The Nucleus ***53****, No. 3 (2016) 200-207*

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)

ISSN 2306-6539 (Online)

Improved Scalable Recommender System

S. Ishtiaq, N. Majeed, M. Maqsood* and A. Javed

Department of Software Engineering University of Engineering and Technology, Taxila, Pakistan

samia.ishtiaq92@yahoo.com; {nadeem.majeed, muazzum.maqsood, ali.javed}@uettaxila.edu.pk

ARTICLE INFO

Article history :
Received : 20 April, 2016
Revised : 24 August, 2016
Accepted : 19 September, 2016

Keywords:
Recommender system
Clustering
Centroid selection
Scalability

ABSTRACT

Recommender systems are known for their ability to recommend items which are new to the user by having some synchronization with user's personal interest. The importance of recommender systems leads to the creation of new approaches that can produce accurate results. As data became large it results in scalability issues. In this work, we have suggested a scalable technique using different methods that work in a sequential manner. A novel centroid selection for clustering based recommender system is proposed. SVD and user representatives are used to handle scalability issues. Experiments on proposed approach with standard datasets showed great improvement in scalability and slight better accuracy.

1. Introduction

Development of Recommender System (RS) has revolutionized the information technology. Recommender Systems are self-learning routines that study user's interest, crawl across available data to find new items. RS estimates about whether or not the user will like the new item and give suggestions on the basis of estimation results. These systems not only abolish need of searching for the item but also recommend the required item and help in decision-making [7]. However, RS also suffer from problems of sparsity, scalability, first-rater problem and unusual user problem. These issue effect the performance of RS and sometimes result in poor recommendation accuracy. To fulfill user expectations different types of recommendation techniques have been applied in RS which includes content based RS , collaborative filtering, and hybrid RS.

Content-based RS takes items contextual information into account to recommend users with items that resemble what they like in the past [4]. This approach describes the item as a set of its textual features. User profiles are made up of description about items that user likes or user past history which includes item purchase behavior and ratings. News Weeder [2] and Pandora are examples of such recommender systems. This technique works well when item description is simple. But such systems face issues of cold start user and poor data description.

Collaborative filtering (CF) works by finding users with similar taste and then recommend the new item to a user based on ratings of his similarity group [3, 10, 23]. This approach works on special matrix known as a user-item matrix that contains a rating of items provided by different users. To suggest new item systems works by calculating the distance of between users to find out neighbors. Ratings of neighbors are used to calculate possible rating value of the new item for the active user. Collaborative filtering can be divided into two types 1) Memory-based [23] 2) Model-based [24]. Amazon [25] and Ringo are examples of collaborative filtering systems. Such algorithm suffers from sparsity issues, cold start user and cold start product [6].

Both the above mention recommendation approaches have some limitations. To maximize recommendation scope and minimize its limitations, researchers established optimal solution known as Hybrid RS [5]. Hybrid RS are systems that combine multiple recommendation approaches in a certain way that guarantees high performance. Different techniques of hybrid RS has been proposed including Switching, Mixed, Feature Augmentation, Cascade and Meta-level [12].

Scalability is one of a major issue in RS as described above. As information grows in terms of user data or item data, needs to perform calculations also increases. Most of the recommendation algorithm perform well with small data but are not capable of coping with data growth which reduces performance capabilities of RS.

This paper presents a novel scalable recommendation approach that produces recommendation with high accuracy and scalability. Algorithm combined different scalability techniques minimizing processing requirement to produce recommendation with a large amount of data.

* Corresponding author

A system implemented latest centroid selection method which is then followed by dimensionality reduction methods and user representatives. The paper also has a comparison of system accuracy with previous approaches. Movie lens dataset is used for this work.

Remaining paper is organized as follows. We start in section 2 by giving some overview of previous work in the field of RS. In Section 3 we explained problem background, in section 4 we explained our proposed methodology. Section 5 explains the results and discussion followed by a conclusion and future work in Section 6.

2. Related Work

Recommender system can be divided into three major subtypes. Content Based recommendation [26], Collaborative Filtering [24] and Hybrid recommendation [12].A survey has been conducted to review all existing techniques of recommender system [5].

Content-based recommendation approach works on the concept that users can possibly like resembling items in the future as they have preferred in the past [27]. News recommender system followed access behavior of users, use it as implicit feedback that is finally used by CB algorithm [29]. A system has been proposed using CB for twitter based recommendation that made use of two features popularity and activity which show a slight improvement in performance [28]. Various other systems implied different CB techniques including heuristic [30], Linear classifier [31] etc. Two major problems with CB as observed in most of the systems are the inability to express well-grounded information and over-specialization [24].

Collaborative filtering digs to find users having similar taste [24, 32, and 33]. One way to calculate similarity is by k nearest neighbor [24]. A new measure for similarity known as Proximity Impact Popularity has been proposed [15]. Improved Pearson correlation called weighted Pearson correlation coefficient has been proposed [16] to resolved problem of traditional Pearson correlation. A system has been proposed that uses JMSD matrix which incorporates numerical as well as non-numerical data for rating [34]. Researchers proposed a new way of making are commendation by using trust factor between to user in combination to item-based CF [46]. Another process known as Pareto dominance has been proposed that select user representation is used as nearest neighbors [35]. However finding similarity using KNN has a major issue of scalability [36]. CF does not need any description about user or item to make recommendations. CF faces problems of cold start item [37, 38] and cold start user [39]

Survey has been done based on various techniques of a hybrid recommender system. The way they are implemented and their comparison [12]. The system proposed a hybrid RS technique known as hydra that combined content based RS and collaborative filtering into unified model [13]. SVD approach is then used for factorization of hybrid RS. The approach, however, does not resolve the cold start problem. Worked has been done by Combining item based collaborative filtering and content-based by clustering item based content and user ratings [22]. The main focus of this approach is cold start problem. Another system explained hybrid technique formed by combining neural network and collaborative filtering [17]. Researchers have proposed a combination of content and collaborative filtering using unified Boltzmann Machines to improve accuracy and prediction [19]. Different combinations of collaborative filtering algorithms were used including SVD, Restricted Boltzmann Machine, Global Effects, Asymmetric Factor Model and Neighborhood Based Approaches [20]. The issue with this recommender system is a lack of ability to resolve cold start. Also, this approach has large training time. The system was proposed that combined CF and SOM neural networks to form hybrid system [18]. Another system combined collaborative filtering, content-based recommendation and demographic filtering [27]. Researchers worked on hybrid approach using content-based approach to making improvements in data and then applying collaborative filtering to make recommendation process better [21]. However, this approach is not scalable.

To address scalability issue in recommender systems clustering approach is widely used. K-means clustering is a most common technique used by many systems [40-42]. The problem with simple k-means is random initial centroid selections. Bradley proposed the idea of selecting initial centroid to minimize the scalability problem [43]. An algorithm has been proposed known as k-means ++ that produces high-quality clusters using a probabilistic approach to select initial centroids [44]. Paper [45] was based on a comparison of clustering algorithms and results showed that k-mean ++ is better as compared to other clustering approaches. Researchers worked on various algorithms to improve clustering technique based on new centroids selections methods [47].

A solution to improve the scalability has led to the creation of SVD. SVD is abbreviation singular value decomposition .A way to factorize matrix which can reduce the data dimensionality [9, 14]. For information retrieval purpose LSI (Latent Semantic Indexing) used SVD to address issues in polysemy and synonymy [48]. SVD was widely used to improve scalability. Work has been done to improve prediction by using SVD and neural networks [49]. This approach used SVD to make CF as a classification problem. Its output was submitted to artificial neural networks algorithm which can be prepared to make better predictions. SVD was also used

in different RS by Group Lens. A comparative study has been done to find out results of prediction generated by SVD and simple CF [50]. Two different types of experiments were carried out. First experiments showed a comparison between SVD and CF in terms of perdition effectiveness. The second experiment showed a comparison of effectiveness between SVD and CF while generation Top-N predictions. Another on SVD and CF has been done [11] that applied both of these approaches in a sequential manner. First, SVD was applied to reduce the dimensions of input data. Then user-based CF was applied on reduced dimensionality. SVD was used to improve scalability

3. Problem Background

Let suppose we have x number of users that are represented by notation m and M represents the whole set of users as $M = \{m_1, m_2, m_3, \ldots . m_x\}$. Item in dataset is denoted by n and set of items are represented by $N = \{n_1, n_2, n_3, \ldots \ldots n_y\}$ where y is a total number of items. Rating of user i for an item j is represented as r_{m_i,n_j} .

3.1 KMeans^{PlustLogPower}

KMeans^{PlustLogPower} [47] algorithm describes a new method of centroid selection as initial centroid selection greatly affects clusters outcome. KMeans^{PlustLogPower} made use of distances and numbers of ratings as two bases for centroid selection. Centroid selection in KMeansPlus^{LogPower} rely on the concept that any new centroid selected should have huge distance with existing centroids and probability proportional to the log of similarity. KMeansPlus^{LogPower} requires a number of clusters as input. KMeansPlus^{LogPower} *the* algorithm works by taking power user as the first centroid where power user is described as a user with a maximum number of ratings. Utilizing power user next centroid is selected based on probability as given by

$$\text{Prob} = \text{dist(u)} + \log\left[\frac{1}{\text{p(x)}} + 1\right] \quad (1)$$

Equation 1 describes the formula for calculating the probability based on the distance between users and number of ratings. In Eq. 1 dist(u) is used to find out the distance between the active user and power user.

$$\text{dist} = \begin{cases} \frac{1}{\text{sim}} & \text{if sim} \neq 0, \\ \text{MAX}_{\text{DIST}} & \text{otherwise} \end{cases} \quad (2)$$

In Eq. 2 distance is defined as the inverse of similarity, considering the fact that greater the distance between two points lower will be the similarity and vice versa. As results of Pearson correlation for calculating similarity can be negative, however distance cannot be negative. So to avoid this confusion a factor of 1 is added to each Pearson correlation similarity. If similarity is 0 then MAX_{DIST} is used where MAX_{DIST} is described as maximum distance that can be obtained between two data points. In Eq.1 p(x) is ratio between total ratings given by active user and power user calculated as

$$p(x) = \frac{|I_u|}{|I_{u_p}|} \quad (3)$$

In Eq. 3 I_u and I_{u_p} represent number of ratings by user u and u_p.After finding centroids equal to number of number of clusters, users are grouped in different clusters based on their distance with centroids. Once clusters are formed iterations are carried out. In each iteration mean is calculated using all data points of clusters and centroids is updated. Again clusters are formed. This process continues until no data point is shifted in new cluster or number of iteration become equal to iteration input. Result of this approach is set of clusters.

3.2 SVD

Singular Value Decomposition commonly termed as SVD is known for reducing dimensionality. The concept of SVD is based on mathematical law according to which any rectangular matrix can be shown as a product of its orthogonal matrix, diagonal matrix, and the transpose of an orthogonal matrix.

$$A_{mn} = U_{mm} S_{mn} V_{nn}^T \quad (4)$$

Where A is an original matrix with dimension m*n. U is an orthogonal matrix with dimension m*m. S is a singular matrix of dimension m*n. V is an orthogonal matrix with transpose and n*n dimension.

To reduce the dimension of the initial input matrix, the low-rank approximation is used that is calculated by keeping only starting K diagonals of matrix S and by deleting $r - k$columns from matrix U and $r - k$ rows from matrix V. Thus producing resultant matrix as shown in next equation.

$$A_{mn} = U_{mk} S_{kk} V_{kn}^T \quad (5)$$

Or we can simply write as $A_k = U_k S_k V_k^T$. Using these matrixes prediction about rating of user u for item i is given as

$$r_{i,u} = U_k . \sqrt{S_k}^T(u) . \sqrt{S}_k . V_k^T(i) \quad (6)$$

Recommendation produced by SVD is mainly affected by the type of imputation used. Most common imputation techniques are filled by zero, fill by random numbers, fill by user average, fill by item average etc.

3.3 User Representative

User representative is a simple concept where a set of users are used to represent data they are associated to. We have used a simple approach to select the user representatives. Considering the fact that users who had rated a large number of items have more knowledge of

items, we have selected power users as user representatives.

4. Proposed Methodology

We proposed a new scalable recommender system with improved scalability and accuracy. Collaborative filtering works by taking utility matrix as a base factor for estimating prediction about the item for given user. But as data increases in size time to process utility matrix also increases. To solve this issue different approaches are used in a sequential way. First $\text{KMeansPlus}^{\text{LogPower}}$ is applied to data which is the latest centroid selection technique [47]. Inserting results of clustering to SVD which is then followed by user representative outcomes as scalability improving technique.

Algorithm starts by taking a number of clusters as input, which in return tells about a number of initial centroids to be formed by the procedure. The first centroid is formed by selecting power user, which is defined as a user with a maximum number of rating in the whole dataset. After the first centroid, all other remain centroids i-e(k-1) are selected based on their probability as given in Step 4 until all centroids are formed. Next procedure Cluster is used for making clusters of given data based on centroids calculated in the previous method. This procedure takes user train data, the number of clusters and iterations as input. Procedure cluster basically associates each user of the dataset to its near centroid based on their similarity as calculated in step 12. Once the cluster is formed the centroids are updated using an average of rating provided by all users as shown in step 13. Updating centroids lead to clustering again. Step 12 to 14 keep on looping until no change in clusters is observed or we have reached to a maximum number of iterations. The result of this procedure is set of clusters.

Algorithm: $\boldsymbol{Rec_{KSU}}$

Input: User item matrix

Output: Set of recommendations

procedure $\text{KMeansPlus}^{\text{LogPower}}$ (no. of cluster)

Select initial centroid c_1 to be a power user u_p

repeat

Select the next centroid c_i where $c_i = \acute{u} \epsilon U$ with the probability

$$\text{Prob} = \text{dist}(u) + \log\left[\frac{1}{p(x)} + 1\right]$$

until k centroids are found

return $\{c_1, c_2, c_3, \ldots, c_k\} \triangleright$ k centroids

end procedure

procedure Cluster (u, k, iteration)

C = Centroid Select

a=0

repeat

Set the cluster g_j for each $j \epsilon 1,2, \ldots \ldots, k$ to be the set of users in U that is closer to $c_j$than they are to c_i for all $l \neq j$.

Set c_j, for each $j \epsilon 1,2, \ldots \ldots, k$ to be the center of mass of all users in g_j, i.e. $c_j = \frac{1}{|g_j|}\sum_{u \epsilon g_j} u$

$$a = a + 1$$

until (C changes no more) OR ($a =$ iteration)

return(C)

end procedure

if(C is large) **then**

repeat

SVD(cluster C_i)

until clusters are reduced

procedure User Representative(cluster C_i)

$$U^{\text{power}} = \{u_1, u_2 \ldots . u_f\}$$

return$\{u_1, u_2 \ldots . u_f\}$

end procedure

procedure Recommend

find out cluster number of active user

find set of user representative for resulted cluster

Use the average of rating for targeted item as given by user representatives.

end procedure

else

Use average user rating for given user

end if

Cluster produced are then checked based on a number of users they have. If the cluster is large then it is sent for dimensionality reduction by SVD in step 20. This step is repeated for all large clusters. Matrix resulted from SVD is then used to find outset of power users in step 23 which are considered as user representatives. Average rating of these user representatives is used for recommending the active users. But if on the other hand, our cluster has very small number of users which may mean that these users have much different taste than other users in large clusters. So in order to give is commendation to those users, we will user average rating as shown in step 31.

5. Results and Discussion

5.1 Dataset

For this research work, we have used Movie Lens datasets that are publically available named as Movie Lens 100 K Ratings and Movie Lens 1 M Ratings. Movie Lens 100 K Rating dataset contains ratings about 1682 movies given by 943 users. A total number of ratings available in the dataset are 100000. While Movie Lens 1

M Rating dataset has ratings of almost 3900 movies provided by 6040 users and 1000,000 ratings. Ratings for both datasets are on a numerical scale ranging from 1 to 5 where 1 is considered as lowest and 5 is considered as highest. To find the sparsity of given datasets we have used simple formula $(1 - \frac{\text{non-zeros entries}}{\text{total number of ratings}})$. So $\left(1 - \frac{100000}{1586126}\right) = 0.93$, which means Movie Lens 100 K Rating dataset is 93% sparse.

5.2 *Results*

To check the performance of our proposed algorithm we have used Movie Lens 100K dataset and Movie lens 1 M dataset. For experiments on both datasets, we have partitioned the data into training data and test data. Using 5 fold cross-validation, 20% of original data for each user is used as test set while remaining data is used as training set.

We have done comparisons of our technique in terms of MAE with four other techniques including user-based CF via default voters, simple K-Means, $\text{KMeansPlust}^{\text{ProbPower}}$. $\text{KMeansPlust}^{\text{LogPower}}$,Results have shown that our proposed approach Rec_{KSU} produce results with slight improvement in MAE.

Table 1: Comparison of proposed technique with previous ones for MAE and coverage

Algorithm	**MAE(100 K)**	**MAE (1 M)**
UBCF_{DV}	.721	.766
KMeans	.745	.863
$\text{KMeansPlust}^{\text{LogPower}}$	.740	.852
Rec_{KSU}	**.710**	**.834**
$\text{KMeansPlust}^{\text{LogPower}}$	.738	.854

Reduction in MAE clearly shows that our technique can produce better quality recommendations even with improved scalability when tested on SML.

5.3 *Optimal Number of Cluster*

A number of clusters initially set have a huge influence on the output of clustering.$\text{KMeansPlus}^{\text{LogPower}}$ When checked on various clusters size, gives optimal result when cluster number is large.

MAE is quite large when tested on 10 number of clusters. However, MAE tends to fall gradually with slight variation as the number of clusters is increased. While testing on 130 clusters MAE is found to be very low [47].

However, when we used this technique in our approach we found out that having large number cluster results in small data for each cluster while reducing the number of clusters we may get large data in each cluster but then the similarity between data is questionable.

Keeping in mind both of the above-mentioned problems we have to find an optimal value that can gather enough data in each cluster while keeping the most similar points in each cluster. So our approach works best when a number of clusters are 100 for 100 K ratings with MAE of .723 and 120 clusters for 1 M rating dataset with MAE of .839 based on recommendation generated for users in large clusters only. One thing should be worth noting here that MAE result produced by 100 and 120 clusters is tested along with SVD and user representative approach.

Table 2: Results of MAE for different number of clusters

Number of clusters	Rcm KSU (100 K)	Rcm KSU(1 M)
60	.735	.844
80	.735	.841
100	**.723**	.840
120	.725	**.839**
140	.729	.841
160	.732	.844
180	.739	.844
200	.745	.845

5.4 *Optimal number of Neighbors*

Experiments on $\text{KMeansPlus}^{\text{LogPower}}$ for a number of neighbors showed that with an increase in neighbor size MAE is decreased as shown in Figs. 1 and 2.

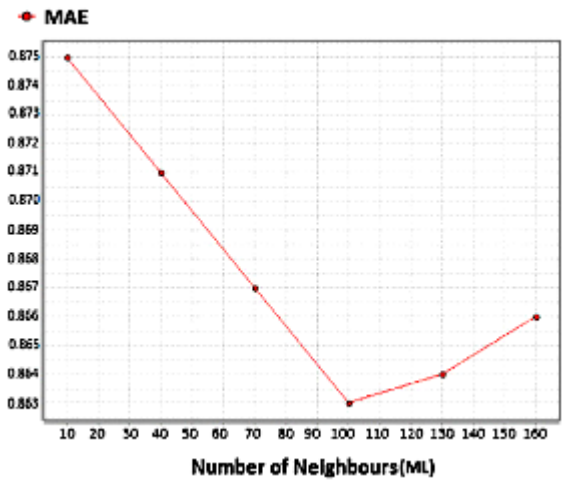

Fig. 1: Result of MAE for different number of neighbors for ML dataset [47]

The graph clearly shows that MAE is high when tested with 10 neighbors which decrease as more neighbors are involved. Results of MAE are found to decrease at neighbor size 30 and 100 for dataset 100 K and 1 M which is still very large and requires a large computational cost.

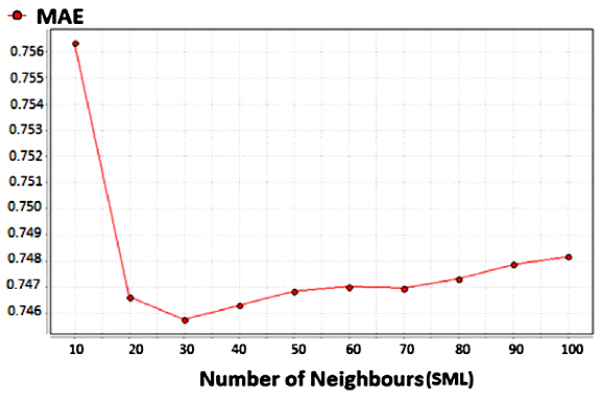

Fig. 2: Result of MAE for different number of neighbors for SML dataset [47]

However, as we have used user representatives due to which our processing cost is very low as our technique does not require to calculate neighborhood for each user at the run time. We have tested on a different number of user representatives to find the optimal value. Results are shown in table 3. As we can see that with an increase in user representatives MAE is increased. The reason behind is that as a number of user representatives are increased less cluster will fall in large cluster category and that large cluster may have low similarity due to which selected user representatives are not able to work well for each user. A number of clusters and user representative are dependent on each other. For a number of clusters other than our optimal value, a different number of user representatives will be used.

Table 3: Result of MAE with different number of user representatives

Number of user representative	MAE(100 K)	MAE (1 M)
5	0.729	0.850
6	0.738	0.843
7	0.764	0.839
8	0.79	0.839
9	0.831	0.841
10	0.832	0.849

5.5 Optimal number of dimension for SVD

SVD when applied on 100 K and 1 M dataset with different dimensions gives optimal results on a different number of a dimension based on the type of imputation used. Complete results of SVD on both datasets are shown in Table 4.

However, when applied SVD in our approach following clustering and finally user representatives. As we have to differentiate between large and small clusters to produce recommendation using a different approach. We have selected value 5 and 8 as a threshold value for 100 K and 1 M dataset and set the remaining parameters on its base. So we have tested dimension number from 1 to 8 and found that MAE is reduced at 5 dimensions for 100 K ratings. We cannot test dimensions above the threshold as some of our large clusters may have total data points equal to the threshold. At this point, one can clearly state that as there are three different techniques used in a sequential way so parameters of each of these dependent on others.

Table 4: Result of MAE with different imputations and number of dimensions on SML dataset

Imputed technique	MAE (100 K)	Number of dimensions(100 K)	MAE (1M)	Number of dimensions (1 M)
Zeros	2.32	12	2.40	26
Random	1.072	4	1.09	17
User average	.778	8	.759	22
Item average	.774	10	.730	22

Table 5: Result of MAE for different number of dimensions using Rcm KSU

Number of dimensions	Rcm KSU(100 K)	Rcm KSU(1 M)
1	.80	.94
2	.78	.92
3	.76	.91
4	.74	.89
5	.728	.88
6	.74	.87
7	.75	.85
8	.76	.839

In short, our approach gives a result with a large improvement in scalability and MAE while using a small number of neighbors, clusters, and reduced dimensions. For small clusters we have used average user ratings so overall MAE for whole 100 K test set is found to be .71 using our approach. While for 1 M rating dataset overall MAE for all clusters is .734.

6. Conclusion and Future Work

In this paper, we have proposed a scalable collaborative recommender system. The collaborative recommender system is based on k-mean clustering which itself improve the scalability issues. A novel centroid selection technique is used to improve k-mean clustering. Our technique with a combination of different scalability approaches resulted in an increase in accuracy of recommendation and large scalability improvement. The experiment of the standard dataset is done in order to show results that show the improved performance of the system. For future work, we will try to test our technique

on more standard datasets and also replace user average for small clusters with some other technique.

Reference

[1] Burke and Robin. "Hybrid web recommender systems, "The adaptive web.Springer Berlin Heidelberg, pp. 377-408, 2007.

[2] K. Lang, "Newsweeder: Learning to filter netnews", Proc. of the 12th Int. Conf. on Machine Learning, vol. 12, pp. 331-339, ICML 1995.

[3] Vozalis, Emmanouil and K.G. Margaritis, "Analysis of recommender systems algorithms", Proc. of the 6th Hellenic European Conf. on Computer Mathematics and its Applications (HERCMA-2003), Athens, Greece. pp. 732-745, 2003.

[4] M. Pazzani and D. Billsus of Part, "Content-based recommendation systems", The Adaptive Web, P. Brusilovsky, Berlin: Springer, vol. 4321, pp. 325–341, 2007,.

[5] M. Rehman and T. Ahmad. "Optimized k-Nearest Neighbor Search with Range Query", The Nucleus, vol. 52, no. 2, pp. 45-49, 2015.

[6] Zou, Haitao, et al. "TrustRank: a cold-start tolerant recommender system", Enterprise Information Systems, vol. 9.2, pp.117-138, 2015.

[7] Resnick, Paul and Hal R. Varian. "Recommender systems", Communications of the ACM, vol. 40.3, pp. 56-58, 1997.

[8] B. Mobasher, "Recommender systems", KunstlicheIntelligenz, Special Issue on Web Mining, vol. 3, pp. 41–43, 2007.

[9] Azar and Yossi et al., "Spectral analysis of data", Proc. of the 33rd Annual ACM Symposium on Theory of Computing, pp. 619-626, 2001.

[10] Goldberg and David et al., Using Collaborative Filtering to Weave an Information Tapestry", Communications of the ACM, vol. 35, no. 12, pp. 61-70, 1992.

[11] Sarwar, "Sparsity, scalability, and distribution in recommender systems", Ph.D. thesis, University of Minnesota, 2001.

[12] Burke and Robin, "Hybrid recommender systems: Survey and experiments", User Modeling and User-adapted Interaction, vol. 12, no. 4, pp. 331-370, 2002.

[13] Spiegel, Stephan, JérômeKunegis and Fang Li, "Hydra: a hybrid recommender system [cross-linked rating and content information]", Proc. of the 1st ACM Int. Workshop on Complex Networks Meet Information &Knowledge Management. ACM, pp. 75-80, 2009.

[14] M.E. Wall, A. Rechtsteiner and L.M. Rocha, "Singular value decomposition and principal component analysis", A Practical Approach to Microarray Data Analysis, Daniel P. Berrar: Springer US , vol. 2003, pp. 91-109, 2003.

[15] Ahn and Hyung Jun, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem", Information Sciences, vol. 178, no. 1, pp. 37-51, 2008.

[16] J.L. Herlocker et al., "An algorithmic framework for performing collaborative filtering", Proc. of the 22nd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 230-237, 1999.

[17] Ssiliou and Charalampos, et al., "A recommender system framework combining neural networks &collaborative filtering", Proc. of the 5th WSEAS Int. Conf. on Instrumentation, Measurement, Circuits and Systems, World Scientific and Engineering Academy and Society (WSEAS), pp. 285-290, 2006.

[18] Lee, Meehee, P. Choi and Y. Woo. "A hybrid recommender system combining collaborative filtering with neural network", Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems, pp. 531-534, Springer Berlin Heidelberg, 2002.

[19] Gunawardana, Asela and C. Meek. "A unified approach to building hybrid recommender systems", Proc. of the 3rd ACM Conf. on Recommender Systems, pp. 117-124. ACM, 2009.

[20] Jahrer, Michael, A. Töscher, and R. Legenstein, "Combining predictions for accurate recommender systems", Proc. of the 16th ACM SIGKDD Int.Conf. on Knowledge Discovery and Data Mining, pp. 693-702, 2010.

[21] Melville, Prem, R.J. Mooney and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations", Aaai/iaai, pp. 187-192. 2002.

[22] Li, Qing and B.M. Kim, "An approach for combining content-based and collaborative filters", Proc. of the 6th Int. Workshop on Information Retrieval with Asian Languages, vol. 11, pp. 17-24, 2003.

[23] Shardan, Upendra and P. Maes, "Social information filtering: algorithms for automating "word of mouth", Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, pp. 210-217, ACM Press/Addison-Wesley Publishing Co., 1995.

[24] Adomavicius, Gediminas and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions", IEEE transactions on knowledge and data engineering 17, vol. no. 6, pp. 734-749, 2005.

[25] Linden, Greg, B. Smith and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering", Internet Computing, IEEE, vol. 7, no. 1, pp. 76-80, 2003.

[26] Balabanović, Marko and Y. Shoham, "Fab: content-based, collaborative recommendation", Communications of the ACM, vol.40, no. 3, pp. 66-72, 1997.

[27] Pazzani and J. Michael, "A framework for collaborative, content-based and demographic filtering", Artificial Intelligence Review, vol. 13, pp. 393-408, 1999.

[28] Garcia, Ruth and Xavier Amatriain. "Weighted content based methods for recommending connections in online social networks", Workshop on Recommender Systems and the Social Web, pp. 68-71, 2010.

[29] Billsus, Daniel, M.J. Pazzani and J. Chen, "A learning agent for wireless news access", Proc. of the 5th Int. Conf. on Intelligent user Interfaces, pp. 33-36. ACM, 2000.

[30] Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. Modern information retrieval, vol. 463. New York: ACM press, 1999.

[31] Harvard Joachims and Thorsten, "Text categorization with support vector machines: Learning with many relevant features", European Conf. on Machine Learning, pp. 137-142, 1998.

[32] J.L. Herlocker, J.A. Konstan, J.T. Riedl and L.G. Terveen, "Evaluating collaborative filtering recommender systems", ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5–53,2004.

[33] Su, Xiaoyuan and T.M. Khoshgoftaar, "A survey of collaborative filtering techniques", Advances in Artificial Intelligence, vol. 2009, pp. 4-23, 2009.

[34] Bobadilla, Jesús, Francisco Serradilla and Jesus Bernal. "A new collaborative filtering metric that improves the behavior of recommender systems." Knowledge-Based Systems, vol. 23, no. 6, pp. 520-528, 2010.

[35] Ortega and Fernando et al., "Improving collaborative filtering-based recommender systems results using Pareto dominance", Information Sciences, vol. 239, pp 50-61, 2013.

[36] Luo, Xin, Y. Xia and Q. Zhu, "Incremental collaborative filtering recommender based on regularized matrix factorization", Knowledge-Based Systems, vol.27, pp. 271-280, 2012.

[37] Park, Seung-Taek and W. Chu, "Pairwise preference regression for cold-start recommendation", Proc. of the 3rd ACM Conf. on Recommender Systems, pp. 21-28. ACM, 2009.

[38] Park, Yoon-Joo and A. Tuzhilin, "The long tail of recommender systems and how to leverage it", Proc. of the 2008 ACM Conference on Recommender Systems, pp. 11-18, 2008.

[39] du Boucher-Ryan, Patrick, and D. Bridge", Collaborative recommending using formal concept analysis", Knowledge-Based Systems, vol. 19, no. 5, pp. 309-315, 2006.

[40] Sarwar, M. Badrul, G. Karypis, J. Konstan and J. Riedl. "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering." In Proceedings of the fifth international conference on computer and information technology, vol. 1, pp. 128-134, 2002.

[41] Xue, Gui-Rong, C. Lin, Q. Yang, W. Xi, H. J. Zeng, Y. Yu and Z. Chen. "Scalable collaborative filtering using cluster-based smoothing", Proc. of the 28th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 114-121. 2005.

[42] A.M. Rashid, S.K. Lam, G. Karypis and J. Riedl, "ClustKNN: A highly scalable hybrid model-& memory-based CF algorithm, Proc. of WebKDD, 2006.

[43] P.S Bradley and U. M. Fayyad, "Refining initial points for K-means clustering", ICML, vol. 98, pp. 91-99, 1998.

[44] Arthur, David and S. Vassilvitskii, "k-means++: The advantages of careful seeding", Proc. of the 18th Anual ACM-SIAM Symp. on Discrete Algorithms, pp. 1027-1035, 2007.

[45] Shindler and Michael of Part, "Approximation algorithms for the metric k-median problem", Efficient Approximation and Online Algorithms, E. Bampis, Berlin: Springer, vol. 2006, pp. 292-320, 2008.

[46] Jamali, Mohsen and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation", Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 397-406. 2009.

[47] Zahra and Sobia et al., "Novel centroid selection approaches for KMeans-clustering based recommender systems", Information Sciences, vol. 320, pp. 156-189, 2015.

[48] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, Indexing by latent semantic analysis Journal of the American Society for Information Science, vol. 41, no. 6, pp. 391–407, 1990.

[49] Billsus, Daniel and M.J. Pazzani, "Learning Collaborative Information Filters", Icml, vol. 98, pp. 46-54. 1998.

[50] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender systems – A case study". Proc. of the ACM WebKDD Workshop, vol. 2, pp. 212-224, 2000.