WSRank：A Collaborative Ranking Approach for Web Service Selection

Linlin Meng, Jianxin Li, Hailong Sun
School of Computer Science and Engineering
Beihang University
Beijing, China
{mengll, lijx, sunhl}@act.buaa.edu.cn

ABSTRACT

As Cloud Computing has emerged as new computing paradigm, more and more services have been deployed and provided on the cloud platform with a SaaS model, thereby how to select a qualified service is becoming a key issue. Several approaches based on service feedback ranking e.g., rating-oriented collaborative filtering (CF) have been proposed. Traditional CF approaches predict the potential QoS values that a user would assign to unused services, so that service candidates are mainly ranked with the predicted values. In this paper, we present WSRank, a novel ranking-oriented CF approach that addresses the service ranking problem through directly modeling user preferences derived from past QoS values. Different from the existing similar approaches, WSRank firstly presents a QoS model which allows users to express their preferences flexibly while providing combination of multiple QoS properties to give an overall rating to a service. Second it measures the similarity between users based on the correlation between their rankings of services rather than the rating values. Experimental results show that our approach outperforms the competing approaches significantly on the AP measure for evaluating ranked results.

Keywords

Web service; QoS; service ranking; collaborative filtering

1. INTRODUCTION

Recently, cloud computing has become a popular computing paradigm with centralized hardware and software infrastructure. Software as a Service (SaaS) is also becoming a new software providing mode. All of these technologies show a centralized trend of web services. In particular, service-oriented architecture (SOA) and related technologies have also been extensively used. As an important type of SOA realization technologies, more and more web services have been provided over the Internet.

Web service discovery has been extensively studied, which mainly deals with functional properties [1]. However, due to the large amount of services with identical or similar functionalities, users will be overwhelmed by the candidates. While web service discovery alone cannot tackle this problem, effective approaches to web service selection have become more necessary, which is a key issue in the field of service computing [2]. With consideration to the distributed and dynamic nature of web services, user preferences and more service properties should be considered because the similar functional services generally have very different quality. These non-functional service properties are also known as Quality of Service (QoS). Many researchers [3,4,9,20] propose that QoS is a key factor in the success of building critical service-oriented applications.

The most straightforward approach [5,6] of personalized service selection is to evaluate all the services and select the services based on the observed QoS values. However, it's not practical for a consumer to evaluate the comprehensive QoS information of each candidate service, since it is time-consuming, and some properties are hard to measure through several web service invocations, such as reputation and reliability.

In recent years, much work based on service feedback have been proposed e.g. rating-oriented CF [7,8,10,11], ranking-oriented CF [16], and etc. Among them, collaborative filtering (CF) is a reprehensive work on service selection. However, users' requirements during web service selection cannot be fully satisfied since these CF approaches only make value prediction or ranking prediction for single QoS property which is not sufficient to represent the overall quality of web service.

The existing CF approaches have two limitations. One is the order has a high priority rather than its value during a service ranking predication. Moreover, higher accuracy in rating prediction does not necessarily lead to better ranking effectiveness. Suppose we have three services s_i, s_j and s_k, for which the true ratings are known to be 3, 5 and 7 respectively and two different methods have predicted the ratings on s_i, s_j and s_k to be (2, 8, 12) and (7, 6, 5) respectively. In terms of rating prediction accuracy measured by the absolute deviation (MAE) from the true rating, the second prediction is better than the first one. While using the prediction (7, 6, 5), service s_i, s_j and s_k will be incorrectly ordered while the prediction (2, 8, 12) ensures the correct order.

Another is better service should have better ranking accuracy. In the ranking-oriented CF approach proposed by Liu et al., although Kendall'sτ seems to be a reasonable choice for comparing two rankings, there still exist some issues. Let's consider an example with eight different services. We assume their actual ranking order is (1, 2, 3, 4, 5, 6, 7, 8) and two alternate methods have given ranked order (4, 3, 1, 2, 5, 6, 7, 8) and (1, 2, 3, 4, 8, 7, 5, 6) to them respectively. Hence, based on the Kendall's τ values, the two methods are equivalent (both equal to 0.6429). The Kendall's τ does not distinguish between the errors that occur towards the top of the ranking list from the errors towards the bottom of the list [20]. While in most cases of web service selection, we care more about the services that are ranked towards the top of the list because of their higher probability of being selected.

To address the issues mentioned above, we propose a collaborative ranking approach named WSRank, which directly addresses the service ranking problem without going through the intermediate step of rating prediction. The main contributions of this paper are as follows:

- We design a QoS model which allows users to express their preferences flexibly while providing combination of multiple QoS properties to give an overall rating on services.
- We describe a similarity measure sensitive to errors' position in the ranking list, which is used to determine a set of users that share similar preferences to the active user. Then we present an efficient ranking algorithm for web service selection.
- We conduct extensive experiments to study the performance of WSRank compared with other approaches. Experimental results show that our approach outperforms the competing approaches significantly. WSRank is also integrated into our ServiceXchange[1] which is a web service repository and search engine.

The remainder of this paper is organized as follows. Section 2 introduces the prototype design and the typical CF approaches. Section 3 elaborates a QoS model and describes our collaborative ranking approach. Section 4 presents experimental results and their analysis. Section 5 briefly discussed some related work. Finally section 6 concludes the whole paper.

2. Preliminaries

2.1 Prototype Design

Figure 1 shows the system architecture of ServiceXchange designed by our team in BUAA. ServiceXchange is a web service repository and search engine like Seekda that provides functions of web service registering, searching, monitoring, etc.

[1] http://www.servicexchange.cn.

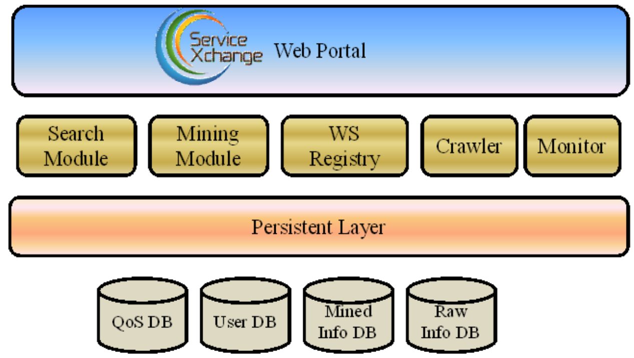

Figure 1. System Architecture of ServcieXchange

The following scenario shows why personalized service ranking is needed:

As a web service repository, there exist huge numbers of distributed and reusable services in ServiceXchange. When searching services in ServiceXchange, an active user usually will obtain a large set of services with same functionality. Then the active user wants to select the optimal service from the set of functionally equivalent service candidates. Thus personalized service quality ranking of the functionally equivalent services is required for the active user to make an optimal service selection.

2.2 The Typical CF Approaches

One widely used collaborative filtering approach is the neighborhood-based CF approach. The rating-oriented approach and the ranking-oriented approach are two types of common approaches to neighborhood-based collaborative filtering.

The rating-oriented approach usually tries to predict the missing values in user-service matrix as accurately as possible. In rating-oriented CF Approach, the Person Correlation Coefficient (PCC) [21] is a popular similarity computation approach, measures the similarity between users and web services. Three well-known rating-oriented approaches based on PCC are user-based model [8] (named UPCC), item-based model [10] (named IPCC), the user-based and item-based combined model [7] (named UIPCC) respectively.

In the ranking-oriented CF approach proposed by Liu et al. [16], user preferences are modeled as personal rankings derived from user ratings. The Kendall Rank Correlation Coefficient is used to measure the similarity between users by computing the correlation of corresponding personal rankings. To predict the full rankings of unrated items, this approach defines a preference function that represents predicted preference relation between items in the form of $\Psi : I \times I \rightarrow \mathbb{R}$. In the end, a greedy order algorithm is used to compute full rankings.

3. WSRank: A Collabrative Ranking Approach

This section presents our collaborative quality ranking approach for web services, which is designed as a four-phase process. During Phase 1, we introduce the QoS model and the utility function to obtain the user-service rating matrix. Then in Phase 2, we calculate the similarity of the users with the active user and identify a set of similar users. After that, in Phase 3 a preference function is defined to present the priority of two services. Finally, during Phase 4, a greedy order algorithm is proposed to rank the service candidates by making use of the past usage experiences of other similar users. Details of these phases are presented at Section 3.1 to Section 3.4, respectively.

3.1 QoS Model

The aim of our approach is to discover web services with optimal QoS ratings for an active user. Let $Q = (q_{i,1}, q_{i,2}, \dots, q_{i,l})$ is a QoS vector of service i from a user u, where $q_{i,j}$ represents the j th criterion value of service i. For each user u, we can obtain the matrix E_u by putting all the services' QoS vectors together. Each row in E_u represents a service, while each column represents a QoS criterion value.

To represent user priorities and preferences, two steps are involved. First, due to the fact that QoS criteria vary in units and magnitude, values of E_u ($q_{i,j}$) must be normalized to perform QoS-based ranking. Second, the weighted evaluation on criteria needs to be carried out for representing user's constraints, preference and special requirements.

In normalization step each criterion value is transformed to a real value between 0 and 1 by using the Gauss normalization [12]. Thus, the QoS matrix E_u is transformed into a normalized matrix E'_u. Note that for some criteria, smaller values mean higher service quality. For such criteria, we have E'_u ($h_{i,j}$) = $1/ E'_u$ ($h_{i,j}$).

A weight vector $W = (w_1, w_2, \dots, w_l)$ is used to represent user's priorities on preferences given to different criteria with $w_k \in R_0^+$ and $\sum_{k=1}^{l} w_k = 1$. The final QoS ratings vector $R_u = (r_{u,1}, r_{u,2}, \dots, r_{u,n})$ of service candidates are therefore can be computed with the following formula:

$$r_{u,i} = \sum_{k=1}^{l} w_k h_{i,k}, 1 \le i \le n \tag{1}$$

Finally, we can get a user-service matrix $R_{m\times n}$ where $r_{i,j}\,(1 \le i \le m, 1 \le j \le n)$ represents the QoS ratings of user i to service j. Note that in real situation, users are used to give few QoS values. Therefore, all the above computation is just for those services which have been fully rated. If any criterion value of a service by user does not exist, then user's rating on this service will be set to ϕ.

Note that our QoS model allows users to show their preferences flexibly while providing combination of multiple QoS properties to give an overall rating on services. If users only want to employ a specific QoS property, they just need to set the weight of this property as 1 and others as 0.

3.2 Ranking Similarity Computation

The KRCC measure described in section 2 does not consider positions of concordant pairs appearing in the list, which means that the concordant pairs ranked higher are as important as those ranked lower. Yilmaz et al. therefore proposed a new rank correlation coefficient, namely Average Precision Correlation Coefficient (AP) [14], taking into account the position of concordant pairs in the rank list. Given two rank lists l_1 and l_2, assuming l_2 is an actual rank list, the AP coefficient is defined as:

$$\tau_{ap}(l_1, l_2) = \frac{2}{N-1} \cdot \sum_{i=2}^{N} \left(\frac{C(i)}{i-1} \right) - 1 \tag{2}$$

where $C(i)$ is the number of services ranked higher than rank i and correctly ranked with respect to the service at rank i in l_1.

In our approach, the similarity of any two users is computed via AP Correlation Coefficient between the two different rankings of commonly invoked services given by $Sim(u,v) = \tau_{ap}\left(R_u^{S_u \cap S_v}, R_v^{S_u \cap S_v}\right)$ where $R_u^{S_u \cap S_v}$ is the ranking given by user u over common set of services that user u and v both invoked. By calculating the AP similarity values between the active user and other users, the users similar to the active user can be identified. In our approach, the set of similar users N_u based on AP is identified for the current user u by: $N_u = \{ v | Sim(u,v) > 0.5, v \neq u \}$. The value of $Sim(u,v)$ can be calculated by Eq. (2).

3.3 Preference Function

A user's preference on a pair of services is modeled in the form of $\Psi : S \times S \to \mathbb{R}$, where $\Psi(i,j) > 0$ means that service i is more preferable to j for user u and vice versa. The preference function $\Psi(i,j)$ is anti-symmetric, i.e. $\Psi(i,j) = -\Psi(j,i)$. We set $\Psi(i,i) = 0$ for all $i \in S$. We used user-based model to predict the value of preference function $\Psi(i,j)$ of candidate services.

$$\Psi(i,j) = \begin{cases} r_{u,i} - r_{u,j} & i,j \in S_u \\ \dfrac{\sum_{v \in N_u^{i,j}} Sim(u,v) \cdot (r_{v,i} - r_{v,j})}{\sum_{v \in N_u^{i,j}} Sim(u,v)} & other \end{cases} \tag{3}$$

where v is a similar user of the active user u, $N_u^{i,j}$ is a subset of similar users of u, who have invoked both service i and j.

Given a preference function Ψ which assigns a score to every pair of services $i, j \in S$, we want to choose a quality ranking of services in S that agrees with the pair wise preferences defined by Ψ as much as possible. Let ρ be a ranking of services in S such that $\rho(i) > \rho(j)$ if and only if i is ranked higher than j. We can define a value function $V^{\Psi}(\rho)$ as follows that measures the consistency of the ranking with the preference function.

$$V^{\Psi}(\rho)=\sum_{i,j:\rho(i)>\rho(j)}\Psi(i,j) \tag{4}$$

Therefore our goal is to produce a ranking ρ^* that maximizes the above objective value function.

3.4 Greedy Order Algorithm

One possible approach to solve the service ranking problem is to search through all the possible rankings and select the optimal ranking ρ^* that maximizes the value function defined in Eq. (4). However, Cohen et al. [15] have showed that finding the optimal ranking ρ^* is an NP-Complete problem and proposed a greedy order algorithm for finding an approximately optimal ranking shown in Algorithm 1 below:

Algorithm 1 Greedy Order Algorithm

```
Input: a service set S; a preference function Ψ
Output: a service ranking ρ
// computing the potential value for each service in S
for each i ∈ S do
    π(i) = Σ_{j∈S} Ψ(i, j) − Σ_{j∈S} Ψ(j, i)
end for
while S ≠ ϕ do
// picking the service t with the maximum potential value
    t = argmax_{i∈S} π(i)
// assign the selected service with a rank order of n-|S|+1
    ρ(t) = n − |S| + 1
// delete the selected service t from S
    S = S − {t}
// update the potential values of the remaining services
    for each i ∈ S do
        π(i) = π(i) − Ψ(i,t) + Ψ(t,i)
    end for
end while
```

The greedy algorithm produces the ranking from the highest position to the lowest position by always picking the service t that currently has the maximum potential value. The selected service is assigned to a rank equal to n-$|S|$+1 so that it will be ranked above all the other remaining services in S. The ranks are in the range of [1, n] where n is the number of services and a smaller value indicates higher ranking. Then the selected service is deleted from S and the potential values of the remaining services are updated by removing the effects of t. Algorithm 1 has a time complexity of $O(n^2)$, where n is the number of services.

However, algorithm 1 (named GOAP) treats the explicitly rated items and the unrated items equally, so it does not guarantee that the explicitly rated items will be ranked correctly. In our approach, we take advantage of user's given ratings to make correction for the initial ranking. Suppose G is a set of services whose ratings have given by a user, the correction procedure follows these steps:

1. Rank the services in G based on given ratings. $\rho_g(t)$ stores the ranking, where t is a service and the function $\rho_g(t)$ returns the corresponding order of this service.

2. For all the services in G, find the service j with minimum position in $\rho(t)$.

3. Pick the first service i from $\rho_g(t)$ and exchange the position of i and j in $\rho(t)$. Then delete i from G and its correspond order $\rho_g(t)$. If G is not empty, then goto step 2.

4. Experiments

4.1 Data Sets And Evaluation Metric

We adopt the WSRec [7] dataset, which contains about 1.5 million web service invocation records of 100 web services from more than 20 countries. We extract a subset of 300,000 QoS records and 3000 users are generated, each of whom is associated with a QoS profile of all the 100 services. As for our QoS model, we take two QoS properties, RTT and throughput with equal weight as an example.

The 3000 users are divided into two parts, one as training users and the rest as active (test) users. To simulate the real situation, we randomly remove some QoS entries of the training matrix and testing matrix to make a set of sparse matrices with density ranging from 10% to 60%.

As our approach is ranking-oriented rather than rating-oriented, we have to employ metrics for measuring quality of produced rankings. Since AP Correlation Coefficient gives more weight to the errors appearing at high ranking positions, it is more suitable for measuring the services ranking. In our experiments, we also employ the AP Correlation Coefficient (Eq. (2)) to evaluate the efficiency of WSRank.

4.2 Impact of Parameters

4.3.1 Impact of Service Numbers

The number of services impacts the personalized ranking of user u. In these experiments, we set different values to $|S|$ and compare two ranking-oriented methods (i.e., GOAP and WSRank) on the experiment data. The density of the training matrix and the testing matrix is 0.6 and 0.3 respectively.

Figure 2 shows changes in performances while the limitation of $|S|$ increases from 10 to 100 by a step of 10. We observe that the mean AP Correlation Coefficient gradually increases at first and then decreases when service numbers increases. Lower mean AP Correlation Coefficient at the beginning is likely due to the incomplete user information, i.e., missing enough preference information to establish similarity relationships between users. The decreasing of mean AP Correlation Coefficient may be caused by inappropriate preference relations. We also observe that at max ($|S|$) = 30 the prediction accuracy reaches the best value in our experiment.

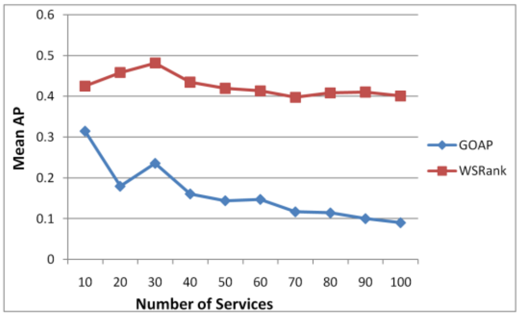

Figure 2. Mean AP Correlation Cofficient VS. max(|S|) Results

4.3.2 *Impact of Matrix Density*

To study the impact of the matrix density on the ranking accuracy, we change the matrix density from 10% to 60% with a step value of 10%. We set the number of services at 30. Two ranking-oriented methods (i.e., GOAP and WSRank) are compared in this experiment.

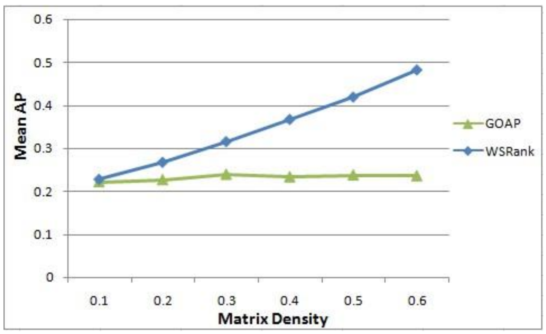

Figure 3. Impact of Matrix Density

Figure 3 shows the experiment results. We can see that when the matrix density is increased from 10% to 60%, our WSRank method outperforms the GOAP method consistently. The ranking accuracy of WSRank is significantly enhanced. This observation indicates that the prediction accuracy of WSRank can be greatly enhanced by collecting more QoS values to make the matrix denser, especially when the matrix is very sparse.

4.4 *Performance Comparison*

To study the personalized services ranking performance, we choose four algorithms as the baselines including UPCC [8], IPCC [11], UIPCC [7] and the greedy order algorithm using Kendall Rank Correlation Coefficient (GOKRCC) [16] compared with our ranking approaches.

In terms of the active users, we vary the number of QoS values given by them from 10, 20 to 30, and name them Given 10, Given 20, and Given 30, respectively. The rankings based on the original full matrix are employed as ideal rankings to study the ranking performance. In this experiment, we set *top-k*=10 and λ = 0.8 for UPCC, IPCC and UIPCC. We set number of services with 30 and employing the 0.3 density training matrix. Table 1 shows the ranking prediction accuracy on the AP Correlation Coefficient of different methods.

Table 1: Average Precision Correlation Coefficient

Training User = 2700 Density = 0.3			
Method	**Given 10**	**Given 20**	**Given 30**
UPCC	-0.6026	-0.5981	-0.6218
IPCC	-0.5419	-0.6029	-0.6248
UIPCC	-0.4245	-0.4747	-0.4913
GOKRCC	0.00069	0.00067	0.00048
GOAP	0.2228	0.2277	0.2389
WSRank	**0.2268**	**0.2753**	**0.3159**

Table 1 shows that:

- Among all the ranking methods, our WSRank obtains better prediction accuracy (larger APCC values) under all the experimental settings consistently.
- The ranking-oriented methods (GOKRCC, GOAP and WSRank) consistently outperform the rating-oriented approaches (UPCC, IPCC and UIPCC), since ranking-oriented CF methods have taken the user's preferences on service pairs into consideration.
- Compared with GOKRCC, GOAP and WSRank consistently achieves better ranking performance, which indicates that the Average Precision Correlation Coefficient is more suitable for similarity computation in ranking-oriented CF approaches.
- When the density of matrix is increased from 10% to 30%, the ranking accuracy is also enhanced, since denser user-item matrix provides more information.
- The approaches that combine user-based and item-based approach (UIPCC) outperforms the user-based approach (UPCC) and item-based approach (IPCC) consistently. This observation indicates that by combing the user-based and item-based approaches, better service ranking performance can be achieved.

5 RELATED WORK

Along with Service Oriented Architecture (SOA), many computing paradigms such as SaaS (Software as a Service) and Cloud computing etc. are becoming more mature, leading to the centralized trend of web services. Web services have a rapidly increase in number, making QoS-aware service selection become a hot topic.

The non-functional properties of services can be measured from either the user or the service provider. Based on the QoS values of services, numbers of service selection approaches [5,6,17,18] have been proposed, which enable optimal services to be identified from a set of candidates and the preference of users. Different from these approaches, our work focuses on service ranking prediction through collaborative filtering approach, which employs the information of similar service users.

Collaborative Filtering, firstly proposed by Rich [19], is widely used in commercial recommender systems like Amazon.com [10]. The collaborative filtering approach is based on the assumption that a user would usually be interested in those items preferred by other users with similar interests. The CF approach does not require any content information about the items, it works by collecting ratings on the items from a large number of users and make ranking of items to a user based on the preference patterns of other users.

One widely used collaborative filtering approach is the neighborhood-based CF approach. There are two types of common approaches to neighborhood-based collaborative filtering. One is the rating-oriented approach and the other is the ranking-oriented approach. There exist three types of rating-oriented approach: the user-based model [8], the item-based model [10,11] and their fusion [7]. In rating-oriented CF Approach, the Person Correlation Coefficient (PCC) [21] is a popular similarity computation approach, measures the similarity between users and web services.

The rating-oriented CF approaches usually try to predict the missing values in user-item matrix as accurately as possible. However in the ranking-oriented scenarios, accurate missing value prediction may not lead to accuracy ranking. Therefore, ranking-oriented CF approaches are becoming more and more attractive.

6 CONCLUSIONS

In this paper, we propose WSRank, a collaborative quality ranking approach for web services. We design a QoS model allowing users to express their preferences flexibly while providing combination of multiple QoS properties to give an overall rating on services. And then we employ a greedy method to select the best service for users based on their preferences. Experimental results show that our approach outperforms existing rating-based collaborative filtering approaches and the traditional greedy method.

For future work, we will investigate different technologies for improving the accuracy of our approach (e.g., random walk, matrix factorization, utilizing content information, etc.).

7 ACKMOWLEDGMENT

We would like to thank Zibin Zheng in the Chinese University of Hong Kong for his providing experiments dataset. The authors gratefully acknowledge the anonymous reviewers for their helpful suggestions and comments. This work is partially supported by Program for New Century Excellent Talents in University 2010 and the Fundamental Research Funds for the Central Universities, National Nature Science Foundation of China (No. 60903149,91018004), China 863 Program (No. 2007AA010301), and China 973 Fundamental R&D Program (No. 2011CB302602).

8 REFERENCES

[1] R. M. Sreenath and M. P. Singh, "Agent-based service selection," Journal of Web Semantics, pp. 261–279, 2003.

[2] L.-J.Zhang, J.Zhang, and H.Cai, Service computing, Springer and Tsinghua University Press, 2007.

[3] Zhang Jia, Trustworthy "Web Services: Actions for Now," IEEE IT` Professional, 2005, 7(1):32-36.

[4] M. P. Papazoglou, P.Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," IEEE Computer, 2007,40(11):38-45.

[5] Yutu Liu, Anne H. H. Ngu, Liangzhao Zeng, "QoS Computation and Policing in Dynamic Web Service Selection," ACM, 2004

[6] Al-Masri, E., and Mahmoud, Q. H., "QoS-based Discovery and Ranking of Web Services," IEEE 16th International Conference on Computer Communications and Networks (ICCCN), 2007, pp. 529-534.

[7] Z. Zheng, H. Ma, M.R. Lyu, and I.King, "WSRec: a collaborative filtering based web service recommendation system," Proc. 7th International Conference on Web Services (ICWS 2009), pp. 437-444, 2009.

[8] L.Shao, J.Zhang, Y.Wei, J.Zhao, B.Xie, and H.Mei, "Personalised QoS prediction for web services via collaborative filtering," Proc. 5th International Conference on Web Services (ICWS 2007), pp. 439-446, 2007.

[9] Jianxin Li, Jinpeng Huai, Chunming Hu, Yanming Zhu. A Secure Collaboration Service for Dynamic Virtual Organizations, Elsevier, Information Sciences, v 180, n 17, p 3086-3107, September 1, 2010

[10] G. Linden, B. Smith, and J. York. "Amazon.com recommendations: item-to-item collaborative filtering," IEEE Internet Computing, Jan./Feb. 2003.

[11] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. "Item-based collaborative filtering recommendation algorithms," In WWW, pages 285–295, 2001.

[12] Ortega M, Rni Y, Chakrabarti k, Mehrotra S, and Huang TS. "Supporting Similarity Queries in MARS," Proc. ACM Multimedia, pages 403–413, 1997.

[13] M. G. Kendall. "A new measure of rank correlation," Biometrika, 30(1/2):81–93, 1938.

[14] E. Yilmaz, J. A. Aslam, and S. Robertson. "A new rank correlation coefficient for information retrieval," In SIGIR, pages 587–594, 2008.

[15] W. W. Cohen, R. E. Schapire, and Y. Singer. "Learning to order things," Journal of Artificial Intelligence Research, 5:243‑270, 1999.

[16] N. N. Liu and Q. Yang. "Eigenrank: a ranking-oriented approach to collaborative filtering," In SIGIR, pages 83‑90, 2008.

[17] V. Cardellini, E. Casalicchio, V. Grassi, and F. L. Presti. "Flow-based service selection for web service composition supporting multiple qos classes," Proc. 5th International Conference on Web Services (ICWS 2007), pp. 743-750.

[18] J. E. Haddad, M. Manouvrier, G. Ramirez, and M. Rukoz. "Qos-driven selection of web services for transactional composition," Proc. 7th International Conference on Web Services (ICWS 2009), pp. 653-660.

[19] E.Rich, "User modeling via stereotypes," Cognitive Science, Vol.3, No.4, 1979.

[20] J. Li, B. Li, T. Wo, C. Hu, J. Huai, L. Liu, K.P. Lam, CyberGuarder: A virtualization security assurance architecture for green cloud computing, Future Generation Computer Systems (2011), doi:10.1016/j.future.2011.04.012

[21] RS. Pindyck, and DL. Rubinfeld. "Econometrics Models and Economic Forecasts," McGraw-Hill, ISBN: 0-07-118831-2, 2000