Feature Reduction for Product Recommendation in Internet Shopping Malls

Hyung Jun Ahn*, Jong Woo Kim **
* hjahn@waikato.ac.nz , Department of Management Systems,
Waikato Management School, University of Waikato, New Zealand
** Department of Management, Hanyang University, Seoul, Korea

Abstract: One of the widely used methods for product recommendation in Internet shopping malls is matching product features against customers' profiles. In this method, it is very important to choose suitable set of features for recommendation efficiency and performance, which has, however, not been rigorously researched so far. In this paper, we build a data set collected from a virtual Internet shopping experiment and adapt and apply feature reduction techniques from pattern matching and information retrieval fields to the data to analyze recommendation performance. The analysis shows that the application of SVD (Singular Value Decomposition) can be the best among the applied methods for recommendation performance.

Keywords: Intelligent agent and data mining technologies in e–services; product recommendation; content based filtering; singular value decomposition.

I. Introduction

There have been a large number of studies for recommendation of products to promote cross-selling or up-selling in Internet shopping malls. The studies can be broadly classified into two types: collaborative filtering methods that use similarity of ratings for products among shoppers, and content-based filtering methods that utilize features or attributes of products and users [2, 3, 7, 10, 15, 16]. Although collaborative filtering methods have been proved to be successful in many studies, it is often very difficult or expensive to collect the ratings, and moreover, when rating data is sparse, recommendation results are usually impaired severely [9, 14, 18]. For these reasons, because it is easier to collect purchase data and associated product features, content-based filtering methods can be a practically better choice for many real world recommendation problems.

In content-based filtering methods, product attributes and user characteristics, often called features, are collected and analyzed with recommendation models such as statistics methods and artificial intelligence models. The features may comprise of various values such as demographic data or user-specified preferences, but in general, for the practical difficulty of collecting data in Internet shopping malls, many methods suggested in previous studies use characteristics of products that shoppers have purchased or shown interests in the past [13, 17].

Although the product characteristics are comparatively easier to acquire, there are often too a large number of available features. For example, in the cases of book or movie recommendation, there can be a large number of keyword features of the products from their themes, genres, and text-based description of the products. These features may directly affect the choice of shoppers, and thus, are often used as key features of recommendation, especially for cultural and content-focused products. This problem may lead to inefficiency or ineffectiveness of recommendation. In terms of efficiency, inaccurate or irrelevant keywords may require huge system memory or lead to increased processing time; in terms of effectiveness, there can be often inaccurate or irrelevant words that may damage the recommendation quality. Consequently, it has been recognized to be very important to extract a more meaningful subset of features that can contribute better to recommendation performance [1, 5, 8].

In this study, we utilized a data set that was constructed from a virtual shopping experiment in an Internet book shopping mall in Korea to find out how feature reduction techniques that have been widely used in pattern matching or information retrieval can be applied to the recommendation problem. We used a Korean lexical analyzer to extract features and use the features to construct user and product profiles. The Term Frequency (TF), Inverse Document Frequency (IDF), TFIDF, mutual information, and Singular Value Decomposition (SVD) methods were chosen and adapted for the experiment.

The remaining part of the paper is organized as follows. In the second section, we provide a brief review of feature selection methods from other disciplines. In the third section, we explain the experiment procedure and present the results of the analysis. The fourth section concludes with discussion and further research issues.

II. Review of Related Research

II. 1 Feature Extraction from Product Description

The studies on feature selection has been mainly the focus of pattern matching, machine learning, or text categorization research [1, 5, 8]. The studies have mainly focused on investigating what types of features affect learning performance more. However, there have been not many studies that have applied those methods to product recommendation in Internet shopping malls. Similar methods for feature reduction used for classification problems are often not directly applicable for recommendation.

In order to extract keyword-based features from product

Proceedings of the Fifth International Conference on Electronic Business, Hong Kong, December 5-9, 2005, pp. 132 - 137.

description, the tools and methods developed in the information retrieval filed can be well utilized. In general, sentences and phrases of product description include many words that are not suitable to be used as features such as articles, conjunctions, or pronouns. After filtering out unsuitable words, it is needed to extract the stem of the chosen words to make the keywords compatible with other variations of the same stem. For example, after stemming, two words 'processing' and 'processes' will have the same stem 'process'. These processes are usually performed using lexical analyzer software programs, and as the result, vector profiles of features such as <w1, w2, w3, …> are generated for each product and shopper, where each wi represents the relative importance of a keyword feature in the profile. Thus, we may compare vectors of users and products to calculate the similarity between them, which can be easily used for recommendation. Two most widely used similarity measures are linear correlation and cosine value between two vectors.

II. 2 Vector Space Model

Vector space model[12] is also widely used in the information retrieval field for representing documents and queries. Documents and queries are modeled as vectors of keywords where the keywords have different weights according to their relative importance in documents and queries. Table 1 shows an example vector for a document and a query. If we assume that there are only three keywords such as <information, management, computer>, the document and query can be represented with 3 dimensional vectors such as <0.3, 0.7, 0.1> and <0.5, 0, 0.8> respectively. Each number in the vectors shows the relative importance of the keywords in the example. We may then calculate the similarity between the two vectors using the cosine measure that will be close to 1 if the two are very similar to each other and 0 if they are very different from each other.

Table 1. Example of vector space representation

Keyword	Document	Query
Information	0.3	0.5
Management	0.7	0
Computer	0.1	0.8

II. 3 TFIDF

We have seen examples of weights in the vector representation in table 1. There can be numerous ways of calculating weights but the most widely used weighting scheme in the information retrieval field is TFIDF [12]. TFIDF combines TF (Term Frequency) and IDF (Inverse Document Frequency), where TF is the simple number of occurrence of a keyword in a given document or query, and IDF is the discriminative power of a given keyword in a given set of corpus. For example, a keyword that appears in most documents in a corpus will have a very low value of IDF, while a keyword that appears in only a small number of documents will have a very high value. As a result, because TFIDF is the product of the two, it considers both the frequency of a keyword within a given document or query and its discriminative power in a given corpus. Formally, TF, IDF, and TFIDF are defined as follows:

$$TF(t,d) = \frac{Occ(t,d)}{MaxOcc(d)} \tag{1}$$

where *Occ(t,d)* is the number of occurrence of the word t in the document d, and *MaxOcc(d)* is the number of occurrence of a word that appeared most in d.

$$IDF(t) = \frac{N}{N(t)} \tag{2}$$

where N is the number of entire documents in a given corpus and *N(t)* is the number of documents that contains the word t.

Accordingly, TFIDF can be calculated as follows:

$$TFIDF(t,d) = \frac{Occ(t,d)}{MaxOcc(d)} \cdot \frac{N}{N(t)} \tag{3}$$

II. 4 Mutual Information

Mutual information was first introduced in Shannon's information theory and is used to represent the amount of information that two probability events provide to each other [1, 9]. That is, MI(*a*, *b*), mutual information for events *a* and *b*, is bigger if the occurrence of an event gives higher information on the occurrence of the other. This value is symmetric for both events, thus, if *a* and *b* are more associated probabilistically, they produce higher mutual information value.

In product recommendation, we can use mutual information to estimate how strongly a feature is associated with a user and recommend products with features of higher mutual information value. A mutual information for two events *a* and *b* can be calculated as follows:

$$MI(a,b) = \log_2 \frac{P(ab)}{P(a)P(b)} \tag{4}$$

where *P(a)* is the probability of *a*'s occurrence, *P(b)* is the probability of *b*'s occurrence, and *P(ab)* is the joint probability of *a* and *b* occurring together.

II. 5 Singular Value Decomposition (SVD)

SVD is one of the matrix decomposition methods in Linear algebra. The basic idea of this method is based on the fact that a small subset of singular values generated by the decomposition can be a good approximation of the entire original matrix. The decomposition gives us two matrices of orthogonal vectors and a diagonal matrix with singular values as diagonal elements. The number of non-zero diagonal elements equals or is smaller than the dimension of the original matrix. It has been shown that reduced matrices

with smaller dimensions and hence smaller singular values may well produce a matrix which is very close to the original matrix in many research problems. For example, it is often used to compress a large image file or reduce the number of keywords for indexing documents in huge document management systems. In particular, in the information retrieval field, it has been found that the dimension reduction leads to the finding of *latent meaning* in documents, which have been proved to be successful for document indexing and retrieval The use of SVD in information retrieval is also called Latent Semantic Indexing (LSI) for the reason. The most significant implication of using SVD is that using a smaller dimension in information retrieval may lead to enhanced retrieval performance, contrary to intuition, because SVD helps utilize the latent meaning structure in documents [6, 11].

Figure 1 shows how a matrix can be decomposed using SVD. The original matrix is transformed into a product of the matrices U, D, and V, where U and V are each orthogonal matrices and D is a diagonal matrix where the diagonal elements are called *singular values* of the decomposition. It follows that the singular value σ_i gets smaller as i increases, which means that the earlier singular values have greater influence if we reassemble the original matrix by multiplying the three matrices with only a subset of the singular values.

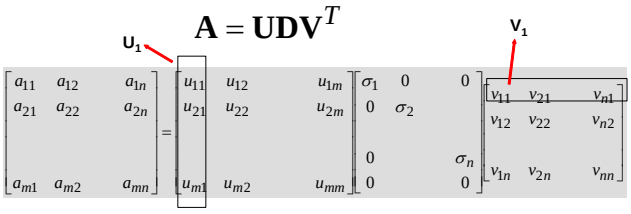

Figure 1. Illustration of singular vector decomposition

III. Experiment and Analysis

III. 1 Data Collection

Data for the experiment and analysis were collected through two activities. First, a virtual shopping experience was performed where 140 examinees browsed freely through a real Internet book shopping mall in Korea and put items they wanted to buy into virtual shopping carts provided by the mall and the result was recorded for each examinee. Table 2 shows the basic statistics of the experiment. Second, they were presented with 32 books as shown in table 3 and were asked to rate the books with scores ranging from 1 to 5, 5 meaning most preferred and 1 meaning least preferred. The 32 books for this experiment were chosen from the 16 categories shown in table 3. We chose only recently published books at the time of the experiment and placed a gap of 1 month between the two experiments to avoid any overlap between the books purchased in the virtual shopping and the 32 books for rating, which was successful. The books collected by the virtual shopping experiment are used to develop user profiles and the 32 books with ratings are used for evaluating the performance of recommendation using different feature reduction methods.

Table 2. Virtual shopping experiment

Average books purchased	10.93
Standard deviation	7.01
The largest number of books purchased among all the examinees	50
The smallest number of books purchased among all the examinees	2

Table 3. 16 categories from which 32 books were chosen for examinees' rating

Computer/ Internet	Management/ Economics	Foreign languages	Children
Hobby/heal th	Comics/Animation	Novel	Poem
Humanity	Essays	Classics	Social science
Science	History	Art	Magazines

Each record collected for the books contains ISBN or ISSN that uniquely identifies a book, as well as title, and description of books.

III. 2 Extraction of Features

For the content-based filtering experiments, first, keywords were extracted from the books of the two experiments. Table 4 shows basic statistics of the extracted keywords.

Table 4. Basic statistics of extracted keywords

	Average no. of keywords	Standard deviation	Maximum no. of keywords among all examinees	Minimum no. of keywords among all examinees
Per each user	551.2	238.8	1374	64
Per each of the 32 books for recommend ation	150.1	132.5	721	15

Among all the keywords extracted, only 3034 keywords that appear at least once in both experiments were chosen because the other words cannot contribute to recommen-dation at all. The frequency of keywords shows an exponential distribution as shown in figure 2. This shows that there are keywords that appear too frequently or too rarely, which in both cases may not be effective for recommendation, and which justifies the attempts to reduce the number of keywords.

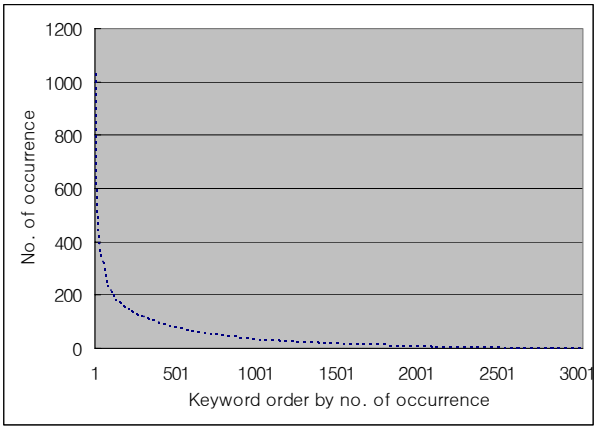

Figure 2. X axis represent each of the 3034 keywords that were ordered by their frequency of appearance. Y axis shows actual frequency of appearance.

On the other hand, figure 3 shows the histogram of keywords according to the frequency of occurrence. This graph roughly shows that a large portion of keywords appear in less than 5 documents and almost all keywords appear in less than 100 documents.

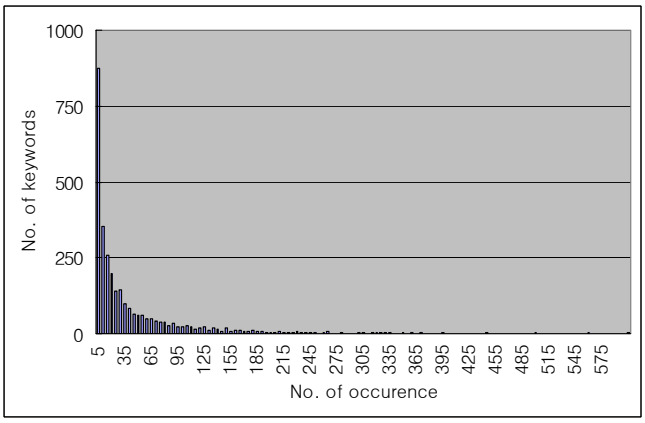

Figure 3. X axis shows the number of occurrences (frequency). Y axis shows the number of keywords for a corresponding frequency on the X axis

III. 3 Recommendation Experiments

With the data collected through the aforementioned methods, 5 recommendation experiments were performed using different ways of feature reduction: TF, IDF, TFIDF, MI (Mutual Information), and SVD. In each of the experiment, we measured the recommendation performance increasing the number of features or dimensions from 1. In using TF, IDF, TFIDF, and MI methods, considering that the average number of keywords in the recommended books is 150, the number of features or keywords was increased up to 200. With the SVD method, since the original matrix is of size 140 by 3034 with only 140 singular values, the experiment was performed using dimensions ranging from 1 to 140. For TF, IDF, TFIDF, and SVD methods, the cosine measure was used for similarity calculation. In the MI method, since we have *information value* for each feature instead of a vector profile, the amount of information measured in bits was used for similarity calculation.

Table 5. Summary of each experiment

Method	Range of features or dimensions	Similarity measure used
TF	1~200	Cosine
IDF	1~200	Cosine
TFIDF	1~200	Cosine
MI	1~200	Amount of information
SVD	1~140	Cosine

(1) Feature reduction using TF

The TF method is the simplest one among the five methods, where we chose those keywords first that appear more in each user profile. We increased the number of keywords for recommendation from 1 to 200.

(2) IDF

With the IDF method, we chose those words first that have higher discriminative power, or higher IDF value. For the calculation of IDF values, we treated all the books purchased by users in the virtual shopping experiment as a corpus.

(3) TFIDF

With this method, the words with higher TFIDF value were chosen first.

(4) MI

Using the MI method, we use the amount of information of each keyword of each user's profile, where the information represents the strength of association between the keyword and the user. For example, if a keyword *w* is highly associated with a user, the user will purchase many books containing the keyword, and conversely, the books purchased by the user should contain many occurrences of *w*. Suppose that *P(B)* represents the portion of the books purchased by a specific user among the entire set of books exposed to the user, and *P(w)* the portion of books containing the word *w*, and *P(Bw)* the probability of the intersection of the two. Then the MI value can be calculated as:

$$MI(B,w) = \log_2 \frac{P(Bw)}{P(B)P(w)} \qquad (5)$$

However, we cannot know the size of the entire books exposed, and hence, we cannot precisely measure *P(B)*, *P(w)*, and *P(Bw)* but should approximate them. First, when the sample size is large enough, we can replace *P(w)* with *P'(w)* where $P(w) \approx \frac{N'(w)}{N} = P'(w)$, *N* is the total number of books found in the virtual shopping experiment, and *N'(w)* is the number of books that contain *w*.

Second, suppose *N(U)* is the number of all the books in

the shopping mall which can be possible exposed to users, and *N(B)* is the the number of books a user has purchased after being exposed to all the books. Since we may not assume that the user has been exposed to all the books, we use an approximation *N'(B)* which should be a subset of *N(B)*: $N'(B) \approx \frac{N(B)}{k}$, where $k > 1$ is an unknown real number. Upon the same assumption, the size of *N(Bw)*, which is the size of the intersection between *B* and *W*, can be approximated in a similar way: $N'(Bw) \approx \frac{N(Bw)}{k}$. Thus, finally, we can approximate the MI value of a specific keyword for a specific user using only observable values as follows:

$$MI(B,w) = \log_2 \frac{P(Bw)}{P(B)P(w)} \approx \log_2 \frac{\frac{N'(Bw)}{N(U)}}{\frac{N'(B)}{N(U)}P'(w)} \quad (6)$$

$$= \log_2 \frac{kN'(Bw)}{kN'(B)\frac{N'(w)}{N}} = \log_2 \frac{N \cdot N'(Bw)}{N'(B)N'(w)}$$

After each MI value for each keyword feature had been calculated, we chose the keywords with higher MI values first for the recommendation test.

(5) SVD

The matrix that utilizes SVD has the size of <No. of keywords × no. of examinees>. Thus, we have a matrix of size 3034 × 140, which was decomposed into U, S, and V as shown in figure 4. Each column vector in the original matrix that corresponds to a single user was composed with TFIDF-based weight values.

$$\begin{bmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ & & \\ a_{m1} & a_{m2} & a_{mn} \end{bmatrix} = \begin{bmatrix} u_{11} & u_{12} & u_{1m} \\ u_{21} & u_{22} & u_{2m} \\ & & \\ u_{m1} & u_{m2} & u_{mm} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & \\ & & \\ 0 & & \sigma_n \\ 0 & & 0 \end{bmatrix} \begin{bmatrix} v_{11} & v_{21} & v_{n1} \\ v_{12} & v_{22} & v_{n2} \\ & & \\ v_{1n} & v_{2n} & v_{nn} \end{bmatrix}$$

3034 X 140 3034 X m m X m m X 140

Figure 4. SVD decomposition of the user profile matrix (U, S, and V')

Figure 5 shows the 140 singular values that were gained as the result of the decomposition. We can also note that there are a small number of large singular values that may contribute highly to the original matrix.

In order to use the decomposed matrices for recommendation, we apply the following formula to calculate the similarity between user *i* and a particular book. Assuming that the book is represented as a vector X_q, the following vector is treated as a row of V: $V_q = X_q' US^{-1}$. And then, for the similarity calculation, cosine distance between V_q and V_i for user *i* is calculated[4].

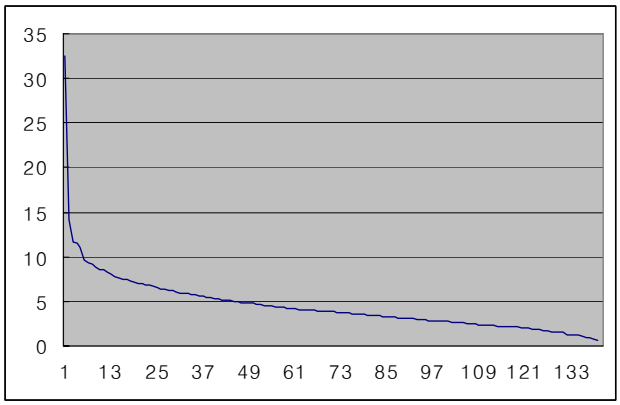

Figure 5. 140 singular values in a descending order

III. 4 Analysis of the Result

Figure 6 shows the result of recommendation experiment with X axis representing the number of features or number of dimensions used for recommendations, and Y axis representing the recommendation performance. In the experiment, 3 books with the highest similarity values were recommended to each user and the average rating of the 3 books by each user was used as a measure of performance.

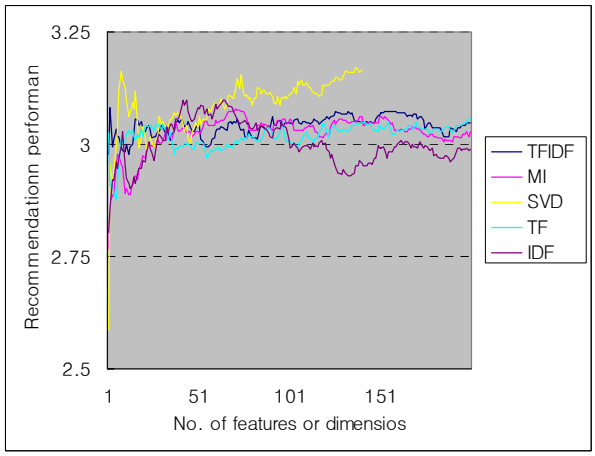

Figure 6. Number of features or dimensions and the recommendation performance

The result in figure 6 can be summarized as follows. First, on the whole, the recommendation using SVD shows better performance than other methods, while there is no different among the performances of the other methods. In TFIDF and TF methods, the results using only a small number of features are not much weaker than the results using much more features. With some variations, it is also observed that the use of additional keywords is not contributing much to increase in the overall recommendation performance. SVD method shows an interesting result where the performance is at its maximum when only 9 dimensions were used for recommendation, while the recommendation performance gradually reaches the same level as the number of dimensions approaches 140. This result is in accordance with the prediction of many studies that have shown that the

dimension reduction of SVD can utilize latent semantics in the data, which may better reflect the nature of data than using the entire set of dimensions.

However, there are a couple of aspects in this result that require more attention. First, although the experiment shows that using SVD may improve the overall recommendation performance, it is not clear how many dimensions should be used in general. For example, in figure 6, when there are approximately 50 features, it is outperforming the SVD method using 50 dimensions. Second, the concept of number of dimension is not exactly the same as the number of features. When using features in other methods than SVD for recommendation, the amount of information the recommendation system should manage is *<Number of features × Number of users>*. In the case of SVD, the amount of information required is *{<Number of features × Number of dimension> + <Number of users × Number of dimension>}*, which is larger than the other methods. This amount is still proportionate to the number of dimensions, and in terms of computational capacity, may not give much burden to recommendation systems. However, when the amount of information is also of great concern, it may be required to consider both aspects when choosing a feature reduction method.

IV. Conclusion and Further Research

In content-based filtering, the number of features has been one of the critical problems in terms of both recommendation efficiency and effectiveness. The contribution of this paper can be summarized as follows. First, we applied feature reduction methods from other disciplines with adaptation to the content-based recommendation problem. Second, we showed that, among the methods, SVD method can present the best recommendation performance with much smaller number of feature dimensions. Further research issues are as follows. First, in the current research, it has not been clearly shown how many features are required in general for various recommendation problems. Second, in order to generalize the findings of this research, experiments using other sets of data from different settings can be required.

References

[1].Al-Ani, A., M. Deriche, and J. Chebil, "A new mutual information based measure for feature selection". *Intelligent Data Analysis*, 2003. 7(1): p. 43

[2].Ansari, A., S. Essegaier, and R. Kohli, "Internet Recommendation Systems". *Journal of Marketing Research (JMR)*, 2000. 37(3): p. 363.

[3].Cohen, W.W. and W. Fan, "Web-collaborative filtering: recommending music by crawling the Web". *Computer Networks*, 2000. 33(1-6): p. 685

[4].DEERWESTER, S., et al., "Indexing by latent semantic analysis". *Journal of the American Society for Information Science*, 1990.41(6): p. 391-407.

[5].Deogun, J.S., et al., "Feature selection and effective classifiers". *Journal of the American Society for Information Science*, 1998. 49(5): p. 423.

[6].Gordon, M.D. and S. Dumais, "Using latent semantic indexing for literature based discovery". *Journal of the American Society for Information Science*, 1998. 49(8): p. 674.

[7].Greco, G., S. Greco, and E. Zumpano, "Collaborative Filtering Supporting Web Site Navigation". *AI Communications*, 2004. 17(3): p. 155.

[8].Huang, D. and T.W.S. Chow, "Effective feature selection scheme using mutual information". *Neurocomputing*, 2005. 63(1-4): p. 325.

[9].Huang, Z., H. Chen, and D. Zeng, "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering". *ACM Transactions on Information Systems*, 2004. 22(1): p. 116.

[10].Konstan, J.A., et al., "GroupLens: Applying Collaborative Filtering to Usenet News". *Communications of the ACM*, 1997. 40(3): p. 77.

[11].Latsche, T.A. and M.W. Berry, "Large-scale information retrieval with latent semantic indexing". *Information Sciences*, 1997.100(1-4): p.105.

[12].Lee, D.L. and H. Chuang, "Document ranking and the vector-space model". *IEEE Software*, 1997. 14(2): p. 67.

[13].Mobasher, B., R. Cooley, and J.Srivastava, "Automatic Personalization Based on Web Usage Mining". *Communications of the ACM*, 2000. 43(8): p. 142.

[14].Sullivan, D.O., B. Smyth, and D. Wilson, "Preserving Recommender Accuracy and Diversity in Sparse Datasets". *International Journal of Arificial Intelligence Tools*, 2004. 13(1): p. 219.

[15].Tout, K., D.J. Evans, and A. Yakan, "Collaborative filtering: special case in predictive analysis". *International Journal of Computer Mathematics*, 2005. 82(1): p. 1.

[16].Vozalis, M. and K.G. Margaritis, "On the combination of user-based and item-based collaborative filtering". *International Journal of Computer Mathematics*, 2004. 81(9): p. 1077.

[17].Weng, S.-S. and M.-J. Liu, "Feature-based recommendations for one-to-one marketing". *Expert Systems with Applications*, 2004.26(4): p. 493.

[18].Wilson, D.C., B. Smyth, and D.O. Sullivan, "Sparsity Reduction in Col laborative Recommendation: A Case-Based Approach". *International Journal of Pattern Recognition & Artificial Intelligence*, 2003. 17(5): p. 863.