Chapter 25
Multi-Criteria Recommender Systems

Gediminas Adomavicius and YoungOk Kwon

25.1 Introduction

The research discipline of recommender systems arose to address the problem of information or choice over-abundance, i.e., to help users find information or items that are most likely to be interesting to them or to be relevant to their needs [4, 7, 12, 38, 39, 73, 74]. Typically, the recommendation problem assumes that there is set *Users* of all the users of a system and set *Items* of all possible items that can be recommended to them. Then, the utility function that measures the appropriateness of recommending item $i \in Items$ to user $u \in Users$ is often defined as $R : Users \times Items \rightarrow R_0$. R_0 typically represents users' possible *preference ratings* for items (e.g., non-negative integers or real numbers within a certain range). The goal of recommender systems is, for each user $u \in Users$, to be able to (a) accurately estimate (or approximate) utility function $R(u, i)$ for item $i \in Items$ for which $R(u, i)$ is not yet known, and then (b) select one or a set of items i for which the predicted value $R(u, i)$ is high (i.e., items that are predicted to be relevant for u) and also possibly satisfy some other desirable conditions (e.g., items with high novelty or diversity [31, 88]).

In most recommender systems, utility function $R(u, i)$ usually estimates a *single-criterion* value, e.g., an overall evaluation or rating of an item by a user. In some recent work, this assumption has been considered as limited [2, 4, 51], because the

---

G. Adomavicius (✉)
Department of Information and Decision Sciences, University of Minnesota,
321 19th Avenue South, Minneapolis, MN 55455, USA
e-mail: gedas@umn.edu

Y. Kwon
Sookmyung Women's University, Cheongpa-ro 47-gil 100,
Yongsan-gu, Seoul 140-742, Korea
e-mail: yokwon@sm.ac.kr

F. Ricci et al. (eds.), *Recommender Systems Handbook*,
DOI 10.1007/978-1-4899-7637-6_25

suitability of the recommended item for a particular user may depend on more than one utility-related aspect that the user takes into consideration when making the choice. Particularly in systems where recommendations are based on the opinion of others, the incorporation of multiple criteria that can affect the users' opinions may lead to more accurate recommendations.

Thus, the additional information provided by *multi-criteria ratings* could help to improve the quality of recommendations because it would be able to represent more complex preferences of each user. As an illustration, consider the following example. In a traditional single-rating movie recommender system, user u provides a single rating for movie i that the user has seen, denoted by $R(u, i)$. Specifically, suppose that the recommender system predicts the rating of the movie that the user has not seen based on the movie ratings of other users with similar preferences, who are commonly referred to as "neighbors" [12, 73, 74]. For example, if two users u and u' have seen three movies in common, and both of them rated their overall satisfaction from each of the three movies as 6 out of 10, the two users are considered as neighbors and the ratings of unseen movies for user u are predicted using the ratings of user u'. Therefore, the ability to correctly determine the users that are most similar to the target user is crucial in order to have accurate predictions or recommendations.

In contrast, in a multi-criteria rating setting, users can provide their subjective preference ratings on multiple attributes of an item. For example, a two-criterion movie recommender system allows users to specify their preferences on two attributes of a movie (e.g., story and visual effects). A user may like the story, but dislike the visual effects of a movie, e.g., $R(u, i) = (9, 3)$. If we simply aggregate the two individual criteria ratings by giving them the same weight in making recommendations, rating the user's overall satisfaction as 6 out of 10 in the single-rating application might correspond to a variety of situations in multi-rating application: (9, 3), (6, 6), (4, 8), etc. Therefore, although the ratings of the overall satisfaction are stated as 6, two users may show different rating patterns on each criterion of an item, e.g., user u gives ratings (9, 3), (9, 3), (9, 3), and user u' gives ratings (3, 9), (3, 9), (3, 9) to the same three movies. This additional information on each user's preferences would help to model users' preferences more accurately, and new recommendation techniques need to be developed to take advantage of this additional information. The importance of studying multi-criteria recommender systems has been highlighted as a separate strand in the recommender systems literature [2, 4, 51], and recently several recommender systems (as we present later in this chapter) have been adopting multiple criteria ratings, instead of traditional single-criterion ratings. Thus, the aim of this chapter is to provide an overview of *multi-criteria recommender systems*.

The use of multi-criteria recommender systems has been proposed for a wide range of applications. As mentioned above, for experiential products (such as movies, books, and music) users may have varying subjective tastes and preferences for multiple product dimensions, and richer information on user preferences helps to improve the quality of recommendations [2, 42, 43, 68]. Other popular domains where multi-criteria recommendation algorithms can be applied include travel and tourism domains. Customers can have different preferences on friendliness, room size, service quality, and tidiness about the hotel, in addition to an overall perspective [34]. Mobile banking business can also adopt the multi-criteria algorithms by tracking each user's behavior data on the mobile service, rather than obtaining explicit ratings [93]. Furthermore, restaurants [44, 83] can be considered with different aspects such as the quality of service, location, value for money, and an overall experience. Similarly, research papers [58, 99] can be recommended with the information on multiple dimensions such as title, keywords, authors, publication year, and the citation links (i.e., representing the papers that cite the target paper as well as the papers cited by the target paper). Multi-criteria recommendation algorithms have also been used to support clinical decision making by combining evidence-based (i.e., disease information) and patient-centric (i.e., patient preferences) information [22].

Generally, recommendation techniques are often classified based on the recommendation approach into several categories: content-based, collaborative filtering, knowledge-based, and hybrid approaches [4, 7]. Content-based recommendation techniques find the best recommendations for a user based on what the user liked in the past [48, 69], and collaborative filtering recommendation techniques make recommendations based on the information about other users with similar preferences [12, 41, 73, 74]. Knowledge-based approaches use knowledge about users and items to find the items that meet users' requirements [14, 17]. The bottleneck of this knowledge-based approach is that it needs to acquire a knowledge base beforehand, but the obtained knowledge base helps to avoid cold start or data sparsity problems that pure content-based or collaborative filtering systems encounter by relying on solely the ratings obtained by users. Hybrid approaches combine content-based, collaborative filtering, and knowledge-based techniques in many different ways [15, 16].

Multi-criteria recommender systems can employ any of these general approaches. However, it is important to note that "multi-criteria" is a very generic term, and we observe that in research literature "multi-criteria recommender systems" may point to several substantially different ideas, including:

- Multi-attribute content search, filtering, and preference modeling;
- Multi-objective recommendation strategies;
- Multi-criteria rating-based preference elicitation.

Below we provide a brief overview of these three categories.

Multi-Attribute Content Search, Filtering, and Preference Modeling These approaches allow the user to specify her current preferences or needs based on various content-based attributes across all items, through searching or filtering processes (e.g., searching for only comedy movies) or by pre-specifying her "favorite" content attributes (e.g., indicating favorite actors or the fact that comedy movies are preferable to action movies), and recommend to the user the items that are the most similar to her preferences and satisfy specified search and/or filtering conditions. Therefore, even though there are some aspects of "multi-criteria" nature due to multiple content attributes, most of these approaches are well represented by the existing paradigms of content-based, hybrid, conversational, case-based reasoning, and some knowledge-based recommender systems as well as traditional information retrieval approaches. For example, several case-based travel recommender systems [76, 77] filter out unwanted items based on each user's preferences on multi-attribute content (e.g., locations, services, and activities), and find personalized travel plans for each user by ranking possible travel plans based on the user's preferences and past travel plans of this or similar users. In addition, some case-based recommender systems [14, 72] allow users to "critique" the recommendation results by refining their requirements as part of the interactive and iterative recommendation process, which uses various search and filtering techniques to continuously provide the user with the updated set of recommendations. For example, when searching for a desktop PC, users can critique the current set of provided recommendations by expressing their refined preferences on individual features (e.g., cheaper price) or multiple features together (e.g., higher processor speed, RAM, and hard-disk capacity). An entire research stream of *conversational recommender systems* is dedicated to these types of approaches [8, 9, 13, 37, 95–97]. Some additional examples of related approaches can be found in Chap. 5.

Multi-Objective Recommendation Strategies Traditionally, the main focus of recommender systems research has been on developing recommendation algorithms that provide *accurate* recommendations, where accuracy can be evaluated using a variety of different measures, such as MAE, RMSE, precision, recall, F-measure, normalized discounted cumulative gain (NDCG), and many others, depending on recommendation task. However, understanding that recommendation accuracy may not always completely align with recommendation usefulness, researchers have been proposing a number of alternative measures, including coverage, diversity, novelty, serendipity, and several others, to evaluate the performance of recommender systems. As a result, modern recommender systems implementations may use multiple *performance* criteria when deciding on the final set of recommendations to be shown to a given user, e.g., using accuracy, diversity, and freshness recommendation criteria in Netflix movie recommendations [6]. In summary, the "multi-criteria" nature in such approaches arises not from the attempts to represent more complex user preferences but rather from optimizing multiple different recommendation performance objectives. This type of work is well represented in recommender systems research stream on performance metrics and evaluation [31, 75, 88].

Multi-Criteria Rating-Based Preference Elicitation This category of recommender systems engage multi-criteria ratings, often by extending traditional collaborative filtering approaches, that represent users' *subjective* preferences for various *components* of individual items. For instance, such systems allow users to rate not only the overall satisfaction from a particular movie, but also the satisfaction from the various movie components (factors), such as the visual effects, the story, or the acting. In other words, these approaches allow a user to specify her individual preferences in a more precise and nuanced manner by rating each item on multiple criteria (e.g., rating the story of movie "Avatar" as 3, and the visual effects of the same movie as 5), and then are able to leverage this more sophisticated preference information in item recommendations. These approaches differ from the multi-attribute content approaches in that the users do not indicate their preference or importance weight on the visual effects component for movies in general or to be used in a particular user query, but rather *how much* they liked the visual effects of the *particular* movie. One example of early research in this area is the Intelligent Travel Recommender system [76], where users can rate multiple travel items within a "travel bag" (e.g., location, accommodation, etc.) as well as the entire travel bag. Then, candidate travel plans are ranked according to these user ratings, and the system finds the best match between recommended travel plans and the current needs of a user. These and similar types of multi-criteria rating-based systems are the focus of this chapter and more exemplar systems and techniques are provided in the later sections.

In summary, as seen above, a number of recommendation approaches that employ traditional content-based, knowledge-based, collaborative filtering, and hybrid techniques can be viewed as multi-criteria recommender systems in some way or another. Some of these approaches model user preferences based on multi-attribute content of items that users preferred in the past, others allow users to specify their current content-related preferences as search or filtering conditions, and yet others try to provide recommendations by balancing several performance metrics at once. However, as mentioned earlier, there is a recent trend in multi-criteria recommendation that studies innovative approaches in collaborative recommendation by attempting to capture and model user preferences in a more comprehensive, more nuanced manner by engaging multi-criteria ratings. We believe that this additional information on users' preferences offers many opportunities for providing novel recommendation support, creating a unique multi-criteria rating environment that has not been extensively researched. Therefore, in the following sections, we survey the state-of-the-art techniques on this particular type of systems that use individual ratings along multiple criteria, which we will refer to as *multi-criteria rating recommenders*.

The remainder of this chapter is organized as follows. In Sect. 25.2, we overview the particular type of multi-criteria recommender systems that use multi-criteria ratings, referred to as *multi-criteria rating recommenders*. In Sects. 25.3 and 25.4, we survey the state-of-the-art algorithms that are used in this type of recommenders for rating prediction and recommendation generation. Finally, Sect. 25.5 discusses research challenges and future research directions for multi-criteria recommender systems, followed by brief conclusions in Sect. 25.6.

25.2 Multi-Criteria Rating Recommendation

In this section, we define the multi-criteria rating recommendation problem by formally extending it from its single-rating counterpart (for more details on traditional single-rating recommender systems refer to Chaps. 2, 3 and 7), and provide some further discussion about the advantages that additional criteria may provide in recommender systems.

25.2.1 Traditional Single-Rating Recommendation Problem

Traditionally recommender systems operate in a two-dimensional space of *Users* and *Items*. The utility of items to users is generally represented by a totally ordered set of ratings R_0. The ratings can be unary (e.g., purchases), binary (e.g., like vs. dislike, high vs. low, good vs. bad), small set of ordered discrete values (e.g., 1-star, 2-stars, ..., 5-stars), or numbers within a certain range (e.g., $[-10, 10]$) [4]. In most recommendation applications, function R is explicitly known only for some subset of the $Users \times Items$ space, e.g., for the items that users have previously consumed and have provided their preference ratings for, and that the majority of the $Users \times Items$ space is unknown. Recommender systems aim to predict the utility of an item for a user. As mentioned earlier, a utility function R can be formally written as follows:

$$R : Users \times Items \to R_0 \tag{25.1}$$

The utility function is determined based on user inputs, such as numeric ratings that users explicitly give to items and/or transaction data that implicitly shows users' preferences (e.g., purchase history). The majority of traditional recommender systems use single-criterion ratings that indicate how much a given user liked a particular item in total (i.e., the overall utility of an item by a user). For example, in a movie recommender system, as shown in Fig. 25.1, user *Alice* may assign a single-criterion rating of 5 (out of 10) for movie *Wanted*, which can be denoted by $R(Alice, Wanted) = 5$. As an illustration, let us assume that the neighborhood-based collaborative filtering technique [73], i.e., one of the most popular heuristic-based recommendation techniques, is used for rating prediction. This technique predicts a user's rating for a given item based on the ratings of other users with similar preferences (i.e., neighbors). Particularly, in this example, the recommender system tries to predict the utility of movie *Fargo* for *Alice* based on the observed ratings. Since *Alice* and *John* show similar rating patterns on the four movies that both of them have previously seen and rated (see Fig. 25.1), for the purpose of this simple example the rating of movie *Fargo* for user *Alice* is predicted using *John*'s rating (i.e., 9), although we would like to note that it is more common to use the ratings of more than one neighbor in a real system.

		Wanted	WALL-E	Star Wars	Seven	Fargo	
Target user	Alice	5	7	5	7	?	Ratings to be predicted
User most similar to the target user	John	5	7	5	7	9	Ratings to be used in prediction
	Mason	6	6	6	6	5	
	⋮	⋮	⋮	⋮	⋮	⋮	

Fig. 25.1 Single-rating movie recommender system

25.2.2 Extending Traditional Recommender Systems to Include Multi-Criteria Ratings

With a growing number of real-world applications, extending recommendation techniques to incorporate multi-criteria ratings has been regarded as one of the important issues for the next generation of recommender systems [4]. Examples of multi-criteria rating systems include Zagat's Guide that provides three criteria for restaurant ratings (e.g., food, décor, and service), Buy.com that provides multi-criteria ratings for consumer electronics (e.g., display size, performance, battery life, and cost), and Yahoo! Movies that show each user's ratings for four criteria (e.g., story, action, direction, and visuals). This additional information about users' preferences provided by multi-criteria ratings (instead of a single overall rating) can potentially be helpful in improving the performance of recommender systems.

Some multi-criteria rating systems can choose to model a user's utility for a given item with an overall rating R_0 as well as the user's ratings $R_1, \ldots, R_k$ for each individual criterion c ($c = 1, \ldots, k$), whereas some systems can choose not to use the overall rating and focus solely on individual criteria ratings. Therefore, the utility-based formulation of the multi-criteria recommendation problem can be represented either with or without overall ratings as follows:

$$R : Users \times Items \rightarrow R_0 \times R_1 \times \cdots \times R_k \tag{25.2}$$

or

$$R : Users \times Items \rightarrow R_1 \times \cdots \times R_k \tag{25.3}$$

Given the availability of multi-criteria ratings (in addition to the traditional single overall rating) for each item, Figs. 25.1 and 25.2 illustrate the potential benefits of this information for recommender systems. While *Alice* and *John* have similar preferences on movies in a single-rating setting (Fig. 25.1), in a multi-criteria rating setting we could see that they show substantially different preferences on several movie aspects, even though they had the same overall ratings (Fig. 25.2). Upon further inspection of all the multi-criteria rating information, one can see that *Alice*

		Wanted	WALL-E	Star Wars	Seven	Fargo	
Target user	Alice	$5_{2,2,8,8}$	$7_{5,5,9,9}$	$5_{2,2,8,8}$	$7_{5,5,9,9}$	$?_{?,?,?,?}$	Ratings to be predicted
User most similar to the target user	John	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$5_{8,8,2,2}$	$7_{9,9,5,5}$	$9_{8,8,10,10}$	
	Mason	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$6_{3,3,9,9}$	$6_{4,4,8,8}$	$5_{2,2,8,8}$	Ratings to be used in prediction
	:	:	:	:	:	:	

Fig. 25.2 Multi-criteria movie recommender system (ratings for each item: overall, story, action, direction, and visual effects)

and *Mason* show very similar rating patterns (much more similar than *Alice* and *John*). Thus, using the same collaborative filtering approach as before, but taking into account multi-criteria ratings, *Alice*'s overall rating for movie *Fargo* would be predicted as 5, based on *Mason*'s overall rating for this movie.

This example implies that a single overall rating may hide the underlying heterogeneity of users' preferences for different aspects of a given item, and multi-criteria ratings may help to better understand each user's preferences, as a result enabling to provide users more accurate recommendations. It also illustrates how multi-criteria ratings can potentially produce more powerful and focused recommendations, e.g., by recommending movies that will score best on the story criterion, if this is the most important one for some user.

Therefore, new recommendation algorithms and techniques are needed that can utilize multi-criteria ratings in recommender systems. Since recommender systems typically calculate and provide recommendations using the following two-phase process, i.e., rating prediction phase and recommendation generation phase, multi-criteria rating information can be used in both of these phases in different ways. A number of approaches have been developed for the prediction or recommendation and there are already several systems implementing such algorithms, which we analyze in the next two sections.

- *Prediction*: the phase in which the prediction of a user's preference is calculated. Traditionally, it is the phase in which a recommender estimates the utility function R for the entire or some part of $Users \times Items$ space based on known ratings and possibly other information (such as user profiles and/or item content); in other words, it calculates the predictions of ratings for the unknown items.
- *Recommendation*: the phase in which the calculated prediction is used to support the user's decision by some recommendation process, e.g., the phase in which the user gets recommended a set of top-N items that maximize his/her utility (such as recommend N items with highly predicted ratings and that also satisfy some additional desirable requirements, e.g., related to item diversity or novelty).

We first classify the existing techniques for multi-criteria rating recommenders into two groups—techniques used during rating prediction and techniques used during recommendation generation—and describe these groups in more detail in the next two sections. The overview of these techniques is presented in Table 25.1.

Table 25.1 Techniques for multi-criteria rating recommenders

<table>
<tr><th>Phase of the recommendation process</th><th colspan="2">Recommendation techniques</th></tr>
<tr><td rowspan="2">Rating prediction</td><td>Heuristic-based approaches</td><td>Model-based approaches</td></tr>
<tr><td>Using multi-criteria ratings to improve user-user or item-item similarity calculation in neighborhood-based collaborative filtering:
• Calculate similarity values on each criterion, aggregate individual similarities into a single similarity (possibly using importance weights for each criterion)
• Calculate similarity values using multidimensional distance metrics directly on multi-criteria rating vectors
Heuristic rating prediction using fuzzy modeling:
• Fuzzy linguistic modeling
• Fuzzy multi-criteria preference aggregation</td><td>Building predictive models to estimate unknown ratings given multi-criteria rating data
• Typical approach: build models to aggregating individual criteria ratings into one overall rating
Representative model-based approaches:
• Simple aggregation functions: simple average, linear regression
• Probabilistic modeling: flexible mixture models, probabilistic latent semantic analysis
• Multi-linear singular value decomposition (MSVD)
• Complex aggregation functions: support vector regression (SVR)</td></tr>
<tr><td>Item recommendation (i.e., determining the best items)</td><td colspan="2">When the overall rating is available (among the multi-criteria ratings)
• Typical approach: rank items by their predicted overall rating
When the overall rating is not available:
• Design a total order for item recommendations, e.g., UTA approach
• Find Pareto optimal item recommendations, e.g., data envelopment analysis, skyline queries
• Use individual rating criteria as recommendation filters</td></tr>
</table>

25.3 Engaging Multi-Criteria Ratings During Prediction

This section provides an overview of the techniques that use multi-criteria ratings to predict an overall rating or individual criteria ratings (or both). In general, recommendation techniques can be classified by the formation of the utility function into two categories: heuristic-based (sometimes also referred to as memory-based) and model-based techniques [4, 12]. Heuristic-based techniques compute the utility of each item for a user on the fly based on the observed data of the user and are typically based on a certain heuristic assumption. For example, a neighborhood-based technique—one of the most popular heuristic-based collaborative filtering techniques—assumes that two users who show similar preferences on the observed items will have similar preferences for the unobserved items as well. In contrast, model-based techniques learn a predictive model, typically using statistical or machine-learning methods, that can best explain the observed data, and then use the learned model to estimate the utility of unknown items for recommendations. Following this classification, we also present the algorithms of multi-criteria rating recommenders by grouping them into heuristic and model-based approaches.

25.3.1 Heuristic Approaches

There has been some work done to extend the *similarity* computation of the traditional heuristic-based collaborative filtering technique to reflect multi-criteria rating information [2, 52, 92]. In this approach, the similarities between users are computed by aggregating traditional similarities from individual criteria or using multidimensional distance metrics. Note that this approach changes only the similarity calculation component of traditional recommendation algorithms; once the similarity is estimated, the overall rating calculation process remains the same.

In particular, the neighborhood-based collaborative filtering recommendation technique predicts unknown ratings for a given user, based on the known ratings of the other users with similar preferences or tastes (i.e., neighbors). Therefore, the first step of the prediction processes is to choose the similarity computation method to find a set of neighbors for each user. Various methods have been used for similarity computation in single-criterion rating recommender systems, and the most popular methods are correlation-based and cosine-based. $R(u,i)$ represents the rating that user u gives to item i, and $\overline{R(u)}$ represents the average rating of user u. Assuming that $I(u,u')$ represents the common items that two users u and u' rated, two popular similarity measures can be formally written as follows:

- *Pearson correlation-based*:

$$sim(u,u') = \frac{\sum_{i \in I(u,u')} (R(u,i) - \overline{R(u)})(R(u',i) - \overline{R(u')})}{\sqrt{\sum_{i \in I(u,u')} (R(u,i) - \overline{R(u)})^2} \sqrt{\sum_{i \in I(u,u')} (R(u',i) - \overline{R(u')})^2}} \tag{25.4}$$

- *Cosine-based*:

$$sim(u,u') = \frac{\sum_{i \in I(u,u')} R(u,i) R(u',i)}{\sqrt{\sum_{i \in I(u,u')} R(u,i)^2} \sqrt{\sum_{i \in I(u,u')} R(u',i)^2}} \tag{25.5}$$

Multi-criteria rating recommenders cannot directly employ the above formulas, because $R(u,i)$ contains an overall rating r_0, and k multi-criteria ratings $r_1, \ldots, r_k$, i.e., $R(u,i) = (r_0, r_1, \ldots, r_k)$.[1] Thus, there are k+1 rating values for each pair of (u,i), instead of a single rating. Two different similarity-based approaches that use $k+1$ rating values in computing similarities between users have been used. The first approach *aggregates traditional similarities that are based on each individual rating*. This approach first computes the similarity between two users separately on each individual criterion, using any traditional similarity computation, such as correlation-based and cosine-based similarity. Then, a final similarity between two users is obtained by aggregating $k+1$ individual similarity values. Adomavicius and Kwon [2] propose two aggregation approaches: an average and the worst-case (i.e., smallest) similarity, as specified in (25.6) and (25.7). As a general approach, Tang and McCalla [92], in their recommender system of research papers, compute an aggregate similarity as a weighted sum of individual similarities over several criteria of each paper (e.g., overall rating, value added, degree of being peer-recommended, and learners' pedagogical features such as interest and background knowledge) as specified in (25.8). In their approach, the weight of each criterion c, denoted by w_c, is chosen to reflect how important and useful the criterion is considered to be for the recommendation.

- *Average similarity*:

$$sim_{avg}(u,u') = \frac{1}{k+1} \sum_{c=0}^{k} sim_c(u,u') \tag{25.6}$$

- *Worst-case(smallest) similarity*:

$$sim_{min}(u,u') = \min_{c=0,\ldots,k} sim_c(u,u') \tag{25.7}$$

- *Aggregate similarity*:

$$sim_{aggregate}(u,u') = \sum_{c=0}^{k} w_c sim_c(u,u') \tag{25.8}$$

The second approach calculates similarity using *multidimensional distance metrics*, such as Manhattan, Euclidean, and Chebyshev distance metrics [2]. The distance between two users u and u' on item i, $d(R(u,i), R(u',i))$, can be calculated as:

- *Manhattan distance*:

$$\sum_{c=0}^{k} |r_c(u,i) - r_c(u',i)| \tag{25.9}$$

[1]In some recommender systems, $R(u,i)$ might not contain the overall ratings r_0 in addition to k multi-criteria ratings, i.e., $R(u,i) = (r_1, \ldots, r_k)$. In this case, all the formulas in this subsection will still be applicable with index $c \in \{1, \ldots, k\}$, as opposed to $c \in \{0, 1, \ldots, k\}$.

- *Euclidean distance*:

$$\sqrt{\sum\nolimits_{c=0}^{k} |r_c(u,i) - r_c(u',i)|^2} \tag{25.10}$$

- *Chebyshev (or maximal value) distance*:

$$\max_{c=0,\ldots,k} |r_c(u,i) - r_c(u',i)| \tag{25.11}$$

The overall distance between two users can be simply an average distance for all common items that both users rated, and it can be formally written as:

$$dist(u,u') = \frac{1}{|I(u,u')|} \sum\nolimits_{i \in I(u,u')} d(R(u,i), R(u',i)) \tag{25.12}$$

The more similar two users are (i.e., the larger the similarity value between them is), the smaller is the distance between them. Therefore, the following simple transformation is needed because of the inverse relationship of the two metrics:

$$sim(u,u') = \frac{1}{1 + dist(u,u')} \tag{25.13}$$

Manouselis and Costopoulou [52] also propose three different algorithms to compute *similarities* between users in multi-criteria rating settings: similarity-per-priority, similarity-per-evaluation, and similarity-per-partial-utility. The similarity-per-priority algorithm computes the similarities between users based on importance weights $w_c(u)$ of user u for each criterion c (rather than ratings $R(u,i)$). In this way, it creates a neighborhood of users that have the same importance weights on multiple criteria with the target user. Then, it tries to predict the overall utility of an item for this user, based on the total utilities of the users in the neighborhood. In addition, the similarity-per-evaluation and similarity-per-partial-utility algorithms create separate neighborhoods for the target user for each criterion, i.e., they calculate the similarity with other users per individual criterion, and then predict the rating that the target user would provide upon each individual criterion. The similarity-per-evaluation algorithm calculates the similarity based on the non-weighted ratings that the users provide on each criterion. The similarity-per-partial-utility algorithm calculates the similarity based on the weighted (using $w_c(u)$ of each user u) ratings that the users provide on each criterion.

In such systems, the similarities between users are obtained using multi-criteria ratings, and the rest of the recommendation process can be the same as in single-criterion rating systems. The next step is, for a given user, to find a set of neighbors with the highest similarity values and predict unknown overall ratings of the user based on neighbors' ratings. Therefore, these similarity-based approaches are applicable only to neighborhood-based collaborative filtering recommendation techniques that need to compute the similarity between users (or items).

In summary, multi-criteria ratings can be used to compute the similarity between two users in the following two ways [2]: by (1) aggregating similarity values that are calculated separately on each criterion into a single similarity and (2) calculating the distance between multi-criteria ratings directly in the multi-dimensional space. Empirical results using a small-scale Yahoo! Movies dataset show that both heuristic approaches outperform the corresponding traditional single-rating collaborative filtering technique (i.e., that uses only single overall ratings) by up to 3.8 % in terms of precision-in-top-N metric, which represents the percentage of truly high overall ratings among those that the system predicted to be the N most relevant items for each user [2]. The improvements in precision depend on many parameters of collaborative filtering techniques, such as neighborhood sizes and the number of top-N recommendations. Furthermore, these approaches can be extended as suggested by Manouselis and Costopoulou [52] by computing similarities using not only known rating information, but also importance weights for each criterion. The latter approaches were evaluated in an online application that recommends e-markets to users, where multiple buyers and sellers can access and exchange information about prices and product offerings, based on users' multi-criteria evaluations on several e-markets. The similarity-per-priority algorithm using Euclidian distance performed the best among their proposed approaches in terms of the mean absolute error (MAE) (i.e., 0.235 on scale of 1–7) with a fairly high coverage (i.e., 93 % of items can be recommended to users) as compared to non-personalized algorithms, such as arithmetic mean and random, that produce higher MAE (0.718 and 2.063, respectively) with 100 % coverage [52].

Maneeroj et al. [50] further investigate the problem of finding the most appropriate neighbors in multi-criteria recommendation settings. In particular, based on the observation that different criteria may have varying importance for different users, they propose an approach that incorporates the individualized importance levels of each criteria into the user-user similarity calculation process. This approach may provide more appropriately chosen neighbors and, consequently, result in better recommendation results.

As mentioned earlier, once the similarity between users or items is computed, the standard neighborhood-based collaborative filtering recommendation technique generally estimates 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 as follows [12]:

- *Weighted sum approach*:

$$R(u,i) = \frac{\sum_{u' \in N(u,i)} sim(u,u')R(u',i)}{\sum_{u' \in N(u,i)} |sim(u,u')|} \tag{25.14}$$

- *Adjusted weighted sum approach*:

$$R(u,i) = \overline{R(u)} + \frac{\sum_{u' \in N(u,i)} sim(u,u')(R(u',i) - \overline{R(u')})}{\sum_{u' \in N(u,i)} |sim(u,u')|} \tag{25.15}$$

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 $R(u, i)$. $N(u, i)$ represents the set of users that are similar to user u among the ones who consumed item i, and the size of set $N(u, i)$ 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 $R(u, i)$.

In the similarity-based approach [2], the above two formulas are typically used to predict the *overall* ratings only, because the recommendations are usually based on the system's predictions of the overall user preferences for items. In other words, $R(u, i)$ here refers to r_0 and not to the entire multi-dimensional rating vector $R(u, i) = (r_0, r_1, \ldots, r_k)$. However, the same formulas can be used to predict each individual criteria rating r_i, if desired. Also, the heuristic similarity-based approach explained above represent the *user-based* approach that uses neighboring users to compute recommendations. As the user-based approach using single criterion ratings can be straightforwardly transformed to the item-based approach that uses neighboring items to compute recommendations [85], the formulas for the user-based approach in the multi-criteria rating settings can be straightforwardly rewritten for the item-based approach.

Furthermore, having to submit precise numeric ratings for multiple criteria of each individual item may represent an increased burden for users. Therefore, it may be advantageous to consider the subjective, imprecise, and vague nature of human ratings when collecting such information. Several studies propose to use fuzzy linguistic approaches for representing and collecting user ratings and to employ fuzzy multi-criteria decision making techniques to rank the relevant items for each user [10, 66]. More specifically, each user's relevance feedback can be collected in the qualitative form (in linguistic terms). For example, in the work of Boulkrinat et al. [10], each user evaluated six criteria of a hotel (i.e., Clean, Comfort, Location, Facilities, Staff, and Value-for-Money), and the user's preferences are expressed through linguistic terms on a scale of 7 levels (i.e., Very High, High, Medium High, Medium, Medium Low, Low and Very Low). The preference for each criterion is then modeled not by a single numeric value (i.e., single level) but rather by a "fuzzy number" (essentially, by a range of levels). The weight of each criterion can be provided by an individual user, representing his or her personal relative importance among the six criteria.

Other fuzzy-based algorithms for multi-criteria CF systems are introduced in the work of Nilashi et al. [61], including Weighted Fuzzy MC-CF and Fuzzy Euclidean MC-CF that use fuzzy-based average similarity and fuzzy-based Euclidean distance respectively, and Fuzzy Average MC-CF that uses a fuzzy-based user- and item-based predictions in a weighted approach. Palanivel and Sivakumar [68] also propose a fuzzy aggregation-based approach that finds preference criterion using a maximum operator, particularly from implicit interest indicators such as time spent on hearing a music item, number of accesses to a music item, and music download status. The use of such implicit interest indicators can further mitigate the burden for the users to keep providing multiple ratings for each consumed item.

25.3.2 Model-Based Approaches

Model-based approaches construct a predictive model to estimate unknown ratings by learning from the observed data. Several existing approaches for multi-criteria rating recommenders fall into this category, including simple aggregation functions, probabilistic modeling, multilinear singular value decomposition (MSVD), and support vector regression (SVR).

Aggregation Function Approach While overall rating r_0 is often considered simply as just another criterion rating in similarity-based heuristic approaches (as illustrated earlier), the aggregation function approach assumes that the overall rating serves as an aggregate of multi-criteria ratings [2]. Given this assumption, this approach finds aggregation function f that represents the relationship between overall and multi-criteria ratings, i.e.,

$$r_0 = f(r_1, \ldots, r_k) \tag{25.16}$$

For example, 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 high, the overall rating of the movie must also be predicted high in order to be accurate.

The aggregation function approach consists of three steps, as summarized in Fig. 25.3. First, this approach estimates k individual ratings using any recommendation technique. That is, the k-dimensional multi-criteria rating problem is decomposed into k single-rating recommendation problems. Second, aggregation function f is chosen using domain expertise, statistical techniques, or machine learning techniques. For example, the domain expert may suggest a simple average function of the underlying multi-criteria ratings for each item based on her prior experience and knowledge. An aggregation function also can be obtained by using statistical techniques, such as linear and non-linear regression analysis techniques, as well as various sophisticated machine learning techniques, such as artificial neural networks. Finally, the overall rating of each unrated item is computed based on the k predicted individual criteria ratings and the chosen aggregation function f.

While the similarity-based heuristic approaches described earlier apply to only neighborhood-based collaborative filtering recommendation techniques, the aggregation function approach can be used in combination with any traditional recommendation technique, because individual criteria ratings are used for the prediction in the first step. As one example of possible aggregation functions, Adomavicius and Kwon [2] use linear regression and estimate coefficients (i.e., importance weights of each individual criterion) based on the known ratings.

Adomavicius and Kwon [2] also note that the aggregation function can have different scopes: total (i.e., when a single aggregation function is learned based on the entire dataset), user-based or item-based (i.e., when a separate aggregation function is learned for each user or item).

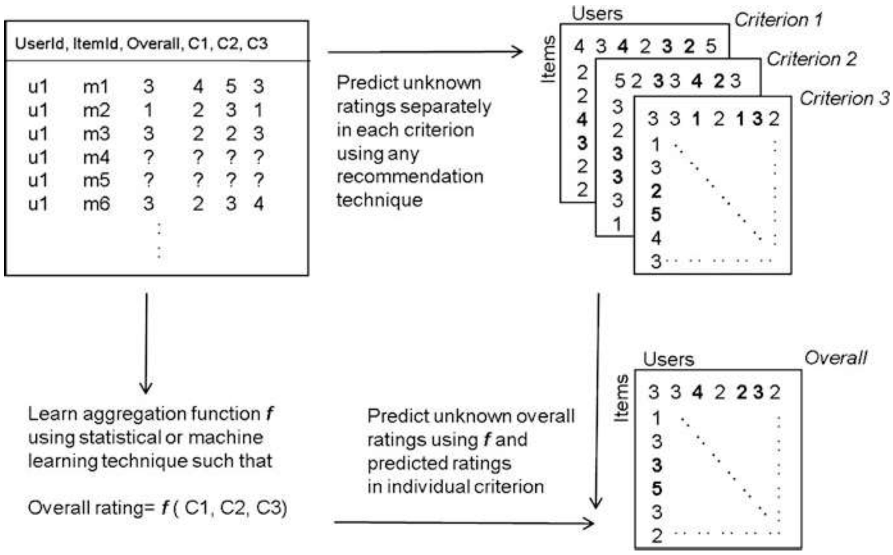

Fig. 25.3 Aggregation function approach (an example of a three-criteria rating system)

Empirical analysis using data from Yahoo! Movies shows that the aggregation function approach (using multi-criteria rating information) outperforms a traditional single-rating collaborative filtering technique (using only overall ratings) by 0.3–6.3 % in terms of precision-in-top-N ($N = 3, 5$, and 7) metric [2].

Probabilistic Modeling Approach Some multi-criteria recommendation approaches adopt probabilistic modeling algorithms that are becoming increasingly popular in data mining and machine learning. One example is the work of Sahoo et al. [80], which extends the flexible mixture model (FMM) developed by Si and Jin [89] to multi-criteria rating recommenders. The FMM assumes that there are two latent variables Z_u and Z_i (for users and items), and they are used to determine a single rating r of user u on item i, as shown in Fig. 25.4a. Sahoo et al. [80] also discover the dependency structure among the overall ratings (r_0) and multi-criteria ratings (r_1, r_2, r_3, and r_4), using Chow-Liu tree structure discovery [19], and incorporate the structure into the FMM, as shown in Fig. 25.4b.

The FMM approach is based on the assumption that the joint distribution of three variables (user u, rating r, and item i) can be expressed using the sum of probabilities over the all possible combinations of the two latent class variables Z_u and Z_i, as follows.

$$P(u, i, r) = \sum_{Z_u, Z_i} P(Z_u)P(Z_i)P(u|Z_u)P(i|Z_i)P(r|Z_u, Z_i) \tag{25.17}$$

In summary, an overall rating of an unknown item for a target user is estimated with the following two steps: learning and prediction. In the first (learning) step, all

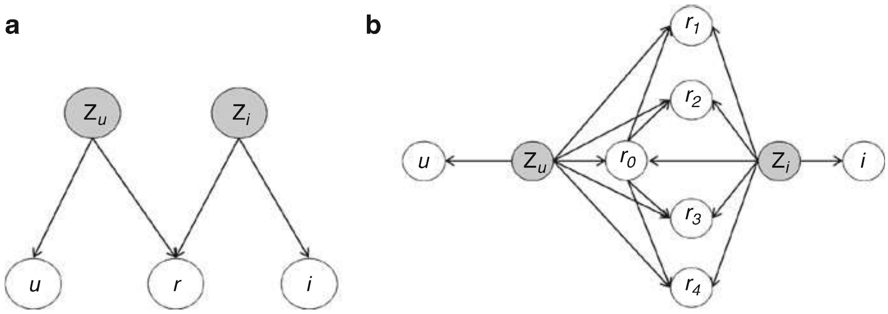

Fig. 25.4 Examples of probabilistic modeling approach in recommender systems. (**a**) Flexible Mixture Model for a single-rating recommender system [89]. (**b**) FMM with multi-criteria rating dependency structure [80]

the parameters of the FMM are estimated using the expectation maximization (EM) algorithm [21]. Using the obtained parameters, in the second (prediction) step, the overall rating of a given unknown item is predicted as the most likely value (i.e., the rating value with the highest probability). This approach has been extended to multi-criteria ratings, and the detailed algorithm can be found in [80].

Sahoo et al. [80] also compare their model in Fig. 25.4b with the model that assumes independence among multi-criteria ratings conditional on the latent variables, and found that the model with dependency structure performs better than the one with the independence assumption. This finding demonstrates the existence of the "halo effect" in multi-criteria rating systems. The "halo effect" is a phenomenon often studied in psychometric literature, which indicates a cognitive bias whereby the perception of a particular object in one category influences the perception in other categories [94]. In multi-criteria recommender systems, the individual criterion ratings provided by users are correlated due to the "halo effect", and particularly more correlated to an overall rating than to other individual ratings [80]. In other words, the overall rating given by the user to a specific item seems to affect how the user rates the other (individual) criteria of this item. Thus, controlling for an overall rating reduces this halo effect and helps to make individual ratings independent of each other, as represented in the Chow-Liu tree dependency structure (Fig. 25.4b).

Using data from Yahoo! Movies, Sahoo et al. [80] show that multi-criteria rating information is advantageous over a single rating when very little training data is available (i.e., less than 15 % of the whole data is used for training). On the other hand, when large training data is available, additional rating information does not seem to add much value. In this analysis, they measure the recommendation accuracy using the MAE metric. However, when they validate this probabilistic modeling approach using precision and recall metrics in retrieving top N items, their model performs better in all cases (i.e., both with small and large datasets) with a maximum of 10 % increase. With more training data, the difference between the model with multi-criteria ratings and the traditional single-rating model diminishes in terms of precision and recall metrics.

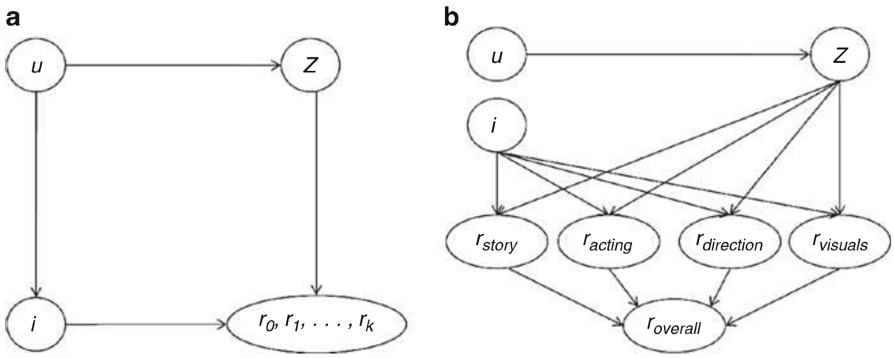

Fig. 25.5 Graphical model representation of multi-criteria PLSA algorithms. (**a**) Full Gaussian PLSA [100]. (**b**) Linear Gaussian Regression PLSA using Yahoo!Movies dataset [100]

Another probabilistic modeling approach was proposed by Zhang et al. [100], who extend the probabilistic latent semantic analysis (PLSA) approach used for single-criteria recommender systems [32] into multi-criteria rating settings. In particular, [100] investigate two multi-criteria PLSA algorithms, based on their modeling of the underlying multi-criteria rating distribution of each user: (1) using full multi-variate Gaussian distribution, and (2) using linear Gaussian regression model. Both proposed approaches provide accuracy improvements over several single-criteria and multi-criteria recommender systems baselines. Graphical model presentations of the two approaches are shown in Fig. 25.5a and b, where r represents a rating of item i by user u and Z is a latent variable. The full gaussian model uses multi-variable nodes $r_0, r_1, \ldots, r_k$ instead of uni-variate node r, and applies the same EM algorithm as used in the single-rating PLSA. Linear Gaussian regression model computes the overall preference (r_0) as the linear combination of preferences on individual criteria ($r_1, \ldots, r_k$). Furthermore, while the work of Sahoo et al. [80] does not employ any normalization scheme for ratings of each user (e.g., adjusting the neutral vote of the individuals to zero and standardizing the scale of all users to the same value), [100] shows that user normalization significantly affects the performance of the multi-criteria PLSA approaches.

Multilinear Singular Value Decomposition (MSVD) Approach Li et al. [46] propose an approach to improve a traditional collaborative filtering algorithm by utilizing the MSVD technique which is a particular realization of the Matrix Factorization approach in multi-criteria rating settings. Singular value decomposition (SVD) techniques have been extensively studied in numerical linear algebra and have also gained popularity in recommender systems applications because of their effectiveness in improving recommendation accuracy [28, 40, 84]. In single-rating recommender systems, these techniques are used to find a lower-dimensional feature space. For example, using K latent features (i.e., rank-K SVD), user u is associated with user-factor vector p_u (the user's preferences on K features), and item i is associated with item-factor vector q_i (the item's importance weights on K features).

After all the values in user- and item-factors vectors are estimated, the preference of how much user u likes item i, denoted by $R^*(u, i)$, is predicted by taking an inner product of the two vectors, i.e.,

$$R^*(u, i) = p_u^T q_i \tag{25.18}$$

More details on the basic SVD techniques can be found in Chap. 7. While the SVD techniques are commonly used as a decomposition method for two-dimensional data (i.e., single criterion ratings), they can be extended for multi-dimensional data (i.e., multi-criteria ratings), referred to as MSVD techniques [20].

For example, Li et al. [46] incorporate contextual information and multi-criteria ratings into recommendation processes. Based on the contextual information, the recommendation problem is defined as a 3-order tensor representing the rating of an item by a user on a criterion under a specific context, and the tensor approximation based on the truncated MSVD technique is then performed. The approximated tensor is finally used to improve neighborhood formation for later use in a neighborhood-based collaborative filtering approach, i.e., identifying the nearest neighbors of each user and computing top-N recommendations.

More specifically, Li et al. [46] use the MSVD to reduce the dimensionality of multi-criteria rating data and evaluate their approach in the context of a restaurant recommender system, where a user rates a restaurant on 10 criteria (i.e., cuisine, ambience, service, etc.). The results demonstrate that their approach improves the accuracy of recommendations (as measured by precision-in-top-N) by up to 5 %, as compared to the traditional single-rating model.

Support Vector Regression (SVR) Approach Several other studies also follow the general aggregation function approach; however, instead of using the traditional linear least squares regression method, they propose to use the Support Vector Regression (SVR) [23] to learn the regression-based rating aggregation functions [25, 34, 35, 81]. While all features can be considered in the regression, [34, 35] propose several ways to choose the most relevant features—by using the chi-squared statistics with respect to the overall ratings for each criterion, applying a genetic feature selection algorithm, or obtaining the advice from a domain expert—and highlight the importance of choosing an adequate subset of item dimensions since it affects the performance of recommendations.

In addition to the higher reported predictive accuracy, another advantage of the SVR technique is that it can be employed in settings with relatively few data points but many features (e.g., many rating dimensions). In particular, Jannach et al. [35] use both user- and item-based SVR approach, i.e., they estimate regression models R^*_{user} individually for each user and regression models R^*_{item} individually for each item. Then, the two predictions can be combined using item and user weights. As described in Fig. 25.6, user- and item-based SV-regression is learned from training data, and criteria ratings can be predicted using any CF technique. Then, overall ratings are estimated using the criteria predictions and SV-regression functions. The final prediction is computed as a weighted combination of the two

Step 1. For each user u and each item i, learn user and item SV-regression-based aggregation functions $R^*_{user}(u,i)$ and $R^*_{item}(u,i)$ from training data
Step 2. Predict individual ratings on multiple criteria for item i and user u (using some standard CF technique)
Step 3. Predict overall ratings using user and item SV-regression-based aggregation functions $R^*_{user}(u,i)$ and $R^*_{item}(u,i)$
Step 4. Use standard gradient descent to compute user and item weights (w_u, w_i)

```
require: #iterations, γ, λ
// Gradient descent iterations:
for 1 to #iterations do
  for each user u do
    for each rated item i of user u do
      // compute prediction with current weights
      R*(u,i) → w_u × R*_user(u,i) + w_i × R*_item(u,i)
      // compare with real rating R(u,i) and determine the error e(u,i)
      e(u,i) ← R(u,i) − R*(u,i)
      // Adjust w_u in gradient step
      w_u ← w_u + γ · (e(u,i) − λ · w_u)
      // Adjust w_i in gradient step
      w_i ← w_i + γ · (e(u,i) − λ · w_i)
return w_u for each user u, and w_i for each item i
```

Step 5. Combine the two predictions using user and item weights
$R^*(u,i) = w_u \times R^*_{user}(u,i) + w_i \times R^*_{item}(u,i)$

Fig. 25.6 Gradient descent algorithm for the weighted support vector regression (SVR) method [35]

overall predictions obtained from user and item-based SV-regression function. Step 4 of Fig. 25.6 describes how weights w_u and w_i are estimated (optimized) in a personalized manner for each user u and item i. A fast, heuristic gradient descent procedure is used to estimate parameters for each user and item by minimizing the prediction error calculated as the difference between the predicted and the actual rating. Here parameter γ determines the size of the correcting step, and λ is used as a regularization to avoid over-fitting.

The results show that the proposed approach using support vector regression with individual and optimized weights for each single user and item compares favorably against a number of existing approaches with respect to multiple evaluation metrics (RMSE, F-measure, precision-in-top-N) on hotel and movie rating datasets. In addition, [35] also evaluated several feature selection strategies that can be useful for multi-criteria recommendation settings with many rating dimensions and showed that using relative simple feature selection procedures (such as chi-square statistics) can lead to further improvements in recommendation accuracy.

In summary, the above approaches represent some of the initial attempts to apply sophisticated learning techniques to address multi-criteria recommendation problems, and we expect to see more such techniques in the future. In the next section, we discuss different approaches to recommending items to users, assuming that the unknown multi-criteria ratings have been estimated using any of the techniques discussed above.

25.4 Engaging Multi-Criteria Ratings During Recommendation

As mentioned above, multi-criteria recommender systems may choose to model a user's utility for a given item by including both the overall rating and ratings of individual item components/criteria or they may choose to include only ratings of individual criteria. If overall ratings are included as part of the model, the recommendation process in such cases is typically straightforward: after predicting all unknown ratings, the recommender system uses the overall rating of items to select the most highly predicted items (i.e., the most relevant items) for each user. In other words, the recommendation process is essentially the same as in traditional, single-criterion recommender systems.

However, without an overall rating the recommendation process becomes more complex, because it is less apparent how to establish the total order of the items. For example, suppose that we have a two-criterion movie recommender system, where users judge movies based on their story (i.e., plot) and visual effects. Further, suppose that one movie needs to be chosen for recommendation among the following two alternatives: (1) movie X, predicted as 8 in story and 2 in visuals, and (2) movie Y, predicted as 5 in story and 5 in visuals. Since there is no overall criterion to rank the movies, it is not easy to judge which movie is better, unless some other modeling approach is adopted, using some non-numerical (e.g., rule-based) way for expressing preferences. Several approaches have been proposed in the recommender systems literature to deal with this problem: some try to design a total order on items and obtain a single global optimal solution for each user, whereas others take one of the possible partial orders of the items and find multiple (Pareto optimal) solutions. Below we briefly mention related work on multi-criteria optimization, describe several approaches that have been used in the recommender systems literature, and discuss other potential uses of multi-criteria ratings in the recommendation process.

25.4.1 Related Work: Multi-Criteria Optimization

Multi-criteria optimization problems have been extensively studied in the operations research (OR) literature [24], although not in the context of recommender systems. This multi-criteria optimization approach assists a decision maker in choosing the best alternative when multiple criteria conflict and compete with each other. For example, various points of view, such as financial, human resources-related, and environmental aspects should be considered in organizational decision making. The following approaches are often used to address multi-criteria optimization problems, and can be applied to recommender systems, as discussed in [4]:

- Finding Pareto optimal solutions;
- Taking a linear combination of multiple criteria and reducing the problem to the single-criterion optimization problem;
- Optimizing only 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.

In multi-criteria rating recommenders, an item can be evaluated differently on a different criterion; thus, it is not an easy task to find the best item overall. Below we describe several recommendation approaches that have been used in the recommender systems literature, all of them having roots in multi-criteria optimization techniques, including: converting the multi-criteria optimization problem into single-criterion ranking problem (Sect. 25.4.2), finding Pareto optimal recommendations (Sect. 25.4.3), and using multiple criteria as constraints (Sect. 25.4.4).

25.4.2 Designing a Total Order for Item Recommendations

In the recommender systems literature there has been some work using multi-attribute utility theories from decision sciences, which can be described as one way to take a linear combination of multiple criteria and find an optimal solution [43], essentially reducing the multi-criteria optimization problem to a simple, single-criteria ranking problem. For example, the approach by Lakiotaki et al. [43] ranks the items by adopting the UTilités Additive (UTA) method proposed by Siskos et al. [90]. Their algorithm aims to estimate overall utility U of a specific item for each user by adding the marginal utilities of each criterion $c (c = 1, \ldots, k)$.

$$U = \sum\nolimits_{c=1}^{k} u_c(R_c) \tag{25.19}$$

which is subject to the following constraints: $u_c(R_c^{worst}) = 0, \forall c = 1, 2, \ldots, k$ and $\sum_{c=1}^{k} u_c(R_c^{best}) = u_1(R_1^{best}) + u_2(R_2^{best}) + \cdots + u_k(R_k^{best}) = 1$. Here R_c is the rating provided on criterion c, and $u_c(R_c)$ is a non-decreasing real-value function (marginal utility function) for a specific user. Assuming that $[R^{worst}, R^{best}]$ is the criterion evaluation scale, R_i^{worst} and R_i^{best} are the worst and the best level of the i-th criterion respectively. The decision maker is asked to provide her global evaluation so as to form a total pre-order of the alternatives (items): $i_1 \succ i_2 \succ \cdots \succ i_m$. The developed utility model is assumed to be consistent with the decision maker's judgment policy so that $U(i_1) > U(i_2) \cdots > U(i_m)$. In developing the global utility model to meet this requirement, there are two types of possible errors which may occur: (1) the under-estimation error when the developed model assigns an alternative to a lower (better) rank than the one specified in the given pre-order (the alternative is under-estimated by the decision maker), and (2) the over-estimation error when the developed model assigns an alternative to a higher (worse) rank than the one specified in the given pre-order (the alternative is over-estimated by the decision maker). The final model is chosen by minimizing the sum of these two errors. Given the estimated ratings on multiple criteria, this can be performed using linear programming techniques.

Since this approach uses the ranking information with ordinal regression techniques, Kendall's tau is used as a measure of correlation between two ordinal-level variables to compare an actual order and the predicted order. The empirical results obtained by using data from Yahoo! Movies show that 20.4 % of users obtain a Kendall's tau of 1 indicating a total agreement of the orders between the ones predicted by the recommender system and the ones stated by users, and the mean value of Kendall's tau across all users is 0.74. This approach is also evaluated using the Receiver Operating Curve (ROC), which depicts relative trade-offs between true positives and false positives. The obtained Area Under Curve (AUC) of 0.81, where 1 represents a perfect classifier and 0.5 represents the performance of a random classifier, demonstrates that multi-criteria ratings provide measurable improvements in modeling users' preferences.

Similarly, Manouselis and Costopoulou [52] propose a method that calculates total utility U either by summing the k predicted partial utilities u_c (in their similarity-per-partial-utility algorithm) or by weighting the predicted ratings that the user would give on each criterion c by the user's importance weights w_c (in their similarity-per-evaluation algorithm). In both cases, the total utility of a candidate item is calculated using an aggregate function of the following form:

$$U = \sum\nolimits_{c=1}^{k} u_c = \sum\nolimits_{c=1}^{k} w_c R_c \tag{25.20}$$

Here individual ratings on multiple criteria are used to rank the candidate items, rather than explicitly estimate overall ratings. Finally, once the total order on the candidate items is established using any of the above techniques, each user gets recommended the items that maximize this total utility.

Akhtarzada et al. [5] also use each user's ratings on items under multiple criteria to rank the items as a recommendation list. To do so, users are first assigned ideal values on each criterion as an average of their past ratings, and the rating on a new item for a specific user is predicted by calculating the distance between the ideal values for all users and the ideal values for the user. Then, when a user sees an item, the most similar item can be recommended based on the similarities between items for the user.

25.4.3 Finding Pareto Optimal Item Recommendations

This approach discovers several good items among large number of candidates (rather than arriving at a unique solution by solving a global optimization problem) when different items can be associated with multiple conflicting criteria and the total order on items is not directly available. *Data envelopment analysis* (DEA), often also called "frontier analysis", is commonly used to measure productive efficiency of decision making units (DMU) in operations research [18]. DEA computes the efficiency frontier, which identifies the items that are "best performers" overall, taking into account all criteria. DEA does not require *a priori* weights for each criterion, and uses linear programming to arrive more directly at the best set of weights for each DMU. Specifically, in the context of multi-criteria recommender systems, given all the candidate items that are available for recommendation to a given user (including the information about their predicted ratings across all criteria), DEA would be able to determine the reduced set of items (i.e., the frontier) that have best ratings across all criteria among the candidates. These items then can be recommended to the user.

While DEA has not been directly used in multi-criteria rating recommenders, the multi-criteria recommendation problem without overall ratings can also be formulated as a data query problem in the database field, using similar motivation [44]. Lee and Teng [44] utilize *skyline* queries to find the best restaurants across multiple criteria (i.e., food, décor, service, and cost). As Fig. 25.7 shows, skyline queries identify a few skyline points (i.e., Pareto optimal points) that are not dominated by any others from a large number of candidate restaurants in two-dimensional data space (food and décor). Here, for a given user, a candidate item is considered to be dominated, if there exists another candidate item that has better or equal ratings on all criteria.

Empirical results using multi-criteria ratings of Zagat Survey in [44] show that the recommender system using skyline queries helps to reduce the number

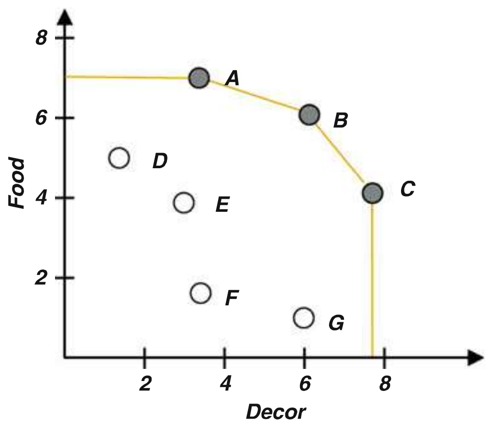

Fig. 25.7 An example of skyline points (the best candidate restaurants) in two-dimensional space

of choices that users should consider from their inquiries. For example, when a user searches for buffet restaurants which are located in New York City with a cost of no more than \$30, the system recommends only two restaurants among twelve candidate restaurants, based on the ratings on four criteria. However, this preliminary work needs to be extended in several directions because the skyline queries may not scale well with the increasing number of criteria, resulting in a large number of skyline points with high computational cost.

25.4.4 Using Multi-Criteria Ratings as Recommendation Filters

Similar to how content attributes can be used as recommendation filters in recommender systems [45, 86], multi-criteria ratings can be used for similar purposes as well. For example, a user may want to specify that only the movies with an exceptionally good story should be recommended to her at a given time, regardless of other criteria, such as visual effects. Then, only the movies that are highly predicted in the story criterion (say, ≥ 9 out of 10) will be recommended to the user. In other words, the dimensionality of multi-criteria optimization problem can be reduced by converting some of the criteria to constraints (filters). This approach is also similar to how content-based [45, 86] or context-aware [3] recommendation approaches filter recommendations; however, it is also slightly different from them, because the filtering is done not based on objective content attributes (e.g., MovieLength $<$ 120 min) or additional contextual dimensions (e.g., TimeOf-Week = weekend), but on the subjective rating criteria (e.g., Story $\geq$ 9), the predicted value of which is highly dependent on user's tastes and preferences.

25.5 Discussion and Future Work

Recommender systems represent a vibrant and constantly changing research area. Among the important recent developments, recommender systems have recently started adopting multi-criteria ratings provided by users, and in this chapter we explored algorithms and techniques for multi-criteria recommender systems. These relatively new systems have not yet been studied extensively, and in this section we present a number of challenges and future research directions for this category of recommender systems.

25.5.1 Developing New Approaches for Multi-Criteria Ratings

Modeling Multi-Criteria Ratings Traditionally, user preferences in recommender systems (including multi-criteria recommender systems) are expressed using simple numeric ratings. Recent work has started to explore alternative approaches for representing and collecting user ratings (e.g., using fuzzy techniques [10, 66]) as well as for modeling ratings in a more nuanced manner (e.g., taking into account semantic interval-scale characteristics of numeric ratings [57]). A comprehensive exploration of user preference modeling, especially in more complex multi-criteria settings, represents an interesting direction for future work.

Intelligent Data Pre-Processing and Segmentation It is well-known that many recommendation settings suffer from the data "sparsity" issue. One possible approach to alleviate this problem is to perform intelligent data segmentation or clustering, where the non-useful dimensions (criteria) are discarded or where data from similar users (or similar items) is merged and the resulting recommendations are calculated (and potentially improved) by taking this aggregation into account. In data mining literature, there has been some work on what the optimal customer segmentation should be [36]. Also, in multi-criteria recommender systems, several approaches have already used a wide variety of specialized user clustering procedures as part of the proposed recommendation algorithms (e.g., [42, 47, 49, 61, 62]). Some researchers have also explored different feature selection techniques for determining the best criteria to use in multi-criteria settings [35]. However, further studies are needed to examine various data pre-processing and segmentation approaches for multi-criteria recommender systems in a more systematic manner.

Predicting Relative Preferences An alternative way to define the multi-criteria recommendation problem could be formulated as predicting the *relative* preferences of users, as opposed to the *absolute* rating values. There has been some work on constructing the correct relative order of items using ordering-based techniques. For example, Freund et al. [26] developed the RankBoost algorithm based on the well-known AdaBoost method and, in multi-criteria settings, such algorithms could be adopted to aggregate different relative orders obtained from different rating criteria for a particular user. In particular, this is an approach taken by the DIVA system [59, 60].

Constructing the Item Evaluation Criteria More research needs to be done on choosing or constructing the best set of criteria for evaluating an item. For example, most of current multi-criteria rating recommenders require users to rate an item on multiple criteria at a single level (e.g., story and special effects of a movie). This single level of criteria could be further broken down into sub-criteria, and there could be multiple levels depending on the given problem. For example, in a movie recommender system, special effects could be again divided into sound and graphic effects. More information with multiple levels of criteria could potentially help to better understand user preferences, and various techniques, such as the analytic hierarchy process (AHP), can be used to consider the hierarchy of criteria [79], as Schmitt et al. [87] propose to do in their system. As we consider more criteria for each item, we may also need to carefully examine the correlation among criteria because the choice of criteria may significantly affect the recommendation quality. Furthermore, it is important to have a *consistent* family of criteria for a given recommender system application, which means that the criteria are monotonic, exhaustive, and non-redundant. In summary, constructing a set of criteria for a given recommendation problem is an interesting and important topic for future research.

Incorporating Domain-Specific Information Many multi-criteria recommender systems are designed without exploiting specific domain knowledge. For example, understanding not just the multiple hotel characteristics (such as cleanliness, location, service, etc.), but also the different segments of population that like to travel (e.g., business travellers, senior travellers, honeymoon/romantic travellers, spring-break travellers, etc.) can provide substantial advantages in designing better recommendation algorithms. Several studies have started exploring the models that can incorporate domain-specific information into multi-criteria recommender systems [11, 27], but there are a lot of further opportunities in this research direction. Similarly, many application domains have rich content information available, and taking advantage of this information (e.g., leveraging tag information for movie recommendation [29, 30] or leveraging job-seeking intent for talent recommendation [78]), can provide further improvements in multi-criteria recommender systems.

25.5.2 Extending Existing Techniques for Multi-Criteria Settings

Reusing Existing Single-Rating Recommendation Techniques A huge number of recommendation techniques have been developed for single-rating recommender systems over the last 15–20 years, and some of them have been extended to multi-criteria rating systems, as discussed in this chapter. For example, neighborhood-based collaborative filtering techniques can take into account multi-criteria ratings using the huge number of design options that Manouselis and Costopoulou [53] suggest (and as discussed in Sect. 25.3.1). There have also been multi-criteria SVD-based and PLSA-based recommendation approaches proposed (as discussed in Sect. 25.3.2), which stem from their single-criterion counterparts. However, among alternative approaches, there has been a number of sophisticated hybrid recommendation approaches developed in recent years [16], and some of them could potentially be adopted for multi-criteria rating recommenders. Finally, more sophisticated techniques, e.g., based on data envelopment analysis (DEA) or multi-criteria optimization, could be adopted and extended for choosing best items in the multi-criteria rating settings.

Investigating Group Recommendation Techniques for Multi-Criteria Settings Some techniques for generating recommendations to groups, as described in Chap. 22, can be adopted in multi-criteria rating settings. According to [33], a group preference model can be built by aggregating the diverse preferences of several users. Similarly, a user's preference for an item in multi-criteria rating settings can be predicted by aggregating the preferences based on different rating criteria. More specifically, there can be many different goals for aggregating individual preferences [55, 64], such as maximizing average user satisfaction, minimizing misery (i.e., high user dissatisfaction), and providing a certain level of fairness (e.g., low variance with the same average user satisfaction). Multi-criteria rating recommenders could investigate the adoption of some of these approaches for aggregating preferences from multiple criteria.

25.5.3 Managing Multi-Criteria Ratings

Managing Intrusiveness The extra information provided by multi-criteria ratings can give rise to an important issue of "intrusiveness", i.e., the requirement for the users to provide this extra information to the system. Specifically, for a recommender system to achieve good recommendation performance, users typically need to provide to the system a certain amount of feedback about their preferences (e.g., in the form of item ratings). This can be an issue even in single-rating recommender systems [39, 56, 63], and some less intrusive techniques to obtain user preferences in multi-criteria recommender systems have been explored [54, 65, 67, 71]. Multi-criteria rating systems are likely to require a more significant level of user involvement because each user would need to rate an item on multiple criteria. Therefore, it is important to measure the costs and benefits of adopting multi-criteria ratings and find an optimal solution to meet the needs of both users and system designers. Preference disaggregation methods could support the implicit formulation of a preference model based on a series of previous decisions. A characteristic example is the UTA (i.e., UTilités Additive) method, which can be used to extract the utility function from a user-provided ranking of known items [43]. Another example is the ability to obtain each user's preferences on several attributes of an item implicitly from the user's written comments, minimizing intrusiveness [1, 54, 70]. There are also some empirical approaches with less computational complexity [82]. Lastly, performing user studies on multi-criteria recommender systems would further examine the impact of having to submit more ratings on the overall user satisfaction.

Dealing with Missing Multi-Criteria Ratings Multi-criteria recommender systems typically would require the users to provide more data to such systems than their single-rating counterparts, thus increasing the likelihood of obtaining missing or incomplete data. One popular technique to deal with missing data is the expectation maximization (EM) algorithm [21] that finds maximum likelihood estimates for incomplete data. In particular, the probabilistic modeling approach for multi-criteria rating prediction proposed by Sahoo et al. [80] uses the EM algorithm to predict values of the missing ratings in multi-criteria rating settings. Similarly, Bayesian models are proposed to handle incomplete missing rating data, for example, missing ratings on one criterion with the ratings on other criteria [91]. The applicability of other existing techniques in this setting should be explored, and novel techniques could be developed by considering the specifics of multi-criteria information, such as the possible relationships between different criteria.

Collecting Large-Scale Multi-Criteria Rating Data Multi-criteria rating datasets that can be used for algorithm testing and parameterization are rare. For this new area of recommender systems to be successful, it is crucial to have a number of standardized real-world multi-criteria rating datasets available to the research community. Some initial steps towards a more standardized representation, reusability, and interoperability of multi-criteria rating datasets have been taken in other application domains, such as e-learning [98].

In this section we discussed several potential future research directions for multi-criteria recommenders that should be interesting to recommender systems community. This list is not meant to be exhaustive; we believe that research in this area is only in its preliminary stages, and there are a number of possible additional topics that could be explored to advance multi-criteria recommender systems.

25.6 Conclusions

In this chapter, we aimed to provide an overview of multi-criteria recommender systems. More specifically, we focused on the category of *multi-criteria rating recommenders*, i.e., techniques that provide recommendations by modelling a user's utility for an item as a vector of ratings along several criteria. We reviewed current techniques that use multi-criteria ratings for calculating the rating predictions and generating recommendations, and discussed open issues and future challenges for this class of recommender systems.

This survey provides a systematic view of multi-criteria recommender systems, a roadmap of relevant work, and a discussion of a number of promising future research directions. However, we believe that this sub-area of recommender systems is still in its early stages of development, and much more research is needed to unlock the full potential of multi-criteria recommenders.

References

1. Aciar, S., Zhang, D., Simoff, S., Debenham, J.: Informed recommender: Basing recommendations on consumer product reviews. IEEE Intelligent systems **22**(3), 39–47 (2007)
2. Adomavicius, G., Kwon, Y.: New recommendation techniques for multicriteria rating systems. IEEE Intelligent Systems **22**(3), 48–55 (2007)

3. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Transactions on Information Systems (TOIS) **23**(1), 103–145 (2005)
4. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering **17**(6), 734–749 (2005)
5. Akhtarzada, A., Calude, C., Hosking, J.: A multi-criteria metric algorithm for recommender systems. Fundamenta Informaticae **110**(1), 1–11 (2011)
6. Amatriain, X., Basilico, J.: Netflix recommendations: Beyond the 5 stars. http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html (2012). Accessed: 2014-06-28
7. Balabanovic, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Communications of the ACM **40**(3), 66–72 (1997)
8. Blanco, H., Ricci, F.: Acquiring user profiles from implicit feedback in a conversational recommender system. In: Q. Yang, I. King, Q. Li, P. Pu, G. Karypis (eds.) RecSys, pp. 307–310. ACM (2013)
9. Blanco, H., Ricci, F., Bridge, D.: Conversational query revision with a finite user profiles model. In: G. Amati, C. Carpineto, G. Semeraro (eds.) IIR, *CEUR Workshop Proceedings*, vol. 835, pp. 77–88. CEUR-WS.org (2012)
10. Boulkrinat, S., Hadjali, A., Mokhtari, A.: Towards recommender systems based on a fuzzy preference aggregation. In: 8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13). Atlantis Press (2013)
11. Brandt, D.: How service marketers can identify value-enhancing service elements. Journal of Services Marketing **2**(3), 35–41 (1988)
12. Breese, J., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proc. of the 14th Conference on Uncertainty in Artificial Intelligence, vol. 461, pp. 43–52. San Francisco, CA (1998)
13. Bridge, D.: Towards conversational recommender systems: A dialogue grammar approach. In: ECCBR Workshops, pp. 9–22 (2002)
14. Burke, R.: Knowledge-based recommender systems. Encyclopedia of Library and Information Systems **69**(Supplement 32), 175–186 (2000)
15. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction **12**(4), 331–370 (2002)
16. Burke, R.: Hybrid web recommender systems. Lecture Notes in Computer Science **4321**, 377–408 (2007)
17. Burke, R., Ramezani, M.: Matching recommendation technologies and domains. In: Recommender Systems Handbook, pp. 367–386. Springer (2011)
18. Charnes, A., Cooper, W., Rhodes, E.: Measuring the efficiency of decision making units. European Journal of Operational Research **2**(6), 429–444 (1978)
19. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Transactions on Information Theory **14**(3), 462–467 (1968)
20. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications **21**(4), 1253–1278 (2000)
21. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society.Series B (Methodological) **39**(1), 1–38 (1977)
22. Dolan, J.: Multi-criteria clinical decision support. The Patient: Patient-Centered Outcomes Research **3**(4), 229–248 (2010). DOI 10.2165/11539470-000000000-00000. URL http://dx.doi.org/10.2165/11539470-000000000-00000
23. Drucker, H., Burges, C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: M. Mozer, M. Jordan, T. Petsche (eds.) NIPS, pp. 155–161. MIT Press (1996)
24. Ehrgott, M.: Multicriteria optimization. Springer Verlag (2005)
25. Fan, J., Xu, L.: A robust multi-criteria recommendation approach with preference-based similarity and support vector machine. In: Advances in Neural Networks–ISNN 2013, pp. 385–394. Springer (2013)

26. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An efficient boosting algorithm for combining preferences. The Journal of Machine Learning Research **4**, 933–969 (2003)
27. Fuchs, M., Zanker, M.: Multi-criteria ratings for recommender systems: An empirical analysis in the tourism domain. In: E-Commerce and Web Technologies, pp. 100–111. Springer (2012)
28. Funk, S.: Netflix update: Try this at home. http://sifter.org/ simon/journal/20061211.html (2006)
29. Gedikli, F., Jannach, D.: Rating items by rating tags. In: Proceedings of the 2010 Workshop on Recommender Systems and the Social Web at ACM RecSys, pp. 25–32 (2010)
30. Gedikli, F., Jannach, D.: Improving recommendation accuracy based on item-specific tag preferences. ACM Transactions on Intelligent Systems and Technology (TIST) **4**(1), 11 (2013)
31. Herlocker, J., Konstan, J., Terveen, L., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) **22**(1), 5–53 (2004)
32. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. **22**(1), 89–115 (2004). DOI 10.1145/963770.963774. URL http://doi.acm.org/10.1145/963770.963774
33. Jameson, A., Smyth, B.: Recommendation to groups. Lecture Notes in Computer Science **4321**, 596–627 (2007)
34. Jannach, D., Gedikli, F., Karakaya, Z., Juwig, O.: Recommending hotels based on multi-dimensional customer ratings. In: Information and Communication Technologies in Tourism 2012, pp. 320–331. Springer (2012)
35. Jannach, D., Karakaya, Z., Gedikli, F.: Accuracy improvements for multi-criteria recommender systems. In: Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 674–689. ACM (2012)
36. Jiang, T., Tuzhilin, A.: Segmenting customers from population to individuals: Does 1-to-1 keep your customers forever? IEEE Trans. on Knowl. and Data Eng. **18**(10), 1297–1311 (2006). DOI 10.1109/TKDE.2006.164. URL http://dx.doi.org/10.1109/TKDE.2006.164
37. Kelly, J., Bridge, D.: Enhancing the diversity of conversational collaborative recommendations: a comparison. Artif. Intell. Rev. **25**(1-2), 79–95 (2006)
38. Konstan, J.: Introduction to recommender systems: Algorithms and evaluation. ACM Transactions on Information Systems (TOIS) **22**(1), 1–4 (2004)
39. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: Grouplens: applying collaborative filtering to usenet news. Communications of the ACM **40**(3), 77–87 (1997)
40. Koren, Y.: Collaborative filtering with temporal dynamics. In: Proc. of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 447–456. ACM New York, NY, USA (2009)
41. Koren, Y., Bell, R.: Advances in collaborative filtering. In: Recommender Systems Handbook, pp. 145–186. Springer (2011)
42. Lakiotaki, K., Matsatsinis, N., Tsoukias, A.: Multicriteria user modeling in recommender systems. IEEE Intelligent Systems **26**(2), 64–76 (2011). DOI http://doi.ieeecomputersociety.org/10.1109/MIS.2011.33
43. Lakiotaki, K., Tsafarakis, S., Matsatsinis, N.: UTA-Rec: a recommender system based on multiple criteria analysis. In: Proc. of the 2008 ACM conference on Recommender systems, pp. 219–226. ACM New York, NY, USA (2008)
44. Lee, H., Teng, W.: Incorporating multi-criteria ratings in recommendation systems. In: IEEE International Conference on Information Reuse and Integration, pp. 273–278 (2007)
45. Lee, W., Liu, C., Lu, C.: Intelligent agent-based systems for personalized recommendations in internet commerce. Expert Systems with Applications **22**(4), 275–284 (2002)
46. Li, Q., Wang, C., Geng, G.: Improving personalized services in mobile commerce by a novel multicriteria rating approach. In: Proc. of the 17th International World Wide Web Conference. Beijing, China (2008)
47. Liu, L., Mehandjiev, N., Xu, D.L.: Multi-criteria service recommendation based on user criteria preferences. In: Proceedings of the fifth ACM conference on Recommender systems, pp. 77–84. ACM (2011)

48. Lops, P., De Gemmis, M., Semeraro, G.: Content-based recommender systems: State of the art and trends. In: Recommender systems handbook, pp. 73–105. Springer (2011)
49. Lousame, F., Sánchez, E.: Multicriteria predictors using aggregation functions based on item views. In: Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on, pp. 947–952. IEEE (2010)
50. Maneeroj, S., Samatthiyadikun, P., Chalermpornpong, W., Panthuwadeethorn, S., Takasu, A.: Ranked criteria profile for multi-criteria rating recommender. In: Information Systems, Technology and Management, pp. 40–51. Springer (2012)
51. Manouselis, N., Costopoulou, C.: Analysis and classification of multi-criteria recommender systems. World Wide Web: Internet and Web Information Systems **10**(4), 415–441 (2007)
52. Manouselis, N., Costopoulou, C.: Experimental analysis of design choices in multiattribute utility collaborative filtering. International Journal of Pattern Recognition and Artificial Intelligence **21**(2), 311–332 (2007)
53. Manouselis, N., Costopoulou, C.: Overview of design options for neighborhood-based collaborative filtering systems. Personalized Information Retrieval and Access: Concepts, Methods and Practices pp. 30–54 (2008)
54. McAuley, J., Leskovec, J., Jurafsky, D.: Learning attitudes and attributes from multi-aspect reviews. In: Data Mining (ICDM), 2012 IEEE 12th International Conference on, pp. 1020–1025. IEEE (2012)
55. McCarthy, J.: Pocket restaurantfinder: A situated recommender system for groups. In: Proc. of the Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems. Minneapolis, MN (2002)
56. Middleton, S., Shadbolt, N., De Roure, D.: Ontological user profiling in recommender systems. ACM Transactions on Information Systems (TOIS) **22**(1), 54–88 (2004)
57. Mikeli, A., Apostolou, D., Despotis, D.: A multi-criteria recommendation method for interval scaled ratings. In: Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on, vol. 3, pp. 9–12. IEEE (2013)
58. Naak, A., Hage, H., Aimeur, E.: A multi-criteria collaborative filtering approach for research paper recommendation in papyres. In: E-Technologies: Innovation in an Open World, pp. 25–39. Springer (2009)
59. Nguyen, H., Haddawy, P.: DIVA: applying decision theory to collaborative filtering. In: Proc. of the AAAI Workshop on Recommender Systems. Madison, WI (1998)
60. Nguyen, H., Haddawy, P.: The decision-theoretic video advisor. In: Proc. of the 15th Conference on Uncertainty in Artificial Intelligence (UAI'99), pp. 494–501. Stockholm, Sweden (1999)
61. Nilashi, M., Ibrahim, O., Ithnin, N.: Hybrid recommendation approaches for multi-criteria collaborative filtering. Expert Systems with Applications **41**(8), 3879–3900 (2014)
62. Nilashi, M., Ibrahim, O., Ithnin, N.: Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and neuro-fuzzy system. Knowledge-Based Systems (2014)
63. Oard, D., Kim, J.: Modeling information content using observable behavior. In: Proc. of the Annual Meeting-American Society for Information Science, vol. 38, pp. 481–488. Washington DC. (2001)
64. O'Connor, M., Cosley, D., Konstan, J., Riedl, J.: PolyLens: A recommender system for groups of users. In: Proc. of the seventh conference on European Conference on Computer Supported Cooperative Work, pp. 199–218. Kluwer Academic Publishers (2001)
65. Oh, J., Jeong, O., Lee, E.: A personalized recommendation system based on product attribute-specific weights and improved user behavior analysis. In: Proceedings of the 4th International Conference on Ubiquitous Information Management and Communication, p. 57. ACM (2010)
66. Palanivel, K., Siavkumar, R.: Fuzzy multicriteria decision-making approach for collaborative recommender systems. International Journal of Computer Theory and Engineering **2**(1), 57–63 (2010)
67. Palanivel, K., Sivakumar, R.: A study on implicit feedback in multicriteria e-commerce recommender system. Journal of Electronic Commerce Research **11**(2) (2010)

68. Palanivel, K., Sivakumar, R.: A study on collaborative recommender system using fuzzy-multicriteria approaches. International Journal of Business Information Systems **7**(4), 419–439 (2011)
69. Pazzani, M., Billsus, D.: Learning and revising user profiles: The identification of interesting web sites. Machine Learning **27**(3), 313–331 (1997)
70. Plantie, M., Montmain, J., Dray, G.: Movies recommenders systems: automation of the information and evaluation phases in a multi-criteria decision-making process. Lecture Notes in Computer Science **3588**, 633–644 (2005)
71. Premchaiswadi, W., Poompuang, P.: Hybrid profiling for hybrid multicriteria recommendation based on implicit multicriteria information. Applied Artificial Intelligence **27**(3), 213–234 (2013)
72. Reilly, J., McCarthy, K., McGinty, L., Smyth, B.: Incremental critiquing. Knowledge-Based Systems **18**(4-5), 143–151 (2005)
73. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. In: Proc. of the 1994 ACM conference on Computer supported cooperative work, pp. 175–186 (1994)
74. Resnick, P., Varian, H.: Recommender systems. Communications of the ACM **40**(3), 56–58 (1997)
75. Ribeiro, M., Lacerda, A., de Moura, E., Veloso, A., Ziviani, N.: Multi-objective pareto-efficient approaches for recommender systems. ACM Transactions on Intelligent Systems and Technology **9**(1), 1–20 (2013)
76. Ricci, F., Arslan, B., Mirzadeh, N., Venturini, A.: ITR: a case-based travel advisory system. Lecture Notes in Computer Science pp. 613–627 (2002)
77. Ricci, F., Venturini, A., Cavada, D., Mirzadeh, N., Blaas, D., Nones, M.: Product recommendation with interactive query management and twofold similarity. Lecture Notes in Computer Science pp. 479–493 (2003)
78. Rodriguez, M., Posse, C., Zhang, E.: Multiple objective optimization in recommender systems. In: Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pp. 11–18. ACM, New York, NY, USA (2012). DOI 10.1145/2365952.2365961. URL http://doi.acm.org/10.1145/2365952.2365961
79. Saaty, T.: Optimization in integers and related extremal problems. McGraw-Hill (1970)
80. Sahoo, N., Krishnan, R., Duncan, G., Callan, J.: Research note-the halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! movies. Information Systems Research **23**(1), 231–246 (2012)
81. Samatthiyadikun, P., Takasu, A., Maneeroj, S.: Bayesian model for a multicriteria recommender system with support vector regression. In: Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on, pp. 38–45. IEEE (2013)
82. Sampaio, I., Ramalho, G., Corruble, V., Prudencio, R.: Acquiring the preferences of new users in recommender systems: the role of item controversy. In: Proc. of the 17th European Conference on Artificial Intelligence (ECAI) Workshop on Recommender Systems, pp. 107–110. Riva del Garda, Italy (2006)
83. Sanchez-Vilas, F., Ismoilov, J., Lousame, F.P., Sanchez, E., Lama, M.: Applying multicriteria algorithms to restaurant recommendation. In: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01, pp. 87–91. IEEE Computer Society (2011)
84. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Application of dimensionality reduction in recommender system - a case study. In: Proc. of the Workshop on Knowledge Discovery in the Web (WebKDD) (2000)
85. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proc. of the 10th International Conference on World Wide Web, pp. 285–295. ACM, New York, NY, USA (2001)
86. Schafer, J.: Dynamiclens: A dynamic user-interface for a meta-recommendation system. In: Proc. of the Workshop on the next stage of recommender systems research at the ACM Intelligent User Interfaces Conf. (2005)

87. Schmitt, C., Dengler, D., Bauer, M.: Multivariate preference models and decision making with the maut machine. In: Proc. of the 9th International Conference on User Modeling (UM 2003), pp. 297–302 (2003)
88. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender systems handbook, pp. 257–297. Springer (2011)
89. Si, L., Jin, R.: Flexible mixture model for collaborative filtering. In: Proc. of the 20th International Conference on Machine Learning, vol. 20, pp. 704–711. AAAI Press (2003)
90. Siskos, Y., Grigoroudis, E., Matsatsinis, N.: UTA methods. Springer (2005)
91. Takasu, A.: A multicriteria recommendation method for data with missing rating scores. In: Data and Knowledge Engineering (ICDKE), 2011 International Conference on, pp. 60–67. IEEE (2011)
92. Tang, T., McCalla, G.: The pedagogical value of papers: a collaborative-filtering based paper recommender. Journal of Digital Information **10**(2) (2009)
93. Tangphoklang, P., Tanchotsrinon, C., Maneeroj, S., Sophatsathit, P.: A design of multi-criteria recommender system architecture for mobile banking business in thailand. In: Proceedings of the Second International Conference on Knowledge and Smart Technologies, vol. 2010 (2010)
94. Thorndike, E.: A constant error in psychological ratings. Journal of applied psychology **4**(1), 25–9 (1920)
95. Trabelsi, W., Wilson, N., Bridge, D.: Comparative preferences induction methods for conversational recommenders. In: P. Perny, M. Pirlot, A. Tsoukiàs (eds.) ADT, *Lecture Notes in Computer Science*, vol. 8176, pp. 363–374. Springer (2013)
96. Viappiani, P., Craig, B.: Regret-based optimal recommendation sets in conversational recommender systems. In: D. Lawrence, A. Tuzhilin, R. Burke, A. Felfernig, L. Schmidt-Thieme (eds.) RecSys, pp. 101–108. ACM (2009)
97. Viappiani, P., Pu, P., Faltings, B.: Conversational recommenders with adaptive suggestions. In: J. Konstan, J. Riedl, B. Smyth (eds.) RecSys, pp. 89–96. ACM (2007)
98. Vuorikari, R., Manouselis, N., Duval, E.: Using metadata for storing, sharing, and reusing evaluations in social recommendation: the case of learning resources. Social Information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively pp. 87–107 (2008)
99. Zarrinkalam, F., Kahani, M.: A multi-criteria hybrid citation recommendation system based on linked data. In: Computer and Knowledge Engineering (ICCKE), 2012 2nd International eConference on, pp. 283–288. IEEE (2012)
100. Zhang, Y., Zhuang, Y., Wu, J., Zhang, L.: Applying probabilistic latent semantic analysis to multi-criteria recommender system. Ai Communications **22**(2), 97–107 (2009)