IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2020, accepted December 6, 2020, date of publication December 10, 2020,
date of current version December 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043773

A Two-Tier Service Filtering Model for Web Service QoS Prediction

MINGYU LI, QIN LU, AND MINGGE ZHANG
School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
Corresponding author: Qin Lu (54903172@qq.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700604.

ABSTRACT Service recommendation technology is the key to realize the personalization of intelligent services. The recommended services need to meet functional requirements as well as non-functional requirements. Therefore, QoS-based service recommendation came into being. To perform intelligent service recommendations, matching users with convenient services based on QoS becomes an inevitable task. However, most of the service recommendation models are based on user interaction records to predict and recommend, ignoring the service-user correlation and unstable QoS values. In this article, we propose a new service recommendation model. We have performed two-tier filtering calculation on a large number of Web Services, filtering the contextual information of users and services and the instability of services. In the first filtering layer, we take the instability of QoS as an indicator to eliminate invalid services, which significantly reduces the service scale and eliminates the interference of invalid services on the recommendation to a certain extent. Further, we process the contextual information of both users and services in the second filtering layer. Considering the impact of the correlation between the service and the user, we use the geographic location information of the user and the service, and solve the combined features generated by the similarity between the user and the service to filter. Considering the sparsity of the service recommendation environment and the influence of noise generated by useless features, we use a model of factorization machine combined with the attention mechanism for computational processing. It effectively distinguishes the interactive importance of different features. We have conducted many experiments on real dataset, and the results show that our model is better than most baseline model in terms of recommendation performance.

INDEX TERMS Service recommendation, QoS, invalid service, contextual information, service filter.

I. INTRODUCTION

“Service” exists in all aspects of our lives. Web services are becoming more and more popular among users, such as booking hotels and buying movie tickets. This makes the number of Web services continue to increase. Web service is an application module that is self-adaptive, self-describing, modular and has good interoperability capabilities. Based on the above advantages, in recent years, more and more software developed to support Web services, and its industrial applications are also increasing. The results of abundant production activities have gradually begun to be shown to us in the form of services, and servicing has become an inevitable trend of industrial development. In this environment, it is difficult for users to choose the services they need due to the large number of services available.Therefore, how to quickly and accurately recommend the most close to the target and good service quality for users, that is, to meet the needs of customers in all aspects of the service, is our ultimate goal.Therefore, service recommendation technology has begun to gain attention [1]–[4].

Service recommendation technology is the key to realize the personalization of intelligent services. In simple terms, the service recommendation task is to recommend a service that can meet the user’s functional requirements. Simultaneously, the service must meet non-functional requirements (Quality of Service, referred to as QoS), such as time, price, reliability, etc., especially with the same or similar services. In the actual service invocation scenario, it is vital to find the optimal choice from the service based on the user’s

The associate editor coordinating the review of this manuscript and approving it for publication was Zhangbing Zhou.

QoS attributes. However, large-scale web service evaluation is a complicated task. The number of historically invoked services by users is usually minimal, resulting in a severe lack of historical QoS values. This situation is an urgent problem to be solved in the service computing environment. Therefore, to perform an intelligent service recommendation, it is an inevitable task to perform high-precision service prediction based on the service QoS value.

In such a complex environment, Researchers continue to study the Web service recommendation method based on QoS, and QoS becomes an important reference index for candidate services. On the one hand, we must first focus on combining contextual information for the recommendation based on alleviating data sparsity. On the other hand, due to the vast service scale, the recommendation efficiency has declined. Therefore, improving service recommendation efficiency is also an important issue that needs to be resolved urgently, and the treatment of services can be used as a preliminary preparation to solve the problems of reliability and recommendation efficiency of later services. In this paper, we use a two-tier service filtering model for efficient personalized service recommendation. To improve the quality of service recommendations, we have performed two-tier filtering. First, we filter unstable and invalid services by judging the stability of the service's QoS value. This operation reduces the service scale to a certain extent, eliminates the interference caused by the appearance of invalid services to service recommendation, and improves recommendation efficiency. To alleviate the impact of context on service recommendation and the relevance between users and services, we consider users and services' geographic location. We achieve filtering by solving similarities between the characteristics of services and users. Furthermore, considering the service recommendation environment's sparsity and noise impact, we use a factorization machine (FM) to solve it. Finally, to further enhance the recommendation effect of the model, we added an attention mechanism based on FM for computational processing, which effectively distinguishes the importance of different interactive features.

The main contributions of our work are summarized as follows:

- We proposed a service filtering strategy to filter invalid services generated during user interaction. We considered the QoS of Web services' negative impact on the prediction results, as well as the reliability and efficiency issues. So we selected invalid services from the mass services and eliminated them to improve the candidate services' purity further.
- By using the geographic location information of users and services, considering the connection between services and users and the importance of different combined features, we used a factorization machine and attention mechanism to predict various combined features and solved the impact of the correlation between services and users.
- Based on 1 and 2, we proposed a two-tier filtered service recommendation model. This model combines the motivation of filtering invalid services and location information. Through two filterings, it not only solves the problem of invalid service interference in the early stage, but also solves the influence of context information and the relationship between service and user, and improves service recommendation efficiency.
- We used the real Web service QoS dataset to carry on the experimental evaluation, the result verifies our proposed model has the good prediction ability.

II. RELATED WORK

Research on QoS prediction [5]–[8] has been widely concerned by academia and industry in recent years. Among all the prediction methods based on QoS, Collaborative Filtering (CF) is the most popular recommendation technology, commonly used in various recommendation systems, mainly because of its simplicity and effectiveness. For example, Zheng *et al.* [9] introduced a similarity calculation method to improve similar neighbors' similarity calculation and the collaborative filtering method. Yao Ming *et al.* [10] integrated content features into collaborative filtering, and use semantic analysis technology to understand user invocation preferences by analyzing Web service description content. Ren *et al.* [11] proposed a CF method based on Support Vector Machine (SVM) to filter services that users do not really like. The traditional service recommendation system usually sorts the candidate services according to the QoS value of the candidate services, such as Matrix Factorization (MF), as a popular method that has attracted extensive attention in the research. Yin *et al.* [12] proposed a QoS prediction model based on matrix factorization and neighborhood selection technology for network location awareness algorithm in this paper. Xin *et al.* [13] proposed a matrix factorization model for QoS prediction and then designed an Expectation-Maximization (EM) estimation scenario. According to the existing QoS data learning model, He *et al.* [14] proposed a location-based Hierarchical Matrix Factorization model (HMF), predicting QoS missing values by combining the results of local matrix factorization and global matrix factorization. Zou *et al.* [15] proposed an enhanced collaborative filtering model for QoS based service recommendation. Although matrix factorization can deal with the sparsity problem, it suffers information loss in most cases [16].It can be seen that these methods complete the prediction task by neighborhood information. However, they do not effectively address the different importance of different services, nor do they give much thought to the connection between users and services. Based on the above analysis, this paper researches it.

Recently, to make recommendations more in line with users' needs, researchers have begun to use contextual information to mine users' interests, and the accurate recommendation results have proved the positive effect of combining contextual information. Jiang *et al.* [17] considered social text

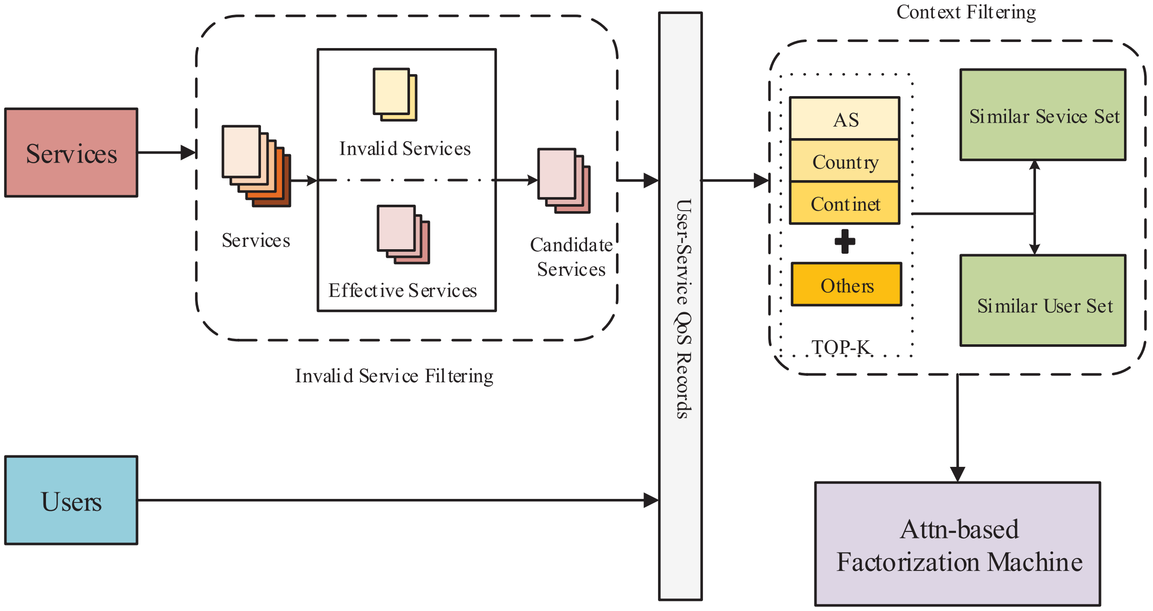


FIGURE 1. A two-tier filtering model: After filtering the model twice, the service called by the original user and similar service, the original user and similar user, similar user and similar service are output with attention-based FM.

information in the recommendation process. Chen *et al.* [18] proposed an MF model based on geographical neighborhood knowledge, which can alleviate the cold start problem. To better represent QoS data, Guo *et al.* [19] proposed three product recommendation models based on implicit user feedback combined with social trust, and introduced a matrix factorization technique to restore user preference between rated items and unrated items according to user-user and item-item similarity. Zhou *et al.* [20] integrated the user's network location information into the MF model. Although many studies have improved the prediction accuracy from a series of aspects, the existing methods have severe limitations and can only extract or learn shallow features. Chen *et al.* [21] proposed a web service recommendation method based on QoS prediction and hierarchical tensor decomposition. The method is called QoSHTD, which is based on location clustering and hierarchical tensor decomposition. Chowdhury *et al.* [22] studied QoS prediction among different users and proposed a layered QoS hybrid filtering prediction scheme considering service and user context information. Liu *et al.* [23] proposed a collaborative preference regression model, which combined probabilistic matrix factorization with probabilistic preference model to form user-related, service-related and subject-related potential factor models, and used them to predict users' interests. Although the existing method takes into account context information and makes the recommendation more personalized, it does not consider that the service processing in the early stage will affect the recommendation efficiency and other issues.This paper will also study the above problems.

III. PROPOSED MODEL

In this section, we will introduce our predicted model in detail. Specifically, the model is mainly divided into the following parts: the service filtering part in III-B, the context information filtering part in III-C, and the attention-based model output part in III-D.

A. OVERVIEW

Our task is to consider the impact of services and users to make accurate service recommendations. Based on this goal, this paper proposes a two-tier filtering model. First of all, we used the coefficient of variation method to filter the effectiveness of services and select effective services, which reduces the scale of services and improves the efficiency and quality of service recommendation. After that, service filtering is carried out based on the context to make the recommendation personalized. By establishing a user-service QoS matrix, the similarity of services invoked by these users is calculated to find out similar users. The similarity is then calculated according to the QoS value of the specific service to find a similar service invoked by the current user. According to the service and similar service called by the original user, original user and similar user, similar user and similar service, these three parts are calculated and output by using the attentional factorization machine model, effectively considering the impact of the importance of service on the results. As shown in Figure 1, it will be introduced in detail below.

B. SERVICE FILTERING MODEL

A necessary part of implementing a service recommendation in such sparse data is to reduce the size of the service and filter out invalid or irrelevant services. On the one hand, service filtering can greatly reduce the number of services, greatly reduce the computational complexity and improve recommendation efficiency. On the other hand, filtering out invalid services can greatly reduce the noise in user behavior data and improve the accuracy of service recommendations.

In general, any small change in location, network, environment, or other aspects will affect the consistency of candidate Web services QoS, which we call QoS instability. For example, if a current user is watching a movie on a mobile phone, due to the influence of various internal and external environments, unstable QoS data will be generated, and the movie may be played smoothly or temporarily. These phenomena are very common in real life. In the process of service recommendation, such a service with a large number of unstable QoS attributes and poor QoS attributes is called ''invalid service''. The presence of these invalid services can reduce the efficiency of service recommendations.

To avoid these invalid services occupying running memory and wasting selection time in the later service recommendation process, we first filtered the services, then removed the invalid ones based on QoS from a vast pool of services. Service filtering is mainly divided into two processes, as shown in Figure 2. One is to use the coefficient of variation method to calculate the coefficient of variation value of the service. Second, aiming at the size of coefficient of variation, the invalid services with unstable QoS attributes are filtered out and the scale of candidate services is reduced.

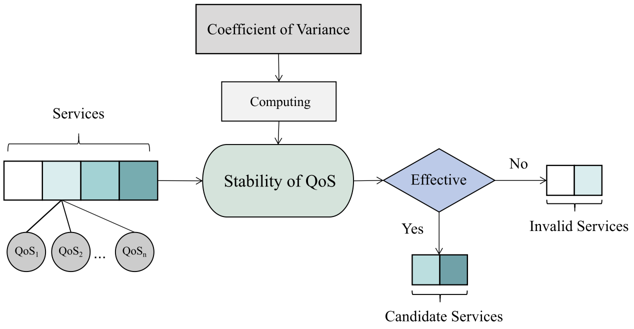


FIGURE 2. **Invalid service filtering model: Mainly filter out those invalid services whose QoS is not stable.**

The coefficient of variation method is used in this paper. Before the weight is calculated, a Web service set is established, represented by W, assuming that the service set contains M services, then $W = \{w_1, w_2, \ldots, w_i, \ldots, w_M\}$, $i \in (1, M)$, Where W_i stands for any Web service, and each service W_i has T attributes. The $W_i = \{q_{i1}, q_{i2}, \ldots, q_{ij}\}$, $j \in T$. where, q_{ij} represents the j^{th} attribute of service W_i. For each q_{ij}, there are n historical records, which can be expressed as $q_{ij} = \{p_1, p_2, \ldots, p_n\}$.

In this paper, the latest N records of QoS are regarded as a group of various random variables. Calculate the coefficient of variation value DU of the service, as shown in equation (1), (2) and (3):

$$\bar{Y}_j = \frac{1}{N}\sum_{n=1}^{N} p_n \tag{1}$$

$$\sigma_j = \sqrt{\frac{\sum\limits_{n=1}^{N} (p_n - \bar{Y}_j)^2}{N-1}} \tag{2}$$

$$DU_j = \frac{\sigma_j}{\bar{Y}_j} \tag{3}$$

The average value of attributes is denoted by $\bar{Y}$, $\bar{Y} \neq 0$; p_n represents the n^{th} record in the latest N QoS record. The mean value (standard deviation) of the attribute is represented by σ_j, and DU_j represents the j^{th} attribute's coefficient of variation value. Besides, when some QoS attributes of services are unstable, or the QoS attributes value does not exist within a certain period, such services are invalid, and the average value of invalid services is 0.

Through the above calculation of coefficient of variation values of service attributes, the coefficient of the variation set $DU_j = \{DU_{i1}, DU_{i2}, \ldots, DU_{ij}\}$ is obtained, DU_{ij} represents the coefficient of variation value of the j^{th} attribute of the i^{th} service. We sum up the coefficient of variation of all attributes of the service, calculate the total coefficient of variation value of each service, form the coefficient of variation set, and then arrange them from large to small. We choose the average value of the service coefficient of the variation set as a stable value. If the coefficient of variation is greater than the average, then the service is invalid and filtered. If the coefficient of variation is less than or equal to the average value, then the service is relatively stable and may meet users or enterprises' needs. The service is left as one of the candidate service sets.

C. CONTEXT FILTERING MODEL

After filtering the invalid service, in this section, we take the geographical location of the service and the user as the context factor, use the geographical location to filter, and then calculate the similarity between the user and the service. By solving the similar user and similar service, we filter out the invalid user and service, finally, we will be precise to recommend a useful combination of feature interaction.

When the geographical location of users or services is different, QoS will be affected, and the QoS of the same service experienced by users in the same area is relatively close. The QoS of various services experienced by users in the same area is often similar, so location information is an essential factor affecting QoS. Therefore, this paper uses user and service geographic location information to find similar users or similar services.

Suppose we have user $U = \{u_1, u_2, \ldots, u_n\}$ and the filtered services, which we call Web Services $C = \{c_1, c_2, \ldots, c_n\}$. When a user invokes a Web service, he (or she) can observe the QoS value of the service from his (or her) own perspective. Each user has a QoS matrix, the rows are the user's service, and the columns are all QoS labels. The numerical values in this matrix represent the specific value of QoS, and each QoS value represents the QoS value generated by the service invoked by the user.

First, we search the Web service users of user A in the same Autonomous System(AS), calculate the similarity between user A and other users, and take top-K similar users. Then, search the Web service users in the same country as user A, calculate the similarity between them, and take top-K similar users. Then search the Web service users on the same Continent as user A, calculate the similarity between them, and take top-K similar users. Finally, the data is summarized, the collection of all remaining users is calculated, and top-K is taken, making a complete query of similar users.

The traditional method can use the Euclidean distance to calculate the similarity between the two, but the Euclidean distance is not satisfactory for most statistical problems. The contribution of each coordinate to the Euclidean distance is equal. When coordinates represent measured values, they are often subject to random fluctuations of different sizes. In this case, a reasonable method is to weigh the coordinates, so that the coordinates with a massive change have a smaller weight coefficient than those with a small change, resulting in various distances. In this paper, an improved similarity calculation method is adopted to improve the user inverse invocation frequency and the service inverse invocation frequency, and the similarity between users is calculated by the service they both invoke. The user inverse invocation frequency calculation for service c:

$$\lambda_c = \log \frac{|u|}{|u_c|} \tag{4}$$

where $|u|$ is the total number of users, $|u_c|$ is the number of users who have already called service c and λ_c represents the frequency of user inverse call. The higher the λ_c of service c, the higher the distinguishing value of service c. The service inverse invocation frequency of user u is as follows:

$$\lambda_u = \log \frac{|c|}{|c_u|} \tag{5}$$

where $|c|$ is the total number of services and $|c_u|$ is the number of services called by user u. Similarly, a higher value means that user u has a higher recognition rate.λ_u represents the frequency of service inverse call. The inverse invocation frequency of users of user u is taken as the weight, and the proposed similarity is based on the Euclidean distance, calculated as:

$$S_{u,v} = \frac{1}{1+\sqrt{\sum_{c=0}^{R}((q_{u,c}-\bar{q}_u)-(q_{v,c}-\bar{q}_v))^2 \cdot \lambda_c \Big/ |R|}} \tag{6}$$

where $S_{u,v}$ is the similarity between user u and user v, and $R = R_u \cap R_v$ is the set of services invoked by user u and user v. $q_{u,c}$ is the real QoS value generated after the target user u invokes the target service c, $q_{v,c}$ is the real QoS value of the target service c. So to keep the denominator from being 0, let's add 1 to the denominator. $\bar{q}_u$ is the average QoS value of user u, and $\bar{q}_u$ is the average QoS value of user v.

The calculation of service similarity is similar to that of the user, as follows:

$$S_{c,h} = \frac{1}{1+\sqrt{\sum_{c=0}^{Z}((q_{u,c}-\bar{q}_c)-(q_{u,h}-\bar{q}_h))^2 \cdot \lambda_u \Big/ |Z|}} \tag{7}$$

where $S_{c,h}$ is the similarity between service c and service h, $Z = Z_c \cap Z_h$ is the user set of service c and service h that has been called before, $\bar{q}_c$ is the average QoS of service c, $\bar{q}_h$ is the average QoS value of service h, Z_c is the set of users who have called service c before, and Z_h is a group of users who have called service h back. According to similar results, selecting the K services with the highest similarity.

D. AN END-TO-END TWO-TIER SERVICE FILTERING MODEL

The FM component is a factorization machine for learning the characteristic interactions of service recommendations. Feature combination is essential for recommendation ordering, and the idea of FM has been a very concise and elegant embodiment of this idea (mainly second-order feature combination). Compared with previous models, the FM model can effectively solve the problem of data sparsity. In addition, FM has a strong generalization ability. Due to this flexible design, FM can train the potential vector v_i (or v_j) when i (or j) appears in the data record. Therefore, in the service data invoked by users, FM can better learn about the characteristic interactions of services that are never invoked or rarely present. At the same time, factor decomposition parameters are introduced, enabling the model to comprehensively consider the relationship between various dependent variables when learning weights.

The interaction between the two features i and j is modeled by FMs as the dot product of their corresponding embedded vectors v_i and v_j:

$$\hat{y}_{FM}(x) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p}\sum_{j>i}^{p} < v_i, v_j > x_i x_j \tag{8}$$

$w_0 \in R, w \in R^p, v \in R^{p\times k}$ is the parameter of the model, and p is the number of variables; $k \ll p$ represents the dimension of factorization; v_i represents the vector in the matrix v, that is, learn a vector with length k to express v_i for each characteristic x_i, that is, the vector v_i is the k-dimension expression of characteristic x_i. w_0 stands for global offset;w_i is the weight of the i^{th} variable; $< v_i, v_j >$ represents the inner product of two vectors, namely the hidden variable. The inner product unit represents the result of second-order characteristic interaction of the potential vector, which can be expressed as follows:

$$\langle v_i, v_j \rangle = \sum_{f=1}^{k} v_{i,f} v_{j,f} \tag{9}$$

The reason the FM can always learn some meaningfully embedded vectors for each feature is that, even if they never or rarely appear in the data at the same time, the dot product is an excellent way to evaluate the interaction between the two features as long as the feature itself appears in the data enough times.

In the process of service recommendation, the FM algorithm uses second-order features to improve the performance of the linear model. When FM uses all second-order cross features, each cross features weight by default, which is 1. However, each cross feature's usefulness is not the same, and some useless interaction features may have a negative impact on model learning and hinder the improvement of model performance. For example, for QoS-based service recommendation, if it is predicted that a user will choose Google map or Baidu map, then the service and the user's location will be more important than the location of the user and the service provider. So attention is used to distinguish the importance of different interaction characteristics, not all of them being equally important.

In order to introduce the attention mechanism, this paper presents the attention network between the ‘‘feature cross layer’’ and the ‘‘output layer’’. The function of the attention network provides weight for each cross feature. Finally, the definition of attention network is as follows:

$$a'_{ij} = h^T \operatorname{Re}LU\left(W\left(v_i \odot v_j\right) x_i x_j + b\right) \tag{10}$$

$$a_{i,j} = \frac{e^{a'_{i,j}}}{\sum_{(i,j)\in R_x} e^{a'_{i,j}}} \tag{11}$$

$$W \in \mathbb{R}^{t*k}, \quad b \in \mathbb{R}^t, \ h \in \mathbb{R}^t \tag{12}$$

where W, b, and h are model parameters, and t represents the size of the hidden layer of the attention network, which is called ‘‘attention factor’’. The Attention score is normalized through the Softmax function. We chose the ReLU function on the activation function, and the effect was also good.

The output of the attention-based pooling layer is a k-dimensional vector, and all feature interactions are compressed by recognizing their respective importance in the embedding space. We map this to the final prediction, and you can see that a_{ij} varies with $v_i \odot v_j$. Namely, the complete equation of the FM based on attention mechanism model is as follows:

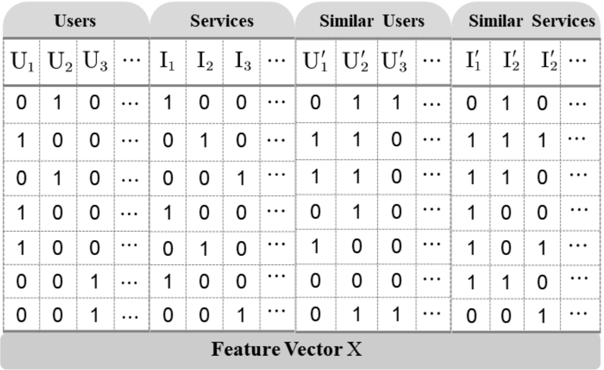
$$\hat{y}_{At}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + p^T \sum_{i=1}^{n} \sum_{j=i+1}^{n} a_{ij}(v_i \odot v_j) x_i x_j \tag{13}$$

According to [24], when user u calls service i, that is, figure 3, only x corresponding to u and i is 1, while the others are 0. Again, the only crossover factor is u, and the x for i is 1. Therefore, the calculation equation of FM based on attention mechanism can be expressed as:

$$\hat{y}(x) = w_0 + w_u + w_i + p^T * a_{ij} < v_u, v_i > \tag{14}$$

Add similar users and similar services as follows:

$$\widehat{y} = w_0 + w_u + w_i + p^T(F1 + F2 + F3) \tag{15}$$



Users				Services				Similar Users				Similar Services			
U_1	U_2	U_3	···	I_1	I_2	I_3	···	U'_1	U'_2	U'_3	···	I'_1	I'_2	I'_2	···
0	1	0	···	1	0	0	···	0	1	1	···	0	1	0	···
1	0	0	···	0	1	0	···	1	1	0	···	1	1	1	···
0	1	0	···	0	0	1	···	1	1	0	···	1	1	0	···
1	0	0	···	1	0	0	···	0	1	0	···	1	0	0	···
1	0	0	···	0	1	0	···	1	0	0	···	1	0	1	···
0	0	1	···	1	0	0	···	0	0	0	···	1	1	0	···
0	0	1	···	0	0	1	···	0	1	1	···	0	0	1	···

Feature Vector X

FIGURE 3. Schematic diagram of eigenvectors The eigenvector is made up of 0,1.

$$\left(\begin{array}{c} F1 = \dfrac{1}{|F_u|} \displaystyle\sum_{m\in F_u} a_{ij}(w_m + < v_u, v_m > + < v_i, v_m >) \\ F2 = \dfrac{1}{|F_i|} \displaystyle\sum_{m\in F_i} a_{ij}(w_n + < v_u, v_n > + < v_i, v_n >) \\ F3 = \dfrac{1}{|F_u||F_i|} \displaystyle\sum_{m\in F_u} \sum_{n\in F_i} a_{ij} < v_m, v_n > \end{array}\right) \tag{16}$$

where F_u and F_i respectively represent similar user sets and similar service sets, $|F_u|$ and $|F_i|$ respectively represent the size of user sets and service sets. w_m and w_n respectively represent the weight of influence of similar users and similar services. The F1, F2 and F3 respectively represent the result of the interaction of feature vectors of user u, similar user and similar service, and service i, similar user and similar service as well as similar user and similar service.

IV. EXPERIMENTAL DESIGN

In this section, we will conduct an experimental evaluation of our model and compare our model with the recommendation performance of the latest service recommendation models and classic recommendation models.

A. DataSet

In this article, we used a public data set of real Web service QoS data from wsdream. com (http://wsdream.github.io/). The data set has a total of 1,974,675 QoS records, including more than 1.5 million QoS records from 339 users and 5,825 Web services distributed around the world. Table 1 shows the statistics of the dataset. In a real service invocation scenario, the user service call matrix is usually sparse. We extract each QoS record to obtain a QoS matrix and get a 339*5828 matrix, which randomly removes part of the data to sparse into the training dataset, and the rest of the data forms the test set. This setup makes our experimental results more convincing. In the next experiment, we evaluate the performance of each model at four different Matrix densities of 5%, 10%, 15%, and 20%. For example, when the Matrix densities is 20%, then 20% of the available QoS

TABLE 1. Statistics of the dataset.

Notations	Values
QoS records	1974675
Number of users	339
Number of Web services	5825
Number of countries where users are located	30
Number of countries where services are located	73
Average response time	1.43s
Standard deviation of response time	31.9s

records are used for training, and the maining 80% of the QoS records are to be predicted. Each set is run 100 times, and the average result is reported.

B. EXPERIMENTAL ENVIRONMENT SETTINGS

We used 70 percent of the data as the training set, 20 percent as the validation set, and 10 percent as the test set. All experimental environments were based on python3.6 and implemented using pytorch 1.5. The model was trained and tested on NVIDIA TITAN X GPU.

C. EVALUATION METRICS

We adopt the widely used Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the prediction accuracy of the prediction methods.

MAE is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{17}$$

RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{18}$$

where n is the total number of predictions, $\hat{y}_i$ is the prediction, and y_i is the actual result. Smaller MAE and RMSE values indicate better prediction accuracy.

D. BASELINES

To verify our proposed model's effect, we selected some classic models on the prediction task as the baselines of our method. The baseline method we chose is as follows:

- NMF: This method was proposed by Lee *et al.* [25]. Compared with the traditional matrix factorization method, it mainly adds an extra constraint to the model and proposes a non-negative matrix factorization method.
- PMF ?This method was proposed by Salakhutdinov and Minh1 [26]. Probability was introduced into the traditional matrix factorization method, and the user-item matrix was used for recommendation.
- FM: FM is a classical factorization machine model. It was first proposed by Steffen Rendle [27] of Konstanz University in 2010, aiming at solving the problem of feature combination under sparse data.
- LACF: Recommended by Tang *et al.* [28] in 2012 for a Web service based on a location-aware collaborative filtering method based on QoS value prediction, LACF represents a hybrid location-aware collaborative filtering algorithm.
- SFNIMF: A Web service recommendation system based on QoS server-side context feature recognition was proposed by Li *et al.* [29] in 2017. This method makes full use of user preference information and Web service features.
- LAFM: In 2018, YANG *et al.* [30] proposed a QoS value prediction method based on the factorization machine, which not only uses QoS information of users and services but also uses QoS information of users and service neighbors.
- LE-MF: MF that extends geographical location information). Xu *et al.* [31] proposed in 2016, LE-MF integrates geographic graphic information and trust mechanisms into the traditional matrix decomposition model to predict QoS values.

V. EXPERIMENT RESULTS

A. PERFORMANCE OF THE BASELINES

In this section, we conduct an experimental evaluation of our approach, comparing our model's recommendation performance with that of the latest service recommendation model and the classic recommendation model. TABLE 2 shows the comparison between our model and various models. The MAE and RMSE values of our model are lower than those of other models, which means that our models generally prefer the results of other models. Compared with other traditional collaborative filtering models and matrix factorization models, the predicted results using FM are superior.

B. ABLATION EXPERIMENTS

1) THE IMPACT OF QoS MATRIX DENSITY ON PREDICTED RESULTS

Matrix density represents the sparsity of training data. We find that MAE and RMSE decrease when matrix density increases. In this experiment, we changed the density matrix from 2 to 20, and the step size is 2. When the training matrix density between 2 and 10, MAE and RMSE both decrease rapidly, which means that the predicted results are greatly improved. As the density increases further, MAE and RMSE begin to decrease slowly. This indicates that as the training set contains more items, the prediction accuracy will be higher. The results are shown in Figure 4.

2) THE IMPACT OF NEIGHBOR SET TOP-K ON PREDICTION RESULTS

The parameter top-K determines the size of the set of similar users and similar service neighbors. To assess its impact, the larger the top-K is, the larger the neighborhood will be. We performed sensitivity analysis at a constant increment of 5 in the range of 5∼25 on a matrix with a density of 10% and

TABLE 2. Results of the comparison with the baseline algorithm.

Models	Matrix Density=5%		Matrix Density=10%		Matrix Density=15%		Matrix Density=20%	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
NMF	0.6801	1.5908	0.7084	1.545	0.6812	1.4598	0.5598	1.4855
PMF	0.6473	1.5783	0.6985	1.4286	0.6081	1.4072	0.5032	1.4007
LACF	0.5821	1.4309	0.5229	1.3628	0.5032	1.3819	0.5086	1.3923
SFNIMF	0.5520	1.5118	0.5377	1.4596	0.4998	1.4002	0.4937	1.3871
FM	0.5381	1.413	0.5216	1.3589	0.4983	1.3554	0.4722	1.3165
LAFM	0.5229	1.3942	0.4839	1.3274	0.4469	1.3011	0.4156	1.2914
LE-MF	0.5279	1.4099	0.5014	1.3412	0.4649	1.3139	0.4434	1.2989
OURS	**0.5089**	**1.3762**	**0.4849**	**1.3058**	**0.3985**	**1.2835**	**0.3847**	**1.1870**

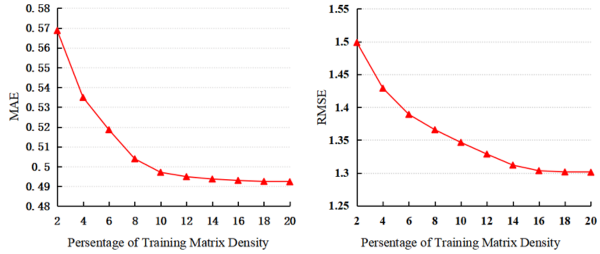


FIGURE 4. The influence of matrix density on prediction results.

15%. As K values increase, MAE and RMSE initially show a downward trend, and the prediction becomes more and more accurate. After K reached a certain threshold, MAE and RMSE also rose slightly. This indicates that too large or too small a top-K value will affect the prediction performance, so the appropriate K-value should be taken and top-K =20 in this paper.The results are shown in Figure 5.

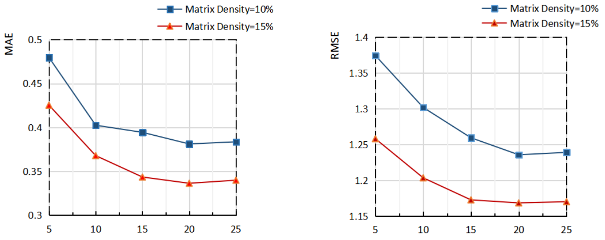


FIGURE 5. The influence of top-K on prediction results.

3) THE IMPACT OF SERVICE FILTERING ON PREDICTION RESULTS

In the model, we first filter the invalid service and verify the validity of the service filter. We set the number of QoS attribute to 2, and use the coefficient of variation method to filter out about 40∼60% of candidate services. This helps us get candidates with high reliability and lays the foundation for the next recommendation. We compared the reliability of the model to see whether the stability of the early service would influence the later recommendation.

$$Reliability = \frac{V_{\max} - \sum_{i=1}^{n} \sigma^2}{V_{\max} - V_{\min}} \times 100\% \tag{19}$$

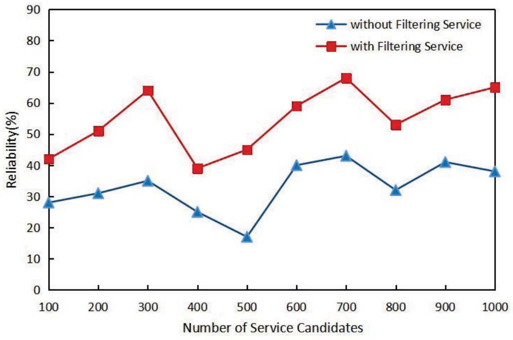


FIGURE 6. The reliability impact of service filtering.

where, V_{max} is the maximum aggregation variance value of all selected services, V_{min} is the minimum aggregation variance value of all selected services, and σ is the standard deviation of the candidate service selected for each service class. Through experimental comparison, it can be found that, as the number of candidate services increases, the reliability of the filtered model is always greater than that of the unfiltered model. This part of the experiment fully verified the influence of unstable filtering on service recommendation. The results are shown in Figure 6.

We carried out experiments on matrices with densities of 5%, 10%, 15% and 20%. The results are shown in Figure 7. After filtering, the MAE and RMSE are both smaller, which means that it is necessary to perform service filtering. Invalid services will have a particular impact on the service recommendation, a particular deviation inaccuracy, and will increase the time efficiency. The service filtering model can effectively filter out those invalid services, significantly reduce the search scope of services, the number of candidate services, and improve the efficiency and accuracy of service recommendation.

4) THE IMPACT OF ATTENTION NETWORK ON PREDICTED RESULTS

In the service recommendation process, we use the FM based on attention mechanism model instead of the FM model. We carried out experiments on the matrix with a density of 5%, 10%,15% and 20% respectively, and obtained the importance of using attention through experimental comparison results, as shown in Figure 8. However, FM assigns the

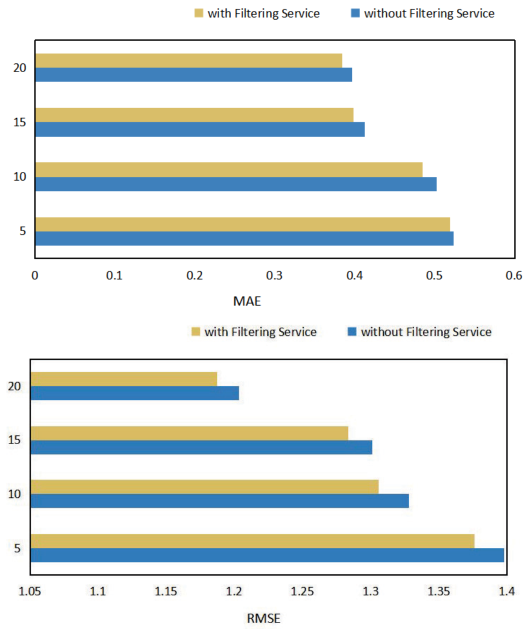


FIGURE 7. The influence of service filtering on prediction results.

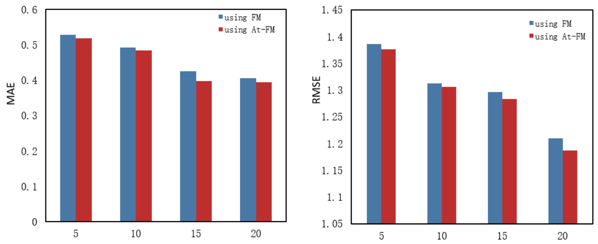


FIGURE 8. The influence of attention net on prediction results.

same importance score to all interactions, resulting in poor prediction performance, and useless features also generate noise, affecting prediction accuracy. In this paper, attention network is used to enhance FM, so that the interaction with similarity has higher importance rather than the same one, and the query results are more in line with users' preferences, and the prediction performance is improved.

VI. CONCLUSION AND FUTURE WORK

In this paper, A two-tier Filtering model based on the attentional factorization machine is proposed for Web service recommendation. We proposed a two-tier filtering model, focusing on the impact of invalid services generated in the recommendation process on the results, so we screened the service's stability and filtered the invalid services obtained. We consider the influence of contextual information, the service and user are filtered based on geographical location respectively, which effectively improves the efficiency of service recommendation. Simultaneously, the attention-based factorization machine model is used to ease data sparsity and noise influence, making it easier for the model to combine useful information related to the current output and effectively improve the quality of service recommendation. We conducted several experiments to verify the validity of our model.

In the future, we will focus on accurate filtering of user services, analyze and identify valid information in user behaviors from multiple perspectives, and provide better decision support for users.

REFERENCES

[1] G. Kang, M. Tang, J. Liu, X. Liu, and B. Cao, "Diversifying Web service recommendation results via exploring service usage history," *IEEE Trans. Services Comput.*, vol. 9, no. 4, pp. 566–579, Jul. 2016.

[2] S. Wang, Y. Zhao, L. Huang, J. Xu, and C.-H. Hsu, "QoS prediction for service recommendations in mobile edge computing," *J. Parallel Distrib. Comput.*, vol. 127, pp. 134–144, May 2019.

[3] L. Chen and W. Ha, "Reliability prediction and QoS selection for Web service composition," *Int. J. Comput. Sci. Eng.*, vol. 16, no. 2, pp. 202–211, 2018.

[4] L. Guo, D. Mu, X. Cai, G. Tian, and F. Hao, "Personalized QoS prediction for service recommendation with a service-oriented tensor model," *IEEE Access*, vol. 7, pp. 55721–55731, 2019.

[5] Y. Yin, W. Zhang, Y. Xu, H. Zhang, Z. Mai, and L. Yu, "QoS prediction for mobile edge service recommendation with auto-encoder," *IEEE Access*, vol. 7, pp. 62312–62324, 2019.

[6] X. Wu, B. Cheng, and J. Chen, "Collaborative filtering service recommendation based on a novel similarity computation method," *IEEE Trans. Services Comput.*, vol. 10, no. 3, pp. 352–365, May 2017.

[7] S. Wang, Y. Ma, B. Cheng, F. Yang, and R. N. Chang, "Multi-dimensional QoS prediction for service recommendations," *IEEE Trans. Services Comput.*, vol. 12, no. 1, pp. 47–57, Jan. 2019.

[8] F. Dahan, H. Mathkour, and M. Arafah, "Two-step artificial bee colony algorithm enhancement for QoS-aware Web service selection problem," *IEEE Access*, vol. 7, pp. 21787–21794, 2019.

[9] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized Web service recommendation via normal recovery collaborative filtering," *IEEE Trans. Services Comput.*, vol. 6, no. 4, pp. 573–579, Oct. 2013.

[10] L. Yao, Q. Z. Sheng, A. Segev, and J. Yu, "Recommending Web services via combining collaborative filtering with content-based features," in *Proc. IEEE 20th Int. Conf. Web Services*, Jun. 2013, pp. 42–49.

[11] L. Ren and W. Wang, "An SVM-based collaborative filtering approach for top-N Web services recommendation," *Future Gener. Comput. Syst.*, vol. 78, pp. 531–543, Jan. 2018.

[12] Y. Yin, S. Aihua, G. Min, X. Yueshen, and W. Shuoping, "QoS prediction for Web service recommendation with network location-aware neighbor selection," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 26, no. 4, pp. 611–632, May 2016.

[13] M. Xin, Y. Zhang, S. Li, L. Zhou, and W. Li, "A location-context awareness mobile services collaborative recommendation algorithm based on user behavior prediction," *Int. J. Web Services Res.*, vol. 14, no. 2, pp. 45–66, Apr. 2017.

[14] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-based hierarchical matrix factorization for Web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, Jun./Jul. 2014, pp. 297–304.

[15] G. Zou, M. Jiang, S. Niu, H. Wu, S. Pang, and Y. Gan, "Qos-aware Web service recommendation with reinforced collaborative filtering," in *Proc. Int. Conf. Service-Oriented Comput.*, 2018, pp. 430–445.

[16] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "Predicting Web service QoS via matrix-factorization-based collaborative filtering under non-negativity constraint," in *Proc. 23rd Wireless Opt. Commun. Conf. (WOCC)*, May 2014, pp. 1–6.

[17] M. Jiang, P. Cui, F. Wang, W. Zhu, and S. Yang, "Scalable recommendation with social contextual information," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 11, pp. 2789–2802, Nov. 2014.

[18] Z. Chen, L. Shen, F. Li, and D. You, "Your neighbors alleviate cold-start: On geographical neighborhood influence to collaborative Web service QoS prediction," *Knowl.-Based Syst.*, vol. 138, pp. 188–201, Dec. 2017.

[19] G. Guo, J. Zhang, F. Zhu, and X. Wang, "Factored similarity models with social trust for top-N item recommendation," *Knowl.-Based Syst.*, vol. 122, pp. 17–25, Apr. 2017.

[20] L. Zhou, Z. Song, S. Zhai, T. Xiao, and Y. Yin, "Predicting Web service QoS via combining matrix factorization with network location," *Int. J. U-E-Service, Sci. Technol.*, vol. 8, no. 3, pp. 303–318, Jun. 2014.

[21] T. Cheng, J. Wen, Q. Xiong, J. Zeng, W. Zhou, and X. Cai, "Personalized Web service recommendation based on QoS prediction and hierarchical tensor decomposition," *IEEE Access*, vol. 7, pp. 62221–62230, 2019.

[22] R. Roy Chowdhury, S. Chattopadhyay, and C. Adak, "CAHPHF: Context-aware hierarchical QoS prediction with hybrid filtering," 2020, *arXiv:2001.09897*. [Online]. Available: http://arxiv.org/abs/2001.09897

[23] X. Liu and I. Fulia, "Incorporating user, topic, and service related latent factors into Web service recommendation," in *Proc. IEEE Int. Conf. Web Services*, Jun./Jul. 2015, pp. 185–192.

[24] S. Rendle, "Factorization machines with libFM," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1–22, May 2012.

[25] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[26] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.

[27] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 995–1000.

[28] M. Tang, Y. Jiang, J. Liu, and X. Liu, "Location-aware collaborative filtering for QoS-based service recommendation," in *Proc. IEEE 19th Int. Conf. Web Services*, Jun. 2012, pp. 202–229.

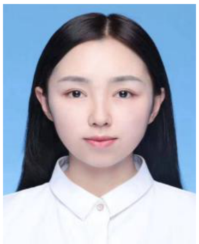
[29] S. Li, J. Wen, F. Luo, M. Gao, J. Zeng, and Z. Y. Dong, "A new QoS-aware Web service recommendation system based on contextual feature recognition at server-side," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 332–342, Jun. 2017.

[30] Y. Yang, Z. Zheng, X. Niu, M. Tang, Y. Lu, and X. Liao, "A location-based factorization machine model for Web service QoS prediction," *IEEE Trans. Services Comput.*, early access, Oct. 17, 2019, doi: 10.1109/TSC.2018.2876532.

[31] Y. Xu, J. Yin, S. Deng, N. N. Xiong, and J. Huang, "Context-aware QoS prediction for Web service recommendation and selection," *Expert Syst. Appl.*, vol. 53, pp. 75–86, Jul. 2016.

MINGYU LI received the B.S. degree in information and computing science from the Qilu University of Technology (Shandong Academy of Sciences), in 2017. She is currently pursuing the M.S. degree with the School of Computer Science and Technology. Her research interests include deep learning, distributed and services computing, and recommendation systems.

QIN LU received the Ph.D. degree in computer application technology from Dalian Maritime University, in 2015. She is currently an Associate Professor and the Dean of the Department of Software Engineering, Qilu University of Technology (Shandong Academy of Sciences). Her research interests include service computing, exception handling, and information integration.



MINGGE ZHANG received the B.Sc. degree in administration from the Tianjin University of Science and Technology, in 2018. She is currently pursuing the M.S. degree in software engineering with the Qilu University of Technology (Shandong Academy of Sciences). Her research interests include deep learning, interpretable reasoning, and recommendation systems.