Adapting Vector Space Model to Ranking-based Collaborative Filtering

Shuaiqiang Wang[1], Jiankai Sun[2], Byron J. Gao[3], Jun Ma[2]
[1]Shandong University of Finance and Economics, Jinan, 250014 China
[2]Shandong University, Jinan, 250101 China
[3]Texas State University-San Marcos, San Marcos, TX 78666, USA
swang@sdufe.edu.cn, sjk2412@gmail.com,
bgao@txstate.edu, majun@sdu.edu.cn

ABSTRACT

Collaborative filtering (CF) is an effective technique addressing the information overload problem. Recently ranking-based CF methods have shown advantages in recommendation accuracy, being able to capture the preference similarity between users even if their rating scores differ significantly. In this study, we seek accuracy improvement of ranking-based CF through adaptation of the vector space model, where we consider each user as a document and her pairwise relative preferences as terms. We then use a novel degree-specialty weighting scheme resembling TF-IDF to weight the terms. Then we use cosine similarity to select a neighborhood of users for the target user to make recommendations. Experiments on benchmarks in comparison with the state-of-the-art methods demonstrate the promise of our approach.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval–*Information filtering*

General Terms: Algorithms, Performance, Experimentation.

Keywords: Recommender systems, Collaborative filtering, Ranking-based collaborative filtering, Vector space model, Term weighting.

1. INTRODUCTION

Ever since the thriving of the Web, the world has been flooded with an overwhelming amount of information. This so-called *information overload* problem represents one of today's major challenges on the Web. As an effective technique addressing the problem, recommender systems generate item recommendations from a large collection in favor of user preferences. In recent years, recommender systems have become a de facto standard and must-own tool for e-commerce to promote business and help customers find products. Prominent examples include eBay, Amazon, Last.fm, Netflix, Facebook, and LinkedIn.

Collaborative filtering. The two main paradigms for recommender systems are content-based filtering and collaborative filtering (CF). Content-based filtering makes recommendations by finding regularities in the textual content information of users and items, such as user profiles and product descriptions [2]. CF is based on the assumption that if users X and Y rate n items similarly or have similar behaviors, they will rate or act on other items similarly [12].

CF only utilizes the user-item rating matrix to make predictions and recommendations, avoiding the need of collecting extensive information about items and users. In addition, CF can be easily adopted in different recommender systems without requiring any domain knowledge [8].

Ranking-based CF methods recommend items for users based on their rankings of items. In particular, such methods utilize similarity measures between two users based on their rankings on the same set of items. A common similarity measure is the Kendall tau rank correlation coefficient [10]. Recent efforts on ranking-based CF [8, 9, 11, 13, 7] have demonstrated their advantages in recommendation accuracy.

Adapting vector space model to ranking-based CF. In this study, we seek recommendation accuracy improvement for ranking-based CF through adaptation of the vector space model [1]. The vector space model is a standard and effective algebraic model widely used in information retrieval (IR). It treats a document as a bag of terms, and uses TF-IDF (or a variant weighting scheme) to weight the terms. Then each document is represented as a vector of TF-IDF weights. Queries are also considered as documents. Cosine similarity is used to compute relevancy of documents with respect to a given query.

To adapt the vector space model to ranking-based CF, we consider each user as a document and her pairwise relative preferences as terms. We then use a degree-specialty term weighting scheme resembling TF-IDF to weight the terms. After representing users as vectors of degree-specialty weights, cosine similarity is used to select a neighborhood of users for the target user to make recommendations.

Degree-specialty weighting. The key component in the adaptation is the degree-specialty weighting scheme. It is straightforward that relative preferences have different *degrees* depending on how strong the preferences are. For example, while both users u and v rank item i_1 higher than item i_2, their actual rating scores for $\{i_1, i_2\}$ may be $\{5, 1\}$ and $\{2, 1\}$ respectively, reflecting the fact that user u prefers i_1 over i_2 much more strongly than user v. Obviously, stronger preferences with larger score differences are more important and should be given a larger degree. Degree resembles *term frequency* (TF), which is defined as the number of occurrences of a term in a given document. A high degree for a preference

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

(term) from a user (document) can be interpreted in a way that the user repeatedly (frequently) confirms her preference.

Now we explain *specialty*. In information retrieval, IDF (inverse document frequency) measures the rarity of a term and contributes positively to the similarity between two documents. A straightforward way of adapting IDF to ranking-based CF is to use the original definition of IDF, which is a dampened ratio of the total number of users over the number of users holding the preference (DF). However, our experiments have returned unsatisfactory results for this method. Example 1 demonstrates that this method does not well measure rarity of preferences.

A deeper analysis shows that although we conceptually treat preferences as textual terms, they have fundamental differences. While a textual term is un-directional involving only one entity, a preference is directional involving two entities. A preference always has an "enemy", which is its opposite preference. A document can either contain or do not contain a textual term. However, a user can contain (hold) or do not contain (hold the opposite of) a preference, or has not rated both items. In practice, most users would belong to the third category. They are irrelevant and should be ignored in measuring rarity. In light of this, we define specialty as a dampened ratio of (the number of users holding the preference + the number of users holding the opposite preference) over the number of users holding the preference.

EXAMPLE 1. *Let $\{i_1, i_2\}$ be two items. Suppose among the total number of 10,000 users, 1,000 have rated both i_1 and i_2, where 800 prefer i_1 to i_2 ($i_1 \succ i_2$) and 200 prefer i_2 to i_1 ($i_2 \succ i_1$). In this case, $i_2 \succ i_1$ is a rare preference because there are 4 times more users holding the opposite preference. The specialty for preference $i_1 \succ i_2$ is based on $\frac{1000}{800}$ instead of $\frac{10000}{800}$, and the specialty for preference $i_2 \succ i_1$ is based on $\frac{1000}{200}$ instead of $\frac{10000}{200}$.*

Why we cannot use the original IDF? Let $\{i_3, i_4\}$ be two items. Suppose among the total number of 10,000 users, 100 have rated both i_3 and i_4 with 80 holding preference $i_3 \succ i_4$ and 20 holding preference $i_4 \succ i_3$. In this case, $i_4 \succ i_3$ is a rare preference and $i_3 \succ i_4$ is a popular one. If IDF is used, then a popular preference $i_3 \succ i_4$ would have a much bigger IDF than a rare preference $i_2 \succ i_1$ because $\frac{10000}{80} > \frac{10000}{200}$.

The effectiveness of our approach can be understood from another perspective. Terms are features. Feature selection and weighting has been one of the most frequently used techniques in pattern recognition, machine learning and data mining for data analysis, in particular, classification tasks. It eliminates irrelevant, redundant and noisy data. Although numerous classification frameworks and algorithms have been proposed, predicting accuracy is upper bounded by the amount of noise in historical data. Reducing noise has the most direct and immediate effects on predicting accuracy [4], as it would for recommendation accuracy.

2. PRELIMINARIES

Ranking-based collaborative filtering. The following notations will be used throughout the paper. Let U be a set of users and I be a set of items. In a recommender system, for each user $u \in U$, a set of items $I_u \subseteq I$ are rated by u. Let R be a rating matrix, where each element $r_{u,m} \in \mathbb{N}$ is the rating score of the m^{th} item i_m with respect to u, and $\mathbb{N}$ is the natural number set indicating different relevance scores.

Collaborating filtering (CF) recommends items to users based on their neighbors (similar users). In particular, for user u, the similarity between u and each user in U is computed from the rating matrix R. Then a set of neighborhood users $U_u \subset U$ are selected, based on which recommendations are made.

Ranking-based CF recommends items based on their rankings derived from the rating matrix R. The similarity between two users u and v, $\tau_{u,v}$, can be computed by the standard Kendall tau rank correlation coefficient [10] based on the two rankings from u and v on their common item set. Formally,

$$\tau_{u,v} = \frac{N_c - N_d}{\frac{1}{2}N(N-1)},$$

where N_c and N_d are the numbers of the concordant pairs and discordant pairs respectively.

For user u, the preference on a pair of items (i_m, i_n) can be predicted with a preference function $\Psi_u(m, n)$ as follows.

$$\Psi_u(m,n) = \frac{\sum_{v \in U_u^{m,n}} \tau_{u,v}(r_{v,m} - r_{v,n})}{\sum_{v \in U_u^{m,n}} \tau_{u,v}}$$

where $U_u^{m,n}$ is the set of similar users of u who have rated both items i_m and i_n.

Based on the predicted pairwise preferences, a total ranking of items for user u can be obtained by applying a preference aggregation algorithm.

Vector space model. The vector space model [1] is a standard algebraic model commonly used in information retrieval (IR). It treats a textual document as a bag of words, disregarding grammar and even word order. It typically uses TF-IDF (or a variant weighting scheme) to weight the terms. Then each document is represented as a vector of TF-IDF weights. Queries are also considered as documents. Cosine similarity is used to compute similarity between document vectors and the query vector. Large similarity indicates high relevancy of documents with respect to the query.

The term frequency $TF_{t,\boldsymbol{d}}$ of term t in document d is defined as the number of times that t occurs in d. It positively contributes to the relevance of d to t.

The inverse document frequency IDF_t of term t measures the rarity of t in a given corpus. If t is rare, then the documents containing t are more relevant to t. IDF_t is obtained by dividing N by DF_t and then taking the logarithm of that quotient, where N is the total number of documents and DF_t is the document frequency of t, i.e., the number of documents containing t. Formally, $IDF_t = \log_{10}\left(\frac{N}{DF_t}\right)$. The TF-IDF value of a term is commonly defined as the product of its TF and IDF values. $TF\text{-}IDF_{t,\boldsymbol{d}} = TF_{t,\boldsymbol{d}} \times IDF_t$.

Cosine similarity is a standard measure estimating pairwise document similarity in the vector space model. It corresponds to the cosine of the angle between two vectors, and it has the effect of normalizing the length of documents. Let $q = \langle w_{q,1}, w_{q,2}, \ldots, w_{q,N}\rangle$ and $d = \langle w_{d,1}, w_{d,2}, \ldots, w_{d,N}\rangle$ be two N-dimensional vectors corresponding a query and a document, their cosine similarity $s_{q,d} = \frac{q \cdot d}{||q|| \times ||d||}$.

3. ADAPTING VECTOR SPACE MODEL TO RANKING-BASED CF

In this section, we adapt the vector space model to ranking-based CF, where users are considered as documents and relative preferences are considered as terms. The terms are weighted by a degree-specialty weighting scheme resembling TF-IDF. The target user u

is considered as a query, which is also a document. The similarity between u and other documents are computed, based on which recommendations are made for u.

3.1 Representation of Users

We consider users as documents and pairwise relative preferences of items as terms. We adopt a bag of words model, where each user is represented as a *bag* of relative preferences, instead of a *set* as in other ranking-based CF methods.

In particular, for a user u, from the set I of items rated by u we can derive a set of relative preferences $\{i_m \succ i_n | i_m, i_n \in I \wedge r_{u,m} > r_{u,n}\}$. Each preference $i_m \succ i_n$ is considered as a term, and the score difference $|r_{u,m} - r_{u,n}|$ indicates the number of "occurrences" of the term in document u.

EXAMPLE 2. *Suppose user u has assigned 4, 3, 2 to items i_1, i_2 and i_3. The document u contains 3 terms and can be represented as "$i_1 \succ i_2, i_1 \succ i_3, i_1 \succ i_3, i_2 \succ i_3$".*

3.2 Term Weighting

Degree. Similar to TF, the degree of term $i_m \succ i_n$ in document u can be defined as the number of occurrences of $i_m \succ i_n$ in u. In this paper, we use a logarithm variant of TF. Formally, let $r_{u,m}$ be the rating score of item i_m by user u, then the degree of term $i_m \succ i_n$ is defined as:

$$w^{(D)}_{u,i_m \succ i_n} = \log_2 (1 + |r_{u,m} - r_{u,n}|)$$

Specialty. Similar to IDF, we want to use specialty to measure the rarity of terms in the corpus (set of users). Let us consider term $i_m \succ i_n$. A straightforward method would be using IDF literally, which is the log value of $\frac{|U|}{N_{i_m \succ i_n}}$, where $|U|$ is the total number of users (documents) and $N_{i_m \succ i_n}$ is the DF, that is, the number of users containing term $i_m \succ i_n$ (holding the preference).

However, we observe that textual terms and preference terms are fundamentally different. While a textual term is un-directional involving only one entity, a preference term is directional involving two entities. A preference term always has an "enemy", which is its opposite preference term. Also, a textual term t is either "contained" or "not contained" in a document d. However, a preference term $i_m \succ i_n$ can be "contained", "not contained", or "unknown" with respect to a user document u.

What is exactly *rarity* for preference terms? We say that *a preference term is rare if there are more opposite preference terms*. With the same interpretation, *a textual term is rare if there are more documents not containing the term*.

The original IDF captures this interpretation of rarity for textual terms, but not for preferences. The numerator of IDF is the total number of documents, which is the number of documents containing the term + the number of documents not containing the term. However, the total number of users is the number of users holding the preference + the number of users holding the opposite preference + the number of users who have not rated both items. Due to the typical sparsity of the rating matrix, most users have not rated both items.

In light of this, instead of using $|U|$ as the numerator, we use $N_{i_m \succ i_n} + N_{i_m \prec i_n}$ as the numerator. This can be considered as a phrasal translation (conveying the sense of the original) of IDF, instead of a literal word for word one.

For each pair of items (i_m, i_n), the relative preferences can be either $i_m \succ i_n$ or $i_m \prec i_n$. For simplicity, we combine the two opposite preference terms into one notation of $i_m \Theta i_n$, where $\Theta \in \{\succ, \prec\}$. Based on the above analysis, a possible way of defining specialty would be as follows:

$$w^{(S)}_{i_m \Theta i_n} = \log_2 \left(\frac{N_{i_m \succ i_n} + N_{i_m \prec i_n}}{N_{i_m \Theta i_n}} \right)$$

Degree-specialty. Resembling TF-IDF, degree-specialty is the product of degree and specialty. Specifically, for a user u, the degree-specialty weight of preference term $i_m \Theta i_n$ is defined as follows:

$$w_{u,i_m \Theta i_n} = w^{(D)}_{u,i_m \Theta i_n} \times w^{(S)}_{i_m \Theta i_n}$$

EXAMPLE 3. *Let $\{i_1, i_2\}$ be two items. Suppose 1,000 users have rated both i_1 and i_2, where 800 prefer i_1 to i_2 ($i_1 \succ i_2$) and 200 prefer i_2 to i_1 ($i_2 \succ i_1$). Suppose user u has assigned scores 2 and 5 and user v has assigned scores 4 and 3 to items i_1 and i_2 respectively.*

Then for user u, the degree-specialty for preference term $i_1 \prec i_2$ can be computed as follows.

$$\begin{aligned} w^{(D)}_{u,i_1 \prec i_2} &= \log_2 (1 + |2 - 5|) = 2, \\ w^{(S)}_{i_1 \prec i_2} &= \log_2 \left(\frac{1000}{200} \right) = 2.32, \\ w_{u,i_1 \prec i_2} &= 2 \times 2.32 = 4.64. \end{aligned}$$

Similarly, for user v, the degree-specialty for preference term $i_1 \succ i_2$ can be computed as follows.

$$\begin{aligned} w^{(D)}_{v,i_1 \succ i_2} &= \log_2 (1 + |4 - 3|) = 1, \\ w^{(S)}_{i_1 \succ i_2} &= \log_2 \left(\frac{1000}{800} \right) = 0.32, \\ w_{v,i_1 \succ i_2} &= 1 \times 0.81 = 0.32. \end{aligned}$$

3.3 Similarity Computation

Cosine similarity is a standard similarity measure in the vector space model, which corresponds to the cosine of the angle between two vectors.

The indicator p of a preference on a pair of items (i_m, i_n) can be defined as a number in {-1, 1}, where $p = -1$ for $i_m \succ i_n$ and $p = 1$ for $i_m \prec i_n$. Let $r_{u,m}$ and $r_{u,n}$ be the rating scores that have been assigned to items i_m and i_n respectively by user u. The value for the preference can be written as

$$p_{u,m,n} = \begin{cases} -1, & \text{if } r_{u,m} > r_{u,n} \\ 1, & \text{if } r_{u,m} < r_{u,n} \end{cases}$$

With degree-specialty weighting, user u is represented as a vector of degree-specialty weights $\hat{w}_u$, where each element is represented as $\hat{w}_{u,m,n} = w_{u,i_m \Theta i_n} p_{u,m,n}$. Then, the similarity between two users u and v can be computed by the standard cosine similarity:

$$\cos_{u,v} = \frac{\hat{w}_u \cdot \hat{w}_v}{||\hat{w}_u|| \times ||\hat{w}_v||}$$

3.4 Ranking Prediction

Generally, ranking-based CF works in two phases: (I) discovery of neighborhood users and (II) prediction of item ranking. For each user, Phase I discovers a set of most similar users as the neighborhood users, based on which Phase II predicts a ranking list of items for recommendation purposes.

We have discussed Phase I, where we use the vector space model to represent users and use degree-specialty to weight the preference terms, and use the cosine similarity $\cos^w_{u,v}$ to estimate similarity between users.

For Phase II, we adopt the ranking prediction method in EigenRank [8] for recommendation, which involves two steps: preference prediction and preference aggregation. During preference prediction, pairwise relative preferences for user u is predicted based on preferences of her neighborhood users. During preference aggregation, such predicted pairwise preferences are aggregated into a total ranking of items for recommendation via a greedy method.

Preference Prediction. Following [8], we define a preference prediction function to predict preferences for users. For a given user u, the preference prediction function $\Psi_u(i_m, i_n) : I \times I \rightarrow \mathbb{R}$ assigns real number confidence scores to documents, where I is the item set and $\mathbb{R}$ is the real number set. $\Psi_u(i_m, i_n) > 0$ indicates that item m is more preferable to n by user u and vice versa. The magnitude of the preference function $|\Psi_u(i_m, i_n)|$ implies the evidence of the preference, and a value of zero means that there is no preference between the two items.

Similar to the rating-based and ranking-based CF methods, for a given user u, preferences are predicted based on the neighborhood user set U_u. The basic idea is that the more often the users N_u prefer item i_m than i_n, the stronger the evidence for the preference prediction. Formally,

$$\Psi_u(i_m, i_n) = \frac{\sum_{v \in U_u^{m,n}} s_{u,v} \times p_{v,m,n}}{\sum_{v \in U_u^{m,n}} s_{u,v}}$$

where $U_u^{m,n}$ is the set of similar users of u who have rated both items i_m and i_n.

Preference Aggregation. Based on the predicted pairwise preferences, a total ranking of items for the target user u can be generated via preference aggregation.

Let τ_u be a ranking of items in I such that $\tau_u(i_m) > \tau_u(i_n)$ if user u prefers item i_m to item i_n. The evidence function $\Lambda(\tau_u)$ measures how consistent the ranking τ_u is with the preference prediction function Ψ_u:

$$\Lambda(\tau_u) = \sum_{\forall (i_m, i_n): \tau_u(i_m) > \tau_u(i_n)} \Psi_u(i_m, i_n)$$

Therefore, our goal is to produce an optimal ranking τ_u^* that maximizes the evidence function. However, Cohen et al. [3] proved that it is a NP-hard problem. Following [8], we use a greedy algorithm for discovery of an approximately optimal ranking.

4. EXPERIMENTS

4.1 Methodology

Datasets. Two movie rating real datasets were used in our experiments, MovieLens (http://www.grouplens.org/) and EachMovie (http://www.grouplens.org/node/76). The MovieLens dataset consists of 1 million ratings assigned by 6040 users to a collection of 3952 movies. The EachMovie dataset contains about 2.8 million ratings, which are made by 74,418 users on 1648 movies. The MovieLens rating scale is from 1 to 5, while the EachMovie rating scale is from 0 to 5.

Evaluation measures. For rating-based collaborative filtering, the standard evaluation criterion is the rating prediction accuracy. Commonly used accuracy measures include the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). Both measures depend on difference between true rating and predicted rating. Since our study focuses on improving item rankings instead of rating prediction, we employ the Normalized Discounted Cumulative Gain (NDCG) [5] metric. This metric is popular in information retrieval for evaluating ranked results, where documents are assigned graded rather than binary relevance judgements.

In the context of collaborative filtering, item ratings assigned by users can naturally serve as graded relevance judgements. Specifically, the NDCG metric is evaluated over some number n of the top items on the ranked item list. Let U be the set of users and $r_{u,p}$ be the rating score assigned by user u to the item at the pth position of the ranked list from u. The NDCG at the nth position with respect to the given user u is defined as follows.

$$NDCG_u@n = Z_u \sum_{p=1}^{k} \frac{2^{r_{u,p}} - 1}{\log(1+p)}$$

For the set of users U, the average NDCG at the nth position is:

$$NDCG_{avg}@n = \frac{1}{|U|} \sum_{u \in U} NDCG_u@n$$

The value of NDCG ranges from 0 to 1. A higher value indicates better ranking effectiveness. NDCG is very sensitive to the ratings of the highest ranked items. This is modeled by the discounting factor $\log(1+p)$ that increases with the position in the ranking. This is a highly desirable characteristic for evaluating ranking quality in recommender systems. This is because, just as in Web search, most users only examine the first few items from the recommended list. The relevance of top-ranked items are far more important than other items [8].

Comparison partners. We used two state-of-the-art ranking-based collaborative filtering algorithms, EigenRank [8] and CoFiRank [13], as our main comparison partners. EigenRank measured similarity between users with Kendall tau rank correlation coefficient for neighborhood selection, predicted pairwise preferences of items with a preference function, and aggregated the predicted preferences into a total ranking with a greedy algorithm. CoFiRank used Maximum Margin Matrix Factorization and employed structured output prediction to directly optimize ranking scores instead of ratings. In addition, we also included comparisons with UVS [6], a conventional user-based collaborative filtering method.

Experiment setup. In our experiments, we randomly selected 80% rated items for training and used the remaining 20% for testing. In order to guarantee that there are adequate number of common rating items between each neighborhood user and the target user, we filtered those users who have rated less than 50 items in MovieLens and 100 items in EachMovie. We ran each algorithm 5 times and reported the average performance.

4.2 Performance

We name our vector space model-based and ranking-based recommendation system `VSRank`. We evaluated the accuracy performance of `VSRank` in comparison with EigenRank, CoFiRank and UVS on EachMovie and MovieLens. Figures 1 and 2 show the performance comparison under the NDCG measure. From the figures we can see that:

- Our proposed degree-specialty weighting scheme can discover a more accurate set of neighborhood users, resulting in improved recommendation accuracy. For the two benchmark datasets, `VSRank` outperformed all other comparison

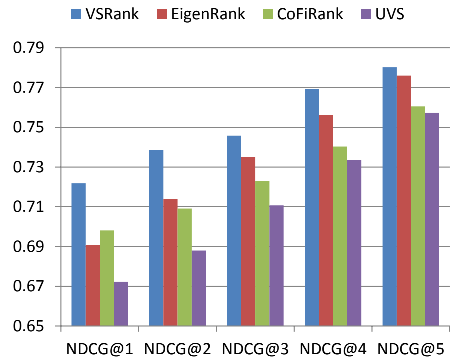

Figure 1: Performance comparison on MovieLens

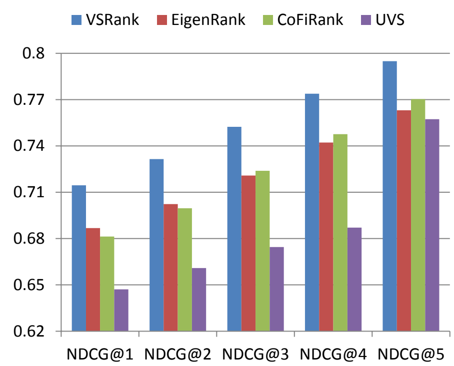

Figure 2: Performance comparison on EachMovie

partners. For example, for MovieLens, `VSRank` achieved 0.7218 and 0.7386 on NDCG@1∼2 comparing to 0.6908 and 0.7138 for EighenRank, gaining 4.49% and 3.47% improvement respectively. For EachMovie, `VSRank` achieved 0.7145 and 0.7315 on NDCG@1∼2 comparing to 0.6813 and 0.6996 for CoFiRank, gaining 4.87% and 4.56% improvement respectively.

- Ranking-based collaborative filtering is more accurate than rating-based methods on the NDCG evaluation measure. In our experiments, all the ranking-based methods outperformed the rating-based method UVS. For example, for MovieLens, `VSRank` achieved 0.7218 and 0.7386 on NDCG@1∼2 comparing to 0.6723 and 0.6880 for UVS, gaining 7.36% and 7.35% improvement respectively. For EachMovie, `VSRank` achieved 0.7145 and 0.7315 on NDCG@1∼2 comparing to 0.6471 and 0.6609 for UVS, gaining 10.42% and 10.68% improvement respectively.

5. CONCLUSION

In this paper, we have adapted the vector space model to ranking-based collaborative filtering for improved recommendation accuracy. Different from existing ranking-based CF methods that treat each user as a set of preferences, we adopt the bag of words model capturing the "frequency" of preferences. Different from existing ranking-based CF methods that treat preferences equally, we use a novel degree-specialty weighting scheme resembling TF-IDF. Users are represented as vectors of degree-specialty weights and cosine similarity is used to compute a highly similar neighborhood of the target user for accurate recommendation.

There are several interesting directions for future work. Firstly, other ranking-based similarity measures can be experimented for improving neighborhood quality. Secondly, knowing that there are many TF-IDF variants, we plan to investigate other possible variants of degree-specialty and study their performance in different applications. Last but not least, the proposed adaptation framework is not limited to ranking-based CF. We plan to explore a similar adaptation of the vector space model to rating-based CF and examine its effectiveness.

6. ACKNOWLEDGEMENT

This research was supported in part by the Natural Science Foundation of China (60970047, 61103151, 61173068, 71171122), the Foundation of Ministry of Education of China (20110131110028, 12YJC630211), the Natural Science Foundation of Shandong Province of China (ZR2012FM037, BS2012DX012), and the National Science Foundation (OCI-1062439, CNS-1058724).

7. REFERENCES

[1] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[2] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35:29–38, 1992.

[3] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[4] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[5] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[6] C. M. K. John S. Breese, David Heckerman. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificia Intelligence (UAI)*, 1998.

[7] M. Kahng, S. Lee, and S. goo Lee. Ranking in context-aware recommender systems. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*, 2011.

[8] N. N. Liu and Q. Yang. Eigenrank: A ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2008.

[9] N. N. Liu, M. Zhao, and Q. Yang. Probabilistic latent preference analysis for collaborative filtering. In *Proceedings of the 18th ACM conference on Information and Knowledge Management (CIKM)*, 2009.

[10] J. I. Marden. *Analyzing and Modeling Rank Data*. Chapman & Hall, New York, 1995.

[11] Y. Shi, M. Larson, and A. Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the 4th ACM conference on Recommender systems (RecSys)*, 2010.

[12] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009.

[13] M. Weimer, A. Karatzoglou, Q. V. Le, and A. J. Smola. Cofirank: Maximum margin matrix factorization for collaborative ranking. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.