December 2004

A Link Analysis Approach to Recommendation under Sparse Data

Zan Huang
The University of Arizona

Daniel Zeng
The University of Arizona

Hsinchun Chen
The University of Arizona

A Link Analysis Approach to Recommendation under Sparse Data

Zan Huang
Department of Management Information Systems
The University of Arizona
zhuang@eller.arizona.edu

Daniel Zeng
Department of Management Information Systems
The University of Arizona
zeng@eller.arizona.edu

Hsinchun Chen
Department of Management Information Systems
The University of Arizona
hchen@eller.arizona.edu

ABSTRACT

Collaborative filtering is one most successful approach to recommendation reported in the literature. It automates the "Word of Mouth" recommendation by suggesting products liked by other consumers who showed similar preference patterns as the target consumer. A serious limitation of the collaborative filtering approach is the *sparsity problem*, referring to the situation where insufficient historical transactions are available for inferring reliable consumer similarities. In this paper, we represent the consumer transactional data as a graph consisting nodes representing consumers and products and links representing transactions. Under this consumer-product graph, we propose to explore the global graph structure to facilitate collaborative filtering under sparse data. We developed a link analysis recommendation algorithm based on the similar ideas implemented in Web graph analysis algorithms. We tested the proposed algorithm using an online bookstore dataset against standard collaborative filtering algorithms that do not consider transitive associations. The experimental results showed that our proposed algorithm outperformed the benchmark algorithms, especially when insufficient amount of transactional data is available.

Keywords

Recommender systems, collaborative filtering, sparsity problem, link analysis.

INTRODUCTION

Recommendation as a social process plays an important role in many applications for consumers, because it is overly expensive for every consumer to learn about all possible alternatives independently. Depending on the specific application setting, a consumer might be a buyer (e.g., in online shopping), an information seeker (e.g., in information retrieval), or an organization searching for certain expertise. Recommender systems have been developed to automate the recommendation process. Examples of research prototypes of recommender systems are: PHOAKS [26], Syskills and Webert [19], Fab [2], and GroupLens [13, 23]. These systems recommend various types of Web resources, online news, movies, among others, to potentially interested parties. Large-scale commercial applications of the recommender systems can be found at many e-commerce sites, such as Amazon, CDNow, Drugstore, and MovieFinder. These commercial systems recommend products to potential consumers based on previous transactions and feedback. They are becoming part of the standard e-business technology that can enhance e-commerce sales by converting browsers to buyers, increasing cross-selling, and building customer loyalty [24].

One of the most commonly-used and successful recommendation approaches is the collaborative filtering approach [9, 20, 25]. When predicting the potential interests of a given consumer, such an approach first identifies a set of similar consumers based on past transaction and product feedback information and then makes a prediction based on the observed behavior of these similar consumers. Despite its wide spread adoption, collaborative filtering suffers from several major limitations including sparsity, system scalability, and synonymy [21].

In this paper, we focus on the sparsity problem, which refers to the lack of prior transactional and feedback data that makes it difficult and unreliable to predict which consumers are similar to a given consumer. For instance, the recommender systems

used by online bookstores use past purchasing history to group consumers and then make recommendations to an individual consumer based on what the other consumers in the same group have purchased. When such systems have access only to a small number of past transaction records (relative to the total numbers of the books and consumers), however, determining which consumers are similar to each other and what their interests are becomes fundamentally difficult.

In our previous study, we have proposed to study the collaborative filtering algorithms in bipartite graphs [10]. In such graphs, one set of nodes represents products, services, and information items for potential consumption. The other set represents consumers or users. The transactions and feedback are modeled as links connecting nodes between these two sets. In this study, we developed a link analysis algorithm for collaborative filtering under this graph-based framework. We will refer to this graph as a *consumer-product graph* with consumer and product nodes in the remainder of the paper. Our research is motivated by link analysis algorithms such as HITS [12] and PageRank [5] for identifying important Webpages in a Web graph. The central research hypothesis is that the proposed link analysis recommendation algorithm is able to extract useful link structure information from the consumer-product graph and facilitates more effective recommendation with sparse transactional data.

The remainder of the paper is organized as follows. We present a brief review on collaborative filtering and the sparsity problem after the introduction section. We then present details of the proposed link analysis recommendation algorithm, followed by an experimental study where we compared the proposed algorithm with standard collaborative filtering algorithms. We conclude the paper by summarizing the key conclusions and future research directions.

LITERATURE REVIEW

In this section, we briefly survey previous research and system development on collaborative filtering and introduce the sparsity problem, which has been identified as one of the major technical challenges hindering the further development and adoption of collaborative filtering systems.

Collaborative Filtering

Collaborative filtering generates personalized recommendations by aggregating the experiences of similar users in the system. Conceptually, this approach automates the process of "word of mouth" recommendation. One key aspect of collaborative filtering is the identification of consumers or users similar to the one who needs a recommendation. Cluster models, Bayesian Network models, and specialized association-rule algorithms, among other techniques, have been used for this identification purpose [4, 15]. Based on similar consumers or neighbors, methods such as the most frequent item approach [21] can then be used to generate recommendations.

Collaborative filtering has been the most successful recommendation system approach to date [21] and has been widely applied in various applications [6, 7, 16-18, 23]. Despite its success in many application settings, the collaborative filtering approach nevertheless has been reported to have several major limitations including the sparsity, scalability, and synonymy problems [22]. The sparsity problem occurs when transactional or feedback data is sparse and insufficient for identifying neighbors and it is a major issue limiting the quality of recommendations and the applicability of collaborative filtering in general. Our study focused on developing an effective approach to making high-quality recommendations even when sufficient data is unavailable. The next subsection will discuss the sparsity problem in detail.

The Sparsity Problem

In collaborative filtering systems, users or consumers are typically represented by the items they have purchased or rated. For example, in an online bookstore selling 2 million books, each consumer is represented by a Boolean feature vector of 2 million elements. The value for each element is determined by whether this consumer has purchased the corresponding book in past transactions. Typically the value of 1 indicates that such a purchase had occurred and 0 indicates that no such purchase has occurred. When multiple consumers are concerned, a matrix composed of all vectors representing these consumers can be used to capture past transactions. We call this matrix the *consumer-product matrix*. The elements in such a matrix can represent purchases, ratings, or other interactions between the consumers and products.

We now introduce some notation to be used throughout the paper. We use C to denote the set of consumers and P the set of items. We denote the consumer-product matrix by a M x N matrix $A = (a_{ij})$ (M=|C|, N=|P|), such that

$$a_{ij} = \begin{cases} 1, & \text{if user } i \text{ purchased item } j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Note that in our study we focused on actual transactions that occurred, so a_{ij} is binary. In other recommendation scenarios such as those that involve ratings, a_{ij} can take other categorical or continuous values (e.g., 5-level rating scales and probabilities of interest).

In many large-scale applications such as major e-commerce Web sites, both the number of items, $|P|$, and the number of consumers, $|C|$, are large. In such cases, even when many transactions have been recorded, the consumer-product matrix can still be extremely sparse, that is, there are very few elements in A whose value is 1. This problem, commonly referred to as the sparsity problem, has a major negative impact on the effectiveness of a collaborative filtering approach. Because of sparsity, it is highly probable that the similarity (or correlation) between two given consumers is zero, rendering collaborative filtering useless [3]. Even for pairs of consumers that are positively correlated, such correlation measures may not be reliable.

Many researchers have attempted to alleviate the sparsity problem. Sarwar et al. proposed an item-based approach to addressing both the scalability and sparsity problems [Sarwar et al. 2001]. Based on the transactional or feedback data, items that are similar to those purchased by the target user in the past are identified and then recommended. Item similarities are computed as the correlations between the corresponding column (item) vectors. It is reported that in certain applications this item-based approach achieved better recommendation quality than the user-based approach, the predominant approach used in recommender systems, which relies on correlations between row (user) vectors.

Another proposed approach, dimensionality reduction, aims to reduce the dimensionality of the consumer-product matrix directly. A simple strategy is to form clusters of items or users and then use these clusters as basic units in making recommendations. More advanced techniques can be applied to achieve dimensionality reduction. Examples are statistical techniques such as Principle Component Analysis (PCA) [8] and information retrieval techniques such as Latent Semantic Indexing (LSI) [3, 22]. Empirical studies indicate that dimensionality reduction can improve recommendation quality significantly in some applications, but performs poorly in others [22]. The dimensionality reduction approach addresses the sparsity problem by removing unrepresentative or insignificant consumers or products to condense the consumer-product interaction matrix. However, potentially useful information might be lost during this reduction process. This may partially explain the mixed results reported on the performance of dimensionality reduction-based collaborative filtering approaches.

In our previous research, we proposed to represent the consumer-product matrix as a bipartite graph, with two groups of nodes representing users and items and links representing transactions [10, 11]. Under this graph representation, we explore the transitive associations among users and items to alleviate the sparsity problem. We applied associative retrieval techniques, specifically various spreading activation algorithms, as computational tools for such transitive association exploration. Our experimental results showed that these algorithms achieved significantly better performance under sparse transactional data. However, these algorithms typically are more computationally expensive than the standard collaborative filtering algorithms.

In this study, we extend from our previous research to study link analysis algorithms that exploit the global link structure in the consumer-product graph for more effective recommendation under sparse data. Motivated by the HITS and PageRank algorithms developed for identifying important webpages in a Web graph, we hypothesize that the incorporation of the global link structure of the consumer-product graph will generate more effective recommendations. In the next section, we present details of the proposed link analysis algorithm for recommendation.

A LINK ANALYSIS ALGORITHM FOR RECOMMENDATION

The most important application of link analysis to date is for identifying important webpages to facilitate effective web searching. The World Wide Web is a graph with linked hypertext documents. A link $p \rightarrow q$ typically indicates that webpage p suggests or recommends that surfers visiting p follow the link and visit q (such a link is called an *informative link* [12] as oppose to links for navigation purposes). Kleinberg [12] distinguished between two types of Web pages which pertain to a certain topic. The first are *authoritative* pages that contain important content information of the topic. The second type are *hub* pages. Hubs are primarily resource lists, linking to many authorities on the topic possibly without directly containing the authoritative information. According to this model, hubs and authorities exhibit a *mutually reinforcing relationship*: good hubs point to many good authorities, and good authorities are pointed at by many good hubs. In light of the mutually reinforcing relationship, hubs and authorities should form communities, which can be pictured as dense bipartite portions of the Web, where the hubs link densely to the authorities.

Another similar link analysis approach to webpage ranking is the link-structure-based ranking approach called *PageRank* employed by the *Google* search engine [5]. This approach can be interpreted as a stochastic analysis of some random-walk behavior through the entire WWW. This algorithm determines ranking of pages by assigning each page p a rank of its importance, called *PageRank*. Specifically, the PageRank of a page p is the probability of visiting p in a random walk of the

entire Web, where the set of states of the random walk is the set of pages, and each random step is of one of the following two types: (1) From the given state *s*, choose at random an outgoing link of *s*, and follow that link to the destination page. (2) Choose a Web page uniformly at random, and jump to it. Kleinberg's approach and PageRank both explore the global structure of a Web graph and infer link-structure properties to indicate importance of a webpage. The two approaches are integrated in [14] as a metaalgorithm to calculate authoritative and hub pages through forward and backward random walks.

The consumer-product graph in our application naturally forms a bipartite graph, and a link $c - p$ indicates certain associations between consumer c and product p. The link is informative in the sense that knowing a previous transaction $<c, p>$, p has the potential to represent part of c's interest and c could indicate potential target consumers of product p. Being different from Web searching applications, recommendation requires the identification of important products for individual consumers rather than generally important products. Thus we extend from the authoritative and hub scores to define a product representativeness score $pr(c^0, p)$ of product p and a consumer representativeness score $cr(c^0, c)$ of c for a particular consumer c^0. $pr(c^0, p)$ denotes how much the product p represents the interest of the target consumer c^0. $cr(c^0, c)$ represents the degree to which the consumer c's behavior can be used to predict the target consumer c^0's behavior.

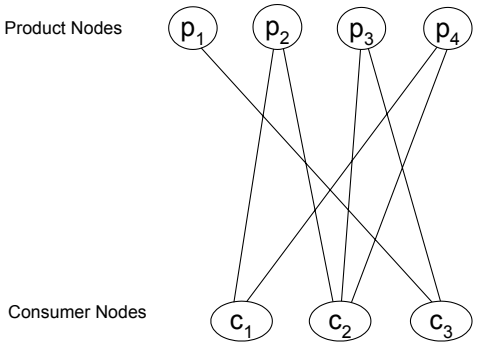

Figure 1. A simple consumer-product graph

We use a simple example consumer-product graph in Figure 1 to illustrate the proposed algorithm. In this example, three consumer nodes and four product nodes are connected by seven links (previous transactions). The corresponding consumer-product matrix, A, is

$$\begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & p_3 & p_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}.$$

Given a consumer-product matrix $A(|P| \times |C|) = \{a_{ij}\}$ as introduced previously. We use a consumer representativeness matrix $CR(M \times M) = \{cr_{it}\}$, where $cr_{it} = cr(i, t)$ and a product representativeness matrix $PR(M \times N) = \{pr_{ik}\}$, where $pr_{ik} = pr(i, k)$ to store the consumer and product representative scores.

The proposed link analysis algorithm is recursive in nature, as similar to Kleinberg's approach [12]. The main idea is that the consumer representativeness scores are derived based on the representativeness scores of the products linked with the consumers. The product representativeness scores are in turn derived based on the representativeness scores of the consumers associated. The key design issue is the score updating function $CR = g(PR)$ and $PR = f(CR)$.

In our simple example, suppose that a product representativeness score matrix PR is available. Consider the product representativeness scores for target consumer c_1, (pr_{11}, pr_{12}, pr_{13}, pr_{14}). For deriving the consumer representativeness scores of target consumer c_1, a simple approach is to sum up the product representativeness scores associated with a particular consumer. For example, the representativeness score of c_2 for the target consumer c_1 can be calculated as $pr_{12} + pr_{13} + pr_{14}$ (following Kleinberg's implementation). The problem with this approach is that if a consumer has links to all products, that consumer will have the highest representativeness scores for all target consumers. However, such a consumer's behavior actually provides little information to the prediction of the behavior of the target consumer. To address this problem, we

normalize the product representativeness score sums by dividing the total number of links the consumer node has to product nodes. Thus the consumer who has links to all products will be assigned relatively low representativeness score unless the target consumer also has links to most products. A consumer that has purchased exactly the same products as the target consumer will be assigned a representativeness score as high as that of the target consumer. Mathematically, the consumer representativeness score updating function *g* can be written as $cr_{it} = \frac{\sum_{a_{ij}=1} pr_{ij}}{\sum_{a_{tj}=1} a_{tj}}$.

Similarly the product representativeness function is written as $pr_{ik} = \frac{\sum_{a_{jk}=1} cr_{ij}}{\sum_{a_{jk}=1} a_{jk}}$. Thus a product that has links to all consumers will be assigned relatively low representativeness score unless the all consumers have high representativeness scores for the target consumer. A product that has been only purchased by the target consumer will be assigned the same representativeness score as the target consumer himself. This approach in fact incorporates the random walk idea implemented in the PageRank algorithm. In this sense, our approach actually integrated the two algorithms as in a different way from that in [14].

We summarize the complete procedure of the proposed link analysis algorithm as follows.

- **Preparation**: Based on the consumer-product matrix *A*, compute matrices *B* and *C* to be used in the updating process. $B = (\frac{a_{ij}}{\sum_i a_{ij}})$ (*M* X *N*), $C = (\frac{a_{ij}}{\sum_j a_{ij}})$ (*M* X *N*). Prepare the target set *T* to store indices of all target consumers that need recommendations.
- **Initialization**: $cr_{it} = \begin{cases} 1, \text{if } t = i \text{ and } i \text{ is in the target set } T, \\ 0, \qquad\qquad \text{otherwise.} \end{cases}$, $pr_{ik} = 0$. Create a source consumer score matrix $CR^0 = CR$.
- **Representativeness score update**: $PR = CR \cdot B$, $CR = PR \cdot C^T + CR^0$. The source consumer scores are added to the updated consumer representativeness score matrix to maintain the high representativeness scores for the target consumers themselves.

The score updating procedure is repeated until convergence or after a specified number of iterations.

EXPERIMENTAL STUDY

We conducted an experiment using data from an online bookstore to evaluate the effectiveness of the proposed link analysis algorithm. In this section, we first describe the experimental data and present the evaluation design and performance measures used in our study. We then summarize our experimental findings with the proposed link analysis algorithm and standard collaborative filtering algorithms.

Experimental Data

A major Chinese online bookstore (www.books.com.tw) provided us with data covering a portion of five years of recent transactions. This data set corresponds to a graph with 9,695 book nodes, 2,000 customer nodes, and 18,771 links (transactions).

Evaluation Design and Measurement

To evaluate the performance of different recommendation methods, we adopted a holdout testing approach similar to those used in [1, 21]. For each target consumer we retrieved the entire set of previously purchased products and sorted them into chronological order by purchase date. The first half of these products was treated as "past" purchases to serve as input to be fed into different methods to generate recommendations. For comparison purposes the second half of these products were treated as "future" purchases of the customer and hidden from the recommender system.

In our study, we use precision, recall, F-measure, and rank score as defined as follows, to measure the effectiveness of a given recommendation approach. The first three measures are widely accepted in information retrieval and recommender system research [3, 21].

$$\text{Precision} = \frac{\text{Number of recommended books that match with future purchases}}{\text{Total number of recommended books}}$$

$$\text{Recall} = \frac{\text{Number of recommended books that match with future purchases}}{\text{Total number of books in future purchases}}$$

$$\text{F'} = \frac{2 \times \text{Precision} \times \text{Recall} + \varepsilon^2}{\text{Precision} + \text{Recall} + \varepsilon}, \varepsilon \to 0$$ [1]

Because the algorithms in our study generate recommendations as a ranked list, we also adopted the rank scoring metric in our study [4]. In this metric, the expected utility of a ranked list of book recommendations (sorted by index *j*) for consumer *i* is defined as:

$$R_i = \sum_j \frac{p(i,j)}{2^{(j-1)/(h-1)}}, \text{ where } p(i,j) = \begin{cases} 1, & \text{if item } j \text{ is in user } i\text{'s future purchase list,} \\ 0, & \text{otherwise.} \end{cases}$$

The parameter *h* is the viewing halflife (the rank of the book on the list such that there is a 50% chance the consumer will review that book), which was set to 10 in our experiments. The rank scoring measure is based on the notion that each successive product in a list is less likely to be viewed by the consumer with an exponential decay. The final recommendation utility score over all the test customers is:

$$R = 100 \frac{\sum_i R_i}{\sum_i R_i^{\max}}$$

where $R_i^{\max}$ is the maximum achievable utility if all future purchases of consumer *i* had been at the top of the ranked list. In our experiments, we set the number of recommendations for all collaborative filtering approaches studied to 10. Thus, the recommendation list contained exactly 10 books.

In our experimental study, we compared the link analysis algorithm with two standard collaborative filtering algorithms that do not incorporate global link structure of a consumer-product graph: the consumer-based algorithm and item-based algorithm that form consumer neighborhoods and product neighborhoods respectively. The consumer-based algorithm calculates consumer similarities using the vector similarity function and then recommends products based on the purchases of customers that are similar to the target customer [Breese et al. 1998]. The item-based algorithm calculates product similarities instead of consumer similarities based on the transactional data and then recommends products that are similar to the target customer's previous purchases [Sarwar et al. 2001]. We applied the vector similarity function to calculate the product similarities.

Experiment Procedures and Results

For evaluation purposes, we used 30% of the transactions that were most recent ones as the test set and use the remaining 70% transaction records as the training set. In order to study the performances of the algorithms under data of different sparsity levels, we form four training datasets by randomly selecting 10%, 20%, 40%, from the 70% transactions and the entire 70% transactions. These training datasets were used to form the consumer-product interaction matrices for generating recommendations. For each training dataset, the test dataset was filtered to only include transactions that are possible to predict (A purchase of a product not appeared in the training dataset or a purchase by a consumer that not appeared in the training set is not possible to predict). All consumers in the filtered test dataset were included into the set of target consumers to generate recommendations. For each consumer, an ordered list of 10 recommendations was generated.

The performances of the three algorithms measured by the metrics described previously are presented in Table 1. We observe that the link analysis algorithm significantly outperformed the consumer-based and item-based collaborative filtering algorithms in all four measures. With the entire set of 70% transactions as training data, the link analysis algorithm recommended about 3% books from the recommendation list that match the consumer's future purchases (the precision of

[1] We modified the standard formulation of the F-measure by adding small number ε and ε^2 to the denominator and nominator respectively. This modification will assure valid values of F'-measure when precision or recall is equal to zero, in which case the F'-measure will be $\varepsilon \sim 0$. When precision and recall take nonzero values, the modified F-measure (F') will be very close to the original F-measure.

0.027749) and about 17% of the consumer's future purchases (the recall of 0.166679). These percentages were significantly lower for the consumer-based and item-based algorithms. The rank score of the link analysis algorithm was also higher than the other two algorithms (20.71553 for the link analysis algorithm with 70% training data), indicating that consumer's future purchases were more likely to be placed on the top of the recommendation list.

Precision

	Training Set Percentage			
Algorithm	**10%**	**20%**	**40%**	**70%**
User-based	0.001047	0.002094	0.005236	0.007853
Item-based	0.003665	0.001047	0	0.002618
Link Analysis	**0.005759**	**0.017277**	**0.014136**	**0.027749**

Recall

	Training Set Percentage			
Algorithm	**10%**	**20%**	**40%**	**70%**
User-based	0.010471	0.015707	0.040576	0.064136
Item-based	0.012216	0.003927	0	0.016754
Link Analysis	**0.047557**	**0.113101**	**0.093468**	**0.166679**

F

	Training Set Percentage			
Algorithm	**10%**	**20%**	**40%**	**70%**
User-based	0.001905	0.00365	0.009078	0.013759
Item-based	0.005249	0.001622	0.000001	0.004348
Link Analysis	**0.010042**	**0.028494**	**0.023346**	**0.045226**

Rank Score

	Training Set Percentage			
Algorithm	**10%**	**20%**	**40%**	**70%**
User-based	1.570681	2.094241	7.068063	10.20942
Item-based	1.22164	1.04712	0	2.443281
Link Analysis	**4.886562**	**19.5637**	**10.45375**	**20.71553**

Table 1. Experimental Results

All three algorithms generally achieved better performances as more training data were used, with the exception that the item-based algorithm failed to generate any correct recommendation when 40% data were used for training. The item-based algorithm generally achieved the lowest performance within the three algorithms. This is mainly because in our dataset there were much larger numbers of books than customers. Thus forming product neighborhoods was actually more difficult than forming consumer neighborhoods. We also observed the link analysis algorithm achieved relatively good performances after 20% of the data were used for training. This shows that the proposed link analysis algorithm has the potential for effective recommendation when only sparse data is available.

CONCLUSION

In this study, we developed a link analysis algorithm that incorporates global link structure of a consumer-product graph for effective recommendation under sparse transactional data. Our experimental study using data from an online bookstore showed that the proposed algorithm achieved significantly better performance than two standard collaborative filtering algorithms that form consumer or product neighborhoods without considering the global link structure. In our future studies, we will test the performances of other forms of the representativeness score updating functions. We will also test the proposed algorithm using various types of recommendation data. Another extension of our research is to compare the link analysis algorithm with other recently proposed algorithms specifically designed for alleviating the sparsity problem of collaborative filtering systems.

ACKNOWLEDGMENTS

This research was supported in part by the following grants: NSF Digital Library Initiative-II, "High-performance Digital Library Systems: From Information Retrieval to Knowledge Management," IIS-9817473, April 1999 – March 2002, and NSF Information Technology Research, "Developing A Collaborative Information and Knowledge Management Infrastructure," IIS-0114011, September 2001 – August 2004. We would also like to acknowledge books.com.tw for providing us with the dataset and their assistance during the project.

REFERENCES

1. Aggarwal, C. C., Wolf, J. L., Wu, K.-L., and Yu, P. S. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. in Proceedings of the Fifth ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'99) (San Diego, CA, 1999), 201-212.
2. Balabanovic, M., and Shoham, Y. Fab: Content-based, collaborative recommendation. Communications of the ACM, 40, 3 (1997), 66-72.
3. Billsus, D., and Pazzani, M. J. Learning collaborative information filters. in Proceedings of the 15th International Conference on Machine Learning (1998), 46-54.
4. Breese, J. S., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. in Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (Madison, WI, 1998), Morgan Kaufmann, 43-52.
5. Brin, S., and Page, L. The anatomy of a large-scale hypertextual web search engine. in Proceedings of the 7th International World Wide Web Conference (Brisbane, Australia, 1998).
6. Burke, R. Semantic ratings and heuristic similarity for collaborative filtering. in Proceedings of the Seventeenth National Conference on Artificial Intelligence (2000).
7. Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. Combining content-based and collaborative filters in an online newspaper. in Proceedings of the ACM SIGIR Workshop on Recommender Systems (1999).
8. GOLDBERG, K., ROEDER, T., GUPTA, D., and PERKINS, C. Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval, 4, 2 (2001), 133-151.
9. Hill, W., Stead, L., Rosenstein, M., and Furnas, G. Recommending and evaluating choices in a virtual community of use. in Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems (1995), 194-201.
10. Huang, Z., Chen, H., and Zeng, D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Transactions on Information Systems (TOIS), 22, 1 (2004), 116-142.
11. Huang, Z., Chung, W., and Chen, H. A graph model for e-commerce recommender systems. Journal of the American Society for Information Science and Technology (JASIST), 55, 3 (2004), 259-274.

12. Kleinberg, J. Authoritative sources in a hyperlinked environment. in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (1998).
13. Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. Grouplens: Applying collaborative filtering to usenet news. Communications of the ACM, 40, 3 (1997), 77-87.
14. Lempel, R., and Moran, S. Salsa: The stochastic approach for link-structure analysis. ACM Transactions on Information Systems, 19, 2 (2001), 131-160.
15. Lin, W., Alvarez, S. A., and Ruiz, C. Efficient adaptive-support association rule mining for recommender systems. Data Mining and Knowledge Discovery, 6, 1 (2002), 83-105.
16. Mobasher, B., H. Dai, T. L., Nakagawa, M., Sun, Y., and Wiltshire, J. Discovery of aggregate usage profiles for web personalization. in Proceedings of the Workshop on Web Mining for E-Commerce - Challenges and Opportunities (2000).
17. Nasraoui, O., Frigui, H., Joshi, A., and Krishnapuram, R. Mining web access logs using relational competitive fuzzy clustering. in Proceedings of the Eighth International Fuzzy Systems Association World Congress - IFSA 99 (1999).
18. Pazzani, M. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13, 5-6 (1999), 393-408.
19. Pazzani, M., and Billsus, D. Learning and revising user profiles: The identification of interesting web sites. Machine Learning, 27, 3 (1997), 313-331.
20. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. Grouplens: An open architecture for collaborative filtering of netnews. in Proceedings of the ACM CSCW'94 Conference on Computer-Supported Cooperative Work (1994), 175-186.
21. Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. Analysis of recommendation algorithms for e-commerce. in Proceedings of the ACM Conference on Electronic Commerce (2000), 158-167.
22. Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Application of dimensionality reduction in recommender systems: A case study. in Proceedings of the WebKDD Workshop at the ACM SIGKKD (2000).
23. Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., and Riedl, J. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. in Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW) (1998), 345-354.
24. Schafer, J., Konstan, J., and Riedl, J. E-commerce recommendation applications. Data Mining and Knowledge Discovery, 5, 1-2 (2001), 115-153.
25. Shardanand, U., and Maes, P. Social information filtering: Algorithms for automating "word of mouth". in Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems (1995), 210-217.
26. Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. Phoaks: A system for sharing recommendations. Communications of the ACM, 40, 3 (1997), 59-62.