A new method of QoS prediction based on probabilistic latent feature analysis and cloud similarity

Weina Lu*

School of Automation Science and Electrical Engineering,
Beihang University,
Beijing 100191, China
and
School of Information Technology Engineering,
Tianjin University of Technology and Education,
Tianjin, 300222, China
Email: luweina1981@126.com
*Corresponding author

Xiaohui Hu and Xiaotao Li

School of Automation Science and Electrical Engineering,
Beihang University,
Beijing 100191, China
Email: hxh@iscas.ac.cn
Email: taosmall@163.com

Yuan Wei

Ming Safety Technology Branch of China Coal Research Institute,
Beijing, 100013, China
Email: weiyuansafety@163.com

Abstract: With the increasing requirements of service mode in cloud computing, predicting accurate quality of service (QoS) is greatly significant in the recommender or composition system to avoid expensive and time-consuming invocations. Unlike previous research approaches which generally stay on the explicit values of QoS data, in this paper, we propose a new prediction method based on probabilistic latent feature analysis and cloud similarity. As user experience quality of service is influenced by the implicit factors, such as network performance, user context and user preference, we first consider these factors as the latent features of user and relate it to the QoS data using pLSA model. Then, the users or services are clustered based on the similar latent features. Finally, after mining the similarity of users in the same cluster by cloud model, the personalised QoS values are predicted by the experience quality of the similar users with the similar services. Experiment results with a real QoS dataset show that the proposed approach can effectively achieve an accurate QoS prediction.

Keywords: service; QoS prediction; experience quality; probabilistic latent feature; cloud model.


Biographical notes: Weina Lu received her PhD degree at the Automation Science and Electrical Engineering Institute, Beihang University, Beijing, China. She received her Master in Control Science and Engineering from the Northwestern Polytechnical University, Xi’an, China, in 2006. Her research interests include service computing and data mining.

Xiaohui Hu received his PhD degree in the Computer Institute, Beihang University, Beijing, China. He is a Professor and Doctoral Supervisor at Beihang University. His research interests mainly include service-oriented computing, big data and information integration.
1 Introduction

In recent years, cloud computing based on the form of various services is more and more popular in the field of computer science. Services are becoming the core technology of cross-platform interactions between machines in an open cloud computing environment. With the growing presence and adoption of services in a lot of internet domains, QoS has become an important differentiating point of services with equivalent functions. In the service application and composition, the consumers consider not only the function of service but also the quality of service (QoS). However, owing to the large number of services and crowded service candidates, the consumers hardly obtain the QoS data of all the services completely, because the real-world service invocations may be expensive and very time-consuming. As a result, the QoS data measured and experienced by client-side is usually sparse. The insufficient information on the QoS makes it inconvenient for service selection or recommendation.

In view of the above actual situation, QoS prediction is inevitably needed to obtain accurate and personalised service quality values on client-side. A most widely adopted way is to reuse other users’ experience, considering that the similar users have similar experience of QoSs. So far, there have been a lot of significant results in many research investigations. As in Shao et al. (2007, 2009), the authors earlier proposed the user-based collaborative filtering (CF) method for the missing QoS prediction. In Zheng et al. (2011) and Jiang et al. (2011), a combined user and item-based CF approach is enhanced to predict QoS values by using historical QoS experience of other similar users and similar web services. All of the above CF methods have some effectiveness in getting personalised service QoS values. Because of some limitations of Pearson correlation coefficient (PCC) in similarity computation, they have confined scalability for large datasets. In order to make progress, the low-rank matrix factorisation (MF) model is widely used in Zheng et al. (2013), Lo et al. (2012), Zhang et al. (2011), Koren (2010) and Yu et al. (2013). MF methods focus on dimensionality reduction techniques to fit the user-item matrix approximately. This improves prediction performance through better addressing sparsity and scalability problems, but it also loses useful information for reducing dimensionality. Besides these methods, there are some other researches on hybrid CF algorithm such as region-based CF prediction (Chen et al., 2013; Li et al., 2008), user cluster-based CF prediction (Shi et al., 2011), invoked feature and CF prediction (Li et al., 2013) and so on. All of these hybrid methods extend user and item-based CF, but at the same time, they also increase complexity and expense for implementation, and further more they need external information that is usually not available. In conclusion, the present researches on prediction of QoS missing values make progresses in memory-based CF (Shao et al., 2007, 2009; Zheng et al., 2011; Jiang et al., 2011; Lu et al., 2014), model-based CF (Zheng et al., 2013; Lo et al., 2012; Zhang et al., 2011; Koren, 2010; Yu et al., 2013; Wu et al., 2007) and hybrid CF recommenders (Chen et al., 2013; Li et al., 2008; Shi et al., 2011; Li et al., 2013). However, these approaches generally calculate from the mathematical point of view, and few studies analyse the latent features of service quality values in the process of prediction. As the information of QoS dataset is not fully exploited, the accuracy and efficiency are not very high.

In this work, we focus on the probabilistic latent features of user experience QoS to mine the similarity of users and services. By introducing personalised potential factors such as user context and user preferences to be the implicit characteristics of users, we firstly cluster users and services with the similar themes using probabilistic latent semantic analysis (pLSA). On this basis, a cloud model is employed to measure similarity of user characteristics, and then QoS missing values are predicted with CF with the purpose of completing the service quality information for consumers.

2 Problem description

The QoS dataset in the prediction is denoted by an $M \times N$ matrix (shown in Table 1), which contains the relationship between $M$ service items and $N$ users. Each entry $q_{ij}$ in this matrix represents a QoS value observed by the service user $j$ on the service item $i$. As shown in Table 1, the QoS sparse matrix includes some missing values which means the user does not invoke and evaluate the corresponding service.
The problem we study in this paper is how to predict the missing values of the user-service matrix effectively and precisely according to the known QoS values. The corresponding formal description is as follows.

$S = \{s_1, s_2, \ldots, s_m\}$ denotes the set of $m$ services; $U = \{u_1, u_2, \ldots, u_n\}$ denotes the set of $n$ users; Given a pair $(i, j)$, $s_i \in S$, and $u_j \in U$, then let $Q = \{q_{ij}, 1 \leq i \leq m, 1 \leq j \leq n\}$ be the set of QoS values with $n$ users invoking $m$ services. Where $q_{ij}$ represents the experienced quality of the $i$-th service with the $j$-th user, and $q_{ij} \in \mathbb{R}^{max} \cup \{\emptyset\}$. $\Gamma = \{q_{ij}, q_{ij} \in \mathbb{R}^{max}\} \subseteq Q$ denotes the set of known QoS value; $\Psi = \{q_{ij}, q_{ij} = \emptyset\} \subseteq Q$ denotes the set of unknown QoS value.

So, we intend to utilise the known $q_{ij} \in \mathbb{R}^{max}$ of set $\Gamma$ to predict the unknown $q_{ij}$ of set $\psi$, and make $q_{ij} \neq \emptyset$ at last.

In practice, the QoS evaluation of user is generally from terminal monitoring or subjective feeling (Zheng et al., 2013; Zhang et al., 2011; Yu et al., 2013). It is actually the user experience QoS. Influenced by network performance, user context, user preference and so on, the experienced quality of the same service from different users may not be same. We consider the potential factors affecting QoS experience as the latent feature of users. On the other hand, there are also similar influencing factors of QoS performance on server-side for the deployment of service. The services with similar non-functional attributes are inferred to as similar services. We think that users with similar latent features will experience alike quality of the similar services. Therefore, we can explore the similarity of user latent features and use the similar quality to approximate the real QoS, so that a complete and effective mechanism of QoS prediction is formed.

### 3 Mining cloud similarity of user latent feature

In order to compare the similarity of users’ latent features, latent variable model was employed to associate each QoS value with latent variable $z$ (i.e., theme). Latent variable implies user context, user interests and other factors of experience quality. The users are clustered together by pLSA according to the potential factors. In the same way, the similar services can be also clustered. Then, we can calculate the cloud similarity of user experience on quality of similar services.

#### 3.1 pLSA generation model of service QoS

pLSA is a proposed generation model based on probability and statistics by Hofmann in 1999. It originated from the semantics research of natural language and text learning. By introducing the latent variable $z$ as a potential theme category and relating it to the observed data in co-occurrence set with semantic correlation, pLSA model establishes the relationship between documents, words and themes. The co-occurrence data is mapped to the probability distribution of a fixed theme, so that the dimension reduction can be achieved. Both the traditional graph model and MF of probability distribution of pLSA are shown in Figure 1, where shaded circle denotes the directly observed variables, including the document $w$ and the word $d$, non-shaded circle denotes the non-directly observed variable $z$ of latent theme. The final aim of pLSA solution is to obtain the document-theme distribution (i.e., $p(z|d)$) of the target to classify and determine the category of the document.

#### Table 1

The user-service QoS matrix

<table>
<thead>
<tr>
<th></th>
<th>User1</th>
<th>User2</th>
<th>User3</th>
<th>User4</th>
<th>User5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service1</td>
<td>0.32</td>
<td>0.31</td>
<td>0.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service2</td>
<td>0.41</td>
<td>0.45</td>
<td>0.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service3</td>
<td>12.3</td>
<td>15.5</td>
<td>17.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service4</td>
<td>0.57</td>
<td>0.52</td>
<td>0.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service5</td>
<td>0.38</td>
<td>0.34</td>
<td>0.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service6</td>
<td>5.90</td>
<td>6.23</td>
<td>8.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service7</td>
<td>0.46</td>
<td>0.43</td>
<td>0.48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1 Probabilistic model diagram of pLSA, (a) pLSA: graph model (b) pLSA: MF of probability distribution

(a)

(b)

Establishing pLSA generation model of QoS is to build model for co-occurrence data of service quality related to the specific theme. The variable set $z$ of implicit semantics is associated with each observed QoS value. Here, we refer to user latent features as category theme $z$, user $u$ as document $d$, and service $s$ as word $w$. The user-service pair is $(s, u)$, $s \in S$, $u \in U$, thus the observed quality value $Q = q(s, u)$ can be regarded as co-occurrence table $N$ of users and services. $p(u)$ represents the probability of observed user $u$, $p(s|z)$ represents the invocation probability of service $s$ in the condition of latent variable $z$. $p(z|u)$ represents the probability distribution (posterior probability) of user $u$ in latent variable space, the joint probability distribution of service $s$, user $u$ and latent theme $z$ is:
\( p(s, u, z) = p(u)p(z|u)p(s|z) \). By finding out \( p(s|z) \) of service and \( p(z|u) \) of the corresponding user, pLSA obtain the probability distribution \( p(s|u) \) of service invocation of a user in implicit factors:

\[
 p(s \mid u) = \sum_z p(s \mid z)p(z \mid u) \tag{1}
\]

The joint distribution of user \( u \) and service \( s \) is:

\[
 p(s, u) = p(u)p(s \mid u) \tag{2}
\]

Put (1) into (2), the joint probability distribution of \( (s, u) \) with latent variable is obtained as follows:

\[
 p(s, u) = p(u)\sum_z p(s \mid z)p(z \mid u) \tag{3}
\]

Therefore, the probability model for QoS dataset is generated as follows:

\[
 P_Q = \prod_s \prod_u p(s, u)^{q(s,u)} = \prod_s \prod_u \left[ p(u)\sum_z p(s \mid z)p(z \mid u) \right]^{q(s,u)} \tag{4}
\]

### 3.2 User and service clustering based on EM

The pLSA generation model is a function of \( p(z|u) \) and \( p(s|z) \) with the assumption that user \( u \) and service \( s \) are independent in the condition of latent feature \( z \). For such parameter estimation of probabilistic model containing latent variable or missing data, the EM algorithm is the appropriate solution. With the maximum likelihood estimation method to solve learning problem of the model, the user (or service) clustering is performed under the latent theme.

The optimisation problem of log-likelihood function for QoS data is as follows:

\[
 \text{max } \mathcal{L} = \sum_u \sum_s q(s,u) \log p(s,u) \\
\approx \sum_u \sum_s q(s,u) \log \left[ p(u)\sum_z p(s \mid z)p(z \mid u) \right] \tag{5}
\]

For the maximum likelihood parameter estimation of pLSA, EM algorithm is divided into two steps: E-step and M-step. For E-step, using Bayes formula to calculate the posterior probability of latent variable in the condition of current parameter values, there is:

\[
 p(z \mid s, u) = \frac{p(s \mid z, u)p(z \mid u)}{p(s, u)} = \frac{p(s \mid z)p(z \mid u)}{\sum_z p(s \mid z)p(z \mid u)} \tag{6}
\]

For M-step, maximising the expected value of log-likelihood function of the complete data with latent variable, as follows:

\[
 \text{max } E[\mathcal{L}] = \sum_u \sum_s q(s,u) \sum_z p(z \mid s, u) \left[ \log p(u) + \log p(s \mid z) + \log p(z \mid u) \right] \tag{7}
\]

For the maximum expectation \( E[\mathcal{L}] \), it is an extreme-value problem of a multivariate function of which the constraint conditions are \( \sum_u p(s \mid z) = 1 \) and \( \sum_z p(z \mid u) = 1 \). The conditional extreme-value problem can be converted to unconditional extreme-value problem. Its form of Lagrange function is:

\[
 \mathcal{H} = E[\mathcal{L}] + \alpha \left[ \sum_u 1 - \sum_u p(u) \right] + \beta \sum_z \left[ 1 - \sum_z p(s \mid z) \right] + \gamma \sum_u \left[ 1 - \sum_z p(z \mid u) \right] \tag{8}
\]

\( \mathcal{H} \) is a function of \( p(u) \), \( p(s|z) \) and \( p(z|u) \). For the three respective partial derivatives, we obtain:

\[
 \frac{\partial \mathcal{H}}{\partial p(u)} = \sum_s \sum_z q(s,u)p(z \mid s, u) - \alpha = 0 \\
\frac{\partial \mathcal{H}}{\partial p(z \mid s)} = \sum_u \sum_z q(s,u)p(z \mid s, u)p(s \mid z) - \beta = 0 \\
\frac{\partial \mathcal{H}}{\partial p(z \mid u)} = \sum_u q(s,u)p(z \mid s, u)p(z \mid u) - \gamma = 0 \tag{9}
\]

After the above equations are solved, the parameters are estimated by maximising the expected value in M-step, as follows:

\[
 p(u) = \frac{\sum_s \sum_z q(s,u)p(z \mid s, u)}{\sum_u \sum_z \sum_z q(s,u)p(z \mid s, u)} \\
 p(s \mid z) = \frac{\sum_u q(s,u)p(z \mid s, u)}{\sum_u \sum_z q(s,u)p(z \mid s, u)} \\
 p(z \mid u) = \frac{\sum_u q(s,u)p(z \mid s, u)}{\sum_u \sum_z q(s,u)p(z \mid s, u)} \tag{10}
\]

EM algorithm is one of the most important parameter estimation methods for sparse data. By repeatedly iterative training of E-step and M-step, a stable relationship is built between latent semantic theme \( z \), user \( u \) and service \( s \). The category with the maximum posterior probability (i.e., max \( p(z|u) \)) is identified as the final result of user clustering. In the same way, we cluster services by pLSA, as long as exchange the mappings of \( u \) and \( s \). That is to say, regard service \( s \) as document \( d \), user \( u \) as word \( w \). By reusing the EM algorithm for model training and learning, the obtained theme \( z \) of maximum posterior probability is the implicit category the service belongs to.

### 3.3 User similarity computation based on cloud model

User and service clusters in a series of latent theme are produced by pLSA clustering and partition. Obviously, the users or services in the same cluster are alike to some extent. Thus, the range of searching similar users and services can be reduced to a cluster. To accurately predict
the unknown QoS values, we need to further calculate the similarity of users in the same cluster.

The traditional measurement of similarity usually employs Person correlation coefficient (Shao et al., 2007, 2009; Zheng et al., 2011; Jiang et al., 2011; Lu et al., 2014), which apply the deviation of user data and its mean values to measure the degree of the user correlation between each other. This approach can only measure linear similarity of subjective QoS attributes with ignoring the nonlinear correlation between objective attributes. In addition, similar users may not be identified as such if they have not both rated any of the same services. Actually, as a large number of candidate services are distributed in cloud environment, the users in the same cluster rarely invoke the same services. So we can only analyse the similarity of the users by their experience quality of similar services. In this work, basing on cloud model, the cloud similarity of users in the same cluster is measured with the assistance of backward cloud algorithm.

### 3.3.1 Cloud model

Cloud model (Li and Du, 2005) is an uncertain transforming model between a qualitative concept and its quantitative expression. It mainly represents two uncertainties in objective things and concepts: fuzziness and randomness. Cloud model was proposed by Academician Li Deyi in 1995. So far, it has been successfully applied to data mining, fuzzy evaluation, evolutionary computation and many other fields.

**Definition 1:** Cloud and cloud drop (Li and Du, 2005): Let $U$ be a universal set described by precise numbers, and $C$ is the qualitative concept related to $U$. If there is a number $x \in U$, which randomly realises the concept $C$, and the certainty degree of $x$ for $C$, i.e., $\mu(x) \in [0, 1]$, is a random value with stabilisation tendency

$$
\mu: U \to [0, 1] \quad \forall x \in U \quad x \to \mu(x)
$$

Thus, the distribution of $x$ on $U$ is defined as cloud, and every $x$ is defined as a cloud drop.

For the qualitative concept to be expressed by cloud model, its global property can be reflected by the digital features of cloud. The three digital features, i.e., $E_x$ (expected value), $E_n$ (entropy) and $H_n$ (hyper entropy) are employed by cloud to express the overall feature of a concept. They are denoted as $C(E_x, E_n, H_n)$ and referred to as the feature vectors of cloud. In this paper, we use them to represent uncertain features of user experience quality. Through the backward cloud algorithm, a set of quantitative experienced QoS data can be transformed into the qualitative concept of user latent feature expressed by the digital features $(E_x, E_n, H_n)$.

### Algorithm 1 Backward cloud algorithm

| Input: $m$ QoS values of similar services experienced by a user, namely $m$ cloud drops $\{x_1, x_2, x_3, \ldots, x_m\}$ |
| Output: user experience feature $(E_x, E_n, H_n)$ |
| Steps: |
| a Calculate the mean of $x_i$, i.e., $\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i$, the first order sample centre distance of $x_i$, i.e., $\frac{1}{m} \sum_{i=1}^{m} |x_i - \bar{x}|$ and the sample variance of $x_i$, i.e., $S^2 = -\frac{1}{m-1} \sum_{i=1}^{m} (x_i - \bar{x})^2$ |
| b The estimated value of $E_x$ is: $\hat{E}_x = \bar{x}$ |
| c The estimated value of $E_n$ is: $\hat{E}_n = \sqrt{\frac{\pi}{2m}} \sum_{i=1}^{m} |x_i - \bar{x}|$ |
| d The estimated value of $H_n$ is: $\hat{H}_n = \sqrt{S^2 - \frac{\pi}{2}}$ |

**Source:** Li and Du (2005)

The cloud models of users in the same cluster are computed for the similar services through backward cloud algorithm. The qualitative representation of user cloud model is a tuple of three parameters, i.e., $CM = (E_x, E_n, H_n)$. It is referred to as user experience feature (or user feature), which is the explicit representation of user latent feature, where $E_x$ is the expectation of user feature cloud drops distributed in the domain space, the entropy $E_n$ represents the uncertainty of user experience feature and the hyper entropy $H_n$ represents uncertainty of entropy. Thus, cloud model of users can reflect not only the overall level but also the uncertainty of user feature. The backward cloud algorithm is an important way to map the quantitative QoS value to the qualitative user feature.

### 3.3.2 Cloud similarity computation

In this section, we are only concerned with the QoS of services similar to the forecast service. The cloud similarity operator is designed for the contrast of two similar user features.

**Definition 2:** Cloud similarity operator:

Cloud similarity operator

$$
OP_{sim}(CM_i, (E_x, E_n, H_n), CM_j, (E_x, E_n, H_n))
$$

is a mapping $f(CM_i, CM_j) \to sim(i, j)$, which transform two cloud models into the similarity between them by satisfying the following conditions:

1. $\text{sim}(i, j) \in [0, 1]$, when $CM_i, CM_j, \text{sim}(i, j) = 1$

2. $\text{sim}(i, j) = \left\{ \begin{array}{ll}
1 & [E_n - E_n] \frac{E_n - E_n}{E_n + E_n} \left(1 - \frac{E_n - E_n}{E_n + E_n}\right) - \frac{H_n - H_n}{H_n + H_n} \\
 \frac{1}{E_n + E_n} & [E_n - E_n] \frac{E_n - E_n}{E_n + E_n} \left(1 - \frac{E_n - E_n}{E_n + E_n}\right) - \frac{H_n - H_n}{H_n + H_n} \\
0 & \frac{1}{E_n + E_n} [E_n - E_n] \frac{E_n - E_n}{E_n + E_n} \left(1 - \frac{E_n - E_n}{E_n + E_n}\right) - \frac{H_n - H_n}{H_n + H_n} \\
\end{array} \right.$
where \( \text{sim}(i, j) = 1 \) means that the similarity between a cloud and itself is 1. \( \text{sim}(i, j) = \text{sim}(j, i) \). It means that the cloud similarity operator has symmetry.

Taking the QoS values in Table 1, for instance, the similarity of two users is calculated in the following way. The rows in the table represent the invoked services \( \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\} \), where, \( s_1, s_2, s_4, s_6, s_7 \) belong to the same service cluster, namely similar services. The columns represent the similar clustered users \( \{u_1, u_2, u_3, u_4, u_5\} \). They make up a complete user cluster. The cloud models of users \( u_1, u_2, u_3, u_4, u_5 \) were calculated respectively as follows:

\[
CM_i = (0.3925, 0.0533, 0.242),
\]
\[
CM_2 = (0.51, 0.0752, 0.0393),
\]
\[
CM_3 = (0.042, 0.0919, 0.0515),
\]
\[
CM_4 = (0.43, 0.1086, 0.0388),
\]
\[
CM_5 = (0.3967, 0.0696, 0.0196).
\]

Then, the user similarity \( \{\text{sim}(i, j), i \in [1, 2, ..., 5], j \in [1, 2, ..., 5], i \neq j\} \) can be finally calculated by the cloud similarity operator.

Since the cloud similarity contains the statistical characteristics of user experience feature for similar services and the analysing range of data has been further reduced, the designed operator is more reasonable and accurate for the similarity computation than traditional PCC.

4 Prediction of QoS missing values

In the traditional CF, the mean QoS value of all services invoked by the similar user is considered in the predicting computation. The problem of this method is indistinguishably relying on all other services to forecast the current service QoS. When the functions or properties of services are quite different to each other, the produced errors will lead to inaccurate results of prediction.

Based on the principle that only the QoS values of similar services are effective for the invoking users, this section choose the users in the same cluster to be neighbours and predict the current QoS with their CF, as follows:

\[
q_{xy} = \bar{q}_y + \sum_{u \in \text{cluster}(a)} \alpha_u (q_{ux} - \bar{q}_u)
\]  

(11)

where \( q_{xy} \) is the QoS value to predict. It represents the experienced QoS \( x \) with user \( y \), \( \bar{q}_y \) is the average quality of similar services of service \( x \) with user \( y \). \( S(a) \) is a user cluster which user \( y \) belongs to. \( q_{ux} \) is the experienced quality of the service \( x \) with similar user \( u \). \( \bar{q}_u \) is the average quality of similar services of service \( x \) with similar user \( u \). \( \alpha_u \) is a deviation weight coefficient of user \( u \) and it is obtained by the following linear calculation:

\[
a_u = \frac{\text{sim}(y, u)}{\sum_{v \in \text{cluster}(a)} \text{sim}(y, v)}
\]  

(12)

where \( \text{sim}(y, u) \) is the cloud similarity between user \( y \) and user \( u \). In this work, \( \overline{q}_y \) and \( \overline{q}_u \) are identified as the mean QoS values of services similar to the forecasting service, thus the predicted result is closer to the reality. For the very few QoS missing values with neither similar services nor users, let the predicted value \( q_{xy} = \text{null} \), so as to avoid the bad results reducing accuracy of the whole predicting process.

5 Experiment

In order to analyse the performance of our approach, we conducted related experiments by stages: the comparative experiments about the impact of category number on prediction accuracy, the comparative experiments about the impact of matrix density on prediction accuracy, and the comparative experiments about the runtime of prediction.

5.1 Datasets and experiment condition

In the real world, to invoke thousands of commercial cloud computing components for experiments is very expensive. As a kind of cloud computing component, web services can be integrated into cloud applications and they can access information or compute services from the remote systems. Therefore, we use the real web services QoS datasets (Zhang et al., 2011; Zheng et al., 2010) to conduct the experimental analysis of the algorithms. This dataset was collected from 339 users with 5,825 web services distributed throughout the world in different time intervals. It contains two attributes: the response-time and throughput. For each attribute, there are 1,974,675 service invocation results which form two 5,825 \times 339 matrixes of service-user QoS data. Without loss of generality, the proposed method can be easily extended to other more QoS attributes. The statistical characteristics of the web service QoS dataset are summarised in Table 2. All experiments are performed on the same hardware and software environment (Intel Core i3-2120 3.3 GHz, 2.0 GB RAM, Windows 7, MATLAB7.8, Java 1.7).

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Statistical characteristics of QoS dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Numerical range</td>
</tr>
<tr>
<td>Response-time</td>
<td>0–20 s</td>
</tr>
<tr>
<td>Throughput</td>
<td>0–1,000 kbps</td>
</tr>
</tbody>
</table>

|          | Number of users | Number of services | Data size |
| Response-time | 339 | 5,825 | 1,974,675 |
| Throughput | 339 | 5,825 | 1,974,675 |
5.2 Experimental results and analysis

To evaluate the accuracy of algorithms, we introduce two evaluation metrics of prediction performance: mean absolute error (MAE) and root mean square error (RMSE). The MAE is the average of absolute values of the differences between a prediction result and the corresponding observation. It is defined as:

$$\text{MAE} = \frac{1}{N} \sum_{x,y} |q_{xy} - \hat{q}_{xy}|$$

where $q_{xy}$ denotes the experience QoS $x$ with user $y$, $\hat{q}_{xy}$ is the predicted value. $N$ is the number of predicted values.

RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{x,y} (q_{xy} - \hat{q}_{xy})^2}$$

5.2.1 Impact of parameters for prediction accuracy

5.2.1.1 Impact of category number

In the task of mining QoS similarity based on pLSA, the dimension of implicit themes spaces directly influence the clustering of users and services, hence, it is closely associated with the accuracy of QoS prediction. In this section, taking the real throughput dataset for example, we observe the changes of prediction performance with different user and service categories to determine the best number of categories. Meanwhile, we compare the accuracy of two calculation methods of cloud similarity (cloudsim in Figure 2) and general cosine similarity (cousim in Figure 2). The matrix density is set to 60%. The condition of convergence for EM algorithm is $\varepsilon \geq 0.0001$. The iterations are 50–200. In Figure 2, the experiment results indicate that, when the sample size of users or services is determined, the too large or small number of categories will both badly influence the clustering and prediction. As shown in Figures 2(a) and 2(b), the MAE and RMSE of our approach (cloudsim) are both achieved the minimum 9.934 and 25.655 with eight user categories, while in Figure 2(c) and Figure 2(d), the MAE and RMSE are achieved respectively 9.5 and 26.2 near the lower level with 25 service categories. Besides that, comparing the MAE and RMSE in the four figures, it also can be seen that whatever the number of category is, the accuracy of cloud similarity is obviously better than the cosine similarity.

5.2.1.2 Impact of matrix density

To show the prediction accuracy of the proposed method, we study the impact of the matrix density on MAE and RMSE with comparing to other two popular methods: UIPCC (Zheng et al., 2011) and NIMF (Zheng et al., 2013). In the experiment, the matrix density varies from 0.05 to 0.2 with a step value of 0.05 and from 0.3 to 0.9 with a step value of 0.1. The numbers of user and service categories are respectively set to 8 and 25.
A new method of QoS prediction based on probabilistic latent feature analysis and cloud similarity

As shown in Figures 3(a), 3(b), 3(c) and 3(d), all the prediction accuracies of the three methods are influenced by the matrix density. In the four figures, with the increasing density of the matrix, the prediction errors of proposed approach (PLFCS) decrease obviously faster than UIPCC and NIMF. It means that the optimisation of prediction accuracy of PLFCS with latent feature clustering is better than others. Furthermore, from the beginning of matrix density 0.2, the prediction errors of PLFCS have reached the lowest level of the three algorithms. In Figures 3(a) and 3(c), it is needed to explain that in the narrow interval of matrix density 5% to 15%, the average prediction error of NIMF is slightly less than PLFCS 4.6% [in Figure 3(a)] and 13.4% [in Figure 3(c)], but in the wide interval of matrix density 20% to 90%, the average prediction error of NIMF is more than PLFCS 21.1% and 28.3%, which are much larger than the preceding two percentages. This shows that NIMF is only suitable for extremely sparse QoS data, while LPFCS is more applicable to most of the matrix density and it has greater versatility. The prediction results of Figure 3 are statistically summarised in Table 3 for the three algorithms, it can be seen that the average prediction errors of PLFCS is always the lowest of the three.

Table 3 The statistical characteristics of three prediction methods

<table>
<thead>
<tr>
<th>QoS dataset</th>
<th>Response-time</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean MAE</td>
<td>Mean RSME</td>
<td>Mean MAE</td>
</tr>
<tr>
<td>UIPCC</td>
<td>0.4648</td>
<td>16.1974</td>
</tr>
<tr>
<td>NIMF</td>
<td>0.4195</td>
<td>13.7462</td>
</tr>
<tr>
<td>PLFCS</td>
<td>0.3761</td>
<td>12.4394</td>
</tr>
</tbody>
</table>

5.2.2 Prediction runtime comparison

For a large number of forecasting data, our approach is on the premise that the system has completed the offline clustering for users and services, and the active user to predict belongs to one of these clusters. This part of experiment compares the average prediction runtime of PLFCS and UIPCC, so as to analyse the efficiency of two methods with different matrix density. As shown in Figure 4, with the increase of density matrix, there are both growth trends of needed runtime in the two methods. This is because the increasing effective information of QoS data causes the increasing of computation amount. In the results, the prediction time of PLFCS is 0.021 s~0.045 s, while UIPCC is 0.124 s~0.196 s, that is about 10 times more than PLFCS. The reason for this situation is that PLFCS similarity calculation with offline clustering is limited to the quality of similar services with users in the same cluster, while the UIPCC includes the quality of all together invoked services with users in the whole matrix. The different data amount involved in computation leads to the different efficiency between the two methods.
6 Conclusions

This paper proposes a new QoS prediction method based on probabilistic latent feature and cloud similarity. Further accuracy improvements are achieved by adequately exploring the implicit information of QoS data, such as network performance, context information and user preference. Firstly, the users and services with similar latent feature are clustered together by pLSA. Secondly, cloud model is employed to measure the similarity of users in the same cluster. Finally, the QoS missing value is predicted through CF of similar users and services. This work combines cluster algorithm and cloud uncertainty into CF method, so that the experimental achievement of accuracy and efficiency is encouraging in QoS prediction. In the future work, we will study how to predict QoS values by classifier for new users or new services which are outside of the known dataset.

References


