

Research Article

Collaborative QoS Prediction for Mobile Service with Data Filtering and SlopeOne Model

Yuyu Yin,^{1,2,3} Wenting Xu,¹ Yueshen Xu,⁴ He Li,⁴ and Lifeng Yu⁵

¹School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China

²Key Laboratory of Complex Systems Modeling and Simulation of Ministry of Education, Hangzhou, Zhejiang 310027, China

³College of Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, China

⁴School of Software, Xidian University, Xi'an, Shanxi 710071, China

⁵Hithink RoyalFlush Information Network Co., Ltd., Hangzhou, Zhejiang, China

Correspondence should be addressed to Yueshen Xu; ysxu@xidian.edu.cn

Received 25 January 2017; Accepted 21 March 2017; Published 22 June 2017

Academic Editor: Jaegel Yim

Copyright © 2017 Yuyu Yin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The mobile service is a widely used carrier for mobile applications. With the increase of the number of mobile services, for service recommendation and selection, the nonfunctional properties (also known as quality of service, QoS) become increasingly important. However, in many cases, the number of mobile services invoked by a user is quite limited, which leads to the large number of missing QoS values. In recent years, many prediction algorithms, such as algorithms extended from collaborative filtering (CF), are proposed to predict QoS values. However, the ideas of most existing algorithms are borrowed from the recommender system community, not specific for mobile service. In this paper, we first propose a data filtering-extended SlopeOne model (filtering-based CF), which is based on the characteristics of a mobile service and considers the relation with location. Also, using the data filtering technique in FB-CF and matrix factorization (MF), this paper proposes another model FB-MF (filtering-based MF). We also build an ensemble model, which combines the prediction results of FB-CF model and FB-MF model. We conduct sufficient experiments, and the experimental results demonstrate that our models outperform all compared methods and achieve good results in high data sparsity scenario.

1. Introduction

Since many mobile services have been or being developed as the interfaces to access resources on mobile environment, the number of services increases dramatically. Users often have to select a mobile service from a series of service candidates with similar function. To solve the selection issue, people develop the service recommender system to select services with better QoS (short for *quality of service*). But in mobile service invocation, most users only have invoked quite a few services before, and a large part of QoS values are unknown. To solve this problem, it is urgent to find an effective method to predict QoS values, which has been a research highlight in service computing community.

The collaborative filtering (CF for short) algorithm is widely used for QoS prediction [1, 2]. The idea of CF algorithm is to first identify the similar neighbors of a user

or a mobile service and then use the historical QoS values of neighbors to predict the unknown values of the target user or service. It can be seen that the prediction accuracy of CF algorithm largely depends on the identification of similar neighbors. In mobile service recommendation, the accuracy of similar neighbor identification is not so well due to the following reasons:

- (1) In similar neighbor identification, there is an assumption that the QoS values are stable and reliable. However, QoS is largely impacted by the mobile network environment in different locations, both in the user side and service side. Due to the instability of mobile network environment, the QoS value is also unstable.
- (2) Along with the increase of data sparsity, the similarity computation becomes much less accurate. In high

data sparsity, the number of services invoked by a single user is quite limited, which leads to the even few number of common invoked mobile services by more than one user. Especially in the extreme case that two users do not have any services commonly invoked, there is no chance for any two users being the similar neighbor of the other. So it is difficult to conduct similar neighbor identification with high accuracy in sparse QoS records.

- (3) In many cases, we need to select the K most similar neighbors from all neighbor candidates, and the value of K brings a nonnegligible impact on prediction accuracy. The optimal value of K often needs to be determined through a series of experiments and is often different in different datasets.

So we decide to propose new models that can handle the above issues, and our models are based on SlopeOne model. For QoS prediction, the SlopeOne model does not need to identify similar neighbors but directly uses the known QoS records to predict missing values [3]. So the SlopeOne model avoids the issue of similar neighbor identification that happens in CF algorithm. However, the SlopeOne model also has a defect; that is, the model needs to use all of the known QoS records for a missing value prediction. On the one hand, such defect increases the time complexity. On the other hand, it is inevitable to involve noise data, which lowers the prediction accuracy. This paper aims to solve those problems and makes the following contributions:

- (1) It proposes a twofold data filtering strategy to filter noise to improve prediction accuracy and lower time complexity, for predicting the QoS values of mobile services. The proposed data filtering strategy is not designed to any specific QoS property of a mobile service but can be used to predict all types of QoS properties.
- (2) It proposes two novel prediction models. One is an ensemble model, and the other is a matrix factorization model.
- (3) It proposes a linear way to combine the results of the two proposed models, further improving the prediction accuracy.
- (4) It conducts sufficient experiments in two real-world datasets, and the experimental results demonstrate the effectiveness of the proposed models. Note that our models only need the QoS records as the input, without the need for any other side information, which brings high feasibility in mobile service invocation scenario.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 presents the framework of our work. Section 4 explains the proposed filtering methods, and Section 5 elaborates the proposed models. Section 6 gives the experimental results, and Section 7 concludes the paper and discusses the future work.

2. Related Work

It is hard for a user to invoke all available mobile services to acquire all QoS values, to select the most suitable one. Thus, QoS prediction is an indispensable task in mobile service selection and recommendation. The collaborative filtering (CF for short) algorithm is widely used in traditional service computing community to predict QoS [1, 4–7].

The CF algorithm is first formally proposed by [8] and has been broadly employed in e-commerce recommender systems [9, 10]. The CF algorithm can be classified into two types, that is, neighbor-based CF and model-based CF. The neighbor-based CF algorithm can be further classified into two categories, that is, user-based CF algorithm and service-based CF algorithm. We take the following prediction task as the example: to predict the QoS value of user u receiving after invoking service j , marking as q_{uj} . The user-based CF algorithm first identifies the similar neighbors of user u with similarity computation, using the historical QoS records [11–13]. Then, the user-based CF algorithm collaboratively uses the historical QoS records of the identified similar neighbors to service j to compute the predicted QoS value q_{uj} . The service-based CF algorithm is similar to the user-based CF algorithm, and the difference is that the first step is to identify the similar neighbors of service j , and the missing value q_{uj} is predicted by collaboratively using the known QoS records of user u to the identified service neighbors [10, 14, 15].

In recent years, several new neighbor-based CF algorithms have been also proposed for QoS prediction in traditional service computing. Sun et al. [16] proposed a new similarity computation method to better identify user neighbors and service neighbors. In detail, the authors normalized the QoS values and computed the similarity based on Euclidean distance. Liu et al. [17] proposed a geographic location-based CF algorithm. They assumed that the users that are located near each other had similar network environment and thus were likely to experience similar QoS. Zheng et al. [18] constructed an ensemble model, which combined the prediction results of user-based CF algorithm and service-based CF algorithm with a predefined parameter.

Another important type of CF algorithm is model-based algorithm, and the idea is to learn the latent features of a user and a service and further learn the relation between the latent features of users and services. The learning process is based on the historical QoS records. The model-based algorithm includes SVM [19], MF (short for *matrix factorization*) [20, 21], Bayesian classifier [13], and latent semantic analysis [22]. The MF model has been verified to be effective and be the first choice in many prediction tasks. He et al. [23] proposed a geographic location-based hierarchical MF model, in which the user-service invocation matrix is partitioned into several local matrices, with K -Means algorithm. The final prediction result is computed as the combination of the results that are achieved using the whole matrix and local matrices, respectively. Xu et al. [24] extended the PMF (probabilistic matrix factorization) with geographical information. In their model, the similar neighbors were identified based on the geographical distance, and the latent feature vector of the

target user was learned together with the feature vectors of similar neighbors.

Lemire and Maclachlan [3] first proposed the SlopeOne model in recommender system community, which was easy to implement and could achieve good performance. Zhang [25] proposed a hybrid model that was the combination of SlopeOne model and item-based CF algorithm. Correspondingly, Wang and Ye [26] proposed a hybrid model as the combination of SlopeOne model and user-based CF algorithm. Mi and Xu [27] first clustered items according to the ratings that the items received, and, in each cluster, the missing ratings were predicted using SlopeOne model.

In service computing, there are not so many works that study the SlopeOne model. In this paper, we employ SlopeOne model as the base to predict QoS values for mobile services, and our proposed models are verified to be effective by sufficient experiments.

3. The Whole Framework

We present the whole framework of this paper in Figure 1, which includes the following components:

- (1) User-service invocation matrix: it stores the known historical QoS records, and the large part of missing values are to be predicted.
- (2) User similarity matrix: it stores the similarity result of two users, which is computed based on the service invocation records.
- (3) Service similarity matrix: it stores the similarity result of two services, which is computed based on the invoked records.
- (4) Global filter: it identifies the user neighborhood and service neighborhood based on the similarity.
- (5) Local filter: it further identifies a fine-grained neighborhood from the neighborhood that is discovered by the global filter.
- (6) FB-CF (filtering-based CF): it is the proposed multimodel combination method, which is composed of three submodels, and can select a suitable submodel to finish the prediction task in different conditions.
- (7) FB-MF (filtering-based MF): using the prediction results of FB-CF, this component first fills the missing entries in the user-service invocation matrix and then factorizes the matrix using the MF model.
- (8) The ensemble model: it combines the FB-CF model and FB-MF model, to further improve the prediction accuracy.

4. The Proposed Filtering Method

In this section, we present our proposed filtering methods, including global data filtering and local data filtering.

4.1. Global Filtering

4.1.1. The Motivation of Global Filtering. The motivation of global filtering is based on the observation of real-world

service invocation data, and we take the response time as the example to explain. First, let us see Table 1. In Table 1, the task is to predict the QoS value after user₁ invokes service₂. Using the basic SlopeOne model, we can get the prediction result as $x = a + (c - b)$. Now let us see Table 2, in which the prediction result is $x = 7 + (0.5 - 0.7) = 6.8$. However, the prediction result is likely to be biased, and such bias should be avoided for a linear prediction model. The analysis is based on the real-world QoS data collected by [28], and more details of this dataset can be found in the experiment section (see Section 6.1). We give the following detailed analysis.

- (1) As shown in Figures 2(a) and 2(b), the response time data have a strong aggregation characteristic, and the values of most data are distributed around a limited value range. More than 80% values are less than the average, and more than 90% values are located in the range of the average adding double standard deviations. Thus, the prediction value 6.8 is quite likely to be deviated from the real value of x .
- (2) The distribution of QoS values shows clear randomness. Assume that the real value of x is close to the prediction value 6.8; then, the QoS value vectors of user₁ and user₂ should have a stable difference in every dimension. If the deviation of the difference of two QoS value vectors is small, the difference between the two vectors is stable. As shown in Figure 3, we randomly select user A and further select user C. The QoS value vector of user C is similar to that of user A. It can be seen that the QoS value of user A can be smaller or larger than the QoS value of user C, which means there is no stable difference between the QoS values of user A and user C, even though the deviation of difference of user C is the smallest compared to user A. So the prediction value 6.8 is likely to be quite different from the real value of x .

Now let us consider the case in Table 3, in which we can get the prediction value $x = 1 + (0.5 - 0.7) = 0.8$. Based on the aggregation effect shown in Figure 2(a), the probability of x being close to 0.8 is large. So naturally more such local matrices are helpful for linear regression to improve the prediction accuracy.

Based on the above analysis, when the difference among a , b , and c is large (as shown in Table 2), the prediction error is likely to be large. In contrast, when the difference among a , b , and c is small (as shown in Table 3), the prediction error is likely to be small. So in this paper, we use global filtering to enlarge the frequency of the cases like Table 3.

4.1.2. Global Filtering with Similarity Computation. The goal of global filtering is to make the values of a , b , and c close to each other. Some papers claim that the users with close geographical location have similar network environment and thus tend to experience similar QoS [17, 29]. However, in mobile environment, the relation between the network configuration and location is more complex. As shown in Figure 4, we randomly select two users A and B that are close to each other geographically. It can be seen that the

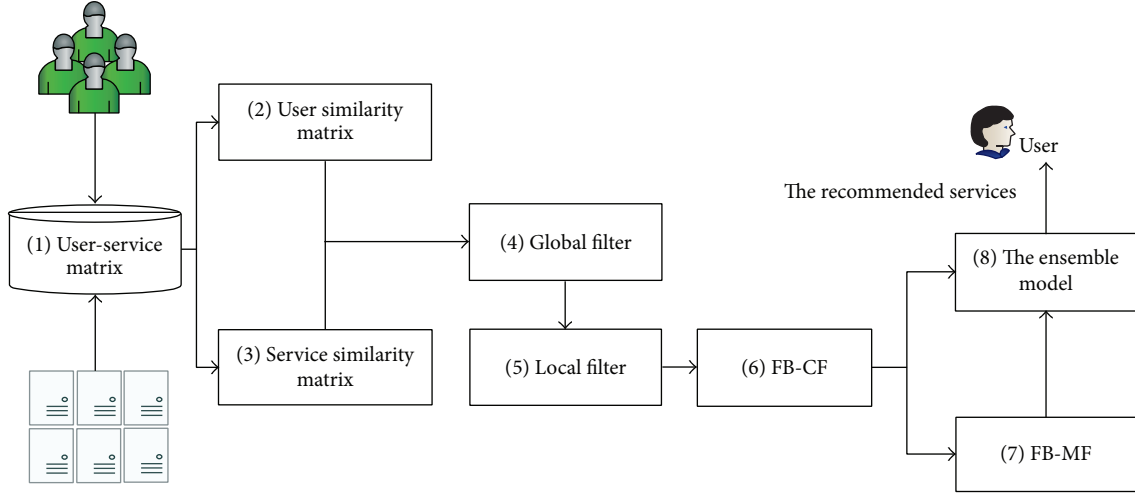


FIGURE 1: The whole framework.

TABLE 1: Example 1.

	service ₁	service ₂
user ₁	a	x
user ₂	b	c

TABLE 2: Example 2.

	service ₁	service ₂
user ₁	$a = 7$	$x = ?$
user ₂	$b = 0.7$	$c = 0.5$

QoS value (response time) of user A can be quite larger than that of user B or be also quite smaller than that of user B. That is, even though two users locate closely, the QoS values that they receive may still be quite different. Besides, if the side information, such as the geographical location, is indispensable for a model, the applicability of the model will be limited. For a model with geographical information as the input, the model will fail to work in the invocation scenario that has no geographical information. In this paper, the proposed models use Manhattan distance as a base to compute the similarity to finish the global filtering. The Manhattan distance is

$$\text{dis}(\vec{x}, \vec{y}) = \sum_{i=1}^n (|x_i - y_i|), \quad (1)$$

where \vec{x} and \vec{y} are the QoS value vectors of two users and n is the number of services that are commonly invoked by the two users. Equation (1) ignores the impact of the number of commonly invoked services, so we use the average Manhattan distance to better compute the similarity of two QoS vectors, which is shown as follows:

$$\text{sim}(\vec{x}, \vec{y}) = \frac{1}{1 + \sum_{i=1}^n (|x_i - y_i|) / n}, \quad (2)$$

where we borrow the idea of Laplacian smoothing in the denominator. In (2), if the QoS vectors are closer, the

similarity of the users will be larger. Note that although we take the user similarity computation as the example to explain, similarly, the service similarity can be also computed in the same way. In mobile service similarity computation, \vec{x} and \vec{y} are the vectors of the invoked records of two mobile services, and n is the number of users that have commonly invoked the mobile services before.

The global filtering is conducted in both the user side and service side and uses a threshold to control the filtering strength. That is, for the target user or target service, the goal is to select the similar neighbors that the corresponding similarity is larger than the threshold. The threshold is not set manually but computed automatically as

$$\tau_{\text{global}} = \frac{\text{avg}(\text{Sim}_{\text{user}}) + \text{avg}(\text{Sim}_{\text{service}})}{2}, \quad (3)$$

where τ_{global} is the threshold, $\text{avg}(\text{Sim}_{\text{user}})$ is the average value of the user similarity matrix, and $\text{avg}(\text{Sim}_{\text{service}})$ is the average value of the mobile service similarity matrix. The automatic computation of the threshold improves the applicability of our method. The experimental results show that, in two real-world datasets, the proposed global filtering achieves good performance.

After global filtering, we get the similar neighbor set for a user or a service. Considering that, under the case of huge data volume, the similar neighbor set can be quite large, to lower the complexity of subsequent computation, we select the K most similar neighbors to form a compact neighbor set. The sensitivity of our proposed models to K will be given in the experiment section.

4.2. Local Filtering. The global filtering is capable of measuring the closeness of QoS vectors, but there may exist huge difference among some local QoS values. Here is an example, where there are two QoS vectors \vec{q}_A and \vec{q}_B :

$$\begin{aligned} \vec{q}_A &= (1, \dots, 1, 10, 2, \dots, 2), \\ \vec{q}_B &= (1, \dots, 1, 3, 2, \dots, 2). \end{aligned} \quad (4)$$

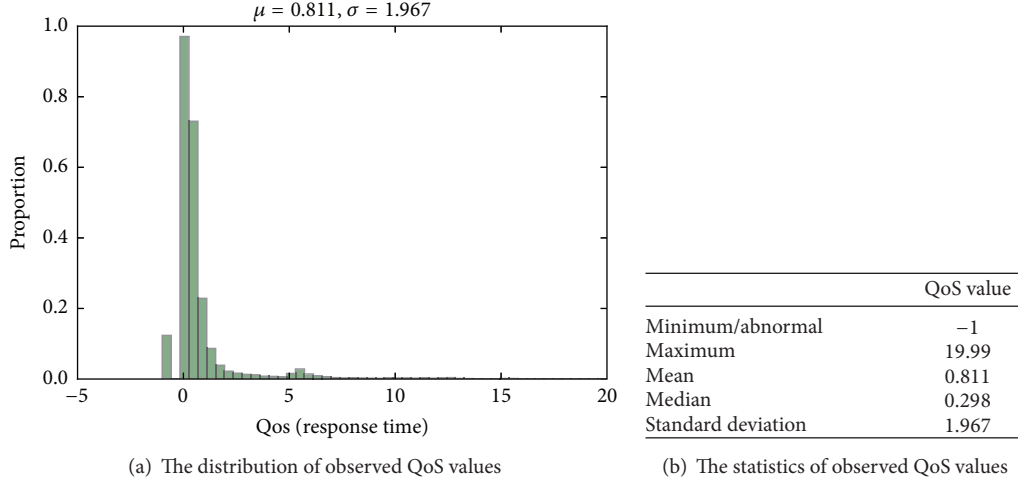


FIGURE 2: Distribution of the number of the observed QoS values.

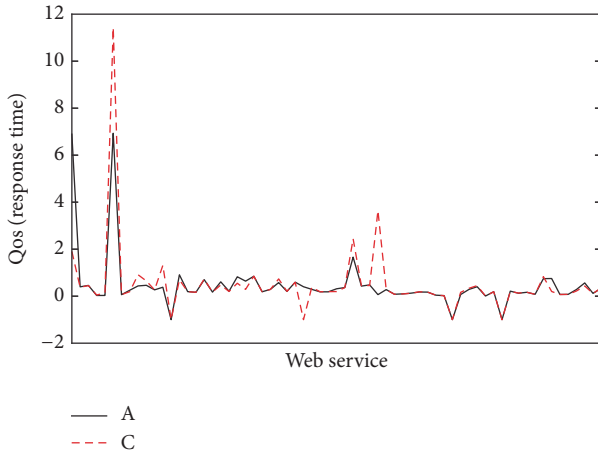


FIGURE 3: The QoS distribution of vectors with small deviation.

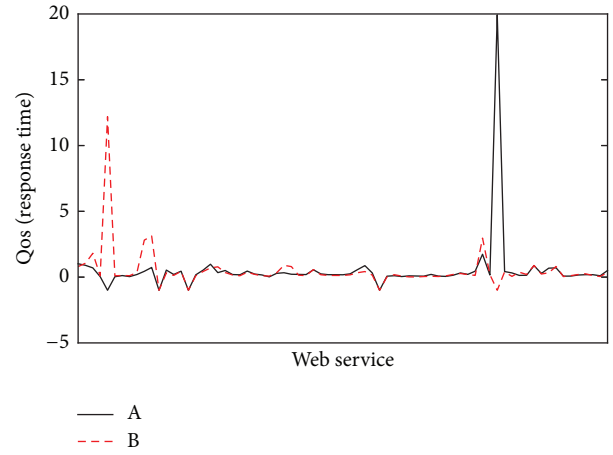


FIGURE 4: The QoS distribution of users in close geographic location.

TABLE 3: Example 3.

	service ₁	service ₂
user ₁	$a = 1$	$x = ?$
user ₂	$b = 0.7$	$c = 0.5$

A and B (A or B could be a user or a service) can receive a quite different QoS value (such as 10 and 3 as shown in the above example), but since the QoS values in other entries are quite similar, overall the similarity should be large. However, in SlopeOne model, using 10 and 3 will lead to large error. So in this paper, we further propose a local filtering method to avoid the above case.

Lemire and Maclachlan [3] proposed the bipolar SlopeOne model, which only uses the data to reach consistency in two-class classification, to be the input of the prediction. This model does the local filtering task to some extent but has the following defects:

- (1) It is hard to decide the classification border: as continuous values, the QoS values are different from

the traditional rating data, which are discrete values. So it is hard to decide the threshold for two-class classification. For example, we set 5 as the threshold, being larger than 5 is positive class, and being smaller than 5 is negative class. In such a case, being 4.9 will be negative class, and being 5.1 will be positive class, but naturally the two values are quite close to each other.

- (2) It is easy to lead to overfiltering: the algorithm requires that, in the local matrix, the classifications of QoS values a , b , and c should be the same. Such strong filtering strategy is likely to lead to too few available data that can be used for prediction, especially in the high data sparsity case.

To solve the above issues, in this paper, we propose a local filtering method based on the dynamic difference classification.

Different from the static two-class classification, which employs a fixed threshold to classify a QoS to one of the two classes, we define that if the difference of two values is smaller

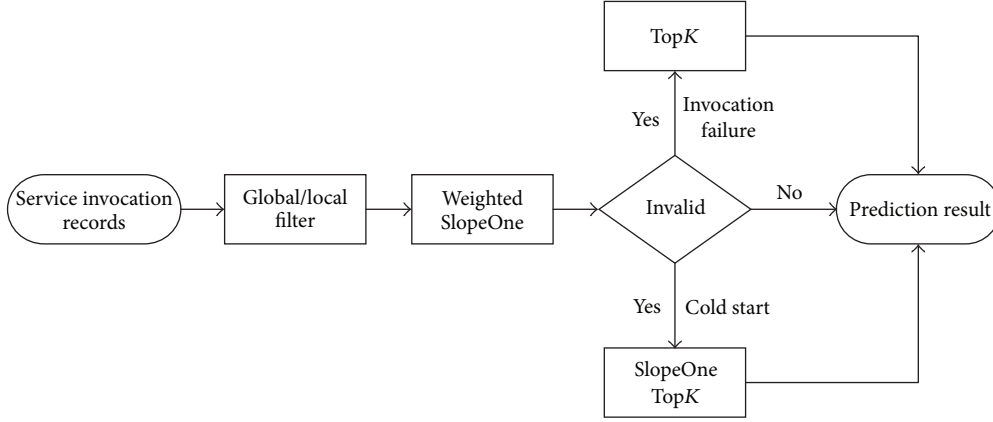


FIGURE 5: The framework of the FB-CF model.

than a threshold, then the class of the two values is the same. That is,

$$\{q_a, q_b \in \text{the same class} \mid |q_a - q_b| < \tau_{\text{local}}\}, \quad (5)$$

where τ_{local} is the classification threshold. Similar to the global filtering threshold τ_{global} , the local filtering threshold τ_{local} in this section does not rely on manual setting either but is computed automatically as the average of all the known QoS values; that is,

$$\tau_{\text{local}} = \text{avg}(\text{dataset}). \quad (6)$$

5. The Proposed Prediction Models

In this section, we will elaborate the proposed prediction models, including three SlopeOne-based models and one MF-based model.

The framework of the proposed model FB-CF (filtering-based CF) is shown in Figure 5. The first step is to use the proposed global filter and local filter to filter the noise data, and the second step is to conduct the prediction using the proposed weighted SlopeOne model (see the following Section 5.1). In the prediction process, if the weighted SlopeOne model finds that the cases of invocation failure or cold-start occur, the framework will turn to the proposed TopK model or SlopeOne TopK model (see the following Sections 5.2 and 5.3). It means that FB-CF model can select a suitable submodel to fit any real invocation case, which further improves the prediction accuracy. We will give detailed explanation of all models in the rest part of this section.

5.1. Data Filtering-Based Weighted SlopeOne. Assuming two vectors $\vec{v} = \{v_i \mid i = 1, 2, \dots, n\}$ and $\vec{w} = \{w_i \mid i = 1, 2, \dots, n\}$, the SlopeOne model uses the linear regression predictor $f(x) = x + b$, $x \in \vec{v}$. If we aim to compute \vec{w} based on \vec{v} , there is only one unknown parameter b . To get the parameter b , we only need to minimize the loss function

$$\min \sum_{i=1}^n (v_i + b - w_i)^2. \quad (7)$$

So the task turns to computing the optimal b . By derivation, we can get $b = \sum_i (w_i - v_i) / n$, which indicates that b is equal to the average deviation of \vec{v} and \vec{w} .

When the SlopeOne model predicts the QoS value of user u invoking mobile service j based on mobile service i , b is the average deviation of the QoS records of service i and service j , and x is the QoS value of user u invoking mobile service i . We can get the following predictor:

$$q_{uj} = q_{ui} + \text{dev}_{ij}, \quad (8)$$

where the average deviation dev_{ij} is computed with

$$\text{dev}_{ij} = \sum_{u \in U_{ij}} \frac{q_{ui} - q_{uj}}{|U_{ij}|}, \quad (9)$$

where U_{ij} represents the user set, in which the users invoke both service i and service j .

5.2. TopK Prediction Model. In the historical invocation records, there exist some QoS values that are not missing but recorded to be negative, which mean that the service invocation fails and the QoS value has not been recorded. In this paper, we also aim to predict the possibility of invocation failure, by studying the possibility of a QoS value being negative. After global and local filtering, if there are some values being negative, which lowers the prediction performance, we use the following TopK model to solve this issue. The user-based TopK model is used to predict the QoS value of user u invoking service j , following the two steps:

- (1) Use TopK algorithm to select the similar neighbor set $N(u)$. This step uses Manhattan distance to compute the similarity to select the K most similar neighbors for user u .
- (2) Predict the missing values based on similar neighbors' historical QoS records. The predictor is

$$q_{uj} \approx \frac{\sum_{v \in N(u)} \text{sim}(u, v) \times q_{vj}}{\sum_{v \in N(u)} \text{sim}(u, v)}, \quad (10)$$

where $\text{sim}(u, v)$ is the similarity of users u and v .

5.3. SlopeOne-TopK Prediction Model. The cold-start problem is a great challenge in QoS prediction, and we propose another model SlopeOne TopK to solve this problem. We take the example of user u invoking mobile service j to explain the following:

- (1) Use TopK algorithm to select the similar neighbor set $N(u)$. This step also uses Manhattan distance to compute the similarity to select the K most similar neighbors for user u . If a neighbor never invoked service j before, we will use the weighted SlopeOne model to first predict the unknown value, to solve the cold-start issue.
- (2) Predict the missing values based on similar neighbors' historical QoS records. The predictor is

$$q_{uj} \approx \frac{\sum_{v \in N(u)} \text{sim}(u, v) \times q_{vj}}{\sum_{v \in N(u)} \text{sim}(u, v)}, \quad (11)$$

where q_{vj} is the QoS value of user v invoking mobile service j . If q_{vj} is unknown, we will first predict q_{vj} using weighted SlopeOne model.

5.4. The Proposed FB-MF Prediction Model. In recent years, the MF model and its extensions are widely used in service recommendation system and have been verified to be effective [16]. In MF model, the user-service matrix $R \in \mathbb{R}^{m \times n}$ is factorized into two low-dimensional matrices $P \in \mathbb{R}^{f \times m}$ and $S \in \mathbb{R}^{f \times n}$, as follows:

$$R = P^T S, \quad (12)$$

where m is the number of users, n is the number of mobile services, and f is the number of latent features. So the missing value of user u invoking service j is shown as follows:

$$q_{uj} \approx P_u \cdot S_j. \quad (13)$$

By minimizing the following loss function, we can get the objective function of MF model:

$$L = \frac{1}{2} \sum_{u=1}^m \sum_{j=1}^n (q_{uj} - P_u \cdot S_j)^2 + \frac{\lambda}{2} (\|P_u\|^2 + \|S_j\|^2), \quad (14)$$

where q_{uj} is the real value of user u invoking service s . We use the regularization terms $\|P_u\|^2 + \|S_j\|^2$ to avoid the overfitting problem. We use the gradient descent algorithm to achieve the local optima of the above loss function; the derivatives are

$$\begin{aligned} \frac{\partial L}{\partial P_u} &= (P_u \cdot S_j - q_{uj}) \cdot S_j + \lambda P_u, \\ \frac{\partial L}{\partial S_j} &= (P_u \cdot S_j - q_{uj}) \cdot P_u + \lambda S_j. \end{aligned} \quad (15)$$

In fact, the user-service matrix R is quite sparse. So in the process of minimizing the loss function, there are many q_{uj} being missing. Such high sparsity seriously impacts

the effectiveness of the model and decreases the prediction accuracy. So we propose a filtering-based MF model (FB-MF for short) to solve the problem.

In FB-MF model, we first use the FB-CF model to finish the prediction task and fill the missing value in user-service matrix R . So in the beginning of FB-MF model, all values are known. Since the prediction result of FB-CF is close to the real value, the existing prediction result can be the base of FB-MF model, to further improve the prediction accuracy.

5.5. The Ensemble Model. Note that the FB-CF model is a local prediction model that uses the filtered local data from the whole QoS records. In contrast, the FB-MF model is a global model that uses the whole QoS records. To further improve the prediction accuracy, we combine the prediction results of the FB-CF model and FB-MF model. We use a parameter to combine the two results linearly, which is shown as follows:

$$q_{uj} \approx \theta \times q_{\text{FB-CF}} + (1 - \theta) \times q_{\text{FB-MF}}. \quad (16)$$

The parameter θ is used to control the weight of two individual models in the final prediction result. If the parameter θ is set to 0, the ensemble model will be degraded to the FB-MF model. If θ is set to 1, the ensemble model will be degraded to the FB-CF model. We name the ensemble model as filtering-based ensemble model (FB-EM for short).

Although, in the current paper, we adopt a static way (θ) to control the weight of the two models, we can see from the experimental results of parameter sensitivity that our model is not sensitive to the value of θ . It indicates that the static setting of θ does not bring much impact on the model performance. We will add the task of dynamic parameter setting into the future work list.

6. Experiment and Evaluation

We conduct sufficient experiments to evaluate the performance of our proposed models, compared to several well-known existing models. The experimental results demonstrate that our models achieve better prediction accuracy and are also not sensitive to the parameters.

6.1. Dataset and Evaluation Metrics. In the experiments, we use a real-world service QoS dataset, WSDream dataset, which is published by [28]. This dataset contains 5825 services and 339 users and contains two types of QoS attributes, that is, response time and throughput. In this paper, we conduct experiments on both response time and throughput records. This dataset has been widely employed to evaluate the prediction accuracy by many researchers [18, 24, 30, 31]. So the experimental results in this paper are convincing.

We use the Mean Absolute Error (MAE) and Normalized Mean Absolute Error (NMAE) to measure the prediction accuracy of our models. The MAE is defined as follows:

$$\text{MAE} = \frac{1}{N} \sum_{u,s} |q_{u,s} - \hat{q}_{u,s}|. \quad (17)$$

TABLE 4: Accuracy comparison (a smaller value means higher accuracy).

Model	Training set density (TD) — response time							
	TD = 5%		TD = 10%		TD = 15%		TD = 20%	
	MAE	NMAE	MAE	NMAE	MAE	NMAE	MAE	NMAE
UserMean	0.8829	1.0879	0.8766	1.0861	0.8785	1.0861	0.8749	1.0849
ItemMean	0.7318	0.9019	0.7121	0.8823	0.7101	0.8779	0.7031	0.8718
SlopeOne	0.7126	0.8757	0.6929	0.8560	0.6886	0.8544	0.6843	0.8492
IPCC	0.6833	0.8422	0.6248	0.7741	0.6037	0.7464	0.5849	0.7253
UPCC	0.6763	0.8335	0.6304	0.7811	0.6172	0.7630	0.6053	0.7506
WSRec	0.6542	0.8063	0.6157	0.7628	0.6026	0.7450	0.5825	0.7223
MF	0.6441	0.7939	0.5405	0.6697	0.5207	0.6438	0.5023	0.6229
FB-CF	0.4991	0.6150	0.4282	0.5305	0.3977	0.4917	0.3788	0.4697
FB-MF	0.4890	0.6027	0.4269	0.5289	0.4019	0.4968	0.3864	0.4791
FB-EM	0.4856	0.5984	0.4195	0.5196	0.3925	0.4852	0.3751	0.4651

The NMAE is defined as follows:

$$\text{NMAE} = \frac{\text{MAE}}{\sum_{u,s} q_{u,s}/N}, \quad (18)$$

where $q_{u,s}$ is the real QoS value in testing set, $\hat{q}_{u,s}$ is the prediction value, and N is the number of QoS values in testing set. A smaller MAE value or a smaller NMAE value means higher prediction accuracy.

6.2. Experiment Setting. In the real-world service invocation, the number of known user-service invocation records is quite limited. To conduct the experiment in a real-world scenario, we randomly select a small part of QoS records from the whole dataset to generate the training set, and the remaining data generate the testing set. In our experiment, we evaluate the prediction accuracy of each model on four different training set densities, that is, 5%, 10%, 15%, and 20%. For example, in the case of training set density being 5%, it means that 5% of the whole data form the training set, while the remaining 95% data are to be predicted. Each set of experiment is conducted for 10 times, and we report the average result. We conduct experiments on both response time and throughput datasets, to give people the confidence that our models can be employed in diverse QoS prediction tasks for mobile service.

In parameter setting, for the FB-CF model, we set the parameter K , including the size of user neighborhood, to be 10 (marked as $K_{\text{user}} = 10$) and the size of service neighborhood, to be 30 (marked as $K_{\text{service}} = 30$). For the FB-MF model, the number of latent factors f is set to be 50, and the regularization parameter λ is set to be 0.01. For the hybrid model, the parameter θ is set to be 0.6. All parameters in the baseline models are set to the same values as in their original papers.

6.3. Performance Comparison. To evaluate the prediction accuracy of our models, we implement several well-known QoS prediction models, as listed below. In those models, UPCC, IPCC, and WSRec are neighborhood-based models, MF is model-based, and SlopeOne is a regression-based model:

- (1) UserMean: the missing QoS value is predicted as the mean of the historical QoS values invoking by the target user.

- (2) ItemMean: the missing QoS value is predicted as the mean of the historical QoS values on the target service invoking by different users.
- (3) UPCC (user-based PCC): UPCC is a user-based collaborative filtering method. This method utilizes the historical QoS records of similar users to predict the missing QoS values in a collaborative way [32].
- (4) IPCC (item-based PCC): IPCC is an item-based collaborative filtering model. This method utilizes the historical QoS records of similar services to predict the missing QoS value [15].
- (5) WSRec: this method is proposed by [18] and linearly combines the prediction results of UPCC and IPCC. WSRec uses a parameter to balance the weighted UPCC and IPCC.
- (6) MF: MF refers to the matrix factorization model and has been explained in Section 5.4.
- (7) SlopeOne: SlopeOne is a linear regression model proposed by [3].

From both Tables 4 and 5, we have the following observations:

- (1) The proposed three models (FB-CF, FB-MF, and FB-EM) all achieve higher prediction accuracy than other baseline models in both datasets and in various density cases. Such an improvement indicates that the proposed filtering strategies, combination model, and the ensemble model are effective. Also, it can be inferred that our proposed filtering strategies and models have high feasibility to different data densities. The reason that the FB-MF model performs better than FB-CF model is as follows:
 - (a) In the initial state of FB-MF model, the sparse user-service matrix is prefilled using the prediction result of FB-CF model. So it can be seen that the prediction procedure of FB-MF model is exactly built on the achieved prediction result. So expectably, the prediction result of FB-MF model should be better than the result of FB-CF model.

TABLE 5: Accuracy comparison (a smaller value means higher accuracy).

Model	Training set density (TD) — throughput							
	TD = 5%		TD = 10%		TD = 15%		TD = 20%	
	MAE	NMAE	MAE	NMAE	MAE	NMAE	MAE	NMAE
UserMean	50.937	1.1729	51.343	1.1684	50.941	1.1676	51.185	1.1639
ItemMean	37.307	0.8597	33.014	0.7508	32.785	0.7490	32.676	0.7421
SlopeOne	31.798	0.7322	31.820	0.7242	31.688	0.7263	31.701	0.7208
IPCC	31.112	0.7164	29.936	0.6813	30.100	0.6899	30.609	0.6960
UPCC	30.829	0.7099	29.054	0.6612	28.357	0.6499	28.114	0.6393
WSRec	29.538	0.6802	28.185	0.6414	27.556	0.6315	27.422	0.6235
MF	58.623	1.3503	30.188	0.6870	24.106	0.5525	22.065	0.5017
FB-CF	24.189	0.5568	19.396	0.4413	17.522	0.4015	16.192	0.3682
FB-MF	21.870	0.5035	18.597	0.4231	17.533	0.4018	16.940	0.3852
FB-EM	21.806	0.5020	18.151	0.4130	16.884	0.3869	16.036	0.3646

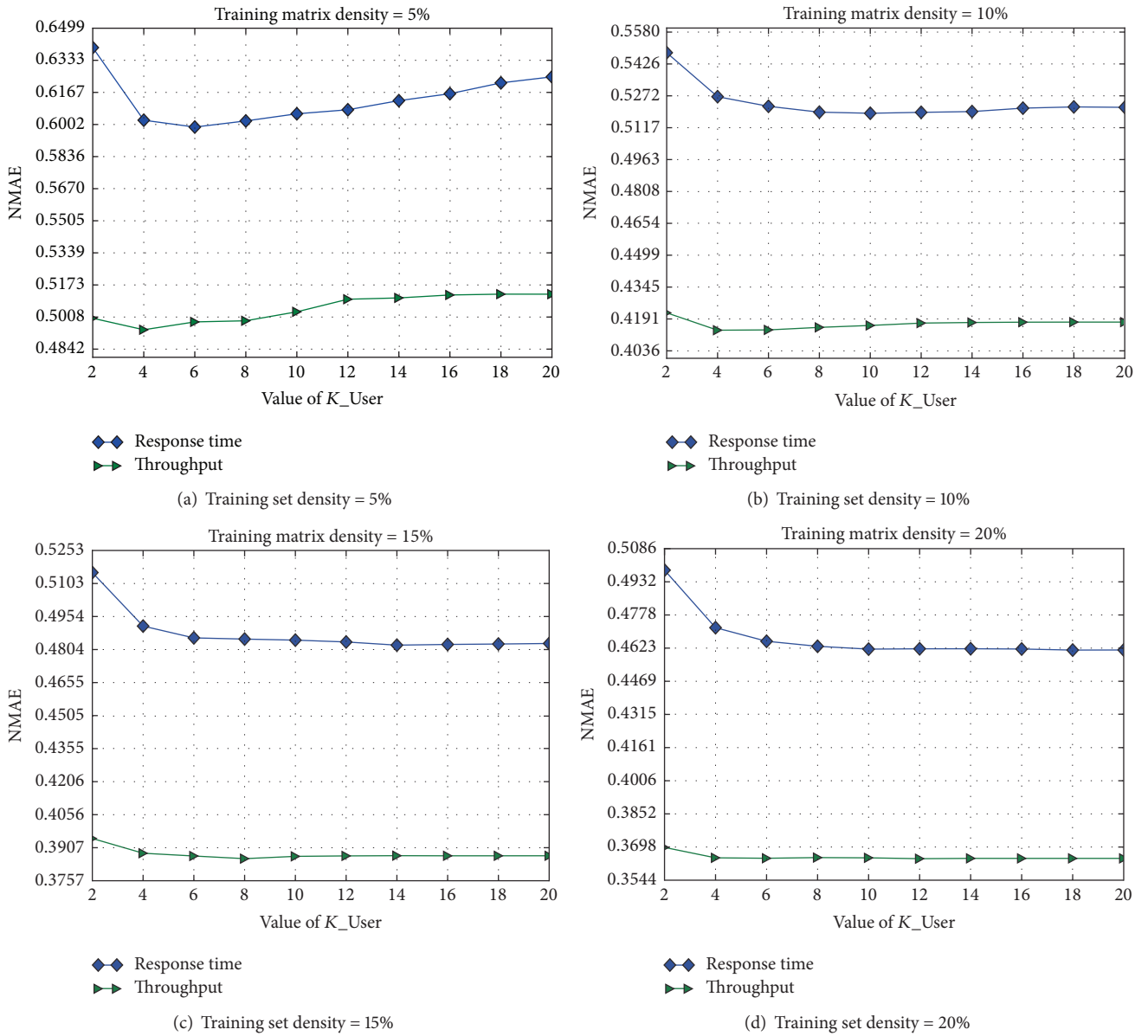
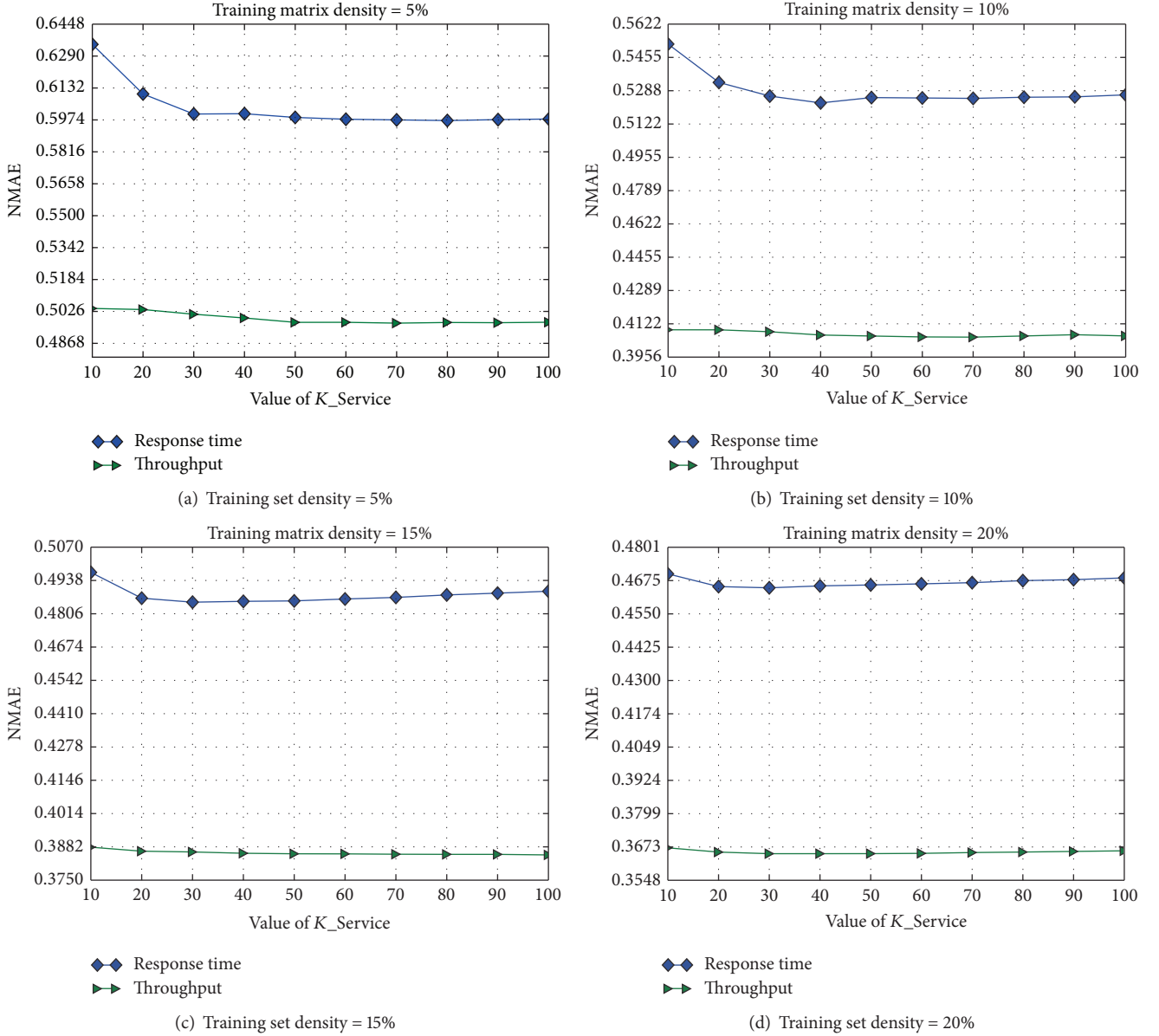


FIGURE 6: Sensitivity to K (user).

FIGURE 7: Sensitivity to K (service).

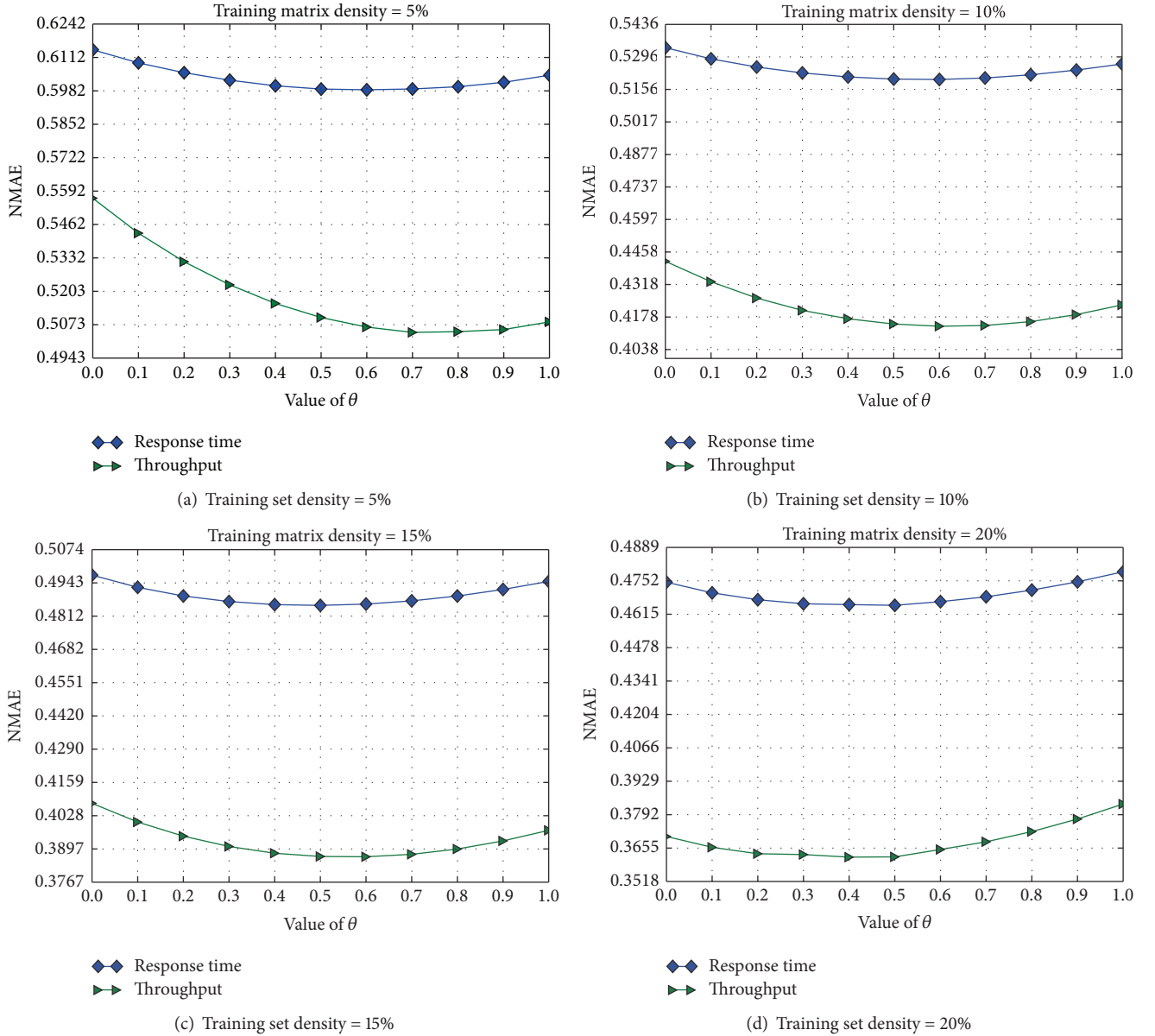
(b) We can notice that, in Tables 4 and 5, the performance of MF model is consistently better than that of collaborative filtering algorithms (e.g., IPCC and UPCC). It indicates that the MF model itself has larger potential to achieve higher prediction accuracy.

- (2) Along with the training set density increasing, MAE and NMAE values decrease. It indicates that more historical invocation records indeed can improve the prediction performance.
- (3) Based on the paired t -tests ($p < 0.001$), the improvements achieved by our three models are all significant.

In the rest part of this section, we will study the sensitivity of our proposed ensemble model FB-EM to the parameters.

6.4. The Sensitivity Analysis of K . In this paper, we use the parameter K to control the number of user or service neighborhood size. Using K lowers the time complexity and saves the time of online prediction. We find that the change trends of MAE and NMAE are quite similar, so we report the result of NMAE here.

The parameter K_{user} controls the number of user neighborhood, and as Figure 6 shows, with the increase of K_{user} , the NMAE value first decreases and then reaches a stable point. At the point of K_{user} being equal to 10, the model achieves the best NMAE value. So we set the default parameter of K_{user} to 10. Note that, in the two datasets of response time and throughput, the change trends of K_{user} and NMAE are quite similar, which illustrates that our model can be used in different prediction tasks.

FIGURE 8: Sensitivity to θ .

The parameter of K_{service} controls the number of service neighborhoods, and as Figure 7 shows, with the increase of K_{service} , the NMAE value first decreases and then also becomes stable at the point of K_{service} being 30, where the model achieves the best NMAE value. So we set the default parameter of K_{service} to 30. Similarly, in the two datasets, the change trends of K_{service} and NMAE are also quite similar.

6.5. The Sensitivity Analysis of θ . The parameter θ is used to balance the weight of two individual models (FB-CF and FB-MF) in the ensemble model. We set the parameter θ in the range of 0 to 1. We report the experimental result in both response time dataset and throughput dataset, in Figure 8.

It can be seen that, in four different training set densities, the optimal value of θ is all in the value of 0.5~0.7. In the whole

range of 0 to 1, the change extent of NMAE value is limited, and in the two datasets, the change trends of NMAE are also quite similar. For one thing, it indicates that our model is not sensitive to the setting of θ . For another thing, our model can be used for multiple QoS prediction tasks.

6.6. The Sensitivity Analysis of Training Set Density. The training set density is the proportion of known mobile service invocation records in the whole dataset. A higher training set density means more information can be used for QoS prediction. To better study the impact of training set density, we conduct comparative experiments on three different values of K_{user} (5, 10, and 15) and three different values of K_{service} (20, 30, and 40). The experimental results

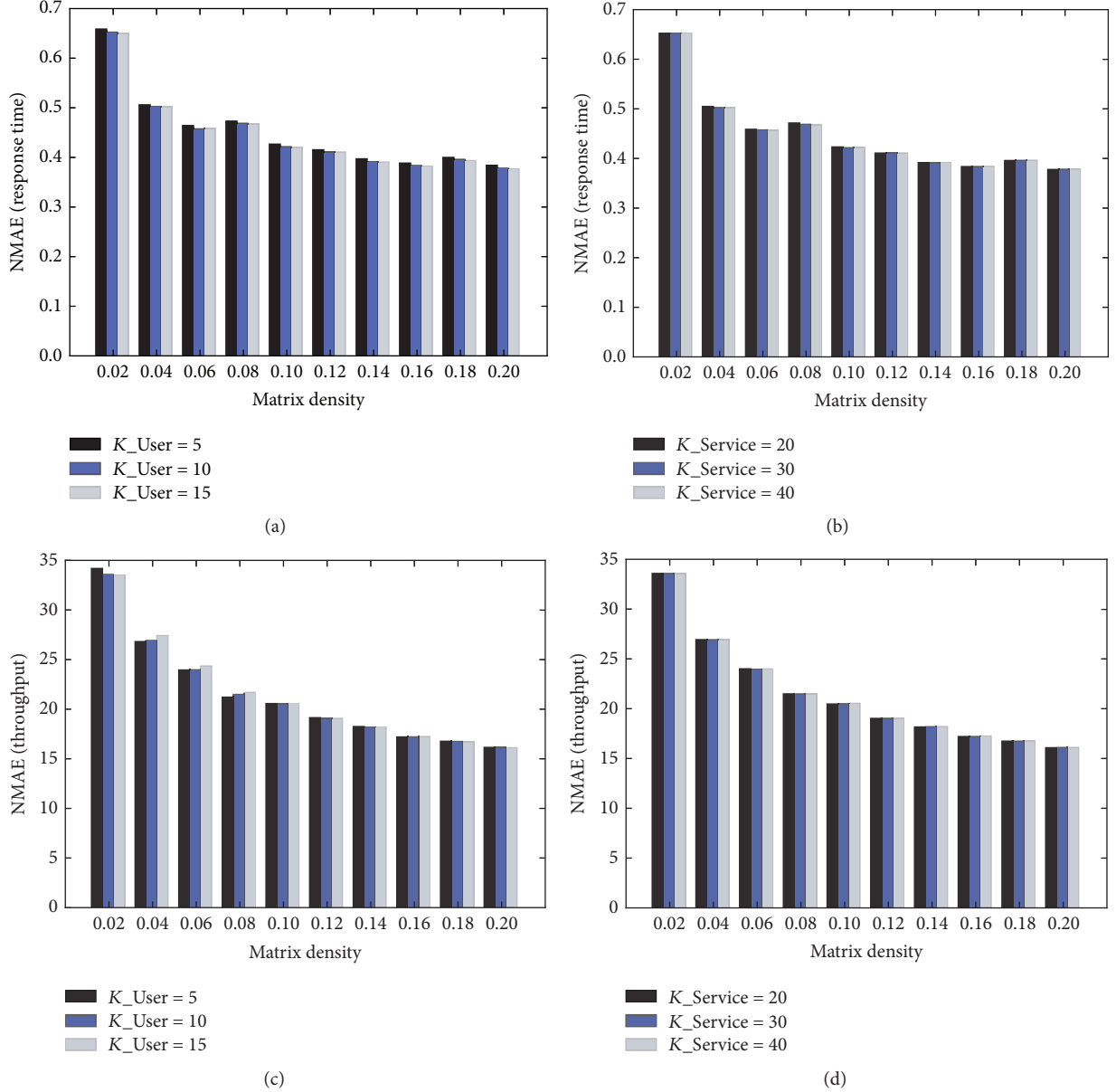


FIGURE 9: Sensitivity to training set density.

are shown in Figure 9, where the density is set to be the value in the range of 2% to 20%.

Figure 9 shows that, with the matrix density increasing, the NMAE value decreases at first. Along with the training set density being larger, the speed of decreasing becomes slower. It means that when there are only limited historical invocation records, the best way to improve prediction accuracy is to collect more QoS data. But when the number of QoS records becomes larger, the key of the prediction task turns to the development of effective models.

7. Conclusion and Future Work

In this paper, we propose two filtering-based models to predict QoS values for mobile services and an ensemble model,

which are FB-CF (filtering-based CF), FB-MF (filtering-based MF), and FB-EM (filtering-based ensemble model). The proposed three models are all based on the proposed filtering methods. The FB-CF model and FB-MF model are extended from SlopeOne model and matrix factorization, respectively. We propose two filtering methods, that is, global filtering and local filtering. The goal of the filtering methods is to filter the noise data that are not suitable for similarity computation. In particular, the FB-CF model and the filtering methods are organized into a unified framework. We conduct sufficient experiments on a real-world dataset, and the experimental results demonstrate the effectiveness of our filtering methods and models.

In the future, we will continue to improve our model from various ways. For example, we plan to use a more flexible

way to combine the two individual models, instead of using a fixed parameter. Second, we also try to improve the filtering methods by investigating more QoS properties of mobile services.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Yueshen Xu and Yuyu Yin contributed equally to this paper and they are co-first authors.

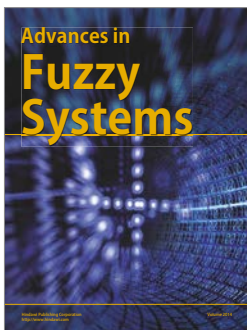
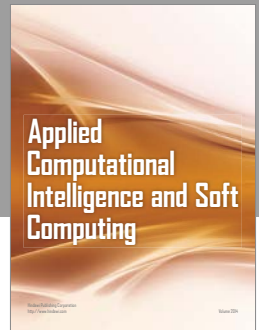
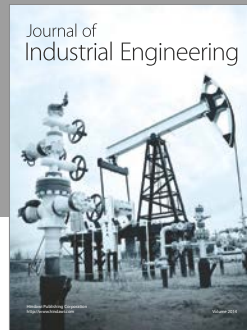
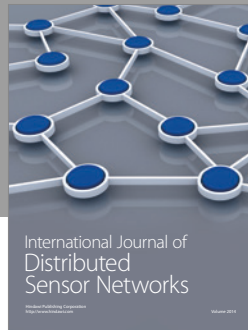
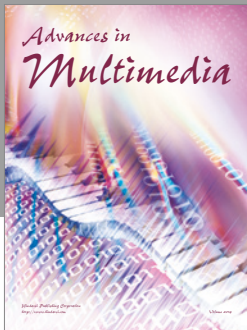
Acknowledgments

This paper is funded by Zhejiang Provincial Natural Science Foundation (no. LY12F02003), China Postdoctoral Science Foundation (no. 2013M540492), the National Key Technology R&D Program (no. 2015BAH17F02), and the National Natural Science Fund of China (nos. 61100043, 61173177).

References

- [1] X. Chen, X. Liu, Z. Huang, and H. Sun, "RegionKNN: a scalable hybrid collaborative filtering algorithm for personalized web service recommendation," in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 9–16, IEEE, July 2010.
- [2] J. Yin, W. Lo, S. Deng, Y. Li, Z. Wu, and N. Xiong, "Colbar: a collaborative location-based regularization framework for QoS prediction," *Information Sciences*, vol. 265, pp. 68–84, 2014.
- [3] D. Lemire and A. Maclachlan, "Slope one predictors for online rating-based collaborative filtering," in *Proceeding of the SIAM International Conference on Data Mining (SDM)*, vol. 5, pp. 1–5, SIAM, 2005.
- [4] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modelling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [5] C. Zhang, L. Zhang, and G. Zhang, "QoS-aware mobile service selection algorithm," *Mobile Information Systems*, vol. 2016, Article ID 4968279, 6 pages, 2016.
- [6] L. Qi, X. Xu, W. D. Dou, J. Yu, Z. Z. Zhou, and X. Zhang, "Time-aware IoE service recommendation on sparse data," *Mobile Information Systems*, vol. 2016, Article ID 4397061, 12 pages, 2016.
- [7] J. Yin, X. Lu, C. Pu, Z. Wu, and H. Chen, "JTangCSB: a cloud service bus for cloud and enterprise application integration," *IEEE Internet Computing*, vol. 19, no. 1, pp. 35–43, 2015.
- [8] E. Rich, "User modeling via stereotypes," *Cognitive Science*, vol. 3, no. 4, pp. 329–354, 1979.
- [9] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *Proceeding of the 16th International World Wide Web Conference (WWW '07)*, pp. 271–280, Alberta, Canada, May 2007.
- [10] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [11] J. L. Herlocker, J. A. Konstan, Al. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 230–237, 1999.
- [12] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.
- [13] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI '98)*, pp. 43–52, Morgan Kaufmann Publishers Inc., 1998.
- [14] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 143–177, 2004.
- [15] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, pp. 285–295, 2001.
- [16] H. Sun, Z. Zheng, J. Chen, and M. R. Lyu, "Personalized web service recommendation via normal recovery collaborative filtering," *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 573–579, 2013.
- [17] J. Liu, M. Tang, Z. Zheng, X. Liu, and S. Lyu, "Location-Aware and Personalized Collaborative Filtering for Web Service Recommendation," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 686–699, 2016.
- [18] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: a collaborative filtering based web service recommender system," in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 437–444, IEEE, July 2009.
- [19] M. Grcar, B. Fortuna, D. Mladenec, and M. Grobelnik, "knn versus svm in the collaborative filtering framework," in *Data Science and Classification*, pp. 251–260, Springer, 2006.
- [20] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain Monte Carlo," in *Proceedings of the 25th International Conference on Machine Learning (ICML '08)*, pp. 880–887, ACM, Helsinki, Finland, July 2008.
- [21] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 713–719, August 2005.
- [22] T. Hofmann, "Collaborative filtering via gaussian probabilistic latent semantic analysis," in *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 259–266, Toronto, Canada, 2003.
- [23] P. He, J. Zhu, Z. Zheng, J. Xu, and M. R. Lyu, "Location-based hierarchical matrix factorization for Web service recommendation," in *Proceedings of the 21st IEEE International Conference on Web Services (ICWS '14)*, pp. 297–304, 2014.
- [24] Y. Xu, J. Yin, W. Lo, and Z. Wu, "Personalized location-aware QoS prediction for web services using probabilistic matrix factorization," in *Proceedings of the Web Information Systems Engineering—(WISE '13)*, Lecture Notes in Computer Science, pp. 229–242, Springer.
- [25] D. Zhang, "An item-based collaborative filtering recommendation algorithm using slope one scheme smoothing," in *Proceedings of the 2nd International Symposium on Electronic Commerce and Security (ISECS '09)*, vol. 2, pp. 215–217, May 2009.
- [26] P. Wang and H. W. Ye, "A personalized recommendation algorithm combining slope one scheme and user based collaborative

- filtering,” in *Proceedings of the International Conference on Industrial and Information Systems, (IIS '09)*, pp. 152–154, April 2009.
- [27] Z. Mi and C. Xu, “A recommendation algorithm combining clustering method and slope one scheme,” in *Proceedings of the International Conference on Intelligent Computing*, pp. 160–167, Springer, 2011.
- [28] Z. Zheng, Y. Zhang, and M. R. Lyu, “Distributed QoS evaluation for real-world Web services,” in *Proceedings of the IEEE 8th International Conference on Web Services (ICWS '10)*, pp. 83–90, Miami, Fla, USA, July 2010.
- [29] W. Lo, J. Yin, S. Deng, Y. Li, and Z. Wu, “Collaborative web service QoS prediction with location-based regularization,” in *Proceedings of the IEEE 19th International Conference on Web Services (ICWS '12)*, pp. 464–471, Honolulu, Hawaii, USA, June 2012.
- [30] D. Yu, Y. Liu, Y. Xu, and Y. Yin, “Personalized QoS prediction for web services using latent factor models,” in *Proceedings of the 11th IEEE International Conference on Services Computing (SCC '14)*, pp. 107–114, July 2014.
- [31] Q. Yu, Z. Zheng, and H. Wang, “Trace norm regularized matrix factorization for service recommendation,” in *Proceedings of the IEEE 20th International Conference on Web Services, (ICWS '13)*, pp. 34–41, July 2013.
- [32] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: an open architecture for collaborative filtering of netnews,” in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pp. 175–186, Chapel Hill, NC, USA, 1994.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

