International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 22 Issue 1 – MAY 2016.

ENHANCING THE ACCURACY OF ITEM BASED SIMILARITY IN WEB SERVICE RECOMMENDATIONS

Devendra Singh[1], Tejaswini Choday[2], Aamir Bilal Mohammed[3], Anirudh Rathinam[4]

Department of Computer Science and Engineering, SRM UniversityRamapuram, India

[1]devendra.yadav2410@gmail.com
[2]tejaswini@gmail.com
[3]aamirdbx@gmail.com
[4]anirudh.rathinam@gmail.com

***Abstract*–Collaborative filtering (CF) is a technique that is often used to make web service recommendations by filtering suitable candidate services according to the preferences of a user. Over the past few years, several attempts have been made to improve the accuracy of predictions despite this, the methods still require significant improvement. This paper proposes a location-aware memory based CF approach that takes into account the influence of an item on a given user. This is done by comparing the similarity of a candidate item with the or services from the user history. In addition, it also considers the location of users and services by ranking the possible candidates for recommendation based on the closeness to the selected user.In this way the recommender is able to provide personalized recommendations to the user by enhancing the accuracy and the quality of predictions made.**

I. INTRODUCTION

Web services are applications that are designed to assist users and support machine to machine interactions over a network. The increase in influence of Service Oriented Architecture (SOA) has resulted in a significant increase of the number of web services over the past decade [2]. Due to the large number of available services, it is necessary to come up with a method to identify the required services accurately and make recommendations to the user. Non functional requirements such as the response time and the throughput must also be taken into account in addition to functional requirements such as user feedback or active user ratings [3][4]. Existing recommendation systems are classified as content-based systems [10] [11], link prediction-based systems [12] [13], and CF based methods [7] [8] [9].
Collaborative Filtering (CF) is a widely used model to provide web service recommendations [5]. However existing approaches rarely consideruser interests and preferences. Due to lack of information about a user's preference, accurate recommendations will not be made.Due to most recommenders placing an emphasis on Web services' quality factors and ignoring users' interests [1] there is a possibility that the functional requirements for the user may not be satisfied. Moreover, very few models consider the location of the user while making recommendation predictions [2]. The location plays a vital role in determining the quality of non functional requirements and hence must be taken into account for accurate predictions.In this paper, we consider both the location of the user and a user's history while making recommendations. Hence the contribution of this paper is three fold: (i) We propose a model to enhance the accuracy and subsequently the quality of the predictions made. (ii) The proposed model takes into account the location of the user in order to find other similar users. (iii) The recommender takes into account the user history to gauge the influence of a candidate item for recommendation.

The paper is organized as follows: In Section 2 we explain a brief overview of the model. Section 3 explains the operation of the location handler module that keeps track of the user location. In the next section Section 4, we explain the prediction system and how the influence of an item may be calculated. Section 5 concludes the paper along with a statement of future work.

II. MODEL OVERVIEW

Based on the traditional CF approaches, several enhanced methods have been proposed to improve the prediction accuracy. Wu et al. [14] improved the CF method by smoothing the available datacontained by the user-service QoS matrix. The reputation of users while making recommendations was included in the system proposed by Qiu et al. [15]. Chen et al. [16] proposed a scalable CF method by recognizing the influence of the location in recommendation systems by grouping users into a set of regions according to their IP addresses and QoS similarities. In this paper we use this to group users according to the location.

Due to web services being deployed on the internet, factors such as the throughput and response time that are essential QoS values, depend on the underlying network's performance [6]. For example, when the user and service are located far from each other, the performance will be negatively affected due to bandwidth limitations and transfer delay. On the other hand, if they are located close to each other, it is likely that the

International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 22 Issue 1 – MAY 2016.

QoS values will be high for the user. Hence the location is a crucial factor affecting the QoS [2].Once the location has been obtained, we may then go on to find similar users and construct a neighborhood on whose basis we may determine the items or service to be recommended. The neighborhood is determined by selecting the k number of most similar elements using the Top-K similar neighbor selection algorithm which is often employed by memory based CF systems [8]. Similar users are chosen on the basis of a variation of the Pearson's Correlation Constant (PCC). Here we use a weighted PCC approach on the basis of user location to construct a neighborhood of k number of users.After the similar user neighborhood has been completed, we obtain highly rated items that act as candidates. For each of these, we evaluate the similarity of the item with other items in the user history with the help of which weights are assigned to them. The item with the highest weight is deemed most similar and is then submitted for recommendation. A diagram representing this is shown below.

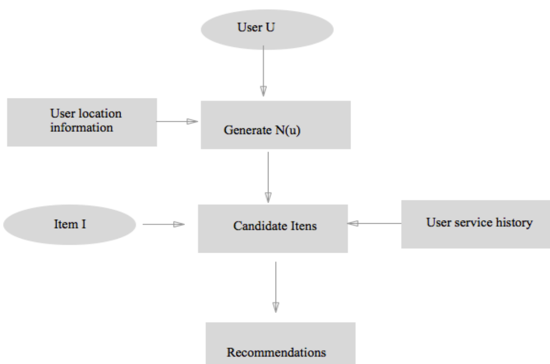

Figure 1: Overview of the System

The steps are as follows:

1: Get location of user u and choose an itemi
3: Sort other users according to their closeness to u
4: Select the top k users to form neighborhood N(u)
4: Sort N(u) according to similarity to user u
5: Get candidate services from N(u) using i
6: Set candidate weights according to u's history
7: Select the top k items for recommendation

Hence in this way, the recommender makes the use of the location and the user history to make personalized web service recommendations.

III. USER HANDLER

The user handler is a module that constructs a neighborhood of k number of similar users by taking into account their location. This paper makes the use of the freegeoip API to extract this information based on a user's IP address. The user location can be extracted in the form of JSON data. The location information obtained includes the zip code data, city, country etc. Here, we only consider similar regions based on the city or the zip code, same countries and different countries for location information. After the user data is extracted, we use PCC to find similar users. The mathematical representation is shown below. It measures the degree of similarity between two users, u and v by stating that they are similar if the rating they have given to all common items are similar.

$$PCC(u,v) = \frac{\sum_i (r(u,i)-\bar{r}(u))(r(v,i)-\bar{r}(v))}{\sqrt{(r(u,i)-\bar{r}(u))^2} * \sqrt{(r(v,i)-\bar{r}(v))^2}}$$

Here, u and v are the users between which the similarity is computed, r = rating given by a user and r' = average rating of all items given by user. The PCC formula may be improved by adding weighted values to it. This is represented by the notation Sim(u, v) as shown below. Here, the weight is multiplied to the PCC formula to avoid overestimation of similarity (McLaughin and Herlocker)[17]. Here, r is a minimum threshold value and I_u, I_v are the items of users u, v.

$$Sim(u,v) = \frac{\max(|I_u \cap I_v|, r)}{r} * PCC$$

The equations above may be used to generate a list of users by sorting them according to their degree of closeness to the current user u. All this is done according to the similarity between user u and v ∊ U. If the requirement of k number of users is fulfilled, then we end else we move on to users within the next zone (according to degree of closeness to the user) and repeat. This process is repeated until we obtain k number of total users. The algorithm is as follows:

Algorithm 1: Neighborhood generation
Input: User u; Number of neighbors k; Set of users $\hat{v}$; Items in neighborhood count
Output: Neighborhood N(u)
1: While count < k:
2: For each v ∊ $\hat{v}$ with same region do:
3: Get Sim(u, v) and append to list y
4: Sort y according to similarity
5: Append(N(u), v) where v ∊ y
6: Repeat 3-5 for users with same country
7: Repeat 3-5 for users with different countries
8: If count < k
9: End while
10: Return N(u)

Hence in this way we may obtain a neighborhood of users. Here, the number of users in the neighborhood is initially 0. The neighborhood is gradually filled on the basis of the location of other users, we keep generating similar users until the requirement of k users has been fulfilled. A potential problem that may be occur here is the possibility of sparse datasets. If for instance even after taking into account the total number of users in the country the requirement of k users is not met, we get this problem. In such cases, we simply exit out of the loop adding as many users as possible. This is shown in line 8 of the algorithm. After the neighborhood of users are generated, we can use it to recommend items based on similarity.

IV. ITEM RECOMMENDATIONS

The next step is the recommendations of items to user u. This is done after the neighborhood of similar users is obtained. This method takes into account the history of the user u to assign a weighted value to the item being recommended. This weighted value is the cumulative influence that the particular item has on the user u since it takes into account all the items held by u. The more similar the item is to the items in user u's history, the higher the weighted value. The most recommended items are then given out as recommendations to the user.

The procedure is described as follows: Generate list of recommended items I based on j, a highly rated service for user u and process them individually. Here, the current item being processed is I_C. Let the list of items in user u's history be I_u. For each itemi in I_u compute similarity between I_C andiand generate the weights. Store this in I_l. Sort the items in I_l in descending order and select k number of required items. If total number of items is less than k, it indicates that we have the sparse data problem in which case we must select and return all items recommended.

A) Similarity Computation

The initial list of items may be generated using a variation of the PCC formula. The items can be limited using a threshold such that only items with a high degree of similarity are selected to be appended to the item neighborhood. Here the weight w represents the influence of the item on the basis of the user's history. The PCC formula is given as follows.

$$\text{PCC}(i,j) = \frac{\sum_u (r(u,i)-\bar{r}(i))(r(u,j)-\bar{r}(j))}{\sqrt{\sum_u (r(u,i)-\bar{r}(i))^2} * \sqrt{\sum_u (r(u,j)-\bar{r}(j))^2}}$$

Here, the summation is done for all common users shared by items i and j. To enhance the similarity of the PCC, we use Inf(u, v) to represent the cumulative similarity based on the user history.Here,Inf refers to the influence that a particular item that is to be recommended as it considers all items in the user's history. The mathematical representation for Inf(i, j) where i and j represent the items are given by the following formula.

$$\text{Inf}(i,j) = \frac{w * \sum_u (r(u,i)-\bar{r}(i))(r(u,j)-\bar{r}(j))}{\sqrt{w * \sum_u (r(u,i)-\bar{r}(i))^2} * \sqrt{w * (r(u,j)-\bar{r}(j))^2}}$$

B) Computation of Weights

The weighted values in this case depend on the history of the user u that is stored in the list I_l. The formula for w is shown below:

$$w = c * w_0 + (1-c) * w_1$$

Here c represents a constant and the values for w_0 and w_1 are:

$$w_0 = \frac{\sum_{i \in I_l} PCC(j, u_i)}{|I_u|}$$

Here for every item i in the history list of user u, we calculate the similarity between the item I and the item being recommended j. This is then normalized between 0 and 1. w_1 is calculated as follows.

$$w_1 = \sqrt{\sum_{u \in U} \frac{n(u,i) - \bar{n}(u)}{|u_i|}}$$

This gives us the standard deviation for the items which is normalized using the formula shown below. The method for the computation of w_1 is similar to the weights generated in method [2]

$$n(u,i) = \frac{r(u,i) - \min r(i)}{\max r(i) - \min r(i)}$$

Thus in this way we may compute the weights using the appropriate algorithm. The next algorithm shows how the item recommendations are generated. The algorithm is as follows:

Algorithm 2: Service Recommender
Input: User *u*; item *j*;I_u, the items in u's history; Neighborhood N(u), similarity threshold t
Output: L, a list of k service recommendations
1: For each v ∈ N(u):
2: For each item i of user v
3: If PCC(i, j) > t
4: Append (I_r, i)
5: For each $I_C \in I_r$ do:
6: Computes =Inf(I_C,i)
7: Append(I_l, *s*)
8: Sort I_l in descending order
9: Return k items from $I_l$for recommendations

In this way we can take into account both the location of the users (from the neighborhood) and the history produce item recommendations.

V. CONCLUSION

This paper presents a personalized location-aware CF method for User-based Web service recommendations by considering a user's history. The paper aims at improving the item/service prediction performance, and hence this by incorporates the location of users by creating a similar neighborhood of users so as to satisfy non functional requirements and the user history by comparing the services. This helps the recommender gauge the functional interests of the user including the user's preferences while making the recommendations.

The paper opens up research opportunities in service computing by proposing new ways to perform the user based approach to collaborative filtering. The same can be applied to services as well in order to form a hybrid memory based CF approach. In the future, we will take more detailed location based information into consideration such as the Internet's AS topology. We will also consider incorporating the time and performance factors as well as theQoS prediction, and obtain real world datasets to evaluate our methods.

REFERENCES

[1] AWSR: Active Web Service Recommendation Based on Usage History Guosheng Kang, Jianxun Liu, Mingdong Tang, Xiaoqing (Frank) Liu, Buqing Cao, Yu Xu

[2] "Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation" - Jianxun Liu, Mingdong Tang, Zibin Zheng, Xiaoqing (Frank) Liu, SaixiaLyu

[3] L.-J. Zhang, J. Zhang, and H. Cai, Service Computing. Springer and Tsinghua University Press, 2007.

[4] M. P. Papazoglou and D. Georgakopoulos, "Service-Oriented computing," Communications of the ACM, 2003, pp. 46(10):24–28.

[5] Y. Jiang, J. Liu, M. Tang, et al. "An effective Web service recommendation based on personalized collaborative filtering". In Proceedings of International Conference on Web Services (ICWS2011). IEEE Computer Society, pp. 211-218, 2011.

[6] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-aware overlay construction and server selection," in Proc. 21st IEEE INFOCOM, New York, NY, June 2002.

[7] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering, " in Proc. 5th International Conference on Web Services, 2007, pp. 439-446.

[8] Z. Zheng, H. Ma, M.R. Lyu, and I. King. "WSRec: A Collabora-tive Filtering Based Web Service Recommendation System," in Proc. 7th International Conference on Web Services , Los Angeles, CA, USA, pp. 437-444, 2009.

[9] Z. Zheng, H. Ma, M. R. Lyu, and I. King "QoS-Aware Web Service Recommendation by Collaborative Filtering", IEEE Trans. on Services Computing, 2011, vol.4, no.2, pp.140-152.

[10] G. Kang, J. Liu, M. Tang, X. Liu, B. Cao, and Y. Xu, "AWSR: Active web service recommendation based on usage history," in Proc. Int. Conf. Web Services, 2012, pp. 186–193.

[11] L. Liu, F. Lecue, and N. Mehandjiev, "Semantic content-based recommendation of software services using context," ACM Trans. Web, vol. 7, no. 3, pp. 17–20, 2013.

[12] K. Huang, Y. Fan, W. Tan. Recommendation in an Evolving Service Ecosystem Based on Network Prediction. IEEE T. Automation Science and Engineering 11(3): 906-920 (2014)

[13] Y. Ni, Y. Fan, K. Huang, J. Bi, W. Tan. Negative-Connection-Aware Tag-Based Association Mining and Service Recommendation. ICSOC 2014: 419-428.

[14] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. Zhou, and Z. Wu, "Predict-ingQoS for Service Selection by Neighborhood-based Collaborative Filtering", IEEE Trans. on System, Man, and Cybernetics, Part A, 2013, vol. 43, no. 2, pp. 428-439.

[15] W. Qiu, Z. Zheng, X. Wang, X. Yang, and M. R. Lyu, "Reputation-aware QoS Value Prediction of Web Services", in Proc. 10th IEEE International Conference on Services Computing , Santa Clara Marriott, CA, USA, June 27-July 2, 2013, pp.41-48.

[16] X.Chen, X.Liu, Z. Huang, and H. Sun, "RegionKNN: A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation", in Proceedings of International Conference on Web Services, Miami, FL, USA, pp.9-16, 2010.

[17] M.R.McLaughlin, J.L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," SIGIR, 2004.