Jin *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:222
https://doi.org/10.1186/s13638-019-1525-y

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Neighborhood-aware web service quality prediction using deep learning

Ying Jin[1], Kaibin Wang[2], Yiwen Zhang[2*] and Yuanting Yan[2]

Abstract

With the rapid growth of web services on the Internet, it becomes more difficult for users who want to choose the high-quality web services from a large number of functionally equivalent candidate services. Therefore, the prediction of quality of service (QoS) values according to the history of web services has received extensive attention. In recent years, deep learning has achieved great success in speech recognition, image processing, and natural language understanding. However, it is rarely applied to the service recommendation field. Therefore, a novel approach for QoS prediction named NDL (neighborhood-aware deep learning) is proposed. NDL first gets the Top-*k* neighbors of the user and the service through the Pearson correlation coefficient according to the service QoS information. Then, it extracts the potential features of the user neighbor and the service neighbor; after that, it inputs the QoS values of the user and the user neighbor as well as the QoS values of the service and service neighbors as a convolutional neural network. The results of experiments conducted on a real-world dataset demonstrate that the NDL significantly outperforms the current QoS prediction method in prediction accuracy.

Keywords: Quality of service (QoS), Quality prediction, Deep learning, Neighborhood information

1 Introduction

Currently, with the rapid growth of web services on the Internet, there are a large number of services with the same functions. Therefore, it becomes more challenging to recommend high-quality services to users. Quality of service (QoS) is widely used to evaluate the nonfunctional properties of services, such as QoS-based service composition [1–5], service discovery [6–9], and service recommendation [10], and has received wide attention from researchers.

However, the QoS information for most services is unknown to the user in reality, and the service quality attributes of the service are vulnerable to the environment [11, 12]. Different QoS values may be observed by users in different locations on the same service. Moreover, some service quality attributes (e.g., trust) are difficult to evaluate, requiring long-term observation and a large number of calls. Therefore, more efficient methods are needed to get QoS information about web services.

Collaborative filtering (CF) is an effective method to predict service quality [2, 3], and it can generally be divided into memory-based methods and model-based methods. The main steps of a memory-based CF method are to first identify similar users (similar services) by calculating the similarity between users (services) through the Pearson correlation coefficient (PCC), and then predict missing service quality values by using similar service quality information about the current user (service). However, when the data are sparse, PCC is not able to accurately measure the similarity between the users (services) to confirm similar users (services) which results in low prediction accuracy.

The model-based CF method needs to obtain a complex model based on the training data and machine learning method, and combine the historical data of similar users to predict the service quality of the target service. In recent years, matrix factorization (MF) technology has been successfully applied to model-based service quality prediction [13, 14]. The prediction based on MF can solve the data sparsity problem, which is the most popular method of service quality prediction at present.

Recently, due to its strong expressive and learning ability, deep learning has been successfully applied in

*Correspondence: zywahu@qq.com
[2]School of Computer Science and Technology, Anhui University, Hefei, 230601, China
Full list of author information is available at the end of the article

many fields, including speech recognition [15], image processing [16], and natural language understanding [17]. In the field of recommended systems, the restricted Boltzmann machine was first proposed to simulate a user's explicit rating of items [18], and autoencoders and denoising autoencoders were also used for recommendations [19–21]. However, research using deep learning in the field of services recommendation is seldom. Therefore, we propose the neighborhood-aware deep learning method to predict the service quality of web services. It can effectively learn the nonlinear relationship between users and user neighbors, and services and service neighbors through multi-layer perceptron (MLP) and convolution neural networks (CNN), thus significantly improving the accuracy of service quality prediction. The main contributions of this paper are as follows:

- We propose a novel deep learning model for predicting the QoS value of web services, which can effectively learn the nonlinear relationship between users and user neighbors through an MLP.
- The proposed model not only considers the service quality of the target users and target services but also uses the CNN to extract the potential nonlinear feature relationships between the information of user neighbors and the service neighbors.
- Extensive experiments have been conducted on the WS-Dream real-world dataset, and it indicates that the NDL significantly outperforms the existing service quality prediction approach.

The rest of this paper is organized as follows. Section 2 reviews the related work of QoS prediction. Section 3 explains the implementation process and details of NDL. Section 4 presents the experimental settings and results analysis, and Section 5 concludes the paper.

2 Related work

CF has been widely used in various commercial recommendation systems [22, 23]. Breese et al. [24] proposed a user-based CF method. Sarwar et al. proposed an item-based CF approach [25]. McLaughlin and Herlocker improved user similarity calculations by modifying the traditional Pearson correlation coefficient (PCC) [26]. Liu et al. calculated the similarity between users through public items evaluated by users [27]. Currently, CF methods have been successfully applied to the service recommendation field [5, 28]. Shao et al. [5] introduced user-based CF methods into service quality prediction.

However, these methods only consider the set of user neighbors and have low prediction accuracy. To solve this problem, a hybrid collaborative filtering method named WSRec [29] has been proposed by Zheng et al. This method obtains the user neighbor set and the service neighbor set by calculating the similarities between users and services through the PCC to predict the missing QoS value of the user to the target service. In addition, weights have been introduced to balance the user-based and service-based CF methods, which achieved better prediction accuracy.

In recent years, to further improve prediction accuracy, researchers have proposed some more complicated CF-based prediction methods. They can be broadly divided into two categories. The first type of method attempts to dig deeper into the user service matrix to use more information for prediction. For example, Jiang et al. [30] proposed a personalized CF service quality prediction method by discovering QoS personalized features of the user and service to improve the similarity calculation method. Wu et al. [31] used the data s*f*moothing technology of the user service QoS matrix to propose an improved collaborative filtering method, which can alleviate the problem of data sparsity to a certain extent and improve prediction accuracy. The second type of method attempts to use additional user and service information to assist the prediction. Chen et al. [32] added IP address information to the PCC when calculating the similarity between users and services and then used the Top-*k* mechanism to obtain similar users and similar services. Lo et al. [33] considered using the user's geographic location information (longitude and latitude) to obtain a similar set of users, but we know that a similar geographic location between users does not indicate a similar network environment and QoS experience. Therefore, Tang et al. [34] improved the prediction accuracy of QoS by dividing the autonomous regions of users and services to obtain similar users and similar services.

The existing improved CF-based QoS prediction methods demonstrate a significant improvement in prediction accuracy when compared with the traditional CF-based method, but still face problems such as data sparsity. To solve the problem, matrix factorization (MF) becomes another widely used QoS prediction technology in recommendation systems. Zhang et al. [10] combined user's neighbor information with MF methods to predict missing service quality values, and used the PCC to calculate similarity between users. Lo et al. combined location-based neighborhood information of users in an MF-based predictive model and assumed that users in similar geographic areas tend to share similar QoS experiences in web services [33]. Xu et al. [35] used a similar method, which confirmed the user's neighbors based on the geographical accuracy and latitude between users. However, two users who are physically close are not necessarily close in the network. Tang et al. proposed a network-aware MF-based prediction method that combines a network map into a predictive model, uses a network map to measure the network distance between users, and then identifies the user's neighbors [14].

The above matrix MF-based prediction methods use the inner product of the user's feature vector and the service's feature vector. The method considers the relationship between the user and the service to be linear, but in real life, this relationship may not only be linear, and the deep learning method can theoretically fit any nonlinear function [36]. Therefore, we propose the application of the deep learning method to QoS prediction. This paper not only considers the service quality information of the user and the service but also uses the CNN to learn the neighbor information of the user service, which effectively improves the prediction accuracy of the service quality.

3 Model

3.1 Problem definition

First, we formally define two concepts, namely neighbor users and neighbor web services.

Definition 1 (Neighbor Users) *Given a user u, its neighbor users* $N(u)$ *are defined as the user that have similar QoS experiences as user u on a set of commonly invoked web services.*

Definition 2 (Neighbor Web Services) *Given a web service i, its neighbor web services* $N(i)$ *are defined as the web service that delivers similar QoS experiences as service i to a group of users.*

Given a user u, a web service i, a set of users $U = \{u_1, u_2, \cdots, u_m\}$, and a set of web services $I = \{i_1, i_2, \cdots, i_n\}$, the method proposed in this paper mainly consists of two major operations: (1) to identify $N(u)$ from U and $N(i)$ from I, and (2) to predict the quality of web service i for user u based on $N(u)$ and $N(i)$.

3.2 Neighbor selection

Given a user service matrix R which includes m users and n services, and every element, R_{ij} in R shows the value of a client service quality attribute of service j observed by user i. R_{ij} is empty if service j has not been invoked by the users before. Pearson's correlation coefficient (PCC) or vector space similarity (VSS) can calculate the similarity between different users by using the available service quality values in the user item matrix collected from different users. It is common to use PCC and VSS as similarity calculation methods. As [29] and [30], since PCC takes into account differences in user value styles and enables high precision, it can commonly get higher performance than VSS, which is the reason we use PCC as the similarity calculation method. On the basis of the QoS values they observed in the jointly invoked web service, it can calculate the similarity between user u and v by using PCC, as calculated by Eq. (1):

$$\mathrm{PCC}(u,v) = \frac{\sum_{j\in J}(R_{u,j} - \overline{R}_u)(R_{v,j} - \overline{R}_v)}{\sqrt{\sum_{j\in J}(R_{u,j} - \overline{R}_u)^2}\sqrt{\sum_{j\in J}(R_{v,j} - \overline{R}_v)^2}} \tag{1}$$

where J represents the set of services invoked by users u and v, and $R_{u,j}$ shows user u's QoS value for service j. $\overline{R}_u$ and $\overline{R}_v$ are the average QoS value of the services which are evaluated by users u and v, respectively. The value range of the $\mathrm{PCC}(u,v)$ is $[-1, 1]$, and the larger the PCC value, the greater the similarity between users.

PCC value can identify a set of Top-k similar users by calculating the similarity between current user and other users. However, a user may have a limited number of similar users. And this problem is ignored by Top-k algorithm which still includes different users with negative PCC values; thus, it will lead to the great influence on the prediction accuracy. In our method, different users (negative *PCC* values) with negative correlations are excluded by us. Thus, user u can identify a group of similar users by using the following equation:

$$N(u) = \{v | v \in \mathrm{Top} - k(u), \mathrm{PCC}(u,v) > 0, u \neq v\} \tag{2}$$

where Top-$k(u)$ is a set of Top-k similar users to the current user u; $\mathrm{PCC}(u,v)$ is a similarity value between the users u and v and can be calculated by Eq. (1). Note that the Top-k relationship is asymmetric. User v is in the Top-k neighbor of user u which does not mean that user u is also a Top-k neighbor of user v.

Similarly, on the basis of the QoS values they have invoked on the user, it can calculate the similarity between services i and j by using the PCC, as calculated by Eq. (3):

$$\mathrm{PCC}(i,j) = \frac{\sum_{v\in V}(R_{v,j} - \overline{R}_i)(R_{v,j} - \overline{R}_j)}{\sqrt{\sum_{v\in V}(R_{v,j} - \overline{R}_i)^2}\sqrt{\sum_{v\in V}(R_{v,j} - \overline{R}_j)^2}} \tag{3}$$

where V represents the set of users invoking services i and j, and $R_{v,j}$ the service quality value of user v for service j. $\overline{R}_i$ and $\overline{R}_j$ are the average service quality values of services i and j, respectively.

According to the PCC value, a set of Top-k similar services after calculating the similarity between the current service and other services can be identified. Service i can identify a group of similar users by the following equation:

$$N(i) = \{j | j \in \mathrm{Top} - k(i), \mathrm{PCC}(i,j) > 0, i \neq j\} \tag{4}$$

where Top-$k(i)$ represents a set of Top-k similar services with current service i. *PCC*(i,j) is the similarity value between services i and j and can be calculated by Eq. (3). Similarly, the Top-k relationship is asymmetric.

3.3 NDL model

Figure 1 shows the overall framework of the method in this paper, and it mainly has the following three functional modules:

1) User information learning module: To generate user QoS vector and user neighbor QoS matrix according to historical QoS information and Top-k neighbor users of target users, respectively, as input of MLP and CNN for nonlinear learning to get the user feature vector and the user neighbor feature vector. Then, the user feature vector and the user neighbor feature vector are combined as the input of the multi-layer perceptron (MLP) to perform nonlinear learning, and the vector p is obtained.

2) Service information learning module: To generate service QoS vector and service neighbor QoS matrix according to historical QoS information of target services and Top-k neighbor services of target services, as input of the MLP and CNN respectively for nonlinear learning to get the service feature vector and the service neighbor feature vector. Then, the service feature vector and the service neighbor feature vector are combined and used as the input of the MLP to perform nonlinear learning, and the vector q is obtained.

3) Prediction module: The predicted QoS value is obtained through the inner product of the vector p obtained by the user information learning module and the vector q obtained by the service information learning module.

In the following sections, we will detail the implementation procedure of the above three functional modules.

3.3.1 User information learning module

Here, we assume that the target user is u, and from Section 3.2, we can get the neighbor $N(u)$ of user u according to the user service matrix R. Then, the QoS vector V_u of the user u and the QoS matrix M_u of the Top-k neighbors of the user u are extracted from the user service matrix R, respectively. V_u and M_u are the input of MLP and CNN, respectively, as shown on the left side of Fig. 1. More formally, the forward propagation process is defined as follows:

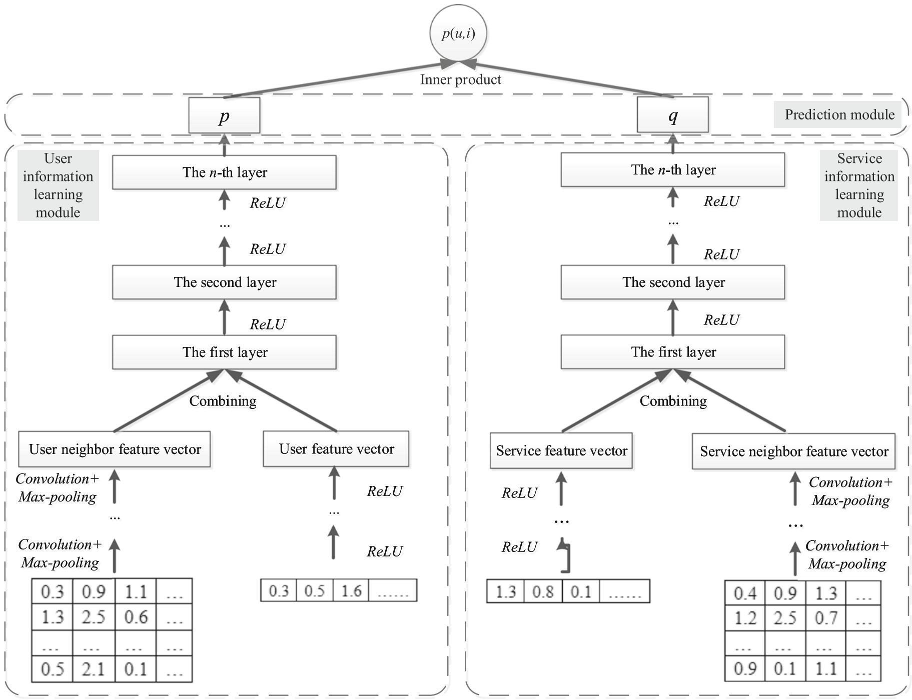
$$x = V_u \tag{5}$$

$$x_u = f_n\left(W_n^T(f_{n-1}(\cdots f_1(W_1^T x + b_1) + b_{n-1} + b_n\right) \tag{6}$$

$$a_n^u = f_n^u(f_{n-1}(\cdots f_1(\mathrm{convolution} + \max - \mathrm{pooling}(M_u)))) \tag{7}$$

$$v_n = \mathrm{flatten}\left(a_n^u\right) \tag{8}$$

It can be seen from the above formulas that the user QoS vector V_u is nonlinearly learned as the input of the MLP to obtain the vector x_u. The user neighbor QoS matrix M_u is nonlinearly learned as the input of the CNN to obtain



0.3	0.9	1.1	…
1.3	2.5	0.6	…
…	…	…	…
0.5	2.1	0.1	…

0.3	0.5	1.6	……

1.3	0.8	0.1	……

0.4	0.9	1.3	…
1.2	2.5	0.7	…
…	…	…	…
0.9	0.1	1.1	…

Fig. 1 Overall framework of NDL method

the vector v_u, where w_k, b_k, and f_k represent the weight matrix, offset vector, and activation function of the kth layer perceptron, respectively. Convolution + max − pooling are the convolution and pooling operations, f_n^u is the activation function, and flatten function is used to flatten the input tensor into a one-dimensional vector. For the activation function of the MLP and the CNN, *Sigmoid*, hyperbolic tangent function (*Tanh*), and rectifier (*ReLU*) can be freely selected.

The *Sigmoid* function limits each neuron to (0,1), which may limit the performance of the model in this paper. It is well known that when the output of a neuron approaches 0 or 1, learning will stop. Moreover, this paper aims to predict the QoS value of web services, which is essentially a regression problem, and the *Sigmoid* function is more suitable for a binary classification problem. Even if the *Tanh* function is a better choice and widely used, it only relieves the problem of the *Sigmoid* function to a certain extent because it can be regarded as an upgraded version of $Sigmoid(Tanh(x/2) = 2\sigma(x) - 1))$. Similar to the *Sigmoid* function, the *Tanh* function is more suitable for a binary classification problem. Therefore, we chose the *ReLU* function, which is more reasonable and proven to be unsaturated. In addition, it encourages sparse activation, is ideal for sparse data, and makes the model less likely to be overfitted.

Then, x_u and v_u are combined into the input of the MLP. More formally, the forward propagation process is defined as follows:

$$x = \begin{bmatrix} x_u \\ v_u \end{bmatrix} \tag{9}$$

$$a_n = f_n(W_n^T(f_{n-1}(\cdots f_1(W_1^T x + b_1) + b_{n-1} + b_n) \tag{10}$$

$$p = a_n \tag{11}$$

where w_k, b_k, and f_k represent the weight matrix, the offset vector, and the activation function of the kth layer perceptron, respectively. We use the *ReLU* function as the activation function here.

3.3.2 Service information learning module

Similar to Section 3.3.1, here, we assume that the target service is i, and from Section 3.2, we can get the neighbor $N(i)$ of service i according to the user service matrix R. Then, the QoS vector V_i of service i and the QoS matrix M_i of the Top-k neighbors of the service i are extracted from the user service matrix R, respectively. V_i and M_i are input as the MLP and the CNN, respectively, as shown on the right side of Fig. 1. More formally, the forward propagation process is defined as follows:

$$x = V_i \tag{12}$$

$$x_i = f_n(W_n^T(f_{n-1}(\cdots f_1(W_1^T x + b_1) + b_{n-1} + b_n) \tag{13}$$

$$a_n^i = f_n^i(f_{n-1}(\cdots f_1(\text{convolution+max−pooling}(M_i)))) \tag{14}$$

$$v_i = \text{flatten}\left(a_n^i\right) \tag{15}$$

It can be seen from the above formulas that the service QoS vector V_i is nonlinearly learned as the input of the MLP to obtain the vector x_i. The service neighbor QoS matrix M_i is used as the input of the CNN for nonlinear learning to obtain the vector v_i, where w_k, b_k, and f_k represent the weight matrix, the offset vector, and the activation function of the kth layer perceptron, respectively. Similarly, convolution + max − pooling are the convolution and pooling operations, f_n^i is the activation function, and the flatten function is used to flatten the input tensor into a one-dimensional vector. We use the *ReLU* function here for the activation function of the MLP and the CNN.

Then, we combine x_i and v_i into the input of the MLP. More formally, the forward propagation process is defined as follows:

$$x = \begin{bmatrix} x_i \\ v_i \end{bmatrix} \tag{16}$$

$$a_n = f_n(W_n^T(f_{n-1}(\cdots f_1(W_1^T x + b_1) + b_{n-1} + b_n) \tag{17}$$

$$q = a_n \tag{18}$$

where w_k, b_k, and f_k represent the weight matrix, the offset vector, and the activation function of the kth layer perceptron, respectively. We use the *ReLU* function as the activation function here.

As for the design of the network structure, a common solution is to follow the tower mode, with the widest bottom layer and each successive layer having a small number of neurons. The precondition is that by using a small number of hidden units at a higher level, they can learn more about the abstract characteristics of the data. We have empirically implemented the structure of the tower so that the size of the layer is half that of each successive higher layer.

3.3.3 Prediction module

The vector p and the vector q are obtained according to Sections 3.3.1 and 3.3.2, and the final predicted value can be obtained as follows:

$$p(u, i) = p^T q \tag{19}$$

3.4 Model learning

Another important problem of the QoS prediction model is to define an appropriate objective function for model optimization based on observed data and unobserved feedback.

The general objective function is as follows:

$$L = \sum_{y \in Y} l(y, p(u,i)) \tag{20}$$

where $l(\cdot)$ represents the loss function and y is the nonzero element in the training user service matrix R.

For recommendation system, the loss function is the most important part of the objective function. In many existing models, the square loss is mainly manifested as follows:

$$L = \sum_{y \in Y} (y, p(u,i))^2 \tag{21}$$

For the NDL model to start training from scratch, we use adaptive moment estimation (Adam), which performs a small update to frequent updates and performs large updates on infrequent parameters to accommodate each parameter learning. The Adam method converges faster than the normal SGD for both models and alleviates the difficulty of adjusting the learning rate.

The total implementation procedure of NDL is presented in Algorithm 1.

4 Experimental results and analysis

4.1 Dataset

To evaluate the effectiveness of the method, we use a real-world web service dataset, WS-Dream, as the test dataset. The dataset contains 1,974,675 QoS records, which were obtained from 5825 web services distributed in 73 countries invoked by 339 computers (users) distributed in 30 countries. There is a QoS record generated by each invocation between each user and each web service. Each QoS record has two attributes, response time (RT) and throughput (TP). For more details about the dataset, please refer to [37].

4.2 Evaluation metrics

This paper uses two basic indicators to evaluate the predictive performance of QoS prediction methods, including MAE (mean absolute error) and RMSE (root mean square error), and these two evaluation indicators have been widely used in the field of service computing [7, 8, 24].

MAE is defined as:

$$\text{MAE} = \frac{\sum_{u,i} |R(u,i) - P(u,i)|}{L} \tag{22}$$

and RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{u,i} (R(u,i) - P(u,i))^2}{L}} \tag{23}$$

where $R(u,i)$ and $P(u,i)$ are the real and predicted QoS values, respectively, and L is the total number of predicted values. The lower the MAE and RMSE, the higher the prediction accuracy.

Algorithm 1: NDL

```
Input:
  $Iters$: # of training iterations
  $R$: original rating matrix
  $M_u$: QoS matrix of the Top-$k$ neighbors of user $u$
  $M_i$: QoS matrix of the Top-$k$ neighbors of service $i$
Output:
  $W_k (k = 1, 2, \cdots, N-1)$: weight matrix
  $b_k (k = 1, 2, \cdots, N-1)$: bias vector
Begin
  Initialisation :
    randomly initialize $W$ and $b$;
    set $Y$ $\leftarrow$all none zero elements in $R$;
    for $it$ from 1 to $Iters$ do
      for each element of user $u$ and service $i$ in $Y$ do
        set $x_u$ $\leftarrow$ use Equation 5-6 with input of $u$;
        set $v_u$ $\leftarrow$ use Equation 7-8 with input of $u$ and $M_u$;
        set $x_i$ $\leftarrow$ use Equation 12-23 with input of $i$;
        set $v_i$ $\leftarrow$ use Equation 14-15 with input of $i$ and $M_i$;
        set $p$ $\leftarrow$ use Equation 9-11 with input of $u$ and $M_u$;
        set $q$ $\leftarrow$ use Equation 16-18 with input of $i$ and $M_i$;
        set $p(u,i)$ $\leftarrow$ use Equation 19 with input of $p$ and $q$;
        set $y$ $\leftarrow$ input of $u, i$ and $Y_{ui}$;
        set $L$ $\leftarrow$ user Equation 21 with input of $p(u,i)$ and $y$;
        use back propagation to optimize model parameters
      end for
    end for
  End
```

4.3 Comparing methods

In this section, we compare our method with the following methods to verify the accuracy of the prediction. The comparison methods are as follows:

(1)UPCC [5]: A user-based collaborative filtering algorithm that first uses the PCC to calculate the similarity between different users, and then uses the neighboring user's QoS value for prediction.

(2)IPCC [25]: A service-based collaborative filtering algorithm that first uses the PCC to calculate the similarity between different web services, and then uses the QoS value of the neighbor service for prediction.

(3)UIPCC [29]: A hybrid collaborative filtering method that combines the QoS prediction values of UPCC and IPCC to predict missing values, and adds a parameter to balance the effects.

(4)CMF [38]: The classical matrix factorization method that does not consider user neighbor information and service neighbor information.

(5) NMF [39]: Nonnegative matrix factorization, a matrix factorization method that maintains nonnegative constraints.

(6) PMF [40]: Probability matrix factorization, a method that adds probability distribution based on the traditional matrix factorization method and uses the Bayesian method to derive the posterior probability of implicit features of users and services, and finally combines with matrix factorization.

(7) NIMF [13]: An improved matrix factorization method that uses the PCC to obtain similar users, and then combines QoS information of similar users with the matrix factorization method for QoS prediction.

(8) NAMF [14]: An improved matrix factorization method. This method is different from [10] because it uses the geographic location information of the user to obtain similar users, and then combines the QoS information of similar users with the matrix decomposition method for QoS prediction.

4.4 Experimental setup

In the comparison experiment, in order to simulate the actual web service application scenario, a part of the QoS data in the user service matrix is randomly removed to make the data sparse, and the matrix densities are 5%, 10%, 15%, and 20%. The sparse matrix is used as training data, while the removed QoS value is used as a test set.

All experiments were performed on a machine equipped with an Intel i7-4790 3.60 GHz CPU, 4 GB RAM, running Ubuntu 16.04. Based on the TensorFlow framework, we implemented the proposed method. For the neural network, we used the Gaussian distribution (avg = 0, stdev = 0.01) to randomly initialize the model parameters, and used a small batch of Adam to optimize the model. To make the experiment more complete, we have listed the important parameters of NDL, as shown in Table 1

Table 1 Important parameters of NDL

Parameters	Values
Batch size	32
Learning rate	0.0001
Layers of MLP	3
Layers of CNN	3
Strides	[2,2]
Filter	[2,2]

4.5 Effectiveness

This section gives the prediction accuracy of the NDL and the methods in Section 4.3 at 5% to 20% sparsity as well as the prediction accuracy of the NDL at the parameter matrix density and *k* value.

(1) Prediction accuracy

From Table 2, it can be concluded that NDL has the highest prediction accuracy compared to the methods in Section 4.3 because NDL gets the lowest MAE and RMSE at different matrix densities.

Memory-based CF methods, including UPCC, IPCC, and UIPCC, achieve the worst prediction accuracy. In particular, the MF-based method is significantly better than the memory-based CF method when the data are very sparse. However, the deep learning-based prediction method NDL in this paper is superior to matrix factorization-based prediction methods in RMSE and MAE. As the density increases, the performance of all methods increases. This indicates that higher quality information can improve prediction accuracy. A significant advantage of NDL over matrix factorization methods is that it takes into account the nonlinear relationship between users and services, and considers the nonlinear relationship of neighbor information.

(2) Impact of matrix density on prediction accuracy

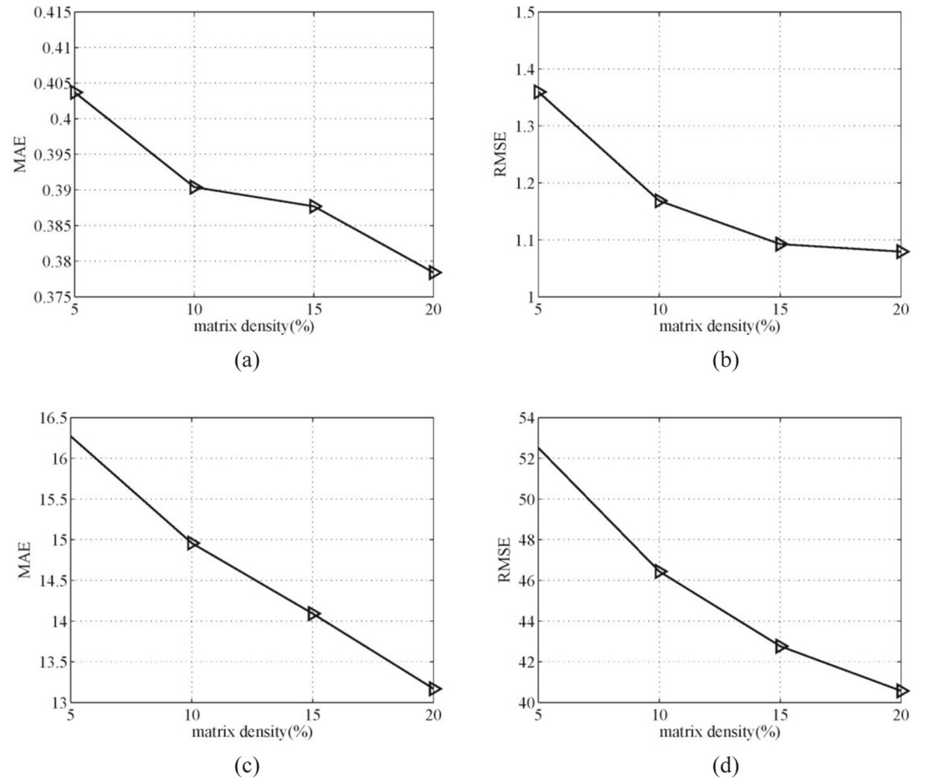
To further evaluate the impact of matrix density on NDL, we varied the density of the user-service matrix used in the experiment from 5% to 20%. Fig. 2 a and b are the MAE and RMSE values in RT (response time), respectively; c and d are the MAE and RMSE values in TP (throughput), respectively. It can be seen from the figure that as the matrix density increases, both the MAE and RMSE values decrease, indicating that the prediction accuracy is improved. This illustrates the importance of adequate input data for service quality prediction. In particular, Fig. 2 shows that the decrease in both MAE and RMSE is more significant at the beginning of the increase in matrix density, which shows the remarkable advantage of NDL.

(3) Impact of *k* on prediction accuracy

This section uses a 5% density matrix to evaluate the impact of Top-*k* on the prediction accuracy of this method. The parameter Top-*k* controls the size of the trusted user set. The smaller the *k* value is, the smaller the user's set of trusted users will be, and vice versa. The *k* values are 5, 10, 15, and 20, respectively. The experimental results in RT are shown in Fig. 3 a and b are the MAE and RMSE values in RT (response time), respectively; c and d are the MAE and RMSE values in TP (throughput),

Table 2 Comparison of MAE and RMSE on the WS-DREAM dataset

Attributes	Methods	Matrix density = 5%		Matrix density = 10%		Matrix density = 15%		Matrix density = 20%	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
RT	UPCC	0.9553	2.1269	0.7823	1.8569	0.6716	1.7264	0.5972	1.7177
	IPCC	1.1026	2.2583	0.8780	1.9893	0.7840	1.8628	0.7223	1.7948
	UIPCC	0.8471	1.9208	0.7290	1.7308	0.6128	1.5906	0.5520	1.5878
	CMF	0.6116	1.4142	0.5169	1.3562	0.4917	1.2163	0.4591	1.1988
	NMF	0.6182	1.5746	0.6040	1.5494	0.5990	1.5345	0.5982	1.5331
	PMF	0.5678	1.4735	0.4996	1.2866	0.4720	1.2163	0.4492	1.1828
	NIMF	0.5514	1.4075	0.4854	1.2745	0.4534	1.1980	0.4357	1.1678
	NAMF	0.5384	1.3853	0.4850	1.2592	0.4529	1.2071	0.4350	1.1443
	NDL	*0.4037*	*1.3596*	*0.3904*	*1.1686*	*0.3877*	*1.0924*	*0.3783*	*1.0793*
TP	UPCC	56.4816	95.4345	47.3569	78.1629	41.6976	70.9251	37.2768	67.9981
	IPCC	46.5634	79.6976	42.5893	73.5783	36.5033	68.4784	34.3576	65.4433
	UIPCC	40.0451	74.5033	36.4308	64.9208	33.8068	59.5171	29.2445	56.5301
	CMF	30.8275	69.2836	26.4586	59.3657	21.7944	52.9409	17.8588	47.1635
	NMF	25.7529	65.8517	17.8411	53.9896	15.8939	51.7322	15.2516	48.6330
	PMF	19.9034	54.0508	16.1755	46.4439	15.0956	43.7957	14.6694	42.4855
	NIMF	17.9297	52.6573	16.0542	46.9409	14.4346	43.1596	13.7099	41.1689
	NAMF	18.0837	52.8658	15.9808	46.9788	14.6661	43.0206	13.9386	40.7481
	NDL	*16.2700*	*52.5069*	*14.9585*	*46.4381*	*14.0924*	*42.7671*	*13.1675*	*40.5656*



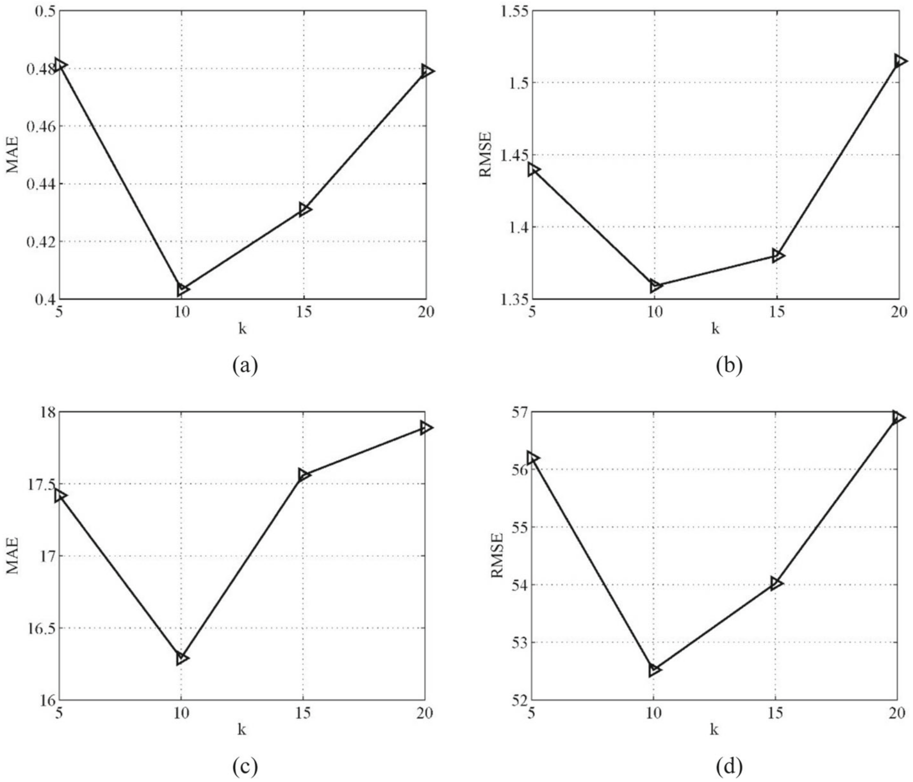
(a)

(b)

(c)

(d)

Fig. 2 Impact of matrix density on prediction accuracy



(a)

(b)

(c)

(d)

Fig. 3 Impact of *k* on prediction accuracy

respectively. As the value of k increases, the values of MAE and RMSE decrease. However, when k exceeds a certain threshold, the values of MAE and RMSE rise accordingly. This shows that taking appropriate values for k is beneficial to improve prediction accuracy in that when the value of k is too small, the user's trusted user set will be undersized, some of the user's trusted users will be ignored, and the information of the trusted user is not fully utilized, thereby reducing the prediction accuracy. When the value of k is too large, the user's set of trusted users will be oversized and may contain some untrusted users. The QoS values of these users are actually noise data, thus reducing the measurement accuracy. Figure 3 shows that both MAE and RMSE reach the lowest value when $k = 10$.

5 Conclusion

In this paper, we propose NDL, a method based on neighborhood-aware deep learning to predict the QoS values for web services. This method applies the deep learning method to capture the nonlinear relationship between users and services, and combines the neighbor information of users and services into the deep learning method, which further improves the prediction accuracy of service quality. First, Pearson's correlation coefficient is used to obtain the neighbor information of the user and the service; then, the user (service) information and the user (service) neighbor information perform learning as the input of MLP and CNN, respectively. In this paper, a series of experiments are conducted on a real web service dataset. The experimental results show that the QoS prediction accuracy of this method is significantly improved when compared with the previous methods. In our future work, we aim to apply the method of this paper to multiple datasets to verify the effectiveness of the proposed method.

Abbreviations

b_k: The offset vector of the kth layer perceptron;
Convolution + max − pooling: The operations of convolution and pooling; CF: Collaborative filtering; CNN: Convolution neural networks; f_k: The activation function of the kth layer perceptron; f_n^u: The activation function; $l(\cdot)$: The loss function; MAE: Mean absolute error; MF: Matrix factorization; MLP: Multi-layer perceptron; *Mu*: The QoS matrix of the Top-*k* neighbors of the user *u*; NDL: Neighborhood-aware deep learning; $N(i)$: The set of neighbor web service; $N(u)$: The set of neighbor users; PCC: Pearson's correlation coefficient; QoS: Quality of service; *R*: The user service matrix; RMSE: Root mean square error; $\overline{R}_u$: The average QoS value of the services that are evaluated by user *u*; $R_{u,j}$: The QoS value of user *u* for service *j*; Top-$k(u)$: The set of Top-*k* similar users to the current user *u*; VSS: Vector space similarity; V_u: The QoS vector of the user *u*; w_k: The weight matrix of the kth layer perceptron

Acknowledgements

This work is supported by the National Science Foundation of China (No. 61872002) and the Natural Science Foundation of Anhui Province of China (No. 1808085MF197). Yiwen Zhang is the corresponding author of this paper.

Authors' contributions

YJ and KW carried out the study and drafted the manuscript. YZ conceived the idea and participated in the design of the algorithm. YJ and KW performed the experiment and results analysis. YZ and YY participated in the technical discussion and helped perform the data analysis. All authors read and approved the final manuscript.

Availability of data and materials

The QoS data used to support the findings of this study can be accessed publicly in the website https://wsdream.github.io/dataset/wsdream_dataset1.html.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Management, Hefei University, Hefei, 230601, China. [2]School of Computer Science and Technology, Anhui University, Hefei, 230601, China.

Received: 17 April 2019 Accepted: 22 July 2019
Published online: 04 September 2019

References

1. Y. Guo, S. Wang, K. S. Wong, et al., Skyline service selection approach based on QoS prediction. Int. J. Web. Grid. Serv. **13**(4), 425–447 (2017)
2. Y. Zhang, Z. Zheng, M. R. Lyu. Wsexpress: A Qos-aware search engine for web services. IEEE International Conference on Web Services (ICWS) (IEEE, 2010), pp. 91–98
3. Y. Zhang, G. Cui, S. Zhao, et al., IFOA4WSC: a quick and effective algorithm for QoS-aware servicecomposition. IJWGS. **12**(1), 81–108 (2016)
4. Y. Zhang, G. Cui, S. Deng, et al., Efficient query of quality correlation for service composition. IEEE Trans. Serv. Comput. (2018). https://doi.org/doi: 10.1109/TSC.2018.2830773
5. L. Shao, J. Zhang, Y. Wei, et al., in *IEEE International Conference on Web Services (ICWS)*. Personalized Qos prediction for web services via collaborative filtering (IEEE, 2007), pp. 439–446
6. Y. Wu, C. Yan, L. Liu, et al., An adaptive multilevel indexing method for disaster service discovery. IEEE Trans. Comput. **64**(9), 2447–2459 (2014)
7. L. Liu, N. Antonopoulos, M. Zheng, et al., A socioecological model for advanced service discovery in machine-to-machine communication networks. ACM Trans. Embed. Comput. Syst. (TECS). **15**(2), 38 (2016)
8. J. Cui, H. Zhou, H. Zhong, et al., Akser: attribute-based keyword search with efficient revocation in cloud computing. Inf. Sci. **423**, 343–352 (2018)
9. J. Cui, H. Zhou, Y. Xu, et al., OOABKS: Online/offline attribute-based encryption for keyword search in mobile cloud. Inf. Sci. **489**, 63–77 (2019)
10. Y. Zhang, K. Wang, Q. He, et al., Covering-based web service quality prediction via neighborhood-aware matrix factorization. IEEE Trans. Serv. Comput. https://doi.org/10.1109/TSC.2019.2891517
11. Q. Wu, M. Zhou, Q. Zhu, Y. Xia, VCG auction-based dynamic pricing for multigranularity service composition. IEEE Trans. Autom. Sci. Eng. **15**(2), 796–805 (2018)
12. Y. Xia, M. Zhou, X. Luo, Q. Zhu, J. Li, Y. Huang, Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds. IEEE Trans. Autom. Sci. Eng. **12**(1), 162–170 (2015)
13. Z. Zheng, H. Ma, M. R. Lyu, et al., Collaborative web service Qos prediction via neighborhood integrated matrix factorization. IEEE Trans. Serv. Comput. **6**(3), 289–299 (2013)
14. M. Tang, Z. Zheng, G. Kang, et al., Collaborative web service quality prediction via exploiting matrix factorization and network map[J]. IEEE Transactions on Network and Service Management. **13**(1), 126–137 (2016)
15. G. Hinton, L. Deng, D. Yu, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Sig. Process. Mag. **29**(6), 82–97 (2012)
16. C. Szegedy, V. Vanhoucke, S. Ioffe, et al., in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Rethinking the inception architecture for computer vision, (2016), pp. 2818–2826
17. R. Collobert, J. Weston, in *Proceedings of the 25th International Conference on Machine Learning*. A unified architecture for natural language processing: deep neural networks with multi-task learning (ACM, 2008), pp. 160–167
18. R. Salakhutdinov, A. Mnih, G. Hinton, in *Proceedings of the 24th International Conference on Machine Learning*. Restricted Boltzmann machines for collaborative filtering (ACM, 2007), pp. 791–798
19. S. Li, J. Kawale, Y. Fu, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. Deep collaborative filtering via marginalized denoising auto-encoder (ACM, 2015), pp. 811–820
20. S. Sedhain, A. K. Menon, S. Sanner, et al. Autorec: Autoencoders meet collaborative filtering[C]. Proceedings of the 24th International Conference on World Wide Web. (ACM, 2015), pp. 111–112
21. F. Strub, J. Mary, Collaborative filtering with stacked denoising autoencoders and sparse inputs. NIPS. Workshop. Mach. Learn. eCommerce., 1–8 (2015)
22. G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering. IEEE. Internet. Comput. **1**, 76–80 (2003)
23. M. Wang, L. Shi, L. Liu, et al., Hybrid recommendation-based quality of service prediction for sensor services. Int. J. Distrib. Sensor Netw. **14**(5), 1550147718774012 (2018)
24. J. S. Breese, D. Heckerman, C. Kadie, in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Empirical analysis of predictive algorithms for collaborative filtering (Morgan Kaufmann Publishers Inc, 1998), pp. 43–52
25. B. Sarwar, G. Karypis, J. Konstan, et al., in *Proceedings of the 10th International Conference on World Wide Web*. Item-based collaborative filtering recommendation algorithms (ACM, 2001), pp. 285–295
26. M. R. McLaughlin, J. L. Herlocker, in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. A collaborative filtering algorithm and evaluation metric that accurately model the user experience (ACM, 2004), pp. 329–336
27. R. R. Liu, C. X. Jia, T. Zhou, et al., Personal recommendation via modified collaborative filtering. Phys. A Stat. Mech. Appl. **388**(4), 462–468 (2009)
28. R. Zhu, X. Zhang, X. Liu, et al., ERDT: Energy-efficient reliable decision transmission for intelligent cooperative spectrum sensing in industrial IoT. IEEE Access. **3**, 2366–2378 (2015)
29. Z. Zheng, H. Ma, M. R. Lyu, et al., in *IEEE International Conference on Web Services (ICWS)*. Wsrec: A collaborative filtering-based web service recommender system (IEEE, 2009), pp. 437–444
30. Y. Jiang, J. Liu, M. Tang, et al., in *IEEE International Conference on Web Services (ICWS)*. An effective web service recommendation method based on personalized collaborative filtering (IEEE, 2011), pp. 211–218
31. J. Wu, L. Chen, Y. Feng, et al., Predicting quality of service for selection by neighborhood-based collaborative filtering. IEEE Trans. Syst. Man. Cybern. Syst. **43**(2), 428–439 (2012)
32. X. Chen, Z. Zheng, Q. Yu, et al., Web service recommendation via exploiting location and QoS information. IEEE Trans. Parallel. Distrib. Syst. **25**(7), 1913–1924 (2014)
33. W. Lo, J. Yin, S. Deng, et al., in *IEEE International Conference Web Services (ICWS)*. Collaborative web service Qos prediction with location-based regularization (IEEE, 2012), pp. 464–471
34. M. Tang, Y. Jiang, J. Liu, et al., in *IEEE International Conference Web Services (ICWS)*. Location-aware collaborative filtering for QoS-based service recommendation (IEEE, 2012), pp. 202–209
35. Y. Xu, J. Yin, W. Lo, et al., in *International Conference on Web Information Systems Engineering*. Personalized location-aware QoS prediction for web services using probabilistic matrix factorization (Springer, Berlin, 2013), pp. 229–242
36. G. B. Huang, L. Chen, C. K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural. Netw. **17**(4), 879–892 (2006)
37. Z. Zheng, Y. Zhang, Lyu M.R., Investigating QoS of real-world web services. IEEE Trans. Serv. Comput. **7**(1), 32–39 (2014)
38. Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems. Computer. **8**, 30–37 (2009)
39. D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization. Nature. **401**(6755), 788 (1999)
40. A. Mnih, R. R. Salakhutdinov, in *Proceedings of Advances in Neural Information Processing Systems*. Probabilistic matrix factorization, (2008), pp. 1257–1264

Publisher's Note