A Comparative Study of Recommendation Algorithms in E-Commerce Applications*

Zan Huang, Daniel Zeng, Hsinchun Chen
Department of Management Information Systems
Eller College of Management
University of Arizona

Abstract

We evaluate a wide range of recommendation algorithms on e-commerce-related datasets. These algorithms include the popular user-based and item-based correlation/similarity algorithms as well as methods designed to work with sparse transactional data. Data sparsity poses a significant challenge to recommendation approaches when applied in e-commerce applications. We experimented with approaches such as dimensionality reduction, generative models, and spreading activation, which are designed to meet this challenge. In addition, we report a new recommendation algorithm based on link analysis. Initial experimental results indicate that the link analysis-based algorithm achieved the overall best performances across several e-commerce datasets.

Keywords

Recommender systems, collaborative filtering, algorithm design and evaluation, e-commerce

* A short version of this paper is currently under review at IEEE Intelligent Systems. The link analysis algorithm was originally reported in a conference paper presented at the Tenth Annual Americas Conference on Information Systems (AMCIS 2004), which received the Best Paper Award in SIGDSS (Special Interest Group in Decision Support Systems).

1. Introduction

Recommender systems are being widely used in many application settings to suggest products, services, and information items to potential consumers. For example, a wide range of companies such as *Amazon.com, Netflix.com*, *Half.com*, *CDNOW*, *J.C. Penney*, and *Procter & Gamble* have successfully deployed commercial recommender systems and reported increased Web and catalog sales and improved customer loyalty [1]. Many software companies provide stand-alone generic recommendation technologies. The top five providers include *Net Perceptions*, *Epiphany*, *Art Technology Group*, *BroadVision*, and *Blue Martini Software.* These five companies combined have a total market capital of over $600 million as of December 2004 (finance.yahoo.com). Recent years have also seen development of industry-specific recommender systems embedded in various business and e-commerce systems (e.g., *TripleHop Technologies*' *TripMatcher* for the travel industry and *Yahoo!*'s *LAUNCHcast* for online music broadcasting).

At the heart of recommendation technologies are the algorithms for making recommendations based on various types of input data. In e-commerce, most recommendation algorithms take as input the following three types of data: product attributes, consumer attributes, and previous interactions between consumers and products (e.g., buying, rating, and catalog browsing).

Models based on the product or consumer attributes attempt to explain consumer-product interactions based on these intrinsic attributes (e.g., [2-4]). Intuitively these models learn either explicitly or implicitly rules such as "Joe likes science fiction books" and "well-educated consumers like *Harry Potter.*" Techniques such as regression and classification algorithms have been used to derive such models. The performances of these approaches, however, rely heavily on high-quality consumer and product attributes that are often difficult or expensive to obtain.

Collaborative filtering-based recommendation takes a different approach by utilizing only the consumer-product interaction data and deliberately ignoring the consumer and product attributes [5-7]. Based solely on interaction data, consumers and products are characterized implicitly by their previous interactions. The simplest example of recommendation based only on interaction data is to recommend the most popular

products to all consumers. Collaborative filtering has been reported to be the most widely adopted and successful recommendation approach and researchers are actively advancing collaborative filtering technologies in various aspects including algorithmic design, human-computer interaction design, consumer incentive analysis, and privacy protection (e.g., [8-10]).

Despite significant progress made in collaborative filtering research, there are several problems limiting its applications in e-commerce. One major problem is that most research has focused on recommendation from multi-graded rating data that explicitly indicate consumers' preferences, whereas the available data about consumer-product interactions in e-commerce applications are typically binary transactional data (e.g., whether a purchase was made or not). Although algorithms developed for multi-graded rating data can be applied, typically with some modifications, to binary data, these algorithms are not able to exploit the special characteristics of binary transactional data to achieve more effective recommendation. A second problem is lack of understanding of relative strengths and weaknesses of different types of algorithms in e-commerce applications. The need for such comparative studies is evident in many recent studies that have proposed new algorithms but only conducted limited comparative evaluation. The third problem with collaborative filtering as a general-purpose e-commerce recommendation approach is the *sparsity problem*, which refers to the lack of prior transactional and feedback data that makes it difficult and unreliable to infer consumer similarity and other patterns for prediction purposes. Research on high-performance algorithms under sparse data is emerging [4, 11-13], but substantial additional research effort is needed to provide solid understanding of them.

Our research is focused on addressing the above problems. Our ultimate goal is to develop a meta-level guideline that "recommends" an appropriate recommendation algorithm for a given application that demonstrates certain data characteristics. In this article, we present the initial results of experimental work towards this goal with two specific objectives: (a) evaluating collaborative filtering algorithms with different e-commerce datasets and (b) assessing the effectiveness of different algorithms with sparse data.

2. Recommendation Algorithms

We now present six types of representative collaborative algorithms including those designed to alleviate the sparsity problem and a new algorithm we recently developed based on the ideas from link analysis.

We first introduce a common notation for describing a collaborative filtering problem. The input of the problem is an $M \times N$ *interaction matrix* $A = (a_{ij})$ associated with M consumers $C = \{c_1, c_2, ..., c_M\}$ and N products $P = \{p_1, p_2, \ldots, p_N\}$. We focus on recommendation that is based on transactional data, thus a_{ij} has two possible values of 0 and 1. A value of 1 represents an observed transaction between c_i and p_j (for example, c_i has purchased p_j). We consider the output of a collaborative filtering algorithm to be *potential scores* of products for individual consumers that represent possibilities of future transactions. A ranked list of K products with the highest potential scores for a target consumer serves as the recommendations.

2.1 User-based Algorithm

The user-based algorithm, which has been well-studied in the literature, predicts a target consumer's future transactions by aggregating the observed transactions of similar consumers. The algorithm first computes a consumer similarity matrix $WC = (wc_{st})$, $s, t = 1, 2, \ldots, M$. The similarity score wc_{st} is calculated based on the row vectors of A using a vector similarity function (such as in [10]). A high similarity score wc_{st} indicates that consumers s and t may have similar preferences since they have previously purchased many common products. $WC \cdot A$ gives potential scores of the products for each consumer. The element at the cth row and pth column of the resulting matrix aggregates the scores of the similarities between consumer c and other consumers who have purchased product p previously.

2.2 Item-based Algorithm

The item-based algorithm is different from the user-based algorithm only in that product similarities are computed instead of consumer similarities. This algorithm has been shown to provide higher efficiency and comparable or better recommendation quality

than the user-based algorithm for many datasets [14, 15]. This algorithm first computes a product similarity matrix $WP = (wp_{st})$, s, t = 1, 2, …, N. Here, the similarity score wp_{st} is calculated based on column vectors of A. $A \cdot WP$ gives the potential scores of the products for each consumer.

2.3 Dimensionality Reduction Algorithm

The dimensionality reduction-based algorithm first condenses the original interaction matrix and generates recommendations based on the condensed and less sparse matrix to alleviate the sparsity problem [11, 16]. The standard *singular vector decomposition* procedure is applied to decompose the interaction matrix A into $U \cdot Z \cdot V'$, where U and V are two orthogonal matrices of size $M \times R$ and $N \times R$ respectively and R is the rank of matrix A. Z is a diagonal matrix of size $R \times R$ having all singular values of matrix A as its diagonal entries. The matrix Z is then reduced by retaining only k largest singular values to obtain Z_k. U and V are reduced accordingly to obtain U_k and V_k. Consumer similarities are derived based on U_k and $Z_k^{1/2}$. Recommendations are then generated in the same fashion as described in the user-based algorithm.

2.4 Generative Model Algorithm

Under this approach, latent class variables are introduced to explain the patterns of interactions between consumers and products [13, 17]. Typically one can use one latent class variable to represent the unknown cause that governs the interactions between consumers and products. The interaction matrix A is considered to be generated from the following probabilistic process: (1) select a consumer with probability $P(c)$; (2) choose a latent class with probability $P(z|c)$; and (3) generate an interaction between consumer c and product p (i.e., setting a_{cp} to be 1) with probability $P(p|z)$. Thus the probability of observing an interaction between c and p is given by $P(c, p) = \sum_z P(c)P(z \mid c)P(p \mid z)$. Based on the interaction matrix A as the observed data, the relevant probabilities and conditional probabilities are estimated using a maximum likelihood procedure called *Expectation Maximization* (*EM*). Based on the estimated probabilities, $P(c, p)$ gives the potential score of product p for consumer c.

2.5 Spreading Activation Algorithm

In our previous research, we have proposed a graph-based recommendation approach based on the ideas of associative information retrieval [12]. This approach addresses the sparsity problem by exploring transitive associations between consumers and products in a bipartite *consumer-product graph* that corresponds with the interaction matrix A. The spreading activation algorithms developed in associative information retrieval can then be adopted to accomplish transitive association exploration efficiently. In this study we used an algorithm with competitive performance in recommendation applications, the *Hopfield net* algorithm [12]. In this approach, consumers and products are represented as nodes in a graph, each with an activation level μ_j, $j = 1, \ldots, N$. To generate recommendations for consumer c, the corresponding node is set to have activation level 1 ($\mu_c = 1$). Activation levels of all other nodes are set at 0. After initialization the algorithm repeatedly performs the following activation procedure: $\mu_j(t+1) = f_s\left[\sum_{i=0}^{n-1} t_{ij}\mu_i(t)\right]$, where f_s is the continuous *SIGMOID* transformation function or other normalization functions; t_{ij} equals η if i and j correspond to an observed transaction and 0 otherwise ($0 < \eta < 1$). The algorithm stops when activation levels of all nodes converge. The final activation levels μ_j of the product nodes give the potential scores of all products for consumer c.

2. 6 Link Analysis Algorithm

In this article we report a new recommendation algorithm based on the ideas from link analysis research. Link analysis algorithms have found significant application in Web page ranking and social network analysis (notably, *HITS* [18] and *PageRank* [19]). Our algorithm is an adaptation of the *HITS* algorithm in the recommendation context.

The original *HITS* algorithm distinguishes between two types of Web pages that pertain to a certain topic: *authoritative* pages, which contain definitive high-quality information, and *hub* pages, which are comprehensive lists of links to authoritative pages. A given webpage i in the Web graph has two distinct measures of merit, its authority score a_i and its hub score h_i. The quantitative definitions of the two scores are recursive. The authority score of a page is proportional to the sum of hub scores of pages linking to it, and conversely, its hub score is proportional to the authority scores of the pages to

which it links. These definitions translate to a set of linear equations: $a_i = \sum_j G_{ji} h_j$ and $h_i = \sum_j G_{ij} a_j$, where G is the matrix representing the links in the Web graph. Using the vector notation, $a = (a_1, a_2, \ldots, a_n)$ and $h = (h_1, h_2, \ldots, h_n)$, we can express the equations in compact matrix form: $a = G' \cdot h = G' \cdot G \cdot a$ and $h = G \cdot a = G \cdot G' \cdot h$. The solutions of the above equations correspond to eigenvectors of $G' \cdot G$ (for a) and $G \cdot G'$ (for h). Computationally, it is often more efficient to start with arbitrary values of a and h and repeatedly apply $a = G' \cdot h$ and $h = G \cdot a$ with a certain normalization procedure at each iteration. Subject to some mild assumptions, this iterative procedure is guaranteed to converge to the solutions [18].

In our recommendation application, the consumer-product graph forms a bipartite graph consisting of two types of nodes, consumer and product nodes. A link between a consumer node c and a product node p indicate that p has the potential to represent part of c's interest and at the same time c could partially represent product p's consumer base. Compared to Web page ranking, recommendation requires the identification of products of interest for individual consumers rather than generally popular products. As such, we adapt the original authority and hub scores definitions for recommendation purposes. Specifically, we define a product representativeness score $pr(p, c^0)$ of product p with respect to consumer c^0, which can be viewed as a measure of the "authority" of product p in terms of the level of interest it will have for consumer c^0. Similarly, we define a consumer representativeness score $cr(c, c^0)$ of c with respect to consumer c^0, which measures how well consumer c, as a "hub" for c^0, associates with products of interest to c^0.

In contrast to the vector representation of Web page authority and hub scores, we denote by $PR = (pr_{ik})$ the $N \times M$ product representativeness matrix, where $pr_{ik} = pr(i, k)$, and by $CR = (cr_{it})$ the $M \times M$ consumer representativeness matrix, where $cr_{it} = cr(i, t)$. Directly following the idea of the recursive definition of authority and hub scores, one would define the product and consumer representativeness scores as $PR = A' \cdot CR$ and $CR = A \cdot PR$. Intuitively, the sum of the product representativeness scores of the products linked to a consumer gives the consumer representativeness score and vice versa.

However, these definitions have several inherent problems. First, 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 for predicting the behavior of the target consumer. A more fundamental problem with these definitions is that with the convergence property shown in [18], *PR* and *CR* defined above will converge to matrices with identical columns, amounting to scores representing product ranking independent of particular consumers, thus providing only limited value for recommendation.

To address these problems, we have adopted the following consumer representativeness score definition: $CR = B \cdot PR + CR^0$, where $B = (b_{ij})$ is an $M \times N$ matrix derived from A: $b_{ij} = \frac{a_{ij}}{(\sum_j a_{ij})^{\gamma}}$ and CR^0 is the source consumer representativeness score matrix: $cr_{ij}^0 = \begin{cases} \eta, \text{ if } i = j, \\ 0, \text{otherwise} \end{cases}$ (i.e., $CR^0 = \eta\ I_M$, where I_M is an $M \times M$ identity matrix). This adaptation adopts ideas similar to those successfully applied in network spreading activation models [20]. With the introduction of matrix *B*, we normalize the representativeness score a consumer receives from linked products by dividing it by the total number of products she is linked to. In other words, a consumer who has purchased more products needs to have more overlapping purchases with the target consumer than a consumer with a smaller number of total purchases to be considered equally representative of the target consumer's interest. The parameter γ controls the extent to which a consumer is penalized because of having made large numbers of purchases. This type of adjustment is well studied in modeling the decaying strength in the spreading activation literature [20]. We have tested γ within the range of 0 to 1, and used 0.9 in our final experiments. The source matrix CR^0 is included to maintain the high representativeness scores for the target consumers themselves and to customize the representativeness score updating process for each individual consumer. In order to maintain consistent levels of consumer self-representativeness, in the actual computation we normalize the matrix multiplication result $B \cdot PR$ before adding the source matrix CR^0. The normalization process is such that each column of $B \cdot PR$ (corresponding to the consumer representativeness score for each target consumer) adds up to 1. Such

normalization also helps to maintain the numerical precision of large-scale matrix multiplications.

The complete procedure of our proposed link analysis recommendation algorithm is summarized as follows:

Step 1. Construct the interaction matrix A and the associating matrix B based on the sales transaction data: $A = (a_{ij})$ and $B = (b_{ij})$, where $b_{ij} = \frac{a_{ij}}{(\sum_j a_{ij})^{\gamma}}$.

Step 2. Set the source consumer representativeness matrix CR^0 to be $\eta\, I_M$ and let it be the initial consumer representativeness matrix: $CR(0) = CR^0$.

Step 3. At each iteration t, perform the following:

Step 3.1. $PR(t) = A' \cdot CR(t-1)$;

Step 3.2. $CR(t) = B \cdot PR(t)$;

Step 3.3. Normalize $CR(t)$, such that $\sum_{i=1}^{M} cr_{ij} = 1$;

Step 3.4. $CR(t) = CR(t) + CR^0$.

Perform Step 3.1 to 3.4 until convergence or the specified number of iterations T is reached (T=5 is sufficient for the e-commerce datasets in our experiments).

3. Evaluation of Recommendation Algorithms

Several previous studies have been devoted to evaluating multiple recommendation algorithms [10, 21], but they mainly focused on variations of the user-based algorithms. Furthermore, newly proposed algorithms are typically only compared with the user-based algorithm. As a result, a comprehensive understanding of existing recommendation algorithms is far from complete.

In our study, we selected 20% most recent interactions of each consumer's interactions to form the testing set and designated the remaining 80% (earlier interactions) to be the training set. To understand the performance of the algorithms under sparse data, we also study recommendation performance with a *reduced training set* by randomly selecting from the training set (referred to as the *unreduced training set*) only 40% of the consumer's total interactions (or 50% of the interactions in the training set). The algorithms were set to generate a ranked list of recommendations of K products. For each

consumer, the recommendation quality was measured based on the number of *hits* (recommendations that matched the products in the testing set) and their positions in the ranked list. We adopted the following recommendation quality metrics from the literature regarding the relevance, coverage, and ranking quality of the ranked list recommendation (e.g., [10]):

(1) Precision: $P_c = \frac{\text{Number of } hits}{K}$,

(2) Recall: $R_c = \frac{\text{Number of } hits}{\text{Number of products consumer } c \text{ interacted with in the testing set}}$,

(3) F Measure: $F_c = \frac{2 \times P_c \times R_c}{P_c + R_c}$, and

(4) Rank Score: $RS_c = \sum_j \frac{q_{cj}}{2^{(j-1)/(h-1)}}$, where j is the index for the ranked list; h is the viewing *halflife* (the rank of the product on the list such that there is a 50% chance the user will purchase that product); $q_{cj} = \begin{cases} 1, & \text{if } j \text{ is in } c\text{'s testing set,} \\ 0, & \text{otherwise} \end{cases}$.

For precision, recall, and F measure, an average value over all consumers tested was adopted as the overall metric for the algorithm. For the rank score, an aggregated rank score RS for all consumers tested was derived as $RS = 100 \frac{\sum_c RS_c}{\sum_c RS_c^{\max}}$, where $RS_c^{\max}$ was the maximum achievable rank score for consumer c if all future purchases had been at the top of a ranked list. The precision, recall, and F measure are standard performance measures to estimate the relevance and coverage of the recommended items compared with the consumers' potential interests. The rank score measure was proposed in [10] and adopted in many follow-up studies (e.g., [12, 15, 21]) to evaluate the ranking quality of the recommendation list.

4. An Experimental Study and Observations

We used three e-commerce datasets in our experimental study: a retail dataset provided by a leading U.S. online clothing merchant, a book dataset provided by a Taiwan online bookstore, and a movie rating dataset provided by the *MovieLens* Project. The retail

dataset contained 3 months of transaction data with about 16 million transactions (household-product pairs) involving about 4 million households and 128,000 products. The book dataset contained 3 years of transactions of a sample of 2,000 customers. There were about 18,000 transactions and 9,700 books involved in this dataset. The movie dataset contained about 1 million ratings on about 6,000 movies given by 3,500 users over 3 years. For the movie rating dataset we treated a rating on product p by consumer c as a transaction ($a_{cp} = 1$) and ignored the actual rating. Such adaptation has been adopted in several recent studies such as [15]. Assuming that a user gives a rating to a movie based on her experience with the movie, we recommend only whether a consumer will watch a movie in the future and do not deal with the question of whether or not she will like it.

For our experiment, we used samples from these three datasets. We included consumers who had interacted with 5 to 100 products for meaningful testing of the recommendation algorithms. This range constraint resulted in 851 consumers for the book dataset. For comparison purposes, we sampled 1,000 consumers within this range from the retail and movie datasets for the experiment. The details about the final samples we used are shown in Table 1. The statistics of the complete datasets are also reported in Table 1 in the parentheses.

Dataset	# of Consumers	# of Products	# of Transactions	Density Level*	Avg. # of purchases per consumer	Avg. sales per product
Retail	1,000 (~4 million)	7,328 (~128,000)	9,332 (~16 million)	0.1273% (~0.0031%)	9.33 (~4)	1.27 (~125)
Book	851 (~2,000)	8,566 (~9,700)	13,902 (~18,000)	0.1907% (0.0928%)	16.34 (~9)	1.62 (~1.86)
Movie	1,000 (~3,500)	2,900 (~6,000)	50,748 (~1 million)	1.7499% (4.7619%)	50.75 (~166)	17.50 (~285.71)

Table 1. Characteristics of the datasets (* the *density level* of a dataset is defined as the percentage of the elements valued as 1 in the interaction matrix)

Following the evaluation procedure described above, we prepared an unreduced training set, a reduced training set, and a testing set for evaluation. Thus we had six experimental configurations for each of the seven algorithms (3 datasets by 2 training sets). We set the number of recommendations to be 10 ($K = 10$) and the halflife for the

rank score to be 2 ($h = 2$). In addition to the six collaborative filtering algorithms, we also included a simple approach (referred to as the "Top-N Most Popular" or "Top-N" algorithm) that recommends to a consumer the top 10 most popular unseen products as a comparison benchmark.

Based on existing literature and our understanding of the algorithms, we expect to have the following findings: (1) Most algorithms should generally achieve better performance with the unreduced (dense) dataset; (2) Algorithms that were specifically designed for alleviating the sparsity problem should generally outperform the standard correlation/similarity-based algorithms and the "Top-N" algorithm, especially for the reduced (sparse) datasets; (3) The link analysis algorithm, with the global link structure taken into consideration and a flexible control on penalizing frequent consumers and products, is hypothesized to generally outperform other collaborative filtering algorithms.

Measure	Algorithm	Dataset						Avg. algorithm rank
		Retail		Book		Movie		
		Reduced	Unreduced	Reduced	Unreduced	Reduced	Unreduced	
Precision	User-based	0.0028	0.0042	0.0048	0.0122	0.0399	0.0516	6.00
	Item-based	0.0051	0.0106	0.0068	0.0093	**0.0462**	**0.0759**	4.00
	Dimensionality Reduction	0.0050	0.0064	0.0097	0.0191	0.0384	0.0530	5.17
	Generative Model	0.0056	0.0070	0.0226	**0.0251**	0.0388	0.0444	4.17
	Spreading Activation	0.0063	**0.0130**	0.0201	0.0231	0.0437	0.0607	**3.17**
	Link Analysis	**0.0081**	**0.0133**	0.0218	**0.0267**	**0.0471**	0.0624	**1.50**
	Top-N Most Popular	0.0069	0.0062	**0.0268**	**0.0258**	0.0373	0.0452	4.00
Recall	User-based	0.0163	0.0305	0.0351	0.0753	0.0446	0.0576	5.83
	Item-based	0.0359	0.0731	0.0268	0.0443	**0.0540**	**0.0864**	4.00
	Dimensionality Reduction	0.0411	0.0408	0.0564	0.1026	0.0428	0.0580	4.67
	Generative Model	0.0329	0.0466	0.1337	**0.1273**	0.0412	0.0426	4.67
	Spreading Activation	0.0531	**0.0863**	0.1070	0.1155	0.0457	0.0618	**3.00**
	Link Analysis	**0.0703**	**0.0891**	0.1212	**0.1282**	**0.0510**	0.0649	**1.83**
	Top-N Most Popular	0.0429	0.0326	**0.1553**	**0.1316**	0.0377	0.0440	4.00
F	User-based	0.0046	0.0073	0.0083	0.0202	0.0401	0.0518	6.00
	Item-based	0.0088	0.0182	0.0091	0.0144	**0.0475**	**0.0769**	3.83
	Dimensionality Reduction	0.0088	0.0109	0.0157	0.0305	0.0386	0.0528	5.00
	Generative Model	0.0093	0.0118	0.0366	**0.0393**	0.0380	0.0415	4.33
	Spreading Activation	0.0111	**0.0219**	0.0320	0.0362	0.0426	0.0583	**3.17**
	Link Analysis	**0.0144**	**0.0224**	0.0349	**0.0415**	**0.0466**	0.0605	**1.67**
	Top-N Most Popular	0.0113	0.0100	**0.0431**	**0.0405**	0.0357	0.0425	4.00
Rank Score	User-based	1.4323	2.5770	1.8164	4.9332	3.7750	5.3500	5.67
	Item-based	2.0313	4.9866	1.3172	3.2146	4.1667	**8.7750**	4.33
	Dimensionality Reduction	3.4896	3.0120	3.0227	6.9486	4.1000	6.8000	4.33
	Generative Model	0.7552	2.1084	**11.9800**	**11.0287**	4.2000	4.0500	4.33
	Spreading Activation	3.7500	**5.2209**	8.6800	9.4955	**4.7500**	7.4000	**2.83**
	Link Analysis	**4.4035**	**6.4074**	9.9902	10.3835	**5.3387**	7.4319	**1.83**
	Top-N Most Popular	2.0052	1.3889	**12.8397**	**10.7814**	3.7000	5.0750	4.67
# of target consumers		320	498	601	674	1000	1000	
Avg. consumer purchase		9.33		16.34		50.75		

Table 2. Experimental results: actual performance measures (boldfaced measures were not significantly different from the highest measure in each configuration at 5% level)

We report the recommendation quality measures in Table 2. The boldfaced measures correspond to the algorithms that were not significantly different from the highest measure in each configuration at the 5% significance level. To provide a summary of each algorithm's overall performance across different datasets, we reported the average rank of each algorithm with respect to the four measures [22]. For example, for the precision measure the link analysis algorithm's average rank is 1.50, which corresponds to the average of its ranks for individual datasets (1, 1, 3, 1, 2, and 2). Boldfaced average ranks are the top 2 average ranks. As collaborative filtering algorithms can only recommend products to the consumers that appeared in the training transactions, for consumers with no future purchases in the testing set appeared in the training set, no successful recommendation is possible. To make the performance measures more meaningful, we only evaluate recommendations for target consumers for whom successful recommendations are possible. Therefore, for the same dataset the reduced and unreduced training sets resulted in different numbers of target consumers. These numbers of target consumers are also reported in Table 2.

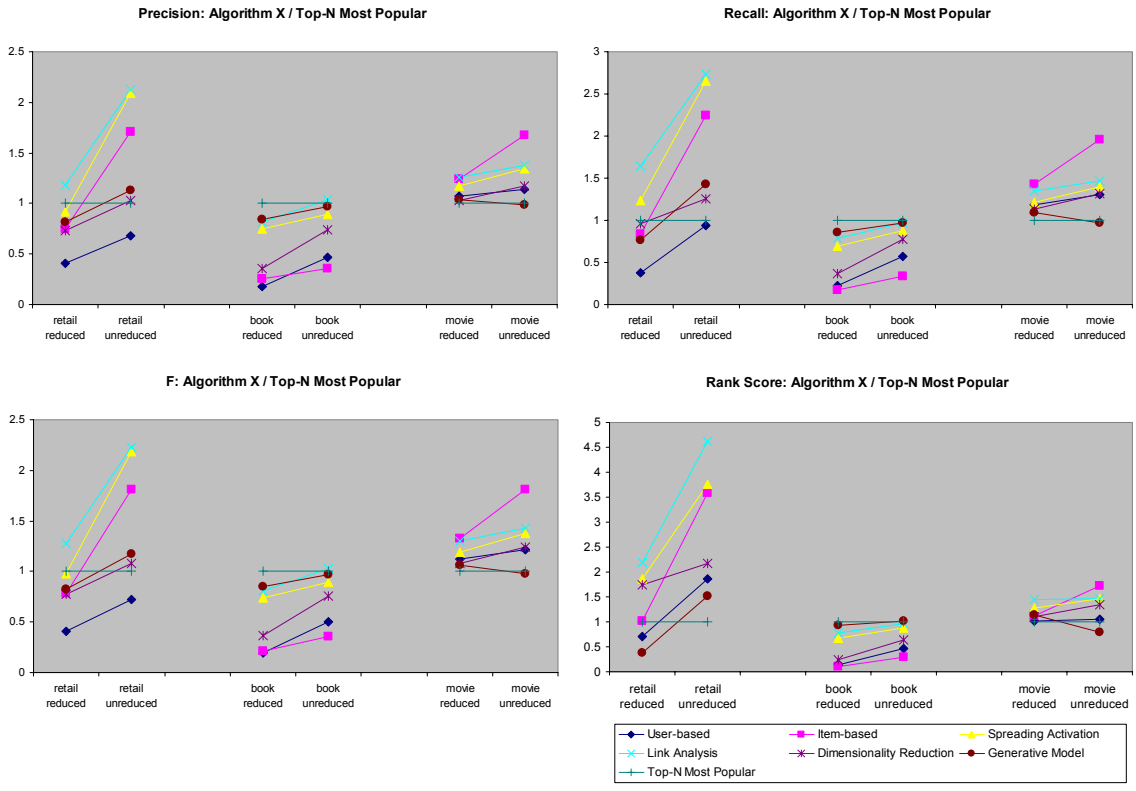

Figure 1. Experimental results: relative performance measures

For easy interpretation of the results, we present in Figure 1 the relative performances of the individual algorithms compared to the "Top-N" algorithm. For example, the link analysis algorithm's value in the precision diagram for the unreduced retail dataset was 2.13, meaning its precision was 113% higher than that achieved by the "Top-N" algorithm.

Based on these results we report the following observations.

- All algorithms achieved better performance with the unreduced data. The performance measures shown in Table 2 were generally larger with the unreduced datasets than with the reduced datasets. The difference is even more significant when the numbers of target consumers are taken into account. For example, the average precision measure of 0.81% for 320 target consumers under the reduced retail dataset should be adjusted to 0.52% (0.81%×320/498) when compared with the average precision of 1.33% for 498 target consumers under the unreduced dataset. The general upward line pattern in Figure 1 visually demonstrates this finding. There were 3 exceptions in the total of 84 data-algorithm-measure configurations: the recall and rank score measures of the Top-N algorithm under the reduced book dataset were slightly higher than their counterparts under the unreduced book dataset after target consumer adjustment; the rank score of the generative model algorithm under the reduced movie dataset was slightly higher than its counterpart under the unreduced movie dataset after target consumer adjustment.
- The link analysis algorithm generally achieved the best performance across all configurations except for the movie dataset. Table 2 shows that the link analysis algorithm achieved the highest average ranks for the precision, recall, F, and rank score measures (1.5, 1.83, 1.67, and 1.83). The spreading activation algorithms achieved the second highest average ranks (3.17, 3, 3.17, and 2.83). The average ranks for all other algorithms were between 4 and 6. This result clearly shows the quality advantage of the link analysis algorithm over the other algorithms. The good performance of both the link analysis and spreading activation algorithms also shows that additional valuable information (such as transitive associations) in sales transactions can be exploited by graph-based algorithms. The link analysis

algorithm's dominance over other algorithms was most evident with the unreduced retail datasets. It achieved about 150% higher precision, recall, and F measures and a more than 350% higher rank score than the Top-N algorithm.

- Most other algorithms showed mixed performances under different datasets. The item-based algorithm performed exceptionally well for the unreduced movie dataset, but had relatively lower quality with the retail datasets and the worst performance with the book dataset. The good performance of the item-based algorithm with the unreduced movie dataset may be associated with the dataset's much higher transaction density (1.75%) and average sales per product (17.50) than other datasets. The generative model algorithm achieved relatively good performance with the book datasets but had the worst performance with the unreduced movie dataset. This divergent trend from the performance with the item-based algorithm may also be associated with the transaction density level and the average sales per product. The dimensionality reduction algorithm consistently achieved a mediocre performance across all configurations and the user-based algorithm was almost always dominated by other algorithms.
- An interesting result from our experiment is that the Top-N recommendations were not necessarily a bad choice for many configurations, especially for the book dataset. On the other hand, in-depth analysis of the recommendations showed that a large portion of the collaborative filtering recommendations were different from those given by the Top-N algorithm. Collaborative filtering recommendations therefore may still provide value to consumers in addition to the simple popularity-based recommendations.

5. Conclusion

A unique contribution of our study is a comprehensive evaluation of a wide range of algorithms using e-commerce datasets. No single algorithm was observed to dominate all other algorithms across all configurations, while the link analysis algorithm proposed in this article achieved the overall best performances. Although the sparsity level of a consumer-product interaction data had a general effect on recommendation quality it

could not explain all quality variations of different algorithms. Additional descriptors of the interaction data are needed to prescribe the most appropriate recommendation algorithm for a particular application. We expect that these additional descriptors should include row/column density of the interaction matrix and measures from graph/network modeling literature, such as node degree distribution, average path length, and cluster coefficient [23]. Our ongoing research is exploring these descriptors as part of the meta-level recommendation framework for recommendation algorithms.

6. Acknowledgement

This research is partly supported by an NSF Digital Library Initiative-II grant, "High-performance Digital Library Systems: From Information Retrieval to Knowledge Management," IIS-9817473, April 1999 – March 2002, and by an NSF Information Technology Research grant, "Developing a Collaborative Information and Knowledge Management Infrastructure," IIS-0114011, September 2001 – August 2005.

7. References

[1] J. Schafer, J. Konstan, and J. Riedl, "E-commerce recommendation applications," *Data Mining and Knowledge Discovery*, vol. 5, pp. 115-153, 2001.

[2] M. Balabanovic and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, pp. 66-72, 1997.

[3] M. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, pp. 393-408, 1999.

[4] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lavrence, "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments.," presented at 17'th Conference on Uncertainty in Artificial Intelligence (UAI 2001). 2001.

[5] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," presented at ACM Conference on Computer-Supported Cooperative Work, 1994.

[6] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices In a virtual community of use," presented at ACM Conference on Human Factors in Computing Systems, CHI'95, 1995.

[7] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," presented at Human Factors in Computing Systems, 1995.

[8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," presented at ACM CSCW'94 Conference on Computer-Supported Cooperative Work, 1994.

[9] P. Resnick and H. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, pp. 56-58, 1997.
[10] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," presented at Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, 1998.
[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender systems: a case study," presented at WebKDD Workshop at the ACM SIGKKD, 2000.
[12] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, pp. 116-142, 2004.
[13] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information Systems*, vol. 22, pp. 89-115, 2004.
[14] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Item-based collaborative filtering recommendation algorithms," presented at Tenth International World Wide Web Conference, 2001.
[15] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Transactions on Information Systems*, vol. 22, pp. 143-177, 2004.
[16] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, pp. 133-151, 2001.
[17] L. H. Ungar and D. P. Foster, "A formal statistical approach to collaborative filtering," presented at CONALD'98, 1998.
[18] J. Kleinberg, "Authoritative sources in a hyperlinked environment," presented at ACM-SIAM Symposium on Discrete Algorithms, 1998.
[19] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," presented at 7th International World Wide Web Conference, Brisbane, Australia, 1998.
[20] J. R. Anderson, "A spreading activation theory of memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 22, pp. 261-295, 1983.
[21] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, pp. 5-53, 2004.
[22] P. Domingos and M. Pazzani, "Beyond independence: Conditions for the optimality of the simple Bayesian classifier," presented at 13th International Conference on Machine Learning, Bari, Italy, 1996.
[23] R. Albert and A.-L. Barabasi, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, pp. 47-97, 2002.