

## Recommender systems: from algorithms to user experience

Joseph A. Konstan · John Riedl

Received: 10 November 2010 / Accepted in revised form: 31 December 2010 /  
Published online: 10 March 2012  
© Springer Science+Business Media B.V. 2012

**Abstract** Since their introduction in the early 1990's, automated recommender systems have revolutionized the marketing and delivery of commerce and content by providing personalized recommendations and predictions over a variety of large and complex product offerings. In this article, we review the key advances in collaborative filtering recommender systems, focusing on the evolution from research concentrated purely on algorithms to research concentrated on the rich set of questions around the user experience with the recommender. We show through examples that the embedding of the algorithm in the user experience dramatically affects the value to the user of the recommender. We argue that evaluating the user experience of a recommender requires a broader set of measures than have been commonly used, and suggest additional measures that have proven effective. Based on our analysis of the state of the field, we identify the most important open research problems, and outline key challenges slowing the advance of the state of the art, and in some cases limiting the relevance of research to real-world applications.

**Keywords** Recommender systems · User experience · Collaborative filtering · Evaluation · Metrics

---

J. A. Konstan (✉) · J. Riedl  
GroupLens Research, Department of Computer Science and Engineering,  
University of Minnesota, Minneapolis, MN 55455, USA  
e-mail: konstan@cs.umn.edu

J. Riedl  
e-mail: riedl@cs.umn.edu

## 1 Introduction

In the early-to-mid 1990's, as use of the Internet rapidly spread, recommender systems based on collaborative filtering were invented to help users address information overload by building prediction models that estimate how much the user will like each of a large set of items. The GroupLens system (Resnick et al. 1994) built on the intuition that every time a user read a Usenet News article she formed and then threw away a valuable opinion, captured those opinions as “ratings” and used the ratings of like-minded readers to produce personal predictions that were displayed as part of the article header. The Ringo system (Shardanand and Maes 1995) provided recommendations for music artists using a similar technique they termed “social information filtering.” And Video Recommender Hill et al. (1995) employed similar algorithms to support recommendations through e-mail and the web among a virtual community of movie fans. Recommender systems quickly became popular, both in research and in commercial practice. By 1996, several companies were marketing recommender engines (including Agents, Inc., which grew out of the Ringo project, and Net Perceptions, which grew out of the GroupLens project), and the first in a long series of research workshops on the field was held (in March 1996 in Berkeley, CA).

Since that start, the field has advanced through both basic research and commercial development to the point where today recommender systems are embedded in a wide range of commerce and content applications (both online and offline), where recommender systems handbooks and texts have been published (e.g., Jannach et al. 2011; Ricci et al. 2011), where universities are offering courses on recommender systems, and where there is a dedicated annual conference on the topic (the ACM Recommender Systems Conference). The scope of recommender systems has also broadened; while the term originally grew out of work in collaborative filtering, it quickly expanded to include a broader range of content-based and knowledge-based approaches. While such systems are important, we limit our focus to recommender systems that are based on collaborative filtering, though many of the interface issues we discuss apply to recommenders based on different approaches. This limitation reflects both our own expertise and the practical limitations of addressing so broad a field in a single article.

We do not attempt to offer a comprehensive review of past algorithmic research. Indeed, there have been a number of thorough surveys that focus on the algorithms behind recommenders (Adomavicius and Tuzhlin 2005; Burke 2002; Ekstrand et al. 2011; Herlocker et al. 1999, 2004), and we refer the interested reader to them. Rather, we present an overview of the most important developments in the field that touch on the *user experience* of the recommender. By user experience we mean the delivery of the recommendations to the user and the interaction of the user with those recommendations. The user experience necessarily includes algorithms, often extended from their original form, but these algorithms are now embedded in the context of the application. Our review looks at research grounded in specific recommender systems and their evaluations, and stands in contrast to Knijnenburg et al. (2012) which approaches user experience from more of an experience-model and social-experimental approach. In the rest of this section we highlight the main directions of work in the early years of recommender systems, including the beginning of the shift away from thinking of recommenders as prediction engines to considering them in the context

of user experience. The rest of the paper then reviews research directed at the user experience in recommender systems.

### 1.1 A focus on prediction algorithms

The early research recommender systems all used similar variants of a weighted,  $k$ -nearest-neighbor prediction algorithm. Intuitively, this algorithm predicts how much a target user  $u$  will like a target item  $i$  by first selecting a neighborhood of other users with tastes most similar to that of  $u$ . Neighborhood selection is performed by computing a similarity measure between  $u$ 's prior ratings and the ratings of other users (commonly using Pearson's correlation coefficient or a vector cosine similarity measure) and selecting the most similar users as neighbors. Then the ratings those neighbors assigned to item  $i$  are normalized and mixed into a weighted average (with the similarity between users as the weight), resulting in a prediction for user  $u$ . (This overview is simplified; Herlocker et al. (1999) provides more detail on variations in neighborhood formation, weighting, and normalization, along with experimental results comparing alternatives).

With prediction as the task, it is not surprising that the most popular evaluation strategies used were to measure the accuracy of the predictions. Nearly all early published research on recommender systems evaluated the recommenders using an error or correlation measure. Error measures such as mean absolute error and mean squared error provide an assessment of how well the predicted ratings match actual ratings. Correlation provides a similar measure, but focuses on correct relative prediction rather than absolute prediction values. In either case, these metrics were applied to a part of the rated data (withheld from the recommender) to assess accuracy. We discuss some of the weaknesses of this quality metric below, but should point out one significant one here—the mismatch between user need and the metric. Error and correlation scores do a good job testing recommenders as an approach to recovering missing data, but do much less well at assessing whether they can recommend valuable items *previously unknown to the user*—exactly the items they were designed to recommend in the first place.

### 1.2 Recommender algorithms in the commercial world

As recommender use expanded rapidly among online retailers and online content providers, applications of recommenders grew more diverse but the underlying algorithms converged to a few particularly useful ones. The classic recommender algorithm describe above, known as user-user collaborative filtering because the correlation is measured between pairs of users, was widely recognized as providing high-quality predictions and recommendations (see, for instance, (Breese et al. 1998)), but in practice often performed too slowly to be suitable for real-time use in applications with hundreds of thousands or millions of users. Item-item collaborative filtering (Sarwar et al. 2001) was developed as an alternative algorithm; it builds correlations between pairs of items, and then computes recommendations by finding items with high similarity to the set of items already rated favorably by the user. Many ecommerce stores have

many more customers than items, and more stable relationships between items than between customers. In these stores the item–item algorithm has faster online response time than the user–user algorithm, especially if the item relationships are precomputed. The item–item algorithm, which also extends nicely to *unary* rating sets (sets where the database has either positive information or no information at all, such as sales data), quickly became popular in commercial applications.

Alternative algorithms based on dimensionality reduction (Billsus and Pazzani 1998, Sarwar et al. 2002) showed early promise for commercial application and have been adapted in many ways to deliver high performance and high quality recommendations. These methods, commonly based on singular value decomposition, start with the recognition that a user–item ratings matrix actually has too many independent dimensions and thus loses some of the underlying relationships between user tastes. The algorithms reduce the dimensionality to an underlying set of latent taste dimensions, expressing both user preferences and item characteristics in terms of these latent dimensions. Once this dimensionalization (which is costly to compute) is established, prediction and recommendation are quite efficient, even for very large datasets. One challenge is that the singular value decomposition algorithm is too expensive to re-compute for each new rating that arrives. In the SVD literature a technique called *folding-in* is used to incorporate new data into an existing decomposition (Deerwester et al. 1990). After a significant amount of folding-in the decomposition loses its accuracy, and must be updated. More recent algorithms enable the SVD to be updated “in place” (Zha et al. 1999), but these algorithms are themselves complicated and computationally intensive.

From the earliest adoption of recommender systems, businesses recognized the need to move away from “pure” recommender algorithms to adapt them both to provide a better customer experience and to better fit with their sales and marketing efforts. In part, these impurities involved the integration of business logic (e.g., rules to prevent recommending out-of-stock goods or goods sold as loss leaders). But commercial recommenders also were concerned with shaping recommendations in ways that would later be integrated into research systems. Businesses didn’t want to waste a recommendation on a product customers would likely purchase anyway (e.g., bananas in a supermarket), and thus favored more “serendipitous” recommendation. Businesses also had to face the challenge of insufficient data on new products and new customers (cold-start problems), and thus integrated demographic, content-based, and other recommendations with the pure rating-based techniques. Researchers picked up on and advanced these themes; content-based and hybrid recommenders (see Burke’s excellent survey (2002)) have been active areas of research, and the concept of serendipity has been broadened to a wider range of control over recommendations, including the diversity of recommendation sets, discussed below.

### 1.3 A turning point: beyond accurate prediction

Business applications also brought a new vocabulary and new metrics for evaluation. As researchers encountered business applications (often through the companies launched out of academic research projects), they found little interest in MAE and

much more interest in metrics such as lift and hit rate (measures of the increase in response caused by using the recommender and of the percentage of recommendations that are converted into sales or the equivalent). These practical concerns led researchers—particularly those with a background in human-centered computing—to think more broadly about both the evaluation and the design of recommender systems and interfaces. Concern with prediction did not disappear. Indeed, the Netflix Challenge, which brought many machine learning and data mining researchers into the field, focused entirely on prediction accuracy. But there was at first a powerful undercurrent, and then a growing consensus, that small changes in MAE were not the path to significant improvements in user experience (Swearingen and Sinha 2001).

Measuring user experience, while natural in a business environment, is often challenging for recommender systems research. Pure algorithmic work can be done by simply using existing datasets; measuring user experience requires developing a system, including both algorithms and user interface, and carrying out field studies with long-term users of the system—the only reliable way of measuring behavior in a natural context. The research infrastructure used in this way fits into three categories:

- *Development of systems dedicated to experimental use.* An example of this work is Pearl Pu's work on building user trust in recommenders. Chen and Pu (2007a) included a study of 54 users that showed that explanation was more effective when deployed through the organization of the result set rather than as an annotation on an unorganized list. Similarly, our own TechLens research project has created research paper recommenders used for a series of one-shot experiments to assess recommendation algorithms for different user needs (Kapoor et al. 2007a,b,c; Torres et al. 2004; McNee et al. 2002; Ekstrand et al. 2010; McNee et al. 2006b). Coyle and Smyth (2008) applied this same technique to field studies, when studying the use of a collaborative web search tool by 50 employees in a software company over a four week trial. Similarly, Linton and Schaefer's OWL recommender for word processor commands was deployed among users in a corporate environment (Linton and Schaefer 2000).
- *Collaboration with operators of live systems to carry out recommender systems experiments.* Ziegler et al. (2005) worked with the operators of BookCrossing.com to recruit subjects to test his diversifying recommender (a one-shot experiment). Cosley et al. (2007) built a recommender for Wikipedia tasks and deployed it, entirely through public Wikipedia interfaces; Wikipedia's interfaces provided the data to run the recommender, access to users, and the data to evaluate the recommender over a longer field study.
- *Development and maintenance of research systems and user communities.* All three of the original automated collaborative filtering systems (GroupLens, Ringo, and the Video Recommender) were tested over months and years with large sets of users. That trend has continued with our GroupLens Research group's MovieLens movie recommender system; with Kautz et al.'s ReferralWeb system for recommending people based on social networks and expertise Kautz et al. (1997); with Burke et al.'s FindMe knowledge-based recommenders (Burke et al. 1997), including the Entree restaurant recommender; and many others.

There are significant challenges to conducting such human-centered research, including the challenge of finding and maintaining user communities for research, but there have also been many significant results showing the value of this approach. [Cosley et al. \(2003\)](#) showed this value directly when studying the effect of intentionally incorrect recommendations on users' subsequent ratings and overall satisfaction. In that experiment, users who received incorrect predictions (one full star higher or lower than actually predicted) showed a bias in their subsequent ratings in the direction of the error. It was somewhat encouraging, however, that users who received the incorrect predictions (2/3 of their predictions would be incorrect) did have a lower opinion of the system than those who received the actual predictions generated by the system. Interestingly, while satisfaction was reduced, users did not directly notice the cause of their reduced satisfaction.

Looking at recommenders from the user experience perspective provides some results that are counterintuitive, at least when viewed from the perspective of accurate predictions and recommendations. [McNee et al.'s \(2003\)](#) experiments on new user interfaces found that a slower initial rating interface that gave users more control (at the cost of more effort) led to higher user retention even though it did not improve actual prediction quality. And [Sinha and Swearingen \(2001\)](#) found that users find recommendations from friends to be more useful than those from "systems," even though the recommender systems have a greater range of items over which to provide accurate predictions. (Interestingly, even though they found the individual recommendations from their friends more useful, more than half the users still preferred the recommendations from the systems overall, perhaps because of the great coverage).

Frameworks for evaluation and design of recommender systems now recognize a wide range of user goals, systems objectives and measures. [Herlocker et al. \(2004\)](#) presents an evaluation framework built on different user tasks. He recognizes, for instance, that there is a fundamental difference between using a recommender system to get a few suggestions to try (in which case all that matters is the quality of the top few; errors in the bottom 90% of items may be irrelevant) and using it to check pre-specified items (in which case coverage matters, and large errors on any items would be bad). McNee's Human-Recommender Interaction theory ([McNee 2006](#); [McNee et al. 2006a](#)) adopts the same approach to recommender design, proposing a model for mapping user needs to recommender system design options through desired attributes.

The next four sections of this paper review a wide range of user-experience centered research on recommender systems. We group these into themes:

- The user-recommender lifecycle, including how recommender systems can adapt to different needs of new users vs. experienced users, and how they can balance short-term with longer-term value;
- Notions of quality that move beyond prediction accuracy, including exploring the quality of baskets of recommendations presented together, considering the dimensionality and flexibility of ratings, and exploring the underlying quality of the ratings dataset;
- Risks of recommenders, including risks to privacy and the challenge of preventing manipulation and shilling; and

- Giving users more control over recommendations, including increased transparency of recommendations and exploring ways to better understand and incorporate the context of recommendation use.

We conclude the paper with a brief exploration of broad challenges for the field.

## 2 User-recommender lifecycle

Over the course of their participation in a recommender system, user experiences, needs, and interests change. The recommender must be designed to understand the needs of the users at these different stages, and to serve them appropriately. For instance, new users may need recommendations specifically tailored to improve their trust in the system, while more experienced users may be ready for “stretch” recommendations that help broaden the user model the recommender system has for them, or that benefit the community as a whole.

### 2.1 Handling new users

Recommender systems depend on a model of a user (generally in the form of user ratings) to provide personalized recommendations. Users who do not yet have a model, therefore, cannot receive personalized ratings. The challenge of how to serve these users, and how best to learn their preferences, is known as the “new user problem” in recommender systems. Many commercial systems simply fall back to a simpler form of recommendation—either a contextual “if you’re looking at that, you might like this” product association, or a demographic recommendation or even just a set of most-popular choices. For systems that want or need to achieve personalization quickly, however, carefully selecting the interface and items to present to the user for rating can have significant effects. [Rashid et al. \(2002\)](#) explored six algorithms for soliciting ratings from new users. They found that the best performance was a hybrid that selected items (movies in this experiment) that were popular, but that also had high ratings entropy. Entropy, a measure of disagreement among raters, proved to be a good predictor of the amount of information provided to the system by a rating. This hybrid outperformed either individual strategy, since entropy alone resulted in too many obscure films, and popularity alone resulted in too little information in each rating. Interestingly, an algorithm that progressively predicted which movies a user had seen did very well in finding movies a user could rate, but poorly at building an effective profile, in part because the ratings tended to cluster in a narrow part of the movie space. [McNee et al. \(2003\)](#) compared interfaces that presented movies for rating against those where users typed in and searched for movies to rate; users took longer to build comparably good profiles by searching, but perceived the system as being fast at learning their profile, and subsequently exhibited greater loyalty to the system.

### 2.2 User lifecycle

Over the course of a user’s interactions with a recommender system, their interests may change. A young movie buff might love epics like *Star Wars* and *Lord of the*

*Rings*, and over time develop a taste for psychological dramas like *Citizen Kane* and *The Aviator*. Taste change in recommender systems may also occur as the user learns about the system. For instance, (McNee 2006) discussed the change in a new user from the mode of “checking out” whether the recommender works to the mode of exploiting the recommender for valuable suggestions. In the former mode, the user will be most interested in recommendations for items she has already consumed so that she can evaluate their quality, while during the later stage recommendations for novel items will be more valuable. Ekstrand et al. (2010) predicted a similar phenomenon for recommendations of scientific research papers. The idea of the evolution of goals has been studied in earlier agent-based systems as well. Mitchell et al. (1994) explored how active a learning personal agent should be in making automated decisions on a user’s calendar. In the initial stages the agent would watch, but take no action, building knowledge about how the person responds to requests. Later, the agent would suggest actions. Finally, after gaining the user’s trust, the agent might automatically act upon some requests where it has sufficient internal confidence, such as for a meeting with the user’s supervisor. Maes (1994) frames the challenge around the twin issues of competence of the agent in executing user tasks, and trust of the user in the competence of the agent. In recommender systems these issues have been directly studied in (Harper et al. 2005), which used economic modeling to examine users’ motivations for contributing to recommender systems. Over time, they found that it made economic sense for users to reduce their ratings contributions because the value of their ratings to themselves was reduced once the system had a rich user model for them. Thus, more experienced users were more motivated to contribute by editing movie information, or participating in the discussion forum than by providing additional ratings. Richer models of the user lifecycle promise to help recommender systems recommend activities to users that help them develop as community members, while maximizing the value of the system to them and to the community as a whole.

### 2.3 Service, community, and exploitation

Recommender systems are generally viewed as serving the user to whom they are immediately delivering a recommendation. However, in practice, they frequently have a conflict between benefit to that user and benefit to the community as a whole. This conflict is most visible when new items are introduced to the system. To whom should these items be recommended? Recommending a new item is risky, because there is little data on which to base the recommendation, but if new items are never recommended to any users, the system will never have enough data to effectively recommend them. In some ways the situation is similar for new users, who must enter an initial set of ratings for the system to learn about them. Generally the system selects the items, and asks the user to rate them. Should those items be the best ones for the system to learn about this specific new user (Rashid et al. 2002), or should they be the ones the system most needs ratings for? In the context of Wikipedia, Cosley et al. (2007) explored directing users to work that was identified by the community as valuable. In this case, the conflict is between recommending



work to a user that he is most qualified to complete versus recommending work that he is most interested in doing (SuggestBot chose the latter approach.) One way to resolve these conflicts is to give users choice, but with information about how their choices will provide value to the community. For instance, [Rashid et al. \(2006\)](#) demonstrated that showing users how their contributions will help other users induced them to change the types of contributions. Specifically, the users contributed more to items that provided value to other users who were similar to them in their taste for movies.

### 3 More sophisticated notions of quality

Recommender systems have been most frequently measured by the average error across all predictions they are capable of forming. Average error measures miss the features that are most important to user satisfaction in many applications of recommenders. New notions of quality have emerged that provide a richer picture of the performance of recommenders in practice, and look beyond simply the algorithm to explore the ways in which the rating process itself can impact quality. Designers of recommender systems should consider carefully both the evaluation tools they use to measure the value they are providing to users and the underlying means for obtaining quality ratings data.

#### 3.1 Avoiding bad mistakes

While the earliest work on recommender systems was valuable for simply showing that predicted ratings correlated with actual withheld ones, researchers quickly realized that not all mistakes were created equally. Shardanand and Maes in their paper on the RINGO personalized music recommendation system ([1995](#)) looked at the magnitude of errors (the mean absolute error), but also recognized that the performance of the system mattered most on music that users felt strongly about, so they separately analyzed the errors in cases where the user ratings of artists were very high or low (1, 2, 6, or 7 on a 7-point scale). They would elsewhere refer to major errors that could undermine trust as *reversals*—errors that would give a low score to a highly-rated item or a high-score to a low-rated one. [Miller et al. \(1997\)](#) reported *mean squared error* along with mean absolute error and the correlation between prediction and rating, indicating a desire to report a metric that disproportionately punishes large errors. Later [Herlocker et al. \(1999\)](#) introduced the use of the Receiver Operating Characteristic curve for evaluating prediction algorithms. The ROC curve, long used in signal processing, measures the ability to distinguish signal from noise, and thus the tradeoff between false positives and false negatives. ROC measurements are based on a cut-off value between positive and negative ratings, and only penalize cases where a prediction would cause a positively-rated item to be missed or a negatively-rated one to be consumed. [Herlocker et al. \(2004\)](#) later showed that the various “decision support” metrics (i.e., those intended to measure how well a recommender distinguishes between good and bad, rather than overall accuracy) perform similarly.

### 3.2 The shift to top-N recommenders

The earliest retail sales applications of recommender systems (including websites such as Amazon.com and CDnow.com, as well as telephone sales marketing applications) focused on recommending a small number of products to customers in the hopes of making a sale or adding an additional product to a sale. These personalized recommendations were used in place of generic product placements (e.g., on a site homepage) or hand-generated cross-sell lists (e.g., people-who-bought this recommendations or check-out-time recommendations). Researchers rapidly recognized the importance of considering recommendation lists as opposed to simply predictions. Breese et al. (1998) introduced a ranked scoring metric often referred to as the Breese score. This metric recognizes that the value of a recommendation is a function both of its position in the list (based on evidence that the likelihood of selecting a recommended item decays exponentially as the item is lower on the list) and the utility of the item (e.g., the probability that the user will like it, or more generally the value of the recommendation to the user if taken). Sarwar et al. (2000) evaluated collaborative filtering algorithms using top-N *precision* and *recall* metrics, adapted from the field of information retrieval. In commercial applications, metrics such as hit-rate and lift measure the actual frequency that recommended items are selected and the resulting increased sales. One of the criticisms of the Netflix Challenge was the fact that it measured success only on the accuracy of predictions. Under this approach, small distinctions among the worst movies are as important as differences among the top items that might actually change the list of recommendations.

### 3.3 Diversity and evaluating lists of recommendations

As recommender systems have become more commonly used for producing sets or lists of recommendations, rather than simply individual predictions, attention has shifted to the value of the recommendation list as a whole and not simply the quality of each individual recommendation. A particular concern expressed in certain domains concerns “pigeonholing” users—identifying a single narrow interest and making many similar recommendations. Consider a music recommender that, knowing you enjoyed “Stairway to Heaven” played only Led Zeppelin music. Or consider, as Ziegler did, the case of Amazon.com which at times would recommend five books by the same author—each of them a good recommendation, but collectively a poor recommendation list (Ziegler et al. 2005). Ali and Van Stam (2004) identify this problem with collaborative filtering recommenders as the portfolio problem; indeed, this problem is well-known to editors of newspapers who must choose how much of the paper, or the front page, to devote to the most important stories versus how much to diversify the contents. Ziegler et al. (2005) showed how to address this challenge with topic diversification, using an external taxonomy (from Amazon.com) to create a similarity measure, and then injecting diversity into the resulting recommendation set (in his experiments, using a dataset from BookCrossing.com). He found that users noticed that individual recommendations were not as good in the diversified lists, but that users preferred lists that were diversified to a moderate

extent (about 30%), particularly when those lists came from an item-item recommendation algorithm. [Torres et al. \(2004\)](#) experimented with a variety of collaborative filtering, content filtering, and hybrid algorithms for research paper recommendation, finding that different algorithms performed better at generating different types of recommendation lists (e.g., related research for a paper, a broad introduction to a field).

### 3.4 Multidimensional ratings and tags

One design question that has challenged recommender systems designers is the question of when a single-dimensional “rating” and “prediction” are sufficient, and when users would benefit from more detailed ratings and recommendations for specific attributes of a product. Zagat’s popular restaurant recommendations, for example, are based on ratings along the four dimensions of Food, Decor, Service, and Cost; but few other recommenders ask their users to provide such detailed ratings. Our own early work ([Konstan et al. 1998](#)) found that users resisted multidimensional ratings. More recent work by [Adomavicius and Kwon \(2007\)](#) suggests that multicriteria ratings can enable better recommendation, and evaluated new recommendation algorithms using a multicriteria rating set from Yahoo! Movies (with separate ratings for story, acting, direction, and visuals). This result suggests that in some circumstances, at least, multidimensional ratings are acceptable to users and can improve overall recommender performance. The increased prevalence of tagging systems, which help users organize, categorize, and search for content, seems to help fill the gap that might otherwise be filled by multi-dimensional ratings, with the added benefit of having an arbitrarily large number of dimensions. One challenge in implementing tagging is that coming up with tags requires careful thought from users, which can slow down their contributions. Further, having each user individually come up with tags may create a system with a wide variety of nearly synonymous tags. For instance, in MovieLens there are tags for “Tarantino”, “Quentin Tarantino”, and “tarantino is god”, all apparently referring to the same director. Interestingly, recommender systems have been employed specifically for the task of suggesting tags to users with the goal of helping the tagging folksonomy converge ([Sen et al. 2006](#)). For instance, [Jaschke et al. \(2007\)](#) propose a graph-based algorithm based on PageRank, which they find does better than previously proposed algorithms. The task of recommending tags based on content analysis of the item or of metadata about the item was chosen for the ECML PKDD 2009 Discovery Challenge task (footnote: <http://www.kde.cs.uni-kassel.de/ws/dc09/online>). The top two competitors in the challenge use: (1) an algorithm that crosses a profile of the item (tags that have been applied to the item before and text content in the item) with a profile of the user (tags the user has shown interest in before) to create a set of tags that are related to both the item and the user ([Lipczak et al. 2009](#)); and (2) an algorithm that pulls tags from a wide variety of resources across the Internet, and weights the tags provided by those resources based on their prior accuracy ([Mrosek et al. 2009](#)). An active area of research is extending recommenders like these to work interactively with users, for instance by suggesting appropriate tags as they begin typing a tag.

### 3.5 Data quality

One of the challenges in recommender systems is that the data underlying the recommendations may be noisy, corrupted, or just plain wrong. [Amatriain et al. \(2009\)](#) demonstrated that asking users to rerate items they had previously rated could reduce errors in prediction by as much as 14%. They used an algorithm to carefully select items that were likely to be unusually noisy to reduce the additional effort required from the user. Further, [Marlin and Zemel \(2009\)](#) demonstrated that most existing algorithms are based on the flawed assumption that the missing ratings are uniformly distributed. They showed that by carefully correcting for the distribution of missing ratings they could improve the accuracy of well-known algorithms significantly. Most existing academic studies of recommender systems are based on datasets with explicit ratings of items, such as the MovieLens datasets. However, in commercial practice it is much more common to have datasets based on implicit actions users take, such as which links they click on, or how long they dwell on an item detail page. Though implicit measures of interest are noisier than explicit measures, they are so ubiquitous that using them effectively can lead to high quality recommendations nonetheless ([Morita and Shinoda 1994](#); [Konstan et al. 1997](#)). Additional research is needed to further develop the science of effectively using all of the available data for forming recommendations.

## 4 Hidden dangers

Recommender systems can enhance our lives, by helping us filter through the too-many available options to select the ones most value to us. However, there are also social risks to the wide deployment of recommenders, from the obvious risks to individual privacy, to more subtle risks to the structure of social relationships ([Toch et al. 2012](#)). Recommender technology can help ameliorate these risks—but only if deployed thoughtfully.

### 4.1 Privacy

[Canny \(2002\)](#) explored recommender systems that were able to provide recommendations for individuals while accessing only encrypted forms of their preference vectors. This approach tackled a key challenge for recommender systems: those users might not be willing to provide their preference data to potentially untrusted central servers. In Canny's system users would each circulate an encrypted vector of their preferences, which could be used to compute a public "aggregate" that can be shared by all users, and used to create personalized predictions for each user. One of the advantages of this approach is that in principle it can be used to create decentralized recommender systems, with no central agent who has privileged access to information. Such an approach may be part of the answer to privacy challenges like those uncovered in [Frankowski et al.'s \(2006\)](#) paper that showed that if people have access to de-identified preferences they can with high probability tie one preference profile to a public (and thus, identified) preference profile available elsewhere. Thus, a music recommender

company could figure out which of its users were openly blogging about music on a separate public blog site. [Miller et al. \(2004\)](#) built on Canny's approach to develop recommenders that could be implemented entirely on PDAs, which could be carried with the user. The PDA would use a personal area network, such as Bluetooth, to communicate with other system users' PDAs, sharing encrypted preference profiles. Each PDA would individually create the aggregate that it needs to form predictions solely for its own user. Another approach to privacy-preserving recommenders is to introduce noise into the computations. ([McSherry and Mironov 2009](#)) show how to take the leading algorithms in the Netflix prize computation and modify them to make drawing inferences about the original ratings difficult. They show that with careful design they can create modified algorithms with formal privacy guarantees, with relatively modest loss in recommendation quality. These papers demonstrate that there is enormous potential for privacy aware recommender technologies, which have the potential to create high quality recommendations completely under the control of the end user.

#### 4.2 Social effects of recommenders

As recommenders become ubiquitous, they are being designed to fit into social contexts. Over time, the recommenders will interact with the social group, leading to new demands on the recommender, and creating new norms for the group. For instance, [Cosley et al. \(2003\)](#) found that showing users predictions that had been manipulated either up or down could substantially influence the ratings that the users would then give the item. The authors hypothesize that this effect opens the way to recursive shilling: by causing one user to see a falsely positive rating, that user may rate the item inappropriately high, causing other users to see falsely positive ratings, and so on. Another social risk of recommenders is cyberbalkanization. First popularized by [Van Alstyne and Brynjolfsson \(1996\)](#), cyber-balkanization refers to the potential for increased specialization of online communities, as people choose to interact primarily with people who are like themselves. The idea is that the reduced friction of discovery online will make it easier for people to find others with whom they share political, religious, and cultural beliefs, and that they may choose to spend more of their time with these "similar others". Recommenders can exacerbate cyber-balkanization, by making it even easier for people to find the information, people, and products they are most comfortable with ([Van Alstyne and Brynjolfsson 2005](#)). The concern is that balkanization will make it more difficult for people to talk to each other, creating increasingly isolated sub communities. Interestingly, some studies predict that recommenders will lead to an opposed bad outcome—homogenization. [Fleder and Hosanger \(2009\)](#) have pursued the question of what will happen in ecommerce as recommenders are ubiquitous. Their simulations predict that in some cases the use of recommenders may lead to an even stronger winner-take-all system, in which the leading products are seen by so many consumers, and rated highly by so many consumers, that they come to completely dominate their market segment. Research is needed to understand how recommenders can be tuned to achieve positive social outcomes—and how groups can determine what outcomes they desire. One example of such a positive social outcome

is making visible to users the value of their contributions to a group (Rashid et al. (2006)). Because the effects of their work was visible to users, they were more motivated to contribute to the group, and specifically more motivated to contribute in ways that helped other users the most.

#### 4.3 Robustness and manipulation resistance

From the early days of recommenders systems there has been speculation from users that other users are providing dishonest preference profiles in order to skew the recommendations. For instance, MusicFX users reportedly changed their music preferences to make sure that music they liked a lot was more likely to play than music they could merely tolerate. Lam and Riedl (2004) showed that even simple attack techniques would be able to significantly distort the recommendations formed by the most common recommender algorithms. Among others, Mobasher and his colleagues have produced a stream of research refining the understanding of the performance of a variety of attack methods against a variety of recommender algorithms (Sandvig et al. 2008; Mobasher et al. 2007; Williams et al. 2007). More recently, Resnick and Sami have demonstrated the theoretical limits on the ability of any type of attack to manipulate recommendations, as a function of the size of that attack, and have explored algorithms that are resistant to manipulation by design (Resnick and Sami 2007, 2008). Research in this area has established that even blunt attacks can change recommendations, and that these attacks might be difficult to detect in practice. However, in a large recommender community, an effective attack on popular items is difficult because of the volume of countervailing data (Lam and Riedl 2004).

### 5 Putting the user in control

A strength of recommender systems is that they reduce the workload on users who are overwhelmed by the choices available to them. However, users are often more satisfied when they are given control over how the recommender functions on their behalf—even, in some cases, when that control increases the effort required of them, and when the resulting recommendations are objectively *less* accurate. The sweet spot is recommenders that balance serving users effectively, while ensuring that the users have the control they desire.

#### 5.1 Interactive recommendation dialogues

Burke et al. (1997) recognized that often users are better at critiquing presented alternatives than at specifying what they want. In multiple domains, including restaurant search, they implemented FindMe-style recommender systems where users could navigate among options by selecting attributes to change. For example, a user looking at a restaurant could choose to see restaurants like a recommended one but with Italian food or a quieter atmosphere. They deployed this system at the 1996 Democratic National Convention, and found that users found the experience to be useful, intuitive, and

fulfilling. The FindMe systems spawned a wide range of more interactive, exploratory recommender systems, including many of the systems now known as critiquing-based systems (Chen and Pu 2012). More advanced systems generate suggestions to users of “compound critiques” that enable them to navigate simultaneously along several of the key dimensions that are most important to the user (Chen and Pu 2007b; Reilly et al. 2007). From them, we learn the importance of thinking creatively about the role of the user of such a system—classic recommenders simply asked users to supply ratings; these interactive systems often invite the user to dynamically guide the search process, and as a result may be more engaging as tools of discovery. Pu et al. (2012) provides a more detailed review of preference elicitation and refinement research to improve recommender system user experience.

## 5.2 Recommender context

Adomavicius et al. 2005 argues that in different contexts, such as at different times of the day, users may be interested in different types of recommendations. For instance, a recommender for a newspaper might learn to recommend business news early in the morning on weekdays, but leisure articles late in the afternoon on Friday. They built a recommender for the movie domain that is able to incorporate such contextual features in its recommendations. Park et al. (2007) describe a recommender for a mobile device that uses the user’s location, the current time, and even the weather as contextual information in choosing the best items to recommend. Tintarev and Masthoff (2007) interviewed users to understand their motivations for using recommendations, and discovered that users felt that their mood would influence their taste in genre and other features of movies. The success of these projects demonstrates the importance of context in producing valuable recommendations. However, there is an opportunity for systematic tools for incorporating the broad variety of types of context into the full spectrum of recommender scenarios.

## 5.3 Recommending for groups

Most recommenders systems have been developed for individuals, but there are a number of situations in which groups of people want to do an activity together, and would benefit from recommendations tailored to the entire group. One of the earliest explorations of this idea was the MusicFX recommender by McCarthy and Anagnost (1998) for “work out music” to be played in a gym while people exercise together. This system worked by having people swipe their ID cards as they entered the gym, so a central server knew who was in the gym at any given time. Each person filled out a profile for what kind of music he or she preferred to exercise to. The recommender considered the profiles of all currently active participants in choosing what music to play at any given time. O’Connor et al. (2001) explored similar ideas in developing a recommender for groups of people who wanted to go to a movie together. In both of these systems one of the interesting challenges was how to aggregate the preferences of the different members of the group to form predictions that would

best satisfy the needs of that group. Is it better to make sure that one person is as happy as possible? Or to make sure that no person is too unhappy? Partly to address questions such as these (Masthoff and Gatt 2006) have shown that the affective state of individuals in the group can be modeled, and that doing so makes it possible to reason about the overall effects of the sequence of recommenders on the satisfaction of the group. Their model includes components that estimate individual satisfaction, and the effect of the individual reactions on the other members of the group (“contagion”). Research in this area has demonstrated the value of automated support for group recommenders, and the need to understand the goals and constraints of the group.

#### 5.4 Explanations and transparency

Herlocker et al. (2000) suggested that one of the reasons recommenders have had more success in low risk domains such as books, movies, and CDs, and less success in high-risk domains such as cars or real estate is that users are unwilling to risk very much based on a recommendation they do not understand. One possible solution is to provide *explanations* that give users confidence to act on the recommendation. Tintarev and Masthoff (2007) describe three motivations for explanations in recommender systems: (1) *transparency*, which shows the user how the recommendation was formed, so the user knows how much to trust it; (2) *trust*, which encourages the user to take the recommendation *independent of how accurate the recommendation is*, and (3) *scrutability*, which enables the user to take action to let the system know of mistakes in the data used for the recommendations, so future recommendations can be improved. The explanations can be of a wide range of types. For instance Maes (1994) discusses a booking agent that represents its internal state as a facial expression intended to give the users insight into its confidence. Herlocker et al. (2000) studied a variety of statistical presentations that explained the recommendation in terms of the neighborhood of similar users, and found that very simple explanations were more effective than more sophisticated representations of how the algorithm actually worked. (Vig et al. 2009) showed that explanations can be effective even if they are in a completely different dimension (tags) than the dimension the algorithm was using for the computation (ratings). Studies have shown that users like explanations, and use the explanations to help make decisions about which items to purchase (Herlocker et al. 2000; Tintarev and Masthoff 2008; Vig et al. 2009). Further, users feel the explanations help them understand the recommendations better, and make choices that better fit their current mood or interests (Vig et al. 2009). Hook et al. (1996) discussed ways to form explanations in adaptive systems. Tintarev and Masthoff (2012) provide a more detailed review of research on explanations in recommender systems. We share their conclusion that additional work is needed in the field, particularly to explore the extent to which the recommendations are actually improving user decision-making, to explore the costs and benefits of scrutability in improving recommendations, and to understand the effects of explanations in the context of live, deployed systems.



## 6 Looking forward

Recommender systems has grown as an area of both research and practice. Over the past four years, for example, the annual ACM Recommender systems conference has grown from 120 to 320 attendees, and from 35 to 129 full paper submissions. Furthermore, a wave of second generation recommender companies offer more complete commerce and content solutions. As the field moves forward, we believe user experience concerns will continue to drive advances in algorithms, interfaces, and systems.

### 6.1 A key challenge

A key challenge to the field of recommender systems as a whole is the *integration of content-based approaches* (including use of product information and other metadata, user demographics, and profiles of user content preferences), *collaborative approaches* (including explicit and implicit ratings, tagging, and other behavioral indicators of user preference), and *contextual approaches* (including understanding business rules, location, user task and mood, interfaces and interaction appropriate to the situation, and other contextual cues and constraints) into comprehensive, practical recommender systems. We are already seeing indications of strong interest in this direction. While the Netflix challenge focused solely on prediction, a 2010 contest challenged researchers to develop context-aware recommendations, looking at factors such as mood and time of year (the Challenge on Context-Aware Movie Recommendation, <http://www.dai-labor.de/camra2010>).

Looking at issues that cut across the fifteen themes we discuss above, we see three specific “choke points” that must be addressed to keep the field moving forward towards addressing this key challenge:

*Scalability.* As we continue to integrate more information into recommender systems, and as we expect them to adapt to greater contextual challenges, the problems of algorithmic and system scalability only grow. One long-running challenge is to recommend the next item in a sequence of items to be explored (e.g., the next lesson in an online learning system). Such a recommender requires not only ratings of the individual lessons, but also information on sequence and dependency. Similar challenges exist with other forms of context and with recommenders that hope to tap the wide range of implicit behavior that can today be recorded through web browsing behavior. In response, we will need both fundamental computational advances (such as better fast approximations for dimensionality reduction) and recommender-specific performance research.

*Better exploitation of user-contributed content.* The Web has permanently become a participatory medium, and the wealth of available information is staggering. Aside from implicit behavior data such as time-spent reading and links followed, we now have access to an unprecedented amount of review data (consider, for instance, the over 40 million reviews and opinions at TripAdvisor.com, multiplied by hundreds of other community review sites), comments, tags, blog posts, tweets, facebook updates,

and much more. A key challenge for recommender system developers is to find ways to use this content to improve recommender systems while still respecting individual privacy.

*Research infrastructure.* As the field grows, we face significant challenges in research infrastructure and standards. While there are a few widely-used datasets, they are mostly concentrated on entertainment domains such as movies, and even for these datasets the field lacks standards for reporting experimental results (one notable problem is that some researchers ignore predictions their system cannot make, while others penalize themselves or substitute a non-personalized prediction). The challenge is that much larger when considering live user studies, especially field studies. Over the next several years, we hope to see the community develop not only common research standards and metrics, but also a set of shared research resources where novel user experiences can be presented to long-term users of ongoing recommender communities. Eventually, we expect any significant algorithmic advance to go through stages of being tested on standard datasets, and then if promising tested on real users in one or more live systems.

**Acknowledgements** We are grateful for the rich and intellectually-stimulating interactions we have had with our many colleagues in the recommender systems research community. We particularly thank our close collaborators Paul Resnick and Loren Terveen, and the students and visitors who have formed the GroupLens Research group. This article is based upon work supported by the National Science Foundation under Grants DGE 95-54517, IIS 96-13960, IIS 97-34442, IIS 99-78717, IIS 01-02229, IIS 03-24851, IIS 05-34420, IIS 05-34939, IIS 08-08692, and IIS 09-64695. We are grateful for their support.

## References

- Adomavicius, G., Kwon, Y.: New recommendation techniques for multicriteria rating systems. *IEEE Intel. Syst.* **22**(3), 48–55 (2007)
- Adomavicius, G., Tuzhlin, A.: Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **6**(17), 734–749 (2005)
- Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhlin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.* **23**(1), 103–145 (2005)
- Ali, K., Van Stam, W.: Tivo: making show recommendations using a distributed collaborative filtering architecture. In: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 394–401. ACM, New York (2004)
- Amatriain, X., Pujol, J.M., Tintarev, N., Oliver, N.: Rate it again: increasing recommendation accuracy by user re-rating. In: *Proceedings of the third ACM conference on Recommender systems (RecSys '09)*, pp. 173–180. ACM, New York (2009)
- Billsus, D., Pazzani, M.J.: Learning collaborative information Filters. In: *Proceedings of the Fifteenth International Conference on Machine Learning* **54**, 47 (1998)
- Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 43–52 (1998)
- Burke, R.: Hybrid recommender systems: survey and experiments. *User Model. User-Adapt. Inter.* **12**(4), 331–370 (2002)
- Burke, R.D., Hammond, K.J., Young, B.C.: The FindMe approach to assisted browsing. *IEEE Expert: Intel. Syst Their Appl.* **12**(4), 32–40 (1997)
- Canny, J.: Collaborative filtering with privacy via factor analysis. In: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 238–245 (2002)

- Chen, J., Konstan, J.A.: Computer science conference paper selectivity and impact: an analysis. *Commun. ACM.* **53**(6), 79–83 (2009)
- Chen, Y., Konstan, J.: Online field experiments: lessons from communityLab. In: *Proceedings of the Third International Conference on e-Social Science*. Ann Arbor, October 7–9, 2007 (2008)
- Chen, L., Pu, P.: Preference-based organization interface: aiding user critiques in recommender systems. In: *Proceedings of International Conference on User Modeling (UM2007)*, pp. 77–86. June 25–29, Corfu (2007a)
- Chen, L., Pu, P.: Hybrid critiquing-based recommender systems. In: *Proceedings of International Conference on Intelligent User Interfaces (IUI2007)*, pp. 22–31. Jan 28–31, Hawaii (2007b)
- Chen, L., Pu, P.: Critiquing-based recommenders: survey and emerging trends. *User Model. User-Adapt. Inter.* **22**(1–2), 125–150 (2012)
- Cosley, D., Frankowski, D., Terveen, L., Riedl, J.: SuggestBot: using intelligent task routing to help people find work in wikipedia. In: *Proceedings of the 12th international conference on Intelligent user interfaces (IUI '07)*, pp. 32–41. ACM, New York (2007)
- Cosley, D., Lam, S.K., Albert, I., Konstan, J.A. and Riedl, J.: Is seeing believing? How recommender system interfaces affect users' opinions. In: *Proceedings of the conference on Human Factors in Computing Systems*, pp. 585–592 (2003)
- Coyle, M., Smyth, B.: (Web Search) shared: social aspects of a collaborative, community-based search network. In: *Proceedings of the 5th international conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH '08)*, pp. 103–112 (2008)
- Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *J. Am. Soc. Inform. Sci.* **41**(6) (1990)
- Ekstrand, M.D., Kannan, P., Stemper, J.A., Butler, J.T., Konstan, J.A., Riedl, J.T.: Automatically building research reading lists. In: *Proceedings of the fourth ACM conference on Recommender systems (RecSys '10)*, pp. 159–166. ACM, New York (2010)
- Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact* **4** (2011)
- Fleder, D., Hosanger, K.: Blockbuster culture's next rise or fall: the impact of recommender systems on sales diversity. *Manage. Sci.* **55**(5), 697–712 (2009)
- Frankowski, D., Cosley, D., Sen, S., Terveen, L., Riedl, J.: You are what you say: privacy risks of public mentions. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '06)*, pp. 565–572. ACM, New York (2006)
- Harper, F. Maxwell, Frankowski, D., Drenner, S., Ren, Y., Kiesler, S., Terveen, L., Kraut, R., Riedl, J.: Talk amongst yourselves: inviting users to participate in online conversations. In: *Proceedings of the 12th international conference on Intelligent user interfaces (IUI '07)*, pp. 62–71. ACM, New York (2007)
- Harper, F., Li, X., Chen, Y., Konstan, J.: An economic model of user rating in an online recommender system. In: *Proceedings of the 10th International Conference on User Modeling*, pp. 216–307. Edinburgh (2005)
- Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: *Proceedings of the 2000 ACM conference on computer supported cooperative work (CSCW '00)*, pp. 241–250 (2000)
- Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* **22**(1), 5–53 (2004)
- Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J.: An algorithmic framework for performing collaborative filtering. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 230–237 (1999)
- Hill, W., Stead, L., Rosenstein, M., Furnas, G.: Recommending and evaluating choices in a virtual community of use. In: *Proceedings of the 1995 Conference on Human Factors in Computing Systems*, pp. 194–201. Denver (1995)
- Höök, K., Karlgren, J., Wærn, A., Dahlbäck, N., Jansson, C.G., Karlgren, K., Lemaire, B.: A glass box approach to adaptive hypermedia. *User Model. User-Adapt. Inter.* **6**(2–3), 157–184 (1996)
- Jannach, D., Zanker, M., Felfernig, A., Friedrich, G.: *Recommender systems: an introduction*. Cambridge University Press, New York (2011)
- Jaschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag Recommendations in Folksonomies. In: Kok, J.N., Koronacki, J., De Mantaras, R.L., Matwin, S., Mladeni, D., Jaschke, R., Skowron, A. (eds.) *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007)*, pp. 506–514. Springer-Verlag, Berlin, Heidelberg (2007)

- Kapoor, N., Butler, J.T., Fouty, G.C., Stemper, J.A., Konstan, J.A.: A study of citations in users' online personal collections. In: Proceedings of the ACM 2007 European Conference on Digital Libraries, pp. 404–415. Budapest, Hungary (2007a)
- Kapoor, N., Butler, J.T., Fouty, G.C., Stemper, J.A., Konstan, J.A.: Resolvability of references in users' personal collections. In: Proceedings of the Sixth International Conference on Conceptions of Library and Information Sciences (CoLIS), pp. 43–53. Boras (2007b)
- Kapoor, N., Chen, J., Butler, J., Fouty, G.C., Stemper, J.A., Riedl, J., Konstan, J.A.: TechLens: a researcher's desktop. In: The first ACM conference on recommender systems, extended abstracts, pp. 183–184. Minneapolis (2007c)
- Kautz, H., Selman, B., Shah, M.: Referral web: combining social networks and collaborative filtering. *Commun. ACM* **40**(3), 63–65 (1997)
- Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: Applying collaborative filtering to usenet news: the GroupLens system. *Commun. ACM* **40**(3), 77–87 (1997)
- Konstan, J.A., Riedl, J., Borchers, A., Herlocker, J.L.: Recommender Systems: A GroupLens perspective. In: Proceedings of the AAAI Workshop on Recommender Systems 1998, pp. 60–64. AAAI Technical Report WS-98-08 (1998)
- Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the user experience of recommender systems. *User Model. User-Adapt. Inter.* **22** (2012). doi:[10.1007/s11257-011-9118-4](https://doi.org/10.1007/s11257-011-9118-4)
- Lam, S.K., Riedl, J.: Shilling recommender systems for fun and profit. In: Proceedings of the 13th international conference on World Wide Web (WWW '04), pp. 393–402. ACM, New York (2004)
- Leshed, G., Hancock, J.T., Cosley, D., McLeod, P.L., Gay, G.: Feedback for guiding reflection on teamwork practices. In: Proceedings of the 2007 international ACM conference on Supporting group work (GROUP '07), pp. 217–220. ACM, New York (2007)
- Linton, F., Schaefer, H.-P.: Recommender systems for learning: building user and expert models through long-term observation of application use. *User Model. User-Adapt. Inter.* **10**(2–3), 181–208 (2000)
- Lipczak, M., Hu, Y., Kollet, Y., Milios, E.: Tag sources for recommendation in collaborative tagging systems. In: Proceedings of the ECML-PKDD discovery challenge workshop, pp. 157–172. Bled, Slovenia (2009)
- Maes, P.: Agents that reduce work and information overload. *Commun. ACM* **37**(7), 30–40 (1994)
- Marlin, B.M., Zemel, R.S.: Collaborative prediction and ranking with non-random missing data. In: Proceedings of the third ACM conference on Recommender systems (RecSys '09), pp. 5–12. ACM, New York (2009)
- Masthoff, J., Gatt, A.: In pursuit of satisfaction and the prevention of embarrassment: Affective state in group recommender systems. *User Model. User Adapt. Inter.* **16**, 281–319 (2006)
- McCarthy, J., Anagnost, T.D.: MusicFX: an arbiter of group preferences for computer supported collaborative workouts. In: Proceedings of the 1998 ACM conference on Computer supported cooperative work (CSCW '98), pp. 363–372. ACM, New York (1998)
- McNee, S.M.: Meeting user information needs in recommender systems. Ph.D. Dissertation, University of Minnesota-Twin Cities (2006)
- McNee, S.M., Riedl, J., Konstan, J.A.: Making recommendations better: an analytic model for human-recommender interaction. In: CHI '06 extended abstracts on Human factors in computing systems (CHI '06), pp. 1103–1108. ACM, New York (2006a)
- McNee, S.M., Kapoor, N., Konstan, J.A.: Don't look stupid: avoiding pitfalls when recommending research papers. In: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (CSCW '06), pp. 171–180. ACM, New York (2006b)
- McNee, S.M., Lam, S.K., Konstan, J.A., Riedl, J.: Interfaces for eliciting new user preferences in recommender systems. In: Proceedings of the 9th International Conference on User Modeling, pp. 178–187 (2003)
- McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Proceedings of the 2002 ACM conference on Computer supported cooperative work (CSCW '02), pp. 116–125. ACM, New York (2002)
- McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09), pp. 627–636. ACM, New York (2009)
- Miller, B.N., Konstan, J.A., Riedl, J.: PocketLens: toward a personal recommender system. *ACM Trans. Informat. Syst.* **22**(3), 437–476 (2004)

- Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J.: MovieLens unplugged: experiences with an occasionally connected recommender system. Poster in IUI '03: Proceedings of the 8th international conference on intelligent user interfaces, pp. 263–266 (2003)
- Miller, B.N., Riedl, J.T., Konstan, J.A.: Experience with GroupLens: making usenet useful again. In: Proceedings of the USENIX 1997 Annual Technical Conference, pp. 219–233 (1997)
- Mitchell, T.M., Caruana, R., Freitag, D., McDermott, J., Zabowski, D.: Experience with a learning personal assistant. *Commun. ACM* **37**(7), 80–91 (1994)
- Mobasher, B., Burke, R., Bhaumik, R., Williams, C.: Towards trustworthy recommender systems: an analysis of attack models and algorithm robustness. *ACM Transac. Internet Technol.* **7**(4) Article 23 (2007)
- Morita, M., Shinoda, Y.: Information filtering based on user behavior analysis and best match text retrieval. In: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 272–281. Dublin, Ireland Springer-Verlag, New York (1994)
- Mrosek, J., Bussman, S., Albers, H., Posdziech, K., Hengefeld, B., Opperman, N., Robert, S., Spira, G.: Content- and graph-based tag recommendation: two variations. In: Proceedings of the ECML-PKDD Discovery Challenge Workshop, pp. 189–199. Bled, Slovenia, Europe, September 7th (2009)
- O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: PolyLens: a recommender system for groups of users. In: Prinz, W., Jarke, M., Rogers, Y., Schmidt, K., Wulf, V. (eds.) Proceedings of the Seventh European Conference on Computer-Supported Group Work, pp. 199–218. Bonn, Germany 16–20 September (2001)
- Park, M.-H., Hong, J.-H., Cho, S.-B.: Location-based recommendation system using bayesian user's preference model in mobile devices. In: Indulska, J., Ma, J., Yang, L., Ungerer, L., Cao, J. (Eds.) Proceedings of Ubiquitous Intelligence and Computing, 4611, pp. 1130–1139. Springer, Berlin, Heidelberg (2007)
- Pu, P., Chen, L.: Trust building with explanation interfaces. In: Proceedings of the 11th international conference on Intelligent user interfaces (IUI '06), pp. 93–100. ACM, New York (2006)
- Pu, P., Chen, L., Hu, R.: Evaluating recommender systems from the user's perspective: survey of the state of the art. *User Model. User-Adapt. Inter.* **22** (2012). doi:10.1007/s11257-011-9115-7
- Ramezani, M., Sandvig, J.J., Bhaumik, T., Schimoler, R., Burke, B., Mobasher, B.: Exploring the impact of profile injection attacks in social tagging systems. In: Proceedings of the 2008 WebKDD Workshop, held at ACM KDD'2008 Conference on Knowledge Discovery and Data Mining, pp. 134–143. Las Vegas (2008)
- Rashid, A.M., Ling, K., Tassone, R., Resnick, P., Kraut, R., Riedl, J.: Motivating participation by displaying the value of contribution. In: Grinter, R., Rodden, T., Aoki, P., Cutrell, E., Jeffries, R., Olson, G. (eds.) Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06), pp. 955–958. ACM, New York (2006)
- Rashid, A.M., Albert, I., Cosley, D., Lam, S.K., McNee, S.M., Konstan, J.A., Riedl, J.: Getting to know you: learning new user preferences in recommender systems. In: Proceedings of the 7th international conference on Intelligent user interfaces (IUI '02), pp. 127–134. ACM, New York (2002)
- Reilly, J., Zhang, J., McGinty, L., Pu, P., Smyth, B.: Evaluating compound critiquing recommenders: a real-user study. In: Proceedings of ACM Conference on Electronic Commerce (EC'07), pp. 114–123, June 11–15. San Diego (2007)
- Resnick, P., Sami, R.: 2008. Manipulation-resistant recommender systems through influence limits. *SIGecom Exch.* **7**, 3, Article 10 (November 2008), 4 pages
- Resnick, P., Sami, R.: The influence limiter: provably manipulation-resistant recommender systems. In: Proceedings of the 2007 ACM conference on Recommender systems (RecSys '07), pp. 25–32. ACM, New York (2007)
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work (CSCW '94), pp. 175–186. ACM, New York (1994)
- Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.): Recommender systems handbook. Springer, Berlin (2011)
- Sandvig, J.J., Bhaumik, R., Ramezani, M., Burke, R., Mobasher, B.: A framework for the analysis of attacks against social tagging systems. In: Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, held at AAAI 2008. Chicago, July 2008 (2008)
- Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web (WWW '01), pp. 285–295. ACM, New York (2001)

- Sarwar, B., Karypis, G., Konstan, J.A., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Proceedings of the Fifth International Conference on Computer and Information Technology (2002)
- Sarwar, B. M., Karypis, G., Konstan, J.A., Riedl, J.T.: Analysis of recommendation algorithms for e-commerce. In: Proceedings of the 2nd ACM conference on Electronic commerce (EC '00), pp. 158–167. ACM, New York (2000)
- Sen, S., Lam, S.K., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F.M., Riedl, J.: tagging, communities, vocabulary, evolution. In *Proceedings of ACM CSCW 2006*. Baniff, Alberta, Canada, November. ACM New York, NY, pp. 181–190 (2006)
- Shardanand, U., Maes, P.: Social information filtering: algorithms for automating “word of mouth”. In: Katz, I.R., Mack, R., Marks, L., Rosson, M.B., Nielsen, J. (eds.) Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '95), pp. 210–217. ACM Press/Addison-Wesley Publishing Co., New York (1995)
- Sinha, R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries (2001)
- Smyth, B.: Adaptive information access and the quest for the personalization-privacy sweetspot. In: Proceedings of the 10th international conference on Intelligent user interfaces (IUI '05), pp. 2–2. ACM, New York (2005)
- Swearingen, K., Sinha, R.: Beyond algorithms: an HCI perspective on recommender systems. In: Proceedings of the SIGIR 2001 Workshop on Recommender Systems (2001)
- Tintarev, N., Masthoff, J.: Evaluating the effectiveness of explanations for recommender systems. *User Model. User-Adapt. Inter.* **22** (2012). doi:[10.1007/s11257-011-9117-5](https://doi.org/10.1007/s11257-011-9117-5)
- Tintarev, N., Masthoff, J.: The effectiveness of personalized movie explanations: an experiment using commercial meta-data. In: Nejdl, W., Kay, J., Pu, P., Herder, E., (eds.) Adaptive Hypermedia and Adaptive Web-based Systems, 5149, pp. 204–213. 5th International Conference (AH 2008), Hannover, Germany. July 29–August 1st (2008)
- Tintarev, N., Masthoff, J.: Effective explanations of recommendations: user-centered design. In: Proceedings of the 2007 ACM conference on Recommender systems (RecSys '07), pp. 153–156. ACM, New York (2007)
- Toch, E., Wang, Y., Cranor, L.F.: Personalization and privacy: a survey of privacy risks and remedies in personalization-based systems. *User Model. User-Adapt. Inter.* **22**(1–2), 203–220 (2012)
- Torres, R., McNee, S.M., Abel, M., Konstan, J.A., Riedl, J.: Enhancing digital libraries with TechLens+. In: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries (JCDL '04), pp. 228–236. ACM, New York (2004)
- Van Alstyne, M., Brynjolfsson, E.: Global village or cyber-balkans? Modeling and measuring the integration of electronic communities. *Manage. Sci.* **51**(6), 851–868 (2005)
- Van Alstyne, M., Brynjolfsson, E.: Electronic communities: Global village or cyberbalkans? In: DeGross, J., Jarvenpaa, S., Srinivasan, A. (eds.) Proceedings of the 17th International Conference on Information Systems (ICIS 1996), pp. 80–98. Cleveland, December (1996)
- Vig, J., Sen, S., Riedl, J.: Tagsplanations: explaining recommendations using tags. In: Proceedings of the 13th international conference on Intelligent user interfaces (IUI '09), pp. 47–56. ACM, New York (2009)
- Williams, C., Mobasher, B., Burke, R.: Defending recommender systems: detection of profile injection attacks. *J. Serv. Ori. Comput. Appl.* **1**, 157–170 (2007)
- Zha, H., Zhang, Z.: On matrices with low-rank-plus-shift structure: partial SVD and latent semantic indexing. *SIAM J. Matrix Anal. Appl.* **21**(2), 522–536 (1999)
- Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proceedings of the Fourteenth International World Wide Web Conference (WWW 2005), pp. 25–32. ACM, New York (2005)

## Author Biographies

**Joseph A. Konstan** is Distinguished McKnight University Professor and Distinguished University Teaching Professor at the University of Minnesota, where he teaches and conducts research on human-computer interaction, and particularly on recommender systems, social computing, and persuasive computing. He holds an AB (1987) from Harvard University and an MS (1990) and a Ph.D. (1993) from the University of California, Berkeley, all in Computer Science. He is a Fellow of the ACM, a member of the CHI Academy, and a winner (with John Riedl and others) of the ACM Software Systems Award for the GroupLens Collaborative Filtering Recommender Systems.

**John Riedl** is Professor of Computer Science and Engineering at the University of Minnesota, where he teaches and conducts research on social computing, particularly on recommender systems and the social web. He holds a BS (1983) in Mathematics from the University of Notre Dame and an MS (1985) and a Ph.D. (1990) in Computer Sciences from Purdue University. In 1996, he co-founded Net Perceptions, Inc. to commercialize collaborative filtering recommender systems; he served there as Chief Technology Officer until 1998 and as Chief Scientist from 1998 through 2002. He is a Fellow of the ACM and a winner (with Joseph Konstan and others) of the ACM Software Systems Award for the GroupLens Collaborative Filtering Recommender Systems.