Received 7 May 2013; revised 21 August 2013; accepted 18 September 2013. Date of publication 23 September 2013; date of current version 30 July 2014.

Digital Object Identifier 10.1109/TETC.2013.2283233

Cold-Start Recommendation Using Bi-Clustering and Fusion for Large-Scale Social Recommender Systems

DAQIANG ZHANG[1,2], CHING-HSIEN HSU[3], MIN CHEN[4], QUAN CHEN[5], NAIXUE XIONG[6], AND JAIME LLORET[7]

[1]School of Software Engineering, Tongji University, Shanghai 200092, China
[2]Department of Computer Science, Nanjing Normal University, Jiangsu 210097, China
[3]Department of Computer Science, Chung Hua University, Hsinchu 300-12, Taiwan
[4]Department of Computer Science, Huazhong University of Science and Technology, Wuhan 430074, China
[5]Department of Computer Science, Shanghai Jiao Tong University, Shanghai 201101, China
[6]School of Computer Science, Colorado Technical University, Colorado Springs, CO 80907-3896 USA
[7]Department of Communications, Polytechnic University of Valencia, Valencia 46022, Spain

CORRESPONDING AUTHOR: M. CHEN (minchen@ieee.org)

This work was supported by the National Natural Science Foundation of China (Grant No. 61103185 and 61300224), the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province, China (Grant No. 11KJB520009), and the 9th Six Talents Peak Project of Jiangsu Province (Grant No. DZXX-043).

ABSTRACT Social recommender systems leverage collaborative filtering (CF) to serve users with content that is of potential interesting to active users. A wide spectrum of CF schemes has been proposed. However, most of them cannot deal with the cold-start problem that denotes a situation that social media sites fail to draw recommendation for new items, users or both. In addition, they regard that all ratings equally contribute to the social media recommendation. This supposition is against the fact that low-level ratings contribute little to suggesting items that are likely to be of interest of users. To this end, we propose bi-clustering and fusion (BiFu)-a newly-fashioned scheme for the cold-start problem based on the BiFu techniques under a cloud computing setting. To identify the rating sources for recommendation, it introduces the concepts of popular items and frequent raters. To reduce the dimensionality of the rating matrix, BiFu leverages the bi-clustering technique. To overcome the data sparsity and rating diversity, it employs the smoothing and fusion technique. Finally, BiFu recommends social media contents from both item and user clusters. Experimental results show that BiFu significantly alleviates the cold-start problem in terms of accuracy and scalability.

INDEX TERMS Cold-start problem, collaborative filtering, bi-clustering, smoothing, fusion.

I. INTRODUCTION

Social media sites, e.g., IMDB, MovieLens and Delicious, have achieved widespread success over the past five years. Millions of users are active daily in these sites. They upload media, make comment and share information within social circles [1]. To further promote marketing and maintain user loyalty, many social media sites, together with various Applications for Internet of Things employ Social Recommender Systems (SRSs) [2] [3] to serve users with the most attractive services. Collaborative Filtering (CF) is a fundamental mechanism to SRSs, which automatically aggregates user profiles and finds user preference from the large-scale SRSs [4]. CF has been adopted in social medial sites such as Facebook [5], Amazon [6] and Google [7].

In order to achieve high levels of prediction accuracy or runtime scalability, various CF schemes have been proposed for SRSs [8]–[13]. However, most of them suffer from the cold-start problem. The problem represents a situation that social media sites are incapable of making recommendation for new items, new users or both due to few rating elements available in the matrices (data sparsity is higher than 85%) [14]–[16]. In the cold-start setting, these schemes have trouble in effectively locating similar items and like-minded users [17]. As a consequence, the problem remarkably undermines the SRSs efficiency. Moreover, most CF schemes implicitly assume that all ratings contribute the same to recommendation.

To this end, we propose a cloud-based CF scheme using Bi-clustering and Fusion (BiFu), which takes the trivial ratings, bi-clustering and fusion into consideration. We implement a social recommender system and provide accurate social recommendation as a cloud service. Thus, social medial site can easily enjoy the accurate recommendation. To tackle the cold-start problem through constructing a dense area in the item-user rating matrix, BASA firstly identifies the popular items and their associated ratings. Then, it allocates them into the upper left corner of the matrix. To alleviate the influence of data sparsity, BiFu takes advantage of the bi-clustering technique to aggregate similar items and like-minded users simultaneously. It goes further by smoothing ratings within every user cluster. To recommend products for active users, BiFu predicts unrated items from like-minded user clusters and similar item clusters. Our experiments demonstrate the superiority of BiFu scheme in terms of recommendation accuracy and scalability. The main contributions of this work are three-fold.

- BiFu proposes the concept of trivial ratings for SRSs that identifies user dislike items. It filters out the trivial ratings in the item-user matrix, which considerably reduces the dimensionality of the item-user matrix and improves the recommendation accuracy.
- BiFu introduces the bi-clustering and fusing techniques for handling the cold-start problem, alleviating the influence of data sparsity to SRSs. It simultaneously clusters items and user profiles. Then it smoothes ratings within every user cluster to overcome the diversity of user rating styles. Finally it fuses the recommendation from both item and user clusters.
- We implement a social recommender system and provide it as a cloud service.

The remaining of this paper is organized as follows. Section II introduces the related work. Section III provides the background of our work. Section IV introduces BiFu in detail. Section V reports the experiment settings and results. Section VI concludes our work with future directions.

II. RELATED WORK

In this section, we briefly review the existing works regarding the cold-start problem in SRSs. We also discuss the techniques used in the proposed scheme.

A. COLLABORATIVE FILTERING WITH THE COLD-START PROBLEM

The cold-start problem has not been well tackled so far, despite the fact that the vast majority of CF research has been dedicated to SRSs over the past two decades in attracting users and improving their loyalty. As a result, these works usually achieve low recommendation accuracy as well as poor scalability under the cold-start setting [14], [18], [19].

It is intuitive that incorporating external information, e.g., content information, may help CF bridge the gap between the clues of existing and new items or users, thus enabling CF to solve the notorious problem. The newly-added information can involve various aspects of items and users. Taking a movie as an example, the information may be about actors, directors, composers, genres and singers, or the information of age, gender, job attributes and nationality of audiences. In [15], Zhou et al. integrated user initial review and matrix factorization for prediction. In [19], Schein et al. derived a generative probabilistic model, which augmented the raw movie ratings with actor and director information. In [14], Victor et al. proposed a trust-based CF scheme, which incorporated a trust network among all users in SRSs into the raw ratings. By identifying and leveraging key features on the trust network, this scheme achieved a high level of recommendation accuracy. All these kinds of schemes, however, are seriously restricted by three drawbacks. First, they assume that external information should be ready to be included, which may not hold due to diverse uncertainties and constraints [20], [21]. Second, it incurs heavy cost to supplement profiles of a number of items and users. Finally, the addition of external information increases the dimensionality of the item-user matrix, thus slowing down the speed of operation. These drawbacks do not exist in hybrid systems that combine CF and the content-based filtering schemes.

Another solution is to inject pseudo-items or pseudo-users, namely filterbots, into the item-user rating matrix, and then apply the conventional CF schemes. With these pseudo-items, pseudo-users and corresponding ratings, existing schemes [22] significantly alleviate the sparsity of the item-user matrix. Nevertheless, these schemes do not captured the diversity of the items or users, hence they share the same shortcomings held in conventional CF schemes.

In general, incorporating semantics of items and users into CF will alleviate the influence of the cold-start problem [18]. These kinds of CF schemes are built on top of two conditions. One is that users are able to correctly express their interest or requirements regarding item intrinsic features. The other is that the ratings are tightly correlated with user behaviors. These conditions, however, cannot be satisfied in reality.

B. COLLABORATIVE FILTERING WITH THE BI-CLUSTERING TECHNIQUE

The majority of CF schemes mainly employ one-dimension clustering to cluster items or users individually. However, the one-dimension clustering technique usually ignores the useful information in the opposite dimension. To this end, Bi-clustering technique simultaneously clusters both item dimension and user dimension in the item-user matrix. Correspondingly, the Bi-clustering technique performs better than one-way cluster technique to deal with sparse and high-dimensional recommendation matrices [23]–[26].

To summarize, the cold-start problem is prevalent in SRSs. It still remains open to CF schemes that do not append any other external information.

C. RECOMMENDER SYSTEMS BASED ON CLOUD COMPUTING

Cloud computing is increasing popular in the past several years. A cloud platform often provides the services according to three main models: *Platform-as-a-Service* (PaaS), *Infrastructure-as-a-Service* (IaaS) and *Software-as-a-Service* (SaaS) [27].

Because cloud computing can greatly improve the capacity and reliability of calculation and reduce costs of infrastructure and maintenance, it is a good choice to implement recommender systems under the cloud computing environment. In fact, researchers have proposed several recommender systems based on the SaaS model. In [28], Lee et. al proposed a personalized DTV program recommendation system to refine the channel selecting processes and satisfy the consumers' requirements. CPRS [29] is proposed to recommend programs to the consumers for digital TV platforms. In [30], Jiang et. al proposed a blog personality recommender system to recommend high quality and personalized blogs for different readers based on cloud computing infrastructure. To easily maintain end-users, [31] shows a case of cloud client virtualizaiton, which can be used in cloud services.

III. PRELIMINARY

In this section, we introduce the underlying mechanisms of memory-based CF and bi-clustering technique involved in the proposed scheme. Notations in the design of BiFu are listed in Table 1.

TABLE 1. **Notations in the design of BiFu.**

Notation	Explanation
$\|I\|$	the number of items in the item-user matrix
$\|U\|$	the number of users in the item-user matrix
I_u	the set of items rated by the user u
U_i	the set of users who have rated the item i
$r_{u,i}$	the rating that the user u rates the item i
$\overline{r_i}$	the average rating of the item i
$\overline{r_u}$	the average rating of the user u
$P_{u,i}$	the predicted rating on the item i by the user u
S_i	the set of the item i's similar items
S_u	the set of the user u's like-minded users
$\underline{y}$	the low bound of the rating on a given item
$\overline{\overline{y}}$	the upper bound of the rating on a given item

A. MEMORY-BASED CF

It draws recommendation by identifying similar items or like-minded users over the entire item-user matrix. They often achieve high levels of accuracy, but are unable to scale up [32]. Generally, memory-based CF can be classified into item-based, user-based and IU-based categories according to the rating sources, as shown in Figure 1. In this subsection, we discuss the item-based and user-based CF before going into the introduction to IU-based CF.

1) ITEM-BASED CF

It exploits the idea that similar items may be preferred by the same user. Eq. 1 illustrates how an item-based CF predicts an

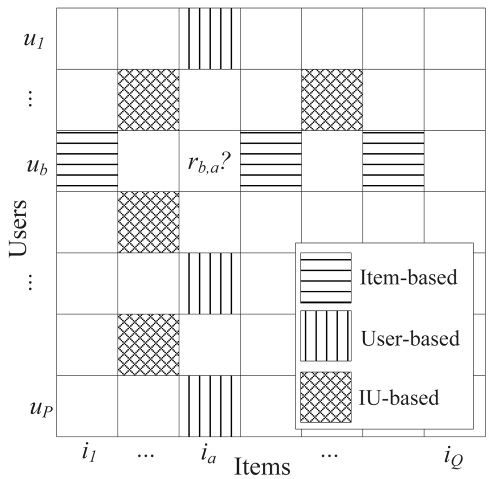


FIGURE 1. **Memory-based CF involves item-based, user-based and IU-based CF. It exploits a phenomenon that like-minded users may go for the same item and the same user may like similar items.**

active user u likes the active item i.

$$P_{u,i} = \overline{r_i} + \frac{\sum_{j\in I_u} sim(i,j)\cdot(r_{u,j}-\overline{r_j})}{\sum_{j\in I_u} sim(i,j)} \tag{1}$$

$$sim(i,j) = \frac{\sum_u (r_{u,i}-\overline{r_i})\cdot(r_{u,j}-\overline{r_j})}{\sqrt{\sum_u (r_{u,i}-\overline{r_i})^2}\cdot\sqrt{\sum_u (r_{u,j}-\overline{r_j})^2}}. \tag{2}$$

The similarity of items can be measured by Pearson Correlation Coefficient (PCC) and Vector Space Similarity (VSS) algorithms. As described in [33], PCC often achieves better performance than VSS. Consequently, BiFu selects PCC to measure the item similarity. Eq. 2 shows the PCC-based similarity, where $u \in U_i \cap U_j$.

2) USER-BASED CF

It shares the similar motivation as item-based CF that like-minded users may go for the same item. Eq. 3 shows the underlying mechanism of user-based CF, where $sim(u, v)$ is the similarity of users u and v.

$$P_{u,i} = \overline{r_u} + \frac{\sum_{v\in U_i} sim(u,v)\cdot(r_{v,i}-\overline{r_v})}{\sum_{v\in U_i} sim(u,v)} \tag{3}$$

User similarity is often calculated by PCC or VSS algorithms [34]. In view of widespread success of PCC in user-based CF, BiFu chooses it as the similarity measurement between users.

3) IU-BASED CF

Both item-based and user-based CF neglect the ratings that like-minded users made on similar items. These ratings reflect the preference of like-minded users, which may provide insights in inferring potential interest of active users. A simple example of IU-based CF is illustrated in [8].

B. BI-CLUSTERING TECHNIQUE

In SRSs, bi-clustering technique (also referred to as two-mode clustering) aims at simultaneously clustering items into k disjoint clusters and users into l clusters [35], [36]. Then we get two clusters, i.e., $IC : \{IC_1, IC_2, \cdots, IC_\alpha, \cdots, IC_k\}$ and $UC : \{UC_1, UC_2, \cdots, UC_\beta, \cdots, UC_l\}$, where IC_α and UC_β are an item cluster and a user cluster, respectively. Thus, the bi-clustering is defined as a tuple (IC, UC). Note that the partitions of items and users are independently, and their combination relies on the entire item-user matrix.

IV. BIFU: A CLOUD-BASED COLD-START RECOMMENDATION SCHEME USING BI-CLUSTERING AND FUSION

In this section, we propose **BiFu:** — A cloud-based Cold-start Recommendation Scheme, which exploits the rating confidence level and bi-clustering techniques. Algorithm 1 illustrates the three phases of BiFu scheme — filtering, bi-clustering, and prediction.

Algorithm 1 BiFu Scheme.

```
procedure BIFU
    Input: An item-user matrix IU
    Output: Recommendation for requests
    procedure FILTERING PHASE
        Filtering trivial ratings in IU
        Removing empty profiles in IU
        Constructing a dense area
    end procedure
    procedure BI-CLUSTERING PHASE
        Bi-clustering
        Smoothing ratings within every item and user
cluster, respectively
    end procedure
    procedure PREDICTION PHASE
        Predicting unrated items for requests
        Making recommendation
    end procedure
end procedure
```

In the filtering phase, BiFu filters trivial ratings in the item-user matrix by introducing the rating confidence level. Then it removes empty items and user profiles, and aggregates popular items and frequent raters to the upper left corner of the item-user matrix. In the bi-clustering phase, BiFu clusters items and users into item clusters and user clusters simultaneously. Within every user cluster, BiFu presents a smoothing strategy to eliminate the diversity of user rating styles. In the prediction phase, BiFu fuses recommendation from both item and user clusters and then makes suggestions. Note that the filtering and bi-clustering phases are accomplished in the offline phase, which hides the heavy computation overhead from disclosing to users. What follows is detailed introduction to BiFu scheme.

A. FILTERING PHASE

This phase aims at preprocessing ratings for the bi-clustering phase, and reducing the dimensionality of the item-user matrix. It involves filtering trivial ratings, removing empty profiles and constructing a dense area.

1) FILTERING TRIVIAL RATINGS

The ratings on a specific item are usually distributed around an average rating. We assume that the ratings follow Gaussian distribution. We argue that a rating less than the average attitude provides discriminative information that the item is unfavorable to a certain user. Such ratings are defined as trivial ratings, which negatively contribute to recommendation. Let μ, σ^2 and r be the mean, variance and the rating of a concrete item, the Probability Density Function (PDF) of the rating r is $f(r; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(r-\mu)^2}{2\sigma^2}}$. Its Cumulative Distribution Function (CDF) can be computed as an integral of the PDF. Its PDF is defined in Eq. 4, where $\underline{y}$ is the lower bound of the rating. The Quantile Function (QF) $F^{-1}(p; \mu, \sigma^2)$ is the inverse function of $F(r; \mu, \sigma^2)$, where p is the belief.

$$F(r; \mu, \sigma^2) = \int_{\underline{y}}^{r} f(t; \mu, \sigma^2)dt = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{\underline{y}}^{r} e^{\frac{-(t-\mu)^2}{2\sigma^2}} dt \quad (4)$$

Definition 1: Rating confidence level α_x is a probability of a rating falls into the interval $[x, \bar{y}]$, where $\underline{y} \le x \le \bar{y}$ and $\bar{y}$ is the upper bound of the rating.

Theorem 1: Given a rating confidence level α_x and PDF $f(r; \mu, \sigma^2)$, then $\alpha_x = \int_x^{\bar{y}} f(r; \mu, \sigma^2)dr$.

Proof: Straightforward. ■

Corollary 1: Given a rating confidence level α_x and CDF $F(r; \mu, \sigma^2)$, the rating confidence interval of α_x is $[x, \bar{y}]$, where $x = F^{-1}(1 - \alpha_x; \mu, \sigma^2)$.

The rating confidence level is the value of $\int_x^{\bar{y}} f(r; \mu, \sigma^2)dr$ in most cases. Usually, the value of $\bar{y}$ is a fixed value (*e.g., 5 in 5-point scale*) in SRSs.

Figure 2 illustrates an example that the rating confidence interval under the given rating confidence level and the rating scale. Once we get the rating confidence interval, we are able to filter trivial ratings that are defined in Definition 2.

Definition 2: Given a rating confidence level α_x, trivial ratings are ratings in the item-user matrix that fall out of the interval $[x, \bar{y}]$.

Theorem 2: Given a concrete item, during the rating aggregation phase, if a rating on the item is r, and $|r - u| \ge k\delta$, the probability that the rating happens is at least $(1 - \frac{1}{k^2})$, where $k > 1$.

Proof: Directly derived from Chebyshev's inequality. ■

As a matter of fact, we have Chebyshev's inequality shown in 2 to ensure the correctness of the rating selection. Given the antagonistical influence of trivial ratings, BiFu employs a function to filter them. Specially, it will check whether every rating of an item is less than x. In this way, BiFu dramatically

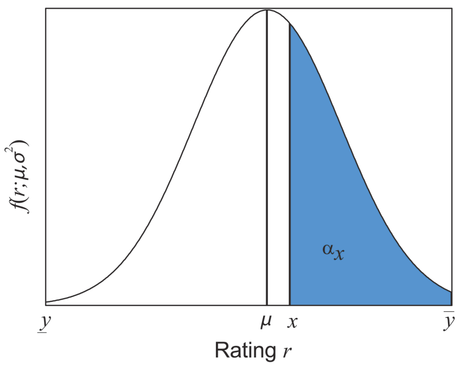


FIGURE 2. Rating confidence level in Gaussian Distribution. Given an item, BiFu dramatically reduces the number of its candidate ratings by selecting the ratings between $(x, \overline{y}]$.

reduces the dimensionality of the item-user matrix. Note that BiFu merely marks trivial ratings as non-rated ratings, thus generating many new empty profiles in the matrix that will be removed in the following step.

2) REMOVING EMPTY PROFILES

Consider that a user makes negative contribution to recommendation when he/she does not rate any items or rate items as low scores. For the same reason, an item is useless for any CF scheme if it has not been rated yet or is lowly rated. These empty item and user profiles bring nothing to SRSs but a large amount of meaningless computation overhead. Therefore, it is always of wisdom to remove them. Note that we do not remove these empty columns and rows from the original item-user matrix, but shift them to the end of the item-user matrix.

Figure 3 illustrates the results of removing empty profiles. BiFu not only simplifies its successive operations, but also gets a dense area that is the basis for the next step.

3) CONSTRUCTING A DENSE AREA

In this step, BiFu introduces the concepts of popular items and frequent raters to aggregate ratings. It identifies and aggregates them to the upper left of the item-user matrix. Popular items are items that are frequently rated by users, and frequent raters are users that rate more items than others. Given an active item i and an active user u, Eq. 5 defines their rating density, where ρ_i is the rating density of the item i and ρ_u is the rating density of the user u. Thus, popular items and frequent raters are defined as Definitions 3 and 4.

$$\rho_i = |U_i|/|U|, \, \rho_u = |I_u|/|I| \tag{5}$$

Definition 3: Popular items are items whose rating density ρ_i is bigger than the threshold $\widehat{\rho_i}$: $\rho_i > \widehat{\rho_i}$, where $\widehat{\rho_i}$ is the average rating density of all items.

Definition 4: Frequent raters are users whose rating density ρ_u is bigger than the threshold $\widehat{\rho_u}$: $\rho_u > \widehat{\rho_u}$, where $\widehat{\rho_u}$ is the average rating density of all users.

By specifying the values of the thresholds $\widehat{\rho_i}$ and $\widehat{\rho_u}$, BiFu easily identifies popular items and frequent raters. At the same time, it aggregates them to the upper left of the item-user matrix, which finally forms a dense area. Figure 4 plots the original rating distribution of a certain part of Netflix dataset. Figure 4 shows the same part of the matrix after preprocessing popular items and frequent raters. With such a dense rating distribution, BiFu is ready for the bi-clustering phase.

Note that BiFu presents an aggregation algorithm that can shift popular items and frequent raters in a linear time. Suppose BiFu plans to get top ς popular items, it randomly chooses the first ς items from the upper left of the item-user matrix and sorts them by heap–sort algorithm. Then, it checks the rest popular items by maintaining the heap structure. Thus, in the worst case, the complexity of shifting popular items to the upper left of the matrix is $O(\varsigma \cdot log(\varsigma) + (n - \varsigma) \cdot \log(\varsigma))$, where n is the number of items that are identified as popular items.

B. BI-CLUSTERING PHASE

To alleviate the influence of data sparsity, BiFu uses the bi-clustering technique to simultaneously cluster similar items and like-minded users. Specifically, it includes two steps in bi-clustering – clustering and smoothing ratings within every user cluster.

BiFu clusters both dimensions of the item-user matrix into item clusters and user clusters by K-means algorithm. After clustering, it divides $|I|$ items into N_{IC} clusters $\{IC_1, IC_2, \ldots, IC_{N_{IC}}\}$ by PCC similarity. Meanwhile, it classifies $|U|$ users into N_{UC} clusters $\{UC_1, UC_2, \ldots, UC_{N_{UC}}\}$ also by PCC similarity. Detail information of using the co-cluster technique for collaborative filtering can refer to [26] and [35].

In every user cluster, we observe two significant phenomena. One is that *many items are not rated owing to the data sparsity in the cold-start settings*. The other is *the diversity of user rating style* that some users tend to express the same level of favorite degree as higher scores than other users. This kind of rating diversity reflects biased arbitrariness.These few ratings or biased arbitrariness negatively decrease the prediction accuracy. Consequently, BiFu presents a smoothing strategy for every user clusters. To be specific, the missing values in each cluster are smoothed by the Eq. 6, where UC_u is the user cluster that includes the user u.

$$r_{u,i} = \overline{r_u} + \sum\nolimits_{u' \in UC_u} \frac{r_{u',i} - \overline{r_{u'}}}{|UC_u|} \tag{6}$$

For the same reason, BiFu leverages the smooth technique for every item cluster to eliminate the data sparsity and rating diversity.

C. PREDICTION PHASE

This phase aims at drawing recommendation for active users in an online manner. To predict the rating of the active user u_a may rate the active item i_b, BiFu firstly predicts the rating by looking for similar items of the item i_b from item clusters

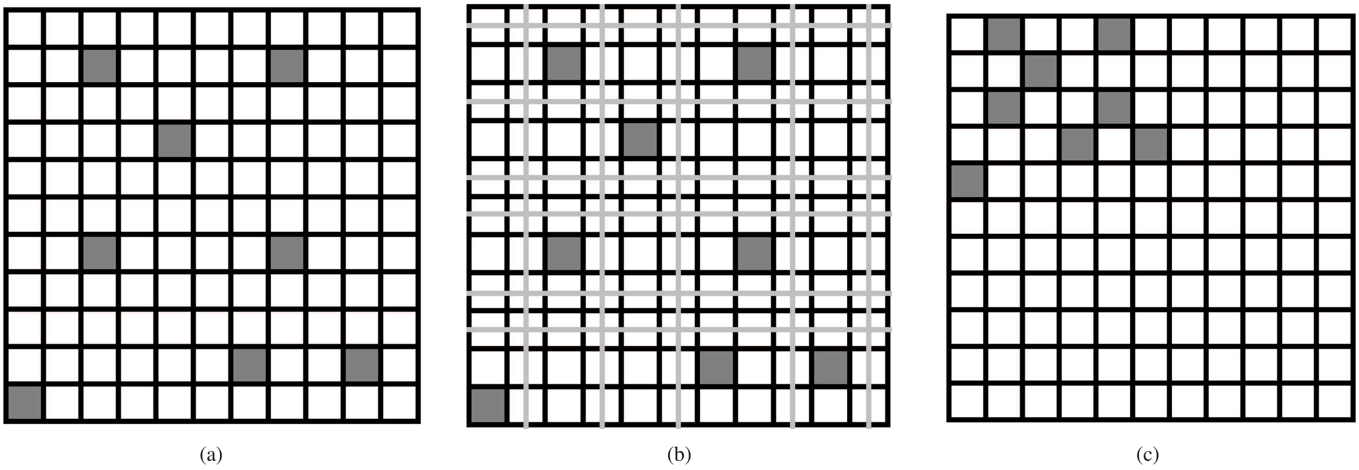


FIGURE 3. BiFu removes empty profiles for the item-user matrix under the cold-start setting. (a) The user rating distribution in the item-user matrix is sparse. (b) Empty profiles of items and users are identified and then cleared. (c) A new user rating distribution is constructed after BiFu removes empty profiles.

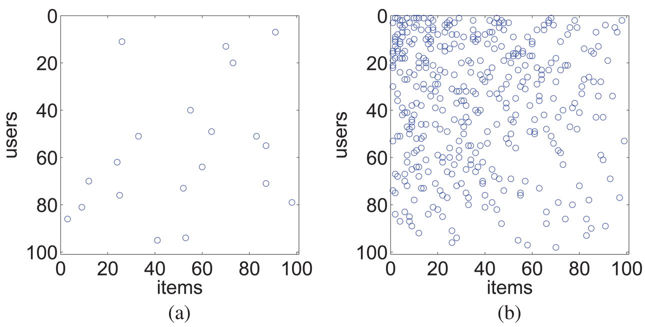


FIGURE 4. BiFu constructs a dense area in Netflix dataset. (a) A snapshot of part of original rating distribution in the item-user matrix. (b) A snapshot of rating distribution after BiFu shifts the popular items and the frequent raters.

and makes the item-based prediction. Then, it predicts the rating by seeking like-minded users of the user u_a from user clusters and draws the user-based recommendation. Finally, BiFu gives the rating P_{u_a,i_b} by fusing these two predictions together.

Note that original and smoothed ratings affect recommendation differently. To distinguish them, BiFu introduces a parameter λ, whose value is between 0 and 1. The weight $w_{u,i}$ is $1-\lambda$ when the user u has rated the item i; otherwise is λ.

To predict the rating for the requested item i_b, BiFu extracts i_b's several nearest similar items from the most similar item clusters that better embody item features. The similarity between the item i_b and an item cluster IC_α is defined as Eq. 7, where U_{IC_α} is the set of users that have rated items in the cluster IC_α, and $u \in U_{i_b} \cap U_{IC_\alpha}$. Then, it extracts a certain number of most similar items from several most nearest item clusters. Usually, BiFu sets the number of most similar items bigger than the number of items in the most nearest cluster. Thus, the ratings extracted may cover user preference as much as possible. The similarity between the item i_b and another item i is defined as Eq. 8, which incorporates the smoothing parameter λ to distinguish different ratings (*e.g., original and smoothed ratings*).

$$sim(i_b, IC_\alpha) = \frac{\sum_u (r_{u,i_b} - \overline{r_{i_b}})(\sum_{i' \in IC_\alpha} \frac{r_{u,i'} - \overline{r_{i'}}}{|IC_\alpha|})}{\sqrt{\sum_u (r_{u,i_b} - \overline{r_{i_b}})^2}\sqrt{\sum_u (\sum_{i' \in IC_\alpha} \frac{r_{u,i'} - \overline{r_{i'}}}{|IC_\alpha|})^2}} \quad (7)$$

$$sim(i_b, i) = \frac{\sum_{u \in U_{i_b}} w_{u,i}(r_{u,i} - \overline{r_i})(r_{u,i_b} - \overline{r_{i_b}})}{\sqrt{\sum_{u \in U_{i_b}} w_{u,i}^2 (r_{u,i} - \overline{r_i})^2}\sqrt{\sum_{u \in U_{i_b}} (r_{u,i_b} - \overline{r_{i_b}})^2}}. \quad (8)$$

Meanwhile, BiFu chooses u_a's top several like-minded users from the most similar user clusters that best match his or her preference. The similarity between u_a and a user cluster UC_β is given in Eq. 9, where I_{UC_β} is the set of items rated in the cluster UC_β, $i \in I_{u_a} \cap I_{UC_\beta}$. Given a specific user, BiFu selects its several most nearest user clusters and then extracts a certain number of most like-minded users as the source of user-based prediction. The similarity between u_a and another user u is given in Eq. 10.

$$sim(u_a, UC_\beta) = \frac{\sum_i (r_{u_a,i} - \overline{r_{u_a}}) \cdot (\sum_{u' \in UC_\beta} \frac{r_{u',i} - \overline{r_{u'}}}{|UC_\beta|})}{\sqrt{\sum_i (r_{u_a,i} - \overline{r_{u_a}})^2} \cdot \sqrt{\sum_i (\sum_{u' \in UC_\beta} \frac{r_{u',i} - \overline{r_{u'}}}{|UC_\beta|})^2}} \quad (9)$$

$$sim(u_a, u) = \frac{\sum_{i \in I_{u_a}} w_{u,i} \cdot (r_{u,i} - \overline{r_u}) \cdot (r_{u_a,i} - \overline{r_{u_a}})}{\sqrt{\sum_{i \in I_{u_a}} w_{u,i}^2 \cdot (r_{u,i} - \overline{r_u})^2} \cdot \sqrt{\sum_{i \in I_{u_a}} (r_{u_a,i} - \overline{r_{u_a}})^2}} \quad (10)$$

$$P_{u_a,i_b} = (\overline{r_{i_b}} + \frac{\sum\limits_{i\in S_{i_b}} w_{u,i}\cdot sim(i_b,i)\cdot(r_{u,i}-\overline{r_i})}{\sum\limits_{i\in S_{i_b}} w_{u,i}\cdot sim(i_b,i)})\cdot\gamma + (\overline{r_{u_a}} + \frac{\sum\limits_{u\in S_{u_a}} w_{u,i}\cdot sim(u_a,u)\cdot(r_{u,i}-\overline{r_u})}{\sum\limits_{u\in S_{u_a}} w_{u,i}\cdot sim(u_a,u)})\cdot(1-\gamma) \quad (11)$$

BiFu involves two kinds of recommendation mechanisms – item-based and user-based mechanisms. It introduces a parameter γ to incorporate them. The incorporation for the rating of the user u_a on the item i_b is given as Eq. 12.

Thus far, we have discussed BiFu that is characterized by introducing the rating confidence level and bi-clustering technique. What follows is our empirical study.

D. IMPLEMENTATION

We implement BASA as a social recommender system at Apache Hadoop platform (http://hadoop.apache.org/) and provide the service in the SaaS model.

To use the cloud service, a social site can simply upload its item-user matrix to the cloud server through web browser. Once the cloud server receives a recommendation request, the server launches a virtual machine to work on this new request and associates the virtual machine with the request. In the virtual machine, a software that implements the BiFu scheme in Algorithm 1 is then started to work on the item-user matrix. Once the software finish the processing, the virtual machine returns the results for recommendation to the cloud server. After that, the cloud server stores the recommendation results, notifies the social site that the recommendation results are ready and destroys the correspondence virtual machine to free the computation resources.

V. EXPERIMENTS

In order to evaluate the proposed scheme, we carried out a series of experiments. Particularly, we try to answer the following questions:

- What is the overall performance of the proposed scheme?
- How do the two fundamental problems of CF, *i.e., data sparsity and scalability*, affect the performance of BiFu?
- How do the parameters influence BiFu? It involves several parameters in its design such as the fusion parameter γ and the smoothing parameter λ.

A. EXPERIMENTAL SETTINGS

We mainly compare our BiFu approach against:

- Average-based CF scheme (AVG), which draws prediction by injecting average filterbots in the item-user matrix [22].
- Item-based CF scheme (IB), which makes recommendation by item similarity.
- User-based CF scheme (UB), which draws suggestions by user similarity.
- SCBPCC, which is a scalable CF scheme using cluster-based smoothing [33].

TABLE 2. Statistical features of MovieLens and Netflix datasets.

Features	MovieLens (ML)	Netflix (NF)
Number of users	71,567	2,649,429
Number of items	10,681	17,770
Number of ratings	10,000,054	100,480,507
Density of data	1.31%	0.21%
Avg. #of rated items/user	139.7	37.92
Avg. #of rated users/Item	936.2	5654.5
Avg. rating globally	3.5124	3.6044

We conducted our experiments over a couple of well-known social media datasets about movies — MovieLens (http://www.grouplens.org/) and Netflix (http://www.netflixprize.com/). The statistical features of these datasets are summarized in Table 2. The lower bounds for identifying trivial ratings (i.e., x in Definition 2) in MovieLens and Netflix were set as 3.1 and 3.7, respectively. All experiments were conducted on a cluster of Ubuntu 64-bit OS that involves 4 computers. Every computer is with 8 GB RAM and Intel Xeon E5405 2.00 GHz dual CPUs.

To thoroughly evaluate the proposed scheme, we designed three types of experiments to study the cold-start problem (*i.e., new items, new users or both*) — horizontal comparison (HC), vertical comparison (VC) and bidirectional comparison (BC), as illustrated in Figure 5.

- In HC experiment, we changed the size of the training set by selecting the first 15%, 30%, 45%, 60%, 75% item profiles of MovieLens and Netflix datasets named as ML_15, ML_30, ML_45, ML_60 and ML_75, and NF_15, NF_30, NF_45, NF_60 and NF_75, respectively. We selected the last 20% columns (new items) of the original item-user matrix as the test set.
- In VC experiment, we used the same scheme as HC to choose the training set, but we extracted user profiles instead of item profiles. The last 20% rows (new users) are selected as the test set.
- In BC experiment, we also shared the same scheme as in HC and VC did to select the training set, but we simultaneously extracted item and user profiles, *e.g., extracting the first 15% item profiles and 15% user profiles together*. We selected the last 20% item and user profiles of the original item-user matrix as the test set.

B. EVALUATION METRICS

In order to keep consistent with most previous research, we select two metrics to measure the recommendation accuracy. One is Mean Absolute Error (MAE), defined in Eq. 12, where T is the test set, and $|T|$ is the size of the test set.

The other metric is Root Mean Squared Error (RMSE), given in Eq. 13, where T is the same meaning as its

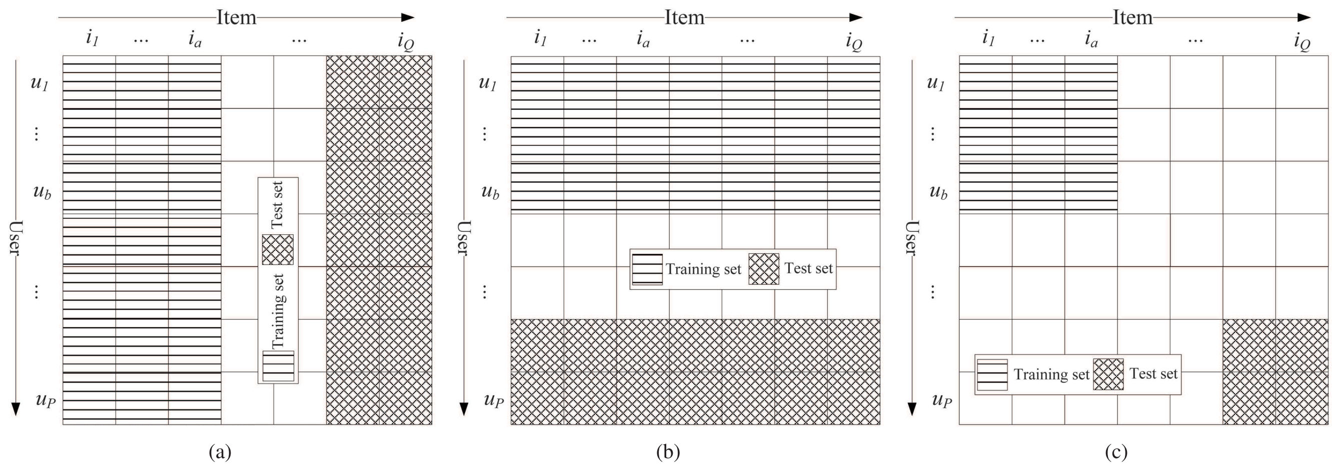


FIGURE 5. Three types of experiments for the cold-start problem – horizontal, vertical and bidirectional comparison, designed for new items, users and both, respectively. (a) Horizontal comparison. (b) Vertical comparison. (c) Bidirectional comparison.

TABLE 3. MAE/RMSE on MovieLens.

Type	Scheme	ML_15	ML_30	ML_45	ML_60	ML_75
HC	AVG	0.8934/1.2314	0.7957/1.0525	0.7704/1.0065	0.7642/0.9903	0.7582/0.9793
	IB	0.7272/0.9307	0.7270/0.9304	0.7274/0.9309	0.7275/0.9311	0.7273/0.9308
	SCB	0.7155/0.9397	0.6522/0.8593	0.6168/0.8187	0.6172/0.8128	0.6065/0.8001
	BiFu	0.7002/0.9128	0.6319/0.8498	0.5879/0.7814	0.5780/0.7614	0.5682/0.7571
VC	AVG	0.7665/0.9966	0.7568/0.9774	0.7521/0.9698	0.7496/0.9662	0.7484/0.9642
	UB	0.7155/0.9880	0.7151/0.9876	0.7153/0.9878	0.7149/0.9874	0.7150/0.9874
	SCB	0.5712/0.7566	0.5619/0.7429	0.5516/0.7302	0.5515/0.7309	0.5512/0.7333
	BiFu	0.5693/0.7385	0.5502/0.7293	0.5377/0.7164	0.5269/0.7011	0.5255/0.7014
BC	AVG	0.7694/0.9839	0.7399/0.9427	0.7345/0.9355	0.7324/0.9324	0.7299/0.9284
	IB	0.7355/0.9417	0.7355/0.9416	0.7357/0.9419	0.7359/0.9421	0.7354/0.9415
	UB	0.6936/0.9194	0.6933/0.9191	0.6937/0.9195	0.6938/0.9195	0.6939/0.9196
	SCB	0.6181/0.8080	0.6136/0.8052	0.6073/0.7972	0.6106/0.8019	0.6086/0.7989
	BiFu	0.5827/0.7666	0.5703/0.7510	0.5648/0.7390	0.5587/0.7218	0.5510/0.7202

counterpart in MAE. The smaller the MAE and RMSE are, the better the recommendation accuracy.

$$MAE = \sum\nolimits_{u \in T} |P_{u,i} - r_{u,i}| / |T| \tag{12}$$

$$RMSE = \sqrt{\sum\nolimits_{u \in T} (P_{u,i} - r_{u,i})^2 / |T|}. \tag{13}$$

C. PERFORMANCE

BiFu can alleviate two CF fundamental problems – data sparsity and scalability under the cold-start setting. In this section, we conducted experiments to evaluate the proposed scheme.

1) RECOMMENDATION ACCURACY

The cold-start problem involves three aspects: new items, new users or both. Therefore, we conducted HC, VC and BC experiments over MovieLens and Netflix datasets correspondingly.

Tables 3 and 4 illustrate the recommendation accuracy over MovieLens and Netflix datasets. With the increase of the training set from 15% to 75% on two datasets, the number of the ratings in the training set increase rapidly, leading to the result that all CF schemes achieved the lower MAE and RMSE over all three types of experiments. Among these schemes, BiFu achieved the lowest MAE and RMSE, indicating that it outperforms other CF schemes in recommendation accuracy. This is owing to two reasons. One is BiFu filters trivial ratings and marks them as non-rated ratings. The other is it removes a large number of empty profiles, including the original empty profiles and new empty profiles produced in the filtering step.

All CF schemes cannot get better prediction accuracy in Netflix dataset than MovieLens dataset. For example, the MAEs/RMSEs of IB in HC experiment over ML_15 and NF_15 datasets are 0.7272/0.9307 and 0.7963/0.9909. This is because the data sparsity in Netflix is less than that of MovieLens dataset. Compared with AVG, IB, UB and SCB, BiFu has the minimal variation in its recommendation accuracy, which shows that BiFu is more robust than the others in handling the data sparsity. Moreover, BiFu also embodies an attractive characteristic that it achieves high levels of accuracy even if in the small training set. Thus, we may train BiFu over a medium size subset of the original item-user matrix, which saves lots of training time.

TABLE 4. MAE/RMSE on Netflix.

Type	Scheme	NF_15	NF_30	NF_45	NF_60	NF_75
HC	AVG	0.8718/1.1520	0.8289/1.0622	0.8169/1.0362	0.8123/1.0273	0.8070/1.0168
	IB	0.7963/0.9909	0.7963/0.9909	0.7963/0.9909	0.7965/0.9911	0.7964/0.9911
	SCB	0.7418/0.9617	0.7101/0.9174	0.7011/0.9085	0.7001/0.9052	0.6985/0.9036
	BiFu	0.7409/0.9571	0.7000/0.9018	0.6869/0.8843	0.6538/0.8733	0.6277/0.8545
VC	AVG	0.8130/1.0189	0.8118/1.0165	0.8111/1.0156	0.8110/1.0153	0.8111/1.0152
	UB	0.7564/1.0210	0.7563/1.0208	0.7561/1.0205	0.7558/1.0198	0.7552/1.0192
	SCB	0.6682/0.8815	0.6640/0.8796	0.6618/0.8612	0.6506/0.8574	0.6446/0.8484
	BiFu	0.6457/0.8699	0.6360/0.8621	0.6296/0.8445	0.6272/0.8376	0.6128/0.8318
BC	AVG	0.8028/1.0060	0.7878/0.9786	0.7838/0.9701	0.7820/0.9672	0.7803/0.9642
	IB	0.8023/0.9973	0.8023/0.9973	0.8025/0.9975	0.8027/0.9978	0.8026/0.9976
	UB	0.7570/0.9809	0.7576/0.9814	0.7574/0.9813	0.7573/0.9812	0.7574/0.9813
	SCB	0.6772/0.8743	0.6644/0.8692	0.6694/0.8676	0.6559/0.8538	0.6492/0.8435
	BiFu	0.6502/0.8561	0.6417/0.8512	0.6435/0.8534	0.6422/0.8475	0.6362/0.8421

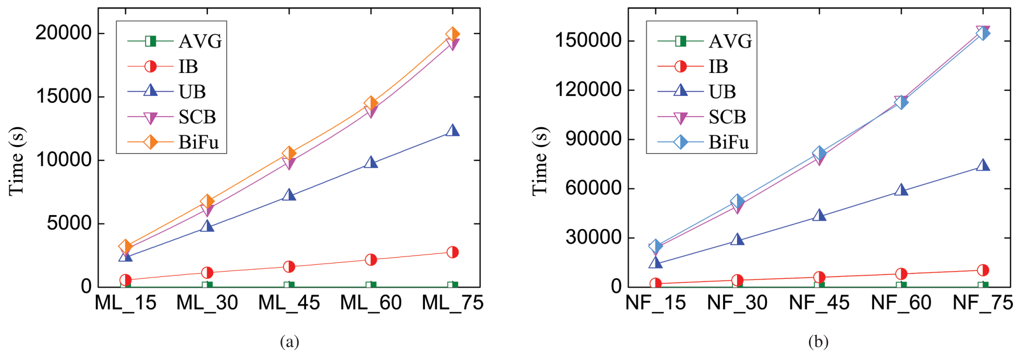


FIGURE 6. Scalability of BiFu in the offline phase. (a) Scalability of BiFu over MovieLens dataset. (b) Scalability of BiFu scheme over Netflix dataset.

In summary, BiFu outperforms the other schemes in recommending products or services. This is attributed to two reasons. Firstly, BiFu employs the bi-clustering technique to locate the most related items and the most like-minded users as the recommendation sources. Secondly, it fuses the ratings for recommendation from related items and like-minded users after filtering out the trivial ratings on top of the proposed concepts — popular items and frequent raters.

2) SCALABILITY

It has an impact on the performance of BiFu scheme that makes recommendation in large-scale SRSs. In the online phase, all CF schemes can respond to request in a few seconds. In the offline phase, BiFu distinguishes itself from the existing CF schemes in making preparations for personalized recommendation. It incorporates the bi-clustering technique that is a computation-intensive process. It also incurs some computation overhead in filtering trivial ratings. Therefore, in this section we conducted experiments to evaluate the scalability of BiFu in the offline phase by varying the training set over MovieLens and Netflix datasets. We changed the training set in the same manner as that in accuracy experiments. We also ran the IB, UB, AVG and SCB schemes as the baselines. Note that in our experiments, we selected the running time as the metric to measure scalability.

Figure 6 shows the response time of BiFu scheme in the offline phase. As the testset grows, the response time of BiFu increased in a linear fashion, indicating that BiFu is scalable. The maximum response time for ML_75 with 100% percentage of the testset is about 20,000 seconds, among which the bi-clustering consumes almost a half time to cluster the large-scale rating matrices (particularly in Netflix dataset).

Note that the non-personalized CF schemes (*e.g., AVG*) ran fast, but they made poor recommendation without personalization. In comparison with the SCB scheme that achieved high scalability, BiFu is a little bit slow, but it significantly improves the recommendation accuracy. This is because BiFu introduces some computation-intensive operations in the offline phase. Generally, the recommendation accuracy rather than scalability in SRSs is the goal. Therefore, we claim that the proposed scheme is highly appropriate to the cold-start problem.

D. SENSITIVITY OF PARAMETERS

This experiment evaluates the sensitivity of the fusion parameter γ and the smoothing parameter λ of BiFu.

1) SENSITIVITY OF FUSION PARAMETER γ

BiFu makes use of γ to fuse recommendations that are achieved in item-based and user-based manners. Thus, we

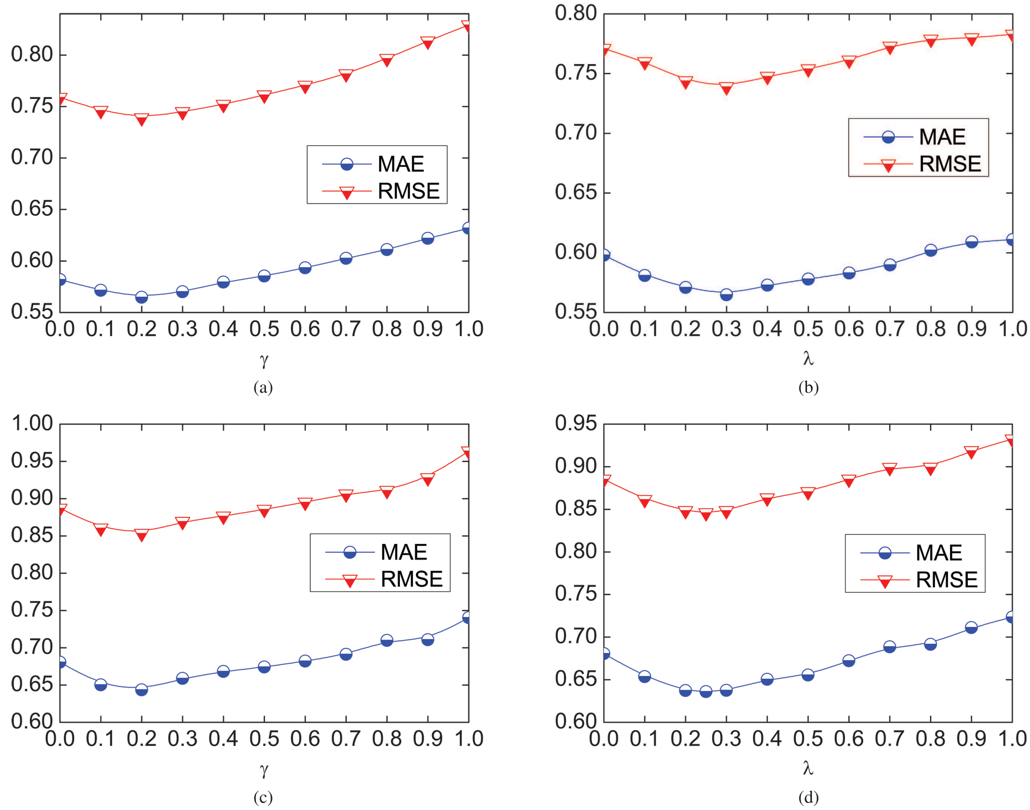


FIGURE 7. Sensitivity of parameters in BC experiment. (a) Sensitivity of γ over MovieLens dataset. (b) Sensitivity of λ over MovieLens dataset. (c) Sensitivity of γ overNetflix dataset. (d) Sensitivity of λ overNetflix dataset.

designed experiments to check how much it affects the performance of BiFu.

Figure 7(a) and (c) illustrates the sensitivity of parameter γ in BC experiment over MovieLens and Netflix datasets. When γ varies from 0 to 1, BiFu firstly gets a high accuracy and lately a low accuracy. When the value of γ is between 0.1 and 0.3, BiFu gets high levels of accuracy. In our experiments, we set the value of γ as 0.2 for both datasets. On the other side, the value of γ is less than 0.5, indicating that the user-based mechanism contributes more in recommendation than the item-based mechanism. This exactly fits a fact that like-minded users usually go for the same item.

2) SENSITIVITY OF SMOOTHING PARAMETERS λ

Figure 7(b) and (d) shows the effect of λ in BC experiment over the MovieLens and Netflix datasets. With the increase of the value of λ, the accuracy of BiFu varies significantly. As the value of λ is between 0.1 and 0.4, BiFu gets better accuracy. This indicates that smoothed ratings have an impact on the performance of BiFu, but not that much as that of the original ratings. We set its value as 0.3 and 0.25 in MovieLens and Netflix experiments, respectively.

VI. CONCLUSION

In this paper, we have proposed a newly-fashioned scheme — BiFu, enabling the social media sites to alleviate the influence of the cold-start problem. We implement a prototype as a cloud service based on the Apache Hadoop. BiFu firstly proposes the concept of trivial ratings that are negative to social recommendation. It further puts forwards a identifying mechanism. Then, it leverages the Bi-clustering technique for both items and users to accurately identifying similar items and like-minded users as possible. Our empirical study have shown the superiority of the proposed scheme.

However, BiFu could be further improved. We are evaluating it on the cloud environment with more users involved. We are also investigating the item or user similarity calculation and dimension shrink of the rating matrix. We plan to make the BiFu as a general social recommender container so that it can support various recommender systems.

ACKNOWLEDGMENT

We would like to thank Mr. Wenyin Wang and Mr. Xi Ma for their discussions and constructive suggestions. We also thank Ms. Nicole Kwoh for her proofread.

REFERENCES

[1] X. Hu, T. H. S. Chu, H. C. B. Chan, and V. C. M. Leung, "Vita: A crowdsensing-oriented mobile cyber physical system," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 148–165, Jun. 2013.

[2] I. Guy and D. Carmel, "Social recommender systems," in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 283–284.

[3] D. Zhang, Q. Zou, and H. Xiong, "CRUC: Cold-start recommendations using collaborative filtering in internet of things," *Energy Proc.*, vol. 13, pp. 3454–3461, Dec. 2011.

[4] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*. New York, NY, USA: Springer-Verlag, Oct. 2011.

[5] J. Noel, S. Sanner, K.-N. Tran, P. Christen, L. Xie, E. V. Bonilla, E. Abbasnejad, and N. Della Penna, "New objective functions for social collaborative filtering," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 859–868.

[6] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.

[7] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: Scalable online collaborative filtering," in *Proc. 16th Int. WWW Conf.*, 2007, pp. 271–280.

[8] D. Zhang, J. Cao, M. Guo, J. Zhou, and R. Vaskar, "An efficient collaborative filtering approach using smoothing and fusing," in *Proc. ICPP*, 2009, pp. 147–154.

[9] M. Khabbaz and L. V. S. Lakshmanan, "TopRecs: Top-k algorithms for item-based collaborative filtering," in *Proc. EDBT*, 2011, pp. 213–224.

[10] N. Koenigstein, G. Dror, and Y. Koren, "Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy," in *Proc. 5th ACM Conf. RecSys*, 2011, pp. 165–172.

[11] J. Noel, S. Sanner, K.-N. Tran, P. Christen, L. Xie, E. V. Bonilla, E. Abbasnejad, and N. Della Penna, "New objective functions for social collaborative filtering," in *Proc. 21st Int. Conf. WWW*, 2012, pp. 859–868.

[12] B. Xu, J. Bu, C. Chen, and D. Cai, "An exploration of improving collaborative recommender systems via user-item subgroups," in *Proc. 21st Int. Conf. WWW*, Apr. 2012, pp. 21–30.

[13] Z. Ma, S. Li, H. Luo, J. Guo, P. Gallinari, and D. Chen, "A cross-domain recommendation model for cyber-physical systems," *IEEE Trans. Emerg. Topics Comput.*, to be published.

[14] P. Victor, C. Cornelis, A. Teredesai, and M. De Cock, "Whom should I trust?: The impact of key figures on cold start recommendations," in *Proc. ACM Symp. Appl. Comput.*, 2008, pp. 2014–2018.

[15] K. Zhou, S.-H. Yang, and H. Zha, "Functional matrix factorizations for cold-start recommendation," in *Proc. 34th Int. ACM SIGIR*, 2011, pp. 315–324.

[16] R M. Rodríguez, M. Espinilla, P. J. Sanchez, and L. M. Lopez, "Using linguistic incomplete preference relations to cold start recommendations," *Internet Res.*, vol. 20, no. 3, pp. 296–315, 2010.

[17] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD*, 2008, pp. 426–434.

[18] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Inf. Sci.*, vol. 178, no. 1, pp. 37–51, 2008.

[19] A. Schein, A. Popescul, L. Ungar, and D. Pennock, "Generative models for cold-start recommendations," in *Proc. SIGIR Workshop*, vol. 6. 2001, pp. 10–15.

[20] W. Chu and S.-T. Park, "Personalized recommendation on dynamic content using predictive bilinear models," in *Proc. 18th Int. Conf. WWW*, 2009, pp. 691–700.

[21] A. K. Menon, K. P. Chitrapura, S. Garg, D. Agarwal, and N. Kota, "Response prediction using collaborative filtering with hierarchies and side-information," in *Proc. 17th ACM SIGKDD*, 2011, pp. 141–149.

[22] S. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste, "Naïve filterbots for robust cold-start recommendations," in *Proc. 12th ACM SIGKDD*, 2006, pp. 705–712.

[23] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Proc. 5th IEEE ICDM*, Nov. 2005, pp. 625–628.

[24] M. Deodhar and J. Ghosh, "A framework for simultaneous co-clustering and learning from complex data," in *Proc. 13th ACM SIGKDD*, 2007, pp. 250–259.

[25] G. Chen, F. Wang, and C. Zhang, "Collaborative filtering using orthogonal nonnegative matrix tri-factorization," *Inf. Process. Manag.*, vol. 45, no. 3, pp. 368–379, 2009.

[26] K. W.-T. Leung, D. L. Lee, and W.-C. Lee, "CLR: A collaborative location recommendation framework based on co-clustering," in *Proc. 34th Int. ACM SIGIR Conf.*, 2011, pp. 305–314.

[27] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[28] S. Lee, D. Lee, and S. Lee, "Personalized DTV program recommendation system under a cloud computing environment," *IEEE Trans. Consumer Electron.*, vol. 56, no. 2, pp. 1034–1042, May 2010.

[29] C.-F. Lai, J.-H. Chang, C.-C. Hu, Y.-M. Huang, and H.-C. Chao, "CPRS: A cloud-based program recommendation system for digital TV platforms," *Future Generat. Comput. Syst.*, vol. 27, no. 6, pp. 823–835, 2011.

[30] J. Jiang, W. Pang, Y. Deng, K. He, and Z. Gu, "A blog personality recommender system based on cloud computing infrastructure," in *Proc. IJCSS*, May 2012, pp. 1–5.

[31] Y. Zhou, Y. Zhang, H. Liu, N. Xiong, and A. V. Vasilakos, "A bare-metal and asymmetric partitioning approach to client virtualization," *IEEE Trans. Services Comput.*, to be published.

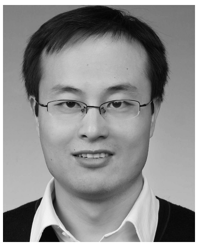
[32] J. Wang, A. P. de Vries, and M. J. T. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proc. SIGIR*, 2006, pp. 501–508.

[33] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proc. 28th Annu. Int. ACM SIGIR Conf.*, 2005, pp. 114–121.

[34] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proc. 30th Annu. Int. ACM SIGIR Conf.*, 2007, pp. 39–46.

[35] I. S. Dhillon, S. Mallela, and D. S. Modha, "Information-theoretic co-clustering," in *Proc. 9th ACM SIGKDD Int. Conf. KDD*, 2003, pp. 89–98.

[36] K. W.-T. Leung, D. L. Lee, and W.-C. Lee, "CLR: A collaborative location recommendation framework based on co-clustering," in *Proc. 34th Int. ACM SIGIR Conf.*, 2011, pp. 305–314.



DAQIANG ZHANG received the B.Sc. degree in management science and the M.Sc. degree in computer science from Anhui University, Anhui, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2010. He was a Full-Time Software Engineer at Autodesk, Inc., Shanghai. From June 2008 to June 2009, he was a jointly supervised Ph.D. candidate at Hong Kong Polytechnic University, Hong Kong. From September to November 2010, he was an Analyst at Goldman Sachs, Inc., Beijing, China. From July 2011 to July 2012, he was a post-doctoral fellow at Institute Telecom, France. Currently, he is an Associate Professor with the School of Software Engineering, Tongji University, Shanghai. He is involved in research on mobile computing, distributed computing, and wireless sensor networks. He has published more than 60 papers in major journals and international conferences in those above areas, including the IEEE Transactions on Parallel and Distributed Systems, the *IEEE Network Magazine*, ICPP, ICC, and WCNC. He received the Best Paper Award from ACCV'2009 and UIC'2012. He is in the editorial board in *European Transactions on Telecommunications* (Wiley publisher), the *International Journal of Big Data Intelligence* (Inderscience publisher), the KSII Transactions on Internet and Information Systems (Korea Society of Internet Information), and the *New Review of Hypermedia and Multimedia* (Taylor & Francis publisher). He is a member of CCF.

CHING-HSIEN HSU received the B.Sc. degree in computer science from Tung Hai University, Taichung, Taiwan, in 1995 and the M.Sc. and Ph.D. degrees in computer science from Feng Chia University, Taichung, in 1996 and 1999, respectively. He is currently a Professor with the Department of Computer Science and Information Engineering, Chung Hua University, Hsinchu, Taiwan.

His research interests include primarily in parallel and distributed computing, cloud and grid computing, big data, and as well as ubiquitous intelligence and computing. He has published over 200 academic papers in journals, books, and conference proceedings. He was awarded five times Annual Outstanding Research Awards and the Distinguished Award for excellence in research from Chung Hua University. He is serving on a number of journal editorial boards. He is the Editor-in-Chief of the *International Journal of Big Data Intelligence* and the *International Journal of Grid and High Performance Computing*. He has severed many international conferences as a chair and committee member.

MIN CHEN is a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University (SNU), Seoul, Korea, from 2009 to 2012. He was a Research and Development Director with Confederal Network, Inc., from 2008 to 2009. He was a Post-Doctoral Fellow in the Department of Electrical and Computer Engineering, University of British Columbia (UBC), Vancouver, BC, Canada. Before joining UBC, he was a Post-Doctoral Fellow at SNU.

He received the Best Paper Award from the IEEE ICC 2012 and the Best Paper Runner-up Award from QShine 2008. He has published more than 170 paper publications. He serves as an Editor or Associate Editor for *Information Sciences, Wireless Communications and Mobile Computing, IET Communications, and IET Networks*, the *Wiley International Journal of Security and Communication Networks*, the *Journal of Internet Technology*, *KSII Transactions on Internet and Information Systems*, and the *International Journal of Sensor Networks*. He is a Managing Editor for IJAACS and IJART. He is a Guest Editor for the IEEE Network, the IEEE Wireless Communications Magazine. He is a Co-Chair of the IEEE ICC 2012-Communications Theory Symposium and the IEEE ICC 2013-Wireless Networks Symposium. He is a General Co-Chair for the 12th IEEE International Conference on Computer and Information Technology (IEEE CIT-2012). He is a Keynote Speaker for CyberC 2012 and Mobiquitous 2012. He is a TPC member for IEEE INFOCOM 2014. His research focuses on Internet of Things, machine to machine communications, body area networks, body sensor networks, E-healthcare, mobile cloud computing, cloud-assisted mobile computing, ubiquitous network and services, mobile agent, and multimedia transmission over wireless network.

QUAN CHEN received the B.S. degree in computer science from Tongji University, Shanghai, China, in 2007 and the M.S. degree in computer science from Shanghai Jiao Tong University, Shanghai, in 2009, where he is currently pursuing the Ph.D. degree with the Embedded and Pervasive Computing Center, Department of Computer Science and Engineering.

His research interests include parallel and distributed processing, task scheduling, cloud computing, and chip multiprocessor. He has published 20 papers consisting of the IEEE Transactions on Parallel and Distributed Systems, ICPP, IPDPS, ICS, and CIT.

NAIXUE XIONG received the Ph.D. degree in software engineering and in dependable networks from Wuhan University, Wuhan, China, and the Japan Advanced Institute of Science and Technology, Ishikawa, Japan. He was a Visiting Assistant Professor and Research Scientist with Georgia State University, Atlanta, GA, USA. Currently, he is a Professor with the School of Computer Science, Colorado Technical University, Colorado Springs, CO, USA.

His research interests include security and dependability, cloud computing, network architecture, and optimization theory. He has published over 100 journal papers. Some of his works were published in IEEE JSAC, IEEE or ACM Transactions, ACM Sigcomm Workshop, IEEE INFOCOM, and IPDPS. He has been a General Chair, Program Chair, Publicity Chair, PC Member and OC Member of over 100 international conferences, and as a reviewer over 100 international journals, including the IEEE JSAC, the IEEE SMC (Park: A/B/C), the IEEE Transactions on Communications, the IEEE Transactions on Mobile Computing, the IEEE Transactions on Parallel and Distributed Systems. He is serving as an Associate Editor or Editor Member for over ten international journals (including *Information Science*), and a Guest Editor for over ten international journals, including *Sensor Journal, WINET*, and *MONET*. He has received the Best Paper Award in the 10th IEEE International Conference on High Performance Computing and Communications (HPCC-08) and the Best student Paper Award in the 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009). He is a member of the IEEE ISATC, the IEEE TCPP, and the IEEE TCSC, and a Chair of Trusted Cloud Computing in the IEEE Computational Intelligence Society.

JAIME LLORET received the M.Sc. degree in physics in 1997, the M.Sc. degree in electronic engineering in 2003, and the Ph.D. degree in telecommunication engineering in 2006. He is a Cisco Certified Network Professional Instructor. He was a network designer and administrator in several enterprises. He is currently an Associate Professor in the Polytechnic University of Valencia, Valencia, Italy. He is the Head of the research group communications and remote sensing of the Integrated Management Coastal Research Institute and he is the Head of the Active and collaborative techniques and use of technologic resources in the education (EITACURTE) Innovation Group.

He has authored 12 books and has published more than 220 research papers in national and international conferences, international journals (more than 50 with ISI Thomson Impact Factor). He has been the co-editor of 15 conference proceedings and a guest editor of several international books and journals. He is an editor-in-chief of the *International Journal Networks Protocols and Algorithms*, IARIA Journals Board Chair (eight journals) and he is an Associate Editor of several international journals. He has been the general chair (or co-chair) of 18 international conferences (chairman of SENSORCOMM 2007, UBICOMM 2008, ICNS 2009, ICWMC 2010, eKNOW 2012, SERVICE COMPUTATION 2013, COGNITIVE 2013, and ADAPTIVE 2013, and cochairman of ICAS 2009, INTERNET 2010, MARSS 2011, IEEE MASS 2011, SCPA 2011, ICDS 2012, 2nd IEEE SCPA 2012, GreeNets 2012, 3rd IEEE SCPA 2013, and local chair of MIC-WCMC 2013). He is a co-chairman AdHocNow 2014 and local chair of the IEEE Sensors 2014. He is fellow of IARIA.