



HAL
open science

Location-Based Service Recommendation for Cold-Start in Mobile Edge Computing

Mengshan Yu, Guisheng Fan, Huiqun Yu, Liang Chen

► **To cite this version:**

Mengshan Yu, Guisheng Fan, Huiqun Yu, Liang Chen. Location-Based Service Recommendation for Cold-Start in Mobile Edge Computing. 17th IFIP International Conference on Network and Parallel Computing (NPC), Sep 2020, Zhengzhou, China. pp.297-309, 10.1007/978-3-030-79478-1_26 . hal-03768745

HAL Id: hal-03768745

<https://inria.hal.science/hal-03768745v1>

Submitted on 4 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Location-based service recommendation for cold-start in mobile edge computing

Mengshan Yu^{1,2}, Guisheng Fan¹✉, Huiqun Yu¹, and Liang Chen¹

¹ Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai, China

{gsfan,yhq}@ecust.edu.cn

² Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai, China

Abstract. With the rapid development of the 5G and Internet of Things (IoT), mobile edge computing has gained considerable popularity in academic and industrial field, which provides physical resources closer to end users. Service recommendation in such a distributed environment is a hot issue. However, the cold-start problem for service recommendation in mobile edge computing is still urgent to be solved. In this paper, we propose a service recommendation method based on collaborative filtering (CF) and user location, by comprehensively considering the characteristic of services at the edge and mobility of users. In detail, we first synthesize the service characteristics of each dimension through multidimensional weighting method. We further introduce the idea of Inverse_CF_Rec to the traditional CF and then predict the lost QoS value to solve the problem of sparse data. Finally, a recommendation algorithm based on predicted QoS value and user geographic location is proposed to recommend appropriate services to users. The experimental results show that our multidimensional inverse similarity recommendation algorithm based on collaborative filtering(MDICF) outperforms Inverse_CF_Rec in terms of the accuracy of recommendation.

Keywords: Mobile edge computing · QoS · Service recommendation · Multiple dimension.

1 Introduction

In recent years, the edge computing paradigm has been widely adopted as a good complement to cloud computing, which brings the processing to the edge of the network [1]. And the service recommendation problem is a hot issue in edge computing. For example, when a mobile user send a service request, edge servers that have enough resources to process the information will intercept and respond to the user request [2]. And the recommendation system becomes very important if there are more than one edge server which can process user requests. Usually, service recommendation algorithms consider quality of service (QoS) as the decision variable [3]. So, how to calculate the QoS value of the service more accurately is one of our key research points.

According to our survey, most service recommendation methods are based on massive historical data [3, 4, 6, 9, 15], such as the memory-based collaborative

filtering (MCF) [3]. However, the mobility of users may lead to the fluctuation of QoS value in mobile edge computing [5]. When a mobile user moves from the wireless coverage range of one edge server to that of another edge server, the historical QoS value will become invalid and the QoS value on the new edge server is null [6]. In the field of service recommendation in edge computing, the data density is generally sparse, that is, there exists the cold start problem. And our proposed method is based on collaborative filtering. However, it is difficult to calculate the similarities between object user and other users when the object user moves to a new edge sever. To solve it, the similarities between the object user and other users who have data on the object user’s historical edge server and new edge server is considered. We also calculate the similarities among users on object user’s new edge server. Because of the sparse data matrix, calculating the similarities among users directly may lead to imprecise results. So authors in Ref [7] introduce the sociological theory, namely an enemy’s enemy is a friend, which represents that the opposite user of a opposite user is likely to be a similar user of the object user. However, their work only consider user requirements on a single dimension. In actual decision, considers on multiple dimensions are more common.

According to aforementioned issues, we propose a multi-dimensional inverse similarity recommendation algorithm based on collaborative filtering (MDICF).

(1) Considering the multi-dimensionality of user needs and service performance at the edge, we adopt a multi-dimensional linear weighting method to take into account the needs of every dimension. For each mobile user, our weight coefficients in different dimensions are dynamically adjusted for a more comprehensive understanding of user needs.

(2) Aiming at the mobility of users and sparse data of services at the edge, we use inverse similarity to predict users QoS value based on collaborative filtering and calculate the distance between the user and the service by obtaining their latitude and longitude. A MDICF method is proposed to recommend appropriate services to users within a reasonable range.

The rest of this paper is organized as follows. Section 2 shows the related work. Section 3 details our prediction recommendation method. Section 4 introduces the design and experimental results of our experiments and Section 5 summarizes our methods and prospects for future work.

2 Related work

Because of high interpretability, collaborative filtering (CF) is widely used in various recommendation systems. CF is mainly divided into two categories: Model-based and Memory-based. Memory-based CF is also called Neighborhood-based CF, which can be divided into user-based similarity CF and content-based similarity CF. The basic idea of that method is to find similar user sets or items based on user or content similarity, and recommend to target users based on ratings of similar users or items. The similarity among users or content is mainly calculated by using Pearsons Correlation Coefficient and Cosine Similarity. In addition, hybrid recommendation algorithm contains a mixture of matrix decomposition and deep learning. This method incorporates different recommendation algorithms, so it has a powerful feature extraction ability and can automatically retrieves a deeper representation of data.

In recent years, there have been many researches on CF in the field of recommendation system. The author of [12] takes into account the number of times a user calls a service when calculating the historical QoS similarity generated when a user calls the same service. They improve authenticity and persuasion of data by using the average of all QoS. The collaborative filtering model based

on matrix factorization [12, 14] has also been widely used in Web service recommendation. As a potential linear model, the performance of matrix factorization is not good enough when capturing complex interactions. With the rapid development of service-oriented computing and cloud computing, many machine learning and deep learning models are applied to recommendation systems in combination with traditional collaborative filtering. The authors in [11] propose a hybrid web recommendation algorithm based on matrix factorization and deep learning which combines collaborative filtering and text content. The invocation between mashups and services and their functions are integrated into a deep neural network to describe the complex interactions between services. Article [12] proposed a model combined a compressed interaction network and DNN. On the level of vectorization, the feature interaction was generated in an explicit way. But they have serious bottlenecks in running efficiency and memory consumption. Such deep learning models can solve many problems, but there are serious bottlenecks in terms of operating efficiency and memory consumption.

In addition, the exiting recommendation algorithm in edge computing rarely take into account the location characteristics of users and services. In fact, the QoS of services in edge computing are usually closely related to the location relationship between users and services. Now that smart phones and other devices that support GPS navigation are widely used, they can collect more and more location information, which also provides reliable conditions for our location-based recommendations. There have been many researches on location-based service recommendation before. They usually introduce location information as a new dimension to recommendation algorithms. The article [16] proposed a location-aware personalized CF web service recommendation method which select similar users or services of target users in users and web services at two locations.

3 Motivation

In this section, we illustrate our method through a scenario. Vehicle-road collaboration is an important part of smart transportation. As a good complement to cloud computing, edge computing performs most of the calculations near the road at the edge. As shown in Fig. 1.

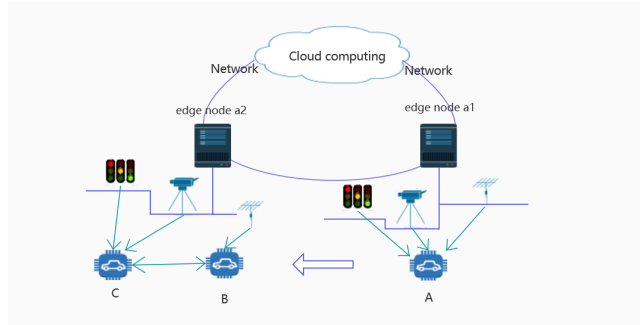


Fig. 1. Service recommendation in mobile edge computing

It integrates a local map system, traffic signal information, nearby moving targets and multiple sensor interfaces [8] at the edge node near the road. A self-driving vehicle *A* drives away from the coverage of edge node *a1* and enters

coverage of edge node $a2$. It needs to obtain the required services in $a2$. And we need to consider the problem of spare data when historical information of A is little at edge node $a2$. In addition, since there are many kinds of integrated at edge nodes near the road, we need to consider multidimensional feature such as response time, throughput and so on. Therefore, we propose a multi-dimensional weighted inverse similarity method to predict unknown QoS values and make recommendations based on geographic location.

4 Multidimensional inverse similarity recommendation algorithm based on CF

According to the aforementioned analysis, we consider using collaboration filtering with reversed user similarity to predict the unknown QoS values, and the problem of multi-dimension for the QoS value is also considered. Then we recommend services to object user based on locations of user and services. Specifically, we adopt linear weighted sum method to integrate QoS in different dimensions into one comprehensive QoS. Then the problem is transformed into a linear programming problem. Our method is mainly divided into two steps:

(1) QoS prediction. This step includes the calculation of user similarity, the determination of users with similar behaviors and similar users and prediction of QoS. We predict unknown QoS through the calculation of user similarity and similar credibility.

(2) Service recommendation. We can get all services within a certain distance from the user through our calculating. We recommend services for the object user based on the predicted QoS in the previous step.

The overview of our approach is shown in Fig. 2 as follow.

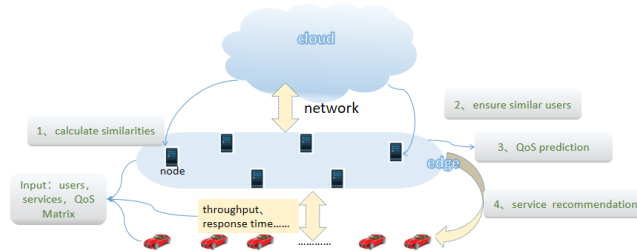


Fig. 2. Overview of MDICF

4.1 QoS prediction

Linear weighted sum method. In mobile edge computing, the focus of services at different locations may be different. Thus, we consider adopting linear weighted sum method to integrate different dimension of QoS in one service with one dimension. This is a method commonly used in the multi-objective optimization problem which keeps the original data in each dimension. The linear weighted sum method is formalized in Equation(1):

$$\begin{cases} q = \sum_{i=1}^{S-q} (p_i * x_i) \\ \sum_{i=1}^{S-q} p_i = 1 \end{cases} \quad (1)$$

wherein $p_i \in [0,1]$, x_i indicates the weight coefficient and QoS value of each dimension in one service, respectively.

In this paper, we consider the response time and throughput of services at edge. So, we transform the Equation(1) into Equation(2):

$$\begin{cases} q_{(i,k)} = \alpha * tp_{(i,k)} + \beta * rt_{(i,k)} \\ \alpha + \beta = 1 \end{cases} \quad (2)$$

wherein α and β are weight coefficients between 0 and 1, $tp_{(i,k)}$ and $rt_{(i,k)}$ respectively represent the quality of service in terms of response time and throughput when $service_k$ is called by $user_i$, $q_{(i,k)}$ represents the weighted quality of service when $service_k$ is called by $user_i$, which is used when we calculate the similarities later. The calculation of α and β will be introduced later.

Calculate the user similarity. Pearson correlation coefficient (PCC) is widely used in the similarity calculation because of its simplicity and efficiency. Thus, we consider using PCC to calculate the similarities of users. We use $U = \{u_1, u_2, u_3, \dots, u_m\}$ to represent the user set that call for services, $ES = \{es_1, es_2, es_3, \dots, es_n\}$ to represent the set of services at edge. In our method, $q_{(i,k)}$ represents the QoS of edge $service_k$ when $user_i$ calls for edge $service_k$ and \bar{q}_i represents the average QoS of $user_i$. $sim_{(i,j)}$ is the similarity between $user_i$ and $user_j$ which is calculated as Equation (3):

$$sim_{(i,j)} = \frac{\sum_{es_k \in IN} (q_{(i,k)} - \bar{q}_i) * (q_{(j,k)} - \bar{q}_j)}{\sqrt{\sum_{es_k \in IN} (q_{(i,k)} - \bar{q}_i)^2} * \sqrt{\sum_{es_k \in IN} (q_{(j,k)} - \bar{q}_j)^2}} \quad (3)$$

In this equation, IN is the set of edge services that both $user_i$ and $user_j$ have called for. If $user_i$ and $user_j$ have not called for the same edge service, IN is equal to $Null$. We calculated the similarity among all users in the user set. In addition, in the user similarity matrix, $sim_{(i,j)}$ is empty when IN is $Null$ (i.e. $sim_{(i,j)} = Null$ if $IN = Null$).

Calculate similar users. We have obtained the similarity matrix of users in previous sections. Now we define and calculate several key variables. $threshold_P$ ($-1 \leq threshold_P \leq 1$) denotes pre-defined similarity threshold and is adjusted in our experiment. $potential_similar$ and $potential_opposite$ represent potential similar user set and potential opposite user set, respectively. $rel_matrix_{opposite}$ and $rel_matrix_{similar}$ denote the reliability matrix of opposite users and similar users, respectively. The calculations about them will be introduced in detail later. $rel_matrix_{opposite}$ is a user set opposite to the object user. As we can see in Equation(4), users whose similarity with the object user is less than the threshold are defined as opposite users.

$$opposite_users_{object_user} = \{user_i | sim_{(object_user,i)} \leq threshold_P\} \quad (4)$$

We determine potential similar users based on the theory that “the enemy is a possible friend” [18]. The inaccurate results will be obtained if we directly define the opposite users of $opposite_users$ as the similar users. So we quote a reliability calculation as Equation(5) [7]. In this equation, $user_j$ is the

opposite user of object user and the $user_k$ is the opposite user of $user_j$, that is $user_j \in opposite_user_{object_user}$ and $user_k \in opposite_user_j$.

$$reliability(object_user, user_k) = sim_{(object_user,j)} * sim_{(j,k)} \quad (5)$$

For each object user, we establish a credibility matrix for the potentially similar user set called rel_matrix_i . The similar credibility constraints between each user in the matrix and object user are as shown in Equation(6).

$$reliability_{similar} = \{k \in potential_similar, \\ reliability(object_user, user_k) \geq -threshold_P\} \quad (6)$$

In addition, we know that the friend of the enemy may be an enemy. Therefore, we use a opposite credibility constraint to determine the opposite user show in Equation(7).

$$reliability_{opposite} = \{k \in potential_opposite, \\ reliability(object_user, user_l) \leq threshold_P\} \quad (7)$$

wherein $user_j$ is the opposite user of $object_user$ and $user_l$ is the similar user of $user_j$, that is, $user_j \in opposite_users_{object_user}$ and $user_l \in similar_user_j$.

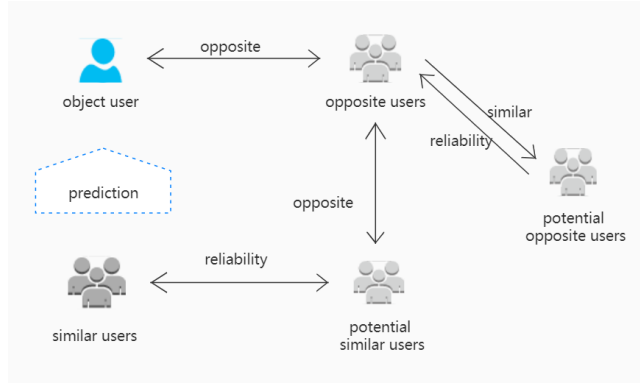


Fig. 3. QoS prediction in mobile edge computing

We then add the opposite users obtained by above constraints to the opposite user set. The relationship between users is shown in Fig. 3. We get $opposite_users$ by calculating user similarities, and the potential similar users of opposite users can be obtained. Then, the similar users of opposite users with reliability constraint $reliability_{opposite}$ can be added into the original opposite user set. Next we get potential similar users with calculating opposite users of users in $opposite_users$. At last, we obtain similar user set $similar_users$ of object user through the constraint of similar reliability($reliability_{similar}$).

QoS prediction. We have obtained the similar users and similar reliability between all similar users and the object user. Now we regard the reliability as

the probability of user similarity. So we can predict the missing QoS when the object user is calling service that has not been called before by Equation(8) and $user_k \in similar_users$. In this equation, $q_{(k,i)}$ and $predict_q_{(object_user,i)}$ denote the QoS when $user_k$ calls $service_i$ and the predicted QoS when the object user is calling $service_i$, respectively.

$$predicted_q_{(object_user,i)} = \frac{\sum_{user_k \in similar_users} reliability(object_user, user_k) * q_{(k,i)}}{\sum_{user_k \in similar_users} reliability(object_user, user_k)} \quad (8)$$

4.2 Service recommendation

It cannot be ignored in service recommendation that the wireless coverage distance of the edge server is limited. To solve this problem, we need to calculate the distance between the object user and the service to ensure that the recommended service is within the wireless coverage area. As shown in Fig.1 in Section3, we need to find a service on Edge node $a2$ that meets the needs of users.

In this paper, we use a general geographic distance calculation method, Haversine equation, to calculate the distances between users and services according to attitude and longitude. Because it uses a sine function, it can maintain enough significant figures even at small distances.

We use (lon_1, lat_1) and (lon_2, lat_2) to represent the longitude and latitude of the object user and the service, respectively. $dis(u, s_i)$ represents the distance between the object user and the i -th service. The distance calculation is shown in Equation(9) as follow.

$$\begin{cases} dis_{(u,s_i)} = 2R \cdot arctan\left(\frac{\sqrt{hav(\theta)}}{\sqrt{1-hav(\theta)}}\right) \\ hav(\theta) = \sin^2\left(\frac{lat_1-lat_2}{2}\right) + \cos lon_1 \cos lon_2 \sin^2\left(\frac{lon_1-lon_2}{2}\right) \end{cases} \quad (9)$$

Wherein θ represents the center angle between the object user and the service, and R represents radius of earth. Here we take 6371km.

Based on the calculation of the above equation, we find the recommendable service set s in the range which is shown in Equation(10). We select the $top - K$ services from the predicted data to recommend to the object user.

$$s = \{s_1, s_2, \dots, s_c\}, dis_{(u,s_i)} \leq \lambda \quad (10)$$

4.3 Algorithm design

Our improved inverse user similarity collaborative filtering algorithm predicts the QoS value of the unknown service that the object user calls for. And the recommendation algorithm is based on geographic location. The implementation of our algorithms is shown as the pseudo-code of Algorithm 1.

Assuming that each QoS matrix includes m users and n service data, and all of them are in k dimensions. The analysis of our time complexity is as follow:

- (1) Linear weighted sum method: The time complexity of QoS linear weighting on k dimensions is $O(k * m * n)$.
- (2) Calculate the user similarity: We calculate the user similarity based on the existing user and service data offline. Here, it is $O(1)$. If he is a new user, the time complexity is $O(m)$.

(3) Calculate similar users: According to the user similarity matrix, we judge the reliability of indirect friends between users. The maximum number of opposite users is m , and the maximum number of opposite users opposite users is m . In other words, the time complexity of finding indirect similar users is $O(m^2)$ under the worst possibility.

(4) Distance calculation: The distance between object user and each service need to be calculated, so the complexity is $O(n)$.

(5) Service recommendation: We maximized the number of indirect friends as m and the number of edge services in the recommended range as n , so the worst time complexity of our QoS prediction and service recommendation is $O(m * n)$. Since indirect similar users judgment and geographical location calculation are conducted at the same time, we assume $l = \text{Max}(m^2, n)$. Based on the above analysis, the time complexity of our algorithm is $O(l + m * n)$.

Algorithm 1 MDICF

Input: U ; E_S ; tp_Matrix ; rt_matrix ;
Output: recommendation service
Begin:

- 1: Initialization: $predict_q = \phi$; $similar_users = \phi$; $threshold_P$
- 2: **for** ($q_{(i,k)}$ in QoS Matrix) **do**:
- 3: $q_{(i,k)} = \alpha * tp_{(i,k)} + \beta * rt_{(i,k)}$
- 4: **end for**
- 5: **for** ($user_i$ in U) **do**:
- 6: **for** ($user_j$ in U) **do**:
- 7: calculate $sim_{(i,j)}$ by Equation(3)
- 8: **end for**
- 9: **end for**
- 10: calculate $opposite_users$ by Equation(4)
- 11: **for** ($user_k$ in $potential_similar$) **do**:
- 12: calculate $reliability_similar$ by Equation(6)
- 13: update $similar_users < - user_k$
- 14: **end for**
- 15: **for** ($user_q$ in $potential_opposite$) **do**:
- 16: calculate $reliability_opposite$ by Equation(7)
- 17: update $opposite_users < - user_q$
- 18: **end for**
- 19: repeat 11-18
- 20: **for** (es_i in E_S) **do**:
- 21: **if** $q_{(object_user,i)} = \phi$ **then**
- 22: calculate $predicted_q_{(object_user,i)}$ by Equation(8)
- 23: **end if**
- 24: **end for**
- 25: **for** (es_i in E_S) **do**:
- 26: calculate $dis(object_user, i)$
- 27: **if** $dis \leq \lambda$ **then**:
- 28: add es_i into $potential_service$
- 29: **end if**
- 30: **end for**
- 31: **return** Top-3 services with $predicted_q$ in $potential_service$

5 Experiment

In this section, we first verify the rationality of our experiment, and then we compare our algorithm with other related algorithms. The results of experiments is calculated by running an average of 100 times.

5.1 Experiment environment and datasets

Datasets. In our experiment, we adopt the public dataset WSDream of the Chinese University of Hong Kong. The dataset describes real-world QoS evaluation results including both response time and throughput values, obtained from 339 users on 5825 web services as a user-service matrix. We use the real response time and throughput in the dataset to simulate the response time and throughput of the service that the user calls for at the edge. In addition, we calculate the distance between the user and service through the latitude and longitude of them in the dataset. Since we mainly consider the case where the data matrix is sparse, we randomly delete some quality items based on the original data set and obtain data matrixes with different data density d . In this paper, the data density d of the nine experimental configurations varies from 5% to 50%, namely 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40% and 50%.

5.2 Experimental design

Our algorithm is an improvement based on the traditional collaborative filtering method. Compared with traditional service recommendation, the mobility of edge users and the immediacy of demand are the important points that we need to consider. At the same time, the demand for computing power and throughput cannot be ignored. Therefore, we comprehensively consider the throughput, response time of the service and mobility of users and recommend services at the edge to end users in this paper.

Parametric design. At the beginning of the experiment, we weight the quality of service in two dimensions. We use the linear weighting sum method in Ref. [17], where α and β are adaptive for different object users. Generally, the most important demand has the maximum weight coefficient value according to the user preferences. Since the α and β for each test user in this experiment are different, no quantitative explanation will be made here.

The threshold of similar user judgement and reliability judgement refer to the design in Ref. [7], tentatively set $threshold_P = -0.6$. We also performed it during the experiment. After verification, we find the results of our algorithm under this threshold is relatively good.

Since this experiment is based on the WSDream dataset, the geographical distances between users and services are larger than those in our scenario, so λ should also be relatively large. Through calculation, the distance between the user and service in the data set ranges from 0 to 50000. In a real scenario at the edge, the value of λ needs to be reset according to the real coverage. Finally, we recommend the $top - 3$ services to target users within this geographic range.

5.3 Experiment results

We use Mean Absolute Error (MAE) to measure the error between our predicted value and the true value, and compare it with other methods.

In this paper, we adopt Mean Absolute Error (MAE) to measure the error between our predicted value and the true value.

$$MAE = \frac{1}{c} \cdot \sum_{i=1}^c |predicted_q_i - q_i| \quad (11)$$

In Equation(11), c represents the number of prediction services, $predict_q_i$ represents the predicted value, and q_i represents the true value. The smaller the MAE value, the higher the prediction accuracy.

When the data density d is 5%, the number of edge services es is 1000, and the number of different users u is varied from 100 to 300, the QoS value that predicted by our proposed method is shown in Fig. 4(a).

When the data density d is 5%, the number of users u is 100, and the number of edge services s is varied from 1000 to 3000, the QoS value that predicted by our proposed method is shown in Fig. 4(b).

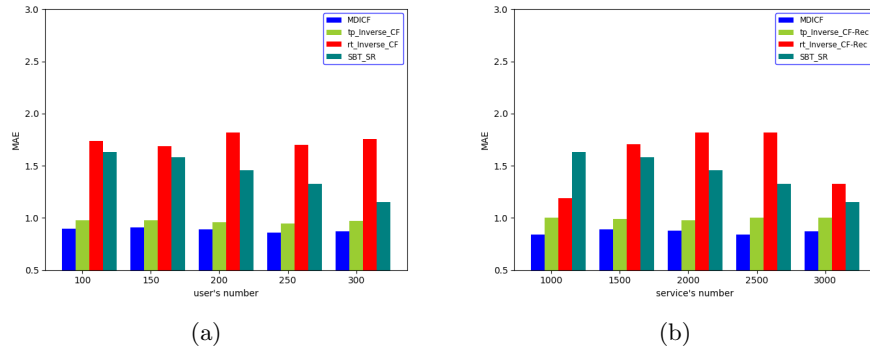


Fig. 4. Recommendation accuracy

As can be seen in the figure, compared with other methods, our method performs significantly better on the MAE evaluation index. Among them, the lowest prediction accuracy is the *Inverse_CF_Rec* method when the response time is the evaluation standard. This method only considers the response time of the service as the evaluation index. However, the response time is affected by the distance besides the the service application at edge. Therefore, it is difficult to make an accurate recommendation based on only one evaluation index. Followed by SBT-SR, this method only considers the theory that “the enemys enemy is a friend”, but ignores “the enemys friend may be an enemy”. It take less consideration about some potential similar users and potential opposite users which may lead to reduce prediction accuracy. The method our proposed has the best performing which comprehensively considers the service response time, throughput and potential similar users. At last, we consider the geographical location of users to improve the feasibility of recommendation.

In addition, we also made comparisons in different data densities. As shown in Fig. 5, when the number of edge services s is 1000, the number of users u is 100 and data density d is varied from 5% to 50%, the QoS value that predicted by our proposed method is significantly smaller than that predicted by other method in the case of data density.

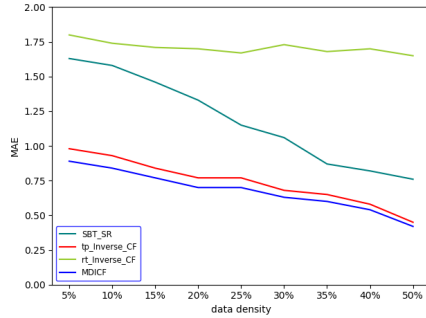


Fig.5. MAE of different data density

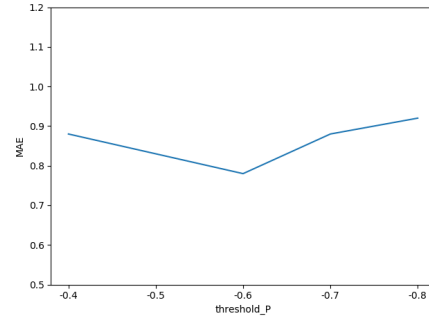


Fig.6. Different threshold on MAE

In this experiment, we selected object users in the test dataset, and verified the rationality of taking $threshold_P$ equal to -0.6 as the similar threshold when the data density is 5% for 100 users and 1000 services. As shown in Fig. 6, we set as below: $threshold_P = -0.4, -0.5, -0.6, -0.7, -0.8$. Regardless of the accuracy to the decimal place, the selected threshold performs well in QoS prediction, and we can consider our value to be reasonable.

6 Conclusion

In this paper, we combine quality indicators of multiple dimensions into one comprehensive QoS value and take the geographical location of end users into consideration. These handling strategies make our method better fit the characteristics of services at edge and end users. To solve the cold start problem for the service recommendation system in edge computing, we propose a multi-dimensional inverse user similarity recommendation algorithm based on collaborative filtering (MDICF). The experimental results demonstrate the effectiveness of our algorithm for service.

In the future, we plan to improve our prediction model through deep learning when facing sparse data. In addition, how to combine the predicted QoS value with geographic location information more closely is also the focus of our future work.

Acknowledgements

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. Wang, S., Zhao, Y., Xu, J., Yuan, J., Hsu, C.H.: Edge server placement in mobile edge computing. *Journal of Parallel and Distributed Computing*, vol. 127, 160–168 (2019)

2. Ahmed, A., Ahmed, E.: A survey on mobile edge computing. In: 2016 10th IEEE International Conference on Intelligent Systems and Control, pp. 1–8 (2016)
3. Zheng, Z., Ma, H., Lyu, M. R., King, I.: Qos-aware web service recommendation by collaborative filtering. *IEEE Transactions on Services Computing*, Vol. 4, No. 2, 140–152 (2011)
4. Kang, G., Tang, M., Liu, J., Liu, X. , Cao, B.: Diversifying web service recommendation results via exploring service usage history. *IEEE Transactions on Services Computing*, Vol. 9, No. 4, 566–579 (2016)
5. Arshad, R., Elsayy, H., Sorour, S., Alnaffouri, T. Y., Alouini, M. S.: Handover management in dense cellular networks: A stochastic geometry approach. In: 2016 IEEE International Conference on Communications, Kuala Lumpur, pp. 1–7 (2016).
6. Wang, S., Zhao, Y., Huang, L., Xu, J., Hsu, C. H.: QoS prediction for service recommendations in mobile edge computing. *Journal of Parallel and Distributed Computing*. Vol. 127 , 134–144 (2019)
7. Zhou, Y., Tang, Z., Qi, L., Zhang, X., Dou, W., Wan, S.: Intelligent service recommendation for cold-start problems in edge computing. *IEEE Access*, Vol. 7, 46637–46645 (2019)
8. Ekiz, N., Salih, T., Kucukoner, S., Fidanboyly, K.: An overview of handoff techniques in cellular networks. *Int. J. Inf. Technol.*, Vol. 2 132–136 (2005)
9. Herlocker, J. L.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, Vol. 5, 287–310 (2002)
10. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, 734–749 (2005)
11. Xiong, R., Wang, J., Zhang, N., Ma, Y.: Deep hybrid collaborative filtering for web service recommendation. *Expert Systems with Application*, Vol. 110, 191–205 (2018)
12. Li, S., Wen, J., Luo, F., Gao, M., Zeng, J., Dong, Z.Y.: A new qos-aware web service recommendation system based on contextual feature recognition at server-side. *IEEE Transactions on Network and Service Management*, 1–1 (2017)
13. Dhas, Y.and Jeyanthi, P.:Track health care using location based QoS web service recommendation system, In: 2017 9th International Conference on Advanced Computing, Chennai, pp. 35–40 (2017)
14. Lian, J., Zhou, X., Zhang, F., Chen, Z., Xie, X., Sun, G.: Xdeepfm: combining explicit and implicit feature interactions for recommender systems. In: 2018 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, London, pp. 1754–1763 (2018)
15. Li, K., Long, Y., Lan, H., Yin, W., Pengju, M., Chuanbin, L., Yujia Z.: A personalized QoS prediction approach for cps service recommendation based on reputation and location-aware collaborative filtering. *Sensors*, Vol. 18, No. 5, (2018)
16. Liu, J., Tang, M., Zheng, Z., Liu, X., Lyu, S.: Location-aware and personalized collaborative filtering for web service recommendation. *IEEE Transactions on Services Computing*, Vol. 9, No. 5, 686–699 (2016)
17. Xie, Y., Zhu, Y., Wang, Y., Cheng, Y., Xu, R., Sani, A. S., Yuan, D., Yang, Y.: A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud-edge environment. *Future Gener. Comput. Syst.*, Vol. 97, 361–378 (2019)
18. Lianyong Q., Xuyun Z., Yiping W., Yuming Z.: A social balance theory-based service recommendation approach. In:2015 9th Asia-Pacific Services Computing Conference, Bangkok, THAILAND, pp. 48–60 (2015)