

Diversifying Web Service Recommendation Results via Exploring Service Usage History

Guosheng Kang, *Student Member, IEEE*, Mingdong Tang, Jianxun Liu, *Member, IEEE*, Xiaoqing(Frank) Liu, *Member, IEEE*, and Buqing Cao, *Member, IEEE*

Abstract—The last decade has witnessed a tremendous growth of Web services as a major technology for sharing data, computing resources, and programs on the Web. With the increasing adoption and presence of Web services, design of novel approaches for effective Web service recommendation to satisfy users' potential requirements has become of paramount importance. Existing Web service recommendation approaches mainly focus on predicting missing QoS values of Web service candidates which are interesting to a user using collaborative filtering approach, content-based approach, or their hybrid. These recommendation approaches assume that recommended Web services are independent to each other, which sometimes may not be true. As a result, many similar or redundant Web services may exist in a recommendation list. In this paper, we propose a novel Web service recommendation approach incorporating a user's potential QoS preferences and diversity feature of user interests on Web services. User's interests and QoS preferences on Web services are first mined by exploring the Web service usage history. Then we compute scores of Web service candidates by measuring their relevance with historical and potential user interests, and their QoS utility. We also construct a Web service graph based on the functional similarity between Web services. Finally, we present an innovative diversity-aware Web service ranking algorithm to rank the Web service candidates based on their scores, and diversity degrees derived from the Web service graph. Extensive experiments are conducted based on a real world Web service dataset, indicating that our proposed Web service recommendation approach significantly improves the quality of the recommendation results compared with existing methods.

Index Terms—Web service recommendation, diversity, user interest, QoS preference, service usage history

1 INTRODUCTION

Web services have been rapidly developed in recent years and played an increasingly significant role in e-commerce, enterprise application integration, and other applications. With the growth of the number of Web services on the Internet, Web service discovery has become a critical issue to be addressed in service computing community [1]. Since there are many Web services with similar functionalities and different non-functional quality, it is important for users to select desirable high-quality Web services which satisfy both users' functional and non-functional requirements.

Recently, recommending qualified and preferred Web services to users has attracted much attention in terms of the information overload problem. Web service recommendation is a process of proactively discovering and recommending suitable Web services to end users. A number of works have been done on service recommendation based on quality of service (QoS). Most of them employed Collaborative Filtering (CF) techniques [2-6], some of them applied content-based approach, and a few of them combined CF approach with content-based techniques. They focus on predicting missing QoS values of Web services used by similar users for an active user.

However, there are drawbacks for these approaches. To begin with, they simply recommend users Web services with the best QoS values on a certain QoS criterion without exploiting the user's potential QoS preferences, which may likely be mined from his/her service usage history [7]. A user's QoS preference for services is certainly important for real service recommendation scenarios, since it can be used for measuring the QoS utility of a Web service in a more accurate and personalized way. Moreover, existing service recommendation approaches may have unneeded similar services in the top-k recommendation lists, since there is a default assumption that all the k results are independent of each other, which may not be true in many times. As a result, the user's satisfaction degree may decrease in their experience of selection in the recommended list due to the redundant services in the limited top-k recommendation list. For example, suppose there is a certain category of services with similar or related function (i.e., in the same service domain) which match a user's interests and have comparatively higher QoS than the services in other categories. It is probable that existing service recommendation approaches will only recommend services in this category to the user in the final short recommendation list. From the user's viewpoint, however, the recommended services with similar functionality are redundant, and services in the other categories which are interesting to the active user should also be incorporated as many as possible in the only limited top-k recommendation list. In order to remove the redundancy in service recommendation list, and at the

- Guosheng Kang is with the School of Computer Science, Fudan University, Shanghai, China, 201203. E-mail: guoshengkang@gmail.com.
- Mingdong Tang, Jianxun Liu and Buqing Cao are with the Department of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China, 411201. E-mail: {tangmingdong, ljx529, buqingcao}@gmail.com.
- Xiaoqing(Frank) Liu is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65401, USA. E-mail: fliu@mst.edu

same time maintain the quality of the recommended services, diversity should be considered in recommendation.

In recommender systems, when the k best recommendations are very similar to each other, many of them may be useless to the user, and thus the usefulness of k recommendations may be very low. It is desirable for a recommender system to return a diverse set of cases in order to provide the user with optimal coverage of the information space [8]. Currently, diversity is considered as important as similarity in many existing recommender systems [9-12]. For example, Zhou et al. [9] discussed the diversity-accuracy dilemma of recommender systems, showing that hybrid method with diversity can improve the recommendation performance. Ziegler et al. [10] proposed that recommendation can be improved through topic diversification. Based on these facts, we argue that diversity is also an important feature in Web service recommendation systems. In this paper, we propose a novel service recommendation approach by taking diversity into consideration. We incorporate the functional relevance, QoS utility, and diversity features of Web services for recommending well diversified top- k services to users. Specifically, the contributions are as follows.

- 1) We mine a user's functional interests and QoS preferences by exploring his/her service usage history. The user interests are two-fold: the historical user interest and the potential user interest. The historical user interest is mined through its own service usage history, query logs and profile, while the potential user interest is derived through collaborative filtering approach. User interests and QoS preferences are used for measuring the functional relevance and QoS utility respectively for Web service candidates.
- 2) We compute a score for each Web service candidate using the functional relevance and QoS utility. Meanwhile, we construct a Web service graph based on the functional similarity between service candidates with a certain level of user interest relevance. A diversity measure is defined based on the Web service graph.
- 3) We perform a novel diversity-aware service ranking algorithm to find the optimal top- k Web services based on a proposed comprehensive ranking measure. The experimental results indicate that the proposed approach improves the performance of service recommendation compared with the existing methods.

The rest of this paper is organized as follows: Section 2 surveys the related works on service recommendation. Section 3 presents our framework of the proposed Web service recommendation approach. Section 4 discusses how to effectively recommend Web services and diversify the service recommendation results in detail, including functional evaluation, non-functional evaluation, diversity evaluation, and ranking of Web services. Section 5 describes the experimental results. Finally, we draw conclusions and outline our future work in Section 6.

2 RELATED WORK

To discover high quality Web services, a number of QoS models for Web services and QoS-driven service selection

approaches have been proposed in the service computing field [13-17]. In their study, it is usually assumed that a user explicitly specifies his/her interests (e.g., by using keywords) and QoS requirements, and submits them to the service discovery system [18]. Then the service discovery system matches the user's interests and QoS requirements with corresponding attributes of Web services, and returns those with the best matching degrees to the user [19-21]. The scenarios for service selection can be divided into two categories. The first scenario aims to select a set of services for a composite service, which is widely studied by existing work on service selection [16,17,22]. The second scenario is to select a single service for a user request [23,24], or to select multiple services with the same function for multiple user requests [15,25]. Recently, there has been an increasing interest in actively recommending qualified and preferred Web services to users without initiating service requests. Existing Web service recommendation approaches can be roughly divided into three categories: CF-based approaches, content-based approaches, and hybrid approaches. In the following, we survey the related work on service recommendation in these three categories, and on diversity-based ranking algorithms.

2.1 CF Methods for Service Recommendation

The main idea of collaborative filtering is to recommend new items of interest to a user regarding the other users' experiences over a set of items. Existing collaborative filtering algorithms can be divided into two categories: memory-based and model-based. Memory based methods are more popular in service recommendation, partially because they are more intuitive to interpret the recommendation results [26]. Memory-based collaborative filtering can be further divided into user-based approaches [27] and item-based approaches [28]. User-based collaborative filtering methods recommend a user the items preferred by the users with similar interests, while item-based collaborative filtering methods recommend a user the items similar to those he/she preferred in the past. Shao et al. [6] proposed a user-based CF approach that uses PCC (Pearson Correlation Coefficient) to compute similarity between users in terms of their experiences on used Web services. Zheng et al. [29] proposed a hybrid CF approach for QoS-aware service recommendation by combining both item-based PCC (IPCC) and user-based PCC (UPCC). They exploited not only similarity among users but also similarity among services for missing QoS prediction. Jiang et al. [2] improved the hybrid CF-based service recommendation approach of [5] by taking the personal characteristics of users and services into consideration when measuring similarity among users and services using PCC. Zhang et al. [30] proposed a context-aware service recommendation model, which simultaneously considered users' experiences, the target user's environment factor and his/her input factor to make recommendation decisions. Chen et al. [4] observed that user-perceived QoS metrics of services are highly related to users' physical locations on the Internet. They proposed an efficient region model with the properties of QoS for QoS prediction. Tang et al. [3] incorporated both users' and services' location information into QoS prediction, and pro-

posed a location-aware CF method for service recommendation. Zhang et al. [31] observed that QoS performance of services is highly related to the service status and network environments which are variable against time. They proposed a QoS prediction framework, called WSPred, to provide time-aware personalized QoS value prediction for different service users. Wu et al. [32] presented a neighborhood-based collaborative filtering approach to predict such unknown values for QoS-based service selection. Most recently, some CF based service recommendation approaches employed the matrix factorization theory to improve the accuracy of QoS prediction [33,34].

2.2 Content-based Methods for Service Recommendation

Content-based service recommendation approaches focused on exploring the description information of Web services and the user's own service usage history. Generally, the Web services which are highly relevant to the user's service usage history and own high QoS utility would be recommended to users. Kang et al. [35] proposed an active Web service recommendation approach based on service usage history which incorporates both user interest and QoS preference into Web service recommendation. With the user interest and QoS preference, recommender systems can recommend top-k optimal services with user-desired functional and non-functional requirements. In [35], a user's potential QoS preference is acquired by the average QoS preference from service usage history. This potential QoS preference is used for all the service candidates. However, the QoS preference may be not accurate because a user may have different QoS preferences to different services. Therefore, this approach should be further improved. Liu et al. [36] proposed a semantic content-based recommendation approach that analyzes the context of intended service use to provide effective recommendations in conditions of scarce user feedback. Hu et al. [37] proposed a personalized search approach for Web service recommendation, in which interests are extracted from users' records. While these two works do not consider QoS preferences and potential user interests of users, which will be addressed in this work.

2.3 Hybrid Methods for Service Recommendation

Hybrid service recommendation approaches combined both collaborative filtering and content-based recommendation techniques. Freddy [38] proposed a semantic content-based recommendation system that provides end-users with recommendations about semantic Web services that could be of their interest. Firstly, this approach considers the neighbors of the active user by computing similarities between different users' personal information. Then, Web services manipulated by similar users, except the services already used by the active user, are ranked depending on their semantic similarity with services the active end-user used to interact with. Finally, the top k services are recommended to the user. Yao et al. [39] proposed a hybrid service recommendation approach by combining CF with content-based features of Web services. This approach exploited both rating data and content data of ser-

vices via using a three-way aspect model. In their work, user interests are represented by a set of latent variables, which is developed offline. However, QoS preferences of users are not considered in these works.

2.4 Diversity-based Ranking algorithms

As mentioned previously, existing service recommendation approaches assumed that all the returned results are independent to each other, which may result in many unneeded similar Web services in a top-k recommendation list. As a result, diversity of the recommended Web services would be quite low. Actually, diversity has been widely considered as an important criterion in the areas of information retrieval and data mining, and there has been a large body of research work on diversifying search results in text, graph and other searching applications [40-43]. Smyth et al. [11] argued that often diversity can be as important as similarity, especially in case-based recommendation systems. Hurley et al. [12] conducted analysis and evaluation on novelty and diversity. They argued that the motivation of diversity research is to increase the probability of retrieving unusual or novel items which are relevant to the user and introduce a methodology to evaluate their performance in terms of novel item retrieval.

In the literature, employing graph models to do diversity-based ranking is the mainstream, and there are two popular techniques. The first one is based on a greedy vertex selection procedure [44,45], and the second one is based on a so-called vertex reinforced random walk [46]. In particular, the greedy vertex selection procedure chooses a vertex with maximum random walk based ranking score at one time, and then removes the selected vertex from the graph. To get the top-k ranking list, this process will repeat K times. There are two algorithms based on this framework: the Grasshopper algorithm [44] and the manifold rank with stop points algorithm [45]. Both of the two algorithms have empirically shown that they can improve diversity in ranking on graph-type data. However, the drawback of this type of algorithm is that they lack theoretical explanation for the algorithm on why it can improve diversity of the ranking results. In [46], Mei et al. proposed a diversified ranking algorithm, named DivRank, based on a vertex reinforced random walk. They also presented an optimization explanation for DivRank so that it can achieve diversity in ranking. However, their optimization explanation cannot be used in directed graphs other than undirected graphs. In addition, the convergence property of DivRank is not clear, because it resorts to some approximation strategies to the original vertex reinforced random walk.

The above diversified ranking algorithms neither are scalable to large graphs due to the time or memory requirements, nor are intuitive and reasonable diversified ranking measures. To overcome the problems in above algorithms, Rong-Hua Li et al. [47] proposed a novel diversified ranking measure on large graphs, which captures both relevance and diversity, and formulate the diversified ranking problem as a sub-modular set function maximization problem. An efficient greedy algorithm with linear time and space complexity was developed.

However, diversifying Web service recommendation re-

sults has received little attention so far, and the existing diversity-based ranking algorithm cannot be applied directly for the practical Web service recommendation. In this paper, we will address this issue by recommending top-k optimal Web services to a user that not only match well with the user's interests and QoS requirements but also have a sound diversity. Modified from literature [47], we also formulate the Web service recommendation with diversity as a sub-modular set function maximization problem, and developed an greedy algorithm with linear time and space complexity which is suitable for handling large number of Web services in a big data environment.

3 FRAMEWORK OF OUR APPROACH

Now we describe the framework of our service recommendation approach which takes diversity into consideration as shown in Figure 1. In the framework, Web Service Recommendation with Diversity (WSRD) is the key component. For simplicity, we suppose that the service usage history and functional description information and QoS information of all services are already provided or acquired. The collected service pool can be updated dynamically by the service search engine. However, we assume that the number of services does not change in the small interval during the process of service recommendation.

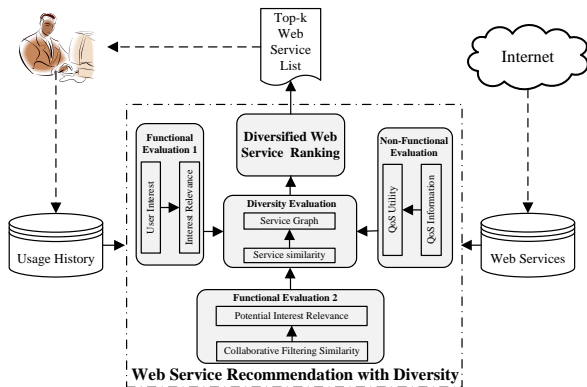


Fig. 1. Framework of Our Service Recommendation Approach

WSRD has four subcomponents: functional evaluation, non-functional evaluation, diversity evaluation, and diversified Web service ranking, as shown in Figure 1. The functional evaluation can be further divided into two parts: Functional Evaluation 1 and Functional Evaluation 2. Functional Evaluation 1 evaluates the relevance of the user's historical interest with Web services based on a content-based similarity measure. Content-based similarity is acquired by text similarity. This work only considers Web services that are described by the Web Service Description Language (WSDL). Nevertheless, it is easy to extend our work to handle other kinds of Web services. The user's historical interest can be mined from his/her own service usage or query history. Functional Evaluation 2 predicts the user's potential interest and evaluates its relevance with Web services by employing collaborative filtering based user similarity. The user similarity is measured based on the service invocation history of all service users. Non-functional Evaluation first infers the user's potential

QoS preference on a service candidate through mining the service's usage history, then calculates the QoS utility of the Web service with the obtained QoS information. Diversity Evaluation first calculates the functional similarity between service candidates, and then constructs a Web service graph with the computed similarity values between service candidates. After functional, non-functional, and diversity evaluation, WSRD performs the diversified service ranking algorithm based on the functional relevance, QoS utility and Web service graph to yield a well diversified top-k service recommendation list for the active user.

4 THE SERVICE RECOMMENDATION APPROACH

Suppose there are M services in the service usage history of an active user u , denoted by $WS_{u,1}, WS_{u,2}, \dots, WS_{u,M}$, and the QoS preference vector specified by the user on $WS_{u,i}$ is denoted by $P_{u,i}$. If the Web services invoked by the user were recommended by the Web service search engine, their QoS preference vectors may also be empty. Suppose there are N Web service candidates for Web service recommendation, which are WS_1, WS_2, \dots, WS_N . With these notations, next we describe our Web service recommendation approach, which includes functional evaluation, non-functional evaluation, diversity evaluation, and diversified Web service ranking.

4.1 Functional Relevance based on Historical User Interest

Terms in WSDL documents of all the available Web service candidates can be looked upon as a corpus and therefore we employ the well-known TF/IDF (Term Frequency/Inverse Document Frequency) [48] algorithm to weight the importance of terms in the corpus. TF/IDF is a statistical measure to evaluate how important a word is to a document in the corpus. The importance increases proportionally to the number of times a word appearing in the document but is offset by frequency of the word in the corpus. Variations of TF/IDF weighting scheme are often used by search engines as a central tool in scoring and ranking documents' relevance given a user query. It analyzes most common terms appearing in each document and appearing less frequently in other documents.

We extract the meaningful words from WSDL documents to form the corpus. The format of a WSDL document conforms to the rules of XML document. There may be many words or tokens which need to be preprocessed in WSDL documents. There are usually three steps for document preprocessing: *normalization*, *word stemming*, and *tokens and stop-words removing*. Specific details on WSDL document preprocessing can be referred to [35,49].

The term count in a given document is the number of times a given term appears in the document. This count is usually normalized to prevent a bias towards longer documents (which may have a larger term count regardless of the actual importance of that term in the document) to give a measure of the importance of the term. Thus the term frequency $tf(t_j, WSDL_i)$ of the j^{th} term of the WSDL document $WSDL_i$ in the corpus is defined in the simplest case as the occurrence count of the term in the document, which

is calculated as follows:

$$tf(t_j, WSDL_i) = \frac{freq(t_j, WSDL_i)}{|WSDL_i|} \quad (1)$$

where t_j is the j^{th} term in the corpus; $WSDL_i$ is the WSDL document of the i^{th} Web service WS_i ; $freq(t_j, WSDL_i)$ is the occurrence number of t_j in $WSDL_i$; $|WSDL_i|$ is the total occurrence number of all meaningful terms in $WSDL_i$. Therefore, we should have $|WSDL_i| = \sum_j freq(t_j, WSDL_i)$.

The inverse document frequency $idf(t_j, WSDL_i)$ is a measure of the general importance of the term t_j (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient), calculated as follows:

$$idf(t_j, WSDL_i) = \log_2 \frac{|WSDL|}{|\{WSDL_i: t_j \in WSDL_i\}|} \quad (2)$$

where $|WSDL|$ is the total number of WSDL documents, which equals to the number of Web services, i.e., $|WSDL| = N$; $|\{WSDL_i: t_j \in WSDL_i\}|$ is the number of documents where the term t_j appears. If the term is not in the corpus, this leads to a division-by-zero. It is therefore common to adjust the denominator to $1 + |\{WSDL_i: t_j \in WSDL_i\}|$.

Since WSDL documents are generally short, we choose to give higher weight to the IDF value to normalize the inherent bias with Formula (3), while the common implementation of TF/IDF gives equal weights to term frequency and inverse document frequency (i.e., $\omega = tf * idf$). The reason behind this modification is to normalize the inherent bias of TF measure in short documents [50].

$$\omega_{i,j} = tf(t_j, WSDL_i) * idf^2(t_j, WSDL_i) \quad (3)$$

A high weight in TF/IDF is achieved through a high term frequency and a low document frequency of the term in the whole collection of documents. The weights hence tend to filter out very common terms.

Suppose there are n terms in the corpus. Then with TF/IDF algorithm, the WSDL document of a Web service WS_i can be transformed into a term weight vector \mathbf{w}_i . To measure the text similarity between WS_i and WS_j , we use the Cosine Similarity.

$$texSim = \cos(\mathbf{w}_i, \mathbf{w}_j) = \frac{\mathbf{w}_i \cdot \mathbf{w}_j}{|\mathbf{w}_i| \times |\mathbf{w}_j|} \quad (4)$$

where $|\mathbf{w}_i|$ and $|\mathbf{w}_j|$ are the Euclidean length of the vector \mathbf{w}_i and \mathbf{w}_j respectively, and the numerator is the dot product of \mathbf{w}_i and \mathbf{w}_j .

It is worth noting that treating WSDL as common text and simply employing text similarity to measure the functional similarity of Web services may be not enough. Two Web services with a low text similarity could be similar because they may contain similar or equivalent service operations. Thus, measuring the similarity of service operations is also needed for assessing two Web services' similarity. A Web service operation consists of three elements $OP_f = (\mathbf{K}, \mathbf{In}, \mathbf{Out})$ [49]. The *keywords* (\mathbf{K}) element of service operation i can be denoted by a vector of words $\mathbf{K}^i = (k_1^i, k_2^i, \dots, k_l^i)$. The *input* (\mathbf{In}) and the *output* (\mathbf{Out}) elements can be denoted by vectors $\mathbf{In}^i = (in_1^i, in_2^i, \dots, in_m^i)$ and $\mathbf{Out}^i = (out_1^i, out_2^i, \dots, out_n^i)$, where in_k^i and out_k^i are terms appeared in the *input* and the *output* respectively.

Thus, service operations are described as sets of terms. By applying the TF/IDF measure into these sets, we can also measure the service operation similarity $opSim$ of two service operations by using Cosine Similarity, i.e., $opSim = \cos(\mathbf{K}^i, \mathbf{K}^j) + \cos(\mathbf{In}^i, \mathbf{In}^j) + \cos(\mathbf{Out}^i, \mathbf{Out}^j)$. Considering that a Web service may contain multiple service operations, we measure the operation similarity of two Web services with the maximal service operation similarity of two service operations from the two Web services respectively. Therefore, if two Web services have high text similarity and high service operation similarity, we can say that they are similar. Based on the above observations, we define the similarity $S_{i,j}^{ws}$ of two Web services as follows, where φ and ϕ are adjustment parameters, satisfying $\varphi + \phi = 1$.

$$wsSim(WS_i, WS_j) = \varphi texSim + \phi opSim \quad (5)$$

Considering that an user's historical interest may change over time, to acquire the user's latest interests accurately, we construct a user interest vector by merging WSDL documents of recently used Web services into one WSDL document, named *BigWSDL*. Then we also add the corresponding query logs and the user's profile into the *BigWSDL*, namely *BigWSDL* + . *BigWSDL* + is transformed into a term vector \mathbf{w}_{ui} via TF/IDF, indicating the user's historical interest. Since \mathbf{w}_{ui} is a full description of the user's historical interest, it is reasonable to use it to yield diversified Web service recommendation results for the active user. We measure the similarity S_i^H between *BigWSDL* + and a Web service candidate WS_i with Formula (5), indicating the very historical user interest relevance.

4.2 Functional Evaluation based on Potential User Interest

The relevance of Web services with the user interest computed above is only based on the active user's own Web service usage history, while the service experiences of the other users are neglected. The service experiences of the other users can be used to predict the potential interest of the active user. Thus, we also apply collaborative filtering approach to find the Web services which are probably interesting to the active user.

In collaborative filtering approach, user similarity is calculated based on the Web service invocation records of a set of users. Similar users usually share the common interest, so they are likely to use Web services with the same functionality. The more commonly invoked Web services two users have in their invocation records, the larger the user similarity between them. Based on this observation, we use the following formula to calculate the user similarity $S_{i,j}^{user}$ between two users.

$$userSim(u_i, u_j) = \frac{2 \times |CS_{ij}|}{|S_{u_i}| + |S_{u_j}|} \quad (6)$$

where S_{u_i} and S_{u_j} are the sets of Web services used by user u_i and u_j respectively, CS_{ij} is the set of Web services used by both u_i and u_j , i.e., $CS_{ij} = S_{u_i} \cap S_{u_j}$. If $|CS_{ij}| = 0$, then we have $userSim(u_i, u_j) = 0$.

In CF based recommendation approach, usually the items used by similar users will be recommended to the active user. As for a Web service, several similar users may

have used it. Thus, in this paper, we use the maximum user similarity to the active user as the potential user interest relevance S_i^p of a Web service candidate to the active user.

4.3 Non-Functional Evaluation

Suppose that m QoS criteria are used for assessing the non-functional quality of WS_i , its QoS vector is denoted by QS_i , i.e., $QS_i = (q_{i,1}, q_{i,2}, \dots, q_{i,m})$, where $q_{i,j}$ represents the value of the j^{th} quality criterion. Generally, there are two types of QoS criteria. A QoS criterion is considered to be negative if the higher the value, the lower the quality, (e.g., Response Time and Cost). On the other hand, if the higher the value, the higher the quality, the QoS criterion is considered to be positive (e.g., Availability and Reliability). Values of different QoS criteria need to be normalized to the same range for uniform measurement purpose. While before normalization, we apply statistical method (i.e., Pauta Criterion method) to preprocess the QoS values in advance to remove the outliers. In this section, we transform each QoS criterion value to a real number between 0 and 1 by comparing it with the maximum and minimum values of the QoS criterion among all available Web service candidates. After such normalization processing, larger value for any quality criterion means better quality. Concretely, for a negative QoS criterion, its value $q_{i,j}$ would be normalized as $q'_{i,j}$ according to Formula (7), and for a positive criterion, $q_{i,j}$ would be normalized using Formula (8), which are defined as follows:

$$q'_{i,j} = \begin{cases} \frac{Q_{max}(j) - q_{i,j}}{Q_{max}(j) - Q_{min}(j)}, & \text{if } Q_{max}(j) \neq Q_{min}(j) \\ 1, & \text{if } Q_{max}(j) = Q_{min}(j) \end{cases} \quad (7)$$

$$q'_{i,j} = \begin{cases} \frac{q_{i,j} - Q_{min}(j)}{Q_{max}(j) - Q_{min}(j)}, & \text{if } Q_{max}(j) \neq Q_{min}(j) \\ 1, & \text{if } Q_{max}(j) = Q_{min}(j) \end{cases} \quad (8)$$

where the maximum value $Q_{max}(j)$ and minimum value $Q_{min}(j)$ of the j^{th} QoS criterion are defined as Formula (9) and (10), respectively. We denote QS'_i as the QoS vector of WS_i after normalization processing.

$$Q_{max}(j) = \max\{q_{i,j}\} \quad (9)$$

$$Q_{min}(j) = \min\{q_{i,j}\} \quad (10)$$

Let $P_{u,i} = (w_1, w_2, \dots, w_m)$ represents the user's QoS preference on different QoS criteria over the Web service candidate WS_i , where $w_j \geq 0$ and $\sum_{j=1}^m w_j = 1$. Then the QoS utility $U_{u,i}$ of WS_i is calculated as follows:

$$U_{u,i} = QS'_i \times P_{u,i}^T = \sum_{j=1}^m w_j \times q'_{i,j} \quad (11)$$

In WSRD, we mine the active user's potential QoS preferences to service candidates from historical QoS preferences of the active user or similar users who used the service candidates. We firstly transform WSDL document of $WS_{u,i}$ into a term vector $w_{u,i}$ by using TF/IDF algorithm. Then we compute the similarity $S_{i,j}^{ws}$ between WS_i and $WS_{u,j}$, i.e., $S_{i,j}^{ws} = wsSim(WS_i, WS_{u,j})$ which is calculated with Formula (5). As for $S_{i,1}^{ws}, S_{i,2}^{ws}, \dots, S_{i,M}^{ws}$, we find the Web services with both the similarity above the threshold ε and valid QoS preferences in service usage history to form the set S_{sim} . It is reasonable to assume that the active user has similar QoS preferences to similar Web service

candidates, i.e., Web services with similar functionalities. Based on this observation, we choose to use the QoS preferences used for the similar Web services in usage history to calculate the potential QoS preference over the current Web service candidate based on their Web service similarities, which is defined as Formula (12).

$$P_{u,i} = \frac{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws} \times P_{u,j}}{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws}} \quad (12)$$

In some cases, the number of services in S_{sim} could be very small. In this situation, the resulting QoS preference may not be accurate enough, since limited information can be used. To address this drawback, we use the QoS preferences from similar users who have used the current service before to compensate the problem of lacking information, which is defined as Formula (13), where $P_{u_k,i}$ is the effective QoS preference of WS_i by user u_k , and where ψ_1 and ψ_2 are adjustment parameters, with $\psi_1 + \psi_2 = 1$.

$$P_{u,i} = \psi_1 \frac{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws} \times P_{u,j}}{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws}} + \psi_2 \frac{\sum_{u_k \in U_{sim}} S_{u_k,i}^{user} \times P_{u_k,i}}{\sum_{u_k \in U_{sim}} S_{u_k,i}^{user}} \quad (13)$$

Based on the above discussion, we propose an algorithm for non-functional evaluation of Web service candidates, which is described in Algorithm 1. Please note that the QoS preference information may be not always available for Web services in the usage history. For example, when Web services used by a user are from recommendation, there will be no QoS preference information recorded in the usage history. Therefore, they cannot contribute to evaluate the potential QoS preference of the active user to the current Web service in Algorithm 1.

Algorithm 1 Non-Functional Evaluation

Input: $WS_{u,1}, WS_{u,2}, \dots, WS_{u,M}; P_{u,1}, P_{u,2}, \dots, P_{u,M}; \varepsilon; WS_1, WS_2, \dots, WS_N; QS_1, QS_2, \dots, QS_N$

Output: $U_{u,1}, U_{u,2}, \dots, U_{u,N}$

- 1: **for** $i=1$ to N **do**
 - 2: $QS'_i = \text{normalize}(QS_i)$;
 - 3: $S_{sim} = \emptyset$;
 - 4: **for** $j=1$ to M **do**
 - 5: $S_{i,j}^{ws} = wsSim(WS_i, WS_{u,j})$;
 - 6: **if** $S_{i,j}^{ws} > \varepsilon$ and $P_{u,j} \neq \emptyset$ **then**
 - 7: add $WS_{u,j}$ into S_{sim} ;
 - 8: **end if**
 - 9: **end for**
 - 10: **if** $|S_{sim}| < Num$ **then** // Num is a threshold number
 - 11: Find the top-10 similar users U_{sim} that have used $WS_{c,i}$;
 - 12: $P_{u,i} = \psi_1 \frac{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws} \times P_{u,j}}{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws}} + \psi_2 \frac{\sum_{u_k \in U_{sim}} S_{u_k,i}^{user} \times P_{u_k,i}}{\sum_{u_k \in U_{sim}} S_{u_k,i}^{user}}$;
 - 13: **else**
 - 14: $P_{u,i} = \frac{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws} \times P_{u,j}}{\sum_{WS_{u,j} \in S_{sim}} S_{i,j}^{ws}}$;
 - 15: **end if**
 - 16: $U_{u,i} = QS'_i \times P_{u,i}$;
 - 17: **end for**
 - 18: **return** $U_{u,1}, U_{u,2}, \dots, U_{u,N}$;
-

4.4 Diversity Evaluation

To evaluate the diversity degree of Web services in a recommendation list, we employ a generalized diversified ranking measure modified from [41] for Web services. For simplicity, we firstly specify the following definitions and

notations for discussing the diversity measurement.

Definition 1. Web Service Graph:

A Web service graph $G = (V, E)$ is an undirected weighted graph consisting of a set of nodes V and a set of edges E , wherein a node denotes a Web service candidate, i.e., $v_i = WS_i$, and an edge denotes that the connected nodes are similar. $|V| = K$ is the number of nodes (i.e., Web services) in the graph. Please note that not all the Web services in the Web service pool are used for constructing the Web service graph. Only the Web services with a certain relevance to user interest are used. We set two thresholds θ_H and θ_p for the historical user interest relevance and potential user interest relevance, respectively. Each node has a score calculated with Formula (14), where α, β , and γ are three parameters, satisfying $\alpha + \beta + \gamma = 1$, indicating how much each factor is important in the final score measurement. For each pair of nodes v_i and v_j , we compute the similarity between them. If the similarity value is no less than the predefined threshold τ , then an edge is added to connect them in the Web service graph G , i.e., $(v_i, v_j) \in E$, indicating that v_i and v_j are similar.

$$Score_{u,i} = \alpha S_i^H + \beta S_i^P + \gamma U_{u,i} \quad (14)$$

Based on the above definition of the Web service graph, Algorithm 2 presents the procedure of Web service graph construction.

Algorithm 2 Web Service Graph Construction

Input: $S_1^H, S_2^H, \dots, S_N^H; S_1^P, S_2^P, \dots, S_N^P; U_{u,1}, U_{u,2}, \dots, U_{u,N}; \theta_H, \theta_p, \alpha, \beta, \gamma$

Output: Web Service Graph $G = (V, E)$

- 1: $V = \emptyset, E = \emptyset;$
 - 2: **for** $i=1$ to N **do**
 - 3: **if** $S_i^H \geq \theta_H$ or $S_i^P \geq \theta_p$ **then**
 - 4: add WS_i to $V;$
 - 5: **end if**
 - 6: **end for**
 - 7: **for** each node in V **do**
 - 8: $Score_{u,i} = \alpha S_i^H + \beta S_i^P + \gamma U_{u,i};$
 - 9: **end for**
 - 10: **for** each pair of nodes v_i and v_j in V **do**
 - 11: **if** $wsSim(WS_i, WS_j) \geq \tau$ **then**
 - 12: add edge (v_i, v_j) to $E;$
 - 13: **end if**
 - 14: **end for**
 - 15: **return** $G = (V, E);$
-

Definition 2. Expanded Set:

Let S be the subset of nodes in the Web service graph $G = (V, E)$, the expanded set of S is $N(S)$ such that $N(S) = S \cup \{v \in (V - S) | \exists u \in S, (u, v) \in E\}$, where the symbol “-” in $V - S$ denotes the set minus operator.

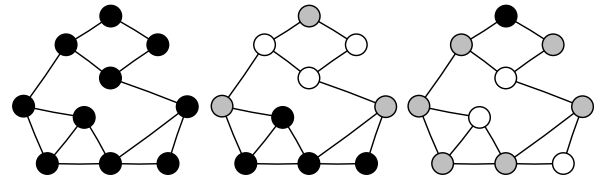
Definition 3. Expanded Ratio:

The expansion of set S is defined as $|N(S)|$, where $N(S)$ is defined in Def. 2, and $|N(S)|$ denotes the cardinality of $N(S)$. The expansion ratio of S is defined as $\sigma = |N(S)|/K$.

It is worth mentioning that the definition of expansion ratio is based on the topological structure of the Web service graph. Intuitively, the larger expansion ratio of a set of nodes implies that the nodes are more scattered in the graph, and thus resulting in a better diversity. Here, the intuition behind is that two nodes are dissimilar if they do

not share the common neighbors in the Web service graph. Therefore, the definition of expansion ratio can be considered as a diversity measure [41].

With Def. 2 and 3, we can infer that a set of nodes with larger expansion ratio are more dissimilar to each other. Given a Web service graph in Figure 2(a), let's select three nodes from it. Figure 2(b) and Figure 2(c) are two different cases, where white nodes represent the nodes selected. There are six nodes (white nodes and gray nodes) in the expanded set for the case in Figure 2(b), and nine nodes for the case in Figure 2(c). We can infer that the expansion ratio of the selected nodes in Figure 2(b) and Figure 2(c) are 0.6 and 0.9, respectively. The selected nodes in Figure 2(b) are well connected, thus they are probably similar to one another. On the contrary, there is no edge between any two selected nodes in Figure 2(c). Therefore, the selected nodes in Figure 2(c) are more diverse than those in Figure 2(b). This example shows that nodes with a larger expansion ratio have better diversity. Our diversified ranking measure is on the above definition.



(a) A Web Service Graph G (b) $\sigma = 0.6$ (c) $\sigma = 0.9$

Fig. 2. Illustration of Expansion Ratio v.s. Diversity (white nodes denote the selected nodes and gray nodes are the expanded nodes)

4.5 Diversified Web Service Ranking

We use expansion ratio to evaluate the diversity of Web services. Based on this diversity measure, a comprehensive measure for ranking Web services and an efficient greedy algorithm for finding top- k optimal Web services are proposed in this section. We aim to find the k nodes (denoted by S) in a Web service graph that have the highest scores and the largest expansion ratio $|N(S)|/K$. Formally, our goal is to maximize the following comprehensive ranking measure.

$$F(S) = (1 - \lambda) \sum_{v \in S} Score_v + \lambda \frac{|N(S)|}{K} \quad (15)$$

where $Score_v$ denotes the score of node v , and $\lambda \in [0,1]$ is a tunable parameter that is used to tradeoff the score and the diversity. The first term in Formula (15) is the sum of scores over the selected nodes, which reflects the functional relevance and QoS utility of the selected services. The second term is the expansion ratio of the selected nodes. As discussed before, larger expansion ratio indicates better diversity. Hence, Formula (15) captures both the score (i.e., the combination of functional relevance and QoS utility) and the diversity of the selected services.

However, Formula (15) does not tell us how to find the optimal k services in the Web service graph. We proposed an algorithm that can address the above issue well. In essence, the problem of finding top- k diversified and optimal Web services on a Web service graph can be formalized as Formula (16), which aims to maximize the non-

decreasing modular function with a cardinality constraint.

$$\begin{aligned} \arg \max_{S \subseteq V} F(S) \\ \text{s. t. } |S| = k \end{aligned} \quad (16)$$

Let $\lambda = 1$ in Formula (15), then the problem of finding the top-k diversified Web services is equivalent to the maximal expansion problem defined in [51] which is known to be NP-hard. As a consequence, our problem defined in Formula (16) is also NP-hard. However, the proposed measure $F(S)$ is a non-decreasing sub-modular function, which will be proved later. The definition of non-decreasing sub-modular is presented in Def. 4. One can also refer to [52].

Definition 4. Nondecreasing Sub-modular Set Function:

Let V be a finite set, a real valued function $f(S)$ on the set of subsets of V , S , is called a nondecreasing sub-modular set function, if the following conditions hold.

- ◇ **Nondecreasing:** For any subsets S and T of V such that $S \subseteq T \subseteq V$, we have $f(S) \leq f(T)$.
- ◇ **Sub-modularity:** Let $\rho_v(S) = f(S \cup \{v\}) - f(S)$ be the marginal gain. Then, for any subsets S and T of V such that $S \subseteq T \subseteq V$ and $j \in V \setminus T$, we have $\rho_v(S) \geq \rho_v(T)$.

We prove that Formula (15) is a nondecreasing sub-modular function with $F(\emptyset) = 0$, where \emptyset is an empty set. We state the theorem as follows.

Theorem 1: The set function $F(S)$ defined in Formula (15) is a nondecreasing sub-modular function with $F(\emptyset) = 0$, where \emptyset denotes an empty set.

Proof: For $\forall S \subseteq T \subseteq V$ and $v \in V \setminus T$, let $\rho_v(S) = F(S \cup \{v\}) - F(S)$, and $\rho_v(T) = F(T \cup \{v\}) - F(T)$. We have

$$\begin{aligned} \rho_v(T) &= (1 - \lambda)Score_v + \lambda \frac{|N(T \cup \{v\})| - |N(T)|}{K} \\ &= (1 - \lambda)Score_v + \lambda \frac{|N(\{v\}) - N(T)|}{K} \\ &\geq 0. \end{aligned}$$

Note that the nondecreasing property of $F(S)$ can be guaranteed by $\rho_v(T) \geq 0$. Similarly, we have $\rho_v(S) = (1 - \lambda)Score_v + \lambda \frac{|N(\{v\}) - N(S)|}{K} \geq 0$. By definition, we have $F(\emptyset) = 0$ and $|N(\{v\}) - N(S)| \geq |N(\{v\}) - N(T)|$. Hence, we conclude $\rho_v(S) \geq \rho_v(T) \geq 0$. This completes the proof.

Many existing diversified ranking measures exhibit sub-modularity [52], resulting in an efficient greedy algorithm in polynomial time with $1 - 1/e$ approximation ratio to maximize it. Since $F(S)$ is a sub-modular function, the top-k diversified ranking problem can be approximately solved by an efficient greedy algorithm. In the following, we present our greedy algorithm for solving the diversified service ranking problem as Algorithm 3.

Algorithm 3 Diversified Web Service Ranking

Input: Web Service Graph $G = (V, E)$, parameter λ , adjacency matrix A

Output: A set S of k ranked Web services

- 1: $S = \emptyset$;
- 2: **while** $|S| \leq k$ **do**
- 3: find $v_{max} = \arg \max_{v \in (V-S)} (1 - \lambda)Score_v + \frac{\lambda}{K} |N\{v\} - N(S)|$;
- 4: $S = S \cup \{v_{max}\}$;
- 5: **end while**
- 6: **return** S ;

We use the Web service graph constructed in Algorithm 2 as the input of Algorithm 3. The algorithm chooses

es a node with maximum marginal gain $\rho_v(S) = (1 - \lambda)Score_v + \frac{\lambda}{K} |N\{v\} - N(S)|$ at each iteration (line 3), and adds it into the output set S (line 4). To get the top-k ranking list, this procedure will repeat k times (line 2-5). Algorithm 3 will eventually produce an ordered ranking list according to $\rho_v(S)$. Since $\rho_v(S)$ satisfies the non-decreasing properties, Algorithm 3 is indeed a reasonable ranking procedure that the node with high ranking score will appear in the top ranking list. The time complexity of Algorithm 3 is $O(K|E|)$. Theoretically, the greedy algorithm (Algorithm 3) is a $(1 - 1/e)$ approximation algorithm for the top-k diversified ranking problem. This can be proved by similar argument that has been used to prove the approximation factor of the greedy algorithm for sub-modular set function maximization problem [52]. And it is worth mentioning that the $(1 - 1/e)$ approximation factor is quite tight [53]. In other words, there are no other polynomial-time algorithm that can obtain a more tight approximate factor unless $P=NP$.

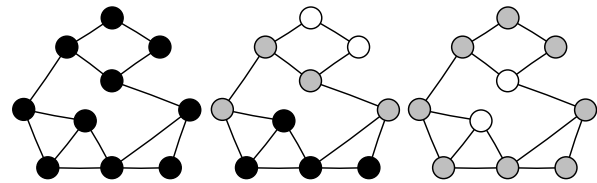
The proposed diversified ranking measure ($F(S)$) in Def. 3, only considers the immediate neighborhood information of S . Naturally, we can generalize the diversified ranking measure $F(S)$ by taking k-hop nearest neighbors into account. We call such a measure a generalized diversified ranking measure and denote it by $F_k(S)$. In the following, we give the definitions of k-hop expanded set and k-hop expansion ratio.

Definition 5. k-hop Expanded Set:

Let S be the subset of nodes in a Web service graph $G = (V, E)$, the k-hop expanded set of S is $N_k(S)$ such that $N_k(S) = S \cup \{v \in (V - S) | \exists u \in S, d(u, v) \leq k\}$, where $d(u, v)$ denotes the length of shortest path from u to v . The symbol “-” in $V - S$ denotes the set minus operator.

Definition 6. k-hop Expanded Ratio:

The k-hop expansion of set S is the cardinality of the k-hop expanded set denoted as $|N_k(S)|$. And, the k-hop expansion ratio of S is defined as $\sigma_k = |N_k(S)|/K$.



(a) A Web Service Graph G (b) $\sigma = 0.6$ (c) $\sigma = 1.0$

Fig. 3. Illustration of 2-hop Expansion Ratio v.s. Diversity (white nodes denote the selected nodes and gray nodes are the expanded nodes)

Consider the Web service graph in Figure 3(a) and select two nodes from it. Figure 3(b) and Figure 3(c) are two different cases, where white nodes represent the nodes selected. There are six nodes (white nodes and gray nodes) in the 2-hop expanded set for the case in Figure 3(b), and ten nodes in the 2-hop expanded set for the case in Figure 3(c). Then, the 2-hop expansion ratio of the selected nodes in Figure 3(b) and Figure 3(c) are 0.6 and 1.0, respectively. The selected nodes in Figure 3(b) are well connected, thus they are probably similar to one another. As a contrast, there is no edge between the two selected nodes in Figure 3(c).

Therefore, the selected nodes in Figure 3(c) are more diverse than the selected nodes in Figure 3(b).

Based on the k-hop expansion, we define the generalized diversified ranking measure $F_k(S)$ as follows.

$$F_k(S) = (1 - \lambda) \sum_{v \in S} Score_v + \lambda \frac{|N_k(S)|}{K} \quad (17)$$

Obviously, $F(S)$ is a special case of $F_k(S)$ when $k = 1$. Like $F(S)$, $F_k(S)$ is also a nondecreasing sub-modular function. We give a theorem as follows. The proof is similar to the proof of the Theorem 1, thus we omit it for brevity.

Theorem 2: The set function $F_k(S)$ defined in Formula (17) is a nondecreasing sub-modular function with $F_k(\emptyset) = 0$, where \emptyset denotes an empty set.

Likewise, the problem of maximizing the set function $F_k(S)$ subject to a cardinality constraint is NP-hard. However, based on the sub-modularity property of $F_k(S)$, we can develop a greedy algorithm to optimize it accurately similar with Algorithm 3. While, now the problem is that the greedy algorithm gain $\rho_v(S) = (1 - \lambda)Score_v + \frac{\lambda}{K}|N_k\{v\} - N_k(S)|$ in each iteration. Therefore, our approach is scalable to any k-hop expansion. In this paper, we mainly focus on the 1-hop expansion, which is evaluated in our experiments presented in Section 5.

5 PERFORMANCE EVALUATION

In this section, we report the performance study of our proposed approach for Web service recommendation. Four parts are included: i) comparing our approach with the other Web service recommendation approaches including collaborative filtering approach, content-based recommendation approach, and their hybrid approach; ii) comparing our approach with the state-of-the-art diversified ranking methods; iii) precision evaluation; iv) studying the sensitivity of our approach under different tradeoffs of different parameters. We first describe the Web service dataset collected for the experiments and then report the experimental results.

5.1 Dataset Setup

To obtain solid experimental results, it is ideal to use a real world Web service dataset. Zheng et al. [29] published a large-scale real world Web service dataset acquired in their WS-DREAM project¹. WS-DREAM is a Web service crawling engine that collects publicly available WSDL file addresses from the Internet. It also collected QoS information of these Web services by using 339 distributed computers to monitor the Web services. This dataset has been widely used for performance evaluation by previous work on Web service recommendation.

In our experiments, we used this dataset as our base dataset. Some Web services in the dataset are unavailable now, thus we only choose the available ones to form a new dataset which initially contains QoS data of 1982 Web services. The characteristics of the processed dataset are described in Table 1. There are 339 users and 1982 Web services in our dataset. The behavior of Web services in each invocation, as well as the observed QoS perfor-

mance (response time and throughput) were recorded in our dataset.

TABLE 1
Characteristics of the Dataset

Parameters	values
Number of users	339
Number of services	1982
Effective invocations	637595
Average effective invocations per user	101
Observed QoS quality	Response time, throughput

For all the 1982 Web services in our dataset, both response time and throughput were employed in our experiments. The values observed by 339 users on the 1982 Web services are presented in Figure 4. Since Web services in the dataset may have different response time and throughput values for different users, for consistency, we use the average of the response time and throughput values of each Web service as its QoS.

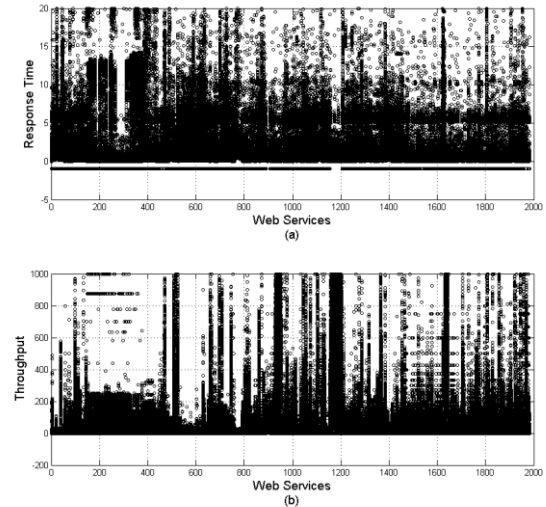


Fig. 4. QoS Values of Web services in the Processed Dataset (a) response time and (b) throughput

In order to simulate the variety of QoS preferences from different users, we divide the 339 users into 5 groups. Users in different groups are supposed to have different weights on response time w_{rt} and throughput w_{tp} for the Web services. The user partition is presented in Table 2.

TABLE 2
QoS Preferences of Users

Users	w_{rt}	w_{tp}
Group 1 (68 users)	0.0~0.2	$1 - w_{rt}$
Group 2 (68 users)	0.2~0.4	$1 - w_{rt}$
Group 3 (68 users)	0.4~0.6	$1 - w_{rt}$
Group 4 (68 users)	0.6~0.8	$1 - w_{rt}$
Group 5 (67 users)	0.8~1.0	$1 - w_{rt}$

5.2 Evaluation Metrics

In our experiments, we randomly choose 10 service users from the dataset as the active (test) users. We report

¹ <http://www.wsdream.net>
1939-1374 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

the average performance evaluation from several aspects.

In our Web service recommendation approach, score and diversity are incorporated. The score involves the functional relevance (w.r.t user interest) and non-functional relevance (w.r.t user QoS requirements) of Web services. Thus, in our experiments, we mainly evaluate the performance on the total score, diversity of the recommendation list, and the overall diversified ranking measure defined in Formula (15). The precision metric will be presented later in Section 5.5. When comes to weighted summation, we do normalization processing with similar method to Formula (7) before weighted summation for each factor.

We employ two metrics to measure the diversity. One is proposed in [46], which makes use of the density of the induced sub-graph by the top-k ranked nodes. The density of a graph is defined as the number of edges (excluding self-links) presenting in the graph divided by the maximal possible number of edges in the graph. The density of sub-graph S formed by the top-k ranked nodes is defined as Formula (18). Intuitively, the density inversely measures the diversity of the top-k ranked nodes, i.e., smaller density means larger diversity.

$$d(S) = \frac{|{(u,v)|u \in S, v \in S, (u,v) \in E}|}{|S| \times (|S|-1)} \quad (18)$$

The second metric is the expansion ratio given in Def. 3. The rational is that the larger expansion ratio of the top-k ranking nodes indicates the better diversity.

Here, we present the parameter settings and experimental environment. In our proposed approach, there are several parameters: Num , ψ_1 , ψ_2 , θ_H , θ_P , α , β , γ , λ , and τ . We set $Num = 10$, $\psi_1 = \psi_2 = 0.5$, $\theta_H = \theta_P = 0.6$, $\alpha = \beta = 0.4$, $\gamma = 0.2$, $\lambda = 0.5$, and τ equals to the average similarity calculated with Formula (19). Therefore, in our experiments, we take the score and diversity as equally important factors. While we also test the impact of variance of parameters in our approach. All the experiments are conducted on a 2.50 GHz CPU and 4 GB PC running Windows 7. All algorithms are implemented by MATLAB 2009a and Visual C++ 6.0.

$$\tau = \frac{2 * \sum S_{i,j}^{WS}}{N(N-1)} \quad (19)$$

5.3 Comparison with Other Service Recommendation Approaches

Baselines: We compare our proposed approach with the Content-based Web service recommendation approach (rank Web services according to their historical user interest relevance and QoS utility), CF-based Web service recommendation approach (rank Web services according to their potential user interest relevance and QoS utility), and Hybrid approach (rank Web services according to the combination of historical user interest relevance, potential user interest relevance, and QoS utility) under the diversity, score, and the ranking measurement defined before.

5.3.1 Diversity Evaluation

In this section, we conduct experiments to study the performance of our approach with the diversity metric and make comparison with its competitors.

As shown in Figure 5(a), WSRD achieves the best diversi-

ty under the density metric, followed by the CF-based approach, then Hybrid and Content-based approaches. Therefore, we can conclude that Hybrid and Content-based approaches exhibit poor performance in density evaluation. With expansion ratio measure, WSRD also outperforms the other competitors, which can be seen from Figure 5(b). The expansion ratio of CF-based and Hybrid approaches are comparatively close. When k becomes larger, the effect is more obvious. Content-based approach achieves the worst expansion ratio. Based on the above observations, we can conclude that WSRD achieves the best diversity performance than its competitors.

5.3.2 Functional and Non-functional Evaluation

The score defined in Formula (14) describes the user interest relevance and QoS utility of Web service candidates, so it can be used to evaluate the functional and non-functional quality of recommended Web services. In this subsection, we conduct experiments to study the performance of WSRD with the overall score of Web services in the recommendation list compared with its competitors.

As shown in Figure 5(c), WSRD, Content-based, CF-based, and Hybrid approaches achieve very similar overall scores for their recommendation lists. Our approach is slightly worse than the Hybrid approach, with a very small difference so that it can be neglected. Content-based approach is slightly worse than WSRD when k is larger than 50. The CF-based approach has the most similar performance as WSRD.

5.3.3 Overall Evaluation

The diversified ranking measure defined in Formula (15) captures the functional, non-functional and diversity feature of Web services in a recommendation list, so it evaluates the overall quality of a Web service recommendation list.

In Figure 5(d), WSRD achieves the best performance under the diversified ranking measure, followed by the CF-based approach, then Hybrid and Content-based approaches. Content-based approach shows the worst performance under the diversified ranking measure. Therefore, WSRD outperforms the competitors when both the diversity and the score of the Web service recommendation list are taken into consideration.

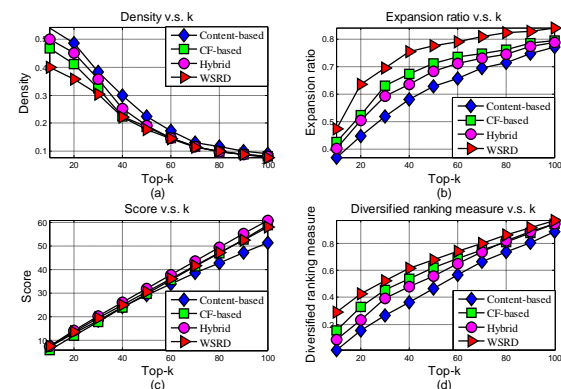


Fig. 5. Comparison of Web Service Recommendation Approaches

5.4 Comparison with Other Diversified Ranking Methods

Baselines: We compare our proposed algorithm with three diversified ranking methods in graph domain again with the diversity, score, and the overall ranking measurement as evaluation metrics. The three baselines are: (1) Grasshopper: Grasshopper is a ranking algorithm that leverages an absorbing random walk to achieve diversity [44]; (2) Manifold ranking with stop points (Mani_stop): The Mani_stop algorithm was proposed in [45], which is very similar to the Grasshopper algorithm; (3) DivRank: The DivRank makes use of the stationary distribution of a vertex reinforced random walk to rank nodes [46].

In this experiment, we implement the proposed algorithm and compare with three baselines described above over our dataset. From Figure 6(a), we can see that WSRD achieves the best diversity under the density metric, followed by Mani_stop algorithm, and then Grasshopper and DivRank algorithms. There is a large gap between WSRD and Grasshopper, implying that the Grasshopper and DivRank algorithm exhibit poor performance to enhance diversity in our Web service dataset. And Mani_stop shows a little worse diversity than the WSRD. Under the expansion ratio measure (i.e., Figure 6(b)), WSRD outperforms the competitors. Grasshopper and DivRank achieve nearly equivalent diversity. And the Mani_stop algorithm performs the worst. For the score (i.e., Figure 6(c)), we can observe that the best algorithm is Grasshopper, followed by DivRank, WSRD and Mani_stop. In terms of this observation, improving diversity will reduce the score. Hence, in practice, we should seek a reasonable tradeoff between score and diversity. In this experiment, we find that WSRD can achieve this end, as it considerably improves diversity but do not significantly sacrifice the performance of the score. As shown in Figure 6(d), WSRD achieve the best performance under the diversified ranking measure, followed by Grasshopper, then DivRank and Mani_stop. And there is a big gap between WSRD and its competitors. Therefore, WSRD outperforms the competitors in the overall diversified ranking.

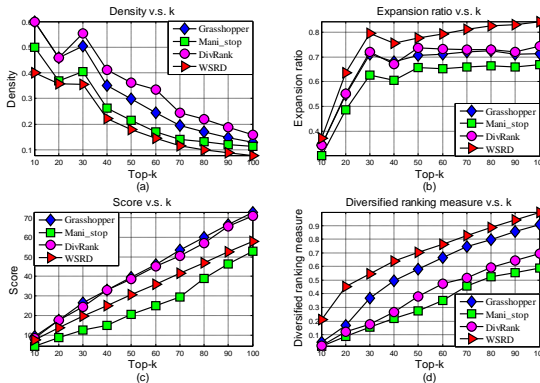


Fig. 6. Comparison of Various Diversified Ranking Methods

5.5 Precision Evaluation

To further evaluate the effectiveness of our approach, we evaluated the precision of our approach and compared it with the state-of-the-art service recommendation ap-

proach—CF based approach as shown in Figure 5(d), and the state-of-the-art diversified ranking method — Grasshopper as shown in Figure 6(d). Since there is no ground truth in Web service datasets, we use the Hybrid approach as the ground-truth rank. The precision is defined by Formula (20).

$$Pre = |S \cap \tilde{S}| / |\tilde{S}| \quad (20)$$

where S denotes the set of services in the top- k diversified ranking list produced by the service recommendation approach or diversified ranking method, and \tilde{S} denotes the set of services in the top- k ranking list by Hybrid service recommendation approach which always yields the k most relevant services as can be seen from Figure 5(c).

In this experiment, we select three groups of 10 active users to observe the average precision (i.e., each group with 10 active users). Figure 7 shows the results. From Figure 7(a), we can clearly see that WSRD consistently outperforms CF-based approach and Grasshopper in the experiment of group 1. In group 2, we can observe that three approaches generate comparable rank, and the performance of WSRD is slightly better than Grasshopper; while CF-based approach is the worst. In group 3, Grasshopper does not perform well, and CF-based approach is in the middle. In contrast, the performance of our approach is very stable in three groups of experiments. With the above observation, we can conclude that WSRD is better than CF-based approach and Grasshopper.

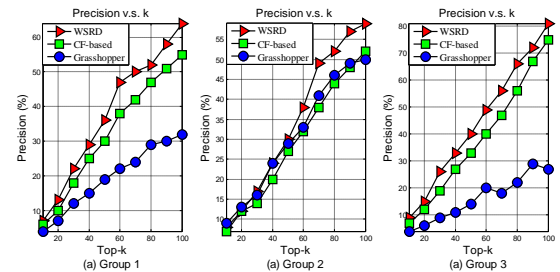


Fig. 7. Comparison of Precision of WSRD, CF-based approach, and Grasshopper

5.6 Impacts of Parameters

In this section, we conduct experiments to study the effects of parameters in our approach. Specifically, we study the impacts of α , β , γ , and λ on our approach. When we study the effects of α , β , γ , we keep λ unchanged with the default value. And When we study the effects of λ , we keep α , β , γ , unchanged with the default values.

We study the effects of parameters α , β , γ in Formula (14) to our approach, which are leveraged to tradeoff the historical user interest relevance, potential user interest relevance, and QoS utility, respectively. The results are presented in Figure 8. In this experiment, α and β are both weights to the functional relevance of Web service candidates, thus we set the equal value to them.

In Figure 8(a), we can see that the density decreases as γ increases, and the effect is especially obvious when k is small. From this observation, we can conclude that larger γ causes better diversity. However, this phenomenon is only obvious when k is less than 20 under expansion ratio evaluation, as can be seen from Figure 8(b). As for the score evaluation,

tion, we can see from Figure 8(c) that larger γ tends to cause higher score, especially when k becomes large. In Figure 8(d), larger γ tends to cause slightly better diversified ranking. And the gap is relatively large when k is less than 20. This phenomenon is very similar to Figure 8(b) when k is less than 20, since the scores are nearly the same when k is less than 20 in Figure 8(c). Theoretically, in our approach, larger γ means the score tends to be more dominated by the non-functional quality (e.g., QoS utility), so the resulting recommended Web services tends to be more dissimilar to each other (indicating better diversity). Therefore, the experimental results in Figure 8 verified the fact.

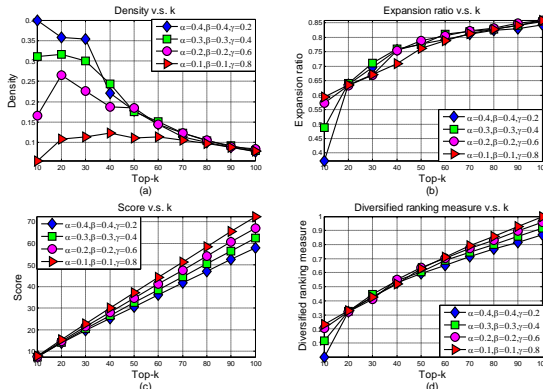


Fig. 8. The Impact of Parameter α, β, γ

Next, we study the effect of parameter λ in Formula (15), which is leveraged to tradeoff the score and diversity. The experimental results are presented in Figure 9.

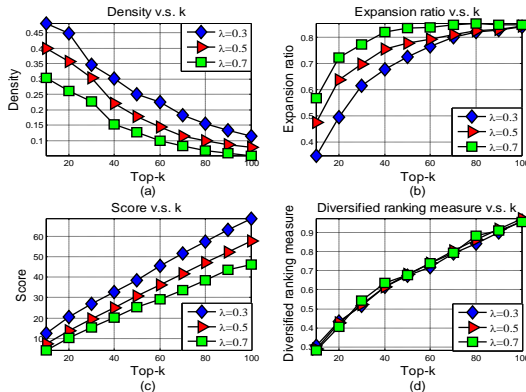


Fig. 9. The Impact of Parameter λ

In Figure 9(a), the density decreases as λ increases, since smaller density means better diversity. And in Figure 9(b), expansion ratio increases as λ increases, since larger expansion ratio means better diversity. Therefore, diversity generally increases as λ increases. This is because a larger λ means more weights are assigned to the diversity feature. In contrast, according to Figure 9(c), the total score decreases slightly as λ increases, but the decrease is trivial. From the above observations, improving the diversity will oppositely reduce the score, while the change of λ is not much sensitive to the diversified ranking measure as can be seen from figure 9(d). Hence, in practice, users can set a reasonable tradeoff between the score and the diversity

according to their preferences.

6 CONCLUSION

In this paper, we presented a Web service recommendation approach with diversity to find desired Web services for users. We incorporate functional interest, QoS preference, and diversity feature for recommending top-k diversified Web services. A diversified Web service ranking algorithm is proposed to find the top-k diversified Web service ranked list based on their functional relevance including historical user interest relevance and potential user interest relevance, non-functional relevance such as QoS utility, and diversity feature. Experimental results on a real world Web service dataset show that the proposed approach improves the Web service recommendation performance in terms of diversity, the combination of functional relevance and QoS utility, and the diversified ranking evaluation.

In future work, we will study Web service clustering methods to improve the similarity computation and conduct real user survey to evaluate the usefulness of our method further. In addition, our proposed diversified ranking measure ($F(S)$) mainly focuses on the immediate neighborhood information of S in the Web service graph. More tests will be performed by our diversified ranking measure with k-hop nearest neighbors in the future work.

ACKNOWLEDGMENT

The authors appreciate the anonymous reviewers for their extensive and informative comments for the improvement of this manuscript. The work was supported by the National NSFC (No.61272063 and No.61100054), Program for New Century Excellent Talents (No.NCET-10-0140), Excellent Youth Foundation of Hunan Scientific Committee (No.11JJ1011), Scientific Research Fund of Hunan Educational Committee (No.11B048).

REFERENCES

- [1] L. Zhang, J. Zhang, H. Cai. Services computing. *Tsinghua University Press*, Beijing, 2007.
- [2] Y. Jiang, J. Liu, M. Tang, X. Liu. "An effective Web service recommendation based on personalized collaborative filtering". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 211-218, 2011.
- [3] M. Tang, Y. Jiang, J. Liu, X. Liu. "Location-Aware Collaborative Filtering for QoS-Based Service Recommendation". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 202-209, 2012.
- [4] X. Chen, X. Liu, Z. Huang, H. Sun. "RegionKNN: a scalable hybrid collaborative filtering algorithm for personalized Web service recommendation". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 9-16, 2010.
- [5] Z. Zheng, H. Ma, M. R. Lyu, I. King. "Wsrec: a collaborative filtering based web service recommender system". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 437-444, 2009.
- [6] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, H. Mei. "Personalized qos prediction for web services via collaborative filtering".

- Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 439-446, 2007.
- [7] M. Gong, Z. Xu, L. Xu, Y. Li, L. Chen. "Recommending Web Service Based on User Relationships and Preferences". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 380-386, 2013.
- [8] S. M. Mcnee, J. Riedl, J. A. Konstan. "Being accurate is not enough: how accuracy metrics have hurt recommender systems". *Proceedings of CHI'06 extended abstracts on Human factors in computing systems*. ACM, pp. 1097-1101, 2006.
- [9] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, Y.-C. Zhang. "Solving the apparent diversity-accuracy dilemma of recommender systems". *Proceedings of the National Academy of Sciences*, Vol. 107, No. 10, pp. 4511-4515, 2010.
- [10] C.-N. Ziegler, S. M. Mcnee, J. A. Konstan, G. Lausen. "Improving recommendation lists through topic diversification". *Proceedings of Proceedings of the 14th international conference on World Wide Web*. ACM, pp. 22-32, 2005.
- [11] B. Smyth, P. McClave. "Similarity vs. diversity". *Case-Based Reasoning Research and Development* Springer, pp. 347-361, 2001.
- [12] N. Hurley, M. Zhang. "Novelty and diversity in top-n recommendation--analysis and evaluation". *ACM Transactions on Internet Technology (TOIT)*, Vol. 10, No. 4, pp. 14, 2011.
- [13] S. S. Yau, Y. Yin. "QoS-Based Service Ranking and Selection for Service-Based Systems". *Proceedings of International Conference on Service Computing*. IEEE Computer Society, pp. 56-63, 2011.
- [14] W. Lin, W. Dou, X. Luo, J. Chen. "A history record-based service optimization method for QoS-aware service composition". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 666-673, 2011.
- [15] G. Kang, J. Liu, M. Tang, X. F. Liu, K. K. Fletcher. "Web Service Selection for Resolving Conflicting Service Requests". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 387-394, 2011.
- [16] L. Qi, Y. Tang, W. Dou, J. Chen. "Combining local optimization and enumeration for QoS-aware Web service composition". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 34-41, 2010.
- [17] M. Alrifai, D. Skoutas, T. Risse. "Selecting skyline services for QoS-based web service composition". *Proceedings of International World Wide Web Conference*. ACM, pp. 11-20, 2010.
- [18] M. Klusch, B. Fries, K. Sycara. "Automated semantic web service discovery with OWLS-MX". *Proceedings of Autonomous Agents and Multiagent Systems*. ACM, pp. 915-922, 2006.
- [19] D. Lee, J. Kwon, S. Lee, S. Park, B. Hong. "Scalable and efficient web services composition based on a relational database". *Journal of Systems and Software*, Vol. 84, No. 12, pp. 2139-2155, 2011.
- [20] D. Ardagna, R. Mirandola. "Per-flow optimal service selection for Web services based processes". *Journal of Systems and Software*, Vol. 83, No. 8, pp. 1512-1523, 2010.
- [21] J. Wu, L. Chen, Z. Zheng, M. R. Lyu, Z. Wu. "Clustering Web services to facilitate service discovery". *Knowledge and Information Systems*, Vol. No., pp. 1-23, 2012.
- [22] G. Kang, J. Liu, M. Tang, X. Yu. "An Effective Dynamic Web Service Selection Strategy with Global Optimal QoS Based on Particle Swarm Optimization Algorithm". *Proceedings of 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW 2012)*. IEEE, pp. 1939-1944, 2012. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
- Computer Society, pp. 2274-2279, 2012.
- [23] G. Kang, J. Liu, M. Tang, B. Cao. "Web service selection algorithm based on principal component analysis". *Journal of Electronics (China)*, Vol. 30, No. 2, pp. 1-9, 2012.
- [24] Y. Liu, A. H. Ngu, L. Z. Zeng. "QoS computation and policing in dynamic web service selection". *Proceedings of In Proceedings of the International World Wide Web Conference*. ACM, pp. 66-73, 2004.
- [25] S. Wang, C.-H. Hsu, Z. Liang, Q. Sun, F. Yang. "Multi-user web service selection based on multi-QoS prediction". *Information Systems Frontiers*, Vol. 16, No. 1, pp. 1-10, 2013.
- [26] D. Goldberg, D. Nichols, B. M. Oki, D. Terry. "Using collaborative filtering to weave an information tapestry". *Communications of the ACM*, Vol. 35, No. 12, pp. 61-70, 1992.
- [27] J. S. Breese, D. Heckerman, C. Kadie. "Empirical analysis of predictive algorithms for collaborative filtering". *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 43-52, 1998.
- [28] B. Sarwar, G. Karypis, J. Konstan, J. Reidl. "Item-based collaborative filtering recommendation algorithms". *Proceedings of the 10th international conference on World Wide Web*. ACM, pp. 285-295, 2001.
- [29] Z. Zheng, H. Ma, M. Lyu, I. King. "QoS-aware web service recommendation by collaborative filtering". *IEEE Transactions on Services Computing*, Vol. 4, No. 2, pp. 140-152, 2011.
- [30] Z. Li, Z. Bin, L. Ying, G. Yan, Z. Zhi-Liang. "A Web Service QoS Prediction Approach Based on Collaborative Filtering". *Proceedings of IEEE Asia-Pacific Services Computing Conference*. IEEE, pp. 725-731, 2010.
- [31] W. Lo, J. Yin, S. Deng, Y. Li, Z. Wu. "Collaborative web service qos prediction with location-based regularization". *Proceedings of 2012 IEEE 19th International Conference on Web Services (ICWS)*. IEEE, pp. 464-471, 2012.
- [32] J. Wu, L. Chen, Y. Feng, Z. Zheng, M. C. Zhou, Z. Wu. "Predicting quality of service for selection by neighborhood-based collaborative filtering". *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 43, No. 2, pp. 428-439, 2013.
- [33] Y. Xu, J. Yin, W. Lo, Z. Wu. "Personalized Location-Aware QoS Prediction for Web Services Using Probabilistic Matrix Factorization". *Proceedings of Web Information Systems Engineering-WISE 2013*. Springer, pp. 229-242, 2013.
- [34] Z. Zheng, H. Ma, M. Lyu, I. King. "Collaborative Web service QoS prediction via neighborhood integrated matrix factorization". *IEEE Transactions on Service Computing*, Vol. 6, No. 3, pp. 289-299, 2013.
- [35] G. Kang, J. Liu, M. Tang, X. Liu, B. Cao, Y. Xu. "AWSR: Active Web Service Recommendation Based on Usage History". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 186-193, 2012.
- [36] L. Liu, F. Lecue, N. Mehandjiev. "Semantic content-based recommendation of software services using context". *ACM Transactions on the Web (TWEB)*, Vol. 7, No. 3, pp. 17-20, 2013.
- [37] R. Hu, W. Dou, J. Liu. "A personalised search approach for web service recommendation". *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 13, No. 2, pp. 83-95, 2013.
- [38] F. Lécué. "Combining collaborative filtering and semantic content-based approaches to recommend web services". *Proceedings of Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*. IEEE, pp. 200-205, 2010.

- [39] L. Yao, Q. Z. Sheng, A. Segev, Yu. "Recommending Web Services via Combining Collaborative Filtering with Content-based Features". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 42-49, 2013.
- [40] L. Qin, J. X. Yu, L. Chang. "Diversifying top-k results". *the VLDB Endowment*, Vol. 5, No. 11, pp. 1124-1135, 2012.
- [41] R.-H. Li, J. X. Yu. "Scalable diversified ranking on large graphs". *Proceedings of 2011 IEEE 11th International Conference on Data Mining (ICDM)*. IEEE, pp. 1152-1157, 2011.
- [42] M. Drosou, E. Pitoura. "Search result diversification". *ACM SIGMOD Record*, Vol. 39, No. 1, pp. 41-47, 2010.
- [43] R. Agrawal, S. Gollapudi, A. Halverson, S. Jeong. "Diversifying search results". *Proceedings of the Second ACM International Conference on Web Search and Data Mining*. ACM, pp. 5-14, 2009.
- [44] X. Zhu, A. B. Goldberg, J. Van Gael, D. Andrzejewski. "Improving Diversity in Ranking using Absorbing Random Walks". *Proceedings of HLT-NAACL*. Citeseer, pp. 97-104, 2007.
- [45] X. Zhu, J. Guo, X. Cheng, P. Du, H.-W. Shen. "A unified framework for recommending diverse and relevant queries". *Proceedings of Proceedings of the 20th international conference on World wide web*. ACM, pp. 37-46, 2011.
- [46] Q. Mei, J. Guo, D. Radev. "Divrank: the interplay of prestige and diversity in information networks". *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1009-1018, 2010.
- [47] R.-H. Li, J. X. Yu. "Scalable diversified ranking on large graphs". *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 9, pp. 2133-2146, 2013.
- [48] K. S. Jones. "A statistical interpretation of term specificity and its application in retrieval". *Journal of documentation*, Vol. 28, No. 1, pp. 11-21, 1972.
- [49] Y. Zhang, Z. Zheng, M. R. Lyu. "WSEXPRESS: a QoS-aware search engine for web services". *Proceedings of International Conference on Web Services*. IEEE Computer Society, pp. 91-98, 2010.
- [50] S. Robertson. "Understanding inverse document frequency: on theoretical arguments for IDF". *Journal of documentation*, Vol. 60, No. 5, pp. 503-520, 2004.
- [51] A. S. Maiya, T. Y. Berger-Wolf. "Expansion and search in networks". *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, pp. 239-248, 2010.
- [52] G. L. Nemhauser, L. A. Wolsey, M. L. Fisher. "An analysis of approximations for maximizing submodular set functions—I". *Mathematical Programming*, Vol. 14, No. 1, pp. 265-294, 1978.
- [53] U. Feige. "A threshold of $\ln n$ for approximating set cover". *Journal of the ACM (JACM)*, Vol. 45, No. 4, pp. 634-652, 1998.



Mingdong Tang received his Ph.D. degree in computer science from the Institute of Computing Technology Chinese Academy of Sciences, China, in 2009. He is now an associate professor in School of Computer Science and Engineering, Hunan University of Science and Technology. His current interests include distributed computing, service computing and cloud computing, big data, social network. He is now a leading researcher in research of service computing and cloud computing in Hunan University of Science and Technology. He has published more than 50 papers in well-know conferences and journals, such as ICWS, SCC, JWRSR, IEEE BigData, SCIENCE CHINA Information Sciences, etc. He is also on the editorial boards of several journals and has served as a program chair for several international conferences.



Jianxun Liu received his M.S. and Ph.D. degree in computer science from the Central South University of Technology in 1997 and Shanghai Jiao Tong University in 2003, respectively. He is now a professor and a vice dean of School of Computer Science and Engineering, Hunan University of Science and Technology. His current interests include workflow management, service computing and cloud computing, semantics and knowledge grid, etc. He is the recipient of the second grade prize of both the Shanghai and Hunan Science Progress Award. He has published more than 60 papers in well-know conferences and journals, such as ICWS, SCC, Information & Management, Future Generation Computer Systems, Concurrency and Computation: Practice and Experience, etc. He served as a program committee member or program chair of the major conferences in his area (ICWS, SCC, SCA, CGC, APBPM, TrustCom, EmbededCom, PICom, DASC, SKG, etc.) and a reviewer for TKDE, TSC, TPDS, etc.



Xiaoqing(Frank) Liu received his Ph.D. degree in computer science from the Texas A&M University in 1995. He is now a professor and associate chair for graduate studies and external affairs in the computer science department at the Missouri University of Science and Technology. He was an interim chair in the computer science department in 2013. His current interests include software engineering, service computing, On-line argumentation and collaborative systems, fuzzy logic, knowledge based systems, and artificial intelligence applications. He also serves as a visiting "lotus scholar chaired professor" of Hunan province in the school of computer science at the Hunan University of Science and Technology. He has published more than 120 papers in reputable conferences and journals.



Buqing Cao received his M.S. degree in computer science from the Central South University of Technology in 2007, and Ph.D. degree from Wuhan University in 2010. He is now a lecturer in School of Computer Science and Engineering, Hunan University of Science and Technology. His current interests include service computing, social network, software engineering. He has published more than 30 papers in well-know conferences and journals, such as ICWS, SCC, JWRSR, IEEE BigData, China Communications, GCC, etc.



Guosheng Kang received his M.S. degree in computer science from Hunan University of Science and Technology. He is currently a Ph.D. candidate in School of Computer Science, Fudan University, Shanghai, China. He has been an IEEE student member since 2010. He was a research assistant at Hunan University of Science and Technology from 2011 to 2012. He has published more than 20 papers. His publications have appeared in some well-know conference proceedings and international journals, such as ICWS, ICSOC, APSCC, IEEE BigData, etc. His research interests include service computing and cloud computing, data-centric business process management, business process configuration, data mining, and social network.