Improved Scalable Recommender System

S. Ishiq, N. Majeed, M. Maqsood and A. Javed
Department of Software Engineering University of Engineering and Technology, Taxila, Pakistan

ABSTRACT
Recommender systems are known for their ability to recommend items which are new to the user by having some synchronization with the user’s personal interest. The importance of recommender systems leads to the creation of new approaches that can produce accurate results. As data become 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. However, RS also suffer from problems of sparsity, scalability, first-rater problem and unusual user problem. These issues affect 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. 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. 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 items 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 algorithms suffer from sparsity issues, cold start user and cold start product [6].

Both the above mentioned recommendation approaches have some limitations. To maximize recommendation scope and minimize its limitations, researchers established optimal solution known as Hybrid RS. 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.

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.
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 selection 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
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 \( M = \{ m_1, m_2, m_3, \ldots, m_y \} \). Item in dataset is denoted by n and set of items are represented by \( N = \{ n_1, n_2, n_3, \ldots, n_y \} \) 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 KMeansPlusLogPower

KMeansPlusLogPower [47] algorithm describes a new method of centroid selection as initial centroid selection greatly affects clusters outcome. KMeansPlusLogPower made use of distances and numbers of ratings as two bases for centroid selection. Centroid selection in KMeansPlusLogPower rely on the concept that any new centroid selected should have huge distance with existing centroids and probability proportional to the log of similarity. KMeansPlusLogPower requires a number of clusters as input. KMeansPlusLogPower 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

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

(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.

\[
\text{dist} = \begin{cases} \frac{1}{\text{MAX}} & \text{if } \text{sim} \neq 0, \\ \text{MAX} & \text{otherwise} \end{cases}
\]

(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 MAXDIST is used where MAXDIST 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

\[
p(x) = \frac{|u|}{|u_p|}
\]

(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.

\[
A_{mn} = U_{mn} S_{mn} V_{mn}^T
\]

(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.

\[
A_{mn} = U_{mk} S_{kk} V_{kn}^T
\]

(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

\[
r_{i,u} = U_k \cdot \sqrt{S_k}(u) \cdot \sqrt{S_k} V_k^T(i)
\]

(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 KMeansPlusLogPower 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: RecKSu

Input: User item matrix
Output: Set of recommendations
procedure KMeansPlusLogPower (no. of cluster)
Select initial centroid c1 to be a power user up
repeat
Select the next centroid c1 where ci = ū∈U with the probability
Prob = dist(u) + log \( \frac{1}{p(x)} + 1 \)
until k centroids are found
return \{c₁, c₂, c₃, ..., cₖ\} = k centroids
end procedure
procedure Cluster (u, k, iteration)
C = Centroid Select
a = 0
repeat
Set the cluster gⱼ for each j=1,2,..., k to be the set of users in U that is closer to cⱼ than they are to ci for all l ≠ j.
Set cⱼ, for each j=1,2,..., k to be the center of mass of all users in gⱼ, i.e. cⱼ = \( \frac{1}{|gⱼ|} \sum_{u ∈ gⱼ} u \)
\( a = a + 1 \)
until (C changes no more) OR (a = iteration)
return (C)
end procedure
if (C is large) then
repeat
SVD(cluster Ci)
until clusters are reduced
procedure User Representative(cluster Ci)
\( U_{power} = \{u₁, u₂ ... uₙ\} \)
return \{u₁, u₂ ... uₙ\}
end procedure
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 use 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 100K Ratings and Movie Lens 1M 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 100,000 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 \(1 - \frac{100000}{15986126}\) = 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, KMeansPlusLogPower. Results have shown that our proposed approach RecKSU produce results with slight improvement in MAE.

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MAE(100 K)</th>
<th>MAE (1 M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UBCFoff</td>
<td>.721</td>
<td>.766</td>
</tr>
<tr>
<td>KMeans</td>
<td>.745</td>
<td>.863</td>
</tr>
<tr>
<td>KMeansPlusLogPower</td>
<td>.740</td>
<td>.852</td>
</tr>
<tr>
<td>RecKSU</td>
<td>.710</td>
<td>.834</td>
</tr>
<tr>
<td>KMeansPlusLogPower</td>
<td>.738</td>
<td>.854</td>
</tr>
</tbody>
</table>

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 KMeansPlusLogPower. 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

<table>
<thead>
<tr>
<th>Number of clusters</th>
<th>Rec KSU (100 K)</th>
<th>Rec KSU (1 M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>.735</td>
<td>.844</td>
</tr>
<tr>
<td>80</td>
<td>.735</td>
<td>.841</td>
</tr>
<tr>
<td>100</td>
<td>.723</td>
<td>.840</td>
</tr>
<tr>
<td>120</td>
<td>.725</td>
<td>.839</td>
</tr>
<tr>
<td>140</td>
<td>.729</td>
<td>.841</td>
</tr>
<tr>
<td>160</td>
<td>.732</td>
<td>.844</td>
</tr>
<tr>
<td>180</td>
<td>.739</td>
<td>.844</td>
</tr>
<tr>
<td>200</td>
<td>.745</td>
<td>.845</td>
</tr>
</tbody>
</table>

5.4 Optimal number of Neighbors

Experiments on KMeansPlusLogPower 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.
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

<table>
<thead>
<tr>
<th>Number of user representative</th>
<th>MAE(100 K)</th>
<th>MAE (1 M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.729</td>
<td>0.850</td>
</tr>
<tr>
<td>6</td>
<td>0.738</td>
<td>0.843</td>
</tr>
<tr>
<td>7</td>
<td>0.764</td>
<td>0.839</td>
</tr>
<tr>
<td>8</td>
<td>0.79</td>
<td>0.839</td>
</tr>
<tr>
<td>9</td>
<td>0.831</td>
<td>0.841</td>
</tr>
<tr>
<td>10</td>
<td>0.832</td>
<td>0.849</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Imputed technique</th>
<th>MAE (100 K)</th>
<th>Number of dimensions(100 K)</th>
<th>MAE (1M)</th>
<th>Number of dimensions (1 M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zeros</td>
<td>2.32</td>
<td>12</td>
<td>2.40</td>
<td>26</td>
</tr>
<tr>
<td>Random</td>
<td>1.072</td>
<td>4</td>
<td>1.09</td>
<td>17</td>
</tr>
<tr>
<td>User average</td>
<td>.778</td>
<td>8</td>
<td>.759</td>
<td>22</td>
</tr>
<tr>
<td>Item average</td>
<td>.774</td>
<td>10</td>
<td>.730</td>
<td>22</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Number of dimensions</th>
<th>Rcm KSU(100 K)</th>
<th>Rcm KSU(1 M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.80</td>
<td>.94</td>
</tr>
<tr>
<td>2</td>
<td>.78</td>
<td>.92</td>
</tr>
<tr>
<td>3</td>
<td>.76</td>
<td>.91</td>
</tr>
<tr>
<td>4</td>
<td>.74</td>
<td>.89</td>
</tr>
<tr>
<td>5</td>
<td>.728</td>
<td>.88</td>
</tr>
<tr>
<td>6</td>
<td>.74</td>
<td>.87</td>
</tr>
<tr>
<td>7</td>
<td>.75</td>
<td>.85</td>
</tr>
<tr>
<td>8</td>
<td>.76</td>
<td>.839</td>
</tr>
</tbody>
</table>

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


