

## Location-Aware Collaborative Filtering for Web Service Recommendations Based on User and Service History

Balika J Chelliah<sup>1,\*</sup> and K. Vivekanandan<sup>2</sup>

<sup>1</sup>Assistant Professor, Department of Computer Science and Engineering, SRM University, India.

<sup>2</sup>Professor, Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India

Received 19 March 2017; Accepted 11 September 2017

### Abstract

The proliferation of technologies based on web services in the past few years has driven the exponential growth of the number of services available to the user. Due to the large number of candidates, it is difficult for a user to select the service best suited to their needs. Thus it is of paramount importance to devise a strategy to recommend the appropriate service to a given user. Collaborative Filtering (CF) is a widely employed technique to filter relevant data in Web Service Recommendations (WSRs). Although several CF based WSR techniques have been proposed over the past few years, their performance still requires significant improvement. In this paper, we propose a CF approach that leverages the location of the users for the filtering process. This ensures a greater measure of similarity between users to aid in making recommendations. Moreover, we also consider the history of the user and the web service to produce accurate recommendations. This is done by assigning weights to a candidate based on the user or service history to produce a similarity measure allowing the system to accurately determine the preferences of a user and make recommendations accordingly. The results of our proposed method are simulated through a set of comprehensive experiments performed on a real world Web Service dataset that is used to determine its performance.

*Keywords:* Service History, Recommendations

### 1. Introduction

The proliferation of Service Oriented Architecture (SOA) has driven the exponential growth of Web Services consequently increasing the number of services available to the user over the past decade [2]. These services play a significant role in domains including e-commerce and enterprise integration systems [1] and have garnered attention from both academia and the industry [4][5]. The large number of services that are now available to a user has resulted in the need to produce good recommendations to aid the user. WSR is a process that involves proactively searching and recommending services to users. To this end, there exist several ways by which this may be done. There are three major approaches to produce recommendations. These are (i) CF based approaches, (ii) Content-Based approaches and (iii) Hybrid approaches. This paper introduces a location-aware CF approach for accurate service recommendations.

CF is a widely used model to predict missing Quality of Service (QoS) values [6] and can be subcategorized into memory-based approaches and model-based approaches. The popularity of the memory-based CF approach stems due to the ease of interpretation of results [8] and is further divided into user based approaches [9] and item based approaches [10]. User-based CF recommends items preferred by the users with similar interests to a user, while

item-based CF methods recommends similar items to those that a user had preferred in the past [1]. Since very few recommendation models consider the location of the user [2] while making predictions, the accuracy of predictions might be reduced. Moreover, many recommenders fail to consider the user and service specific conditions. Our proposed method offers a solution to these drawbacks through a location-aware CF model where the recommender is aware of user and service specific conditions through their history. As such the contribution of this paper is three fold.

1. We propose a model to enhance the accuracy and subsequently the quality of the predictions made.
2. User location is considered when finding similar users to make recommendations
3. The recommender utilizes user history to make informed decisions.

The remainder of this paper is organized as follows. In Section 2, we give a brief overview of the proposed framework including descriptions of modules used by the recommender and their various functions. Section 3 describes the user location handler and the formation of the neighborhood of similar users. The module keeps track of users' locations using information such as their AS numbers to keep track of regional information and their country name to enhance the quality of predictions. The candidate users are weighted depending on the average QoS value taken from the item or service history. This is done because we wish to consider the impact of a particular recommendation. For example, if a service has a history of high QoS values but for a user from a particular location, the QoS value is

\*E-mail address: balika888@gmail.com

ISSN: 1791-2377 © 2017 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.

doi:10.25103/jestr.105.23

low, it can be inferred that the service will not produce as good a QoS value for users of that location hence resulting in a smaller weight. In the next section - Section 4, we describe how the services are recommended to users. Similar to the previous section, this is a location aware system that considers the user history. The history of the user helps the recommender in making informed decisions by providing a weighted recommendation for item similarity. Section 5 explains how the user based and item based information is integrated to predict missing QoS values that may be used for recommendations. The performance evaluation is shown in Section 6 where the experimental results are derived using a real world dataset to verify the effectiveness of our method. We conclude the paper in Section 7 by summarizing the proposed model and by providing a statement of future work.

## 2. Overview of the Proposed Model

Collaborative Filtering (CF) is a method where recommendations are made by automatically predicting user preferences by collecting the preferences of similar users. A CF domain consists of users  $U$ , items/services  $I$  and a user-item matrix  $R$  where each entry  $r(u, i)$  where  $r \in \mathbb{R}$ ;  $u \in U$ ;  $i \in I$  represents the rating user 'u' gave to the item 'i' [2]. This is left empty if the user has not rated or used the item 'i'. Here, CF is used to predict missing values that the user might give to an item 'i' to produce recommendations. To produce recommendations, the proposed method uses a memory-based CF approach where we form a neighborhood of similar users for recommendations. Many recommendation systems use the Pearson Correlation Constant (PCC) for similarity computation. Over the past few years, several attempts have been made to improve the CF approach. McLaughlin and Herlocker [7] proposed a weighted variation of the PCC where the values of weights depended on the measure of similarity, computed by the points of commonality between concerned users. A personalized approach to predict user interests through their records was proposed by Hu et al. [12]. Additional contextual information such as location was incorporated in the method proposed by Adomavicius and Tuzhilin [13].

Our method builds on these prior approaches to produce recommendations. Users are sorted according to location and a neighborhood of similar users is generated. The top-k similar neighbor selection algorithm [14] selects the top 'k' number of similar neighbors. For similarity computation, a weighted PCC approach is used where the weights are determined by the history of the common item used. The services undergo a similar process where a neighborhood of items is generated based on location and a weighted approach based on user specific history is used for similarity computation. The diagram for the proposed system is shown below.

## 3. Similar user Computation

As the system is location-aware, users are filtered according to their location. Similar users are often grouped according to their IP addresses [15] however using the IP to determine the closeness of the user is not always accurate [2]. To determine the regional information, we use the AS number that is unique to each region within a country. The hierarchy of user groups is set according to the closeness to the current

user. Our approach uses a variant of the PCC method to predict the user similarity. This section explains the user similarity computation and the neighborhood formation for a given user.

### 3.1. User Similarity Computation and Missing Value Prediction

The traditional PCC approach used to measure similarity between two users -  $u$  and  $v$  (that share common items) can be defined as follows.

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

Where  $u, v$  are the users;  $i \in I_1 \cap I_2$  is the set of common items shared by the users;  $r(u, i), r(v, i)$  are the ratings given by users  $u$  and  $v$  to the item  $i$  and  $\bar{r}(u), \bar{r}(v)$  represent the average ratings of users  $u$  and  $v$ . The larger the value generated by this equation, the more similar the users are.

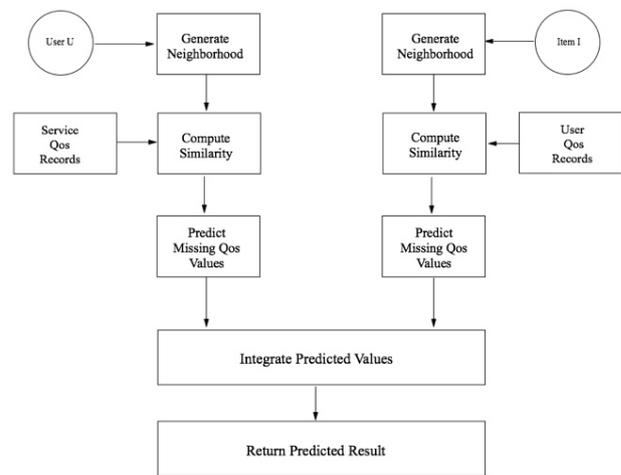


Fig. 1. Overview of the System

Here,  $v \in N(u)$ , the neighborhood for a user  $u$  that is created using the nearest neighbor algorithm. A drawback of this approach is the overestimation of similarity [16] [17] [18]. To avoid this, a threshold parameter can be set to avoid overestimation [7]. In our method, a weighted value is used to generate values based on user history. This can also be normalized in the range of [0, 1]. We do this to take into account the average performance of the item and compare it with the QoS value for the user. The weights are generated as follows:

$$w_u = \frac{[r(u, i) - \bar{r}(i)] - [\min(r(i)) - \bar{r}(i)]}{[\max(r(i)) - \bar{r}(i)] - [\min(r(i)) - \bar{r}(i)]}$$

Here,  $\max(r(i))$  and  $\min(r(i))$  are the maximum and minimum ratings for the item  $i$ . The similarity is measured as follows.

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

### 3.2 User Neighborhood Generation

The next step is to generate the neighborhood of users through which missing values may be predicted. To do this, users are sorted according to their closeness to the current

user. The location-aware neighbor generation algorithm is given below.

**Algorithm 1:** Neighborhood generation

**Input:** User  $u$ ; Number of neighbors  $k$ ; Set of users  $\hat{v}$ ;

**Output:** Neighborhood  $N(u)$

- 1: Get users from same region  $v_r$  and country  $v_c$  where  $v_r, v_c \in \hat{v}$
- 2: While  $N(u)$  not generated
- 3: If number of users in  $v_r > k$
- 4: For each  $v \in$  top  $k$  users in  $v_r$
- 5: Append( $N(u), v$ )
- 6: Else if of users in  $v_r, v_c > k$
- 7: For each  $v \in$  in top  $k$  users in  $v_r, v_c$
- 8: Append( $N(u), v$ )
- 9: Else
- 10: For each  $v \in \hat{v}$
- 11: Append( $N(u), v$ ) until  $k ||$  until  $v =$  null
- 12: End while
- 13: Return  $N(u)$

**4. Similar Item Computation**

Recommendations may also be made depending on the similarity of items. To enhance the item similarity, our method uses a user’s history to determine user preference.

**4.1. Item Similarity Computation and Missing Value Prediction**

The traditional PCC approach to computing the similarity between two items is as follows.

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

Where  $i, j$  are the items;  $u \in U_i \cap U_j$  is the set of common users for both items;  $r(u, i), r(u, j)$  are the ratings given to the items  $i, j$  and  $\bar{r}(i), \bar{r}(j)$  represent the average ratings of items  $i$  and  $j$ . In our method, we introduce a weighted PCC approach that considers the user’s history for each candidate item. This is represented by  $Inf(i, j)$  which is given by the following.

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

Here,  $w_i$  and  $w_j$  are weighted values that represent the similarity between the items and  $j \in N(i)$  the neighborhood for item  $i$ . The value assigned to the weights are computed as follows.

$$w_i = \frac{[r(u,i) - \bar{r}(u)] - [\min(r(u)) - \bar{r}(u)]}{[\max(r(u)) - \bar{r}(u)] - [\min(r(u)) - \bar{r}(u)]}$$

This gives  $w$  a normalized value between  $[0, 1]$ . Here, items that are highly rated by the user are selected to ensure greater accuracy.

**4.2 Item Neighborhood Generation**

Similar to the previous section, the item neighborhood is location aware. Since the QOS of the web services are dependent on the underlying network [11] it is highly likely that the performance of the network increases with lesser distance [2]. The algorithm for the generation of the network neighborhood is similar to the one shown in Section 3.6

where we change the users to services or items.

**5. Missing Value Prediction and Web Service Recommendations**

The missing values in the user-item matrix can be predicted based on user similarity using the equation represented below. A high value indicates that there is a high likelihood that the user would prefer the item. The user similarity based rating is as follows.

$$\hat{r}_u(u, i) = \bar{r}(u) + \frac{\sum_v Sim(u,v)(r(v,i) - \bar{r}(v))}{\sum_v Sim(u,v)}$$

Where  $v \in N(u)$ , the neighborhood for user  $u$ . Similarly, the missing values in the user-item matrix depending on the item or service similarity is computed as follows.

$$\hat{r}_i(u, i) = \bar{r}(u) + \frac{\sum_j Sim(i,j)(r(u,j) - \bar{r}(j))}{\sum_j Sim(i,j)}$$

Where  $j \in N(i)$ , the neighborhood for item  $i$ . The values obtained in both cases can be integrated to give the predicted rating for a list of items. This is done using a parameter  $\alpha$  that is a constant that is used to balance the resultant output.

**5.1. Missing Value Prediction**

The procedure to generate web service recommendations is described in Algorithm 2 given below. It is used to predict the missing QoS values from the user-item matrix. It considers both the user based prediction and the item based predictions and returns an integrated predicted value.

**Algorithm 2:** Generating Web Service Recommendations

**Input:** User  $u$ ; Number of recommendations  $k$ ; User neighborhood  $N(u)$ ; Item neighborhood  $N(i)$ ;

**Output:** Prediction for missing QoS value  $\bar{r}$

- 1: Generate  $\hat{r}_u(u, i)$  using  $N(u)$
- 2: Generate  $\hat{r}_i(u, i)$  using  $N(i)$
- 3: For each predicted value
- 4: If  $\hat{r}_u(u, i) \& \hat{r}_i(u, i)$  not null
- 5:  $\bar{r} = (\alpha * \hat{r}_u(u, i) + (1-\alpha) * \hat{r}_i(u, i))$
- 6: If  $\hat{r}_u(u, i)$  not null &  $\hat{r}_i(u, i) =$  null
- 7:  $\bar{r} = \hat{r}_u(u, i)$
- 8: If  $\hat{r}_u(u, i) =$  null &  $\hat{r}_i(u, i)$  not null
- 9:  $\bar{r} = \hat{r}_i(u, i)$
- 10: Else
- 11:  $\bar{r} =$  null
- 12: Return  $\bar{r}$
- 13: End

In the algorithm,  $\alpha$  is a control variable between 0 to 1 that helps integrate the user and item based predictions. Thus in this way our location-aware method is used to predict the missing QoS values based on user or item history for a given user. An exception to this occurs when  $\hat{r}_u(u, i)$  and  $\hat{r}_i(u, i)$  are both null due to data-sparsity. In cases like this no recommendations will be produced for the user however this occurs very rarely.

**6. Performance Evaluation**

This section consists of a set of comprehensive experiments to determine the effectiveness of our approach. To obtain

accurate results, we have used the real world datasets published by Zheng et al. [3] from their WSDream project. In this section, we evaluate the accuracy of predictions, the impact of control variable  $\alpha$  and the impact of location on the predictions. The experiments were conducted to predict 150 QoS values from the dataset.

### 6.1 Accuracy of Predictions

To compute the accuracy of predictions for our solution, MAE (Mean Absolute Error) is often used. Here the smaller the value, the greater the accuracy of the prediction. The MAE is defined below.

$$MAE = \frac{1}{n} * \sum_{i=1}^n |i - p|$$

To accommodate the large range of values a normalized version called NMAE is used. It is defined below.

$$NMAE = \frac{MAE}{\sum_{i=1}^n r(u,i)/n}$$

To evaluate the prediction accuracy, we compare our method (LAH) to other well known methods methods such as user PCC (UPCC), item PCC (IPCC) and hybrid PCC (HPCC). The results are shown in the table below.

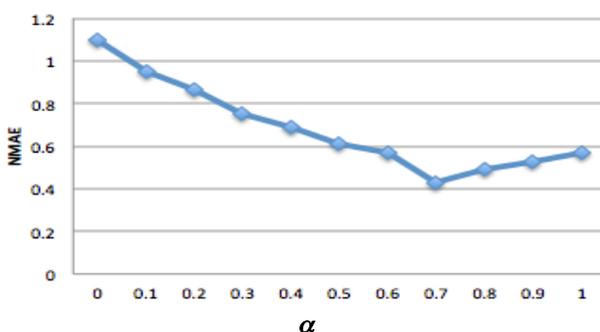
**Table 1.** Accuracy of proposed model

	UPCC	IPCC	HPCC	LAH
<b>NMAE (tp matrix)</b>	1.142	0.903	0.544	0.404
<b>NMAE (rt matrix)</b>	1.025	0.967	0.491	0.425

As observed by the data, LAH has the highest accuracy of prediction.

### 6.2. Impact of $\alpha$

To obtain the best possible value for the variable  $\alpha$  in cases where both  $\hat{r}_u(u, i)$  &  $\hat{r}_i(u, i)$  are not null, we conducted a set of experiments assigning values for  $\alpha$  between the range of 0 to 1 for throughput values. It was determined that the best possible value for  $\alpha$  is 0.7. The results of our experiments are shown in the graph below.



**Fig. 2.** Impact of  $\alpha$

### 6.3. Impact of location based on closeness

Here, we examine how the location influences the accuracy of the predictions. To do so, we conducted a set of experiments where:

1. The neighbors consisted of users/items with same AS values. Since very few of these share the same AS numbers, those that share the country we also included.
2. The neighbors only consisted of elements from the same country
3. The neighbors consisted of elements from different countries.

The results of our experiments are illustrated in the table below.

**Table 2.** Impact of location

	Same ASN	Same Country	Different Countries
<b>NMAE (tp matrix)</b>	0.332	0.409	0.969
<b>NMAE (rt matrix)</b>	0.295	0.470	0.980

Thus it can be inferred from the results of the experiments that the location has an impact on the predicted value such that the closer the user or item, the higher the accuracy of prediction.

## 7. Conclusion

Hence the recommender proposed in this paper can be used for highly accurate QoS value predictions through the use of the user and service locations grouped by the degree of closeness to the user  $u$  and the user/service QoS history that may be used to assign weighted measures to the degree of similarity.

The paper opens new research directions in the field of CF based Web Service Recommendations such as the importance of location in recommender systems and the significance of irregularities in QoS data such as high or low QoS values for certain elements when compared to the average QoS value. This also opens up the possibility of exploring the impact of location and history for functional requirements of the user such as user feedback and product ratings that will help produce even more accurate web service recommendations.

Access article distributed under the terms of the Creative Commons Attribution License



## References

- [1] "Diversifying Web Service Recommendation Results via Exploring Service Usage History" - Guosheng Kang, Mingdong Tang, Jianxun Liu], Xiaoqing(Frank) Liu and Buqing Cao
- [2] "Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation" - Jianxun Liu, Mingdong Tang, Zibin Zheng, Xiaoqing (Frank) Liu, Saixia Lyu
- [3] Zibin Zheng, Yilei Zhang, and Michael R. Lyu, "Investigating QoS of Real-World Web Services", IEEE Transactions on

- Services Computing, vol.7, no.1, pp.32-39, 2014.
- [4] S. S. Yau, Y. Yin, "QoS-based service ranking and selection for service based systems," in Proc. of the International conference on Services Computing, Washington DC, USA, July, 2011, pp. 56 - 63
- [5] G. Kang, J. Liu, M. Tang, X.F. Liu, and K. K. Fletcher, "Web Service Selection for Resolving Conflicting Service Requests," in Proc. 9th International Conference on Web Services, Washington, DC, USA, July, 2011, pp. 387-394.
- [6] Y. Jiang, J. Liu, M. Tang, et al. "An effective web service recommendation based on personalized collaborative filtering". Proceedings of International Conference on Web Services (ICWS2011). IEEE Computer Society, pp. 211-218, 2011
- [7] M.R.McLaughlin, J.L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," SIGIR, 2004.
- [8] D. Goldberg, D. Nichols, B. M. Oki, D. Terry. "Using collaborative filtering to weave an information tapestry". Communications of the ACM, Vol. 35, No. 12, pp. 61-70, 1992.
- [9] J. S. Breese, D. Heckerman, C. Kadie. "Empirical analysis of predictive algorithms for collaborative filtering". Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., pp. 43-52, 1998.
- [10] B. Sarwar, G. Karypis, J. Konstan, J. Reidl. "Item-based collaborative filtering recommendation algorithms". Proceedings of the 10th international conference on World Wide Web. ACM, pp. 285-295, 2001.
- [11] 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.
- [12] R. Hu, W. Dou, J. Liu. "A personalized search approach for web service recommendation". International Journal of Ad Hoc and Ubiquitous Computing, Vol. 13, No. 2, pp. 83-95, 2013.
- [13] G. Adomavicius and A. Tuzhilin. Recommender Systems Handbook, chapter Context-aware Recommender Systems. Springer, 2010.
- [14] Z. Zheng, H. Ma, M.R. Lyu, and I. King. "WSRec: A Collaborative Filtering Based Web Service Recommendation System," in Proc. 7th International Conference on Web Services, Los Angeles, CA, USA, pp. 437- 444, 2009.
- [15] 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.
- [16] Breese, J. S., Heckerman, D., Kadie, C., 1998. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98). Morgan Kaufmann, San Francisco. (pp. 43-52).
- [17] Dahlen, B. J., Konstan, J. A., Herlocker, J. L., Good, N., Borchers, A., Riedl, J., 1998. Jump-starting movie lens: User benefits of starting a collaborative filtering system with "dead data". University of Minnesota TR 98-017.
- [18] Herlocker, J. L., Konstan, J. A., Riedl, J., 2002. An Empirical Analysis of Design Choices in Neighborhood-based Collaborative Filtering Algorithms. Information Retrieval, 5 287-310.