QoS Prediction for Smart Service Management and Recommendation based on the Location of Mobile Users

Lei-lei Shi[1,4], Lu Liu[2], Liang Jiang[1,4], Rongbo Zhu[3], John Panneerselvam[2]
[1]School of Computer Science and Telecommunication Engineering, Jiangsu University, China
[2] School of Informatics, University of Leicester, UK
[3]School of Computer Science, South-Central University for Nationalities, China
[4]Jiangsu Key Laboratory of Security Tech. for Industrial Cyberspace, Jiangsu University, China
Corresponding author: Lu Liu, Email: l.liu@leicester.ac.uk

Abstract—Quality of Service (QoS) directly reflects the degree to which services offered by providers satisfy the non-functional requirements of users. QoS information is not usually available as a priori to providers when recommending services to user queries, this creates uncertainty in offering right services to right queries. Recent researches in service recommendation and management mainly address the issues of sparse data prediction and user personalized recommendation. Recommendation systems require smart strategies of recommending and managing services in accordance with the user queries. Predicting the QoS requirements of user queries before recommending the services can potentially aid in offering the most suitable services to users. This paper proposes a hybrid mobile service recommendation and management model based on semantic recommendation along with location-based quality preference analysis for emerging 5G mobile networks. The proposed model can effectively predict the QoS by exploiting previously invoked services to identify the best matching mobile services based on the similarity between users and services. Performance evaluation based on a published web services dataset demonstrates an enhanced prediction accuracy with an effective reduction in time overheads when compared to other related methods.

Index Terms—QoS Prediction, Service Recommendation and Management, Web Services, Collaborative Filtering

1 INTRODUCTION

Service recommendation and management is usually the concept of offering the best possible matching services to a given user's query. Given the ever-increasing number of mobile web services and the ever-growing size of service repositories, the development of efficient service recommendation and management algorithms has received considerable attention in recent years [28-30]. Collaborative filtering [1] is a kind of service recommendation and management algorithm that offers services based on a prior estimation of the Quality of Service (QoS) parameter attached to a given user's service requirements. Collaborative filtering works based on an assumption that users with similar interests often invoke similar services [2]. Some algorithms of collaborative filtering have addressed the data sparsity problem in service recommendation and management by predicting the missing QoS values, and exploit the predicted QoS to recommend neighbouring services for users. The QoS value required to invoke a potential web service for a given user is mainly calculated using a service recommendation and management method, based on analyzing the correlation between users characterizing similar interests and/or the correlation between similar services invoked by similar users [3].

Traditional collaborative filtering methods of service recommendation and management mostly exploit historical information of user invoked services to predict the QoS value. Despite their wider deployments, such traditional methods do not consider the latent characteristics, between users and their associated services, such as semantic information of services, network type, geographical location, etc. [4-9]. Such information significantly impacts the accuracy achieved in the service recommendation and management process. In addition, the collaborative filtering algorithms [14,22] suffer inaccuracies whilst predicting the service quality requirements under large-scale data sparsity. Data sparsity results when a user tries to invoke a given part of a large number of services. Sparsity of data usually has a significant influence on the accuracy and validity of the recommendation systems, especially under high sparsity. A QoS data matrix of a given service may include several null elements, therefore resolving data sparsity issues are absolutely crucial in order to improve the overall performance of the service recommendation and management systems.

Users are often unique with their expectation in terms of the type and quality of services they receive from the recommendation and management systems. In other words, a given user might like fewer services to be accurately matching the query or might like a diverse range of returning options. To this end, accuracy-focused recommendation and management systems recommend expected services by avoiding unpopular services, especially when some services lack QoS attributes. Diversity-focused recommendation and management systems [10-13] not only recommend popular services, but also recommend and manage services that are cold-blooded but can still meet user preferences. Thus, an efficient recommendation and management system should not only return popular and well-known services, but also other non-popular user-friendly services in order to provide consumers with more diverse options. Cold start is another issue prevailing in traditional recommendation systems when faced with more new users and new services. Memory-based recommendation systems are usually not capable of availing effective recommendations when new users or services enter the repository, due to the lack of sufficient historical information. Time-efficient web services usually characterize a large number of users, resulting in housing massive amount of data. Many recommendation methods, despite characterizing good accuracy and recommendation effect, are usually limited by

complex data calculation process which directly affects their recommendation speed. Therefore, a recommendation system should not only be accurate in offering matching services, but should also reduce the incurring computation and time overheads.

To this end, this paper critically evaluates the current state-of-the-art service recommendation and management methods [12,15,16] and further proposes a novel semantic association-based web services hybrid recommendation and management method with integrated location-oriented quality preference analysis for 5G and beyond networks, which can offer accurate services to user queries alongside reducing the incurring computation and time overheads. Experiments conducted based on a published web services dataset exhibits the efficiencies of the proposed method in terms of the prediction accuracy and reduction in time overheads. Major contributions of this paper are listed as follows.

1) A hybrid web services recommendation and management method named LPOR (Location and Preference Oriented Recommendation) that comprehensively considers QoS location information and user quality preference is proposed. The proposed method applies the Analytic Hierarchy Process (AHP) to extract the weight of a given user's preference of QoS. To exploit the user location information for enhancing the quality of recommendation, the proposed method firstly requests the autonomous system number (ASN) through the IP address of the users or services, and then exploits the user's country ID or service country ID along with ASN to perform area aggregation, and finally predicts the missing QoS value based on regional similarity.
2) The proposed hybrid service recommendation and management method incorporates the advantages of correlation-based and content-based service recommendation methods to resolve the cold start issues, along with achieving diversity whilst recommending and managing services. With this characteristic, the proposed method can not only find accurate matching services, but also can reduce the incurring computation and time overheads.
3) Extensive experiments are conducted based on real-world web services dataset to demonstrate the efficiencies of the proposed method, which exhibits that the proposed method outperforms traditional methods by achieving obvious time and cost improvements.

The rest of this paper is organized as follows. Section II critically reviews the existing works of QoS prediction for recommendation services. Section III describes our proposed QoS prediction method, and Section IV presents and discusses the experimental results. Section V concludes our paper along with outlining our future research directions.

2 Related Work

Given the increasing usage of web services in the recent years [3-6, 47-49], a wide range of works have been proposed to enhance the efficiencies of recommendation and management services [9].

Popular service recommendation and management methods include collaborative filtering (CF)[31-33], content-based, knowledge base-based recommendation and so on. It is common to witness the integration of two or more recommendations methods in an attempt to enhance the quality of the overall recommendation systems. Content-based web service recommendation uses service attributes to predict the web services that are previously invoked by similar users. Collaborative filtering method exploits the similarity among different users, believing that similar users often have similar interests in web services. Collaborative filtering is one of the widely used methods in the context of web service recommendation. Depending on the attributes used for prediction, collaborative filtering methods can be classified into item-based collaborative filtering, user-based collaborative filtering and model-based collaborative filtering accordingly. All of such methods rely on the historical correlation index between similar items, users, or services including user's score, user's attributes, and service characteristics and so on. A typical collaborative filtering method mostly relies on users' general attributes such as user's rating information and service list of previously triggered web services. This method has been widely used in well-known commercial websites, such as Netflix and Amazon [12-13]. Bias SVD [14] is a potential factor model using singular value decomposition (SVD), which exploits user and project bias factors. GM [15] is a greedy approach used for sorting items. CloudRank2 [16] is a ranking method of cloud services using different preference confidence levels. 2RHyRec [12] is a ranking oriented hybrid approach that combines collaborative filtering with potential factors. Model-based methods have used machine learning methods [34-36] such as clustering model [25][37-39], neural networks [17][40-42] and latent semantic model [1][37][43-46].

Considering QoS attributes can potentially improve the effectiveness of web service recommendation and service selection in different methods, particularly when the QoS indicators of different services characterize similar functions [18]. In most of such methods [19], QoS is mainly represented by QoS attribute values, with a presumption that QoS attributes (such as response time) are valid and are easy to obtain. However, most of the QoS attribute values are not easy to obtain, as they are affected by factors such as geographical location, time, network status (such as response time, network latency, availability, etc.). In fact, some of the QoS attributes might characterize severe inconsistency. Such inconsistencies might result due to the fact that providers offering QoS parameters that exceed the actual performance. QoS experiences of users are also not usually consistent with the promises of service providers due to their dynamic service environments. Furthermore, QoS values collected by the registry for all users representing the average performance of the service might also be inconsistent. QoS inconsistency has mainly been the issues of the traditional recommendation methods, since such methods do not consider matching the service semantics with the users' needs.

The content-based recommendation method uses the evaluation data and service attributes of a user's interaction with the previous services in order to evaluate the user's preferences for the content features of the services [20]. New

services are then recommended to the corresponding users by computing the similarity between service content and user characteristics. Various methods have been proposed for extracting service and user features by calculating the topic probability distributions based on topic models. The most common ones are probabilistic semantic analysis (PLSA) [1] and Hidden Dirichlet Distribution (LDA) [25]. PLSA is a general statistical model associated with test data. The core of PLSA is the Aspect Model [21], which is a hidden variable model. With PLSA, user preferences and service attributes can be measured by hidden variables, and it has been applied to automated recommendation [22] and collaborative filtering recommendation [25]. LDA assumes that the topic probability of a document is controlled by K hidden variables (that is, K beta parameters), which determines the probability distribution of topic based on the statistical results of words in the corpus. Beta and alpha determine the probability distribution of topics in a specific document, and topics determine the occurrence of words. LDA is an extension of PLSA's introduction of Dirichlet process as a service subject distribution. LDA has been applied to different fields, such as scientific topic discovery [24-27], information diffusion and text analysis [23]. One of the problems with recommendation systems is the cold start-up, which often occurs when the recommendation system attempts to recommend services without content features or interest features [39]. Content-based recommendation systems or hybrid recommendation systems can avoid this problem, since the relationship among service contents can be established by analyzing service description documents.

Existing works on service recommendation only discuss the relationship between neighborhood-based web services to predict the QoS attributes. Furthermore, user interest attributes have not been given sufficient emphasis on the existing state-of-the-art. Ignoring such user interest attributes might lead to the recommendations of services that do not characterize the capabilities of meeting the user requirements. Moreover, in extreme cases, it is almost impossible to make recommendations for new users and services. Due to such complexities of web service recommendation systems, this paper proposes a novel hybrid recommendation system by incorporating the features of semantic and association-based service recommendations along with location awareness.

3 HYBRID RECOMMENDATION − BASED QUALITY OF SERVICE PREDICTION METHOD

The method proposed in this paper uses the following definitions:

Definition 1 T_k=< $Q_{(k,1)}$, $Q_{(k,2)}$,... $Q_{(k,J)}$ > is a vector, which refers to all QoS data submitted by user u_k.

Definition 2 $Target_{Qw}$=< T_{w1}, T_{w2},... T_{wk}> is a vector that represents the weight of a user's preference for the QoS attribute of a Web Service.

Based on the above definition, the data set used in this article can be formatted into response time R_T and throughput T_P, as shown in Table I and Table II.

3.1 Web service recommendation based on QoS attribute preference

Different users have different preferences for QoS. In order to compute their preferences for web services, AHP is used to quantify the relative fuzzy weights. This relatively simple and effective estimation method can be used to infer the degree of user preferences for QoS attributes, which is measured by the weight function.

Analytic Hierarchy Process (AHP) appeared as early as the 1970s. It is a flexible, changeable and practical multi-criteria decision-making method for quantitative analysis of qualitative problems. It first decomposes the factors affecting decision-making according to their attributes, including multi-level attributes from top to bottom. The number of required hierarchical levels in the decomposition process is decided based on unique cases. After establishing the hierarchical model, we construct the comparison matrix of the factors, and further the comparison matrix is moderated to maintain consistency. Finally, the weight reflecting the relative importance of each level is calculated, and the relative weight of all the factors is calculated based on the corresponding ranks of the involved factors.

Besides AHP, the entropy method can also be used to calculate the weight. The entropy method considers the amount of information contained in an attribute depending on their respective entropy, which affects their resulting weight. An increasing amount of information will incur a significant influence on the evaluation results, further resulting in higher weight values. Therefore, the weight of a given QoS attribute in the entropy method is calculated based on the degree of confusion of service QoS attributes experienced by users. Higher values of QoS attributes usually incur greater impacts on the evaluation, resulting in higher weight values.

Although the entropy method can achieve the same function as AHP, the entropy method tends to be more objective while AHP is a subjective weight evaluation method and it is more flexible. Given this flexibility, this paper uses AHP to quantify the QoS weight. The hierarchy of web services is comprised of three layers, described as follows. First is the overall goal layer, which represents the overall goal; this paper represents the user's optimal QoS weight vector for web services, formally expressed as $TargetQ_w$. Second is the goal layer, which contains several layers of elements, representing the sub-goals involved in achieving the overall goal, formally expressed as $Q_{wi,j}$ scheme in Fig. 1. Finally, the layer represents a feasible solution to implement the previous level.

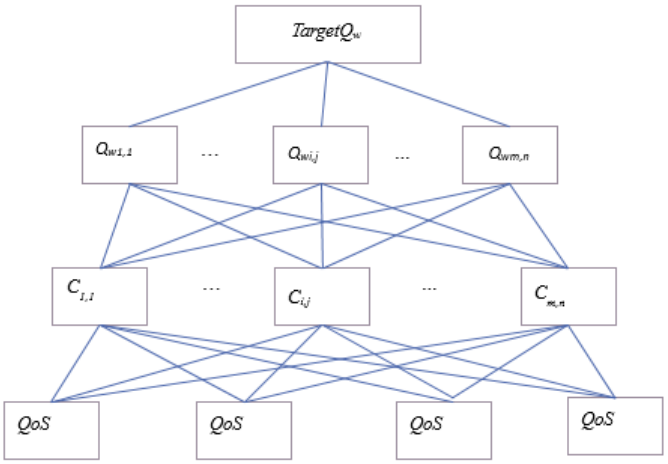

Fig.1. QoS weight calculation hierarchy graph

Table I Formatting QoS data <RT, TP>

<RT,TP>	s_1	s_2	s_3
u_1	<5.728,0.334>	<5.594,17.549>	<NaN,25.316>
u_2	<0.375,6.892>	<0.268,NaN>	<0.276,21.739>
u_3	<NaN,1.97>	<0.266,15.037 >	<0.276,21.739>

Table II Formatting QoS data RT

RT	s_1	s_2	s_3
u_1	5.728	5.594	NaN
u_2	0.375	0.268	0.276
u_3	NaN	0.266	0.276

After the hierarchical graph is established, a comparison matrix *C* of QoS attributes for each user's performance requirements for web services is constructed, in which each element represents the importance of the two-to-two comparison between the QoS attributes of the users for web services. Assuming that a given web service has *n* QoS attributes (usually *n* should not exceed 9 positive numbers), the form of matrix *C* can be represented as follow:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{bmatrix} \quad (1)$$

where c_{ij} denotes the importance of an attribute *i* of QoS relative to another attribute *j*. Matrix *C* is called as the comparison matrix. When a user does not define a QoS attribute, its weight value considered as 0 and fitted as a child node in the hierarchy graph. In this paper, the relative importance of QoS attributes is converted into nine corresponding levels, i.e. marked with 1-9, as shown in Table III.

Table III Classification of QoS attribute

Level	Description
1	The two attributes have the same importance
2	Between the level 1 and level 3.
3	The former attribute is slightly more important than the latter.
4	Between the level 3 and level 5.
5	The former attribute is clearly more important than the latter.
6	Between the level 5 and level 7.
7	The former attribute is much more important than the latter attribute.
8	Between the level 7 and level 9.
9	The former attribute is significantly important than the latter.

3.2 Location based recommendation for Web services

Existing event detection models hardly distilled the popular topics, which results in low quality of posts and users being discovered under popular topics. Therefore, topic filtering method [8] is essential for determining the importance of users under popular topics.

Definition 3 D_u is the virtual aggregation area for users.

Definition 4 *A* as all ASN sets: {a_1, a_2,... a_n}, *n* is the total number of ASN in the dataset, and the element a_i represents an ASN.

Definition 5 User u_i region aggregation returns a list of nearest neighbors in the form of <nearest neighbor user, weight>

$$\mathrm{sim}_{\mathrm{ui}} = \{(u_j, w_j) : u_i \wedge u_j \in a_i, u_i \wedge u_j \in countryID_i, u_i \wedge u_j \in D_i\} \quad (2)$$

Autonomous System (AS) is a set of one or more address prefixes used to specify a unified routing policy in IP networks. In general, AS is controlled by a single entity and network administrators. In the Border Gateway Protocol (BGP), each autonomous system has a unique and corresponding number in the global Internet, namely the Autonomous System Number (ASN). ASN is unique and ASN allocation is usually centralized. A topic with an ASN can be a single manageable network unit (such as a university, an individual enterprise etc.).

Regional aggregation uses the autonomous system number (ASN) and country ID corresponding to the IP address of the user and services to calculate the location proximity of the user and services. This step is based on the assumption that users or services with the same ASN or *countryID* characterize greater similarity. Users or services with same ASN or *countryID* are entered into a Neighbor list. If users or services do not characterize the same ASN or *countryID*, and then Top-K users with the high similarity of aggregation regions are selected for service recommendation.

Zone aggregation first takes advantage of the QoS attributes experienced by a user belonging to the same or similar ASN or *countryID*, and adds other users or services whose user ID u_i (or service s_j) is in the same AS to that user's nearest neighbor list.

In fact, not many users (or services) belong to the same ASN or *countryID*. We need to further calculate the remaining users (or services) those do not belong to the same ASN or *countryID*.

For a given user, the computation of the similarity weights of each ASN is calculated using the following formula:

$$w_i = \frac{|a_i|}{\sum |a_i|} \tag{3}$$

Formula $|a_i|$ denotes the number of the first ASN in which the user has used the Web Service IP, and $\sum |a_i|$ denotes the number of all Web Services used by that user.

For any two users, the region similarity before correction is calculated.

$$sim'(u,v) = \frac{\sum_1^m w_{ui} * w_{vi}}{\sqrt{\sum_1^m w_{ui}^2} * \sqrt{\sum_1^m w_{vi}^2}} \tag{4}$$

where, *sim* (*u,v*) represents the number of common ASN contained in user *u* and *v*.

However, a *countryID* can include multiple ASNs, since ASN is usually smaller than *countryID*. QoS attributes are closely related to the network environment since IP addresses of users under the same ASN generally share the same network environment. Thus, QoS values of two user experiences or services within the same ASN belong to the same country. Although similar users or services belong to the nearest neighbor list, they might have different weights. So the following formula is used to calculate the regional similarity *sim (U, V)*:

$$sim(u,v) = d_c + d_a + sim'(u,v) \tag{5}$$

where, d_a and d_c are adjustable corrections, representing the adjustments when *u* and *v* belong to the same ASN and *countryID*, respectively.

Based on the above calculation, we obtain the similarity between any user *sim(u,v)*. If *sim(u,v)* is larger than the previously set threshold of regional aggregation, user u and v enter the same area.

Now, we filtered out the list of neighbors for user *u*, and obtained the region similarity between users. The list of neighbors includes the same ASN user, the same country ID user, and the same region with the user *u*.

3.3 Location oriented and quality preference-based hybrid recommendation method for Web services

The traditional memory based collaborative filtering algorithm is often inefficient and characterize poor scalability. Through the above process of region aggregation, we obtain the list of neighbor users who are similar to the given user's region, and further obtain the region similarity between any two users. Neighbor users include users with same ASN, users with same *countryID*, and users in the same region of the user *u*, all characterizing similar QoS performances to a given service.

The following formula is used to calculate the predicted value of QoS:

$$Ru, j' = \frac{\sum_{j \in N} sim(u_i, u_j) * r_{u,j}}{\left|sim(u_i, u_j)\right|} \tag{6}$$

where $sim(u_i, u_j)$ is the regional similarity between user $_i$ and user *j*.

After region aggregation, regions are added by calculating the similarity between current users and users in each region. When the similarity is not satisfied, the threshold value is modified over time. Then the Web Services invoked by the users in a given region are recommended to the current users after being processed according to the predicted QoS values, thus reducing the computation time of the Web Services recommendation. At the same time, it is more reasonable to predict the QoS of Web Services based on the user region similarity aggregation, because users in the same region share the same network environment, infrastructure, and often have similar Web Service quality experience.

The QoS value of the service in the recommendation list is calculated by S_q.

$$S_q = TargetQ_w * T_k^T \tag{7}$$

The final list of services returned is usually a list of services that contain user area similarities and similar Web Service quality attribute preferences.

4 Experiments and Results

The experiments presented in this paper are conducted based on real-world QoS datasets to evaluate the effectiveness and accuracy of the proposed method.

4.1 Experimental environment

The experimental verification is carried out in the following environment:

1) Processor: Core i5 2.67Hz
2) Memory: 4GB
3) Operating system: Windows 7 Professional 64bit
4) Software tools: Python 3.4, Java, Matlab

4.2 Dataset

Experiments are conducted using real-world web services QoS dataset [4] comprising more than 170,000 QoS values from 339 users with 5,825 web services distributed in 73 countries, and reflect the user's real QoS experience values.

Since the original dataset does not contain WSDL files, we recrawled the WSDL files for each service using a Python code before the experiment. Some web services are not accessible in China, web services with expired maintenance are filtered out. Finally, a rating matrix of 339 users and 908 web services with more than 300,000 QoS values is constructed. Each *r_ (I, J*) matrix represents the rating of user *I* on the web service *J*.

All WSDL addresses are obtained from the dataset, and all WSDL files are crawled from the network; a total of 908 web service WSDL files are collected by crawling the latest WSDL files. Then, during the service content extraction, a service

description file is processed for the LDA model to calculate the document-topic probability distribution.

In order to accurately predict the QoS value, the experimental dataset is divided into two groups of different matrices, namely the training matrix and the test matrix. The elements in the training matrix are randomly selected from the dataset, and the remaining elements form the test matrices. In many real web services, users use only a part of the web services, it is impossible to use all the web services, and some indicators of the QoS value may also be lost. In order to match the real-world scenario, data density D of ($0<d<1$) has been set up in the experiment. For each sparse matrix, the training set accounts for 80% of the original data, and the test matrix is the remaining 20%. The elements of the training set are used to predict the missing QoS values in the test set. The parameter K represents the number of neighboring services.

The LDA implementation of the hybrid recommendation and management method for web services for association and semantics uses the Gibbs sampling technique [25-27, 37-39] for parameter estimation. For hyper-parameter settings, we use the same value as in [28] that is, $50/T$ (T is the number of hidden topics). All web service ratings use the same metrics based on service QoS values [39, 40] such as response time, throughput, and so on.

4.3 Baseline Approaches

We compare the efficiencies of our proposed method with the below popular methods: UPCC [4] is a user-based collaborative filtering method that uses Pearson correlation index to measure user similarity. IPCC [4] is an item-based collaborative filtering method that uses Pearson correlation index to measure the similarity between items. WSRec [4] is a QoS-aware hybrid Web Service recommendation and management method that uses a linear combination of different weights of UPCC and IPCC methods.

4.4 Evaluation Methods

4.4.1 Accuracy comparison

The prediction accuracy of all the methods has been evaluated with the incremental values of the density matrix ranging from 0.01 to 0.05. As shown in Figure 2 and Figure 3, the prediction accuracy of all the methods increases gradually with increasing values of the density matrix. As shown in Figure 2 and Figure 3, using the same parameter configuration (with different density values from 0.01 to 0.05, k = 50, = 50, = 0.1), we conducted 10 sets of iterations and average the results. Our proposed method exhibits a higher accuracy than other methods, as our proposed method considers the relationship between service relevance and semantic similarity of services. Accuracy is manifested in lower MAE and NMAE values.

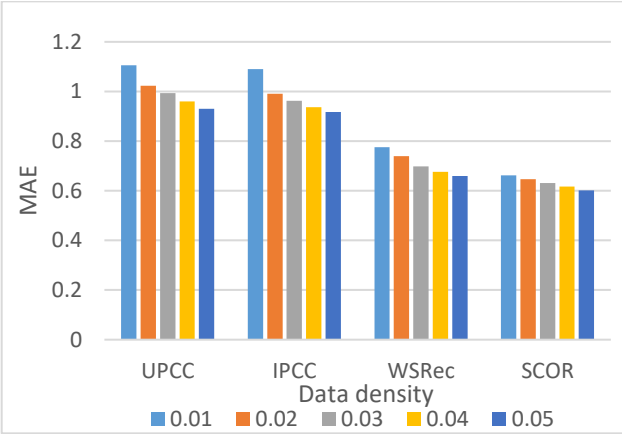

Fig. 2. The MAE value of each method

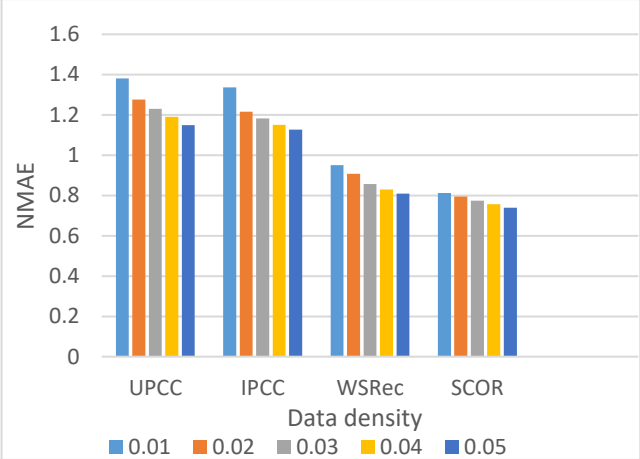

Fig.3. NMAE values of each method

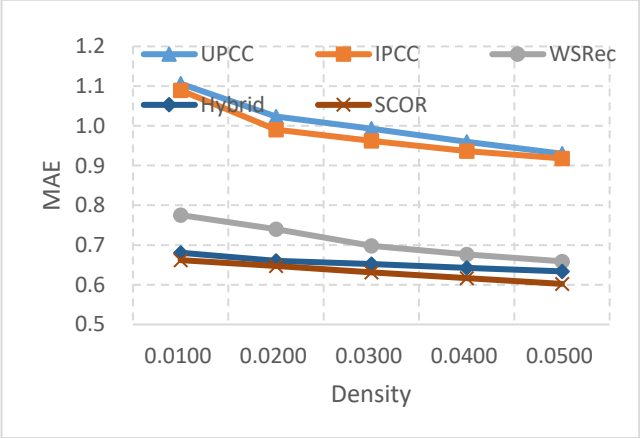

Fig. 4. MAE values affected by sparsity of each method

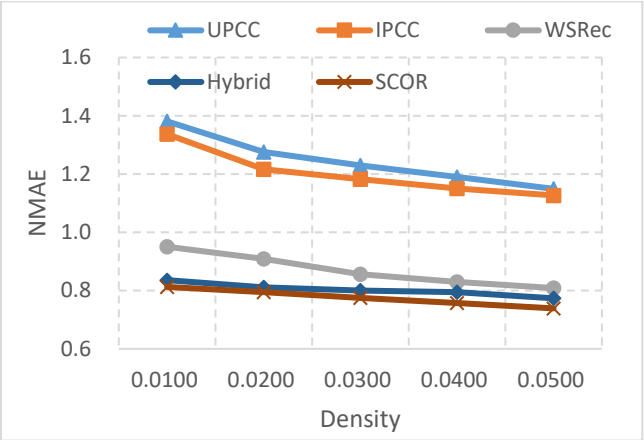

Fig. 5. NMAE values affected by the sparsity of each method

4.4.2 Sparsity effect

In reality, a smaller number of services is often invoked by users, so that the training matrix becomes highly sparse. Therefore, it is necessary to study the density of matrices. As shown in Figures 4 and 5, the prediction accuracy of the methods is enhanced when the matrix density increases from 0.01 to 0.05, witnessing corresponding lower values of MAE and NMAE. The experimental results show that prediction accuracy is enhanced with more abundancy in the data. Furthermore, it can be obviously observed that our proposed method outperforms all the other methods in terms of prediction accuracy.

4.4.3 Recommendation effect

In order to study the actual effect of recommendation, we compare the semantic and association oriented web service recommendation method with the other three methods, namely GM [15], Cloudrank2 [16] and 2RHyRec [12]. Recommendation performance is evaluated by checking the first *n* (*n* = 5, 10, 15 and 20) web service qualities. The average precision and average recall of each first *n* recommendations are used as evaluation metrics.

Average accuracy = number of top *n* recommended services / recommended capacity (8)

Average recall = number of top *n* recommended services / number of user confirmed item sets (9)

In the experiment, this method is compared with collaborative filtering, Top-k Association and content-based methods.

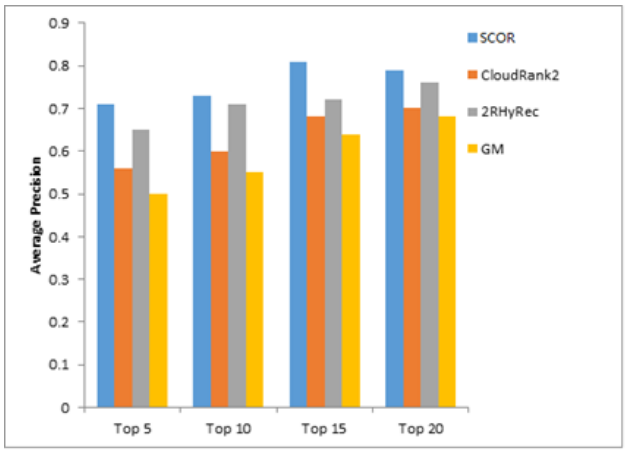

Fig. 6 Average accuracy of each method

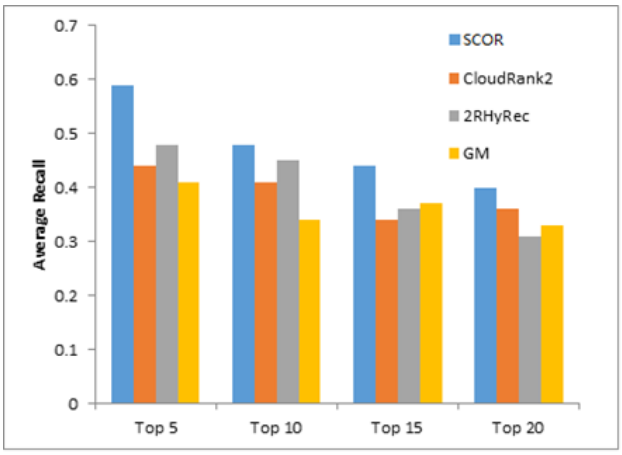

Fig.7 Average recall of each method

It can be seen from Figure 6 and Figure 7 that when *n* = 5, 10, 15 and 20, the precision and recall of the hybrid method are better than other methods. This is because the hybrid method combines the advantages of other methods and can learn from each other in recommendation precision and effect, so as to make up for the incompleteness caused by using a single service recommendation method for users. With the increase of *N*, the recommendation effect is getting better and better, because the recommendation results are more comprehensive performance of a service. When the project base is large and the value of *n* is too small, there will be omission of recommendation results, which also shows that the recommendation method can do better when the number of recommended projects is large.

4.4.4 Algorithm efficiency

In order to verify the time efficiency of the proposed method, the time overheads of different methods are evaluated. The average time consumed by each method is calculated by the response time (*RT*) prediction and throughput (*TP*) prediction (only integers of minutes) in minutes (*m*). The experimental results are shown in Table IV.

Table IV Efficiency comparison

Method	QoS attribute	
	RT	TP
UPCC	50	48
IPCC	99	89
WSRec	149	136
LPOR	19	18

Table IV shows that the service-based recommendation and management method consumes more time than the user-based recommendation and management method, mainly because the number of services is much larger than the number of users, so when the service-based recommendation and management method is used, the amount of computation is relatively large, resulting in more time overheads. The web service recommendation and management method based on location and quality preferences are shorter than the other methods, further characterizing similar *TP* and *RT* prediction time. Experimental results show that the proposed method improves the time efficiency of the recommendation and management method, whilst achieving better prediction accuracy.

4.4.5 Recommended method selection

As shown in Table V, the hybrid association-and-semantic-oriented recommendation and management method is more dynamic and adaptive to various scenarios and exhibits better prediction and time efficiency as it considers the relevance and semantic characteristics of services.

Table V Comparison of recommendation method selection

Method	Sparse Data	Accuracy	Efficiency	Data Object
UPCC	Applicable	Low	high	users
IPCC	Applicable	Medium	higher	services
WSRec	Applicable	Higher	low	users and services
SCOR	Applicable	High	low	users and services
LPOR	Applicable	Higher	high	users and services

5 CONCLUSIONS

The paper proposed a hybrid method of service recommendation and management based on QoS prediction along with location and quality preference analysis to enhance the prediction accuracy with reduced time overheads. By quantifying the weights of users' QoS preferences for services and combining the location characteristics, the proposed hybrid method of service recommendation and management returns the best matching services to user queries. The proposed method is faster whilst ensuring reliable prediction accuracy. Experiments conducted based on real-world datasets showed that the proposed location-based and quality-preference-oriented hybrid recommendation and management, not only be accurate in offering matching services, but also achieves accurate QoS prediction with faster computation.

Despite enhancing the prediction accuracy and service efficiencies of recommendation systems, the proposed hybrid methods of service recommendation and management can still be improved in various perspectives. On the one hand, in future work, we intend to do more experiments based on a larger dataset, including more dynamic service attributes such as performance and scalability etc. Therefore, other types of features such as time series, network type will be considered in the future. On the other hand, services recommendation and management failures caused by the disappearance of services have not been discussed in this article, exploring this aspect is our another research direction. Service recommendation and personalized management of mobile web services are likely to be ubiquitous in the big data age.

Acknowledgements

The work reported in this paper has been supported by the National Natural Science Foundation of China Program

(61502209) and UK-Jiangsu 20-20 World Class University Initiative programme.

REFERENCES

[1] T. Hofmann, "Latent Semantic Models for Collaborative Filtering," ACM Transaction for Info. Syst., vol. 22, no. 1, pp. 89-115, 2004.

[2] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," vol. 13, pp. 259-266, 2003.

[3] Z. Chen, L. Shen, F. Li, D. You, and J. P. B. Mapetu, "Web service QoS prediction: when collaborative filtering meets data fluctuating in big-range," World Wide Web-internet & Web Information Systems, vol. 23, pp. 1715-1740, 2020.

[4] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A Collaborative Filtering Based Web Service Recommender System," in IEEE International Conference on Web Services, 2009, pp. 437-444.

[5] Y. Yin, L. Chen, Y. Xu, J. Wan, H. Zhang, and Z. Mai, "QoS Prediction for Service Recommendation with Deep Feature Learning in Edge Computing Environment," Mobile Networks & Applications, vol. 25, pp. 391-401, 2020.

[6] M. Li, Q. Lu, and M. Zhang, "A Two-Tier Service Filtering Model for Web Service QoS Prediction," IEEE Access, vol. 8, pp. 221278-221287, 2020.

[7] R. Nagarajan and R. Thirunavukarasu, "A Service Context-Aware QoS Prediction and Recommendation of Cloud Infrastructure Services," Arabian Journal Forence & Engineering, vol. 45, pp. 2929–2943, 2020.

[8] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services," IEEE Transactions on Services Computing, vol. 3, pp. 223-235, 2010.

[9] J. Li and J. Lin, "A probability distribution detection based hybrid ensemble QoS prediction approach," Information Sciences, vol. 519, pp. 289-305, 2020.

[10] W. Wang, G. Cassar, G. Cassar, and K. Moessner, "An experimental study on geospatial indexing for sensor service discovery," Expert Systems with Applications, vol. 42, pp. 3528-3538, 2015.

[11] K.HUANG, D.D.WANG, Q.CUI, C.L.HAO, Q.WANG, "Web Service QoS Prediction Based on Auxiliary Features," Computer System Application, vol. 25, pp. 154-161, 2016.

[12] G. Linden, B. Smith, and J. York, "Linden G, Smith B and York J: 'Amazon.com recommendations: item-to-item collaborative filtering', Internet Comput. IEEE" vol. 7, pp. 76-80, 2003.

[13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens:an open architecture for collaborative filtering of netnews," in ACM Conference on Computer Supported Cooperative Work, 1994, pp. 175-186.

[14] H. Kim, Y. Park, J. Kim, and D. Hong, "A Low-Complex SVD-Based F-OFDM," IEEE Transactions on Wireless Communications, vol. 19, pp. 1373-1385, 2020.

[15] W. W. Cohen, R. E. Schapire, and Y. Singer, "Learning to order things," vol. 10, pp. 451-457, 2011.

[16] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, "QoS Ranking Prediction for Cloud Services," IEEE Transactions on Parallel & Distributed Systems, vol. 24, pp. 1213-1222, 2013.

[17] A. Jennings and H. Higuchi, "A user model neural network for a personal news service," User Modeling and User-Adapted Interaction, vol. 3, pp. 1-25, 1993.

[18] X. Wei and W. B. Croft, "LDA-based document models for ad-hoc retrieval," in International ACM SIGIR Conference on Research and Development in Information Retrieval, 2006, pp. 178-185.

[19] N. Thio and S. Karunasekera, "Automatic Measurement of a QoS Metric for Web Service Recommendation," in Australian Conference on Software Engineering, 2005, pp. 202-211.

[20] P. Lops, M. D. Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," Recommender Systems Handbook, pp. 73-105, 2011.

[21] M. J. Paul and R. Girju, "A Two-Dimensional Topic-Aspect Model for Discovering Multi-Faceted Topics," in Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, Usa, July, 2010.

[22] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments," Uncertainty in Artificial Intelligence, vol. 17, pp. 437-444, 2001.

[23] H. Wu, Y. Wang, and X. Cheng, "Incremental probabilistic latent semantic analysis for automatic question recommendation," pp. 99-106, 2008.

[24] T. L. Griffiths and M. Steyvers, "Finding scientific topics," Proc Natl Acad Sci U S A, vol. 101, pp. 5228-5235, 2004.

[25] L. L. Shi, L. Liu, Y. Wu, L. Jiang, and J. Hardy, "Event Detection and User Interest Discovering in Social Media Data Streams," IEEE Access, vol. 5, pp. 20953-20964, 2017.

[26] L. Shi, Y. Wu, L. Liu, X. Sun, and L. Jiang, "Event detection and identification of influential spreaders in social media data streams," Big Data Mining & Analytics, vol. 1, pp. 34-46, 2018.

[27] J. Ge, L. L. Shi, L. Liu, and X. Sun, "User Topic Preferences based Influence Maximization in Overlapped Networks," IEEE Access, vol. 8, pp. 141328-141345, 2019.

[28] M. Ahmed, L. Liu, J. Hardy, B. Yuan, and N. Antonopoulos, "An efficient algorithm for partially matched services in internet of services," Personal & Ubiquitous Computing, vol. 20, pp. 283-293, 2016.

[29] M. Wang, L-L. Shi, L. Liu, M. Ahmed, J. Panneerselvam, Hybrid Recommendation based QoS Prediction for Sensor Services, International Journal of Distributed Sensor Networks, 14 (5), 2018, pp 1-10.

[30] L. Liu, N. Antonopoulos, M. Zheng, Y. Zhan, Z. Ding, A Socio-ecological Model for Advanced Service Discovery in Machine-to-Machine Communication Networks, ACM Transactions on Embedded Computing Systems, Vol 15(2), 2016, pp 38:1–38:26.

[31] S. Tang, S. Yuan, and Y. Zhu, "Convolutional Neural Network in Intelligent Fault Diagnosis Toward Rotatory Machinery," IEEE Access, vol. 8, pp. 86510-86519, 2020.

[32] S. Tang, S. Yuan, and Y. Zhu, "Deep Learning-Based Intelligent Fault Diagnosis Methods Toward Rotating Machinery," IEEE Access, vol. 8, pp. 9335-9346, 2020.

[33] S. Zeng, B. Zhang, Y. Lan, and J. Gou, "Robust collaborative representation-based classification via regularization of truncated total least squares," Neural Computing & Applications, vol. 31, pp. 5689–5697, 2019.

[34] B. Yuan, J. Panneerselvam, L. Liu, N. Antonopoulos and Y. Lu, "An Inductive Content-Augmented Network Embedding Model for Edge Artificial Intelligence," in IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 4295-4305, 2019.

[35] M. Nilashi, P. F. Rupani, M. M. Rupani, H. Kamyab, W. Shao, H. Ahmadi, et al., "Measuring sustainability through ecological sustainability and human sustainability: A machine learning approach," Journal of Cleaner Production, vol. 240, p. 118-162, 2019.

[36] Z. Han, W. K. S. Tang, and Q. Jia, "Event-Triggered Synchronization for Nonlinear Multi-Agent Systems With Sampled Data," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, pp. 3553-3561, 2020.

[37] J. Ge, L. -L. Shi, Y. Wu and J. Liu, "Human-Driven Dynamic Community Influence Maximization in Social Media Data Streams," in IEEE Access, vol. 8, pp. 162238-162251, 2020.

[38] M. Y. Wu, L. Zhang, and M. Imran, "Research on Evolution Model of Employees' Relationship and Task Conflict in Enterprise Clusters: A Complex Network Perspective," Wireless Personal Communications, vol. 102, pp. 3489–3501, 2018.

[39] L. Jiang, L. Shi, L. Liu, J. Yao, and M. A. Yousuf, "User interest community detection on social media using collaborative filtering," Wireless Networks, 2019. https://doi.org/10.1007/s11276-018-01913-4.

[40] G. Chen, L. Xu, X.-B. Liu, Wang, and Zhao, "Picking Robot Visual Servo Control Based on Modified Fuzzy Neural Network Sliding Mode Algorithms," Electronics, vol. 8, pp. 605-622, 2019.

[41] A. Monney, Y. Zhan, J. Zhen, and B. B. Benuwa, "A Multi-Kernel Method of Measuring Adaptive Similarity for Spectral Clustering," Expert Systems with Applications, vol. 159, pp. 157-168, 2020.

[42] C. Y. Mao, J. F. Chen, D. Towey, J. F. Chen, X. Y. Xie, "Search-based QoS ranking prediction for web services in cloud environments", Future Generation Computer Systems, Vol. 50, pp. 111-126, 2015.

[43] H. Lu, S. Liu, H. Wei, and J. Tu, "Multi-kernel fuzzy clustering based on auto-encoder for fMRI functional network," Expert Systems with Applications, vol. 159, pp. 113-123, 2020.

[44] J. Gou, B. Hou, Y. Yuan, W. Ou, and S. Zeng, "A new discriminative collaborative representation-based classification method via l2

regularizations," Neural Computing and Applications, vol. 32, pp. 9479–9493, 2020.

[45] J. Gou, L. Wang, B. Hou, J. Lv, Y. Yuan, and Q. Mao, "Two-Phase Probabilistic Collaborative Representation-Based Classification," Expert Systems with Applications, vol. 133, pp. 9-20, 2019.

[46] T. Guan, F. Han, and H. Han, "A Modified Multi-Objective Particle Swarm Optimization Based on Levy Flight and Double-Archive Mechanism," IEEE Access, vol. 7, pp. 183444-183467, 2019.

[47] H. Q. Wu, L. M. Wang, S. R. Jiang. "Secure Top- k Preference Query for Location-based Services in Crowd-outsourcing Environments" ,The Computer Journal, vol. 61, no. 4, pp. 496-511, 2018.

[48] Xu, Z, Y. Wu, L. Liu, "A novel multilevel index model for distributed service repositories", Tsinghua Science and Technology, vol. 22, no. 3, pp. 273-281, 2017.

[49] Y. Wu, W. Xu, L. Liu, and D. Miao, "Performance formula‐based optimal deployments of multilevel indices for service retrieval," Concurrency & Computation Practice & Experience, vol. 31, pp. 4265.1-4265.12, 2017.