aspects/components of an item. Leveraging this additional information in recommender systems should be beneficial, since it can potentially increase the accuracy of the recommendations. Therefore, new techniques are needed in order to effectively incorporate the multi-criteria rating information into the recommendation process.

The goal of multi-criteria recommender systems is to find items that maximize each user's utility, just as in the single-rating recommender systems. Therefore, one of the important goals of recommendation systems is to be able to predict the overall rating of each item for each user, because the system ultimately needs to compare the items based on their overall ratings and recommend the best items to the users. The difference between single-rating and multi-criteria rating systems is that the latter have more information about the users and items, which can be effectively used in the recommendation process. More formally, the general form of a rating function in a multi-criteria recommender system is:

$$R: Users \times Items \rightarrow R_0 \times R_1 \times \ldots \times R_k \qquad (6)$$

where R_0 is the set of possible overall rating values, and R_i represents the possible rating values for each individual criterion i ($i = 1, \ldots, k$), typically on some numeric scale (e.g., from 1 to 13).

In the remainder of this section we propose two new recommendation approaches and present several different variations of each. The first approach is designed to extend the traditional single-criteria memory-based collaborative filtering algorithm, while the second approach is not restricted to any specific algorithm. In other words, any existing single-criteria recommendation algorithm (i.e., content-based, collaborative, or hybrid) can be used in conjunction with this approach. And, as mentioned earlier, throughout the paper we will use one of the common user-based collaborative filtering techniques for illustration purposes.

4.1 Similarity-Based Approach to Extending Standard Collaborative Filtering Techniques

Consider a movie recommendation application, where users provide the recommender system with a single rating (between 1 and 13) for each movie they have seen. Moreover, suppose that this recommender system is using a traditional user-based collaborative filtering approach for rating prediction, as described in Section 3. In this case, according to Equation (3), any rating that user u would give to yet unseen movie i would be estimated based on how users u' that are similar to target user u rated movie i, i.e., unknown rating $R(u, i)$ is calculated based on ratings $R(u', i)$. Therefore, the more accurately the system determines who the "true peers" (or "nearest neighbors") of u are, the more accurate the rating prediction should be. The traditional (two-dimensional) collaborative filtering calculates the similarity between users u and u' based on how similar their ratings are for the movies they *both* have seen.

Figure 1 illustrates this estimation process with a simple example. Assume, that we have five users $u_1, \ldots, u_5$ and five movies $i_1, \ldots, i_5$. Furthermore, let's suppose that the recommender system needs to estimate how much the target user u_1 would like movie i_5 and, as indicated in Figure 1, that all other ratings of different users to different movies are known. Then, the traditional collaborative filtering approach finds the users that are closest to u_1 and that have seen

movie i_5. In this case, u_2 and u_3 seem to be "perfect matches" for user u_1, since all of them rated the common movies exactly the same (see Figure 1). Since both u_2 and u_3 rate movie i_5 as 9, the value of target rating $R(u_1, i_5)$ will be predicted as 9.

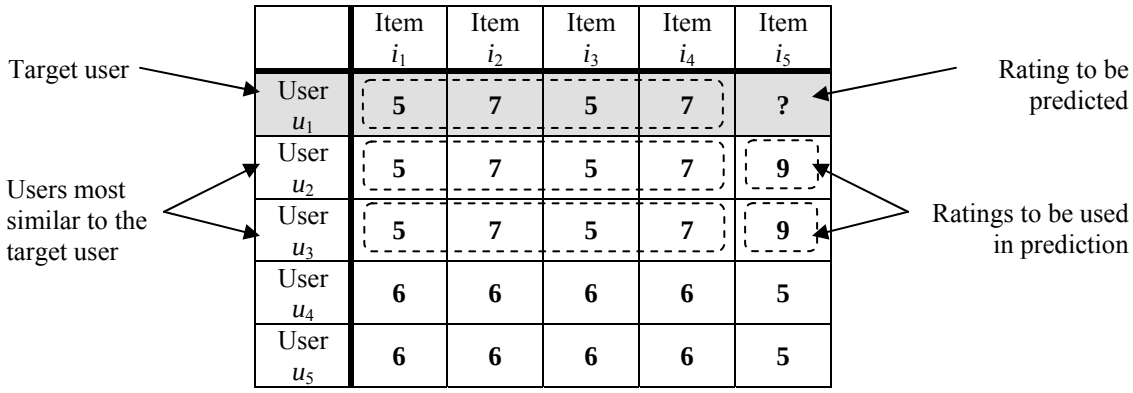

Figure 1. Collaborative filtering in a single-criteria setting.

Now let's consider the same scenario as above, but in a multi-criteria setting. Specifically, let's assume that we have the same five users u_1, …, u_5 and five movies i_1, …, i_5. Also, rating $R(u_1, i_5)$ is unknown and needs to be predicted, and, as indicated in Figure 2, all other overall ratings of different users to different movies are known and are exactly the same as before (in Figure 1). In addition, let's assume that each user is also asked to provide the feedback about the movie on four specific criteria: story, acting, direction, and visuals[2], and that the overall rating in this case is a simple average of the four individual criteria ratings.

Following the idea behind the standard collaborative filtering approach, in order to predict $R(u_1, i_5)$ the recommender system should find the users that are closest to u_1 and that have seen movie i_5. However, because of all the additional information that is available in the form of multi-criteria ratings, one can clearly see that users u_2 and u_3 are quite different in their tastes and preferences from user u_1, even though their overall ratings for each movie match perfectly. In

[2] As is done on some movie review websites, such as Yahoo! Movies (http://movies.yahoo.com).

particular, the movie aspects that u_1 hated (story and acting) were really liked by u_2 and u_3 and vice versa. However, in recommender systems that are based on single-criteria ratings, this information would be "hidden" within the aggregate overall rating, which may lead to inaccurate insights about the true similarity between user preferences (as in this example). Users u_4 and u_5 seem to be much better matches for user u_1 in this example, since not only their overall ratings are similar, but their preferences for different movie aspects were very similar as well (see Figure 2). Since both u_4 and u_5 rate movie i_5 as 5, the value of target rating $R(u_1, i_5)$ would be predicted as 5, which is a very different outcome from the one obtained in a single-criteria rating scenario.

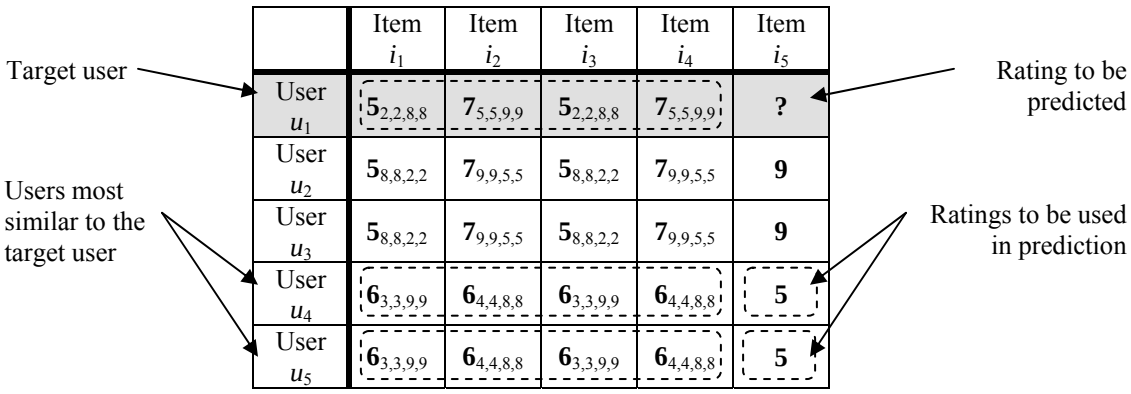

Figure 2. Collaborative filtering in a multi-criteria setting.

In summary, while the overall rating that a user gives to an item provides the information regarding *how much* the user liked the item, multi-criteria ratings provide some insights regarding *why* the user liked the item as much as she did. Therefore, having multi-criteria ratings provides the possibility to estimate the similarity between two users more accurately.

Based on this idea, we propose to extend the standard collaborative filtering algorithm to include multi-criteria ratings. Specifically, we propose several different ways to include multi-criteria rating information in the calculation of the similarity between two different users $sim(u, u')$ or two different items $sim(i, i')$. Then, given the newly calculated similarity, the rating

prediction can be done using the weighted sum or adjusted weighted sum in the same way as with a standard collaborative filtering algorithm, i.e., using Equations (2), (3), or (5). Below we describe two different approaches to leverage multi-criteria ratings in the similarity computation.

Aggregating traditional similarities that are based on each individual rating

This approach can use any standard similarity metric, such as cosine-based (4), and calculates the similarity between users (or items) based on each individual criteria. Let's assume that each rating that user u gives to item i consists of an "overall" rating r_0, and k multi-criteria ratings r_1, …, r_k, i.e.,

$$R(u, i) = (r_0, r_1, \ldots, r_k). \quad (7)$$

Then, k+1 different similarity estimations can be obtained by using some standard metric to measure similarity between users u and u': $sim_0(u, u')$ represents the similarity between u and u' based on the overall rating; $sim_1(u, u')$ – similarity based on the first criteria rating; $sim_2(u, u')$ – similarity based on the second criteria rating; and so on. The overall similarity then can be computed by aggregating the individual similarities in several ways:

- [Average similarity] By averaging all individual similarities, i.e.,

$$sim_{avg}(u,u') = \frac{1}{k+1}\sum_{i=0}^{k} sim_i(u,u'), \quad (8)$$

- [Worst-case similarity] By using the smallest of similarities, i.e.,

$$sim_{min}(u,u') = \min_{i=0,\ldots,k} sim_i(u,u'). \quad (9)$$

Calculating similarity using multidimensional distance metrics

In multi-criteria rating scenario, each rating $R(u, i) = (r_0, r_1, \ldots, r_k)$ represents a point in the k+1-dimensional space. Therefore, one natural approach to compute similarity between different users is to use multidimensional distance metrics. Such metrics are easy to understand and

straightforward to implement. Note that the metrics of distance and similarity are inversely related: the smaller the distance between two users, the higher the similarity. We calculate the similarity between two users in three steps.

First, we have to be able to calculate the distance between two users' ratings for the same item, i.e., $d_{\text{rating}}\left(R(u,i),R(u',i)\right)$, where $R(u,i)=(r_0,r_1,\ldots,r_k)$ and $R(u',i)=(r'_0,r'_1,\ldots,r'_k)$. For this purpose, any of the standard multidimensional distance metrics can be used:

- Manhattan distance: $\sum_{i=0}^{k}\left|r_i-r'_i\right|$; (10)
- Euclidean distance: $\sqrt{\sum_{i=0}^{k}\left|r_i-r'_i\right|^2}$; (11)
- Chebyshev (or maximal value) distance: $\max_{i=0,\ldots,k}\left|r_i-r'_i\right|$. (12)

Second, the overall distance between two users u and u' is simply:

$$d_{\text{user}}(u,u')=\frac{1}{\left|I(u,u')\right|}\sum_{i\in I(u,u')}d_{\text{rating}}\left(R(u,i),R(u',i)\right) \tag{13}$$

where $I(u, u')$ denotes the set of items that both u and u' have rated. In other words, the overall distance between two users u and u' is the average distance between their ratings for all their common items.

Finally, because the collaborative filtering techniques operate with the metric of user similarity (and not user distance), and the distance and similarity are inversely related, we use the simple transformation between the two metrics:

$$sim(u,u')=\frac{1}{1+d_{\text{user}}(u,u')}. \tag{14}$$

Note that this definition of similarity has desired range properties, i.e., the similarity will approach 0 as the distance between two users becomes larger, and it will be 1 if the distance is zero (users are identical).

In summary, both of the approaches presented in this section change only the similarity function in the traditional collaborative filtering technique in order to reflect multi-criteria rating information, which should result in a more accurate identification of similar users and, consequently, in better recommendation quality. We provide some empirical results in Section 5.

4.2 Aggregation Function Based Approach

Approaches to integrate multi-criteria rating information into recommender systems discussed in the previous section apply primarily to the similarity-based recommenders, such as traditional collaborative filtering techniques. In contrast, in this section we present a different approach that is not limited to any specific recommendation algorithm. The intuition behind this approach comes from the assumption that multi-criteria ratings represent user's preferences for the different important components of an item (e.g., story, acting, direction, and visuals aspects in the case of movie recommender systems). Thus, the overall rating of an item is not just another rating that is independent of others, but rather serves as some "aggregation" function f of the multi-criteria ratings of this item, i.e.,

$$r_0 = f(r_1, \ldots, r_k). \tag{15}$$

In other words, this approach assumes that the overall rating has a certain relationship with multi-criteria ratings. For instance, in a movie recommendation application, the story criteria rating may have a very high "priority", i.e., the movies with high story ratings are well liked overall by some users, regardless of other criteria ratings. Therefore, if the story rating of the movie is predicted to be high, the overall rating of the movie must also be predicted as high in order to be accurate.

The proposed approach to rating estimation consists of the following three steps, as illustrated in Figure 3. First, we decompose k-dimensional multi-criteria rating space into k

single-rating recommendation problems and use *any* traditional single-criteria recommendation technique to estimate ratings for each individual criterion. Second, we use statistical or machine learning techniques to estimate aggregation function *f* based on the known ratings. And third, using the multi-criteria ratings estimated in step 1 and function *f* estimated in step 2, we directly calculate the predicted overall rating. Below we discuss each of these steps in more detail.

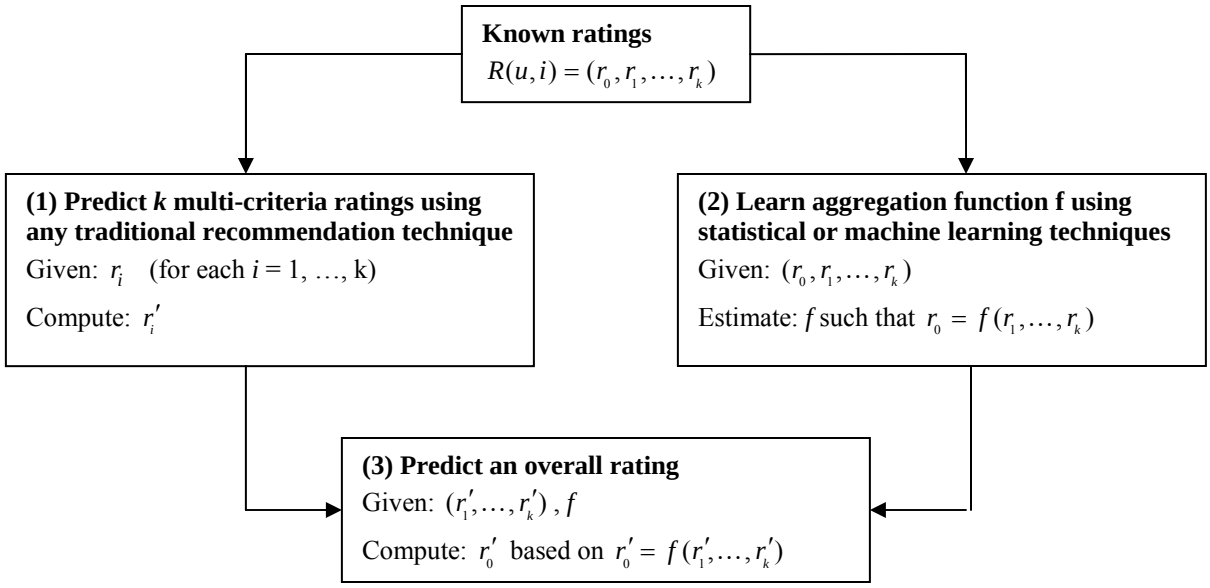

Figure 3. Overview of the aggregation-function-based approach.

Step 1: Predicting multi-criteria ratings

As mentioned earlier, we decompose *k*-dimensional multi-criteria rating space into a set of *k* single-rating recommendation problems, where each problem can be represented with a traditional *Users*×*Items* matrix (like the one in Figure 1) and addresses the rating prediction for one of the individual criteria. In other words, instead of the multi-criteria recommendation problem R:*Users*×*Items* $\rightarrow R_0 \times R_1 \times \ldots \times R_k$ we are dealing with *k* single-rating recommendation problems R:*Users*×*Items* $\rightarrow R_i$ (where i = 1, …, k). This approach provides a lot of flexibility, since (unlike with similarity-based approaches mentioned in previous section) *any* existing

single-criteria recommendation technique (e.g., collaborative, content-based, or hybrid) can be used to estimate unknown ratings for individual criteria.

Step 2: Learning the aggregation function

The goal of this step is to estimate relationship *f* between the overall rating and the underlying multi-criteria ratings of items, such that $r_0 = f(r_1, \ldots, r_k)$. We are already able to predict the individual multi-criteria ratings (see Step 1 above), but one of the important goals of recommendation systems is to be able predict the overall rating of each item for each user, which can be helpful in different situations. For example, having the overall rating for each item enables the recommender system to *rank* all items for each user in terms of their predicted utility (i.e., overall rating) and recommend only the most relevant items. In contrast, to determine the most relevant items without the presence of the overall rating, the recommender system would have to deal with a much more complex multi-criteria optimization problem (Statnikov & Matusov 1995). Thus, finding the aggregation function is crucial for recommender systems, and there are several ways in which this function could be obtained:

- *Domain expertise*. Based on her prior experience and knowledge of the domain, the domain expert may suggest the appropriate aggregation function. For example, it may be the case that the overall rating is a simple *average* of the underlying multi-criteria ratings for each item, i.e., $r_0 = (r_1 + \ldots + r_k)/k$.
- *Statistical techniques*, including various linear and non-linear regression analysis techniques. For example, in the case of *linear regression*, the aggregation function for the overall rating would be a linear combination of the multi-criteria ratings, i.e., $r_0 = w_1 r_1 + \ldots + w_k r_k + c$, where weight w_i associated with criterion *i* can be interpreted as

the importance of this criterion in determining the overall rating. The weights w_i (i = 1, …, k) and constant c are estimated based on the set of known ratings.

- *Machine learning techniques*. Various sophisticated computational learning techniques can also be used for this purpose, e.g., *artificial neural networks* (Mitchell 1997).

Besides the ability to use different learning techniques, the aggregation function can also be of three different *scopes*: total, user-based, or item-based. In particular, f is a *total* aggregation function if it is used to predict all unknown ratings, e.g., if the criteria weights w_i in a regression-based function mentioned above are consistent for all users and items. However, depending on the domain specifics, it may be useful to consider *user-based* or *item-based* aggregation functions in some applications. For example, in a movie recommender system, user u may have a much larger weight on the "story" component that is consistent for all movies, whereas user u' may have a significant weight on the "visuals" component. In this case, it would be advantageous for user u to have her own *user-based* aggregation function f_u, which would be learned exclusively from the known ratings of user u (as opposed to all known ratings) using the aforementioned techniques. Similarly, with the *item-based* aggregation function f_i we would assume that each item i will have its own aggregation function that is consistent for all the ratings involving this item.

Finally, note that a variety of different techniques are available for testing the fitness or accuracy of the predicted aggregation function(s). For example, in the case of linear regression, one can estimate the predictive power using its R^2 value. Or, more generally, one could use standard n-fold cross validation techniques to estimate the predictive accuracy of the aggregation function (Mitchell 1997). Therefore, we have the ability to restrict the use of user-based (or item-based) aggregation functions only to the ones that exhibit sufficient predictive performance,

e.g., whose accuracy is greater than some pre-specified threshold. The remaining users (or items) could use other techniques, e.g., the total aggregation function. As with every data-driven computational learning technique, there will be application domains where this approach will work well (i.e., domains where users/items exhibit consistent preferences on each criterion) and domains where other techniques will be more advantageous.

Step 3: Predicting overall ratings

Finally, as mentioned earlier, we compute each unknown overall rating r_0' directly by using the multi-criteria ratings estimated in step 1 and function f estimated in step 2: $r_0' = f(r_1', \ldots, r_k')$.

4.3 Other Benefits of Multi-Criteria Ratings in Recommender Systems

Up to this point, we have focused on how new techniques can potentially improve the estimation of overall ratings by leveraging multi-criteria rating information. In addition to this enhancement, the usage of multi-criteria ratings in recommender systems can provide other benefits to their users.

In particular, most recommender systems are inflexible in customizing recommendations according to user-specific requests. In other words, recommendations typically are fixed for all users (e.g., "provide 5 most relevant items to each user"), and cannot be adjusted by the users on the fly. There have been interesting attempts to provide recommendation filtering capabilities based on some item content information (see, for example, Schafer 2005); however, while undoubtedly useful, this filtering is typically done on the user-specified information that is fixed to an item and, therefore, same to all the users. For example, in a movie recommender system the users may be able to narrow their movie recommendations based on the movie genre, MPAA rating, film length, etc. (Schafer 2005). However, in a multi-criteria recommender system (similar to the one shown in Figure 2), a certain user may want to request only exceptionally

good "story" movies, where the "story" component of a movie is completely subjective to each user and, as mentioned earlier, is estimated individually for each user. Multi-criteria rating information would allow the recommender systems to respond to users' individual dynamic needs (e.g., expressed as filtering thresholds on individual criteria) in a more personalized manner and adjust the recommendations accordingly.

5. Experimental Results

To evaluate the proposed approaches, we have collected a set of user-submitted movie ratings from Yahoo! Movies website (movies.yahoo.com) for several hundred randomly chosen movies from the last decade. When a user submits movie ratings to Yahoo! Movies, in addition to the overall rating, she is asked to provide four criteria information for each movie: story, acting, direction, and visuals. All ratings have 13 possible values and are based on a standard grading scale from A+ to F; for the analysis purposes we changed them to numerical values from 13 to 1. In the data preprocessing stage, we invoked two constraints on the dataset in order to ensure that the dataset is not extremely sparse and has enough data for rating prediction: (a) there should be at least 10 movie ratings per user and (b) at least 10 user ratings per movie.

As a result, we ended up with a dataset that includes 155 users, 50 movies, and has 2,216 known ratings in total (28.6% of ratings are known). Each user has rated 14.3 movies on average, and the average number of common movies between two users is 5.2. Each movie has been rated on average by 44.3 users, and the average number of common users between two movies is 13.6. The average rating on each criterion is approximately 9 (or "B").

Furthermore, in order to obtain reliable results with a relatively small amount of data, we use a standard 10-fold cross validation technique (Mitchell 1997), where we randomly divide the dataset into 10 disjoint subsets. We use nine-tenths of the data for training, and the remaining

one-tenth for testing rating prediction, and then repeat this process 10 times (each time with a different test dataset) and perform the evaluation on all predicted ratings.

Numerous metrics for evaluating the performance of recommender systems have been proposed and used in the research literature (Herlocker et al. 2004), including the statistical accuracy metrics (e.g., mean absolute error and root mean squared error) as well as decision-support measures that determine how well the recommendation algorithm can predict high-relevance items (i.e., items that would be rated highly by the user). Examples of decision-support metrics include precision (the percentage of truly "high" ratings among those that were predicted to be "high" by the recommender system), recall (the percentage of correctly predicted "high" ratings among all the ratings known to be "high"), and F-measure, which is a harmonic mean of precision and recall (Herlocker et al. 2004).

In this paper, we have focused on the popular variation of the above-mentioned precision metric, i.e., *precision-in-top-N*, which represents the percentage of truly "high" overall ratings among those that were predicted to be *N* most relevant items for each user. This metric was chosen because of its practicality, since many users in real-life personalization and recommendation applications are typically interested in looking only at few highest-ranked item recommendations.

Because precision-related metrics measure the frequency with which a recommender system makes correct decisions about whether an item that is predicted as "highly-ranked" is truly "highly-ranked," we needed to define what "highly-ranked" means in our application. In other words, every rating had to be defined on a binary scale, i.e., as "highly-ranked" or "non-highly-ranked". Since Yahoo! Movies' rating scale (from A+ to F) was not binary, we translated the overall movie ratings into a binary scale by treating the ratings greater than 10.5 (A+, A, A-)

as "highly-ranked" and ratings less than 10.5 as "non-highly-ranked." The threshold of 10.5 was chosen with the assumption that the users would really want to focus on the recommendations about movies that are most relevant to them (i.e., movies they would rate as A+, A, A-), and therefore the correctness of recommendations for such movies is most desirable.

Also note that, in our dataset, the percentage of the "highly-ranked" ratings (i.e., overall ratings above 10.5) was 35.6%, which means that it would be possible to obtain the precision of 35.6% simply by recommending items at random. Any recommender system that does not achieve 35.6% precision would be worse than a random guess and, therefore, essentially useless.

In order to illustrate the performance of the proposed multi-criteria recommendation techniques on real-life data, we performed the empirical analysis of the following five approaches using the above-mentioned movie data (as summarized in Table 1):

- *standard CF* – a traditional single-rating user-based CF approach, which uses adjusted weighted sum and the cosine similarity metric, as described in (3) and (4). This approach is used as a baseline to illustrate the performance of multi-criteria recommendation approaches, as compared to a single-rating recommender system.
- Two similarity-based techniques (as described in Section 4.1) implemented with the traditional user-based CF approach:
    - *cos-min* – an example of a technique that aggregates traditional cosine-based similarities for each individual rating.
    - *Chebyshev* – an example of a technique that uses Chebyshev multidimensional distance metric.
- Two aggregation-function-based techniques (as described in Section 4.2), where individual multi-criteria ratings are estimated using the traditional user-based CF

approach:

- *total-reg* – an example of total aggregation function that is based on linear regression.
- *movie-reg95* – an example of item-based aggregation function that is generated separately for each movie and restricted only for the movies that have the best regression fit (i.e., $R^2 \geq 95\%$).

Note that we use the standard user-based collaborative filtering approach as an integral part of every technique in order to minimize the non-essential differences between the techniques as much as possible and, thus, to maximize the possibility that any differences in performance between the *standard CF* and multi-criteria recommender systems are due to the newly introduced multi-criteria rating information.

Recommendation Approach: user-based CF		**Precision in top 3 (%)**	**Precision in top 5 (%)**	**Precision in top 7 (%)**
Neighborhood size: ALL users	standard CF	70.7	68.7	69.0
	cos-min	70.7	68.8	69.1
	Chebyshev	**74.5**	**70.3**	**70.5**
	total-reg	71.5	**70.9**	**70.4**
	movie-reg95	**71.8**	**74.0*** **	**75.3*** **
Neighborhood size: 3 users	standard CF	64.9	64.9	66.3
	cos-min	**67.1**	**67.1**	**67.8**
	Chebyshev	**66.2**	65.5	64.6
	total-reg	65.2	**66.6**	66.5
	movie-reg95	**69.0*** **	**70.7*** **	**72.2*** **

Table 1. Main experimental results of several recommendation approaches.

Furthermore, for the sake of completeness, we provide results for different CF neighborhood sizes (the neighborhood of all users vs. the neighborhood of the 3 most similar users) and for different precision-in-top-*N* levels (*N* = 3, 5, and 7). The results are summarized in Table 1. The shaded cells represent the performance of the baseline CF approach. Note that nearly every multi-criteria technique performed either better or at least as well as the baseline

technique. The precision figures in regular font represent 0%–1% improvement over the baseline approach. The boldface precision figures represent 1%–4% improvement over the baseline approach, and the boldface precision figures marked with *** represent >4% improvement over the baseline approach. For further comparison, we have also calculated the precision-in-top-*N* metric for a simple "popularity-based" recommendation approach, where each user is recommended *N* movies (*N* = 3, 5, and 7) that are most liked by all other users, based on the average rating for each movie. The results show that the precision-in-top-*N* for this simple approach is: 61.3% (top 3), 53.3% (top 5), and 46.4% (top 7), which performs better than a "random guess" approach mentioned earlier but not as well as collaborative filtering techniques.

Among other notable results:

- For user-based CF, *precision-in-top*-1 measures (as opposed to top-3, top-5, and top-7 measures in Table 1) for various neighborhood sizes were dominated by similarity-based techniques (such as *Chebyshev* and *cos-min*), which typically outperformed both the baseline approach and the aggregation function-based approaches by 2%-6%.
- We also tried movie-based CF (as opposed to user-based CF, as in Table 1), for which *total-reg* performed the best of all the techniques for various neighborhood sizes and typically outperformed the baseline approach by 1%-5%.
- Combining similarity-based and aggregation-function-based multi-criteria recommendation techniques can sometimes improve the predictive performance, which is generally consistent with similar findings in recommender systems literature about the advantages of combining different types of recommender systems (e.g., it has been widely reported that combining content-based and collaborative systems may improve the recommendation accuracy).

As with most recommender systems and, more generally, computational learning techniques, the performance of a specific technique is highly domain-dependent. In other words, its performance depends significantly on the characteristics of the underlying data. Thus, while we expect the proposed techniques to do well in a variety of different application domains, multi-criteria recommendation techniques cannot be expected to have an advantage over traditional single-criterion techniques in all domains where multi-criteria information exists, especially in the ones where multi-criteria ratings do not carry meaningful information or where is no inherent relationship between the overall rating and multi-criteria ratings for the users or items.

6. Conclusions

While single-rating recommender systems have been successful in a number of personalization applications, multi-criteria rating systems are getting to be commonly deployed in many industries. However, in order to take full advantage of existing multi-criteria ratings in personalization applications, new recommendation techniques are required. In this paper, we propose two new recommendation approaches – the similarity-based approach and the aggregation-function-based approach – to incorporating and leveraging multi-criteria rating information. Our experimental results on a real-world dataset confirm that, when available, multi-criteria ratings can be successfully leveraged to improve recommendation accuracy. We expect that the proposed approaches will be useful in other application domains as well, where they will be able to predict overall ratings more accurately by utilizing the available multi-criteria rating information.

The area of recommender systems has made significant progress over the last few years; many techniques have been proposed and many systems have been developed. However, modern recommender systems still require further significant improvements in order to provide

better recommendations and be viable in more complex personalization applications; the ability to leverage multi-criteria rating information constitutes one such improvement. We believe that this paper is just the first step in studying multi-criteria recommender systems and that significant additional work is needed to further explore this issue.

Acknowledgments

The research reported in this paper was supported in part by the National Science Foundation grant IIS-0546443.

References

G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, Jun. 2005.

M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.

M. Bichler, "An experimental analysis of multi-attribute auctions," *Decision Support Systems*, vol. 29, no. 10, pp. 249-268, 2000.

J. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *Proc. 14th Conf. on Uncertainty in Artificial Intelligence*, July 1998.

J. Figueria, S. Greco, and M.Ehrgott, *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer, 2005.

P. E. Green, A. M. Krieger, and Y.Wind, "Thirty years of conjoint analysis: Reflections and Prospects," *Interfaces*, vol. 31, no. 3, pp. 56-73, 2001.

J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5-53, 2004

W. Lee, C. Liu, and C. Lu, "Intelligent agent-based systems for personalized recommendations in Internet commerce," *Expert Systems with Applications*, vol. 22, no. 4, pp. 275-184, May 2002.

T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

F. Ricci and H. Werthner, "Case-Based Querying for Travel Planning Recommendation," *Information Technology and Tourism*, vol. 4, nos. 3–4, pp. 215–226, 2002.

B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," *Proc. 10th Int'l WWW Conf.*, 2001.

J. B. Schafer, "DynamicLens: A Dynamic User-Interface for a Meta-Recommendation systems," Beyond personalization 2005: A workshop on the next stage of recommender systems research at the *ACM Intelligent User Interfaces Conf.*, Jan. 2005.

R. B. Statnikov and J.B. Matusov, *Multicriteria Optimization and Engineering*, Chapman & Hall, 1995.