7
Evaluating recommender systems

In previous chapters we introduced a variety of different recommendation techniques and systems developed by researchers or already in use on commercial platforms. In the future, many new techniques will claim to improve prediction accuracy in specific settings or offer new ways for users to interact with each other, as in social networks and Web 2.0 platforms.

Therefore, methods for choosing the best technique based on the specifics of the application domain, identifying influential success factors behind different techniques, or comparing several techniques based on an optimality criterion are all required for effective evaluation research. Recommender systems have traditionally been evaluated using offline experiments that try to estimate the prediction error of recommendations using an existing dataset of transactions. Some point out the limitations of such methods, whereas others argue that the quality of a recommender system can never be directly measured because there are too many different objective functions. Nevertheless, the widespread use of recommender systems makes it crucial to develop methods to realistically and accurately assess their true performance and effect on the users. This chapter is therefore devoted to discussing existing evaluation approaches in the light of empirical research methods from both the natural and social sciences, as well as presenting different evaluation designs and measures that are well accepted in the research community.

7.1 Introduction

Recommender systems require that users interact with computer systems as well as with other users. Therefore, many methods used in social behavioral research are applicable when answering research questions such as *Do users find interactions with a recommender system useful?*, *Are they satisfied with the*

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

Table 7.1. *Basic characteristics of evaluation designs.*

Subject	Online customers, students, historical user sessions, simulated users, computers
Research method	Experimental, quasi-experimental, or nonexperimental
Setting	Real-world scenarios, lab

quality of the recommendations they receive?, *What drives people to contribute knowledge such as ratings and comments that boost the quality of a system's predictions?* or *What is it exactly that users like about receiving recommendations? Is it the degree of serendipity and novelty, or is it just the fact that they are spared from having to search for them?* Many more questions like these could be formulated and researched to evaluate whether a technical system is efficient with respect to a specified goal, such as increasing customer satisfaction or ensuring the economic success of an e-commerce platform. In addition, more technical aspects are relevant when evaluating recommendation systems, related, for instance, to a system's technical performance such as responsiveness to user requests, scalability, and peak load or reliability. Furthermore, goals related to the system's life cycle, such as ramp-up efforts, maintainability, and extensibility, as well as lowering the cost of ownership, can be thought of and are of interest for evaluation research.

Because of the diverse nature of possible evaluation exercises in the domain of recommendation systems, we start with very basic properties of research methodologies, as depicted in Table 7.1. The table differentiates empirical research based on the units that are subjected to research methods, such as people or computer hardware. Furthermore, it denotes the top-level taxonomy of empirical research methods, namely experimental and nonexperimental research, as well as the distinction between real-world and lab scenarios where evaluations can be conducted. Each of these meta-level concepts will be explained in more detail in the remainder of this chapter.

7.2 General properties of evaluation research

Empirical research itself has been subject to intense scrutiny from areas as diverse as philosophy and statistics (Pedhazur and Schmelkin 1991). Rather than repeating these principles, guidelines, and procedures here, we focus on some particular aspects and discuss them in the context of evaluating recommender systems. We begin with some general thoughts on rigor and validity of empirical evaluations. Finally, we briefly discuss some selected general criteria that must be kept in mind when evaluating recommendation applications with scientific rigor.

7.2.1 General remarks

Thoroughly describing the methodology, following a systematic procedure, and documenting the decisions made during the course of the evaluation exercise ensure that the research can be repeated and results verified. This answers the question of *how* research has been done. Furthermore, criteria such as the *validity, reliability*, and *sensibility* of the constructs used and measured relate to the subject matter of the research itself, questioning *what* is done. Notably, asking whether the right concepts are measured or whether the applied research design is valid is necessary.

Internal validity refers to the extent to which the effects observed are due to the controlled test conditions (e.g., the varying of a recommendation algorithm's parameters) instead of differences in the set of participants (predispositions) or uncontrolled/unknown external effects. In contrast, *External validity* refers to the extent to which results are generalizable to other user groups or situations (Pedhazur and Schmelkin 1991). When using these criteria to evaluate recommender systems, questions arise such as *Is it valid to exploit users' clicks on pages displaying details of an item as an indicator of their opinion about an item?* External validity examines, for instance, whether the evaluated recommendation scenario is representative of real-world situations in which the same mechanism and user interface of the technique would be used, and whether the findings of the evaluation exercise are transferrable to them. For example, will an increase in users' purchase rate of recommended items because of a new hybrid computation mechanism also be observable when the system is put to the field? *Reliability* is another postulate of rigorous empirical work, requiring the absence of inconsistencies and errors in the data and measurements. Finally, *sensibility* necessitates that different evaluations of observed aspects are also reflected in a difference in measured numbers.

Furthermore, issues surrounding research findings include not only their statistical significance but also information about the size of their effect and thus their significance with respect to the potential impact on real-world scenarios. For instance, what is the impact of a 10 percent increase in the accuracy of predicted ratings? Will this lead to a measurable increase in customer loyalty and lower churn rates of an e-commerce platform? Unfortunately, based on the current state of practice, not all these fundamental questions can be answered, but some guidance for designing evaluations is available, and researchers are urged to critically reflect on their own work and on the work of others.

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

7.2.2 Subjects of evaluation design

People are typically the subjects of sociobehavioral research studies – that is, the focus of observers. Obviously, in recommender systems research, the populations of interest are primarily specific subgroups such as online customers, web users, or students who receive adaptive and personalized item suggestions.

An experimental setup that is widespread in computer science and particularly, for instance, in subfields such as machine learning (ML) or information retrieval (IR) is datasets with synthetic or historical user interaction data. The basic idea is to have a collection of user profiles containing preference information such as ratings, purchase transactions, or click-through data that can be split into training and testing partitions. Algorithms then exploit the training data to make predictions with the hidden testing partition. The obvious advantage of this approach is that it allows the performance of different algorithms to be compared against each other. Simulating a dataset comes with the advantage that parameters such as distribution of user properties, overall size, or rating sparsity can be defined in advance and the test bed perfectly matches these initial requirements. However, there is significant risk that synthetic datasets are biased toward the design of a specific algorithm and that they therefore treat other algorithms unfairly. For this reason synthetic datasets are advisable only to test recommendation methods for obvious flaws or to measure technical performance criteria such as average computation times – that is, the computer itself becomes subject of the evaluation rather than users.

Natural datasets include historical interaction records of real users. They can be categorized based on the type of user actions recorded. For example, the most prominent datasets from the movie domain contain explicit user ratings on a multipoint Likert scale. On the other hand, datasets that are extracted from web server logs consist of implicit user feedback, such as purchases or add-to-basket actions. The sparsity of a dataset is derived from the ratio of empty and total entries in the user–item matrix and is computed as follows:

$$sparsity = 1 - \frac{|R|}{|I| \cdot |U|} \tag{7.1}$$

where

$$R = ratings$$

$$I = items$$

$$U = users$$

In Table 7.2 an incomplete list of popular datasets, along with their size characteristics, is given. The well-known MovieLens dataset was derived from

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

Table 7.2. *Popular data sets.*

Name	Domain	Users	Items	Ratings	Sparsity
BX	Books	278,858	271,379	1,149,780	0.9999
EachMovie	Movies	72,916	1,628	2,811,983	0.9763
Entree	Restaurants	50,672	4,160	N/A	N/A
Jester	Jokes	73,421	101	4.1M	0.4471
MovieLens 100K	Movies	967	4,700	100K	0.978
MovieLens 1M	Movies	6,040	3,900	1M	0.9575
MovieLens 10M	Movies	71,567	10,681	10M	0.9869
Netflix	Movies	480K	18K	100M	0.9999
Ta-Feng	Retail	32,266	N/A	800K	N/A

a movie recommendation platform developed and maintained by one of the pioneers in the field, the GroupLens research group[1] at the University of Minnesota. The EachMovie dataset was published by HP/Compaq and, despite not being publicly available for download since 2004, has still been used by researchers since. One additional movie dataset that has recently been made public is Netflix.[2] Published in conjunction with the Netflix Prize,[3] the company promised $1 million for the first team to provide a 10 percent improvement in prediction accuracy compared with its in-house recommender system. This competition stimulated much research in this direction. Finally, this threshold was reached by the team BellKor's Pragmatic Chaos in 2009. None of the aforementioned movie datasets contain item descriptions such as the movies' plots, actors, or directors. Instead, if algorithms require this additional content information, it is usually extracted from online databases such as the Internet Movie Database – IMDB.[4]

The BX dataset was gathered from a community platform for book lovers and contains explicit and implicit ratings for a large number of books (Ziegler et al. 2005). In contrast, the rating data from the joke recommender Jester represents a very dense dataset with only a few different items (Goldberg et al. 2001). The Entree data collection contains historical sessions from a critique-based recommender, as discussed in Chapter 4. Finally, the Ta-Feng dataset provides a representative set of purchase transactions from the retail domain with a very

[1] See http://www.grouplens.org/.
[2] See http://archive.ics.uci.edu/ml/datasets/Netflix+Prize.
[3] See http://www.netflixprize.com/.
[4] See http://www.imdb.com/.

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

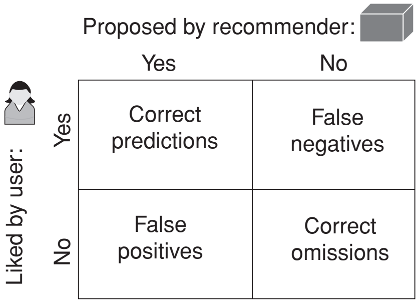


Figure 7.1. Types of errors.

low number of ratings per user. The dataset was exploited to evaluate the hybrid Poisson aspect modeling technique presented by Hsu et al. (2004).

Additional stimuli in the field come from social web platforms that either make their interaction data available to the public or allow researchers to extract this information. From CiteULike[5] and Bibsonomy,[6] tagging annotations on research papers are collected and public datasets can be downloaded from there. The social bookmarking platform del.icio.us[7] is another example for data from the social web that is used for evaluation research.

Nevertheless, the results of evaluating recommender systems using historical datasets cannot be compared directly to studies with real users and vice versa. Consider the classification scheme depicted in Figure 7.1. If an item that was proposed by the recommender is actually liked by a user, it is classified as a *correct prediction*. If a recommender is evaluated using historical user data, preference information is only known for those items that have been actually rated by the users. No assumptions can be made for all unrated items because users might not have been aware of the existence of these items. By default, these unknown item preferences are interpreted as disliked items and can therefore lead to false positives in evaluations – that is, the recommender is punished for recommending items that are not in the list of positively rated items of the historical user session, but that might have been welcomed by the actual user if they were recommended in a live interaction.

In contrast, when recommending items to real users, they can be asked to decide instantly if they like a proposed item. Therefore, both correct predictions and false positives can be determined in this setting. However, one cannot assess whether users would have liked items that were not proposed to them – that is,

[5] See http://www.citeulike.org/.
[6] See http://www.bibsonomy.org/.
[7] See http://www.delicious.com/.

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

the false negatives. Thus, one needs to be aware that evaluating recommender systems using either online users or historical data has some shortcomings. These shortcomings can be overcome only by providing a marketplace (i.e., the set of all recommendable items) that is completely transparent to users who, therefore, rate all items. However, in markets with several hundreds or even thousands of items, dense rating sets are both impossible and of questionable value, as no one would need recommendations if all items are already known by users beforehand.

7.2.3 Research methods

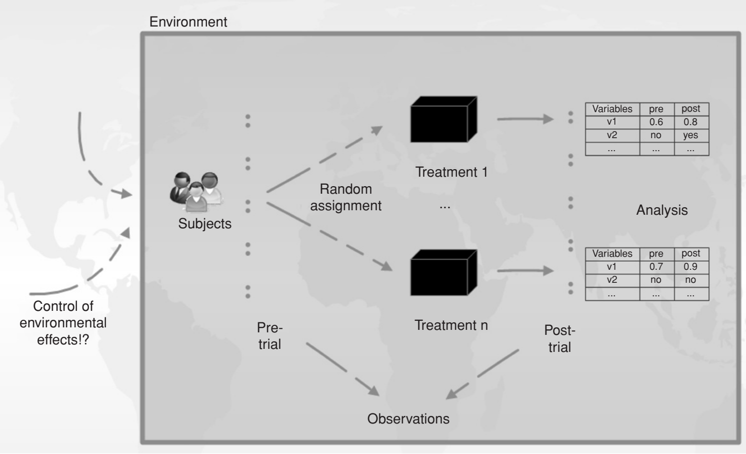
Defining the goals of research and identifying which aspects of the users or subjects of the scientific inquiry are relevant in the context of recommendation systems lie at the starting point of any evaluation. These observed or measured aspects are termed *variables* in empirical research; they can be assumed to be either independent or dependent. A few variables are always independent because of their nature – for example, gender, income, education, or personality traits – as they are, in principle, static throughout the course of the scientific inquiry. Further variables are independent if they are controlled by the evaluation design, such as the type of recommendation algorithm that is applied to users or the items that are recommended to them. Dependent variables are those that are assumed to be influenced by the independent variables – for instance, user satisfaction, perceived utility, or click-through rate can be measured.

In an *experimental research design*, one or more of the independent variables are manipulated to ascertain their impact on the dependent variables:

> An *experiment* is a study in which at least one variable is manipulated and units are randomly assigned to the different levels or categories of the manipulated variables (Pedhazur and Schmelkin 1991, page 251).

Figure 7.2 illustrates such an experiment design, in which subjects (i.e., units) are randomly assigned to different treatments – for instance, different recommendation algorithms. Thus, the type of algorithm would constitute the manipulated variable. The dependent variables (e.g., v_1 and v_2 in Figure 7.2) are measured before and after the treatment – for instance, with the help of a questionnaire or by implicitly observing user behavior. Environmental effects from outside the experiment design, such as a user's previous experience with recommendation systems or the product domain, also need to be controlled – for instance, by ensuring that only users that are sophisticated or novices in the

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press



Variables	pre	post
v1	0.6	0.8
v2	no	yes
...	...	...

Variables	pre	post
v1	0.7	0.9
v2	no	no
...	...	...

Figure 7.2. Example of experiment design.

product domain participate in the experiment (i.e., by elimination or inclusion) or by factorization (i.e., ensuring that sophisticated and novice users have an equal chance of being assigned to a treatment $1 \leq i \leq n$). For a deeper discussion on conducting live-user experiments and alternate experiment designs, the reader is referred to a textbook specifically focusing on empirical research (Pedhazur and Schmelkin 1991).

When experimenting offline with datasets, units (i.e., historical user sessions) do not need to be randomly assigned to different treatments. Instead, all algorithms can be evaluated on all users in the dataset. Although sequentially assigning real users to several treatments would lead to strongly biased results from repeated measurements (e.g., users might remember their initial answers), in offline experiments the historical user behavior will obviously remain static.

A *quasi-experimental design* distinguishes itself from a real experiment by its lacking random assignments of subjects to different treatments – in other words, subjects decide on their own about their treatment. This might introduce uncontrollable bias because subjects may make the decision based on unknown reasons. For instance, when comparing mortality rates between populations being treated in hospitals and those staying at home, it is obvious that higher mortality rates in hospitals do not allow us to conclude that these medical treatments are a threat to people's lives. However, when comparing purchase rates of e-commerce users who used a recommender system with the purchase rates of those who did not, a methodological flaw is less obvious. On one hand, there could be unknown reasons (i.e., uncontrolled variables) that let users who have a strong tendency to buy also use the recommender system, whereas

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

on the other hand, a higher purchase rate of recommender users could really be an indicator of the system's effectiveness. Therefore, the effectiveness of quasiexperimental designs is not undisputed and, as a consequence, their results must be interpreted with utmost circumspection, and conclusions need to be drawn very carefully (Pedhazur and Schmelkin 1991).

Nonexperimental designs include all other forms of quantitative research, as well as qualitative research. Quantitative research relies on numerical measurements of different aspects of objects, such as asking users different questions about the perceived utility of a recommendation application with answers on a seven-point Likert scale, requiring them to rate a recommended item or measuring the viewing time of different web pages. In contrast, qualitative research approaches would conduct interviews with open-ended questions, record think-aloud protocols when users interact with a web site, or employ focus group discussions to find out about users' motives for using a recommender system. For a more elaborate discussion of qualitative research designs, the reader is referred to Miles and Huberman (1994) and Creswell (2009).

One nonexperimental research design that is quite interesting in the context of evaluating recommender systems is *longitudinal research*, in which the entity under investigation is observed repeatedly as it evolves over time. Such a design allows criteria such as the impact of recommendations on the customer's lifetime value to be measured. Such research endeavors are very complex and costly to carry out, however, as they involve observing subjects over a long period of time. Zanker et al. (2006) conducted longitudinal research in which the sales records of an online store for the periods before and after introducing a recommendation system were analyzed and compared with each other. One of the most interesting results was that the top-seller list (i.e., the items that were most frequently sold) changed considerably and some items that were rarely purchased in the period before the introduction of the recommendation system became top-selling items afterward. Further analysis indicated that the increase in the number of pieces sold for these items correlated positively with the occurrence of these items in actual recommendations.

Cross-sectional research designs can also be very promising in the recommender systems domain, analyzing relations among variables that are simultaneously measured in different groups, allowing generalizable findings from different application domains to be identified.

Case studies (Stake 1995, Yin 2002) represent an additional way of collecting and analyzing empirical evidence that can be applied to recommendation systems research when researchers are interested in more principled questions. They focus on answering research questions about how and why and combine whichever types of quantitative and qualitative methods necessary to investigate

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

contemporary phenomena in their real-life contexts. Therefore, to answer the question of how recommendation technology contributed to Amazon.com's becoming the world's largest book retailer would require a case study research design.

7.2.4 Evaluation settings

The evaluation setting is another basic characteristic of evaluation research. In principle, we can differentiate between *lab studies* and *field studies*. A lab situation is created expressly for the purpose of the study, whereas a field study is conducted in an preexisting real-world environment.

Lab situations come with the major advantage that extraneous variables can be controlled more easy by selecting study participants. However, doubts may exist about study participants who are motivated to participate primarily by money or prizes. Therefore, a study needs to be carefully designed to ensure that participants behave as they would in a real-world environment. In contrast, research that is conducted in the field comes with the advantage that users are intrinsically motivated to use a system or even spend their own money when trusting a recommendation system and purchasing the item that was proposed to them. Nevertheless, researchers tend to have little control over the system, as the commercial interests of the platform operator usually prevail. Typically, one has little choice over the different settings, as other factors, such as the availability of data or real-world platforms, will influence the decision.

7.3 Popular evaluation designs

Up to now, experiment designs that evaluate different algorithm variants on historical user ratings derived from the movie domain form by far the most popular evaluation design and state of practice. To substantiate this claim, we conducted a survey of all research articles that appeared on the topic of recommender systems in the reputed publication *ACM Transactions on Information Systems* (*ACM TOIS*) over a period of five years (2004–2008). Twelve articles appeared, as listed in Table 7.3 in chronological order. The first of them has been the most influential with respect to evaluating recommender systems and, in particular, collaborative filtering systems, as it focuses on comparing different accuracy measures for collaborative filtering algorithm variants. As can be seen from Table 7.3, in three-quarters of these articles, offline experiments on historical user sessions were conducted, and more than half of authors chose movie recommendations as their application domain. Adomavicius et al. (2005)

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

Table 7.3. *Evaluation designs in ACM TOIS 2004–2008.*

Reference	Approach	Goal (Measures)	Domain
Herlocker et al. (2004)	Offline experiments	Accuracy (MAE,[a] ROC[b] curve)	ML[c]
Middleton et al. (2004)	Experimental user study	Accuracy (hit rate)	Web pages, e-mails
Hofmann (2004)	Offline experiments	Accuracy (MAE, RMSE[d])	EM[e]
Huang et al. (2004)	Offline experiments	Accuracy (Precision, Recall, F1)	Bookstore
Deshpande and Karypis (2004)	Offline experiments	Accuracy (hit rate, rank metric)	EM, ML, mail order purchases
Miller et al. (2004)	Offline experiments	Accuracy (MAE, Recall), catalog coverage	ML
Adomavicius et al. (2005)	Offline experiments	Accuracy (Precision, Recall, F1)	Movie ratings
Wei et al. (2005)	Offline experiments with simulated users	Marketplace efficiency	Synthetic datasets
Lee et al. (2006)	Qualitative user study	Usage analysis and wish list for improved features	Broadcast news
Ma et al. (2007)	Experimental user study	Search efficiency (mean log search time, questionnaire)	Web pages
Im and Hars (2007)	Offline experiments	Accuracy (MAE–NMAE[f])	Movie ratings, research papers, BX-Books, EM
Wang et al. (2008)	Offline experiments	Accuracy (MAE)	EM, ML

[a] MAE: mean absolute error.

[b] ROC: receiver operating characteristic.

[c] ML: MovieLens dataset.

[d] RMSE: root mean square error.

[e] EM: EachMovie dataset.

[f] NMAE: normalized mean absolute error.

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

and Im and Hars (2007), collected these ratings from specifically designed platforms that also collected situational parameters such as the occasion in which the movie was consumed. The others worked on the then publicly available datasets MovieLens and EachMovie (see Subsection 7.2.2). Experimental studies involving live users (under lab conditions) were done by Middleton et al. (2004) and Ma et al. (2007), who measured the share of clickthroughs from overall recommended items and search efficiency with respect to search time. A qualitative research design was employed only by Lee et al. (2006), who evaluated an automated content-based TV news delivery service and explored the usage habits of a group of sixteen users. The study consisted of pre- and post-trial questionnaires, diaries from each user during the one-month trial, and interaction data. The outcome of the study was a wish list for feature improvements and more insights into the usage patterns of the tool – for example, that users mainly accessed the section on latest news and used the system's search functionality only very rarely.

7.4 Evaluation on historical datasets

Because of the paramount importance of experimental evaluations on historical datasets for recommender systems research, we focus in this section on how they are carried out. Based on a small example, we discuss popular methodologies and metrics, as well as the interpretation of results.

7.4.1 Methodology

For illustrative purposes, we assume that an arbitrary historical user profile contains ten fictitious movie ratings, as depicted in Table 7.4. When evaluating a recommendation method, a group of user profiles is normally used as input to train the algorithm and build a model that allows the system to compute recommendations efficiently at run time. A second group of user profiles, different from the first, is required for measuring or testing the algorithm's performance. To ensure that the measurements are reliable and not biased by some user profiles, the random split, model building, and evaluation steps are repeated several times to determine average results. *N-fold cross-validation* is a stratified random selection technique in which one of N disjunct fractions of the user profiles of size $\frac{1}{N}$ is repeatedly selected and used for evaluation, leaving the remaining $\frac{N-1}{N}$ user profiles to be exploited for building the algorithm's model. Consequently, each user profile is used exactly once to evaluate the algorithm and $N - 1$ times to contribute to the algorithm's model building step. In the

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

Table 7.4. *Example user ratings.*

Row	UserID	MovieID	Rating
1	234	110	5
2	234	151	5
3	234	260	3
4	234	376	5
5	234	539	4[a]
6	234	590	5
7	234	649	1
8	234	719	5[a]
9	234	734	3
10	234	736	2

[a] Randomly selected ratings for testing.

extreme case, in which N is equal to the total number of user profiles, the splitting method is termed *leave one out*. From the computational point of view this method is the most costly, as the model has to be rebuilt for each user. At the same time, however, it allows the algorithm to exploit the maximum amount of community data for learning. Therefore, in situations in which the user base is only very small – a few hundred different profiles – a leave-one-out strategy can make sense to use as much data as possible for learning a model.

In addition, during the testing step, the user profile must be split into two groups, namely, user ratings to train and/or input the algorithm (i.e., determining similar peers in case of collaborative filtering) and to evaluate the predictions. In our example, we assume that the fifth and the eighth rows (see footnote in Table 7.4) of user number 234 have been randomly selected for testing – that is, they constitute the *testing set* and the other eight rows are part of the *training* or *learning set*.

One of two popular variants may be applied to split the rating base of the currently evaluated user into training and testing partitions. The *all but N* method assigns a fixed number N to the testing set of each evaluated user, whereas the *given N* method sets the size of the training partition to N elements. Both methods have their strengths, especially when one varies N to evaluate the sensitivity of an algorithm with respect to different testing or training set sizes. A fixed training set size has the advantage that the algorithm has the same amount of information from each tested user, which is advantageous when measuring the predictive accuracy. In contrast, fixed testing set sizes establish equal conditions for each user when applying classification metrics.

When evaluating algorithms, such as a simple nonpersonalized recommendation mechanism, that suggest the same set of popular items to every user and

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

therefore do not need to identify similar peers or do not require a set of liked items to query the product catalog for similar instances, the evaluation method is effectively *Given 0* – that is, the training set of past ratings of evaluated users is empty and all ratings can be used for testing the algorithm's predictions. Such an evaluation approach also applies to the constraint-based recommendation paradigm (see Chapter 4).

Based on the scale of historical ratings available – that is, unary (purchase) transactions or ratings on Likert scales – an evaluation can examine the prediction or the classification capability of a recommender system. A *prediction task* is to compute a missing rating in the user/item matrix. The prediction task requires Likert scale ratings that have been explicitly acquired from users, such as the ones specified in Table 7.4. The *classification task* selects a ranked list of n items (i.e., the recommendation set) that are deemed to be relevant for the user. The recommendation set typically contains between three and ten items, as users typically tend not to want to scroll through longer lists. To evaluate the accuracy of an algorithm's classifications, Likert scale ratings need to be transformed into relevant and not-relevant items – for instance, classifying only items rated 4 and above as relevant. This leads us back to the discussion in Section 7.2.2 on how to treat items with unknown rating values. The current state of practice assumes that these items are nonrelevant, and therefore evaluation measures reward algorithms only for recommending relevant items from the testing set, as is explained in the next subsection.

7.4.2 Metrics

Herlocker et al. (2004) provide a comprehensive discussion of accuracy metrics together with alternate evaluation criteria, which is highly recommended for reading. We therefore focus only on the most common measures for evaluations based on historical datasets.

Accuracy of predictions. When evaluating the ability of a system to correctly predict a user's preference for a specific item, mean absolute error (MAE) is undisputedly the most popular measure, as confirmed by the outcome of the small survey in Section 7.3. The MAE metric was already discussed in the context of collaborative filtering (see Chapter 2) and when dynamizing a weighted hybridization strategy (Chapter 5). Nevertheless, we restate its computation scheme for reasons of completeness.

$$MAE = \frac{\sum_{u \in U} \sum_{i \in testset_u} |rec(u,i) - r_{u,i}|}{\sum_{u \in U} |testset_u|} \tag{7.2}$$

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

MAE computes the average deviation between computed recommendation scores ($rec(u, i)$) and actual rating values ($r_{u,i}$) for all evaluated users $u \in U$ and all items in their testing sets ($testset_u$). Alternatively, some authors, such as Sarwar et al. (2001), compute the root mean square error (RMSE) to put more emphasis on larger deviations or, similar to Goldberg et al. (2001), create a normalized MAE (NMAE) with respect to the range of rating values.

$$NMAE = \frac{MAE}{r_{max} - r_{min}} \tag{7.3}$$

r_{max} and r_{min} stand for the highest and lowest rating values to normalize $NMAE$ to the interval $0 \ldots 1$. Consequently, the normalized deviations should be comparable across different application scenarios using different rating scales. Im and Hars (2007), for example, used NMAE to compare the effectiveness of collaborative filtering across different domains.

Accuracy of classifications. The purpose of a classification task in the context of product recommendation is to identify the n most relevant items for a given user. *Precision* and *Recall* are the two best-known classification metrics; they are also used for measuring the quality of information retrieval tasks in general. Both are computed as fractions of $hits_u$, the number of correctly recommended relevant items for user u. The Precision metric (P) relates the number of hits to the total number of recommended items ($|recset_u|$).

$$P_u = \frac{|hits_u|}{|recset_u|} \tag{7.4}$$

In contrast, the Recall (R) computes the ratio of hits to the theoretical maximum number of hits owing to the testing set size ($|testset_u|$).

$$R_u = \frac{|hits_u|}{|testset_u|} \tag{7.5}$$

According to McLaughlin and Herlocker (2004), measuring an algorithm's performance based on Precision and Recall reflects the real user experience better than MAE does because, in most cases, users actually receive ranked lists from a recommender instead of predictions for ratings of specific items. They determined that algorithms that were quite successful in predicting MAEs for rated items produced unsatisfactory results when analyzing their top-ranked items. Carenini and Sharma (2004a) also argue that MAE is not a good indicator from a theoretical perspective, as all deviations are equally weighted. From the user's perspective, however, the only fact that counts is whether an item is recommended.

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

Assume that a recommender computes the following item/rating -tuples for user 234, whose rating profile is presented in Table 7.4:

$$recset_{234} = \{(912, 4.8), (47, 4.5), (263, 4.4), (\mathbf{539}, \mathbf{4.1}), (348, 4), \ldots, (\mathbf{719}, \mathbf{3.8})\}$$

Although only a single item from the user's test set is recommended among the top five, an MAE-based evaluation would give favorable results, as the absolute error is on average, 0.65. If the evaluation considered only the top three ranked items, however, Precision and Recall would be 0, and if the recommendation set is changed to contain only the five highest ranked items, $P_{234} = \frac{1}{5}$ and $R_{234} = \frac{1}{2}$.

By increasing the size of a recommendation set, the tradeoff between Precision and Recall metrics can be observed. Recall will typically improve as the chance of hitting more elements from the test set increases with recommendation set size, at the expense of lower Precision. For instance, if item 719 was recommended only on the twentieth and last position of the recommendation list to user 234, Recall would jump to 100 percent, but Precision would drop to 10 percent.

Consequently, the F1 metric is used to produce evaluation results that are more universally comparable:

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{7.6}$$

The $F1$ metric effectively averages Precision and Recall with bias toward the weaker value. Comparative studies on commercial datasets using P, F, and $F1$ have, for example, been conducted by Sarwar et al. (2000b) and Zanker et al. (2007).

Some argue, however, that a classification measure should reflect the proportion of users for which at least one item from the user's test profile is recommended. In other words, the hit rate should be defined as

$$hitrate_u = \begin{cases} 1 & : \text{if } hits_u > 0 \\ 0 & : \text{else} \end{cases} \tag{7.7}$$

Deshpande and Karypis (2004) used this measure to compare their item-based collaborative filtering variant with a user-based one, whereas O'Sullivan et al. (2004) employed it for measuring the quality of TV program guides. Nguyen and Ricci (2007b) assessed different algorithm variants for a mobile critique-based recommender, also based on hit rate. Interestingly, they presented a simulation model that allows one to evaluate historical critiquing sessions by replaying the query input. The logs were derived from user studies on the mobile recommendation application presented by Nguyen and Ricci (2007a).

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

Accuracy of ranks. Rank scores extend the results of classification metrics with a finer level of granularity. They differentiate between successful hits by also taking their relative position in recommendation lists into account. Breese et al. (1998) propose a metric that assumes decreasing utilities based on items' rank. The parameter α sets the half-life of utilities, which means that a successful hit at the first position of the recommendation list has twice as much utility to the user than a hit at the $\alpha + 1$ rank. The rationale behind this weighting is that later positions have a higher chance of being overlooked by the user, even though they might be useful recommendations.

$$rankscore_u = \sum_{i \in hits_u} \frac{1}{2^{\frac{rank(i)-1}{\alpha}}} \tag{7.8}$$

$$rankscore_u^{max} = \sum_{i \in testset_u} \frac{1}{2^{\frac{idx(i)-1}{\alpha}}} \tag{7.9}$$

$$rankscore'_u = \frac{rankscore_u}{rankscore_u^{max}} \tag{7.10}$$

The function $rank(i)$ returns the position of item i in the user's recommendation list. $Rankscore_u^{max}$ is required for normalization and returns the maximum achievable score if all the items in the user's test set were assigned to the lowest possible ranks, i.e. ranked according to a bijective index function $idx()$ assigning values $1, \ldots, |testset_u|$ to the test set items. Thus, for our example user 234, with twenty recommendations and hits on the fourth and twentieth positions, the half-life utility rank score would be computed as:

$$rankscore_{234} = \frac{1}{2^{\frac{4-1}{10}}} + \frac{1}{2^{\frac{20-1}{10}}} = 1.08$$

$$rankscore_{234}^{max} = \frac{1}{2^{\frac{1-1}{10}}} + \frac{1}{2^{\frac{2-1}{10}}} = 1.93$$

$$rankscore'_{234} = \frac{1.08}{1.93} = 0.56$$

Another very simple rank accuracy measure is the *lift index*, first proposed by Ling and Li (1998). It assumes that the ranked list is divided into ten equal deciles and counts the number of hits in each decile as $S_{1,u}, S_{2,u}, \ldots, S_{10,u}$, where $\sum_{i=1}^{10} S_i = hits_u$.

$$liftindex_u = \begin{cases} \frac{1 \cdot S_{1,u} + 0.9 \cdot S_{2,u} + \cdots + 0.1 \cdot S_{10,u}}{\sum_{i=1}^{10} S_{i,u}} & : \text{ if } hits_u > 0 \\ 0 & : \text{ else} \end{cases} \tag{7.11}$$

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

Compared with the rank score of Breese et al. (1998), the lift index attributes even less weight to successful hits in higher ranks. Consequently, for the example user 234, the lift index is calculated as follows:

$$liftindex_{234} = \frac{0.9 \cdot 1 + 0.1 \cdot 1}{2} = 0.5 \tag{7.12}$$

Finally, an example of using the lift index on recommendation results is presented by Hsu et al. (2004). For a discussion on additional rank accuracy metrics, we refer readers to Herlocker et al. (2004).

Additional metrics. One metric that allows evaluators to compare different techniques based on their capability to compute recommendations for a large share of the population is user coverage (*Ucov*). It is of particular interest when one wants to analyze an algorithm's behavior with respect to new users with few known ratings.

$$Ucov = \frac{\sum_{u \in U} \rho_u}{|U|} \tag{7.13}$$

$$\rho_u = \begin{cases} 1 & : \text{if } |recset_u| > 0 \\ 0 & : \text{else} \end{cases} \tag{7.14}$$

It measures the share of users to whom nonempty recommendation lists can be provided. Obviously, it is sensible to measure user coverage only in conjunction with an accuracy metric, as otherwise recommending arbitrary items to all users would be considered as an acceptable strategy.

A similar coverage metric can be computed on the item universe.

$$Ccov = \frac{|\bigcup_{u \in U} recset_u|}{|I|} \tag{7.15}$$

Catalog coverage (*Ccov*) reflects the total share of items that are recommended to a user in all sessions (Herlocker et al. 2004) and can be used as an initial indication for the diversity of an algorithm's recommendations.

However, Ziegler et al. (2005) propose a more elaborate measure of the diversity of recommendation lists, termed *intra-list similarity* (ILS).

$$ILS_u = \frac{\sum_{i \in recset_u} \sum_{j \in recset_u, i \neq j} sim(i, j)}{2} \tag{7.16}$$

For a given similarity function *sim*(i, j) that computes the similarity between two recommended items, ILS aggregates the pairwise proximity between any two items in the recommendation list. ILS is defined to be invariant for all permutations of the recommendation list, and lower scores signify a higher

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

diversity. Ziegler et al. (2005) employed this metric to compare a topic diversification algorithm on the BX books dataset.

7.4.3 Analysis of results

Having applied different metrics as part of an experimental study, one must question whether the differences are statistically meaningful or solely due to chance. A standard procedure for checking the significance of two deviating mean metrics is the application of a pairwise analysis of variance (ANOVA). The different algorithm variants constitute the independent categorical variable that was manipulated as part of the experiment. However, the null hypothesis H_0 states that the observed differences have been due to chance. If the outcome of the test statistics rejects H_0 with some probability of error – typically $p \leq .05$ – significance of findings can be reported. For a more detailed discussion of the application of test statistics readers are referred to Pedhazur and Schmelkin (1991); textbooks on statistics, as well as articles discussing the application of statistical procedures in empirical evaluation research, such as Demšar (2006) or Garcìa and Herrera (2008).

In a second step, the question as to whether the observed difference is of practical importance must be asked. When contemplating the substantive significance of a fictitious finding, like a 5 percent increase in recommendation list diversity caused by an algorithm modification, statistics cannot help. Instead, additional – and more complex – research is required to find out whether users are able to notice this increase in diversity and whether they appreciate it. The effect of higher recommendation list diversity on customer satisfaction or actual purchase rates must be evaluated, a task that can be performed not by experimenting with historical datasets but rather by conducting real user studies. The next section will provide some examples of these.

7.5 Alternate evaluation designs

As outlined in the previous section, recommender systems are traditionally evaluated using offline experiments to try to estimate the prediction error of the recommendations based on historical user records. Although the availability of well-known datasets such as MovieLens, EachMovie, or Netflix has stimulated the interest of researchers in the field, it has also narrowed their creativity, as newly developed techniques tend to be biased toward what can be readily evaluated with available resources. In this section we therefore refer to selected examples of evaluation exercises on recommender systems that adopt alternate

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

evaluation designs and do not experiment on historical datasets. Furthermore, we structure our discussion according to the taxonomy of research designs presented in Section 7.2.3.

7.5.1 Experimental research designs

User studies use live user interaction sessions to examine the acceptance or rejection of different hypotheses. Felfernig and Gula (2006) conducted an experimental user study to evaluate the impact of different conversational recommender system functions, such as explanations, proposed repair actions, or product comparisons. The study, involving 116 participants, randomly assigned users to different variants of the recommender system and applied pre- and post-trial surveys to identify the effect of user characteristics such as the level of domain knowledge, the user's trust in the system, or the perceived competence of the recommender. The results show that study participants appreciate particular functionality, such as explanations or the opportunity to compare products, as it tends to increase their perceived level of knowledge in the domain and their trust in the system's recommendations. A similar study design was applied by Teppan and Felfernig (2009b), who reported on a line of research investigating the effectiveness of psychological theories in explaining users' behavior in online choice situations; this will be examined in more detail in Chapter 10.

An experimental user study was also conducted by Celma and Herrera (2008), who were interested in comparing different recommendation variants with respect to their novelty as perceived by users in the music domain. One interesting aspect of this work is that it combines an item-centric network analysis of track history with a user-centric study to explore novelty criteria to provide recommendations from several perspectives. An intriguing finding of this study is that both collaborative filtering and a content-based music recommender did well in recommending familiar items to the users. However, the content-based recommender was more successful in identifying music from the long tail of an item catalog ranked by popularity (i.e., the less frequently accessed items) that would be considered novel by the participants. As collaborative filtering focuses on identifying items from similar peers, the recommended items from the long tail are already familiar to the music enthusiasts, whereas content-based music recommendation promises a higher chance to hit interesting similar items in different portions of the long tail, according to this study.

Pu et al. (2008) compared the task completion times of users interacting with two different critiquing-based search interfaces. They employed a within-subjects experiment procedure, in which all twenty-two participants

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

were required to interact with both interfaces. This is opposed to a between-subjects test, in which users are randomly assigned to one interface variant. However, to counterbalance bias from carryover effects from evaluating the first interface prior to the second, the order of interfaces was alternated every two consecutive users. Because of the small number of subjects, only a few differences in measurements were statistically significant; nevertheless, the goal of this study, namely, exploring the support for tradeoff decisions of different critiquing-based recommendation interfaces, is of great interest.

In Chapter 8, an online evaluation exercise with real users is described as a practical reference. It employs a between-subjects experiment design in which users are randomly assigned to a specific personalized or impersonalized recommendation algorithm variant and online conversion is measured. This type of online experiment is also known as *A/B testing*.

7.5.2 Quasi-experimental research designs

A quasi-experimental evaluation of a knowledge-based recommender in the tourism domain was conducted to examine conversion rates – that is, the share of users who subsequently booked products (Zanker et al. 2008 and Jannach et al. 2009). The study strongly confirmed that users who interacted with the interactive travel advisor were more than twice as likely to issue a booking request than those who did not. Furthermore, an interesting cultural difference between Italian- and German-speaking users was detected, namely that Italian users were twice as likely to use interactive search tools such as the travel recommender.

7.5.3 Nonexperimental research designs

Swearingen and Sinha (2001) investigated the human-computer interaction (HCI) perspective when evaluating recommender systems, adopting a mixed approach that included quantitative and qualitative research methods. The subjects were observed while they interacted with several commercial recommendation systems, such as Amazon.com. Afterward they completed a satisfaction and usability questionnaire and were interviewed with the aim of identifying factors that can be used to predict the perceived usefulness of a recommendation system to derive design suggestions for good practice from an HCI perspective. Results of that study included that receiving very novel and unexpected items is welcomed by users and that information on how recommendations are derived by the system should be given.

Experiences from fielded applications are described by Felfernig et al. (2006–07). The authors used a nonexperimental quantitative research design in

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

which they surveyed actual users from two commercial recommender systems in the domains of financial services and electronic consumer goods. In the latter domain, a conversational recommender for digital cameras was fielded. Based on users' replies to an online questionnaire, the hypothesis that interactive sales recommenders help users to better orient themselves when being confronted with large sets of choices was also confirmed. In the financial services domain, the installation of constraint-based recommenders was shown to support sales agents during their interaction with prospective clients. Empirical surveys determined that the time savings achieved by the sales representatives while interacting with clients are a big advantage, which, in turn, allows sales staff to identify additional sales opportunities (Felfernig et al. 2006–07).

Another interesting evaluation exercise with a nonexperimental quantitative design is to compare predictions made by a recommendation system with those made by traditional human advisors. Krishnan et al. (2008) conducted such a study and compared the MovieLens recommender system with human subjects. The results of their user study, involving fifty research subjects, indicated that the MovieLens recommender typically produced more precise predictions (based on MAE) than the group of humans, despite the fact that only experienced MovieLens users with long rating records were invited to participate in the survey. However, a subgroup of the human recommenders (i.e., research subjects) produced consistently better results than the employed system, which could, in turn, be used to further improve the algorithm's ability to mimic the human subjects' problem-solving behavior. An additional aspect of this specific evaluation design is that it supports the credibility of the system in the eyes of its users, as it demonstrates its ability to provide better predictions than human experts.

7.6 Summary

After reflecting on the general principles of empirical research, this chapter presented the current state of practice in evaluating recommendation techniques. We discussed the meta-level characteristics of different research designs – namely, subjects, research method, and setting – and consulted authoritative literature for best research practices.

Furthermore, a small survey of highly reputed publications on recommendation systems in the *ACM TOIS* was presented, which gave an overview of research designs commonly used in practice. As a result, we focused in particular on how to perform empirical evaluations on historical datasets and discussed different methodologies and metrics for measuring the accuracy or coverage of recommendations.

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press

From a technical point of view, measuring the accuracy of predictions is a well-accepted evaluation goal, but other aspects that may potentially affect the overall effectiveness of a recommendation system remain largely underdeveloped. Therefore, Section 7.5 presented several examples of evaluation studies that were based not on historical datasets but rather on real user studies. They were grouped according the classification scheme presented in Section 7.2.3, namely, into experimental, quasi-experimental, and nonexperimental research methods. Although the works discussed in Section 7.5 do not cover the complete range of study designs that have been explored so far, this selection can undoubtedly serve as a helpful reference when designing new evaluation exercises.

7.7 Bibliographical notes

Herlocker et al.'s (2004) article on evaluating collaborative filtering recommender systems is the authority in the field and is therefore one of the most frequently cited articles on recommendation systems. Since then, few works have appeared on the topic of evaluating recommender systems in general. One exception is the work of del Olmo and Gaudioso (2008), who criticize existing accuracy and ranking metrics for being overparticularized and propose a new category of metrics designed to measure the capacity of a recommender to make successful decisions. For this reason they present a new general framework for recommender systems that formalizes their recommendation process into several temporal stages. The essence of their approach is that a recommender system must be able to not only choose *which* items should be recommended, but also decide *when* and *how* recommendations should be shown to ensure that users are provided with useful and interesting recommendations in a timely manner. One interesting aspect of this article is its consideration of the interactivity of a recommender system, a property that has not been evaluated in existing approaches.

Furthermore, literature on empirical research in general, such as Pedhazur and Schmelkin (1991), on the interleaved quantitative processes of measurement, design, and analysis, or Creswell (2009), on mixed research designs focusing on qualitative methods, are also relevant when assessing alternate strategies for evaluating the quality and value of recommender systems.

https://doi.org/10.1017/CBO9780511763113.009 Published online by Cambridge University Press