From: AAAI Technical Report WS-00-04. Compilation copyright © 2000, AAAI (www.aaai.org). All rights reserved.

Semantic ratings and heuristic similarity for collaborative filtering

Robin Burke

Department of Information and Computer Science
University of California, Irvine
burke@ics.uci.edu

Abstract

Collaborative filtering systems make recommendations based on ratings of user preference. Usually, the ratings are uni-dimensional (e.g. like vs. dislike), and can be either explicitly elicited from users or, more typically, are implicitly generated from observations of user behavior. This research examines *multi-dimensional* or *semantic* ratings in which a system gets information about the reason behind a preference. Such multi-dimensional ratings can be projected onto a single dimension, but experiments show that metrics in which the semantic meaning of each rating is taken into account have markedly superior performance.

Introduction

Collaborative filtering (CF) is a technique for recommending items to a user's attention based on similarities between the past behavior of the user and that of other users. A canonical example is the GroupLens system that recommends news articles based on similarities between users' reading behavior (Resnick, et al. 1994). This technique has been applied to many areas from consumer products to web pages (Resnick & Varian, 1997; Kautz, 1998), and has become a standard marketing technique in electronic commerce.

The input to a CF system is a triple consisting of a user, an object that the user has an opinion about, and a rating that captures that opinion: $<u, o, r(u,o)>$. As ratings for a given user are accumulated, it becomes possible to correlate users on the basis of similar ratings and make predictions about unrated items on the basis of historical similarity. In other words, the goal is to find other users u_i whose ratings correlate well with some user u_0, and use these users' ratings on some new object o' to predict $r(u_0, o')$.

One of the central problems in applying CF is the task of gathering ratings. Early systems such as Ringo (Shardanand & Maes, 1995) asked users to supply ratings directly, but more recently, user interface concerns have led developers to seek implicit ratings from users, using observable variables such as dwell time on a particular web page, buying behavior, etc. The psychological status of such ratings has received little attention, but it is easy to demonstrate that simple one-dimensional scales do not capture the nuance of user preference. If I dislike the movie "Die Hard," is it because I deem it too violent, or not violent enough? The kind of recommendation that should be made would be vastly different in each case. With sufficient data to pinpoint users' preferences, this kind of ambiguity may be resolved. However, applications may not always have sufficient data about an item or a user. This is particularly a problem for new items, such as newly-released movies, that have few user ratings.

My research has investigated the creation of knowledge-based recommender systems (Burke, 1999b; Burke, in press) that use multi-scaled semantic ratings from users. For example, the restaurant guide Entree[1] (Burke, Hammond & Young, 1997) allows users to critique restaurants as too expensive or too traditional, among other dimensions. Beyond any possible role in collaborative filtering, these ratings have a function in the system's interface: each critique or "tweak" invokes a new retrieval redirecting the user towards items more likely to meet his or her needs.

Earlier work (Burke, 1999a) noted the potential for integrating collaborative and knowledge-based recommendation and described a framework under which such integration could be accomplished. I proposed a hybrid recommender system that uses its knowledge to generate the best possible set of recommendations and then uses collaborative filtering to break ties among them. As noted previously, such a hybrid avoids some of the "ramp-up" or "cold-start" problems associated with CF, since the system can make good recommendations without gathering any

[1] <URL: http://infolab.ils.nwu.edu/entree/>

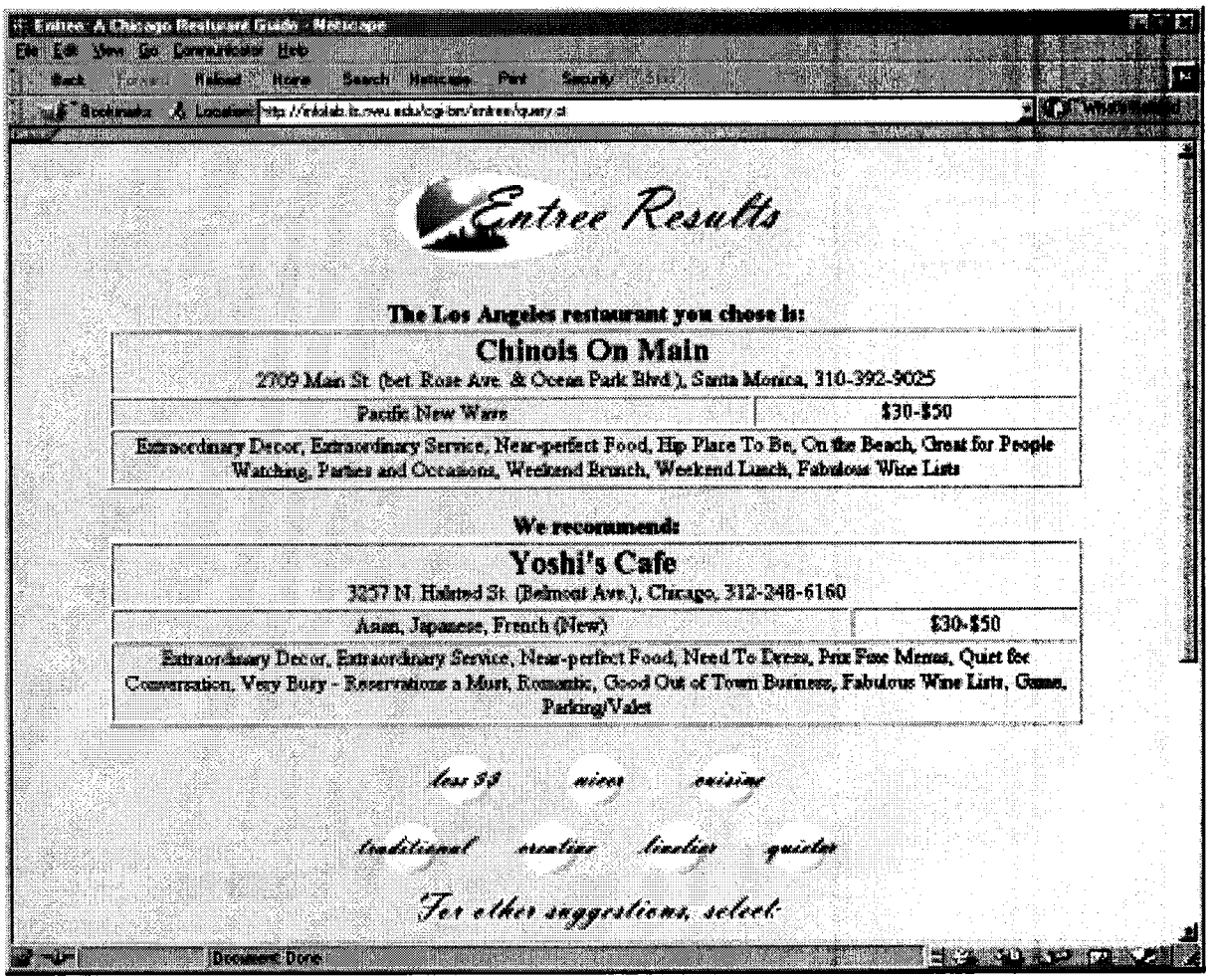

Figure 1. The Entree recommender system.

usage data, and it helps avoid the problems of knowledge-acquisition and database quality by allowing a knowledge-based system to refine its suggestions over time.

Entree

Consider a user who initiates an interaction with Entree using a known restaurant, Wolfgang Puck's "Chinois on Main" in Santa Monica.[2] As shown in Figure 1, the system finds a similar Chicago restaurant that combines Asian and French influences, "Yoshi's Cafe,"[3] as well as other restaurants (not shown) that are ranked by their similarity. The user, however, is interested in a cheaper meal and selects the "Less $$" button. The result (not shown) is a creative Asian restaurant in a cheaper price bracket: "Lulu's." However, the French influence is lost – one consequence of the move to a lower price bracket. The user can continue browsing and critiquing until an acceptable restaurant has been located.

Entree has been in continuous operation as a public web utility since July 1996. The experiments described below use logs through June 1999. The system does not retain any user data – it does not use cookies or other mechanisms to identify a user returning to the site. For this reason, these experiments may show something of a lower bound on the efficacy of CF: a site would normally be expected to gather user ratings over multiple sessions of interaction and tie them directly to a user identifier.[4] The log data was partitioned into sessions identified by IP address and terminating after 10 minutes of

[2] The user may also make a database query based on desired restaurant features.

[3] Note that the connection between "Pacific New Wave" cuisine and its Asian and French culinary components is part of the system's knowledge base of cuisines.

[4] On the other hand, ratings gathered over a longer period of time would reflect a diversity of search goals, a diversity presumably not present in a single search session.

inactivity. There are approximately 50,000 sessions in the 3 years of usage data.[5]

Approach

The research reported here attempts to estimate how much improvement might be expected from adding CF to an existing knowledge-based recommender system, and to determine what filtering algorithm would produce the best performance. The central issues are the generation of ratings and the computation of intra-user similarity.

Each session consists of a list of user actions and displayed restaurants. A number of restaurants are retrieved as likely candidates, but only one is highlighted. There are eight possible actions a user can take: "Less $$," "Nicer," "More Creative," "More Traditional," "Quieter," "Livelier," "Change Cuisine," and "Browse" (the choice to move to a different restaurant in the return list.) In the case of the user choosing an alternative cuisine, the log does not record the user's choice. A user can begin the session with a known restaurant as a starting point or with a query that describes the type of restaurant sought. (These queries were also not logged.) So, for each restaurant, we can associate one of 10 ratings: Entry point, Exit point, or one of the eight actions.

Sessions range in length from one to 20 interactions, but typically contain less than ten ratings. Occasionally, the same restaurant is rated more than once. For example, a user might see a recommendation, browse to the other restaurants in the list, return to the original suggestion, and then perform a tweaking action. We discard all but the most recent rating.

One possible collaborative filtering approach using this data is to project the multi-dimensional ratings onto a single dimension, a simple binary like / dislike scale. For example, if user A sees "Yoshi's Cafe" and says "Give me something cheaper," this can be recorded as a negative rating. If the user stops browsing upon encountering this restaurant, the user is assumed to have found what he or she seeks, and a positive rating is assigned.[6] Ratings produced in this way can be directly plugged into standard CF algorithms.

Another approach is to look only for users with exactly the same ratings: treating ratings on different scales as incommensurable. We would only match user A against others who also thought that "Yoshi's Cafe" was too expensive. We can formulate this version of the problem as an information retrieval task where each term consists of a restaurant / rating combination. For example, we can use a cosine measure of similarity, treating each session as a vector of binary values whose dimensionality is that of all restaurant/rating combinations. The drawback of this "sparse" metric is that only a small number of users would have had exactly the same reactions to a given set of restaurants, meaning that predictions will be based on a smaller number of users than in a collaborative approach.

A third technique takes into account the semantics of the ratings themselves: a similarity metric is created based on the relationships between ratings. For example, if user B looks at "Yoshi's Cafe" and says "Give me something nicer," we should probably rate users A and B as dissimilar even though they both disliked the same restaurant – they did so for essentially opposite reasons. This is the "heuristic similarity" approach. It does not establish a single scale onto which all ratings are projected, but rather looks at similarity on a rating-by-rating basis. A similarity value is assigned to each possible pair of ratings, using an adjacency table generated by considering the semantics of each response type, and a few common-sense considerations:

- A rating is maximally similar to itself.
- "Browse" is not similar to any other rating.

[5] The Entree data set is available at the UCI KDD Archive <URL: http://kdd.ics.uci.edu/ >.

[6] It is possible that the user has given up in frustration, in which case a positive rating would be inappropriate. This kind of ambiguity is common in other CF domains, such as web log analysis. A system that was tied to e-commerce transactions or in the case of restaurants, on-line reservation placement, would have more reliable positive ratings.

Table 1: Adjacency matrix for Entree ratings

Browse	Cheaper	Nicer	Trad.	Creat.	Lively	Quiet	Cuisine	Entry	Exit	
1	0	0	0	0	0	0	0	0	0	Browse
	1	-1	-0.5	-0.5	-0.5	-0.5	0	0	0	Cheaper
		1	0.5	0.5	-0.5	0.5	0	0	0	Nicer
			1	-1	-0.5	0.5	0	0	0	Trad.
				1	0.5	-0.5	0	0	0	Creat.
					1	-1	0	0	0	Lively
						1	0	0	0	Quiet
							1	0	0	Cuisine
								1	1	Entry
									1	Exit

- Some ratings have natural opposites: "Livelier" / "Quieter", "Traditional" / "Creative."

The full comparison table is shown in Table 1. This metric takes the qualitative differences between ratings into account, but it allows more kinds of intra-user comparison than the sparse metric.

Evaluation

To evaluate different CF approaches, the data was randomly partitioned into equal-sized training and test sets of approximately 25,000 sessions. From the test set, only highly-active users were extracted, those with at least 15 restaurants rated, 199 users in total. These users provided enough individual data to evaluate each approach, even though they were a small minority of the test data. With user-identified data, a system would accumulate 15 ratings for a user very quickly.

The goal of CF for Entree is not strictly to predict ratings. A default negative score is highly effective at predicting ratings since almost 80% of the ratings are negative. Rather the task is to improve the quality of the recommendations made by the knowledge-based component. The evaluation technique reflects this application. See the outline of the algorithm in Figure 2. For each session *S*, the system first isolates a positively rated restaurant *r* – an item the user found satisfactory. The goal is to bring this restaurant to the user's attention as soon as possible, so the CF system must pick this positively-rated item from a group of negatively-rated ones.

To simulate this task, the system randomly selects 6 items from *S* with a negative rating, and groups them with *r* as the test partition *T*. Eight items then become training data for a particular user, excess ratings being discarded at random. (Five such training/test splits are generated for each session for cross-validation.) Using the training partition, a prediction is made for the rating of each test item $t \in T$, and the one with the highest predicted score is selected as the recommendation. Additional training data is added in steps, and predictive performance is reported when four, six or eight of the ratings are known to the system.

For the correlation filter, predictions of the rating of a test item *t* are made by selecting all users who have rated *t*, filtering those who meet a

```
Let S = A session: { s0, .., sn}, n>=15 where each si consists of a
        pair <restaurant, rating>
    r = a positive rating from S
    T = test data for the session, initially { }
    P(S, t), a prediction function that predicts the score of test
      item t, given ratings from training data in S.

move r from S to T
move 6 random s from S to T
for i <- 4, 6, 8
  p <- t such that P({s0,..si}, t) is maximized
  if p equals r
    correct prediction (i)
```

Figure 2. Evaluation algorithm

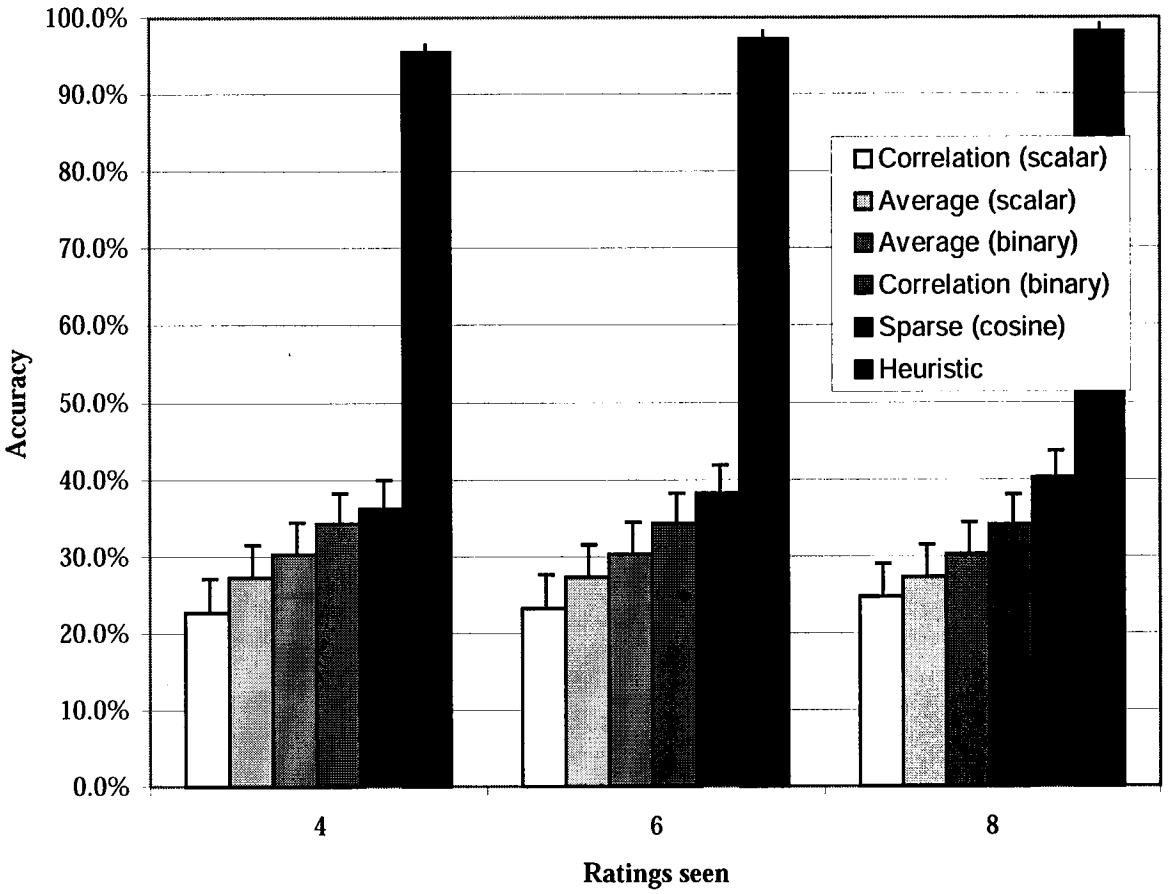

Figure 3. Experimental Results

minimum threshold of correlation with the test user based on the training data, and averaging their ratings of *t*.[7] The same experimental scheme was applied also to the sparse metric, using cosine similarity, and with the heuristic metric, using the average of the adjacency values for all items rated in common. As a baseline, we also used a predictor computed from the average rating for all users.

The experiment evaluated the two different ways of projecting ratings on a scale: a binary method and a scalar method. Under the binary method, most ratings have a score −1 (since they cause the user to move away from the restaurant in question). The only exceptions are "Entry point" and "Exit point," which are both treated as positive ratings (+1). The scalar method adds two additional distinctions: "Exit point" gets a slightly lower rating (0.8) since we cannot be fully confident that it signifies success as opposed to frustration, and "Browse" gets less of a negative rating (-0.5) to reflect the fact that users are not directly critiquing a restaurant, but instead poking around among other returned items. The hypothesis was that the scalar ratings, which incorporate some semantic information, would have performance in between that of the correlation and heuristic techniques.

The experiment therefore had six conditions: intra-user Correlation computed with both binary and scalar ratings, the Sparse metric using cosine similarity, and the Heuristic metric, together with a pair of predictors using the Average rating computed with the two rating techniques. Figure 3 shows the results of the experiment. (Error bars represent the 95% confidence interval.)

Surprisingly, the scalar version of the ratings was not better than the simpler binary projection. Both the Average and Correlation metrics using scalar ratings were slightly worse than the Average using binary ratings. The Sparse metric slightly outperformed Correlation using binary ratings, but this difference is not statistically significant. The Heuristic technique was shown to be the clear winner. It approached 100% accuracy at the task of predicting what restaurant the customer will like, and even with a small

[7] Since the rating data is relatively sparse, a default score of 0 (no preference) is assigned when an item is rated in one session but not in the other (Breese, Heckerman & Kadie, 1998).

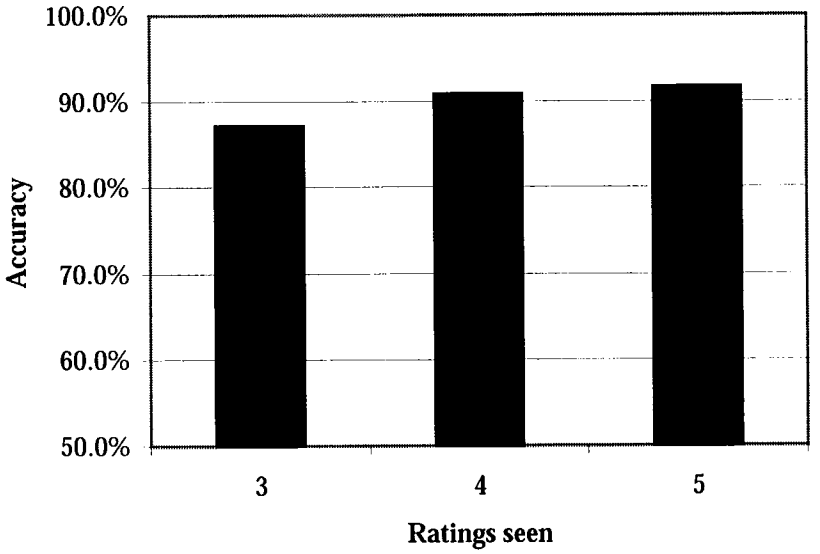

Figure 4. Heuristic prediction for users with 10 ratings or more

amount of data (4 ratings), better than 95% accuracy was achieved.

The performance of the Heuristic technique was further evaluated given a smaller amount of training data. A sample of the data (6600 sessions) was extracted and performance evaluated on shorter sessions (10 ratings or more, total of 264 users), using five ratings for testing and three, four and five ratings for training. Figure 4 shows these results, which are consistent with the performance for more active users. Having seen three ratings, the Heuristic metric can predict the preferred item from the five item test set with 87% accuracy, rising to 92% after all five ratings are seen.

Future Work

The heuristic approach to collaborative filtering described above is being applied at the commercial successor to Entree: Recommender.com, a provider of recommendation generation services to e-commerce sites.

The main drawback to the heuristic approach is that it involves additional knowledge engineering: for n possible ratings, an n^2 adjacency matrix must be generated. This is particularly a problem in domains that have a large number of possible tweaks, such as the domain of movies. A user might, for example, ask for a similar movie but without a particular actor: "Die Hard" without Bruce Willis. It would be impractical to manually construct an adjacency matrix rating the similarity of all such choices. Further research will investigate whether the similarity between two tweaks by learned from user data as follows.

We can begin with the assumption that all ratings are dissimilar, and then try to find evidence for similar. Suppose, for example, there are two users A and B whose ratings for many items are identical, but who differ on their rating of some item o: $r(A, o) = r_1$ and $r(B, o) = r_2$. We would take this as evidence that r_1 and r_2 should be considered similar ratings.

Conclusion

It is hardly surprising that the meaning of actions should be an important consideration in determining similarity between user profiles: why a user likes or dislikes something must surely be important. Nor is it surprising that heuristic similarity has received little attention, given that few applications have access to semantic critiques such as those used in Entree. However, the results described here suggest that even modest efforts invested in improving the semantic richness of a user interface enable the gathering of predictively-powerful usage data. In these experiments, the performance gains obtained by using better data far exceeds any incremental benefit that might be expected from applying better algorithms to data that is semantically weak.

Acknowledgements

These experiments were designed with the assistance of Daniel Billsus of UCI. The

development of Entree was supported at the University of Chicago by the Office of Naval Research under grant F49620-88-D-0058. The interface to the Entree system was designed and created by Robin Hunicke at the University of Chicago. Many others contributed to the FindMe effort at the University of Chicago, including Terrence Asselin, Kai Martin, Kass Schmitt, and Robb Thomas.

References

Breese, J. S.; Heckerman, D. and Kadie, C. 1998. "Analysis of Predictive Algorithms for Collaborative Filtering," In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43-52. San Francisco, CA: Morgan Kaufmann.

Burke, R. In press. Knowledge-based Recommender Systems. In A. Kent (ed.), *Encyclopedia of Library and Information Systems*. In press.

Burke, R. 1999a. Integrating Knowledge-Based and Collaborative-Filtering Recommender Systems. In *AAAI Workshop on AI in Electronic Commerce*. page 69-72. AAAI.

Burke, R. 1999b. The Wasabi Personal Shopper: A Case-Based Recommender System. In *Proceedings of the 11th National Conference on Innovative Applications of Artificial Intelligence*, pages 844-849, AAAI.

Burke, R., Hammond, K., and Young, B. 1997. The FindMe Approach to Assisted Browsing. *IEEE Expert*, 12(4), pp. 32-40.

Kautz, H. (ed.) 1998. *Recommender Systems: Papers from the AAAI Workshop*. AAAI Technical Report WS-98-08. AAAI.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. 1994. GroupLens: an open architecture for collaborative filtering of netnews. In *CSCW '94: Proceedings of the conference on Computer supported cooperative work*, 175-186. New York: ACM Press.

Resnick, P. and Varian, H. R. 1997. Recommender systems. *Communications of the ACM*, 40(3) 56-58.

Shardanand, U. and Maes, P. 1995. Social information filtering algorithms for automating "word of mouth" In *CHI-95: Conference proceedings on Human factors in computing systems*, 210-217. New York: ACM Press.