ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20160708140839
Original scientific paper

Collaborative Filtering with Temporal Dynamics with Using Singular Value Decomposition

Cigdem BAKIR

Abstract: Nowadays, Collaborative Filtering (CF) is a widely used recommendation system. However, traditional CF techniques are harder to make fast and accurate suggestions due to changes in user preferences over time, the emergence of new products and the availability of too many users and too many products in the system. Therefore, it becomes more important to make suggestions that are both fast and take the changes in time into consideration. In the presented study, a new method for providing suggestions customized according to the users' preference and taste as they change over time was developed. By combining the time-dependent changes through the SVD (Singular Value Decomposition), a faster suggestion system was developed. Thus, an attempt was made to enhance product prediction success. In the present study all techniques on Netflix data and the results were compared. The results obtained on the accuracy of the predicted ratings were found out to be promising.

Keywords: Collaborative Filtering; recommendation system; temporal dynamics

1 INTRODUCTION

Recommendation systems implement techniques that discover information in order to make personalized recommendations during interaction with products and services, and they deal with substantial amount of data. They are used widely in today's e-commerce and web applications. However, due to the presence of great numbers of users and products in the system, it may prove difficult to make instant and accurate suggestions to users. It requires substantial processing time and memory.

Yet, particularly in online systems, users expect their requests to be met instantly. In addition, while making recommendations to users, new products and services that change individuals' tastes may emerge.

Also seasonal changes or regional factors [1, 4] may change users' perception and perspective over time. For instance, users may develop a new perspective for an actor or movie genre, or may change the types of movies he or she likes over time. Furthermore, every user may experience a different change. Alternatively, a change in a person itself or the person’s family may change the person's needs.

Due to the variations in users' requests, on the basis of regional or any other various factors, it becomes hard to make accurate and reliable suggestions for every user. The traditional item-based CF technique makes recommendations by considering users' ratings and the similarities across items [8]. However, traditional item-based CF technique cannot make recommendations pertinent to the changing requests and tastes of users. Since these systems ignore such changes over time, they recommend the same product every time. If it is an e-commerce application that implements this technique, the client will start losing interest since the application will not be able to meet the changing interests of the client, and as a result company productivity will fall [1].

A system that will take into consideration time dependent changes and that will observe the changes in every user's behaviour is needed in order to keep the users' interest alive and to enhance company profitability. With item-based CF with temporal dynamics using the age of ratings proposed in this study, temporal variations were taken into consideration and an attempt was made to increase prediction success. The longer a product rated by the users has been in the market, the more users has it been rated by. In other words, older products have remained longer and accordingly have been seen and rated by more users. In prediction systems, the dates on which ratings were given are as important as the ratings. The ratings given in the recent past to a product registered in the system are more valuable than the older ratings the said product has received. Accordingly, in the proposed system the ages of the user ratings were calculated and a plan was developed to increase the weights of newer ratings while decreasing the weights of older ratings. Assessing ratings on the basis of their age has enabled rating prediction success to increase significantly. Furthermore, in order to make faster and more accurate recommendations, the traditional item-based CF technique was integrated with SVD. SVD is a method that is quite effective in solving the problems faced in CF approaches such as data scarcity, scaling, synonym and latency. By means of the SVD method, dimension of data was reduced and the ratings users give to products were calculated faster. In order to enhance product prediction success, the success of the temporal CF technique with SVD method was calculated. With the SVD dimension reduction method, both the required memory and the processing time were minimized. In this way, more accurate recommendations were made at shorter periods of time.

2 RELATED WORK

Recommendation systems constitute a subclass of the data filtering systems that recommend items peculiar to a user on the basis of the rating that user had given to similar items. Today, these systems are used widely in various areas including movies, music, videos, news, books, social networks and web pages.

Recommendation systems generate recommendations in two ways - as Collaborative Filtering (CF) and Content-Based Filtering. CF makes recommendations by evaluating users' past behaviours, interaction with other users or product similarities. Content-based filtering, on the other hand, makes recommendations with consideration of the demographic characteristics of users such as age, race, occupation or where they live, and the typical characteristics of items that separate them into various categories.

Coi and Suh tried to enhance the quality of the recommendations by using similarity function for the selected neighbours of the item under consideration [2]. In order to solve the data scarcity and scalability problems of content-based CF and collaborative filtering methods, Liu proposed a hybrid recommendation system. In his study, Liu developed a method that integrates user and item behaviours. In order to resolve data scarcity, the author assigned average ratings to unrated items. As for the scaling problem, he classified users on the basis of their personal characteristics [3].

Xiang proposed a session-based temporal graph method that integrates short and long-term temporal variations [4]. Lathia, Hailes, Capra and Amarian compared the results they obtained from item-based CF, the k closest neighbour algorithm, and the SVD method - a matrix factorization technique - in terms of rating accuracy. Also with these methods, the authors examined the effects of temporal dynamics on user ratings [5].

For the purpose of solving the inadequacies of memory-based CF algorithms made using several similarity parameters, Hoffman developed model-based CF algorithms based on dimension reduction methods such as clustering, classification and SVD [20]

Yang and Liu used the MapReduce application in order to generate effective recommendations in short periods of time. In this way, the authors managed to reduce the calculation cost in data sets [7]

Liu, Hu and Zhu developed an effective method with a new similarity model to improve the accuracy. This model showed context information on user rating and increased performance [13].

3 ITEM BASED COLLABORATIVE FILTERING

In an item-based approach for k items most similar to the item i rated by the active user, $\{i_1,\ i_2,\ ....\ ,\ i_k\}$ similarities $\{w_{i,1},\ w_{i,2},\ ....\ ,w_{i,k}\}$ are calculated. $w_{i,j}$ shows the similarity item i has to item j. The most important step of item-based CF algorithms is to determine similar elements and to group the closest items. The main idea in calculating similarity is to determine the users that rated both item i and item j, and to calculate the $w_{i,j}$ similarity among the jointly rated items. The similarity between the items i and j is calculated via Pearson correlation as shown in Eq. (1) [10].

$$w_{i,j}=\frac{\sum_{u\in U}(r_{u,i}-\overline{r_i})\,(r_{u,j}-\overline{r_j})}{\sqrt{\sum_{u\in U}\,(r_{u,i}-\overline{r_i})^2}\,\sqrt{\sum_{u\in U}\,(r_{u,j}-\overline{r_j})^2}} \tag{1}$$

where $r_{u,i}$ is the rating user u gave to item i, $\overline{r_i}$ is the average of the ratings given to item i, $r_{u,j}$ is the rating user u gave to item j, and $\overline{r_j}$ is the average of the ratings given to item **j** [2]. In item-based CF technique, after the similarity between items is calculated, a prediction of rating ($p_{a,i}$) is made [10].

$$p_{a,i}=\frac{\sum_{allsimilarityitems,N}(w_{i,N}*r_{a,N}\ )}{\sum_{allsimilarityitems,N}(w_{i,N})} \tag{2}$$

4 SINGULAR VALUE DECOMPOSITION

Singular value decomposition (*SVD*) is a matrix factorization technique. On consideration that $\boldsymbol{A}$ is a $m\times n$ sized matrix and r is a rank, $\boldsymbol{A}$ matrix decomposed through *SVD* is presented in Eq. (3) [15].

$$SVD(\boldsymbol{A})=\ U\ \times\ S\ \times\ V^{\mathrm{T}} \tag{3}$$

$\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrixes of the respective dimensions of $m\times m$ and $n\times n$. $\boldsymbol{S}$, on the other hand, is a singular diagonal matrix, the diagonal entries of which are not zero. r diagonal entries of the **s** singular matrix are subject to the rules $(s_1,\ s_2,\ \ldots,\ s_n)\ s_i>0$ and $s_1\geq s_2\geq\ldots\geq s_r$. The first r columns of the $\boldsymbol{U}$ matrix are the eigenvector of $\boldsymbol{A}$. $\boldsymbol{A}^{\mathrm{T}}$, and signify the left singular vectors of the matrix $\boldsymbol{A}$. r columns of the matrix $\boldsymbol{V}$, on the other hand, are the eigenvectors of $\boldsymbol{A}^{\mathrm{T}}$. $\boldsymbol{A}$ and signify the right singular vectors of matrix $\boldsymbol{A}$.

In order to reduce dimension effectively through SVD, r values other than zero are taken and $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$ are formed at the respective dimensions of $m\times r$, $r\times r$ and $r\times n$.

The SVD technique expresses the matrix $\boldsymbol{A}$ in the best way. $k<<r$ singular values are obtained, the remaining values are removed and the entries of the matrix $\boldsymbol{S}$ are sorted. In this way the similarities and interactions among users that are latent and not visible in the original matrix $\boldsymbol{A}$, yet still significant, are found. While the obtained diagonal matrix is referred to as S_k, the matrixes $\boldsymbol{U}$ and $\boldsymbol{V}$ are referred to as U_k and V_k. U_k is generated by removing the columns r-k from the matrix $\boldsymbol{U}$, while V_k is generated by removing the columns r-k from the matrix $\boldsymbol{V}$. $\boldsymbol{A}_k$ matrix is connoted as shown in Eq. (4) [15]:

$$\boldsymbol{A}_k=U_k\times S_k\times V_k^{\mathrm{T}} \tag{4}$$

5 USE OF SVD IN ITEM BASED CF TECHNIQUES

SVD is used in both memory-based and model-based recommendation systems. It is a Latent Semantic Indexing (LSI) technique and it is quite effective in solving the problems faced in CF approaches such as data scarcity, scaling, synonym and latency [20]. The form where the $\boldsymbol{A}$ $m\times n$ dimensioned original user-item matrix was separated into its singular eigenvalues is as shown below [14, 16, 19]:

$$A_k\ =\ U_k\ \times\ S_k\ \times\ V_k^{\mathrm{T}} \tag{5}$$

In order to detect the latent relations and similarities between items and make more accurate predictions, the matrix $\boldsymbol{A}$ is reduced to the k dimension. While $U\sqrt{S}$ shows the users in the k dimension of the $m\times k$ matrix, $\sqrt{S}V$ shows the items in the dimension $n\times k$. In order to make more reliable predictions to the user, the original matrix **A** is shown at k dimension space with U_k, S_k and V_k. SVD usage in item-based CF techniques is carried out as shown in Eq. (6) [16]:

$$p_{u,i}\ =\ A_u\ +\ U_k\sqrt{S_k}(u)\times\ \sqrt{S_k}V_k^{\mathrm{T}}(i) \tag{6}$$

While $p_{u,i}$ is used for calculating the rating user *u* gave to item i, A_u indicates the average point user *u* gave to all items and $\boldsymbol{U}$, $\boldsymbol{S}$ and $\boldsymbol{V}$ respectively show the eigenvalue, diagonal matrix and eigenvectors obtained after the original matrix $\boldsymbol{A}$ was subjected to *SVD* application. $A_{m\times \mathrm{n}}$=$U_{m\times \mathrm{r}}\sqrt{S_{r\times \mathrm{r}}}\sqrt{S_{r\times \mathrm{r}}}V_{r\times \mathrm{n}}$, in item-based CF techniques SVD is shown as follows [14, 16, 19].

$$A_{m\times \mathrm{n}} = U_{m\times \mathrm{k}}\sqrt{S_{k\times \mathrm{k}}}\ (u)\sqrt{S_{k\times \mathrm{k}}}V^{T}_{n\times \mathrm{k}}(i) \qquad (7)$$

Uses of item-based CF and SVD in recommendation systems are presented in Eqs. (5), (6) and (7). In U eigenvalues, while the active user's (u) line $\sqrt{S_{k\times \mathrm{k}}}$ is multiplied with all columns, all lines of $\sqrt{S_{k\times \mathrm{k}}}$ and $V_{k\times n}$ are multiplied with the column of the item the rating of which is sought (i).

6 OUR SOLUTION

Item based CF with temporal dynamics by using age of ratings is implemented and prediction performance is enhanced [9]. When calculating the similarity between items in the traditional item-based CF technique, only the ratings given by the users to the products are used [11, 12].The ratings which are given to items are presented in Tab. 1, in the form of a user-product matrix. In these systems, an attempt is made to predict the ratings of the active user on the basis of the calculated item similarities. In item-based CF with temporal dynamics using the age of ratings, users' ratings are not predicted solely on the basis of the similarities among items, as it is the case in traditional item-based CF method. As distinct from the traditional item-based CF method, the age of the rating is added into the system. In other words, item similarity and the predicted rating will change according to the age of the ratings.

In item-based CF with temporal dynamics using age of ratings, $Age_{n,m}$ indicates the age of the rating user *n* gave to the product m. Let $r_{a,3}$ be the rating test example the active user gave to the product number 3. While calculating product similarity, also the age of each rating is needed. For instance, when calculating the age of $r_{1,3}$ the difference between the rating dates of $r_{1,3}$ and test data will be taken into consideration. Since the date on which the test data was to be calculated is 27.01.2005 and the date of $r_{1,3}$ is 27.01.2003, the difference between these two dates is 731 days or approximately 2 years [18]. This period indicates the age of $r_{1,3}$ on the basis of the present date (test date 27.01.2005). As it can also be seen in Tab.1, the ages of the ratings are dynamic pieces of information that change as per the date on which they are viewed.

Table 1 User-item matrix, based on the scores of the test sample table, with the age of the scores given in days

Users/Items	i_1	i_2	i_3	i_m
u_1	$r_{1,1}$ $Age_{1,1}$ (352 days) 10.02.2004		$r_{1,3}$ $Age_{1,3}$ (731 days) 27.01.2003	
u_2	$r_{2,1}$ $Age_{2,1}$ (434 days) 20.11.2003			$r_{2,m}$ $Age_{2,m}$ (26 days) 01.01.2005
a **Active user**	$r_{a,1}$ $Age_{a,1}$ (468 days) 17.10.2003		$r_{a,3}$=? $Age_{a,3}$ **(27.01.2005)** **Test sample**	
u_n	$r_{n,1}$ $Age_{n,1}$ (135 days) 14.09.2004			$r_{n,m}$ $Age_{n,m}$ (235 days) 06.06.2004

As can be seen in Tab. 1, the ages of the ratings are calculated based on the number of days and according to the date of test. In the developed system, the ages of these ratings were converted to years and used in that form. The objective here was to reduce errors in rating predictions by using the age of the ratings given in the item-based CF technique. The following paragraphs will outline the results of various tests.

In the proposed method, the weights of the ratings given by users were decreased for older ratings and increased for younger ratings. For this increase or decrease of rating weights, various conversion functions were tested and it was tried to determine the most suitable conversion function. Fig. 1 presents the conversion functions used to weight the ages of ratings. This conversion function was implemented as shown in Eq. (8) [18].

$$Age_{\mathrm{w}} = mAge_r + n \qquad (8)$$

In this function, while Age_r stands for the age of the present rating, Age_{w} indicates the weighted age of the present rating.

Since the proposed method was carried out on the basis of the ratings Netflix customers gave to a selection of movies in the last two years, Age_r takes a value within the range of [0,2]. Age_{w} takes the values calculated with the conversion function shown in Fig. 1.

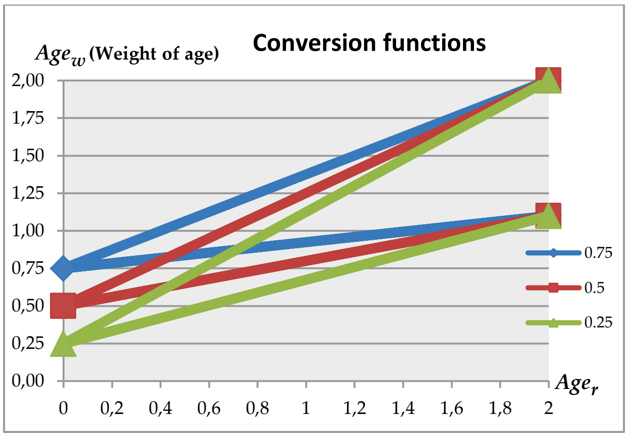

Figure 1 Conversion functions used for weighting the ages of scores

The Age_w calculated with the conversion function is used for calculating the new temporal value of the present rating obtained in consequence of weighing, as shown in Eq. (9) [18].

$$r_{aged} = r_{org}/Age_w \qquad (10)$$

Here, r_{org} is the current rating and r_{aged} is the rating weighted in terms of time. In conventional item-based CF technique explained in Section 2, r_{aged} is used instead of r.

Linear conversion is made through the $Age_w = mAge_r + n$ function. Tab. 2 shows varying conversion ranges for m while n = 0.25 is kept fixed. While Age_w takes the value 0.25 for the smallest Age_r value 0, conversion results for $Age_r =$ 2 were respectively calculated to be 2, 1.7, 1.5, 1.4, 1.3, 1.2 and 1.1. For n = 0.25, if Age_r=0 in the conversion function, when Age_w = 0.25, in consequence of the $r_{aged} = r_{org}/Age_w$operation, the original rating increases 4 times. On the other hand, when Age_w = 1.1, the weight of the original rating is decreased by 10 %. It is found m values using same conversion functions for other values of n. In this way the conversion range with which the most accurate and reliable predictions are made was tried to be determined.

Table 2 User-item matrix, based on the scores of the test sample table, with the age of the scores given in days

Age_r	Age_w	m
[0.....2] ⇒	[0.25.....2]	0.875
	[0.25.....1.7]	0.725
	[0.25.....1.5]	0.625
	[0.25.....1.4]	0.575
	[0.25.....1.3]	0.525
	[0.25.....1.2]	0.475
	[0.25.....1.1]	0.425

7 EXPERIMENTAL STUDY

For the purpose of comparatively measuring the success of the proposed system, the ratings Netflix clients gave to various movies during the last two years were taken into consideration. Netflix dataset consists of the ratings given by approximately half a million of clients to 17770 movies from 1999 to 2005 [17]. There are more than 100 million entries of ratings. It is a real dataset obtained from an U.S. company operating in the area of movie and video rentals and it is frequently used in recommendation systems. The results of the conducted application were calculated through *RMSE* (Root Mean Squared Error). *RMSE* is among the most widely used evaluation parameters in the Netflix dataset.

$$RMSE = \sqrt{\frac{1}{n}\sum_{\{i,j\}}(p_{i,j} - r_{i,j})^2} \qquad (11)$$

In this equation, while n shows the total number all users rated, $p_{i,j}$shows the predicted rating user i gave to item j and $r_{i,j}$ shows the actual rating user i gave to item j [1]. Traditional item based CF results are shown in Tab. 3.

Table 3 Traditional item based CF result

Traditional Item Based CF	RMSE	Percentage Error (%)
	1.976	39.524

In Fig. 2 was showed *RMSE* error rate obtained with temporal dynamics item-based CF. In Fig. 3 the results obtained from the neighborhoods of the classical item-based CF technique and the SVD method were compared to the results obtained from the neighborhoods of CF technique with temporal dynamics and SVD method. CF with temporal dynamics combined with SVD method provided more successful results than the results of the traditional item-based CF technique.

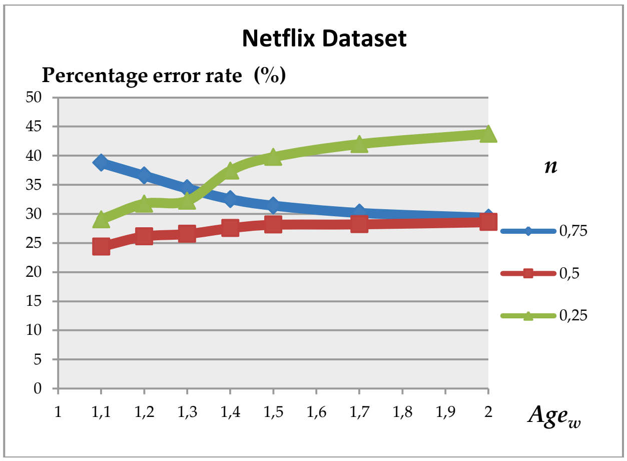

Figure 2 RMSE error rate obtained with temporal dynamics item-based CF

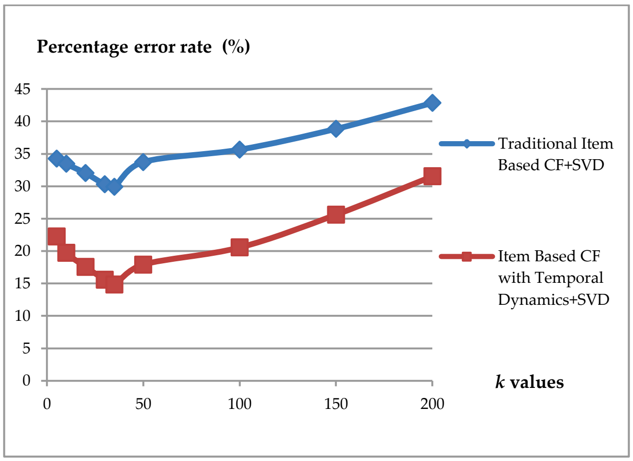

Figure 3 Item based with temporal dynamics combined SVD and item based combined SVD

Tab. 4 presents the results obtained by applying SVD on the Age_w [0.5...1.1] conversion range where the most accurate recommendations were made. By using both SVD and temporal conversion coefficient, it was ensured that the obtained result is increased.

Table 4 Item based collaborative filtering with temporal dynamics using SVD (for Age_w [0.5...1.1])

k values	*RMSE*	Percentage Error (%)
5	1.128	22.25
10	0.987	19.75
20	0.878	17.57
30	0.779	15.59
35	0.743	14.86
50	0.896	17.92
100	1.027	20.55
150	1.280	25.61
200	1.576	31.53

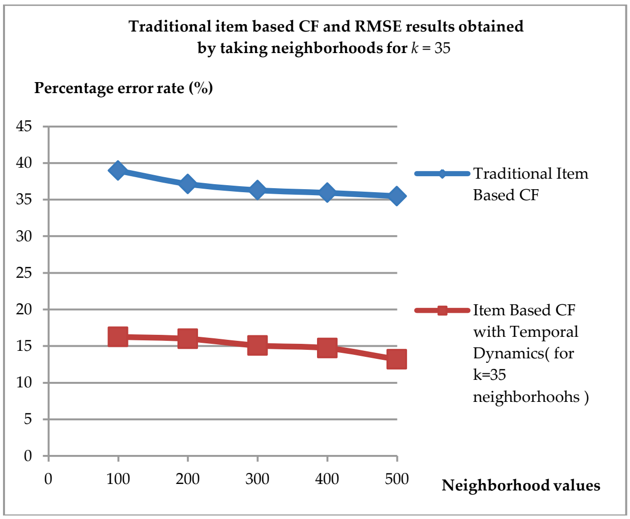

Figure 4 Traditional Item based CF and item based CF with temporal dynamics results by taking neighbourhoods

From Fig. 4 it can be seen that the results obtained from the neighborhoods for k = 35 of the application of SVD on the Age_w [0.5...1.1] range where the most successful ratings were obtained with the temporal item-based CF technique, are more successful than the results obtained from traditional item-based CF technique.

8 CONCLUSION

The traditional item-based CF technique makes recommendations through the sole consideration of user ratings. For this reason, it usually falls short in making recommendations suitable for the changing requests, tastes and interests of users. Also, it requires too much processing time. With the present study, an attempt was made to rectify the inadequacies of the traditional item-based CF technique. Available ratings were associated with their ages and through the use of several conversion coefficients, they were either increased or decreased in weight. In this way a more accurate, reliable and faster system that generates recommendations suitable for users' changing interests was designed. By developing a method that keeps users' interest alive and considers temporal dynamics, product prediction success was enhanced.

9 REFERENCES

[1] Koren, Y. (2009). Collaborative Filtering with Temporal Dynamics. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 447-456. https://doi.org/10.1145/1557019.1557072

[2] Choi, K. & Suh, Y. (2013). A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge based Systems, 37*, 146-153. https://doi.org/10.1016/j.knosys.2012.07.019

[3] Zhang, Z. & Lin, H. (2013). A hybrid fuzzy-based personalized recommender system for telecom products/services. *Elsevier Information Sciences*, 117-129. https://doi.org/10.1016/j.ins.2013.01.025

[4] Xiang, L. & Yuan, Q. (2010). Temporal Recommendation on Graphs via Long-and Short-term Prefence Fusion. *KDD'10 of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 723-732.

[5] Lathia, N. & Hailes, S. (2008). Trust-based collaborative filtering. *IFIP International Federation for Information Processing, 263*, 119-134. https://doi.org/10.1007/978-0-387-09428-1_8

[6] Yang, X. & Liu, P. (2014). Collaborative Filtering Recommendation using Matrix Factorization: A Map Reduce Implementation. *International Journal of Grid Distributed Computing, 2*(7), 1-10.

[7] Luo, C. & Cai, X. (2014). Self-training Temporal Dynamics Collaborative Filtering. *PAKDD'14*, 461-472. https://doi.org/10.1007/978-3-319-06608-0_38

[8] Shrot, T. & Rosenfeld, A. (2014). CRISP - An Interruption management algorithm based on collaborative filtering. *CHI'14, One of a Chind*, 3035-3044.

[9] Choi, K. & Suh, Y. (2013). A new similarity function for selecting neighbors for each target item in collaborative filtering. *Knowledge based Systems, 37*, 146-153. https://doi.org/10.1016/j.knosys.2012.07.019

[10] Sarwar, B. & Karypis, G. (2001). Item-based Collaborative Filtering Recommendation Algorithms. *WWW'01: International Conference on World Wide Web*, 285-295. https://doi.org/10.1145/371920.372071

[11] Herlocker, J. L. & Konstan, J. A. (1999). An Algorithmic Framework for Performing Collaborative Filtering. *ACM Conference on Research and Development Information Retrieval*, 230-237. https://doi.org/10.1145/312624.312682

[12] Gao, M. & Wu, Z. (2011). User rank for item-based Collaborative Filtering Recommendation. // *Inf. Process. Lett. 111*(9), 440-446. https://doi.org/10.1016/j.ipl.2011.02.003

[13] Liu, H. & Hu, Z. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems, 56*, 156-166. https://doi.org/10.1016/j.knosys.2013.11.006

[14] Chang, T. & Hsiao, W. (2014). An ordinal regression model with SVD hebbian learning for collaborative filtering. *Journal of Information Science and Engineering, 30*, 387-401.

[15] Sarwar, B. M., Karypis, G., Konstan, J. A. & Riedl, J. (2002). Incremental singular value decomposition algorithms for highly scalable recommender systems. *Proceedings of the 5th International Conference on Computer and Information Technology (ICCIT 2002)*.

[16] Gong, S. & Ye, H. (2009). Combining Singular Value Decomposition and Item-based Recommender in Collaborative Filtering. *WKDD'09 of the 2009 Second International Workshop on Knowledge Discovery and Data Mining*, 769-772. https://doi.org/10.1109/WKDD.2009.132

[17] Zhou, Y. & Wilkinson, D. (2008). Large-scale parallel collaborative filtering for the netflix prize. *Lecture Notes in Computer Science, 5034*, 337-348. https://doi.org/10.1007/978-3-540-68880-8_32

[18] Cigdem, B. (2014). Hybrid Collaborative Filtering with Temporal Dynamics. *International Journal of Recent and Innovations Trends in Computing and Communication, 2*(11), 3428-3432.

[19] Vozalis, M. G. & Margaritis, K. G. (2005). Applying SVD on Item-based filtering. Intelligent Systems Design and Applications 5th International Conference (ISDA205), 464-469. https://doi.org/10.1109/ISDA.2005.25

[20] Hoffman, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS), 22*(1). https://doi.org/10.1145/963770.963774

Contact information:

Cigdem BAKIR
Yildiz Technical University,
Davutpasa Street Esenler,
34220 İstanbul, Turkey
E-mail: cigdem@ce.yildiz.edu.tr