Cambridge University Press
978-0-521-49336-9 - Recommender Systems: An Introduction
Dietmar Jannach, Markus Zanker, Alexander Felfernig and Gerhard Friedrich
Excerpt
More information

1
Introduction

Most Internet users have come across a recommender system in one way or another. Imagine, for instance, that a friend recommended that you read a new book and that you subsequently visit your favorite online bookstore. After typing in the title of the book, it appears as just one of the results listed. In one area of the web page possibly called "Customers Who Bought This Item Also Bought," a list is shown of additional books that are supposedly of interest to you. If you are a regular user of the same online bookstore, such a personalized list of recommendations will appear automatically as soon as you enter the store. The software system that determines which books should be shown to a particular visitor is a *recommender system*.

The online bookstore scenario is useful for discussing several aspects of such software systems. First, notice that we are talking about *personalized* recommendations – in other words, every visitor sees a different list depending on his or her tastes. In contrast, many other online shops or news portals may simply inform you of their top-selling items or their most read articles. Theoretically, we could interpret this information as a sort of impersonal buying or reading recommendation as well and, in fact, very popular books will suit the interests and preferences of many users. Still, there will be also many people who do not like to read *Harry Potter* despite its strong sales in 2007 – in other words, for these people, recommending top-selling items is not very helpful. In this book, we will focus on systems that generate personalized recommendations.

The provision of personalized recommendations, however, requires that the system knows something about every user. Every recommender system must develop and maintain a *user model* or *user profile* that, for example, contains the user's preferences. In our bookstore example, the system could, for instance, remember which books a visitor has viewed or bought in the past to predict which other books might be of interest.

Although the existence of a user model is central to every recommender system, the way in which this information is acquired and exploited depends on the particular recommendation technique. User preferences can, for instance, be acquired *implicitly* by monitoring user behavior, but the recommender system might also *explicitly* ask the visitor about his or her preferences.

The other question in this context is what kind of *additional information* the system should exploit when it generates a list of personalized recommendations. The most prominent approach, which is actually used by many real online bookstores, is to take the behavior, opinions, and tastes of a large community of other users into account. These systems are often referred to as *community-based* or *collaborative* approaches.

This textbook is structured into two parts, reflecting the dynamic nature of the research field. Part I summarizes the well-developed aspects of recommendation systems research that have been widely accepted for several years. Therefore, Part I is structured in a canonical manner and introduces the basic paradigms of collaborative (Chapter 2), content-based (Chapter 3), and knowledge-based recommendation (Chapter 4), as well as hybridization methods (Chapter 5). Explaining the reasons for recommending an item (Chapter 6) as well as evaluating the quality of recommendation systems (Chapter 7) are also fundamental chapters. The first part concludes with an experimental evaluation (Chapter 8) that compares different recommendation algorithms in a mobile environment that can serve as a practical reference for further investigations. In contrast, Part II discusses very recent research topics within the field, such as how to cope with efforts to attack and manipulate a recommender system from outside (Chapter 9), supporting consumer decision making and potential persuasion strategies (Chapter 10), recommendation systems in the context of the social and semantic webs (Chapter 11), and the application of recommender systems to ubiquitous domains (Chapter 12). Consequently, chapters of the second part should be seen as a reference point for ongoing research.

1.1 Part I: Introduction to basic concepts

1.1.1 Collaborative recommendation

The basic idea of these systems is that if users shared the same interests in the past – if they viewed or bought the same books, for instance – they will also have similar tastes in the future. So, if, for example, user *A* and user *B* have a purchase history that overlaps strongly and user *A* has recently bought a book that *B* has not yet seen, the basic rationale is to propose this book also to *B*. Because this selection of hopefully interesting books involves filtering the most promising ones from a large set and because the users implicitly collaborate with one another, this technique is also called *collaborative filtering* (CF).

Today, systems of this kind are in wide use and have also been extensively studied over the past fifteen years. We cover the underlying techniques and open questions associated with collaborative filtering in detail in the next chapter of this book. Typical questions that arise in the context of collaborative approaches include the following:

- How do we find users with similar tastes to the user for whom we need a recommendation?
- How do we measure similarity?
- What should we do with new users, for whom a buying history is not yet available?
- How do we deal with new items that nobody has bought yet?
- What if we have only a few ratings that we can exploit?
- What other techniques besides looking for similar users can we use for making a prediction about whether a certain user will like an item?

Pure CF approaches do not exploit or require any knowledge about the items themselves. Continuing with the bookstore example, the recommender system, for instance, does not need to know what a book is about, its genre, or who wrote it. The obvious advantage of this strategy is that these data do not have to be entered into the system or maintained. On the other hand, using such characteristics to propose books that are actually similar to those the user liked in the past might be more effective.

1.1.2 Content-based recommendation

In general, recommender systems may serve two different purposes. On one hand, they can be used to stimulate users into doing something such as buying a specific book or watching a specific movie. On the other hand, recommender systems can also be seen as tools for dealing with *information overload*, as these systems aim to select the most interesting items from a larger set. Thus, recommender systems research is also strongly rooted in the fields of *information retrieval* and *information filtering*. In these areas, however, the focus lies mainly on the problem of discriminating between relevant and irrelevant *documents* (as opposed to the artifacts such as books or digital cameras recommended in traditional e-commerce domains). Many of the techniques developed in these areas exploit information derived from the documents' contents to rank them. These techniques will be discussed in the chapter on content-based recommendation[1].

At its core, content-based recommendation is based on the availability of (manually created or automatically extracted) item descriptions and a profile that assigns importance to these characteristics. If we think again of the bookstore example, the possible characteristics of books might include the genre, the specific topic, or the author. Similar to item descriptions, user profiles may also be automatically derived and "learned" either by analyzing user behavior and feedback or by asking explicitly about interests and preferences.

In the context of content-based recommendation, the following questions must be answered:

- How can systems automatically acquire and continuously improve user profiles?
- How do we determine which items match, or are at least similar to or compatible with, a user's interests?
- What techniques can be used to automatically extract or learn the item descriptions to reduce manual annotation?

When compared with the content-agnostic approaches described above, content-based recommendation has two advantages. First, it does not require large user groups to achieve reasonable recommendation accuracy. In addition, new items can be immediately recommended once item attributes are available. In some domains, such item descriptions can be automatically extracted (for instance, from text documents) or are already available in an electronic catalog. In many domains, however, the more subjective characteristics of an item – such as "ease of use" or "elegance of design" – would be useful in the recommendation process. These characteristics are hard to acquire automatically, however, meaning that such information must be manually entered into the system in a potentially expensive and error-prone process.

1.1.3 Knowledge-based recommendation

If we turn our attention to other application domains, such as consumer electronics, many involve large numbers of one-time buyers. This means that we cannot rely on the existence of a purchase history, a prerequisite for collaborative and content-based filtering approaches. However, more detailed and structured *content* may be available, including technical and quality features.

[1] Some authors use the term "content-based filtering" instead of content-based recommendation.

Cambridge University Press
978-0-521-49336-9 - Recommender Systems: An Introduction
Dietmar Jannach, Markus Zanker, Alexander Felfernig and Gerhard Friedrich
Excerpt
More information

Take, for instance, a recommender system for digital cameras that should help the end user find a camera model that fits his or her particular requirements. Typical customers buy a new camera only once every few years, so the recommender system cannot construct a user profile or propose cameras that others liked, which – as a side note – would result in proposing only top-selling items.

Thus, a system is needed that exploits additional and means–end knowledge to generate recommendations. In such *knowledge-based approaches*, the recommender system typically makes use of additional, often manually provided, information about both the current user and the available items. *Constraint-based recommenders* are one example of such systems, which we will consider in our discussion of the different aspects of knowledge-based approaches. In the digital camera domain, a constraint-based system could use detailed knowledge about the features of the cameras, such as resolution, weight, or price. In addition, explicit constraints may be used to describe the context in which certain features are relevant for the customer, such as, for example, that a high-resolution camera is advantageous if the customer is interested in printing large pictures. Simply presenting products that fulfill a given set of requested features is not enough, as the aspect of personalization is missing, and every user (with the same set of requested features) will get the same set of recommendations. Thus, constraint-based recommender systems also need to maintain user profiles. In the digital camera scenario the system could, for instance, ask the user about the relative importance of features, such as whether resolution is more important than weight.

The other aspect covered in this chapter is "user interaction", as in many knowledge-based recommender systems, the user requirements must be elicited interactively. Considering the bookstore example and collaborative recommendation techniques once again, we see that users can interact with the system in only a limited number of ways. In fact, in many applications the only possibility for interaction is to rate the proposed items – for example, on a scale from 1 to 5 or in terms of a "like/dislike" statement. Think, however, about the digital camera recommender, which should also be able to serve first-time users. Therefore, more complex types of interaction are required to determine the user's needs and preferences, mostly because no purchase history is available that can be exploited. A simple approach would be to ask the user directly about his or her requirements, such as the maximum price, the minimum resolution, and so forth. Such an approach, however, not only requires detailed technical understanding of the item's features but also generates additional cognitive load in scenarios with a large number of item features. More elaborate approaches, therefore, try to implement more conversational interaction styles, in which the system tries to incrementally ascertain preferences within an interactive and personalized dialog.

Overall, the questions that are addressed in the chapter on knowledge-based recommender systems include the following:

- What kinds of domain knowledge can be represented in a knowledge base?
- What mechanisms can be used to select and rank the items based on the user's characteristics?
- How do we acquire the user profile in domains in which no purchase history is available, and how can we take the customer's explicit preferences into account?
- Which interaction patterns can be used in interactive recommender systems?
- Finally, in which dimensions can we personalize the dialog to maximize the precision of the preference elicitation process?

1.1.4 Hybrid approaches

We have already seen that the different approaches discussed so far have certain advantages and, of course, disadvantages depending on the problem setting. One obvious solution is to combine different techniques to generate better or more precise recommendations (we will discuss the question of what a "good" recommendation is later). If, for instance, community knowledge exists and detailed information about the individual items is available, a recommender system could be enhanced by hybridizing collaborative or social filtering with content-based techniques. In particular, such a design could be used to overcome the described ramp-up problems of pure collaborative approaches and rely on content analysis for new items or new users.

When combining different approaches within one recommender system, the following questions have to be answered and will be covered in the chapter on hybrid approaches:

- Which techniques can be combined, and what are the prerequisites for a given combination?
- Should proposals be calculated for two or more systems sequentially, or do other hybridization designs exist?
- How should the results of different techniques be weighted and can they be determined dynamically?

1.1.5 Explanations in recommender systems

Explanations aim to make a recommendation system's line of reasoning transparent to its users. This chapter outlines how the different recommendation strategies can be extended to provide reasons for the recommendations they propose to users. As knowledge-based recommendation systems have a long tradition of providing reasons to support their computed results, this chapter focuses on computing explanations for constraint-based and case-based recommender systems. In addition, efforts to explain collaborative filtering results are described to address the following topics:

- How can a recommender system explain its proposals while increasing the user's confidence in the system?
- How does the recommendation strategy affect the way recommendations can be explained?
- Can explanations be used to convince a user that the proposals made by the system are "fair" or unbiased?

1.1.6 Evaluating recommender systems

Research in recommender systems is strongly driven by the goal of improving the quality of the recommendations that are produced. The question that immediately arises is, of course, how can we actually measure the quality of the proposals made by a recommender system?

We start the chapter on evaluating recommender systems by reflecting on the general principles of empirical research and discuss the current state of practice in evaluating recommendation techniques. Based on the results of a small survey, we focus in particular on empirical evaluations on historical datasets and present different methodologies and metrics.

We also explore alternate evaluation approaches to address the necessity of, for instance, better capturing user experience or system goals. Evaluation approaches are classified into experimental, quasi-experimental, and nonexperimental research designs. Thus, the questions answered in the chapter include the following:

- Which research designs are applicable for evaluating recommender systems?
- How can recommender systems be evaluated using experiments on historical datasets?
- What metrics are applicable for different evaluation goals?

- What are the limitations of existing evaluation techniques, in particular when it comes to the conversational or business value aspects of recommender systems?

1.1.7 Case study

The final chapter of the book's first part is devoted to an experimental online evaluation that compares different personalized and impersonalized recommendation strategies on a mobile Internet portal. The purpose of this large-scale case study of a commercial recommender system is to address questions such as

- What is the business value of recommender systems?
- Do they help to increase sales or turn more visitors into buyers?
- Are there differences in the effectiveness of different recommendation algorithms? Which technique should be used in which situation?

1.2 Part II: Recent developments

Although many of the ideas and basic techniques that are used in today's recommender systems were developed more than a decade ago, the field is still an area of active research, in particular because the web itself has become an integral part of our everyday life and, at the same time, new technologies are constantly emerging.

In the second part of the book we will therefore focus – in the form of shorter chapters – on current research topics and recent advancements in the field. Among others, the following questions will be addressed:

- *Privacy and robustness.* How can we prevent malicious users from manipulating a recommender system – for instance, by inserting fake users or ratings into the system's database? How can we ensure the privacy of users?
- *Online consumer decision making.* Which consumer decision-making theories are the most relevant? Can the insights gained in traditional sales channels be transferred to the online channel, and in particular, how can this knowledge be encoded in a recommender system? Are there additional techniques or new models that can help us to improve the (business) value or acceptance of a recommendation service?
- *Recommender systems in the context of the social and the semantic web.* How can we exploit existing trust structures or social relationships between users to improve the recommender's accuracy? How do Semantic Web technologies

affect recommendation algorithms? What is the role of recommenders in Web 2.0?

- *Ubiquitous applications.* How do current technological advances, for instance in the area of mobile solutions, open new doors for building next-generation recommender systems? How do ubiquitous application domains affect recommendation algorithms – for instance, by placing more emphasis on contextual and situational parameters?

Cambridge University Press
978-0-521-49336-9 - Recommender Systems: An Introduction
Dietmar Jannach, Markus Zanker, Alexander Felfernig and Gerhard Friedrich
Excerpt
More information

PART I

Introduction to basic concepts