IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 16, 2017, accepted March 25, 2017, date of publication April 19, 2017, date of current version May 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2695657

QoS Recommendation in Cloud Services

XIANRONG ZHENG[1], (Member, IEEE), LI DA XU[1], (Fellow, IEEE), AND SHENG CHAI[2], (Member, IEEE)
[1]Information Technology and Decision Sciences Department, Old Dominion University, Norfolk, VA 23529 USA
[2]School of Computing, Queen's University, Kingston, ON K7L 2N8, Canada

Corresponding author: Xianrong Zheng (x1zheng@odu.edu)

ABSTRACT As cloud computing becomes increasingly popular, cloud providers compete to offer the same or similar services over the Internet. Quality of service (QoS), which describes how well a service is performed, is an important differentiator among functionally equivalent services. It can help a firm to satisfy and win its customers. As a result, how to assist cloud providers to promote their services and cloud consumers to identify services that meet their QoS requirements becomes an important problem. In this paper, we argue for QoS-based cloud service recommendation, and propose a collaborative filtering approach using the Spearman coefficient to recommend cloud services. The approach is used to predict both QoS ratings and rankings for cloud services. To evaluate the effectiveness of the approach, we conduct extensive simulations. Results show that the approach can achieve more reliable rankings, yet less accurate ratings, than a collaborative filtering approach using the Pearson coefficient.

INDEX TERMS Cloud computing, quality of service (QoS), recommender systems, collaborative filtering.

I. INTRODUCTION

Cloud computing refers to a large pool of virtualized resources that can be dynamically reconfigured to provide elastic services over the Internet [14]. It has the potential to increase business agility, improve efficiencies, and reduce costs. As cloud computing becomes increasingly popular, cloud providers, including leading IT companies like Amazon, Google, and Microsoft, compete to offer the same or similar services over the Internet.

As an example, Amazon Simple Storage Service (Amazon S3) offers durable and massively scalable object storage. Google Cloud Storage provides durable and highly available object storage. Microsoft Azure Storage provides reliable and economical storage for small and big data. Indeed, there are more than a dozen cloud providers offering online storage services, and the number is still growing.

As the cloud market becomes more open and competitive, quality will be more important. According to the American Society for Quality, quality is "the totality of features and characteristics of a product or service that bears on its ability to satisfy stated or implied needs." [5]. It can help companies to obtain a competitive advantage by improving business operations, building good reputation, reducing product liability, and competing effectively in the global economy.

In cloud computing, Quality of Service (QoS) is non-functional properties of cloud services, which describe how well a service is performed, such as availability, reliability, responsiveness, and security. Indeed, QoS is an important differentiator among functionally equivalent services. It can help a firm to satisfy and win its customers. As a result, how to assist cloud providers to promote their services and cloud consumers to identify services that meet their QoS requirements becomes an important problem.

Recommender systems, which have been developed to alleviate the information overload problem, can help users to find useful information and products. They can generate suggestions that match users' interests and preferences. Recommender systems are personalized information filtering techniques, which are employed to either predict whether a user will like an item (prediction problem) or find a set of items that will be of interest to a user (top-N recommendation problem) [3].

Collaborative Filtering (CF) is considered as the most successful technique to build recommender systems [2], [3], [7]. CF has been studied in electronic commerce for many years. It recommends items to users based on the opinions of a set of users sharing the same or similar interests. It can consider the quality of items, and can recommend serendipitous items to users, i.e., really good items that are not obvious to users [1].

In this paper, we argue for QoS-based cloud service recommendation. A rating-based CF approach using the Pearson coefficient can achieve accurate ratings, but may incur unreliable rankings. To address this issue, we propose a ranking-based CF approach using the Spearman coefficient to recommend cloud services. The approach can achieve more reliable rankings than a CF approach using the Pearson coefficient.

The paper's main contributions are as follows:

1) A ranking-based CF approach using the Spearman coefficient. The approach is used to predict both ratings and rankings;
2) Extensive simulations to evaluate the effectiveness of the approach. Results show that the approach can achieve reliable rankings.

The rest of the paper is structured as follows. Section II describes the CF approach using the Spearman coefficient. Section III reports simulation results. Section IV reviews related work. Section V concludes the paper.

II. RANKING-BASED CF USING THE SPEARMAN COEFFICIENT

A user-based CF approach, which can generate high quality recommendations [3], is adopted to build a recommender system for cloud services. The rationale is that each user belongs to a group of similarly behaving individuals. So, items frequently purchased by members of the group can be used to determine the items to be recommended for the user.

A. NOTATIONS AND DEFINITIONS

In collaborative filtering, there exist a set of *users*, $U = \{u_1, u_2, \ldots, u_m$, and a set of *items*, $I = \{i_1, i_2, \ldots, i_n$. A user, $u \in U$, has rated a subset of items, $I_u \subset I$. Also, an item, $i \in I$, has been rated by a subset of users, $U_i \in U$. The *active user*, $a \in U$, is the user for whom N recommendations need to be determined. In the paper, *cloud consumers* and *cloud services* are used as synonyms for users and items, respectively. The *user-item matrix* is an $m \times n$ matrix, where $r_{u,i}$ is a rating given by user u for item i. If user u rates item i, then $0 \leq r_{u,i} \leq 1$. Otherwise, $r_{u,i} = \emptyset$. In the latter case, collaborative filtering can be used to predict its value. For user $u \in U$, the subset of its ratings is $r_{u.} = \{r_{u,i} \in R | i \in I_u$, and the mean of its ratings is $\bar{r_{u.}}$.

B. SYSTEM ARCHITECTURE

The recommender system for cloud services is shown in Fig. 1. It has three components, and works as follows. In the first step, similarity coefficients between the active user and other users are computed. In the second step, a set of nearest neighbors of the active user are identified. In the third step, a user-based CF approach is adopted to predict the QoS of the active user for an unrated item, and determine a set of recommended items for the active user.

C. RANKING-BASED CF APPROACH

1) SIMILARITY COMPUTATION

In the paper, the Spearman coefficient is adopted to calculate the similarity between the active user a and user u. In Section III, a CF approach using the Pearson coefficient is compared with a CF approach using the Spearman coefficient.

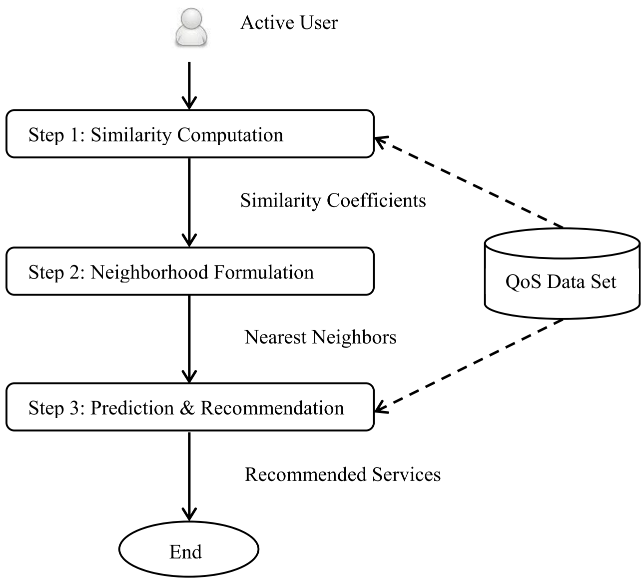

FIGURE 1. **Recommender systems for cloud services.**

-Pearson correlation coefficient (the Pearson coefficient in short). It is one of the most popular techniques to calculate the similarity between two users [7].

$$r(a,u) = \frac{\sum_{i \in I_a \cap I_u} (r_{a,i} - \bar{r_{a.}})(r_{u,i} - \bar{r_{u.}})}{\sqrt{\sum_{i \in I_a \cap I_u} (r_{a,i} - \bar{r_{a.}})^2 \sum_{i \in I_a \cap I_u} (r_{u,i} - \bar{r_{u.}})^2}} \quad (1)$$

The value of the Pearson coefficient is between -1 and 1, where -1 implies total negative correlation, and 1 implies total positive correlation.

-Spearman rank correlation coefficient (the Spearman coefficient in short). It measures how well the relationship between two variables can be described using a monotonic function [13]. Given two rankings on a set of items, the Spearman coefficient calculates their similarity by considering the difference of the two rankings for each item.

$$\rho(a,u) = 1 - \frac{6\sum d^2}{n\left(n^2 - 1\right)} \quad (2)$$

where n is the number of items, and d is the difference of the two rankings for each item. Here, each ranking has distinct values. The value of the Spearman coefficient is between -1 and 1, where -1 implies that one ranking is the opposite of the other, and 1 implies that two rankings are the same.

2) NEIGHBORHOOD FORMULATION

In the paper, the k-nearest neighbor technique is used to find a set of similar neighbors U_k for the active user, where k is a positive integer. However, it is not easy to find a good value for k.

3) PREDICTION AND RECOMMENDATION

In the paper, a user-based CF approach is adopted to predict the QoS for item i, where the contribution of a neighbor is weighted by its similarity with the active user a [7].

$$p\,(a,i) = \overline{r_{a.}} + \frac{\sum_{u \in U_k} \rho(a,u)(r_{u,i} - \bar{r}_{u.})}{\sum_{u \in U_k} \rho(a,u)} \tag{3}$$

Next, the approach computes a set of items whose predicted QoS are ranked top-N. The items will be recommended to the active user, along with their ratings and rankings.

D. ALGORITHMIC DESCRIPTION

Algorithm 1 implements a CF approach using the Spearman coefficient. It works as follows. Firstly, in lines 1 to 2, U_k and I_N, which are a set of neighbors of the active user a and a set of items to be recommended, respectively, are initialized to null.

Secondly, in lines 3 and 4, for each user $u \in U \wedge u \neq a$, the similarity $\rho(a,u)$ between the active user a and user u is calculated by (2). In line 5, tuple (u,ρ) is added to map S, where u is the key, and ρ is the value. In line 6, S is sorted by its values, and U_k, which is a set of k most similar neighbors, is identified.

Algorithm 1 Spearman Approach (U, I, M, a, k, N)

Input: *set* U with a set of m users;
set I with a set of n items;
array M with a user-item matrix;
int a which indicates the active user;
parameter k which indicates the number of neighbors;
parameter N which indicates the number of items to be recommended;
set U_k which indicates a set of k nearest neighbors;
set I_p which indicates a set of items to be predicted;
Output: *set* I_N which indicates a set of top-N items to be recommended;

1 $U_k \leftarrow \emptyset$
2 $I_N \leftarrow \emptyset$
3 **for** $u \in U \wedge u \neq a$
4   use (2) to calculate $\rho(a,u)$
5   add tuple (u,ρ) to map S
6 $U_k \leftarrow sort(S)$
7 **for** each item $i \in I_p$
8   use (3) to calculate $p(a,i)$
9   add tuple (i,p) to map P
10 $I_N \leftarrow sort(P)$
11 **return** I_N

Finally, in lines 7 and 8, for each item $i \in I_p$, the rating $p\,(a,i)$ of the active user a for item i is predicted by (3). In line 9, tuple (i,p) is added to map P, where i is the key, and p is the value. In line 10, P is sorted by its values, and I_N, which is a set of top-N items to be recommended, is determined. In line 11, I_N is returned.

The computational complexity of Algorithm 1 can be determined as follows. Assume that there are m users, n items, and k nearest neighbors to be identified, and that $k \leq m$ and $k \leq n$. Firstly, the time required to compute all similarities is $O(mn^2)$, since $(m-1)$ similarities need to be computed for the active user and other $(m-1)$ users, and at most n^2 operations are required to calculate each similarity. Secondly, the time required to compute k nearest neighbors is $O(km)$, since at most $(m-1)$ operations are required to find each neighbor among m users, and k neighbors need to be identified. Finally, the time required to compute top-N recommendation is $O(kn)$, since at most n items need to be accessed for each neighbor, and there are k neighbors. So, the computational complexity of Algorithm 1 is $O\left(mn^2\right) + O\,(km) + O\,(kn) = O\left(mn^2\right)$, where $k \leq m$ and $k \leq n$.

III. EVALUATION AND ANALYSIS

To evaluate the ranking-based CF approach using the Spearman coefficient, we conduct extensive Monte Carlo simulations. In the following, the experimental setup, evaluation metrics, and simulation results are reported.

A. EXPERIMENTAL SETUP

All simulations are conducted on a Dell OptiPlex desktop with a 2.93 GHz Intel Core i7 CPU and a 16.0 GB RAM, running Microsoft Windows 7 Enterprise Operating System. The simulations are implemented with Java under Oracle NetBeans IDE 8.0.2 with JDK 8u60. Three well-known storage clouds, namely, Amazon S3, Microsoft Windows Azure Blob Storage (Azure Blob), and Aliyun Open Storage Service (Aliyun OSS) are chosen to provide real QoS data, on which our simulation data is based.

In this paper, the QoS considered is the upload responsiveness of the three storage clouds. Here, an upload operation transfers a 1 MB, 10 MB, and 100 MB file, respectively, from a personal desktop to the three storage clouds under a 100 Mbps Internet connection. The responsiveness is a normalized value between 0 and 1. The greater the responsiveness, the better the storage cloud. Refer to Zheng (2014) for details of how the upload operation is performed, and how the responsiveness is defined. In the paper, a CF approach using the Spearman coefficient is compared with a CF approach using the Pearson coefficient.

In our software prototype, there is a user-item matrix as input. Also, there is parameter *variance* that can be used to generate a random number within a certain interval of a value, such that the impact of a specific data set on recommendation performance can be reduced, if not completely removed. The Spearman and Pearson coefficients can be chosen to compute the similarity among users. Also, there are parameters k denoting a neighborhood size, and N representing a recommendation size. The output is top-N items recommended along with their ratings and rankings.

In the simulations, the user-item matrix is a 21×21 matrix. The matrix is relatively small compared with commercial recommender systems. However, it is sufficient for

TABLE 1. The user-item matrix.

	I1	…	I7	I8	…	I14	I15	…	I21
U0	0.320		?	0.917		?	0.987		?
…									
U7	0.362			0.922			0.993		
…									
U13	0.407			0.923			0.993		
…									
U20									

the purpose of illustration, and would work for storage clouds, where the providers are a few dozen now. The 21 users are organized into three groups. User groups 1, 2, and 3 include users 0-6 (starts from 0), users 7-13, and users 14-20, and mimics those perform an upload operation on a 1 MB, 10 MB, and 100 MB file, respectively.

The 21 items are organized into three groups too. Item groups 1, 2, and 3 include items 1-7 (starts from 1), items 8-14, and items 15-21, and imitate Aliyun OSS, Amazon S3, and Azure Blob, respectively. The ratings of users 0, 7, and 13 for items 1, 8, and 15, as shown in Table 1, are obtained from our previous experiments [14], whereas the other ratings of the user-item matrix are randomly generated data based on the real observations. In other words, the ratings of the same user group for the same item group are within a certain variance of the real data.

In the simulations, user 0 is set as the active user, without loss of generality. Also, N is set as 3, for simplicity's sake. A CF approach that employs Spearman and Pearson coefficients, respectively, will be used to predict the ratings and rankings of user 0 for items 7, 14, and 21, as shown with question marks in Table 1. Here, the expected ratings for items 7, 14, and 21 are 0.320, 0.917, and 0.987, respectively. The expected ranking is item 21 > item 14 > item 7, which means that items 21, 14, and 7 are ranked top 1, 2, and 3, respectively.

It should be noted, here, that cross validation is a powerful method to estimate the prediction performance of an algorithm on a fixed data set. However, it is not applicable in the paper. The reason is that our Monte Carlo simulations use randomly generated dynamic data to minimize the impact of a specific data set on the recommendation performance.

B. EVALUATION METRICS

Two metrics are used to evaluate the ranking-based CF approach. The Root-Mean Square Error (RMSE) is a metric widely used to evaluate predicted ratings [12]. RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{u,i}(p_{u,i} - \hat{p}_{u,i})^2}{N}} \tag{4}$$

where $p(u, i)$ is a predicted value by user u for item i, $p(u, i)$ is the expected value of user u for item i, and N is the number of predicted values.

The Normalized Discounted Cumulative Gain (NDCG) is a popular metric used to evaluate predicted rankings [12]. NDCG is defined as

$$NDCG_N = \frac{DCG_N}{IDCG_N} \tag{5}$$

where DCG_N and $IDCG_N$ are the Discounted Cumulative Gain (DCG) of top-N items of a predicted ranking and the ideal ranking, respectively. DCG_N is calculated by

$$DCG_N = \sum_{i=1}^{N} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{6}$$

where rel_i is the value of the item at position i of a ranking. The value of NDCG is between 0 and 1, where a larger value means a better ranking, and 1 implies the ideal ranking.

C. MONTE CARLO SIMULATIONS

Monte Carlo simulations perform random sampling, and conduct experiments on computer. Firstly, the impact of a neighborhood size on the recommendation performance is studied, and parameter k is varied from 2 to 20 with a step of 2. The variance is 0.01, the recommendation size is 3, and the active user is user 0. The results are shown in Table 2.

TABLE 2. The impact of the neighborhood size (k = 10 to 20).

	Pearson			Spearman		
k	I7	I14	I21	I7	I14	I21
2	0.345	0.902	0.969	0.373	0.889	0.952
4	0.344	0.894	0.973	0.337	0.900	0.979
6	0.339	0.908	0.980	0.334	0.912	0.982
8	0.337	0.905	0.978	0.339	0.904	0.976
10	0.337	0.905	0.972	0.339	0.905	0.973
12	0.353	0.904	0.972	0.360	0.900	0.970
14	0.336	0.910	0.980	0.347	0.903	0.975
16	0.341	0.905	0.976	0.343	0.905	0.975
18	0.345	0.904	0.973	0.345	0.905	0.974
20	0.342	0.901	0.970	0.342	0.902	0.970

Take $k = 10$ as an example. For the CF approach using the Pearson coefficient, the predicted ratings are 0.337, 0.905, 0.972 for items 7, 14, and 21, respectively, and the ranking is item 21 > item 14 > item 7. The RMSE is 0.013 by (4), and the NDCG is 1.000 by (5) and (6). For the CF approach using the Spearman coefficient, the predicted ratings are 0.339, 0.905, and 0.973 for items 7, 14, and 21, respectively, and the ranking is item 21 > item 14 > item 7. The RMSE and NDCG are 0.015 and 1.000, respectively. In this case, in terms of RMSE (i.e., 0.013 vs. 0.015), the CF approach using the Pearson coefficient slightly performs better than the CF approach using the Spearman approach. In terms of NDCG (i.e., 1.000 vs. 1.000), the two approaches are the same.

In the ten cases, for the CF approach using the Pearson coefficient, the average RMSE and NDCG are 0.017 and 1.000, respectively. For the CF approach using the Spearman coefficient, the average RMSE and NDCG are 0.019 and 1.000, respectively. So, in terms of RMSE (i.e., 0.017 vs. 0.019), the CF approach using the Pearson coefficient performs slightly better than the CF approach using the Spearman coefficient. In terms of NDCG (i.e., 1.000 vs. 1.000), the two approaches are the same.

TABLE 3. The impact of the variance (var = 0.01 to 0.10).

		Pearson			Spearman	
var	I7	I14	I21	I7	I14	I21
0.01	0.336	0.905	0.975	0.353	0.895	0.966
0.02	0.345	0.902	0.962	0.350	0.898	0.958
0.03	0.351	0.907	0.957	0.350	0.910	0.958
0.04	0.344	0.899	0.970	0.340	0.894	0.964
0.05	0.354	0.886	0.963	0.366	0.892	0.957
0.06	0.361	0.911	0.951	0.357	0.905	0.955
0.07	0.345	0.895	0.953	0.352	0.898	0.953
0.08	0.353	0.911	0.929	0.352	0.909	0.933
0.09	0.311	0.875	0.928	0.335	0.873	0.921
0.10	0.355	0.907	0.935	0.362	0.894	0.933

In fact, as the neighborhood size increases, RMSE decreases with some fluctuations, whereas NDCG remains the same here. From our observations, a value around 10 is reasonable for the neighborhood size in the simulations.

Secondly, the impact of the variance on the recommendation performance is studied, and it is varied from 0.01 to 0.10, with a step of 0.01. The neighborhood size is 10, the recommendation size is 3, and the active user is user 0. The results are shown in Table 3.

In the ten cases, for the CF approach using the Pearson coefficient, the average RMSE and NDCG are 0.029 and 1.000, respectively. For the CF approach using the Spearman coefficient, the average RMSE and NDCG are 0.032 and 1.000, respectively. So, in terms of RMSE (i.e., 0.029 vs. 0.032), the CF approach using the Pearson coefficient slightly performs better than the CF approach using the Spearman coefficient. In terms of NDCG (i.e., 1.000 vs. 1.000), the two approaches are the same. In fact, as the variance increases, RMSE increases with some fluctuations, whereas NDCG remains the same here. From our observations, a value around 0.10 is reasonable for the variance in the simulations.

Thirdly, the CF approach using the Spearman coefficient is compared with the CF approach using the Pearson coefficient. In the simulations, the variance is 0.10, the neighborhood size is 10, the recommendation size is 3, and the active user is user 0. The simulations are run for 10 times. The results are shown in Table 4.

TABLE 4. Recommendation results (var = 0.10, k = 10).

		Pearson			Spearman	
run	I7	I14	I21	I7	I14	I21
1	0.360	0.923	0.927	0.342	0.920	0.937
2	0.347	0.905	0.930	0.362	0.909	0.925
3	0.360	0.919	0.925	0.359	0.915	0.934
4	0.353	0.929	0.942	0.359	0.920	0.929
5	0.346	0.891	0.934	0.347	0.881	0.923
6	0.329	0.884	0.939	0.312	0.894	0.928
7	0.373	0.894	0.928	0.340	0.903	0.930
8	0.363	0.854	0.898	0.364	0.847	0.896
9	0.378	**0.950**	**0.935**	0.350	0.933	0.936
10	0.349	0.892	0.946	0.362	0.892	0.942

In the simulations, for the CF approach using the Pearson coefficient, the average RMSE and NDCG are 0.042 and 0.971, respectively. For the CF approach using the Spearman coefficient, the average RMSE and NDCG are 0.042 and 1.000, respectively. So, in terms of RMSE (i.e., 0.042 vs. 0.042), the two approaches are the same. In terms of NDCG (i.e., 0.971 vs. 1.000), the CF approach using the Spearman coefficient performs better than the CF approach using the Pearson coefficient.

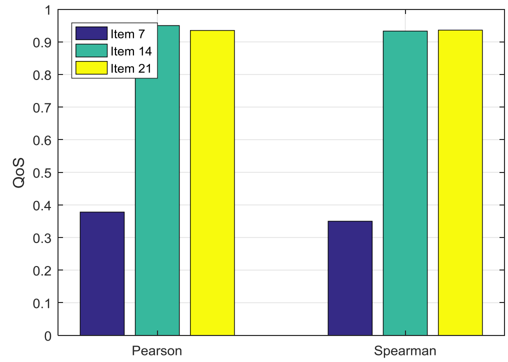

FIGURE 2. An unreliable ranking caused by the pearson coefficient.

In run 9, for the CF approach using the Pearson coefficient, the RMSE is 0.016. However, the ranking is item 14 > item 21 > item 7. So, the NDCG is 0.710. For the CF approach using the Spearman coefficient, the RMSE is 0.012. However, the ranking is item 21 > item 14 > item 7. So, the NDCG is 1.000. This indicates that the CF approach using the Pearson coefficient makes a mistake in its ranking, whereas the CF approach using the Spearman coefficient avoids the mistake. Fig. 2 illustrates the prediction that causes an unreliable ranking by the Pearson coefficient for items 14 and 21.

In conclusion, the Monte Carlo simulations show that in terms of RMSE, the CF approach using the Pearson coefficient performs slightly better, with a few exceptions, than the CF approach using the Spearman coefficient - a ranking-based approach. The Monte Carlo simulations also show that in terms of NDCG, the CF approach using the Spearman coefficient performs better than or the same as the CF approach using the Pearson coefficient. So, the CF approach using the Spearman coefficient can achieve more reliable rankings than the CF approach using the Pearson coefficient - a rating-based approach.

IV. RELATED WORK

A. CLOUD QUALITY MODELS

"QoS is an important research topic in cloud computing" [12]. Hwang, Fox, and Dongarra (2012) propose a compound metric named Quality of Cloud Services (QoCS) to measure the overall quality of cloud systems. QoCS considers five attributes, i.e., throughput and efficiency, scalability, availability, security, and cost effectiveness. The value of the five attributes is normalized to be 1, where a lower value means poor performance while a higher value means better performance.

Zheng et al. (2014) present a model named CLOUDQUAL to measure the quality of cloud services. The model has both quality dimensions and metrics, and targets general cloud services. CLOUDQUAL contains six quality dimensions, i.e., usability, availability, reliability, responsiveness, security, and elasticity. All quality dimensions are objective except usability. A formal specification is given for each quality dimension.

Gonzales et al. (2016) present a cloud reference architecture that contains various security controls and best practices, and a cloud security model called Cloud-Trust that provides quantitative assessments of IaaS clouds. Cloud-Trust, which is based on a Bayesian network model, can determine probabilities of APT (Advanced Persistent Threats) infiltration, and APT detection. However, it is hard to obtain accurate probabilities by analytical means.

B. CLOUD SERVICE RECOMMENDATION

Zheng et al. (2011) argue that as the number of web services is increasing, effective techniques are required to recommend web services to users, and that QoS is important in describing web services. They propose a CF approach called WSRec to make web service recommendation, which requires no additional invocations of web services, but exploits the past experiences of users. However, the approach cannot be adopted for cloud service recommendation without modifications, as cloud services differ from web services in several aspects.

Sun et al. (2014) argue that the explosion of cloud services on the Internet brings new challenges for cloud service selection. They conduct a comprehensive analysis of cloud service selection approaches from five perspectives, i.e., decision making methods, data representation models, parameters and characteristics of cloud services, contexts, and purposes. Also, open issues are identified and discussed for cloud service selection. Even the two topics are related, cloud service recommendation is different from cloud service selection.

Zheng et al. (2013) argue that QoS provides valuable information for users to select cloud services from a set of functionally equivalent services. They propose a QoS ranking prediction framework named CloudRank for cloud services. They report that the ranking-based approaches outperform existing rating-based approaches. However, the ranking-based approaches can predict QoS ranking, but no QoS rating is mentioned, which may impede users to make informed decisions. Also, the experiments are conducted on the data set of web services instead of cloud services.

V. CONCLUSIONS AND FUTURE WORK

As cloud computing becomes popular, the same or similar services are delivered over the Internet. QoS is an important differentiator among functionally equivalent services. In this paper, recommender systems are employed to assist cloud providers to promote their services and cloud consumers to identify services that meet their QoS requirements.

Collaborative filtering is the most successful and widely used technique to build recommender systems. In the paper, we argue for QoS-based cloud service recommendation, and propose a ranking-based CF approach using the Spearman coefficient. The approach can predict both ratings and rankings for cloud services. To demonstrate the effectiveness of the approach, we conduct extensive simulations, and compare the approach with a rating-based CF approach using the Pearson coefficient. Results show that the CF approach using the Spearman coefficient can achieve more reliable rankings, yet less accurate ratings, than the CF approach using the Pearson coefficient.

To achieve better performance, we plan to use a mixed approach in our next step. In other words, we first use the CF approach using the Spearman coefficient to predict rankings, and then use the CF approach using the Pearson coefficient to predict ratings. In this way, the mixed approach could achieve more accurate ratings, while still obtaining reliable rankings. In addition, we plan to compare the CF approach using the Spearman coefficient with other ranking-based approaches in our future work.

REFERENCES

[1] F. Cacheda, V. Carneiro, D. Fernandez, and V. Formoso, "Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems," *ACM Trans. Web*, vol. 5, no. 1, p. 2, Feb. 2011.

[2] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, p. 13, Jan. 2015.

[3] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, Jan. 2004.

[4] D. Gonzales, J. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, "Cloud-trust—A security assessment model for infrastructure as a service (IaaS) clouds," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2015.2415794.

[5] J. Heizer and B. Render, *Operations Management*, 7th ed. Upper Saddle River, NJ, USA: Pearson, 2004.

[6] K. Hwang, G. C. Fox, and J. J. Dongarra, *Distributed and Cloud Computing From Parallel Computing to the Internet of Things*, 1st ed. Waltham, MA, USA: Morgan Kaufmann, 2012.

[7] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, 1st ed. New York, NY, USA: Cambridge Univ. Press, 2010.

[8] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, p. 3, Jul. 2014.

[9] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, "Cloud service selection: State-of-the-art and future research directions," *J. Netw. Comput. Appl.*, vol. 45, pp. 134–150, Oct. 2014.

[10] L. A. Tawalbeh, R. Mehmood, E. Benkhlifa, and H. Song, "Mobile cloud computing model and big data analysis for healthcare applications," *IEEE Access*, vol. 4, pp. 6171–6180, Sep. 2016.

[11] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *ACM Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr./Jun. 2011.

[12] Z. Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, "QoS ranking prediction for cloud services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1213–1222, Jun. 2013.

[13] X. Zheng, P. Martin, K. Brohman, and L. D. Xu, "CLOUDQUAL: A quality model for cloud services," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1527–1536, May 2014.

[14] X. Zheng, *QoS Representation, Negotiation and Assurance in Cloud Services*. Kingston, ON, Canada: Queen's Univ., 2014.

XIANRONG ZHENG (M'08) received the Ph.D. degree in computer science from Queen's University, Kingston, ON, Canada, in 2014. He is currently an Assistant Professor with the Information Technolgoy and Decision Sciences Department, Old Dominion University, Norfolk, VA, USA. His research areas are computing and informatics. His current research interest is cloud service recommendation. His broad research interests include data science (cloud computing and big data) and electronic commerce (recommender systems and reputation systems).

LI DA XU (M'86–SM'11–F'16) is an Academician of the Russian Academy of Engineering. He is a Highly Cited Researcher named by Thomson Reuters (Clarivate). He is the Founding Chair of IFIP TC8 WG8.9 and of the IEEE SMC Society Technical Committee on Enterprise Information Systems, the Founding Editor-in-Chief of the *Journal of Industrial Information Integration* (Elsevier) and *Enterprise Information Systems* (Taylor & Francis), and the Founding Co-Editor-in-Chief of *Frontiers of Engineering Management* (official journal of the Chinese Academy of Engineering), the *Journal of Management Analytics* (Taylor & Francis), and the *Journal of Industrial Integration and Management* (World Scientific).

SHENG CHAI (M'04) is currently pursuing the Ph.D. degree at the School of Computing, Queen's University, Kingston, ON, Canada. His research interests lie in the areas of big data, machine learning, data mining, and information retrieval. He has served on the Technical Program Committee of the KICSS Conference.