

# Recommender Systems: Techniques, Applications, and Challenges



Francesco Ricci, Lior Rokach, and Bracha Shapira

## 1 Introduction

Recommender systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user [21, 56, 58]. The suggestions usually relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read.

“Item” is the general term used to denote what the system recommends to users. An RS normally focuses on a specific type of item (e.g., movies or news articles) and accordingly, its design, its graphical user interface, and the core recommendation technique used to generate the recommendations are all customized to provide useful and effective suggestions for that specific type of item.

RSs are primarily directed at individuals who lack sufficient personal experience or competence to evaluate the potentially overwhelming number of items that a website may offer [58]. A prime example is an e-commerce recommender engine that assists users in selecting items to purchase. On the popular website, Amazon.com, the site employs an RS to personalize the online store for each customer [42, 70]. Since recommendations are usually personalized, different users or user groups benefit from diverse, tailored suggestions. However, there are also non-personalized recommendations, which are much simpler to generate. Typical examples include the top 10 editor selections of books or movies. While they may be useful and effective in certain situations, for example, when not enough information

---

F. Ricci (✉)

Faculty of Computer Science, Free University of Bozen-Bolzano, Bozen-Bolzano, Italy  
e-mail: [fricci@unibz.it](mailto:fricci@unibz.it)

L. Rokach · B. Shapira

Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beersheba, Israel  
e-mail: [liorrk@bgu.ac.il](mailto:liorrk@bgu.ac.il); [bshapira@bgu.ac.il](mailto:bshapira@bgu.ac.il)

© Springer Science+Business Media, LLC, part of Springer Nature 2022

F. Ricci et al. (eds.), *Recommender Systems Handbook*,

[https://doi.org/10.1007/978-1-0716-2197-4\\_1](https://doi.org/10.1007/978-1-0716-2197-4_1)

about the target user's preferences or interests is available, these types of non-personalized recommendations have not been the primary focus of the RS research.

In their simplest form, personalized recommendations are provided as ranked lists of items. In performing this ranking, RSs try to predict the most suitable products or services, based on the user's preferences and constraints. In order to complete such a computational task, RSs collect information from users regarding their preferences, which are either explicitly expressed, for example, as ratings for products, or inferred by interpreting the actions performed by the user while interacting with the system. For instance, an RS may consider the purchase of a particular item as an implicit sign of the user's preference for the item.

The development of RSs initiated from a rather simple observation: individuals often rely on recommendations provided by others in making routine, daily decisions [56, 68]. For example, it is common to turn to one's peers for recommendations when selecting a book to read; employers rely on recommendation letters in their recruitment decisions; and when selecting a movie to watch, individuals tend to read and rely on the movie reviews written by film critics or ask their friends for recommendations.

In seeking to mimic this behavior, the first RSs applied algorithms in order to leverage recommendations produced by a community of users and deliver these recommendations to an "active" or "target" user looking for suggestions. The recommendations were for items that similar users, or those with similar tastes, liked. This approach is termed collaborative filtering [57], and it is based on the rationale that if the active user agreed with certain users in the past, then other recommendations coming from these similar users should be relevant and of interest to the active user.

As e-commerce websites began to develop, there was a pressing need to provide recommendations derived from filtering the entire range of available alternatives. Users found it difficult to choose between the extremely wide variety of items (products and services) that these websites offered.

The explosive growth and variety of information and e-commerce related services available on the Web (selling, product comparison, auctions, etc.) frequently overwhelm users, leading them to make poor decisions. The large availability of choices, instead of being beneficial, may affect users' wellbeing. It came to be understood that while choice is good, more choice is not always better. Indeed, choice, with its implications of freedom, autonomy, and self-determination, may become burdensome and ultimately cause freedom to be regarded/perceived as a kind of misery-inducing tyranny [66].

In recent years, RSs have proven to be valuable means of coping with the information overload problem. An RS addresses this phenomenon by pointing a user toward new, not-yet-experienced items, or toward items that the user may want to reconsume, which are relevant to the user's current task or context. Upon a user's request, which can be articulated, depending on the recommendation approach, by the user's context and need, RSs generate recommendations using various types of knowledge and data about users, the available items, and previous transactions stored in customized databases. The user can then browse the recommendations,

choose whether or not to accept them, and may provide (immediately or at a later stage) implicit or explicit feedback. This user action and feedback can be stored in the RS database for use in generating new recommendations in future user-system interactions.

The study of recommender systems is relatively new compared to research on other classical information system tools and techniques (e.g., databases or search engines). Recommender systems emerged as an independent research area in the mid-1990s [10, 32, 56, 68]. In recent years, interest in recommender systems has dramatically increased, as shown in the examples below:

1. Recommender systems play an important role in widely used Internet sites such as Amazon.com, YouTube, Netflix, Spotify, LinkedIn, Facebook, Tripadvisor, Last.fm, and IMDb. Moreover many media companies are now developing and deploying RSs as part of the services they provide to their subscribers. Like Netflix, that provided real data, and awarded a million dollar prize to the team that substantially improved its recommender system's performance, other companies have shared data for RS-related competitions [41, 48].
2. There are conferences and workshops dedicated specifically to the field, namely the Association of Computing Machinery's (ACM) Conference Series on Recommender Systems (RecSys), established in 2007. This conference is the premier annual event on RSs technology research and applications. In addition, sessions dedicated to RSs are frequently included in more traditional conferences in the area of databases, information systems, user modeling, artificial intelligence, machine learning, data science, and adaptive systems. Additional noteworthy conferences within this scope include: ACM's Special Interest Group on Information Retrieval (SIGIR); User Modeling, Adaptation and Personalization (UMAP); Intelligent User Interfaces (IUI); World Wide Web (WWW); ACM's Special Interest Group on Management of Data (SIGMOD); and the Knowledge and Data Discovery (KDD) conference which is a main venue for data science related research.
3. At institutions of higher education around the world, undergraduate and graduate courses are now dedicated entirely to RSs, and tutorials on RSs are very popular at computer science conferences. A few books on different aspects of recommender systems have been published; some introduced the origin of RSs and RS techniques [38, 65], others cover practical aspects of implementing RSs [27, 33], and there is even a textbook dedicated to the subject [4]. Springer is publishing several books on specific recommender system topics in its series: Springer Briefs in Electrical and Computer Engineering.
4. There have been several special issues of academic journals which cover research and developments in the RS field. Among the journals that have dedicated issues to RSs are: ACM Transactions on Intelligent Systems and Technology (2015); IEEE Intelligent Systems (2019); Electronic Markets (2019); Cognitive Computation (2020); and User Modeling and User Adapted Interaction (2020).

In this introductory chapter, we briefly discuss basic RS ideas and concepts. Our main goal is not to present a self-contained comprehensive survey on RSs but rather

to provide a structured overview of the chapters in this handbook and to help the reader navigate the rich and detailed content it contains. The interested reader can also consult other available resources on recommender systems [17, 40, 44, 54, 61]. At the end of this chapter, we identify some research challenges that we believe are particularly important and merit further research attention.

The handbook is divided into five sections: (1) general recommendation techniques; (2) special recommendation techniques; (3) value and impact of recommender systems; (4) human-computer interaction; and (5) applications.

The first section presents the fundamental techniques most commonly used today for building RSs, such as collaborative filtering techniques; semantic and content-based methods, context-aware techniques, and deep neural networks based approaches.

The second section surveys special machine learning techniques and approaches aimed at addressing unique RS challenges and domains, such as session-based RSs, group RSs, people-to-people RSs, cross-domain RSs, adversarial ML in RSs, and the use of NLP in RSs.

The third section discusses the specific value of recommender systems for distinguished actors and the role they play for these actors, who are referred to as stakeholders. This section introduces multi-stakeholder RSs and presents methods and metrics to assess the RS's value for different actors from various perspectives. In addition to reviewing classic evaluation metrics, this section discusses more recently considered qualities, such as diversity, novelty, bias and fairness.

The fourth section is dedicated to human-computer interaction aspects of recommender systems and discusses decision-making aware design and approaches to assess and enhance RS explainability as a means of improving users' trust and perception of recommendations. This section also discusses user's personality acquisition and utilization aimed at improving users' RS experience.

Finally, the fifth section presents various recommender system applications and unique use cases. This section contains a list of chapters covering a broad spectrum of recommendation techniques used to recommend food, multimedia, as well as fashion and social RS techniques.

## 2 Recommender Systems' Value

We have defined RSs as software tools and techniques that provide users with suggestions for items that a user may wish to utilize. Now we refine this definition to illustrate a range of possible roles that an RS can play. In fact, in the last years it emerged that RSs must be considered as a platform playing different roles for distinguished actors. The end user of an RS is an important actor but not the only one. In Chapter "Multistakeholder Recommender Systems" is first introduced the notion of a recommendation stakeholder, which is any group or individual that can affect, or is affected by, the delivery of recommendations to users. Three important stakeholders are identified: Consumers (aka users), Providers (aka suppliers) and the

System owners. Consumers are the end users who receive item recommendations. Providers are those actors that supply the recommended items (e.g., a book publisher in Amazon). Finally, the organization itself, which has created the platform and the associated RS in order to match consumers with items is an important actor. For instance, a travel recommender system is typically offered by a travel intermediary, such as, Booking.com or AirBnB, or a destination management organization, such as Visitfinland.com. The first type of actors primarily aims at selling more (hotel rooms and other travel services), while the second type is interested in increasing the number of tourists visiting the various regions of a destination, in a fair and sustainable way [18, 60]. While, the user's primary motivations for accessing such a system would be to find a suitable hotel, interesting events or attractions when visiting a destination or planning the travel to a destination. Actually, the providers and the system owner, can use the RS to influence the choices of the end users and finally determining the success of the platform and the profit of the suppliers.

Chapter "Value and Impact of Recommender Systems" focuses on various ways RSs create value for different stakeholders and also discusses the possible risks and the potentially negative impacts of using such systems. The chapter deals with the organizational and business-oriented perspective by reporting how practical systems are evaluated and which effects have been observed in real-world deployments. Here, we want to give some of the basic reasons as to why both item providers and recommender system owners may want to exploit RSs.

- *Increase the number of items sold.* This is probably the most important function for a commercial RS, i.e., to be able to sell an additional set of items compared to those usually sold without any kind of recommendation. This goal is achieved because the recommended items are likely to suit the user's needs and wants. Presumably the user will recognize this after having tried several recommendations. Non-commercial systems have similar goals, even if often there is no cost for the user that is associated with selecting an item. For instance, a public content network such as BBC (UK) or RAI (IT) aims at increasing the number of news items read on its site. In general, we can say that from the service provider's point of view, the primary goal for introducing an RS is to increase the conversion rate, i.e., the number of users that accept the recommendation and consume an item, compared to the number of simple visitors that just browse through the information.
- *Sell more diverse items.* Another major function of an RS is to enable the user to select items that might be hard to find without a precise recommendation, i.e., to not select only the "blockbusters". For instance, in a tourist RS the service provider is interested in promoting all the places of interest in a tourist area, not just the most popular ones. This is a major objective for a tourism RS maintained by a public destination management organisation. Similar goals are posed to public content network managers. But, attaining this goal could be difficult without an RS, since the system provider at the same time cannot afford the risk of advertising places (or media) that are not likely to suit a particular user's taste.

- *Increase the user satisfaction.* A well designed RS can also improve the experience of the user with the site or the application. The user will find the recommendations interesting, relevant and, with a properly designed human-computer interaction, he or she will also enjoy using the system. The combination of effective, accurate recommendations and a usable interface will increase the user's subjective evaluation of the system. This, in turn, will increase system usage and the likelihood that the recommendations will be accepted.
- *Increase user fidelity.* A user should be loyal to a website which, when visited, recognizes the old customer and treats her as a valued visitor. This is a standard feature of an RS since many RSs compute recommendations, thus leveraging the information acquired from the user during previous interactions such as the user's ratings of items or user's visited pages. Consequently, the longer the user interacts with the site, the more refined the user's model becomes: the system's representation of the user's preferences develops and the effectiveness of the RS output to customize and match to the user's preferences is increased.
- *Better understanding of what the user wants.* Another important function of an RS, which can also be leveraged by other applications of the RS owner, is the description of the user's preferences, which are collected either explicitly or predicted by the system. The system owner, but also the item providers, may decide to reuse this knowledge for a number of other goals, such as improving the management of the item's stock or production. For instance, in the travel domain, destination management organizations can decide to advertise a specific region to new customer sectors or advertise a particular type of promotional message derived by analyzing the data collected by the RS (transactions of the users).

We mentioned above some important motivations as to why the system owner introduces an RS and the item providers offer their services/items through the RS platform. But naturally users will use an RS if it will effectively support their tasks or goals. Consequently an RS must balance the needs of these three stakeholder and offer a service that is valuable to all of them. Hence advanced evaluation techniques must be introduced in order to properly assess the value and the impact of an RS across multiple groups of stakeholders, as it is discussed in Chapters "Multistakeholder Recommender Systems" and "Value and Impact of Recommender Systems".

Without fully describing the range of dimensions that drive the adoption of RSs, which is discussed in Chapter "Value and Impact of Recommender Systems", we want now to focus on the end user side and refer to Herlocker et al. [36], a paper that has become a classical reference in this field. The authors define eleven popular user's tasks that an RS can assist in implementing. Some may be considered as the main or core tasks that are normally associated with an RS, such as offering suggestions for items that may be useful to a user. Others might be considered as more "opportunistic" ways to exploit an RS. As a matter of fact, this task differentiation is similar to what happens with other digital "tools", e.g., a search engine. Search engines primary function is to locate documents that are relevant to the user's information need, but it can also be used to check the importance of

a webpage (looking at the position of the page in the result list of a query) or to discover the various usages of a word in a collection of documents.

- *Find Some Good Items*: Recommend to a user some items as a ranked list along with predictions of how much the user would like them (e.g., on a scale of one-to-five stars). This is the main recommendation task that many commercial systems address (see, for instance, Chapters “Music Recommendation Systems: Techniques, Use Cases, and Challenges” and “Multimedia Recommender Systems: Algorithms and Challenges”). We note that some systems do not show the predicted rating, especially when users may not be interested in understanding why the item is recommended (e.g., in music).
- *Find all good items*: Recommend all the items that can satisfy the user needs, in such cases it is insufficient to just find some of them. This is especially true when the number of items is relatively small or when the RS is mission-critical, such as in medical or financial applications. In these situations, in addition to the benefit derived from carefully examining all the possibilities, the user may also benefit from a motivated ranking of these items or from additional explanations that the RS generates (see Chapter “Beyond Explaining Single Item Recommendations”).
- *Annotation in context*: Given an existing context, e.g., a list of items, emphasize some of them depending on the user’s long-term preferences. For example, a TV recommender system might annotate which TV shows displayed in the electronic program guide (EPG) are worth watching (Chapter “Social Recommender Systems”, provides interesting examples of this task).
- *Recommend a sequence*: Instead of focusing on the generation of a single recommendation, the idea is to recommend a sequence of items, one by one or as a whole. Typical examples include recommending, a next point of interest to visit, a TV series, a book on RSs after having recommended a book on data mining, or a compilation of musical tracks (see Chapters “Music Recommendation Systems: Techniques, Use Cases, and Challenges”, and “Session-Based Recommender Systems”, for more examples and discussion).
- *Recommend a bundle*: Suggest a group of items that fits well together. For instance, a travel plan may be composed of various attractions, destinations, and accommodation services that are located in a delimited area. From the point of view of the user, these various alternatives can be considered and selected as a single travel destination [45].
- *Just browsing*: In this task, the user browses the catalog without any imminent intention of purchasing an item. The task of the RS is to help the user to browse the items that are more likely to fall within the scope of the user’s interests for that specific browsing session. This is a task that has also been supported by adaptive hypermedia techniques [20].
- *Find credible recommender*: Some users do not trust recommender systems, thus they play with them to see how good they are at making recommendations. Hence, a certain system may also offer specific functions to let the users test its behavior in addition to those just required for obtaining recommendations.

- *Improve the profile*: This relates to the capability of the user to provide (input) information to the recommender system about what he or she likes and dislikes. This is a fundamental task that is strictly necessary to provide personalized recommendations. If the system has no specific knowledge about the active user, then it can only provide the same recommendations that would be delivered to an “average” user (see Chapter “People-to-People Reciprocal Recommenders” for an application domain where preference elicitation is critical).
- *Express self*: Some users may not care about the recommendations at all. Rather, what is important to them is that they be allowed to contribute with their ratings and express their opinions and beliefs. The user satisfaction for that activity can still act as leverage, resulting in the user’s continued loyalty to the application (as we mentioned prior, in discussing the service provider’s motivations).
- *Help others*: Some users are happy to contribute with information, e.g., their evaluation of items (ratings), because they believe that the community benefits from their contribution. This could be a major motivation for entering information into a recommender system that is not used routinely. For instance, with an automobile RS, a user who has already purchased a new car is aware that the rating entered in the system is more likely to be useful to other users rather than to oneself, the next time a new-car-purchase is contemplated.
- *Influence others*: In Web-based RSs, there are users whose main goal is to explicitly influence other users into purchasing particular products. As a matter of fact, there are also malicious users that may use the system simply to promote or penalize certain items.

As these various points indicate, the role of an RS within an information system can be quite diverse. This diversity calls for the exploitation of a range of different knowledge sources and techniques. In the next two sections, we discuss the data that an RS manages and the core technique used to identify the right recommendations.

### 3 Data and Knowledge Sources

RSs are information processing systems that actively gather various kinds of data in order to build their recommendations. The data used within an RS can relate to three kinds of objects: items, users, and interactions between the users and the items. Some systems collect users’ features, such as their areas of interest, age, or gender, while other gather items’ characteristics, such as the genre of a movie or the price of an item. All systems, in order to learn users’ preferences and generate recommendations, gather users’ interactions with items. In practice, the data and knowledge sources available for an RS can be very diverse, and their effective exploitation depends on the recommendation technique and the available computational resources (see Sect. 4).

**Items** Items are the objects that are recommended. Available items data depends on the complexity of acquiring them, for instance item features, which also



depends on the availability of meta-data. For example, in a multimedia RS, items such as images, require special image analysis algorithms to extract their features from raw content. Text-based items, such as news, or items that have long descriptions, s.a., reviews, require special algorithms to analyze the text and extract meaningful relevant features from the text. Chapter “Natural Language Processing for Recommender Systems” describes NLP methods that can be used to extract meaningful information from textual descriptions of items (and users). For example, by using such methods, features can be extracted from experts reviews [55], and they may alleviate the cold-start scenario, i.e., when dealing with items that have not gained yet enough interactions. Chapter “Semantics and Content-Based Recommendations” presents methods for the representation of items (and users) adopting semantic-aware representations of textual content. While early models were not able to extract from the textual content a complete description of the item, nor to encode semantic relationships between terms, semantic aware systems enable to give meaning to information expressed in natural language and obtain a deeper comprehension of the information conveyed by textual content, hence leading to better RSs. Chapter “Semantics and Content-Based Recommendations” reviews the main two approaches for semantic representation: endogenous, i.e., relying on the distribution of keywords in documents, and exogenous, which resolves semantic synonymy, polysemy and other language challenges, by utilizing external knowledge, like taxonomies ontologies or other language related information.

**Users** Users of an RS, as we mentioned above, may have very diverse goals and characteristics. In order to personalize the recommendations and the human-computer interaction, RSs exploit a range of information about the users. This information can be structured in various ways. User data is said to constitute the user model [16, 30]. The user model profiles the user, i.e., encodes her preferences and needs. Various user modeling approaches have been used and, in a certain sense, an RS can be viewed as a tool that generates recommendations by building and exploiting user models [13, 14]. The selection of what information to model depends on the recommendation technique, the availability of data and the efforts required to gather and extract the knowledge.

For instance, in collaborative filtering, a user is modeled as a simple list of his ratings for certain items. In context aware RS (see Chapter “Context-Aware Recommender Systems: From Foundations to Recent Developments”) the user model incorporates the contextual information to be utilized during the recommendation process in order to recommend items to users under specific contextual situations. For example, by using the temporal context, a travel recommender system would include contextual features describing the weather, and the time of year when the vacation was consumed, so that a vacation recommendation in winter can be very different from the one recommended in summer. In content-based RSs (see Chapter “Semantics and Content-Based Recommendations”) users could be modeled by the content features of the items they consumed, and, similarly to the items, such features may derive from of a variety of tabular and unstructured content.

For example, textual user generated reviews about consumed items may be utilized to enrich the user model [22, 55].

**Interactions** RSs record the interactions between their users and items in log-like data that stores important information generated during the interactions. This information is useful for the recommendation generation algorithm that the system is using. For instance, a log may contain a reference to the item purchased by the user, the events that led to the buying event (e.g., browsing, clicking, inclusion in the basket, etc.), the rating that the user assigned to the item, or any other information that the system collects about the time, the location of the interaction. An important part of the information about the interactions is the user feedback to the item. User feedback can be explicitly provided by users or implicitly inferred from the users' behaviour.

Explicit feedback is known to be more reliable than implicit one and provides a level of preference of the user for an item (e.g., ratings 1–5). However, explicit feedback is often not available, or very sparse, since many users would not bother to provide it. Ratings are the most popular form of explicit feedback data that RSs collect [59]. According to [63], ratings can take on a variety of forms:

- Numerical ratings such as the 1–5 stars used in the RS offered by Amazon.com.
- Ordinal ratings, such as “strongly agree, agree, neutral, disagree, strongly disagree” where the user is asked to select the term that best indicates his or her opinion regarding an item (usually via questionnaire).
- Binary ratings that model choices in which the user is simply asked to decide if a certain item is good or bad.

Implicit feedback relates to the inference of the user's preference for an item from the observed interaction of the user with the item, for instance during a recommendation session (see Chapter “Session-Based Recommender Systems”). Implicit feedback is typically available for many more items than those available for explicit feedback. In fact, various types of user-item interactions (clicking, buying, adding to the basket, and many more) provide some implicit indication of the user's preference. Such feedback can be considered as unary ratings, and does not indicate a specific level of preference. The interactions are typically interpreted as positive implicit preferences. However, the absence of an interaction cannot be simply considered as a sort of negative feedback. Usually, there is not enough data to precisely derive the reasons for a missing interaction: the user may dislike the item, or may have not seen it at all, or may have consumed it already in the past. Chapter “Item Recommendation from Implicit Feedback” describes the challenges of building RSs from implicit feedback, and provides methods to address them. First, since implicit feedback refers to items that the users consumed in the past, the data is noisy and gives a weak signal of positive preferences. Most of the items do not have interactions and considering all these absence of interactions as a weak negative signals of preference leads to high training costs. That makes standard training algorithms hard to apply.

An important data characteristic dimension relates to the time span of the data. A user may have a long-term or a short-term profile depending of his or her interaction with the system. If the user is registered and known to the system, the system can indeed learn user's preferences from his past interactions. However, for scenarios where the user is a guest of the system, not registered, or would like to stay anonymous, the system can learn the user's preferences only from the current session. In such session-based scenarios, the RS aims to tailor item suggestions to the short-term needs and the predicted intents of the user that are learned from the interactions during the session. This is termed session-based recommendation. In addition, it can happen that the user is known to the system but may change her intents from session to session, depending on her specific task. For instance, in one session a user could like to buy a present for an elderly person (parent), and in a different session the same user may buy items for personal use. Chapter "Session-Based Recommender Systems" elaborates on practical application scenarios for session-based recommender systems, defines the problem, outlines the challenges, and finally describes approaches to address these challenges.

In addition, a user model in context-based RSs, as described in Chapter "Context-Aware Recommender Systems: From Foundations to Recent Developments", is dynamic and changes based on contextual features such as temporal, location, mood or any other relevant features. Usually in context-aware RSs the users are known to the system and have user models that should be altered according to the contextual information to address the short-term preferences emerging in a specific context.

We should stress that utilizing the right data is fundamental to the performance of an RS. As described above, a variety of data and knowledge sources can be leveraged in various RSs techniques. The decision about the data to use for a system, and how to use it should be done carefully while considering availability of data, the recommendation algorithm, the required effort, and the available resources.

## 4 Recommendation Techniques

To implement its core function, identifying useful items for the user, an RS must *predict* that an item is worth recommending. To achieve this goal, the system must be able to predict the utility of some items, or at least compare the utility of some items, and then decide which items to recommend based on this comparison. The prediction step may not be explicit in the recommendation algorithm, but we can still apply this unifying model to describe the general role of an RS. Here, our goal is to provide the reader with a unifying perspective rather than an account of all the different recommendation approaches that are illustrated in this handbook.

To illustrate the prediction step of an RS, consider, for instance, a simple and non-personalized recommendation algorithm that recommends only the most popular songs. The rationale for using this approach is that in the absence of more precise information about the user's preferences, a popular song, i.e., one that is liked (high utility) by many users, will also most likely appeal to a generic user, or at least with

a higher likelihood than another randomly selected song. Hence, the utility of such popular songs is predicted to be reasonably high for this generic user.

This view of the core recommendation computation as the prediction of the “utility” of an item for a user has been suggested in [3] and updated in [61] by referring to a more generic concept of “evaluation” of the user for an item, rather than utility. Both papers model this degree of utility (evaluation) of the user  $u$  for the item  $i$  as a (real-valued) function  $R(u, i)$ , as it is usually done in collaborative filtering by considering users’ ratings for items. Then the fundamental task of an RS is to predict the value of  $R$  over pairs of users and items, or in other words, to compute  $\hat{R}(u, i)$ , where we denote with  $\hat{R}$  the estimation, computed by the RS, for the true function  $R(u, i)$ . Consequently, having computed this prediction for the active user  $u$  on a set of items, i.e.,  $\hat{R}(u, i_1), \dots, \hat{R}(u, i_N)$ , the system will recommend the items  $i_{j_1}, \dots, i_{j_K}$  ( $K \leq N$ ) with the largest predicted utility.  $K$  is typically a small number, that is, much smaller than the cardinality of the items set or the items on which a user utility prediction can be computed, i.e., RSs “filter” the items that are recommended to users.

As mentioned above, some RSs do not fully estimate the utility before making a recommendation. Instead, they may apply some heuristics to hypothesize that an item may be of use to a user. This is typical, for instance, in knowledge-based systems. These systems use various kinds of knowledge about users, items, and the utility function itself. However, these approaches are much less prevalent in real-world scenarios. Thus, we will not cover them in this handbook edition (the reader is welcome to read the corresponding chapters in the second edition of this handbook).

It is also important to note that sometimes the user utility for an item depends on other variables, which we generically call “contextual”. For instance, the utility of an item for a user can be influenced by the time and user location when the recommendation is requested. For instance, users may be more interested in items (e.g., restaurant) closer to their current location. Consequently, the recommendations must be adapted to these specific additional conditions and as a result, it becomes increasingly more difficult to correctly estimate what the right recommendations are.

This handbook presents several different types of recommender systems that vary in terms of the addressed domains and especially regarding the recommendation algorithm, i.e., how the prediction of the utility of a recommendation is made. Other differences relate to how the recommendations are finally assembled and presented to the user in response to user requests. These aspects are discussed as well, later in this introduction.

To provide an initial overview of the different types of RSs, we want to quote a taxonomy provided by [21] that has become a classical way of distinguishing between recommender systems and referring to them. Burke [21] distinguishes between six different classes of recommendation approaches:

**Content-Based** The system learns to recommend items that are similar to the ones that the user liked in the past. The similarity of items is calculated based on the

features associated with the compared items. For example, if a user has positively rated a movie that belongs to the comedy genre, then the system can learn to recommend other movies from this genre [43].

Classic content-based recommendation techniques aim at matching the attributes of the user profile against the attributes of the items. In most cases, the items' attributes are simply keywords that are extracted from the items' descriptions. Semantic indexing techniques represent the item and user profiles using concepts instead of keywords. Chapter "Semantics and Content-Based Recommendations" presents a comprehensive survey of semantic indexing techniques to overcome the main problems of classical keyword-based systems. The authors present two main groups of semantic indexing techniques: exogenous and endogenous. Techniques in the former group rely on the integration of external knowledge sources, such as: ontologies, encyclopedic knowledge (such as Wikipedia) and data from the Linked Data cloud, while techniques in the latter group rely on a lightweight semantic representation based on the hypothesis that the meaning of words depends on their usage in large corpora of textual documents. Chapter "Semantics and Content-Based Recommendations" demonstrates how to utilize semantic approaches to realize a new generation of semantic content-based recommender systems by describing their main potentials and limitations.

**Collaborative Filtering** The original and most simple implementation of this approach [32] makes recommendations to the active user based on items that other users with similar tastes liked in the past. The similarity in the taste of two users is calculated based on the similarity in the rating history of the users. This is the reason why [64] refers to collaborative filtering as "people-to-people correlation." Collaborative filtering is considered to be the most popular and widely implemented technique in RS. Chapter "Trust Your Neighbors: A Comprehensive Survey of Neighborhood-Based Methods for Recommender Systems" presents a comprehensive survey of neighborhood-based methods for collaborative filtering. Neighborhood-based methods focus on relationships between items or, alternatively, between users. An item-based approach models the preference of a user to an item based on ratings of similar items by the same user. Neighborhood-based methods have obtained considerable popularity due to their simplicity, efficiency, and ability to produce accurate and personalized recommendations. Chapter "Trust Your Neighbors: A Comprehensive Survey of Neighborhood-Based Methods for Recommender Systems" describes the main benefits of such methods, as well as their principal characteristics. Moreover, the chapter addresses the essential decisions that are required while implementing a neighborhood-based recommender system and gives practical information on how to make such decisions. Perhaps the decision that has the greatest impact on the rating prediction and computational performance of the recommender system is the choice between a user-based and an item-based method. In typical commercial recommender systems where the number of users exceeds the number of available items, item-based approaches should be preferred since they provide more accurate recommendations, while being more computationally efficient and requiring less frequent updates. On the other hand,

user-based methods usually provide more original recommendations, which may lead users to a more satisfying experience [26].

Finally, the problems of sparsity and limited coverage, often observed in large commercial recommender systems, are discussed by exploring two research directions: dimensionality reduction and graph-based techniques. Dimensionality reduction provides a compact representation of users and items that captures their most significant features. An advantage of such an approach is that it allows to obtain meaningful relations between pairs of users or items, even though these users have rated different items, or these items were rated by different users. On the other hand, graph-based techniques exploit the transitive relations in the data. These techniques also avoid the problems of sparsity and limited coverage by evaluating the relationship between users or items that are not directly connected. However, unlike dimensionality reduction, graph-based methods also preserve some of the “local” relations in data.

Chapter “Advances in Collaborative Filtering” presents several techniques available for building Collaborative Filtering (CF) RSs. Specifically, the authors discuss latent factor models, such as matrix factorization (e.g., Singular Value Decomposition, SVD). These methods map both items and users to the same latent factor space. The latent space is then used to explain ratings by characterizing both products and users in term of factors automatically inferred from user feedback. The authors elucidate how SVD can handle additional features of the data, including implicit feedback and temporal information. They also describe techniques to address shortcomings of neighborhood techniques by suggesting more rigorous formulations using global optimization techniques. Utilizing such techniques makes it possible to lift the limit on neighborhood size and to address implicit feedback and temporal dynamics. The resulting accuracy is close to that of matrix factorization models, while offering a number of practical advantages.

Particular attention should be given to the scenario in which the users give no explicit feedback. In Chapter “Item Recommendation from Implicit Feedback” the authors provide a comprehensive review of the core challenges in learning item recommendation from implicit feedback. The authors present various methods to address these challenges: sampling methods for general scoring functions based on pointwise, pairwise and softmax losses are described. Moreover, efficient training algorithms that can be applied to dot product models and square losses are discussed. Finally, they present sublinear time approaches for the retrieval task in which the most recommended items are quickly detected from the whole long item catalogue.

As deep learning gained popularity in various domains such as computer vision, more and more researchers started to incorporate deep learning methods in RS [73]. In particular, collaborative filtering tasks that were solved using SVD methods has been transformed into their corresponding deep learning solution. Since 2010, a variety of deep learning designs and methods have been developed for RS. Deep learning has many advantages in the context of RS. First, it can save time for the practitioners because deep learning requires less feature engineering work. Moreover, it can process unstructured raw data such as text, sound, image and video, which are very common in many RSs.

In Chapter “Deep Learning for Recommender Systems” the authors overview some key methods and highlight the impact of deep learning techniques in the recommender systems field. They present a range of challenging tasks in recommendation, such as cold-start problem, explainability, temporal dynamics, and robustness that can be addressed using deep neural networks. In the first part of the chapter, the authors describe the basics of deep learning techniques that are widely used in RS, including multilayered perceptrons, convolutional neural networks (CNN), recurrent neural networks (RNN), deep reinforcement learning, etc. Then the authors explain how users, items and their interactions can be modeled by a deep learning architecture. Finally, they describe various applications of deep learning in RS-related domains, such as e-commerce, online entertainment, news and point-of-interests.

**Community-Based** This type of systems recommends items based on the preferences of the users friends. This technique follows the epigram, “Tell me who your friends are, and I will tell you who you are” [7, 12]. Evidence suggests that people tend to rely more on recommendations from their friends than on recommendations from similar but anonymous individuals [69]. This observation, combined with the growing popularity of open social networks, is generating a rising interest in community-based systems or, as they are usually referred, social recommender systems [31] (see Chapters “Social Recommender Systems” and “People-to-People Reciprocal Recommenders”). This type of RS models and acquires information about the social relations of the users and the preferences of the users friends. The recommendation is based on ratings that were provided by the user’s friends. In fact these RSs are following the rise of social-networks and enable a simple and comprehensive acquisition of data related to the social relations of the users.

There are a few more types of RSs that will not be covered explicitly in this edition, but we shortly discuss them here for the sake of completeness:

**Demographic** This type of systems recommends items based on the demographic profile of the user [17]. The assumption is that different recommendations should be generated for different demographic niches. Many websites adopt simple and effective personalization solutions based on demographics. For example, users are dispatched to particular websites based on their language or country. Or, suggestions may be customized according to the age of the user. While these approaches have been quite popular in the marketing literature, there has been relatively little proper RS research on demographic systems.

**Knowledge-Based** Knowledge-based systems recommend items based on specific domain knowledge about how certain item features meet users’ needs and preferences and, ultimately, how the item is useful for the user. Notable knowledge-based recommender systems are case-based [19, 24, 28, 39, 62]. In these systems, a similarity function estimates how much the user’s needs (problem description) match the recommendations (solutions of the problem). Here, the similarity score can be directly interpreted as the utility of the recommendation for the user. Knowledge-based systems tend to work better than others at the beginning of their

deployment but if they are not equipped with learning components, they may be surpassed by other shallow methods that can exploit the logs of the human/computer interaction (as in CF).

**Hybrid Recommender Systems** These RSs are based on the combination of the above mentioned techniques. A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B. For instance, CF methods suffer from new-item problems, i.e., they cannot recommend items that have no ratings. This does not limit content-based approaches since the prediction for new items is based on their description (features) that are typically easily available. Given two (or more) basic RSs techniques, several ways have been proposed for combining them to create a new hybrid system (see [21] for the precise descriptions). Moreover, the emergence of deep learning techniques made it much easier to build hybrid RS [53].

As we have already mentioned, the context of the user when he or she is seeking a recommendation can be used to better personalize the output of the system. For example, in a temporal context, vacation recommendations in winter should be very different from those provided in summer [11]. Or a restaurant recommendation for a Saturday evening with one's friends should be different from that suggested for a workday lunch with co-workers [1]. Chapter "Context-Aware Recommender Systems: From Foundations to Recent Developments" reviews the topic of context-aware recommender systems (CARS). It presents the general notion of context and how it can be modeled in RSs. As it discusses the possibilities of combining several context-aware recommendation techniques into a single unified approach, the authors also provide a case study of one such combined approach.

Three popular different algorithmic paradigms for incorporating contextual information into the recommendation process are discussed: reduction-based (pre-filtering), contextual post filtering, and context modeling. In reduction-based (pre-filtering) methods, only the information that matches the current usage context, e.g., the ratings for items evaluated in the same context, are used to compute the recommendations. In contextual post filtering, the recommendation algorithm ignores the context information. The output of the algorithm is filtered/adjusted to include only the recommendations that are relevant in the target context. In the contextual modeling, the more sophisticated of the three approaches, context data is explicitly used in the prediction model.

## 5 Special Recommendation Techniques

It is clear from the previous sections that RS research is evolving in numerous and diverse directions, and special topics are emerging or becoming more important subjects of investigation. In this handbook, we cover some of these topics. Indeed, some have already been mentioned, such as: context-aware recommendations (Chapter "Context-Aware Recommender Systems: From Foundations to Recent



Developments”), which addresses the challenges and reviews methods to leverage contextual information into the recommendation process; and session-based recommendations (Chapter “Session-Based Recommender Systems”) that deals with challenges emerging from the scenario of short-term modeling of anonymous users during sessions. Other important topics are covered in this handbook and we will now briefly introduce them.

Chapter “Group Recommender Systems: Beyond Preference Aggregation” deals with situations in which the system should recommend information or items that are relevant to a group of users rather than to an individual. For instance, an RS may select television programs for a group to view or a sequence of songs to listen to, based on preference models of all the group members. Recommending to groups is clearly more complicated than recommending to individuals. Assuming that we know precisely what is good for individual users, the issue is how to combine the individual user models of the group members. In this chapter, the authors present usage scenario for group recommendation and discuss how to acquire information about individual users preferences in a group setting. The chapter includes a discussion of how the system knows who forms the group, i.e., who is present, and how to display and explain group recommendations. Moreover, the chapter includes a description of methods to help users reach a final decision, thus optimizing multiple criteria, while considering fairness (see also Chapter “Fairness in Recommender Systems”), and adopting methods from the multistakeholder RSs that are described in Chapter “Multistakeholder Recommender Systems”.

Chapter “Adversarial Recommender Systems: Attack, Defense, and Advances” discusses the vulnerability of RSs to adversarial attacks where an intelligent and adaptive attacker may deliberately manipulate data thereby violating the underlying stationary assumption of ML-based systems [15], that is, that the training data and test data are sampled from a similar (and possibly unknown) distribution. Adversarial attacks risk the security of RSs, and may compromise the integrity of the system, usually with the goal to harvest recommendation outcomes toward an illegitimate benefit, e.g., pushing some targeted items into the top-K list of users for market penetration. RSs should perhaps be designed with security-awareness so that to face adversarial attacks. The chapter is focused on machine-learned attacks based on the adversarial machine learning paradigm, rather than on traditional hand-crafted shilling attacks. The chapter introduces foundation concepts of adversarial machine learning (AML), and reviews the widely adopted attack and defense strategies focusing on adversarial attacks and defense against recommendation models, as well as on possible evaluation methods and metrics for evaluating them.

Chapter “People-to-People Reciprocal Recommenders” describes a special recommendation scenario where the RS recommends users to other users, such as employers to employees, or matching dating options to pairs of users. Thus, the system is required to satisfy the two parties involved in the recommendation (reciprocity) rather than the common one sided recommendation of items to an individual. The authors discuss the distinctive characteristics of reciprocal RSs, review previous works in various domains including social networks, mentor-mentee matching, recommending students to their colleagues in on-line courses

setting, job recommendation, and on-line dating. The authors also present a complete case study in on-line dating.

The idea of relating recommendations in different systems or domains by exploiting user data collected in one system or domain to produce recommendations in another, is at the base of the research on cross-domain recommender systems illustrated in Chapter “Design and Evaluation of Cross-Domain Recommender Systems”. In fact, recommender system applications are typically restricted to a particular type of item (movies, food, etc.), however, e-commerce sites like Amazon, eBay, and Alibaba regularly collect user feedback for items across multiple domains. Cross domain preferences of users might also be collected from their social media content or in parallel from several systems (one for each domain). It may, therefore, be beneficial to leverage all the available user data provided in various systems and domains. Recommendation quality can be improved, especially by mitigating the cold start and sparse-data challenges where knowledge acquired in a “source” domain can be used in the sparse-data target domain. Moreover, cross-domain RSs can enable cross-selling recommendations, where items from multiple domains are suggested together (e.g., a movie and a book on a recommended singer). In this chapter, the authors formally define the cross-domain recommendation problem, and try to provide a unifying perspective by merging ideas and approaches which arise in distinct disciplines. The chapter provides an analytical categorization of prior work, and identifies open issues for future research.

## 6 Recommender Systems Evaluation

Recommender systems research is being conducted with a strong emphasis on practice and commercial applications. One very important issue related to the practical side of RS deployment is the necessity of evaluating the quality and value of the system. Evaluation is required at different stages of the system’s life cycle and for various purposes [2, 36] (see Chapter “Evaluating Recommender Systems”).

At design time, evaluation is required to verify the selection of the appropriate approach for the specific system goals, of the various system’s stakeholders that have been identified. For instance, if a tourism RS is designed to improve the visitors’ knowledge of the destination and increase the visits’ count in the less known areas, then a technique characterised by high predictive precision is not in order, while a model that can manage well “new items”, i.e., items almost never consumed before is pivotal [46]. In the design phase, evaluation should be implemented off-line and the recommendation algorithms, i.e., their computed recommendations, are compared with the stored user interactions. An off-line evaluation consists of running several algorithms on the same datasets of user interactions and comparing their performances along a set of metrics measuring alternative and complementary dimensions, e.g., precision vs. diversity of the recommendations (see Chapter “Novelty and Diversity in Recommender Systems”). This type of evaluation is usually conducted on existing public benchmark data if

appropriate data is available, or, otherwise, on collected data. The design of the off-line experiments should follow known experiment design practices [9] in order to ensure reliable results.

Off-line experiments can give some indications on the quality of the chosen recommendation algorithm in fulfilling the system defined task and goals. However, such an evaluation cannot provide any insight about the user satisfaction, acceptance or experience with the system. That is, the algorithm might be very accurate in solving the core recommendation problem, i.e., identifying relevant items, but for some other reason the system may not be accepted by users, for example, because the performance of the system was not as expected. It is also worth noting that the RS may find the “correct” recommendations for a user, i.e., items that are matching the user needs and wants, but the user may not be able to immediately assess it, especially in those domains where the user may not be able to experience the recommended items at once. For instance, while the songs suggested by a music RS may immediately be played and evaluated, how much a book will make us feel is difficult to assess, before actually reading large part of the book itself.

Therefore, a user-centric evaluation is also required. It can be performed online after the system has been launched, or as a focused user study. During online evaluation, real users interact with the system without being aware of the full nature of the experiment running in the background. It is possible to run various versions of the algorithms on different groups of users for comparison and analysis of the system logs in order to enhance system performance. In addition, most of the RS algorithms include parameters, such as weight thresholds, the number of neighbors in collaborative filtering, etc., requiring constant adjustment and calibration.

Focused user studies are conducted when the online evaluation of the system in the real context of usage is not feasible or too risky. In this type of evaluation, a controlled experiment is planned where a small group of users are asked to perform different tasks with various versions of the system. It is then possible to analyze the user’s performance and to distribute questionnaires so that users may report on their experience. In such experiments, it is possible to collect both quantitative and qualitative information about the systems.

In recent years there has been an increased interest in user-centric evaluation procedures and metric for recommender systems. Researchers realized that recommender systems’ goals extend beyond the accuracy of the algorithms [40] as tools to provide a helpful and enjoyable, personalized experience that leads to user retention and satisfaction. This approach broadened the range of evaluated aspects of an RS to included aspects such as the form of preference elicitation, the presentation of the recommended results (e.g., one top item, top N items, or predicted ratings), and finally, the evaluation of the explanations provided to the users. Explanations may serve a few goals: the most popular is the justification of results, i.e., explaining to the user why the system decided to recommend a specific item. Other goals may include increasing trust in the system, persuading the user to purchase the recommended item, and helping a user with their decision making. When designing the evaluation of the recommendation explanation, it is therefore

important to identify the goal of the explanation and adjust a suitable metric for measuring it.

Chapter “Evaluating Recommender Systems” details the three previously mentioned types of experiments that can be conducted in order to evaluate recommender systems, namely, offline, online and user studies. It presents their advantages and disadvantages, and defines guidelines for choosing the methods for evaluating them by considering the properties that are to be evaluated. Unlike existing discussions of RS evaluation in the literature that usually focus mainly on the accuracy of an algorithm’s prediction [36] and related measures, this chapter is unique in its approach to the evaluation discussion since it focuses on property-directed evaluation. It provides a large set of properties (other than accuracy) that are relevant to the system’s success. For each of the properties, the appropriate type of experiment and relevant measures are suggested. Among the list of properties are: coverage, cold start, confidence, trust, novelty, risk, and serendipity. The chapter describes the difficulties and pitfalls of each of the properties and guidelines for the selection of the suitable evaluation type and properties for a given recommendation task and system.

As we have already discussed in Sect. 2, RSs are used by multiple stakeholders and for diverse and sometime competing goals. Chapter “Multistakeholder Recommender Systems” also discusses how system evaluation can take into account the perspectives of multiple stakeholders. A multistakeholder perspective on evaluation, in fact, focuses on additional aspects of system performance that may be quite important. For instance, a multi-sided platforms such as eBay, Etsy, Offer Up, or AirBnB, have a key business requirement of attracting and satisfying the needs of providers as well as users. Hence, a system of this type will also need to evaluate its performance from the provider perspective. The importance of such an evaluation should be evident. For instance, providers whose items are not recommended may experience poor engagement from users and lose interest in participating in the platform. Fewer providers on the platform could then lead to a decline in user interest in the platform as they may find it not comprehensive enough. Or, in an e-commerce platform, the profit of each recommended item may be a factor in ordering and presenting recommendation results. This marketing function of recommender systems was apparent from the start in commercial applications, but rarely included as an element of research systems.

Chapter “Value and Impact of Recommender Systems” completes the treatment of the topic of evaluating RSs by focusing on the capability of RSs to influence the users choices and their behavior. This is especially important in systems where the recommendations are central to the user experience of the service. For instance, in a system like YouTube or Netflix a large fraction of the observed interactions stems from recommendations. RSs can even increase this fraction with a self-enforcing cycle; if the content that users come across during browsing has already been pre-filtered and personalized. The use of recommender systems therefore also bears some risks and can potentially even lead to negative and undesired effects, such as limiting the range of items that are actually recommended, hence negating one of the main goals of RSs: information discovery. The chapter discusses how the impact

of recommender systems is currently measured, with a particular focus on measures that are used in practice.

A related and very important topic that also emerged in the last years is the fairness of the RS behaviour (Chapter “Fairness in Recommender Systems”). This defines another dimension for the evaluation of RSs, which goes beyond accuracy or user satisfaction. In fact, RSs pose unique challenges for investigating the fairness and non-discrimination concepts that have been developed in other machine learning literature. The multistakeholder nature of recommender applications, the ranked outputs, the centrality of personalization, and the role of user response complicate the problem of identifying precisely what type of fairness may be relevant, and how to precisely measure and evaluate it. In Chapter “Fairness in Recommender Systems” the authors distinguish various ways a recommender system may be unfair and provide a conceptual framework for identifying the fairness that arise in an application and designing a project to assess and mitigate them.

## 7 Recommender Systems and Human-Computer Interaction

As we have illustrated in the previous sections, researchers have been chiefly concerned with designing a range of technical solutions and leveraging various sources of knowledge to achieve better predictions about the “utility” of the various items for the target user, or what is going to be chosen by the target user, in order to “anticipate” these choices with targeted recommendations. The underlying assumption behind this research activity is that presenting these “correct” recommendations, or the best options, would be sufficient. In other words, the recommendations should speak for themselves and the user should definitely accept the recommendations if they are correct. This is clearly an overly simplified account of the recommendation problem and the delivery and acceptance of recommendations is not so straightforward.

In practice, users need recommendations because they do not have enough knowledge to make an autonomous decision. Consequently, it may not be easy for them to evaluate the proposed recommendation. Hence, various researchers have tried to understand the factors that lead to the acceptance of a recommendation by a given user [8, 25, 35, 50, 67, 71].

Swearingen and Sinha [71] were among the first to point out that the effectiveness of an RS is dependent on factors that go beyond the quality of the prediction algorithm. In fact, the RS must also convince users to try (read, buy, listen, watch, etc.) the recommended items. This, of course, depends on the individual characteristics of the selected items, and therefore on the recommendation algorithm. The process also depends, however, on the particular human/computer interaction supported by the system when the items are presented, compared, and explained. Swearingen and Sinha [71] found that from a user’s perspective, an effective recommender system must inspire trust in the system and it must have a system logic that is at least somewhat transparent. Additionally, the authors note that it should point

users towards new, not-yet-experienced items, and should provide details about recommended items, including pictures and community ratings, and finally, it should present ways to refine recommendations.

Swearingen and Sinha [71] and other similarly oriented researchers do not diminish the importance of the recommendation algorithm, but claim that its effectiveness should not be evaluated only in terms of the accuracy of the prediction, i.e., with standard and popular Machine Learning or Information Retrieval metrics, such as Mean Absolute Error (MAE), precision, or Normalized Discounted Cumulative Gain (NDCG) (see Chapter “Evaluating Recommender Systems”). Other dimensions should be measured that relate to the acceptance of the recommender system and its recommendations. These ideas have been remarkably well presented and discussed also by McNee et al. [50]. In that work, the authors propose user-centric directions for evaluating recommender systems, including: the similarity of recommendation lists, recommendation serendipity, and the importance of user needs and expectations in a recommender. These seminal papers have played an important role in driving the RSs research towards a more user-centric analysis. In this handbook some, already mentioned contributions, deal with these novel evaluation dimensions: Chapters “Fairness in Recommender Systems”, “Novelty and Diversity in Recommender Systems”, and “Value and Impact of Recommender Systems”.

An essential goal of recommender systems is to help users make better choices [23, 34, 37]. Thus, it is important to understand how people make choices and how the human decision making process can be supported. Chapter “Individual and Group Decision Making and Recommender Systems” begins with a compact overview of the psychology of everyday choice and decision making that is based on a large literature of psychological research and formulated so as to be relevant and accessible to recommender systems research community. The authors explain how recommender systems can be viewed as one of many available tools for facilitating choice. Then, the authors provide a high-level overview of strategies for helping people make better choices, indicating how recommender systems fit into the greater picture of choice. The authors show how an understanding of human decision making can inform research and practice concerning these processes. The revised version of this chapter extends the analysis to choices made by groups and their support by recommender systems for groups.

Personality accounts for the most important way in which individuals differ in their enduring emotional, interpersonal, experiential, attitudinal and motivational styles. Several recent studies have shown that personality can be especially useful at tackling the issues of the cold start problem in RSs and in generating diverse recommendations. Hence, the exploitation of the user personality can improve the user-system interaction in several ways. Chapter “Personality and Recommender Systems” discusses first how personality relates to user preferences and how to use personality in recommender systems. The authors present the Five Factor Model (FFM) of personality. This model appears suitable for usage in RSs as it can be easily quantified in terms of features corresponding to the main factors. The acquisition of the personality factors for an observed user can be made either explicitly through questionnaires or implicitly using machine learning approaches on data

derived from social media streams or mobile phone call logs. The authors survey personality acquisition methods, strategies for using personality in recommender systems, and available datasets to use in offline recommender systems experiment.

Finally, Chapter “Beyond Explaining Single Item Recommendations” tackles an additional aspect of the user-centric approach to evaluation and highlights an important aspect of RS: explanations of the recommendation results to the users. It examines the reasons that make an evaluation “good” and the effects that explanations might have on RS acceptance. The chapter first analyses the interaction between the recommender system and the explanation in terms of preference elicitation methods and the presentation of results, as well as the recommendation algorithm. Then, explanation styles are described, along with examples of explanation in existing systems. The goals of explanations are listed from which metrics that measure the success of explanations in achieving these goals are described. The chapter concludes with challenges related to explanations. This includes the context in which explanations should be shown, and a major challenge in evaluating the interaction between acceptance of recommendation and explanations, as well as how to assure that explanations are indeed helpful and do not lead users to make poor decisions.

## 8 Recommender Systems Applications

Recommender systems research, aside from its theoretical contribution, is generally aimed at practically improving industrial RSs and involves research about various practical aspects that apply to the implementation of the systems. Indeed, an RS is an example of large-scale machine learning and data mining algorithms in commercial practice [5, 6]. The common interest in the field, both from the research community and the industry has leveraged the availability of data for research on one hand and the evolution of enhanced algorithms on the other hand. Practical related research in RSs examines aspects that are relevant to different stages in the life cycle of an RS, namely, the design of the system, its implementation, evaluation, maintenance and enhancement during system operation. The Netflix Prize announced in 2006, described in Chapter “Advances in Collaborative Filtering”, was an important event for the recommender systems research community and industry, and their mutual interaction. It highlighted the importance of the recommendation of items to users and accelerated the development of many new data mining recommendation techniques. Even though the Netflix Prize initiated a lot of research activities, the prize addressed a simplification of the full recommendation problem. It reduced the recommendation problem to the prediction of user’s ratings while optimizing the Root Mean Square Error (RMSE).

The first factor to consider while designing an RS is the application’s domain, as it has a major effect on the algorithmic approach that should be taken. In [51] the authors provide a taxonomy of RSs and classify existing RS applications to specific application domains. Based on these specific application domains, we

define more general classes of domains for the most common recommender systems applications:

- Entertainment—recommendations for movies, music, games, and IPTV.
- Content—personalized newspapers, recommendation for documents, recommendations of webpages, e-learning applications, and e-mail filters.
- E-commerce—recommendations of products to buy such as books, cameras, PCs etc. for consumers.
- Services—recommendations of travel services, recommendation of experts for consultation, recommendation of houses to rent, or matchmaking services.
- Social—recommendation of people in social networks, and recommendations of content social media content such as tweets, Facebook feeds, LinkedIn updates, and others.

As recommender systems became more popular, interest roused in the potential advantages of new and diverse applications, such as recommending insurance policies, or recommending questions for question-answering systems. As the above list cannot cover all the application domains that are now being addressed by RS techniques: it gives only an initial description of the various types of application domains.

The developer of an RS for a certain application domain should understand the specific facets of the domain, its requirements, application challenges and limitations. Only after analyzing these factors one can be able to select an appropriate recommendation algorithm and to design an effective human-computer interaction. In the current version of the handbook, some of the chapters in this section describe applications of recommender systems in specific domains. Each of these chapters describes the requirements of an RS for a specific domain, its precise challenges and the suitable technologies and algorithms for addressing them.

A popular domain for recommendation is music, presented in Chapter “Music Recommendation Systems: Techniques, Use Cases, and Challenges”. Unique features of music items that pose various challenges for recommendations should be considered when designing and evaluating RSs for music. Such challenges include, for example, the short time that it takes a user to gain an opinion about a recommended item, as compared to a movie or a book, or the fact that the same item can be recommended many times. In addition, music can be recommended as a single item, a playlist, and abstracted by genre, performer, or band. Music RSs, as opposed to many other domains, rely heavily on content-based recommendation which brings specific challenges to the domain [49].

Another notable example of RSs that emerged with the diffusion of new communication technologies are recommender systems related to the social web, and specifically those that target the social media domain. With the rise of social networks (e.g., Facebook, LinkedIn, Tweeter, Flickr, and others), users are overloaded with information, activities and interactions. Social recommender systems are RSs that aim at assisting the user in identifying relevant content (e.g., tweets, feeds or images), and engage only in relevant activities and interactions (e.g., discussions, or comments). Apart from the RSs that have been developed to be dedicated to



social media, recommender systems in other domains can benefit from the new types of data that social media introduces about users to enhance the quality of standard RSs [31]. The term Social RS covers many types of RSs that are relevant to social media platforms. Chapter “Social Recommender Systems” describes two main types: recommendations of social media content and recommendations of people. For recommendations of social content, the chapter reviews various social content media domains, and provides a detailed case study and insights learned from a recommender system operated in the enterprise which suggests mixed social media items. The chapter lists three different types of people recommendations, namely, the recommendation of familiar people (e.g., classmates, family members) that are not connected in the network; recommendations of interesting people (to connect with, or follow), and recommendations of strangers (to date, to hire, or for various other purposes). It explains the complexity of people-recommendation and lists key topics that should be considered and should be further investigated. The list includes: the need for explanation, privacy concerns, social relationships, trust and reputation, as well as the need to define special evaluation measures.

In Chapter “Multimedia Recommender Systems: Algorithms and Challenges” the authors present multimedia recommender systems. These systems utilize multimodal data (e.g., image, text, audio, video) to represent the items. This representation is then used in a content-based filtering RS or a hybrid RS that combines collaborative filtering and content-based filtering. Multimedia recommender systems are commonly used in various application domains. For example, they are frequently used to recommend videos on entertainment platforms. They are also used for recommending photos on social media platforms or recommending paintings to visit in a (virtual) museum. Other domains include recommending news videos and short user-generated videos. Typical examples of audio recommendation include music creation (e.g., recommending sounds such as drum loops to a creator of electronic music), music consumption (e.g., automatic playlist generation), and information (e.g., podcast recommendation). Multimedia recommender systems can be used to extract features for obtaining a proxy multimedia representation of items to recommend non-media items. For example, many fashion and food RSs are leveraging the visual modality to enhance recommendation. In Chapter “Multimedia Recommender Systems: Algorithms and Challenges” the authors survey the state-of-the-art research related to multi-media RS and, in particular focusing on techniques that integrate item or user side information into a hybrid recommender. They also present various methods for automatically extracting features from multimedia data and creating an item representation based on various modalities (image, video, and audio).

Some new application domains for recommendation evolved with the emergence multi-media RS. One example, detailed in Chapter “Fashion Recommender Systems”, is presented by fashion RSs. Due to the growth of fashion e-commerce sites, there is an increasing need for personalized clothing recommendations that match users preferences. Fashion RSs can help users in two main scenarios. Firstly, they can help users to find alternative items to a given item (e.g. evening dress) that has the same style or look and feel. The second common scenario in fashion RS

is complementary item recommendations, i.e., helping a user find fashion items that complement a given one. For example, a belt and a tie that go with a given suit. Most of the state-of-the-art fashion RSs that are explored in Chapter “Fashion Recommender Systems” are based on deep learning methods due to their ability to extract meaningful features from the items (such as image, video, text, audio). Deep learning methods overcome the limitations of more traditional recommendation methods (e.g. cold start in nearest neighbors methods) and outperform them.

Another detailed example of RSs designed for a specific domain is described in Chapter “Food Recommender Systems”, which deals with recommender systems for food. The authors first present the problem of food recommender systems by detailing the numerous possible scenarios such as: (a) cooking RSs that given available time and ingredients, aim to fit the users taste preferences, health needs (e.g., allergies and intolerance), religious (e.g., Kosher food) and other dietary restrictions (e.g., vegetarian); (b) grocery RSs; and (c) restaurant RSs help groups of people who want to visit a restaurant together as a social. The authors review several food RS solutions proposed in the literature including well performing algorithms: traditional algorithms for user-item ranking problem that are adapted to the food domain and context-aware RS algorithms that take into consideration time and space. In addition, they present algorithms that can change behavioral habits into healthier one in attempt to prevent illnesses associated with being overweight, diabetes and hypertension. The authors also present the importance of user interfaces in food RSs as demonstrated in various recent researches. Several successful user-interface for food RS are presented.

## 9 Challenges

The list of newly emerging and challenging RS research topics is not limited to those described in the previous sections of this introduction. Moreover, it is impossible to cover all of them within this short introduction. The reader is referred to the final discussion sections that are included in almost all of the chapters published in this handbook for other important challenges for RS research.

Below, we briefly introduce some of them, which we deem as particularly important.

### 9.1 Preference Acquisition

A number of open issues are related to the critical stage of acquiring reliable information about the user preferences in order to generate a useful user profile.

It is clear that in many real-world applications of RSs, implicit feedback is much more readily available and requires no extra effort on the user’s side. For instance, on a web site it is easy to log the users visiting a URL, or clicking on an ad.

The system can treat these actions as a form of positive feedback to the displayed items. That information contains relevant information for predicting future users' actions. For that reason many recent RSs focus only on the use of implicit feedback. This topic is largely covered in Chapter "Item Recommendation from Implicit Feedback", which is fully dedicated to this subject, but also in Chapters "Session-Based Recommender Systems" and "Advances in Collaborative Filtering". In cases only implicit feedback is available, the core computational problem becomes the prediction of the probability that a user will interact with a given item, by performing a target action, such as listening to a music track or browsing a web page. In such a setting, very often, there is no clear negative signal, i.e., an action that signal a lack of interest for an item, hence the available data is either positive or missing. The missing data includes both items that the user explicitly chose to ignore because they were not appealing and items that would have been perfect recommendations but were never presented to the user or discovered by her. Hence, also when managing implicit feedback one has to consider the biases of the available data and the implication for the construction of effective and fair systems. These topics are further discussed in Chapters "Novelty and Diversity in Recommender Systems", "Value and Impact of Recommender Systems", "Multistakeholder Recommender Systems", and "Fairness in Recommender Systems". But more effective solutions for tackling it must be further developed.

Notwithstanding the large availability of implicit feedback, this data cannot completely substitute the usage of explicitly user-made evaluations, at last for some item categories. This is signalled by the fact that many web sites and applications of RSs are still offering to their users the possibility to enter explicitly their opinion on the recommended or consumed items. Moreover, in certain application domains explicit feedback is still central. This is clearly illustrated in Chapters "Social Recommender Systems", "Group Recommender Systems: Beyond preference aggregation", and "People-to-People Reciprocal Recommenders". These are application domains or settings where the RS cannot totally operate as a black box; users are interacting also with themselves, not only with items. In these applications the RS acts as a mediator between users, enabling the users to better understand each other and browse recommendations that are often referring to actions or preferences of other users. This is clearly seen in a group RS where the goal of the system is to support the choices of a group of users, which often requires the reciprocal understanding of the group members opinions and preferences. Hence, collecting, storing, extracting relevant information from explicit feedback of the group members is pivotal. Ample discussion of these topics is provided also in Chapter "Individual and Group Decision Making and Recommender Systems", where the authors clearly expand the role of a RS beyond the narrow functionality of predicting the most relevant items for each user.

Therefore, simplifying the cognitive cost of preference acquisition is of primary importance. For an RS to achieve good recommendation performance, users typically need to provide the system with a certain amount of feedback about their preferences (e.g., in the form of item ratings). Therefore, it is important to measure the costs and benefits of adopting alternative rating approaches and scales,

and to find an optimal solution to meet the needs of both the users, the suppliers and the platform owners. In fact, the acquired preference can have a profound impact on the quality and distribution of the recommendations for all the involved stakeholders. This topic is discussed extensively in Chapters “Multistakeholder Recommender Systems”, “Fairness in Recommender Systems” and “Novelty and Diversity in Recommender Systems”, where the current research challenges receive ample discussion.

An interesting direction of research for easing the preference acquisition process consists of the exploitation of personality, mood and emotions. This is becoming a popular topic, especially because it is clear that more and more techniques will be developed in order to automatically and unobtrusively acquire such information. In Chapter “Personality and Recommender Systems”, the authors stress the challenge of acquiring personality information in a nonintrusive fashion. Nowadays, only the longest questionnaires, which consist of around one hundred questions, can provide an accurate evaluation of the user’s personality. Hence, non-intrusive approaches are necessary and the research in this area is just starting. Mining user activity for extracting personality information is an option, but also the fast penetration of portable devices and new sensor types, which can potentially life-logging the user’s activity, can offer a promising platform that is worth exploring.

Another line of research aimed at tackling the cold start problem and reducing the user preference elicitation effort is cross-domain recommender systems. These techniques (see Chapter “Design and Evaluation of Cross-Domain Recommender Systems”) could be used as an alternative path to user preferences’ elicitation as they are able to build detailed user profiles without the need to collect explicit user assessment of the target domain items.

## **9.2 Interaction**

A major challenge that RS research is now facing, and is clearly addressed in several chapters of this handbook, is that we still need to broaden the scope of research to the system aspects of a recommender system. This means that aside from the algorithms, which are used to predict the user preference and behaviour, and compute the recommendations, the mechanism through which users provide their input and the means by which they receive the systems output, play a significant role and can play an even larger role in determining the success or failure of a recommender system. We still need to better understand the general qualities of alternative solutions to preference elicitation, as we mentioned previously, but also to recommendation presentation, and to develop personalized solutions for these phases of the interaction with the system.

It must be observed that while interacting with a recommender system, users make various types of decisions. The most important one is surely selecting an item from the recommendation list. But, before making the final decision, users often have to decide how to explore the information space and what additional

input they could provide to the system. For instance, they could have to select a specific group of recommendations (e.g., the recommendations about items recently added to the catalogue), or to indicate search conditions, such as the category of searched products, that can help to narrow down the typically large set of received recommendations. Moreover, users often do not know or do not reflect on their preferences beforehand, especially when users approach an RS for information discovery. In such cases, the system-supported interaction and visualization contribute to the user construction of their preferences within a specific recommendation session. As it has been illustrated in Chapter “Individual and Group Decision Making and Recommender Systems”, there are several challenges with the full support of user decision making in a recommender system. Our understanding of the situational context generated by the system and its effect on item selection processes is still incomplete and we need to better connect RS research to psychology and decision making disciplines. While it is clear that an RS helps to make decisions, there is still the need for further develop research that takes theories from decision psychology and cognitive psychology into account when explaining users’ preference construction and decision making process in the context of recommender systems [52].

Considering the user interaction with the recommender system, the topic of explaining the system recommendations still poses a number of interesting and open issues. For instance, it is still not completely clear whether explanations bring more overall benefits than risks. In Chapter “Beyond Explaining Single Item Recommendations” it is shown that explanations are part of a cyclical process: the explanations affect the acceptance of particular recommendations, the users’ mental model of the system, and in turn, this affects the ways users interact with the explanations. But, whether the users are influenced in such a way that their choices are improved, and not biased, is not clear. Explanations may even increase the information overload that RSs are supposed to tame, or make the decision problem, in some cases, even more complex. Moreover, while some research has been conducted on explaining recommendations to individual users, explaining recommendations for a group is still an unexplored subject [29, 72]. For instance, one might think that showing how satisfied other group members are, and how this prediction is incorporated in the recommendations selected for a group, could improve users’ understanding of the recommendation process and perhaps make it easier to accept items they do not like. However, users need for privacy is likely to conflict with their need for transparency. Moreover, showing the preferences of other users may move the group discussion on the preferences, i.e., how much the system is taking care of the other group member preferences, rather than on the value of the recommended items. We definitely need more research on these topics.

In a discussion about the human-computer interaction in recommender systems, we cannot forget the issue of the assessment of the value of the recommendations, which is not only related to what extent the recommended items are liked by the user. The general relationships between the value of the RS, the supported interaction and the instruments adopted to measure if the RS meets the goal of their users, are discussed in two chapters of this book (Chapters “Value and Impact of

Recommender Systems” and “Evaluating Recommender Systems”). In fact, there is a number of specific factors that may influence the real usefulness of a recommendation during a specific human-computer interaction. For instance, the time value of recommendations, which is partially discussed in the chapter on context-aware RS (Chapter “Context-Aware Recommender Systems: From Foundations to Recent Developments”), refers to the fact that a given set of recommendations may not be applicable forever but there could be a time interval when these items can be recommended. This is clear, for instance, when it comes to news items: people want to be informed about the most recent events and news lose their value even one day after the initial announcement. The time value of a recommendation is clearly dependent on the novelty and diversity of the recommended items. We still need more theoretical, methodological and algorithmic developments on these aspects. For instance, modelling feature-based novelty in probabilistic terms in order to unify discovery and familiarity models would be an interesting line for future work. Aspects such as the time dimension during which items may recover part of their novelty value, or the variability among users regarding their degree of novelty-seeking are examples of issues that require further research and are mentioned in Chapter “Novelty and Diversity in Recommender Systems”.

### ***9.3 New Recommendation Tasks***

The application of recommender systems is still dominated by solutions for recommending relatively simple and inexpensive products, like movies, music, news and books. While there are systems managing more complex item types, such as financial investment or travel, these item categories are considered as atypical cases. Inevitably, complex domains require more elaborated solutions. Complex products are typically configurable or offered in several variants. This feature still poses a challenge to RS, which are instead designed to consider different configurations as different items. Identifying the more suitable configuration requires reasoning on the relations between configurations (classifying and grouping items) and calls for addressing the specificity of the human decision making task generated by the selection of a configuration. In this handbook, Chapter “Fashion Recommender Systems” focuses on such a complex type of products. For instance in the fashion domain, users rarely think about what to wear and buy in isolation, so the RS must enable users to find fashion items that complement a given one. In general, addressing new types and more structured types of items can call for the introduction of new and interesting research lines. For instance, Chapter “People-to-People Reciprocal Recommenders” clearly shows how different the recommendation techniques must be in domains where reciprocal recommendations are needed, as in dating or job finding applications.

As we already indicated in the previous editions of this handbook, RSs that optimise a sequence of recommendations, e.g., a new book recommendation every week, are not frequent and we believe that this is still an open issue. A considerable

amount of new research has focused on Session-based RS, as it is clearly illustrated in Chapter “Session-Based Recommender Systems”. But, the current research is still motivated by the goal to predict the sequential choice behaviour of users, which is extracted from the available data, rather than to construct sequences of recommendations that have specific properties that the users may appreciate, such as nudging the user towards better and better choices, or more diverse choices. This is an important dimension in many application, for instance in eLearning but even more clearly in the food domain. For instance, while some initial attempts to build a complete meal plan have been tried, still current food recommender systems are not able to effectively generate such sequences of recommendations (see Chapter “Food Recommender Systems”).

Moreover, it is important to study the sequential dimension of users’ decision making both within a recommendation session and between recommendation sessions. Here, we want to further note the importance of such a topic in group RSs (see Chapter “Group Recommender Systems: Beyond Preference Aggregation”). In these systems, sequential recommendations are a natural setting, since stable groups, such as friends or families, repeatedly choose items of the same type, e.g., when deciding where to go for vacation or what to eat at home. More research is needed on algorithms and user interfaces for producing coherent sequences of recommendations. In particular, one should model the effect on users of several contextual conditions such as the manner in which already-shown-items could influence the user evaluation of the next recommendations, or the social role and relationships of the group members.

Most of the popular RSs are now accessed through mobile devices that follow their owners throughout their daily life, and are always within an arm’s reach of their owners. In this scenario, RSs can proactively send notifications to their users about items of potential interest that are relevant because of the contextual situation of the user. The challenge is finding true relevant items for the user situation and not overburdening the user with a stream of irrelevant interruptions. To address this goal, we must better exploit implicit feedback derived from user usage, which means, we should try to build models that can better generalise from the observed behaviour and identify novel and diverse recommendations (see Chapter “Novelty and Diversity in Recommender Systems”, and [47]). But, such systems should also learn to better identify contextual situations that require a push of the recommendations. We believe that this depends on the detection of contextual changes that are significant to the user and therefore justify a recommendation. For instance, when it is the ideal time for a pause in writing a paper, i.e., the context is changing from work to leisure, a recommendation of a relevant, or personalized, article of sports news can be delivered. Understanding when context changes, or it could be forced to change, and when a user may be receptive to a recommendation push is a challenging issue for further research. As it is also suggested in Chapter “Context-Aware Recommender Systems: From Foundations to Recent Developments”, in order to develop these new and compelling context aware systems, we need to explore novel engineering solutions, including: novel

data structures, storage systems, user interface components and service oriented architectures.

Finally, we hope that this handbook, as a useful tool for practitioners and researchers, will contribute to further developing knowledge in this exciting and useful research area and provide a baseline for further exploring the above mentioned issues. Currently the research on RSs has greatly benefited from the combined interest and efforts that industry and academia have invested in this field. We therefore wish the best to both groups as they read this handbook and we hope that it will attract even more researchers to work in this highly interesting and challenging field.

## References

1. G. Adomavicius, K. Bauman, B. Mobasher, F. Ricci, A. Tuzhilin, M. Unger, Workshop on context-aware recommender systems, in *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22–26, 2020*, ed. by R.L.T. Santos, L.B. Marinho, E.M. Daly, L. Chen, K. Falk, N. Koenigstein, E.S. de Moura (ACM, New York, 2020), pp. 635–637
2. G. Adomavicius, A. Tuzhilin, Personalization technologies: a process-oriented perspective. *Commun. ACM* **48**(10), 83–90 (2005)
3. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **17**(6), 734–749 (2005)
4. C.C. Aggarwal, *Recommender Systems - The Textbook* (Springer, New York, 2016)
5. X. Amatriain, Mining large streams of user data for personalized recommendations. *SIGKDD Explor. Newsl.* **14**(2), 37–48 (2013)
6. X. Amatriain, J. Basilico, Past, present, and future of recommender systems: an industry perspective, in *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15–19, 2016*, ed. by S. Sen, W. Geyer, J. Freyne, P. Castells (ACM, New York, 2016), pp. 211–214
7. O. Arazy, N. Kumar, B. Shapira, Improving social recommender systems. *IT Prof.* **11**(4), 38–44 (2009)
8. H. Asoh, C. Ono, Y. Habu, H. Takasaki, T. Takenaka, Y. Motomura, An acceptance model of recommender systems based on a large-scale internet survey, in *Advances in User Modeling - UMAP 2011 Workshops, Girona, July 11–15, 2011, Revised Selected Papers* (2011), pp. 410–414
9. R.A. Bailey, *Design of Comparative Experiments* (Cambridge University Press, Cambridge, 2008)
10. M. Balabanovic, Y. Shoham, Content-based, collaborative recommendation. *Commun. ACM* **40**(3), 66–72 (1997)
11. L. Baltrunas, F. Ricci, Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Model. User-Adapt. Interact.* **24**(1–2), 7–34 (2014)
12. D. Ben-Shimon, A. Tsikinovsky, L. Rokach, A. Meisels, G. Shani, L. Naamani, Recommender system from personal social networks, in *AWIC, Advances in Soft Computing*, vol. 43, ed. by K. Wegrzyn-Wolska, P.S. Szczepaniak (Springer, New York, 2007), pp. 47–55
13. S. Berkovsky, T. Kuflik, F. Ricci, Mediation of user models for enhanced personalization in recommender systems. *User Model. User-Adapted Interact.* **18**(3), 245–286 (2008)
14. S. Berkovsky, T. Kuflik, F. Ricci, Cross-representation mediation of user models. *User Model. User-Adapted Interact.* **19**(1–2), 35–63 (2009)



15. B. Biggio, I. Corona, B. Nelson, B.I., Rubinstein, D. Maiorca, G. Fumera, G. Giacinto, F. Roli, Security evaluation of support vector machines in adversarial environments, in *Support Vector Machines Applications* (Springer, New York, 2014), pp. 105–153
16. D. Billsus, M. Pazzani, Learning probabilistic user models, in *UM97 Workshop on Machine Learning for User Modeling* (1997). <http://www.dfki.de/bauer/um-ws/>
17. J. Bobadilla, F. Ortega, A. Hernando, A. Gutierrez, Recommender systems survey. *Knowl. Based Syst.* **46**(0), 109–132 (2013)
18. J. Borràs, A. Moreno, A. Valls, Intelligent tourism recommender systems: a survey. *Expert Syst. Appl.* **41**(16), 7370–7389 (2014)
19. D. Bridge, M. Göker, L. McGinty, B. Smyth, Case-based recommender systems. *Knowl. Eng. Rev.* **20**(3), 315–320 (2006)
20. P. Brusilovsky, Methods and techniques of adaptive hypermedia. *User Model. User-Adapted Interact.* **6**(2–3), 87–129 (1996)
21. R. Burke, Hybrid web recommender systems, in *The Adaptive Web* (Springer, Berlin, 2007), pp. 377–408
22. M. Chelliah, S. Sarkar, Product recommendations enhanced with reviews, in *Proceedings of the Eleventh ACM Conference on Recommender Systems* (2017), pp. 398–399
23. L. Chen, M. de Gemmis, A. Felfernig, P. Lops, F. Ricci, G. Semeraro, Human decision making and recommender systems. *TiS* **3**(3), 17 (2013)
24. L. Chen, P. Pu, Critiquing-based recommenders: survey and emerging trends. *User Model. User-Adapt. Interact.* **22**(1–2), 125–150 (2012)
25. D. Cosley, S.K., Lam, I., Albert, J.A., Konstant, J. Riedl, Is seeing believing? How recommender system interfaces affect users’ opinions, in *Proceedings of the CHI 2003 Conference on Human factors in Computing Systems, Fort Lauderdale, FL* (2003), pp. 585–592
26. M.D. Ekstrand, F.M. Harper, M.C. Willemsen, J.A. Konstan, User perception of differences in recommender algorithms, in *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA*, 06–10 Oct 2014, pp. 161–168
27. K. Falk, *Practical Recommender Systems* (Manning Publications, Shelter Island, 2019)
28. C. Feely, B. Caulfield, A. Lawlor, B. Smyth, Using case-based reasoning to predict marathon performance and recommend tailored training plans, in *Case-Based Reasoning Research and Development - 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020*, ed. by I. Watson, R.O. Weber. *Lecture Notes in Computer Science*, vol. 12311 (Springer, New York, 2020), pp. 67–81
29. A. Felfernig, N. Tintarev, T.N.T. Trang, M. Stettinger, Designing explanations for group recommender systems. *CoRR abs/2102.12413* (2021)
30. G. Fisher, User modeling in human-computer interaction. *User Model. User-Adapted Interact.* **11**, 65–86 (2001)
31. J. Golbeck, Generating predictive movie recommendations from trust in social networks, in *Trust Management, 4th International Conference, iTrust 2006 Proceedings, Pisa*, 16–19 May 2006, pp. 93–104
32. D. Goldberg, D. Nichols, B.M. Oki, D. Terry, Using collaborative filtering to weave an information tapestry. *Commun. ACM* **35**(12), 61–70 (1992)
33. S.K. Gorakala, M. Usulli, *Building a Recommendation System with R* (Packt Publishing Ltd., Birmingham, 2015)
34. N. Hazrati, M. Elahi, F. Ricci, Simulating the impact of recommender systems on the evolution of collective users’ choices, in *HT '20: 31st ACM Conference on Hypertext and Social Media, Virtual Event, 13–15 July 2020*, ed. by U. Gadiraju (ACM, New York, 2020), pp. 207–212
35. J. Herlocker, J. Konstan, J. Riedl, Explaining collaborative filtering recommendations, in *Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work* (2000), pp. 241–250
36. J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22**(1), 5–53 (2004)

37. A. Jameson, Recommender systems seen through the lens of choice architecture, in *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, IntRS 2015, Co-located with ACM Conference on Recommender Systems (RecSys 2015), Vienna, 19 Sept 2015, CEUR Workshop Proceedings*, vol. 1438, ed. by J. O'Donovan, A. Felfernig, N. Tintarev, P. Brusilovsky, G. Semeraro, P. Lops (2015), p. 1. [CEUR-WS.org](http://CEUR-WS.org)
38. D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, *Recommender Systems: An Introduction* (Cambridge University Press, Cambridge, 2010)
39. J.L. Jorro-Aragoneses, M. Caro-Martínez, J.A. Recio-García, B. Díaz-Agudo, G. Jiménez-Díaz, Personalized case-based explanation of matrix factorization recommendations, in *Case-Based Reasoning Research and Development - 27th International Conference, ICCBR 2019, Otzenhausen, Germany, 8–12 Sept 2019, Proceedings*, vol. 11680, ed. by K. Bach, C. Marling, Lecture Notes in Computer Science (Springer, New York, 2019), pp. 140–154
40. J.A. Konstan, J. Riedl, Recommender systems: from algorithms to user experience. *User Model. User-Adapted Interact.* **22**(1–2), 101–123 (2012)
41. Y. Koren, R.M. Bell, C. Volinsky, Matrix factorization techniques for recommender systems. *IEEE Comput.* **42**(8), 30–37 (2009)
42. G. Linden, B. Smith, J. York, [Amazon.com](http://Amazon.com) recommendations: item-to-item collaborative filtering. *IEEE Internet Comput.* **7**(1), 76–80 (2003)
43. P. Lops, M. de Gemmis, G. Semeraro, Content-based recommender systems: state of the art and trends, in *Recommender Systems Handbook*, ed. by F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Springer, New York, 2011), pp. 73–105
44. L. Lu, M. Medo, C.H. Yeung, Y.C. Zhang, Z.K. Zhang, T. Zhou, Recommender systems. *Phys. Rep.* **519**(1), 1–49 (2012)
45. T. Mahmood, F. Ricci, A. Venturini, Learning adaptive recommendation strategies for online travel planning, in *Information and Communication Technologies in Tourism 2009* (Springer, New York, 2009), pp. 149–160
46. D. Massimo, F. Ricci, Clustering users' pois visit trajectories for next-poi recommendation, in *Information and Communication Technologies in Tourism 2019, ENTER 2019, Proceedings of the International Conference in Nicosia, Cyprus, 30 Jan–1 Feb 2019*, ed. by J. Pesonen, J. Neidhardt (Springer, New York, 2019), pp. 3–14
47. D. Massimo, F. Ricci, Enhancing travel experience leveraging on-line and off-line users' behaviour data, in *IUI '20: 25th International Conference on Intelligent User Interfaces, Cagliari, 17–20 March 2020, Companion* (ACM, New York, 2020), pp. 65–66
48. J. McAuley, C. Targett, Q. Shi, A. Van Den Hengel, Image-based recommendations on styles and substitutes, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2015), pp. 43–52
49. B. McFee, T. Bertin-Mahieux, D.P. Ellis, G.R. Lanckriet, The million song dataset challenge, in *Proceedings of the 21st International Conference Companion on World Wide Web, WWW '12 Companion* (ACM, New York, 2012), pp. 909–916
50. S.M. McNea, J. Riedl, J.A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, in *CHI '06: CHI '06 Extended Abstracts on Human Factors in Computing Systems* (ACM Press, New York, 2006), pp. 1097–1101
51. M. Montaner, B. López, J.L. de la Rosa, A taxonomy of recommender agents on the internet. *Artif. Intell. Rev.* **19**(4), 285–330 (2003)
52. M. Otsuka, T. Osogami, A deep choice model, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 12–17 Feb 2016, Phoenix, Arizona*, ed. by D. Schuurmans, M.P. Wellman (AAAI Press, Menlo Park, 2016), pp. 850–856
53. T.K. Paradarani, N.D. Bastian, J.L. Wightman, A hybrid recommender system using artificial neural networks. *Expert Syst. Appl.* **83**, 300–313 (2017)
54. D.H. Park, H.K. Kim, I.Y. Choi, J.K. Kim, A literature review and classification of recommender systems research. *Expert Syst. Appl.* **39**(11), 10059–10072 (2012)
55. M. Perano, G.L. Casali, Y. Liu, T. Abbate, Professional reviews as service: a mix method approach to assess the value of recommender systems in the entertainment industry. *Technol. Forecast. Soc. Change* **169**, 120800 (2021)

56. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in *Proceedings ACM Conference on Computer-Supported Cooperative Work* (1994), pp. 175–186
57. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (1994), pp. 175–186
58. P. Resnick, H.R. Varian, Recommender systems. *Commun. ACM* **40**(3), 56–58 (1997)
59. M. Reusens, W. Lemahieu, B. Baesens, L. Sels, A note on explicit versus implicit information for job recommendation. *Decis. Support Syst.* **98**, 26–35 (2017)
60. F. Ricci, Travel recommender systems. *IEEE Intell. Syst.* **17**(6), 55–57 (2002)
61. F. Ricci, Recommender systems: models and techniques, in *Encyclopedia of Social Network Analysis and Mining*, ed. by R. Alhajj, J.G. Rokne, 2nd edn. (Springer, New York, 2018)
62. F. Ricci, D. Cavada, N. Mirzadeh, A. Venturini, Case-based travel recommendations, in *Destination Recommendation Systems: Behavioural Foundations and Applications*, ed. by D.R. Fesenmaier, K. Woeber, H. Werthner (CABI, Wallingford, 2006), pp. 67–93
63. J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in *The Adaptive Web* (Springer, Berlin, 2007), pp. 291–324
64. J.B. Schafer, J.A. Konstan, J. Riedl, E-commerce recommendation applications. *Data Min. Knowl. Disc.* **5**(1/2), 115–153 (2001)
65. M. Schrage, *Recommendation Engines* (MIT Press, Cambridge, 2020)
66. B. Schwartz, *The Paradox of Choice* (ECCO, New York, 2004)
67. M. van Setten, S.M. McNee, J.A. Konstan, Beyond personalization: the next stage of recommender systems research, in *IUI*, ed. by R.S. Amant, J. Riedl, A. Jameson (ACM, New York, 2005), p. 8
68. U. Shardanand, P. Maes, Social information filtering: algorithms for automating “word of mouth”, in *Proceedings of the Conference on Human Factors in Computing Systems (CHI’95)* (1995), pp. 210–217
69. R.R. Sinha, K. Swearingen, Comparing recommendations made by online systems and friends, in *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries* (2001)
70. B. Smith, G. Linden, Two decades of recommender systems at amazon.com. *IEEE Internet Comput.* **21**(3), 12–18 (2017)
71. K. Swearingen, R. Sinha, Beyond algorithms: an HCI perspective on recommender systems, in *Recommender Systems, papers from the 2001 ACM SIGIR Workshop, New Orleans, LA*, ed. by J.L. Herlocker (2001)
72. T.N.T. Tran, M. Atas, A. Felfernig, V.M. Le, R. Samer, M. Stettinger, Towards social choice-based explanations in group recommender systems, in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2019, Larnaca, Cyprus, 9–12 June 2019*, ed. by G.A. Papadopoulos, G. Samaras, S. Weibelzahl, D. Jannach, O.C. Santos (ACM, New York, 2019), pp. 13–21
73. S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: a survey and new perspectives. *ACM Comput. Surv.* **52**(1), 1–38 (2019)