*Journal of AI and Data Mining*
*Vol 8, No 1, 2020, 83-93.*
DOI: 10.22044/JADM.2019.7461.1888

# QoS-based Web Service Recommendation using Popular-dependent Collaborative Filtering

**S. Adeli[1*] and P. Moradi[2]**

[1]*Faculty of Information Technology and Computer Engineering, Urmia University of Technology, Urmia, Iran.*
[2] *Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran.*

Receive15 September 2018; Revised 05 July 2019; Accepted 17 September 2019
* *Corresponding author:somayeh.adeli_it@yahoo.com (S. Adeli).*

**Abstract**
Since most organizations present their services electronically, the number of functionally-equivalent web services is increasing as well as the number of users that employ those web services. Consequently, plenty of information is generated by the users that leads the users to be in trouble with finding their appropriate web services. Therefore, it is required to provide a recommendation method for predicting the quality of web services (QoSs) and recommending web services. Most of the existing collaborative filtering approaches do not operate efficiently in recommending web services due to ignoring some effective factors such as the dependency among users/web services, popularity of users/web services, and location of web services/users. In this paper, a web service recommendation method called Popular-Dependent Collaborative Filtering (PDCF) is proposed. The proposed method handles the QoS differences experienced by the users as well as the dependency among users on a specific web service using the user/web service dependency factor. Additionally, the user/web service popularity factor is considered in PDCF which significantly enhances its effectiveness. We also propose a location-aware method called LPDCF, which considers the location of web services into the recommendation process of PDCF. A set of experiments is conducted to evaluate the performance of PDCF and investigate the impression of the matrix factorization model on the efficiency of PDCF with two real-world datasets. The results obtained indicate that PDCF outperforms other competing methods in most cases.

**Keywords***: Recommender System, Web Service, Collaborative Filtering, QoS-based Recommendation, Quality of Service.*

## 1. Introduction
Nowadays, the number of functionally-equivalent web services is being permanently increased, because most of the organizations present their services on the Internet [1, 2]. As the number of the functionally-equivalent web services are being enhanced, the role of the Quality of Service (QoS) in the web service selection is highlighted [3]. Consequently, it is required to build a web service recommendation system based on QoS to help users in choosing their appropriate web services [4, 5]. A key component of the web service recommendation techniques is the computation of similarity of the users/web services [5, 6]. The values of client-side properties (e.g. response time, invocation failure rate) are dependent on the users' context, for example, users' location, the workload of users' system, and users' network conditions. Thus dependency among users/web services on the client-side's QoS properties is created, that should be considered in computing similarity.
Moreover, some web services are frequently requested by the users, and some of the users often request many web services. These types of users/web services overlap with most of the users/web services that cause errors in predicting

Hence, the importance of this type of web services and users must be decreased in similarity calculating. The available methods do not consider the mentioned problems [6-10]. In this paper, a novel collaborative filtering algorithm, called Popular-Dependent Collaborative Filtering (PDCF), is proposed for web service recommendation. The proposed method enhances the accuracy of predictions using the user/web service popularity and the user/web service dependency factors. Moreover, the proposed method is expended to consider the location of the web services in its recommendation process. This method is so-called the Location-aware PDCF (LPDCF). The experiments conducted with real QoS records indicate that the proposed PDCF method outperforms other competing techniques such as UPCC, IPCC, User-based Normal Recovery Collaborative Filtering (UNRCF), Item-based Normal Recovery Collaborative Filtering (INRCF) and Location-aware Low rank Matrix Factorization (LLMF) [7-9]. The results obtained show that in most cases, the proposed method perform much better that the others.
The rest of this paper is organized as what follows. Section 2 surveys the related works. Section 3 describes the proposed PDCF. Section 4 provides the experimental evaluations to test the precision of PDCF. At last, some conclusions are given in Section 5.

## 2. Related works
The goal of this section is to present the existing QoS-based web service recommendation techniques. Many research works have been done in the field of the web service recommendation in the recent years. Heretofore, various approaches have been extended to recommend web services according to their QoS properties. The Collaborative Filtering (CF) is an effective, well-known, and frequently used recommendation method that has ever been used in many research papers. The CF algorithms are classified into memory-based and model-based methods [4, 6, 7, 8, 9, 10, 11, 12]. The authors in [4] have grouped users and web services into different classes. They have used the Euclidean distance for clustering the users and web services. Then the mean of each cluster has been used to predict the value of the web service that the users have not invoked before [4]. The proposed method in [4] has only considered the mean of similarities in each cluster and does not regard the effect of different values of similarities in the prediction of missing values. Pearson Correlation Coefficient (PCC) has been applied as a similarity measure in [2] as well as the User-based PCC (UPCC) and Item-based PCC (IPCC) in the estimation of missing values. PCC may overrate the similarity values than the actual similarities. Thus they have employed a significant weight to reduce the error in prediction [2]. The aforesaid method is good but it does not consider the QoS difference between different users. Though, the proposed PDCF has solved this problem using the user/web service dependency factor. Researchers in [8] have proposed Normal Recovery Collaborative Filtering (NRCF) for QoS-based web service recommending. They used the Euclidean similarity measure with some changes to improve its performance [8]. In [9], the authors have employed the Location-aware Low rank Matrix Factorization (LLMF) technique to improve the precision of the prediction process in the QoS-based web service recommendation. LLMF utilizes L1-norm low rank matrix factorization with location information of web services to increase the performance of the low-rank matrix factorization method. In [10], the authors have utilized hierarchical tensor decomposition and users/web services clustering based on their location to solve the data sparsity problem [10]. Actually, the methods used in [9, 10] are model-based collaborative filtering algorithms. One of main disadvantages of model-based algorithms is that the models should be reconstructed whenever a new user/web service is inserted in the recommender system [12]. Hence, the memory-based collaborative filtering algorithms have been employed in this paper, although, some researchers have employed the memory-based collaborative filtering algorithms in [4, 7, 9]; however, none of them have considered dependency among the users/web services and popularity of the users/web services. In the present paper, the proposed method employed the dependency factor and popularity factor that leads to a high performance. Moreover, the information location of the web services was incorporated with PDCF owing to the efficacy of the web services' location on the prediction process.

## 3. Popular-dependent collaborative filtering
In this section, the details of the proposed approach are presented based on the memory-based collaborative filtering. In addition to considering the QoS discrepancy among the users/web services,

PDCF uses the user/web service popularity factor for computing the users/web services similarity that increases the accuracy of PDCF. Also the existing dependency among the users/web services is utilized by PDCF that contributes to a high precision in predicting the missing values.

PDCF predicts the missing values in the User-Item matrix (UI matrix) that is a sparse matrix, as follows:

$$UI=\begin{bmatrix} rt_{11} & \cdots & rt_{12} & \cdots & rt_{1n} \\ \vdots & \ddots & & & \vdots \\ rt_{i1} & & \ddots & & rt_{in} \\ \vdots & & & \ddots & \vdots \\ rt_{m1} & \ldots & rt_{ij} & \ldots & rt_{mn} \end{bmatrix}$$

In the UI matrix, the rows represent the users, the columns show the web services, and the cells are the QoS values. Suppose that M is the total number of the users, N is the total number of the web services, and Nqos is the number of the QoS properties (e.g. response time, failure rate). Thus there will be $N_{qos}$ numbers of the UI matrix which each of them includes M×N elements. In this paper, the response time was considered as the intended QoS. Thus $rt_{ij}$ represents the response time of web service i for user j in the above-mentioned UI matrix. Some elements in the UI matrix are null since the users have not invoked for some web services. In PDCF, first the QoS values in the UI matrix are normalized in the range of [0,1], and then the user similarity and the web service similarity are computed. Equations (1) and (2) are employed for the user-based and web service-based normalizations, respectively [13, 14].

$$nurt_{i,j}=\begin{cases} \dfrac{rt_{ij}-rt_{umin}}{rt_{umax}-rt_{umin}}, & \text{if } rt_{umin}\neq rt_{umax} \\ 1, & \text{if } rt_{umin}=rt_{umax} \end{cases} \qquad (1)$$

where $rt_{umin}$ and $rt_{umax}$ denote the minimum and maximum values of response time that user u has observed, respectively.

$$nsrt_{i,j}=\begin{cases} \dfrac{rt_{ij}-rt_{smin}}{rt_{smax}-rt_{smin}}, & if \;\; rt_{smin}\neq rt_{smax} \\ 1, & \text{if } rt_{smin}=rt_{smax} \end{cases} \qquad (2)$$

where $rt_{smin}$ and $rt_{smax}$ denote the minimum and maximum values of response time that are provided by the web service s for all users, respectively.

Different similarity measures can be used for calculating similarity among the users or the web services such as Pearson correlation, Cosine similarity, and Euclidean similarity [15, 16]. However, the PDCF method has been proposed as a new method to compute similarity and predict the missing value with a high performance. In the PDCF method, the user popularity factor and the web service popularity factor are applied as new similarity measures that are described in Section 3.1. Further, PDCF employs the user dependency factor and the web service dependency factor for predicting meticulously, which are presented in details in Section 3.2.

### 3.1. User/web service popularity factor

Some web services are frequently requested by the users, which are so-called popular web services. Suppose that there are two users u1 and u2 that have requested for k popular web services. Again, suppose that there are two other users u3 and u4 that have k co-invoked web services. Then the user similarity between u1 and u2 should be less than the user similarity between u3 and u4. Since the frequent web services are the web services that many users are interested in and when two users have requested for many frequent web services, this does not mean that these two users are similar. Thus, the importance of frequent web services should be reduced in the similarity computation between the users. For this purpose, the web service popularity factor is dedicated to each web service according to (3).

$$WSP_i = log\,\frac{M}{M_i} \qquad (3)$$

where $WSP_i$ denotes the popularity factor of web service i, M determines the total number of the users, and $M_i$ is the number of the users that have requested web service i. In this work, $WSP_i$ was employed with user-based Euclidean similarity to compute the similarity between users according to (4).

$$sim(a,u)=1-\frac{\sqrt{\sum_{i\in S} WSP_i^{-2}\times(r_{u,i}-r_{a,i})^2}}{\sqrt{\sum_{i\in S} WSP_i^{-2}}} \qquad (4)$$

where S denotes the set of the web services that both users a and u have requested, and $r_{u,i}$ and $r_{a,i}$ are the response time of web service i for users u and a, respectively.

Similarly, some of the users often request for a high number of web services that are called popular users. Therefore, when two web services are co-invoked by many popular users, this does not mean that these two web services are very similar. For reducing the impact of popular users in web service similarity computation, (5) is proposed in this paper.

$$sim(i,j)=1-\frac{\sqrt{\sum_{u\in U} UP_u^{-2}\times(r_{u,i}-r_{u,j})^2}}{\sqrt{\sum_{u\in U} UP_u^{-2}}} \tag{5}$$

where $UP_u$ is the user popularity factor of user u that is obtained by (6), U determines a set of the users who have requested for both services i and j, and $r_{u,i}$ and $r_{u,j}$ denote the response time provided by web services i and j for user u.

$$UP_u=\log\frac{N}{N_i} \tag{6}$$

In (6), N is the total number of web services and $N_u$ is the number of web services that have been requested by user u.

### 3.2. User/web service dependency factor

Selecting neighbors is an important task in predicting the missing values that influences on the prediction accuracy. In this work, top-k neighbor selection is used for selecting user/web service neighbors. As regards, different users live in different locations, and the workload of their systems and their network conditions are different from each other. Therefore, various users observe different QoS values on a specific web service, and different web services have different QoS values for a specific user [7, 16]. If the observed QoS values by user v on web service i is very different from the QoS values observed by other users on web service i, the importance of user v should be decreased in a user-based prediction. Most of the prediction methods do not consider the existent dependency among the users on a specific web service or dependency among web services that are used by a particular user. For this purpose, the user dependency factor and web service dependency factor are proposed in this paper. Equation (7) is applied for computing the user dependency factor.

$$UD_v=\frac{\sum_{k=1}^{|I_v|} c^{-(r_{v,k}-\overline{r_k})}}{|I_v|} \tag{7}$$

where $|I_v|$ is the number of web services that user v has used, c is a constant value that controls the disagreement degree, $r_{v,k}$ is the QoS value of web service k for user v, and $\overline{r_k}$ denotes the average of QoS values of web service k that has been seen by different users.

Similarly, if the QoS values observed by user u on web service i is very different from the QoS values observed by that user on other web services, then the importance of web service i should be decreased in the web service-based prediction. In this paper, (8) is employed to calculate the web service dependency factor.

$$WSD_j=\frac{\sum_{l=1}^{|U_j|} c^{-(r_{l,j}-\overline{r_l})}}{|U_j|} \tag{8}$$

In this equation, $|U_j|$ is the number of users that have requested for web service j, $r_{l,j}$ is the QoS value of web service j for user l, and $\overline{r_l}$ determines the average of QoS values of web services that have been requested by user l.

In this paper, a combination of the user-based and web service-based predictions is employed. In addition, the user dependency factor and the web service dependency factor should be taken into account in prediction to access more accuracy. Consequently, (9) is applied to predict the missing values in the proposed PDCF.

$$\begin{aligned} P_{a,i} &= \mu\times(r_{u\min}+(r_{umax}-r_{umin})\times \\ &\frac{\sum_{v\in V}\mathrm{UDv}\times sim(a,v)\times r_{v,i}}{\sqrt{\sum_{v\in V}\mathrm{UDv}\times sim(a,v)}})+ \\ &((1-\mu)\times(r_{s\min}+(r_{s\max}-r_{s\min}))\times \\ &\frac{\sum_{j\in J}\mathrm{WSDj}\times sim(i,j)\times r_{a,j}}{\sqrt{\sum_{j\in J}\mathrm{WSDj}\times sim(i,j)}}) \end{aligned} \tag{9}$$

where μ is a random value in the interval of [0,1] that compromises between the user-based prediction and the web service-based prediction. V and J determine the user neighbors and web service neighbors, respectively. Since the QoS values in the UI matrix have been normalized in the interval of [0,1], the predicted values have been returned to their initial values in (9).

### 3.3. Location-aware PDCF

As regards, the web services are provided on the Internet, QoS of web services (such as response time and throughput) are affected by the sub-structure network. On the other hand, the nodes that are nearby geographically, intend to partake equivalent network sub-structures. Therefore, geographically-close web services represent similar QoS properties. For this reason, the location of web services should be considered in the prediction process [10]. In this paper, web services are grouped based on the continent. In other words, the UI matrix is divided into five sub-matrices entitled UIA, UIE, UINA, UISA, and UIO; each sub-matrix corresponds to one continent. The proposed PDCF is enforced on each sub-matrix to predict the missing values in them. Afterward, the completed sub-matrices are rearranged based on their primary places in the UI matrix.

## 4. Experiments and results

In this section, some experiments have been accomplished in order to represent the efficiency of the proposed method, called the PDCF. Experiments were done on two datasets provided by Zheng et al.; both of them are described as follow:

The first one includes 1.5 million invocation records on 100 web services by 150 users [17]. Thus there will be a UI matrix with 150×100 elements, in which each element represents the response time. The second one consists of 1,974,675 elements that represent the response time experienced by 339 users on 5825 web services [18]. Therefore, there are UI matrix equivalents to 339 rows and 5825 columns, in the second dataset. In order to create a situation similar to the real world, some numbers of elements in the UI matrix were eliminated randomly. Thus, the density of the UI matrix can be determined as follows:

$$density = \frac{G}{M \times N} \tag{10}$$

where G denotes the number of elements that have remained in the UI matrix.

The effectiveness of different parameters such as μ, neighborhood size, density, and c was investigated using the first dataset, and the results obtained illustrated in Sections 4.2, 4.3, 4.4, and 4.5. PDCF was compared with PCC and NRCF using the first dataset in Section 4.6. As regards, the first dataset is smaller than the second, and if the UI matrix is too small, the effectiveness of location information on the efficiency of PDCF becomes contrary; the second dataset was applied to evaluate LPDCF and the results obtained were demonstrated in Section 4.7.

### 4.1. Evaluation metric

The mean absolute error (MAE) and root mean squared error (RMSE) are employed as metrics for appraising the performance of the proposed PDCF in this paper. MAE and RMSE determine how much a predicted value is far from the real value. MAE and RMSE are defined according to (۱۱) and (12).

$$MAE = \frac{\sum_{u,s} \left| r_{u,s} - p_{u,s} \right|}{N_p} \tag{11}$$

$$RMSE = \sqrt{\frac{\sum_{u,s} \left( r_{u,s} - p_{u,s} \right)^2}{N_P}} \tag{12}$$

where $r_{u,s}$ is the real QoS value of web service s for user u, $p_{u,s}$ is the predicted QoS value for user u on web service s, and $N_p$ denotes the total number of the predicted values.

### 4.2. Impact of μ

The parameter μ is a balancing parameter that compromises between the user-based prediction and the web service-based prediction. Different datasets have different attributes that influence the precision of prediction. For this reason, a balancing parameter is required to adopt the predictions with different datasets. If μ=0, prediction will only be based on the web service similarity. μ=1 means that prediction will only be based on the user similarity. However, if 0< μ <1, the prediction will be based on a combination of the user similarity and the web service similarity. In this work, for investigating the impact of parameter μ on the proposed PDCF, the value of parameter μ was changed from 0 to 1 with increment steps=0.1. We set | J |=10, |V|=10, training users=100, GN=10, C=1.9, and density=0.14. Figure 1 represents the impact of μ on the obtained prediction accuracy using PDCF. The results obtained indicate that MAE is increased by enhancing μ, moderately. Consequently, the minimum value of MAE is provided by μ=0.

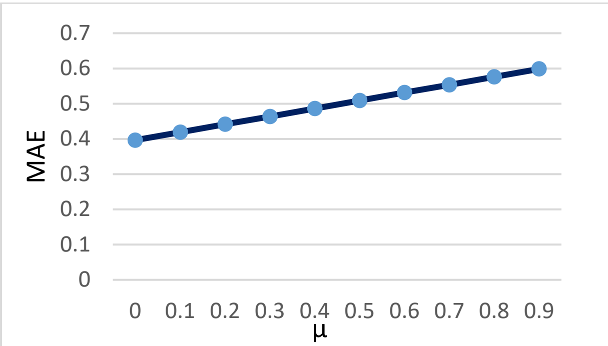



**Figure 1. Impact of μ.**

### 4.3. Impact of neighborhood size

The number of users or the web services applied to predict the missing values in the UI matrix is specified by the neighborhood size. Indeed, the neighborhood size equals the user neighbors and the web service neighbors. Therefore, the neighborhood size affects the prediction accuracy of the proposed PDCF. In this experiment, for investigating the impact of the neighborhood size, all parameters are similar to Section 3.2, and the number of neighbors is varied from 10 to 50 with a step of 10. As observed in figure 2, the maximum value of MAE is provided using the neighborhood size=10, and the minimum values are represented in the neighborhood size=40 and the neighbor size=50.

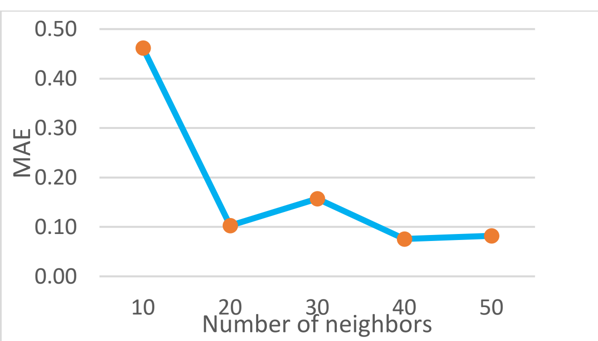



**Figure 2. Impact of the neighborhood size.**

### 4.4. Impact of density

Density determines the volume of the existing information in the UI matrix to predict the lost values. Thus the performance of PDCF is affected by density. In this section, an experiment is accomplished to study the impact of density on MAE using parameters similar to Section 3.2. Then values of density are modified between 0.04 and 0.2. Figure 3 indicates that MAE is reduced by increasing density.

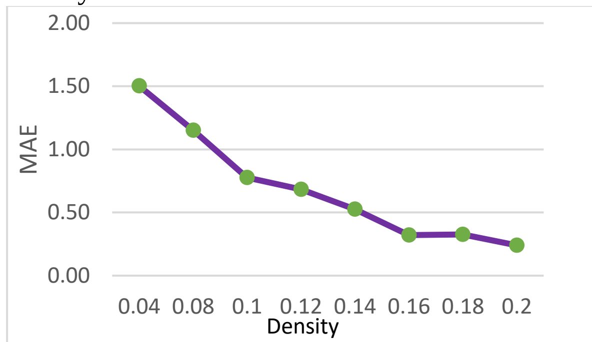



**Figure 3. Impact of density.**

More information is achieved about the user/web service popularity factor and user/web service dependency factor by enhancing density. Thus MAE reduces with an increasing value of density.

### 4.5. Sensitivity analysis over c

In this section, some experiments are performed to find the best value of c, which controls the disagreement degree in the user/web service dependency factor. The best value of c is the value that decreases MAE. For this purpose, the value of parameter c is varied from 0.1 to 3 by steps=0.1. Figure 4 illustrates that the best value of c is 0.1 using density=0.1, μ=0.1, and neighborhood size=10.

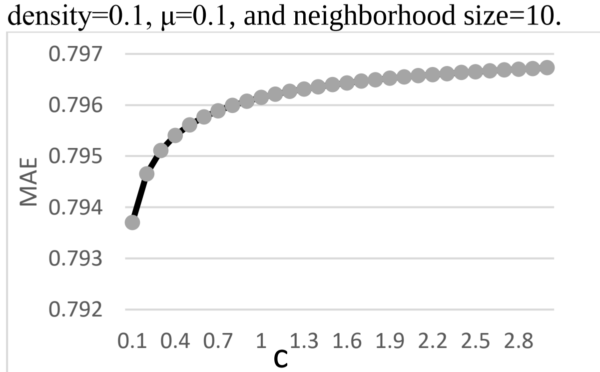



**Figure 4. Relationship between c and MAE.**

Additionally, two experiments were performed to find the best value of c according to different values of density and neighborhood size. In both experiments, the value of μ was considered 0.1. In the first test whose result is represented in figure 5, density is modified from 0.04 to 3, and the best value of c is selected in each density. In horizontal axes, top numbers and bottom numbers indicate c and density, respectively. Figure 5 shows that the minimum value of MAE is provided by c=3 in density=0.18.

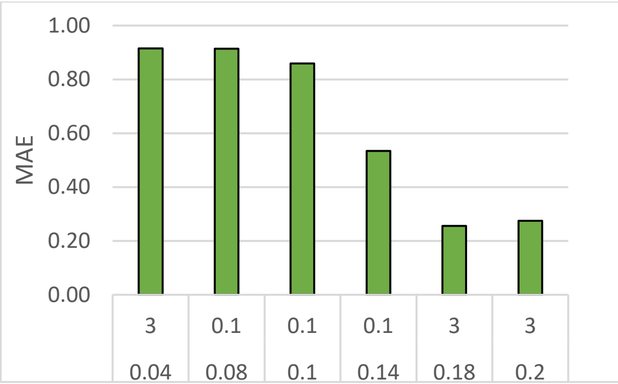



**Figure 5. Best value of c based on different values of density.**

The best value of c according to neighborhood size was studied in the second test. Figure 6 indicates that when the neighborhood size is 40, the best value for c is 1.5. Indeed, the least value for MAE was acquired by c=1.5.

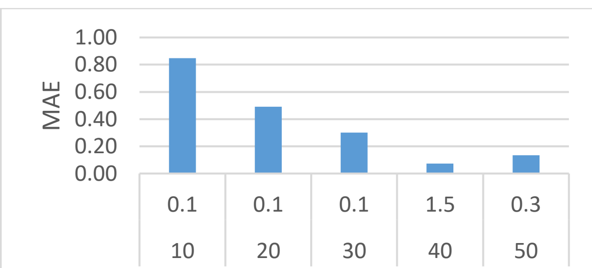



**Figure 6. Best value of c according to various values of neighborhood size.**

### 4.6. Evaluating performance of the PDCF using dataset 1

In this section, the efficiency of the proposed PDCF method is evaluated in the web service recommendation. Performance of PDCF is compared with PCC and NRCF, using some experiments.

In the first experiment, we set density=0.14, neighborhood size=10, and c=0.1, and modify values of μ from 0 to 1 with a step value of 0.1. Figure 7 indicates that PDCF is more accurate than PCC and NRCF. Compared to the PCC and NRCF techniques, PDCF improves the accuracy under various μ values. The minimum MAE values provided by PDCF, PCC, and NRCF are 0.5136, 0.9502, and 0.8450, respectively. Consequently, it can be said that the PDCF method significantly outperforms PCC and NRCF.

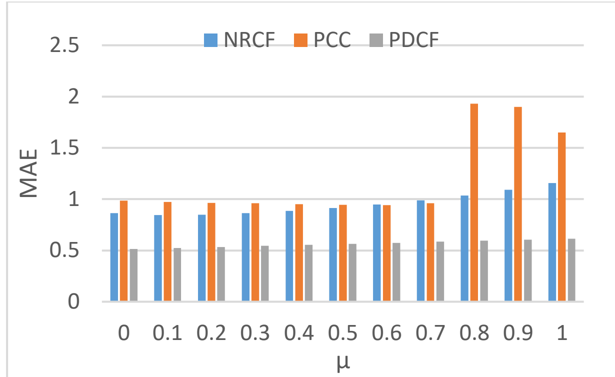
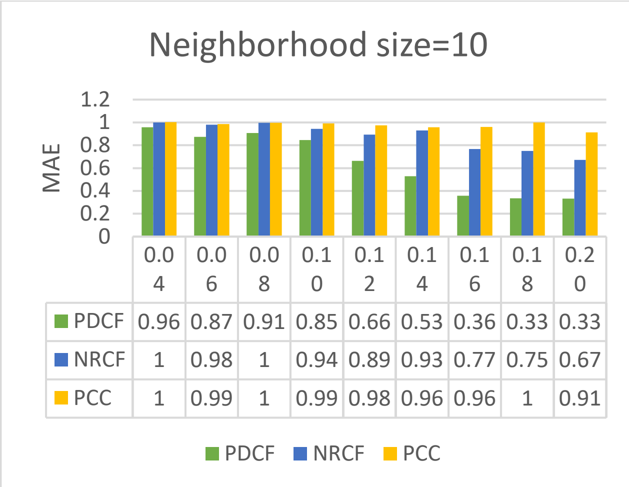



**Figure 7. Performance comparison between PDCF, NRCF, and PCC under different μ.**

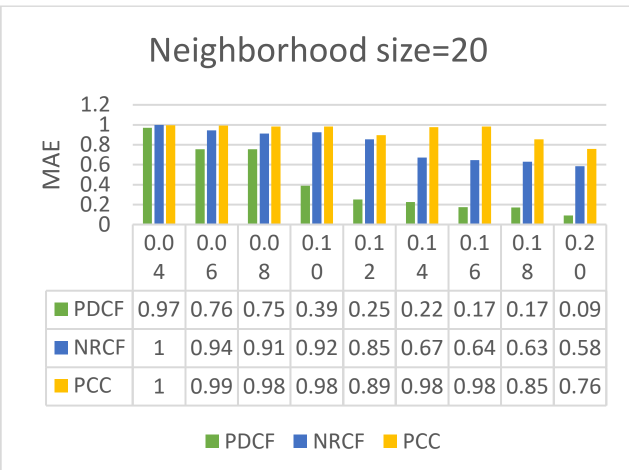
Performance of PDCF, PCC, and NRCF was compared with each other under different densities and neighborhood sizes in the second experiment. This experiment includes five subtests with the neighborhood size=10, 20, 30, 40, and 50 whose values of density are varied between 0.04 and 0.2 in each neighborhood size.

Figure 8 demonstrates the results obtained from evaluating the accuracy of PDCF, NRCF, and PCC measured by MAE. What is observed in all parts of figure 8 is that PDCF operates more effective than NRCF and PCC.

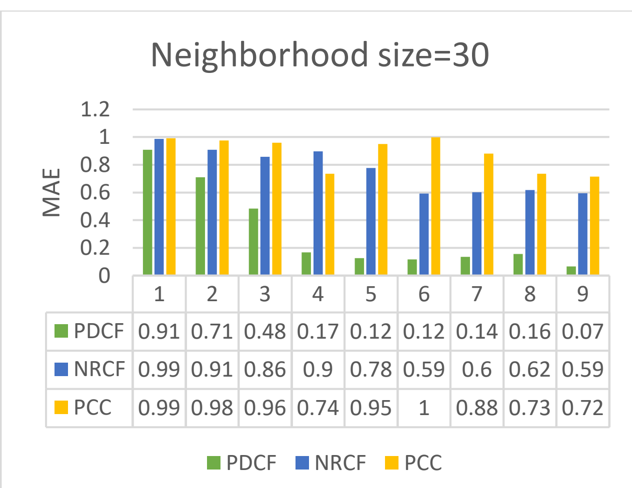
As apperceived in figure 8, the accuracy of PDCF improves with increasing density and neighborhood size, insofar as MAE significantly declines to 0.01 with the neighborhood size=50 and density=0.2. When the UI matrix gets denser, popular users and popular web services are recognized with more accuracy, and the importance of popular users and popular web services is reduced in calculating the missing values. As a result, the precision of prediction is enhanced and MAE is decreased. PDCF outperforms other approaches even when the UI matrix is sparse.



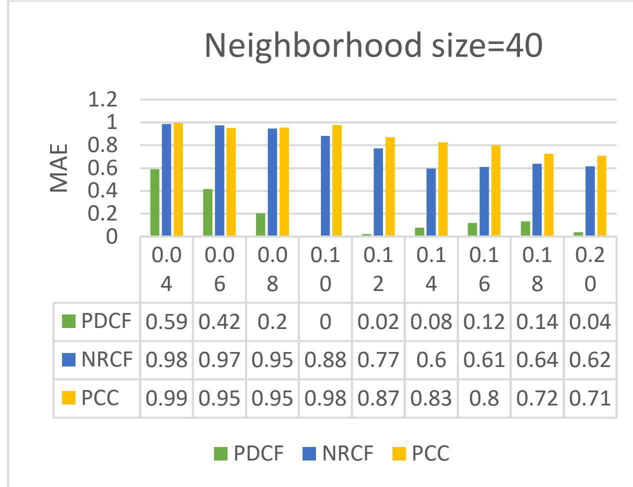

| | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.20 |
|---|---|---|---|---|---|---|---|---|---|
| PDCF | 0.96 | 0.87 | 0.91 | 0.85 | 0.66 | 0.53 | 0.36 | 0.33 | 0.33 |
| NRCF | 1 | 0.98 | 1 | 0.94 | 0.89 | 0.93 | 0.77 | 0.75 | 0.67 |
| PCC | 1 | 0.99 | 1 | 0.99 | 0.98 | 0.96 | 0.96 | 1 | 0.91 |



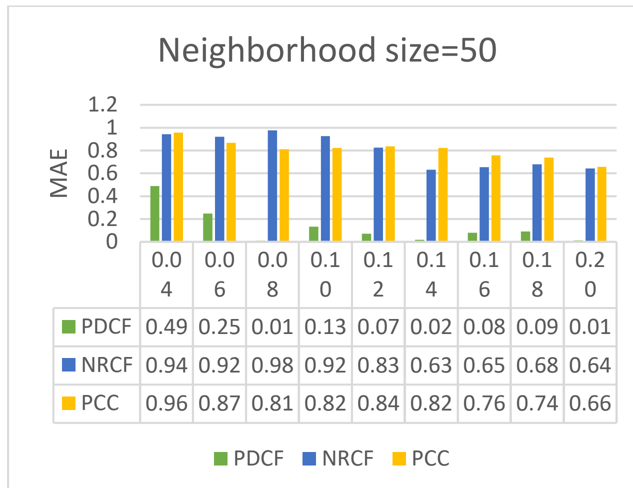

| | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.20 |
|---|---|---|---|---|---|---|---|---|---|
| PDCF | 0.97 | 0.76 | 0.75 | 0.39 | 0.25 | 0.22 | 0.17 | 0.17 | 0.09 |
| NRCF | 1 | 0.94 | 0.91 | 0.92 | 0.85 | 0.67 | 0.64 | 0.63 | 0.58 |
| PCC | 1 | 0.99 | 0.98 | 0.98 | 0.89 | 0.98 | 0.98 | 0.85 | 0.76 |




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| PDCF | 0.91 | 0.71 | 0.48 | 0.17 | 0.12 | 0.12 | 0.14 | 0.16 | 0.07 |
| NRCF | 0.99 | 0.91 | 0.86 | 0.9 | 0.78 | 0.59 | 0.6 | 0.62 | 0.59 |
| PCC | 0.99 | 0.98 | 0.96 | 0.74 | 0.95 | 1 | 0.88 | 0.73 | 0.72 |




| | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.20 |
|---|---|---|---|---|---|---|---|---|---|
| PDCF | 0.59 | 0.42 | 0.2 | 0 | 0.02 | 0.08 | 0.12 | 0.14 | 0.04 |
| NRCF | 0.98 | 0.97 | 0.95 | 0.88 | 0.77 | 0.6 | 0.61 | 0.64 | 0.62 |
| PCC | 0.99 | 0.95 | 0.95 | 0.98 | 0.87 | 0.83 | 0.8 | 0.72 | 0.71 |




| | 0.04 | 0.06 | 0.08 | 0.10 | 0.12 | 0.14 | 0.16 | 0.18 | 0.20 |
|---|---|---|---|---|---|---|---|---|---|
| PDCF | 0.49 | 0.25 | 0.01 | 0.13 | 0.07 | 0.02 | 0.08 | 0.09 | 0.01 |
| NRCF | 0.94 | 0.92 | 0.98 | 0.92 | 0.83 | 0.63 | 0.65 | 0.68 | 0.64 |
| PCC | 0.96 | 0.87 | 0.81 | 0.82 | 0.84 | 0.82 | 0.76 | 0.74 | 0.66 |

**Figure 8. Performance comparison between the PDCF, NRCF and PCC under various densities and neighborhood size.**

### 4.7. Evaluating performance of the LPDCF using dataset 2

In this section, several experiments were performed to evaluate the effectiveness of the location information and matrix factorization model on the proposed PDCF. In addition, the performance of PDCF and LPDCF was investigated compared with the competing methods such as LWSPCC, LUPCC, LWSNRCF, LUNRCF, and LLMF using MAE and RMSE [4, 8, 9].

All the mentioned methods were combined with location information as, explained in Section 3.3. For this reason, they have been entitled LWSPCC, LUPCC, LWSNRCF, and LUNRCF. Additionally, we incorporated LLMF with LPDCF, named LLMF+LPDCF, in order to investigate the impact of matrix factorization on improving LPDCF. The values predicted by LLMF contribute to complete the UI matrix and decrease data sparsity. The completed UI matrix by the LLMF algorithm was applied as input for LPDCF. The results of evaluations are demonstrated in table I. WS-LPDCF, LWSNRCF, LWSPCC, and LLMF+WS-LPDCF represent that item-based collaborative filtering is performed in their prediction process ($\mu$=0) and the user-based collaborative filtering algorithm is used in U-LPDCF, LUNRCF, LUPCC, and, LLMF+U-LPDCF ($\mu$=1).

Besides, other parameters (such as c and neighborhood size) are used in the mentioned approaches whose values are adjusted according to c=3 and neighborhood size=20 in this experiment. As observed in table I the WS-PDCF, U-PDCF, WS-LPDCF, U-LPDCF, LLMF+WS-LPDCF, and LLMF+U-LPDCF generally outperform other competing methods.

WS-LPDCF and U-PDCF weigh the users/web services based on the dependency factor and popularity factor, and they consider the location of the web services all of which are effective factors on the performance of the prediction methods. Consequently, WS-LPDCF and U-PDCF outnumber other competing memory-based collaborative filtering algorithms.

LLMF is a state-of-the-art model-based collaborative filtering method that has been proposed in [10]. LLMF utilizes L1-norm with low rank matrix factorization (LMF) to improve the precision of LMF. However, WS-PDCF/U-PDCF and WS-LPDCF/U-LPDCF operate better than LLMF as observed in table I. Furthermore, the results obtained illustrate a combination of WS-LPDCF/U-LPDCF and, LLMF causes a higher precision than other competing methods. That is because LLMF reduces data sparsity of the UI matrix.

Howsoever, WS-PDCF/U-PDCF operates better than WS-LPDCF/U-LPDCF and LLMF + WS-LPDCF/ LLMF + U-LPDCF in some conditions. For example, when the density is 30%, MAE of WS-PDCF is 0.459, which is lower than MAE of WS-LPDCF and LLMF + WS-LPDCF. In WS-LPDCF and LLMF+WS-LPDCF, the web services are divided into some groups based on their location; the dependency factor and popularity factor are computed in each group, and similarity among web services is calculated in each group. Under location-based web service grouping, the popularity factor and dependency factor fail to utilize overall information about web services. Hence, when density raises, WS-PDCF performs more accurate than WS-LPDCF and LLMF+WS-LPDCF.

## 5. Conclusion

In this paper, a novel method called popular-dependent collaborative filtering (PDCF) was proposed for QoS-based web service recommendation. The proposed PDCF method not only considers the QoS differences experienced by the users but also applies the user/web service popularity factor for decreasing impact of the popular users/web services in similarity computing that leads to an accuracy more than other competitive methods. Besides, different from other methods, the existent dependency among users/web services is taken into account using the user/ web service dependency factor in PDCF. The conducted experiments with two real-world datasets indicated effectiveness of PDCF compared to PCC, NRCF, and LLMF. In addition, the proposed PDCF method was combined with location information of web services and low rank matrix factorization, entitled LLMF+LPDCF. The results obtained demonstrated that this combination operated extremely efficiently.

**Table I. MAE and RMSE comparison between the proposed PDCF and competing methods based on different densities.**

| Method | | Density | | | | | |
|---|---|---|---|---|---|---|---|
| | | 5% | 10% | 15% | 20% | 25% | 30% |
| WS-LPDCF | MAE | 0.9474 | 0.7600 | 0.6961 | 0.5120 | 0.4995 | 0.4993 |
| | RMSE | 2.2056 | 1.9088 | 1.8077 | 1.4994 | 1.4699 | 1.4438 |
| U-LPDCF | MAE | 0.7722 | 0.6247 | 0.591 | 0.5614 | 0.5300 | 0.5162 |
| | RMSE | 1.9706 | 1.7176 | 1.5659 | 1.51612 | 1.4202 | 1.3930 |
| LLMF+WS-LPDCF | MAE | 0.8874 | 0.7224 | 0.5412 | 0.5121 | 0.4759 | 0.4675 |
| | RMSE | 2.1146 | 1.8773 | 1.5971 | 1.4799 | 1.4238 | 1.3861 |
| LLMF+U-LPDCF | MAE | 0.7669 | 0.6241 | 0.5726 | 0.5280 | 0.5090 | 0.4955 |
| | RMSE | 1.9622 | 1.7123 | 1.6176 | 1.4713 | 1.4186 | 1.3712 |
| LWSPCC | MAE | 1.528 | 1.3077 | 1.1491 | 1.0355 | 1.0106 | 0.9895 |
| | RMSE | 2.541 | 2.5053 | 2.5017 | 2.3229 | 2.1275 | 2.0660 |
| LUPCC | MAE | 1.0618 | 0.8822 | 0.7720 | 0.7005 | 0.6311 | 0.6037 |
| | RMSE | 2.8040 | 2.734 | 2.6870 | 2.3892 | 1.4582 | 1.4209 |
| LWSNRCF | MAE | 0.9516 | 0.8607 | 0.7505 | 0.5965 | 0.5044 | 0.4988 |
| | RMSE | 2.2120 | 1.9191 | 1.8176 | 1.5006 | 1.4727 | 1.4573 |
| LUNRCF | MAE | 0.7725 | 0.6347 | 0.6089 | 0.5759 | 0.638 | 0.5561 |
| | RMSE | 1.9707 | 1.7216 | 1.6651 | 1.6456 | 1.645 | 1.409 |
| LLMF | MAE | 0.903 | 0.9087 | 0.908 | 0.931 | 0.913 | 0.910 |
| | RMSE | 2.1372 | 2.1344 | 2.1177 | 2.1139 | 2.1082 | 2.108 |
| WS-PDCF | MAE | 1.0397 | 0.9041 | 0.5836 | 0.501 | 0.4785 | 0.459 |
| | RMSE | 2.3159 | 2.0693 | 1.6481 | 1.5113 | 1.4421 | 1.376 |
| U-PDCF | MAE | 0.7223 | 0.6117 | 0.6098 | 0.5665 | 0.5058 | 0.4933 |
| | RMSE | 1.894 | 1.6746 | 1.6434 | 1.5169 | 1.4166 | 1.3863 |

## References

[1] Chen, X., Zheng, Z., & R. Lyu, M. (2014). QoS AwareWeb Service Recommendation via Collaborative Filtering, New York: Springer Science+Business Media.

[2] Ge, J., Chen, Z., Peng, J., Li, T., & Zhang, L. (2010). Web Service Recommendation Based on QoS Prediction Method. 9th IEEE Int. Conf. on Cognitive Informatics, pp. 109-112

[3] Hamidzadeh, M. M. (2019). Ensemble-based Top-k Recommender System Considering Incomplete Data. Journal of AI and Data Mining, vol. 7, pp. 393-402.

[4] Jiang, Y., Liu, J., Tang, M., & Liu, X. (2011). An Effective Web Service Recommendation Method based on Personalized Collaborative filtering. IEEE International Conference on Web Services(ICWS11). IEEE computer society,pp. 211-218.

[5] Lakhal, R. B., & Chainbi, W. (2012). A Multi Criteria Approach for Web Service Discovery. The 9th International Conference on Mobile Web Information Systems (MobiWIS), Elsevier PCS, pp. 609-616.

[6] Lin, C.-F., Sheu, R.-K., Chang, Y.-S., & Yuan, S.-M. (2011). A relaxable service selection algorithm for QoS-based web service composition. Information and Software Technology, pp.1370-1381.

[7] Majlesi Esfahani, P., Varaee, T., & Habibi, J. (2012). Application of Social Harmony Search Algorithm on Composite Web Service Selection based on Quality Attributes. 2012 Sixth International Conference on Genetic and Evolutionary Computing. IEEE Computer Society, pp. 526-529.

[8] Nikita R.Gurjar, S. V. (2015). Personalized QoS-Aware Web Service Recommendation via Exploiting Location and Collaborative Filtering. International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, no. 1, pp. 695-698.

[9] Pan, Y., Wu, D., & Olson, D. (2017). Online to Offline (O2O) Service Recommendation Method Based on Multi-dimensional Similarity Measurement. Decision Support Systems, vol. 8, no. 3, pp. 1-8.

[10] Rajeswari, M., Sambasivam, G., Balaji, N., Saleem Basha, M., Vengattaraman, T., & Dhavachelvan, P. (2013). Appraisal and analysis on various web service. Journal of King Saud University – Computer and Information Sciences, vol. 26.

[11] Rong, W., Peng, B., Ouyang, Y., Liu, K., & Xiong, Z. (2015). Collaborative personal profiling for web service ranking and recommendation. Inf Syst Front, vol. 17, pp. 1265–1282.

[12] Sachan, A., & Richariya, V. (2013). A Survey on Recommender Systems based on Collaborative Filtering Technique. International Journal of Innovations in Engineering and Technology (IJIET), vol. 2, pp. 8-14.

[13] Su, K., Xiao, B., Liu, B., Zhang, H., & Zhang, Z. (2017). TAP: A personalized trust-aware QoS prediction approach for web service recommendation. Knowledge-Based Systems, vol. 115 pp. 55-65.

[14] Sun, H., Zheng, Z., Chen, J., & R. Lyu, M. (2013). Personalized Web Service Recommendation via Normal Recovery Collaborative Filtering. IEEE TRANSACTIONS ON SERVICES COMPUTING, pp. 573-579.

[15] TIAN CHENG, J. W. (2019). Personalized Web Service Recommendation Based on QoS Prediction and Hierarchical Tensor Decomposition. IEEEAccess, vol. 7, pp. 62221-62230.

[16] Xiaoke Zhu, X.-Y. J. (2018). Similarity-maintaining Privacy Preservation and Location-aware Low-rank Matrix Factorization for QoS Prediction based Web Service Recommendation. IEEE TRANSCATIONS ON SERVICES COMPUTING.

[17] Zheng, Z., Ma, H., R. Lyu, M., & King, I. (2011). QoS-Aware Web Service Recommendation by Collaborative Filtering. IEEE TRANSACTIONS ON SERVICES COMPUTING, vol. 4, pp. 140-152.

[18] Zibin Zheng, Y. Z. (2010). Distributed QoS Evaluation for Real-World Web Services. 8th International Conference on Web Services, pp. 83-90.

نشریه هوش مصنوعی و داده کاوی

# ارائه‌ی روش پیشنهادگر مبتنی بر ویژگی‌های کیفیت سرویس با استفاده از روش فیلترینگ مشارکتی عمومیت-وابستگی

**سمیه عادلی [۱،*] و پرهام مرادی [۲]**

[۱] دانشکده‌ی مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی ارومیه، ارومیه، ایران.

[۲] دانشکده‌ی مهندسی کامپیوتر، دانشگاه کردستان، سنندج، ایران.

ارسال ۲۰۱۸/۰۹/۱۵؛ بازنگری ۲۰۱۹/۰۷/۰۵؛ پذیرش ۲۰۱۹/۰۹/۱۷

**چکیده:**

ازآنجایی‌که امروزه اکثر سازمان‌ها خدماتشان را تحت وب ارائه می‌دهند و تعداد سرویس‌های وب با عملکرد یکسان روزبه‌روز در حال افزایش است؛ کاربران برای پیدا کردن سرویس‌های وب مناسب خود دچار مشکل شوند. بنابراین، ارائه‌ی یک روش مناسب برای پیش‌بینی ویژگی‌های کیفی سرویس‌های وب و پیشنهاد مناسب‌ترین سرویس وب به کاربران امری ضروری است. به دلیل اینکه بیشتر روش‌های مبتنی بر فیلترینگ مشارکتی، فاکتورهای تأثیرگذار در سیستم‌های پیشنهادگر مانند وابستگی بین کاربران/ سرویس‌های وب، عمومیت کاربران/ سرویس‌های وب و مکان کاربران/ سرویس‌های وب را نادیده می‌گیرند، بیشتر روش‌های مبتنی بر فیلترینگ مشارکتی از کارایی لازم برای پیشنهاد سرویس‌های وب مناسب برخوردار نیستند. در این مقاله، یک روش جدید به نام روش فیلترینگ مشارکتی عمومیت-وابستگی ارائه می‌شود، که علاوه بر اینکه تفاوت ویژگی‌های کیفیت سرویس تجربه‌شده توسط کاربران را در نظر می‌گیرد، وابستگی بین کاربران/سرویس‌های وب را روی یک سرویس وب/ کاربر خاص با استفاده از فاکتور وابستگی کاربر/سرویس وب نیز در نظر می‌گیرد. علاوه براین، فاکتور عمومیت کاربر/سرویس وب در روش فیلترینگ مشارکتی عمومیت- وابستگی در نظر گرفته می‌شود که به‌طور چشمگیری کارایی آن را افزایش می‌دهد. همچنین روشی به نام روش فیلترینگ مشارکتی عمومیت- وابستگی مکان آگاه در این مقاله ارائه‌شده است که علاوه بر فاکتورهای مذکور مکان سرویس‌های وب را نیز در فرایند پیشنهاد در نظر می‌گیرد. به‌منظور ارزیابی کارایی روش‌های پیشنهادشده و بررسی تأثیر تکنیک مدل‌سازی تجزیه‌ی ماتریسی روی روش پیشنهادشده، آزمایش‌هایی با استفاده از دو مجموعه داده‌ی حقیقی انجام می‌شود. نتایج آزمایش‌ها نشان می‌دهد که روش فیلترینگ همکاری عمومیت- وابستگی بهتر از سایر روش‌ها عمل می‌کند.

**کلمات کلیدی**: سیستم پیشنهادگر، سرویس وب، روش فیلترینگ مشارکتی، پیشنهاد مبتنی بر ویژگی‌های کیفیت سرویس، کیفیت سرویس.