

Beyond Algorithms: An HCI Perspective on Recommender Systems

Kirsten Swearingen & Rashmi Sinha

SIMS, UC Berkeley, 94720

{kirstens, sinha}@sims.berkeley.edu

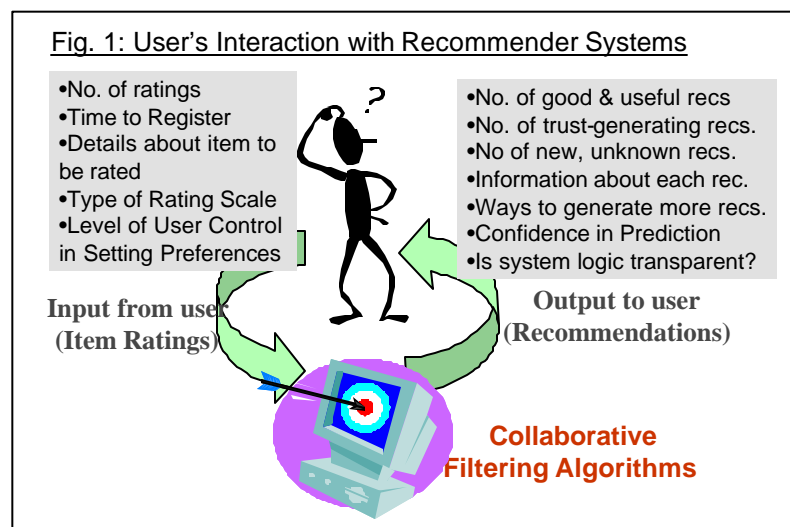
Abstract: The accuracy of recommendations made by an online Recommender System (RS) is mostly dependent on the underlying collaborative filtering algorithm. However, the ultimate effectiveness of an RS is dependent on factors that go beyond the quality of the algorithm. The goal of an RS is to introduce users to items that might interest them, and convince users to sample those items. What design elements of an RS enable the system to achieve this goal? To answer this question, we examined the quality of recommendations and usability of three book RS (Amazon.com, RatingZone & Sleeper) and three movie RS (Amazon.com, MovieCritic, Reel.com). Our findings indicate that from a user's perspective, an effective recommender system inspires trust in the system; has system logic that is at least somewhat transparent; points users towards new, not-yet-experienced items; provides details about recommended items, including pictures and community ratings; and finally, provides ways to refine recommendations by including or excluding particular genres. Users expressed willingness to provide more input to the system in return for more effective recommendations.

INTRODUCTION

A common way for people to decide what books to read or movies to watch is to ask their friends for recommendations. Online Recommender Systems (RS) attempt to create a technological proxy for this social filtering process. Previous studies of RS have mostly focused on the collaborative filtering algorithms that drive the recommendations (Delgado 2000, Herlocker 2000, Soboroff 1999). We conducted an empirical study to examine user's interactions with several online book and movie RS from an HCI perspective. We had two specific goals. Our first goal was to examine users' interaction with RS (i.e., input to the system, output from the system, and other

interface factors) in order to isolate design features that go into the making of an effective RS. Our second goal was to compare, from the user's perspective, two ways of receiving recommendations: (a) from online RS and (b) from friends (the social recommendation process).

The user's interaction with the RS can be divided into two stages: Input to the system and Output to the System (see Figure 1). Issues related to the Input stage comprise (a) number of ratings user had to provide, (b) if the initial rating items were user/system generated, (c) if the system provided information about the rated item, (d) the rating scale and (e) if the system allowed filtering by



metadata e.g., book author / genre. The output stage involves (a) the number of recommendations received, (b) information provided about each recommended item, (c) whether user had previously experienced the recommendation or not, (d) if system logic was transparent, (e) interface issues, and (f) ease of generating new sets of recommendation.

Our study involved an empirical analysis of users' interaction with three book RS (Amazon.com, RatingZone's QuickPicks, and Sleeper) and three movie RS (Amazon.com, Moviecritic, and Reel.com). We chose the RS based on differences in interfaces (layout, navigation, color, graphics, and user instructions), types of input required, and

information displayed with recommendations (see Appendix for the RS comparison chart). An RS may take input from users implicitly or explicitly, or a combination of the two (Schafer et. al 1999). Our study examined systems that relied upon explicit input.

We were also interested in comparing the two ways of receiving recommendations (friends and online RS) from the users' perspective. While researchers (Resnick & Varian, 1999) have compared RS with social recommendations, there is no reported research on how the two methods of receiving recommendations compare. Our hypothesis was that friends would make superior recommendations since they know the user well, and have intimate knowledge of his / her tastes in a number of domains. In contrast, RS only have domain-specific knowledge about the users. Also, information retrieval systems do not yet match the sophistication of human judgment processes.

METHODOLOGY

Participants: A total of 19 people participated in our experiment. Each participant tested either 3 book or 3 movie systems, and evaluated recommendations made by 3 friends. Study participants were mostly students at the University of California, Berkeley. *Age range:* 20 to 35 years. *Gender ratio:* 6 males and 13 females. *Technical background:* 9 worked in or were students in technology-related fields, the other 10 were studying or working in non-technical fields.

Procedure: This study was completed during November 2000 – January 2001. For each of the three book/movie recommendation systems (presented in a random order), users completed the following tasks: (a) Completed online registration process (if any) using a false e-mail address so that any existing buying/browsing history would not color the recommendations provided during the experiment. (b) Rated items on each RS in order to get recommendations. (Some systems required users to complete a second step, where they were asked for more ratings to refine recommendations.) (c) Reviewed list of recommendations. (d) If the initial set of recommendations did not provide anything that was both new and interesting, users were asked to look at additional items. They were to stop looking when they found at least one book/movie they were willing to try, or they grew tired of searching. (e) Completed satisfaction and usability questionnaire for each RS. After the user had tested and evaluated all three systems, we conducted a post-test interview.

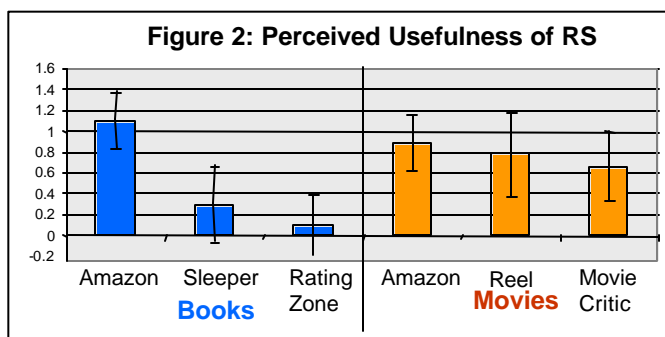
Independent Variables: (a) Item domain: books or movies (b) Source of recommendations: friend or online RS (c) Recommender System itself.

Dependent Measures:

- (a) Quality of recommendations was evaluated using 3 metrics.
 - *Good Recommendations:* Percentage of recommended items that the user liked. Good Recommendations were divided into the following two subcategories.
 - *Useful Recommendations* were “good” recommendations that the user had not experienced before. This is the sum total of useful information for the user—ideas for new books to read / movies to watch.
 - *Previously Liked Recommendations (Trust-Generating Recommendations)* were “good” recommendations that the user had already experienced and enjoyed. These are not “useful” in the traditional sense, but our study showed that such items indexed users' confidence in the RS.
- (b) Overall satisfaction with recommendations and with RS.
- (c) Time measures – time spent registering and receiving recommendations from the system

RESULTS & DISCUSSION

The goal of our analysis was to find out if users perceived RS as an effective method of finding about new books / movies. To answer these questions, we did a comprehensive analysis of



all the data we gathered in the study: time & behavioral logs, questionnaire about subjective satisfaction, rating of recommended items, self report during test & observations made by tester. Results pertaining to general satisfaction with RS are discussed first. Subsequently, we discuss specific aspects of user's interaction with the RS, focusing on the system input / output elements identified earlier. For each input / output element, we have identified a few design choices. If possible, we also offer design suggestions for RS. These design suggestions are based on our interpretation of the study results. For some system elements, we do not have any specific recommendations (since the results did not allow any strong inferences). In such cases, we have attempted to define a range of design options, and the factors to consider in choosing a particular option

I) Users' General Perception of Recommender Systems

Results showed that the users' friends consistently provided better recommendations, i.e., higher percentage of "good" and "useful" recommendations as compared to online RS (see Fig. 1). However, further analysis and post-test interviews revealed that users did find value in the online RS. (For a detailed discussion of the RS vs. friends' methodology and findings, see Sinha & Swearingen, 2001.)

a) **Users Perceived RS as being Useful:** Overall, users expressed a high level of overall satisfaction with online RS. Their qualitative responses in the post-test questionnaire indicated that they found the RS useful and intended to use the systems again.

b) **Users did not Like All RS Equally:** However, not all RS performed equally well. As Figure 2 shows, though most systems were judged at least somewhat useful, Amazon Books was judged the most useful, RatingZone was judged not useful, while Sleeper was judged only moderately useful. This corresponds to the results of the post-test interviews, in which, of the 11 users who said they preferred one of the online systems, 6 named Amazon as the best (3 for Amazon-books and 3 for Amazon-movies), 3 preferred Sleeper, and 3 liked MovieCritic.

c) **What Factors Predicted Perceived Usefulness of System:** What factors contributed to the perceived usefulness of a system? To examine this question, we computed correlations between Perceived Usefulness and other aspects of a Recommender System (see Table 1). We found that certain elements correlated strongly with perceived usefulness, while others showed a very low correlation.

As Table 1 shows, Perceived Usefulness correlated most highly with % Good and % Useful Recommendations. % Good Recommendations is indicative of the accuracy of the algorithm, and it is not surprising that it plays an important role in determining Perceived Usefulness of System. However, these two metrics (Good and Useful Recommendations) do not tell the whole story. For example, RatingZone's performance was comparable to

TABLE 1	
Factors that predict RS Usefulness	
No. of Good Recs.	0.53 **
No. of Useful Recs.	0.41 **
Detail in Item Description	0.35 **
Know reason for Recs?	0.31 *
Trust Generating Items	0.30 *
Factors that don't predict RS Usefulness	
Time to get Recs.	0.09
No. of recs.	-0.02
No of items to rate	-0.15

* significant at .05

** significant at .01



Amazon and Sleeper, (in terms of Good and Useful recommendations); but RatingZone was neither named as a favorite nor deemed "Very Useful" by subjects. On the other hand, MovieCritic's performance was poor relative to Amazon and Reel, but several users named it as a favorite. Clearly, other factors influenced the users' perception of RS usefulness. Our next task was to attempt to isolate those factors.

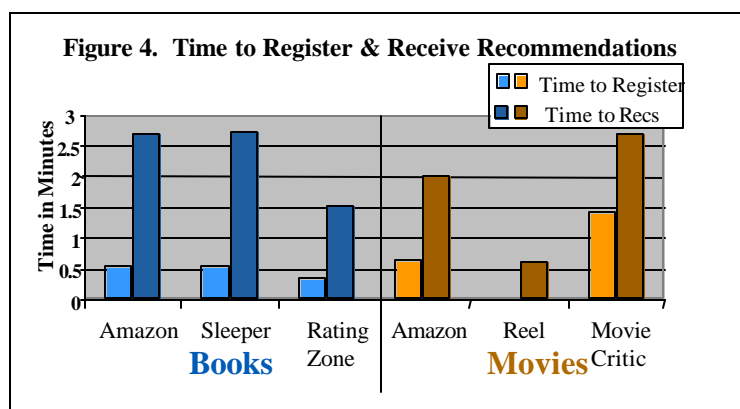
II) Design Suggestions: System Input Elements

II-a) Number of Ratings Required to Receive Recommendations / Time to Register

Our results indicate that an increase in the number of ratings required does not correlate with ease of use (see Table 1, above). Some of the systems that required the user to make many ratings (e.g. Amazon, Sleeper) were rated highly on satisfaction and perceived usefulness. Ultimately what mattered to users was whether they got what they came for: useful recommendations. Users appeared to be willing to invest a little more time and effort if that outcome seemed likely. They did express some impatience with systems that required a large number of ratings, e.g., with MovieCritic (required 12 ratings) and Rating Zone (required 50 ratings). However, the users' impatience seemed to have less to do with the absolute number of ratings and more to do with the way the information was displayed (e.g., only 10 movies on each screen, no detailed information or cover image with the title, necessitating numerous clicks in order to rate each item). For more details on presentation of rating information and interface issues, see sections I-b and II-e, below.

Also, time to register and receive recommendations did not correlated with the perceived usefulness of the system (see Table 1). As Figure 3 shows, systems that took less time to give recommendations were not the ones that provided the most useful suggestions.

We had also asked users if they thought any system asked for too much personal information during the registration process. Most systems required users to indicate information such as name, e-mail address, age, and gender. The users did not mind providing this information and it did not take them a long time to do so.



- "... there wasn't a lot of variation in the results... I'd be willing to do more rating for a wider selection of books." (Comment about Amazon)
- "There could be a few (2 or 3) more questions to gain a clearer idea of my interests...maybe if I like historical novels, etc.?"(Comment about RatingZone)

Design Suggestion: Designers of recommendation systems are often faced with a choice between enhancing ease of use (by asking users to rate fewer items) or enhancing the accuracy of the algorithms (by asking users to provide more ratings). Our suggestion is that it is fine to ask to the users for a few more ratings if that leads to substantial increases in accuracy.

II-b) Information about Item Being Rated

The systems differed in the amount of information they provided about the item to be rated. Some, such as RatingZone (version 1), provided only the title. If a user was not sure whether he/she had read the item, there was no way to get more information to jog his/her memory. Other systems, such as MovieCritic, Amazon and RatingZone (version 2), provided additional information but located it at least one click away from the list of items to be rated. Finally, systems such as Sleeper provided a full plot synopsis along with the cover image. Sleeper differed from the other RS in another important way. Rather than trying to develop a gauge set of popular items that people would be likely to have read or seen, Sleeper circumvented the problem by selecting a gauge set of obscure items, then asking "how interested are you in books like this one?" instead of "what did you think of this book?"

This meant that users were empowered to rate every item presented, instead of having to page through long lists, hoping to find rate-able items.

- *9 of the 15 she hadn't heard of—"I have to click through to find out more info." (Sighing.) "Lots of clicking!" (Comment about Amazon)*
- *Worried because she hadn't read many of the books [to be rated]. (Comments about RatingZone)*
- *"I don't read too many books--brief descriptions were helpful" (Comment about Sleeper)*

Design Suggestion: Satisfaction and ease-of-use ratings were higher for the systems that collocated some basic information about the item being rated on the same page. Cover image and plot synopses received the most positive comments, but future studies could identify other crucial elements for inclusion.

II-c) Rating Scales for Input Items

The RS used different kinds of rating scales for input ratings. MovieCritic used a 9-point Likert Scale, Amazon asked users for a favorite author / director, while Sleeper used a continuous rating bar. Some users commented favorably on the continuous rating bar used by Sleeper (See Figure 4), which allowed them to express gradations of interest level. Part of the reaction seemed to be to the novelty of the rating method. The only negative comments on rating methods were regarding Amazon's open text-box for "Favorite item." "Three of the users did not want to select a single item (artist, author, movie, hobby) as "favorite;" one user tried to enter more than one item in the "Favorite Movie" textbox, only to receive an error.

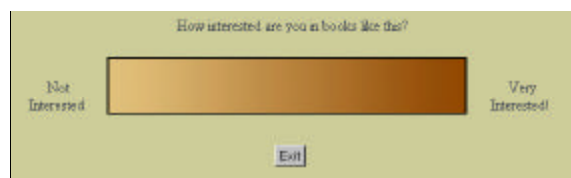


Figure 5. Sleeper Rating Scale

- *"I liked rating using the shading" (Comment about Sleeper's rating scale)*
- *"Interesting approach, [it was] easy to use." (Comment about Sleeper's rating scale).*

Design Suggestion: We do not have design suggestions in this area, but recommend pre-testing the rating scale with users; we also think that user's preference for continuous scale vs. discrete scales should be studied further.

II-d) Filtering by Genre

MovieCritic provided examples of both effective and ineffective ways to give users control over the items that are recommended to them. The system allowed users to set a variety of filters. Almost all of the users commented favorably on the genre filter—they liked being able to quickly set the "include" and "exclude" options on a list of about 20 genres. However, on the same screen, MovieCritic offered a number of advanced features, such as "rating method" and "sampling method" which were confusing to most users. Because no explanation of these terms was readily available, users left the features set to their default values. Although this did not directly interfere with the recommendation process, it may have negatively affected the sense of control which the genre filters had so nicely established.

- *"Good they show how to update—I like this." (Comment about MovieCritic)*
- *"Amazon should have include/exclude genre, like MovieCritic" (Comment about Amazon & MovieCritic)*
- *"No idea what a rating method or sampling method are [in Preferences]" (Comment about MovieCritic)*

Design Suggestion: Our design suggestion is to include filter-like controls over genres, but to make them as simple and self-explanatory as possible.

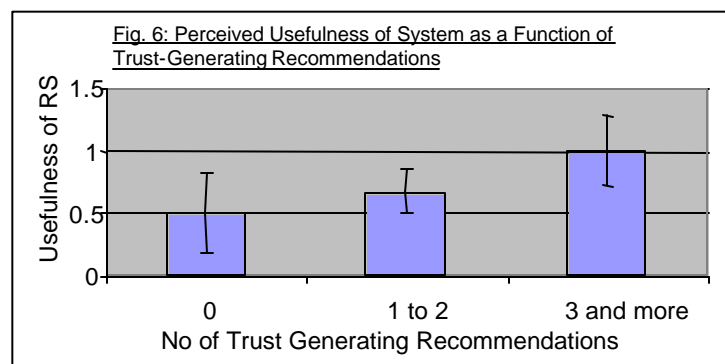
III) Design Suggestions: System Output Elements

III-a) Accuracy of Algorithm

As discussed earlier, Perceived Usefulness of systems correlated highly with % Good and % Useful recommendations. Both our qualitative and quantitative data give support for the fact that accurate recommendations are the backbone of an effective RS. The design suggestions that we are discussing are useful only if the system can provide accurate recommendations.

III-b) Good Recommendations that have been Previously Experienced (Trust-Generating Recommendations)

As Table 1 shows, Good Recommendations with which the user has previously had a positive experience correlate with Perceived Usability of systems. Such recommendations are not useful in the traditional sense (since they do not offer any new information to the user), but they index the degree of confidence a user can feel in the system. If a



system recommends a lot of "old" items that the user has liked previously, chances are, the user will also like "new" recommended items.

Figure 6 shows that the perceived usefulness of a recommender system went up with an increase in the number of trust-generating recommendations.

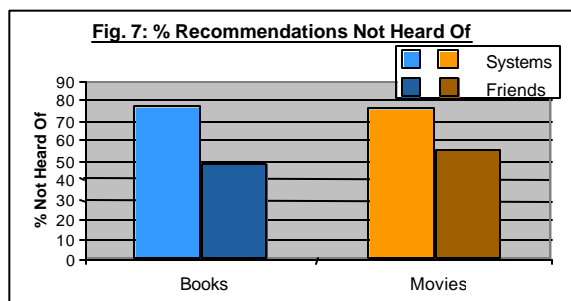
- *"I made my decision because I saw the movie listed in the context of other good movies" (Comment about Reel)*

Design Suggestion: Our design suggestion is that systems should take measures to enhance user's trust. However, it would be difficult for any system to insure that some percentage of recommendations was previously experienced. A possible way to facilitate this would be to generate some very popular recommendations, classics that the user is likely to have watched / read before. Such items might be flagged by a special label of some kind (e.g., "Best Bets").

III-c) Recommendations of New, Unexpected Items

Again, this concern has less to do with design and more to do with the algorithm driving the recommendations. It complements the previous point regarding trust-generating items. Five of our users stated that their favorite RS succeeded by expanding their horizons, suggesting items they would not have encountered otherwise.

- *"A number of things I hadn't heard of. Some guesses were more out there than friends, but it*

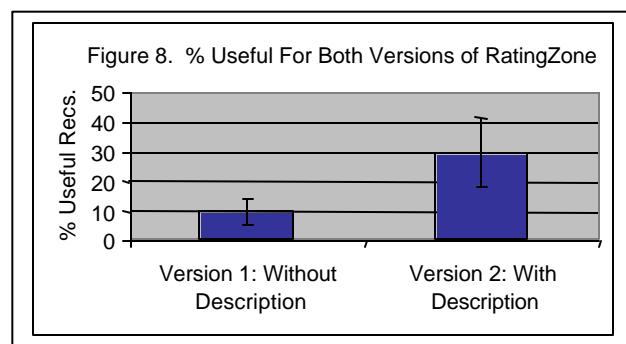


was] nice to be surprised....90% of friends' books I'll want to read, but I already knew I wanted to read these. I want to be stretched, stimulated with new ideas.”(Comment about Amazon)

- *“Sleeper suggested books I hadn’t heard of. It was like going to Cody’s [a local bookstore]—looking at that table up front for new and interesting books.” (Comment about Sleeper)*

Design Suggestion: To achieve this design goal, RS could include recommendations of new, just released items. Such recommendations could be a separate category of recommendations, leaving the choice of accessing them to the user.

III-d) Information about Recommended Items



The presence of longer descriptions of recommended items correlated positively with both the perceived usefulness and ease of use of RS. Users like to have more information about the recommended item (book / movie description, author / actor / director, plot summary, genre information, reviews by other users). Reviews and ratings by other users seemed to be especially important. Several users indicated that reviews by other users helped them in their decision-making. Similarly, people commented that pictures of the item recommended were very helpful in decision-making. Cover images often helped users recall

previous experiences with the item (e.g., they had seen that movie in the video store, read a review of the book etc.).

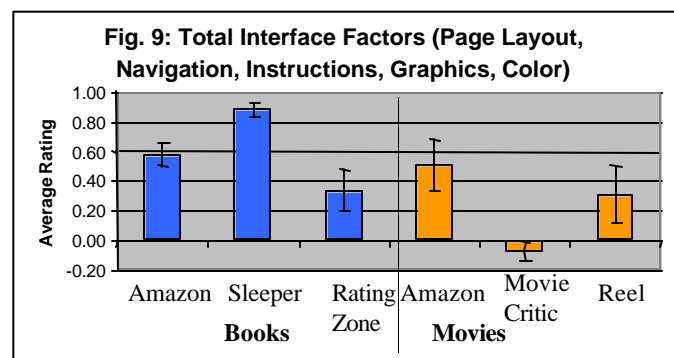
This finding was reinforced by the difference between the two versions of Rating Zone (see Figure 8). The first version of RatingZone's Quick Picks did not provide enough information and user evaluations were almost wholly negative as a result. The second version provided a link to the item description at Amazon. This small design change correlated with a dramatic increase in % useful recommendation. A different problem occurred at MovieCritic, where detailed information was offered but users had trouble finding it, due to poor navigation design.

- *“Of limited use, because no description of the books.”(Comment about RatingZone, Version 1)*
- *“Red dots [Predicted ratings] don't tell me anything. I want to know what the movie's about.”(Comment about MovieCritic)*
- *“I liked seeing cover of box in initial list of result... The image helps.”(Comment about Amazon)*

Design Suggestion: We recommend providing clear paths to detailed item information. This can be done by content maintained on the RS itself, or by linking to appropriate sources of information. We also recommend offering some kind of a community forum for users to post comments as an easy way to dramatically increase the efficacy of the system.

III-e) Interface Issues

From the user’s point of view, interface matters, mostly when it gets in the way. Navigation and layout seemed to be the most important factors—they correlated with ease of use and perceived usefulness of system, and generated the most comments, both favorable and unfavorable. For example, MovieCritic was rated negatively on layout and navigation. In



general MovieCritic performed well in terms of Good and Useful recommendations. Users' comments indicated that the navigation problems with MovieCritic might have lead to its low overall rating. Users did not have strong feelings about color or graphics and these items did not correlate strongly with perceived usefulness.

- *"Don't like how recommendations are presented. No information easily accessible. Not clear how to get info about the movie. Didn't like having to use the Back button [to get back from movie info]"*(Comment about MovieCritic)
- *"Didn't like MovieCritic--too hard to get to descriptions."*(Comment about MovieCritic)

Design Suggestion: Our design suggestion is to design the information architecture and navigation so that it is easy for users to access information about recommended item, and it is easy to generate new sets of recommendations.

III-f) Predicting the Degree of Liking for Recommended Items

Some RS also predict the degree of liking for the recommended item. Within our sample of systems, only Sleeper and MovieCritic provided such predictions (Amazon has recently added such a rating to its recommendation engine).

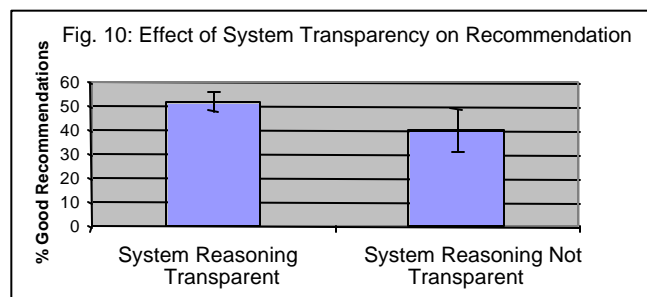
Users seemed to be mostly neutral about the "degree of liking" predictions; they did not help or hinder users' interactions with the system. However, such ratings can make users more critical of the recommendations. For example, a user might lose confidence in a system that predicted a high degree of liking for an item he/she hates. Another potential problem is if the system recommends items with low or medium "predicted liking" ratings. In such cases (as with Sleeper) users were confused about why the system recommended such items—the sparsity of items in the database was not visible, so users were left feeling like "hard to please" customers, and feeling unsure about whether to seek out the items given such tepid endorsements by the RS.

- *"All recommendations were in the middle of the Interested/Not Interested scale."*(Comment about Sleeper)
- *"So, so [in terms of usefulness]. Many books it recommended were ones I would be very interested in, yet they thought otherwise."*(Comment about Sleeper)

Design Suggestion: The predicted degree of liking is a high-risk feature. A system would need to have a very high degree of accuracy for users to benefit from this feature. Predicted liking could be used to sort the recommended items. Another possibility is to express the degree of liking categorically, (as with MovieCritic). MovieCritic divided items into "Best Bets" "Worst Bets" and some users liked this approach.

III-g) Effect of System Transparency

Users liked to understand what was driving a system's recommendations. Figure 10 shows that % Good Recommendations was positively related to Perceived System Transparency. This effect also surfaced in the comments made by users. On the other hand, some users, particularly those with a technical background, were irritated when a system's algorithm seemed too simplistic: "Oh, this is another Oprah book," or "These are all books by the author I put in as a Favorite."



- *"I really liked the system, but did not understand the recommendations."* (Comment about Sleeper)
- *"Don't know why computer books were included in refinement step. Didn't like any of them."* (Comment about Amazon)

- *“This movie was recommended because Billy Bob Thornton is in it. That’s not enough.” (Comment about MovieCritic)*
- *“They only recommended books by the author I picked. Lazy!” (Comment about Amazon)*

Design Suggestion: Users like the reasoning of RS to be at least somewhat transparent. They are confused if all recommendations are unrelated to the items they rated. RS should try to recommend at least some items that are clearly related to the items that the user had rated.

Recipe for an Effective Recommender System: Different Strokes for Different Folks

Our review above suggests that users want RS to satisfy a variety of needs. Some users want items that are very similar to ones they rated, while other users want items from other genres. We also noticed that some users are critical if the system logic seems too simplistic, other users like understanding system logic. Clearly, the same RS is satisfying very different needs. Below, we have tried to identify the primary kinds of recommendation needs that we observed.

- **Reminder recommendations**, mostly from within genre (“I was planning to read this anyway, it’s my typical kind of item”)
- **“More like this” recommendations**, from within genre, similar to a particular item (“I am in the mood for a movie similar to GoodFellas”)
- **New items**, within a particular genre, just released, that they / their friends do not know about
- **“Broaden my horizon” recommendations** (might be from other genres)

One way to accommodate these different needs is for an RS system to find a careful balance between the different kinds of items. However, we believe that a better design solution is for an RS to embrace these different needs and structure itself around them. There are two possible design options here. One solution is to divide recommended items into subsets so that the user can decide what kind of recommendations he/she would like to explore further. For example, recommended items could be divided into (a) new, just released items, (b) more by favorite author / director, (c) more from same genre, and (d) from different genres etc.

Another design solution is to explicitly ask users in the beginning of the session, the kind of recommendations they are looking for, and then recommend only those kinds of items. In either case, an RS needs to communicate clearly its purpose and usage, so as to manage the expectations of those who invest the time to use it. Communicating the reason a specific item is recommended also seems to be good practice. Amazon added this capacity after our study was completed so we were unable to gather feedback on its utility.

LIMITATIONS OF PRESENT STUDY

Conclusions drawn from this study are somewhat limited by several factors. (a) One limitation of our experiment design was that we handicapped the systems' collaborative filtering mechanisms by requiring users to simulate a first-time visit, without any browsing, clicking, or purchasing history. This deprived systems such as Amazon and MovieCritic of a major source of strength--the opportunity to learn user preferences by accumulating information from different sources over time. (b) A second limitation is that we did not study a random sample of online RS. As such, our results are limited to the systems we chose to study. (c) Finally, this study suffers from the same limitations as any other laboratory study: we do not know if users will behave in the same way in real life as in the lab.

ACKNOWLEDGEMENTS

This research was supported in part by NSF grant NSF9984741. We thank Marti Hearst and Hal Varian for their general support of the project and for the feedback they gave us at various points. We also thank Jennifer English, Ken Goldberg & Jonathan Boutelle for feedback about the paper, as well as this workshop's anonymous reviewers for helping to improve our presentation of this material.

REFERENCES

- Joaquin Delgado. “Agent-Based Information Filtering and Recommender Systems.” Ph.D thesis. March 2000.
- David Goldberg, Daniel Nichols, Brian M. Oki, and Douglas Terry. Using Collaborative Filtering to Weave an Information Tapestry.” *Communications of the ACM*, December 1992. 32 (12)
- Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins, “Eigentaste, A Constant-Time Collaborative Filtering Algorithm,” *Information Retrieval*, 4(2), July 2001.
- Jonathan L. Herlocker, Joseph A. Konstan, John Riedl. “Explaining collaborative filtering recommendations.” In *Proceeding on the ACM 2000 Conference on computer supported cooperative work*, 2000, Pages 241 – 250
- Don Peppers and Martha Rogers, Ph.D. “I Know What You Read Last Summer,” Inside 1to1. Oct. 21, 1999. <http://www.1to1.com/articles/il-102199/index.html>
- P. Resnick and H.R. Varian, “Recommender systems.” *Communications of the ACM*, 1997. 40(3) 56-58.
- Rashmi Sinha and Kirsten Swearingen. “Benchmarking Recommender Systems.” *Proceedings from DELOS workshop on personalization and recommender systems*, June 2001
- Ian M. Soboroff and Charles K. Nicholas “Combining Content and Collaboration in Text Filtering,” *Proceedings of the IJCAI 99 Workshop on Machine Learning and Information Filtering*, Stockholm, Sweden, August 1999.
- Shawn Tseng and B. J. Fogg, “Credibility and Computing Technology,” *Communications of the ACM*, special issue on Persuasive Technologies, 42 (5), May 1999.

APPENDIX: Description of Recommender Systems Examined in Study

Note: This study was completed during November 2000 – January 2001. Since then, 3 of the RS sites (Amazon, RatingZone, and MovieCritic) have altered their interfaces to various degrees.

<i>Description of Recommendation System</i>					
<i>User Input Aspect</i>	Amazon (both books and movies)	Sleeper	RatingZone	Reel	MovieCritic
How many items must a user rate to receive recommendations?	1 favorite item in each of 4 different categories, 16 more items in refinement step	15 items to rate (mandatory)	50 items to review, all optional to rate	1 item at a time	12 items to rate (mandatory)
Who generates items to rate?	User, initially.	System	System	User	System or user
Demographic information required	Name, e-mail address, age	Name, e-mail address	Name, e-mail address, age, gender, and zip	Nothing	Name, e-mail address, gender, age
Item rating scale	Favorite, then checkbox for “recommend items like this”	Shaded bar (range from “interested” to “not interested”)	Checkbox for “I liked it”	No rating, just enter the movie you want matched	11 point scale (“Loved it” to “Hated it” to “Won’t see it”)
Users could specify interest in particular item type or genre	No	No	Yes	No	Yes
<i>System Rec. Aspects</i>	Amazon	Sleeper	RatingZone	Reel	MovieCritic
Item information (titles only, cover images, synopsis etc.)	Title, cover image, synopsis	Title, cover image, synopsis	RZ Version 1: Title, # of pages, year of pub. RZ Version 2: added link to Amazon.	Title, cover image, brief description,	Screen 1: title. Screen 2: predicted ratings and other ratings Screen 3: IMDB

Information about system's confidence in recommendation	No	Yes	No	No	Yes
Information on other users' ratings	Yes	No	No	No	Yes