

Chapter 24

Multi-Criteria Recommender Systems

Gediminas Adomavicius, Nikos Manouselis and YoungOk Kwon

Abstract This chapter aims to provide an overview of the class of multi-criteria recommender systems. First, it defines the recommendation problem as a multi-criteria decision making (MCDM) problem, and reviews MCDM methods and techniques that can support the implementation of multi-criteria recommenders. Then, it focuses on the category of *multi-criteria rating recommenders* – techniques that provide recommendations by modelling a user’s utility for an item as a vector of ratings along several criteria. A review of current algorithms that use multi-criteria ratings for calculating predictions and generating recommendations is provided. Finally, the chapter concludes with a discussion on open issues and future challenges for the class of multi-criteria rating recommenders.

24.1 Introduction

The problem of recommendation has been identified as the way to help individuals in a community to find information or items that are most likely to be interesting to them or to be relevant to their needs [4, 39, 72]. Typically, it 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 $R : Users \times Items \rightarrow R_0$, where R_0 typically is represented by non-negative integers

Gediminas Adomavicius, YoungOk Kwon
Department of Information and Decision Sciences
Carlson School of Management, University of Minnesota, Minneapolis, MN 55455, USA
e-mail: {gedas,kwonx052}@umn.edu

Nikos Manouselis
Greek Research and Technology Network (GRNET S.A.)
56 Messogeion Av., 115 27, Athens, Greece
e-mail: nikosm@grnet.gr

or real numbers within a certain range [4]. It is assumed that this function is not known for the whole $Users \times Items$ space but is specified only on some subset of it. Therefore, in the context of recommendation, we want for each user $u \in Users$ to be able to (a) estimate (or approximate) the utility function $R(u, i)$ for item $i \in Items$ for which $R(u, i)$ is not yet known, and (b) choose one or a set of items i that will maximize $R(u, i)$, i.e.,

$$\forall u \in Users, i = \arg \max_{i \in Items} R(u, i) \quad (24.1)$$

In most recommender systems, the utility function usually considers a single-criterion value, e.g., an overall evaluation or rating of an item by a user. In recent work, this assumption has been considered as limited [2, 4, 48], because the 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” [71]. 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. 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' .

In contrast, in a multi-criteria rating setting, users can provide 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 use two ratings with the same weight in making recommendations, rating their 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, 48], 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 systems that use multiple criteria to support recommendation (referred to as *multi-criteria recommender systems*), with a particular emphasis on *multi-criteria rating* ones.

The remainder of this chapter is organized as follows. First, we overview the generic recommendation problem under the prism of multi-criteria decision making (MCDM), and demonstrate the potential of applying MCDM methods to facilitate recommendation in multi-criteria settings. Second, we focus on the particular type of multi-criteria recommender systems that use multi-criteria ratings, referred to as *multi-criteria rating recommenders* because, while it has not been extensively researched, this type of systems has significant potential for better recommendation performance. We survey the state of the art algorithms for this type of recommender systems. Finally, research challenges and future research directions in multi-criteria recommender systems are discussed.

24.2 Recommendation as a Multi-Criteria Decision Making Problem

In order to introduce multiple criteria in the generic recommendation problem, one of the classic MCDM methodologies can be followed. To facilitate the discussion on how MCDM methods and techniques can be used when developing a recommender system, we followed the steps and notations proposed by Bernard Roy (one of the 1960s pioneers in MCDM methods) in the generic modeling methodology for decision making problems [77]. The discussion could also follow some other generic MCDM modeling methodologies [24, 34, 95, 97], since the scope of this section is to provide some initial insights into issues that recommender systems researchers should consider when designing a multi-criteria recommender.

Roy's [77] methodology includes four steps when analyzing a decision making problem:

1. *Defining the object of decision.* That is, defining the set of alternatives (items) upon which the decision has to be made and the rationale of the recommendation decision.
2. *Defining a consistent family of criteria.* That is, identifying and specifying a set of functions that declare the preferences of the decision maker (targeted user) upon the various alternatives. These should cover all the parameters affecting the recommendation decision and be exhaustive and non-redundant.
3. *Developing a global preference model.* That is, defining the function that synthesizes the partial preferences upon each criterion into a model that specifies the total preference of a decision maker regarding a candidate alternative.

4. *Selection of the decision support process.* This covers the design and development of the procedure, methods, or software systems that will support a decision maker when taking a decision about the set of alternatives (items), in accordance to the results of the previous steps.

We briefly review these steps in separate subsections below, and mention how each of them pertains to recommender systems.

24.2.1 Object of Decision

In recommender systems, the object of decision is item i that belongs to the set of all the candidate items. The elements of this set are referred to as *alternatives* or actions in related literature [30]. To express the rationale behind the decision, Roy [77] refers to the notion of the decision “problematics.” Four types of decision problematics are identified:

- *Choice*, which concerns the selection of one or more alternatives that can be considered as more appropriate from all candidate ones;
- *Sorting*, which refers to the classification of the alternatives into a number of pre-defined categories;
- *Ranking*, which involves ranking all the alternatives, from the best one to the worst;
- *Description*, which concerns the description of each alternative in terms of how it performs upon each criterion.

All four types of decision problematics can be considered valid for the recommendation problem:

- Choosing and recommending one or more items as more suitable for a particular user;
- Classifying (or sorting, as Roy defines it) all available items into pre-defined categories according to their suitability, e.g., into “recommended for purchase” and “recommended for viewing” items;
- Ranking all available items from the most suitable to the least suitable ones for a particular user, and presenting a ranked list of recommendations to the user;
- Describing how suitable a particular item is for a specific user, based on how it is evaluated upon each criterion. It corresponds to a full analysis of the item performance upon all criteria, illustrating the suitability of an item for the specific user (that is, in a personalized manner that aims to help the user to make a selection).

24.2.2 Family of Criteria

The performance of alternatives in set *Items* is analyzed upon a set of criteria for each user, in order to model all their characteristics, attributes, effects, or consequences [77, 97]. In recommender systems, the criteria may refer to the multiple features of an item (often the case in content-based recommendations) or to the multiple dimensions upon which the item is being evaluated/rated.

Any criterion c can be represented by function $g_c(i)$ that expresses the preferences of one user (therefore is user-specific), in order for the user to be able to decide between two alternatives i_1 and i_2 , i.e., whether $g_c(i_1) > g_c(i_2)$, in the case that alternative i_1 is preferred to alternative i_2 , or whether $g_c(i_1) = g_c(i_2)$, in the case that the two alternatives are considered equivalent (i.e., perfectly substitutable for the particular user on this criterion). To be able to make rational decisions using multiple criteria, it has to be ensured that the whole set of these functions creates a *consistent* family of criteria [77]. A family of criteria is said to be consistent when it has the following three properties:

1. *Monotonic*: a family of criteria is monotonic only if, for each pair of alternatives i_1 and i_2 , for which $g_{c_1}(i_1) > g_{c_1}(i_2)$ for one criterion c_1 and $g_c(i_1) = g_c(i_2)$ for every other criterion $c \neq c_1$, it can be assumed that alternative i_1 is preferred to alternative i_2 .
2. *Exhaustive*: a family of criteria is exhaustive only if, for each pair of alternatives i_1 and i_2 , for which $g_c(i_1) = g_c(i_2)$ upon each criterion c , we can assume that i_1 and i_2 are equivalent.
3. *Non-redundant*: a family of criteria is non-redundant only if the removal of any one of the criteria leads to the violation of one of the other two properties.

In the remainder of this chapter, unless explicitly specified otherwise, we will assume that we have a consistent family of k criteria, i.e., g_1, g_2, \dots, g_k . The design of a consistent family of criteria for a given recommendation application has been largely ignored in the recommender systems literature and constitutes an interesting and important problem for future research. Four types of criteria are usually found in MCDM [30]:

- *Measurable*, i.e., a criterion that allows its quantified measurement upon some evaluation scale;
- *Ordinal*, i.e., a criterion that defines an ordered set of acceptable values that allow its evaluation using a qualitative or a descriptive scale;
- *Probabilistic*, i.e., a criterion that uses probability distributions to represent uncertainty in its evaluation;
- *Fuzzy*, i.e., a criterion whose evaluation is represented in relation to its possibility to belong in one of the intervals of a qualitative or descriptive evaluation scale.

From a broad perspective, a family of criteria can be used to facilitate the representation of user preferences in recommender systems as well. Therefore, we can

assume that all types of criteria could be potentially engaged in multi-criteria recommender systems, although (as shown later) it seems that some types are used in currently developed systems more often than others.

24.2.3 Global Preference Model

The development of a global preference model provides a way to aggregate the values of each criterion g_c (where $c = 1, \dots, k$) in order to express the preferences between the different alternatives of the set *Items*, depending on the selected decision problematics. In the MCDM literature, a number of methodologies have been developed, which can be classified in different categories according to the form of the global preference model that they use and the process of creating this model. According to [30] and [64], the following categories of global preference modeling approaches can be identified:

- *Value-Focused models*, where a value system for aggregating the user preferences on the different criteria is constructed. In such approaches, marginal preferences upon each criterion are synthesized into a total value function, which is usually called the utility function [33]. These approaches are often referred to as multi-attribute utility theory (MAUT) approaches.
- *Multi-Objective Optimization models*, where criteria are expressed in the form of multiple constraints of a multi-objective optimization problem. In such approaches, usually the goal is to find a Pareto optimal solution for the original optimization problem [102]. They are also sometimes referred to as multi-objective mathematical programming methodologies.
- *Outranking Relations models*, where preferences are expressed as a system of outranking relations between the items, thus allowing the expression of incomparability. In such approaches, all items are pair-wise compared to each other, and preference relations are provided as relations “ a is preferred to b ”, “ a and b are equally preferable”, or “ a is incomparable to b ” [76].
- *Preference Disaggregation models*, where the preference model is derived by analyzing past decisions. Such approaches are sometimes considered as a subcategory of other modeling categories mentioned above, since they try to infer a preference model of a given form (e.g., value function or outranking relations) from some given preferential structures that have led to particular decisions in the past. Inferred preference models aim at producing decisions that are at least identical to the examined past ones [30].

Methodologies from all categories can be used in order to create global preference models for recommender systems, depending on the selected decision problematic and the environment in which the recommender system is expected to operate.

24.2.4 Decision Support Process

In this step, a final decision for a given MCDM problem is made by choosing an appropriate method among the ones defined in each of the previous steps. Like in traditional MCDM, multi-criteria recommendation problems may also need to use different methods for different domains or applications. Note, however, that this MCDM perspective is broad and not very restrictive when modeling multi-criteria recommendation problems, because many existing recommender systems can be thought to fit directly in the MCDM category, since they usually take into account information from multiple sources (e.g., user profiles and item attributes), thus making them *de facto* multi-criteria decision makers. Therefore, later in the chapter, we will focus on a particular category of MCDM recommender systems that can be differentiated from most existing recommender systems.

In Tables 24.1-24.3, we provide an overview of some sample recommender systems that could be broadly classified as MCDM (or multi-criteria recommender) systems based on the work of [48]. This survey covers systems that use one of the MCDM methods discussed in the previous section and, thus, provides insights into the way that existing MCDM approaches can be employed to support the decision-making in recommender systems.

The multi-criteria recommender systems are categorized according to the decision problematic they support (Table 24.1), the types of criteria they use (Table 24.2), and the global preference modelling approach they follow (Table 24.3). Based on Table 24.1, it is interesting to note that most of the existing research focuses on the decision problematic of *ranking* the items (i.e., ranking candidates for recommendation). There are also several systems that support the *sorting* of items into different categories according to their suitability for the user (e.g., recommended vs. non-recommended items). Very few systems support the *choice* and *description* problematic, although clearly there exist some applications in which they would prove relevant. Furthermore, as Table 24.2 illustrates, the families of criteria used are mainly *measurable*: that is, users rate items upon a measurable scale for each criterion. Nevertheless, there are also several systems that engage *fuzzy*, *ordinal*, and *probabilistic* criteria for the expression of user preferences regarding the candidate items. Finally, Table 24.3 indicates that only a few of the multi-criteria recommenders engage in the creation of the global preference model using a *multi-objective optimization* or *outranking relations*. On the contrary, the vast majority uses some *value-focused* model that typically calculates prediction in the form of an additive utility function. There are also some systems that do not synthesize the predictions from the multiple criteria, but rather use the raw vector models as their outcome (e.g., by providing a vector of ratings from all the criteria).

It is important to note that existing systems are sometimes violating the consistency rules that Roy's methodology proposes (e.g., not using an exhaustive set of dimensions). Nevertheless, experimental results often indicate that performance of multi-criteria systems is satisfactory (e.g., see the survey of algorithms that follows) even in cases where no formal modelling methodology has been followed.

This could mean that a modelling inconsistency does not always imply problematic performance, although this is an issue that calls for further investigation.

Table 24.1: Decision problematics supported by existing multi-criteria recommender systems

Choice	Ariely et al. 2004 [6], Falle et al. 2004 [23], Kleinberg and Sandler 2003 [38], Lee et al. 2002 [45], Lee 2004 [44], Price and Messinger 2005 [69], Tewari et al. 2003 [93]
Sorting	Cantador et al. 2006 [12], Choi and Cho 2004 [15], Emi et al. 2003 [22], Guan et al. 2002 [28], Kim and Yang 2004 [36], Liu and Shih 2005 [47], Masthoff 2003 [53], Montaner et al. 2002 [57], Nguyen and Haddawy 1999 [60], Nguyen and Haddawy 1998 [59], Stolze and Rjaibi 2001 [90], Wang 2004 [99], Yu 2002 [100], Yu 2004 [101], Zimmerman et al. 2004 [103]
Ranking	Adomavicius and Kwon 2007 [2], Ardissono et al. 2003 [5], Balabanovic and Shoham 1997 [7], Ghosh et al. 1999 [27], Karacapilidis and Hatzieleftheriou 2005 [32], Kerschberg et al. 2001 [35], Kim et al. 2002 [37], Lakiotaki et al. 2008 [42], Lee and Tang 2007 [43], Lee et al. 2002 [45], Li et al. 2008 [46], Manouselis and Costopoulou 2007b [49], Manouselis and Costopoulou 2007c [50], Manouselis and Sampson 2004 [52], Mukherjee et al. 2001 [58], Noh 2004 [61], Perny and Zucker 1999 [66], Perny and Zucker 2001 [67], Plantie et al. 2005 [68], Ricci and Werthner 2002 [73], Ricci and Nguyen 2007 [74], Sahoo et al. 2006 [79], Schafer 2005 [82], Schickel-Zuber and Faltings 2005 [83], Srikumar and Bhasker 2004 [89], Tang and McCalla 2009 [92], Tsai et al. 2006 [96]
Description	Aciar et al. 2007 [1], Cheetham 2003 [14], Denguir-Rekik et al. 2006 [19], Herrera-Viedma et al. 2004 [29], Schmitt et al. 2002 [84], Schmitt et al. 2003 [85], Stolze and Stroebl 2003 [91]

24.3 MCDM Framework for Recommender Systems: Lessons Learned

While, as mentioned earlier, the recommender systems surveyed in Tables 24.1-24.3 can be considered to be multi-criteria recommender systems according to the MCDM framework, it is important to understand where the existing types of recommender systems fall within this framework and also whether this MCDM framework gives rise to any novel types of recommender systems.

Recommendation techniques are often classified based on the recommendation approach into several categories: content-based, collaborative filtering, knowledge-based, and hybrid approaches [7]. Content-based recommendation techniques find the best recommendations for a user based on what the user liked in the past [65], and collaborative filtering recommendation techniques make recommendations based on the information about other users with similar preferences [8]. Knowledge-based approaches use knowledge about users and items to find the items

Table 24.2: Criteria types engaged in existing multi-criteria recommender systems

Measurable	Adomavicius and Kwon 2007 [2], Ariely et al. 2004 [6], Balabanovic and Shoham 1997 [7], Cantador et al. 2006 [12], Choi and Cho 2004 [15], Falle et al. 2004 [23], Ghosh et al. 1999 [27], Guan et al. 2002 [28], Kerschberg et al. 2001 [35], Kim and Yang 2004 [36], Kim et al. 2002 [37], Lakiotaki et al. 2008 [42], Lee and Tang 2007 [43], Lee 2004 [44], Lee et al. 2002 [45], Li et al. 2008 [46], Liu and Shih 2005 [47], Manouselis and Costopoulou 2007b [49], Manouselis and Costopoulou 2007c [50], Manouselis and Sampson 2004 [52], Masthoff 2003 [53], Montaner et al. 2002 [57], Mukherjee et al. 2001 [58], Noh 2004 [61], Plantie et al. 2005 [68], Ricci and Werthner 2002 [73], Ricci and Nguyen 2007 [74], Sahoo et al. 2006 [79], Schafer 2005 [82], Schickel-Zuber and Faltings 2005 [83], Schmitt et al. 2003 [85], Schmitt et al. 2002 [84], Srikumar and Bhasker 2004 [89], Stolze and Rjaibi 2001 [90], Tang and McCalla 2009 [92], Tewari et al. 2003 [93], Tsai et al. 2006 [96], Yu 2002 [100], Yu 2004 [101], Zimmerman et al. 2004 [103]
Ordinal	Aciar et al. 2007 [1], Cheetham 2003 [14], Emi et al. 2003 [22], Nguyen and Haddawy 1998 [59], Nguyen and Haddawy 1999 [60]
Fuzzy	Herrera-Viedma et al. 2004 [29], Karacapilidis and Hatzieleftheriou 2005 [32], Perny and Zucker 1999 [66], Perny and Zucker 2001 [67], Stolze and Stroebel 2003 [91], Wang 2004 [99]
Probabilistic	Ardissono et al. 2003 [5], Kleinberg and Sandler 2003 [38], Price and Messinger 2005 [69]

that meet users’ requirements [9]. 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 [10]. Upon more in-depth analysis of the representative MCDM recommender systems surveyed in the previous section, we discover that the multi-criteria nature of the majority of these systems can be classified in the following three general categories:

- *Multi-attribute content preference modeling.* Even though these systems typically use single-criterion ratings (e.g., numeric or binary ratings), for any given user these systems attempt to understand and model the commonalities of multi-attribute content among the items the user preferred in the past, and recommend to the user the items that best match this preferred content. For example, in a movie recommender system, these commonalities may be represented by specific genres, actors, directors, etc. that the user’s preferred movies have in common.
- *Multi-attribute content search and filtering.* These systems allow a user to specify her general preferences on content-based attributes across all items, through searching or filtering processes (e.g., searching for only “comedy” movies or specifying that “comedy” movies are preferable to “action” movies), and rec-

Table 24.3: Global preference models used in existing multi-criteria recommender systems

Value-focused models	Aciar et al. 2007 [1], Adomavicius and Kwon 2007 [2], Ariely et al. 2004 [6], Balabanovic and Shoham 1997 [7], Cantador et al. 2006 [12], Choi and Cho 2004 [15], Denguir-Rekik et al. 2006 [19], Falle et al. 2004 [23], Ghosh et al. 1999 [27], Guan et al. 200 [28], Herrera-Viedma et al. 2004[29], Karacapilidis and Hatzieleftheriou 2005 [32], Kerschberg et al. 2001 [35], Kim and Yang 2004 [36], Kim et al. 2002 [37], Kleinberg and Sandler 2003 [38],Lakiotaki et al. 2008 [42], Lee 2004 [44], Lee et al. 2002 [45], Li et al. 2008 [46], Liu and Shih 2005 [47], Manouselis and Costopoulou 2007b [49], Manouselis and Costopoulou 2007c [50], Manouselis and Sampson 2004 [52], Masthoff 2003 [53], Montaner et al. 2002 [57], Mukherjee et al. 2001 [58], Noh 2004 [61], Perny and Zucker 1999 [66], Perny and Zucker 2001 [67], Plantie et al. 2005 [68], Ricci and Werthner 2002 [73], Sahoo et al. 2006 [79], Schafer 2005 [82], Schickel-Zuber and Faltings 2005 [83], Schmitt et al. 2003 [85], Schmitt et al. 2002 [84], Srikumar and Bhasker 2004 [89], Stolze and Stroebel 2003 [91], Stolze and Rjaibi 2001 [90],Tang and McCalla 2009 [92], Tsai et al. 2006 [96], Yu 2004 [101], Yu 2002 [100], Zimmerman et al. 2004 [103]
Optimization	Lee and Tang 2007 [43], Price and Messinger 2005 [69], Tewari et al. 2003 [93]
Outranking relations	Emi et al. 2003 [22], Nguyen and Haddawy 1999 [60], Nguyen and Haddawy 1999 [59]
Other preference models	Ardissono et al. 2003 [5], Cheetham 2003 [14], Lee et al. 2002 [45], Ricci and Nguyen 2007 [74], Wang 2004 [99]

ommend to the user the items that are the most similar to her preferences and satisfy specified search and/or filtering conditions.

- *Multi-criteria rating-based preference elicitation.* These systems allow a user to specify her individual preferences by rating each item on multiple criteria (e.g., rating the story of movie *Wanted* as 2 and the visual effects of the same movie as 5), and recommend to the user the items that can best reflect the user’s individual preferences based on the multi-criteria ratings provided by this and other users.

Multi-attribute content preference modeling. One way to model user preferences is by analyzing multi-attribute content of items that users purchased or liked. Many multi-criteria recommender systems incorporate these content-based features either directly into the recommendation process (i.e., use a content-based approach) or in combination with collaborative recommendation techniques (i.e., use a hybrid approach). In these systems, users are typically allowed to implicitly or explicitly express their preferences with single-criterion ratings (e.g., item purchase history or single numeric ratings). Using these ratings, recommender systems then can learn users’ content-based preferences in an automated fashion by finding the commonalities among the individual content attributes of items that the users purchased or liked, e.g., by identifying favorite content attributes (e.g., “comedy” movies) for each user. As a result, recommendations are made taking into account these fa-

favorite content attributes [7]. Numerous traditional recommender systems that employ content-based, knowledge-based, or hybrid approaches in combination with some multi-attribute preference modeling of users can be found in this category. Several scoring or utility functions have been developed and used to rank the candidate items based on users' content-based preferences, including information retrieval-based and model-based techniques, such as Bayesian classifiers and various machine learning techniques [4]. More details on these techniques are discussed in other chapters 3.

Multi-attribute content search and filtering. In some systems, users can explicitly provide their *general* preferences on multi-attribute content of items that can be used by various searching and filtering techniques to find the most relevant items. For example, in [82] users can identify the movie genre, MPAA rating, and film length that they like and specify which attribute is the most important for their decision in choosing the movies at the current time. Then the recommender system narrows down the possible choices by searching for the items that match these additional explicit user preferences. For example, if a user indicates that she wants to watch “comedy” movies and the movie genre is the most important attribute for her, she will be recommended only comedy movies. Similarly, in [45], users also can provide to the recommender system both the preferred specifications for different content attributes as well as the corresponding importance weights for the different attributes.

Some of knowledge-based recommender systems [35, 37] can also be classified into this category, because users can provide their general preferences by building their own hierarchical taxonomy tree (i.e., where all item features are modeled in a hierarchical manner) and assigning the relative importance level to each component in the tree. As a result, the systems recommend the most relevant items according to users' preferences upon the user-defined multiple attributes of item taxonomy. Furthermore, some of hybrid recommender systems with knowledge-based approach would also fit in this category, particularly case-based reasoning recommender systems, where items are represented with multi-criteria content in a structured way (i.e., using a well-defined set of features and feature values) [88]. These systems allow users to specify their preferences on multi-attribute content of items in their search for items of interest. For example, several case-based travel recommender systems [73, 75] 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 [9, 70] 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).

Multi-criteria rating-based preference elicitation. This category of recommender systems engage multi-criteria ratings, often by extending traditional collaborative filtering approaches, that show 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. They differ from the above-surveyed systems 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 such system is the Intelligent Travel Recommender system [73], 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, many recommender systems that employ traditional content-based, knowledge-based, and hybrid techniques can be viewed as multi-criteria recommender systems, since they model user preferences based on multi-attribute content of items that users preferred in the past or allow users to specify their content-related preferences – i.e., search or filtering conditions for multi-attribute content of items (e.g., identifying the preferred movie genre or providing preferences on multiple pre-defined genre values). However, as mentioned earlier, there is a recent trend in multi-criteria recommendation that studies innovative approaches in collaborative recommendation 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*.

24.4 Multi-Criteria Rating Recommendation

In this section, we define the multi-criteria rating recommendation problem by formally extending it from its single-rating counterpart, and provide some further discussion about the advantages that additional criteria may provide in recommender systems.

24.4.1 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 R_0 (e.g., non-negative integers or real numbers within a certain range), and 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:

$$R : Users \times Items \rightarrow R_0 \tag{24.2}$$

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 Table 24.4, 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 [71], 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 Table 24.4), 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.

Table 24.4: Single-rating movie recommender system

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

24.4.2 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, \dots, R_k for each individual criterion c ($c = 1, \dots, k$), 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:

$$R : Users \times Items \rightarrow R_0 \times R_1 \times \dots \times R_k \quad (24.3)$$

or

$$R : Users \times Items \rightarrow R_1 \times \dots \times R_k \quad (24.4)$$

Given the availability of multi-criteria ratings (in addition to the traditional single overall rating) for each item, Tables 24.4 and 24.5 illustrate the potential benefits of this information for recommender systems. While *Alice* and *John* have similar preferences on movies in a single-rating setting (Table 24.4), 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 (Table 24.5). Upon further inspection of all the multi-criteria rating information, one can see that *Alice* 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.

Table 24.5: Multi-criteria movie recommender system (ratings for each item: overall, story, action, direction, and visual effects)

Target user		Wanted	WALL-E	Star Wars	Seven	Fargo	
User most similar to the target user	↘ Alice ↗	5,2,2,8,8	7,5,5,9,9	5,2,2,8,8	7,5,5,9,9	2,2,2,2,2	⌊ Ratings to be predicted
	John	5,8,8,2,2	7,9,9,5,5	5,8,8,2,2	7,8,8,2,2	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
	:	:	:	:	:	:	

Therefore, new recommendation algorithms and techniques are needed that can utilize multi-criteria ratings in recommender systems. There are already several systems implementing such algorithms, which we analyze in the next section.

24.5 Survey of Algorithms for Multi-Criteria Rating Recommenders

Recommender systems typically calculate and provide recommendations using the following two-phase process:

- *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/hers utility (i.e., N items with highest-predicted ratings).

Multi-criteria rating information can be used in both of these phases in different ways, and a number of approaches have been developed for the prediction or recommendation. Therefore, we classify the existing techniques for multi-criteria rating recommenders into two groups – techniques used during prediction and techniques used during recommendation – and describe these groups in more detail in separate subsections below.

24.5.1 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]. 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.

24.5.1.1 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, 49, 92]. In this approach, the similarities between users are computed by aggregating traditional similarities from individual criteria or using multi-dimensional distance metrics.

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. Assuming that $R(u, i)$ represents the rating that user u gives to item i , and $I(u, u')$ represents the common items that two users u and u' rated, two popular similarity measures can be formally written as follows:

- *Pearson correlation-based*:

$$sim(u, u') = \frac{\sum_{i \in I(u, u')} (R(u, i) - \overline{R(u)}) (R(u', i) - \overline{R(u')})}{\sqrt{\sum_{i \in I(u, u')} (R(u, i) - \overline{R(u)})^2} \sqrt{\sum_{i \in I(u, u')} (R(u', i) - \overline{R(u')})^2}} \quad (24.5)$$

- *Cosine-based*:

$$sim(u, u') = \frac{\sum_{i \in I(u, u')} R(u, i) R(u', i)}{\sqrt{\sum_{i \in I(u, u')} R(u, i)^2} \sqrt{\sum_{i \in I(u, u')} R(u', i)^2}} \quad (24.6)$$

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, \dots, r_k , i.e. $R(u, i) = (r_0, r_1, \dots, r_k)^1$. 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 (24.7) and (24.8). 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 (24.9). 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:*

$$sim_{avg}(u, u') = \frac{1}{k+1} \sum_{c=0}^k sim_c(u, u') \quad (24.7)$$

- *Worst-case(smallest) similarity:*

$$sim_{min}(u, u') = \min_{c=0, \dots, k} sim_c(u, u') \quad (24.8)$$

- *Aggregate similarity:*

$$sim_{aggregate}(u, u') = \sum_{c=0}^k w_c sim_c(u, u') \quad (24.9)$$

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:*

$$\sum_{c=0}^k |R_c(u, i) - R_c(u', i)| \quad (24.10)$$

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

- *Euclidean distance*:

$$\sqrt{\sum_{c=0}^k |R_c(u, i) - R_c(u', i)|^2} \quad (24.11)$$

- *Chebyshev (or maximal value) distance*:

$$\max_{c=0, \dots, k} |R_c(u, i) - R_c(u', i)| \quad (24.12)$$

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:

$$\text{dist}(u, u') = \frac{1}{|I(u, u')|} \sum_{i \in I(u, u')} d(R(u, i), R(u', i)) \quad (24.13)$$

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:

$$\text{sim}(u, u') = \frac{1}{1 + \text{dist}(u, u')} \quad (24.14)$$

Manouselis and Costopoulou [49] 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 expressed preferences 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 (i) aggregating similarity values that are calculated separately on each criterion into a single similarity and (ii) calcu-

lating 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 [49] 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 to 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 [49].

24.5.1.2 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 aggregation function, probabilistic modeling, and multilinear singular value decomposition (MSVD).

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.,

$$r_0 = f(r_1, \dots, r_k) \quad (24.15)$$

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. 24.1. First, this approach estimates k individual ratings using any recommendation technique. That is, the k -dimensional multi-criteria rating problem is decom-

posed 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 .

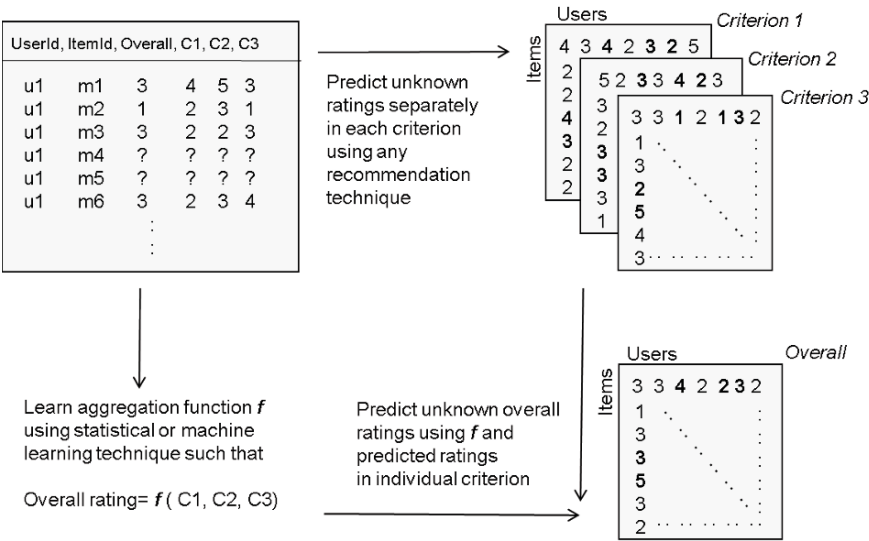


Fig. 24.1: Aggregation function approach (an example of a three-criteria rating system)

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).

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. [79], which extends the flexible mixture model (FMM) developed by Si and Jin [86] 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. 24.2a. Sahoo et al. [79] 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 [16], and incorporate the structure into the FMM, as shown in Fig. 24.2b.

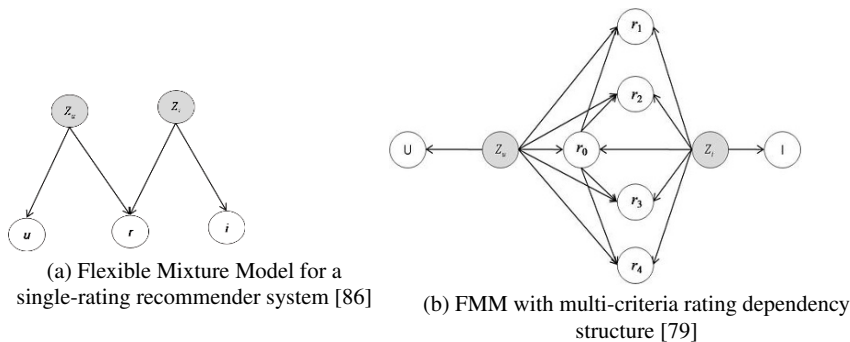


Fig. 24.2: Examples of probabilistic modeling approach in recommender systems

The FMM approach is based on the assumption that the joint distribution of three variables (user u , item i , and rating r) 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.

$$P(u, i, r) = \sum_{Z_u, Z_i} P(Z_u)P(Z_i)P(u|Z_u)P(i|Z_i)P(r|Z_u, Z_i) \quad (24.16)$$

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 the parameters of the FMM are estimated using the expectation maximization (EM)

algorithm [18]. 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 [79].

Sahoo et al. [79] also compare their model in Fig. 24.2b 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 [79]. 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. 24.2b).

Using data from Yahoo! Movies, Sahoo et al. [79] 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.

Multilinear singular value decomposition (MSVD) approach. Li et al. [46] propose a novel approach to improve a traditional collaborative filtering algorithm by utilizing the MSVD technique. Singular value decomposition (SVD) techniques have been extensively studied in numerical linear algebra and have recently gained popularity in recommender systems applications because of their effectiveness in improving recommendation accuracy [26, 41, 81]. In single-rating recommender systems, these techniques identify latent features of items including well-defined item dimensions and uninterpretable dimensions. In particular, using K latent features (i.e., rank- K SVD), user u is associated with a user-factors vector p_u (the user’s preferences on K features), and item i is associated with an item-factors 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.,

$$R^*(u, i) = p_u^T q_i \quad (24.17)$$

While the SVD techniques are commonly used as a decomposition method for two-dimensional data, the MSVD techniques [17] can be used for multi-dimensional data, such as multi-criteria ratings. In particular, 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.). More specifically, they use MSVD techniques to uncover relationships among users, items, and criteria and then use this information for identifying the nearest neighbors of each user and computing top- N recommendations. 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.

In summary, the above approaches represent 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 subsection, 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.

24.5.2 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 very 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: (i) movie X, predicted as 8 in story and 2 in visuals, and (ii) 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.

24.5.2.1 Related work: multi-criteria optimization

Multi-criteria optimization problems have been extensively studied in the operations research (OR) literature [21], 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.

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.

24.5.2.2 Designing a total order for item recommendations

In the recommender systems literature there has been some work using multi-attribute utility theories from decision science, which can be described as one way to take a linear combination of multiple criteria and find an optimal solution [42].

For example, the approach by Lakiotaki et al. [42] ranks the items by adopting the UTilités Additive (UTA) method proposed by [87]. Their algorithm aims to estimate overall utility U of a specific item for each user by adding the marginal utilities of each criterion c ($c = 1, \dots, k$).

$$U = \sum_{c=1}^k u_c(g_c) \quad (24.18)$$

where g_c is the rating provided on criterion c , and $u_c(g_c)$ is a non-decreasing real-value function (marginal utility function) for a specific user. Since this model 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. Their model 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 [49] 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:

$$U = \sum_{c=1}^k u_c = \sum_{c=1}^k w_c r_c \quad (24.19)$$

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.

24.5.2.3 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 [13]. 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 [43]. Lee and Teng [43] utilize *skyline* queries to find the best restaurants across multiple criteria (i.e., food, décor, service, and cost). As Fig. 24.3 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 [43] show that the recommender system using skyline queries helps to reduce the number 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.

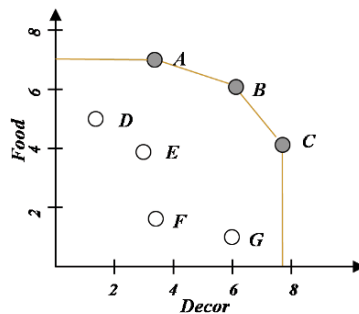


Fig. 24.3: An example of skyline points (the best candidate restaurants) in two-dimensional space

24.5.2.4 Using multi-criteria ratings as recommendation filters

Similar to how content attributes can be used as recommendation filters in recommender systems [45, 82], 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. This approach is similar to how content-based [45, 82] 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., $\text{MovieLength} < 120$ minutes) or additional contextual dimensions (e.g., $\text{TimeOfWeek} = \text{weekend}$), but on the subjective rating criteria (e.g., $\text{Story} \geq 9$), the predicted value of which is highly dependent on user's tastes and preferences.

24.6 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 investigated algorithms and techniques for multi-criteria recommender systems. These 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.

Managing intrusiveness. The extra information provided by multi-criteria ratings can give rise to an important issue of intrusiveness. 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, and some less intrusive techniques to obtain user preferences have been proposed [40, 56, 62]. Multi-criteria rating systems may 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 [42]. 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, 68]. There are also some empirical approaches with less computational complexity [80]. 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.

Reusing existing single-rating recommendation techniques. A huge number of recommendation techniques have been developed for single-rating recommender systems over the last 10-15 years, and some of them could potentially be extended to multi-criteria rating systems. For example, neighborhood-based collaborative filtering techniques may possibly take into account multi-criteria ratings using the huge number of design options that Manouselis and Costopoulou [51] suggest. As another example, there has been a number of sophisticated hybrid recommendation approaches developed in recent years [11], and some of them could potentially be adopted for multi-criteria rating recommenders. Finally, more sophisticated tech-

niques, 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.

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. [25] 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 [78], as Schmitt et al. [85] 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, as mentioned earlier, it is important to have a *consistent* family of criteria for a given recommender system application because then 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.

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 [18] that finds maximum likelihood estimates for incomplete data. In particular, the probabilistic modeling approach for multi-criteria rating prediction proposed by [79] uses the EM algorithm to predict values of the missing ratings in multi-criteria rating settings. 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.

Investigating group recommendation techniques for multi-criteria settings. Some techniques for generating recommendations to groups can be adopted in

multi-criteria rating settings. According to [31], 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, 63], 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.

Developing new MCDM modeling approaches. From the MCDM perspective, the recommendation problem is posing novel challenges to the decision modellers. On the one hand, there is a plethora of additional techniques that can be readily adopted and used in such systems, such as including a sensitivity analysis step in the algorithm, as [68] proposes. On the other hand, some studies indicate that recommendation is not a single decision making problem, since there are several decision problems that have to be addressed simultaneously, and each individual has an influence on the recommendation provided to other individuals [54]. Neither it is considered to be a typical group decision making problem nor a negotiation between individuals [66]. Therefore, new MCDM modelling approaches should be proposed and tested for multi-criteria recommendations [20].

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.

24.7 Conclusions

In this chapter, we aimed to provide an overview of multi-criteria recommender systems. We first defined the recommendation problem as an MCDM problem and reviewed the MCDM methods and techniques that can support the implementation of multi-criteria recommenders. Then, 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 prediction

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 is only a first step towards exploring this problem-rich area of recommender systems, and much more research and development are needed to unlock the full potential of multi-criteria recommenders.

Acknowledgements Research of G. Adomavicius was supported in part by the National Science Foundation grant IIS-0546443, USA. Research of N. Manouselis was funded with support by the European Commission and more specifically, the project ECP-2006-EDU-410012 “Organic.Edunet: A Multilingual Federation of Learning Repositories with Quality Content for the Awareness and Education of European Youth about Organic Agriculture and Agroecology” of the eContentplus Programme.

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 Multi-Criteria 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. Towards 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. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P. Intrigue: Personalised Recommendation of Tourist Attractions for Desktop and Handset Devices. *Applied Artificial Intelligence*, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries, 17(8):687–714, 2003.
6. Ariely, D., Lynch, J.G.Jr., Aparicio, M. Learning by Collaborative and Individual-based recommendation Agents. *Journal of Consumer Psychology*, 14(1&2):81–95, 2004.
7. Balabanovic, M., Shoham, Y. Fab:Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
8. Breese, S., Heckerman, D., Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. of the 14th Conference on Uncertainty in Artificial Intelligence*, volume 461, pages 43–52. San Francisco, CA, 1998.
9. Burke, R. Knowledge-Based Recommender Systems. *Encyclopedia of Library and Information Systems*, 69(Supplement 32):175–186, 2000.
10. Burke, R. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
11. Burke, R. Hybrid Web Recommender Systems. In P. Brusilovsky, A. Kobsa, W., Nejdl (Eds.), *The Adaptive Web: Methods and Strategies of Web, Personalization, Lecture Notes in Computer Science*, 4321:377–408, 2007.
12. Cantador, I., Fernandez, M., Castells, P. A Collaborative Recommendation Framework for Ontology Evaluation and Reuse. In *Proc. of the International ECAI Workshop on Recommender Systems*. Riva del Garda, Italy, 2006.

13. Charnes, A., Cooper, W., Rhodes, E. Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6):429–444, 1978.
14. Cheetham, W. Global Grade Selector: A Recommender System for Supporting the Sale of Plastic Resin. *Technical Information Series*, GE Global Research, TR 2003GRC261, 2003.
15. Choi, S.H., Cho, Y.H. An utility range-based similar product recommendation algorithm for collaborative companies. *Expert Systems with Applications*, 27(4):549–557, 2004.
16. Chow, C., Liu, C. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.
17. 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.
18. 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.
19. Denguir-Rekik, A., Montmain, J., Mauris, G. A fuzzy-valued Choquet-integral-based multi-criteria decision-making support for marketing and benchmarking activities in e-commerce organizations. In *Proc. of the MCDM 2006*. Chania, Greece, 2006.
20. DIMACS/LAMSADE. Computer science and decision theory: Applications of notions of consensus description of the research issues in the project. In *COST Action ICO602*. <http://dimacs.rutgers.edu/Workshops/Lamsade/propmod.pdf>, 2004.
21. Ehrgott, M. *Multicriteria Optimization*. Springer Verlag, 2005.
22. Emi, Y., Suetoshi, E., Shinohara, I., Toshikazu, K. Development of a recommendation system with multiple subjective evaluation process models. In *Proc. of the 2003 International Conference on Cyberworlds (CW03) 0-7695-1922-9/03*. Singapore, 2003.
23. Falle, W., Stoefer, D., Russ, C., Zanker, M., Felfernig, A. Using knowledge-based advisor technology for improved customer satisfaction in the shoe industry. In *Proc. of International Conference on Economic, Technical and Organisational aspects of Product Configuration Systems*, Technical University of Denmark, Copenhagen, 2004.
24. Figueira, J., Greco, S., Ehrgott, M. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer Verlag, 2005.
25. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4:933–969, 2003.
26. Funk, S. Netflix update: Try this at home. <http://sifter.org/simon/journal/20061211.html>, 2006.
27. Ghosh, S., Munde, M., Hernandez, K., Sen, S. Voting for movies: the anatomy of a recommender system. In *Proc. of the 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM, 1999.
28. Guan, S., Ngoo, C.S., Zhu, F. Handy broker: an intelligent product-brokering agent for m-commerce applications with user preference tracking. *Electronic Commerce Research and Applications*, 1(3-4):314–330, 2002.
29. Herrera-Viedma, E., Pasi, G., Lopez-Herrera, A.G. Evaluating the Information Quality of Web Sites: A Qualitative Methodology Based on Fuzzy Computing with Words. *Journal of the American Society for Information Science and Technology*, 57(4):538–549, 2006.
30. Jacquet-Lagrèze, E., Siskos, Y. Preference disaggregation: 20 years of mcda experience. *European Journal of Operational Research*, 130(2):233–245, 2001.
31. Jameson, A., Smyth, B. Recommendation to groups. Brusilovsky P., Kobsa A., and Nejdl W., eds. *Adaptive Web: Methods and Strategies of Web Personalization*, Springer, Berlin, 596–627, 2007.
32. Karacapilidis, N., Hatzieleftheriou, L. A hybrid framework for similarity-based recommendations. *International Journal of Business Intelligence and Data Mining*, 1(1):107–121, 2005.
33. Keeney, R.L. *Value-focused Thinking: A Path to Creative Decisionmaking*. Cambridge MA: Harvard Univ Press, 1992.
34. Keeney, R.L., Raiffa, H. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Cambridge Univ Press, 1976.

35. Kerschberg, L., Kim, W., Scime, A. Websifter /uppercaseII: A Personalizable Meta-Search Agent based on Semantic Weighted Taxonomy Tree. In *Proc. of the International Conference on Internet Computing*, pages 14–20. Las Vegas, Nevada, 2001.
36. Kim, T.H., Yang, S.B. Using Attributes to Improve Prediction Quality in Collaborative Filtering. In *Proc. of the E-Commerce and Web Technologies: 5th International Conference, EC-Web 2004*, Zaragoza, Spain, 2004.
37. Kim, W., Kerschberg, L., Scime, A. Learning for automatic personalization in a semantic taxonomy-based meta-search agent. *Electronic Commerce Research and Applications*, 1(2):150–173, 2002.
38. Kleinberg, J., Sandler, M. Convergent Algorithms for Collaborative Filtering. In *Proc. of the 4th ACM Conference on Electronic Commerce*, pages 1–10. San Diego, CA, 2003.
39. Konstan, J.A. Introduction to Recommender Systems: Algorithms and Evaluation. *ACM Transactions on Information Systems (TOIS)*, 22(1):1–4, 2004.
40. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, 40(3):77–87, 1997.
41. Koren, Y. Collaborative Filtering with Temporal Dynamics. In *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456. ACM New York, NY, USA, 2009.
42. 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*, pages 219–226. ACM New York, NY, USA, 2008.
43. Lee, H.H., Teng, W.G. Incorporating Multi-Criteria Ratings in Recommendation Systems. In *IEEE International Conference on Information Reuse and Integration*, pages 273–278, 2007.
44. Lee, W.P. Towards agent-based decision making in the electronic marketplace: interactive recommendation and automated negotiation. *Expert Systems with Applications*, 27(4):665–679, 2004.
45. Lee, W.P., Liu, C.H., Lu, C.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, D.R., Shih, Y.Y. Integrating AHP and data mining for product recommendation based on customer lifetime value. *Information & Management*, 42(3):387–400, 2005.
48. 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.
49. Manouselis, N., Costopoulou, C. Experimental Analysis of Design Choices in Multi-Attribute Utility Collaborative Filtering. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(2):311–332, 2007.
50. Manouselis, N., Costopoulou, C. Towards a Design Process for Intelligent Product Recommendation Services in e-Markets. *Artificial Intelligence and Integrated Information Systems: Emerging Technologies and Applications*, pages 398–417, 2007.
51. Manouselis, N., Costopoulou, C. Overview of design options for neighborhood-based collaborative filtering systems. *Personalized Information Retrieval and Access: Concepts, Methods and Practices*, pages 30–54, 2008.
52. Manouselis, N., Sampson, D. A Multi-criteria Model to Support Automatic Recommendation of e-Learning Quality Approaches. In *Proc. of the 16th World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA)*. Lugano, Switzerland, 2004.
53. Masthoff, J. Modeling the Multiple People That Are Me. In *Proc. of the International Conference on User Modelling (UM2003)*, pages 258–262. Johnstown, USA, 2003.
54. Matsatsinis, N.F., Samaras, A.P. MCDA and preference disaggregation in group decision support. *European Journal of Operational Research*, 130(2):414–429, 2001.

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.E., Shadbolt, N.R., Roure, D.C. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.
57. Montaner, M., Lopez, B., de la Rosa, J.L., Opinion-Based Filtering Through Trust. In *Proc. of the 6th International Workshop on Cooperative Information Agents VI*, LNCS 2446, pages 164–178, Springer-Verlag, 2002.
58. Mukherjee, R., Dutta, P.S., Jonsdottir, G., Sen, S. MOVIES2GO: An Online Voting Based Movie Recommender System. In *Proc. of the 5th International Conference on Autonomous Agents*, pages 114–115. Montreal, Canada, 2001.
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)*, pages 494–501. Stockholm, Sweden, 1999.
61. Noh, S. Implementing Purchasing Assistant Using Personal Profile. In *Proc. IADIS International Conference on Applied Computing*. Lisbon, Portugal, 2004.
62. Oard, D., Kim, J. Modeling information content using observable behavior. In *Proc. of the Annual Meeting-American Society for Information Science*, volume 38, pages 481–488. Washington DC., 2001.
63. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J. PolyLens: A Recommender System for Groups of Users. In *Proc. of the 7th Conference on European Conference on Computer Supported Cooperative Work*, pages 199–218. Kluwer Academic Publishers, 2001.
64. Pardalos, P., Siskos, Y., Zopounidis, C. *Advances in Multicriteria Analysis*. Dordrecht: Kluwer Academia Publishers, 1995.
65. Pazzani, M., Billsus, D. Learning and Revising User Profiles: The Identification of InterestingWeb Sites. *Machine Learning*, 27(3):313–331, 1997.
66. Perny, P., Zucker, J.D. Collaborative Filtering Methods based on Fuzzy Preference Relations. In *Proc. of EUROFUSE-SIC 99*, pages 279–285, 1999.
67. Perny, P., Zucker, J.D. Preference-based Search and Machine Learning for Collaborative Filtering: the Film-Conseil Movie Recommender System. *Information, Interaction, Intelligence*, 1(1):1–40, 2001.
68. Plantie, M., Montmain, J., Dray, G. Movies Recommenders Systems: Automation of the Information and Evaluation Phases in a Multi-criteria Decision-Making Process. In *Proc. of the DEXA 2005*, LNCS 3588, pages 633–644. Berlin Heidelberg: Springer-Verlag, 2005.
69. Price, B., Messinger, P.R. Optimal Recommendation Sets: Covering Uncertainty over User Preferences. *Proc. of the National Conference on Artificial Intelligence*, 2005.
70. Reilly, J., McCarthy, K., McGinty, L., Smyth, B. Incremental Critiquing. *Knowledge-Based Systems*, 18(4-5):143–151, 2005.
71. Resnick, P., Iacovou, N., Sushak, 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*, pages 175–186, 1994.
72. Resnick, P., Varian, H.R. Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997.
73. Ricci, F., Arslan, B., Mirzadeh, N., Venturini, A. ITR: A Case-Based Travel Advisory System. *Advances in Case-Based Reasoning*, LNAI 2416, pages 613–627. Springer-Verlag: Berlin Heidelberg, 2002.
74. Ricci, F., Nguyen, Q.N. Acquiring and Revising Preferences in a Critique-Based Mobile Recommender System. *IEEE Intelligent Systems*, 22(3):22–29, 2007.
75. Ricci, F., Venturini, A., Cavada, D., Mirzadeh, N., Blaas, D., Nones, M. Product Recommendation with Interactive Query Management and Twofold Similarity. *Proc. of the 5th International Conference on Case-Based Reasoning*, Trondheim, Norway, 2003.

76. Roy, B., Bouyssou, D. *Aide multicritère à la décision: méthodes et cas*. Economica Paris, 1993.
77. Roy, B., McCord, M.R. *Multicriteria Methodology for Decision Aiding*. Springer, 1996.
78. Saaty, T.L. *Optimization in Integers and Related Extremal Problems*. McGraw-Hill, 1970.
79. Sahoo, N., Krishnan, R., Duncan, G., Callan, J.P. Collaborative Filtering with Multi-component Rating for Recommender Systems. In *Proc. of the 16th Workshop on Information Technologies and Systems*. Milwaukee, WI, 2006.
80. 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*, pages 107–110. Riva del Garda, Italy, 2006.
81. Sarwar, B.M., Karypis, G., Konstan, J.A., 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.
82. Schafer, J.B. DynamicLens: A Dynamic User-Interface for a Meta-Recommendation Systems. In *Proc. of the Workshop on the Next Stage of Recommender Systems Research at the ACM Intelligent User Interfaces Conference*, 2005.
83. Schickel-Zuber, V., Faltings, B. Heterogeneous Attribute Utility Model: A new approach for modeling user profiles for recommendation systems. In *Proc. of the Workshop on Knowledge Discovery in the Web (WebKDD)*. Chicago, Illinois, 2005.
84. Schmitt, C., Dengler, D., Bauer, M. The MAUT-Machine: An Adaptive Recommender System. In *Proc. of the Workshop on Adaptivitat und Benutzermodellierung in Interaktiven Softwaresystemen (ABIS)*. Hannover, Germany, 2002.
85. 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)*, pages 297–302, 2003.
86. Si, L., Jin, R. Flexible Mixture Model for Collaborative Filtering. In *Proc. of the 20th International Conference on Machine Learning*, volume 20, pages 704–711. AAAI Press, 2003.
87. Siskos, Y., Grigoroudis, E., Matsatsinis, N.F. *UTA Methods*. Springer, 2005.
88. Smyth, B. Case-Based Recommendation. *The Adaptive Web: Methods and Strategies of Web Personalization*, 2007.
89. Srikumar, K., Bhasker, B. Personalized Product Selection in Internet Business. *Journal of Electronic Commerce Research*, 5(4):216–227, 2004.
90. Stolze, M., Rjaibi, W. Towards Scalable Scoring for Preference-based Item Recommendation. *IEEE Data Engineering Bulletin*, 24(3):42–49, 2001.
91. Stolze, M., Stroebe, M. Dealing with Learning in eCommerce Product Navigation and Decision Support: The Teaching Salesman Problem. In *Proc. of the 2nd Interdisciplinary World Congress on Mass Customization and Personalization*. Munich, Germany, 2003.
92. Tang T.Y., McCalla, G. The Pedagogical Value of Papers: a Collaborative-Filtering based Paper Recommender. *Journal of Digital Information*, 10(2), 2009.
93. Tewari, G., Youll, J., Maes, P. Personalized location-based brokering using an agent-based intermediary architecture. *Decision Support Systems*, 34(2):127–137, 2003.
94. Thorndike, E.L. A Constant Error in Psychological Ratings. *Journal of Applied Psychology*, 4(1):25–29, 1920.
95. Triantaphyllou, E. *Multi-Criteria Decision Making Methods: A Comparative Study*. Kluwer Academic Pub, 2000.
96. Tsai, K.H., Chiu, T.K., Lee, M.C., Wang, T.I. A Learning Objects Recommendation Model based on the Preference and Ontological Approaches. In *Proc. of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT'06)*, 2006.
97. Vincke, P. *Multicriteria Decision-Aid*. New York: J. Wiley, 1992.
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 Ef-*

- fectively*, in Go D.H. & Foo S. (Eds.), pages 87–107. Hershey, PA: Idea Group Publishing, 2008.
99. Wang, P. Recommendation Based on Personal Preference. *Computational Web Intelligence: Intelligent Technology for Web Applications*, in Zhang Y., Kandel A., Lin T., and Yao Y.(Eds.), World Scientific Publishing Company, 2004.
 100. Yu, C.C. Designing a Web-Based Consumer Decision Support Systems for Tourism Services. In *Proc. of the 4th International Conference on Electronic Commerce*, pages 23–25. Hong Kong, 2002.
 101. Yu, C.C. Innovation, management and strategy: A web-based consumer-oriented intelligent decision support system for personalized e-service. In *Proc. of the 6th International Conference on Electronic Commerce*, pages 429–437, 2004.
 102. Zeleny, M. *Linear Multiobjective Programming*. New York: Springer, 1974.
 103. Zimmerman, J., Kurapati, K., Buczak, A.L., Schaffer, D., Martino, J., Gutta, S. TV Personalization System: Design of a TV Show Recommender Engine and Interface. *Personalized Digital Television: Targeting Programs to Individual Viewers*, in Ardissono L., Kobsa A., Maybury M. (Eds.), Human-Computer Interaction Series 6, 2004.