Chapter 7
Recommender Systems: Sources of Knowledge and Evaluation Metrics

Denis Parra and Shaghayegh Sahebi

Abstract. *Recommender* or *Recommendation Systems* (*RS*) aim to help users dealing with information overload: finding relevant items in a vast space of resources. Research on *RS* has been active since the development of the first recommender system in the early 1990s, Tapestry, and some articles and books that survey algorithms and application domains have been published recently. However, these surveys have not extensively covered the different types of information used in *RS* (sources of knowledge), and only a few of them have reviewed the different ways to assess the quality and performance of *RS*. In order to bridge this gap, in this chapter we present a classification of recommender systems, and then we focus on presenting the main sources of knowledge and evaluation metrics that have been described in the research literature.

7.1 Introduction

Recommender or *Recommendation Systems* (*RS*) aim to help a user or a group of users in a system to select items from a crowded item or information space [70]. In order to generate recommendations, a *RS* might try to match users' characteristics with items' characteristics by performing content filtering, or it might look at previous interactions of the user in the system to match users with similar patterns [53]. A typical domain where RS are useful is the World Wide Web (WWW): with its

Denis Parra
School of Information Sciences, University of Pittsburgh
135 North Bellefield Avenue, Pittsburgh, PA 15260
e-mail: dap89@pitt.edu

Shaghayegh Sahebi
Intelligent Systems Program, University of Pittsburgh
Sennott Square, Pittsburgh, PA 15260, USA
e-mail: ssahebi@cs.pitt.edu

J.D. Velásquez et al. (Eds.): Advanced Techniques in Web Intelligence-2, SCI 452, pp. 149–175.
springerlink.com © Springer-Verlag Berlin Heidelberg 2013

overwhelming growth of available information and the continuously growing number of different devices that can be used to access it *RS* have taken on an important role in the daily lives of people to find relevant resources, such as movies [41], books [56], music [18], tourism destinations [12], or cooking recipes [26].

The first recommender system, Tapestry [32], was introduced almost 20 years ago by Goldberg et al. to deal with the increasing amount of messages that users received by email. This early system –as well as GroupLens developed by Paul Resnick et al. [96] and Ringo by Shardanand and Maes [107]– made use of a technique called *Collaborative Filtering* (CF) to provide recommendations to a center user based on previous actions performed by herself and by like-minded users, denoted as nearest neighbors. All these systems make use of some form of deviance measure between a predicted and a real value of preference for evaluation. In their seminal paper, Herlocker et al. [42] survey different tasks and metrics for *RS*, introducing, among others, the concepts of serendipity and novelty. However, these concepts started to have a larger impact on the evaluation of *RS* after the Netflix prize.

The *Netflix Prize*[1] was a contest created by the movie rental company Netflix[2] in October of 2006 [11]. The *Netflix Prize* challenged the data mining, machine learning and computer science communities to improve the algorithm Cinematch by at least 10% in terms of predicting the ratings that users assigned to movies. The winners of this challenge would receive a $1 million dollar prize. Netflix released a dataset of 100 million anonymous movie ratings and the evaluation was based on Root Mean Square Error (*RMSE*), a metric that we explain in section 7.4.1. Although the community of researchers engaged in *RS* existed well before this contest, the *Netflix Prize* attracted a large amount of people from different areas. It might not be a coincidence that the ACM Recommender Systems conference, targeted specifically for *RS*, began in 2007. Despite the benefit of attracting a large community of researchers to the field, the Netflix Prize had the negative effect of focusing on accuracy in the active evaluation period, giving less importance to important characteristics of the recommendations such as coverage, novelty, or diversity. By the time the challenge was finished, the *RS* community started to show more interest in other quality metrics.

Some studies have gone beyond accuracy to evaluate *RS* such as recommendation diversification by Ziegler et al. in 2005 [128] and Zhou et al. in 2010 [125], serendipity by Murakami et al. in 2008 [80] and by Zhang et al. in 2011 [124], and coverage by Ge et al. in 2010 [29]. More recently Vargas and Castells try to combine accuracy and serendipity in a single evaluation framework [113]. These new trends in *RS* evaluation stem from several factors, among which we count:

- **Accuracy and user satisfaction are not always related**: Some articles showed that rating prediction accuracy is not always correlated with other metrics [95], and most importantly, not necessarily correlated with user satisfaction [39] [70]. This result supported the need for creating new evaluation measures that better

[1] `http://www.netflixprize.com`

[2] `http://www.netflix.com`

predicted the final goal which is a user-centric evaluation of the *RS* rather than only an off-line evaluation.

- **Lack of explicit user feedback**: Although curiosity is a human trait, turning users from lurkers into real contributors to a system is a challenging task [92]. For this reason, algorithm and evaluation metrics that rely on implicit user feedback have become more frequent in recent years.
- **New sources of knowledge**: In the early days of *RS*, two contemporary popular technologies were not available: Smartphones and social networks. The first can provide a good deal of contextual information, such as temporal data, location, and additional ways to interact than a desktop computer does. The second, social networks, provides contextual information that impacts the development of trust-based methods: real family and friends. In addition, users contribute with long-term (birthday, preferred sports, art, or politics) and short-term information (*likes* on a specific comment or picture), giving *RS* different signals to produce recommendations.

In the following sections, we review *RS* by presenting a classification in section 7.2. Then, in section 7.3 we describe the main sources of knowledge used to provide recommendations, to continue with section 7.4 presenting the metrics used to evaluate quality and performance of *RS*. In section 7.5, we present all of the aforementioned concepts in the context of Web Recommendation, and we finalize summarizing the chapter adding a list of ongoing and future challenges in this area.

7.2 Classification of Recommender Systems

The ultimate goal of any user-adaptive system is to provide users with what they need without asking them explicitly [79] [115]. This identifies the difference between personalization and customization. The difference between these two is in the actor who controls the creation of user profiles as well as the presentation of interface elements to the user. In customization, the users usually control their preferences or requirements manually. On the other hand, in personalization, the user profiles are created and potentially updated by the system automatically and with minimal explicit control by the user [73]. These systems can reduce the amount of time a user spends to find her required items [27]. The process of web personalization is consisted of three phases: data preparation and transformation, pattern discovery, and recommendation [81]. In traditional collaborative filtering approaches, the pattern discovery phase (e.g., neighborhood formation in the k-nearest neighbor method) as well as the recommendation phase is performed in real time. In contrast, personalization systems which are based on web usage mining, perform the pattern discovery phase in an online state. The data preparation phase transforms raw web log files into clickstream data that can be processed through data mining tasks. A variety of data mining techniques can be applied to the clickstream or Web application data in the pattern discovery phase, such as clustering, association rule mining, and sequential pattern discovery. A recommendation engine considers the active user session in conjunction with the discovered patterns to provide personalized content [116]. The personalized content can take the form of recommended links or products, or targeted advertisements [81]. At first, traditional Recommender Systems were defined as systems that collected user opinions about various subjects and guided users towards their items of interest. This was done using collaborative filtering approaches [96], [97]. After a while, these systems started using broader research approaches and played a more active role related to users. As a result, any system that produces individualized recommendations as its output or has the effect of guiding users to interesting or useful objects is defined as a personalization system [16]. Generally, personalization is based on a mapping between users and items to interest values [3]. The learning process of Recommender Systems is divided into two general methods: memory-based (lazy-learning) Recommender Systems and model-based Recommender Systems [73]. In memory-based models, the entire data is stored and used in the memory while calculating the recommendations. As a result, these systems are sensitive to scalability issues. On the other hand, the expensive learning process in these systems gets completed offline. Model-based systems are more scalable in high data volumes.

Generally, recommender systems are divided into three groups based on their input data type, approaches to create user profiles, and algorithmic methods utilized to produce recommendations: rule-bases, content-based, and usage-based systems [73]. Each of these three groups are discussed in the following sections.

7.2.1 Rule-Based Recommender Systems

In rule-based recommender systems, decisions are made based on some rules that are extracted, either manually or automatically, from user profiles. The goal in these systems is to find factors that influence users' choice of an item or product. Many of the existing e-commerce websites use manual rule-based recommender systems. These systems permit the site administrators to set the rules based on statistical, psychological, and demographic information about users. In some cases, the rules are very domain dependent and reflect the business goals of the website. These rules are used to improve the contents provided to a user when her profile matches at least one of the conditions. Like many other rule-based systems, this method of recommendation depends on the knowledge engineering abilities of the system designers to build a suitable rule-base for specific characteristics of the domain and market. User profiles are usually achieved by explicit interaction with users. Some research has been done on the learning methods for categorizing users into different groups based on their statistical information and then inferring the required rules for recommendation [90]. These methods aim to extract personalized rules for each user by use of reasoning approaches [17]. The general mechanism in these systems is that the user announces her interests to the system and then the system assesses each of existing items for each user, based on the knowledge base it has. We can name ISCREEN [91] as one of the rule-based systems that uses manually generated rules to filter its messages. Another example is Expertise Recommender [69] which recommends expert software engineers to programmers, based on the problems they report in programming. One of the advantages of these systems is the users' capability to express characteristics of their favorite items. One of the problems in these systems, in addition to the limitations of knowledge engineering, is the method used to generate user profiles. The input to these systems is user explanations about their personal interests and as a result, it is a biased input. Profiles in these systems are usually static and consequently, the performance of the systems degraded is by time passing and aging user profiles.

7.2.2 Content-Based Recommender Systems

Content-based Recommender Systems provide recommendations to users based on comparing items or products to the items that user showed interest in. A user profile in these systems represents explanations of product characteristics that user chose before. These explanations are illustrated by a set of characteristics or features describing the products in a user profile. The act of producing recommendations usually includes comparing features of items unseen or unrated by the user with her profile's content description. The items that are similar enough to the user's profile are recommended to her.

Content-based recommender systems usually rely on Information Retrieval techniques such as classification, clustering and text analysis [77]. In most of the content-based recommender systems, especially in the web-based and e-commerce systems, content descriptions are textual features extracted from web pages or product descriptions. Typically these systems rely on known document modeling approaches, which are rooted in information retrieval and information filtering research [99] [10]. User profiles and items can be shown as weighted vectors of words (e.g. based on *tf.idf* weightening model). Predicting a user's interest in an specific item can be done based on calculating vector similarity (such as cosine similarity measure) between the user profile vector and the item profile vector or based on probabilistic methods (such as bayesian classifiers). Additionally, despite collaborative filtering methods, user profiles are created individually, based only on the items seen or rated by the user himself/herself.

We can name Letizia [65], NewsWeeder [57], Personal WebWatcher [71], InfoFinder [55], and Syskill-Webert [89] among the first examples of content-based recommender systems.

One of the problems in content-based recommender systems, due to relying on user's previous ratings and interests, is the tendency to specification in choosing items [72]. However, user studies show that users tend to be more interested in novel and surprising items suggested by recommender systems [108]. Additionally, the practical relationships between items, such as their co-occurrence of use, or being complements for accomplishing a specific task, is not considered here. Another problem is that some items based cannot be represented with specific features, such as textual, so they won't be available in these recommender systems.

7.2.3 Collaborative Filtering Recommender Systems

Collaborative filtering [41] aims to solve some of the problems in rule-based and content-based recommender systems. Collaborative filtering-based recommender systems have achieved an acceptable success in e-commerce sites [104]. These models usually include matching item ratings of the current user (like ratings on books, or movies) to similar users (close neighbors) to recommend items that are not yet seen/rated by this user. In the standard case, these systems are memory-based. Traditional collaborative filtering systems used a standard memory-based classification approach based on k-nearest neighbor (kNN) method. In this algorithm, the target user profile is compared to other user profiles to identify the first k users who have similar interests to this user. In traditional collaborative filtering, the predicted rating of active user a on each item j is calculated as a weighted sum of similar users' rankings on the same item: Equation 7.1. Where n is the number of similar users we would like to take into account, α is a normalizer, $v_{i,j}$ is the vote of user i on item j, $\bar{v}_i$ is the average rating of user i and $w(a, i)$ is the weight of this n similar users.

$$p_{a,j} = \bar{v}_a + \alpha \sum_{i=1}^{n} w(a, i)(v_{i,j} - \bar{v}_i) \tag{7.1}$$

The value of $w(a, i)$ can be calculated in many ways. Common methods are Cosine similarity, Euclidean similarity, or Pearson Correlation on user profiles.

Although these systems aim to provide a solution to issues in previous models of recommender systems, they suffer from their own problems. The most important problem of traditional memory-based collaborative filtering systems is that they are not scalable. In the kNN algorithm, formation of neighbors should be done in an online method. In other words, contrary to the model-based methods in which the model learning phase is done offline on the training data, the modeling phase in these systems is performed as an online task. With increase in users and items, this method can be unacceptably slow to produce dynamic recommendations during the interaction with users.

Another problem is due to the sparse nature of most of the datasets. More items in the dataset result in a decreased density of the user profile. As a consequence, the probability of similarity of seen items among users decreases, which results in less confidence in correlation calculations. Besides, collaborative filtering models perform at their best when there are explicit non-binary ratings for items while it is not the case for many websites. In some websites collecting user information for personalization is easier to be done using visited pages or products or asking for a product's information or changes in the shopping cart. These sources of information are considered as implicit feedback, which is discussed in section 7.3.

This method also suffers from the "new item" problem. When a new item or product is added to the item-set, it has never been seen or rated by any users. As a result, it does not exist in any user profile and the recommender system cannot recommend it to any user. The lack of ability to explain recommendations to users is another problem of these systems. Since collaborative filtering recommender systems do not use other information resources, like the content or semantic data, they cannot explain the reason for recommending a specific item to user.

To solve the sparsity and scalability problems, some use optimization techniques [5] [103] [123]. These methods include dimensionality reduction techniques, similarity indexing, and offline clustering of user profile in the past to search in the matched cluster while generating recommendations.

Another method which is based on collaborative filtering is item-based collaborative filtering [102]. In this method, a similarity matrix of items is produced based on rating data of user profiles in an offline way. This matrix is used to generate recommendations in the online phase. In other words, instead of relying the similarity between items in their content descriptions, it is calculated based on user ratings of them. Each item is shown as a vector and the similarities are calculated based on measures such as cosine similarity or based on correlation-based similarities such as Pearson or Spearman correlation. The process of generating recommendations predicts the rating of the target user to an unseen target item, by a weighted sum of given ratings to similar items to the target item. The same can be done on the item profiles. Evaluation of this method shows that it can produce recommendations with similar qualities to the model-based collaborative filtering recommendations [19].

Most of the personalization data mining methods are an extension of collaborative filtering. In these methods a pattern recognition algorithm takes prior user profiles or ratings as its input and generates an aggregated model of users. These models can be used with the current user profile to generate recommendations or predict user behavior in the future.

7.2.4 Hybrid Recommender Systems

As mentioned in the past sections, both content-based and collaborative filtering recommender systems have their own problems. Content-based recommenders cannot capture and utilize various types of similarities such as co-occurrence among items. Collaborative filtering methods have the "new item" problem. Hybrid recommender systems aim to solve the problems of content-based and collaborative filtering recommenders by use of various sources of information and combining both methods [63] [20] [21] [76]. They use both usage data of users and content data of items. Consequently, in addition to capturing the content similarities between items, these systems are able to reveal other relationships, such as associations and co-occurrences, between them. Another new direction in hybrid recommender systems is in using semantic web mining to extract semantic relationships between users and items [14] [9] [126]. Since using only keywords in finding similarities between objects has problems such as polysemy and synonymy, these models use the domain knowledge in form of a dictionary, ontology, or concept hierarchy to solve them. Some of these systems have used other sources of information such as the hierarchical link structure of a website as an additional domain knowledge [82] [98]. In general, these systems showed better results in predicting user interests.

7.3 Sources of Knowledge for Recommender Systems

7.3.1 Ratings

Ratings have been the most popular source of knowledge for *RS* to represent users's preferences from the early 1990s [96], [107], [101], to more recent years [61], [2], [51], [54]. The foundational *RS* algorithm collaborative filtering, presented in section 7.2.3, tries to find like-minded users by correlating the ratings that users have provided in a system. The goal of the algorithm is predicting users' ratings, under the assumption that this is a good way to estimate the interest that a user will show for a previously unseen item. This *rating prediction* task was the main objective of the Netflix Prize, and new algorithms were created that significantly improved the performance of the *Cinematch* algorithm. However, it has recently been shown that relying on additional information about the user or her context improves the performance of RS [4], [28]. Furthermore, in numerous occasions there are no ratings available and methods based on implicit feedback must be used [44]. The following sections describe these additional or alternative sources of knowledge.

7.3.2 Implicit Feedback

This source of knowledge refers to actions that the user performs over items, but that cannot be directly interpreted as explicit interest, i. e., the user explicitly stating her preference or the relevance of an item. This characteristic may seem as too noisy to consider using it in recommendations, however, mapping implicit and explicit feedback has been studied for several years, showing a strong correlation between both that makes implicit feedback a suitable source of knowledge to represent users' interests. Already in 1994, Morita and Shinoda [78] proved that there was a correlation between reading time on online news and self-reported preference. Konstan et al. [49] did a similar experiment with the larger user base of the Grouplens project and again found this to be true. Oard and Kim [83] performed experiments using not only reading time, but also other actions like printing an article, to find a positive correlation between implicit feedback and ratings.

Lee et al. [60] implement a recommender system based on implicit feedback by constructing "pseudo-ratings" using temporal information. In this work, the authors introduce the idea that recent implicit feedback should contribute more positively towards inferring the rating. The authors also use the idea of distinguishing three temporal bins: old, middle, and recent. Two recent works approach the issue of implicit feedback in the music domain. Jawaheer et. al analyze the characteristics of user implicit and explicit feedback in the context of last.fm music service [47]. However, their results are not conclusive due to limitations in the dataset since they only used explicit feedback available in the last.fm profiles, which is limited to the love/ban binary categories. This data is very sparse and, as the authors report, almost non-existent for some users or artists. On the other hand, Kordumova et. al use a Bayesian approach to learn a classifier on multiple implicit feedback variables [50]. Using these features, the authors are able to classify liked and disliked items with an accuracy of 0.75, uncovering the potential of mapping implicit feedback directly to preferences. In the music domain, Parra et al. [85] [87] mapped implicit feedback to explicit preference on the consumption of music albums. They found a significant effect of the number of times people listened to music and how recently the did it on the users' explicit preference (users' ratings). In a different domain, Fang and Si [23] propose a matrix co-factorization method that integrates user profile information and implicit feedback to provide recommendations of articles in the scientific portal nanohub.org.

7.3.3 Social Tags

Social Tagging systems (STS) allow users to attach free keywords, also known as tags, to items that users share or items that are already available in the system. Common examples of these systems are CiteULike[3], Bibsonomy[4] , or Mendeley[5] (mainly for academic resources), Delicious[6] (URLs), Flickr[7] (photographs), and last.fm (music). In these systems, the primary user action is the "social annotation" or "instance of tagging", corresponding to a tuple (u, i, t) where $u \in$ Users, $i \in$ Items, and $t \in$ Tags. These systems have been studied in IR (Information Retrieval) to assess their potential to improve web search. Although there are some limitations especially in terms of coverage, as social bookmarking systems capture a rather small portion of the World Wide Web, they have shown promising results [43] [120].

In these systems, the recommendation of tags and resources (urls, photographs, academic articles) has several years of research. In [46], Jschke et al. evaluate tag recommendations comparing simple heuristics methods with an adapted user-based CF method, and FolkRank, which became state-of-the-art algorithm for tag recommendations. Furthermore, Tso-Sutter et al. [112] go further by using the user

[3] www.citeulike.org

[4] www.bibsonomy.org

[5] www.mendeley.com

[6] www.delicious.com

[7] www.flickr.com

annotations to recommend items (flickr photographs) instead of tags. They evaluate several methods using recall, and the best performing one is a method that "fuses" *user* x *item*, *item* x *tag*, and *user* x *tag* dimensions. Bogers [13] performs several evaluations combining and comparing content-based information with usage-based approaches. He uses MAP (Mean Average Precision) as fundamental evaluation metric, finding positive results for methods that fuse content and usage information, but he also warns about the spam and duplicates in the social bookmarking systems as a major threat to its more wide usage as source of user interest. Parra and Brusilovsky [86] also propose two variations of user-based collaborative filtering (CF) by leveraging the users' tags in citeulike to recommend scientific articles, showing that the proposed tag-based enhancements to CF result in better precision, rank and larger coverage than traditional rating-based approaches when used on these collections.

7.3.4 Online Social Networks

Social Recommender Systems (SRSs) are recommender systems that target the social media domain [34]. The main goals for these systems are to improve recommendation quality and solve the social information overload problem. These recommender systems provide people, web pages, items, or groups as recommendations to users. They use familiarity [36] [38], as connections on social web, similarity of users who might not be familiar with each other [35] [62], and trust [59] [6] as useful features of the social web. Also, a combination of these different features can be used in a hybrid social recommender system [37].

Social recommender systems can be categorized by three groups: social recommenders for recommending items, social recommenders for recommending people, and group recommender systems. In the first category, social relationships help collaborative filtering approaches to find more accurate recommendations [31] [33]. These recommendations can come from people the user knows and thus can judge them easily. They are based on both familiarity and similarity factors and as a result they are more effective for new users. In [38], Guy et. al. showed that familiarity results in more accurate recommendations while similarity results in more diverse items.

Group Recommender Systems (GRSs) provide recommendations to a group of people. Polylens was an early group recommendation system evaluated on a large scale, built to recommend movies to groups of people [84]. In the study, O'Connor et al. showed that users value the system, and are even willing to yield some privacy to get the benefits of group recommendation. In [105] , Senot et al. evaluate different group profiling strategies on a large-scale dataset of TV viewings, showing that the utilitarian strategy was the best but acknowledging that further study was needed to generalize the results to other domains. Another study by Baltrunas et al. show that when individual recommendations are not effective, group recommendation can result in better suggestions [7].

Trust. An important line of research in *RS* has been the influence of trust in the decisions the user makes to choose recommended items. Goldbeck adopts Sztompka's definition of trust in a research where she performs several experiments relating trust, similarity and derivations of trust from either one: "Trust is a bet about the future contingent actions of others" [30]. The influence of trust and its relationship with similarity have been already shown by Sinha and Swearingen, where people tended to prefer recommendations from friends than from systems, suggesting that it is because people have more trust for friends. This connection was most strongly clarified by Ziegler and Goldbeck, showing that the more similar two people were, the greater the trust between them [127]. Similarity is one of the core components of Collaborative Filtering, but Goldbeck's results show that trust captures more nuanced facets of correlation between users in a system than only similarity [30]. Other important works in this area include Massa and Avesani's research showing how some weaknesses of *RS* can be effectively alleviated by incorporating trust [68], and also Walter et al. who investigates a model of trust-based *RS* with agents that use their social network to reach information and their trust relationships to filter it [118].

One of the main drawbacks of this technique, as pointed out by Victor et al. in [117], is the lack of publicly available datasets (other than Epinions.com, the most used on this area) that allow to test trust-based approaches.

7.3.5 Context

7.3.5.1 Location

Unlike years ago, location information about the users is now widespread with the proliferation of mobile devices that incorporate GPS technology. This has allowed the field of *RS* to incorporate this information in the recommendation process, either as the single input information or as a complementary source of knowledge. One of the earliest systems to consider location to provide recommendation in a mobile-device was CityVoyager [110] which recommended places to shop in Tokyo. The design of the system was innovative, but the user study was too small to generalize results. They asked 11 users to freely shop and evaluate their shopping experience –the shopping stores–, and with the data gathered they tuned a recommendation model and evaluated the recommendation with just two users.

Another location-aware shopping system was developed and evaluated by Yang et al. [121]. In this casev they proposed a system for recommending vendors' webpages –including offers and promotions– to interested customers. They compared four recommendation approaches (content-distance-based, content-based, distance-based, and random) in a user study with 136 undergraduate and graduate students that used the system for a period of a year and a half (January 2004 to August 2005). The evaluation measured satisfaction of the recommendations, and the content-distance-based approach had the best results overall. A more recent work by Quercia et al. [94] studied the recommendation of social events in the Boston, MA area using a mobile location-aware recommendation system. They sampled the location estimation of one million mobile users, and then combined the sample with social events in the same area, in order to infer the social events attended by 2,519 residents. Upon this data, they tested a variety of algorithms for recommending social events and found that the most effective algorithm recommends events that were popular among residents of an area. The least effective, instead, recommends events that are geographically close to the area. They evaluated the quality of the recommendations through several variations of percentile-ranking, the same metric used by Hu et al. in [44] and Fang and Si in [23], but under a different name.

7.3.5.2 Time

Although time or temporal information cannot always be considered directly as a source of preference, several methods and systems make use of time in their recommendations, especially in combination with other sources of user interest. As already mentioned in the section 7.3.3 regarding implicit feedback, Lee et al. [60] conflate implicit feedback and temporal information in a mobile e-commerce site, measuring its success by the increase in sales per recommendations provided. Another successful method incorporating time is TimeSVD++, introduced by Koren in [52], which accounts for temporal effects in the rating behavior of users and the rating pattern for items over the time. In a different approach, Lathia et al. [58] present a study of temporal effects in user preference. They study the effect on recommendations given that users continue to rate items over time, and they also investigate "the extent that the same items are being recommended over and over again". In the article, they also introduce two metrics to measure diversity and novelty, which are described in the section 7.4.

7.3.6 Heterogeneous Sources of Knowledge

Combining different sources of information has proven to be beneficial in some research cases. Fernandez-Tobias et al. present a cross-domain approach based on information obtained from the Linked Data project [25]. Using semantic representations, the authors recommend music artists based on places of interest: music venues. Another interesting case of heterogeneous data usage is the one presented by Fazel-Zarandi et al., which provides personalized expert recommendation based on semantic-data, a theoretical framework of social drivers, and social network analysis which shows promising results [24].

7.4 Evaluation Metrics for Recommender Systems

Although accuracy metrics have been frequently used to evaluate *RS* [15, 96, 107, 40], there are more dimensions that need to be assessed to capture their performance.

In a broad sense, the paper written by Herlocker et al. in 2004 [42] is a cornerstone for the evaluation of *RS*, as it describes several recommendation tasks that go beyond providing a plain list of recommended items, and many more evaluation metrics than accuracy. From this paper and further research stem the idea that the quality of a *RS* as perceived by a user is related to additional characteristics such as diversity of the recommended items [128], or how much user information and feedback needs the *RS* to perform well [111]. In the upcoming subsections, we describe several measures that have been used to evaluate these dimensions. Moreover, we include in the Section 7.4.5 the description of two frameworks recently introduced that fill the gap in the evaluation of the user experience of *RS*.

7.4.1 Prediction-Based Metrics

Prediction metrics allow one to compare which *RS* algorithm makes fewer mistakes when inferring how a user will evaluate a proposed recommendation. Predicting the ratings that a user will give to an item is the main optimization performed in rating-based *CF* recommender systems. The first of these measures is the *Mean Absolute Error* (*MAE*), which measures the mean of the absolute deviance between the predicted and the actual rating given by the users in the system.

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N} \tag{7.2}$$

In equation 7.2, p_i is the predicted rating, r_i is the actual rating and N is the total number of predictions. In order to give more importance to cases with larger deviances from the actual ratings, *Mean Squared Error* (*MSE*) is used instead of *MAE*.

$$MSE = \frac{\sum_{i=1}^{N} (p_i - r_i)^2}{N} \tag{7.3}$$

A variant of *MSE* is the *Root Mean Squared Error* (*RMSE*), which was the error metric used in the Netflix Prize.

$$RMSE = \sqrt{MSE} \tag{7.4}$$

7.4.2 Information Retrieval Related Metrics

In an scenario where a user is provided with a list of recommendations in which she can evaluate the items as relevant or not relevant, metrics used in information retrieval such as Precision, Recall, or DCG are useful to assess the quality of a recommendation method. For instance, tag-based recommendations rely heavily on these metrics since users do not usually state their preference by rating the items [13, 86].

Precision is the fraction of recommended items that are relevant [67]. It is defined as

$$Precision = \frac{|\text{relevant items recommended}|}{|\text{items in the list}|} \tag{7.5}$$

The number of items recommended in a list can be very high depending on the recommendation method and the size of the dataset, and it is not feasible that a user will be able to check and evaluate all of them. For that reason, the evaluation metric will consider only the top items, which is called Top-N recommendation [19], and it is usually presented in articles as *Precision@n*. Precision or precision@n are used to evaluate the system in the context of a single user. In order to obtain a single metric that accounts for the precision of the recommendation method over the whole set of users, *Mean Average Precision* (*MAP*) is used. *MAP* is obtained by calculating the mean over the average precision of the list of recommendations from each user, as

$$MAP = \sum_{n=1}^{N} \frac{AveP(n)}{N} \tag{7.6}$$

In the equation, $AveP(n)$ is the average precision for user n, i.e., the average of the precision values obtained for the set of top-N recommendations after each relevant recommendation is retrieved [67].

Recall is another typical metric used in information retrieval. It is defined as the fraction of relevant recommendations that are presented to the user [67]

$$Recall = \frac{|\text{relevant items recommended}|}{|\text{relevant items}|} \tag{7.7}$$

However, as described by Herlocker et al. [42], *recall* is useless in its pure sense for evaluating *RS*, since it requires knowing all the items that are relevant to a *center user*. The authors of the paper cite previous research by Sarwar et al. [100] that have approximated recall by considering those items held in the test dataset of a cross-validation evaluation as the set of relevant items. They express that this metric might be useful, but should be used carefully. Researchers must be aware of the bias underlying this metric since the items in the test dataset are just a sample of the the items that could be considered relevant. In addition, they point out that this approximated recall should be used in a comparative fashion on the same dataset and not as an absolute measure.

Usually the list of recommended items is ranked from most to less relevant. When that is the case, a useful metric is the Discounted Cumulative Gain [45], which measures how effective the recommendation method is at locating the most relevant items at the top and the less relevant items at the bottom of the recommended list. Discounted Cumulative Gain is defined as

$$DCG = \sum_{i}^{p} \frac{2^{rel_i - 1}}{\log_2(1 + i)} \tag{7.8}$$

Usually *normalized DCG* (*nDCG*) [45] is used more frequently, since it allows one to compare the DCG of lists with different length. It is calculated by normalizing the discounted cumulative gain of an ordered list of recommended items by the ideal order of those items if they were ranked perfectly

$$nDCG = \frac{DCG}{iDCG} \tag{7.9}$$

7.4.3 Diversity, Novelty and Coverage

Diversity has been shown to be an important factor in user satisfaction regarding system recommendations [128, 124]. Ziegler et al. study how diversity affects a user's opinion, and they derive the *Intra-list Similarity* metric

$$ILS(P_{w_i}) = \frac{\sum_{b_k \in P_{w_i}} \sum_{b_k \in P_{w_i}, b_k \neq b_c} c_o(b_k, b_c)}{2} \tag{7.10}$$

Higher scores of ILS denote lower diversity. Based on this metric, the authors propose a topic diversification algorithm. The results of offline and a large online user study show that "the user's overall liking of recommendation lists goes beyond accuracy and involves other factors, e.g., the users' perceived list diversity" [128].

On a different approach, Lathia et al. [58] introduced two metrics to measure diversity and novelty respectively. They use these measures to evaluate the *RS* performance when considering the drift in users' preferences over time. The metrics are diversity at depth N (7.11) and novelty (7.12)

$$diversity(L1, L2, N) = \frac{|\frac{L2}{L1}|}{N} \tag{7.11}$$

The ratio $L2/L1$ corresponds to the fraction of elements in the list *L2* that are not in the list *L1*. The second metric is novelty, which compares the current list *L2* to the set of all items that have been recommended to date A_t

$$novelty(L2, N) = \frac{|\frac{L2}{A_t}|}{N} \tag{7.12}$$

Coverage usually refers to the proportion of items that a *RS* can recommend, a concept also called *catalog coverage*. There are also some alternatives to measure coverage during an off-line or on-line experiment, where it is desirable to weight the items by popularity or utility in order, as described in [106]. The same authors describe coverage from the users' point of view, user coverage, understood as the proportion of users for which the system can produce recommendations, as used by Parra and Brusilovsky in [88].

7.4.4 *Implicit Feedback and Partial Knowledge of User Preferences*

In recent years, the research on *RS* has expanded beyond rating-based systems to cope with systems that do not rely on ratings and, even more, that rely mainly on implicit feedback from the users. Under this scenario, several metrics have been introduced, the most important being the *Mean Percentage Ranking* (*MPR*), also known as *Percentile Ranking*. It is used when the knowledge source of user interest is implicit feedback. It is a recall-oriented metric, because the authors that have used it [23] [44] state that precision based metrics are not very appropriate as they require knowing which resources are undesirable to a user. Lower values of *MPR* are more desirable. The expected value of MPR for random predictions is 50%, and thus MPR ¿ 50% indicates an algorithm no better than random.

$$MPR = \frac{\sum_{ui} r^t_{ui} \cdot \overline{rank_{ui}}}{\sum_{ui} r^t_{ui}} \tag{7.13}$$

Where r_{ui} indicates if the user u consumed the item i and $\overline{rank_{ui}}$ denotes the percentile-ranking of i within an ordered list. In this way, $\overline{rank_{ui}} = 0\%$ means that i is at the top of the list [44].

Another metric intended for implicit feedback datasets is *AP Correlation*. It was introduced by Yilmaz et al. [122] as a modification to Kendall's Tau in order to penalize mistakes made regarding highly relevant items more than for less relevant ones. AP correlation finds the precision between two orders at each index in the list and takes the average of these values

$$\tau_{a\rho} = \frac{2}{N-1} \cdot [\sum_{i \in I} \frac{C(i)}{index(i) - 1}] - 1 \tag{7.14}$$

N is the number of ranked items in the list, $C(i)$ is the number of items at an index less than $index(i)$ that are correctly ranked according to the ground truth. *AP correlation* ranges from +1 to -1. One problem with this metric is that it assumes that the ground truth list and the evaluated list give a total order, so when just partial orders are available, it is unusable.

In order to deal with partial orders, the *Expected Discounted Rank Correlation* (*EDRC*) introduced by Ackerman and Chen [1], combines *AP correlation* with *nDCG* to measure the similarity between two sets of pairwise preferences. Similar to both of them, EDRC emphasizes preserving the order of the user's most preferred items and applying a penalty for less preferred items. This metric tries to solve an important evaluation issue, that has been well introduced but not yet tested.

7.4.5 Beyond Metrics: User Study Frameworks

Evaluating the users' experience in *RS* has lagged compared to off-line evaluations, since it has not been standardized and it is usually time-consuming. Only recently, in the Recommender Systems Conference of 2011[8], two user evaluation frameworks were introduced, one by Knijnenburg et al. [48] and the other by Pu et al. [93].

The Knijnenburg et al. framework is characterized by subjective and objective evaluations of the *user experience* (*UX*). Figure 7.1 illustrates the framework. To start the evaluation, they consider *objective system aspects* (*OSA*): algorithms, visual and interaction design of the system, the way recommendations are presented and other traits such as social networking. The *subjective system aspects* (*SSA*) contain the users' perception of the *OSA* which are evaluated with questionnaires: their main objective is showing whether the objective aspects (personalization) are perceived at all.

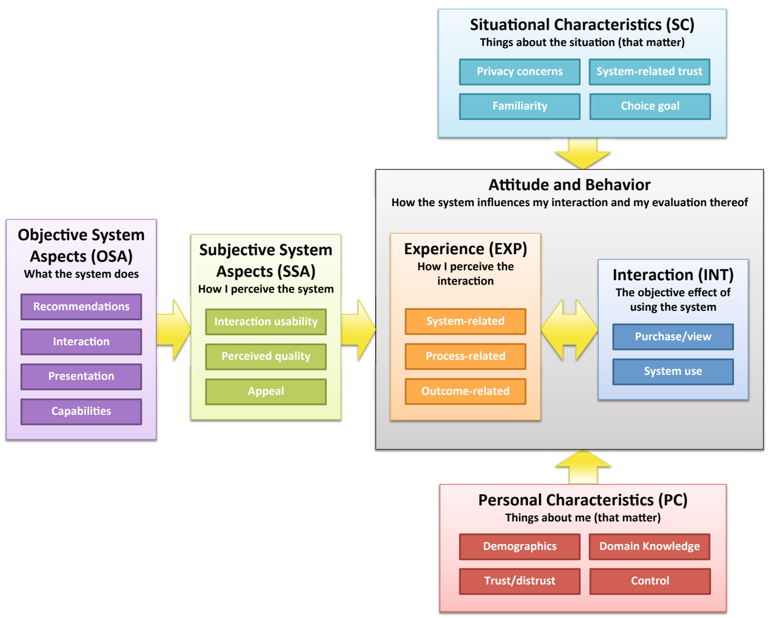

Fig. 7.1 Kijnenburg's et al. [48] UX evaluation framework

To distinguish between attitude and behavior, Knijnenburg et al. model considers the *experience* (*EXP*) and *interaction* (*INT*). The experience consists of the users' evaluation of the system, also evaluated by questionaries, as *SSA*, and is divided

[8] http://recsys.acm.org/2011

into the evaluation of the system, the decision process, and the final decisions made. The interaction is, on the other side, the observable behavior of the user. Finally, the model also considers that experience and interaction are influenced by *personal* (*PC*) and *situational* (*SC*) characteristics. Personal characteristics include demographics, trust, domain knowledge and perceived control. The second set, situational characteristics, depend on the context of the interaction.

In [93], Pu et al. introduced a unifying evaluation framework called ResQue (Recommender systems' Quality of user experience). They built this framework upon well-known usability evaluation models such as TAM (Technology Acceptance Model) and SUMI (Software Usability Measurement Inventory), although Knijnenburg et al. also make use of the first one to develop his framework. Pu et al. cite Kninenburg's framework in their related work but they argue that it fails to relate users perception to the likelihood of user adoption of the system. The main component of ResQue model are four dimensions: the perceived system qualities, users' beliefs, their subjective attitudes, and their behavioral intentions. The first, perceived system qualities, refers to user's perception of the objective characteristics of a recommender system (recommendations quality, interface adequacy, interaction adequacy, and information sufficiency and explicability). The second dimension, Beliefs, refers to a higher level of user perception of the system, influenced by perceived qualities (perceived usefulness, ease of use, and control and transparency). The third dimension, attitudes, refers to the user's overall feeling toward a recommender, likely to be derived from experience (overall satisfaction, confidence inspiring, and trust). Finally, the fourth dimension is about behavioral intentions towards a system that can influence a user's decision to use the system or consume some of the recommended resuts.

7.5 Web Recommendations

Although one of the main motivations for developing *RS* is, as described in the abstract of this book chapter, the amount of information available on the Web, Web *RS* are more closely referred to as part of Web Usage Mining in literature than to the approaches explained in Section 7.2. In this section, we aim to provide a bridge between Web Usage Mining and the techniques for building *RS*, i.e., for adaptive web personalization.

7.5.1 Sources of Knowledge for Web Recommendation

Facca et al. [22] identify three main sources of data for web usage mining: server side, proxy side and client side. At the server level, web server logs are typically found in three ways: Common Log Format, Extended Log Format, or LogML. Other sources from the server side are cookies and TCP/IP packet sniffers. The second main source of data, the proxy side, is similar to the data that can be captured from the server side, but it collects data of groups of users by accessing a large group of serves. Finally, on the client side, Javascript, Java Applets or modified browsers allows us to capture usage data. Some researchers have explored combining other sources of data for web site recommendation, such as Li et al. [63] who combine usage data with content and structure for web site recommendation. More recent research has also shown the use of additional features such as gender, age, and geographical information and they have proved to be beneficial for recommendation, such as Li et al. work on recommending personalized news in the Yahoo! portal [64].

7.5.2 Methods for Web Recommendation

In [74], Mobasher identifies the primary methods used in Web *RS* for off-line model building –preferred over memory-based models due to performance and scalability issues–, which are Clustering, Association Rule Discovery, Sequential Pattern Discovery, Markov Models, and Latent Models. Baraglia et al. introduce the *SUGGEST 3.0* system that uses clustering in the first of two steps of their method to produce recommendations [8]. Velasquez et al. also show the effectiveness of clustering for online navigation recommendations [114]. Association rules is a frequently used method in web usage mining and for web recommendations. Mobasher et al. use association rules in conjunction with clustering in [75] to recommend URLs using as dataset the Web site of the Association for Consumer Research logs. Lin et al. make use of association rules with an underlying collaborative approach [66] to produce recommendations. In Markov models, one distinguishing example of *Markov Decision Process* (*MDP*) is the *RS* implemented by Shani et al. in 2005. The authors change the usual approach of seeing the recommendations as a rating prediction problem, and they turn it into a sequential optimization process, implementing it in a commercial system. Regarding latent models, a tensor factorization method for personalized web search recommendation called CubeSVD is introduced by Sun et al. in [109]. An alternative approach is taken by Xu et al., who make use of *Latent Dirichlet Allocation* (*LDA*) in a collaborative Web Recommendation framework to model the latent topic space and discover associations between user sessions via probability inference [119].

7.5.3 Evaluation Metrics for Web Recommendation

Evaluation metrics used on Web recommendation do not differ too much from those presented in section 7.4. However, in e-commerce the success of a recommendation method is usually measured by the increase in sales or some signal of user engagement. Mobasher et. al use in [75] precision, coverage, F1 (the harmonic mean between precision and coverage) and weighted average visit percentage (WAVP) to evaluate individual profile effectiveness. This last measure, is defined as:

$$WAVP = \left(\sum_{t \in T_{pr}} \frac{\mathbf{t} \cdot \mathbf{pr}}{|\mathbf{t}|} \right) \left(\sum_{p \in pr} weight(p, pr) \right) \tag{7.15}$$

where t is a specific transaction, T_{pr} the subset of transactions whose elements contain at least one page from pr.

7.6 Summary

In this chapter, we have presented *RS* beginning with its historical evolution from the early nineties to present day. In order to give users new to this area an introduction to the most common methods, we provided a classification of the main *RS* approaches. Then, we focused on the sources of knowledge and evaluation measures used to assess *RS* performance and quality. In the last section, we tried to bridge the trends seen in *RS* research with web recommendations, which is the main focus of this book. In the coming years, we expect to see an increasing amount of commercially-available recommender systems, since they are mature in several domains as a technology to engage users and alleviate information overload. New challenges are presented by the growing amount of devices and heterogeneous sources of knowledge available, at different levels of analysis, to provide recommendations. Some of these challenges go beyond the current trends of scalability and big data: data sparsity; how to deal with the new user and new item problems; how to automatically select a recommendation method given an special context; add transparency, diversity and serendipity to *RS*; how to leverage social networks; how to use implicit feedback; how to assure that off-line evaluation results correlates with on-line user satisfaction; among others. All of these issues remain at least partially unsolved and we expect to see a good deal of applications and research around these topics.

Acknowledgements. The first author, Denis Parra, has been partially supported by Conicyt (Comisin Nacional de Investigacin Cientfica y Tecnolgica, Gobierno de Chile.) to pursue his PhD program in Information Sciences at the University of Pittsburgh.

References

1. Ackerman, B., Chen, Y.: Evaluating rank accuracy based on incomplete pairwise preferences. In: UCERSTI 2: Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces, vol. 2 (2011)
2. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. ACM Trans. Inf. Syst. 23(1), 103–145 (2005)
3. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17, 734–749 (2005)

4. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 217–253. Springer US (2011), doi:10.1007/978-0-387-85820-3
5. Aggarwal, C.C., Wolf, J.L., Yu, P.S.: A new method for similarity indexing of market basket data. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD 1999, pp. 407–418. ACM, New York (1999)
6. Avesani, P., Massa, P., Tiella, R.: A trust-enhanced recommender system application: Moleskiing. In: SAC 2005: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 1589–1593. ACM Press (2004)
7. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 119–126. ACM, New York (2010)
8. Baraglia, R., Silvestri, F.: An online recommender system for large web sites. In: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2004, pp. 199–205. IEEE Computer Society, Washington, DC (2004)
9. Beemanapalli, K., Rangarajan, R., Srivastava, J.: Incorporating Usage Information into Average-Clicks Algorithm, pp. 21–35 (2007)
10. Belkin, N.J., Bruce Croft, W.: Information filtering and information retrieval: two sides of the same coin? Commun. ACM 35(12), 29–38 (1992)
11. Bennett, J., Lanning, S., Netflix, N.: The netflix prize. In: KDD Cup and Workshop in Conjunction with KDD (2007)
12. Berger, H., Denk, M., Dittenbach, M., Pesenhofer, A., Merkl, D.: Photo-Based User Profiling for Tourism Recommender Systems. In: Psaila, G., Wagner, R. (eds.) EC-Web 2007. LNCS, vol. 4655, pp. 46–55. Springer, Heidelberg (2007)
13. Bogers, T.: Recommender Systems for Social Bookmarking. PhD thesis, Tilburg University (December 2009)
14. Bose, A., Beemanapalli, K., Srivastava, J., Sahar, S.: Incorporating Concept Hierarchies into Usage Mining Based Recommendations. In: Nasraoui, O., Spiliopoulou, M., Srivastava, J., Mobasher, B., Masand, B. (eds.) WebKDD 2006. LNCS (LNAI), vol. 4811, pp. 110–126. Springer, Heidelberg (2007)
15. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)
16. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction 12, 331–370 (2002)
17. Burke, R.D., Hammond, K.J., Young, B.C.: The findme approach to assisted browsing. IEEE Expert. 12, 32–40 (1997)
18. Celma, Í., Serra, X.: Foafing the music: Bridging the semantic gap in music recommendation. Web Semant. 6, 250–256 (2008)
19. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. 22(1), 143–177 (2004)
20. Eirinaki, M., Lampos, H., Vazirgiannis, M., Varlamis, I.: Sewep: Using site semantics and a taxonomy to enhance the web personalization process, pp. 99–108 (2003)
21. Eirinaki, M., Lampos, C., Paulakis, S., Vazirgiannis, M.: Web personalization integrating content semantics and navigational patterns. In: Proceedings of the 6th Annual ACM International Workshop on Web Information and Data Management, WIDM 2004, pp. 72–79. ACM, New York (2004)
22. Facca, F.M., Lanzi, P.L.: Mining interesting knowledge from weblogs: a survey. Data & Knowledge Engineering 53(3), 225–241 (2005)

23. Fang, Y., Si, L.: Matrix co-factorization for recommendation with rich side information and implicit feedback. In: Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec 2011, pp. 65–69. ACM, New York (2011)
24. Fazel-Zarandi, M., Devlin, H.J., Huang, Y., Contractor, N.: Expert recommendation based on social drivers, social network analysis, and semantic data representation. In: Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec 2011, pp. 41–48. ACM, New York (2011)
25. Fernández-Tobías, I., Cantador, I., Kaminskas, M., Ricci, F.: A generic semantic-based framework for cross-domain recommendation. In: Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec 2011, pp. 25–32. ACM, New York (2011)
26. Freyne, J., Berkovsky, S., Smith, G.: Recipe Recommendation: Accuracy and Reasoning. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 99–110. Springer, Heidelberg (2011)
27. Fry, C., Bow Street Software, Weitzman, L., Ibm: Why surf alone?: Exploring the web with reconnaissance agents, internet (accessed: 20, 2001)
28. Gantner, Z., Rendle, S., Schmidt-Thieme, L.: Factorization models for context-/time-aware movie recommendations. In: Proceedings of the Workshop on Context-Aware Movie Recommendation, CAMRa 2010, pp. 14–19. ACM, New York (2010)
29. Ge, M., Delgado-Battenfeld, C., Jannach, D.: Beyond accuracy: evaluating recommender systems by coverage and serendipity. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys 2010, pp. 257–260. ACM, New York (2010)
30. Golbeck, J.: Trust and nuanced profile similarity in online social networks. ACM Trans. Web 3(4), 12:1–12:33 (2009)
31. Golbeck, J., Hendler, J.: Filmtrust: Movie recommendations using trust in web-based social networks, vol. 6, pp. 497–529. ACM, New York (2006)
32. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM 35, 61–70 (1992)
33. Groh, G.: Recommendations in taste related domains: Collaborative filtering vs. social filtering. In: Proc ACM Group 2007, pp. 127–136 (2007)
34. Guy, I., Carmel, D.: Social recommender systems. In: Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011, pp. 283–284. ACM, New York (2011)
35. Guy, I., Jacovi, M., Perer, A., Ronen, I., Uziel, E.: Same places, same things, same people?: mining user similarity on social media. In: Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW 2010, pp. 41–50. ACM, New York (2010)
36. Guy, I., Jacovi, M., Shahar, E., Meshulam, N., Soroka, V., Farrell, S.: Harvesting with sonar: the value of aggregating social network information. In: Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, CHI 2008, pp. 1017–1026. ACM, New York (2008)
37. Guy, I., Ronen, I., Wilcox, E.: Do you know?: recommending people to invite into your social network. In: Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI 2009, pp. 77–86. ACM, New York (2009)
38. Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S., Ofek-Koifman, S.: Personalized recommendation of social software items based on social relations. In: Proceedings of the Third ACM Conference on Recommender Systems, RecSys 2009, pp. 53–60. ACM, New York (2009)

39. Hayes, C., Hayes, C., Massa, P., Cunningham, P., Avesani, P., Cunningham, P.: An online evaluation framework for recommender systems. In: Workshop on Personalization and Recommendation in E-Commerce (Malaga). Springer (2002)
40. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Inf. Retr. 5(4), 287–310 (2002)
41. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999, pp. 230–237. ACM, New York (1999)
42. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22, 5–53 (2004)
43. Heymann, P., Koutrika, G., Garcia-Molina, H.: Can social bookmarking improve web search? In: First ACM International Conference on Web Search and Data Mining, WSDM 2008 (February 2008)
44. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, pp. 263–272. IEEE Computer Society, Washington, DC (2008)
45. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. 20, 422–446 (2002)
46. Jäschke, R., Marinho, L., Hotho, A., Lars, S.-T., Gerd, S.: Tag recommendations in social bookmarking systems. AI Commun. 21, 231–247 (2008)
47. Jawaheer, G., Szomszor, M., Kostkova, P.: Comparison of implicit and explicit feedback from an online music recommendation service. In: HetRec 2010: Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 47–51. ACM, New York (2010)
48. Knijnenburg, B.P., Willemsen, M.C., Kobsa, A.: A pragmatic procedure to support the user-centric evaluation of recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys 2011, pp. 321–324. ACM, New York (2011)
49. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: Grouplens: applying collaborative filtering to usenet news. Commun. ACM 40(3), 77–87 (1997)
50. Kordumova, S., Kostadinovska, I., Barbieri, M., Pronk, V., Korst, J.: Personalized Implicit Learning in a Music Recommender System. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 351–362. Springer, Heidelberg (2010)
51. Koren, Y.: Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: ACM KDD, pp. 426–434 (2008)
52. Koren, Y.: Collaborative filtering with temporal dynamics. In: ACM KDD, Paris, France, pp. 89–97 (2009)
53. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer 42(8), 30–37 (2009)
54. Koren, Y., Sill, J.: Ordrec: an ordinal model for predicting personalized item rating distributions. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys 2011, pp. 117–124. ACM, New York (2011)
55. Krulwich, B., Burkey, C.: Learning user information interests through extraction of semantically significant phrases. In: Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access, pp. 100–112 (1996)
56. Kuroiwa, T., Bhalla, S.: Book recommendation system for utilisation of library services. Int. J. Comput. Sci. Eng. 5, 207–213 (2010)

57. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the 12th International Machine Learning Conference, ML 1995 (1995)
58. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, pp. 210–217. ACM, New York (2010)
59. Lee, D.H.: Pittcult: trust-based cultural event recommender. In: Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, pp. 311–314. ACM, New York (2008)
60. Lee, T., Park, Y., Park, Y.: A time-based approach to effective recommender systems using implicit feedback. Expert Syst. Appl. 34(4), 3055–3062 (2008)
61. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: Proceedings of SIAM Data Mining SDM 2005 (2005)
62. Lerman, K.: Social networks and social information filtering on digg. CoRR, abs/cs/0612046 (2006)
63. Li, J., Zaïane, O.R.: Combining Usage, Content, and Structure Data to Improve Web Site Recommendation. In: Bauknecht, K., Bichler, M., Pröll, B. (eds.) EC-Web 2004. LNCS, vol. 3182, pp. 305–315. Springer, Heidelberg (2004)
64. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation. In: Proceedings of the 19th International Conference on World Wide Web, WWW 2010, pp. 661–670. ACM, New York (2010)
65. Lieberman, H.: Letizia: An agent that assists web browsing. In: International Joint Conference on Artificial Intelligence, pp. 924–929 (1995)
66. Lin, W., Alvarez, S.A., Ruiz, C.: Efficient adaptive-support association rule mining for recommender systems. Data Min. Knowl. Discov. 6(1), 83–105 (2002)
67. Manning, C.D., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
68. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys 2007, pp. 17–24. ACM, New York (2007)
69. McDonald, D.W., Ackerman, M.S.: Expertise recommender: a flexible recommendation system and architecture. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW 2000, pp. 231–240. ACM, New York (2000)
70. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: CHI 2006 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2006, pp. 1097–1101. ACM, New York (2006)
71. Mladenic, D.: Personal webwatcher: design and implementation (1996)
72. Mladenic, D.: Text-learning and related intelligent agents: A survey. IEEE Intelligent Systems 14(4), 44–54 (1999)
73. Mobasher, B.: Data Mining for Web Personalization. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 90–135. Springer, Heidelberg (2007)
74. Mobasher, B.: Data Mining for Web Personalization. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 90–135. Springer, Heidelberg (2007)
75. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Discovery and evaluation of aggregate usage profiles for web personalization. Data Min. Knowl. Discov. 6(1), 61–82 (2002)
76. Mobasher, B., Dai, H., Luo, T., Sun, Y., Zhu, J.: Integrating Web Usage and Content Mining for More Effective Personalization. In: Bauknecht, K., Madria, S.K., Pernul, G. (eds.) EC-Web 2000. LNCS, vol. 1875, pp. 165–176. Springer, Heidelberg (2000)

77. Mooney, R.J., Roy, L.: Content-based book recommending using learning for text categorization. In: Proceedings of the Fifth ACM Conference on Digital Libraries, DL 2000, pp. 195–204. ACM, New York (2000)
78. Morita, M., Shinoda, Y.: Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval. In: SIGIR 1994: Proceedings of the 17th Annual International ACM SIGIR Conference, pp. 272–281. Springer-Verlag New York, Inc., New York (1994)
79. Mulvenna, M.D., Anand, S.S., Büchner, A.G.: Personalization on the net using web mining: introduction. Commun. ACM 43, 122–125 (2000)
80. Murakami, T., Mori, K., Orihara, R.: Metrics for Evaluating the Serendipity of Recommendation Lists. In: Satoh, K., Inokuchi, A., Nagao, K., Kawamura, T. (eds.) JSAI 2007. LNCS (LNAI), vol. 4914, pp. 40–46. Springer, Heidelberg (2008)
81. Nakagawa, M., Mobasher, B.: A Hybrid Web Personalization Model Based on Site Connectivity
82. Nasraoui, O., Frigui, H.: Extracting web user profiles using relational competitive fuzzy clustering (2000)
83. Oard, D., Kim, J.: Modeling information content using observable behavior. In: Proc. of the ASIST Annual Meeting, pp. 481–488 (2001)
84. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work, ECSCW 2001, pp. 199–218. Kluwer Academic Publishers, Norwell (2001)
85. Parra, D., Amatriain, X.: Walk the Talk: Analyzing the Relation between Implicit and Explicit Feedback for Preference Elicitation. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 255–268. Springer, Heidelberg (2011)
86. Parra, D., Brusilovsky, P.: Collaborative filtering for social tagging systems: an experiment with citeulike. In: Proceedings of the Third ACM Conference on Recommender Systems, RecSys 2009, pp. 237–240. ACM, New York (2009)
87. Parra, D., Karatzoglou, A., Amatriain, X.: Implicit Feedback Recommendation via Implicit-to-Explicit Ordinal Logistic Regression Mapping, vol. 1 (2011)
88. Parra-Santander, D., Brusilovsky, P.: Improving collaborative filtering in social tagging systems for the recommendation of scientific articles. In: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, pp. 136–142. IEEE Computer Society, Washington, DC (2010)
89. Pazzani, M., Billsus, D., Michalski, S., Wnek, J.: Learning and revising user profiles: The identification of interesting web sites. In: Machine Learning, pp. 313–331 (1997)
90. Pazzani, M.J.: A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review 13, 393–408 (1999)
91. Pollock, S.: A rule-based message filtering system. ACM Trans. Inf. Syst. 6, 232–254 (1988)
92. Preece, J., Shneiderman, B.: The reader-to-leader framework: Motivating technology-mediated social participation. AIS Transactions on Human Computer Interaction 1(1), 13–32 (2009)
93. Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys 2011, pp. 157–164. ACM, New York (2011)
94. Quercia, D., Lathia, N., Calabrese, F., Di Lorenzo, G., Crowcroft, J.: Recommending social events from mobile phone location data. In: Proceedings of IEEE ICDM 2010 (December 2010)

95. Redpath, J., Glass, D.H., McClean, S., Chen, L.: Collaborative Filtering: The Aim of Recommender Systems and the Significance of User Ratings. In: Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S., van Rijsbergen, K. (eds.) ECIR 2010. LNCS, vol. 5993, pp. 394–406. Springer, Heidelberg (2010)
96. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW 1994, pp. 175–186. ACM, New York (1994)
97. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM 40, 56–58 (1997)
98. Sahebi, S., Oroumchian, F., Khosravi, R.: An enhanced similarity measure for utilizing site structure in web personalization systems. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 03, WI-IAT 2008, pp. 82–85. IEEE Computer Society, Washington, DC (2008)
99. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)
100. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM Conference on Electronic Commerce, EC 2000, pp. 158–167. ACM, New York (2000)
101. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, WWW 2001, pp. 285–295. ACM, New York (2001)
102. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Itembased collaborative filtering recommendation algorithms. In: Proc. 10th International Conference on the World Wide Web, pp. 285–295 (2001)
103. Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.T.: Application of dimensionality reduction in recommender system – a case study. In: ACM Webkdd Workshop (2000)
104. Ben Schafer, J., Konstan, J., Riedi, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic commerce, EC 1999, pp. 158–166. ACM, New York (1999)
105. Senot, C., Kostadinov, D., Bouzid, M., Picault, J., Aghasaryan, A.: Evaluation of group profiling strategies. In: IJCAI, pp. 2728–2733 (2011)
106. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Recommender Systems Handbook, pp. 257–297 (2011)
107. Shardanand, U., Maes, P.: Social information filtering: algorithms for automating word of mouth. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York (1995)
108. Sinha, R.R., Swearingen, K.: Comparing Recommendations Made by Online Systems and Friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001)
109. Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., Chen, Z.: Cubesvd: a novel approach to personalized web search. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 382–390. ACM, New York (2005)
110. Takeuchi, Y., Sugimoto, M.: CityVoyager: An Outdoor Recommendation System Based on User Location History. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, pp. 625–636. Springer, Heidelberg (2006)

111. Tkalcic, M., Kunaver, M., Kosir, A., Tasic, J.: Addressing the new user problem with a personality based user similarity measure. In: Masthoff, J., Grasso, F., Ham, J. (eds.) UMMS 2011: Workshop on User Models for Motivational Systems: The Affective and the Rational Routes to Persuasion (2011)
112. Tso-Sutter, K.H.L., Marinho, L.B., Schmidt-Thieme, L.: Tag-aware recommender systems by fusion of collaborative filtering algorithms. In: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC 2008, pp. 1995–1999. ACM, New York (2008)
113. Vargas, S., Castells, P.: Rank and relevance in novelty and diversity metrics for recommender systems. In: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys 2011, pp. 109–116. ACM, New York (2011)
114. Velasquez, J.D., Bassi, A., Yasuda, H., Aoki, T.: Mining web data to create online navigation recommendations. In: Perner, P. (ed.) ICDM 2004. LNCS (LNAI), vol. 3275, pp. 551–554. Springer, Heidelberg (2004)
115. Velásquez, J.D., Palade, V.: Adaptive Web Sites: A Knowledge Extraction from Web Data Approach. IOS Press, Amsterdam (2008)
116. Velsquez, J.D., Palade, V.: Building a knowledge base for implementing a web-based computerized recommendation system. International Journal on Artificial Intelligence Tools 16(05), 793 (2007)
117. Victor, P., De Cock, M., Cornelis, C.: Trust and recommendations. In: Recommender Systems Handbook, pp. 645–675 (2011)
118. Walter, F.E., Battiston, S., Schweitzer, F.: A model of a trust-based recommendation system on a social network. Autonomous Agents and Multi-Agent Systems 16(1), 57–74 (2008)
119. Xu, G., Zhang, Y., Yi, X.: Modelling user behaviour for web recommendation using lda model. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT 2008, vol. 3, pp. 529–532 (December 2008)
120. Yanbe, Y., Jatowt, A., Nakamura, S., Tanaka, K.: Can social bookmarking enhance search in the web? In: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL 2007, pp. 107–116. ACM, New York (2007)
121. Yang, W.-S., Cheng, H.-C., Dia, J.-B.: A location-aware recommender system for mobile shopping environments. Expert Systems with Applications 34(1), 437–445 (2008)
122. Yilmaz, E., Aslam, J.A., Robertson, S.: A new rank correlation coefficient for information retrieval. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information retrieval, SIGIR 2008, pp. 587–594. ACM, New York (2008)
123. Yu, P.S.: Data mining and personalization technologies. In: Proceedings of the Sixth International Conference on Database Systems for Advanced Applications, DASFAA 1999, pp. 6–13. IEEE Computer Society, Washington, DC (1999)
124. Zhang, Y.C., Séaghdha, D.Ó., Quercia, D., Jambor, T.: Auralist: introducing serendipity into music recommendation. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM 2012, pp. 13–22. ACM, New York (2012)
125. Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J.R., Zhang, Y.-C.: Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences 107(10), 4511–4515 (2010)
126. Zhu, T., Greiner, R., HŁubl, G.: An effective complete-web recommender system (2003)
127. Ziegler, C.-N., Golbeck, J.: Investigating interactions of trust and interest similarity. Decis. Support Syst. 43, 460–475 (2007)
128. Ziegler, C.-N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the 14th International Conference on World Wide Web, WWW 2005, pp. 22–32. ACM, New York (2005)