Introduction To Recommender Systems: Algorithms and Evaluation

Recommender systems use the opinions of members of a community to help individuals in that community identify the information or products most likely to be interesting to them or relevant to their needs. These systems, originally referred to as collaborative filtering systems, were developed to address two challenges that could not be addressed by existing keyword-based information filtering systems. First, they addressed the problem of overwhelming numbers of *on-topic* documents—ones which would all be selected by a keyword filter—by filtering based on human judgement about the quality of those documents. Second, they addressed the problem of filtering non-text documents based on human taste. For example, the Ringo system [Shardanand and Maes, 1995] applied collaborative filtering to recommend music to individuals and later research and commercial systems applied the same techniques to other art forms.

Early research in this area focused largely on the ability of these systems to generate recommendations that were valued by the users of the system. And, indeed, these systems generated substantial enthusiasm and support from their users. In 1996, at the first of a series of workshops on collaborative filtering, it first became clear that some fairly simple algorithms (namely weighted k-nearest-neighbor algorithms applied to a sparse matrix of the ratings that users assigned to particular items or documents) worked well for several different research groups and application areas. This workshop also started using the term "Recommender Systems" and led to the publication of a special issue of *Communications of the ACM* on the topic (March 1997).

At this point, the Recommender Systems research field diverged. Substantial commercial interest focused attention on a variety of practical questions, including the speed with which recommendations could be generated, the scale of problems that could be addressed, and the assessment of the value of recommendations to the business itself or to the customers. At the same time, a broad range of machine learning researchers (broadly defined) started applying a wide variety of techniques to recommendation problems, exploring issues of improving accuracy of algorithms, better exploiting knowledge about the

Editor's address: GroupLens Research Group, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455; email: konstan@cs.umn.edu.

document/product domain, and achieving more rapid start-up for systems and users.

It was this split in the field that created the motivation for this special issue. In particular, several recommender systems researchers observed that our field was missing two key resources: (1) a single venue to serve as a guide to the different types of recommender system being developed and tested by researchers, and (2) a broad guide to evaluating recommender systems to encourage these researchers from different specialties to create comparable results. As editor of this issue, I am gratified by the strong response received to the call for paper submissions, and am further gratified that a number of articles that could not appear in this issue will likely appear in later issues of *TOIS*.

Articles in This Issue

Herlocker et al.'s article on evaluating collaborative filtering recommender systems reflects more than five years of thinking on how to conduct meaningful research evaluations of new algorithms and systems. It was prompted most directly by an observation that several papers published in the last several years have shown nearly identical performance (measured as mean absolute error between predicted values and actual user ratings) on a set of widely-distributed ratings sets. This raised the question of whether it was worthwhile to continue investigating algorithms, and whether these were all indeed equally good. This article addresses those questions in two ways. First, it experimentally evaluates a wide range of already-published metrics used by researchers to evaluate the quality of recommendations produced by their algorithms, showing that not all metrics measure the same things, but that the metrics can effectively be grouped into clusters that yield highly-correlated measurements. Second, it works forward from user tasks, assessing which evaluation mechanisms most directly reflect the suitability of a recommender for specific purposes. In doing so, it argues that accuracy measures only capture a small part of the needed understanding of a recommender system's usefulness, and that other factors such as the novelty of recommendations are often unmeasured and unreported.

Middleton's article on ontological user profiling is an excellent example of applying content-based techniques to the recommendation problem. This article describes the use of an ontology of research article topics that is used to build more effective profiles of user interests for use in a recommender system. This article is notable for three reasons: First, it shows as effective a hybrid technique that can be generalized to other domains for which an ontology exists (or can be constructed) and where recommendations can appropriately be guided by a profile of user interest areas. Second, it shows how an existing external ontology can help address the cold-start problem in recommender systems (i.e., the problem that systems based purely on collaborative filtering cannot provide much value to their early users, and indeed cannot provide much value to new users until after they've populated their profiles). Third, this work includes significant field-study evaluation of the effectiveness of ontological profiles in the recommender system.

Hoffman's article on latent semantic models presents a model-based collaborative filtering algorithm that uses probabilistic latent semantic analysis and expectation-maximization algorithms to construct a compact and accurate reduced-dimensionality model of a community preference space. The intuition behind this model is that there is a set of independent underlying factors for which users have preferences, and that their preferences can be expressed as a vector of weights assigned to these factors. Similarly, items in the space can be expressed in terms of the same factors. This model can then be used for efficient prediction, and has the additional benefit of providing insight into clusters of related items. In addition to showing that the best of these algorithms is quite accurate, Hoffman shows that it scales well for prediction time (which does not increase as the number of items or users increases).

Huang et al.'s article on associate retrieval takes a different approach toward addressing the challenge of sparsity in a recommender system. Using data from a Chinese online bookstore, they explore how a spreading-activation algorithm (specifically a Hopfield net algorithm) can help improve recommendation quality for users by helping exploit transitive association. Intuitively, this addresses one of the challenges of pure collaborative filtering algorithms. If two users have both read and liked similar books, but not the exact same ones, the relationship between them is lost. This article shows that applying a spreading-activation algorithm can help in the recommendation process—for new users in particular, but also for users in general. It also shows that higher density can lead to an overactivation effect so that the benefits of spreading activation trail off as the ratings density increases.

Deshpande and Karypis' article on item-based recommenders addresses the specific challenge of recommending a list of candidates (a top-N list) rather than making predictions for most or all of an item set. This task is particularly common in electronic commerce applications where a system is used to select a small number of items to suggest on a personalized Web page, in a promotional message, or at transaction time. This article shows how algorithms based on co-purchase (or co-rating) similarities between items, or between item-sets and items, can produce efficient and high quality recommendations. In addition to evaluating two key techniques that provide this quality (similarity measures based on conditional probability and higher-order, item-based models), this article does an excellent job of using a diverse collection of data sets (eight collected from real systems and thirty-six synthetic ones) to validate the results.

ACKNOWLEDGMENTS

I would like to thank three groups of people for their work in making this issue happen. First, I would like to thank the many anonymous reviewers who not only volunteered to review these articles, but also provided valuable detailed feedback on a tight schedule. Second, I would like to thank the authors for their speed and responsiveness in addressing reviewer concerns; I was delighted to be able to go through two revisions on many articles and still keep the special issue on schedule. Finally, I would like to thank Gary Marchionini, Editor-in-Chief of *TOIS*, who agreed to take on this project and supported it through

realization, and Doug Oard, Associate Editor of *TOIS*, who worked with me throughout the process, managed reviews of articles where I had conflicts-of-interest, and served as a mentor and guide in creating this issue.

REFERENCE

SHARDANAND, U., MAES, P. 1995. Social information filtering: Algorithms for automating "Word of Mouth". In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'95).* ACM, New York, 210–217.

JOSEPH A. KONSTAN
University of Minnesota
Guest Editor