Accurate is not always good: How Accuracy Metrics have hurt Recommender Systems

Sean M. McNee
GroupLens Research
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455 USA
mcnee@cs.umn.edu

John Riedl
GroupLens Research
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455 USA
riedl@cs.umn.edu

Joseph A. Konstan
GroupLens Research
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455 USA
konstan@cs.umn.edu

CHI 2006, April 22–27, 2006, Montreal, Canada.
ACM 1-xxxxxxxxxxxxxxxxxx.

Abstract
Recommender systems have shown great potential to help users find interesting and relevant items from within a large information space. Most research up to this point has focused on improving the accuracy of recommender systems. We believe that not only has this narrow focus been misguided, but has even been detrimental to the field. The recommendations that are most accurate according to the standard metrics are sometimes not the recommendations that are most useful to users. In this paper, we propose informal arguments that the recommender community should move beyond the conventional accuracy metrics and their associated experimental methodologies. We propose new user-centric directions for evaluating recommender systems.

Keywords
Recommender systems, personalization, collaborative filtering, metrics

ACM Classification Keywords
H3.3. Information Search and Retrieval: Information Filtering, Search Process, Retrieval Models

Introduction

Imagine you are using a travel recommender system. Suppose all of the recommendations it gives to you are for places you have already traveled to? Even if the system was very good at ranking all of the places you have visited in order of preference, this still would be a poor recommender system. Would you use such a system?

Unfortunately, this is exactly how we currently test our recommender systems. In the standard methodology, the travel recommender would be penalized for recommending new locations instead of places the users have already visited! Current accuracy metrics, such as MAE [Herlocker 1999], measure recommender algorithm performance by comparing the algorithm's prediction against a user's rating of an item. The most commonly used methodology with these metrics is the leave-n-out approach [Breese 1998] where a percentage of the dataset is withheld from the recommender and used as test data. In essence, we reward a travel recommender for recommending places a user has already visited, instead of rewarding it for finding new places for the user to visit.

By focusing on this way of testing recommenders, are we really helping users find the items they are interested in? We claim there are many different aspects to the recommendation process which current accuracy metrics do not measure. In this paper, we will review three such aspects: the similarity of recommendation lists, recommendation serendipity, and the importance of user needs and expectations in a recommender. We will review how current methodologies fail for each aspect, and provide suggestions for improvement.

Similarity

More frequently than not, recommendation lists contain similar items on them. Going to Amazon.com for a book by Robert Heinlein, for example, will give you a recommendation list full of all of his other books. We have seen this behavior in algorithms as well. The Item-Item collaborative filtering algorithm can trap users in a 'similarity hole', only giving exceptionally similar recommendations (e.g. once a user rated one Star Trek movie she would only receive recommendations for more Star Trek movies) [Rashid 2001]. This problem is more noticeable when there is less data on which to base recommendations, such as for new users to a system. It is these times when a poor recommendation could convince a user to leave the recommender forever.

Accuracy metrics cannot see this problem because they are designed to judge the accuracy of individual item predictions; they do not judge the contents of entire recommendation lists. Unfortunately, it is these lists that the users interact with. All recommendations are made in the context of the current recommendation list and the previous lists the user has already seen. The recommendation list should be judged for its usefulness as a complete entity, not just as a collection of individual items.

One approach to solving this problem was proposed in [Ziegler 2005], with the introduction of the Intra-List Similarity Metric and the process of Topic Diversification for recommendation lists. Returned lists can be altered to either increase or decrease the diversity of items on that list. Results showed that these altered lists performed worse on accuracy measures than unchanged lists, but users preferred the altered lists.

Depending on the user's intentions, the makeup of items appearing on the list affected the user's satisfaction with the recommender more than the changes in the accuracy of the item on the list.

Serendipity

Serendipity in a recommender is the experience of receiving an unexpected and fortuitous item recommendation. There is a level of emotional response associated with serendipity that is difficult to capture in any metric. But even if we remove that component, the unexpectedness part of this concept—the novelty of the received recommendations—is still difficult to measure. The converse of this concept, the *ratability* of received recommendations, is quite easy to measure, however. And it is ratability that a leave-n-out approach measures.

When applied to a classification problem, a leave-n-out methodology works great: items belonging to a particular segment are withheld and the classifier is judged on how it classifies these items given what it knows of the dataset. Similarly, a leave-n-out methodology judges a recommender algorithm by how well it classifies, or recommends, withheld items to the users they were withheld from.

We define the ratability of an item in a recommender to be the probability that this item will be the next item that the user will consume (and then rate) given what the system knows of the user's profile. From a machine learning point-of-view, the item with the highest ratability would be the next item a classifier would place into the user's profile. Thus, recommender algorithms which score well on accuracy metrics using a leave-n-out methodology can generate recommendation for items with high ratability. These algorithms are good at predicting what else will appear in that user's profile. Indeed, many machine learning algorithms score very well as recommenders based on a leave-n-out methodology [Billsus 1998, Breese 1998].

The implicit assumption is that a user is always interested in the items with the highest ratability. While this assumption is true in classification problems, we claim it often isn't true in recommenders. Users often judge recommendations that are for items they would not have thought of themselves. For instance, we once were helping an online music store with recommendations using a User-User Collaborative Filtering algorithm. The most common recommendation was for the Beatle's "White Album". From an accuracy perspective these recommendations were dead-on: most users like that album very much. From a usefulness perspective, though, the recommendations were a complete failure: every user either already owned the "White Album", or had specifically chosen not to own it. Even though it was highly ratable, "White Album" recommendations were almost never acted on by users, because they added almost no value.

Our previous research [McNee 2002, Torres 2004] has given us one more piece to this puzzle: recommender algorithms generate qualitatively different recommendations lists from each other. When asked, users preferred lists from different recommender algorithms based on their current task. Users chose different words to describe the recommendations they received (e.g. User-based collaborative filtering was considered to generate "novel" recommendations).

This suggests that we need other ways to classify recommender algorithms. While a 'serendipity metric' may be difficult to create without feedback from users, other metrics to judge a variety of algorithm aspects would provide a more detailed pictures of the differences between recommender algorithms.

User Experiences and Expectations

As we have shown in previous work, user satisfaction does not always correlate with high recommender accuracy [McNee 2002, Ziegler 2005]. There are many other factors important to users that need to be considered.

New users have different needs from experienced users in a recommender. New users may benefit from an algorithm which generates highly ratable items, as they need to establish trust and rapport with a recommender before taking advantage of the recommendations it offers. Previous work shows that the choice of algorithm used for new users greatly affects the user's experience and the accuracy of the recommendations the system could generate for them [Rashid 2001].

Our previous work also suggested that differences in language and cultural background influenced user satisfaction [Torres 2004]. A recommender in a user's native language was greatly preferred to one in an alternate language, even if the items themselves recommended were in the alternate language (e.g. a Portuguese-based research paper recommender recommending papers written in English).

Moving Forward

Accuracy metrics have greatly helped the field of recommender systems; they have given us a way to compare algorithms and create robust experimental designs. We do not claim that we should stop using them. We just cannot use them alone to judge recommenders. Now, we need to think closely about the users of recommender systems. They don't care about using an algorithm that scored better on a metric, they want a meaningful recommendation. There a few ways we can do this.

First, we need to judge the quality of recommendations as users see them: as recommendation lists. To do this, we need to create a variety of metrics which act on recommendation lists, not on items appearing in a list. There are already a few, such as the Intra-List Similarity metric, but we need more in order to understand other aspects of these lists.

Second, we need to understand the differences between recommender algorithms and measure them in ways beyond their ratability. Users can tell the difference between recommender algorithms. For example, when we changed the algorithm running the MovieLens movie recommender, we received many emails from users wondering why MovieLens had become so "conservative" with its recommendations. Both algorithms scored well on MAE measures, but were clearly different from each other.

Finally, users return to recommenders over a period of time, growing from new users to experienced users. Each time they come to the system, they have some reason for coming: they have a purpose. We need to judge the recommendations we generate for each user based on whether or not we were able to meet their need. Until we acknowledge this relationship with

users, recommenders will continue to generate mismatched recommendations [Zaslow 2002].

In the end, recommenders exist to help users. We, as a community, have created many algorithms and approaches to studying recommender algorithms. It is now time to also study recommenders from a user-centric perspective to make them not only accurate and helpful, but also a pleasure to use.

Acknowledgements

We would like to thank our collaborators, Cai-Nicolas Ziegler and Roberto Torres, and all of GroupLens Research. This work was funded by the National Science Foundation, grants DGE 95-54517, IIS 96-13960, IIS 97-34442 IIS 99-78717, and IIS 01-02229.

References

[1] Billsus, D., and Pazzani, M.J. Learning Collaborative Information Filters., In Proc. of ICML 1998, Morgan Kaufmann (1998), 46-54.

[2] Breese, J.S., Heckerman, D., and Kadie, C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proc. of UAI 1998*, Morgan Kaufmann (1998), 43-52.

[3] Herlocker, J.L., Konstan, J.A., Terveen, L.G., and Riedl, J.T. Evaluating Collaborative Filtering Recommender Systems. *ACM TOIS* 22, 1 (2004), 5-53.

[4] Herlocker, J.L., Konstan, J.A., Borchers, A., and Riedl, J. An algorithmic framework for performing collaborative filtering. In *Proc. of ACM SIGIR 1999*, ACM Press (1999), 230-237.

[5] McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A. and Riedl, J. On the Recommending of Citations for Research Papers. In *Proc. of ACM CSCW 2002*, ACM Press (2002), 116-125.

[6] Rashid, A.M., Albert I., Cosley D., Lam, S.K., McNee, S.M., Konstan, J.A. and Riedl, J. Getting to know you: learning new user preferences in recommender systems. In *Proc. of ACM IUI 2002*, ACM Press (2002), 127-134.

[7] Torres, R., McNee, S.M., Abel, M., Konstan, J.A., and Riedl, J. Enhancing digital libraries with TechLens+. In *Proc. of ACM/IEEE JCDL 2004*, ACM Press (2004) 228-236.

[8] Zaslow, J. If TiVo Thinks You Are Gay, Here's How To Set It Straight --- Amazon.com Knows You, Too, Based on What You Buy; Why All the Cartoons? *The Wall Street Journal*, sect. A, p. 1, November 26, 2002.

[9] Ziegler, C.N., McNee, S.M., Konstan, J.A., and Lausen, G., Improving Recommendation Lists through Topic Diversification. In *Proc. of WWW 2005*, ACM Press (2005), 22-32.