Recommender Systems: An Overview

*Robin Burke, Alexander Felfernig,
and Mehmet H. Göker*

■ *Recommender systems are tools for interacting with large and complex information spaces. They provide a personalized view of such spaces, prioritizing items likely to be of interest to the user. The field, christened in 1995, has grown enormously in the variety of problems addressed and techniques employed, as well as in its practical applications. Recommender systems research has incorporated a wide variety of artificial intelligence techniques including machine learning, data mining, user modeling, case-based reasoning, and constraint satisfaction, among others. Personalized recommendations are an important part of many online e-commerce applications such as Amazon.com, Netflix, and Pandora. This wealth of practical application experience has provided inspiration to researchers to extend the reach of recommender systems into new and challenging areas. The purpose of the articles in this special issue is to take stock of the current landscape of recommender systems research and identify directions the field is now taking. This article provides an overview of the current state of the field and introduces the various articles in the special issue.*

The prototypical use case for a recommender system occurs regularly in e-commerce settings. A user, Jane, visits her favorite online bookstore. The homepage lists current best-sellers and also a list containing recommended items. This list might include, for example, a new book published by one of Jane's favorite authors, a cookbook by a new author, and a supernatural thriller. Whether Jane will find these suggestions useful or distracting is a function of how well they match her tastes. Is the cookbook for a style of cuisine that she likes (and is it different enough from ones she already owns)? Is the thriller too violent? A key feature of a recommender system therefore is that it provides a personalized view of the data, in this case, the bookstore's inventory. If we take away the personalization, we are left with the list of best-sellers — a list that is independent of the user. The aim of the recommender system is to lower the user's search effort by listing those items of highest utility, those that Jane might be most likely to purchase. This, of course, is beneficial to Jane as well as the e-commerce store owner.

Recommender systems research encompasses scenarios like this and many other information access environments in which a user and store owner can benefit from the presentation of personalized options. The field has seen a tremendous expansion of interest in the past decade, catalyzed in part by the Netflix Prize (Bennett and Lanning 2007) and evidenced by the rapid growth of the annual Association for Computing Machinery (ACM) Recommender Systems conference. At this point, it is worthwhile to take stock, to consider what distinguishes recommender systems research from other related areas of research in artificial intelligence, and to examine the field's successes and new challenges.

What Is a Recommender System?

The definition of a recommender system has evolved over the past 14 years. In Resnick and Varian's seminal article (1997), the authors describe a recommender system as follows:

In a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases the primary transformation is in the aggregation; in others the system's value lies in its ability to make good matches between the recommenders and those seeking recommendations.

Note that this definition places the emphasis on the recommender systems as supporting the collaboration between users. Later researchers have expanded the definition to include systems that suggest items of interest, regardless of how those recommendations are produced: "any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options" (Burke 2002). This more general definition was formalized by Adomavicius and Tuzhilin (2005):

More formally, the recommendation problem can be formulated as follows: Let C be the set of all users and let S be the set of all possible items that can be recommended. Let u be a utility function that measures the usefulness of item s to user c, that is, u: C x S $\Rightarrow$ R, where R is a totally ordered set (for example, nonnegative integers or real numbers within a certain range). Then, for each user $c \in$ C, we want to choose such item $s' \in$ S that maximizes the user's utility.

This definition opens up the field of recommender systems to any application that computes a user-specific utility, encompassing many problems commonly thought of as database or information retrieval applications. Even this broad definition may be too narrow as some recommenders may operate on configurations — as opposed to a fixed set S of all items — and others make recommendations for groups (utility is computed for a subset $C^* \subset C$ of the users rather than a single user). The definition may also be a bit misleading in that many recommender systems do not explicitly calculate utilities when they produce a ranked list of recommended items. The authors are careful to say that the goal is to choose the items with the best utility, not necessarily to compute the utility in some explicit way.

From these considerations, two basic principles stand out that distinguish recommender systems research:

A recommender system is *personalized*. The recommendations it produces are meant to optimize the experience of one user, not to represent group consensus for all.

A recommender system is intended to help the user select among discrete options. Generally the items are already known in advance and not generated in a bespoke fashion.

The personalization aspect of recommender systems distinguishes this line of research most strongly from what is commonly understood as research in search engines and other information retrieval applications. In a search engine or other information retrieval system, we expect the set of results relevant to a particular query to be the same regardless of who issued it.[1] Many recommender systems achieve personalization by maintaining profiles of user's activity (long term or short term) or stated preferences (Schafer et al. 2007). Others achieve a personalized result through conversational interaction (McGinty and Reilly 2011).

A Recommender System Typology

Recommender systems research is characterized by a common problem area rather than a common technology or approach. An examination of the past four ACM Recommender System conferences shows that a wide variety of research approaches have been applied to the recommender systems problem, from statistical methods to ontological reasoning, and a wide variety of problems have been tackled, from choosing consumer products to finding friends and lovers.

One lesson that has been learned over the past years of recommender systems research is that the application domain exerts a strong influence over the types of methods that can be successfully applied. Domain characteristics like the persistence of the user's utility function have a big impact: for example, a user's taste in music may change slowly but his or her interest in celebrity news stories may fluctuate much more. Thus, the reliability of preferences gathered in the past may vary. Similarly, some items, such as books, are available for recommendation and consumption over a long period of time—often years. On the other hand, in a technological domain, such as cell phones or cameras, old products become rapidly obsolete and cannot be usefully recommended. This is also true of areas where timeliness matters such as news and cultural events. See Burke and Ramezani (2011) for a more complete description of the factors that influence the choice of a recommendation approach.

It is not surprising therefore that there are multiple strands of research in recommender systems, as researchers tackle a variety of recommendation domains. To unify these disparate approaches, it is useful to consider the AI aspects of recommendation, in particular, the knowledge basis underlying a recommender system.

Knowledge Sources

Every AI system draws on one or more sources of knowledge in order to do its work. A supervised machine-learning system, for example, would have a labeled collection of data as its primary knowledge source, but the algorithm and its parameters can be considered another implicit
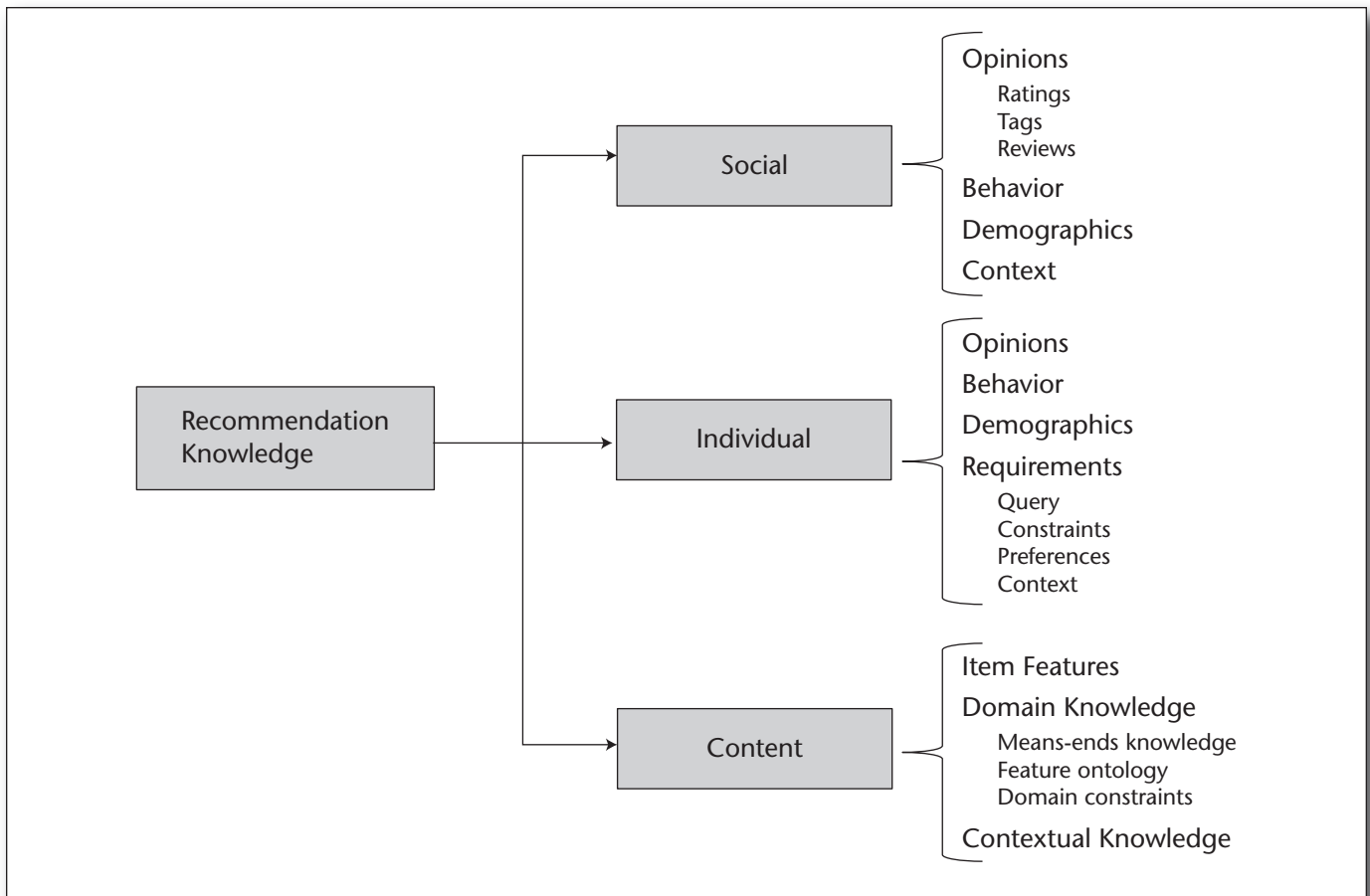
Figure 1. Taxonomy of Knowledge Sources in Recommendation.

After Felfernig and Burke (2008).

kind of knowledge that is brought to bear on the classification task.

Recommendation algorithms can also be classified according to the knowledge sources that they use. Figure 1 shows the taxonomy of knowledge sources used in Felfernig and Burke (2008). There are three basic types of knowledge: (1) social knowledge about the user base in general, (2) individual knowledge about the particular user for whom recommendations are sought (and possibly knowledge about the specific requirements those recommendations need to meet), and finally (3) content knowledge about the items being recommended, ranging from simple feature lists to more complex ontological knowledge and means-ends knowledge that enable the system to reason about how an item can meet a user's needs.

Different recommendation approaches draw from different parts of this spectrum of knowledge sources. The terms of the Netflix Prize competition made available only opinions in the form of ratings but no requirements or demographic information about users (Bennet and Lanning 2007). Good domain knowledge is notoriously difficult to

assemble in this domain because of the complexity of representing and reasoning about narrative content, directorial style, and so on. The problem thus lent itself to a mathematical approximation technique working exclusively from ratings both social and individual (Bell, Koren, and Volinsky 2007).

By contrast, the problem of recommending investment options reported in Felfernig and Burke (2008) can benefit from detailed knowledge about the customers' income and financial status, the other items in their portfolio, and their attitude toward risk. Other users' opinions and choices may be useful but are insufficient to make high-quality recommendations in this domain.

Research Questions in Recommender Systems

The most prominent technique in recommendation is *collaborative recommendation* (Schafer et al. 2007). The basic insight for this technique is a sort of continuity in the realm of taste — if users Alice

and Bob have the same utility for items 1 through *k,* then the chances are good that they will have the same utility for item *k* + 1. Usually, these utilities are based on ratings that users have supplied for items with which they are already familiar.

The key advantage of collaborative recommendation is its simplicity. The problem of computing utility is transformed into the problem of extrapolating missing values in the ratings matrix, the sparse matrix where each user is a row, each item a column, and the values are the known ratings. This insight can be operationalized in a number of ways. Originally, nearest-neighbor techniques were applied to find neighborhoods of like-minded peers. However, matrix factorization and other dimensionality-reduction techniques are now recognized as superior in accuracy (Bell and Koren 2007).

Some problems with collaborative recommendation are well established. New items cannot be recommended without relying on some additional knowledge source. Extrapolation depends on having some values from which to project. Indeed, sparsely rated items in general present a problem because the system lacks information on which to base predictions. By the same token, users who have supplied few ratings will receive noisier recommendations than those with more substantial histories. The problems of new users and new ratings are collectively known as the "cold start" problem in collaborative recommendation.

The distribution of ratings and user preferences in many consumer taste domains is fairly concentrated: a small number of "blockbuster" items receive a great deal of attention, and there are many, many rarely rated items.

Malicious users may be able to generate large numbers of pseudonymous "sybil" profiles and use them to bias the recommendations of the system in one way or another.

There is still a great deal of algorithmic research focused on the problems of collaborative recommendation: more accurate and efficient estimates of the ratings matrix, better handling of new users and new items, and the extension of the basic collaborative recommendation idea to new types of data including multidimensional ratings and user-generated tags, among others.

Content-Based Recommendation

Before the advent of collaborative recommendation in the 1990s, earlier research in personalized information access had concentrated on combining knowledge about items with information about user's preferences in order to locate appropriate items. Both Rich's early (1979) work on Grundy (book recommendation) and Rocchio's (1971) method (information retrieval) can now be seen as early examples of recommender systems although the term had not yet been coined. This approach, because of its reliance on the content knowledge source, in particular, item features, has come to be known as *content-based recommendation.*

Content-based recommendation is closely linked with supervised machine learning. We can view the problem as one of learning a set of user-specific classifiers where the classes are "useful to user *X"* and "not useful to user *X."*

One of the key issues in content-based recommendation is feature quality. The objects to be recommended need to be described so that meaningful learning of user preferences can occur. Ideally, every object would be described at the same level of detail and the feature set would contain descriptors that correlate with the discriminations made by users. Unfortunately, this is often not the case. Descriptions may be partial or some parts of the object space may be described in greater detail than others.

The match between the feature set and the user's utility function also needs to be good. One of the strengths of the popular Pandora streaming music service is that the feature set it uses for musical selections is manually chosen by music-savvy listeners. Automatic music processing is not yet good enough to reliably extract features like "bop feel" from a Charlie Parker recording.

In addition to the development and application of new learning algorithms for the recommendation task, research in content-based recommendation also examines the problem of feature extraction in different domains. A further subtype of content-based recommendation is knowledge-based recommendation, in which the reliance on item features is extended to other kinds of knowledge about products and their potential utilities for users. An example of this kind of system is the investment recommender mentioned earlier that has to know about the risk profiles and tax consequences of different investments and how these interact with the financial position of the investor.

As with other knowledge-based systems, knowledge acquisition, maintenance, and validation are key issues. Also, since knowledge-based recommenders can make use of detailed requirements from the user, user interface research has been paramount in developing knowledge-based recommenders that do not place too much of a burden on users.

Evaluation

Because of the difficulties of running large-scale user studies, recommender systems have conventionally been evaluated on one or both of the following measures:

Prediction accuracy. How well do the system's predicted ratings compare with those that are known, but withheld?

Precision of recommendation lists. Given a short list of recommendations produced by the system (typically all a user would have patience to examine), how many of the entries match known "liked" items?

Both of these conventional measures are deficient in some key respects and many of the new areas of exploration in recommender systems have led to experimentation with new evaluation metrics to supplement these common ones.

One of the most significant problems occurs because of the long-tailed nature of the ratings distribution in many data sets. A recommendation technique that optimizes for high accuracy over the entire data set therefore contains an implicit bias toward well-known items, and therefore may fail to capture aspects of utility related to novelty. An accurate prediction on an item that the user already knows is inherently less useful than a prediction on an obscure item. To address this issue, some researchers are looking at the balance between accuracy and diversity in a set of recommendations and are working on algorithms that are sensitive to item distributions.

Another problem with conventional recommender systems evaluation is that it is essentially static. A fixed database of ratings is divided into training and test sets and used to demonstrate the effectiveness of an algorithm. However, the user experience of recommendation is quite different. In an application like movie recommendation, the field of items is always expanding; a user's tastes are evolving; new users are coming to the system. Some recommendation applications require that we take the dynamic nature of the recommendation environment into account and evaluate our algorithms accordingly.

Another area of evaluation that is relatively underexamined is the interaction between the utility functions of the store owner and the user, which necessarily look quite different. Owners implement recommender systems in order to achieve business goals, typically increased profit. The owner therefore may prefer an imperfect match with a high profit margin to a perfect match with limited profit. On the other hand, a user who is presented with low utility recommendations may cease to trust the recommendation function or the entire site. Owners with high volume sites can field algorithms side by side in randomized trials and observe sales and profit differentials, but such results rarely filter out into the research community.

This Special Issue

The aim of this special issue is to give a brief overview of the history and current status of recommender system research, to describe the current state of recommender systems in practical use, and

to highlight new directions in recommender systems research that may be of interest to *AI Magazine* readers. The articles have been chosen to illuminate the state of the art in recommendation and to illustrate some of the challenges that must be faced in extending current techniques in recommendation to meet new domains and new requirements.

The special issue starts off with an article by Francisco J. Martin, Justin Donaldson, Adam Ashenfelter, Marc Torrens, and Rick Hangartner. The authors, all of whom have considerable experience in both recommender systems research and industrial system development, give a historical overview of the field and describe their view of what the future holds for recommender systems research. The second article is by Susan Aldrich, an analyst from the Patricia Seybold Group. This article looks at the commercial recommender systems landscape from the perspective of an industry analyst and describes how commercial recommender systems are designed, deployed, and evaluated. Given the maturity of the field in terms of commercial applications, we found it important to convey the industrial viewpoint here.

Next, Barry Smyth, Jill Freyne, Peter Briggs, and Maurice Coyle explain their work on collaborative web search in the HeyStaks system.[2] The authors describe how the standard, one-size-fits-all web search can be made more personalized by using information about searches done by peers or collaborators. Enhancing search with this social aspect increases the quality of the results and makes them more relevant.

The theme of social recommendation continues with the article by Robin Burke, Jonathan Gemmel, Andreas Hotho, and Robert Jäschke that highlights how recommender applications can leverage data coming from social applications and how recommendation algorithms need to advance to meet the new challenges these systems pose.

Òscar Celma and Paul Lamere describe recommender systems in the field of music recommendation. They analyze existing approaches to music recommendation, explain these recommendation styles using data from Last.fm, and point out research challenges that need to be addressed.

Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alex Tuzhilin analyze the impact of context on recommender systems. Context-aware recommendation goes beyond what we normally would consider personalization and takes additional information such as the environment and conditions the user is operating under into account. The authors analyze how these contextual factors can change and how they affect system design. They provide examples of implementations and describe challenges and future research directions for this field.

Software engineering is a relatively new application area for recommender systems. Bamshad Mobasher and Jane Cleland-Huang show how various areas of requirements engineering can benefit from recommendation: stakeholder identification, domain analysis, requirements elicitation, and decision support. They describe various approaches from literature and point out areas for future research.

Gerhard Friedrich and Markus Zanker look at the problem of explaining the recommendations that a system gives. Such explanations may be important in securing user confidence and acceptance of recommended items. This article describes the various means by which recommender systems generate such explanations and points out open research issues.

The last article in the issue, by Andreas Falkner, Alexander Felfernig, and Albert Haag, looks at recommendation in domains with configurable products. Many of the techniques appropriate for predefined product catalogs are not appropriate for products with many configurable parts as the number of possible complete configurations is exponential. The authors discuss existing approaches to such configuration problems and open issues.

We hope that the articles in the issue provide background information for researchers that are new to the field, guidance for researchers who want to commercialize their work, and new ideas and motivation to researchers who want to expand the already impressive amount of work in the relatively young field of recommender systems.

As the editors of this issue, we would like to thank all of our authors and *AI Magazine* for their hard work and support.

Note

1. Personalized search, for example, removes this distinction, and is therefore by this definition a recommender systems application.
2. See www.heystaks.com.

References

Adomavicius, G., and Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* 17(6): 734–749.

Bell, R. M., and Koren, Y. 2007. Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining,* 43–52. Los Alamitos, CA: IEEE Computer Society.

Bell, R. M.; Koren, Y.; and Volinsky, C. 2007. The BellKor Solution to the Netflix Prize. Technical Report, AT&T Labs Research, Forham Park, NJ.

Bennett, J., and Lanning, S. 2007. The Netflix Prize. Paper presented at the KDD Cup Workshop, San Jose, CA, 12 August.

Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12(4): 331–370.

Burke, R., and Ramezani, M. 2011. Matching Recommendation Technologies and Domains. In *Recommender Systems Handbook,* ed. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, 367–386. Berlin: Springer.

Felfernig, A., and Burke, R. 2008. Constraint-Based Recommender Systems: Technologies and Research Issues. In *Proceedings of the 10th International Conference on Electronic Commerce* (ICEC '08). New York: Association for Computing Machinery.

McGinty, L., and Reilly, J. 2011. On the Evolution of Critiquing Recommenders. In *Recommender Systems Handbook,* ed. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, 419–453. Berlin: Springer.

Resnick, P., and Varian, H. R. 1997. Recommender Systems. *Communications of the ACM* 40(3): 56–58.

Rich, E. 1979. User Modeling via Stereotypes. *Cognitive Science* 3(4): 329–354.

Rocchio, J. J. 1971. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing,* ed. G. Salton, 313–323. Englewood Cliffs, NJ: Prentice-Hall.

Schafer, J.; Frankowski, D.; Herlocker, J.; and Sen, S. 2007. Collaborative Filtering Recommender Systems. In *The Adaptive Web,* ed. P. Brusilovsky, A. Kobsa, and W. Nejdl, 291–324. Lecture Notes in Computer Science 4321. Berlin: Springer.

Robin Burke is an associate professor in the College of Computing and Digital Media at Depaul University. His research interests in recommender systems include knowledge-based recommendation, hybrid recommendation, and recommendation applications for social web systems. He has published more than 140 articles and conference papers in these and related fields and is a member of the steering committee and current cochair for the ACM Recommender Systems conference.

Alexander Felfernig is a professor of applied software engineering at the Graz University of Technology. In his research he focuses on different aspects of the application of recommendation and configuration technologies in industrial environments. Felfernig is a cofounder of ConfigWorks. He is author and coauthor of more than 150 papers in international journals, conferences, and workshops. For his research he received the Heinz Zemanek Award from the Austrian Computer Society in 2009.

Mehmet H. Göker is vice president of data and analytics (CFL) at Salesforce.com. He has been implementing and deploying intelligent, human-centered applications for more than 15 years. His main interest is developing and leveraging AI technologies to solve real-world problems. Göker has a Dr.-Ing from the Darmstadt University of Technology in Germany.