Explaining Recommendations: Satisfaction vs. Promotion

Mustafa Bilgic
Computer Science Dept.
University of Maryland at College Park
College Park, MD 20742
mbilgic@cs.umd.edu

Raymond J. Mooney
Computer Sciences Dept.
University of Texas at Austin
Austin, TX 78712-0023
mooney@cs.utexas.edu

ABSTRACT

Recommender systems have become a popular technique for helping users select desirable books, movies, music and other items. Most research in the area has focused on developing and evaluating algorithms for efficiently producing accurate recommendations. However, the ability to effectively explain its recommendations to users is another important aspect of a recommender system. The only previous investigation of methods for explaining recommendations showed that certain styles of explanations were effective at convincing users to adopt recommendations (i.e. promotion) but failed to show that explanations actually helped users make more accurate decisions (i.e. satisfaction). We present two new methods for explaining recommendations of content-based and/or collaborative systems and experimentally show that they actually improve user's estimation of item quality.

Introduction

The use of personalized recommender systems to aid users' selection of reading material, music, and movies is becoming increasingly popular and wide-spread. Most of the research in recommender systems has focused on efficient and accurate algorithms for computing recommendations using methods such as collaborative filtering [4, 5], content-based classifier induction [9, 8], and hybrids of these two techniques [1, 7]. However, in order for users to benefit, they must trust the system's recommendations and accept them. A system's ability to explain its recommendations in a way that makes its reasoning more transparent can contribute significantly to users' acceptance of its suggestions. In the development of expert systems for medicine and other tasks, systems' ability to explain their reasoning has been found to be critical to users' acceptance of their decisions [12].

Several recommender systems provide explanations for their suggestions in the form of similar items the user has rated highly, like Amazon, or keywords describing the item that caused it to be recommended [8, 2]. However, Herlocker et al. [6] provide the only systematic study of explanation methods for recommenders. Their experimental results showed that certain styles of explanation for collaborative filtering increased the likelihood that the user would adopt the system's recommendations. However, they were unable to demonstrate that any style of explanation actually increased users' satisfaction with items that they eventually chose.

Arguably, the most important contribution of explanations is not to convince users to adopt recommendations (promotion), but to allow them to make more informed and accurate decisions about which recommendations to utilize (satisfaction). If users are convinced to accept recommendations that are subsequently found to be lacking, their confidence in the system will rapidly deteriorate. A good explanation is one which accurately illuminates the reasons behind a recommendation and allows users to correctly differentiate between sound proposals and inadequately justified selections.

This paper evaluates three different approaches to explaining recommendations according to how well they allow users to accurately predict their true opinion of an item. The results indicate that the *neighbor style* explanations recommended by [6] based on their promotion ability perform poorly, while the *keyword style* and *influence style* explanations that we introduce perform much better.

Methods for Recommender Systems

Recommender systems suggest information sources and products to users based on learning from examples of their likes and dislikes. A typical recommender system has three steps: *i.*) Users provide examples of their tastes. These can be explicit, like ratings of specific items, or implicit, like URLs simply visited by the user [10]; *ii.*) These examples are used to compute a *user profile*, a representation of the user's likes and dislikes; *iii.*) The system computes recommendations using these *user profiles*.

Two of the traditional approaches to building a user profile and computing recommendations are collaborative filtering (CF) and content-based (CB) recommendation. Hybrid systems that integrate these two different approaches have also been developed.

Workshop: Beyond Personalization 2005
IUI'05, January 9, 2005, San Diego, California, USA
http://www.cs.umn.edu/Research/GroupLens/beyond2005

CF systems recommend items by matching a user's tastes to those of other users of the system. In the nearest-neighbor model [5], the *user profiles* are user-item ratings matrices. Recommendations are computed by first finding *neighbors*, similar users whose ratings correlate highly with those of the active user, and then predicting ratings for the items that the active user has not rated but the *neighbors* have rated using the *user profiles* and the correlation coefficients.

CB systems recommend items based on items' content rather than other users' ratings. The *user profiles* consist of concept descriptions produced by a machine-learning algorithm such as naive Bayes using a "bag of words" description of the items [8, 9]. Recommendations are computed based on predictions of these models which classify items as "good" or "bad" based on a feature-based description of their content.

Both CF and CB systems have strengths and weaknesses that come from exploiting very different sources of information. Consequently, a variety of different methods for integrating these two different approaches have recently been developed. Some of these hybrid methods use other users' ratings as additional features in a fundamentally content-based approach [1]. Others use content-based methods to create *filterbots* that produce additional data for "pseudo-users" that are combined with real users' data using CF methods [11]. Still others use content-based predictions to "fill out" the sparse user-item ratings matrix in order to allow CF techniques to produce more accurate recommendations [7]. A survey of hybrid recommenders can be found at [3].

Our Recommender System

We have previously developed a recommender system called LIBRA (Learning Intelligent Book Recommending Agent) [8]. The current version employs a hybrid approach we developed called *Content Boosted Collaborative Filtering* (CBCF) [7]. The complete system consists of three components. The first component is the Content Based Ranker that ranks books according to the degree of the match between their content and the *active user's* content-based profile. The second component is the Rating Translator that assigns ratings to the books based on their rankings. The third component is the Collaborative Filterer, which constructs final recommendations using an enhanced user-item ratings matrix.

LIBRA was originally developed as a purely content-based system [8] and has a database of approximately 40,000 books. Content descriptions are stored in a semi-structured representation with Author, Title, Description, Subject, Related Authors, and Related Titles. Each slot contains a bag of words, i.e. an unordered set of words and their frequencies. These data were collected in 1999 by crawling Amazon. Once the user rates a set of training books, the Content Based Ranker composes a *user profile* using a bag-of-words Naive Bayesian text classifier. The user profile consists of a table that has three columns: a slot column, a column for the token in that slot, and the strength column. The strength for a token t in a slot s is: $\frac{P(t|c_l,s)}{P(t|c_d,s)}$ where c_l is the category of *likes*, and c_d is the category of *dislikes*. A score for a test item is then computed by multiplying the strengths of each token t in slot s of the book. Lastly, the books are ranked based on their scores. This gives us the "Ranked Items" vector.

One of the main problems with CF methods is that the user-item ratings matrix upon which predictions are based is very sparse since any individual user rates only a small fraction of the available items. The basic idea of CBCF is to use content-based predictions to "fill out" the user-item ratings matrix. In [7], a 6-way CB classifier was used to predict integer ratings in the range 0–5. However, a 6-way classifier is less accurate than the 2-way (like vs. dislike) classifier originally used in LIBRA. Here, we use a Rating Translator as a bridge between the Content Based Ranker and the Collaborative Filterer.

The Rating Translator converts rankings into ratings by looking at the rating pattern of the user. However, since the rating pattern of a user usually tends to be skewed towards positive ratings, these data are first smoothed using a source of unskewed data: the rating pattern of several users who rated randomly selected items (Table 5 in [8]).

Once the user-item ratings matrix is filled-out using content-based predictions, we use a version of the CF method recommended in [5]. The system first computes correlations between the *active user* and other users of the system. The n users with the highest correlations are chosen as the *neighbors*. Predictions are computed using the *neighbors'* ratings for the test examples. Finally, the test items are sorted based on their predicted ratings and the top items are presented to the user as recommendations.

The Explanation Systems

A variety of recommender systems are now available. Some are developed for research purposes such as GroupLens [10], and some are in commercial use such as Amazon and NetFlix. Although a few of these provide some form of explanation for their recommendations, most are black boxes with respect to why they recommend a specific item [6]. Thus, the users' only way to assess the quality of a recommendation is to try the item, e.g. read the book or watch the movie. However, since users use recommender systems to reduce the time they spend exploring items, it is unlikely they will try an item without trusting that it is worth the effort. Herlocker et al. have shown that explanation systems increase the acceptance of collaborative filtering systems [6].

The effectiveness of an explanation system can be measured using two fundamentally different approaches: the *promotion* approach and the *satisfaction* approach. For the *promotion* approach, the best explanation is the one that is most successful at convincing the user to adopt an item. For the *satisfaction* approach, the best explanation is the one that lets the users assess the quality of the item the best.

Unfortunately, there is little existing research on explaining recommender systems. The only detailed study is that of Herlocker et al. [6] in which twenty-one different styles of explanations where compared. The title of a recommended item was removed in order to prevent any bias it might cause, and the user was asked to rate a recommended item by just looking at its explanation. Herlocker et al. generally present explanation systems that produce the highest mean rating as the best. We believe that *satisfaction* is more important than *promotion*. If the users are satisfied with their selections in the end, they will develop trust in the system and continue to use it. Although in a second study in the same paper, Herlocker et al. did examine the effect of explanation on "filtering performance," they failed to find any consistent effect. Consequently, we explore how well an explanation system helps the user accurately estimate their actual opinion of an item.

We have used three explanation systems in our study: *keyword style explanation* (KSE), *neighbor style explanation* (NSE), and *influence style explanation* (ISE). Two factors played a role in choosing these three explanation styles. One factor is the type of information required, i.e. content and/or collaborative. We included KSE for systems that are partly or purely content-based, and NSE for systems that are partly or purely collaborative. ISE is not dependent on the recommendation method as described below. The second factor that affected our selection of these styles is that we wanted to test how KSE and ISE perform compared to NSE, which was the best performing explanation method (from the standpoint of promotion) in Herlocker et al. 's study.

Keyword Style Explanation (KSE)

Once a user is provided a recommendation, he is usually eager to learn "What is it about the item that speaks to my interests?" KSE is an approach to explaining content-based recommendations that was included in the original version of LIBRA. KSE analyzes the content of a recommended item and finds the strongest matches with the content in the user's profile. In LIBRA, the words are matched against the table of feature strengths in the user profile described above. For each token t occurring c times in slot s of the item's description, a strength of $c * strength(t)$ is assigned, where $strength(t)$ is retrieved from the user-profile table. Then, the tokens are sorted by strength and the first twenty entries are displayed to the user. An example is presented in Figure 1. This approach effectively presents the aspects of the item's content that were most responsible for the item being highly ranked by the system's underlying naive-Bayesian classifier.

Slot	Word	Count	Strength	Explain
DESCRIPTION	HEART	2	94.14	Explain
DESCRIPTION	BEAUTIFUL	1	17.07	Explain
DESCRIPTION	MOTHER	3	11.55	Explain
DESCRIPTION	READ	14	10.63	Explain
DESCRIPTION	STORY	16	9.12	Explain

Figure 1: The Keyword Style Explanation

If the user wonders where a particular keyword came from, he can click on the *explain* column, which will take him to yet another table that shows in which training examples that word occurred and how many times. Only positively rated training examples are included in the table. An example of such a table is presented in Figure 2. This approach effectively presents which user-rated training examples where responsible for this keyword having its high strength.

Title	Author	Rating	Count
Hunchback of Notre Dame	Victor Hugo, Walter J. Cobb,	10	11
Till We Have Faces : A Myth Retold	C. S. Lewis, Fritz Eichenberg,	10	10
The Picture of Dorian Gray	Oscar Wilde, Isobel Murray,	8	5

Figure 2: Explanation of Which Positively-Rated Books Have a Keyword

For more information on LIBRA's KSE method, see [8]. Billsus and Pazzani's news recommender provides similar explanations [2].

Neighbor Style Explanation (NSE)

If the recommender system has a collaborative component, then a user may wonder how other similar users rated a recommended item. NSE is designed to answer this question by compiling a chart that shows how the *active user's* CF *neighbors* rated the recommended item. To compute the chart, the *neighbors'* ratings for the recommended item are grouped into three broad categories: Bad (ratings 1 and 2), Neutral (rating 3), and Good (ratings 4 and 5). A bar chart is plotted and presented, as shown in Figure 3. NSE was tested

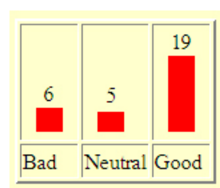

Figure 3: Explanation Showing Ratings of a User's Neighbors

along with twenty other explanation systems by Herlocker et al. [6] and performed the best from a *promotion* perspective. Grouping the rating into 3 coarse categories was found to be more effective than using a histogram with all 5 original ratings levels.

Influence Style Explanation (ISE)

ISE presents to the user a table of those training examples (which the user has already explicitly or implicitly rated) that had the most impact on the system's decision to recommend a given item. Amazon and NetFlix have a similar style of explanation, however it is unclear how they actually select the explanatory training items. LIBRA presents a table of training books that had the most impact on its recommendation. Each row in the table has three entries: the book that the *active user* rated, the rating they gave the book, and the *influence* of that book on this recommendation. An example of such an explanation is shown in Figure 4.

BOOK	YOUR RATING Out of 5	INFLUENCE Out of 100
Of Mice and Men	4	54
1984	4	50
Till We Have Faces : A Myth Retold	5	50
Crime and Punishment	4	46
The Gambler	5	11

Figure 4: Influence Style Explanation

The ideal way to compute influences is to remove the book whose influence is being computed from the training set, recompute the recommendation score for each of the test items, and measure the resulting difference in the score of the recommended book. Therefore, unlike KSE or NSE, ISE is completely independent of the underlying recommendation algorithm. For purely collaborative or purely content based approaches, removing a training example and re-scoring the test examples can be done fairly efficiently. However, for the full CBCF algorithm currently used by LIBRA, this would require recomputing every single user's content-based user-profile and re-scoring every item for every user to update the "filled in" user-item matrix. Doing this to compute the influence of every training example is infeasible for a real-time explanation system.

To compute the influences efficiently, we compute two influences, the content influence and the collaborative influence, separately, rescale both and then average the two. The content influence of an item on the recommendation is computed by looking at the difference in the score of the recommendation computed by training the Bayesian classifier with and without the item. The collaborative influence is computed similarly: the correlation constants and predictions are computed with and without the item; the difference in the prediction for the recommended item is the collaborative influence. So that the users can easily interpret the results, we wanted the final influence to be in a fixed range [-100, 100]. Since the ranges for content influences and collaborative influences were different (content influence is a difference of log probability ratios and collaborative influence is a difference in predicted ratings), we re-scale them separately to a common range, [-100, 100], and then we compute the final influence by averaging the two. We sort the table using this final influence and present all positive influences to the user.

Experimental Methodology and Results

Methodology

To evaluate these three forms of explanation, we designed a user study in which people filled out an online survey. The ideal way to implement a survey to measure satisfaction is:

1. Get sample ratings from the user.
2. Compute a recommendation r.
3. For each explanation system e

3.1 Present r to the user with e's explanation.

3.2 Ask the user to rate r

4. Ask the user to try r and then rate it again.

If we accept that a good explanation lets the user accurately assess the quality of the item, the explanation system that minimizes the difference between the ratings provided in steps 3.2 and 4 is best. In step 1, we ask the *active user* to provide LIBRA with ratings for at least three items, ranging from 1 (dislikes) to 5 (likes), so that LIBRA can provide him a decent recommendation along with some meaningful explanations. We remove the title and author of the book in the step 3 because we do not want the user to be influenced by it. The ratings in step 3.2 are based solely on the information provided in the current explanation. To avoid biasing the user, we tell him that each explanation is for a different book (since the explanations present very different information, the user has no way of knowing they are actually for the same item.) Moreover, we randomize the order of the explanation systems used in each run to minimize the effect of seeing one explanation before another. Since running this experiment would be very time consuming should we asked the users to read the books recommended to them, we slightly modified step 4. Instead of reading the book, the *active user* is asked to read the Amazon pages describing the book and make a more informed rating based on all of this information.

We hypothesized that: 1. NSE will cause the users to overestimate the rating of an item. 2. KSE and ISE will allow users to accurately estimate ratings. 3. Ratings provided at step 3.2 and 4 should be positively correlated, with ISE and KSE correlating with the final rating better than NSE.

We believed that NSE would cause overestimation since the presented histograms are always highly skewed towards the top ratings since otherwise the book would not have been recommended. We believed that ISE and KSE would give better correlations since they do not suffer from this problem and they present additional information about this or similar books that we believed was more useful.

Results

Thirty-four subjects were recruited to fill out the online survey, most were students in various departments at the University of Texas at Austin. Since the system allowed the users to repeat the process with more than one recommendation, we were able to collect data on 53 recommendations. We use the following definitions in the rest of the paper. *Explanation-ratings* are the ratings given to an item by the users in step 3.2 by just looking at the explanation of the recommendation. *Actual-ratings* are the ratings that users give to an item in step 4 after reading detailed information about the book.

Since LIBRA tries to compute good recommendations, we expect both *explanation-ratings* and *actual-ratings* to be high. As can be seen from the Table 1, the mean ratings are pretty high, at least 3.75.

Table 1: Means and Std Deviations of Ratings

Type	μ	σ
Actual	3.75	1.02
ISE	3.75	1.07
KSE	3.75	0.98
NSE	4.49	0.64

We expect to have approximately normal distributions for the differences between the *explanation-ratings* and the *actual-ratings*. The histograms of the differences are displayed in Figure 5. The means of the differences can be seen in Table 2.

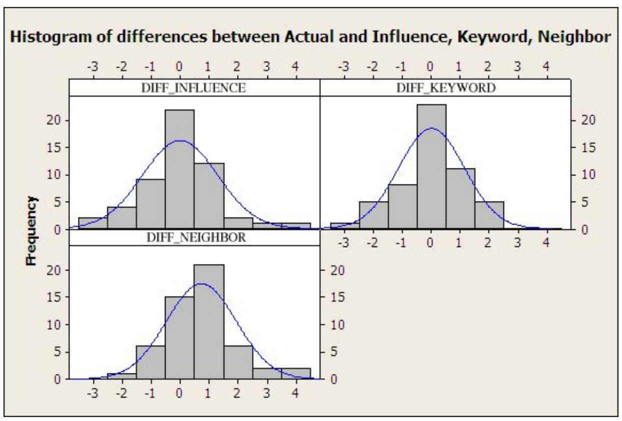

Figure 5: Histograms of Differences Between Explanation and Actual Ratings

Table 2: Means, Std Deviations, and Confidence Intervals of Differences

Type	μ	σ	**95% Conf. Int.**
ISE	0.00	1.30	(-0.36, 0.36)
KSE	0.00	1.14	(-0.32, 0.32)
NSE	0.74	1.21	(0.40, 1.07)

According to the *satisfaction* approach, the best explanation is the one that allows users to best approximate the *actual-rating*. That is, the distribution of *(explanation-ratings* $-$ *actual-ratings)* for a good explanation should be centered around 0. Thus, the explanation whose μ_d (the mean of the difference between *explanation-rating* and *actual-rating*) is closest to 0 and that has the smallest standard deviation σ_d in Table 2 is a candidate for being the best explanation. KSE wins with $\mu_d = 0.00$ and $\sigma_d = 1.14$. When we look at the confidence intervals, we see that both KSE and ISE are very close. This table also shows that, with high probability, NSE causes the user to overestimate the *actual-rating* by 0.74 on average. Considering that the mean for *actual-ratings* is 3.75, and that the highest rating is 5.00, a 0.74 overestimate is a significant overestimation. This table supports both Hypotheses 1 and 2.

We have also run paired t-tests to find out whether these differences were likely to be due to chance only. The null hypothesis we used for all three types of explanations is $H_0(\mu_d = 0)$. Since we did not have prior estimates on whether KSE and ISE would cause the user to overestimate or underestimate should they estimate wrong, the alternative hypothesis for these explanation systems is $H_a(\mu_d \neq 0)$. However, since we postulated that the NSE would cause the user to overestimate the *actual-ratings*, the alternative hypothesis for NSE is $H_a(\mu_d > 0)$. The results in Table 3 clearly show that we can reject the null hypothesis for NSE, because the probability of having $\mu_d = 0$ is 0.00. (i.e. $P = 0.00$). So, we accept the alternative hypothesis for NSE. For ISE and KSE on the other hand, we cannot reject the null hypothesis, because $P = 1.00$. Thereby, the t-tests justify Hypothesis 1.

Table 3: t-tests

	Hypotheses	**P**
ISE	$H_0(\mu_d = 0), H_a(\mu_d \neq 0)$	1.00
KSE	$H_0(\mu_d = 0), H_a(\mu_d \neq 0)$	1.00
NSE	$H_0(\mu_d = 0), H_a(\mu_d > 0)$	0.00

One other thing that needs to be noted is that the means themselves might be misleading. Consider the following scenario. Assume that we have a new style of explanation called, the *fixed style explanation* (FSE), such that no matter what type of recommendation the user is given, FSE presents such an explanation that it makes the user think that the quality of the item is 3 out of 5. If the *actual-ratings* are equally distributed in the interval $[1, 5]$, then the mean difference between the *explanation-ratings* and the *actual-ratings* for FSE will be 0. However, this does not necessarily mean that FSE is a good explanation. *Explanation-ratings* for a good explanation style should have $\mu_d = 0$, a low σ_d, plus they should strongly correlate with the *actual-ratings*.

We have calculated the Pearson correlation between *actual-ratings* and *explanation-ratings* along with their respective probabilities of being non-zero due to chance for all explanation styles. Results are presented in Table 4. The most strongly correlating explanation is KSE at 0.34. The prob-

Table 4: Correlations and P-Values

	Actual	
	r	**P**
ISE	0.23	0.10
KSE	0.34	0.01
NSE	-0.02	0.90

ability of getting this high of a correlation due to chance is only 0.01. ISE has a correlation of 0.23 and the probability of having this high of a correlation by chance of 0.1. Even though it does not meet the standard value of 0.05, it is close. The correlation constant for NSE is negative, however, the chance of having this small of a negative correlation is 90%. The correlation table supports our Hypothesis 3 fully for KSE and partially for ISE. NSE does not result in any correlation, indicating that it is ineffective at helping users evaluate the quality of a recommendation.

Future Work

Because of time issues, we had to ask the users to read the Amazon's pages instead of the books themselves. The experiment can be repeated in a domain where trying out the recommended item does not take much time, like a movie or music domain. Moreover, there are twenty other explanation styles described in Herlocker et al.'s paper [6]. The experiment could be repeated with these other explanation styles as well. Note that they found that NSE was the best explanation from a *promotion* perspective. Another style in that study could perform better from a *satisfaction* viewpoint.

Conclusions

The ability of recommender systems to effectively explain their recommendations is a potentially crucial aspect of their utility and usability. The goal of a good explanation should not be to "sell" the user on a recommendation, but rather, to enable the user to make a more accurate judgment of the true quality of an item. We have presented a user-study that evaluated three different approaches to explanation in terms of how accurately they allow users to predict a more in-depth evaluation of a recommendation. Our results demonstrate that the "neighborhood style" explanation for collaborative filtering systems previously found to be effective at promoting recommendations [6], actually causes users to overestimate the quality of an item. Such overestimation would lead to mistrust and could eventually cause users to stop using the system. Keyword-style explanations, which present content information about an item that caused it to be recommended, or influence-style explanations, which present ratings previously provided by the user that caused an item to be recommended, were found to be significantly more effective at enabling accurate assessments.

Acknowledgments

This research was partially supported by an undergraduate research assistant fellowship to the first author from the Dept. of Computer Sciences at the Univ. of Texas at Austin and by the National Science Foundation through grant IIS-0117308.

REFERENCES

1. C. Basu, H. Hirsh, and W. W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 714–720, Madison, WI, July 1998.
2. D. Billsus and M. J. Pazzani. A personal news agent that talks, learns and explains. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pages 268–275. ACM Press, 1999.
3. R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
4. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the Association for Computing Machinery*, 35(12):61–70, 1992.
5. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, Berkeley, CA, 2000. ACM Press.
6. J. Herlocker, J. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, pages 241–250, Philadelphia, PA, 2000. ACM Press.
7. P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, pages 187–192, Edmonton, Alberta, 2002.
8. R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 195–204, San Antonio, TX, June 2000.
9. M. J. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 54–61, Portland, OR, Aug. 1996.
10. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 175–186. ACM Press, 1994.
11. B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, pages 345–354. ACM Press, 1998.
12. W. Swartout, C. Paris, and J. Moore. Explanations in knowledge systems: Design for explainable expert systems. *IEEE Expert: Intelligent Systems and Their Applications*, 6(3):58–64, 1991.