DOI:10.1145/2891406

The future success of these systems depends on more than a Netflix challenge.

BY DIETMAR JANNACH, PAUL RESNICK, ALEXANDER TUZHILIN, AND MARKUS ZANKER

Recommender Systems—Beyond Matrix Completion

THE USE OF recommender systems has exploded over the last decade, making personalized recommendations ubiquitous online. Most of the major companies, including Google, Facebook, Twitter, LinkedIn, Netflix, Amazon, Microsoft, Yahoo!, eBay, Pandora, Spotify, and many others use recommender systems (RS) within their services.

These systems are used to recommend a whole range of items, including consumer products, movies, songs, friends, news articles, restaurants and various others. Recommender systems constitute a mission-critical technology in several companies. For example, Netflix reports that at least 75% of its downloads and rentals come from their RS, thus making it of strategic importance to the company.[a]

In some ways, the systems that produce these recommendations are remarkable. They incorporate a variety of signals about characteristics of the users and items, including people's explicit or implicit evaluations of items. The systems process these signals at a massive scale, often under real-time constraints. Most importantly, the recommendations are of significant quality on average. In empirical tests, people choose the suggested items far more often than they choose suggested items based on unpersonalized benchmark algorithms that are based on overall item popularity.

In other ways, the systems that produce these recommendations are sometimes remarkably bad. Occasionally, they make recommendations that are embarrassing for the system, such as recommending to a faculty member an introductory book from the "for dummies" series on a topic she is expert in. Or, they continue recommending items the user is no longer interested in. Shortcomings like these motivate ongoing research both in industry and academia, and recommender systems are a very active field of research today.

To provide an understanding of the state of the art of recommender systems, this article starts with a bit of history, culminating in the million-dollar Netflix challenge. That challenge led to a formulation of the recommendation problem as one of matrix completion: Given a matrix of users by items, with item ratings as cells, how well can an algorithm predict the values in some

» key insights

- **Recommender systems have become a ubiquitous part of our daily online user experience and support users in a variety of domains.**
- **Today, the scientific community operationalizes the research problem mainly on principles from information retrieval and machine learning, leading to a well-defined but narrow problem characterization.**
- **We briefly review the history of the field, report on the recent advances, and propose a more comprehensive research approach that considers both the consumer's and the provider's perspective.**

a http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html

IMAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

cells that are deliberately held out? However, algorithms with maximum accuracy at the matrix completion task are not sufficient to make the best recommendations in many practical settings. We will describe why, review some of the approaches that current research is taking to do better, and finally sketch ways of approaching the recommendation problem in a more comprehensive way in the future.

A Brief History

Many fields have contributed to recommender systems research, including information systems, information retrieval (IR), machine learning (ML), human-computer interaction (HCI), and even more distant disciplines like marketing and physics. The common starting point is that recommendations must be personalized or adapted to the user's situation, with different people typically getting different item suggestions. That implies maintaining some kind of user history or model of user interests.

Building user profiles: Information filtering roots. In many application domains, for example, in news recommendation, recommenders can be seen as classic information filtering (IF) systems that scan and filter text documents based on personal user preferences or interests. The idea of using a computer to filter a stream of incoming information according to the preferences of a user dates back to the 1960s, when first ideas were published under the term "selective dissemination of information."[17] Early systems used explicit keywords that were provided by the users to rank and filter documents, for example, based on keyword overlap counts. Later on, more elaborate techniques like weighted term vectors (for example, TF-IDF vectors) or more sophisticated document analysis methods like latent semantic indexing (LSI) were applied to represent documents, with corresponding representations of user interests stored as user models. Recommender systems based on these techniques are typically called "content-based filtering" approaches.

Leveraging the opinions of others. As early as 1982, then ACM president Peter J. Denning complained about "electronic (email) junk" and advocated the development of more intelligent systems that help to organize, prioritize, and filter the incoming

Recommendation as Matrix Completion

The recommendation problem viewed as a matrix completion problem as done in the Netflix Prize.

1. **Given a sparse matrix *M*, create a matrix *M′* by randomly hiding a subset H of the known ratings.**
2. **Predict the values (*H**) of the hidden ratings using *M′*.**
3. **Assess the difference between the predicted and the true ratings using the Root Mean Square Error (RMSE).**

RMSE: Given a vector of predictions *H of length n and the vector containing the true values *H*, the RMSE is computed as**

$$\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(H_i^* - H_i)^2}$$

M

Users/Items	I1	I2	I3	....	I100000
U1	4	3			2
U2		2	5		
U3	3	5			2
...					
U10000	2	2	5		

x Known ratings
Missing ratings

M′

Users/Items	I1	I2	I3	....	I100000
U1	4	3			2
U2		2	?		
U3	3	5			?
...					
U10000	2	?	5		

H

x Known ratings
Missing ratings
? Withheld ratings

***Common techniques.* The goal of *matrix factorization* techniques in RS is to determine a low-rank approximation of the user-item rating matrix by decomposing it into a product of (user and item) matrices of lower dimensionality (latent factors).**

The idea of *ensemble methods* is to combine multiple alternative machine learning models to obtain more accurate predictions.

streams of information.[8] One of his proposals included the idea to use "trusted authorities" that assess document quality; receivers would only read documents that surpass some defined quality level. In 1987, the "Information Lens" personal mail processing system was proposed.[25] The system was mainly based on manually defined filtering rules but the authors already envisaged a system where email receivers could endorse other people whose opinions they value. The number and strength of the endorsements would then prioritize incoming messages. The Tapestry email filtering system at Xerox PARC[14] adopted a similar approach of employing user-specified rules. It also introduced the idea that some readers could classify (rate) messages and other readers could access this information, which was called "collaborative filtering" (CF).

In 1994, Resnick et al.[33] presented the GroupLens system, which continued the ideas of Tapestry and introduced a system component, the "Better Bit Bureau," which made automated predictions about which items people would like based on a nearest-neighbor scheme. Other research groups working independently developed similar ideas.[18,36] The idea was "in the air" that opinions of other people were a valuable resource and the race was on to turn the idea into practical results.

It works in e-commerce! In 1999, only five years after the first CF methods were proposed, Shafer et al.[35] reported on several industrial applications of recommender systems technology in e-commerce; for example, for the recommendation of books, movies, or music. Amazon.com is mentioned as one of the early adopters of recommendation systems. In 2003, Linden et al.[23] report that Amazon's use of item-to-item CF techniques as a targeted marketing tool had a huge impact on its business in terms of click-through and conversion rates. In the same report, several challenges in practical environments were discussed, in particular the problem of scalability and the need to create recommendations in real time.

The matrix completion problem. By this time, the research community had developed some standard concepts and terminology. The core element is the user-item rating matrix, as illustrated in the upper part of the accompanying sidebar. Rows represent users, columns represent items, and each cell represents a user's subjective preference for an item, determined based on an explicit report (for example, 1–5 stars) or based on user behavior (for example, clicking, buying, or spending time on the item).

The user-item matrix is generally sparse: most users have not interacted with most items. One formulation of the recommender problem, then, is that of a matrix completion problem. That is, the problem is to predict what the missing cells will be, in other words, how will users rate items they haven't rated yet?

With that formulation, it was natural to apply and adapt machine-learning techniques from other problem settings, including various forms of clustering, classification, regression, or singular value decomposition.[3,4] Correspondingly, the community adopted evaluation measures from the IR and ML fields like precision/recall and the root mean squared error (RMSE) measure. These evaluation techniques withhold some of the known ratings, use the algorithm to predict ratings of those cells, and then compare the predicted ratings with the withheld ones. The availability of some common rating datasets, distributed by vendors and academic projects, enabled researchers to conduct bake-offs comparing the performance of alternative matrix filling algorithms against each other.

The Netflix Prize. Netflix, which saw the strategic value in improving its recommendations, supercharged the bake-off process for matrix completion algorithms in 2006. Netflix offered a million-dollar prize, a dataset for training, and an infrastructure for testing algorithms on withheld data. The training dataset included 100 million real customer ratings. The prize was for the first algorithm to

outperform Netflix's in-house system by 10% on RMSE (see the sidebar). Interest in this competition was huge. More than 5,000 teams registered for the competition and the prize was finally awarded in 2009. Substantial progress was made with respect to the application of ML approaches for the rating prediction task. In particular, various forms of matrix factorization as well as ensemble learning techniques were further developed in the course of the competition and proved to be highly successful.

Beyond Matrix Completion

At the conclusion of the Netflix Prize competition, it might have been plausible to think that recommender systems were a solved problem. After all, many very talented researchers had devoted themselves for an extended period of time to improve the prediction of withheld ratings. The returns on that effort seemed to be diminishing quite rapidly, with the final small improvements that were sufficient to win the prize coming from combining the efforts of many independent contestants.

However, it turns out that recommender systems are far from a solved problem. Here, we first give examples of why optimizing the prediction accuracy for held-out historical ratings might be insufficient or even misleading. Then we discuss selected quality factors of recommender systems not covered by the matrix completion task at all and give examples of recent research that goes beyond matrix completion.

Pitfalls of matrix completion setups. *Postdiction ≠ prediction.* Predicting held-out matrix entries is really predicting the past rather than the future. If the held-out rating entries are representative of the hidden rating entries, then the distinction does not matter. However, in many recommender settings, the held-out ratings are not representative of the missing ratings.

One reason is the missing ratings are generally not missing at random. Even for items that people have experienced, if rating requires any effort at all they are more likely to rate items that they love or hate rather than those that they feel lukewarm about. Moreover, people are more likely to try items they expect to like. For example, in one empirical study, researchers found that ratings of songs randomly assigned to users had a very different distribution than ratings of songs users had chosen to rate.[26]

As a result, algorithms that predict well on held-out ratings that users provided may predict poorly on a random set of items the user has not rated. This can mean that algorithms tuned to perform well on past ratings are not the best algorithms for recommending in the real world.[7]

In addition, the matrix completion problem setup is not suitable to assess the value of reminding users of items they have already purchased or consumed in the past. However, such repeated recommendations can be a desired functionality of recommenders, for example, in domains like music recommendation or the recommendation of consumables.

In the end, the standardized evaluation setup and the availability of public rating datasets made it attractive for researchers to focus on accuracy measures and the matrix completion setup and may have lured them away from investigating the value of other information sources and alternative ways of evaluating the utility of recommendations.

Today, a growing number of academic studies try to evaluate the performance of their methods using A/B tests on live customers in real industrial settings (for example, Dias et al,[9] Garcin et al.,[12] and Gorgoglione et al.[16]). This is a very positive trend that requires cooperation from a commercial vendor who may not agree to make data publicly available, thus making it difficult for results to be checked or reproduced by others.

Not all items and errors are equally important. RMSE, the evaluation metric used in the Netflix Prize, equally weights errors of prediction on all items. However, in most practical settings items with low predicted ratings are never shown to users, so it hardly matters if the correct prediction for those items is 1, 2, or 3 stars. Intuitively, it is more appropriate in these domains to optimize a ranking criterion that focuses on having the top items correct.

In recent years, a number of learning-to-rank approaches have been proposed in the literature to address this issue, which aim to optimize (proxies of) rank-based measures. When applying such IR measures in the recommendation domain, the problem remains that the "ground truth" (that is, whether or not an item is actually relevant for a user) is available only for a tiny fraction of the items. The results of an empirical evaluation can depend on how the items with unknown ground truth are treated when determining the accuracy metrics. In addition, the problem of items not missing at random also exists for learning-to-rank approaches and at least some of them exhibit a strong bias to recommend blockbusters to everyone, which might be of little value for the users.[19]

In some domains, like music recommendation, it is also important to avoid very "bad" recommendations as they can greatly impact the user's quality perception.[6,21] Omitting some "good" recommendations is not nearly so harmful, which would argue for risk-averse algorithm designs that mostly recommend items with a high average rating and low rating variance. Recommending only such generally liked, non-controversial items might however not be particularly helpful for some of the users.

System quality factors beyond accuracy. The Netflix Prize with its focus on accuracy has undoubtedly pushed recommender systems research forward. However, it has also partially overshadowed many other important challenges when building a recommender system and today even Netflix states "there are much better ways to help people find videos to watch than focusing only on those with a high predicted star rating."[15] Next, we give examples of quality factors other than single-item accuracy, review how recent research has approached these problems, and sketch open challenges.

Novelty, diversity, and other components of utility. Making good rating predictions for as-yet unrated items is almost never the ultimate goal. The true goal of providing recommendations is rather some combination of a certain value for the user and profit for the site. In some domains, user ratings may represent a general quality assessment but still not imply the item should be recommended. As an example, consider the problem of recommending restaurants to travelers. Most people dining at a Michelin-starred location may give it five stars, but budget travelers may be

annoyed to see it recommended. As a result, some researchers have tried to recommend based on a more encompassing model of utility. For the budget traveler, that utility or "economic value" might increase with predicted rating but decrease with cost.[13] Therefore, predicting how much a user will "like" an item—as done in the Netflix Prize—can in many domains be insufficient. The problem in reality often is to additionally predict the presumed utility of a recommendation for the user.

The novelty and non-obviousness of an item are, for instance, factors that may affect the utility of item recommendations. Proposing the purchase of bread and butter in a grocery shop is obvious and will probably not generate additional sales. Similarly, recommending sequels of a movie that a user liked a lot and will watch anyway or songs by a user's favorite artist will not help the user discover new things.

In many domains, it is not even meaningful to assess the utility of a single recommendation, but only sets of recommendations. In the movie domain, once a recommendation list includes one Harry Potter movie, there is diminished value from additional Harry Potter movies. Quality measures like novelty, diversity and unexpectedness have therefore moved into the focus of researchers in recent years.[5]

While quite some progress was made over the past few years and researchers are increasingly aware of the problem that being accurate might be insufficient, a number of issues remain open. It is, for example, often not clear if and to what extent a certain quality characteristic like novelty is truly desired in a given application for a specific user. Similarly, too much diversity can sometimes be detrimental to the user experience. Finding the right mix of novel and familiar items can be challenging, and more research is required to better understand the requirements and success factors in particular domains.

Another issue is that estimating the strength of quality factors like diversity based on offline experiments is problematic. Measures like Intra-List-Diversity have been proposed in the literature but up to now it is unclear to what extent such objective measures correlate with the users' diversity perception.

Generally, the literature focuses on

Predicting held-out matrix entries is really predicting the past rather than the future.

the usefulness of the recommendations from the perspective of the end user. The providers of recommendation services, however, often try to optimize their algorithms based on A/B tests using very different measures, including sales volumes, conversion rates, activity on the platform, or sustained customer loyalty in terms of revisiting customers or renewed subscriptions. These measures vary significantly across businesses and sometimes even over time when the importance of different key performance indicators varies over time.

Context matters. Even if we generally know how to assess the usefulness of an item for a user, this usefulness might not be stable and depend on the user's current context. Assume, for example, that we have built a recommendation system for restaurants and have done everything right so far. Our algorithm is good at matching the users' preferences and the recommendations themselves are a good mix of familiar and new options as well as popular choices and insider tips. The recommendations can still be perceived as poor. A restaurant in a northern climate with acceptable food and indoor ambience but an exceptional outdoor patio overlooking the harbor will probably be a good recommendation, but only when visited in summer. Traditional CF techniques unfortunately do not account for such time aspects. In many domains context-aware algorithms are therefore required as they are able to vary their recommendations depending on contextual factors such as time, location, mood, or the presence of other people.

Over the past decade, a number of context-aware recommendation capabilities have been developed in academia and applied in a variety of application settings, including movies, restaurants, music, travel, and mobile applications.[2] Typical context adaptation strategies are to filter the recommendable items before or after the application of a non-contextualized algorithm, to collect multiple ratings for the same item in different context situations, or to design recommendation techniques that factor in context information into their machine learning models.

Contextualization has also become a common feature in real applications today. For example, many music web-

sites, such as Spotify, ask listeners for their current mood or adapt the recommendations depending on the time of the day. Online shopping sites look at the very recent navigation behavior and infer short-term shopping goals of their visitors. Mobile recommender systems finally constitute a special case of context-aware recommenders, as more and more sensor information becomes available, for example, about the user's location and local time.

From a research perspective, context is a multifaceted concept that has been studied in various research disciplines. Over the last 10 years significant progress has been made also in the field of context-aware recommenders and the first comparative evaluations and benchmark datasets were published.[31] Nonetheless, much more work is required to fully understand this multifaceted concept and to go beyond what is called the representational approach with its predefined and fixed set of observable attributes.

Interacting with users. Coming back to our restaurant recommender, let us assume we have extended its capabilities and it now considers the user's time and geographical location when making recommendations. But what happens if the user—in contrast to her past preferences—is in the mood to try out something different, for example, a vegan restaurant. How would she tell the system? And if she did not like it afterward, how would she inform the system not to recommend vegan restaurants again in the future?

In many application domains, short-term preferences must be elicited and recommending cannot be a one-shot process of determining and presenting a ranked list of items. Instead, various forms of user interactions might be required or useful to put the user in control. Examples of typical interaction patterns are interactive preference elicitation and refinement procedures, the presentation of explanations and persuasive arguments, or the provision of mechanisms that help users explore the space of available options. The design of the user experience and the provided means of interacting with the system can be a key quality factor for the success of the recommendation service.

In the research literature, conversational recommender systems were proposed to elicit user preferences interactively and engage in a "propose, feedback, and revise" cycle with users.[37] They are employed in domains where consumers are confronted with high involvement buying decisions, such as financial services, real estate, or tourism. Most approaches use forms-based dialogues to let users choose from predefined options or use natural language processing techniques to cope with free-text or oral user input. Recent alternative approaches also include more emotional ways of expressing preferences, for example, based on additional sensors to determine the user's emotional state or by supporting alternative ways of user input such as selecting from pictures.[30] Furthermore, the integration of better recommendation functionality in voice-controlled virtual assistants like Apple's Siri represents another promising path to explore by the RS research community.

One key insight in conversational systems is that users may not initially understand the space of available items, and so do not have well-formed preferences that can be expressed in terms of attributes of items. An interactive or visualization-based recommender can help users explore the item space and incrementally reveal their preferences to the system. For example, critiquing-based interfaces invite users to say "Show me more like restaurant A, but cheaper." Although these approaches attracted considerable interest in research, they are not yet mainstream in practice.[27]

Overall, with interactive systems, the design challenge is no longer simply one of choosing items to recommend but also to choose a sequence of conversational moves as proposed by Mahmood et al.[24] who developed an adaptive conversational recommendation system for the tourism domain.

Manipulation resistance. Moving on from the specific problems of how the preferences are acquired and which algorithms are used in our restaurant recommender, the question could arise of whether we can trust that the ratings of the community are honest and fair. Interested parties might manipulate the output of a recommender to their advantage, for example, by creating fake profiles and ratings.

In the long run, customers who were misled by such manipulated reviews would distrust the recommendations made by the system and in the worst case the online service as a whole. Being resilient against such manipulations can therefore be crucial to the long-term success of a system.

There has been considerable research on manipulation resistance, where resistance is defined as attackers having only a limited ability to change the rating predictions that are made. Most of it identifies archetypal attack strategies and proposes ways to detect and counteract them. For example, one "shilling" or "profile injection" attack creates profiles for fake users, with ratings for many items close to the overall average for all users. Then, these fake users give top (or bottom) ratings to the items that are being manipulated.[22] This line of research has identified algorithms that are more or less resistant to particular attack strategies.[29]

In recent research, the textual reviews provided by users on platforms like TripAdvisor are used instead or in combination with numerical ratings to understand long-term user preferences. These textual reviews do not only carry more detailed information than the ratings, they can also be automatically analyzed to detect fake entries.[20] Research suggests that in some domains the fraction of manipulated entries can be significant.

Generally, to resist manipulation, algorithms take some countermeasure that discards or reduces the influence given to ratings or reviews that are suspected of not being trustworthy. However, this has the effect of throwing away some good information. There is a lower bound on the good information that must be discarded in any attempt to prevent attacks by statistical means of noticing anomalous patterns.[34] No easy solution to this problem seems to exist, unless attackers can be prevented from injecting fake profiles.

Trust and loyalty. Manipulation resistance is not the only requirement for building a trustworthy system. Let us return to our restaurant recommender and assume that our user has eventually decided to try out one of our recommendations. Thus, from a provider perspective, we were successful in driv-

ing the user's short-term behavior. But what if the user is dissatisfied afterward with her choice and in particular feels that our recommendations were biased and not objective?

As a result, she might not trust the service in the future and even the most relevant recommendations might be ignored. In the worst case, she will even distrust the competence and integrity of the service provider.[6] An important quality factor of a recommendation system is that it is capable of building long-term loyalty through repeated positive experiences.

In e-commerce settings, users can rightly assume that economic considerations might influence what is placed in the recommendation lists and can be worried that what is being proposed is not truly optimal for them but for the seller. Transparency is therefore an important factor that has been shown to positively influence the user's trust in a system: What data does the RS consider? How does the data lead to recommendations? Explanations put the focus on providing additional information in order to answer these questions and justify the proposed recommendations.

In the research literature, a number of explanation strategies have been explored over the past 10 years. Many of them are based on "white-box" strategies that expose how a system derived the recommendations.[11] However, many challenges remain open. One is how to explain recommendations that are created by complex ML models. Another is how to leverage additional information such as the browsing history or the user's social graph to make recommendations look more plausible or familiar to the user.[32]

From Algorithms to Systems

Our brief survey on the history of the field indicates that recommender systems have arrived at the Main Street with broad industry interest and an active research community. Furthermore, we have seen the recommender systems community address a variety of topics beyond rating prediction and item ranking, for example, concerning the system's user interface or long-term effects.

Beyond the computer science perspective. Many of the proposals discussed earlier focus on algorithmic aspects, for example, how to combine context information with matrix completion approaches, how to find the most "informative" items that users should be asked to rate, or how to design algorithms that balance diversity and accuracy in an optimal way. As suspected in Wagstaff[38] for the ML community, the RS research community, to some extent, still seems too focused on benchmark datasets and abstract performance metrics. Whether or not the reported improvements actually matter in the real world for a certain application domain, and the needs of the users—are they, for instance, actually looking for new items to discover or are they seeking "more of the same" for comparison purposes—this question is seldom asked.

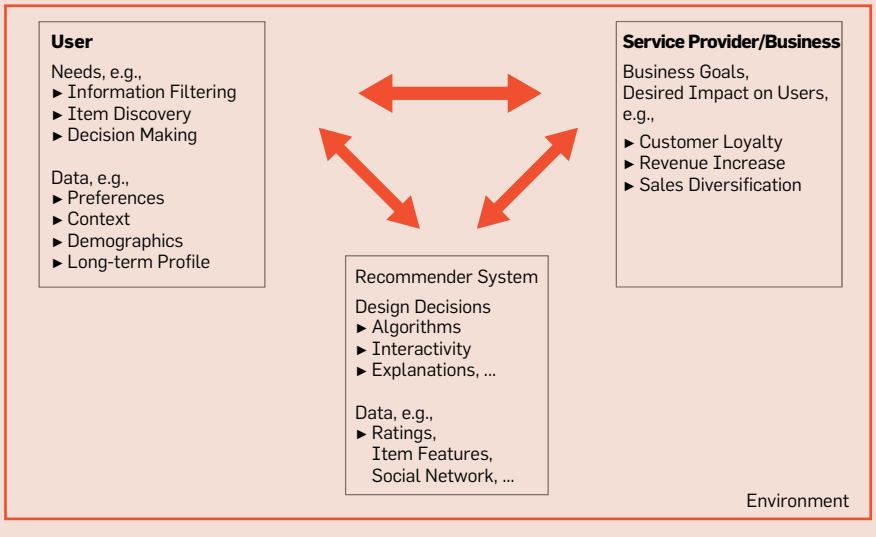
Due to their high practical relevance, RS are naturally a field of research in disciplines other than computer science (CS), including information systems (IS), e-commerce, consumer research, or marketing. Research work like Xiao and Benbasat[39] that develop a comprehensive conceptual model of the characteristics, use, and impact of e-commerce "recommendation agents" are largely unnoticed in the CS literature. In their work, the authors develop 28 propositions that center around two practically relevant questions in e-commerce settings: How can RS help to improve the user's decision process and quality? and Which factors influence the user's adoption of and trust toward the system? The process of actually generating the recommendations—which is the focus in the CS field—is certainly important, but only one of several factors that contribute to the success of an RS.

Research questions in the context of a RS should therefore be viewed from a more comprehensive perspective as sketched in Figure 1. Whenever new technological proposals are made, we should ask which specific need or requirement in a given domain are addressed. Making better buying decisions can be one need from the user's perspective; guiding customers to other parts of the product spectrum can be a desired effect from the provider's side. Correspondingly, these goals determine the choice of the evaluation measure that is chosen to assess the effectiveness of the approach.

At the end, the goals of a recommendation system can be very diverse, ranging from improved decision making over item filtering and discovery, to increased conversion or user engagement on the platform. Abstract, domain-independent accuracy measures as often used today are typically insufficient to assess the true value of a new technique.[15]

Focusing on business- and utility-oriented measures and the consideration of novelty, diversity, and serendipity aspects of recommendations—as discussed earlier—are important steps into that direction. In

Figure 1. A more comprehensive view on the recommendation problem.



any case, which measures are actually chosen for the evaluation, always has to be justified by the specific goals that should be achieved with the system. Furthermore, in offline experimentations, multi-metric evaluation schemes, application-specific measures, and the consideration of recommendation biases represent one way of assessing desired and potentially undesired effects of a RS on its users.[19]

However, to better understand the effectiveness of a RS and its impact on users, more user-centric and utility-oriented research is generally required within the CS community and the algorithmic works should be better connected with the already existing insights from neighboring fields.

Putting the user back in the loop. A recommender system is usually one component within an interactive application. The minimal interaction level provided by such a component is that a list of recommendations is displayed and users can select them for inspection or immediate consumption, for example, on media streaming platforms.

RS have one of their roots in the field of human-computer interaction (HCI) and the design of the user interface, the choice of the supported forms of interactivity, or the selection of the content to be displayed can all have an impact on the success of a recommender. However, the amount of research dedicated to these questions is comparably low, particularly when compared to the huge amount of research on item-ranking algorithms.

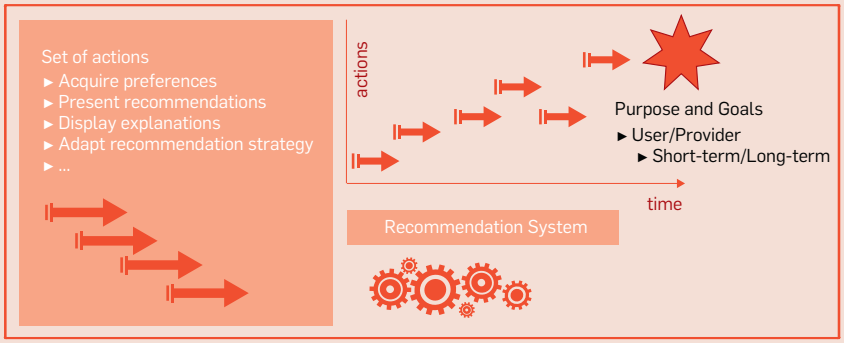
Therefore, our second tenet is that the CS community should put more effort on the HCI perspective of RS, as has been advocated earlier, for example, in Konstan and Riedl[21] and McNee et al.[28] Current research largely relies on explicit ratings and automatically observable user actions—often called *implicit feedback*—as preference indicators. Many real-world systems allow users to explicitly specify their preferences, for example, in terms of preferred item categories. Recommendation components on websites could be much more interactive and act, for example, in the e-commerce domain as "virtual advisers"[39] and social actors that ask questions, adapt their communication to the current user, provide explanations when desired, present alternative or complementary shopping proposals and, in general, put the user more into control and allow for new types of interactions.

When looking at the recommendations provided by Amazon.com on their websites, we can see that various forms of user interaction already exist that are underexplored in academia. Amazon.com, for example, provides multiple recommendation lists on its landing page. Amazon's system also supports explanations for the made recommendations and even lets the user indicate if a past user action observed by the system (for example, a purchase) should no longer be considered in the recommendation process. However, many questions, such as how to design such interactive elements in the best possible way, how much cognitive load for the user is acceptable, or how the system can stimulate or persuade people to do certain actions are largely unexplored.

Furthermore, in case a system supports various forms of interactivity and is at the same time capable of acquiring additional information from the user, additional algorithmic and computational challenges arise. An intelligent system might, for example, decide on the next conversational move, or whether to display an explanation or not, depending on the current state of the interaction or the estimated expertise and competence of the user. Some approaches in that direction were proposed in the literature in the past, but they often come at expense of considerable ramp-up costs in terms of knowledge engineering and they might appear to be quite static if they have no built-in learning capabilities.[10,24]

Finally, mobile and wearable devices have become the personal digital assistants of today. With the recent developments in speech recognition, gesture-based interactions, and a multitude of additional sensors of these devices, new opportunities arise regarding how we interact with recommender systems.

Figure 2. A new characterization of the recommendation problem.



Toward a more comprehensive characterization of the recommendation task. In the research literature, an often-cited definition of the recommendation problem is to find a function that outputs a relevance score for each item given information about the user profile and her contextual situation, "content" information about the items, and information about preference patterns in the user community.[1] Although the development of even better techniques for item selection and ranking will remain at the core of the research problem, the discussions here indicate this definition seems too narrow. To conclude our considerations related to the HCI perspective on recommenders and the more comprehensive consideration of the interplay between users, organizations, and the recommendation system, we propose a new characterization of the recommendation problem (Figure 2).

A new problem characterization. A recommendation problem has the following three components: an overall goal that governs the selection and ranking of items; a set of available actions centered on the presentation of recommended items; and an optimization timeframe:

▸ The overall goal constitutes the operationalized measure or a set of

measures that should be optimized by an appropriate selection and ranking of items from a (large) set. Optimizing a specific rank measure can be such a goal, but more utility-oriented goals and corresponding measures like user satisfaction, decreased decision efforts, revenues, or loyalty might be equally important. Generally, the goals can be derived from the user's perspective, the provider's perspective, or both.

▸ Depending on the application domain, a set of actions is available for the recommendation system to take. The central action typically is the selection and presentation of a set of items. Additional possible moves are varying its strategy to recommend items, providing specific explanations or other communication content, requesting feedback or alternative variants of user input. These conversational moves are building blocks for goal achievement. The selection of the most helpful next action and its timing can be the result of a reasoning process itself.

▸ The timeframe or optimization horizon signifies the time window over which the goal should be optimized. The explicit consideration of the time dimension allows us to differentiate between single one-shot interactions and longer time spans that can be more relevant to businesses and users.

The recommendation problem finally can be defined as: Find a sequence of conversational actions and item recommendations for each particular user that optimizes the overall goal over the specified timeframe.

Summary

Recommender systems have become a natural part of the user experience in today's online world. These systems are able to deliver value both for users and providers and are one prominent example where the output of academic research has a direct impact on the advancements in industry.

In this article, we have briefly reviewed the history of this multidisciplinary field and looked at recent efforts in the research community to consider the variety of factors that may influence the long-term success of a recommender system. The list of open issues and success factors is still far from complete and new challenges arise constantly that require further research. For example, the huge amounts of user data and preference signals that become available through the Social Web and the Internet of Things not only leads to technical challenges such as scalability, but also to societal questions concerning user privacy.

Based on our reflections on the developments in the field, we finally emphasize the need for a more holistic research approach that combines the insights of different disciplines. We urge that research focuses even more on practical problems that matter and are truly suited to increase the utility of recommendations from the viewpoint of the users. C

References

1. Adomavicius, G. and Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowledge and Data Engineering 17*, 6 (2005), 734–749.
2. Adomavicius, G. and Tuzhilin, A. Context-aware recommender systems. *Recommender Systems Handbook*. Springer, 2011, 217–253.
3. Billsus, D. and Pazzani, M.J. Learning collaborative information filters. In *Proceedings ICML '98* (1998), 46–54.
4. Breese, J.S., Heckerman, D. and Kadie, C.M. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings UAI '98* (1998), 43–52.
5. Castells, P., Wang, J., Lara, R. and Zhang, D. Introduction to the special issue on diversity and discovery in recommender systems. *ACM Trans. Intell. Syst. Technology 5*, 4 (2014), 52:1–52:3.
6. Chau, P.Y.K., Ho, S.Y., Ho, K.K.W. and Yao, Y. Examining the effects of malfunctioning personalized services on online users' distrust and behaviors. *Decision Support Syst. 56* (2013), 180–191.
7. Cremonesi, P., Garzotto, F. and Turrin, R. Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study. *ACM Trans. Interact. Intell. Syst. 2*, 1 (2012), 11:1–11:41.
8. Denning, P.J. ACM president's letter: Electronic junk. *Commun. ACM 25*, 3 (Mar. 1982), 163–165.
9. Dias, M.B., Locher, D., Li, M., El-Deredy, W. and Lisboa, P.J. The value of personalised recommender systems to e-business: A case study. In *Proceedings RecSys'08* (2008), 291–294.
10. Felfernig, A., Friedrich, G., Jannach, D. and Zanker, M. An integrated environment for the development of knowledge-based recommender applications. *Int. J. Electron.* Commerce 11, 2 (2006), 11-34.
11. Friedrich, G. and Zanker, M. A taxonomy for generating explanations in recommender systems. *AI Magazine 32*, 3 (2011), 90–98.
12. Garcin, F., Faltings, B. Donatsch, O., Alazzawi, A., Bruttin, C. and Huber, A. Offline and online evaluation of news recommender systems at swissinfo.ch. In *Proceedings RecSys '14* (2014), 169–176.
13. Ghose, A., Ipeirotis, P.G. and Li, B. Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content. *Marketing Science 31*, 3 (2012), 493–520.
14. Goldberg, D., Nichols, D., Oki, B. and Terry, D. Using collaborative filtering to weave an information tapestry. *Commun. ACM* (1992), 61–70.
15. Gomez-Uribe, C.A. and Hunt, N. The Netflix Recommender System: Algorithms, business value, and innovation. *ACM Trans. Manage. Inf. Syst. 6*, 4 (2015), 13:1–13:19.
16. Gorgoglione, M., Panniello, U. and Tuzhilin, A. The effect of context-aware recommendations on customer purchasing behavior and trust. In *Proceedings RecSys '11* (2011), 85–92.
17. Hensley, C.B. Selective dissemination of information (SDI): State of the art in May, 1963. In *Proceedings of AFIPS '63* (Spring), 1963, 257–262.
18. Hill, W., Stead, L., Rosenstein, M. and Furnas, G. Recommending and evaluating choices in a virtual community of use. In *Proceedings CHI '95* (1995), 194–201.
19. Jannach, D., Lerche, L., Kamehkhosh, I. and Jugovac, M. What recommenders recommend: An analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction* (2015), 25:1–65.
20. Jindal, N. and Liu, B. Opinion spam and analysis. In *Proceedings WSDM '08*, (2008), 219–230.
21. Konstan, J. and Riedl, J. Recommender systems: From algorithms to user experience. *User Modeling and User-Adapted Interaction 22*, 1-2 (2012), 101–123.
22. Lam, S.K. and Riedl, J. Shilling recommender systems for fun and profit. In *Proceedings of WWW '04*, (2004), 393–402.
23. Linden, G., Smith, B. and York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing 7*, 1 (2003), 76–80.
24. Mahmood, T., Ricci, F. and Venturini, A. Improving recommendation effectiveness: Adapting a dialogue strategy in online travel planning. *J. of IT & Tourism 11*, 4 (2009), 285–302.
25. Malone, T.W., Grant, K.R., Turbak, F.A., Brobst, S.A. and Cohen, M.D. Intelligent information-sharing systems. *Commun. ACM 30*, 5 (May 1987), 390–402.
26. Marlin, B.M. and Zemel, R.S. Collaborative prediction and ranking with non-random missing data. In *Proceedings RecSys '09* (2009), 5–12.
27. McGinty, L. and Reilly, J. On the evolution of critiquing recommenders. *Recommender Systems Handbook*, Springer, 2011, 419–453.
28. McNee, S.M., Riedl, J. and Konstan, J.A. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *Proceedings CHI '06*, (2006), 1097–1101.
29. Mobasher, B., Burke, R., Bhaumik, R. and Williams, C. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Technology 7*, 4 (Oct. 2007).
30. Neidhardt, J., Seyfang, L., Schuster, R. and Werthner, H. A picture-based approach to recommender systems. *J. of IT & Tourism 15* (2015), 1–21.
31. Panniello, U., Tuzhilin, A. and Gorgoglione, M. Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction 24*, 1-2 (2014), 35–65.
32. Papadimitriou, A., Symeonidis, P. and Manolopoulos, Y. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Min. Knowl. Discovery 24*, 3 (2012), 555–583.
33. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of CSCW '94* (1994), 175–186.
34. Resnick, P. and Sami, R. The information cost of manipulation-resistance in recommender systems. In *Proceedings RecSys '08* (2008), 147–154.
35. Schafer, J.B., Konstan, J. and Riedl, J. Recommender systems in e-commerce. In *Proceedings ACM EC '99* (1999), 158–166.
36. Shardanand, U. and Maes, P. Social information filtering: Algorithms for automating "word of mouth." In *Proceedings CHI '95* (1995), 210–217.
37. Shimazu, H. Expertclerk: Navigating shoppers' buying process with the combination of asking and proposing. In *Proceedings IJCAI '01* (2001), 1443–1448.
38. Wagstaff, K. Machine learning that matters. In *Proceedings ICML* (2012), 529–536.
39. Xiao, B. and Benbasat, I. E-commerce product recommendation agents: Use, characteristics, and impact. *MIS Q. 31*, 1 (Mar. 2007), 137–209.

Dietmar Jannach (dietmar.jannach@udo.edu) is a Chaired Professor of Computer Science at TU Dortmund, Germany.

Paul Resnick (presnick@umich.edu) is the Michael D. Cohen Collegiate Professor of Information at the University of Michigan School of Information, Ann Arbor, MI.

Alexander Tuzhilin (atuzhili@stern.nyu.edu) is the Leonard N. Stern Professor of Business in the Stern School of Business, New York University, NY.

Markus Zanker (markus.zanker@unibz.it) is an associate professor of computer science at Free University of Bozen-Bolzano, Italy.