EFMLP: A Novel Model for Web Service QoS Prediction

Kailing Ye[1,2], Huiqun Yu[1,3(✉)], Guisheng Fan[1], and Liqiong Chen[4]

[1] Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
yhq@ecust.edu.cn
[2] Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201112, China
[3] Shanghai Engineering Research Center of Smart Energy, Shanghai, China
[4] Department of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai 201418, China

Abstract. With the emergence of service-oriented architecture, quality of service (QoS) has become a crucial factor in describing the non-functional characteristics of Web services. In the real world, the user only requests limited Web services, the QoS record of Web services is sparsity. In this paper, we propose an approach named factorization machine and multi-layer perceptron model based on embedding technology (EFMLP) to solve the problem of sparsity and high dimension. First, the input data will be sent to embedding layer to reduce the data dimension. Then, the embedded feature vector will send to the factorization machine. After that, the first-order and second-order weights of the factorization machine are used as the initial weights of the first layer of the multi-layer perceptron. And the multi-layer perceptron is trained to adjust the weights. Finally, 1,974,675 pieces of data from an open dataset is used as experiment data to validate the model, and the result shows that our EFMLP model can predict QoS value accurately on the client side.

Keywords: Embedding model · Factorization machine · Neural network · QoS prediction

1 Introduction

A mount of Web services has been provided in the world. How to choose a web service which satisfies the user has been an urgent problem. Usually, when users access the network, a number of services with the same function can be provided by multiple service providers. Choosing a best Web service to provide users with a Web service recommendation system can greatly improve the user's Internet experience.

In real world, it's also hard to obtain the QoS value by calling the corresponding service from the client side. First, not all Web services are free, some Web services require a fee. Second, the number of services is huge. It will be a huge overhead in

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 350, pp. 375–385, 2021.
https://doi.org/10.1007/978-3-030-67540-0_22

time when each user requests each web service. Finally, most users are not experts, they do not have the experience and skills to measure Web services, which results in the measurement results not being credible.

In brief, the main contributions of this paper are as follows:

First, we propose the EFMLP model. The factorization machine and the multi-layer perceptron based on embedding model are combined to predict the QoS value.
Second, a real-world dataset is used to verify the effectiveness of the EFMLP model. Experimental results show that the method is more accurate than traditional collaborative filtering and matrix decomposition methods.

The rest part of this paper is organized as follows. The second section describes the missing QoS prediction problem and gives the framework of the EFMLP model. Section 3 introduces the EFMLP model to predict personalized QoS values. Section 4 discusses and analyzes the experimental results. Section 5 contains related work, and Sect. 6 summarizes the full paper.

2 The Overview of EFMLP Model

In this section, the prediction problem of missing QoS value is first described in Sect. 2.1. Then we introduce the framework of EFMLP model in Sect. 2.2.

2.1 Problem Description

Our aim is using the historical QoS information to accurately predict the missing QoS value. The existing data usually contains a user-item matrix, where $U = \{u_1, u_2 u_3, \ldots . u_i, \}$ is the user set, containing i users, $S = \{s_1, s_2 s_3, \ldots . s_j, \}$ is a collection of j Web services. v_{ij} represents the QoS value of the service j observed by the service from user i. Therefore, we construct an $i \times j$ user-service sparse matrix $M \in R^{i \times j}$. In this user-service matrix, each item v_{ij} represents a QoS attribute value. Figure 1 shows a tiny example. Figure 1(a) shows the existed QoS attribute values in the user-service matrix, which represented by set H. When the number of users and services is large, the sparsity of the matrix will be huge. Figure 1(b) shows the complete QoS value after prediction, which is represented by set M.

	S_1	S_2	S_3	S_4	S_5
U_1	5.982		0.228		
U_2		0.262		0.652	
U_3	0.854		0.649		0.115

(a) User-Service Matrix

	S_1	S_2	S_3	S_4	S_5
U_1	5.982	5.394	0.228	0.453	0.527
U_2	2.134	0.262	0.328	0.652	0.554
U_3	0.854	0.352	0.649	0.238	0.115

(b) Predicted User-Service Matrix

Fig. 1. A tiny example for missing QoS prediction

The missing QoS value can be represented by the set $P = M - H$. The predicted QoS value is to obtain the set P by using the existing set H. The EFMLP model is based

on the factorization machine model and the neural network, which is trained through the existing data, finally predict the missing QoS. Then a complete and accurate QoS value set M is obtained.

2.2 The Framework of EFMLP Model

In this section, we show the framework of EFMLP model. As shown in Fig. 2, the EFMLP model training is divided into two stages. In the first stage, the input data is converted into feature vectors by one-hot encoding, and then the embedding layer is used for dimensionality reduction and obtain information existing between input vectors. In the second stage, the output of the embedding layer is sent to the factorization machine for training, and the first-order weights and second-order weights of the factorization machine are used as the initial weights of the first layer of the multi-layer perceptron. Then train the multi-layer perceptron.

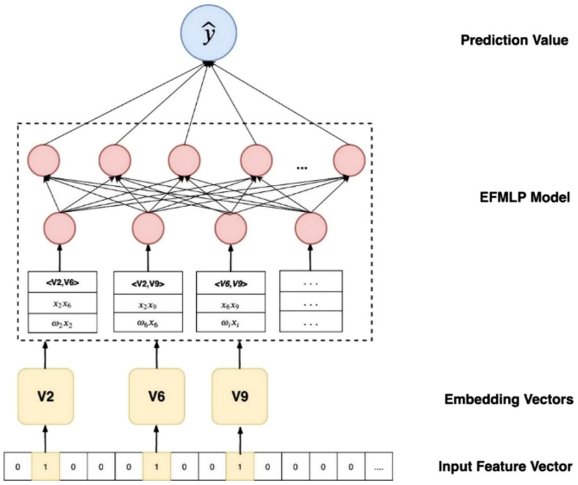

Fig. 2. The framework of EFMLP model

The following we will introduce the components of the model in detail.

(1) Obtain the feature vector. The input data includes user id and service id, both of them are integer types. One-hot encoding is applied to user id and service id respectively.
(2) Embedding vector. The Embedding layer compresses the input feature vectors into low-dimension, dense real-valued vectors, as to keep the feature information and ease the training pressure of the model in the next stage.
(3) EFMLP model. The factorization machine is used to model the embedding vector of the user id and the embedding vector of the service id, thus obtaining the interactive relationship between the user id and the service id. The first and second order coefficients of the factorization machine are used to initialize the first layer parameters of the multilayer perceptron to train the multilayer perceptron.

(4) Obtain the predicted value. After using all historical data to train EFMLP model, the model parameters will be determined. The model will output the predicted QoS value using the new data.

3 Details of EFMLP Model

The prediction process of EFMLP model can be divided into extracting embedding features phase, sending the embedding feature vectors to factorization machine phase. Finally, train the multilayer perceptron. The model is implemented with Pytorch1.3.1. We train the model on a workstation with Ubuntu18.04 LTS, Intel Xeon 8 cores processor and 32 GB memory.

3.1 Extracting Embedding Features

The input data is the user id and service id, and the data type is integer. First, the user id and service id will encode with one-hot layer, one-hot uses a bit status register to encode a state. The total number of item types is represented by a bit value of 1 and all other bit values of 0. For example, there are three users, u_1, u_2, and u_3. Figure 3 is a toy example of one-hot encoding in three users.

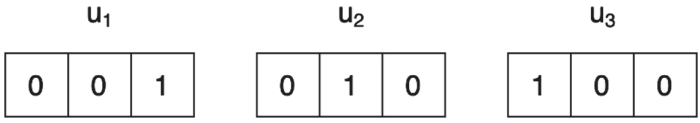

Fig. 3. The toy example of user id one-hot encoding

After obtaining the one-hot encoding of the user id and service id, the one-hot encoded vector is used as the input of the fully connected layer to output the embedding vector of the user id and service id. The role of the embedding layer is to map the original data from the source space to the target space and retain its structure. The data dimension sent to the model is kept at a low dimension, to speed up the model training process.

3.2 The Core of EFMLP Model

First, the factorization machine part of the EFMLP model is used for training, after that the factorization machine model trained weight is used as the initial weight of the first layer of the multi-layer perceptron to train the multi-layer perceptron. Finally, the model is combined with the factorization machine model to predict the missing QoS value.

The one-hot encoded user id and service id are sent to the embedding layer, and the embedding layer outputs low-dimension embedding features, then the embedding user id and service id are sent to the factorization machine. The expression of the factorization machine here is:

$$y_{FM}(x) = \omega_0 + \sum_{i=1}^{N} w_i x_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \langle V_i, V_j \rangle x_i x_j \tag{1}$$

Here, we use W_0^i to represent the weight value of the i-th layer of the MLP in the EFMLP model. W_0^i is initialized by the deviation term w_{i} and the vector $\mathrm{v_i}$ (for example, $W_0^i[0]$ is initialized by w_i, $W_0^i[1]$ initialized by v_i^1, $W_0^i[2]$ initialized by v_i^2). $z_i \in R^{K+1}$ is the parameter vector of the i-th field in the factorization machine.

$$z_i = W_0^i x[start_i : end_i] = \left(w_i, v_i^1, v_i^2, v_i^3, \ldots, v_i^K\right) \tag{2}$$

Where $w_0 \in R$ is a global scalar parameter, and n is the total number in all fields.

$$z = (w_0, z_1, z_2, \ldots z_i, \ldots, z_n) \tag{3}$$

Here $W_1 \in R^{M \times j}$, $b_1 \in R^M$ and $z \in R^J$.

We choose the sigmoid function as the activation function of the multilayer perceptron, where $sigmoid(x) = \frac{1}{1+e^{-x}}$, $W_1 \in R^{M \times J}$, $b_1 \in R^M$ and $z \in R^J$. Here are:

$$l_1 = sigmoid(W_1 z + b_1) \tag{4}$$

The predicted QoS value is:

$$\hat{y} = W_2 l_1 + b_2 \tag{5}$$

Where $W_2 \in R^{L \times M}$, $b_2 \in \mathbb{R}^L$ and $l_1 \in R^M$.

In the EFMLP model, the factorization machine part can establish a model for data expression in the latent space, which is conducive for further learning. In order to measure the effect of the EFMLP model, a loss function must be used to evaluate the error between the predicted value and the original value. Therefore, the loss function of the EFMLP model is defined as follows:

$$\min_{\theta} L\left(y, \hat{y}\right) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} \left(y_{ij} - \hat{y}_{ij}\right)^2 \tag{6}$$

I_{ij} is an indicator function. When I_{ij} is 1, it indicates that user u_i invoked the web service v_j, otherwise not invoked. The optimization of the loss function refers to minimizing the sum of squared errors. The initial weight of the factorization machine is learned through stochastic gradient descent (SGD). Using the chain rule in back propagation, the weights in the factorization machine are effectively updated. The weight update formula in the factorization machine is:

$$\frac{\partial L(y,\hat{y})}{\partial W_0^i} = \frac{\partial L(y,\hat{y})}{\partial z_i} \frac{\partial z_i}{\partial W_0^i} = \frac{\partial L(y,\hat{y})}{\partial z_i} x[start_i : end_i] \tag{7}$$

$$W_0^i \leftarrow W_0^i - \eta \cdot \frac{\partial L(y,\hat{y})}{\partial z_i} x[start_i : end_i]. \tag{8}$$

4 Experiment

In this chapter, we conduct a series of experiments on the EFMLP model using a real dataset, validate the effectiveness of the EFMLP model, compare with 7 methods, and analyze the results.

4.1 Description of Dataset

WS-Dream is a client side QoS dataset publicly released on the Internet. It contains 339 users and 5825 Web services. Table 1 is the statistical data of WS-Dream dataset.

Table 1. Statistics of WS-dream dataset

Statistics	Value
Number of service users	339
Number of web services	5825
Number of web services invocations	1,974,675
Range of response time	1–20 s
Range of throughput	1–1000 kbps

As mentioned above, there are many attributes of QoS. Although we focus on response time and throughput in this experiment, the EFMLP model can be easily apply to other attributes.

4.2 Metrics

In this experiment, we use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure the error between the predicted QoS value and the real QoS value.

MAE is given by:

$$MAE = \frac{\sum_{ij}|\hat{p}_{ij} - p_{ij}|}{N}, \tag{9}$$

and RMSE is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{ij}(\hat{r}_{ij} - r_{ij})^2}{N}}. \tag{10}$$

4.3 Performance Comparison

In order to verify the performance of EFMLP model on predicting the missed client's QoS value, the following methods are used as baselines.

(1) UMEAN (User Average). Calculate the average QoS value of a service that a user has called to predict the QoS value of a service has not been called.
(2) IMEAN (item average). This method calculates the average QoS of a Web service that has been called by other users as the QoS value of a user who has not called this service.
(3) UPCC (user-based collaborative filtering method using Pearson correlation coefficient) [9]. This method predicts the QoS value by looking for users with high similarity.

(4) IPCC (Project-based collaborative filtering method using Pearson correlation coefficient) [17]. This method is widely used by companies in industry, such as Amazon. In this paper, we look for similar Web services (items) to predict QoS values.
(5) UIPCC [5]. Combine user-based collaborative filtering with item-based collaborative filtering to predict QoS values based on similar users and similar Web services.
(6) PMF (Probability Matrix Factorization) [18]. This method was proposed by Salakhutdinov and Minh. QoS prediction is performed by using a user-item matrix through probability matrix decomposition.
(7) NIMF (Neighbor Synthesis Matrix Factorization). This method first finds a series of similar users for the current user, aggregates the information of these similar users and the available QoS values, and applies the concept of user-collaboration to make QoS predictions.

In this experiment, the sparsity of the matrix is controlled by removing part of the QoS entities from the dataset and filling with 0 to simulate the real situation of client service invocation. For example, a 20% matrix sparsity means that 20% of the user-service matrix will be removed. For the EFMLP model, 80% of the QoS records are used as the training set, and 20% of the QoS records are used as the test set. The greater the sparsity of the matrix, the smaller the amount of data can be used as the training set. The results of QoS missing value prediction are shown in Table 2 and Table 3. The EFMLP model is more accurate than the seven existing prediction methods in both response time and throughput QoS attributes.

Table 2. QoS prediction accuracy comparison (Response Time)

QoS properties	Methods	Matrix sparsity = 90%		Matrix sparsity = 10%	
		MAE	RMSE	MAE	RMSE
Response time	UMEAN	0.8776	1.8542	0.8728	1.8522
	IMEAN	0.6892	1.6416	0.6770	1.5256
	UPCC	0.5561	1.3092	0.4010	1.0851
	IPCC	0.5962	1.3423	0.3476	1.0280
	UIPCC	0.5836	1.3298	0.3447	1.0178
	PMF	0.4865	1.3130	0.3738	1.0514
	NIMF	0.4782	1.2914	0.3674	1.0397
	EFMLP	**0.4274**	**1.1987**	**0.2786**	**0.9730**

4.4 Impact of Matrix Sparsity

For QoS prediction, matrix sparsity is a factor that affects the accuracy of prediction. The lower sparsity means that more user-service entity information is available. In order

Table 3. QoS prediction accuracy comparison (Throughput)

QoS properties	Methods	Matrix sparsity = 90%		Matrix sparsity = 10%	
		MAE	RMSE	MAE	RMSE
Throughput	UMEAN	53.8810	110.3452	53.6560	109.7987
	IMEAN	26.8735	64.8038	26.5214	63.2873
	UPCC	22.6036	54.5220	13.5389	38.8792
	IPCC	26.1821	60.2451	16.7591	42.9217
	UIPCC	22.3639	54.3285	13.1787	38.1428
	PMF	15.8972	48.1782	11.9224	35.8713
	NIMF	15.1392	47.0476	11.7735	35.5175
	EFMLP	**14.7826**	**42.0129**	**8.2694**	**28.1795**

to study the influence of matrix sparsity, the matrix sparsity was adjusted from 10% to 90%. At the same time, we set the embedding size with 80 for response time and 400 for throughout, the number of hidden layer units of multilayer perceptron was 128. The following are the experimental results:

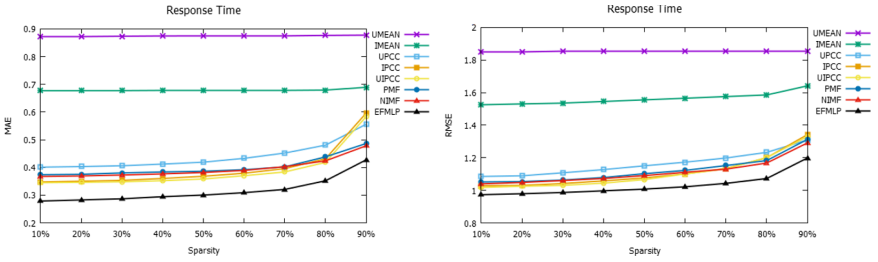

Fig. 4. Impact of matrix sparsity (QoS property is response time)

The experiment results in Fig. 4 and Fig. 5 show that when the matrix sparsity is set from 90% to 50%, the prediction error shows a trend of decline. When the sparsity of

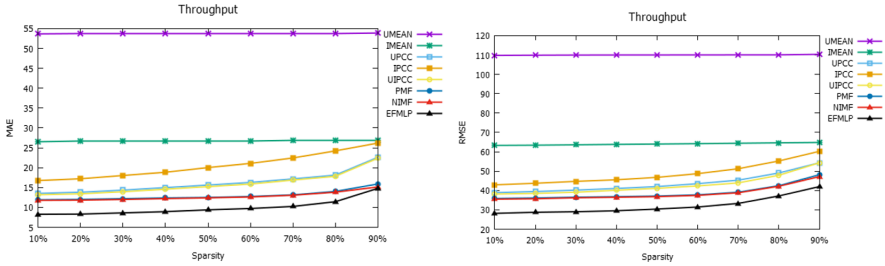

Fig. 5. Impact of matrix sparsity (QoS property is throughput)

the matrix is between 50% and 10%, the prediction error continues to decrease, but the rate of decline becomes slow. The lower the sparsity of the matrix, the more amount of information known as QoS. Therefore, the amount of QoS information will affect the prediction accuracy. When the QoS information reaches a certain level (such as 50%), the effect of QoS information on the prediction accuracy will not be obvious.

4.5 Impact of Dimensionality

The dimension of the embedding layer is a hyper parameter. The dimension determines the number of feature vectors extracted from the user id and service id. In the experiment, the dimension of the embedding layer for response time is set from 50 to 100 with an interval of 10, and the dimension of throughput is set from 200 to 800 with an interval of 100. The matrix sparsity is 90% in the experiment. Figure 6 and 7 are the experiment results of response time and throughput respectively.

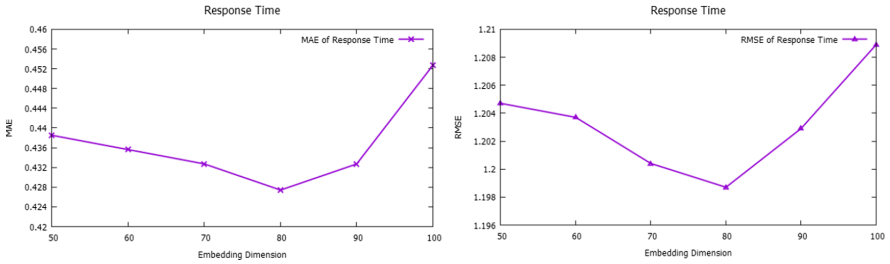

Fig. 6. Impact of embedding vector dimensionality (QoS property is response time)

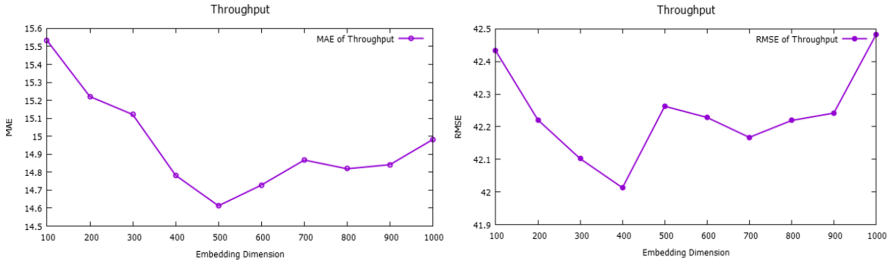

Fig. 7. Impact of embedding vector dimensionality (QoS property is throughput)

The experimental results show that different embedding dimension value satisfy the different QoS attributes. Figure 6 shows that when the dimension changes from 50 to 80, the error of the response time shows a downward trend, which means that the prediction accuracy is improving; when the dimension changes from 80 to 100, the prediction accuracy shows an upward trend. For the throughput, when we choose the MAE as evaluation standard, the dimension changes from 100 to 1000, the trend of both the MAE and RMSE are roughly the same. It is obviously that when the dimension of the RMSE changes from 400 to 500, the accuracy has decreased, indicating that the model is

overfitting when the dimension is set to 500. The reason why the embedding dimension of the response time is smaller than the throughput due to the range of the response time is relatively small.

5 Related Work

In this section, we introduce some existing methods about the QoS value prediction research. Most of these methods have been applied to service-oriented architecture scenarios, here are a lot of investigations. Ardagna and Pernici [15] use five QoS attributes (for example, execution time, availability, price, data quality, and reputation) to facilitate flexible configuration during the adaptive service composition process. By studying common QoS attributes and QoS attributes of designated domains, Alrifai and Risse [16] proposed an effective service composition method. This article focuses on the prediction method of the QoS value that the client can observe. In the study of client QoS missing value prediction, the most extensive and in-depth research is collaborative filtering.

The memory-based, model-based and hybrid collaborative filtering methods have two disadvantages:

(1). Missing QoS value, the relevant research fields have been lacking real datasets as experimental research materials.
(2). The dataset has the characteristics of high dimension and huge sparsity.

6 Conclusion

According to the existing QoS attribute data has the features of huge sparsity and high dimensions, we propose a model called EFMLP. Embedding is used to reduce the dimensionality of sparse data, and a factorization machine and multi-layer perceptron are used to build a model for QoS missing value prediction and verify it on a public dataset with lots of records. The results show that the EFMLP model has better accuracy.

In the future, the model can be used to predict missing values of other QoS attributes on the client side (such as fault tolerance, compatibility, reliability.). In addition, other client side factors can be incorporated into the model, to improve model performance.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 61702334, 61772200), the Project Supported by Shanghai Natural Science Foundation (No. 17ZR1406900, 17ZR1429700) and the Planning Project of Shanghai Institute of Higher Education (No. GJEL18135).

References

1. Ding, S., Li, Y., Wu, D., Zhang, Y., Yang, S.: Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and RIMA model. Decis. Support Syst. **107**, 103–115 (2018)

2. Zeng, L., Benatallah, B., Ngu, A.H., Dumas, M., Kalagnanam, J., Chang, H.: QoS-aware middleware for Web services composition. IEEE Trans. Softw. Eng. **30**(5), 311–327 (2004)
3. Wu, Y., Xie, F., Chen, L., Chen, C., Zheng, Z.: An embedding based factorization machine approach for web service QoS prediction. In: Maximilien, M., Vallecillo, A., Wang, J., Oriol, M. (eds.) ICSOC 2017. LNCS, vol. 10601, pp. 272–286. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69035-3_19
4. Su, K., Ma, L., Xiao, B., Zhang, H.: Web service QoS prediction by neighbor information combined non-negative matrix factorization. J. Intell. Fuzzy Syst. **30**(6), 3593–3604 (2016)
5. Zheng, Z., Ma, H., Lyu, M.R., King, I.: QoS-aware Web service recommendation by collaborative filtering. IEEE Trans. Serv. Comput. **4**(2), 140–152 (2011)
6. Zheng, Z., Ma, H., Lyu, M.R., King, I.: Collaborative web service QoS prediction via neighborhood integrated matrix factorization. IEEE Trans. Serv. Comput. **6**(3), 289–299 (2013)
7. Rendle, S.: Factorization machines. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 995–1000. IEEE (2010)
8. Harris, D., Harris, S.: Digital design and computer architecture, 2nd edn. Morgan Kaufmann, San Francisco (2012). p. 129. ISBN 978-0-12-394424-5, (2012-08-07)
9. Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., Mei, H.: Personalized QoS prediction for web services via collaborative filtering. In: IEEE International Conference on Web Services (ICWS 2007), Salt Lake City, UT, pp. 439–446 (2007)
10. Athman, B., et al.: A service computing manifesto: the next 10 years. Commun. ACM **60**, 4 (2017)
11. Kim, M., Oh, B., Jung, J., Lee, K.H.: Outlierrobust web service selection based on a probabilistic QoS model. Int. J. Web Grid Serv. **12**(2), 162–181 (2016)
12. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet Comput. **7**(1), 76–80 (2003)
13. Zhu, J., He, P., Zheng, Z., Lyu, M.R.: Online QoS prediction for runtime service adaptation via adaptive matrix factorization. IEEE Trans. Parallel Distrib. Syst. **28**(10), 2911–2924 (2017)
14. Liu, A., et al.: Differential private collaborative Web services QoS prediction. World Wide Web **22**(6), 2697–2720 (2018). https://doi.org/10.1007/s11280-018-0544-7
15. Ardagna, D., Pernici, B.: Adaptive service composition in flexible processes. IEEE Trans. Softw. Eng. **33**(6), 369–384 (2007)
16. Alrifai, M., Risse, T.: Combining global optimization with local selection for efficient QoS-aware service composition. In: Proceedings of 18th International Conference on World Wide Web (WWW 2009), pp. 881–890 (2009)
17. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: GroupLens: an open architecture for collaborative filtering of Netnews. In: Proceedings of ACM Conference Computer Supported Cooperative Work, pp. 175–186 (1994)
18. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization. In: NIPS, vol. 1, pp. 2–1 (2007)
19. Zhang, W., Du, T., Wang, J.: Deep learning over multi-field categorical data – a case study on user response prediction. In: ECIR (2016)