A Feedback-corrected Collaborative Filtering for Personalized Real-world Service Recommendation

S. Zhao, Y. Zhang, B. Cheng, J.-L. Chen

Shuai Zhao*, Yang Zhang, Bo Cheng, Jun-liang Chen
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, 100876, China
*Corresponding author: zhaoshuaiby@bupt.edu.cn

Abstract: The emergence of Internet of Things (IoT) integrates the cyberspace with the physical space. With the rapid development of IoT, large amounts of IoT services are provided by various IoT middleware solutions. So, discovery and selecting the adequate services becomes a time-consuming and challenging task. This paper proposes a novel similarity-measurement for computing the similarity between services and introduces a new personalized recommendation approach for real-world service based on collaborative filtering. In order to evaluate the performance of proposed recommendation approach, large-scale of experiments are conducted, which involves the QoS-records of 339 users and 5825 real web-services. The experiments results indicate that the proposed approach outperforms other compared approaches in terms of accuracy and stability.

Keywords: Internet of Things, service recommendation, similarity measurement, collaborative filtering.

1 Introduction

1.1 Background

Internet of Things (IoT) seamlessly integrates user requirement, cyberspace and physical space, which enables the dynamic cooperation of Ą°human-machine-thingĄą. As the adopting of SOA (service oriented architecture) paradigm in IoT environment [1], real-world devices will be able to offer their functionality via service interfaces, which enable other components to interact with them dynamically. The functionalities provided by these devices (e.g. the provisioning of online sensing data) are referred to as *real-world services* because they are provided by embedded systems that are directly related to the physical world [2]. With the developments of IoT, lots of middleware solutions like OpenIoT [3], GSN [4], COSM [5] are proposed which act as service provision platforms. These platforms access real-world resources around the world and provide their capability in form of millions of *real-world services*, which enable sharing, monitoring and controlling environmental data on the web. However, these existing platforms only provide limited functions for service selecting and recommendation. As the rapid increase of available services, selecting the appropriate services becomes challenging and time-consuming [6]. Therefore, service discovery becomes the critical issues of IoT development. Other than web-service discovery based on functional property which has been deeply studied [7], the studies of discovery based on non-functional property are far from mature, i.e. it is difficult to differentiate services with similar or identical functions. As the user-received performance of service is tightly related to the personalized information of specific users, identifying the optimal one for service users is difficult and costly in the case of many services with equivalent functions. Effective personalized service selection and recommendation based on non-functional property become more and more important [8]. Quality of Service (QoS) (e.g. observation-accuracy, round-trip time (RTT), etc.)

is served as key non-functional property, which acts as important element considered when discovering and selecting services [9, 10]. The values of QoS are usually influenced by the specific environment (such as network quality) of users and tend to vary with each user. Because conducting actual service invocation is time and resource consuming, it is unpractical to evaluate the QoS of all candidate services for every user [11]. So, the idea of making personalized QoS prediction for users using a small amount of available QoS value is extremely useful. Based on the predicted QoS values, personalized service recommendation is available for service-users. It enables users to select the service with optimal QoS from a number of services which are function-equivalent.

1.2 Motivation

Collaborative filtering (CF) are widely used in recommender systems [12]. The algorithms of CF can be divided into two categories: memory based and model based. Cosine-based approach (COS) [14, 15] and Pearson Correlation Coefficient (PCC) [12, 13] are two of the most popular memory-based approaches [16] to calculate the similarity between items. A number of works that employ COS-CF (Cosine based Collaborative-filtering) or PCC-CF (Pearson Correlation Coefficient based Collaborative filtering) for QoS based service recommendation and selection have been proposed recently [11] [17–19]. However, the performance that using PCC and COS to measure similarity leaves much to be desired and the prediction accuracy of these works cannot satisfy the requirement of practical application. Moreover, the experiments of these existing works are not convincing enough. The existing service recommendation [17, 19] approaches are short of sufficient-scale and systematic evaluation to verify their recommendation results. Some of them employ item dataset (such as MovieLens [20]) instead of real service dataset to evaluate their approaches.

In order to address these issues, this paper proposes a novel similarity-measurement for computing the similarity between service users and introduces a new personalized recommendation approach for real-world service based on collaborative filtering, which named feedback-corrected Tan-NED (Tanimoto Normalized Euclidean Distance). The contributions of this paper are summarized as the following aspects:

- This paper proposes a novel similarity measurement for memory based CF, which avoids the shortcomings of existing approaches for service recommendation and takes the characteristics of real-world service QoS into account. Therefore, it finds similar users more accurately and obtains more accurate QoS-prediction.

- A feedback-corrected CF approach is proposed, which significantly improves the performance of service recommendation in comparison with existing approaches.

- To evaluate the performance of proposed approach, we conduct comprehensive evaluative experiments based on a large-size real-service QoS dataset which includes 5825 real web-services and 339 users.

The following parts of this paper are organized as follows. In section 2, we review existing similarity measurement approaches for memory-based CF. Section 3 presents the proposed feedback-corrected Tan-NED CF approach. The experiment results of proposed approach are discussed in section 4. Section 5 concludes the work and discusses the future work.

2 Existing Similarity Measurement

The service similarity measurements are divided into two types, namely, functional-measurement and non-functional-measurement. Our previous work [21] focuses on measuring the functional similarity between real-world services based on semantic model. Other than functional similarity, this paper focuses on non-functional similarity, i.e. QoS similarity. The idea of memory-based CF is inspired by the fact that users trust the recommendations from the one who have similar context and preference. These methods predict the QoS of a particular service for a user based on the QoS obtained from users who have similar context and preference. In memory-based CF, Cosine-based approach (COS) [14,15] and Pearson Correlation Coefficient (PCC) [12,13] are two of the most popular algorithms to measure the similarity.

Assuming that a service recommender system includes N services and M users, then we obtain an $M \times N$ user-service matrix, in which the entry $r_{m,n}$ denotes the QoS value of service n observed by user m. If the entry $r_{m,n} = \emptyset$, it indicates that user m has never invoked service n. PCC-CF can be used to calculate the similarity between user u and user v by following formula:

$$Sim(u,v) = \frac{\sum_{i \in I}(r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I}(r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{i \in I}(r_{v,i} - \overline{r_v})^2}} \tag{1}$$

Where $I = I_u \cap I_v$ is the set of services that are co-invoked by users u and v, $r_{u,i}$ denotes the quantized QoS value of service i observed from the view of user u, and $\overline{r_u}$ is the average value on the QoS of services in I observed by user u. The values of PCC range from -1 to 1 according to the definition of equation (1).

In COS CF, the similarity between users can be measured by calculating the cosine similarity of the vectors between them:

$$sim(u,v) = \frac{\sum_{i \in I} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I} r_{u,i}^2}\sqrt{\sum_{i \in I} r_{v,i}^2}} \tag{2}$$

Table 1: An example of user-service QoS matrix

	$service_1$	$service_2$	$service_3$	$service_4$	$service_5$
$user_1$	2	4	2	4	5
$user_2$	1	2	1	2	5
$user_3$	2	4	2	$\emptyset$	$\emptyset$
$user_4$	2	2	2	1	4
$user_5$	5	5	5	4	$\emptyset$
$user_6$	3	2	3	3	1
$user_7$	3	2	3	4	$\emptyset$
$user_8$	3	1	$\emptyset$	$\emptyset$	$\emptyset$

Table 1 is an example of QoS matrix which contains 5 services ($service_1$ to $service_5$) and 8 users ($user_1$ to $user_8$). The values from 1 to 5 are the minimum to the maximum QoS-values of the user-service matrix. $\emptyset$ denotes the user has never invoked the corresponding service before.

We calculate similarity value adopting COS approach (Eq. (2)), and get the following arithmetic expression:

$$Sim(user_3, user_1) = Sim(user_3, user_2)$$

It indicates that $user_1$ is similar to $user_3$ as much as $user_2$ is. Actually, $user_1$ is more similar to $user_3$ than $user_2$, which can be easily observed according to the values in Table 1. Hence, calculation results and facts are contradictory. We can also compute the similarity between $user_4$ and $user_5$:

$$Sim(user_4, user_5) = 0.9885$$

We can draw conclusion that $user_4$ and $user_5$ are very similar according to this computation result. However, this is in conflict with the fact shown in Table 1, since $user_4$ and $user_5$ almost have the opposite QoS-values. $user_5$'s QoS-values are approximated to the maximum QoS-value 5 in the user-service matrix while $user_4$'s values are approximated to the minimum QoS-value 1 of the matrix. This contradiction arises from that when measuring the similarity between two vectors, COS only considers the angle between two vectors and does not consider the length of vector.

LetĄŻs consider another example. If we employ PCC defined in Eq.(1) to measure the similarity between users, we can get the following result:

$$Sim(user_7, user_6) < Sim(user_7, user_8)$$

It indicates that $user_6$ is less similar to $user_7$ than $user_8$ is. Actually, $user_6$ is more similar to $user_7$ than $user_8$, because there are the same value in three dimensions and a difference of 1 in one dimension for $user_7$ and $user_6$, and there are the same value in one dimension and a difference of 1 in one dimension for $user_7$ and $user_8$. Therefore, the calculation results are in conflict with the facts. This contradiction arises from that PCC does not consider the number of services co-invoked which implies the similarity of selection preference and style between users.

In addition to above mentioned approaches, some other similarity measurements are also proposed such as rated-item pools (RIPs) user similarity [22], proximity impact popularity (PIP) measure [13], and mean squared difference [14], which are either for special purposes, or for special situations, or not used widely. Among the traditional similarity measurement approaches, the approaches that we elaborated above are strongly representative.

3   Feedback-corrected Tan-NED CF

3.1   Tan-NED Similarity Measurement

In order to address the issues of existing similarity measurement, this paper proposes a novel similarity measurement named Tanimoto [23] Normalized Euclidean Distance (Tan-NED). Compared with the traditional similarity measurement approaches, our approach measures the similarity based on normalized Euclidean distances of difference multidimensional vector spaces.

Although Euclidean-distance approach can also be employed to measure similarity, Tan-NED is completely different from it. Since the number co-invoked services is different deal with different couples of users in a recommender system, the Euclidean distances of different couples of users tend to compute in the different dimensions of vector spaces. Moreover, the maximal values of Euclidean distances in different vector spaces are usually very different. A value which denotes the maximal value in one vector space may be a very small Euclidean distance in another vector space. Therefore, putting them together to measure similarity is meaningless. For instance, if users a, b both invoked the same 10 services while users c, d both invoked the same 300 services, $dist(a, b)$ is the Euclidean distance between the two users a and b, $dist(c, d)$ is the Euclidean distance between the two users c and d, then it will be meaningless to mention $dist(a, b)$ and $dist(c, d)$ in the same breath, because $dist(a, b)$ is calculated in 10-dimension vector space, and

$dist(c, d)$ is computed in 300-dimension vector space. Therefore, to measure the similarity of vectors we should consider the difference of dimension-number.

Our NED normalizes the QoS values of different users to the same range in order to address the different maximal-value issue. Then, it unifies the similarity metrics of different vector spaces. Before measuring the similarity between users, it uses the maximal and minimal QoS-value of each row to normalize every value of the same row in the original matrix M. After that, the QoS-values of each row are normalized to $[0,1]$. In consequence, the original QoS value matrix M is mapped into a row-normal matrix M_{nu}. Assuming that the number of co-invoked services by user u, v is num, and user u, v have the observed QoS-value vectors $\vec{u}, \vec{v}$ respectively in matrix M_{nu}. Then, the Euclidean distance between vector $\vec{u}$ and $\vec{v}$ in the num dimensions can be calculated by $dist(u, v)$. Besides, the maximal Euclidean distance of the num dimensions are calculated by $dist_{max}$. Since the matrix M_{nu} has been normalized, each dimension ranges from 0 to 1, and the maximal distance of each dimension is 1. In M_{nu}, $nr_{(u,i)}$, $nr_{(v,i)}$ are the QoS-value of service i towards user u, v respectively. The similarity between user u and v can be calculated by NED as follows:

$$nr_{u,i} = \frac{r_{u,i} - r_{umin}}{r_{umax} - r_{umin}}, nr_{v,i} = \frac{r_{v,i} - r_{vmin}}{r_{vmax} - r_{vmin}}$$

$$dist(\vec{u}, \vec{v}) = \sqrt{\sum_{i \in I}(nr_{u,i} - nr_{v,i})^2}$$

$$Sim_{ned}(u, v) = 1 - \frac{dist(\vec{u}, \vec{v})}{dist_{max}} = 1 - \frac{\sqrt{\sum_{i \in I}(\frac{r_{u,i} - r_{umin}}{r_{umax} - r_{umin}} - \frac{r_{v,i} - r_{vmin}}{r_{vmax} - r_{vmin}})^2}}{\sqrt{\sum_{k=1}^{|I|}(1-0)^2}}$$

i.e.,

$$Sim_{ned}(u, v) = 1 - \frac{\sqrt{\sum_{i \in I}(\frac{r_{u,i} - r_{umin}}{r_{umax} - r_{umin}} - \frac{r_{v,i} - r_{vmin}}{r_{vmax} - r_{vmin}})^2}}{\sqrt{|I|}} \tag{3}$$

Where $I = I_u \cap I_v$ is the set of services which is co-invoked by user u and v; $|I|$ is the number of I; r_{umin} and r_{umax} are the minimal and the maximal QoS-values of user u in the original matrix M, $r_{(u,i)}$ denotes the QoS-value of service i towards user u in M. The results of Eq.(3) range from 0 to 1, i.e., $Sim_{ned}(u, v) \in [0, 1]$, where $Sim_{ned}(u, v) = 0$ represents that two users are dissimilar and $Sim_{ned}(u, v) = 1$ indicates that these two user are exactly similar even the same.

Further, in order to use more information of the two users, we also take the number of invoked services by each user and that by both users which implies the QoS preference and style of users into account. We propose Tan-NED which combines Tanimoto similarity coefficient [23] with NED. The formula of Tan-NED is as follow:

$$Sim(u, v) = \frac{|I|}{|I_u| + |I_v| - |I|} \times Sim_{ned}(u, v)$$

i.e.,

$$Sim(u, v) = \frac{|I|}{|I_u| + |I_v| - |I|} \times (1 - \frac{\sqrt{\sum_{i \in I}(\frac{r_{u,i} - r_{umin}}{r_{umax} - r_{umin}} - \frac{r_{v,i} - r_{vmin}}{r_{vmax} - r_{vmin}})^2}}{\sqrt{|I|}}) \tag{4}$$

All the contractions mentioned in section 2 can be eliminated by our Tan-NED. Adopting Eq.(4) we get the following results which are consistent with the facts:

$$Sim(user_3, user_1) > Sim(user_3, user_2)$$

$$Sim(user_4, user_5) = 0.3381$$

$$Sim(user_7, user_6) > Sim(user_7, user_8)$$

3.2   Feedback-corrected Tan-NED Collaborative Filtering

Tan-NED can calculate the similarity between two users, based on it, a novel memory-based
CF approach named Tan-NED CF is proposed. Tan-NED CF predicts the unknown QoS-value
$r_{(u,i)}\prime$ of service i towards user u based on the already available QoS-values of service i towards
other users that are similar with user u. The more similar user v to user u is, the greater user
v's QoS-value influences on $r_{(u,i)}\prime$. The normalized predicting value $\hat{r}_{(u,i)}\prime$ can be calculated by
Eq.(5), and then we recovers the normalized value to the original scale of user u by the maximal
and minimal values of user u. The QoS-value predicted by Tan-NED CF is defined as follow:

$$\hat{r}_{(u,i)}\prime = \frac{\sum_{v \in U} Sim(u, v) \times nr_{v,i}}{\sum_{v \in U} Sim(u, v)} \tag{5}$$

$$r_{(u,i)}\prime = r_{umin} + (r_{umax} - r_{umin})\hat{r}_{(u,i)}\prime \tag{6}$$

U is the set that contains the similar users to user u. Each element $v \in U$ has also invoked
service i. $nr_{v,i}$ denotes the normalized QoS-value of user v on service i in matrix M_{nu} which is
row-normal. $Sim(u, v)$ can be calculated by Eq.(4), and r_{umax} and r_{umin} are the maximal and
minimal QoS-values of user u in the original matrix M. Then we employ the feedback (i.e., the
difference value between prediction QoS value and real QoS value of invoked services) to correct
the following prediction. The correction is a continuous process, so uncertain abnormal real QoS
values may appear which cause "dirty" feedback to the correction process. For example, if a users
router is congested then the RTT of service invoked by this user will become very long, which
will not response the real QoS of service. In order to reduce the impact of "dirty" feedback, we
adopt a Gaussian distribution coefficient to control the correction efforts of feedback. The QoS
value prediction of feedback corrected Tan-NED is as follow:

$$r_{u,i} = \Delta r \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{(\Delta r - \frac{\Delta r_{min} + \Delta r_{max}}{2})^2}{2\sigma^2}} + r_{u,i}\prime \tag{7}$$

Where Δr denotes the difference value between real QoS and prediction QoS of the previous
service invoked by user u, i.e., $\Delta r = r_{real} - r_{pred}$. Δr_{min} is the minimum value of historical
Δr and Δr_{max} is the maximum value of historical Δr. σ is the standard deviation of Gaussian
distribution which determines the amplitude of distribution, and an appropriate σ value will be
obtained by experiments. The feedback correction is a continuous recursive process along with
every service invoking.

3.3   Feedback-corrected Tan-NED CF for Service Recommendation

In the case of candidate services having equivalent functions, the predicted QoS-values which
are calculated by Feedback-corrected Tan-NED can be used for service recommendation, and the
service that has the best predicted QoS performance will be recommended to the corresponding
user. Then, our approach enables personalized service recommendation using a small amount
of available QoS value without the time-consuming and costly actual service invocation. In
service recommender system, either services or users may be remotely distributed in different

location. Besides, the network performances which influence the QoS of services are highly dynamic. Hence, the QoS styles and preferences of user are quite different from each other. Since the proposed approach considers the diversity of QoS style and preference and adopts the information of similar users to make the prediction, it is applicable to a variety of environments.

4   Evaluative Experiments

4.1   QoS Dataset

In order to has sufficient data to evaluate our approach, we use the web-service QoS set [24] which contains 1873838 real RTT (route trip time) records on 5825 real web-services from 339 distributed service users. To our knowledge, this is the largest dataset in the domain of service-computing. In order to collect the data, Zheng et al. monitor 5825 web-services using 339 distributed planet-lab computers. Assuming that a $M \times N$ user-service matrix contains N services, M users and T non-null records, then the density of this matrix can be defined as $density = \frac{T}{(M \times N)}$. According to this concept, the density of user-service matrix used for evaluation is 94.9%.

4.2   Evaluation Metric

We adopt the MAE (Mean-Absolute-Error) metric to evaluate the prediction accuracy of proposed approach. MAE denotes the average-absolute-deviation between the ground-truth values and the predictions values. It is defined as follow:

$$MAE = \frac{\sum_{u,i} |\bar{r}_{u,i} - r_{u,i}|}{N} \qquad (8)$$

$r_{u,i}$ is the prediction RTT of web-service i towards user u, $\bar{r}_{u,i}$ is the real RTT of i observed by u, N is the number of predicted RTTs. The less the value of MAE is, the better the accuracy of prediction is.

4.3   Experimental Setup

The experiments are divided into three parts, namely, performance of similarity measures, impact parameters of prediction, and performance of prediction approaches. First, we compare the performance of our Tan-NED with other similarity measures. In this experiment, we use the original user-service matrix. The RTT records of matrix are divided into two parts: 80% of the records as the training set and 20% of the records as the test set.

Then, we measure the impact of σ (controls the correction effort of feedback), neighbour size k (top-k similar users to calculate the prediction QoS value), and density of matrix. In the third part, we compare the performance of our feedback-corrected Tan-NED with other prediction approaches. In order to evaluate the accuracy of RTT value prediction by different prediction algorithms, the user-service RTT records in the original matrix is removed randomly to generate ten sparse matrices. As Section 4.1 defined, the densities of these ten matrices are incremental with the step-size of 2%, their densities range from 2% to 20%. We adopt these small density matrices in order to get closer to the practical situation that a user may only invokes limited number of services in large amount of available services. So, the real user-service matrix is generally very sparse. We divide each of the ten matrices into three parts: 70% of the RTT records as the training set, 10% as the feedback set, 20% as the test set.

4.4   Performance Evaluation of Similarity Measurements

In order to validate the validity of proposed Tan-NED similarity measurement (Eq.(4)), we compare its performance with other two well-known similarity measurements: COS (Eq.(2)) and PCC (Eq.(1)). Then, we combine Tan-NED, COS, and PCC with formula Eq.(6) to predict the missing RTT values respectively. Figure 1 presents the accuracy of predictions by different similarity measurements. As Figure 1 shows, the proposed Tan-NED consistently outperforms other compared approaches under different k-values; even the worst-case of Tan-NED still over-match the best cases of the compared approaches. Therefore, compared with COS and PCC, the proposed Tan-NED significantly improves the accuracy of prediction.
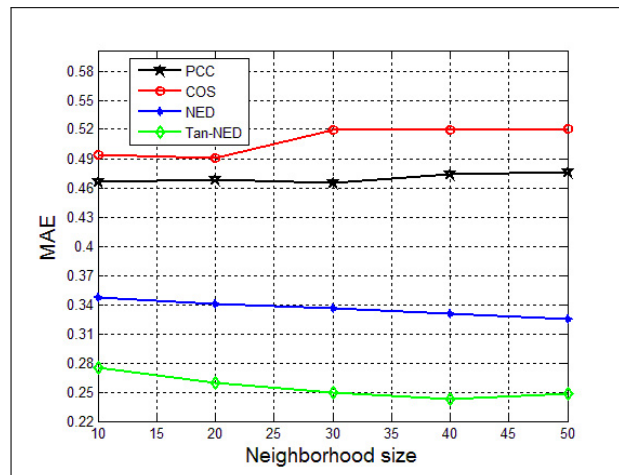
Figure 1: Performance comparison of similarity measures

4.5   Impact of Parameters

Impact of σ

The parameter σ determines the amplitude of distribution which controls the correction efforts of feedback. The higher the value, the more flat the distribution, which means that the probability distribution of Δr ($r_{real} - r_{pred}$) is dispersive. Then, the correction efforts of different Δr value are approximate. Whereas, the lower the value, the steeper the distribution. It means that a few values which close to the median of Δr have strong efforts, however, other values have relatively weak efforts. In this experiment, we increase the parameter σ from 0.1 to 1.0 with the step-size of 0.1 in order to study the impact of ŚŇ on the prediction result. The original user-service matrix is adopted, and the neighbour size k is set to 30. The influence of parameter σ on prediction is presented in Figure 2. As it shows, the values of MAE first slightly decline and then sharply rise. When $\sigma = 0.4$, it hits the bottom. This experiment result indicates that the prediction accuracy can be improved by adjusting the amplitude of correction-effort distribution.

Impact of neighbour size k

In the proposed Tan-NED approach, the neighbour size k determines the number of similar users used for missing value prediction. It acts an important role in the prediction performance. If the value of k is too low, many similar users will be filtered out. If the value of k is too high, dissimilar usersĄŻ records will be considered to calculate the prediction QoS value. The neighbor size k is increased from 10 to 100 with the step-size of 10 in order to study the impact
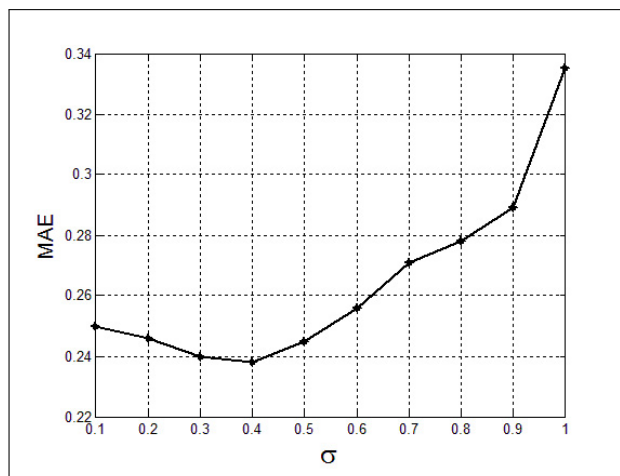
Figure 2: Impact of Sigma

of k. The original user-service matrix is employed, and the parameter σ is set to 0.4. The impact of k on the prediction of Tan-NED is presented in Figure 3. As it shows, the value of MAE first slightly declines and then slightly rises, indicating that our Tan-NED achieves best performance for this dataset when $k = 50$. The deviation between the highest value and the lowest value is only 0.041, which means that our approach is not sensitive to the neighbour size. It is because that our approach uses the value of similarity degree to restrict the effects of each similar user when calculating the prediction QoS value (shown in Eq.(5)).
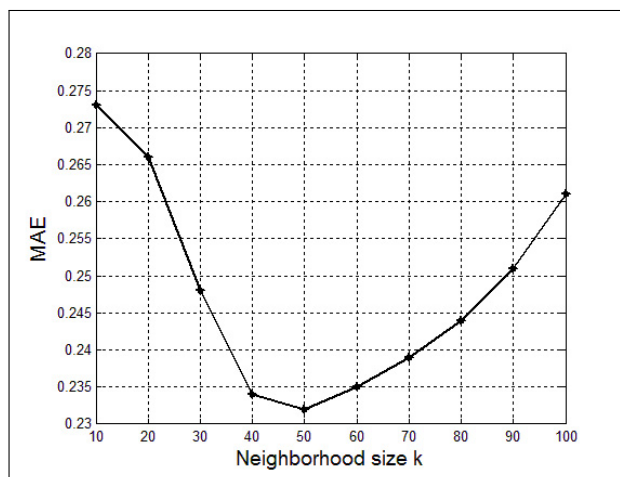
Figure 3: Impact of neighbour size

Impact of User-service Matrix Density

Matrix density denotes the proportion of non-null records that can be used for missing value prediction in the matrix. This section studies the impact of the matrix density on the accuracy of Tan-NED. In this experiment, the density is increased from 0.04 to 0.2 with the step-size of 0.02. The parameter σ is set to 0.4 and k is set to 50. The impact of matrix density on the accuracy of Tan-NED prediction is presented in Figure 4. As it shows, the value of MAE declines as the density increase. The result of this experiment denotes that the prediction of Tan-NED becomes more accuracy as the matrix density increase. The reason is that a more intensive

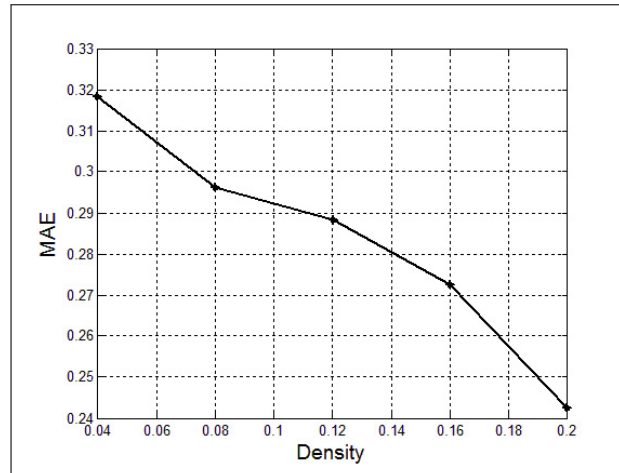matrix provides more reference information for Tan-NED to predict missing value.

Figure 4: Impact of user-service matrix density

4.6 Performance Evaluation of Prediction Approaches

We compare the proposed feedback corrected Tan-NED with other three prediction approaches: UPCC (User-based CF adopting PCC), UMEAN (User-Mean) and WSRec (Web-services recommender) in order to validate its effectiveness. UPCC refers the information of similar users to predict the missing value [7, 19]. WSRec [11] is a novel memory-based CF for web-service recommendation, which achieves a relatively good performance. UMEAN uses the average RTT values of other web-services from the same user to make missing value prediction. The predictions of these four approaches are influenced by the neighbour size k. The neighbour size is increased from 10 to 50 with the step-size of 20 in this experiment. The parameter σ of Tan-NED is set to 0.4 and the confidence weight of WSRec is set to 0.1[1].

Figure 5 presents the prediction accuracy measured by MAE of the evaluated approaches. The three subfigures of Figure 5 correspond to the neighbour-size of 10, 30, and 50 respectively. We increase the density of matrix from 0.02 to 0.2 with the step-size of 0.02. Each subfigure shows the value of MAE with the matrix density changes. As shown in Figure 5, our feedback corrected Tan-NED is significantly superior to other compared approaches. When the density of user-service matrix becomes sparser, the co-valued dimensions between vectors decrease. It means that the number of available values used for missing value prediction is limited, which expands the gap of performance between Tan-NED and compared approaches. As the increase of density, the improvement rate of Tan-NED declines due to each approach has enough available values for prediction. However, in practical situation, the user-service matrix is usually very sparse. Moreover, as each subfigure shows, the deviation between the maximum MAE and the minimum MAE of tan-NED is small. It means that Tan-NED keeps a stable MAE performance under the different density of user-service matrix.

The results of this experiment indicate two features of Tan-NED: 1) compared with other existing approaches, the sparser the user-service matrix is, the more superior of Tan-NED is; 2) the performance of Tan-NED is insensitive to the decrease of matrix density, namely, even with limited available QoS records Tan-NED can also make relatively accurate prediction. These two features of Tan-NED are suitable for the actual situation of real-world service environment. In
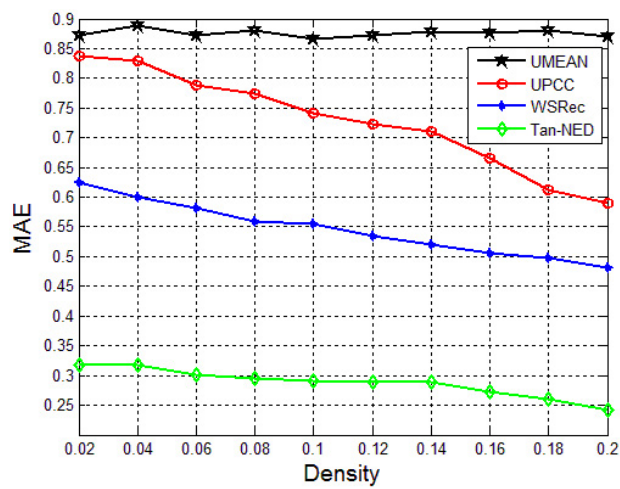
---

[1]The confidence weight in WSRec denotes the impact of user-based method on the final prediction result. As [11] discussed, when confidence weight is set to 0.1, it achieves the best performance.
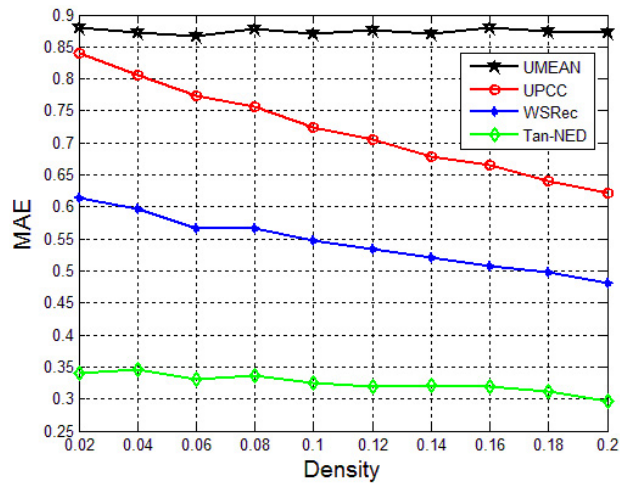
(a) Set the number of neighbour to 10
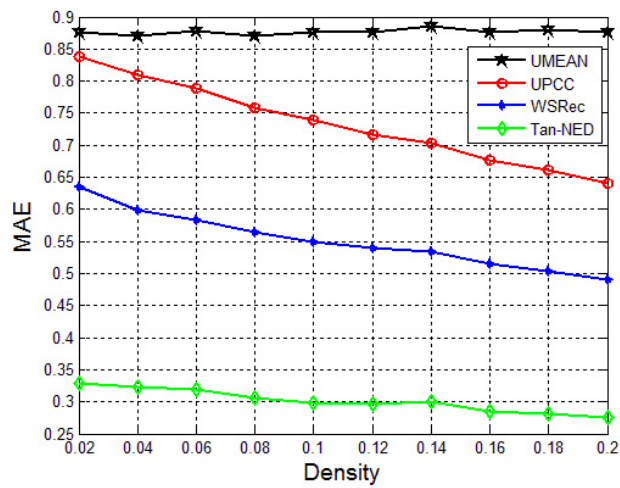
(b) Set the number of neighbour to 30

(c) Set the number of neighbour to 50

Figure 5: The Performance of compared prediction approaches

real-world service recommendation, the density of matrix is generally very sparse. Therefore, compared with existing approaches, our approach can make more accurate and stable prediction for QoS-value.

5   Conclusion

This paper proposes a feedback corrected Tan-NED approach to solve the issue of real-world service personalized recommendation. It studies the features of the QoS-values of real-world services, and proposes a novel similarity measurement which seeks similar users more accurately and provides a basis for accurate QoS-value prediction. Then, the proposed approach can use a small number of available QoS-values from similar users to predict the service QoS-value for the user according to his personalization. In the service recommender system, the proposed approach assists service users to select the service with optimal QoS from a number of function-equivalent services instead of conducting the costly actual service invocation. In order to evaluate the performance of feedback corrected Tan-NED, this paper conducted comprehensive evaluative experiments using a real-world web-service dataset which has sufficient QoS records. Experiment results indicate that compared with existing approaches the proposed approach improves the accuracy of QoS-value prediction significantly.

Since dynamic is a new feature of IoT environment, the QoS of real-world service changes with time frequently. In the future work, we will focus on enhancing the efficiency of our approach to handle the dynamic QoS issue. It may be addressed by using the latest advanced technologies of machine learning.

Acknowledgement

Bibliography

[1] Gustafaason, J. (2011); Integration of wireless sensor and actuator nodes with IT infrastructure using service-oriented architecture, *IEEE Trans Industrial Informatics*, ISSN 1551-3203, 6(1): 1-10.

[2] Guinard, D.; Trifa, V. (2010); Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services, *IEEE Trans Services Computing*, ISSN 1939-1374, 3(3): 223-235.

[3] ICT FP7 OPEN IoT Project. Open source solution for the internet of things into the cloud, (2011). http://vmusm03.deri.ie/.

[4] EPFL GSN project (2009). http://sourceforge.net/apps/trac/gsn/.

[5] Cosm. Cosm platform, (2007). https://cosm.com/.

[6] Perera, C.; Zaslavsky, A.; Christen, P. (2013). Context-aware sensor search, selection and ranking model for internet of things middleware. *14th IEEE International Conference on Mobile Data Management*, 314–322.

[7] Sreenath, R.M.; Singh, M.P. (2003); Agent-based service selection, *J Web Semantics*, ISSN 1570-8268, 1(3): 261-279.

[8] Zhang, L.J.; Zhang, J.; Cai, H. (2007) Services computing, *Springer and Tsinghua University Press*, ISSN 0895-4852.

[9] Moser, O.; Rosenberg, F.; Dustdar, S. (2008). Non-intrusive monitoring and service adaptation for ws-bpel, *17th Intl Conf. on World Wide Web*, 815–824.

[10] Papazoglou, M; Georgakopoulos, D. (2003). Service-oriented computing, *Communications of the ACM*, ISSN 0001-0782, 46(10): 25–28.

[11] Zheng, Z.; Ma, H.; Lyu, M.R.; King, I. (2009). Wsrec: A collaborative filtering based web service recommender system, *7th Intl Conf. Web Services*, 437–444.

[12] Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P.; Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of net news, *ACM Conf. Computer Supported Cooperative Work*, 175–186.

[13] Shardanand, U.; Maes, P. (1995). Social information filtering: Algorithms for automating word of mouth, *SIGCHI Conf. Human Factors in Computing Systems*, 210–217.

[14] Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms, *10th Intl Conf. World Wide Web*, 285–295.

[15] Breese, J.; Heckerman, D.; Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering, *14th Intl Conf. on Uncertainty in artificial intelligence*, 43–52.

[16] Adomavicius, G.; Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans on Knowledge and Data Engineering*, ISSN: 1041-4347, 17: 734–749.

[17] Chen, X.; Zheng, Z.; Liu, X.; Huang, Z.; Sun, H. (2013). Personalized qos-aware web service recommendation and visualization, *IEEE Trans on Service Computing*, ISSN: 1939-1374, 6(1):35-47.

[18] Karta, K. (2005). An investigation on personalized collaborative filtering for web service selection. *Honours Programme thesis, University of Western Australia*.

[19] Shao, L.S.; et al. (2007). Personalized qos prediction forweb services via collaborative filtering. *Intl Conf. on Web Services*, 439–446.

[20] Miller, B.; Albert, I.; Lam, S.; Konstan, J.; Riedl, J. (2003). MovieLens unplugged: Experiences with an occasionally connected recommender system. *8th International Conference on Intelligent User Interfaces*, 263–266.

[21] Zhao, S.; Zhang, Y.; et al. (2013). A multidimensional resource model for dynamic resource matching in internet of things. *Concurrency and Computation: Practice Experience*.

[22] Thio, N.; Karunasekera, S. (2005). Automatic measurement of a qos metric for web service recommendation, *Australian Software Engineering Conference*, 202–211.

[23] Lipkus, A.H. (1999). A proof of the triangle inequality for the Tanimoto distance, *Journal of Mathematical Chemistry*, ISSN: 0259-9791, 263-265.

[24] Zheng, Z.; Ma, H.; Lyu, M.R.; King, I. (2011). QoS-aware Web service recommendation by collaborative filtering, *IEEE Trans on Service Computing*, ISSN: 1939-1374, 4(2): 140-152.