Recommending Web Services via Combining Collaborative Filtering with Content-based Features

Lina Yao and Quan Z. Sheng
School of Computer Science
The University of Adelaide
Adelaide, SA 5005, Australia
{lina, qsheng}@adelaide.edu.au

Aviv Segev
Department of Knowledge Service Engineering
KAIST, Daejeon, 305-701, Korea
aviv@kaist.edu

Jian Yu
Faculty of ICT
Swinburne University of Technology
Melbourne, VIC 3122, Australia
jianyu@swin.edu.au

***Abstract*—With increasing adoption and presence of Web services, designing novel approaches for efficient Web services recommendation has become steadily more important. Existing Web services discovery and recommendation approaches focus on either perishing UDDI registries, or keyword-dominant Web service search engines, which possess many limitations such as insufficient recommendation performance and heavy dependence on the input from users such as preparing complicated queries. In this paper, we propose a novel approach that dynamically recommends Web services that fit users' interests. Our approach is a *hybrid* one in the sense that it combines collaborative filtering and content-based recommendation. In particular, our approach considers simultaneously both rating data and content data of Web services using a three-way aspect model. Unobservable user preferences are represented by introducing a set of latent variables, which is statistically estimated. To verify the proposed approach, we conduct experiments using 3,693 real-world Web services. The experimental results show that our approach outperforms the two conventional methods on recommendation performance.**

***Keywords*-Web service recommendation, collaborative filtering, content-based recommendation, three-way aspect model**

I. Introduction

After a decade of research and development, Web services have become one of the standard technologies for sharing data and software and the number of Web services available on the Internet is consistently increasing [1], [2], [3], [4]. According to recent statistics[1], there are 28,606 Web services available on the Web, provided by 7,739 different providers. This increasing adoption and presence of Web services calls for novel approaches for efficient Web services recommendation and selection, which is a fundamental issue in service oriented computing [5], [6], [7].

Web services recommendation is the process of automatically identifying the usefulness of services and proactively discovering and recommending services to end users. We can also view service recommendation as the process of service selection augmented with end user behavior analysis to achieve relevant and accurate service suggestions. Traditional Web service discovery centers around UDDI (Universal Description, Discovery and Integration) registries [8], [9]. Unfortunately, UDDI is no longer the choice for publishing Web services, evidenced by the shutdown of the public UDDI registries by big players such as IBM, Microsoft, and SAP [10]. Over the last few years, a considerable number of Web services searching approaches have been proposed [11], [12] and several Web services search engines, such as *Web Service List*[2], *XMethods*[3], and *seekda*[4], have emerged. These search engines largely exploit keyword-based search techniques and are insufficient to catch the functionalities of Web services. Furthermore, considerations on non-functional characteristics (e.g., quality of service) of Web services during the service selection and recommendation are very limited [13]. In a recent work by Zheng et al. [5], [12], a Web services search engine is designed and developed that ranks Web services not only by functional similarities to a user's query, but also by non-functional QoS characteristics of Web services.

The main goal of our work is to advance the current state-of-the-art on Web services selection and recommendation. More specifically, our work is inspired by the following observations. To find desirable Web services by using Web service search engines, a user normally has to execute queries herself and is often at a loss as to what queries are appropriate (e.g., which keywords should be used, what values should be set for a QoS attribute). Another problem is that Web services that do not satisfy user's searching query are completely excluded from the recommendation list. It is therefore desirable that a recommendation system selects Web services that people probably prefer by estimating user preferences without requiring users to explicitly specify queries.

In this paper, we propose a novel approach for Web service recommendation by combining *collaborative filter-*

[1]http://webservices.seekda.com, as of 07/01/2013.

[2]http://www.webservicelist.com.
[3]http://www.xmethods.net.
[4]http://webservices.seekda.com.

978-0-7695-5025-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ICWS.2013.16

ing [14], [15], [5], [16] and *content-based recommendation* [17], [18]. Collaborative filtering is a technique widely used for recommending items to a user by considering other similar users' ratings on the items. For instance, suppose that a user likes Web services s_a and s_b. If there are many other users who like s_a and s_b also like service s_c, s_c should probably be recommended to the user. Although the technique is effective, a big problem is that Web services without ranking information (e.g., newly deployed Web services) cannot be recommended (also known as the *cold start* problem). Content-based methods recommend Web services based on the similarity of user preferences and content of Web services (e.g., functionalities). Unrated Web services can be recommended by this technique. Unfortunately, associating user preferences with Web service content is not a trivial task and very few solutions have been proposed. In current Web services search engines, queries that represent user preferences are typically prepared by users. Our approach exploits the advantages of both techniques by proposing a hybrid method that considers both rating and content information of Web services. The main contribution of our work is as follows:

- We identify three main requirements that are important for conducting an effective Web services recommendation,
- We propose a novel hybrid approach that combines collaborative filtering and semantic content-based methods. Our approach exploits a *three-way aspect* model that simultaneously considers the similarities of users and content of Web services. User preferences are represented using a set of latent variables that can be statistically estimated, and finally
- We conduct extensive experiments using real-world Web services to verify the proposed approach. A dataset[5] consisting 5,825 Web services is carefully examined and 3,693 live Web services are selected and used in the experiments. The experimental results show that our approach achieves better recommendation performance than the conventional collaborative filtering and content-based methods.

The remainder of the paper is organized as follows. Section II discusses Web service recommendation requirements and overviews two complementary recommendation approaches. Section III introduces our hybrid Web services recommendation approach. Section IV reports our experimental results. Finally, Section V overviews the related work and Section VI offers some concluding remarks.

II. Web Services Recommendation

In this section, we first discuss the requirements on Web services recommendation, and then briefly introduce two typical recommendation approaches, collaborative filtering and content-based recommendation, which will be used for comparison with our proposed approach in Section IV.

A. Requirements in Service Recommendation

There are three main requirements in order to conduct an effective service recommendation task:

- *High recommendation accuracy.* A good recommendation system should recommend more favorite Web services and fewer disliked ones, particularly in the situations where available information might be not sufficient (e.g., missing QoS of some services).
- *Recommendation diversity.* Recommending services that are well-known to a user is often found unsatisfactory or meaningless. If the recommended services are unfamiliar to a user, the chances of finding new Web services that match the user's requirements would increase.
- *Overcoming the cold-start problem.* Solving this problem not only enables users to find newly-deployed Web services, but also enhances the recommendation diversity.

Our approach will unify both methods for effective services recommendation. In the rest of this section, we will briefly introduce collaborative filtering and content-based recommendation.

B. Collaborative Filtering

Collaborative filtering predicts rating scores of a user for Web services by considering other users' rating on the services. A widely used approach employs Pearson Correlation Coefficient to calculate similarities between users and predicts QoS value based on similar users [19], [5]. In [20], a model combining latent features and memory-based QoS prediction is proposed to enhance the prediction performance. Based on the predicted QoS values, the Web service with the best score or the top n Web services are selected for the recommendation. The probability of a service s being recommended to a user u can be calculated using this method as:

$$\hat{y}_{u,s} = \frac{\sum_{u' \in \mathcal{U}} w_{u,u'} y_{u',s}}{\sum_{u' \in \mathcal{U}} w_{u,u'}} \tag{1}$$

where $y_{u',s}$ is the estimated value, and $w_{u,u'}$ measures the preference similarity of users u and u', using the following formula:

$$w_{u,u'} = \frac{\sum_{s \in \mathcal{S}} (r_{u',s} - \overline{r'_u})(r_{u,s} - \overline{r_u})}{\sqrt{\sum_{s \in \mathcal{S}} (r_{u',s} - \overline{r'_u})^2} \sqrt{\sum_{s \in \mathcal{S}} (r_{u,s} - \overline{r}_u)^2}} \tag{2}$$

Where $r_{u',s}$ is the score given to service s by user u', $\overline{r_u}$ and $\overline{r_{u'}}$ represent the average rating values of user u and

[5] http://www.wsdream.net/dataset.html.

u' respectively ($u, u' \in \mathcal{U}$), and $s \in \mathcal{S}$ is the Web services rated by both users u and u'. It should be noted that there are usually very few of these services. Particularly when the number of Web services is large, the above formula often fails. A possible solution is to replace the empty scores in QoS matrix with a default score. For example, if Web services are rated on a 0 to 4 scale, we could set the default value as 2.

C. Content-based Recommendation

Content-based service recommendation is based on the analysis of the similarities of the content (e.g., WSDLs and short descriptions) between Web services. There have been two main approaches in content-based Web services recommendation: *syntactic* based approaches [21] and *semantic* based approaches [22]. We discuss only semantic based approaches in this paper since syntactic based approaches have limitations in suggesting high quality recommendations.

The semantics of a Web service s can be represented by a set of semantic attributes: i) functional category $\mathcal{F}(s)$, ii) functional parameters (i.e., inputs $\mathcal{IP}(s)$ and outputs $\mathcal{OP}(s)$), and iii) requirements (i.e., preconditions $\mathcal{P}(s)$ and effects $\mathcal{E}(s)$). We assume that these attributes can be provided by a domain ontology through semantic annotations, which will ensure to provide users with recommendations that are semantically similar to Web services previously invoked. It is possible to construct a domain ontology by analyzing Web service descriptions (WSDLs and free text descriptors). Interested readers are referred to [23] for an approach for bootstrapping ontologies based on Web service descriptions.

The semantic similarity of Web services s_i and s_j can be calculated using:

$$q(s_i, s_j) = \sum_{l \in \{\mathcal{F}, \mathcal{IP}, \mathcal{OP}, \mathcal{P}, \mathcal{E}\}} w_l \times (q_{cd}(l(s_i), l(s_j)), q_m(l(s_i), l(s_j))) \tag{3}$$

where $w_l \in [0, 1]$ is the weight assigned to the l^{th} service description attribute and $\sum_{l \in \{\mathcal{F}, \mathcal{IP}, \mathcal{OP}, \mathcal{P}, \mathcal{E}\}} w_l = 1$. Preferences on some particular service attribute can be done by simply adjusting the value of w_l. The result returned by the formula is a pair of values in [0,1] × [0,1], representing the common description rate q_{cd} and matching quality q_m between s_i and s_j respectively. The matching quality between two semantic descriptions (i.e., $q_m(sd_i, sd_j)$) is a value in [0, 1] defined by a matchmaking function (i.e., 1 for exact match, and 0 for disjoint). The common description rate reflects the degree of similarity between the semantic descriptions of two Web services. Formally, the rate can be calculated using:

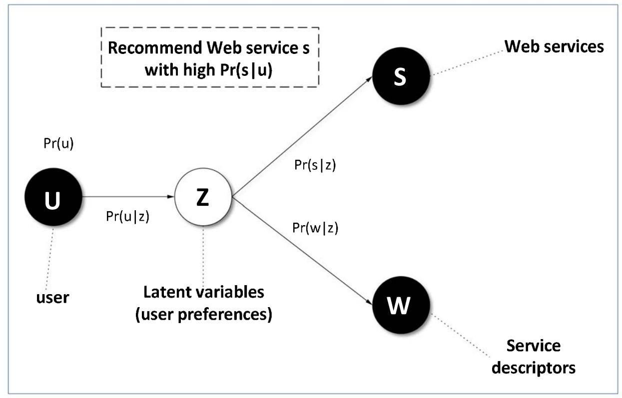


Figure 1. Graphical representation of our approach

$$q_{cd}(sd_i, sd_j) = \frac{|lcs(sd_i, sd_j)|}{|sd_j \backslash sd_i| + |lcs(sd_i, sd_j)|} \tag{4}$$

where $lcs(sd_i, sd_j)$ is the least common subsumer of sd_i and sd_j, which refers to information shared by sd_i and sd_j. $sd_j \backslash sd_i$ represents all the information which is a part of sd_i but not a part of sd_j. The expression in between | refers to the size of ALE concept descriptions of DL (Description Logics) [24].

III. The Hybrid Service Recommendation Model

To meet the three requirements described in Section II-A, in this paper, we propose a hybrid approach that combines collaborative filtering techniques and the content-based approach. To achieve this, it is necessary to reflect both rating and content data in modeling of user preferences. Unfortunately, user preferences are only indirectly represented and observable data such as ratings or content (e.g., semantic descriptions) do not completely reflect the preferences.

To solve the problem, we propose a hybrid approach that associates rating and content data with newly-introduced variables that represent user preferences. Our work is mainly inspired by a *three-way aspect model* presented in [25]. This model has a set of latent variables that directly describe substantial preferences, which cannot be observed directly. The preferences are statistically estimated using *expectation maximization* (EM) that thereafter contribute to better recommendation. In the rest of this section, we will describe how to adapt this model for Web services recommendation.

A. Model Description

The graphical representation of the three-way aspect model for Web services recommendation can be found in Figure 1. The model includes four components: a user set $\mathcal{U} = \{u_1, u_2, ..., u_m\}$, a Web service set $\mathcal{S} = \{s_1, s_2, ..., s_n\}$, content of Web services $\mathcal{W} =$

$\{w_1, w_2, ..., w_n\}$ where w_i is a semantic description of Web service, and a set of latent variables $\mathcal{Z} = \{z_1, z_2, ..., z_k\}$ that governs the recommendation process. The model captures a three-way co-occurrence data among users, Web services, as well as the content of Web services in the form of semantic descriptions. An observation is typically a triple (u, s, w) that corresponds to an event where a user u accesses a Web service s that contains a semantic description w. In the three-way aspect model, observation data is associated with one of the latent variables ($z_i \in \mathcal{Z}$). The latent variables represent user preferences of Web services. It is assumed that users, Web services, and semantic descriptions are independent in the model. It is also worth noting that the aspect model allows multiple semantic descriptions per user, unlike most clustering methods that assign each user with a single class.

In the context of Web services recommendation, an event of a user $u \in \mathcal{U}$ accessing a service $s \in \mathcal{S}$ containing semantic description $w \in \mathcal{W}$, is considered to be associated with one of the latent variables $z \in \mathcal{Z}$. Conceptually, users choose (latent) topics z, which in turn generate both Web services and their content description. Therefore, a latent variable in this new model is associated with not only a distribution of services but also a distribution of service content. The joint probability distribution $Pr(u, s, w, z)$ over user set $\mathcal{U}$, latent topic variables $\mathcal{Z}$, Web service set $\mathcal{S}$ and service content $\mathcal{W}$ is given by

$$Pr(u, s, w, z) = Pr(u)Pr(z|u)Pr(s, w|z) \tag{5}$$

Since we consider that the distribution of s and w are independent in our model, we can have $Pr(s, w|z) = Pr(s|z)Pr(w|z)$. The above equation can be rewritten as:

$$Pr(u, s, w, z) = Pr(u)Pr(z|u)Pr(s|z)Pr(w|z) \tag{6}$$

An equivalent specification of the joint probability distribution that treats users and items symmetrically is

$$Pr(u, s, w, z) = Pr(z)Pr(u|z)Pr(s|z)Pr(w|z) \tag{7}$$

Marginalizing out z, we obtain the joint probability distribution $Pr(u, s, w)$ over $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{W}$ as the following:

$$Pr(u, s, w) = \sum_z Pr(z)Pr(u|z)Pr(s|z)Pr(w|z) \tag{8}$$

This model has a set of parameters $Pr(z)$, $Pr(u|z)$, $Pr(s|z)$ and $Pr(t|z)$, which for simplicity is represented as θ. The model parameters are learned by mining the user-service history data $\mathcal{H} = \{< u, s, w >\}$. One way to learn θ is to maximize the log-likelihood of history data which is:

$$\mathcal{L}(\theta) = \sum_{<u,s,w>\in\mathcal{H}} n(u, s, w) \log(Pr(u, s, w|\theta)) \tag{9}$$

In our work, we adopt the EM algorithm to find a local maximum of the log-likelihood of the training data. The detailed model lof the earning process will be presented in Section III-B.

After the model is learned, the inference of Web services can be ranked for a given user according to $Pr(s|u) \propto \sum_w Pr(u, s, w)$, i.e., according to how likely it is that the user will invoke the corresponding Web service. Web services with high $Pr(s|u)$ that the user has not yet invoked are good candidates for recommendation. This addresses the requirement of recommendation diversity raised in Section II-A.

B. Model Learning

Let $n(u, s, w) = r(u, s) \times n(s, w)$, where $n(u, s, w)$ indicates how much a user u prefers the semantic descriptor w in Web service s, $r(u, s)$ is the rating score of user u for service s, and $n(s, w)$ is the number of times semantic descriptor w occurs in Web service s. Given training data of this form, the log likelihood $\mathcal{L}$ of the data is:

$$\mathcal{L}(\theta) = \log \prod_{<u,s,w>\in\mathcal{H}} Pr(u, s, w|\theta) \tag{10}$$

which can be rewritten as:

$$\mathcal{L}(\theta) = \sum_{<u,s,w>\in\mathcal{H}} n(u, s, w) \log(Pr(u, s, w)|\theta) \tag{11}$$

One way to learn θ is to maximize the log-likelihood of the history data. However, directly maximizing $\mathcal{L}(\theta)$ is hard. The EM algorithm applies an iterative method to improve model parameters. The Equation 11 can be derived as:

$$\begin{aligned}
\mathcal{L}(\theta) &= \sum_{<u,s,w>\in\mathcal{H}} n(u, s, w) \log(Pr(u, s, w)|\theta) \\
&= \sum_{<u,s,w>\in\mathcal{H}} \log Pr(u, s, w|\theta) \\
&= \sum_{<u,s,w>\in\mathcal{H}} \log \sum_z Pr(u, s, w, z|\theta) \\
&= \sum_{<u,s,w>\in\mathcal{H}} \log \left(\sum_z Pr(z|u, s, w, \theta^{(t)}) \frac{Pr(u, s, w, z|\theta)}{Pr(z|u, s, w, \theta^{(t)})} \right) \\
&\geq \sum_{<u,s,w>\in\mathcal{H}} \sum_z Pr(z|u, s, w, \theta^{(t)}) \log \left(\sum_z Pr(z|u, s, w, \theta^{(t)}) \right. \\
&\left. \frac{Pr(u, s, w, z|\theta)}{Pr(z|u, s, w, \theta^{(t)})} \right) \triangleq \mathcal{Q}(\theta|\theta^{(t)})
\end{aligned} \tag{12}$$

Therefore, instead of maximizing $\mathcal{L}(\theta)$ directly, the EM algorithm tries to find the model parameters $\theta^{(t+1)}$ to maximize $\mathcal{Q}(\theta|\theta^{(t)})$. So:

$$\theta^{t+1} = \arg\max\{\mathcal{Q}(\theta|\theta^{(t)})\}$$
$$= \arg\max_{\theta}\left\{\sum_{<u,s,w>\in\mathcal{H}}\sum_{z} Pr(z|u,s,w,\theta^{(t)})\right.$$
$$\left.\log Pr(u,s,w,z|\theta)\right\}$$
$$= \arg\max_{\theta}\left\{\sum_{<u,s,w>\in\mathcal{H}} \mathbb{E}_{z|u,s,w,\theta^{(t)}}\{\log Pr(u,s,w,z|\theta)\}\right\} \quad (13)$$

Then we can use the EM algorithm to solve the equation with training dataset. In particular, E step and M step are iterated alternately until the log-likelihood $\mathcal{L}$ converges to a local maximum. It should be noted that both content and collaboration data can influence recommendations. The relative weight of each type of data depends on the nature of the given data for training. For practical use, it is better to adopt an extended version of the EM algorithm to cope with the data sparseness. One such extension is the deterministic annealing EM algorithm and interested readers are referred to [26] for more details.

IV. Performance Evaluation

This section focuses on reporting the performance study of our proposed hybrid approach for Web services recommendation. In particular, we conduct two experiments to: i) compare our hybrid approach with the conventional methods including collaborative filtering and content-based recommendation and ii) study the sensitivity of the hybrid approach under different markoff ratios. All experiments were conducted on a Core 2 Quad 2.70 GHz machine with 8GB RAM.

A. Dataset Setup

To perform reliable experiments, it is ideal to use large-scale real Web service data that is sufficient to a certain extent. Unfortunately, constructing such data is extremely time-consuming. Luckily, there is a recent effort made by Zheng et al. [5] in their WS-DREAM project[6], which shares a large-scale real Web service dataset. WS-DREAM developed a Web crawling engine to crawl publicly available WSDL file addresses from the Internet. It also collected non-functional attributes (e.g., QoS) of these Web services, which are observed by 339 distributed computers located in 30 different countries, from Planet-Lab[7]. We use this dataset as our base dataset and perform the following pre-processing:

1) We traversed all 5,825 WSDL addresses offered from the dataset and retrieved WSDL documents of 3,693 live Web services. We then calculated the tf/idf to find out the weighted descriptors for each Web service from its operations name, method names and WSDL address by exploiting the approach developed in our previous work [23]. Consequently, each Web service has a keyword list describing the Web service.
2) For some Web services, we collected their corresponding rating scores directly from seekda.com. For those Web services whose rating scores are not available from seekda.com, we determined rating scores (e.g., from 1 to 5) based on their QoS values (e.g., response time, throughput) using a multi-attribute utility function [27].

[6]http://www.wsdream.net
[7]http://www.planet-lab.org

Table I
ORIGINAL DATASET STATISTICS

Number of Users	339
Number of Web Services	5825
User-Service (Response Time) Matrix Density	5.11×10^{-2}
User-Service (Throughput) Matrix Density	7.26×10^{-2}

Table II
PROCESSED DATASET STATISTICS

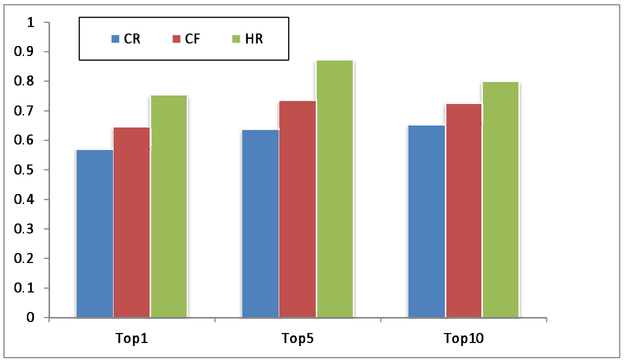
Number of Users	339
Number of Web Services	3693
User-Service (Response Time) Matrix Density	5.32×10^{-2}
User-Service (Throughput) Matrix Density	7.67×10^{-2}
Average Words for each Services	12.79
Average QoS Ratings	3.84

Table I and II show the dataset statistics for the original dataset and the one after the pre-processing.

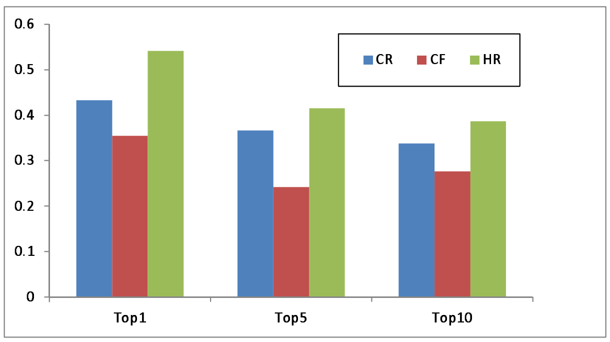
B. Performance Comparison

In order to study the recommendation performance, we compared our proposed hybrid recommendation approach (HR) with the other two methods: collaborative filtering (CF) and content-based recommendation (CR). The matrix (see Section IV-A) was randomly divided into the training matrix and the evaluation matrix by masking 70% of actual scores. The reason that we set the ratio as 70% is from the experimental study shown in Section IV-C.

The recommendation performance was evaluated by examining the quality of top x rankings of Web services (x = 1, 5, 10). Specifically, after each model is learned, we used the model parameters to find $\forall s$, $Pr(s|u)$ for all users. The Web services in the testing dataset were ranked based on their $Pr(s|u)$. The average precisions and average recalls for top x recommendations were used as the evaluation metrics. Average precision was calculated as the ratio of the number of top x recommendation hits to the recommendation size; and average recall was calculated as the ratio of the number of top x recommendation hits to the size of user's validation item set. We calculated the average precisions and recalls of



(a)



(b)

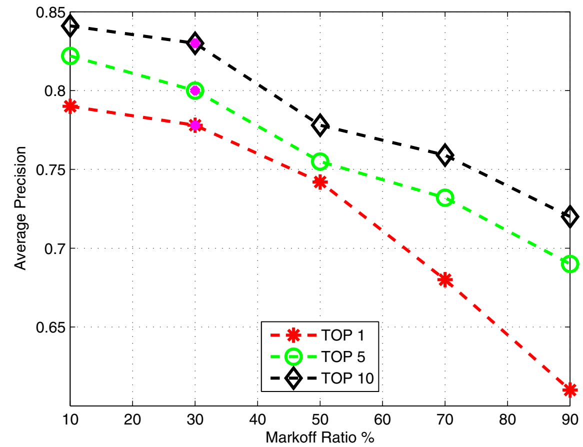
Figure 2. Recommendation performance comparison between HR, CF, and CR with top 1, 5, 10 Web services: (a) average precision and (b) average recall

all users for three different methods. Figure 2 shows the result.

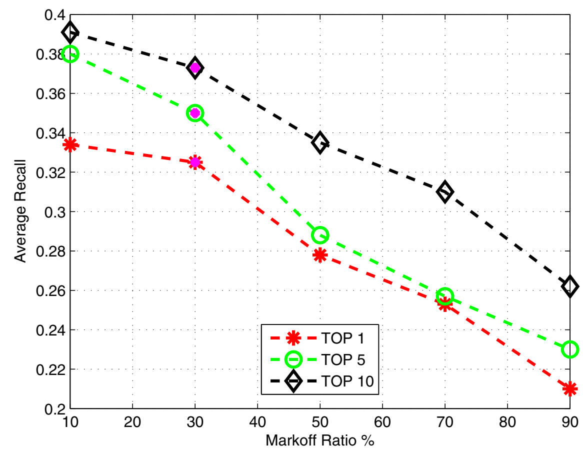
From Figure 2 (a) we can see that, the top x precision values of our hybrid approach (HR) are higher than collaborative filtering (CF) and content-based recommendation approach (CR). In Figure 2 (b), the top x recall values of HR are also higher than the other two approaches. It is clear that our approach outperforms the other two approaches and more relevant Web services can be recommended by our approach.

C. Sensitivity of Markoff Ratio

As mentioned previously, we divided the whole matrix into training and testing matrices. In this experiment, we studied the impact of the markoff ratios on the performance of our proposed hybrid approach. We randomly marked off $y\%$ (y = 10, 30, 50, 70) of actual scores and the rest of the matrix is used as training dataset to infer model parameters. Our algorithm was then used to recover the information that has been marked off. We applied cross-validation method to find the average precisions and recalls for top x (x = 1, 5, 10) Web services recommendation. Figure 3 shows the result.



(a)



(b)

Figure 3. Recommendation performance with top 1, 5, 10 Web services: (a) average precision and (b) average recall

From the figure, we can see that with the increase of the markoff ratio, the overall recommendation performance decreases. This is easy to explain: a higher markoff ratio means less data available for training the approaches, therefore worse recommendation performance. Interestingly, we notice that the recommendation performance increases quickly until markoff ratio reaches 70%, and then increases slowly when markoff ratio decreases.

V. Related Work

Web services recommendation and selection has been a fundamental research issue since the dawn of Web service technologies. Traditional Web services discovery centers on UDDI registries such as the work presented in [28], [8], [9]. Unfortunately, UDDI is no longer the choice of publishing Web services. The available Web services search engines such as *seekda* largely exploit keyword-based search techniques and are insufficient to catch the functionalities of Web services. These search engines barely consider non-functional characteristics (QoS) of Web services. Furthermore, users normally have to specify and execute queries themselves. The performance of Web services recommendation of these search engines is therefore quite limited.

Over the past few years, a heavily researched topic in Web services recommendation centers on QoS-based Web services selection that supports optimized Web services selection by considering QoS attributes of Web services with similar functionalities, as well as the preferences from service users [27], [29], [12], [30]. The quality of the recommendation from these approaches depends on the quality of available QoS information for Web services. Most QoS-based service selection approaches assume that the QoS information (e.g., availability of Web service) is pre-existing and readily accessible with guaranteed quality, which unfortunately is not true, as indicated by Zheng et al. in [5]. Service providers may not be able to deliver the QoS they promised and some QoS properties (e.g., network latency, invocation failure-rate, etc.) are highly related to the locations and network conditions of the service users. Thus, this kind of approaches are impractical for use in many applications. In this work, we consider classical recommendation methods and propose a novel approach that automatically recommends Web services by considering the information of similar service users and the content of similar Web services.

There are two main recommendation methods, namely *collaborative filtering* and *content-based recommendation*. The content-based approaches recommend items (Web services in our context) similar to those that a user appreciates based on the item's characteristics while the collaborative filtering approaches recommend items based on the similarity of different users. Zheng et al. [5] propose a collaborative filtering approach to predict missing QoS based on the information of similar Web users and services. The work by Chen et al. [14] presents RegionKNN, a collaborative filtering algorithm that is designed for large-scale Web services recommendation. This approach considers service users' physical locations and proposes a region model by considering the QoS characteristics of Web services. A refined nearest-neighbor algorithm is then developed for QoS-based service recommendation. Blake and Nowlan [21] develop a Web services recommender system by exploiting an enhanced syntactic approach to compare the content of Web services. Both approaches have weaknesses as discussed in Section II.

Our work presents a hybrid approach for better Web services recommendation by systematically combining both methods together. In particular, we propose a three-way aspect model that considers both QoS ratings and the semantic content of Web services. User preferences are modeled as a set of latent variables in the aspect model [25], which can be statistically estimated using the expectation maximization (EM) method. To the best of our knowledge, this is the first of few approaches that combine collaborative filtering and content-based approach for Web services recommendation.

VI. Conclusion and Future Work

Web services recommendation and selection is a fundamental issue in service oriented computing. Existing Web services discovery and recommendation approaches focus on either perishing UDDI registries, or keyword-dominant, QoS-based Web service search engines. Such approaches possess many limitations such as insufficient recommendation performance and heavy reliance on the input from users (e.g., preparing queries). In this paper, we have proposed a novel hybrid approach for effective Web services recommendation. Our approach exploits a three-way aspect model that systematically combines classic collaborative filtering and content-based recommendation. The proposed hybrid approach simultaneously considers the similarities of user ratings and semantic Web service content. Our approach is validated by conducting several experimental studies using 3,693 real-world Web services publicly available from the Internet. The experimental results show that our approach outperforms the conventional collaborative and content-based methods in terms of recommendation performance.

Our future work includes conducting more experiments to study the performance of the proposed approach (e.g., capability for dealing with new services). We also plan to explore more refined/personalized Web services recommendation by considering the specific contexts (e.g., goals an end user would like to achieve, physical situations, etc.).

References

[1] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," *IEEE Computer*, vol. 40, no. 11, pp. 38–45, 2007.

[2] Q. Yu, X. Liu, A. Bouguettaya, and B. Medjahed, "Deploying and Managing Web Services: Issues, Solutions, and Directions," *The VLDB Journal*, vol. 17, no. 3, pp. 537–572, 2008.

[3] Q. Z. Sheng, J. Yu, and S. Dustdar, Eds., *Enabling Context-Aware Web Services: Methods, Architectures, and Technologies*. CRC Press, 2010.

[4] J. Ma, Q. Z. Sheng, K. Liao, and A. H. Ngu, "WS-Finder: A Framework for Similarity Search of Web Services," in *Proceedings of the 10th International Conference on Service Oriented Computing (ICSOC 2012)*, Shanghai, China, November 2012.

[5] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Wsrec: A collaborative filtering based web service recommender system," in *Proceedings of IEEE International Conference on Web Services (ICWS 2009)*, 2009, pp. 437–444.

[6] M. Huhns and M. Singh, "Service-oriented computing: Key concepts and principles," *IEEE Internet Computing*, vol. 9, no. 1, pp. 75–81, 2005.

[7] E. Al-Masri and Q. Mahmoud, "Investigating Web Services on the World Wide Web," in *Proceedings of the 17th International Conference on World Wide Web (WWW'08)*, Beijing, China, April 2008.

[8] D. Kourtesis and I. Paraskakis, "Combining SAWSDL, OWL-DL and UDDI for Semantically Enhanced Web Service Discovery," in *Proceedings of the 5th European Semantic Web Conference (ESWC'08)*, 2008.

[9] J. Yu and G. Zhou, "Web Service Discovery and Dynamic Invocation based on UDDI/OWL-S," in *Proceedings of the Third International Conference on Business Process Management (BPM'05)*, 2005.

[10] P. Krill, "Microsoft, IBM, SAP Discontinue UDDI Registry Effort," http://www.infoworld.com/d/architecture/microsoft-ibm-sap-discontinue-uddi-registry-effort-777, visited on 07 May 2012.

[11] P. Bonatti and P. Festa, "On Optmal Service Selection," in *Proceedings of 14th International World Wide Web Conference (WWW 2005)*, Chiba, Japan, 2005.

[12] Y. Zhang, Z. Zheng, and M. Lyu, "WSExpress: A QoS-Aware Search Engine for Web Services," in *Proceedings of 8th IEEE International Conference on Web Services (ICWS 2010)*, Miami, Florida, USA, 2010.

[13] L. Chen, L. Hu, Z. Zheng, J. Wu, J. Yin, Y. Li, and S. Deng, "WTCluster: Utilizing Tags for Web Services Clustering," in *Proceedings of 9th International Conference on Service-Oriented Computing (ICSOC 2011)*, Paphos, Cyprus, 2011.

[14] X. Chen, X. Liu, Z. Huang, and H. Sun, "Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Proceedings of IEEE International Conference on Web Services (ICWS 2010)*, 2010, pp. 9–16.

[15] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *Proceedings of International Joint Conference on Artificial Intelligence*, vol. 16, 1999, pp. 688–693.

[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[17] K. Elgazzar, A. E. Hassan, and P. Martin, "Clustering WSDL Documents to Bootstrap the Discovery of Web Services," in *Proceedings of the International Conference on Web Services (ICWS 2009)*, 2009.

[18] W. Liu and W. Wong, "Web Service Clustering Using Text Mining Techniques," *International Journal of Agent-Oriented Software Engineering*, vol. 3, no. 1, pp. 6–26, 2009.

[19] J. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.

[20] Y. Zhang, Z. Zheng, and M. R. Lyu, "Exploring latent features for memory-based qos prediction in cloud computing," in *Proceedings of the 30th IEEE Symposium on Reliable Distributed Systems (SRDS 2011)*, 2011, pp. 1–10.

[21] M. B. Blake and M. F. Nowlan, "A Web Service Recommender System Using Enhancing Syntactical Matching," in *Proceedings of IEEE International Conference on Web Services (ICWS'07)*, 2007.

[22] F. Lécué and A. Delteil, "Making the Difference in Semantic Web Service Composition," in *Proceedings of the 22nd International Conference on Artificial Intelligence (AAAI'07)*, 2007.

[23] A. Segev and Q. Z. Sheng, "Bootstrapping Ontologies for Web Services," *IEEE Transactions on Services Computing*, vol. 5, no. 1, pp. 33–44, 2012.

[24] W. Cohen, A. Borgida, and H. Hirsh, "Computing Least Common Subsumers in Description Logics," in *Proceedings of International Conference on Artificial Intelligence (AAAI'92)*, 1992.

[25] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-based Recommendation in Sparse-Data Environments," in *Proceedings of the 17th International Conference on Uncertainty in Artificial Intelligence (UAI'01)*, 2001.

[26] N. Ueda and R. Nakano, "Deterministic Annealing EM Algorithm," *Neural Networks*, vol. 11, no. 2, pp. 271–282, 1998.

[27] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality Driven Web Services Composition," in *Proceedings of the 12th International World Wide Web Conference (WWW 2003)*, Budapest, Hungary, May 2003.

[28] T. Qiu, L. Li, and P. Lin, "Web Service Discovery with UDDI Based on Semantic Similarity of Service Properties," in *Proceedings of the Third International Conference on Semantics, Knowledge and Grid (SKG'07)*, 2007.

[29] N. Thio and S. Karunasekera, "Automatic measurement of a qos metric for web service recommendation," in *Proceedings of 2005 Australian Software Engineering Conference*, 2005, pp. 202–211.

[30] H. Zheng, J. Yang, W. Zhao, and A. Bouguettaya, "QoS Analysis for Web Service Compositions Based on Probabilistic QoS," in *Proceedings of the 9th International Conference on Service Oriented Computing (ICSOC 2011)*, Paphos, Cyprus, December 2011.