# User Relevance for Item-Based Collaborative Filtering

R. Latha, R. Nadarajan

**HAL Id: hal-01496080**
**https://inria.hal.science/hal-01496080**

Submitted on 27 Mar 2017

# User Relevance for Item-based Collaborative Filtering

R. Latha, R. Nadarajan

Department of Applied Mathematics and Computational Sciences,
PSG College of Technology, Coimbatore
Tamil Nadu, India
lathapsg@yahoo.co.in, nadarajan_psg@yahoo.co.in

**Abstract.** A Collaborative filtering (CF), one of the successful recommendation approaches, makes use of history of user preferences in order to make predictions. Common drawback found in most of the approaches available in the literature is that all users are treated equally. i.e., all users have same importance. But in the real scenario, there are users who rate items, which have similar rating pattern. On the other hand, some users provide diversified ratings. We assign relevance scores to users based on their rating pattern in order to improve the quality of predictions. To do so, we incorporate probability based user relevance scores into the similarity calculations. The improvement of predictions of benchmark item based CF approach with the inclusion of user relevance score is demonstrated in the paper.

**Keywords:** Collaborative filtering, Recommendation System, Information Retrieval, User Relevance

## 1 Introduction

With the tremendous growth of Web, volume of data available in net based systems become large and thus results in large repositories. So it becomes too difficult for individuals to handle the data effectively and efficiently. This scenario is commonly referred to as 'information overload' problem. Recommender system addresses the problem [6]. Many online business systems such as Amazon.com and Netflix are using recommender systems to provide personalized suggestions.

Two common approaches are available for making recommendations. They are Collaborative Filtering (CF) [3] and Content Based filtering (CB) [8]. Pazzani, et. al [8] defines that content-based algorithms base their recommendations on the contents of items and profiles of users. The profiles allow programs to associate users with matching products. Content Based Filtering uses the assumption that items with similar objective features will be rated similarly and users with similar profile/taste will prefer items in a similar manner. CB approaches are specific to a domain and the scope of the approach is limited to the domain for which they are proposed.

Collaborative Filtering (CF) is the most popular and successful approach for recommendation systems. CF relies only on the past user behavior (ratings, preferences, purchase history, time spent etc) [6]. Breese et. al [3] classifies Collaborative Filtering techniques into two categories: Memory-based and Model based techniques. Memory based algorithms predict the rating of users using the previously rated items by the users and other users who have similar tastes. They operate over the entire user database to make predictions. The most common memory-based models are based on the notion of nearest neighbors, using a variety of distance measures [8]. Model based systems are based on a compact model inferred from the data. We have considered model based CF approach for our work.

In Collaborative Filtering techniques, all users are treated uniformly.   But in the real scenario, there are some users who are consistent in providing preferences, where as some users provide ad hoc preferences which do not form any pattern. Ultimately the former kind of users must be assigned higher score, and so their preferences must be given higher importance. We assign scores to users based on their rating pattern in order to improve the quality of predictions. We assign probability based relevance score to users.

The contribution of the proposed work can be divided into two phases, namely offline phase and online phase. In offline phase user relevance scores are calculated and a model is built on relevance scores.   In on line phase rating predictions are made based on the model created.

The rest of this paper is organized as follows.

Section 2 presents the overview of item based CF and user relevance problem, Section 3 discusses about the proposed approach to address the problem. Section 4 presents user relevance model building and Section 5 discusses about experimental evaluations of our approach. MovieLens [1] database is used for proving the results. Finally, Section 6 provides some concluding remarks and an outline of the future research.

## 2      Overview of Item Based CF and User Relevance Problem

In this section we discuss about Item based CF techniques and user relevance problem.

### 2.1      Item Based CF

To address the scalability concerns of user-based recommendation techniques, item-based recommendation techniques (also known as model-based) have been developed [9]. These techniques compute similarity between items, and then use these similarity values to compute top-N recommendations or make predictions. The reason behind these techniques is that a customer will more likely purchase items that are similar or related to the items that he/she has already purchased. These approaches are faster than user based approaches, since the similarity computations can be done offline which results in faster recommendation engines.

Different kinds of similarity measures are available in the literature for computing relationship between items. In [5], the author has discussed about cosine similarity, adjusted cosine similarity and Pearson correlation coefficient.

In Cosine based approach [4] the similarity between items $i$ and $j$ is calculated as cosine of the angle between them. Only the items which are rated by both the users will be considered for computing the angles. The similarity $Sim_{i,j}$ is defined as

$$Sim_{i,j} = \frac{\sum_{u \in u(i) \cap u(j)} r_{u,i} \times r_{u,j}}{\sqrt{\sum_{u \in u(i) \cap u(j)} r_{u,i}^2} \sqrt{\sum_{u \in u(i) \cap u(j)} r_{u,i}^2}} \quad (1)$$

where $u(i)$ represents set of users who have rated for the item $i$ and $r_{u,i}$ represents the ratings given by the user $u$ for the item $i$. According to B. Sarwar et.al, [5] Cosine similarity does not account for the difference in user ratings. Adjusted Cosine Similarity is suggested to overcome the problem, in which the average rating of the user is subtracted from his actual ratings. The function used to calculate the similarity between item $i$ and item $j$ is given below:

$$Sim_{i,j} = \frac{\sum_{u \in u(i) \cap u(j)} (r_{u,i} - \bar{r_u}) \times (r_{u,j} - \bar{r_u})}{\sqrt{\sum_{u \in u(i) \cap u(j)} (r_{u,i} - \bar{r_u})^2} \sqrt{\sum_{u \in u(i) \cap u(j)} (r_{u,j} - \bar{r_u})^2}} \quad (2)$$

where $u(i)$ represents the users who have rated for the item $i$ and $r_{u,i}$ represents the ratings given by user $u$ for item $i$. $\bar{r_u}$ is the average rating of user $u$. There are many ways to compute predictions [4]. Weighted sum is a commonly used approach for predicting unknown ratings which is calculated as given in (3)

$$p_{u,i} = \frac{\sum_{j \in S(i)} Sim_{i,j} * r_{u,i}}{\sum_{j \in S(i)} |Sim_{i,j}|} \quad (3)$$

Here S(i) is the set of items that are similar to $i$, the item to be predicted. $p_{u,i}$ is the prediction for item i for the user u. Although CF techniques take different kinds of input, we focus on the input in the form of *m x n* user item matrix.

**Definition 1. User-item Matrix R**

If there are *m* users who have given ratings for *n* items, then the ratings data can be represented as an *m x n* matrix with rows representing users and columns representing items. The matrix is called user-item matrix *R*. Each element $R_{u,i}$ is an ordinal value which ranges from $R_{min}$ to $R_{max}$. Unrated values are considered to be zero. A sample rating matrix *R* is shown in Table 1.

## 2.2 User Relevance in Item Based CF

In Collaborative Filtering techniques we discussed above, the importance of users is not taken into account. There are some users who prefer items with similar rating

pattern. i.e, if a user rates an item and most of its similar items, then the user is consistent in rating. By assigning higher weight to them brings accuracy in recommendations. On the other hand there are some users who rate items which do not have any similar rating pattern. They rate dissimilar items. Such users are divergent in thinking and assigning higher weight to them improves diversity in recommendations. But current recommender systems treat all users uniformly.

Min Gao et. al [11] proposed a technique to assign rank to users based on popularity and use the rank values in similarity calculations. The technique improves accuracy of predictions.

Our approach proposed in this paper is similar to the work described in [11], but instead of user rank, user relevance score is calculated. Moreover the proposed technique improves accuracy as well as diversity of recommendations.

Despite of much work available in the literature to improve quality of predictions, up to our knowledge no other work is available in the literature to assign weights to users and treat them differently.

**Table 1.** Original rating Matrix R

|     | I1 | I2 | I3 | I4 | I5 | I6 |
|-----|----|----|----|----|----|----|
| U1  | 1  | 2  | 0  | 5  | 0  | 0  |
| U2  | 0  | 3  | 0  | 0  | 5  | 0  |
| U3  | 0  | 0  | 5  | 0  | 0  | 2  |
| U4  | 0  | 1  | 0  | 0  | 0  | 0  |

## 3      Probability based Relevance Score for Users

This section describes the procedure for building a model for assigning relevance score for the users. To calculate the relevance score for users we apply a probability based technique. The model building takes two phases namely item-item similarity calculation phase and relevance score computation phase.

### 3.1      Item-item Similarity Calculation Phase

In order to define pair-wise similarity between item vectors many similarity measures namely Pearson correlation coefficient, cosine similarity, adjusted cosine similarity etc are available in the literature. The similarity measure we have considered is based on number of users who agree on item vectors $i$ and $j$. The Correlation Rating Matrix ($CRM$) stores the similarities between each pair of item vectors. $CRM_{i,j}$ between item vectors  $i$ and $j$ is defined as given in (4)

$$CRM_{i,j} = \begin{cases} 1, & \text{if} \quad |ui \cap uj| > 0, \quad i,j = 1,2,3,\dots,n \\ 0, & Otherwise \end{cases} \qquad (4)$$

where ui is the set of users who have rated for item i. $|u_i \cap u_j|$ represents cardinality of the set $(u_i \cap u_j)$.

### Definition 2. Correlation Rating Matrix (*CRM*)

For the given $m \times n$ user item rating matrix *R*, item-item Correlation Rating Matrix can be represented as an $n \times n$ matrix. The matrix rows and columns represent items and each $CRM_{i,j}$ represents 1 if at least one user rates both the items. $CRM_{i,j}= 0$, implies that item *i* is not related to item *j*.

*CRM* is a symmetric matrix. But if an item i is rated by few users and item j is rated by many users, then their correlation should differ. So we normalize *CRM* in to Normalized Correlation Similarity Matrix *NCRM*.

### Definition 3. Normalized Correlation Rating Matrix (*NCRM*)

NCRM is a matrix that records the relationship between items as ratio of how the user is correlating in the current item and the remaining items. NCRM can be calculated based on the formula given in (5)

$$NCRM_{i,j} = \frac{CRM_{i,j}}{\sum_i CRM_{i,j}} \tag{5}$$

The rows and columns of *NCRM* represent items and it is an asymmetrical matrix. For each item, the *NCRM* value is between 0 and 1. Since the relevance score computation procedure we employ requires binary similarity matrix for relevance score computations. So we convert *NCRM* into S, a binary similarity matrix of various items of the rating matrix. *CRM* and *NCRM* of original rating matrix *R* are shown in Fig. 1.

| CRM | I1 | I2 | I3 | I4 | I5 | I6 |
|-----|----|----|----|----|----|----|
| I1 | 1 | 1 | 0 | 1 | 0 | 0 |
| I2 | 1 | 1 | 0 | 1 | 1 | 0 |
| I3 | 0 | 0 | 1 | 0 | 0 | 1 |
| I4 | 1 | 1 | 0 | 1 | 0 | 0 |
| I5 | 0 | 1 | 0 | 0 | 1 | 0 |
| I6 | 0 | 0 | 1 | 0 | 0 | 1 |

| NCRM | I1 | I2 | I3 | I4 | I5 | I6 |
|------|------|------|-----|------|-----|-----|
| I1 | 0.333 | 0.333 | 0 | 0.333 | 0 | 0 |
| I2 | 0.25 | 0.25 | 0 | 0.25 | 0.25 | 0 |
| I3 | 0 | 0 | 0.5 | 0 | 0 | 0.5 |
| I4 | 0.333 | 0.333 | 0 | 0.333 | 0 | 0 |
| I5 | 0 | 0.5 | 0 | 0 | 0.5 | 0 |
| I6 | 0 | 0 | 0.5 | 0 | 0 | 0.5 |

**Fig. 1.** CRM and NCRM of original matrix R given in Table1

### Definition 4. Item-item Similarity Matrix (S)

For the given $m \times n$ user item rating matrix *R*, item-item similarity matrix is a binary similarity matrix(*S*) which can be calculated by the formula

$$S_{i,j} = \begin{cases} 1, & NCRM_{i,j} > 0.5, \quad i,j = 1,2,3,\dots,n \\ 0, & \text{Otherwise} \end{cases} \qquad (6)$$

S is a binary similarity matrix. Similarity matrix S of R is shown in Table 2.

**Table 2.** Binary Similarity Matrix, S

|    | I1 | I2 | I3 | I4 | I5 | I6 |
|----|----|----|----|----|----|----|
| I1 | 0  | 0  | 0  | 0  | 0  | 0  |
| I2 | 0  | 0  | 0  | 0  | 0  | 0  |
| I3 | 0  | 0  | 1  | 0  | 0  | 1  |
| I4 | 0  | 0  | 0  | 0  | 0  | 0  |
| I5 | 0  | 1  | 0  | 0  | 1  | 0  |
| I6 | 0  | 0  | 1  | 0  | 0  | 1  |

### 3.2 Relevance Score Computation Phase.

From the binary similarity matrix $S$, for each item, the items with similarity value 1 are called relevance items ($R$) and with similarity value 0 are called non-relevant items ($NR$). Next we investigate how user $u$ has rated an item $i$ and all its similar and dissimilar items. If he has rated most of the similar items of $i$, then he can be assigned a good score for his consistency in rating. At the same time high consistence reveals monotonic rating behavior of users which needs to be balanced.

On the other hand if the user has rated an item and most of the dissimilar items of it, then he can be assigned a good score for his divergent thinking and rating. So we consider both scores in order to have a tradeoff between them. We combine both of them to calculate relevance score of the user. The relevance score of the user is estimated using (7) and (8). This is based on the basic probability model proposed by Robertson and Spark [15] in Information Retrieval domain. This feature weighing scheme is used by search engines to rank matching documents according to their relevance to a given search query.

$$CW_{u,i} = \log \frac{(r_i + 0.5) * (N - R - n_i + r_i + 0.5)}{(n_i - r_i + 0.5) * (R - r_i + 0.5)} \qquad (7)$$

$$DW_{u,i} = \log \frac{(nr_i + 0.5) * (N - NR - n_i + nr_i + 0.5)}{(n_i - nr_i + 0.5) * (NR - nr_i + 0.5)} \qquad (8)$$

where $CW_{u,i}$ and $DW_{u,i}$ are the consistency and diversity scores of user u for item i. $CW_{u,i}$ and $DW_{u,i}$ are log odd ratios of user u in relevant and non-relevant items of item i respectively. N is the number of items in the collection. R is the number of items similar to item $i$. $NR$ is the total number of dissimilar items of $i$. $n_i$ is the num-

ber of items in the collection for which the user $u$ has rated and $r_i$ is the number of similar items of $i$ for which user u has rated, $nr_i$ is the number of dissimilar items of $i$ for which user $u$ has rated. Weight assigned to each user is calculated as the total scores assigned to all items and is calculated as given in (9) and (10).

$$CW_u = \sum_{i \in I} CW_{u,i} \tag{9}$$

$$DW_u = \sum_{i \in I} DW_{u,i} \tag{10}$$

where I is the set of items and $CW_u$ and $DW_u$ are called consistency and diversity scores of user $u$. Consistency score signifies user's consistency in rating items where as diversity signifies user's diversified thinking in analyzing items. Since both the scores are important as one improves accuracy of predictions where as other improves diversity in predictions, we consider average of the two scores as a common score for users, which we call as relevance score. Relevance score of users is calculated as given in (11).

$$RW_u = \frac{CW_u + DW_u}{2} \tag{11}$$

## 4     Building User Relevance Model for CF

In the previous section we discussed about computing user relevance score. We incorporate user relevance score into the most widely used similarity measure namely Cosine Similarity. A new similarity measure which we call User Relevance based Cosine Similarity, $RelSim_{i,j}$ is formulated as given in (12).

$$RelSim_{i,j} = \frac{\sum_{u \in u(i) \cap u(j)} (r_{u,i}) \times (r_{u,j}) * RW_u{}^2}{\sqrt{\sum_{u \in u(i) \cap u(j)} r_{u,i}{}^2 * RW_u{}^2} \sqrt{\sum_{u \in u(i) \cap u(j)} r_{u,i}{}^2 * RW_u{}^2}} \tag{12}$$

$RWu$ is the relevance score of the user who rated both the items $i$ and $j$. After calculating the similarity values, a model is built as described in [5].

## 5     Experimental Evaluation

This section discusses about the data sets used, accuracy of predictions and computational complexities. Experiments are conducted using Item-based collaborative filtering recommendation algorithms [5] and Userrank for item-based Collaborative filtering recommendation [11], in order to prove the efficiency of the proposed technique.

## 5.1    Data Set Used

The experiments are run on two datasets namely Movielens[1] and a subset of rating selected from it. The subset is populated by taking top 100 users based on total number of rating and top 100 items rated by those users. We call it as MovieTop100. The datasets have ratings given for movies in the range 1 to 5.

The details of the data sets are given in Table 3. We partition the data sets into training set with 80% of the ratings and test data set with 20% of the ratings. Five cross validation is done for all experiments.

**Table 3.** Data sets used

|           | Movielens | MovieTop100 |
|-----------|-----------|-------------|
| # users   | 943       | 100         |
| # items   | 1682      | 100         |
| # ratings | 100,000   | 2474        |

## 5.2    Experimental Metrics and Evaluation Methodologies.

Here we outline the experiments we have taken to study the significance of the proposed technique in improving the quality of predictions in terms of accuracy and diversity.

**Accuracy Measures**

In order to evaluate the performance of the proposed technique in terms of accuracy of predictions, we follow the approach used in [14]. The metrics used are HR (Hit Ratio), MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). HR is the ratio of the number of hits to the size of test data set used. The predicted rating is a called as a hit if its rounded value is equal to the actual rating given in the test data set. MAE is average absolute deviation between predicted and actual ratings. MAE penalizes each wrong prediction by its distance to the actual rating, whereas RMSE emphasizes larger deviations.  These measures are as formulated below.

$$MAE = \frac{\sum_{i=1}^{n} |p_i - a_i|}{n} \quad RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - a_i)^2}{n}} \quad HR = \frac{no\ of\ hits}{n}$$

Where $p_i$ is the predicted rating and $a_i$ is the actual rating for the item $i$. $n$ is the number of test cases.

**Diversity Measure (Cosine diversity)**

In order to show that the proposed approach improves diversity of predictions we employ average inter list dissimilarity of predicted values. The cosine coefficient is a commonly used similarity measure. Based on cosine similarity a new diversity measure called cosine dissimilarity is proposed in [16]. Cosine dissimilarity is calculated for each pair of predicted ratings and its average is taken as diversity value.

We compared our technique with ItemCF [5] and ItemCF(U-Rank) [11]. Table 4 shows the results of computations for various measures. The quality of predictions increases as HR and Diversity increase whereas quality decreases as MAE and RMSE increase. The values of various measures from Table 4 show that the proposed approach performs better than the benchmark techniques ItemCF and ItemCF(U_Rank) on both the datasets.

**Table 4.** Comparison of various measures by ItemCF, ItemCF(U_Rank) and ItemCF(U_Rel)

|  | Movielens | | | MovieTop100 | | |
|---|---|---|---|---|---|---|
|  | ItemCF | ItemCF (U_Rank) | ItemCF (U_Rel) | ItemCF | ItemCF (U-Rank) | ItemCF (U_Rel) |
| MAE | 0.9318 | 0.9306 | 0.9155 | 0.9208 | 0.9055 | 0.9020 |
| RMSE | 1.2041 | 1.1976 | 1.1778 | 1.1710 | 1.1575 | 1.1519 |
| HR | 0.3386 | 0.3388 | 0.3441 | 0.3233 | 0.3278 | 0.3288 |
| DIV | 0.0243 | 0.0244 | 0.0250 | 0.0152 | 0.0175 | 0.0177 |

**Fig. 2** and **Fig. 3** show the comparison of all the three algorithms using different measures on Movielens dataset.
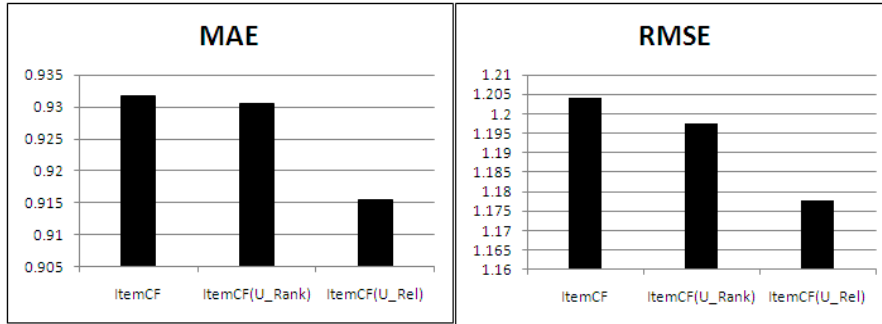


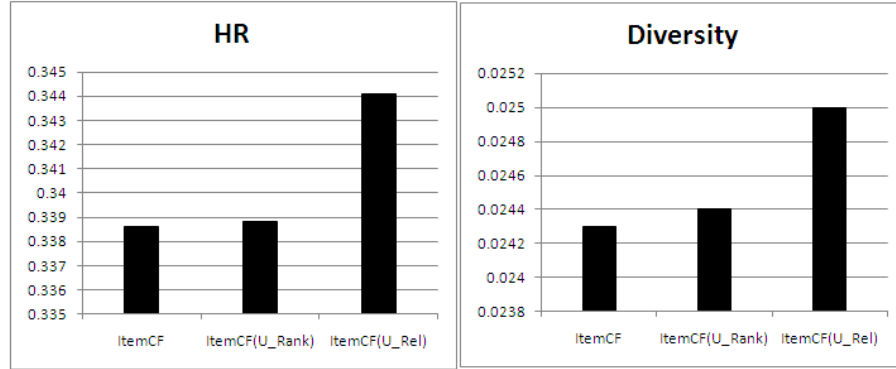**Fig. 2.** Comparison of MAE and RMSE measures for Movielens dataset

**Fig. 3.** Comparison of HR and Diversity measures for Movielens dataset

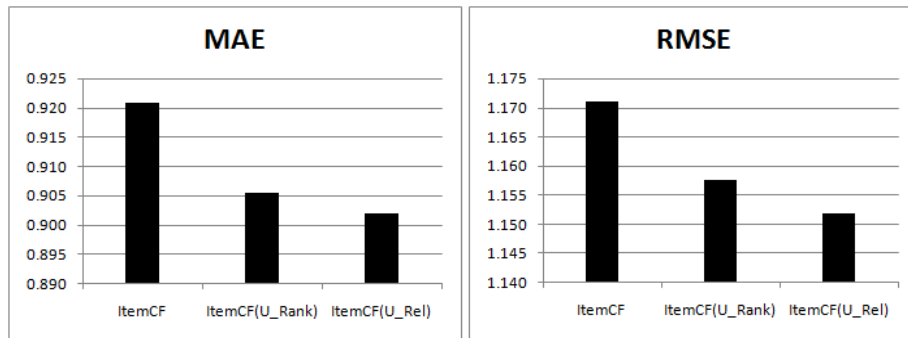**Fig. 4** and **Fig. 5** show the comparison of all the three algorithms using different measures on MovieTop100 dataset.



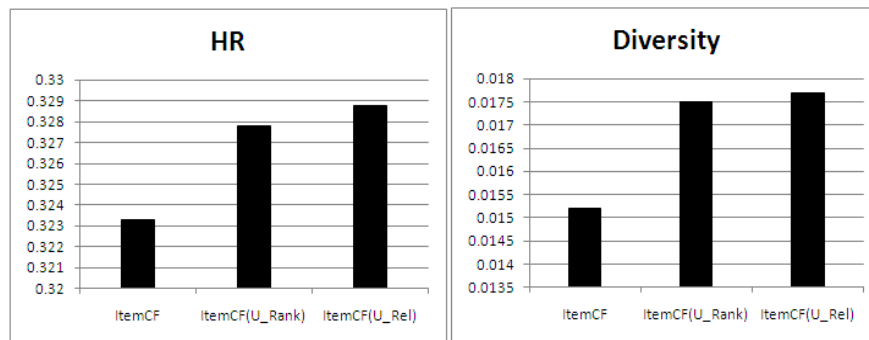**Fig. 4.** Comparison of MAE and RMSE measures for MovieTop100 dataset



**Fig. 5.** Comparison of HR and Diversity measures of MovieTop100 dataset

### 5.3    Comparing Computational Complexities.

High computational complexity is often needed to enhance the predictions.  In the model based view, the computational complexity can be split into complexity in offline phase and complexity in online phase. Offline phase includes similarity computations, relevance score calculations and model building. The complexity of offline computations is $O(mn2+n2+kmn)$ for $m$ users, $n$ items and top $k$ neighbours. Online phase includes prediction computations of n test items with complexity $O(kn)$. For most recommender systems, the online complexity is more important than offline complexity.

## 6    Conclusion

The goal of this work is to improve accuracy and diversity of predictions by including relevance score of users. The proposed approach sets user relevance score as a combination of user's consistency and diversity scores and a new similarity measure is proposed based on user relevance score. A bench mark algorithm with the inclusion of user relevance score is examined for proving the efficiency of results. The experimental results show that the inclusion of relevance weight helps in improving predictions. In future we plan to include the content based attribute values in order to calculate the relevance score of the users.

### References

1.  MovieLens data, http://www.grouplens.org/
2.  Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Recommender Systems Handbook, Springer 2011.
3.  Breese, J. S., Heckerman, D., Kadie, C., Empirical Analysis of Predictive Algorithms for Collaborative Filtering, UAI, 1998.
4.  Gediminas Adomavicius , Alexander 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, June 2005
5.  B Sarwar, G Karypis, J Konstan, J Reidl, Item-based collaborative filtering recommendation algorithms, World Wide Web, 285-295, 2001.
6.  Prem Melville and Vikas Sindhwani, Recommender Systems, Encyclopedia of Machine Learning, 2010.
7.  Micheal D. Ekstrand, John T. Reiedl and Joseph A. Konstan, Collaborative filtering Recommendation  systems, Foundations@Trends, Human-Computer Interaction, 2010.
8.  Pazzani, Michael J, and Daniel Billsus, "Content-based recommendation systems: The Adaptive Web", Springer Berlin Heidelberg, 2007.
9.  George Karypis, Evaluation of Item-Based Top-N Recommendation Algorithms, CIKM, 2001, Atlanta, Georgia, USA.
10. Xiaoyuan Su and Taghi M.Khoshoftaar, A Survey of Collaborative Filtering Techniques, Hindawi Publishing Corporation, Advances in Artificial Intelligence, 2009.
11. Min Gao, Zhongfu Wu, Feng Jiang, Userrank for item-based Collaborative filtering recommendation, Information processing letters, Elsevier, 2011.

12. Daniel Lemire, Anna Maclaclan, Slope One Predictors for Online Rating-Based  Collaborative Filtering, Society for Industrial and Applied Mathematics International conference on data mining, SDM, 2005.
13. Schafer, J.B.Konstan,J, and Riedl,J,  Recommender systems in E-Commerce, ACM E-Commerce conference, 1999
14. Hilmi Yildirim and Mukkai S. Krishnamoorthy, A Random Walk Method for Alleviating the Sparsity Problem in Collaborative Filtering, RecSys'08,
15. S. E Robertson, K. Sparck Jones, Relevance Weighting of Search Terms, Journal of American Society of Information Science, 1976.
16. Hurley, N. and Zhang, M. Novelty and diversity in top-*n* recommendation – Analysis and evaluation. ACM Trans. Internet Technology.  March 2011.