entropy

MDPI

Article

Collaborative Service Selection via Ensemble Learning in Mixed Mobile Network Environments

Yuyu Yin [1,2,†], Yueshen Xu [3,*,†], Wenting Xu [1], Min Gao [1], Lifeng Yu [4] and Yujie Pei [5]

1 School of Computer, Hangzhou Dianzi University, Hangzhou 310018, China; yinyuyu@hdu.edu.cn (Y.Y.); 151050024@hdu.edu.cn (W.X.); 152050068@hdu.edu.cn (M.G.)
2 Key Laboratory of Complex Systems Modeling and Simulation of Ministry of Education, Hangzhou 310018, China
3 School of Software, Xidian University, Xi'an 710126, China
4 Hithink RoyalFlush Information Network Co., Ltd., Hangzhou 310023, China; yulifeng@myhexin.com
5 Fushun Power Supply Branch, State Grid Liaoning Electric Power Supply Co., Ltd., Fushun 113008, China; yyy718@gmail.com
* Correspondence: ysxu@xidian.edu.cn
† The two authors Yuyu Yin and Yueshen Xu contribute equally to this paper, and they are co-first authors.

Received: 18 May 2017; Accepted: 10 July 2017; Published: 20 July 2017

Abstract: Mobile Service selection is an important but challenging problem in service and mobile computing. Quality of service (QoS) predication is a critical step in service selection in 5G network environments. The traditional methods, such as collaborative filtering (CF), suffer from a series of defects, such as failing to handle data sparsity. In mobile network environments, the abnormal QoS data are likely to result in inferior prediction accuracy. Unfortunately, these problems have not attracted enough attention, especially in a mixed mobile network environment with different network configurations, generations, or types. An ensemble learning method for predicting missing QoS in 5G network environments is proposed in this paper. There are two key principles: one is the newly proposed similarity computation method for identifying similar neighbors; the other is the extended ensemble learning model for discovering and filtering fake neighbors from the preliminary neighbors set. Moreover, three prediction models are also proposed, two individual models and one combination model. They are used for utilizing the user similar neighbors and servicing similar neighbors, respectively. Experimental results conducted in two real-world datasets show our approaches can produce superior prediction accuracy.

Keywords: Mobile Service Selection; Mobile Network; Ensemble Learning; QoS Prediction; Abnormal QoS

1. Introduction

With the emergence of 5G era, various mobile devices and tablets have been widely developed and used in enterprises' marketing activities [1]. Developing mobile services has become an increasingly important way for various enterprises to deliver their marketing applications that only customs' functional demands and satisfy their expected non-functional requirements. Since many mobile services have been or are being developed as interfaces to access resources on mobile environments, including 4G, 5G, and WIFI, the number of mobile services has increased dramatically. Meanwhile, users tend to find those services that satisfy their business requirements and also provide high quality service. With a growing number of available services with similar functions, the problem of selecting a suitable candidate service has become an urgent task.

5G networks provide faster speeds and better quality of service (QoS). QoS is a key factor in service selection, both in industry and academia [2]. Recently, QoS-aware service selection has been a critical problem in service and mobile computing [3–5]. This selection is based on a primary premise that all QoS values are pre-provided [6–9]. In real-world cases, however, such a premise is hardly true, since it is impossible to invoke all services at the same time. Therefore, it is an indispensable task to predict QoS.

There has been much research into QoS prediction [10–12]. For example, collaborative filtering (CF) uses historical QoS records to predict unknown QoS values. Many works demonstrate that CF-based prediction approaches have better prediction accuracy [13–15] for utilizing invocation records of the target user/service to identify the similar user/service neighbors. They then predict the missing QoS values collaboratively. Prediction accuracy largely relies on the quality of the identified similar services or users. Several similarity computation methods (e.g., Pearson correlation coefficient or PCC and cosine similarity) are used for the identification of similar user neighbors or service neighbors, but the similarity between two users/services is always overestimated in existing methods, which leads to lower prediction results. We also notice that recent related works pay attention to the improvement of similarity computation. However, in a complex network environment, such as the mixed mobile networks in which networks of different types co-exist (for example, a mobile network mixed with 5G, 4G, Wi-Fi), the abnormal QoS values are easier to generate, and a part of the abnormal QoS data is likely to be mixed with normal QoS data. The abnormal QoS data have the potential to impair the prediction accuracy significantly. Another key issue in QoS prediction is ignored in most current research. Therefore, we aim to detect abnormal QoS data and reduce their impact in QoS prediction in this paper.

To address the above problems, a novel collaborative prediction approach inspired by ensemble learning (i.e., AdaBoost) is proposed in this paper. The proposed approach is oriented to a mixed mobile network environment. Our approach has the capability to improve similarity between two users/services. Moreover, it can also handle abnormal QoS data. We first employ two techniques to discovery two sets of user similar neighbors based on the historical QoS records. The computations and DBScan clustering [16] of these two techniques are similar. Then, we propose an extended ensemble learning model based on an extension of the AdaBoost method that identifies abnormal QoS data and produces clusters of QoS data. AdaBoost, short for adaptive boosting, is a well-known ensemble learning method [17]. Meanwhile, the probability belonging to each cluster can be generated. Next, we filter abnormal users from QoS clusters by investigating the clusters with the proposed ensemble learning method and remove abnormal users from the neighbors set. In this way, the final two neighbors sets can be formed, and the weight of each similar neighbor from the two sets is decided by the probability of belonging to each individual cluster. Following that, we propose two individual collaborative prediction methods using the discovered similar neighbors sets. Finally, we design a combined method to generate the final prediction result. In summary, the main contributions of this paper are as follows:

1. We propose a new neighbors selection method extended from the DBScan algorithm that performs well in handling high data sparsity.
2. We propose an ensemble learning method extended from AdaBoost that can identify abnormal QoS data. The false neighbors can be filtered from the candidate neighbors.
3. We propose two individual collaborative prediction methods, one for user and the other for service. We also propose a combined method that can combine the prediction results of the two individual methods.
4. Experimental results conducted in two real-world datasets show our approaches can produce superior prediction accuracy and have strong flexibility to the experiment setting.

This paper is organized as follows: related work is discussed in Section 2, our framework is presented in Section 3, our prediction models are explained in Section 3, and experimental results are given in Section 4. Conclusion and future work is presented in Section 5.

2. Related Work

Service selection aims to discover quality services for target users, and the effectiveness of a prediction method is the key in service selection. The collaborative filtering (CF for short) method has been popularly used for QoS prediction [6–9]. There are two groups in the CF family, i.e., neighbor-based (focusing on identifying the similar relationship among users or services) and model-based (learning both the latent features of a user and a service and the relation between the latent features of users and services) [18–27].

There are three types of the neighbor-based CF methods: user-based, service-based, and hybrid. Shao et al. [4] standardized QoS values and presented a user-based CF algorithm. In this method, the similarity was computed by PCC, and the results showed that similar users tended to share similar QoS values. Sun et al. [5] normalized QoS values to 0 and 1. They used Euclidean distance to compute the similarity. Context information, like network location or geographical location, is closely related to QoS [6]. Some researchers integrated such context information into the CF method, achieving more accuracy in prediction. Liu et al. [14] proposed a network location–based CF method that identified the potential autonomous systems (AS) based on the IP addresses of users and services. They assumed that the users that are located near to each other had similar network environments and thus were likely to experience similar QoS. Chen et al. [10] constructed a bottom-up hierarchical clustering algorithm utilizing the user geographical location to mine similar regions and further integrated the region similarity into the CF algorithm. Yao et al. [12] proposed a content-based CF algorithm that utilized the description content extracted from WSDL files to mine user preference to service invocation. Wu et al. [18] proposed a time-aware QoS prediction approach. The most advantage is derived from collaborative filtering. This approach first computed user-service pairs that had historical invocation experiences and then used CF-based method to predict QoS values. Chen et al. [7] proposed a hybrid model that combined the service-based CF method and latent semantic analysis. Jiang et al. [8] also proposed a CF-based hybrid model that was a linear combination of user-based CF and service-based CF. Zheng et al. [9] constructed a combination model that combined the prediction results of user-based CF algorithm and service-based CF algorithm with a predefined parameter. However, most of the existing methods ignore the abnormal QoS data. However, involving abnormal data can significantly lower prediction accuracy. In addition, many approaches suffer from data sparsity.

Yin et al. [11] presented three prediction models that all adopted matrix factorization (MF for short) and network location-aware neighbor selection. He et al. [21] proposed a geographic location-based hierarchical MF model, in which the user-service invocation matrix was partitioned into several local matrices, using K-means algorithm. The final prediction result was computed as the combination of the results that had been produced using the whole matrix and local matrices, respectively. Tang et al. [22] proposed a network-aware QoS prediction approach by integrating MF with network information. By employing network information, they computed the network distances among users and further identified user neighborhoods. Xu et al. [23] extended the PMF model (short for probabilistic matrix factorization) with geographical information [28]. Based on the geographical location of the target user, their method learned the user latent feature vector, investigating the impact of the features of similar neighbors. However, model-based methods are vulnerable to data sparsity, which is common in the mixed network environment. This defect can easily lead to inaccurate prediction results.

Additionally, Ma et al. [29] unified the modeling of multi-dimensional QoS data via multi-linear algebra tools and tensor analysis for predicting QoS. In [30], Wu et al. proposed CAP: credibility-aware prediction. CAP employed K-means clustering for identifying the untrustworthy users and cluster QoS values for untrustworthy index calculation and users.

There are quite limited existing studies for QoS prediction considering abnormal QoS data, which is a key issue, especially in the mixed network environment. In this paper, we aim to solve the abnormal QoS data problem in QoS prediction.

3. Collaborative QoS Prediction via Ensemble Learning

Due to the variety of network conditions and the potential changing of user location, the abnormal QoS data can be generated from normal users. If we can identify the abnormal QoS data and further filter the corresponding false neighbors from the neighbors set, the prediction accuracy will have a high probability of improvement.

Considering the following scenario: the three users u_1, u_3, and u_4 are the candidate neighbors of u_2, and x_1 and x_2 are the missing values in Table 1. We find that the QoS values of u_3 and u_4 on s_2 are close, but the QoS value of u_1 is much greater than those of u_3 and u_4. Supposing that the real value of x_1 is 1.5, when we want to predict the value of x_1 based on the known QoS values of u_1, u_3, and u_4, the prediction result will significantly deviate from the real value. Thus, the QoS value of u_1 is the potentially abnormal QoS value. Clearly, if u_1 can be removed from the candidate neighbors set of u_2, the prediction accuracy is likely to be promoted. On the service side, supposing that the real value of x_2 is 8.3, when we try to predict the value of x_2, the smallest QoS value of s_3 should be regarded as one abnormal value. It is difficult to manually set a fixed threshold or determine the false neighbors by comparing the QoS values to a fixed threshold. In this paper, we propose an ensemble learning method, extended from AdaBoost, with the frequency feature vector as the input. According to the value range of historical QoS values, we can create several categories of QoS data by classifications. The AdaBoost method can generate the probability of every missing QoS value belonging to each category. We select the top K categories with the largest probabilities. If a QoS value is not contained in any category, this value will be regarded as an abnormal value. Consequently, its corresponded neighbors will be removed from the candidate neighbors.

Table 1. An example of a QoS matrix.

-	s_1	s_2	s_3	s_4
u_1	1.3	8.5	1.7	7.7
u_2	1.1	x_1	8.6	1.8
u_3	1.2	1.5	x_2	8.2
u_4	1.4	1.3	8.0	8.4

Figure 1 illustrates the complete procedure of our approach, which basically consists of the following three steps:

1. Similar neighbors selection. We use a DBScan co-occurrence matrix to compute the similarity between users. The similarity computation result is used to build the similar neighbors set DC_N(u).
2. Neighbors filtering. We discover a feature vector by combining the frequency vectors of a user and a service for the prediction of the corresponding missing QoS value. The feature vector is the input of the ensemble learning model used to generate the probability of belonging to each category. After that, we can filter false neighbors DC_N(u) by selecting the top K categories with the highest probabilities.
3. Missing QoS value prediction. The probabilities associated with different categories are leveraged as the weights that are assigned to all remaining neighbors. Based on DC_N(u), we employ the user-based CF model and the service-based CF model to generate two sets of prediction results. The final prediction results are computed as the linear combination of the two individual results.

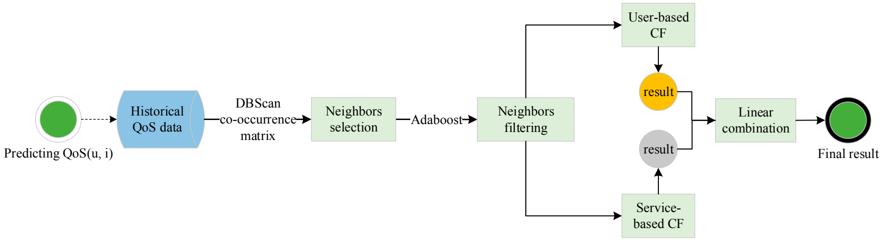

Figure 1. The framework of quality of service (QoS) prediction.

3.1. Similar Neighbors Selection

Similar neighbors selection is a critical step in CF-based method for QoS prediction. Some similarity computation methods (for example, PCC and Euclidean distance) are employed to select neighbors in existing works [4,9,10]. However, most existing methods suffer from the problem of similarity overestimation [4,9], especially in the case of high sparsity. Meanwhile, due to common abnormal QoS data in a mixed mobile network environment (for example, a mobile network mixed with 4G, 5G, and Wi-Fi), the prediction accuracy can easily decline. In this paper, we propose a novel similarity computation method based on DBScan clustering aiming to solve this problem. DBScan is a density-based clustering algorithm in which the clustering structure is generated according to the connection among data. The data to be clustered are interconnected according to connectivity. DBScan computes the data distribution with a group of parameters (ε and *MinSamples*). Compared to K-means, DBScan does not require us to pre-define the number of clusters and is applicable to the clustering task of data with various shapes rather than as K-means, which usually only performs well on data with a sphere shape. DBScan is also able to better identify noise and is more robust to outliers. As a result, the proposed similarity computation method is less affected by abnormal QoS data. Two phases are involved in our method: co-occurrence matrix construction and similar neighbors selection.

3.1.1. Phase 1: Co-Occurrence Matrix Construction

The $m \times n$ matrix M denotes m users and n services, representing the user-service invocation relationship. Each entry $q_{i,j}$ in matrix M denotes the QoS value of user i invoking service j. Users invoked the same service are clustered by employing the DBScan algorithm. The users or services having similar QoS values will be clustered into one same group. C_i^f or C_j^f denotes the f_{th} cluster of user i or service j, where f is the cluster index.

After QoS values clustering, we construct the co-occurrence matrix to store the clustering result. The initial value of each entry is set as 0 in the co-occurrence matrix. The co-occurrence matrix is an $m \times n$ symmetric matrix, notated as A, and the entry $a_{i,j}$ of A denotes the times of user u_i and user u_i being clustered in the same group. For example, user u_1 and user u_3 are clustered in the same group on two services s_0 and s_2 in Figure 1, then the entry $a_{1,3}$ = $a_{3,1}$ = 2 in Table 2. A larger entry value in matrix A means a higher similarity between the two users associated with the entry.

Table 3 shows an example of a 3 × 8 QoS matrix consisting of three services and eight users. We use DBScan algorithm to cluster the QoS values of each service. The clustering results are given in Figure 2. The co-occurrence matrix A can be constructed according to the clustering results and shown in Figure 3. The values of $a_{5,8}$ and $a_{8,5}$ is 2, because they are clustered into the same group of C_0^1 (on s_0) and C_2^0 (on s_2).

Table 2. Co-occurrence matrix A.

-	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
u_0	0	0	0	0	0	0	0	0	0
u_1	0	0	0	2	0	1	0	1	1
u_2	0	0	0	0	0	2	0	0	1
u_3	0	2	0	0	0	1	2	0	1
u_4	0	0	0	0	0	0	0	0	0
u_5	0	1	2	1	0	0	0	0	2
u_6	0	0	0	0	0	0	0	0	0
u_7	0	0	0	0	0	0	0	0	0
u_8	0	1	1	1	0	2	0	0	0

Table 3. An example of QoS matrix M.

-	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8
s_1	-	1.1	2.7	1.2	0.9	2.9	-	-	2.8
s_2	-	1.2	0.2	0.6	-	0.3	0.6	1.0	-
s_3	-	1.6	-	1.5	1.6	1.6	0.7	-	1.6

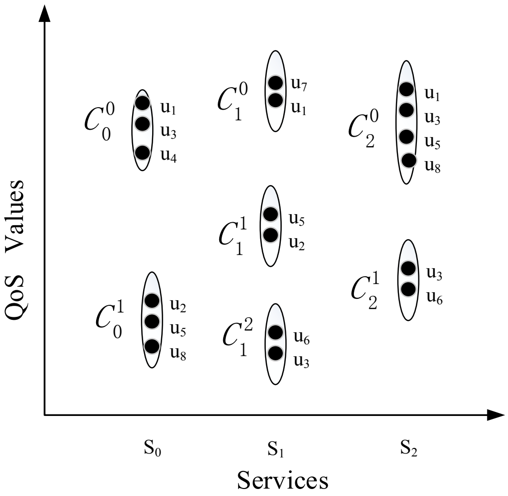

Figure 2. DBScan clustering result.

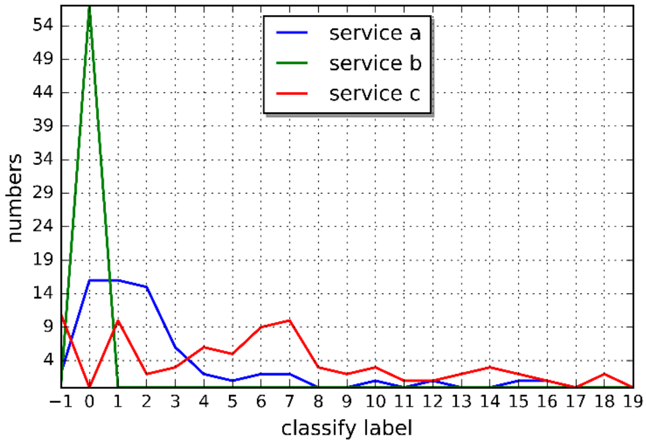

Figure 3. The service frequency vector.

3.1.2. Phase 2: Neighbors Selection

After constructing the user co-occurrence matrix and service co-occurrence matrix, we can identify the similar neighbors. Similar neighbors selection is a core step in achieving QoS prediction with high accuracy, since false similar neighbors are likely to impair the prediction accuracy.

In the co-occurrence matrix, an entry represents the similarity between two users. We choose the top K most similar neighbors to predict the missing value.

The CF algorithm assumes that the invocation behavior of a user will be relatively stable. Such an assumption is reasonable in many cases, but in some cases, it is not applicable. This is because the historical QoS records are likely to contain noise data, and the similarity computation on the noise data tends to fail to select the proper neighbors. Also, if the preference of the target user or service changes, which indeed can happen, it is naturally inaccurate to predict missing QoS values based on the assumed fixed preference.

To fix such issues, we further filter the similar neighbors by utilizing the classification result of AdaBoost. Such filtering can improve the selection accuracy of similar neighbors. Figure 4 shows the framework of the AdaBoost classifier. In this paper, we adopt an ensemble learning method (i.e., AdaBoost algorithm) and use the decision tree as the weak classifier to further filtering similar neighbors. The weak classifier learns different weights to form different classifiers from the distribution of samples. The ensemble classifier aggregates the classification results from the individual classifiers to produce the final result. The detailed explanation of the proposed classifier is given in the following section.

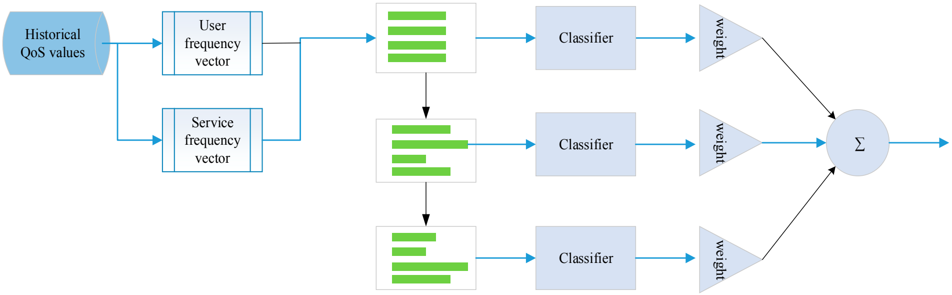

Figure 4. The Adaboost classifier.

3.2. Similar Neighbors Filter

3.2.1. Feature Selection

In this paper, we discover a new set of features: the frequency feature vector, which can better depict the individual features of user-service relationship,

In real dataset [9], QoS values are within a certain range; for example, the response time is [0, 20]. By rounding off the response time, we can build 21 discrete categories. The label of each category is assigned to an element in the set $\{-1, 0, 1, \ldots, 19\}$. Thus, the round-off QoS value of a special service can fall into a special category.

The entry in the frequency feature matrix is defined as the frequency of times occurring in a certain category of the QoS value generated by the invocation of a user to a service. Formally, the user frequency feature matrix is defined as U: M × d, and the service frequency feature matrix is defined as S: N × d. M is the number of users, N is the number of services, and d is the number of categories.

We use the frequency vectors of users and services to construct the feature vector of the target user invoking the target service. For a missing value, the input feature vector is a vector with integers as the elements and 42 dimensions (21 user features plus 21 service features). The input feature vector combines the features of user and service to improve the depicting capability and classification accuracy.

3.2.2. Frequency Feature Vector

Figures 3 and 5 show three services feature vectors and three user feature vectors. The vertical ordinate numbers are the times of round-off QoS value of a user invoking a service on a special classify label. Some latent attributes of users and services can be seen from the two figures.

The feature quality has a great impact on the generalization of the learning model. The frequency feature vector of user-service pairs can better depict the relation of the target user-service pair in order to distinguish from other user-service pairs and further improve the prediction accuracy of classification.

First, let us focus on *service b* and *user b*. Almost all QoS values are in the same category, which indicates the features being stable with little variability. For instance, all QoS values of service b are classified into Classify Label 0. It can be further inferred that if the target service and target user are *service b* or *user b*, the corresponding category is highly likely to be the current category. Thus, for these types of users and services, the classification result is easy to generate.

In contrast, the QoS values of *service a* and *service c* are distributed in many different categories. Especially for *service c*, whose stability is low and variability is high. For such type of services, the classification result is hard to predict, but if we can fully utilize the user features, the classification accuracy can be improved. For example, when the task is to predict the missing QoS of *user a* invoking *service c*, although *service c* is unstable, *user a* has two stable categories, so the classification result is in the two categories with high probability.

Even if both the user and the service are unstable (e.g., the pair of *user c* and *service c*), the curves of the frequency vectors are clearly different. In the historical records, the combination of frequency vectors of different users and different services corresponds to different categories. The AdaBoost algorithm is mainly employed for the supervised learning problems and can learn different combination patterns.

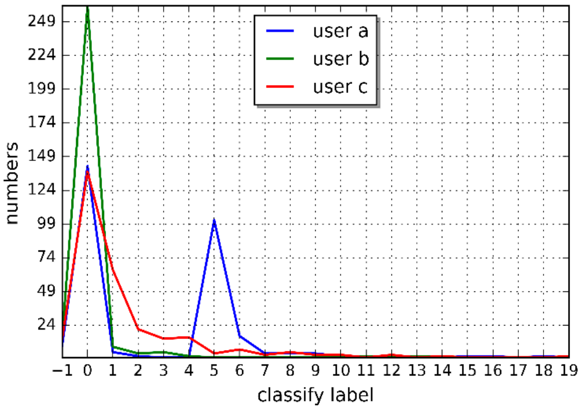

Figure 5. The user frequency vector.

3.2.3. Similar Neighbors Filter

We can calculate the probability of the QoS value belonging to each label using the ensemble learning model. Then we select the K labels with the largest probabilities in all labels. The similar neighbors set of a user is constructed as follows:

$$\overline{N}(u) = \{v|v \in N(u), q_{v,j} \in \textit{label set}\} \tag{1}$$

where $N(u)$ is the similar neighbor set of user u and $q_{v,j}$ is the real QoS value of similar user v having invoked service j. The *label set* is the set of the k labels with largest probabilities of the missing QoS values. $\overline{N}(u)$ is the subset of $N(u)$, where the members satisfy the condition $q_{v,j} \in \textit{label set}$.

Here is an example of $\overline{N}(u)$. Assuming that $N(u) = \{v_1, v_2, v_3, v_4\}$, $\textit{label set} = \{1,2,3\}$, and $q_{v1,j}$ = 1.1, $q_{v2,j}$ = 2.1, $q_{v3,j}$ = 3.1, $q_{v4,j}$ = 7.1. Clearly, because of $q_{v_4,j} \notin \textit{label set}$, we filter the neighbor user v_4 from $N(u)$. We can therefore get the new similar neighbor set $\overline{N}(u) = \{v_1, v_2, v_3\}$.

Similarly, the similar neighbors of a service can be produced as follows:

$$\overline{N}(j) = \{h|h \in N(j), q_{u,h} \in \textit{label set}\} \tag{2}$$

where $N(j)$ is the preliminary similar neighbors set of service j and $q_{u,h}$ is the real QoS value of user u having invoked the similar service h. $\overline{N}(j)$ is the subset of the set $N(i)$, where the members satisfy $q_{u,h} \in \textit{label set}$.

3.3. The Proposed Prediction Methods

We propose a new user-based CF method that utilizes the probabilities associated with the labels of the QoS values, replacing the traditional similarity in existing CF-based methods. Since the ensemble learning model takes an important role in our framework, we name the proposed method User-based CF with Ensemble learning (UCF-E), and the corresponded prediction is given by,

$$UCF - E_{u,j} = \frac{\sum\limits_{v \in \overline{N}(u)} w_{v,j} \cdot q_{v,j}}{\sum\limits_{v \in \overline{N}(u)} w_{v,j}} \tag{3}$$

where $w_{v,j}$ is the probability of $q_{v,j}$ belonging to the label and $q_{v,j}$ is the real value that user v received after invoking service j.

In a similar way, we propose a new service-based CF method using the probability belonging to the labels of the missing QoS values, also replacing the traditional similarity. This proposed method is named Service-based CF with Ensemble learning (SCF-E), and the prediction is given as,

$$SCF - E_{u,j} = \frac{\sum\limits_{h \in \overline{N}(j)} w_{u,h} \cdot q_{u,h}}{\sum\limits_{h \in \overline{N}(j)} w_{u,h}} \tag{4}$$

where $w_{u,h}$ is the probability of $q_{u,h}$ belonging to the label. $q_{u,h}$ is the real value after user u invoked service h.

However, considering the high sparsity of the service invocation records, the UCF-E method or SCF-E method probably does not fully utilize all of the information in the historical QoS records. To further improve prediction accuracy, we propose a hybrid prediction method that combines the prediction results of UCF-E and SCF-E, aiming to fully take advantage of the whole QoS data. We name this method Hybrid CF with Ensemble learning (HCF-E), given as follows:

$$HCF - E_{u,j} = \theta \cdot UCF - E_{u,j} + (1 - \theta) \cdot SCF - E_{u,j} \tag{5}$$

where the parameter θ is used to control the proportions of the two individual models. UCF-$\mathrm{E_{u,j}}$ and SCF-$\mathrm{E_{u,j}}$ are the prediction values of the two individual models respectively. In the extreme case of being 0, the hybrid model is degraded to SCF-E. If θ is 1, the hybrid model is degraded to the UCF-E.

4. Experimental Results

4.1. Dataset and Experiment Setting

The experiment is conducted with WSDream dataset [8], which is a real-world dataset. It has two sub-datasets and consists of 339 users and 5825 services. The aim of WSDream is mainly to evaluate throughput and response time.

As for the experiment setting, we select part of QoS records from the dataset randomly for training set, and all the other for testing set. In the experiment, four different training set densities are configured: 5%, 10%, 15%, and 20%. If the training set density is configured 15%, it means that 15% of the whole data is used for training set, while the other 85% data are for testing. Every set of experiments is run 10 times, and the average result is used for evaluation. The experimental results are given in Table 4 (response time dataset) and Table 5 (throughput dataset).

Table 4. Accuracy comparison (a smaller value means higher accuracy, response time dataset).

Model	Training Set Density (Response Time)							
	Density = 5%		Density = 10%		Density = 15%		Density = 20%	
	MAE	NMAE	MAE	NMAE	MAE	NMAE	MAE	NMAE
UMean	0.8821	1.0816	0.8793	1.0782	0.8791	1.0780	0.8793	1.0782
IMean	0.7221	0.8854	0.7083	0.8685	0.7011	0.8597	0.7005	0.8590
UPCC	0.7573	0.9286	0.7128	0.8740	0.6872	0.8426	0.6207	0.7611
IPCC	0.7132	0.8745	0.7352	0.9015	0.6921	0.8487	0.6587	0.8077
UIPCC	0.6622	0.8120	0.6347	0.7783	0.6261	0.7677	0.5972	0.7323
SVD	0.5731	0.7027	0.5605	0.687	0.5478	0.6717	0.5312	0.6513
LBR	0.5520	0.6769	0.5374	0.6589	0.5189	0.6363	0.4957	0.6078
NIMF	0.5323	0.6527	0.5136	0.6297	0.5011	0.6144	0.4710	0.5775
CAP	0.5452	0.6730	0.5040	0.6184	0.4865	0.5969	0.4636	0.5688
SCF-E	**0.5406**	**0.6633**	**0.4777**	**0.5861**	**0.4458**	**0.5470**	**0.4383**	**0.5378**
UCF-E	**0.5149**	**0.6318**	**0.4395**	**0.5393**	**0.4178**	**0.5126**	**0.4094**	**0.5024**
HCF-E	**0.5091**	**0.6247**	**0.4357**	**0.5346**	**0.4098**	**0.5029**	**0.3929**	**0.4821**

Table 5. Accuracy comparison (a smaller value means higher accuracy, throughput dataset).

Model	Training Set Density (Throughput)							
	Density = 5%		Density = 10%		Density = 15%		Density = 20%	
	MAE	NMAE	MAE	NMAE	MAE	NMAE	MAE	NMAE
UMean	50.937	1.1729	51.343	1.1684	50.941	1.1676	51.185	1.1639
IMean	31.798	0.7322	31.820	0.7242	31.688	0.7263	31.701	0.7208
UPCC	30.829	0.7099	29.054	0.6612	28.357	0.6499	28.114	0.6393
IPCC	31.112	0.7164	29.936	0.6813	30.100	0.6899	30.609	0.6960
UIPCC	29.538	0.6802	28.185	0.6414	27.556	0.6315	27.422	0.6235
SVD	58.623	1.3503	30.188	0.6870	24.106	0.5525	22.065	0.5017
LBR	28.032	0.6340	27.445	0.6208	26.443	0.5981	25.112	0.5680
NIMF	27.331	0.6182	26.405	0.5981	25.413	0.5755	24.102	0.5454
CAP	26.331	0.5955	24.442	0.5528	24.113	0.5454	23.678	0.5355
SCF-E	**26.954**	**0.6277**	**23.582**	**0.5408**	**22.832**	**0.5164**	**21.822**	**0.4975**
UCF-E	**24.409**	**0.5684**	**20.268**	**0.4648**	**19.046**	**0.4307**	**17.836**	**0.4066**
HCF-E	**24.012**	**0.5591**	**19.932**	**0.4571**	**18.964**	**0.4289**	**17.753**	**0.4047**

4.2. Performance Comparison

Some well-known QoS prediction models are implemented for evaluating the proposed model. They are explained below.

1. UMean: Use the mean of each user's historical QoS value as prediction value.
2. IMean: Use the mean of each user's historical QoS value as prediction value.
3. UPCC: User-based collaborative filtering algorithm that uses the historical QoS records of similar users to predict the missing values [12].
4. IPCC: Service-based collaborative filtering algorithm that uses the historical QoS records of similar services to predict the missing values [13].
5. WSRec: Combination of UPCC and IPCC [17].
6. SVD: As a matrix factorization model, this method tries to learn latent factors to mine the user latent features and service latent features [20].
7. LBR: This method selects similar users with geographical location information and take advantage of similar users in matrix factorization [23].
8. NIMF: Contain three predictions models and employs two techniques of matrix factorization and location-aware neighbors selection [5].

9. CAP: Identifies false neighbors and then use reliable clustering results [24] to predict missing QoS values.

Both Mean Absolute Error (MAE) and Normalized Mean Absolute Error (NMAE) are adopted for measuring prediction accuracy. Tables 4 and 5 show the response time and throughput results respectively.

1. All the three proposed models (SCF-E, UCF-E and HCF-E) are better in prediction accuracy.
2. As the training set densities increase, MAE and NMAE values also decrease. Therefore, the more historical QoS records, the better prediction accuracy will be.
3. UCF-E achieves higher prediction accuracy than SCF-E. This is mainly from dataset, the number of users is only 339, but the number of services is 5825. A larger number of services are likely to introduce neighbors not so similar as noise, further to reduce the prediction accuracy.

Parameter sensitivity is also studied in the below subsections.

4.3. Sensitivity Analysis of Classification Precision

The parameter topKLabel (L) controls the number of potential labels of the prediction values. For example, in the case of L being 3 and the classification precision being 0.9017, the probability of prediction value belonging to the three most likely labels is 0.9017. The experimental results are shown in Figure 6, where the training set density is from 5% to 20%.

When L increases, the classification precision first increases rapidly and then tends to converge. It indicates that a small interval of labels can better predict the label. Besides, the classification precisions are close to each other due to convergence in different densities (5% to 20%), which means that our proposed method has a stable performance in high data sparsity.

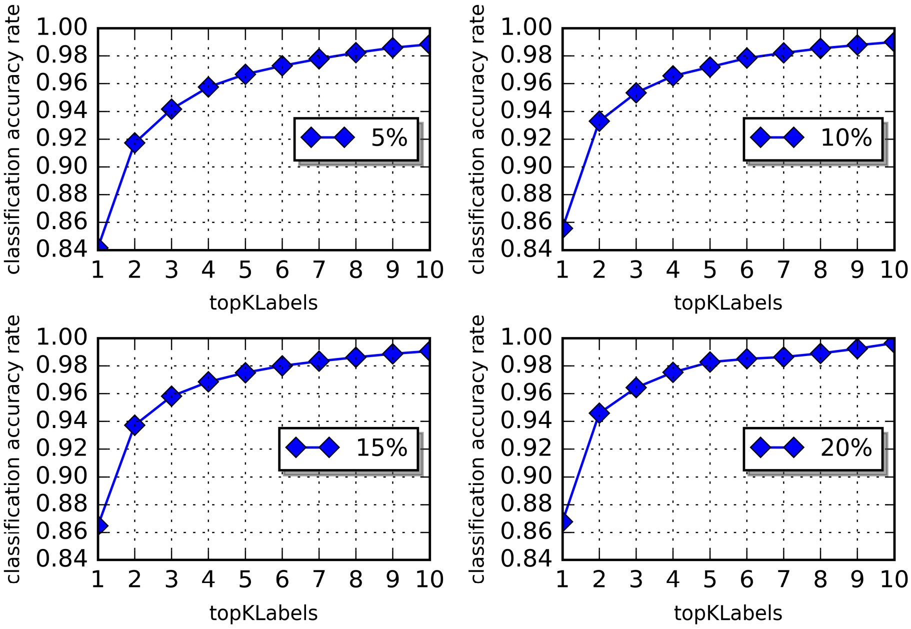

Figure 6. The classification precision with topKLabel (L) increasing in different training set densities.

4.4. Sensitivity Analysis of θ

The parameter θ is used to control the weight of the two individual models (UCF-E and SCF-E) in the combination model. We investigate the sensitivity of our method to θ in the range of 0 to 1. The experimental results are shown in Figure 7, and the training set density is from 5% to 20%. In four different training set densities, the optimal value of θ is all in the value of 0.7–0.9. We set θ to 0.8 as the default. The result indicates that our combination model can achieve better performance by the utilization of the results of both UCF-E and SCF-E. Besides, the MAE in low data sparsity (20%) is clearly lower than that in high data sparsity (5%). This indicates that collecting more QoS data is an effective way to improve the prediction accuracy.

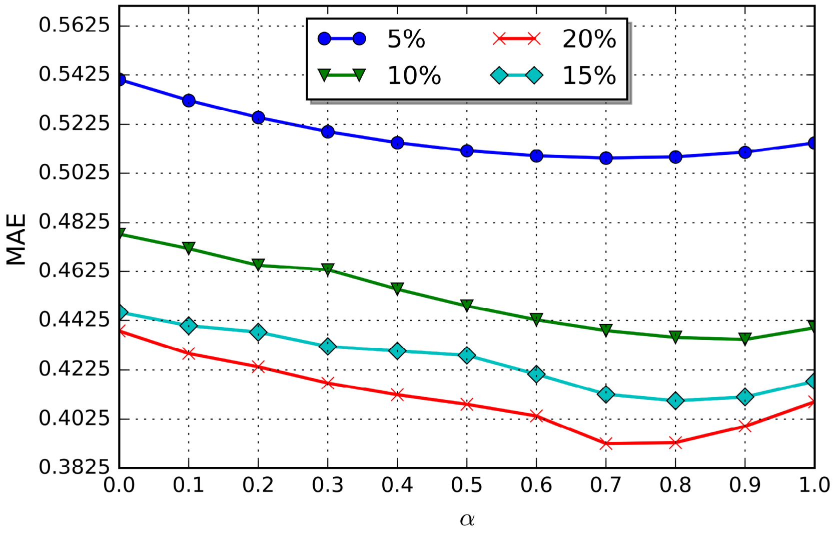

Figure 7. The estimation error with θ increasing in different training set densities.

4.5. Sensitivity Analysis of topKNeighbors (T)

In this paper, we use the parameter topKNeighbors (T) to control the size of the user or service neighborhood. A smaller T can reduce time complexity and reduces the prediction time. We find that the change trends of MAE and NMAE are quite similar, so we report the result of MAE in Figure 8.

In Figure 8, the MAE value first decreases with the increase of T and then increases, and the whole change is slight in the whole value range. After T is larger than 6, some neighbors that are not so similar can reduce the prediction accuracy. At the point of T being 6, the model achieves the best MAE value. Therefore, we set the default value of T as 6.

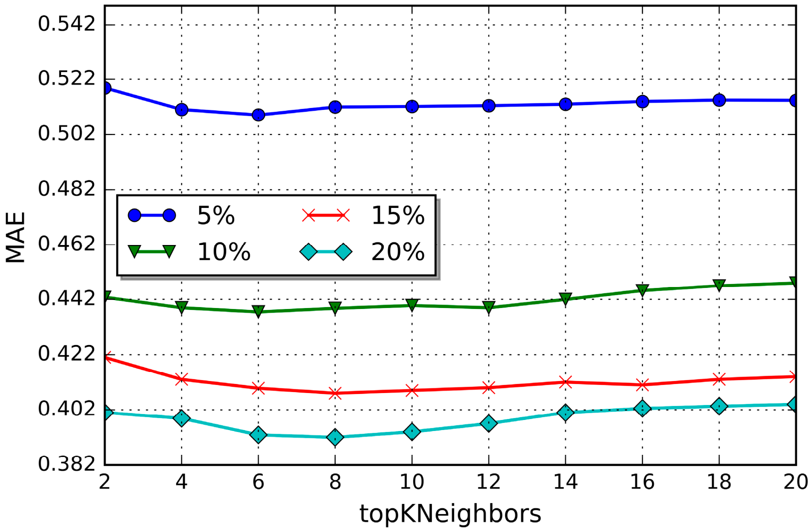

Figure 8. The estimation error with topKNeighbors (T) increasing in different training set densities.

4.6. Sensitivity Analysis of ε

The parameter ε controls the size of core object in DBS can clustering. Using DBScan method, we can select the similar user or service neighbors. In this paper, we use a relatively small to distinguish different QoS values.

As Figure 9 shows, with the increase of ε, the MAE value decreases from 0.01 to 0.04 and then increases. After ε is larger than 0.04, the connectivity of QoS begins to relax, which probably brings some neighbors that are not similar to the target user or service. At the value of ε being 0.04, the model achieves the best MAE value. Therefore, we set the default value of ε to 0.04.

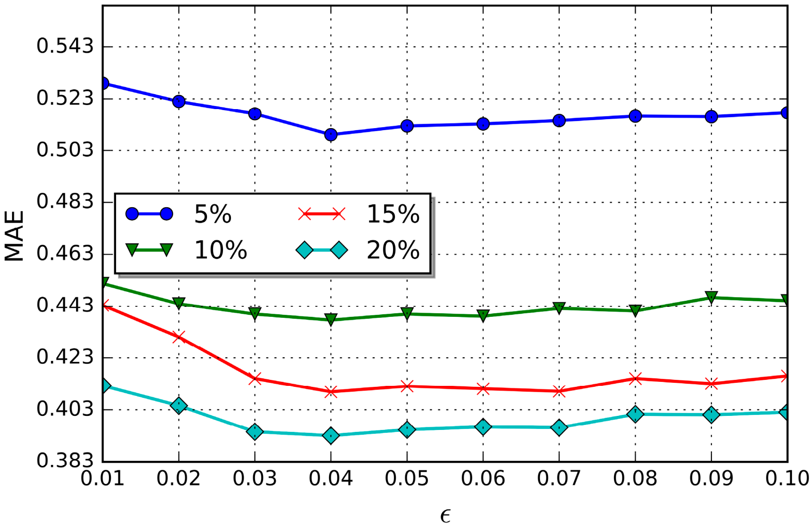

Figure 9. The estimation error with ε increasing in different training set densities.

5. Conclusions and Future Work

A novel collaborative QoS prediction framework that consists of a novel neighbor selection method is proposed in this paper. The proposed novel neighbor selection is based on the DBScan algorithm, which is verified to be effective, especially in cases of high data sparsity. Our approach can also filter false neighbors using the proposed ensemble learning model to generate a quality neighbors set. We also propose a hybrid model that can utilize the results of the two individual models. Experimental results conducted in two real-world datasets show our approaches can produce superior prediction accuracy.

Although our proposed approach has successfully demonstrated that the ensemble learning model can identify false neighbors, some challenges still exist. For example, we plan to analyze the performance of the proposed method in other QoS properties. We also plan to construct a real-world service selection system for mixed mobile networks.

Acknowledgments: This paper is supported by the Zhejiang Provincial Natural Science Foundation (No. LY12F02003), the National Key Technology Support Program (No. 2015BAH17F02), the National Natural Science Foundation of China (No. 61100043), the China Postdoctoral Science Foundation (No. 2013M540492), and the Fundamental Research Funds for Central Universities (JBX171007).

Author Contributions: Yuyu Yin and Yuesheng Xu proposed the main idea and design framework. Wenting Xu and Min Gao designed and performed the experiments. Lifeng Yu and Yujie Pei analyzed the results. Yuesheng Xu and Yuyu Yin carried out the first draft of this paper. Lifeng Yu and Yujie Pei revised the draft and done some write-up.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Choi, J.; Cho, H.; Yi, J.H. Personal Information Leaks with Automatic Login in Mobile Social Network Services. *Entropy* **2015**, *17*, 3947–3962. [CrossRef]
2. Zhang, Y.; Ge, L.; Gao, K. EWnFM: An Environment States Oriented Web Service Non-Functional Property Model. *Entropy* **2015**, *17*, 509–527. [CrossRef]
3. Bichier, M.; Lin, K.J. Service-oriented computing. *Computer* **2005**, *3*, 99–101.
4. Shao, L.; Zhang, J.; Wei, Y. Personalized QoS prediction for web services via collaborative filtering. In Proceedings of the International Conference on Web Services, Salt Lake City, UT, USA, 9–13 July 2007; pp. 439–446.
5. Sun, H.; Zheng, Z.; Chen, J.; Lyu, M. Personalized web service recommendation via normal recovery collaborative filtering. *IEEE Trans. Serv. Comput.* **2013**, *6*, 573–579. [CrossRef]
6. Fan, X.; Hu, Y.; Zhang, R. Context-aware web services recommendation based on user preference. In Proceedings of the Asia-Pacific Services Computing Conference, Fuzhou, China, 4–6 December 2014; pp. 55–61.
7. Chen, M.; Ma, Y.; Hu, B.; Zhang, L.-J. A ranking-oriented hybrid approach to QoS-aware web service recommendation. In Proceedings of the International Conference on Services Computing, New York, NY, USA, 27 June–2 July 2015; pp. 578–585.
8. Jiang, Y.; Liu, J.; Tang, M.; Liu, X. An effective web service recommendation method based on personalized collaborative filtering. In Proceedings of the International Conference on Web Services, Washington, DC, USA, 5–10 July 2011; pp. 211–218.
9. Zheng, Z.; Ma, H.; Lyu, M.R.; King, I. Wsrec: A collaborative filtering based web service recommender system. In Proceedings of the International Conference on Web Services, Los Angeles, CA, USA, 6–10 July 2009; pp. 437–444.
10. Chen, X.; Liu, X.; Huang, Z.; Sun, H. Regionknn: A scalable hybrid collaborative filtering algorithm for personalized web service recommendation. In Proceedings of the International Conference on Web Services, Miami, FL, USA, 5–10 July 2010; pp. 9–16.
11. Yin, Y.; Aihua, S.; Min, G. QoS Prediction for Web Service Recommendation with Network Location-Aware Neighbor Selection. *Int. J. Softw. Eng. Knowl. Eng.* **2016**, *26*, 611–632. [CrossRef]
12. Yao, L.; Sheng, Q.; Segev, Z.; Yu, J. Recommending web services via combining collaborative filtering with content-based features. In Proceedings of the International Conference on Web Services, Santa Clara, CA, USA, 28 June–3 July 2013; pp. 42–49.
13. Zhang, Y.; Zheng, Z.; Lyu, M.R. WSPred: A Time-Aware Personalized QoS Prediction Framework for Web Services. In Proceedings of the IEEE International Symposium on Software Reliability Engineering, Hiroshima, Japan, 29 November–2 December 2011; pp. 210–219.
14. Liu, J.; Tang, M.; Zheng, Z. Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation. *IEEE Trans. Serv. Comput.* **2015**, *9*, 686–699. [CrossRef]
15. Zheng, H.; Yang, J.; Zhao, W. QoS Analysis and Service Selection for Composite Services. In Proceedings of the International Conference on Services Computing, Miami, FL, USA, 5–10 July 2010; pp. 122–129.
16. Wang, W.T.; Wu, Y.L.; Tang, C.Y. Adaptive density-based spatial clustering of applications with noise (DBSCAN) according to data. In Proceedings of the International Conference on Machine Learning and Cybernetics, Guangzhou, China, 12–15 July 2015; pp. 445–451.
17. Schapire, R.E.; Singer, Y. Improved Boosting Algorithms Using Confidence-Rated Predictions. *Mach. Learn.* **1999**, *37*, 297–336. [CrossRef]
18. Wu, C.; Qiu, W.; Wang, X. Time-Aware and Sparsity-Tolerant QoS Prediction Based on Collaborative Filtering. In Proceedings of the International Conference on Web Services, San Francisco, CA, USA, 27 June–2 July 2016; pp. 637–640.
19. Yu, D.; Liu, Y.; Xu, Y.; Yin, Y. Personalized QoS Prediction for Web Services Using Latent Factor Models. In Proceedings of the IEEE International Conference on Services Computing, Anchorage, AK, USA, 27 June–2 July 2014; pp. 107–114.
20. Lee, K.; Park, J.; Baik, J. Location-Based Web Service QoS Prediction via Preference Propagation for Improving Cold Start Problem. In Proceedings of the IEEE International Conference on Web Services, New York City, NY, USA, 27 June–2 July 2015; pp. 177–184.

21. He, P.; Zhu, J.; Zheng, Z.; Xu, J.; Lyu, M.R. Location-based hierarchical matrix factorization for web service recommendation. In Proceedings of the International Conference on Web Services, Anchorage, AK, USA, 27 June–2 July 2014; pp. 107–114; 297–304.
22. Tang, M.; Zheng, Z.; Kang, G. Collaborative Web Service Quality Prediction via Exploiting Matrix Factorization and Network Map. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 126–137. [CrossRef]
23. Xu, Y.; Yin, J.; Lo, W.; Wu, Z. Personalized location-aware QoS prediction for web services using probabilistic matrix factorization. In Proceedings of the International Conference on Web Information Systems Engineering, Nanjing, China, 13–15 October 2013; pp. 229–242.
24. Wei, L.; Yin, J.; Deng, S.; Li, Y.; Wu, Z. Collaborative Web Service QoS Prediction with Location-Based Regularization. In Proceedings of the IEEE International Conference on Web Services, Honolulu, HI, USA, 24–29 June 2012; pp. 464–471.
25. Xie, Q.; Zhao, S.; Zheng, Z.; Zhu, J.; Lyu, M.R. Asymmetric Correlation Regularized Matrix Factorization for Web Service Recommendation. In Proceedings of the IEEE International Conference on Web Services, San Francisco, CA, USA, 27 June–2 July 2016; pp. 204–211.
26. Qi, K.; Hu, H.; Song, W.; Ge, J.; Lv, J. Personalized QoS Prediction via Matrix Factorization Integrated with Neighborhood Information. In Proceedings of the IEEE International Conference on Services Computing, New York, NY, USA, 27 June–2 July 2015; pp. 186–193.
27. Huang, Y.; Huang, J.; Cheng, B.; He, S.; Chen, J. Time-Aware Service Ranking Prediction in the Internet of Things Environment. *Sensors* **2017**, *17*, 974. [CrossRef] [PubMed]
28. Chen, T. A Fuzzy Parallel Processing Scheme for Enhancing the Effectiveness of a Dynamic Just-in-Time Location-Aware Service System. *Entropy* **2014**, *16*, 2001–2022. [CrossRef]
29. Ma, Y.; Wang, S.; Yang, F.; Chang, R.N. Predicting QoS Values via Multi-dimensional QoS Data for Web Service Recommendations. In Proceedings of the IEEE International Conference on Web Services, New York, NY, USA, 27 June–2 July 2015; pp. 249–256.
30. Wu, C.; Qiu, W.; Zheng, Z. QoS Prediction of Web Services Based on Two-Phase K-Means Clustering. In Proceedings of the IEEE International Conference on Web Services, New York, NY, USA, 27 June–2 July 2015; pp. 161–168.