Research Challenges in Recommender Systems

John Riedl

1 Modest Goals

This position paper lays out a set of challenges for the recommender research community. These challenges are specifically intended to be areas of research that are (a) not receiving much attention from active researchers at present; and (b) of importance to the practitioner community. The intent is to help researchers understand problems they might investigate that might lead to practically important results.

Of course, researchers should not solely – or even primarily – focus on problems of merely practical importance. After all, these problems will receive plenty of attention from those who stand to gain market share, increase customer loyalty, and make more money if the sought-after improvements are achieved. Part of the reason for having a professional research community is precisely to focus on problems that are not of short-term commercial interest. Nevertheless, it is valuable to have a flow of information back and forth between practitioners and researchers. At some points in a research project, understanding the needs of the eventual consumers of the research may be helpful, and in some cases even inspiring.

This paper arises from an invited "tutorial" that I was asked to lead at the Recommender Systems conference in New York in October 2009. ACM RecSys was first held in Minnesota in October 2007, and has now grown to 280 attendees. The attendees are very interested in the needs of the industrial practitioners, at companies like Google, Yahoo!, Netflix, and Amazon. The practitioners sometimes feel that the researchers are continuing to work on boring problems just because they are well-defined, and that more innovative and exploratory research would be more interesting and useful for them. Hence the tutorial, at which two leaders from industry (Jon Sanders of Netflix, and Todd Beaupre of Yahoo!) agreed to provide a critique of research directions in the field, and to suggest and motivate alternatives. My role was to help the presenters organize their thoughts in the months leading up to the conference, and to emcee the proceedings. I also chose, on my own initiative, to steal Jon and Todd's good ideas, and share them with you. Any flaws in this writing are likely in my recollection or recounting, not in the original ideas.

2 Key Challenges

In this section we discuss the key challenges presented in the tutorial. For each challenge we introduce the challenge in a paragraph, and then present a hint of some directions in another paragraph or two.

Transparency. The worst recommendation I ever made was in my senior year of college. I suggested to my friend Art that he would love the movie *A Clockwork Orange*. I still maintain that the recommendation was a good one – though we are both certain it was a poor choice for his first date with Angela. If I had taken the time to explain to him what he would like about the movie – the scary vision of a fascist futuristic society – the disaster would have been avoided. The goal of *transparency* in recommenders is to provide these sorts of insight.

Transparency can mean different things in different applications. Sometimes it means to help the user make a good decision, by providing him with selected subsets of information. Other times it means to help the user decide whether a particular recommendation fits a goal or mood. The challenges of developing effective transparency include (a) choosing whether to trace the workings of an algorithm, or to put together an independent argument that justifies the results of the algorithm; (b) what data to use, including other items, other users, or metadata; and (c) how to present the results to the user.

Exploration versus Exploitation. One of the longest-standing challenges in recommenders is how to deal with the *cold start* problem for new items and new users, for which the recommender does not yet have enough information to form reliable recommendations. If information is not collected to form recommendations, the items will never be shown to any users, so information will never be collected, ...

One approach is to take turns presenting the user with items he is likely to like (exploitation), and low information items to get his input so they can be effectively recommended to other users (exploration). The problem can be modelled by estimating the value provided to this user by a good recommendation, and comparing it to the value obtained for the community by getting this user's input on a low information item. (Generally the value of additional input on a high information item is low, because of diminishing returns.) Of course, a key component of the estimate must be the risk of driving away the user if he is shown too many items he is not interested in.

Guided Navigation. Rather than passively consuming a set of recommended items, users often want to be in control of their interaction with the items in a system. The recommendations are often best used as guides through a complex item space, letting the user choose in what directions to move at each step.

One set of key implementation challenges is providing the user with navigation options along dimensions that she cares about, such as: mood, tags, and familiarity versus freshness. Another challenge is to incorporate conversation in the interface, so the system helps the user by suggesting key questions she might answer.

Time Value. In some item spaces, such as books or movies, the relationships between users and items and users changes slowly. In other item spaces, such as daily news, items change in relevance rapidly, and different users have different taste in items. For instance, one user might value above all always knowing the most recent information, while another user might prefer the deeper insights of careful reporting that takes days or weeks to complete.

Key question to explore include:

- How should user input decay in value over time, as the user changes preferences in response to earlier experiences?
- How can short-term (e.g., planning a trip to Hong Kong) and long-term (e.g., new job) preference changes be distinguished?
- How can user models recognize different preferences for timeliness?
- What should be done about evolving user tastes in long-term items, such as books, movies, or wines?

User Action Interpretation. Explicit ratings are a valuable signal of user interest. However, in most systems at least an order of magnitude more information is available in the implicit signals of user interest, such as what items he clicks on, how long he reads them, which items are added to a wish list or shopping cart, etc.

One deep challenge in these data are interpreting negative choices in addition to positive choices. What information is contained in a user *not* choosing to click on a news item? In systems with effective recommenders, users are usually presented with items they will like, so most ratings are high. The Best Paper at RecSys 2009 analyzed the ways this distortion in the available preference data causes problems for the recommender algorithms, and suggests statistical techniques to correct for the distortion.

Evaluating Recommenders. How should recommender systems be evaluated? At the bottom line, companies that deploy recommenders care most about (a) purchase behavior; and (b) customer retention. Incorporating purchase behavior in recommender evaluation is relatively straight-forward (though the problems of the previous section must be dealt with).

Incorporating customer retention is more difficult, because it takes weeks or months to measure. For this reason, companies seek proxy measures they believe reflect eventual retention, such as click-through rates, or time on site. These measures often show very different results from offline algorithm tests. In practice, interface changes often lead to much more dramatic changes in customer behavior than algorithmic changes. The most effective way to compare alternatives is through A/B testing, in which customers are separated into groups, which are subject to different treatments. Carrying out such studies can be difficult for researchers who do not have access to recommender systems with a steady flow of visitors.

3 Missing Challenges

In preparing for the tutorial we also discussed a number of other relevant issues that we did not have time to present in detail. The most interesting are:

Information Diet: Users often want a mixture of items, not a set that includes only items of one type. For instance, a user might only want to see one story on Paris Hilton a day.

Serendipity: Traditional algorithmic measures often evaluate how well a recommender does at finding items a user already knows about. Often the most valuable recommendations in practice are for items a user will like, but has no idea they even exist.

Privacy: Can recommenders be developed that use privacy protected data? Can they prove to a user that the data is safe?

Balkanization: Do recommenders lead to increasingly focused sub-cultures on shared values and interests? Is this dangerous? If so, can algorithms be developed that moderate the balkanization?

Driving Traffic Strategically: Companies often have external goals they wish to meet with their recommenders. For instance, they might want to only recommend items that are in stock, or, given items of nearly equal value to the user, recommend the one with a higher profit margin.

Combining Content and Collaborative Recommenders: Recommenders work in a world with rich metadata and rich data about users. How can these diverse data be combined most effectively?

4 Closing Thoughts

Yes, this paper is parochial in its narrow focus on recommender systems, which are but a small part of the big picture. Nevertheless, I hope these ideas have helped inspire some relevant thoughts about the area of social computing you find most interesting. Though most of these problems are under-researched, I can suggest citations to work that will provide a starting point if you ask.