Is it all About Connections? Factors Affecting the Performance of a Link-Based Recommender System

Miles Efron and Gary Geisler
{efrom, geisg}@ils.unc.edu
School of Information and Library Science
University of North Carolina, Chapel Hill *

August 23, 2001

Abstract

This study reports on a recent evaluation of the similarity model used by Recommendation Explorer, an automatic recommender system. In particular, we consider the role of several system-internal factors in determining the quality of recommendation. More generally, we discuss factors in the recommendation task itself that complicate the construction and evaluation of recommender systems, and reflect on the implications of our findings for research in this area.

1 Introduction

Evaluating information retrieval systems is a notoriously steep challenge. The subjectivity of such crucial variables as the relevance and quality of information complicates the matter of evaluation. Automatic recommender systems are similar to IR systems in many ways [10, 11]. Among these similarities, the problem of evaluation is particularly vexing [14]. In this paper we focus on the evaluation problem by analyzing the factors that affect the performance of Recommendation Explorer, an automatic recommender system.

Borrowing from bibliometrics and hypertext analysis, Recommendation Explorer (RecEx) operates on a square matrix that describes the paths of inter-recommendation between database items. Due to the conceptual similarity between recommendation, citation, and hypertextual linking, we refer to RecEx as a "link-based" system. *Link* here suggests the type of relationship found in bibliometric or hypertextual structures. Thus our use of the term differs somewhat from the graph-theoretic sense used in other research [1, 16].

RecEx applies the technique of singular value decomposition (SVD) to the square item-item recommendation matrix. SVD allows evocative high-order semantic patterns to inform the system's similarity model. In previous work Sarwar *et al.* [13, 12] used SVD to obtain robust recommendations from sparse correlational data. Our work differs from theirs insofar as we use SVD to analyze inter-item relationships, not relations between users.

In a previous study [7] this method of recommendation yielded promising results. The present study thus focuses on problems of interpreting the results of conventional evaluation methods. In particular, we analyze system performance in close detail to discover not only how well our system works, but when it works well and when it fails. We detail several aspects of the recommendation task that bear on the quality of our system's performance, reflecting on their implications for system design and evaluation.

*This research was funded in part by the School of Information and Library Science at the University of North Carolina, and by Ibiblio.org

Figure 1: User-defined interest profiles enable quick, personalized recommendations

2 Recommendation Explorer

Recommendation Explorer (RecEx) is an experimental recommender system under development at the School of Information and Library Science at The University of North Carolina, Chapel Hill. In its current implementation RecEx uses a database of 12,726 popular film titles. Each film in the database is represented by a metadata record that contains a plot summary, production information, and a list of other films that human editors have recommended for those who like the film.

Like all recommender systems, RecEx faces a steep challenge: to accommodate the multidimensional and dynamic nature of recommendation. The demands that each user puts on a recommender system are highly specific and subject to change. Accounting for such complexity in user motivations, expectations, and context is difficult, but critical for a system that provides personalized recommendations. RecEx addresses this difficulty by tightly coupling the system's interface with algorithms that derive a powerful model of inter-item similarity.

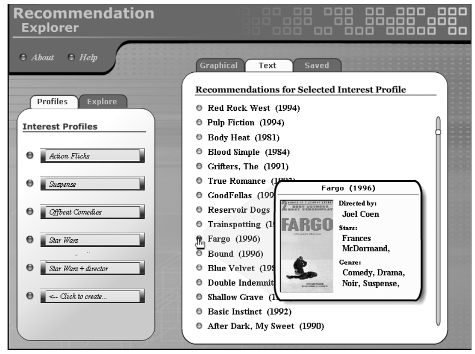

Figure 2: Interface for viewing and updating recommendations

Suppose that user A likes *The Usual Suspects* because of its stars, while user B likes it because of its style. The interface to RecEx allows each user to identify these axes of his or her information need. Using stored, named profiles (Figure 1), the user specifies items that he likes and the item attributes that are important to him. After recommendations are generated from the profile parameters, the user can view and explore the results, previewing and saving items and developing an understanding of the results (Figure 2). This understanding might lead the user to adjust the expression of his information need.

This type of interaction depends upon a robust similarity model. That is, if user A liked *The Usual Suspects* because of its actors, the system needs a way to find actors *similar* to those in *The Usual Suspects*. A model of film-film similarity is important because it allows us to define similar actors, directors, etc: e.g. similar actors act in similar films.

The remainder of this paper describes the construction and evaluation of an experimental module of RecEx. This module implements an inter-item similarity model based on the singular value decomposition of the film-film recommendation matrix. The similarity module derives a mapping of the system's recommendation space, collocating "similar films." We devote particular attention to the problems of evaluating such a module.

3 The RecEx Similarity Model

The database behind RecEx derives from the recommendation service of reel.com, a database of movie information on the Internet. For each film in the database, we record two sets of information: 0 or more close recommendations and 0 or more creative recommendations. A close recommendation from film A to film B implies an obvious link between the two, such as a common director or subject matter. A creative recommendation describes a more tenuous relationship. All films in the database have at least one of the following: close recommendation, creative recommendation, incoming recommendation (i.e. recommended by another film). On average, each film contains 4.035 recommendations, 2.294 close and 1.741 creative.

These recommendations were compiled by human editors of the reel.com database. The proposed method attempts to exploit the expertise of these editors to the greatest extent possible. However, other types of data could inform a system such as we describe. In an e-commerce setting, an item-item matrix $\mathbf{A}$ could be constructed wherein each cell a_{ij} contains a count of the number of times item i was purchased in the same order as item j. A digital library might create a matrix from the citation or hyperlink patterns among documents.

To define item-item similarity we begin with the square film-film recommendation matrix $\mathbf{A}$ where each cell a_{ij} records the type of recommendation that the *ith* film makes regarding the *jth* film. If film j is a close recommendation for i, $a_{ij} = 2$. If film m is a creative recommendation for i, $a_{im} = 1$. We also set cells on the main diagonal equal to 5. Thus $a_{ii} = 5$ (these values were chosen because they led to good performance).

In its original implementation RecEx worked directly on this matrix. For a given seed film i, the system returned N recommendations by recursively following the links from i supplied by the reel.com editors. We refer to this method as the use of the "raw" link structure. To generate robust similarity judgements, the new method transforms the film-film matrix $\mathbf{A}$ through application of the singular value decomposition (SVD).

3.1 Singular Value Decomposition: Motivation

Used widely in information retrieval (where it goes by the name latent semantic indexing, or LSI) [3, 2] SVD is a least-squares dimensionality reduction technique. A type of factor analysis, SVD is closely related to principal components analysis and multidimensional scaling. The goal of LSI is to represent items along axes that manifest the "latent semantic structure" of a matrix. To accomplish this LSI uses SVD to project a matrix $\mathbf{A}$ of rank r onto a space of k dimensions, where $k \ll r$. The resulting k-dimensional matrix $\mathbf{A}_k$ is the closest rank-k approximation of $\mathbf{A}$, in the least squares sense. Its proponents argue that this projection into k-space reduces noise in the matrix $\mathbf{A}$.

In information retrieval, use of LSI is motivated by the suspicion that lexical features provide noisy evidence about document relationships. While lexical ambiguity is not an issue in our film-film recommendation matrix, SVD is still valuable for RecEx. This is due to SVD's analysis of high-order relationships between matrix elements. If film i recommends film j, and j recommends m, our system will recognize a transitive affinity between i and m.

3.2 Singular Value Decomposition: Mathematics

To compute the singular value decomposition[1], we begin with the film-film matrix $\mathbf{A}$, described above (in our case $\mathbf{A}$ is square, but it need not be). During the SVD, our $n \times n$ matrix $\mathbf{A}$ of rank r is factored into the product of three special matrices (Formula 1).

$$\mathbf{A} = \mathbf{T}\mathbf{\Sigma}\mathbf{D}' \tag{1}$$

Matrices $\mathbf{T}$ and $\mathbf{D}$ are orthonormal: $\mathbf{T}'\mathbf{T} = \mathbf{D}'\mathbf{D} = \mathbf{I}_n$ and the columns of $\mathbf{T}$ and $\mathbf{D}$ are of unit length. $\mathbf{T}$ and $\mathbf{D}$ comprise the left and right singular vectors of $\mathbf{A}$, respectively. The $r \times r$ diagonal Matrix $\mathbf{\Sigma}$ contains the singular values of $\mathbf{A}$ in descending order on the main diagonal. The

[1] A full description of SVD is beyond the scope of this study. For a more detailed treatment see [9]

singular values are the positive square roots of the eigenvalues of $\mathbf{A}'\mathbf{A}$ and $\mathbf{A}\mathbf{A}'$. Thus the ith singular value indicates how much of the input matrix's variance is described by the ith axis of factor space.

Matrix $\mathbf{T}$ represents the rows of the original matrix $\mathbf{A}$. Thus the ith column of $\mathbf{T}$ describes the ith film as a vector in factor space. $\mathbf{D}$ represents the columns. In the case of the Recommendation Explorer, $\mathbf{T}$ and $\mathbf{D}$ are equivalent.

The dimensionality reduction in LSA comes about by truncating the matrix $\mathbf{\Sigma}$ and then recombining it with the matrices $\mathbf{T}$ and $\mathbf{D}$. Because SVD by definition will find r factors for matrix $\mathbf{A}$ where $rank(\mathbf{A}) = r$, as we approach the rth factor, the amount of variance described by each axis will be very small. Because the last singular values are small, we suspect that they represent noise, that they describe random variance. By choosing a dimensionality k, setting all singular values i for $i > k$ equal to 0, and amending $\mathbf{T}$ and $\mathbf{D}$ accordingly, by matrix multiplication we project $\mathbf{A}$ onto the best $k-$dimensional space, in the least-squares sense.

3.3 Implementation

To compute the SVD of our film-film matrix, we used SVDPACKC [8], a suite of programs for solving eigensystems of sparse matrices. After computation, we project the left singular vectors into $k-$space. Similarity judgements are performed on the matrix $\hat{\mathbf{T}} = \mathbf{T_k}\mathbf{\Sigma_k}$, where $\mathbf{T_k}$ contains the first k rows of $\mathbf{T}$ and $\mathbf{\Sigma_k}$ is the $k \times$k matrix defined by the first k rows and columns of $\mathbf{\Sigma}$.

The matter of choosing an optimal k value is an open question in the LSA research [4, 15]. Common practice in IR applications indicates a dimensionality between 50 and 300. After some trial and error, we selected $k = 50$.

$$cos(\overrightarrow{u}, \overrightarrow{v}) = \frac{\overrightarrow{u} \cdot \overrightarrow{v}}{||\overrightarrow{u}||\,||\overrightarrow{v}||} \tag{2}$$

Similarity between two films v and u is thus defined as the cosine (Formula 2) of each film's vector in the k space defined by $\hat{\mathbf{T}}$.

4 Experimental Evaluation

To gauge the effectiveness of SVD for our application we conducted an experiment to compare the performance of the SVD module against performance based on the raw link structure defined in matrix $\mathbf{A}$.

4.1 Methodology

Evaluation was conducted using 10 films, listed in Table 1. These "seed" films were chosen to represent a variety of film genres and audience types.

Evaluation occurred in two phases. In the first phase, volunteer reviewers defined a generous list of recommendations for each seed film. Six reviewers were chosen from a sample of convenience, based on their self-identified interest in popular films. Each reviewer chose the 5 seed films on which he felt most competent to make recommendations. For each chosen seed, each reviewer created a list of approximately 30 candidate recommendations. These lists were then combined to create a "recommendation space" for each seed. The largest pooled space was for *Austin Powers*, with 75 members. The smallest was *The Terminator*'s 37. The mean recommendation space was 51.9 titles; the median was 51.

To help them compile their 30-film list for each seed, reviewers used three online resources: The Internet Movie Database (http://www.imdb.com), The Movie Critic (http://www.moviecritic.com), and The Sepia Video Guide (http://vguide.sepia.com). IMDB provides recommendations that derive from several sources: user suggestions, IMDB editor picks, and an undisclosed automatic system. The Movie Critic uses the LikeMinds collaborative filtering system. Sepia is a simple online reference work that contains information about films, but does not make explicit recommendations. In addition, reviewers were permitted to add any titles to the list not found in the online systems.

In the second evaluation phase, 131 new reviewers picked approximately 15 recommendations for each of several seed films. On average these 131 reviewers made recommendations for 3.9 seeds. *Fargo* received the most reviews, 68. *Spanish Prisoner* received the fewest, with 19. The average number of reviews per

Title	Raw	SVD
alien	0.287	0.435
austin powers	0.240	0.431
english patient	0.048	0.061
fargo	0.247	0.373
full monty	0.192	0.048
room view	0.256	0.456
sleepless	0.122	0.185
spanish prisoner	0.233	0.265
star wars	0.471	0.364
terminator	0.484	0.403

Table 1: Average Precision of recommendations based on raw link structure and links analyzed by SVD

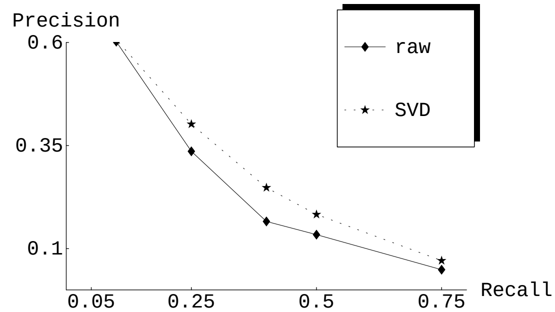

Figure 3: Average precision/recall for the raw data and data transformed by SVD

seed was 51.4, with a median of 54.

Using an online form, each reviewer consulted the pooled recommendation space for each of his selected seeds, marking all those films that he had seen and the 10-15 best recommendations for fans of the seed.

Finally, these reviews were pooled into a "key" for each seed—a list of films that constitute good recommendations for a given seed. The key contains all candidate films selected by any reviewer. Using our key as a point of reference, we evaluate recommendation performance in terms of precision and recall. Precision is defined as the ratio of the number of returned key-members to total films retrieved (percent of returned items that are relevant). Recall is the ratio of the number of key-members retrieved to the total number total key members (percent of all relevant items returned).

4.2 Results

4.2.1 Quality of SVD recommendations

Table 1 compares the two methods, raw and SVD, using average precision at five levels of recall (.1, .25, .4, .50, .75). The SVD method provides better average precision for seven out of the ten seed films. Figure 3 plots precision against recall for each method, with each recall/precision point averaged across all ten seeds. SVD appears to offer the most improvement over the raw link structure at middling levels of recall. This result is somewhat surprising, as we anticipated that SVD would yield greater performance improvements for high-recall searches. A partial explanation for the tapering off of SVD's benefit at high recall may lie in the nature of the task imposed by this experiment. Since candidates are only counted as "relevant" if they appear in the user-defined key, we may be penalizing recommendations that would be perfectly viable to a user, but that were simply not included in the seed key. This amounts to a problem of defining the set of all "relevant" candidates for a given seed, a problem to which we return in the following section.

4.2.2 Dimensionality of the Similarity Model

To gauge how the choice of k, the dimensionality of the reduced space, bears on recommendation quality, we generated spaces of varying dimensionality and measured precision/recall using each space.

Figure 4 charts a dramatic improvement in precision performance as we increase k from 45 to 50. As k increases from 50 to 100, performance degrades slightly. Spaces of dimensionality much lower than 45 appear insufficiently informative for the recommendation task, while the 100-dimensional space is slightly less effective than the 50-dimensional space.

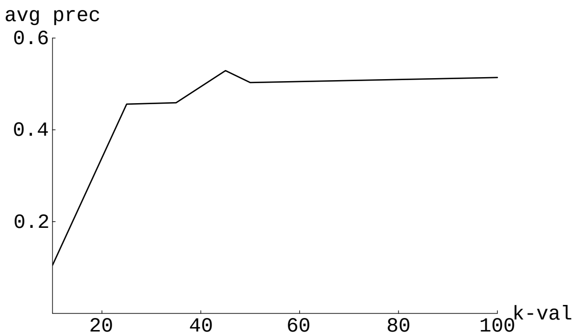

Figure 4: Average precision in terms of the dimensionality of the SVD space.

4.3 Factors Affecting Recommendation Quality

Having analyzed our data, it is clear that the similarity model matches the tastes of our volunteer reviewers better for some seed films than others. For example, average precision for *The English Patient* was extremely low, while the similarity model delivered fairly good precision for *A Room with a View*. This section pursues an analysis of which factors bear on this discrepancy in performance.

A series of linear models were constructed to measure the effects of several independent variables on system performance. In each case, performance for a given seed is modeled as a linear combination of factors. The goal of this modeling is to gain an appreciation for why certain seeds led to decreased or increased performance in our experiment. In particular we were eager to discover how much variability in performance derives from defects in our similarity model, and how much is due to other causes.

Because the SVD-based similarity model depends on the link structure defined by patterns of inter-item recommendation we suspected that performance would be affected by the degree to which a given seed film is linked to other films. To assess this effect we defined for each film in the database two data–its in-degree and out-degree. The in-degree for film A counts the number of films that recommend A. Likewise, A's out-degree describes the number of films that A recommends. Not surprisingly most popular films have a higher in-degree than out-degree. For example 35 films recommend *Alien*, while *Alien* recommends 9. For the sake of comparison, we included as a seed film *The Spanish Prisoner*, with in-degree of only 1 and out-degree of 9.

Formula 3 describes the linear model that was constructed to gauge the relation between performance and a seed film's degree of connection to the database recommendation structure:

$$y_i = \beta_0 + \beta_1(in_i) + \beta_2(out_i) + \beta_3(out/in_i) + \epsilon_i \quad (3)$$

where y_i=average precision for seed i. To our surprise, however, this model fit the data very poorly, yielding R^2=0.119. The model describes less than 12% of the variance among seed film performance. The p-value for the null hypothesis H_0 : *no linear relation adheres among the variables* was 0.845. We supplemented this analysis with a second model, identical to the prior model with the exception that in this case we replaced y_i with precision at 25% recall, rather than average recall. The second model also suffered from a poor fit, with R^2=0.182 and p-value=0.73.

These findings indicate that in-degree and out-degree bear very little on the performance for the observed data. Instead, we suspected that variations in precision might derive from qualities native to the seed films themselves. Perhaps certain films do not lend themselves to comparison by our similarity model. For such films, the idea of 100% recall might not be meaningful. In other words, perhaps certain seeds are simply harder to provide general recommendations for than others. To help measure the "difficulty" of recommending for a given seed, we consider three variables. *Reviewers* counts the number of participants who reviewed a given seed A. Seed film A's *total* records the total number of reviews generated for A during the experiment. Thus *total* divided by *reviewers* yields the average number of films reviewed by a single participant for seed A. Finally, *unique* measures the number of distinct candidate films recommended for A.

Using these variables we constructed another model, shown in formula 4:

$$y_i = \beta_0+\beta_1(reviewers_i)+\beta_2(total_i)+\beta_3(unique_i)+\epsilon_i \quad (4)$$

where y_i=average precision for seed i. This model yielded a much better fit than the previous models, with R^2=0.64. The p-value for H_0 was 0.087. Using y_i=precision at 25% yielded R^2=0.742, with p-value=0.034. Encouraged by the fit of these models, we plotted the residuals against the predicted values for each. Data in both plots appeared to be distributed randomly, suggesting that little or no systematic variance is omitted by the new models.

4.4 Assessing the variability of Observed Statistics

An obvious limitation of the modeling described here lies in the small number of seeds in the study. Due to our low N, a high degree of variability must be expected in trying to generalize our results. To gain a sense of this variability we conducted a round of bootstrap resampling. Bootstrapping (described in [5, 6]) is a computer-intensive method for assessing the variability of a statistic S. The process involves repeated resampling from the original data, and concomitant re-computation of the statistic of interest. For a data set D of size N, we create M bootstrap samples D^* by sampling N observations from D, with replacement. For each bootstrap sample D^*, we compute our statistic S^*. By repeating this process M times, where M is large, we can estimate the stability of S by noting the variability of S^*.

For this application we set M=1000, thus creating 1000 bootstrap samples from our original data. For each of these 1000 bootstrap samples, we computed R^{2*}. We then calculated the standard deviation of R^{2*}. For the model where y_i=average precision, mean R^{2*}=0.793, and $s(R^{2*})$=0.161. With y_i=precision at 25% recall, mean R^{2*}=0.633, and $s(R^{2*})$=0.305. Although our observed R^2 is quite variable we can still glean useful information from our models. If we take $s(R^{2*})$=0.161 to be an estimate of the standard deviation of the true R^2, it is highly unlikely that the true value of R^2 is less than 0.632 (i.e. two standard deviations away from the observed $\hat{R^2}$). Thus it seems that variables *reviewers*, *total*, and *unique* do account for a sizable portion of the performance variation among seed films.

A potential weakness of the link-based similarity model used by RecEx is that weak patterns of co-recommendation in the data might limit performance. The relationship between performance and seed connection measures reported above, however, indicate that this is not the case. Our method appears to discern film-film similarity regardless of the in- and out-degree of seed films.

A strong correlation was observed between system performance and measures of reviewer agreement. This finding suggests that in some cases a similarity model based on static, high-level relationships between films cannot suitably address the complexity of individual users' tastes. In the case of *The English Patient*, for instance, many of the reviewers' top candidates were films based on romantic novels such as *Sense and Sensibility*, *Howard's End*, and *Emma*. On the other hand, other *English Patient* reviewers selected films with Ralph Fiennes (*The English Patient's* co-star) as an actor. Finally, many recommendations for *The English Patient* were period pieces. *The English Patient* describes events that occurred during World War II and before. Thus many reviewers recommended films made during the 1940s, or films that depict that period. On the other hand, the similarity model was unable to pursue these alternative types of interest in the film, recommending an array of "epic" stories, linked more by the scope of their narrative than any single, concrete factor. SVD recommendations for *The English Patient* are general in the face of a recommendation space that is highly specialized and disjointed.

Our results indicate that even a strong model of item-item similarity will have difficulty satisfying information needs as diverse as those evidenced by *The English Patient's* reviewers. To improve recommendation quality we propose supplementing static similarity judgements with user-specific information that "fine-tunes" the input to the recommendation system. Such tuning might include information concerning aspects of seed films that are important to the user at query time.

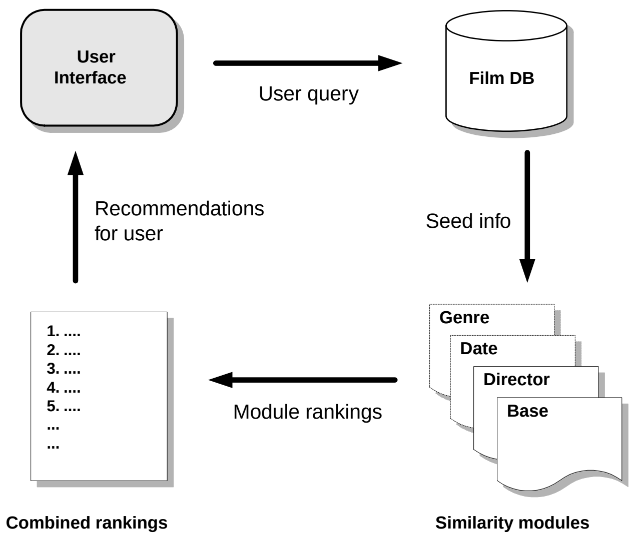

Figure 5: Modular Architecture of RecEx System

As an example, RecEx enables users to specify not only which movies they like, but also why they like them. A user might request movies similar to *The English Patient*, for instance, but also indicate that director and date are important to him. These supplementary similarity judgments are made by separate modules of the system (Figure 5). The director module, for instance, defines similarity between directors in terms of film-film similarity: similar directors make similar movies. This modular approach capitalizes on the film-film similarity model in order to deliver personalized recommendations with minimal user effort.

5 Conclusion

Recommendation Explorer's model of item-item similarity uses the Singular Value Decomposition to discover high-order relationships between database items. Based on patterns of co-recommendation among films, a system based on this similarity model has yielded promising results for automatic recommendation. A potential defect of such a link-based model (poor recommendation for weakly connected items) was not observed in the study reported here. However, this study suggests that even a robust similarity model can be supplemented by data specific to a given user at a given time. A detailed analysis of system performance suggests that human motivations in recommendation are diverse and highly specific. We anticipate improved performance after supplementing the static film matching module described here with modules that permit more nuanced articulations of information needs. Involving the more actively in the recommendation process, we propose, will improve the utility of our model of inter-item similarity. However, the diversity of reviewer opinion found in this study calls into question methods of evaluating such complex systems. Thus the matter of recommender evaluation itself remains a matter to be evaluated.

References

[1] Charu Aggarwal, Joel Wolf, Kun-Lung Wu, and Philip Yu. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In Surajit Chaudhuri and David Madigan, editors, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 201–212, N.Y., August 15–18 1999. ACM Press.

[2] M. W. Berry, Susan T. Dumais, and G. W. OBrien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. of the American Society for Information Science*, 41(6):391–407, 1990.

[4] C. H. Q. Ding. A similarity-based probability model for latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[5] B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.

[6] B. Efron. *An Introduction to the Bootstrap.* Chapman and Hall, 1993.

[7] M. Efron and G. Geisler. Using dimensionality reduction to improve similarity judgements for recommendation. In *Proceedings of the Second DELOS Network of Excellence Workshop on 'Personalisation and Recommender Systems in Digital Libraries'*, number No. 01/W03 in ERCIM Workshop Proceedings, 2001. http://www.ercim.org/publication/ws-proceedings/DelNoe02/index.html.

[8] M. W. Berry *et al.* Svdpackc (version 1.0) user's guide, university of tennesse tech. report cs-93-194. Technical report, University of Tennessee, 1993 (Revised October 1996). Available at http://www.netlib.org/svdpack/index.html.

[9] G. H. Golub and C. F. van Loan. *Matrix Computations.* Baltimore, The Johns Hopkins University Press, 1989.

[10] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[11] D. Riecken. Introduction: personalized views of personalization. *Communications of the ACM*, 43(8):26–28, 2000.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommender algorithms for e-commerce. In *ACM E-Commerce 2000 Conference*, October 2000.

[13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems–a case study. In *ACM WebKDD Workshop*, 2000.

[14] A. F. Smeaton and J. Callan. Joint delos-nsf workshop on personalisation and recommender systems in digital libraries. In *ACM SIGIR Forum.* Association for Computing Machinery, 2001. Forthcoming.

[15] R. E. Story. An explanation of the effectiveness of latent semantic indexing by means of a bayesian regression model. *Information Processing and Management*, 32(3):329–344, 1996.

[16] L. Terveen and B. Amento. Constructing, organizing, and visualizing collections of topically related web resources. *ACM Transactions on Human-Computer Interaction*, 6(1):67–94, 1999.