Using Dimensionality Reduction to Improve Similarity Judgements for Recommendation

Miles Efron and Gary Geisler
{efrom, geisg}@ils.unc.edu

May 28, 2001

Abstract

Recommendation Explorer is an experimental recommender system that attempts to address the provisional and contextual nature of user information needs by coupling the system's interface and recommendation algorithms. This study reports on the development and evaluation of a new similarity module for RecEx. Based on dimensionality reduction via the Singular Value Decomposition (SVD), the new module discovers high-order relationships among database items, thus obtaining a robust model of item-item similarity.

1 Introduction

To help users manage large volumes of data, developers of online information resources have begun turning to personalization and recommendation systems. The growing use of these technologies in popular Web sites (Amazon, MyYahoo), the emergence of companies that develop recommender technologies (NetPerceptions, Yodlee), and special issues of academic journals [7, 8] indicate that such systems promise a new approach to addressing the problem of information overload.

Despite researchers' keen interest, recommendation remains an unsolved problem. Current systems differ with regard to what data they use and how they use it. This article introduces Recommendation Explorer, an experimental recommender system. Borrowing from bibliometrics and hypertext analysis Recommendation Explorer operates on a square matrix that describes the paths of inter-recommendation between database items. We report an experiment on the effectiveness of dimensionality reduction by applying the singular value decomposition (SVD) to this item-item matrix. SVD allows evocative second- and third-order semantic patterns to inform the system's similarity model. In previous work Sarwar *et al.* [10, 9] used SVD to get 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.

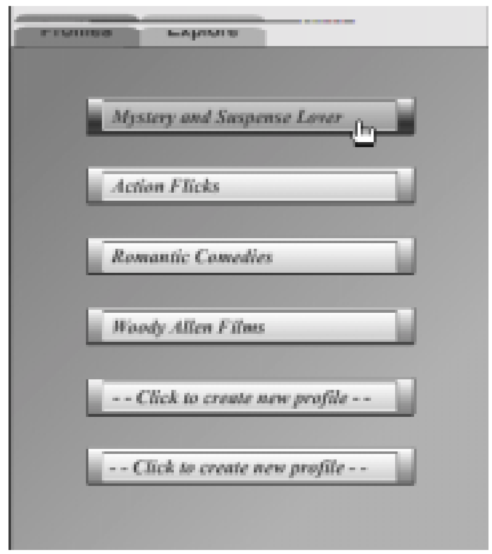

Figure 1: User-defined interest profiles enable quick, personalized recommendations

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 multidimen-

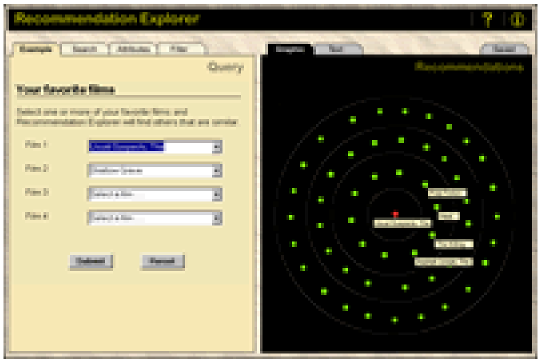

Figure 2: Interface for viewing and updating recommendations

sional 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.

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: similar actors act in similar films.

The remainder of this paper describes an experimental module of RecEx that attempts to provide a suitable film-film similarity model by use of the singular value decomposition. The film similarity module derives a mapping of the system's recommendation space, collocating "similar films."

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) [2, 1] 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 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.

[1] A full description of SVD is beyond the scope of this study. For a more detailed treatment see [5]

3.3 Implementation

To compute the SVD of our film-film matrix, we used SVDPACKC [4], 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 [3, 11]. Common practice in IR applications indicates a dimensionality between 50 and 300. After some trial and error, we selected $k = 50$.

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

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

Title	Full	1/3	1/2
Alien	46	12	3
Austin Powers	67	7	2
English Patient	47	10	3
Fargo	46	6	3
Full Monty	48	4	0
Room with a View	46	13	3
Sleepless in Seattle	35	7	2
Spanish Prisoner	43	15	5
Star Wars	61	8	6
Terminator	35	14	5

Table 1: Seed films and the number of candidates comprising their three keys

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 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 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 three "keys" for each seed—lists of films that constitute good recommendations for a given seed. The full key contains all candidate films selected by any reviewer. The third-key contains titles selected by at least one-third of the seed's reviewers. The half-key contains those films recommended by at least half of the seed's reviewers (*Full Monty* had very diverse ratings; no candidates were chosen by half of all reviewers). Table 1 shows the number of titles in each seed's three keys. The three key levels impose decreasingly stringent requirements on what constitutes a "good" recommendation. The full key requires no consensus, while the half-key requires a high degree of user consensus. Thus we expect that the full key contains more idiosyncratic selections than the half- or third-keys, and will contain items more difficult to retrieve.

Using our keys as a point of reference, we evaluate recommendation performance by using two metrics, precision/recall and a weighted variant of the average search length (ASL) [6]. 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).

$$\frac{\sum_{i=1}^{n} rank(s)_i * rank(u)_i}{\sum_{i=1}^{n} rank(u)_i} \tag{3}$$

We supplement precision/recall with weighted ASL for a number of reasons. Calculating a useful range of recall on small relevance sets such as seen in the half-keys was not meaningful. We also desired a metric that accounts for the fact that members of each seed's key are ranked. ASL measures the quality of a retrieval system by returning the average position of a key-member in the system's ranked output. Low ASL indicates good performance (relevant items near the front). We weight our ASL score by considering the number of reviewers that voted for each member of the key. Thus a film that all reviewers agree is a good recommendation affects evaluation more heavily than those only selected by a few reviewers. Weighted ASL is defined in formula 3, where n is the number of films in the key, $rank(s)_i$ is the position of the *ith* film in the algorithm's (either SVD or raw[2]) output, $rank(u)_i$ is the position of the *ith* film in the key, ordered by reviewer votes (ties are sorted randomly). If a candidate is present in the key but not in the first 1000 films returned by the algorithm, we count it as a failure, and assign it a penalty rank of 7000.

4.2 Results

4.2.1 Quality of SVD recommendations

Table 2 compares the two methods, raw and SVD, using average precision at five levels of recall (.1,

[2] To approximate an ordering by the raw link method we take note of how many levels of recursion a given candidate is from the seed. i.e. Items at 1 level out are considered to have a higher ranking than items 2 or 3 levels down. Candidates at the same level are then "sorted" randomly

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

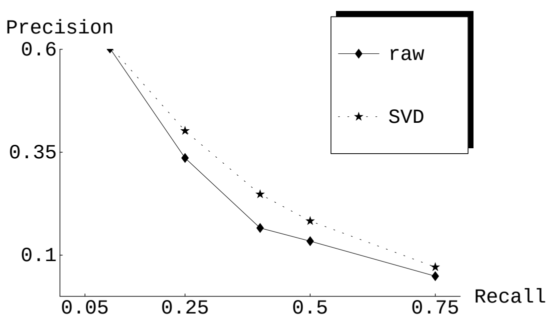

Figure 3: Average precision/recall for the raw data and data transformed by SVD

.25, .4, .50, .75). For this comparison relevance was defined by the "full" key. The SVD method provides better average precision for seven out of the ten seed films. Figure 3 plots precision against recall for each method. Each point is averaged across all ten seeds. SVD appears to offer the most improvement over the raw link structure at middling levels of recall.

Table 3 compares SVD and raw recommendations via the weighted ASL method. In the full key analysis (shown here), SVD improved upon the raw link structure for all seeds but one, (*Terminator*). Results were also favorable under third- and half-keys, with SVD leading to lower weighted ASL than raw links seven and eight times out of ten, respectively. The intuitive impression that SVD leads to lower weighted ASL was largely borne out by a series of hypothesis tests, summarized in Table 4. Although the distinction between weighted ASL based on SVD and raw linkage is high for all three keys, our hypothesis test is especially strong under the full key. This suggests that the SVD analysis improves the similarity model implied by the raw link

Seed	Raw	SVD
alien	6045.46	1882.72
austin powers	7337.10	2030.94
english patient	11550.83	3379.36
fargo	7325.73	4406.15
full monty	12919.89	2798.39
room/view	13461.87	294.44
sleepless	11072.86	3172.37
spanish prisoner	7336.16	5323.54
star wars	8221.43	4107.95
terminator	1445.93	2929.54

Table 3: Weighted ASL of Recommendations based on raw links structure and links analyzed by SVD, (Full Key)

			$H_0 : \mu_{raw} = \mu_{svd}$
KEY	$\overline{x}_{raw}$	$\overline{x}_{svd}$	p-value
Full	8671.73	3032.54	0.0007
1/3	2520.28	1202.86	0.0734
1/2	1839.69	677.552	0.1015

Table 4: Hypothesis tests concerning the equality of mean weighted ASL for raw and SVD-derived recommendations

structure. This improvement is especially evident for those circumstances where even the most idiosyncratic candidate films are considered relevant. However, the benefit of SVD is less clear when measured by precision/recall. Average precision (across recall levels and seed films) using the raw matrix was 0.258; SVD averaged 0.3021. The p-value for the test $H_0 : \mu_{raw} = \mu_{svd}$ using precision/recall was 0.50617, suggesting limited benefit from SVD.

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 ASL and precision/recall using each space.

Figure 4 charts a dramatic improvement in ASL performance as we increase k from 45 to 50. As k increases from 50 to 100, performance degrades slightly. A similar dynamic emerges when performance is measured by average precision at each level of k. Average precision peaks near $k = 50$. Spaces of lower dimensionality again appear insufficiently informative for the recommendation task, while the 100-dimensional space is slightly less effective than the 50-dimensional space.

Figure 4 plots weighted ASL for each of the three

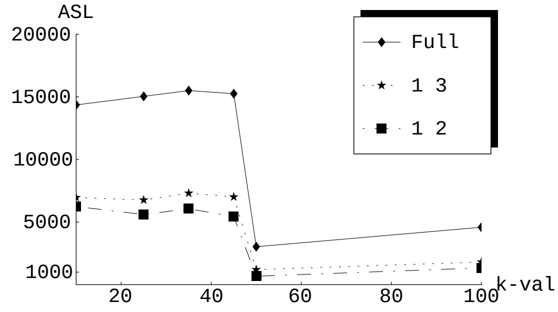

Figure 4: Weighted ASL in terms of the dimensionality of the SVD space, for 3 key levels.

key levels tracked in this experiment. While all three definitions of relevance show the same region of optimal dimensionality, the most dramatic improvement in performance is seen in the case of the full key. Keys based on 1/3 or 1/2 consensus appear to be more tolerant of an improperly chosen k value than the full key. This suggests that the features that allow the system to discern similarity between a seed film and a "weakly" relevant candidate are easily elided during dimensionality reduction. Thus systems such as RecEx that hope to use the similarity apparatus of the reduced space as part of a personalized recommendation system must be sensitive to this parameter.

5 Conclusion

Applying SVD to an item-item recommendation matrix has yielded promising results. The SVD module of Recommendation Explorer was able to match reviewer tastes more closely than a system based on untransformed data. This suggests that dimensionality reduction of the type of link structure described in this study permits useful and non-obvious patterns to inform recommendation.

Future research will improve the method described here. We plan to pursue analyses of the choice of k— the dimensionality of the reduced space—and of other data suitable for dimensionality reduction.

Finally, however useful it may be, a robust similarity model is only one element among many in a personalized recommendation system. In future work we will explore how to use the model described here to permit users to articulate their preferences and needs easily and specifically.

References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] R. M. Losee. *Text Retrieval and Filtering: Analytic Models of Performance*. Kluwer, Boston, 1998.

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

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

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

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

[11] 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.