DOI: 10.1609/aaai.12015

Human-centered recommender systems: Origins, advances, challenges, and opportunities

Joseph A. Konstan | **Loren G. Terveen**

GroupLens Research Group, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455

Correspondence
Joseph A. Konstan, Department of Computer Science and Engineering, University of Minnesota, 4-192 Keller Hall, Minneapolis, MN 55455.
Email: konstan@umn.edu

Abstract
From the earliest days of the field, Recommender Systems research and practice has struggled to balance and integrate approaches that focus on recommendation as a machine learning or missing-value problem with ones that focus on machine learning as a discovery tool and perhaps persuasion platform. In this article, we review 25 years of recommender systems research from a human-centered perspective, looking at the interface and algorithm studies that advanced our understanding of how system designs can be tailored to users objectives and needs. At the same time, we show how external factors, including commercialization and technology developments, have shaped research on human-centered recommender systems. We show how several unifying frameworks have helped developers and researchers alike incorporate thinking about user experience and human decision-making into their designs. We then review the challenges, and the opportunities, in today's recommenders, looking at how deep learning and optimization techniques can integrate with both interface designs and human performance statistics to improve recommender effectiveness and usefulness

INTRODUCTION

Recommender systems use data, including data reflective of user preferences, to customize experiences for those users. In their simplest form, these systems use product preference data to determine which products to display (or "recommend"). Even this simple form of recommendation raises questions. What data? The customer's prior purchases? The customer's reviews, ratings, or explicit preferences? The customer's pattern of responses to prior displays? And what about data from other customers? Data about the products? These questions often form the core concern of recommender systems researchers–how do I squeeze the most value out of data to make better recommendations.

This article, however, looks at a very different set of questions. What does it mean for a recommendation to be good? Should I recommend things the customer likes and buys often? Should I introduce the customer to new products? How many products are too many to recommend? How many are too few? Should I show the best recommendations all at once, or save some for later? When should the recommendations be diverse, and with respect to each other or the customer's history? What type of recommendations lead to immediate satisfaction and which lead to long-term use and customer happiness?

We use the term *Human-Centered Recommender Systems* to describe an approach to recommender systems research and practice that focuses on understanding the characteristics of recommender systems, the characteristics of recommender systems users, and the relationships between them. The goal of human-centered recommender systems research is to design the algorithms and interactions of recommender systems to better fulfill the goals of users and of the organizations engaging with these users.

We often use this term in contrast to *missing value estimation* or *simplistic machine learning*. The contrast can be seen in a simple example often used by the late John Riedl, one of the pioneers of the field. When critiquing *click through* or *purchase rate* as a measure of recommender system success, he would suggest that he could create a very high purchase-rate supermarket shopping cart recommender. It would be a printed sign saying "buy bananas and bread." He would then observe that these two products are among the most frequent purchases in American supermarkets, and therefore the recommender would be very successful. But it would also provide very little value (to either the shopper or the market), since almost all of these customers would have already known about and chosen whether to buy bananas or bread without the recommendation. And yet – perhaps there are times when such recommendations are useful; early in a person's usage of a system, even if a recommendation provides no new information, it may build their trust that the system actually "understands" them.

Indeed, as this example suggests, there is an intimate relationship between human-centricity and the metrics used to evaluate a recommender system (or an innovative component to be incorporated into one). As we discuss in more detail below, substantial advances in human-centric recommender systems have generally required deployments of systems with real users. But experimenting with real users is expensive, high-effort, and often risky. So many innovations are tested using historic user data (which may not be able to assess impact on user behavior). Part of our discussion below addresses how the field is increasing the amount of experiment-based theory to permit better human-centered evaluation through offline analyses.

And no discussion of human-centered recommender systems research would be complete without considering the nature of the human decision-making that such recommender systems are intended to facilitate or influence. While much of the offline study of recommender systems is unfortunately divorced from the context of use, other researchers have built on studies of decision-making and consumer behavior to understand how such decision-making is and can be affected by recommendations (indeed, often as a surrogate for human salespeople). Chen et al. (2013) edited a special issue of *ACM TiiS* on this topic, and outlined in their introduction both the state of research at that time and the key questions that are still being explored today.

The rest of this article is organized into four roughly chronological (but overlapping) sections. We discuss the early days of recommender systems, particularly as those systems first emerged in research and industry. Then we discuss the widespread commercial growth of these systems, followed by the great influx of algorithmic research spurred on by the Netflix Prize competition, and finally the current day and future of the field. We end with a look at how human-centered recommender systems approaches may merge into more simplistic machine learning approaches, offering the best of both approaches.

One final usage note. The term "recommender systems" can be used both narrowly (typically to describe systems that implement some form of collaborative filtering) and broadly (including content-based techniques and much more). We focus here on systems and algorithms with a collaborative filtering lineage (in other words, systems that use some form of ratings data—either explicit or implicit—from many users to provide personalized recommendations to each user). But many of the systems we discuss are hybrids of collaborative and content-based approaches, and we believe that the human-centered issues we raise apply across the broad space of recommender systems.

THE EARLY SYSTEMS (THE MID 1990S): HUMAN-CENTERED ORIGINS AND A QUEST FOR EFFECTIVE METRICS

Fundamental framing. By the late 1980s, researchers had identified *information overload* as a crucial problem. Malone et al. (1987) wrote 'It is already a common experience in mature computer-based messaging communities for people to feel flooded with large quantities of electronic "junk mail" .... it is also common for people to be ignorant of facts that would facilitate their work and that are known elsewhere in their organization.' In response, they framed the computational task of *information filtering*, encompassing both omitting some items (of less interest or quality) and selecting other items (of greater interest or quality) from a large universe. They further defined several methods of filtering, two of which are directly relevant to our concerns: *cognitive filtering* filters items based on their content, and *social filtering* filters items based on some relationship between the person for whom the filtering is being done and other people. Subsequent work elaborated these two approaches using the terms *content-based* and *collaborative filtering*, later unified under the rubric of *recommender systems*.

The first generation of recommender systems—roughly through most of the 1990s—focused on inventing techniques that addressed the information overload problem. More specifically, they attempted to provide one or more of the following services to their users:

> *Prediction* – estimating how much a user will want each of a set of items (assigning a score to each item);

> *Recommendation* – selecting a set of items (from a large, mostly static collection) to present to the user (choosing items the user is expected to want to consume);
>
> *Filtering*– processing a stream of items to choose which ones to present (or not present) to the user.

These correspond to the services that predated modern recommender systems. Hotel and restaurant guides provide a form of star-rating prediction (though not personalized). Advertising servers and the Book-of-the-Month club (and its many similar services) provide recommendations (as do many human agents). Clipping services provide a form of positive filtering; email spam filters provide a form of negative filtering.

While the development of information filtering systems to address information overload provides a useful framing (and arguably is the most common framing in the field) for considering human-centered issues, other framings existed at the same time. Most notably, a long line of work on support for interactive querying and exploration of datasets, boosted by the introduction of techniques from case-based reasoning and user modeling, led to a set of systems organized around the notion of user-system dialogue. These systems provided mechanisms through which the user would interact with the system through a mixture of querying and "critiquing" of system-displayed items, continually refining the model of user interests, until the user finds the item(s) they are satisfied with. We will touch on some systems from this tradition as well, where they help us illuminate relevant issues.

Human-centered origins of early systems. We consider six early recommender systems, looking at their human-centered goals and the key decisions made in their design and evaluation:

Tapestry (Goldberg et al. 1992) was an organizational message database that allowed users to annotate items, essentially serving as an endorsement of the item contents. Users could use these annotations to query the item database, e.g., to find items endorsed by a particular person. While this work reported on an early, experimental version of the system, it was clear that the goals were to reduce overload and to make it possible for users to create queries that automatically "appraised" and routed important content to them.

GroupLens (Konstan et al. 1997; Resnick et al. 1994), Ringo (Shardanand and Maes 1995), and the Bellcore Video Recommender (Hill et al. 1995) were three similar ratings-based collaborative filtering systems, though each had different models for user interaction. GroupLens operated on a stream of content (Usenet News articles) while Ringo and the Video Recommender operated on (regularly updated) sets of music artists and movies, respectively.

GroupLens was integrated into Usenet News reader software; it asked users to rate articles (on a 1–5 scale) and provided a personalized prediction (on the same scale) for each yet-unread article based on the ratings of those who'd rated it. While the original GroupLens paper focused on the architecture of the system and scalability, it also brought forward human-centered questions of social implications of such a system and incentives for rating (vs. waiting and free-riding on the ratings of others). When the system was later evaluated in a field study, metrics included the correlation between predictions and user ratings (showing the value of personalization), predictive utility (helping users make good decisions), and cold start problems for new users and new content. GroupLens also showed that time-spent reading was a good surrogate for explicit ratings, building on the earlier work on Morita and Shinoda (1994).

Ringo provided web-based and e-mail based interfaces. Users were provided with a list of artists to rate (initially 125 of them, on a 7-point rating scale), with the top of the list including frequently-rated artists to provide better matching with others (a strategy later taken to the extreme by Jester, a joke recommender where everyone saw and rated the same first jokes (Golberg et al. 2001)). Ringo would then respond to user queries for recommendations, negative recommendations (artists you'd hate), or predictions. It would provide a confidence level along with the score. Shardanand and Maes conducted systematic evaluation of recommender algorithms including evaluating neighborhood formation strategies. They focused on measuring predictive error (both mean absolute error and standard deviation of errors) looking at how the error distribution changed with errors, but they also looked at user feedback, observing the cold-start issues for both the system as a whole and for new users.

Bellcore Video Recommender provided an e-mail interaction in which users were sent a list of 500 videos to rate (as a research interface, not intended to be a long-term interface), of which 250 were common to all users and the other 250 were randomly chosen. Users rated on a scale of 1–10 along with "must-see," "not-interested," and "unseen;" they received back a set of recommendations with predictions, but also analyses of video category preferences, a list of highly-correlated neighbors, and explicitly explained recommendations from those neighbors. Hill et al. were trying to build a social community that would be augmented by the recommender, not replaced by it. They explored reliability of ratings (finding a .83 correlation between same-user ratings 6 weeks apart), using that to estimate an upper limit on prediction performance (they also looked at correlation between ratings and predictions). Notably, they compared neighborhood-based recommendation with following advice of critics, finding much higher performance for neighborhood-based recommendation; they also reported subjective user feedback.

These three neighborhood-based collaborative filtering systems were different in many ways, but they shared several key similarities. Each of them started with a basis of prediction–of estimating what score the user would assign to an item. Each of them was based on the assumption that prediction should be personalized–that people have different tastes. And each of them believed that the key to making good personalized predictions was to find the right set of neighbors–other users with a history of similar tastes. Each system assumed that the best predictions were those that were most accurate, and the good recommendations flowed from good predictions. And each of them explored concerns surrounding the awkward challenge of cold start – both the challenge of providing useful information to new users who had too sparse a profile to accurately model tastes and the challenge of providing useful information about sparsely rated items. The approaches were somewhat different – GroupLens defaulted to providing new users with average predictions while Ringo and the Video Recommender created artificial density through their start-up rating process.

The next two early systems took different approaches within the family of recommender systems.

Pointing the Way (Maltz and Ehrlich 1995) built upon the ideas from Tapestry to incorporate a notion of "active collaborative filtering" into Lotus Notes. Recognizing that certain individuals in groups tend to review lots of content and find what is relevant to others, they enabled users to easily send "pointers" to other individuals or groups. Their evaluation focused on ease of use and early usage, noting that active collaborative filtering provides an easy-to-use and flexible alternative in a context where users know each other (such as a workplace).

PHOAKS (Terveen et al. 1997) investigated whether recommendations of pages (in Usenet articles) could be aggregated to find useful resource pages relevant to that newsgroup community. The system first mined articles for these recommendations, looking for URLs in a context that suggests endorsement (positive language around it, not quoted or in a signature). It then aggregated these recommendations to provide a "frequently mentioned resources" for each group. PHOAKS provided transparency by making it easy to view the recommendations themselves, not just the recommended resource. Terveen et al. evaluated the approach showing that resources recommended by the most users tended also to be listed in newsgroup FAQ lists.

As we look at these six early systems, three distinct sets of human-centered recommendation questions arise:

> Input/Data: What information is obtained and used to compute recommendations? How is it obtained (implicit, explicit)? On what scale is it input? How reliable is it? Can users examine it and understand its role in the process?;
>
> Algorithm/Output: What is computed and how? For predictions, what data accompanies the prediction that can explain its confidence? For recommendations, are they simply top predictions, or something else? What data comes with them?
>
> Presentation: How are the predictions or recommendations presented to the user? What interaction is available to explore that data?

The input and presentation are what most classically relate to human-computer interaction, but in the field of human-centered recommender systems the data, algorithm, and output have been quite prominent.

This early era also corresponds to the end of the systems-centric research period in this field. Building new systems is hard, as each of these studies showed. Over the following decades, fewer research systems were built and maintained (and in turn more commercial systems would be built and deployed). Substantial research would be conducted on a few research systems, but the vast majority of research would be conducted using datasets – either for offline studies and simulations or as a basis for short-term experiments. As a result, metrics would become even more important than they were in the early days of recommender systems.

As we move forward through later stages of the field of recommender systems, we explore how these three sets of questions are addressed through research and practice.

THE COMMERCIAL BOOM (1995–2005)

Early collaborative filtering systems had the good fortune to bloom in the early days of the dot com boom. As a result, several research projects quickly commercialized even as the corporate sector was finding its own footing in recommender systems. The result of these early ventures was rapid learning about what mattered in commercial applications–engineering algorithms for high throughput and low latency. While none of these original research-spawned companies still survive, the lessons they learned and innovations they spawned still shape the field.

Pattie Maes and her group at MIT founded Agents, Inc. (later Firefly Networks) in late 1995. The company promoted a network model where their clients would integrate with the Firefly system, with Firefly using end-customer data to provide personalized recommendations. Firefly's goal was to grow its network of end-users, and through that the value of the network for each client. Firefly provided a music recommender BigNote that encouraged end-users to sign up to receive music recommendations. Firefly attracted high-profile customers such as Barnes & Noble, America Online, and Yahoo!. As a company, it developed technology to support profiles of personal data (the Firefly Passport). In the end Microsoft acquired Firefly, primarily for the Passport and technical talent.

The GroupLens team at Minnesota founded Net Perceptions in mid-1996. The company took a very different approach to delivery of recommender services. It produced a recommender engine that companies could install and operate within their own computing systems. This engine received product ratings and produced various forms of prediction and recommendation through an API. Net Perceptions quickly signed up major customers such as Amazon.com, Best Buy, and J.C. Penney. The company quickly learned that user-user collaborative filtering could not scale to the needs of these large businesses and implemented new algorithms geared towards low-latency and high-throughput. The company grew rapidly during the boom, but then could not sustain itself through the dot com bust.

In 1997, Iconomic Systems by founded by Pearl Pu, Boi Faltings, and their team at EPFL. The company took the group's research on dialog-based recommendation and exploration interfaces for large product spaces and applied it to travel recommendation. The company was later acquired in 2001 by i:FAO, a German corporate travel agency.

And in 2004 Francisco Martin, Jon Herlocker, and Tom Diettrich at Oregon State University founded MyStrands (with offices in Oregon and Barcelona). Their first product was the MusicStrands music recommender, though they later broadened their scope to include financial and retail recommendation. Now named Strands, the company is still active in providing recommendation products and services.

The lessons these companies (and other startups of the era) learned were important ones that shaped the research agenda in recommender systems for nearly a decade to follow. Recommendations were useful, but only if they could be delivered on time. The surprising delight of an unexpected gem could be sacrificed for reliably providing good recommendations, even if somewhat predictable. And businesses wanted to see evidence of the quality of a recommender system. Metrics became an important part of the sales process – businesses might start by looking at accuracy metrics, but eventually would want to measure something more substantial–typically a business-specific metric such as click through rate, conversion rate, or lift (the added purchase value).

With the benefit of growing datasets, in this period we see the first comprehensive evaluations of algorithms. Herlocker et al. (1999) published an evaluation framework specific to neighborhood-based user-user nearest neighbor algorithms. The analysis was entirely offline; it explored a wide range of algorithmic choices (neighborhood formation, weighting of neighbors, normalization), and looked at mean absolute error for predictions, coverage (the percentage of items for which a personalized prediction could be made), and a decision-support metric ROC-4. ROC-4 reflected the recommender's performance as a filter (the area under a receiver operating characteristic curve) when the goal is to only recommend items with a true user rating of 4 or higher. These offline analyses presaged a wave of algorithmic innovation built on offline analysis, though many of the later works simplified the analysis to a single error metric (usually root mean squared error).

By late 1996 it had become clear, as well, that the "classic" user-user neighborhood-based collaborative filtering algorithm would not scale well enough for large business applications. Shardanand and Maes (1995) had mentioned the possibility of computing relationships among items; Amazon built a scalable recommender system using that model, and the item-item algorithm became widely known, studied, and used (Sarwar et al. 2001, which continued to use MAE as a metric). At the same time, early versions of dimensionality reduction algorithms (later known as latent factor algorithms) were emerging (e.g., Sarwar et al. 2000, which compared these algorithms using top-N Precision and Recall to better assess recommendation performance). The shift from evaluating predictions to evaluating "the success of the top-N" reflected the common business application in use – most retail, information, and advertising services provided one or a small number of top items to the user; few of them had frequent need for accurate predictions outside that top few.

Recommender systems were a hot industry in the late 1990's. Businesses were seeing significant value from making suggestions to customers. Advertisers were improving click-through rates (at a time when their business model required delivering a certain click through rate to collect full payment). And new algorithms promised significant improvements in scale and performance with little cost in accuracy.

But it was just then that many in the research community renewed their focus on user experience. Specifically, the ongoing adoption of web-based systems created a variety of new challenges and opportunities.

Lieberman (1995); Pazzani et al. (1996), and created systems to assist in web browsing. Lieberman's Letizia served as an "over the shoulder" agent watching the user's interaction and suggesting potential content of interest. Pazzani et al's Syskill and Webert explored a variety of AI techniques using user ratings along with page selection to suggest both content and future searches. Each was an attempt to tame the unstructured vastness of the early web.

Burke et al. (1996) introduced the FindMe knowledge-based recommenders. These recommenders allowed users to navigate an information space (restaurant listings in Entreé, the most famous instantiation). Users could start from an example and then navigate through information space by specifying directions of interest – "less expensive," "more lively," "make it Italian" until they found what they were looking for. The database was shared, but the results were personal due to each user's personal path of discovery. In this approach, the user drove the interaction by critiquing the examples generated by the system, driving it in a direction to approach their (perhaps implicit) goal. Others would later extend this approach to more complex critiquing, including Reilly et al. (2007) which reports on the evaluation of compound critiquing recommenders (bringing together lines of research from Barry Smyth and Pearl Pu, both of whom advanced the case-based approach to recommender systems). Pu et al. (2010) brings together a long line of research to provide research-based usability guidelines for critiquing-based product recommenders. Much later, Taijala et al. (2018) adapted this approach to provide a mechanism for navigating through latent item space with "more like" and "less like" options over an item set.

At the same time, ratings-based collaborative filtering was exploring issues of user experience. About a month after the GroupLens team transitioned its MovieLens system to an item-based recommender system (a decision made to address performance issues), we started receiving complaints from users about the new system not being "bold enough" and feeling insufficiently personalized. This led to extensive study of the qualitative aspects of recommender systems and how we might improve the user experience.

Herlocker et al. (2000) experimented with explanations within a recommender system, finding that explaining the deep underlying statistics was counterproductive but that simpler explanations with reference to item attributes or broad statements about other users were well received. This work was later extended by Tintarev and Masthoff (2007) who developed a framework for analyzing recommendation explanations.

McCarthy and Anagnost (1998) explored recommendation for music in shared spaces in MusicFX, a collaborative music-listening recommender for a gym. They found that people would state extreme preferences to manipulate the algorithm. O'Connor et al. (2001) introduced PolyLens to explore the idea of recommending for intentional groups, including comparing strategies of merging a group into a "single pseudo-user" (which was not effective) and combining prediction and recommendation lists for individuals in a group. Jameson and Smyth (2007) provide a good review of work from that era on group recommendation. And Baltrunas et al. (2010) provide a detailed study of rank-list group recommendations, including the impacts of group size and diversity on recommendation fit.

Ziegler et al. (2005) may have been the first experimental study to show that users prefer to have at least a certain amount of diversity in their recommendation lists (in particular, showing that diversified lists were rated higher by users than less diverse lists even when users rated the items lower individually). This fit into a long line of studies showing that "just producing the top-predicted items" may not be the best strategy for a recommender system. Indeed, industry leaders had figured out already that striking the balance between the items expected to be recommended and the items that might prompt new interest is a tricky challenge.

Cosley et al. (2003) reported on an experiment showing that recommendations affect user perceptions of the items recommended, and in turn showing that erroneous, biased, or manipulated recommendations could in turn propagate (rather than self-correct) through their effect on subsequent user perceptions and ratings.

We published a review of the state of the art in evaluating collaborative filtering recommender systems in 2004 (Herlocker et al. 2004). This review looked at both evaluation processes and metrics. The review looked in depth at accuracy metrics, finding that there were three families of such metrics that had reflected different properties of the systems' performance. And it looked at a wide range of user experience metrics, from the novelty and serendipity (Swearingen and Sinha 2001) of recommendations to confidence in the recommendations to directly measured short- and long-term user experience.

Indeed, it would appear that the commercial era was a fruitful one for human-centered recommender systems work, though much of it had moved away from the recommendation algorithms and into new interactions, new ways to understand the data, or new ways to present and combine the data. Recommender systems of that era were still seen as systems where the algorithm (sometimes referred to as the "engine") was just one component. But the whole field was about to be shaken up by a compelling challenge.

THE NETFLIX PRIZE AND ITS AFTERMATH (2006–2016)

It would be difficult to overstate the impact the Netflix Prize has had on the field of Recommender Systems. The announcement that Netflix would award $1 million to whoever could improve upon its Cinematch recommender algorithm by at least 10%, combined with the provision of an extensive dataset and a leaderboard with annual progress prizes, led to an incredible influx of machine learning and data mining researchers. Individuals, small teams, and larger teams–from academia and industry labs–tried their hands at this challenge.

It took nearly 3 years and a set of mergers of top teams that had complementary approaches to produce a grand prize winner. By that time hundreds of researchers had been introduced to the challenges and data sets of recommender systems. And they learned that this was an area that industry, at least, felt was worthy of substantial investment.

Let us take a moment to credit the Netflix Prize for what it did well. It brought out a diverse set of researchers and techniques that advanced machine learning and data science. One can debate whether the winning stacking hybrid algorithm was elegant, but it did show how to squeeze the most out of a set of simpler algorithms. And the field has learned that lesson well. The field has remained enlarged, with ever more technical approaches to recommendation in place (many of which are proving useful to more complex situations) and with an explosion of industry impact as well.

From the human-centered perspective, the Netflix Prize was a disaster out of which great things bloomed. It was a disaster in that all of this magnificent talent was brought to bear on a problem that is largely irrelevant to users. The Netflix Prize was about optimizing the RMSE (root mean squared error) of predictions from a withheld set of user ratings. The closer the algorithm is to accurately predicting those ratings, the better their score. But let's consider why this might not be the most useful metric (and in turn why it may be that Netflix never chose to replace its own algorithms with the winner).

For Netflix, the most important question is which movies to show to a user, and whether the movies shown are likely to lead to the user selecting a movie to watch. The Netflix Prize accuracy measure fails to deliver on this most important question in three ways: (1) All of the movies being evaluated are ones the user has already watched; except in rare circumstances, one would want to see recommendations for new content the user had not already seen. (Of course, this is a fundamental challenge of offline evaluation; you can't evaluate new items without delivering those items to users.); (2) The RMSE metric weighs all predictions equally. An improvement in the accuracy of predicting a low-ranked item helps the score just as much as an improvement on a top-ranked item. But for Netflix, this clearly isn't true. It is not important to be able to predict just how much you dislike a particular movie, just that you dislike it. But it may make a big difference whether something is in your top 3 vs. your top 20, since there may not be screen space to show all 20. (This is very similar to search; order of items on the 20th page of search results does not matter as much as on the 1st page.) This is a problem with the metric chosen. Other metrics can do a better job focusing on top items (top-k precision, normalized discounted cumulative gain, mean average precision, etc.); (3) User preferences are contextual. Recommendations should consider the context in which the user is watching. Someone who just finished a sci-fi trilogy at 3 in the morning may be less likely to want another multi-movie series (and indeed, may not find that this is a good time for a documentary). The Netflix Prize task was entirely devoid of context, but modern commercial recommender systems are generally quite context rich.

Or to put it more bluntly, evaluating an algorithm based on how well it can predict what users have already done (but is hidden) selects for algorithms that produce some of the worst possible real-world recommendations–recommendations for items the user has already seen or experienced. And more generally, for items most similar to the ones the user has already experienced, in other words, conservative and boring recommendations. This realization brought us back to the complaints that had been raised when MovieLens switched to item-item – the recommendations became less bold. One of the significant strengths of user-user nearest neighbor collaborative filtering was its ability to recommend an item strongly based on a single neighbor's top rating (this is also what could cause it to make terrible recommendations at times!).

The effects of the Netflix Prize were felt right away. Not only did many people flood into the field, but they produced a flood of research papers presenting small improvements to recommender systems (usually evaluated based on RMSE) with the argument that even small improvements are significant because ensemble algorithms will be able to squeeze out the most from each improvement. The first ACM Recommender Systems conference in 2007 had 120 attendees and 16 papers – four on interaction and user issues, four on trust and privacy, four on core collaborative filtering algorithms, and four on machine learning algorithms (half focused on user issues as well). By 2009 the conference had over 300 attendees and only two out of 24 papers that focused primarily on user issues.

The response was also quick. Those committed to human-centered recommender systems research and practice organized workshops, promulgated best practices, taught tutorials, conducted studies, and started actively working to ensure that recommender systems were not simply "yet another machine learning problem" to be solved by people with no idea what recommender systems were actually used for in practice. Some examples of the work of this period include: McNee et al. made the case for how accuracy metrics have hurt recommender systems (2006a) and presented an analytical model for human-recommender interaction (2006b) that maps users to recommender algorithms through a set of human-recommender interaction pillars (dialogue, personality, and task) and a set of associated metrics; Ge et al. (2010) more formally developed measures of coverage and serendipity, showing how the field could move beyond accuracy in quantitative evaluation; Pu et al. (2011, 2012) developed and disseminated a state-of-the-art framework for user-centric evaluation of recommender systems, looking at the usability, usefulness, interface and interaction qualities, user satisfaction, and influence of the above on user behavioral intentions. This model borrows an approach based on validated survey instruments; Knijnenburg et al. (2011, 2012) developed a framework for evaluating the user experience of recommender systems through a set of subjective system aspects and experience constructs. This model was validated using field trials and analyzed with structural equation modeling to provide a greater focus on understanding the aspects of recommender systems that lead to particular user experiences.

And at a more detailed level, researchers took a new look at the wide range of factors within a recommender system.

Several researchers looked at *ratings and rating scales*. Amatriain et al. (2009) looked at the natural noise in user ratings, showing how strategic re-rating can be even more useful than obtaining new ratings on unseen items. Sparling and Sen (2011) studied the time, cognitive effort, and user satisfaction of four different ratings scales. Kluver et al. (2012) built on that study and took an information theoretical approach to understand how much information about user preference was captured for each rating of different scales. Building on early systems (such as the Zagat Guides, where users separately rate food, decor, and service) researchers explored the utility of multi-criteria ratings. Adomavicius et al. (2011) provide a review and framework, including algorithms for incorporating multi-dimensional ratings into recommender systems producing a single ranking or prediction. Fuchs and Zanker (2012) studied TripAdvisor's multi-dimensional ratings and found that while the seven specific rating dimensions captured most of the signal of the overall rating, the individual weights varied with customer segment.

Extensive research also explored *context-aware recommendation*. While this thread started with early work on tourism recommendation (recommendations based on where you are, what time it is, current weather), it soon broadened to consider numerous factors including information about who else is present, what activities the user is carrying out, etc. Adomavicius et al. (2005) explains the core multidimensional approach for context-aware recommendation. Such context-awareness has been a major topic in the field ever since.

Several efforts have also been made to address the user *cold start experience*. Rather early, Rashid et al. (2002) explored the question of which items users should be asked to rate as part of start-up (a balance between entropy and popularity, so as to have informational value but not be frustrating) and McNee et al. (2003) explored the tradeoff between user-directed and system-directed initiation (user-directed takes more time, but also created greater satisfaction and loyalty). Since then, however, the idea of prompting users for a list of ratings fell into disfavor. Commercial sites generally started with generic or stereotyped profiles rather than risk losing a customer, and would then build those profiles from implicit data. Vig et al. (2009) showed how tags–data that many users provide as part of social interaction–could be used as a form of preference indication, indeed finding tag-based algorithms to outperform rating-based ones. Chang et al. (2015) took a different approach, evaluating interfaces where users express preference for clusters of content–a process that does not require finding individual items the user can rate. They found high satisfaction and half the start-up time of traditional item rating. Similarly, Graus and Willemsen (2015) studied choice-based preference elicitation as an alternative for initial ratings. They found that users could navigate through a series of choices structured along the latent feature dimensions of the item space.

All this time, the core algorithms of recommender systems continued to evolve. Item-item correlation mostly gave way to latent-factor models (Koren et al. 2009). These models started as an attempt to factor the ratings matrix (which of course, was highly sparse) but quickly evolved to simply estimate the underlying dimensions (and associated vectors for each user and item). Through various optimization techniques (gradient descent, alternating least squares) we found good approximations that modeled only the present ratings. Other researchers focused on the challenge that rated items are not chosen at random and developed algorithms to address these selection effects. And while latent factor models are still highly prevalent, other machine learning approaches (including neural network approaches, reinforcement learning, and the full suite of machine learning approaches) are now being applied to recommendation.

TODAY AND BEYOND: TOWARDS A UNIFIED FIELD (2017)

As we look at recent years, the field of recommender systems has been driving towards consensus. The combination of research and industry experiences has made it clear that accuracy alone is not enough. Nor is it sufficient to carry out isolated user studies. Rather, as a field we know we need frameworks to bring together human-centered recommender systems research with the best machine learning algorithms to achieve scalable, efficient human-centered recommender systems.

This holistic approach was articulated in Ferro et al. (2018), the result of a cross-disciplinary workshop bringing together researchers in information retrieval, recommender systems, and natural language processing to explore how to move those fields towards more predictable performance. The end result was an end-to-end model, much like McNee's HRI model but more general, that linked measurable outcomes to properties of algorithms and of users/tasks/data. Key to that model is a set of intermediate variables (some not observable) that can be identified through research (or perhaps, 1 day through sufficiently rich machine learning).

Several examples of such next-generation approaches have emerged in recent years, and many more are on the near horizon:

Understanding non-selection as an input. Zhao et al. (2018b) builds a rich model for interpreting user's failure to select a recommended item. Starting with eye-tracking analysis to better understand the likely scan order on the screen (Zhao et al. 2016), the work then incorporates survey data, estimates of familiarity, and the context of recommendation to predict whether this recommendation should be repeated again soon or suppressed for a while. Finally, Zhao et al. (2019) test incorporating this non-selection data into machine learning models to evaluate the performance improvements.

Modeling user perceptions of algorithmic differences. Ekstrand et al. (2014) conducts an extensive user study of recommender algorithms to understand how users perceive their performance (against objective measures), including dimensions of accuracy, diversity, novelty, personalization, and overall satisfaction. The work builds a structural equation model showing the surprising result that excessive novelty actually hurts satisfaction. More generally, the model shows the factors that should be included in an analysis of algorithm performance. This work was followed by a field study (Ekstrand et al. 2015) that confirmed that users given the choice to choose their own algorithm actually select the algorithms identified by the model.

Understanding choice overload. Building on an extensive literature on choice overload (essentially, the seeming contradiction that people are made less well off by having additional choices), Willemsen et al. (2016) studied choice overload in the context of recommender systems. They looked in particular at the question of whether the diversity of items recommended affected the effort required to make a choice; they found both that more diverse lists removed effort and that diversity led to higher-satisfaction choices, which were not always the highest-scoring choices.

Machine learning for different objectives. Zhao et al. (2018a) showed how machine learning recommenders can be designed to meet different objectives; in particular, they showed that a recommender built to optimize for user engagement (rather that predictive accuracy) leads to recommendations that increase subsequent user engagement (compared with predictive accuracy recommenders). Wen et al. (2019) similarly showed how to improve performance by incorporating post-click data into post-click-aware ranking metrics.

Broadening interaction. While off-line analysis may have steered the field away from interaction, the future clearly involves much more thoughtful design and evaluation of interactive interfaces for recommendation. Jannach et al. (2020) surveys conversational recommender systems, informed by the recent booming of chatbot interaction.

This path seems highly promising. The future of human-centered recommender systems will depend on the following key factors: (a) Good human-centered science to understand how people make decisions, express judgments, and otherwise take on tasks related to recommendation (consider as an example Rook et al. 2020, a detailed three-factor study of user engagement); (b) ongoing advances in machine learning that allow us to digest increasing amounts of user data, item data, preference data, and context data in an effort to product high-quality recommendations; (c) continued research on real applications that allow us to have data that incorporates diverse contexts including interaction modalities (voice/audio vs. text vs. visual interaction) and decision nature (health/habit, low- vs. high-stakes, etc.); (d) realization that even as such advances lead to better recommendation, the field still must continue to adopt rigorous methods for evaluating the user experience and user satisfaction, such as the structural equation modeling examples we noted; and (e) metrics and measures that allow the scientific findings to be translated into a form that the machine learning systems can ingest, process, and optimize for.

We believe this future is already being created and will continue to push forward the advancement of human-centered recommender systems.

REFERENCES

Adomavicius, G., N. Manouselis, and YO Kwon. 2011. "Multi-Criteria Recommender Systems." In *Recommender Systems Handbook*, edited by Paul B Kantor, Francesco Ricci, Lior Rokach and Bracha Shapira. Boston, MA: Springer https://doi.org/10.1007/978-0-387-85820-3_24

Adomavicius, G., R. Sankaranarayanan, S. Sen, and A. Tuzhilin. 2005. "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach." *ACM Transactions on Information Systems* 23(1): 103–45. https://doi.org/10.1145/1055709.1055714

Baltrunas, L., T. Makcinskas, and F. Ricci. 2010. "Group Recommendations with Rank Aggregation and Collaborative Filtering." In Proceedings of the fourth ACM conference on Recommender systems (RecSys '10). Association for Computing Machinery, New York, NY, USA, 119–26. https://doi.org/10.1145/1864708.1864733

Burke, R., K. Hammond, and E. Cooper. 1996. "Knowledge-Based Navigation of Complex Information Spaces." In Proceedings of the 13th National Conference on Artificial Intelligence, Vol. 1: 462–8.

Chang, S., F. M. Harper, and L. Terveen. 2015. "Using Groups of Items for Preference Elicitation in Recommender Systems." In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15). Association for Computing Machinery, New York, NY, USA, 1258–69. https://doi.org/10.1145/2675133.2675210

Chen, L., M. de Gemmis, A. Felfernig, P. Lops, F. Ricci, and G. Semeraro. 2013. "Human Decision Making and Recommender Systems." *ACM Transactions on Interactive Intelligent Systems* 3: 1, Article 17 (October 2013).https://doi.org/10.1145/2533670.2533675

Cosley, D., S. K. Lam, I. Albert, J.A. Konstan, and J. Riedl. 2003. "Is seeing believing? How Recommender System Interfaces Affect Users' Opinions." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03). Association for Computing Machinery, New York, NY, USA, 585–92. https://doi.org/10.1145/642611.642713

Ekstrand, M. D., D. Kluver, F. M. Harper, M.C. Willemsen, and J.A. Konstan. 2014. "User Perception of Differences in Recommender Algorithms." In Proceedings of the 8th ACM Conference on Recommender systems (RecSys '14). Association for Computing Machinery, New York, NY, USA, 161–8. https://doi.org/10.1145/2645710.2645737

Ekstrand, M. D., D. Kluver, F. M. Harper, and J.A. Konstan. 2015. "Letting Users Choose Recommender Algorithms: An Experimental Study." In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15). Association for Computing Machinery, New York, NY, USA, 11–8. https://doi.org/10.1145/2792838.2800195

Ferro, N., N. Fuhr, G. Grefenstette, J. A. Konstan, P. Castells, E. M. Daly, T. Declerck, et al. 2018. "From Evaluating to Forecasting Performance: How to Turn Information Retrieval, Natural Language Processing and Recommender Systems into Predictive Sciences (Dagstuhl Perspectives Workshop 17442)." *Dagstuhl Manifestos* 7(1): 96–139.

Fuchs, M., and M. Zanker. 2012. "Multi-Criteria Ratings for Recommender Systems: An Empirical Analysis in the Tourism Domain." In Proceedings of the 13th International Conference on Electronic Commerce and Web Technologies, Springer, Vienna, Austria.

Ge, M., C. Delgado-Battenfeld, and D. Jannach. 2010. "Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity." In Proceedings of the fourth ACM conference on Recommender systems (RecSys '10). Association for Computing Machinery, New York, NY, USA, 257–60. https://doi.org/10.1145/1864708.1864761

Goldberg, D., D. Nichols, B. M. Oki, and D. Terry. 1992. "Using Collaborative Filtering to Weave an Information Tapestry." *Communications of the ACM* 35(12): 61–70. https://doi.org/10.1145/138859.138867

Goldberg, K., T. Roeder, D. Gupta, and C. Perkins. 2001. "Eigentaste: A Constant Time Collaborative Filtering Algorithm." *Information Retrival Journal* 4(2): 133–51. https://doi.org/10.1023/A:1011419012209

Graus, M. P., and M. C. Willemsen. 2015. "Improving the User Experience during Cold Start through Choice-Based Preference Elicitation." In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15). Association for Computing Machinery, New York, NY, USA, 273–6. https://doi.org/10.1145/2792838.2799681

Herlocker, J. L., J. A. Konstan, A. Borchers, and J. T. Riedl. 1999. "An Algorithmic Framework for Performing Collaborative Filtering." In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99). Association for Computing Machinery, New York, NY, USA, 230–7. https://doi.org/10.1145/312624.312682

Herlocker, J. L., J. A. Konstan, and J. T. Riedl. 2000. "Explaining Collaborative Filtering Recommendations." In Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW '00). Association for Computing Machinery, New York, NY, USA, 241–50. https://doi.org/10.1145/358916.358995

Herlocker, J. L., J. A. Konstan, L. G. Terveen, and J. T. Riedl. 2004. "Evaluating Collaborative Filtering Recommender Systems." *ACM Transactions on Information Systems* 22(1): 5–53. https://doi.org/10.1145/963770.963772

Hill, W., L. Stead, M. Rosenstein, and G. Furnas. 1995. "Recommending and Evaluating Choices in a Virtual Community of Use." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95). ACM Press/Addison-Wesley Publishing Co., USA, 194–201. https://doi.org/10.1145/223904.223929

Jameson, A., and B. Smyth. 2007. "Recommendation to Groups." In *The Adaptive Web*, edited by P. Brusilovsky, A. Kobsa, and W. Nejdl. LNCS 4321: pp. 596–627.

Jannach, D., A. Manzoor, W. Cai, and L. Chen. 2020. A Survey on Conversational Recommender Systems. ArXiv abs/2004.00646.

Kluver, D., T. T. Nguyen, M. Ekstrand, S. Sen, and J. Riedl. 2012. "How Many Bits Per Rating?" In Proceedings of the sixth ACM conference on Recommender systems (RecSys '12). Association for Computing Machinery, New York, NY, USA, 99–106. https://doi.org/10.1145/2365952.2365974

Knijnenburg, B. P., M. C. Willemsen, and A. Kobsa. 2011. "A Pragmatic Procedure to Support the User-Centric Evaluation of Recommender Systems." In Proceedings of the fifth ACM conference on Recommender systems (RecSys '11). Association for Computing Machinery, New York, NY, USA, 321–4. https://doi.org/10.1145/2043932.2043993

Knijnenburg, B. P., M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. 2012. “Explaining the User Experience of Recommender Systems.” *User Modeling and User-Adapted Interaction* 22: 441–504. https://doi.org/10.1007/s11257-011-9118-4

Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. 1997. “GroupLens: Applying Collaborative Filtering to Usenet News.” *Communications of the Acm* 40(3): 77–87. https://doi.org/10.1145/245108.245126

Koren, Y., R. Bell, and C. Volinsky. 2009. “Matrix Factorization Techniques for Recommender Systems.” *Computer* 42(8): 30–7. https://doi.org/10.1109/MC.2009.263.

Lieberman, H. 1995. “Letizia: An Agent that Assists Web Browsing.” In Proceedings of the International Joint Conference on Artificial Intelligence.

Maltz, D., and K. Ehrlich. 1995. “Pointing the Way: Active Collaborative Filtering.” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’95). ACM Press/Addison-Wesley Publishing Co., USA, 202–9. https://doi.org/10.1145/223904.223930

McCarthy, J. F., and T. D. Anagnost. 1998. “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). Association for Computing Machinery, New York, NY, USA, 363–72. https://doi.org/10.1145/289444.289511

McNee, S. M., S. K. Lam, J. A. Konstan, and J. Riedl. 2003. “Interfaces for Eliciting New User Preferences in Recommender Systems.” In Proceedings of the 9th international conference on User modeling (UM’03). Springer-Verlag, Berlin, Heidelberg, 178–87.

McNee, S. M., J. Riedl, and J. A. Konstan. 2006a. “Being Accurate is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems.” In CHI ’06 Extended Abstracts on Human Factors in Computing Systems (CHI EA ’06). Association for Computing Machinery, New York, NY, USA, 1097–101. https://doi.org/10.1145/1125451.1125659

McNee, S. M., J. Riedl, and J. A. Konstan. 2006b. “Making Recommendations Better: An Analytic Model for Human-Recommender Interaction.” In CHI ’06 Extended Abstracts on Human Factors in Computing Systems (CHI EA ’06). Association for Computing Machinery, New York, NY, USA, 1103–8. https://doi.org/10.1145/1125451.1125660

Morita, M., and Y. Shinoda. 1994. “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 (SIGIR ’94). Springer-Verlag, Berlin, Heidelberg, 272–81. https://doi.org/10.5555/188490.188583

O’Connor, M., D. Cosley, J.A. Konstan, and J. Riedl. 2001. “PolyLens: A Recommender System for Groups of Users.” In *ECSCW 2001*, edited by W. Prinz, M. Jarke, Y. Rogers, K. Schmidt and V. Wulf. Dordrecht: Springer. https://doi.org/10.1007/0-306-48019-0_11

Pazzani, M., J. Muramatsu, and D. Billsus. 1996. “Syskill & Webert: Identifying Interesting Web Sites.” In Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1 (AAAI’96). AAAI Press, 54–61.

Pu, P., L. Chen, and R. Hu. 2011. “A User-Centric Evaluation Framework for Recommender Systems.” In Proceedings of the fifth ACM conference on Recommender systems (RecSys ’11). Association for Computing Machinery, New York, NY, USA, 157–64. https://doi.org/10.1145/2043932.2043962

Pu, P., L. Chen, and R. Hu. 2012. “Evaluating Recommender Systems from the User’s Perspective: Survey of the State of the Art.” *User Modeling and User-Adapted Interaction* 22: 317–55. https://doi.org/10.1007/s11257-011-9115-7

Pu, P., B. Faltings, L Chen, J. Zhang, and P. Viappiani. 2010. “Usability Guidelines for Product Recommenders Based on Example Critiquing Research.” In *Recommender Systems Handbook*, edited by F. Ricci, L. Rokach, B. Shapira and P.B. Kantor. Springer, Chapter 16, 511–46.

Rashid, A.M., I. Albert, D. Cosley, S.K. Lam, S.M. McNee, J.A. Konstan, and J. Riedl. 2002. “Getting to Know You: Learning New User Preferences in Recommender Systems.” In Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI ’02). Association for Computing Machinery, New York, NY, USA, 127–34. https://doi.org/10.1145/502716.502737

Reilly, J., J. Zhang, L. McGinty, P. Pu, and B. Smyth. 2007. “Evaluating Compound Critiquing Recommenders: A Real-User Study.” In Proceedings of ACM Conference on Electronic Commerce (EC’07), 114–23, San Diego, USA.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. 1994. “GroupLens: An Open Architecture for Collaborative Filtering of Netnews.” In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW ’94). Association for Computing Machinery, New York, NY, USA, 175–86. https://doi.org/10.1145/192844.192905

Rook, L., A. Sabic, and M. Zanker. 2020. “Engagement in Proactive Recommendations.” *Journal of Intelligent Information Systems* 54: 79–100. https://doi.org/10.1007/s10844-018-0529-0

Sarwar, B., G. Karypis, J. Konstan, and J. Riedl. 2000. “Analysis of Recommendation Algorithms for E-Commerce.” In Proceedings of the 2nd ACM Conference on Electronic Commerce (EC ’00). Association for Computing Machinery, New York, NY, USA, 158–67. https://doi.org/10.1145/352871.352887

Sarwar, B., G. Karypis, J. Konstan, and J. Riedl. 2001. “Item-Based Collaborative Filtering Recommendation Algorithms.” In Proceedings of the 10th International conference on World Wide Web (WWW ’01). Association for Computing Machinery, New York, NY, USA, 285–95. https://doi.org/10.1145/371920.372071

Shardanand, U., and P. Maes. 1995. “Social Information Filtering: Algorithms for Automating “word of Mouth.” In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’95). ACM Press/Addison-Wesley Publishing Co., USA, 210–7. https://doi.org/10.1145/223904.223931

Sparling, E. I., and S. Sen. 2011. “Rating: How Difficult Is It?” In Proceedings of the Fifth ACM Conference on Recommender Systems (RecSys ’11). Association for Computing Machinery, New York, NY, USA, 149–56. https://doi.org/10.1145/2043932.2043961

Swearingen, K., and R. Sinha. 2001. “Beyond Algorithms: An HCI Perspective on Recommender Systems.” In Proceedings of the SIGIR 2001 Workshop on Recommender Systems.

Taijala, T. T., M. C. Willemsen, and J. A. Konstan. 2018. “MovieExplorer: Building an Interactive Exploration Tool From Ratings and Latent Taste Spaces.” In Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC ’18). Association for Computing Machinery, New York, NY, USA, 1383–92. https://doi.org/10.1145/3167132.3167281

Terveen, L., W. Hill, B. Amento, D. McDonald, and J. Creter. 1997. “PHOAKS: A System for Sharing Recommendations.” *Communications of the Acm* 40(3): 59–62. https://doi.org/10.1145/245108.245122

Tintarev, N., and J. Masthoff. 2007. “A Survey of Explanations in Recommender Systems.” In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop (ICDEW ’07). IEEE Computer Society, USA, 801–10. https://doi.org/10.1109/ICDEW.2007.4401070

Wen, H., L. Yang, and D. Estrin. 2019. “Leveraging Post-Click Feedback for Content Recommendations.” In Proceedings of the 13th ACM Conference on Recommender Systems (RecSys ’19). Association for Computing Machinery, New York, NY, USA, 278–86. https://doi.org/10.1145/3298689.3347037

Zhao, Q., S. Chang, F. M. Harper, and J. A. Konstan. 2016. “Gaze Prediction for Recommender Systems.” In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys ’16). Association for Computing Machinery, New York, NY, USA, 131–8. https://doi.org/10.1145/2959100.2959150

Zhao, Q., M. C. Willemsen, G. Adomavicius, F. M. Harper, and J. A. Konstan. 2018a. “Explicit or implicit feedback? Engagement or Satisfaction? A Field Experiment on Machine-Learning-Based Recommender Systems.” In Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC ’18). Association for Computing Machinery, New York, NY, USA, 1331–40. https://doi.org/10.1145/3167132.3167275

Zhao, Q., M. C. Willemsen, G. Adomavicius, F. M. Harper, and J. A. Konstan. 2018b. “Interpreting User Inaction in Recommender Systems.” In Proceedings of the 12th ACM Conference on Recommender Systems (RecSys ’18). Association for Computing Machinery, New York, NY, USA, 40–8. https://doi.org/10.1145/3240323.3240366

Zhao, Q., M. C. Willemsen, G. Adomavicius, F. M. Harper, and J. A. Konstan. 2019. “From Preference Into Decision Making: Modeling User Interactions in Recommender Systems.” In Proceedings of the 13th ACM Conference on Recommender Systems (RecSys ’19). Association for Computing Machinery, New York, NY, USA, 29–33. https://doi.org/10.1145/3298689.3347065

Ziegler, C. -N., S. M. McNee, J. A. Konstan, and G. Lausen. 2005. “Improving Recommendation Lists Through Topic Diversification.” In Proceedings of the 14th international conference on World Wide Web (WWW ’05). Association for Computing Machinery, New York, NY, USA, 22–32. https://doi.org/10.1145/1060745.1060754

AUTHOR BIOGRAPHIES

Joseph A. Konstan is Distinguished McKnight Professor and Distinguished University Teaching Professor in Computer Science and Engineering at the University of Minnesota, where he also serves as Associate Dean for Research in the College of Science and Engineering. He has been conducting research on recommender systems since 1995, and has focused that research on understanding how to designing recommender algorithms and interfaces to enhance user experience.

Loren Terveen is Distinguished McKnight Professor and Associate Department Head of Computer Science and Engineering at the University of Minnesota. He has been conducting research on recommender systems and human computer interaction since the mid-1990s, focusing on interaction designs that simplify and improve key recommender systems tasks.

How to cite this article: Konstan, J.A., and L.G. Terveen. 2021. “Human-centered recommender systems: Origins, advances, challenges, and opportunities.” *AI Magazine*. 42: 31–42. https://doi.org/10.1609/aaai.12015