

Location and QoS based Web Service Recommender System

Afreen Rafiq

M.Tech, Dept of Computer
Science and Engineering
Stanley college of Engineering
and Technology for Women
Osmania University,
Telangana

Ayesha Ameen

Associate Professor, IT
Deccan College of Engineering
and Technology, Osmania
University, Telangana

Srinivasu Badugu, PhD

Associate Professor, CSE
Stanley College of Engineering
and Technology for Women
Osmania University,
Telangana

ABSTRACT

Web services are the components usually called as integrated software components which support the interoperable machine-to-machine interaction over any network. In today's world, the availability of web services is increasing actively. A recommender system is presented that helps the users to select services with good Quality-of-Service (QoS) performance. This recommender system extracts the information from two parameters called location and QoS values. The basic idea is to predict Web service QoS values and recommend the best one for users based on historical Web service QoS records. Firstly, Locations are clustered using Euclidean distance and then the similarity of users of that particular location is calculated. Likewise, Web services are clustered based on the similarity and then prediction is done based on the clusters formed. After Clustering, the missing data is being predicted using the Pearson correlation of nearest neighboring approach. Prediction is done from both service perspective and user perspective. Once missing data is obtained, services are ready to be recommended. Lastly, users are recommended with services which are being used by them in the history with better QoS values. QoS values (throughput and response time) help in finding best services to be utilized by the user. The system is evaluated using MAE, which is majorly used to evaluate the predicted values.

General Terms

Item based collaborative filtering techniques; User based collaborative filtering techniques.

Keywords

Recommender system, Collaborative filtering, mean absolute error.

1. INTRODUCTION

Web service is defined as method of communication between two electronic devices over network. Web services are widely used in business development and academia. It can be described as collection of open protocols. Web services are used for replacing data between applications. Software applications are written in various types of programming languages and running on various platforms which can be used in web services to transfer data over computer networks like the Internet. QoS (Quality of Service) [1] describes the behavior of services usually they are called non-functional properties of Web services (including security, response time, throughput, availability, portability, non repudiation and other attributes), which are vital factors for service requestors to distinguish identical Web services. In order to deal with information overloading phenomena it is necessary to use wide variety of techniques and tools that will ease us in

finding relevant information. This set of techniques and tools are provided by the research field of information retrieval. The Recommendation Process [2] describes concepts and terms used in recommendation systems now it is important to explain how these systems work and give recommendations to the users. In a general way every recommender system follows a specific process in order to generate recommendations. The process of recommendation is usually described as black box. In the case of audio or video recommendation systems information databases of huge dimensions are produced. The final representation of these recommendations depends on the recommender system [3] itself but it may range from specific ordered lists of items, images of the items, or the whole items.

Presently the size of data available to people is in huge quantity and it is growing every second. It is impossible to go through all the items and to resolve whether they can be useful or not, and thus finding exact or similar items becomes more difficult. There are many cases in which there are many options to choose. Suppose if a user want to book a table in restaurant from an on-line booking system then user gets thousands of web services. Since most of the restaurants web service may be not known to user, user cannot choose whether user would find related or not, but among all these restaurants web service, there must be some restaurants which user will like. Finding the relevant item by browsing through the millions of restaurants provided by the online booking system is impossible, since you probably cannot judge the item's importance without going. This problem is related to finding relevant restaurants, hotels, cuisines and many more.

The amount of usage of web services is increasing hugely. Users to large extent depend on web services for their business to run and return profit. Profits of business are indirectly dependent with the amount of quality work done by the workers and workers depend on various factors to work, among those factors one of the factor on which they rely is web services. If good quality of web service is given the business may achieve better performance. The recommendation process is done by recommending the best services to the users using collaborative filtering approach. This approach finds the similarity of users and recommends the web services which are of best quality.

Collaborative techniques [4] are best appropriate to problems where the density of user interest is comparatively high across a small and static set of items. If the set of items change rapidly, old ratings will be of little or no value to new users who will not be able to have their ratings compared to those of the existing users. If the set of items are large and user interest is too low, then the probability of overlap with other users will be very small. Collaborative recommenders [5] grind best for

a user who fits into a corner with many neighbors of similar taste. In Memory-based Approaches, User rating data is used to find similarity between users or items. This is used for making recommendation and these recommendations are used commercially. It is easy to implement and is effective. User-based, Item based and similarity fusion collaborative filtering techniques are widely used to generate recommendations. Memory-based [6] CFAs use the total or a sample of the user-item database to produce a prediction. By finding the neighbors of a new user, a prediction of preferences on new items for him or her can be generated. When the task is to generate a top-*N* recommendation, we need to find the most similar users or items after computing the similarities, and then combine the neighbors to get the top-*N* most repeated items as the recommendation.(i) User Based Collaborative Filtering Algorithm[7].(ii) Item Based Collaborative Filtering Algorithm[8]are memory based approaches of collaborative filtering to implement recommender system using past usage history of user.

2. SYSTEM ANALYSIS

2.1 Limitation of Recommender System

Selecting a best quality Web service among a large number of web services is a strenuous task. Rather than using publicly available web services some developer choose to implement to implement their own services, which gives additional overhead in time, money and resources. Using an improper service, may add budding risk to the business process. Therefore, useful approaches to service selection [9] and recommendation are in immediate need, which can help service users reduce risk of loss to the business and deliver high-quality business processes which may in turn return profits.

Therefore, various users may observe different QoS values of the same Web service [10]. In other words, QoS values calculated by one user cannot be employed directly by another user for selection of services. It is also illogical for users to collect QoS information by calculating all services by themselves, since employing real world Web service invocation is time and resource-consuming. Moreover, some properties of QoS e.g., reliability are difficult to be calculated as long-duration observation is prescribed. To strike this challenge, this paper presents personalized QoS value prediction for service where past usage experience of web services is used. This approach doesn't invoke additional Web services. Based on the predicted QoS values of Web services generated from client side and server side, personalized QoS Web service recommendations help users to select the ideal service among the functionally identical ones.

2.2 Proposed Approach

People depend on recommendations from other people by spoken words, reference letters, and news reports from news media, general surveys, travel guides. Recommender systems support and expand this natural social process to help people analyze through availability of books, news articles, web pages, movies and music, restaurants, jokes, grocery products, to find the most interesting and vital information for them. "Collaborative filtering (CF)", this approach was first coined by the first recommender system called Tapestry where rule-based recommenders and user-customization were included ,which has been widely followed by not concerning the facts that recommenders may not exactly collaborate with users and recommendations[11] may offer particular interesting items, in addition to indicating those that should be knocked(filtered)out. The central theory of CF is that if

users and rate items similarly, or have similar behaviors (e.g., purchasing, watching, listening, playing), and hence will rate on other items similarly.

2.2.1 Location Based Recommender

In classical recommender systems the recommendation process is done by taking user's past rating history and items features (tagging) and primarily no contextual information is taken for generating recommendations. However, there are many factors which influence users' decision in reality. Besides, demand of users' might vary with context as well (e.g. time of the place, location of the user and weather of that location etc) which cannot be gained through old recommender systems. As a result information about users, items and/or few contextual parameters are not sufficient to give precise real-time recommendations; however, more and precise information is needed in order to give items to users under given conditions. For example, adding the temporal, or context, in recommender systems would provide more precise recommendations which also might give back user's emotions as well. With the implementation of the new generation recommender system, contextual information has become one of the most important knowledge sources to enhance recommendations output and provide more user specific recommendations under that identical circumstance (i.e., contexts) are related with similar user tastes.

2.2.2 Process Logic

Pearson Correlation Coefficient: It is a measure of the power of a linear association between two variables and is denoted by *r*. Pearson correlation attempts to draw a line of most suitable data through the data of two variables, and the Pearson correlation coefficient, *r*, represents to what extent these data points are best suitable. The Pearson correlation coefficient, *r*, can take a range of values from +1 to -1. A value of 0 represents that there is no association (no similarity) between the two variables. A value greater than 0 represents a positive association (similarity); that is, as the value of one variable increases, so does the value of the other variable. A value less than 0 represents a negative association (dissimilarity); that is, as the value of one variable gain, the value of the other variable decline.

$$sim(a,b) = \frac{\sum_{i=1} (r_{(a,i)} - \bar{r}_a)(r_{(b,i)} - \bar{r}_b)}{\sqrt{(r_{(a,i)} - \bar{r}_a)^2} \sqrt{(r_{(b,i)} - \bar{r}_b)^2}} \dots\dots\dots \text{Equation [1]}$$

Where Sim(a,b) is the similarity between Web services a and b. The range of Sim(a,b) is[-1, 1]. Sim(a,b) = null when there is no user who has used similar services. Where ra,i represents that the QoS values of Web service i observed by service user a, ra and rb represent the average QoS values observed by service users a and b respectively. The PCC similarity of two service users, Sim(a, b) ranges from -1 to 1. Positive PCC value shows that two users have similar usage history Web service, while negative PCC value represents usage of web services to larger extent is opposite. Sim(a,b)=null represents that two users have not called common Web service. PCC only deals with the difference of QoS between services called by both users, which may overstate the similarity of two users that are dissimilar but appear to have services with very identical QoS records.

The main intention of this dataset which contains web service QoS values is to give real-world data for Web Service researchers. These web services are monitored by Planet-lab. This dataset is freely available for research purposes. This data set is available with the link www.wsdream.net.

3. SYSTEM ARCHITECTURE

Social networking sites and self publishing sites motivate users to share knowledge [12] and learn from others. This system applies the idea of collaboration of different and gives a platform for users to share observed Web service QoS values and search Web services. This system generates service recommendations based QoS values which were shared by users. The more QoS records users add, the more precise the recommendations will be, since more information can be utilized from the user contributed QoS values.

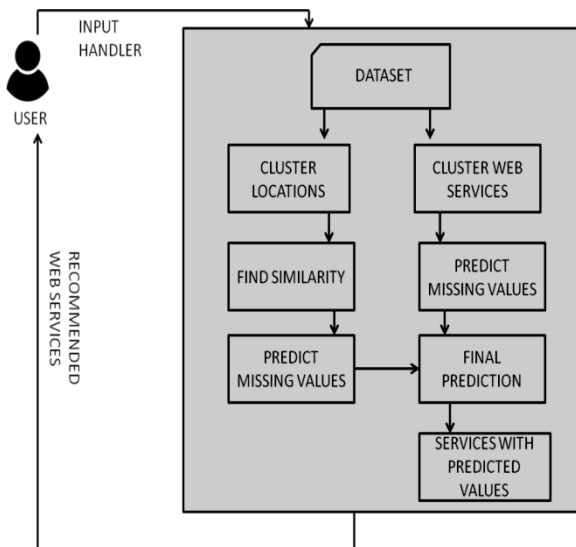


Figure 1: Architecture of the Recommender system

In this paper, it is believed that users are trustworthy. How to identify and handle malicious users [13] and incorrect QoS values will be addressed in the future work. Fig 1 shows the architecture of the recommender system, which includes the following implementation procedure:

- Users of web service log on to system and share observed Web service QoS readings with other users. In this paper, training users are the users who have submitted Web service QoS records to the system. If a training user needs Web service recommendation, then that training user becomes an active user. QoS values of training users will be utilized to extract personalized web service recommendation for the active user.
- This system clusters training users into different regions according to their physical locations and these clusters of locations are utilized to find similarity of users based on past web service usage experiences.
- Similar Web services based on their QoS similarities are also clustered. It also maps the active user to a user region where the active user belongs based on historical QoS and user location.
- This recommender system predicts the missing QoS values of Web services for the active user and recommends the best service to the user.
- At the end, the Active user acquire the predicted QoS values of Web services and also the

recommendation, which can be used in making decisions for choosing best service.

4. ALGORITHM

```

Algorithm: LQR
START
1. Obtain locations from data set and cluster the locations using Euclidean distance.
2. If the user location is not found see the cluster which is formed by nearest distance to that location.
3. Calculate similarity among the users of the particular location.
4. Calculate similarity among the web services based on QoS values.
5. Merge the similar services and form cluster of similar services.
6. Calculate Pearson Correlation for nearest neighborhood of the users within the given location to predict missing values.
7. Calculate Pearson Correlation for nearest neighborhood of the services within the cluster of services to predict missing values.
8. Obtain highest predicted value to each service.
9. Calculate weights.
10. Recommend web services to the users based on the past history.
STOP
  
```

Fig 2: Algorithm of the Proposed System

5. IMPLEMENTATION

In the field of service computing, Web service and its non functional properties called Quality of Service have been considered by number of researches in a number of research investigations for displaying the non-functional aspects of the web services. Ratings of some Quality of service (QoS) properties (e.g., response time) on the same web service differ quite differently from one user to other. For example, the response time of a service found by closely located users usually swings mildly around a certain value. On the other hand, the response times noted by users who are far away from each other sometimes differ significantly. Based on this observation, this recommendation algorithm takes location data into consideration to improve the recommendation preciseness. This recommendation algorithm is designed as a four phase process, i.e., 1) user region creation by clustering users and then finding similarity of users among the clusters, 2) service region creation by clustering web services based on similarity and 3) QoS prediction from user and service perspective 4) Recommendation of web services.

5.1 Clustering Location and Users

In this very first phase, users will be clustered [14] into different regions according to their locations. At the beginning, user's rough location is retrieved. User's location can also be extracted by their IP addresses. The location information displays country, city, latitude or longitude of user's location. Then users from the same place will be clustered together to form early or initial regions. These small regions will be assembled into large ones.

At the end clusters of countries which are very much nearer to one another are retrieved. Initially locations of users are taken as initial seed, once location are known they are kept as individual location or grouped with another location as the situation demands. It is known that when the two countries are nearer their web service usage experience may also be same to some extent. Locations are clustered based on similarity metric. For grouping the location of the user's, Euclidean distance metric (Equation [2]) is used employed.

After Regions are clustered, similarity of users is found based on the clustered regions. The similarity is calculated using similarity metric Pearson correlation Equation (Equation [1]) is which calculates the similarity between two users of a location. When finding the similarity of users a matrix is obtained and the QoS values of the users of the particular location are obtained and then the similarity metric is applied.

- Euclidean distance

$$dis = \sqrt{\sum_i (x_i - y_i)^2} \dots\dots\dots \text{Equation [2]}$$

5.2 Clustering Web Services

Item clustering techniques usually work by analyzing groups of items who appear to have similar ratings. Once clusters are formed, predictions for a target service can be made by taking average of the values of other items in that cluster. Some clustering techniques depict partial participation of each item in several groups. Once the item clustering is completed, performance can be very good, since the cluster size that must be identified is much smaller.

Firstly, web service’s similarity using (Equation [3]) is found by comparing with all the services and to retrieve the similarity of one service median value taken of all the values obtained during the findings of similarity.

At the end once the similarity values are obtained they are compared with all the other services. The similarity value which is very much nearer to the other service similarity value they are clustered. This process is repeated until all the services are clustered based on the nearer similar values of web service similarity values.

$$sim(x, y) = \frac{\sum_{i \in I} (r_{(x,i)} - \bar{r}_x)(r_{(y,i)} - \bar{r}_y)}{\sqrt{\sum_{i \in I} (r_{(x,i)} - \bar{r}_x)^2} \sqrt{\sum_{i \in I} (r_{(y,i)} - \bar{r}_y)^2}} \dots\dots \text{Equation [3]}$$

5.3 Prediction of Missing QoS values

The first two phases where user’s location and Web services are grouped based on their similarities. QoS predictions can be calculated from both service regions and user regions. With the usage of compressed QoS data, neighborhood searching and Predicting Web service QoS [15] for an active user can be calculated accurately than conventional methods.

5.3.1 User’s Perspective

Rather than computing the similarity between the each training user and active user, similarity between the active user with the region that user belongs is computed. More often, users who belong to same region likely to have identical QoS experience on the same Web service. In order to predict the unused QoS value in the dataset web service s for active user a, the following steps are to be followed:

- Identify the user region of active user a.
- For services, the prediction value will be generated considering QoS values submitted from similar regions. Eq. (1) is employed to calculate the similarity between the active user and users of that region that has evaluated service s.
- k most similar values with positive values c1; c2; . . . ; ck will be calculated. Once similarity has calculated then the positive values are taken into attention.

- When similarity values are calculated they are used for predicting the missing values of the given region (active user).
- If the active user’s region has QoS value of s, the prediction is computed using the following equation

$$r_u = r_{c,s} + \frac{\sum_{j=1}^k sim'(a, c_j)(r_{c_j,s} - r_{c_j})}{\sum_{j=1}^k sim'(a, c_j)} \dots\dots\dots \text{Equation [4]}$$

Where $r_{c,s}$ is the missing value column mean, $Sim'(a,c_j)$ is the similarity of the users who belong to one region. $r_{c_j,s}$ is the missing column value and r is the mean of the column.

- After applying this equation to the data set missing values are retrieved.
- The active user will be considered as a member of a new region if appropriate region is not found.
- When a user from new region is found cluster of that location is considered.
- The user from new region given the nearest region by knowing the user region with IP address. The user is provided with nearest region to which it belongs and given accordingly the QoS values.
- Once the QoS values are given then the same procedure is followed to calculate the missing values of that region.

5.3.2 Service’s Perspective

Clustering of Web services give another chance to view and use the data set. It can improve the prediction accuracy when one has very less knowledge of user tastes. To predict the missing QoS value of service s observed by user a from the service perspective, one should use the Web service clusters. According to this experiment, good prediction accuracy is obtained with the rough prediction [16] of missing values. To achieve a better prediction result following equation is used.

$$r_i = r_{a,c} + \frac{\sum_{j=1}^k sim'(s, c_j)(r_{a,c_j} - \bar{r}_{c_j})}{\sum_{j=1}^k sim'(s, c_j)} \dots\dots\dots \text{Equation [5]}$$

$r_{a,c}$ is the column missing value, $sim'(s,c_j)$ is the similarity of the services with the missing value service in the cluster of services which is also called as service region. r_{a,c_j} is the neighborhood service values \bar{r} is the mean of neighborhood services.

As in the user perspective the system has tried to predict the values in the same way have to be applied in service perspective.

- When calculating the predicted values, the clusters of similar services are considered, which were made in the phase two of recommender system.
- These clusters are now taken into consideration and the then the prediction is done within the clusters of the services.
- Initially the clusters are called and then the services which are belonging to the particular group.
- Then the similarities of the services which are the members of the clusters are calculated.
- Then the Pearson correlation of nearest neighbor formula is applied to calculate the missing values.

- This Predicted values are now replaced with the missing values.

5.3.3 Generating Prediction

For user a, the final prediction QoS of service s consists of two parts: prediction from user perspective and from service perspective

$$r_{a,s} = \omega * r_i + (\omega - 1)r_u \dots\dots\dots \text{Equation [6]}$$

Where (r_i) is the QoS prediction generated from user regions, (r_u) is the prediction from clusters formed from Web services, and parameter ω decides how much the present system is dependent on prediction result. The table 1 shows final prediction values.

Example:

Table 1: Services and the Throughput and response time values

Service	Values
0	13.23127
1	3.188253
2	3.447221
3	2.907756
4	2.908822
5	8.71850
6	37.15808
7	3.90343392
8	4.331973
9	3.343392
10	6.0803

5.4 Recommendation

- In order to recommend web services it is necessary to save all the predicted final values of all the services.
- Once the final values are ready recommendation process becomes very easy.
- Recommendation can be done region wise or user wise.
- Initially the user is given a login page.
- The user needs to submit the login credentials with the location name.
- Once the login credentials are given they are verified.
- Once they are verified they are then directed to the Home page.
- The user is given with the web services and the QoS values.

6. RESULT ANALYSIS

6.1 Evaluation Metrics

A recommender system uses different types of measures to evaluate the quality of a recommender system.

- Statistical accuracy metrics, these metrics evaluate the accuracy of a system. They compare the numerical recommendation values [17] against the actual ratings of user for the user-item pairs in the training dataset.

Mean Absolute Error (MAE) it is a measure of the deviation of recommendations from their true user-specified values. For

each pair of ratings-prediction $\langle p_i, q_i \rangle$ this metric pretends that the absolute error between them i.e., $|p_i - q_i|$ is equal. The MAE is calculated by taking sum of these absolute errors of the N corresponding ratings-prediction pairs and then determining the average.

The lower the MAE, the more accurately and precisely the recommendation engine predicts user ratings. In terms of this experiment, MAE is used to calculate the complete recommendation accuracy. Experiments were started by dividing the data set into training and a test portion. Initially 500 web services were considered and these services were divided as test data and training data along with these services different users of various locations were taken but when recommendations were completed, users of one particular location were considered. Performance of prediction were compared by entering the training ratings set into a recommendation engine of collaborative filtering that uses the Pearson nearest neighbor algorithm (user-user). Flexible prediction engine was implemented with user-based CF algorithms. The Pearson's nearest neighbor algorithm was configured to deliver the highest quality prediction without concern for performance.

6.2 Experiment Results

In this section experimental results were presented by applying collaborative filtering techniques for generating predictions. Amount of services on the internet, when compared with the number of services consumed by each user is very small. Generally the data set of recommender systems will be very sparse. This system examines how data sparseness affects the results of prediction from two aspects: the density of training matrix which shows how many QoS records are accumulated from all the users which are present in dataset and QoS values contributed by active users.

6.2.1 Comparing Performance:

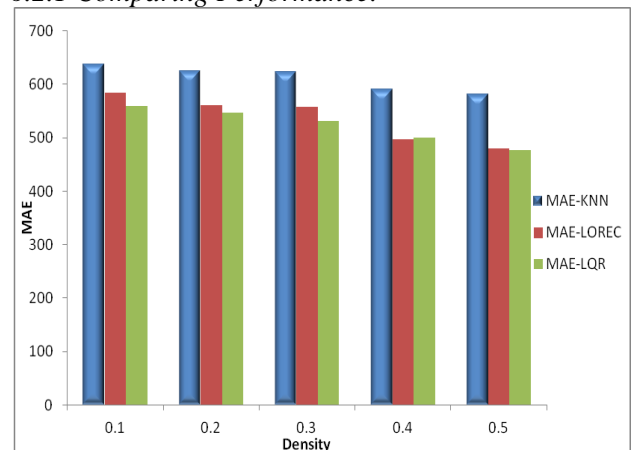


Figure 3: Comparing Performance

In this experiment, one should randomly remove 80 percent throughput values of the initial training matrix to generate a sparse matrix with density 20 percent. Now vary the number of throughput values submitted by active users from 10, 20 to 50. The removed records of active users in order to make the matrix dense are used to study the prediction accuracy. 500 services are taken, and then these services are now divided into training and test matrix. As said before 80% of 500 are 400 services are treated as training matrix with 349 users and remaining services are treated as test matrix. As from the above graph fig.3 it is explored that besides region KNN and LoRec, this system has better accuracy as the accuracy when

compared to region KNN is good but with LoRec its better. In region KNN the top K neighbours are taken to predict the missing values where as in LoRec, doesn't use Euclidean distance. But LQR which is "Location and QoS based web service recommender system" which uses Euclidean distance in initial process for predicting user values of QoS.

6.2.2 User Prediction Performance

When 0.1% density is applied to matrix, it will make matrix 10 % dense and 90% sparse. Whenever the matrix is denser the MAE high with other densities as the system gets users with some QoS values which help in finding similarity of the user.

The prediction of user gets changes with the change of location so the users of particular location considered for evaluation. From the fig.4 it is inferred that whenever the matrix is denser the Mean absolute error becomes low. This is because when make matrix 10% denser we get 90% sparse matrix. This 90% sparse matrix has very less QoS values. With less QoS values in each row the system may not get correct values of similarity as the values are very much less and as it is known that these values are used for predicting the missing values. Thus when the density increases the sparsity of the matrix decreases which gives the less MAE values. With less MAE values the performance of the recommender system increases.

Likewise for 80% training matrix it is needed to make matrix dense with the density of 0.2 which means that matrix should be populated with only 20% data. Remaining 80% should be kept as missing data. As the missing data decreases MAE also decreases. From the graph it clearly explains the whenever the density increases the MAE decreases. It is inferred that the MAE is very much less when compared to the 70% training matrix. And the curve is decreasing in nature as the density increases.

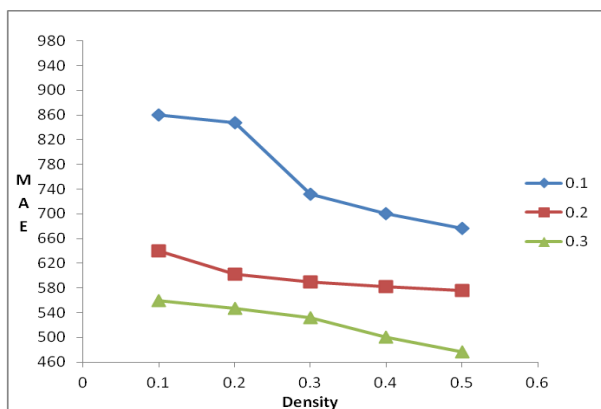


Figure 4: Density v/s MAE

6.2.3 Prediction Accuracy of Service Prediction

This graph depicts how the prediction accuracy changes in the item prediction. The accuracy is tested with training matrices of 70%, 80% and 90% of 500 services. Once the training matrix is obtained with 10% density, 20% density to 50% density, one can see that with 30% test data which is actually 70% training matrix higher MAE values are obtained as the matrix is very much sparse and system can't find similarity. Similarly the 80% training matrix has less MAE when compared to the 30% test data. Also the graph fig. 5 depicts that much less MAE is obtained with 90% training matrix. With the curves it is clearly seen that whenever the matrix is dense the MAE is decreasing.

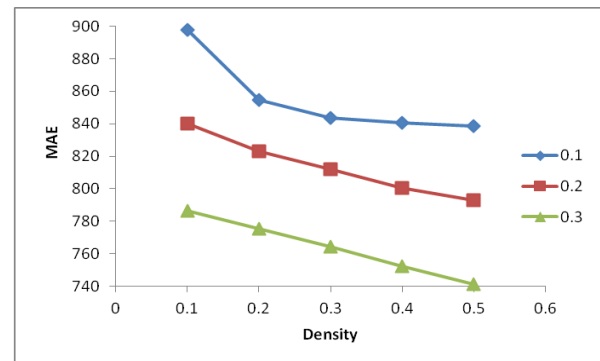


Figure 5: Density v/s MAE

6.2.4 Impact of threshold value

Parameter ω makes prediction method more flexible and adaptable to different data sets. With $\omega=1$, predictions are mainly based on user information, and if $\omega=0$, Web services information is consider. The weights applied to the user prediction and item prediction gives the final prediction values which help the user to get better performance.

Initially, the training matrix of 90%, 80% and 70% are taken. When 90% training matrix is applied and user prediction and item prediction is calculated, 0.1 to the user prediction and 0.9 to the service prediction is applied. When this is done with 90% matrix stable MAE is obtained. Final predictions with 10% test data stable MAE is achieved. But when training matrix is changed with 20% test data curve with increasing MAE is seen in fig. 6. However, for a dense data set, better prediction accuracy is achieved with smaller ω , which means more information provided by similar Web services.

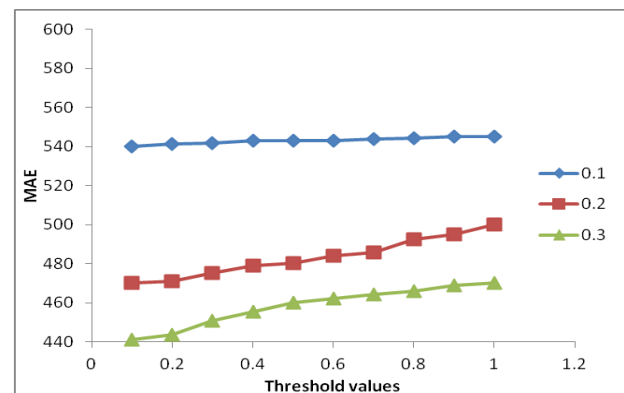


Figure 6: Threshold v/s MAE

6.2.5 Actual and estimated values

When item Prediction file is run with response time values following values are obtained. The actual values and the predicted values are being displayed in the table.2 which indicates that proposed system retrieves much similar values with the existing system.

Table 2: Actual value and Predicted value

Actual value	Predicted value
5.394	2.402714
0.581	0.5808
0.552	0.7303
0.405	0.491
0.865	0.614
0.938	0.2646
2.271	0.7078

One can see that the predicted values and actual values in the table. The actual values are tested and when the system is run predicted values are obtained.

7. CONCLUSION AND FUTURE WORK

Recommender systems are a powerful and robust new technology for generating further value for a business from its user databases. Recommender systems help users in finding items they want to purchase or utilize from a business. These systems benefit users by letting them to discover items they like. Indirectly, these systems help the business to generate more sales. These systems are swiftly becoming a vital tool in E-commerce on the Web. They are being stressed by the large volume of user information in existing collaborative databases. New technologies are in need that can change the scalability of recommender systems.

Location and QoS based Web service recommendation predicts Web service QoS values and it recommends the best service to active users by considering past Web service QoS records. This system combines prediction results which are generated from service regions and user regions, which gives better results than existing methods. It is also found that the combination result of user based and item based approaches is much better than the result obtained from the prediction generated from user regions or the prediction extracted from Web service regions. The two methods analyze the problem from two different perspectives and the combination of them prevents the error of individual methods.

The work presented in the paper can be extended for improvement in number of ways to give researchers scope for future research in this area.

- In terms of the clustering; probabilistic algorithms like EM can be considered to improve scalability of the recommender system. EM needs one scan of the database with limited amount of memory.
- To improve the recommendation accuracy, the recommender system can utilize contextual information as it can influence Web service QoS performance to larger extent.
- Similar to results of Web page search, users may only end up in considering the top three or five recommended services. Thus improving the accuracy of these top recommended services is another task to examine.
- This work can be extended by finding Correlation between different QoS properties, and detecting malicious or false users with false QoS information.

8. ACKNOWLEDGMENT

I thank Almighty without His grace this project would not have been possible. I am obliged to my family for helping supporting and encouraging me throughout my project. My sincere thanks to Mrs. Ayesha Ameen for helping giving precious support for completion of the project.

9. REFERENCES

- [1] N. Thio and S. Karunasekera, "Automatic Measurement of a QoS Metric for Web Service Recommendation," in Proc. Australian Softw. Eng. Conf., 2005, pp. 202-211.
- [2] Ian H. Witten, Mark A. Hall, Eibe Frank, "Data Mining: Practical Machine Learning Tools and Techniques(Third Edition)" Elsevier.
- [3] Joost de, "Evaluating Recommender Systems," TNO Information and Communication Technology, Delft, May 16, 2008
- [4] Xiaoyuan Su and Taghi M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, pp. 1-19, Aug. 2009.
- [5] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," in Proc. 22nd Int'l ACM SIGIR Conf. Res. Dev. Inf. Retrieval, 1999, pp. 230-237.
- [6] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Model. User-Adapt. Interact., vol. 12, no. 4, pp. 331- 370, Nov. 2002.
- [7] R. Jin, J. Chai, and L. Si, "An Automatic Weighting Scheme for Collaborative Filtering," in Proc. 27th Int'l ACM SIGIR Conf., 2004, pp. 337-344.
- [8] M. Deshpande and G. Karypis, "Item-Based Top-n Recommendation Algorithms," ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 143- 177, Jan. 2004.
- [9] R.M. Sreenath and M.P. Singh, "Agent-Based Service Selection," J. Web Semantics, vol. 1, no. 3, pp. 261-279, Apr. 2003.
- [10] M. Ouzzani and A. Bouguettaya. Efficient access to web services. *IEEE Internet Computing*, 8(2):34–44, 2004.
- [11] Mohd Abdul Hameed, Omar Al Jadaan, and S. Ramachandram, Collaborative Filtering based Recommendation System- A Survey. *IJCSE*, Vol. 4 No. 05 May 2012.
- [12] Christopher D. Manning, Prabhakar Raghavan ,Hinrich Schütze," An Introduction to Information Retrieval",Cambridge University Press ,Cambridge,pp 377-401,2009
- [13] J. Canny, "Collaborative Filtering with Privacy via Factor Analysis," in Proc. 25th Int'l ACM SIGIR Conf., 2002, pp. 238-245.
- [14] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen, "Scalable Collaborative Filtering Using Cluster-Based Smoothing," in Proc. 28th Int'l ACM SIGIR Conf. Res. Dev. Inf. Retrieval, 2005, pp. 114-121.
- [15] Hao Ma, Irwin King and Michael R. Lyu," Effective Missing Data Prediction for Collaborative Filtering" SIGIR 2007 Proceedings, Routing and Filtering pp 39-46,2007
- [16] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," in Proc. 14th Annu. Conf. UAI, 1998, pp. 43-52.
- [17] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. (2000). Analysis of Recommendation Algorithms for E-Commerce. In Proceedings of the ACM EC'00 Conference. Minneapolis, MN. pp. 158-167