IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 5, Ver. IV (Sep – Oct. 2014), PP 145-153
www.iosrjournals.org

Improving Recommendation using Forecast based approach and Re-ranking Approache

Mr. Mapari Vikas Prakash, Prof. Patil Pramod
Department of Computer Engineering Dr. D.Y.P.C.O.E. Pimpri, Pune, India
Department of Computer Engineering Dr. D.Y.P.C.O.E. Pimpri, Pune, India

Abstract: *Collaborative filtering is one of the most promising techniques in recommender systems, providing personalized recommendations to users based on their previously expressed preferences in the form of ratings and those of other similar users. A recommender system uses Collaborative Filtering or Content-Based methods to predict new items of interest for a user. Although both methods have their own and distinct advantages but individually they fail to provide good recommendations in many situations. Incorporating components from collaborative and content based methods, can overcome these challenges like data sparsity, stability, accuracy and correlation of traditional recommender systems. Inadequate ratings lot of time gives poor quality of recommendations in terms of accuracy. Various approaches are used for overcome these issues: i) Firstly, we propose to improve data sparsity and correlation. ii) Secondly, we aim to tackle the problem of rank and relevance and we improve recommender system in novelty & diversity using rank & relevance technique.*
Keywords: *Recommender systems, recommendation diversity, sparsity, correlation, ranking functions, Collaborative Filtering, Content based filtering.*

I. Introduction

In the current age of information overload, it is becoming increasingly harder to find relevant content. Recommender systems try to predict the ratings of unknown items for each user, often using other users' ratings, and recommend top N items with the highest predicted ratings. As one of the most promising recommender techniques, Collaborative Filtering (CF) predicts the potential interests of an active user by considering the opinions of users with similar taste.

Recommender systems predict the ratings of unknown items for each and every user, often using other user's ratings, and recommend top N items with the highest predicted ratings. In online applications, items are rated with more or less rating (Rating is scaled in between the range of 1 to 5, from lower to higher order). Common online applications are online shopping, games, movies, music, videos etc. One of the most promising recommender techniques, Collaborative Filtering (CF) predicts the potential interests of an active user by considering the opinions of users with same taste. Simple algorithms and accurate recommendation are two main aspect of Collaborative

Filtering technique, memory based CF. Memory based CF detect the users ratings on different items by asking the user or by observing his/her interaction with the systems to store them into a table known as the rating matrix. Then, memory based CF methods use similarity measurement methods to filter users (or items) that are similar to the active user (or the target item) and calculate the prediction from the ratings of these neighbors.

A. What are Recommender / Recommendation System?

Online recommender systems in which they are used to either predict whether a particular user will like a particular item (prediction), or to identify a set of N items that will be of interest to a certain user (top-N recommendation). Recommender systems (RS) are used in a variety of applications. Examples are web stores, online communities, and music players.

Currently, people mostly tend to associate recommender systems with e-commerce sites, where recommender systems are extensively used to recommend items / products to the customers and to provide customers with information to help them decide buy which products. Products can be based on the top overall sell on a site, on the demographics of the consumers, or on an analysis of the past buying behavior of the consumers as a prediction for future buying behavior.

B. Types of recommendation system

Figure 1 shows the approaches to Recommender Systems categorized as follows:

- **Content Based Recommendation:** In content based recommendation items those are similar in content to items the user has liked in the past or matched to attributes of the user are recommended.

- **Collaborative Filtering (CF):** In Collaborative Filtering systems a user is recommend items based on the previous ratings of all users collectively.
- **Hybrid Approaches:** These methods combine either collaborative and content based approaches or different approaches from CF or CB.

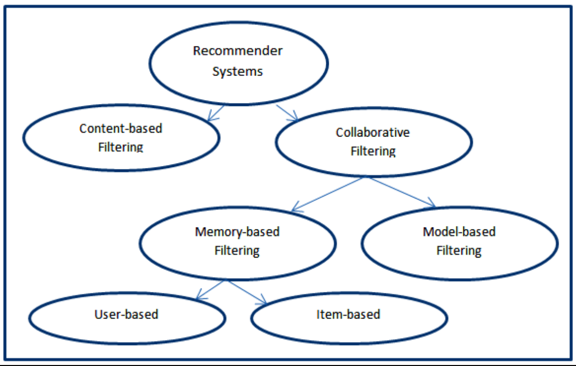

Fig. 1. An overview of recommender techniques

1) Content-based filtering

In content-based methods, items with similar content features comparing to a user's past favorite items will be recommended to the user. The weakness of this kind of methods is that it depends on the features, and effective features are difficult to find in some recommendation applications. . For example, in the Amazon, many users have very incomplete pro ling information, and the items in their history have a quantity of diversity. Thus there are not enough features for accurate predictions. This is the reason that collaborative filtering comes up.

2) Collaborative filtering

Collaborative filtering (CF) is different from other filtering technologies in that information is filtered by using evaluation instead of analysis, thus categorizing information based on the user's opinion of the information instead of the information itself. In addition, CF stresses the concept of community by letting recommendations be a result of the opinions of the current user and other similar users. As figure 1shows, all users contribute with ratings based on their preferences. Recommendations for the current user are produced by matching the user's ratings with ratings given by other users. In this way, similar users are linked together to form a community.

The prediction is based on the common behavior patterns analyzed from the large real dataset. The key point is that CF finds similar users for each user, according to the similarity of their rating history. Then the prediction is made by the ratings of his/her similar users.

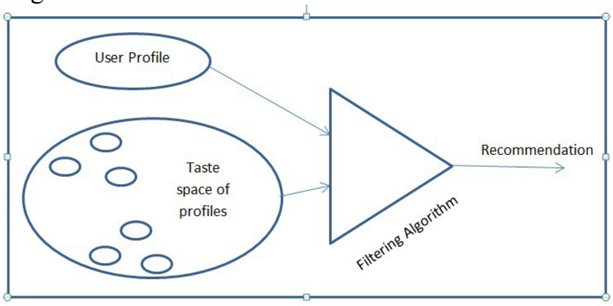

Fig. 2. Collaborative Filtering process

C. Challenges of collaborative filtering

- **Data sparsity:** Data sparsity refers to amount of data we have for a particular dimension/entity of the model. In practice, many recommender systems are based on huge datasets. Huge and sparse dataset can cause the challenge in the performance of the recommendation.

New items are inserted in system are need to be rated with a considerable number of users before it recommended to user who have similar taste once rated them. If there are many items to be recommended, even if there are many users, the user/ratings matrix is sparse, and it is hard to find users that have rated the same items.

- **Diversity:** In collaborative filtering increase in diversity help us to discover a new product among the several choices. Rating of item or popularity is the main constraints for collaborative filtering recommendation algorithm, the products cannot usually recommended with limited past data. It will cause effect for popular products or items, leading to less diversity.

- **Stability:** Stability is the consistent agreement of predictions made on the same items by the same algorithm, when any new incoming ratings are in complete agreement to systems prior estimations. Stability is an important and desired property of recommender systems, and has a number of potential implications related to users trust and acceptance of such systems.
- **Synonyms:** Items with different names or entries have same or very similar item tendency. Recommender systems not able to find this synonym association and thus treat these products differently. The degree of inconsistency in descriptive term usage is greater than commonly supposed. The performance of collaborating filtering systems is getting reduced due to occurrence of synonyms.
- **Cold Start:** There needs to be enough other users already in the system to find a match.
- **Novelty:** The ability of a CF system to recommend items that the user was not already aware of.

D. Applications of Recommendation Systems

- **Product Recommendations:** Perhaps the most important use of recommendation systems is at on-line retailers. We have noted how Amazon or similar on-line vendors strive to present each returning user with some suggestions of products that they might like to buy. These suggestions are not random, but are based on the purchasing decisions made by similar customers or on other techniques.
- **Movie Recommendations:** Netfix offers its customers recommendations of movies they might like. These recommendations are based on ratings provided by users. The importance of predicting ratings accurately is so high, that Netfix offered a prize of one million dollars for the first algorithm that could beat its own recommendation system.
- **News Articles:** News services have attempted to identify articles of interest to readers, based on the articles that they have read in the past. The similarity might be based on the similarity of important words in the documents, or on the articles that are read by people with similar reading tastes.
- **E-commerce:** Recommendations for consumers of products to buy such as books, cameras, PCs etc.
- **Services:** Recommendations of travel services, recommendation of experts for consultation, recommendation of houses to rent, or matchmaking service.

II. Literature survey

Recommender systems use the opinions of a community of users to help individuals in that community more effectively identify content of interest from a potentially overwhelming set of choices [Resnick and Varian 1997]. One of the most successful technologies for recommender systems, called collaborative filtering, has been developed and improved over the past decade to the point where a wide variety of algorithms exist for generating recommendations. Each algorithmic approach has adherents who claim it to be superior for some purpose. Clearly identifying the best algorithm for a given purpose has proven challenging, in part because researchers disagree on which attributes should be measured, and on which metrics should be used for each attribute.

Evaluating recommender systems and their algorithms is inherently difficult for several reasons. First, different algorithms may be better or worse on different data sets. Many collaborative filtering algorithms have been designed specifically for data sets where there are many more users than items. The second reason that evaluation is difficult is that the goals for which an evaluation is performed may differ. Much early evaluation work focused specifically on the accuracy of collaborative filtering algorithms in predicting withheld ratings. Shardanand and Maes [1995] measured reversals large errors between the predicted and actual rating; the signal-processing measure of the Receiver Operating Characteristic curve [Swets 1963] is used to measure a recommenders potential as a filter [Konstan et al. 1997]. Other work has speculated that there are properties different from accuracy that has a larger effect on user satisfaction and performance.

A range of research and systems have looked at measures including the degree to which the recommendations cover the entire set of items [Mobasher et al. 2001], the degree to which recommendations made are nonobvious [McNee et al. 2002], and the ability of recommenders to explain their recommendations to users [Sinha and Swearingen 2002].

Collaborative Filtering (CF) [1,2] is one of the most successful recommender techniques, and it includes memory based CF techniques such as similarity based or neighborhood based CF algorithm; model based CF techniques such as clustering CF algorithms; and hybrid CF techniques such as personality diagnosis. As a representative memory based CF technique, similarity based methods represent one of the most successful approaches to recommendation. They have been notably deployed into commercial systems and been extensively studied. This class of algorithm can be further divided into user based and item based methods. The former is based on the basic assumption that people who have similar past preferences tend to agree in their future tastes. Hence, for the target user, the potential interest on an object is predicted according to the ratings from users who are similar to the target user.

In a typical memory based CF scenario, there is a set of n users U= $u_1, u_2,..., u_n$ and a set of m items I= $i_1, i_2,..., i_m$, and the user-item rating matrix. The ratings can either be explicit indications, such as an integer number from 1 to 5, or implicit indications, such as purchases or click troughs. For example, the implicit user behaviors can be converted to a user-item rating matrix R, where the R(k,l) in k^{th} row and l^{st} column of the matrix stands for the k^{th} users rating for the l^{st} item. Would the k^{th} user have not rated the l^{st} item yet, the null value is assigned to R(k,l). Thus, the recommendation problem is reduced to predicting the unrated entries.
Generally, the process of this type of CF methods consists of two steps: similarity measurement and rating prediction.

E. Similarity Measurement Between Users And Items

The critical part in memory based CF algorithms is the similarity measurement between users or items. In user based CF method, the similarity $s(u_x,u_y)$ between users u_x and u_y is found by comparing the items that both have rated. For item based CF method, the similarity $s(i_x,i_y)$ between items i_x and i_y is determined by the users who have rated both of the two items.

There are various methods to compute similarity between two users or items. The two most commonly used methods are Cosine Distance (CD) and Pearson Correlation (PC). To define them, let RI be the set of all items rated by both users u_x and u_y, and let R_U be the set of all users who have rated both items i_x and i_y. Then, the co-rated entries related to object O_k in $\{u_x, u_y, i_x, i_y\}$ form a D-dimensional vector, where D is equal to the size of set R_I or R_U.

1) Cosine Distance (CD)

For Cosine Distance approach, the cosine of the angle between two vectors represents the similarity between

$$s(O_k,O_l) = \cos\theta = \frac{\vec{D}_{Ok}.\vec{D}_{Ol}}{\|D_{Ok}\|\|D_{Ol}\|} \tag{1}$$

Where ". " denotes the dot product of two vectors, and "‖ ‖" is the vector modulus. $\overrightarrow{D_{Ok}}$ And $\overrightarrow{D_{Ol}}$ are two D-dimensional vectors constructed by the interactions between object O_k and object O_l, where O_k and O_l can be the pair of u_x and u_y, or i_x and i_y. Therefore, the bigger the cosine of the angle (θ), the more similar the two objects will be.

2) Pearson correlation (pc)

We should note that, in the computation of similarity, it is necessary to remove rating correlations, such as the average rating of the user, to improve the importance of similarity. This approach can improve the accuracy of similarity computation to some extent. For User base CF, the Pearson Correlation between two users is:

$$s(u_x, u_y) = \frac{\sum_{i\in I}(R_{ux,i} - \overline{R_{ux}})(R_{uy,i} - \overline{R_{uy}})}{\sqrt{\sum_{i\in I}(R_{ux,i} - \overline{R_{ux}})^2}\sqrt{\sum_{i\in I}(R_{uy,i} - \overline{R_{uy}})^2}} \tag{2}$$

Where, $R_{ux,i,}r_{uy,i}$are the ratings of users u_x, u_y on item i and , $\overline{R_{u_x}}$, $\overline{R_{u_y}}$are the average ratings of users u_x, u_y respectively.
Similarly, for ICF, the Pearson Correlation between two items can be formulated as:

$$s(i_x, i_y) = \frac{\sum_{u\in U}(r_{u,ix} - \overline{r_{ix}})(r_{u,iy} - \overline{r_{iy}})}{\sqrt{\sum_{u\in U}(r_{u,ix} - \overline{r_{ix}})^2}\sqrt{\sum_{u\in U}(r_{u,ix} - \overline{r_{iy}})^2}} \tag{3}$$

Where, $r_{u,ix,}r_{u,iy}$ are the ratings of user u on items i_x, i_y and ,are the average ratings of all users on items i_x, i_y, respectively.

F. Rating Prediction

The phase of rating prediction aims to predict the value that the active user will give to the target item. The KNN-based method is usually utilized to generate prediction by weighting sum of the ratings that similar users give to the target item or the ratings of the active user on similar items depending on whether one uses UCF or ICF.

1) user based CF (UCF)

The UCF algorithm is based on the basic assumption that people who share the similar past tastes will be interested in same items. The algorithm uses the following steps: the first step is to compute the similarities between users with the similarity measurement methods; then one produces the prediction for the active user by taking the weighted average of all the ratings of the similar users on a certain item according to the formula, the items with highest predicted ratings will be recommended to the user.

$$p_{u,I} = \bar{r}_u + \frac{\sum_{v\in U_{(u)}} s(v,u)(r_{v,i}-\bar{r}_v)}{\sum_{v\in U_{(u)}} |s(v,u)|} \quad (4)$$

Where $\bar{r}_u$, $\bar{r}_v$ and are the average ratings of users u and v, respectively; s(v, u) is the similarity between user v and user u calculated using similarity measurement methods; and U(u) denotes the set of similar users of user u. $p_{u,i}$ is the prediction of user u on item i.

2) Item based CF (ICF)

The ICF algorithm recommends to users the items similar to those already consumed. Similarly, after calculating the similarities between items, the unknown rating of user u on item i can be represented as an aggregation of user u on similar items:

$$p_{u,i} = \bar{r}_i + \frac{\sum_{j\in I_{(i)}} s(j,i)(r_{u,j}-\bar{r}_j)}{\sum_{i\in I_{(i)}} |s(j,i)|} \quad (5)$$

Where $\bar{r}_i$, $\bar{r}_j$ and are the average ratings of all users on item i and item j, respectively; s(j, i) is the similarity between item j and item i calculated using similarity measurement methods; and $I_{(i)}$ denotes the set of similar items of item i. $p_{u,i}$ denotes the prediction of user u on item i.

G. Ranking Approach:

Ranking products according to the rating variance of neighbors of a particular user for a particular item. There exist a number of different ranking approaches that can improve recommendation diversity by recommending items other than the ones with topmost predicted rating values to a user.

H. Problem Analyses

After using the co-rated entries as a vector to represent the object, the Cosine Distance measures the similarity between two users or items by computing the cosine of the angle. The bigger the value is, the more similar the two users or items will be. Pearson Correlation takes the rating correlation into consideration to eliminate the influence of average rating. Obviously, this class of similarity measurement method is a variation of Cosine Distance.

Taking UCF as an example, we pick the items that both users have rated before, and then use the ratings of each user on these items to construct a d-dimensional vector such as (r_{u,i_1}, r_{u,i_2}, ….. r_{u,i_d}), where d is the number of co-rated items. If we subtract each element by the average rating of user u, the vector will be changed to (r_{u,i_1} - u,, r_{u,i_2} - u,,….. r_{u,i_d} - u,). In this case, the Pearson Correlation is equivalent to Cosine Distance. With Pearson Correlation, the accuracy of similarity computation can be improved to a certain extent. However, it still suffers from many issues.

Data Sparsity: It's difficult to find co-rated entries when the data is sparse. Similarity between them cannot be computed with existing methods. Furthermore, the similarities between users or items may not be obtained in the same dimensionality.

Data Correlation: Data correlation corresponds to the common features hidden in the data coming from the similar attributes among users or items. Correlations among the ratings result in the non-orthogonal vector space since the elements in different dimensions are not independent. Although the Pearson Correlation has eliminated the influence of average rating, such rating correlation still exists. Therefore, the similarities computed with these similarity measurement methods are not accurate.

Because of these issues, the similarity between two users or items computed with Cosine Distance or Pearson Correlation is not accurate. Consequently, if we take a weighted average of the ratings using the similarities to produce the prediction directly, we may not get a good result. We abstract these problems as data sparsity and data correlation, and use the Forecast model for rating prediction.

I. Forecast Model Based Approach

The process for Forecast to make prediction can be described as: The Cosine Distance method is used to measure the similarity between two items. Then, a similarity matrix will be generated, where m is the number of items. Although the similarity computation is not accurate, the value can represent the degree of similarity.

Thus, algorithm doesn't use the exact value of similarity but rather just rank the items according to them. Then, to generate the prediction of the active user u on item i, the k most similar items that have been rated by the active user to item i are chosen.

Finally, algorithm use these items as input to construct a Forecast model and predict the rating of the active user u on item i. If user u didn't rate k items, the fixed value will be used to complete k ratings. The fixed value can be the median value of rating scale. For example, when the rating scale is 1 to 5, number 3 is selected as the fixed value. With this method, there are three main contributions.

- **Overcoming Data Sparsity:** Although the data is sparse and few items has been rated by each user, only a few neighbors are needed to construct the Forecast model for our algorithm and the experimental results show that the prediction accuracy is still high even when k is equal to 5. Therefore, the proposed algorithm can efficiently address the data sparsity problem.
- **Benefiting From Data Correlation:** The stronger the data correlations are, the more accurate the Forecast model will be. In other words, the proposed algorithm can efficiently benefit from the data correlations rather than eliminate them.
- **Obtaining Accurate Prediction:** We test our algorithm on two datasets, users' dataset and MovieLens. The experimental results compared with UCF and ICF (with Cosine Distance for similarity measurement) show that proposed algorithm gets better performance in prediction accuracy. Specially, with the MovieLens dataset, the accuracy has been improved. Moreover, the value of k can be very small without losing in accuracy.

J. Ranking-Based Approach

Recommender systems compute ratings of items (or products) that are yet to be consumed by users, based on the ratings of items already consumed. A recommender system tries to predict the ratings of unknown items for each user, often using other user's ratings, and recommend top N items with the highest predicted ratings. There are many new algorithms that can improve the predictive accuracy of recommendations.

The quality of recommendations can be relying on the accuracy of recommendations may not be enough to find the most relevant items for each user. One of the goals of recommender systems is to provide a user with highly personalized items, and more diverse recommendations result in more opportunities for users to get recommended such items. With this motivation proposing new recommendation method that can increase the diversity of recommendation for individual user, often measured by an average dissimilarity between all pairs of recommended items, while maintaining an acceptable level of accuracy.

It is possible to obtain higher diversity simply by recommending less popular items; however, the loss of recommendation accuracy. Proposed approaches can increase the diversity of recommendations with only a minimal (negligible) accuracy loss using recommendation ranking techniques. Traditional recommender systems rank the relevant items in a descending order of their predicted ratings and recommend top N items to each user, resulting in high accuracy. The proposed approaches consider item popularity, when ranking the recommendation list to substantially increase recommendation diversity while maintaining comparable levels of accuracy.

1) Standard Ranking Approach

Recommender systems predict unknown ratings based on known ratings, using any traditional recommendation technique. Then, the predicted ratings are used to support the user's decision making. Each user u gets recommended a list of top N items, selected according to some ranking criterion. More formally, item i_x is ranked ahead of item i_y (i.e., $i_x < i_y$) if $rank(i_x) < rank(i_y)$, where rank: I R is a function representing the ranking criterion: rankStandard(i) = R*(u, i)-1 (6)

The power of -1 in equation indicates that the items with highest-predicted ratings R*(u, i) are the ones being recommended to user. Recommending the most highly predicted items selected by the standard ranking approach is designed to help improve recommendation accuracy, but not recommendation diversity.

Therefore, new ranking criteria are needed in order to achieve diversity improvement. Since recommending best-selling items to each user typically leads to diversity reduction, recommending less popular items intuitively should have an effect towards increasing recommendation diversity. Following this motivation, we explore the possibility to use item popularity as a recommendation ranking criterion, and in the next subsection we show how this approach can affect the recommendation quality in terms of accuracy and diversity.

II. Proposed Methodology

A. Forecast Based Approach

A recommender system can be regarded as a Grey system and with our algorithm; this model is used to yield the rating prediction. The Grey Forecast model utilizes accumulated generation operations to build differential equations, which benefit from the data correlations. It has another significant characteristic of requiring less data so it overcomes data sparsity problem. The rating sequence generated in the phase of rating preprocessing is all that is needed as input and future forecasting. The general procedure for this model is derived as follows:

Step 1: Assume the original rating sequence to be $r_u^{(0)}$

$r_u^{(0)} = \{ r^{(0)}(t) \}$, t = 1, 2, ….. k (6)

Where $r^{(0)}$ corresponds to the original rating of user u on the $(k-t+1)^{th}$ most similar item or the t^{th} value of the rating sequence. k is the number of neighbors or the length of the rating sequence and must be equal to or larger than 4.

Step 2: A new sequence $r_u^{(1)}$ is produced by the Accumulated Generating Operation (AGO).

$$r_u^{(1)} = \{r^{(1)}(t)\},\ t = 1, 2, \ldots k \tag{7}$$

Where $r^{(1)}(t) = \sum_{j=1}^{t} r^{(0)}(j)$, t = 1, 2, …..k

Step 3: Build a first-order differential equation.

$$dr^{(1)} / dt + az^{(1)} = b \tag{8}$$

Where $z^{(1)}(t) = \alpha r^{(1)}(t) + (1-\alpha) r^{(1)}(t+1)$, t = 1, 2, …..k-1. α $(0 < \alpha < 1)$ denotes a horizontal developing coefficient. The selecting criterion of α is to yield the smallest prediction error rate. We conducted extensive experiments with different values of α, and find that when α<0.5, this model based method performs well. Therefore, in our experiments, we will set α=0.2.

Step 4: From Step 3, we get the forecasting model GM(1,1):

r ̂(1) (t+1) = (r(0) (1) – b / a) e-at + b/ a

Where a is the development coefficient, and b is grey action, and,

$$\begin{bmatrix} a \\ b \end{bmatrix} = (B^T B)^{-1} B^T Y,\ B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ \ldots\ldots\ldots & \ldots \\ -z^{(1)}(k) & 1 \end{bmatrix},\ Y = \begin{bmatrix} r^{(0)}(2) \\ \ldots \\ r^{(0)}(k) \end{bmatrix} \tag{9}$$

Step 5: Inverse Accumulated Generation Operation (IAGO). Because the Grey Forecast model is formulated using the data of AGO rather than original data, we should use IAGO to convert the data of AGO to an actual rating prediction:

$$\hat{r}^{(0)}(t+1) = \hat{r}^{(1)}(t+1) - \hat{r}^{(1)}(t) = (r^{(0)}(1) - b / a)\, e^{-at}(1-e^{a}) \tag{10}$$

When we set t=k, the rating prediction $p_{u,j}$ of user u on item i can be represented by r ̂(0)(k+1).

Obviously, during the estimate of parameters and in Step 4, a matrix inverse operation is needed. Hence, we cannot always forecast the ratings using Grey Forecast model. In these cases, the average of k ratings is used as the rating prediction of the active user on the target item.

B. Ranking Approach

Recommender systems predict unknown ratings based on known ratings, using any traditional recommendation technique. Then, the predicted ratings are used to support the user's decision-making. In particular, each user u gets recommended a list of top-N items, denoted by $L_N(u)$, selected according to ranking criterion. More formally, item i_x is ranked ahead of item i_y (i.e., $i_x \prec i_y$) if $rank(i_x) < rank(i_y)$, where rank: is a function representing the ranking criterion. The vast majority of current recommender systems use the predicted rating values the ranking criterion or, more formally:

$$rank_{Standard}(i) = R^*(u, i)^{-1} \tag{11}$$

The power of -1 in the above expression indicates that the items with highest-predicted (as opposed to lowest-predicted) ratings R*(u, i) are the ones being recommended to user.

1) Item Popularity-Based Ranking

This approach ranks items directly based on item popularity, based on low rank to high rank. Popularity of item is given by the number of known ratings that each item has.

$$rank_{ItemPop}(i) = |U(i)|,\ \text{where } U(i) = \{u \in U \mid \exists R(u, i)\} \tag{12}$$

Further we will compare the performance of the item popularity based ranking approach with the standard ranking approach using Datasets, and we want to show that, as compared to the standard ranking approach, the item popularity-based ranking approach increase recommendation diversity.

III. Experimental Results

We describe the datasets used; the performance improvement compared to the traditional memory based collaborative filtering methods.

The proposed work i.e. forecast model and raking approach is deployed as well as User based Collaborative Filtering and Item based Collaborative Filtering methods on dataset. Dataset is collected globally and individually available rating datasets. There are different sizes of available datasets. In this paper the dataset which consists of 100,000 ratings (1-5) from 943 users on 1682 movies.

Dataset attribute Information

The data is randomly ordered. This is a tab separated list of File Names and format:

user id :: item id :: rating :: timestamp

The data obtained from following web site:

http://www.movielens.org/

A. Results Using Forecast Approach

Following are the results obtained from executing algorithm over item with various rating as input. The Cosine Distance method is used for the similarity computation between users or items. For proposed work let set the horizontal developing coefficient = 2. When certain user didn't rate k items, the fixed value 3 will be used with lowest similarity so that the rating sequence will always have k numbers.

The results illustrated in Fig. 3 shows that Forecast model based method improves prediction so the mean absolute error (MAE) is minimized. As shown in Fig. 4, the proposed work also minimizes the root mean square error (RMSE) with respect to User based Collaborative Filtering.

The experimental results show that the proposed algorithm can significantly overcome the limitation of the data sparsity and cope with data correlation. In particular, the accuracy of the MovieLens dataset has been improved by over 20% in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

B. Result Using Ranking Approach

The proposed recommendation ranking approaches were pre-processed with each dataset to include users and dataset past ratings that will gives assurance to have a sufficient amount of high predicted items/products for the recommendation to individual user. Every ranking approach performance is measured in terms of diversity-in-top-N (N=1, 5, 10), and its diversity gain and precision loss with respect to the standard ranking approach was calculated. Every re-ranking approach will improve the diversity of recommendations.

The results illustrated in Fig. 3 shows that ranking approach improves prediction so the mean absolute error (MAE) is minimized. As shown in Fig. 4, the proposed work also minimizes the root mean square error (RMSE) with respect to User based Collaborative Filtering.

In this paper we are trying to improve the overall recommendation system, instead of improving it on individual factor (like Data Sparsity, Diversity separately).

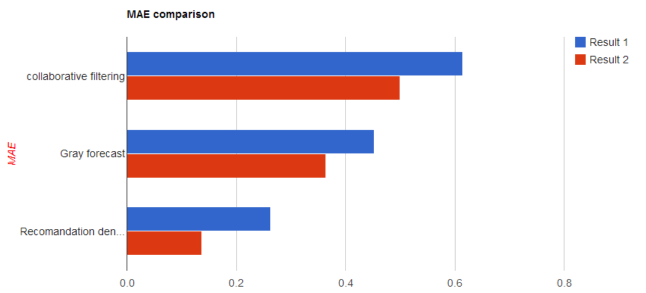

Fig. 3. Comparison result between similarity measurement and forecast model and Ranking approach (MAE)

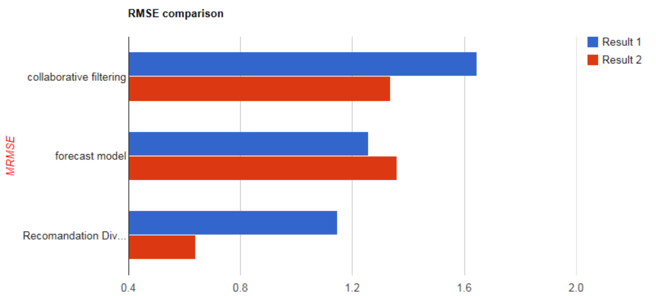

Fig. 4. Comparison result between similarity measurement and forecast model and Ranking approach (RMSE)

IV. Conclusion And Future Work

Since the existing similarity measurement methods, such as Cosine Distance and Pearson Correlation, cannot compute the similarities between users or items accurately when the data is sparse and there exists strong data correlations, user based CF and item based CF methods don't perform well in prediction accuracy. The Forecast model for rating prediction in recommender systems can overcome data sparsity, benefit from data correlations. As an effective rating prediction method, the Forecast model still has room for improvement, when the user didn't rate enough k items, we will use the average of the user's ratings on all items instead of the fixed value to complete k numbers.

However, in most cases, new techniques are designed to improve the accuracy of recommendations, whereas the recommendation diversity has often been overlooked. Ranking recommendations according to the predicted rating values (which is a de facto ranking standard in recommender systems) provides good predictive accuracy; it tends to perform poorly with respect to recommendation diversity.

Also in future work, we can consider the factors that can be challenges the recommendation system such as: Stability, cold start, scalability, synonyms etc. If we improve these factors with this approach then it will cause to improve the overall recommendation system.

Acknowledgment

This is a great pleasure & immense satisfaction to express my deepest sense of gratitude & thanks to everyone who has directly or indirectly helped me in completing my work successfully.

I am highly indebted to our Head of Department and Guide Prof. Patil Pramod for his guidance and constant supervision as well as for providing necessary information regarding my work.

My thanks and appreciations also go to my colleague in developing the paper and people who have willingly helped me out with their abilities.

References

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry, volume 35. ACM Press, http://doi.acm.org/10.1145/138859.138867, 1992. 61-70. [cited at p. 1, 10, 13]

[2] Paul Resnick and Hal R. Varian. Recommender systems, volume 40. ACM Press, http://doi.acm.org/10.1145/245108.245121, 1997. [cited at p. 1, 8]

[3] R. Mooney, P. Bennett, and L. Roy. Book recommending using text categorization with extracted information. In Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98, volume 8, 1998.

[4] J. Breese, D. Heckerman, C. Kadie, et al. Empirical analysis of predictive algorithms for collaborative filtering. In Proc. of UAI98, pages 4352, 1998.

[5] J. Herlocker, J. Konstan, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proc. of SIGIR99, pages 230237. ACM New York, NY,USA, 1999.

[6] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 7(1):7680, 2003.

[7] T. Hofmann. Latent semantic models for collaborative filtering. ACM Transaction son Information Systems, 22(1):89115, 2004.

[8] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J. Riedl, Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems, 22(1), pp. 5-53, 2004.

[9] S.M.. McNee, J. Riedl, J. A. Konstan, Being Accurate is Not Enough: How Accuracy Metrics have hurt Recommender Systems, Conf. on Human Factors in Computing Systems, pp. 1097-1101, 2006.

[10] K. Bradley and B. Smyth, Improving Recommendation Diversity, Proc. of the 12th Irish Conf. on Arti cial Intelligence and Cognitive Science, 2001.

[11] Feng Xie, Zhen Chen, Jiaxing Shang, Geo rey C. Fox, Grey Forecast Model for Accurate Recommendation in Presence of Data Sparsity and Correlation, ACM 1-58113-000-0/00/0010, RSWeb 12, September 913, 2012, Dublin, Ireland.

[12] Cacheda, F., Carneiro, V., Fernndez, D., and Formoso, V. 2011. Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Trans. on Web 5, 1(Feb. 2011), 2:1-2:33.

[13] Jonathan l. Herlocker, joseph a. Konstan, loren g. Terveen, and john t. Riedl evaluating collaborative filtering Recommender systems, ACM Transactions on Information Systems, Vol. 22, No. 1, January 2004, Pages 553.

[14] Ma, H., Zhou, T.C., Lyu, M.R., and King, I. 2011. Improving recommender systems by incorporating social contextual information. ACM Trans. On Information Systems 29, 2 (April 2011), 9:1-9:23.

[15] J. Delgado and N. Ishii, Memory-Based Weighted-Majority Prediction for Recommender Systems, Proc. ACM SIGIR 99 Workshop Recommender Systems: Algorithms and Evaluation, 1999.

[16] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J. Riedl, Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems,22(1), pp. 5-53, 2004

[17] D. Lemire, S. Downes, and S. Paquet, Diversity in open social networks, published online, 2008.

[18] J. Carbonell and J. Goldstein, The user of MMR, diversity-based reranking for reordering documents and producing summaries, Proc. of the ACM Conf. on Research and Development in Information Retrieval (SIGIR), pp. 335-336, 1998.

[19] S.T. Park and D.M. Pennock, Applying collaborative filtering techniques to moviesearch for better ranking and browsing, Proc. of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 550-559, 2007.

[20] B. Smyth and P. McClave, Similarity vs. Diversity, Proc. of the 4th Intl. Conf. on Case-Based Reasoning: Case-Based Reasoning Research and Development, 2001.

[21] Hsu, L.C. 2003. Applying the grey prediction model to the global integrated circuit industry. Technological Forecasting and Social Change 70, 6 (July 2003), 563-574.