2016 IEEE International Conference on Web Services

Asymmetric Correlation Regularized Matrix Factorization for Web Service Recommendation

Qi Xie*, Shenglin Zhao†, Zibin Zheng‡, Jieming Zhu† and Michael R. Lyu†
*School of Computer Science and Technology, Southwest University for Nationalities, Chengdu, China
†Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China
‡School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
Email: qi.xie.swun@gmail.com, {slzhao, zbzheng, jmzhu, lyu}@cse.cuhk.edu.hk

***Abstract*—Web service recommendation has recently drawn much attention with the growing amount of Web services. Previous work usually exploits the collaborative filtering techniques for Web service recommendation, but suffers from the data sparsity problem that leads to inaccurate results. Our analysis on a real-world Quality of Service (QoS) dataset shows that there is a hidden correlation among users and services. We define such hidden correlation with an asymmetric matrix (namely asymmetric correlation), in which each entry presents the hidden correlation between a user pair or between a service pair. The goal of this work is to employ such asymmetric correlation among users and services to alleviate the data sparsity problem and further enhance the prediction accuracy in service recommendation. Specifically, we propose an asymmetric correlation regularized matrix factorization (MF) framework, in which asymmetric correlation and asymmetric correlation propagation have been naturally integrated. Finally, experimental results on a well-known real-world QoS dataset validate that the use of asymmetric correlation among users and services is effective in improving prediction accuracy for Web service recommendation.**

***Index Terms*—Web service; service recommendation; matrix factorization; collaborative filtering; random walk**

I. Introduction

Web services on the Internet are abundant due to the wide adoption of service-oriented architectures and cloud platforms [1]–[3]. It becomes challenging for users to find the optimal Web services satisfying their needs from the large number of services with the same functional characteristics. Therefore, effective recommendation approaches arise to help users analyze the available information for better service selection and application development.

To recommend the optimal Web services for target users, Collaborative Filtering (CF) based Web service recommendation approaches have been widely studied in recent years [1], [4]–[11]. These approaches typically employ historical QoS (Quality of Service) usage data to predict the unobserved QoS values. However, they usually suffer from the data sparsity problem. In practice, the available training data are usually highly sparse (e.g., two users have only a few co-invoked Web services) and can lead to inaccurate prediction results. For example, one critical issue for the memory-based CF [8] is to find the best similar users or services through calculating the Pearson Correlation Coefficient (PCC) between each pair of users and services. But it is likely that the similarity measured by PCC is inaccurately estimated when the historical QoS data are sparse [1], [12].

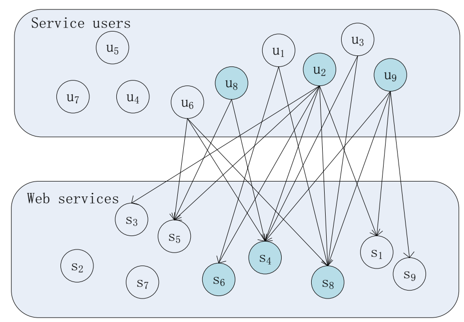


Fig. 1. Web Services Invocation Scenario

Recently, the correlation among users and services is incorporated in CF approaches to circumvent the data sparsity problem [1], [7], [13], [14]. Those CF approaches incorporated the correlation based on the intuition that a pair of services share higher similarity if they are commonly invoked by more users [13]. As shown in Fig. 1, user u_2 has invoked service s_4 and s_6, and we want to predict the QoS value of user u_2 invoking service s_8. Fig. 1 shows that 4 users invoking service pair (s_8, s_4) is larger than 2 users invoking service pair (s_8, s_6). Then it is natural to suppose that the QoS value of s_4 is liable to be a better predictor for service s_8 than the QoS value of s_6 [1], [13]. Existing approaches utilize a significance weight to decrease the influence of a small number of similar co-invoking users or co-invoked services. However, these approaches only consider the correlation to improve the computation of similarity.

Motivated by the above intuition, we exploit a hidden correlation among users and services named as Asymmetric Correlation (AC) in this paper. The asymmetric correlation is defined with an asymmetric matrix, where each entry presents the hidden correlation between a user pair or between a service pair. Furthermore, the higher value of asymmetric correlation user pair (service pair) is, the more similar Web service invocation experience user pair (service pair) enjoys.

The goal of this work is to employ such asymmetric correlation among users and services to alleviate the data sparsity in existing approaches, and to further enhance the prediction accuracy in service recommendation. Moreover, we also explore the following problems:

978-1-5090-2675-3/16 $31.00 © 2016 IEEE
DOI 10.1109/ICWS.2016.34

- Can we exploit the asymmetric correlation with matrix factorization to improve the recommendation performance, when the available data are sparse?
- Can we take advantages of both user asymmetric correlation and service asymmetric correlation?

To address these problems, this paper proposes an asymmetric correlation regularized Matrix Factorization (MF) framework for Web service recommendation, in which asymmetric correlation and asymmetric correlation propagation have been naturally integrated under three models. Specifically, this paper makes the following contributions:

- We construct both user asymmetric correlation and service asymmetric correlation by taking into account dependent relations and propagation.
- We systematically demonstrate how to design three asymmetric correlation based models of MF, in which asymmetric correlation among both users and services has been naturally integrated.
- We evaluate our models on a well-known real-world QoS dataset, and the results show that both user asymmetric correlation and service asymmetric correlation are effective in enhancing the prediction accuracy.

The rest of this paper is organized as follows. Section II presents the problem description. Section III details the asymmetric correlation regularized matrix factorization. Section IV shows the experimental results. Section V describes related work and Section VI concludes the paper.

II. Problem Description

In this section, we demonstrate the asymmetric correlation regularized matrix factorization via a toy example. Before describing the example, we define several terms first.

Definition 1 (Co-occurrence criterion): *Co-occurrence criterion* describes the co-invoked services which are invoked by user pairs or co-invoking users who invoke services pairs, shown in Fig. 2(b) and Fig. 2(c) respectively.

Definition 2 (Dependent relation): *Dependent relation* contains not only the *co-occurrence criterion* of each pair but also the *co-occurrence criterion* of the other pairs. *Dependent relation* can be obtained by constructing the stochastic matrix illustrated in Fig. 2(d) and Fig. 2(e) respectively.

Definition 3 (Asymmetric correlation): *Asymmetric correlation* is the hidden correlation among users and services defined with an asymmetric matrix. And every entry in asymmetric matrix presents the *dependent relation* related to a *co-occurrence criterion* of user pair or service pair.

To illustrate the concept of asymmetric correlation regularized matrix factorization more clearly, a toy example is given in Fig. 2. According to Web services invocation scenario in Fig. 1, we extract three users (u_2, u_8, u_9) and three services (s_4, s_6, s_8) to construct a toy 3×3 user-service QoS matrix. Furthermore, Fig. 2(a) shows the 3×3 user-service QoS matrix where every entry $r_{i,j}$ represents the QoS values (e.g., response time, throughput, etc.) invoked by user u_i on service s_j.

Based on the intuition that the higher value of asymmetric correlation user pair (service pair) is, the more similar Web service invocation experience user pair (service pair) enjoys, the overview of asymmetric correlation regularized matrix factorization is illustrated in the toy example as follows.

- Firstly, the 3×3 user-user correlation matrix and the 3×3 service-service correlation matrix can be constructed according to the *co-occurrence criterion* in 3×3 user-service QoS matrix, which are shown in Fig. 2(b) and Fig. 2(c) respectively.
- Secondly, as shown in Fig. 2(d) and Fig. 2(e), user asymmetric correlation matrix and service asymmetric correlation matrix can be obtained separately, by using corresponding correlation matrices and *dependent relation* of correlation.
- And then, asymmetric correlation propagation is taken into consideration by employing random walk algorithm. Thus, the asymmetric correlation ranks of users and services are produced respectively as shown in Fig. 2(f) and Fig. 2(g).
- After that, three asymmetric correlation based regularization terms shown in Fig. 2(h) are designed by employing the sets of Top-*K* high rank values of user asymmetric correlation and service asymmetric correlation. That can construct corresponding three unified MF models where *asymmetric correlation* among both users and services is systematically integrated.

The problem we address in this paper is how to predict the missing QoS values of the user-service QoS matrix and recommend the optimal Web services for target users effectively by exploiting asymmetric correlation among both users and services.

III. Asymmetric Correlation Regularized Matrix Factorization

A. Asymmetric Correlation Calculation

Given an $m \times n$ user-service QoS matrix including m users and n services, we calculate the asymmetric correlation by two steps: asymmetric correlation matrix construction and asymmetric correlation propagation.

1) Asymmetric Correlation Matrix Construction: We first define the symmetric User Correlation Matrix (UCM) and the symmetric Service Correlation Matrix (SCM). $U(u_i, u_f)$ is a set of services which invoked by both user u_i and user u_f with a co-occurrence criterion, and if $u_i = u_f$ then the value of $U(u_i, u_f)$ is $\emptyset$. It is formulated as

$$U(u_i, u_f) = \begin{cases} \{s_g : (r_{i,g} \neq \emptyset) \wedge (r_{f,g} \neq \emptyset)\} & (u_i \neq u_f) \\ \emptyset & (u_i = u_f) \end{cases} \quad (1)$$

where g is in the range of $[1, n]$, and $r_{i,g}$ and $r_{f,g}$ are the vectors of QoS values of service g invoked by user u_i and u_f respectively. Further, we define the entry of UCM as

$$UCM_{u_i,u_f} = |U(u_i, u_f)| \quad i, f = 1, \ldots, m, \quad (2)$$

where $|U(u_i, u_f)|$ is the number of $U(u_i, u_f)$.

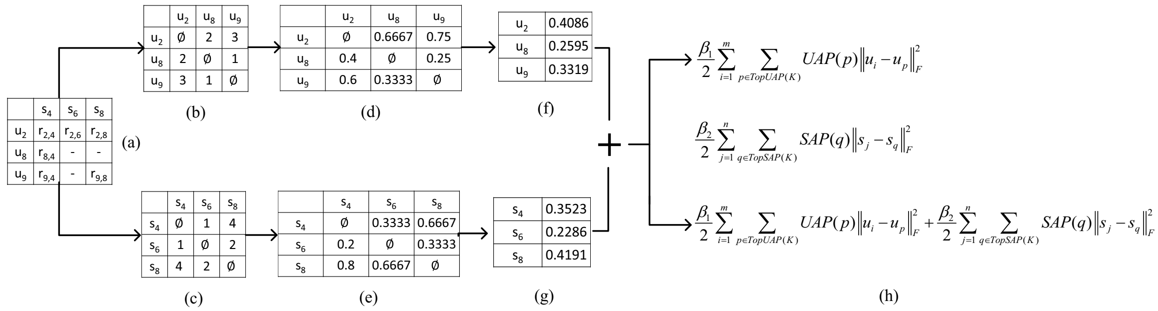


Fig. 2. A Toy Example

Similar to user correlation matrix, the service correlation matrix is defined as:

$$SCM_{s_j,s_g} = |S(s_j, s_g)| \quad j, g = 1, \ldots, n, \tag{3}$$

where $S(s_j, s_g)$ is a set of users which invoke both service s_j and service s_g.

By incorporating the dependent relations among users and services, we define the asymmetric UCM and SCM. Dependent relations can be exploited from above correlation matrices by constructing a stochastic matrix. That is to say, each entry in the asymmetric correlation matrix is related to not only the numbers of co-occurrence criterion of each pair but also the numbers of co-occurrence criterion of the other pairs. Therefore, the Asymmetric Correlation of user u_i and u_f is defined by the ratio of entry UCM_{u_i,u_f} among the sum of entries in f*-th* column of user correlation matrix. In formal, we define the User Asymmetric Correlation Matrix ($UACM$) as

$$UACM_{u_i,u_f} = \begin{cases} \frac{|U(u_i,u_f)|}{\sum\limits_{s\in m}|U(u_s,u_f)|} & (u_i \neq u_f) \\ \emptyset & (u_i = u_f) \end{cases}, \tag{4}$$

where i, f, and s are in the range of $[1, m]$, and the matrix entry $UACM_{u_i,u_f}$ presents the dependent relation of correlation between user u_i and user u_f.

Likewise, we define the service asymmetric correlation matrix ($SACM$) as:

$$SACM_{s_j,s_g} = \begin{cases} \frac{|S(s_j,s_g)|}{\sum\limits_{l\in n}|S(s_l,s_g)|} & (s_j \neq s_g) \\ \emptyset & (s_j = s_g) \end{cases}, \tag{5}$$

where j, g and l are in the range of $[1, n]$, the matrix entry $SACM_{s_j,s_g}$ presents the dependent relation of correlation between service s_j and service s_g.

2) Asymmetric Correlation Propagation: The asymmetric correlation propagation is the key to exploit the asymmetric correlation. Following [14], we leverage the asymmetric correlation propagation to find the top users and services through the random walk algorithm. Observing from (4) and (5), we see that the $UACM$ and $SACM$ are the stochastic matrices which can be considered as transition probability matrices in the random walk algorithm. Then, every entry in $UACM$ represents the probability of user u_i with the next state u_f, similar to matrix $SACM$.

Like computing the PageRank [15], the asymmetric correlation of users and services can spread throughout the corresponding transition probability matrices. Then the random walk algorithm with User Asymmetric correlation Propagation (UAP) is designed by employing the following equations:

$$\begin{cases} UAP(u_i^0) = \frac{1}{|m|} \cdot 1_{|m|} \\ UAP(u_i^{t+1}) = a \cdot UACM \cdot UAP(u_i^t) + (1-a) \cdot \frac{1}{|m|} \cdot 1_{|m|}, \end{cases} \tag{6}$$

where $UAP(u_i)$ is a vector with length of m, $1_{|m|}$ is a $|m|$ long vector of ones, and a is a decay factor which is set to 0.85. $UAP(u_i)$ means the rank value of asymmetric correlation for user u_i, and $UACM$ is the user asymmetric correlation matrix. At the beginning of the iteration, the initial value of $UAP(u_i)$ is set to $UAP(u_i^0)$ which means that all users obtain a same rank value. Then, $UAP(u_i)$ iterates finite steps to convergence.

Likewise, the random walk algorithm with Service Asymmetric correlation Propagation (SAP) is defined as:

$$\begin{cases} SAP(s_j^0) = \frac{1}{|n|} \cdot 1_{|n|} \\ SAP(s_j^{t+1}) = a \cdot SACM \cdot SAP(s_j^t) + (1-a) \cdot \frac{1}{|n|} \cdot 1_{|n|}, \end{cases} \tag{7}$$

where $SAP(s_j)$ is a vector with length of n, $1_{|n|}$ is a $|n|$ long vector of ones, and a is also a decay factor set to 0.85. $SACM$ is the service asymmetric correlation matrix, and $SAP(s_j)$ means the rank value of asymmetric correlation for service s_j. We calculate the value of $SAP(s_j)$ via the same strategy of computing $UAP(u_i)$.

From (6) and (7), we see that the values UAP and SAP generate a rank of users and services according to their Web service invocation experience after a random walk. The higher is the rank value of UAP (SAP), the higher is the probability that the user (service) is popular with services (users) [16].

B. Asymmetric Correlation based Regularization

Three asymmetric correlation based regularization terms are constructed from UAP and SAP to enhance the Web service recommendation. According to [14], [16], target users (services) are liable to have similar Web service invocation experience with popular users (services) which have high rank values of asymmetric correlation. Based on this intuition, a set of users $TopUAP(K)$ with top-K high rank values of user asymmetric correlation and a set of services $TopSAP(K)$

with top-K high rank values of service asymmetric correlation can be discovered in the UAP and the SAP respectively.

1) User Asymmetric Correlation based Regularization (UACR): We define $TopUAP(K)$ as the set of users with top-K high rank values of user asymmetric correlation except the service user u_i itself. We propose a user asymmetric correlation based regularization term as follows:

$$\frac{\beta_1}{2}\sum_{i=1}^{m}\sum_{p\in TopUAP(K)} UAP(p)\left\|u_i-u_p\right\|_F^2 \qquad (u_p\neq u_i), \tag{8}$$

where user u_p belongs to $TopUAP(K)$, and β_1 is the regularization parameter with $\beta_1>0$. $UAP(p)$ is a vector of the rank values of user asymmetric correlation for user u_p.

Furthermore, (8) is employed to minimize the invocation preference between user u_i and other popular users with top-K high rank values of user asymmetric correlation. As a consequence, we gain the first asymmetric correlation regularized matrix factorization model,

$$\begin{aligned}\min_{U,S}\mathcal{L}_1(R,U,S)=&\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}(R_{ij}-U_i^TS_j)^2\\&+\frac{\beta_1}{2}\sum_{i=1}^{m}\sum_{p\in TopUAP(K)}UAP(p)\left\|u_i-u_p\right\|_F^2\\&+\frac{\lambda_1}{2}\left\|U\right\|_F^2+\frac{\lambda_2}{2}\left\|S\right\|_F^2,\end{aligned} \tag{9}$$

where R is the $m\times n$ user-service QoS matrix including m users and n services, in which every entry represents a vector of QoS values. The R matrix can be approximately divided into two submatrices which are $U\in\mathbb{R}^{d\times m}$ and $S\in\mathbb{R}^{d\times n}$ with dimensionality $d<min(m,n)$. $\|\cdot\|$ represents the $Frobenius\ norm$. I_{ij} is a indicator function. When service s_j is invoked by user u_i, the I_{ij} is set to 1 and set to 0 otherwise. In addition, two regularization terms including $\frac{\lambda_1}{2}\left\|U\right\|_F^2$ and $\frac{\lambda_2}{2}\left\|S\right\|_F^2$ are employed to avoid overfitting where $\lambda_1,\lambda_2>0$

In order to obtain the local minimum of the above objective· function, we exploit the gradient descent algorithm to learn U_i and S_j,

$$\begin{aligned}\frac{\partial\mathcal{L}_1}{\partial U_i}&=\sum_{j=1}^{n}I_{ij}(U_i^TS_j-R_{ij})S_j+\lambda_1U_i\\&\quad+\beta_1\sum_{p\in TopUAP(K)}UAP(p)(u_i-u_p),\\\frac{\partial\mathcal{L}_1}{\partial S_j}&=\sum_{i=1}^{m}I_{ij}(U_i^TS_j-R_{ij})U_i+\lambda_2S_j.\end{aligned} \tag{10}$$

2) Service Asymmetric Correlation based Regularization (SACR): Similar to user asymmetric correlation based regularization, the regularization can be expanded by taking advantages of service asymmetric correlation. And the service asymmetric correlation based regularization term can defined as follows:

$$\frac{\beta_2}{2}\sum_{j=1}^{n}\sum_{q\in TopSAP(K)}SAP(q)\left\|s_j-s_q\right\|_F^2 \qquad (s_q\neq s_j), \tag{11}$$

where $SAP(q)$ is the set of rank values of service asymmetric correlation for service s_q. And service s_q is the set of services who belong to $TopSAP(K)$ except the service s_j itself. In addition, β_2 is also the regularization parameter with $\beta_2>0$. Moreover, (11) is employed to minimize the invocation preference between service s_j and other popular services with top-K high rank values of service asymmetric correlation.

Same as the first model, the second asymmetric correlation regularized matrix factorization model can be formulated by:

$$\begin{aligned}\min_{U,S}\mathcal{L}_2(R,U,S)=&\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}(R_{ij}-U_i^TS_j)^2\\&+\frac{\beta_2}{2}\sum_{j=1}^{n}\sum_{q\in TopSAP(K)}SAP(q)\left\|s_j-s_q\right\|_F^2\\&+\frac{\lambda_1}{2}\left\|U\right\|_F^2+\frac{\lambda_2}{2}\left\|S\right\|_F^2.\end{aligned} \tag{12}$$

The gradient descent algorithm of both U_i and S_j is used to find a local minimum of the objective function in (12) by employing the following equations:

$$\begin{aligned}\frac{\partial\mathcal{L}_2}{\partial U_i}&=\sum_{j=1}^{n}I_{ij}(U_i^TS_j-R_{ij})S_j+\lambda_1U_i,\\\frac{\partial\mathcal{L}_2}{\partial S_j}&=\sum_{i=1}^{m}I_{ij}(U_i^TS_j-R_{ij})U_i+\lambda_2S_j\\&\quad+\beta_2\sum_{q\in TopSAP(K)}SAP(p)(s_j-s_q).\end{aligned} \tag{13}$$

3) Hybrid Asymmetric Correlation based Regularization (HACR): From subsection III-B1 and subsection III-B2, we can take advantages of both user asymmetric correlation and service asymmetric correlation. Asymmetric correlation will possibly benefit the Web service recommendation. Then, the hybrid model can be constructed as the third asymmetric correlation regularized matrix factorization model by:

$$\begin{aligned}\min_{U,S}\mathcal{L}_3(R,U,S)=&\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}I_{ij}(R_{ij}-U_i^TS_j)^2\\&+\frac{\beta_1}{2}\sum_{i=1}^{m}\sum_{p\in TopUAP(K)}UAP(p)\left\|u_i-u_p\right\|_F^2\\&+\frac{\beta_2}{2}\sum_{j=1}^{n}\sum_{q\in TopSAP(K)}SAP(q)\left\|s_j-s_q\right\|_F^2\\&+\frac{\lambda_1}{2}\left\|U\right\|_F^2+\frac{\lambda_2}{2}\left\|S\right\|_F^2,\end{aligned} \tag{14}$$

where the objective function $\mathcal{L}_3$ minimizes the invocation preferences among both users and services.

Thus, the local minimum of the objective function $\mathcal{L}_3$ can also be searched by employing gradient descent algorithm of both U_i and S_j as:

$$\begin{aligned} \frac{\partial \mathcal{L}_1}{\partial U_i} &= \sum_{j=1}^{n} I_{ij}(U_i^T S_j - R_{ij}) S_j + \lambda_1 U_i \\ &+ \beta_1 \sum_{p \in TopUAP(K)} UAP(p)(u_i - u_p), \\ \frac{\partial \mathcal{L}_2}{\partial S_j} &= \sum_{i=1}^{m} I_{ij}(U_i^T S_j - R_{ij}) U_i + \lambda_2 S_j \\ &+ \beta_2 \sum_{q \in TopSAP(K)} SAP(p)(s_j - s_q), \end{aligned} \tag{15}$$

where we set the $\lambda_1 = \lambda_2$ in all experiments for simplicity.

IV. Experiment

In this section, we conduct experiments to explore the following questions:

- Is asymmetric correlation effective in enhancing the performance of Web service recommendation by using MF?
- Can we take advantages of both user asymmetric correlation and service asymmetric correlation?
- What are the effects of parameters on the models performance?

A. Dataset Description

We employ the well-known real-world QoS dataset provided by Zheng et al. [1] to study the prediction accuracy of our proposed approaches. The dataset contains about 1,974,675 Web service response time invocations from 339 users on 5825 Web services.

In order to keep consistence with the real-world QoS data sparsity, we randomly choose different number of entries, named *Given Service* (*GS*), from 5825 Web services. Then, the new $339 \times GS$ user-service QoS matrix is divided into two parts, one part as the training matrix which covers 80% of $339 \times GS$ matrix and the other part as the test matrix. As to the training matrix, we randomly remove entries to gain different densities. There are 9 parameters in our experiments including a, β_1, β_2, λ_1, λ_2, top-*K*, *GS*, *denstiy*, and *dimensionality*. The parameter settings are as follows, $\lambda_1 = \lambda_2 = 30$, $a = 0.85$, and $\beta_1 = \beta_2 = \beta$ in all experiments for simplicity.

B. Metrics

Two popular metrics, Mean Absolute Error (MAE) and Normalized Mean Absolute Error (NMAE) are employed to measure the prediction accuracy by using the following equations:

$$MAE = \frac{\sum_{i,j} |R_{i,j} - \hat{R}_{i,j}|}{N}, \tag{16}$$

$$NMAE = \frac{MAE}{\sum_{i,j} R_{i,j}/N}, \tag{17}$$

where $R_{i,j}$ and $\hat{R}_{i,j}$ denote the real QoS value and the predicted QoS value of service *j* invoked by user *i* respectively. And N is the number of missing QoS values in user-service QoS matrix.

C. Comparison

To study the effectiveness of our asymmetric correlation regularized MF approaches (UACR, SACR and HACR), we compare the performance with four state-of-the-art Web service recommendation approaches: UPCC [4], IPCC [1], WSRec [1], and PMF [17]. UPCC and IPCC are user-based CF approach and item-based CF approach respectively, which use PCC to measure the similarity and find the best services. When measuring the performance of UPCC and IPCC, we set the negative predicted QoS values as 0 to calculate MAE and NMAE. WSRec is a model that integrates both UPCC and IPCC systematically. PMF namely probabilistic matrix factorization is a state-of-the-art MF model for recommendation task. In this paper, the compared four approaches and our proposed three approaches employ the same error measures and the same datasets.

Table 1 shows the performance comparison of seven approaches on response time with $\beta = 30$, *density=10%*, *dimensionality=10*, *GS* from 1000 to 3000 with a step value of 1000. From Table 1, we draw the following conclusions:

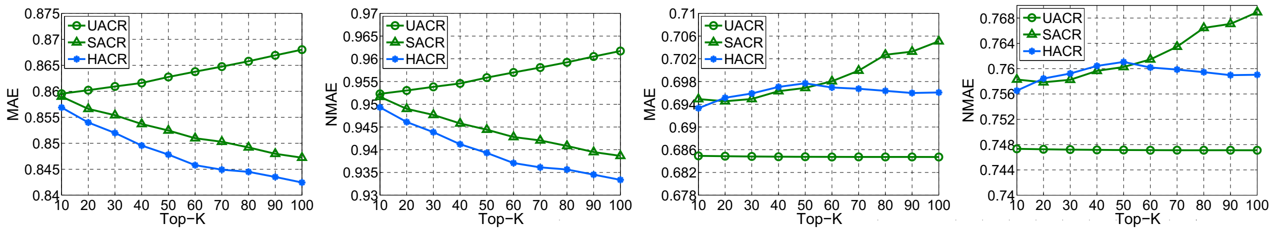
- When *GS=1000*, we set top-*K*=60 for UACR, top-*K*=20 for SACR and top-*K*=10 for HACR. Table 1 shows that our proposed three approaches obtain smaller MAE and NMAE values than other approaches, and UACR gains the smallest MAE and NMAE values than others. This observation shows that both user asymmetric correlation and service asymmetric correlation can enhance the performance of Web service recommendation effectively by using MF.
- When *GS=2000* and *GS=3000*, we set top-*K*=60 for UACR, top-*K*=70 for SACR and top-*K*=50 for HACR. As shown in Table 1, SACR and HACR approaches outperform other five approaches. This observation shows that service asymmetric correlation based approaches can obtain better prediction accuracy.
- Our three approaches by using asymmetric correlation can obtain smaller MAE and NMAE values than others, which indicates better prediction accuracy. That confirms the asymmetric correlation is effective in enhancing the performance of Web service recommendation.
- Moreover, it also can be seen from Table 1 that our three approaches experience a downward trend with the increase GS from 1000 to 3000. This indicates that more QoS values can improve the prediction accuracy. In addition, the HACR can obtain smaller MAE and NMAE values. This observation shows that taking advantages of both user and service asymmetric correlation in a unified model can also enhance the prediction accuracy.

D. Impact of Top-K

In our asymmetric correlation regularized matrix factorization approaches, the parameter top-*K* is employed to control

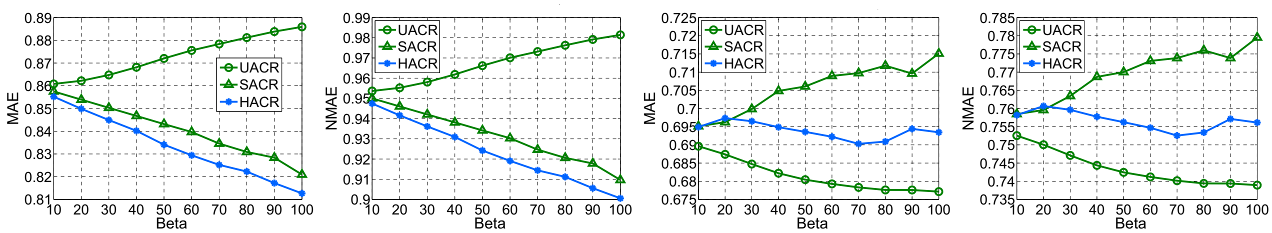
TABLE I
PERFORMANCE COMPARISON

GS	Metric	Methods						
		UPCC	IPCC	WSRec	PMF	UACR	SACR	HACR
1000	MAE	0.8337	0.6991	0.7050	0.7019	0.6847	0.6945	0.6933
	NMAE	0.9204	0.7763	0.7742	0.7755	0.7471	0.7578	0.7565
2000	MAE	0.8437	0.7075	0.7191	0.6303	0.6417	0.6279	0.6275
	NMAE	0.9216	0.7761	0.7726	0.6885	0.7010	0.6860	0.6855
3000	MAE	0.8352	0.7071	0.7115	0.5958	0.6100	0.5945	0.5943
	NMAE	0.9199	0.7820	0.7801	0.6563	0.6718	0.6548	0.6546



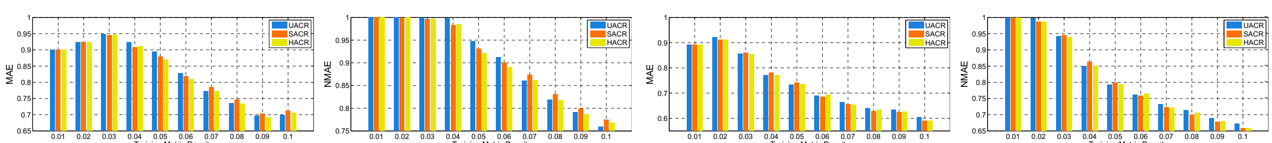
(a) Density=5% (b) Density=5% (c) Density=10% (d) Density=10%

Fig. 3. Impact of Top-*K*



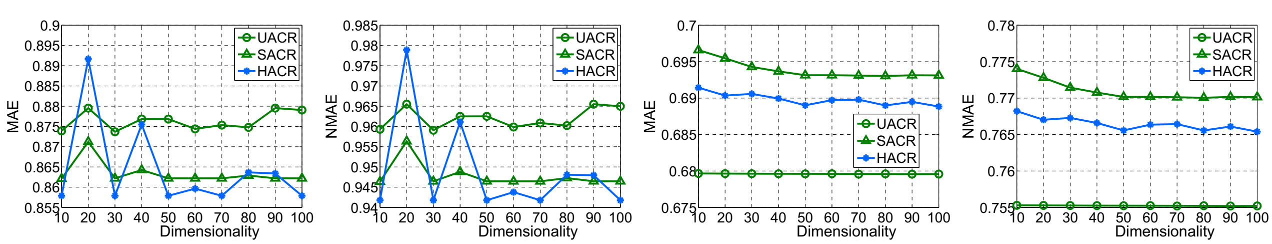
(a) Density=5% (b) Density=5% (c) Density=10% (d) Density=10%

Fig. 4. Impact of Beta



(a) GS=1000 (b) GS=1000 (c) GS=3000 (d) GS=3000

Fig. 5. Impact of Training Matrix Density



(a) Density=5% (b) Density=5% (c) Density=10% (d) Density=10%

Fig. 6. Impact of Dimensionality

the number of popular entries which include users and services. Fig. 3 illustrates the impact of top-*K* on the prediction accuracy of our three approaches (UACR, SACR and HACR) with $\beta = 30$, *GS=1000*, *dimensionality=10*, *density=5%*, *density=10%*, and top-*K* from 10 to 100 with a step size of 10.

When *density=5%*, it can be seen from the Fig. 3 that the MAE and NMAE values of both SACR and HACR become smaller with the increase of top-*K*, indicating larger popular services provides more information for recommendation. Moreover, UACR can obtain the optimal performance with top-*K*=10, while the SACR and HACR can obtain the optimal performance when top-*K*=100. This indicates that popular services provide more information than users for recommendation. Also when *density=10%*, UACR obtains better performance consistently versus SACR and UACR. It shows that the performance of SACR, HACR and UACR has been affected differently with the increase of top-*K*. Whatever the values of top-*K* are, one of our three approaches can obtain the optimal performance, indicating AC based approaches can effectively enhance the performance of Web service recommendation.

E. Impact of Beta

In our approaches, β_1 and β_2 are the regularization parameters to determine how much the asymmetric correlation of users and services influences to the objective functions respectively. Moreover, we set top-*K*=70, *GS=1000*, *dimensionality=10*, *density=5%* and *density=10%* respectively. And we also vary β from 10 to 100 with a step value of 10. As shown in Fig. 4, when *density=5%*, the performance of SACR and HACR becomes better but the performance of UACR becomes worse with the increase of β. And when *density=10%*, the performance of SACR and HACR becomes little worse but the performance of UACR becomes stable with better results with the increase of β. This observation shows that the values of β affect our three approaches differently, indicating AC has an important impact on objective functions. Also whatever the values of β are, one of our three approaches can obtain the optimal performance. It indicates that AC based approaches can effectively enhance the performance.

F. Impact of Training Matrix Density

To study the impact of training matrix density, we set $\beta = 30$, *top*-K=*70*, *GS=1000*, *GS=3000* and change the *density* from 1% to 10% with a step of 1%. Fig. 5 shows that our proposed three approaches experience a downward trend with the increase of *density* from 1% to 10%, indicating that more QoS values can enhance the prediction accuracy and the AC is effective in enhancing the performance. Moreover, the UACR firstly performs little worse than SACR when *GS=1000* and *density* increases from 3% to 6%. After that the UACR performs better than SACR when *density* increases from 7% to 10%. This observation indicates that more QoS values can enhance the prediction accuracy of UACR with *GS=1000*. But as to *GS=3000*, the MAE and NMAE values of SACR become smaller than the ones of UACR with *density* increase from 1% to 10%, indicating that more Web services and QoS values can remarkably increase the performance accuracy of SACR with *GS=3000*.

G. Impact of Dimensionality

Aiming to discuss the influence of *dimensionality* which controls how many latent features related to MF, we set $\beta = 30$, *top*-K=*70*, *GS=1000*, *density=5%* and *density=10%* respectively. And we also tune the *dimensionality* from 10 to 100 with a step size of 10. Fig. 6(a) and Fig. 6(b) show SACR and HACR firstly fluctuate, after that they tend to gentle with the growth of *dimensionality* from 50 to 100, indicating that *dimensionality* has an important impart on the stability and accuracy of prediction performance. In addition, Fig. 6(c) and Fig. 6(d) show that the performance of our three asymmetric correlation based approaches are stable with the increase of *dimensionality* from 10 to 100 when *density=10%* and *GS=1000*. This observation also indicates that more QoS values may enhance the stability of prediction performance.

V. Related Work

In this section, we review some approaches on Web service recommendation, which mainly include two types of CF-based approaches: memory-based [1], [6], [7] and model-based [18], [19]. Currently, some matrix factorization approaches have been presented for Web service recommendation [6], [8], [19]. However, CF-based service recommendation approaches suffer from the data sparsity. Then context information such as location and time is considered to tackle this problem [8], [10], [14]. But context information may be missing for privacy protection of Web services. Hence, researchers attempt to dig correlation hidden in user-service QoS matrix to solve this problem [1], [7], [13], [20], [21].

Some approaches have been proposed to by incorporating the correlation among users or services to improve the deviation or similarity [1], [7], [13]. D. Lemire et al. [13] and Zheng et al. [1] employ the number of similar co-invoked items or co-invoking users to improve the deviation or similarity. Jiang et al. [7] employ the degree of users and services to compute the personalized similarity. Gao et al. [22] incorporate the user weight into the computation of item similarities and differentials. Hu et al. [14] perform a hybrid personalized random walk algorithm to handle data sparsity by inferring more indirect user and service similarities. And Gori et al. [16] present ItemRank that exploits a random walk based scoring algorithm to rank products according to expected user preferences and recommend top-rank items to potentially interested users. Tang et al. [23] propose a random walk method combining location-aware and collaborative filtering method for Web service recommendation. These approaches only consider the correlation to improve the deviation or similarity, but fail to study the influence of the correlation on recommendation performance directly. In specific, the dependent relation and correlation propagation are not involved.

Our approaches are most related to [20], [21], [24]. Yet our proposed approaches distinguish these methods in model construction and means of incorporating the asymmetric correlation. Gong et al. [24] leverage a user relationship and preference, which are derived from the feedback from user to service. Different from this work using the relationship from user to service, we exploit the relationship from user to user and service to service respectively in user-service QoS matrix. Furthermore, the relationship of users and services in our approaches has different definition which is related to a co-occurrence criterion in Web services. In our previous work, [20] combines asymmetric correlation among users and propagation into the deviation computation of different service items in QoS prediction; [21] employs the asymmetric correlation among services to improve the process of choosing Top-K different services by using slope one approach. Different from these approaches, our proposed approaches in this paper leverage the asymmetric correlation both of users and services to construct three corresponding regularization terms which are integrated in an MF model.

VI. Conclusion and Future Work

In this paper, we aim to employ asymmetric correlation among users and services to enhance the prediction accuracy in Web service recommendation. Our methods are based on the intuition that the higher value of asymmetric correlation user pair (or service pair) is, the more similar Web service invocation experience user pair (or service pair) enjoys. We propose three asymmetric correlation based matrix factorization approaches for Web service recommendation. In the proposed approaches, we leverage both user asymmetric correlation and service asymmetric correlation in which asymmetric correlation propagation is taken into consideration via the random walk algorithm. Furthermore, the experimental analysis on a real-world QoS dataset shows that the asymmetric correlation among users and services indeed has an important effect on prediction accuracy.

In future, we will continue to optimize our models in terms of stability and prediction accuracy. Moreover, more QoS properties will be considered, and other contextual features of Web services (e.g., location, time, etc.) can be integrated into our framework to further enhance the performance of Web service recommendation.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61502401), the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14205214 of the General Research Fund), Guangdong Natural Science Foundation (Project No. 2014A030313151).

References

[1] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *IEEE Transactions on Service Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[2] S. Wang, Z. Zheng, Z. Wu, M. R. Lyu, and F. Yang, "Reputation measurement and malicious feedback rating prevention in web service recommendation systems," *IEEE Transactions on Service Computing*, vol. 8, no. 5, pp. 755–767, 2015.

[3] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Towards online, accurate, and scalable qos prediction for runtime service adaptation," in *Proc. 34th Int'l Conf. Distributed Computing Systems ICDCS'14*, 2014, pp. 318–327.

[4] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering," in *Proc. 5th Int'l Conf. Web Services (ICWS'07)*, 2007, pp. 439–446.

[5] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based Web service recommender system," in *Proc. 7th Int'l Conf. Web Services (ICWS'09)*, 2009, pp. 437–444.

[6] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-aware Web service recommendation and visualization," *IEEE Transactions on Services Computing*, vol. 6, no. 1, pp. 35–47, 2013.

[7] Y. Jiang, J. Liu, M. Tang, and X. Liu, "An affective web service recommendation method based on personalized collaborative filtering," in *Proceedings of the 18th IEEE International Conference on Web Services (ICWS'11)*. IEEE, 2011, pp. 211–218.

[8] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, "Collaborative Web service qos prediction with location-based regularization," in *Proceedings of the 19th IEEE International Conference on Web Services (ICWS'12)*. IEEE, 2012, pp. 464–471.

[9] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-based hierarchical matrix factorization for Web service recommendation," in *Proceedings of the 21th IEEE International Conference on Web Services (ICWS'14)*. IEEE, 2014, pp. 297–304.

[10] K. Lee, J. Park, and J. Baik, "Location-based Web service QoS prediction via preference propagation for improving cold start problem," in *Proceedings of the 22th IEEE International Conference on Web Services (ICWS'15)*. IEEE, 2015, pp. 177–184.

[11] Z. Zheng, H. Ma, M. R. Lyu, and 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.

[12] M. R. McLaughlin and J. L. Herlocker, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience," in *Proc. 27th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'04)*, 2004, pp. 329–336.

[13] D. Lemire and A. Maclachlan, "Slope one predictors for online rating-based collaborative filtering," in *SDM*, vol. 5. SIAM, 2005, pp. 1–5.

[14] Y. Hu, Q. Peng, and X. Hu, "A time-aware and data sparsity tolerant approach for Web service recommendation," in *Proceedings of the 21th IEEE International Conference on Web Services (ICWS'14)*. IEEE, 2014, pp. 33–40.

[15] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web." 1999.

[16] M. Gori, A. Pucci, V. Roma, and I. Siena, "Itemrank: A random-walk based scoring algorithm for recommender engines," in *International Joint Conference on Artificial Intelligence*, vol. 7, 2007, pp. 2766–2771.

[17] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, vol. 20, 2008.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 931–940.

[19] H. Ma, "An experimental study on implicit social recommendation," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval(SIGIR'13)*. ACM, 2013, pp. 73–82.

[20] Q. Xie, B. Tang, Z. Zheng, and M. Cui, "Qos prediction in dynamic web services with asymmetric correlation," in *Proceedings of the 15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'15)*. Springer, 2015, pp. 445–459.

[21] Q. Xie, Z. Zheng, L. Liu, and M. Cui, "Correlation-based top-k recommendation for web services," in *Proceedings of the 13th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'15)*. IEEE, 2015, pp. 1903–1909.

[22] M. Gao, Z. Wu, and F. Jiang, "Userrank for item-based collaborative filtering recommendation," *Information Processing Letters*, vol. 111, no. 9, pp. 440–446, 2011.

[23] M. Tang, X. Dai, B. Cao, and J. Liu, "Wswalker: A random walk method for QoS-Aware Web service recommendation," in *Proc. 22th Int'l Conf. Web Services (ICWS'15)*. IEEE, 2015, pp. 591–598.

[24] M. Gong, Z. Xu, L. Xu, Y. Li, and L. Chen, "Recommending web service based on user relationships and preferences," in *Proc. 20th Int'l Conf. Web Services (ICWS'13)*. IEEE, 2013, pp. 380–386.