Collaborative Filtering based on User Trends

Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos Papadopoulos, and Yannis Manolopoulos

Aristotle University, Department of Informatics, Thessaloniki 54124, Greece
{symeon, alex, apostol, manolopo}@delab.csd.auth.gr

Abstract. Recommender systems base their operation on past user ratings over a collection of items, for instance, books, CDs, etc. Collaborative Filtering (CF) is a succesful recommendation technique. User ratings are not expected to be independent, as users follow trends of similar rating behavior. In terms of Text Mining, this is analogous to the formation of higher-level concepts from plain terms. In this paper, we propose a novel CF algorithm which uses Latent Semantic Indexing (LSI) to detect rating trends and performs recommendations according to them. Our results indicate its superiority over existing CF algorithms.

1 Introduction

The "information overload" problem affects our everyday experience while searching for valuable knowledge. To overcome this problem, we often rely on suggestions from others who have more experience on a topic. In Web case, this is more manageable with the introduction of Collaborative Filtering (CF), which provides recommendations based on the suggestions of users who have similar preferences.

Latent Semantic Indexing (LSI) has been extensively used in informational retrieval, to detect the latent semantic relationships between terms and documents. Thus, higher level concepts are generated from plain terms. In CF, this is analogous to the formation of users' trends from individual preferences.

In this paper, we propose a new algorithm that is based on LSI to produce a condensed model for the user-item matrix. This model comprises a matrix that captures the main user trends removing noise and reducing its size.

Our contribution and novelty are summarized as follows: (i) based on Information Retrieval, we include the pseudo-user concept in CF (ii) We implement a novel algorithm, which tunes the number of principal components according to the data characteristics. (iii)We generalize the recommendation procedure for both user- and item-based CF methods. (iv) We propose a new top-N

generation list algorithm based on SVD and the Highest Prediction Rated items.

The rest of this paper is organized as follows. Section 2 summarizes the related work, whereas Section 3 contains the analysis of the CF factors. The proposed approach is described in Section 4. Experimental results are given in Section 5. Finally, Section 6 concludes this paper.

2 Related work

In 1992, the Tapestry system (Goldberg et al. (1992)) introduced Collaborative Filtering (CF). In 1994, the GroupLens system (Resnick et al. (1994)) implemented a CF algorithm based on common users preferences. Nowadays, it is known as user-based CF algorithm. In 2001, another CF algorithm was proposed. It is based on the items' similarities for a neighborhood generation of nearest items (Sarwar et al. (2001)) and is denoted as item-based CF algorithm.

Furnas et al. (1988) proposed Latent Semantic Indexing (LSI) in Information Retrieval area to detect the latent semantic relationships between terms and documents. Berry et al.(1994) carried out a survey of the computational requirements for managing (e.g., folding-in, which is a simple technique that uses existing SVD to represent new information.) LSI-encoded databases. He claimed that the reduced-dimensions model is less noisy than the original data.

Sarwar et al.(2000) applied dimensionality reduction for only the user-based CF approach. In contrast to our work, Sarwar et al. included test users in the calculation of the model as were apriori known. For this reason, we introduce the notion of pseudo-user in order to insert a new user in the model (folding in), from which recommendations are derived.

3 Factors affecting CF

In this section, we identify the major factors that critically affect all CF algorithms. Table 1 summarizes the symbols that are used in the sequel.

Symbol	Definition	Symbol	Definition
k	number of nearest neighbors	$p_{u,i}$	predicted rate for user u on item i
N	size of recommendation list	c	number of singular values
$\mathcal{I}$	domain of all items	A	original matrix
$\mathcal{U}$	domain of all users	S	Singular values of A
u, v	some users	V'	Right singular vectors of A
i, j	some items	A^*	Approximation matrix of A
$\mathcal{I}_u$	set of items rated by user u	$\mathbf{u}$	user vector
$\mathcal{U}_i$	set of users rated item i	$\mathbf{u_{new}}$	inserted user vector
$r_{u,i}$	the rating of user u on item i	n	number of train users
$\overline{r}_u$	mean rating value for user u	m	number of items
$\overline{r}_i$	mean rating value for item i	U	Left singular vectors of A

Table 1. Symbols and definitions.

Similarity measures: A basic factor for the formation of user/item neighborhood is the similarity measure. The most extensively used similarity measures are based on correlation and cosine-similarity (Sarwar et al (2001)). Specifically, user-based CF algorithms mainly use Pearson's Correlation (Equation 1), whereas for item-based CF algorithms, the Adjusted Cosine Measure is preferred (Equation 2)(McLauglin and Herlocker (2004)).

$$\mathrm{sim}(u,v) = \frac{\sum_{\forall i \in S} (r_{u,i} - \overline{r}_u)(r_{v,i} - \overline{r}_v)}{\sqrt{\sum_{\forall i \in S} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{\forall i \in S} (r_{v,i} - \overline{r}_v)^2}}, S = \mathcal{I}_u \cap \mathcal{I}_v. \quad (1)$$

$$\mathrm{sim}(i,j) = \frac{\sum_{\forall u \in T} (r_{u,i} - \overline{r}_u)(r_{u,j} - \overline{r}_u)}{\sqrt{\sum_{\forall u \in \mathcal{U}_i} (r_{u,i} - \overline{r}_u)^2} \sqrt{\sum_{\forall u \in \mathcal{U}_j} (r_{u,j} - \overline{r}_u)^2}}, T = \mathcal{U}_i \cap \mathcal{U}_j. \quad (2)$$

Generation of recommendation list: The most often used technique for the generation of the top-N list, is the one that counts the frequency of each item inside the found neighborhood, and recommends the N most frequent ones. Henceforth, this technique is denoted as Most-Frequent item recommendation (MF). MF can be applied to both user-based and item-based CF algorithms.

Evaluation Metrics: Mean Absolute Error (MAE) has been used in most of related works. Although, it has received criticism as well(McLauglin and Herlocker (2004)). Other extensively used metrics are *precision* (ratio of relevant and retrieved to retrieved) and *recall* (ratio of relevant and retrieved to relevant). We also consider F_1, which is another popular metric for CF algorithms and combines the two previous ones.

4 Proposed Method

Our approach applies Latent Semantic indexing in CF process. To ease the discussion, we will use the running example illustrated in Figure 1 where I_{1-4} are items and U_{1-4} are users. As shown, the example data set is divided into train and test set. The null cells(no rating) are presented as zeros.

	I_1	I_2	I_3	I_4
U_1	4	1	1	4
U_2	1	4	2	0
U_3	2	1	4	5

(a)

	I_1	I_2	I_3	I_4
U_4	1	4	1	0

(b)

Fig. 1. (a) Train Set ($n \times m$), (b) Test Set.

Applying SVD on train data: Initially, we apply SVD on train data $n \times m$ matrix A that produces three matrices. These matrices obtained by SVD can

give by performing multiplication the initial matrix as the following Equation 3 and Figure 2 show:

$$A_{n\times m} = U_{n\times n} \cdot S_{n\times m} \cdot V'_{m\times m} \tag{3}$$

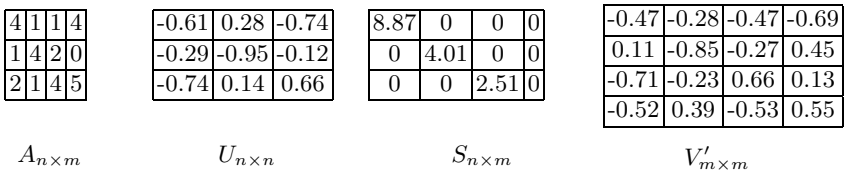


Fig. 2. Example of: $A_{n\times m}$ (initial matrix A), $U_{n\times m}$ (left singular vectors of A), $S_{n\times m}$ (singular values of A), $V'_{m\times m}$ (right singular vectors of A).

Preserving the Principal Components: It is possible to reduce the $n \times m$ matrix S to have only c largest singular values. Then, the reconstructed matrix is the closest rank-c approximation of the initial matrix A as it is shown in Equation 4 and Figure 3:

$$A^*_{n\times m} = U_{n\times c} \cdot S_{c\times c} \cdot V'_{c\times m} \tag{4}$$

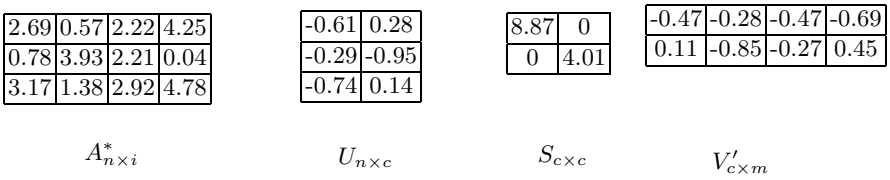


Fig. 3. Example of: $A^*_{n\times m}$ (approximation matrix of A), $U_{n\times c}$ (left singular vectors of A^*), $S_{c\times c}$ (singular values of A^*), $V'_{c\times m}$ (right singular vectors of A^*).

We tune the number, c, of principal components (i.e., dimensions) with the objective to reveal the major trends. The tuning of c is determined by the information percentage that is preserved compared to the original matrix. Therefore, a c-dimensional space is created and each of the c dimensions corresponds to a distinctive rating trend. We have to notice that in the running example we create a 2-dimensional space using 83% of the total information of the matrix (12,88/15,39).

Inserting a test user in the c-dimensional space: It is evident that, for user-based approach, the test data should be considered as unknown in the c-dimensional space. Thus a specialized insertion process should be used. Given the current ratings of the test user u, we enter pseudo-user vector in the c-dimensional space using the following Equation 5(Furnas et al (1988)). In the current example,we insert U_4 into the 2-dimensional space, as it is shown in Figure 4:

$$\mathbf{u_{new}} = \mathbf{u} \cdot V_{m\times c} \cdot S^{-1}_{c\times c} \tag{5}$$

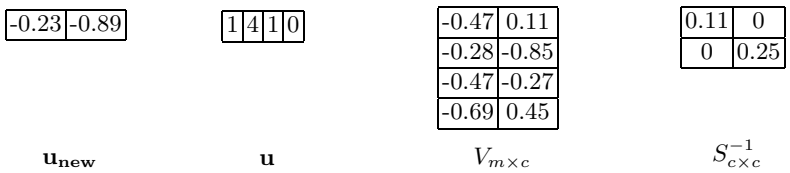


Fig. 4. Example of: $\mathbf{u_{new}}$ (inserted new user vector), $\mathbf{u}$ (user vector), $V_{m\times c}$ (two left singular vectors of V), $S^{-1}_{c\times c}$ (two singular values of inverse S).

Generating the Neighborhood of users/items: Having a reduced dimensional representation of the original space, we form the neighborhoods of users/items in that space. For the user-based approach, we find the k nearest neighbors of pseudo user vector in the c-dimensional space. The similarities between train and test users can be based on Cosine Similarity. First, we compute the matrix $U_{n\times c} \cdot S_{c\times c}$ and then we perform vector similarity. This $n \times c$ matrix is the c-dimensional representation for the n users.

For the item-based approach, we find the k nearest neighbors of item vector in the c-dimensional space. First, we compute the matrix $S_{c\times c} \cdot V_{c\times m}$ and then we perform vector similarity. This $c \times m$ matrix is the c-dimensional representation for the m items.

Generating and Evaluating the Recommendation List: As it is mentioned in Section 3, existing ranking criteria, such as MF, are used for the generation of the top-N list in classic CF algorithms. We propose a ranking criterion that uses the predicted values of a user for each item. Predicted values are computed by Equations 6 and 7, for the cases of user-based and item-based CF, respectively. These equations have been used in related work for the purpose of MAE calculation, whereas we use them for generation of top-N list.

$$p_{u,i} = \overline{r}_u + \frac{\sum_{v\in U} \text{sim}(u,v)(r_{v,i} - \overline{r}_v)}{\sum_{v\in U} |\text{sim}(u,v)|} \tag{6}$$

$$p_{u,i} = \overline{r}_i + \frac{\sum_{j\in \mathcal{I}} \text{sim}(i,j)(r_{u,j} - \overline{r}_j)}{\sum_{j\in \mathcal{I}} |\text{sim}(i,j)|} \tag{7}$$

Therefore, we sort (in descending order) the items according to predicted rating value, and recommend the first N of them. This ranking criterion, denoted as Highest Predicted Rated item recommendation (HPR), is influenced by the good accuracy of prediction that existing related work reports through the MAE. HPR opts for recommending the items that are more probable to receive a higher rating.

5 Experimental configuration

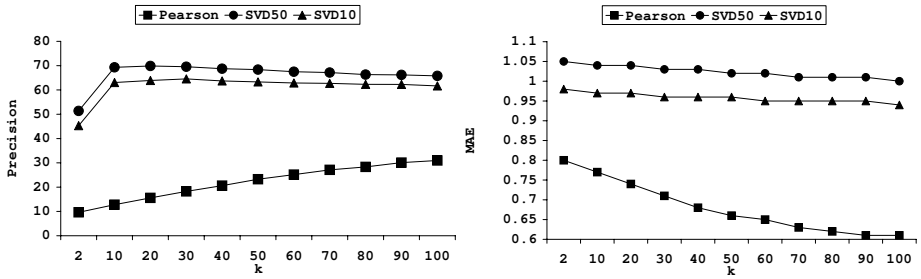
In the sequel, we study the performance of the described SVD dimensionality reduction techniques against existing CF algorithms. Both user-based and

item-based algorithms are tested. Factors, that are treated as parameters, are the following: the neighborhood size (k, default value 10), the size of the recommendation list (N, default value 20),and the size of train set (default value 75%). The metrics we use are recall, precision, and F_1.

We perform experiments in a real data set that has been used as benchmarks in prior work. In particular, we examined MovieLens data set with 100,000 ratings assigned by 943 users on 1,682 movies. The range of ratings is between 1(bad)-5(excellent) of the numerical scale. Finally, the value of an unrated item is considered equal to zero.

5.1 Results for user-based CF algorithm

Firstly, we compare existing user-based CF algorithm that uses Pearson similarity against two representative SVD reductions(SVD50 and SVD10). These are percentages of the initial information we keep from the initial user-item matrix after applying SVD. The results for precision and MAE vs. k are displayed in Figure 5a and b, respectively.



(a) (b)

Fig. 5. Performance of user-based CF vs. k: (a) precision, (b) MAE.

As shown, the existing Pearson measure performs worst than SDV reductions. The reason is that the MovieLens data set is sparse and relatively large (high n value). The SVD reductions reveal the most essential dimensions and filter out the outliers and misleading information.

We now examine the MAE metric. Results are illustrated in Figure 5b. Pearson yields the lowest MAE values. This fact indicates that MAE is good only for the evaluation of prediction and not of recommendation, as Pearson measure did not attain the best performance in terms of precision.

Finally, we test the described criteria for the HSR top-N list generation algorithm. The results for precision and recall are given in Figure 6. As shown, the combination of the SVD similarity measure with HPR as list generation algorithm, clearly outperforms the Pearson with HPR. This is due to the fact that in the former the remaining dimensions are the determinative ones and outliers users have been rejected. Note that in the SVD50 we preserve only 157 basic dimensions instead of 708 train users for the latter.

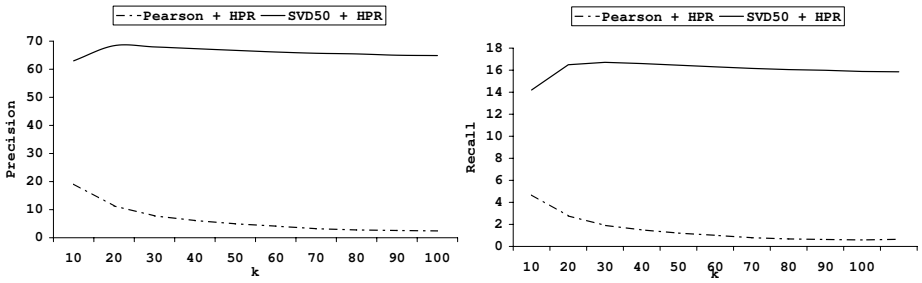


Fig. 6. Comparison HPR criteria for the generation of top-N (a) precision, (b) recall list for user-based CF vs. k

5.2 Results for item-based CF algorithms

We perform similar measurements for the case of item-based CF. Thus, we examine the precision and recall for the existing Adjusted Cosine Measure (considers co-rated items) against SVD50 and SVD10 for the item-based case. The results are depicted in Figure 7 and are analogous to those of the user-based case. MAE and HPR results are also analogous to the user-based case.

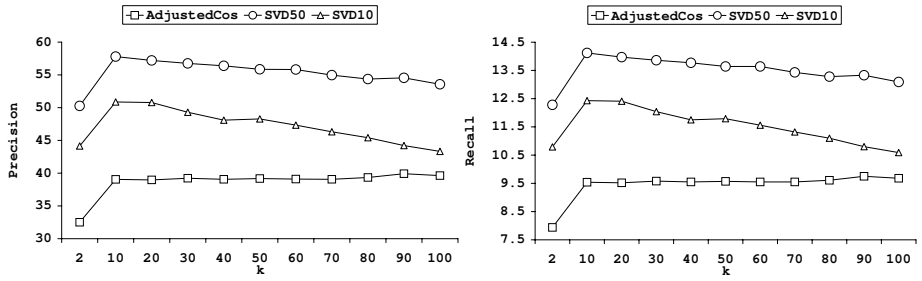
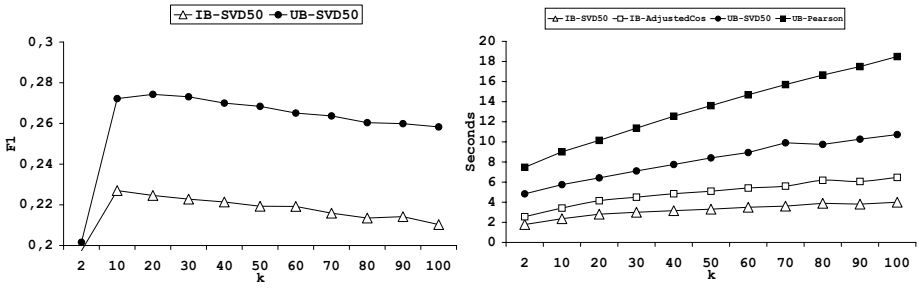


Fig. 7. Performance of item-based CF vs. k: (a) precision, (b) Recall.

5.3 Comparative results

We compared user-based and item-based SVD50. For the generation of the top-N list, for both of them we use MF. The results for F_1 are displayed in Figure 8a, which demonstrate the superiority of user-based SVD50 against item-based SVD50.

Regarding the execution time, we measured the wall-clock time for the on-line parts for all test users. The results vs. k are presented in Figure 8b. Item based CF needs less time to provide recommendations than user-based CF. The reason is that a user-rate vector in user-based approach has to be inserted in the c-dimensional space. The generation of top-N list for the user-based approach further burdens the CF process.



(a) (b)

Fig. 8. Comparison between item-based and user-based CF in terms of precision vs. k and in terms of execution times.

6 Conclusions

We performed experimental comparison of the proposed method against well known CF algorithms, like user-based or item-based methods (that do not consider trends), with real data sets. Our method show significant improvements over existing CF algorithms, in terms of accuracy (measured through recall/precision). In terms of execution times, due to the use of smaller matrices, execution times are dramatically reduced. In our future work we will consider the issue of an approach that would combine user and item based approaches, attaining high accuracy recommendations in the minimum responding time.

References

Berry, M. Dumais, S. and OBrien G. (1994): Using linear algebra for intelligent information retrieval. *SIAM Review,37(4):573-595.*

Furnas, G. Deerwester, S. Dumais. S. et al. (1988): Information retrieval using a singular value decomposition model of latent semantic structure.*In Proc. ACM SIGIR Conf., pages 465-480.*

Goldberg, D. Nichols, D. Brian, M. and Terry D. (1992): Using collaborative filtering to weave an information tapestry. *ACM Communications, 35(12):61-70.*

McLauglin R. and Herlocher J. (2004): A collaborative filtering algorithm and evaluation metric that accurately model the user experience.*In Proc. ACM SIGIR Conf., pages 329-336.*

Resnick, P. Iacovou, N. Suchak, M. Bergstrom, P. and Riedl. J. (1994): Grouplens-An open architecture for collaborative filtering on netnews.*In Proc. Conf. Computer Supported Collaborative Work, pages 175-186.*

Sarwar, B. Karypis, G. Konstan, J. and Riedl. J. (2000): Application of dimensionality reduction in recommender system-a case study.*In ACM WebKDD Workshop.*

Sarwar, B. Karypis, G. Konstan, J. and Riedl. J. (2001): Item-based collaborative filtering recommendation algorithms.*In Proc. WWW Conf., pages 285-295.*