IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 29, 2018, accepted July 31, 2018, date of publication August 6, 2018, date of current version September 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2863543

A Recommendation Method for Social Collaboration Tasks Based on Personal Social Preferences

JIAQIU WANG[1], ZHONGJIE WANG[1], (Member, IEEE), AND JIN LI[2]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[2]School of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Corresponding author: Jiaqiu Wang (wangjiaqiu@hit.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1400604, in part by the Natural Science Foundation of China under Grant 61772155 and Grant 61472106, in part by the Fundamental Research Funds for the Central Universities under Grant HEUCFP201808, and in part by the National Key R&D Program of China under Grant 2017YFC0820700.

ABSTRACT In the social-collaboration scenario, the social-collaboration tasks need to be completed/coordinated by multiple people. For example, in the field of open-source software development, most software developments require multiple developers to collaborate with each other. With the increase in the number and variety of social-collaboration tasks, it is difficult for individuals to discover social-collaboration tasks that they can participate. If we can help match the social-collaboration tasks with appropriate users, the quality and speed of these tasks will be improved, thus social organizations (e.g., companies, teams, and research institutions) and individuals can improve productivity, which is very significant. However, most related work in recommending individuals to participate in social-collaboration tasks mainly focus on individual's data features/characteristics (e.g., personal behaviors and attributes), and little work is based on the social collaborative data features/characteristics within the individual's participation in the social-collaboration tasks, such as the types of social-collaboration tasks that individuals participate, collaborative content, collaborative behavior, collaborative intensity, and so on. This paper proposes a universal recommendation method based on personal social-collaboration preferences, which is used to recommend the social-collaboration tasks that individuals can participate. The characteristics of social-collaboration data contain a large number of individuals' preferences for social-collaboration, which helps improve recommendation performance. We perform a large number of experiments to verify the effectiveness of our proposed method, based on our collected real-world data sets in the open source software development services (Bugzilla and Github). Experimental results show that the proposed algorithm can be well applied to the social-collaboration task recommendation.

INDEX TERMS Social-collaboration tasks, personal social-collaboration preferences, recommendation method, real-world data sets.

I. INTRODUCTION

Nowadays, more and more individuals are involved in social-collaboration tasks. The social-collaboration platform based on the Internet can help multiple people achieve collaborative tasks such as Github [1] service that is a web-based open source software development and Wiki [2] service that is a website on which users collaboratively modify content and structure. In addition to Github and Wiki services, TracWiki, MediaWiki, and PukiWiki [3], [4] are network-based multi-user social-collaboration services. They are mainly used to communicate, share, and cooperate with employees within the enterprise. When individuals use the web-based social-collaboration platform, a large amount of personal behavior data is generated. These personal behavior data can reflect individuals' preferences and interests in participating in the social-collaboration. An illustration of social-collaboration with multiple users involved is shown in Figure 1. Users (users 3 and 4 in Figure 1) can participate in multiple social-collaboration tasks with their own interests. In many fields, online social-collaboration service providers extract personal

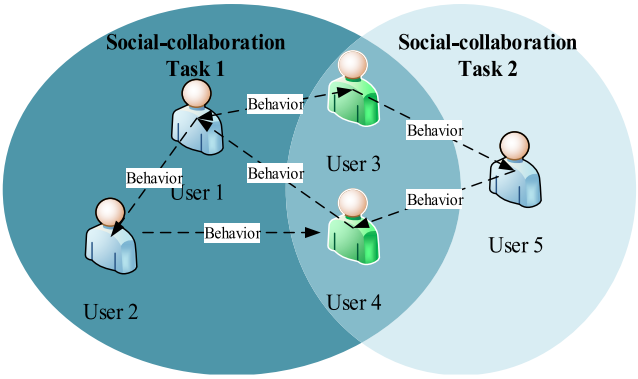


FIGURE 1. An illustrative example shows that multiple users are participating in social-collaboration tasks.

behavior preferences and interests from these personal behavior data to achieve different purposes.

In the network-based social-collaboration, individuals have to face enormous and complicated social-collaboration tasks, which is difficult for social tasks to quickly and accurately discover/locate appropriate individuals that they can participate/interest. If we can help social tasks match the appropriate individuals, it could improve the quality of the social-collaboration tasks, which is very significant to help companies/teams/individuals improve their productivity. How to quickly and accurately recommend social tasks that individuals can participate requires a personalized social-collaboration task recommendation method.

The personalized social-collaboration task recommendation method is that recommends appropriate individuals to social-collaboration tasks based on individuals' interests, preferences, and behavior habits. Personalized recommendation technology has been widely studied and applied in e-commerce, search engine, and software engineering [5]–[7]. Nowadays, due to the vigorous development of web-based social-collaboration platforms, related theories and applications have gradually become the focus in research. In most related work, the recommendation for social-collaborative tasks is mainly oriented to specific areas (such as the multi-user collaborative creation of hypertext, software development, etc.), and little work proposes a universal social collaborative task recommendation algorithm applied to multiple scenarios.

On another hand, most related work is mainly based on the user's own degree of interest in the recommended products, or based on his/her behavioral preferences. However, they rarely consider the social preferences that users embody in their participation in social-collaboration tasks. For example, how frequently do users interact with other users when they participate in social-collaboration tasks? The higher is the frequency, the stronger is the willingness of the user to participate in collaborative tasks. The contribution rate of the user to a social-collaboration task, i.e., the user's workload as a percentage of all users' workload, the higher the contribution rate indicates that the user is more interested in this type of social-collaboration tasks. We believe that individual's social preferences are more conducive to improving recommendation performance, especially when social collaboration tasks are recommended.

This paper proposes a universal recommendation method based on personal social preferences. This method integrates the social preferences that embody in their participation in social-collaboration tasks. It is used for recommending suitable users to social-collaboration tasks. This paper is based on the assumptions that we convert users' social preferences into some data features that embody in users' historical data set. We extract some data features from the raw data for use in model construction. Data features [25] are the characteristics (variables, predictors) selection in machine learning, which are used to enable the algorithm to achieve better performance.

Each user has a different degree of interest for the same data feature, and we regard the degree of interest on the same data feature as the preference. Our recommendation method extracts multiple data features (e.g., task-type, task-severity, individual behavior, and collaborative relationship, etc.) from user-involved social-collaboration tasks. If a user is interested in a social-collaboration task, that is, the user is interested in the data features contained in these social tasks. Our recommendation method also attempts to extract the degree of interest of different data features. By comparing the different user's data features (including the degree of interests) with the data features of new social-collaboration tasks, the users with the greatest relevance is recommended to each social-collaboration task.

In order to verify the effectiveness of the proposed method, we compare the recommendation performance (e.g., precision, recall, etc.) with other methods which are based on personal behavioral preferences (not social preferences). All algorithms are running on two real-world data sets that we have collected from web-pages. One data set is from open-source software development Github, and another data set is from Bugzilla that is a web-based general-purpose bug-tracker and testing tool. These two data sets are fit for our scenario that multiple users participate in social-collaboration tasks. The experimental results show that our proposed method based on personal social preferences can improve the performance of recommending social-collaboration tasks.

The main contributions of this paper are as follows:

- We propose a universal recommendation method based on personal social preferences. This method extracts various data features from social-collaboration tasks that user participates and computes the degree of interest for each data feature. The goal is to recommend suitable users to social-collaboration tasks.
- We combine personal social preferences and personal behavioral preferences extracted from engaged social tasks, which helps to improve the performance of recommendation.
- We conduct comprehensive experiments on two real-world data sets crawled from Github and Bugzilla with three different metrics. The experimental results

demonstrate that our method can be well applied to social-collaboration task recommendation.

The remainder of this paper is organized as follows. Section 2 gives the notions and the problem definition of social-collaboration task recommendation. Section 3 presents a personal social preferences model. Section 4 designs and implements the proposed recommendation algorithm. Section 5 analyzes the performance of the recommendation algorithm by performing experiments. Section 6 shows the related work. Section 7 summarizes this paper and presents the future work.

II. NOTIONS AND PROBLEM DEFINITION

This section presents definitions of related notions and gives the formal description of the problem of recommending social-collaboration tasks.

Definition 1 (Personal Preferences (PP)): Since different users have different degrees of interest for different data features. We regard the collection of degrees of interest on different data features as personal preferences. In other words, personal preferences is a collection of multiple data features (df) with their degrees of interest (di). Formally, PP $= \{(df_1, di_1), (df_2, di_2), \ldots, (df_n, di_n)\}$, $(n > 2)$. We assume that personal preferences consist of personal social preferences (PSP) and personal behavioral preferences (PBP). Formally, PP = {PSP, PBP}, PSP = {CTP, CRP}.

Definition 2 (Personal Social Preferences (PSP)): It refers to the user's degree of social interest in some data features that user participates in social-collaboration tasks. PSP can be subdivided into collaborative-task preferences (CTP) and collaborative-relationship preferences (CRP).

Definition 3 (Collaborative-Task Preferences (CTP)): It refers to the degree of interest in data features on social-collaboration tasks. For example, the types of tasks that users interested can be classified (by text classification) as {A: online knowledge editing, B: software development, C: internal collaboration, D: . . . , etc.}, task-severity: {Critical, Major, Normal, Minor, Trivial, . . . ,etc.}, task-priority: {P1, P2, P3, . . . , etc.}, task-status: {New, Resolved, Reopen, Closed} and so on.

Definition 4 (Collaborative-Relationship Preferences (CRP)): It refers to the degree of interest in the data features on the collaborative-relationship with other users. For example, collaborative-scale in social-collaboration tasks: {less: 2-5 individuals, generally: 5-10 individuals, more: 11 or more individuals}, the collaborative-intensity that interacts with other users: {Strong, Medium, Weak}, the impact on other users: {has no effect, can be referred, must learn}, collaborative-contribution: {A: 1%-30%, B: 31%-70%, C: 71%-99%}, collaborative-role: {member, leader, collaborator} and so on.

Definition 5 (Personal Behavioral Preferences (PBP)): It refers to user's degree of interest in the data features on personal behaviors (emphasizing his own behaviors and not interacting with other users). For example, the user's interested personal behaviors: {update, comment, ask, request}, the ability that user possesses: language ability: (English, Chinese, French), knowledge and skills and so on.

Our proposed method attempts to extract CTP, CRP, and PBP described above (e.g., task type, task severity, individual behavior, and collaboration relationship, etc.) from user-involved social-collaboration tasks. Our method also attempts to quantify these degrees of interest for various preferences.

Definition 6 (Social-Collaboration Task Recommendation Problem (SCTRP)): Given a social-collaboration team/organization *Group*, a set $User = \{u_1, u_2, \ldots, u_n\}$, $(n > 2)$ denotes a collection of users in that *Group*. $SocialTask_m = \{st_1, st_2, \ldots, st_m\}$ denotes a collection of social-collaboration tasks, which are jointly completed by these users over T time period. We first extract some data features of various preferences (CTP, CRP, and PBP) from $SocialTask_m$. Then, we calculate the degrees of interest on these data features. Formally, $\text{CTP}_{k1} = \{(df_1^{ctp}, di_1^{ctp}), \ldots, (df_{k1}^{ctp}, di_{k1}^{ctp})\}$, $k1$ is the number of data features on CTP. $\text{CRP}_{k2} = \{(df_1^{crp}, di_1^{crp}), \ldots, (df_{k2}^{crp}, di_{k2}^{crp})\}$, $k2$ is the number of data features on CRP. $\text{PBP}_{k3} = \{(df_1^{pbp}, di_1^{pbp}), \ldots, (df_{k3}^{pbp}, di_{k3}^{pbp})\}$, $k3$ is the number of data features on PBP.

If new social-collaboration tasks $ST_w = \{st_1, st_2, \ldots, st_w\}$ are generated in the future $T+1$ period, our goal is to recommend these social-collaboration tasks ST_w to interested users. Formally, a recommendation algorithm using these historical data CTP_{k1}, CRP_{k2}, PBP_{k3}, ST_w, and we aim at learning a function f:

$$f : (\text{CTP}_{k1}, \text{CRP}_{k2}, \text{PBP}_{k3}, ST_w) \mapsto set\{u_i, u_j, u_z, etc.\} \quad (1)$$

where $u_i, u_j, u_z \in User$.

III. PERSONAL SOCIAL PREFERENCES MODEL

We adopt the vector space model [8] to model personal social preferences. Our model differs from the traditional vector space model in that we aggregate the data features with the highest weight from the user's various preferences. The weights of various data features represent the degree of interest for personal social preferences. The model is represented as a two-dimensional table. Personal preferences are influenced by CTP, CRP, and PBP, which can be represented as:

$$\text{PP} = \{\text{CTP}, \text{CRP}, \text{PBP}\} \quad (2)$$

Among these various features, in order to clearly distinguish the category of data feature and quantify the degree of interest, we extend the **Social Preferences two-dimensional Table** (SPT) by integrating the **Feature Value Variable** v_k^p and the **Weight Variable** w_k^p into PP ($p \in \{ctp, crp, pbp\}$, $k \in \{k1, k2, k3\}$). Thus, the SPT can be represented:

$$SPT = \begin{Bmatrix} df_1^{ctp} & \cdots & df_{k1}^{ctp} \cdots & df_{k2}^{crp} & \cdots & df_{k3}^{pbp} \\ v_1^{ctp} & \cdots & v_{k1}^{ctp} \cdots & v_{k2}^{crp} & \cdots & v_{k3}^{pbp} \\ w_1^{ctp} & \cdots & w_{k1}^{ctp} \cdots & w_{k2}^{crp} & \cdots & w_{k3}^{pbp} \end{Bmatrix} \quad (3)$$

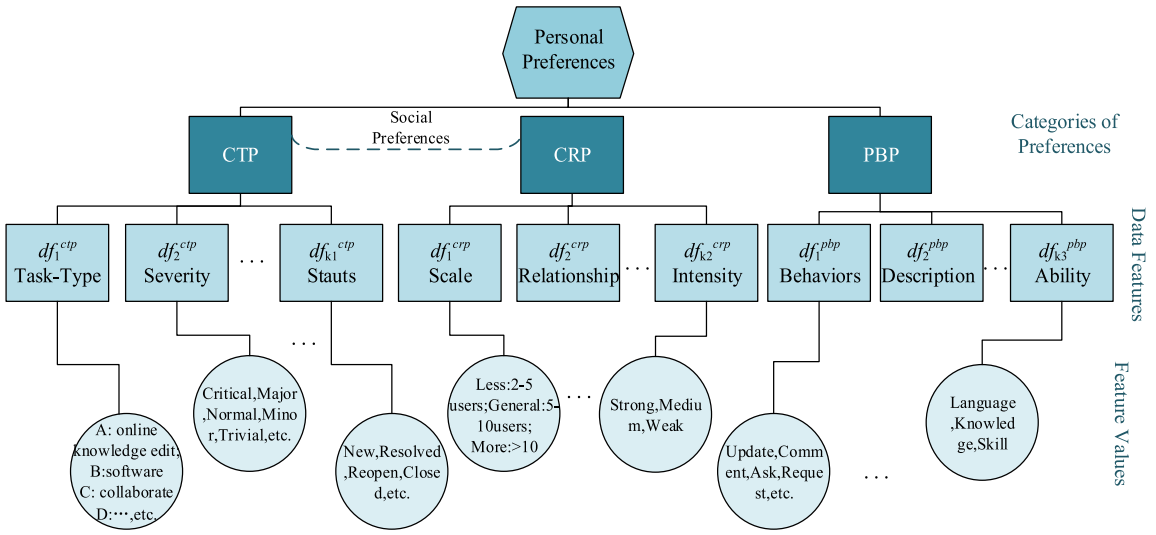


FIGURE 2. A tree structure of "Preferences-Data Features-Feature Values".

where *ctp*, *crp*, *pbp* denote different categories of data features, $v_{k1}^{ctp}, v_{k2}^{crp}, v_{k3}^{pbp}$ denote the feature value corresponding to each data feature, $w_{k1}^{ctp}, w_{k2}^{crp}, w_{k3}^{pbp}$ denote the weight of the feature value, respectively. For convenience, SPT can be abbreviated to SPT $= \{\langle df_1^{ctp}, v_1^{ctp}, w_1^{ctp}\rangle, \ldots, \langle df_{k1}^{ctp}, v_{k1}^{ctp}, w_{k1}^{ctp}\rangle, \ldots \langle df_{k2}^{crp}, v_{k2}^{crp}, w_{k2}^{crp}\rangle, \ldots, \langle df_{k3}^{pbp}, v_{k3}^{pbp}, w_{k3}^{pbp}\rangle\}$. The weight indicates the degree of interest in a certain feature value. The sum of weights of all feature values is 1, that is, $w_1^{ctp}+\ldots+w_{k1}^{ctp}+w_1^{crp}+\ldots+w_{k2}^{crp}+w_1^{pbp}+\ldots+w_{k3}^{pbp}=1$. For example, the categories of data features in CTP can be divided into task-types, task-descriptions, task-status, task-severity, task-priority, and task-platform, etc. The categories of data features in CRP can be divided into the number of people involved in the task, the collaborative intensity of interactions with other users, the impact on other users, the type of collaboration relationships, etc. The categories of data features in PBP can be divided into the personal abilities which are used to complete tasks, personal behaviors, personal data description, etc. Therefore, a tree structure of "Preferences-Data Features-Feature Values" can be constructed as shown in Figure 2.

IV. RECOMMENDATION METHOD

This section presents the proposed recommendation algorithm based on personal social preferences in detail.

A. PREFERENCES EXTRACTION

We construct the *SPT* by observing and extracting various data features from historical logs that users participate in social-collaboration tasks. In order to distinguish the different interests of different users for the same feature value, we adopt the weight variable w_k^p to indicate the degree of interest of a user for a certain feature value. We believe that different data features and feature values have different effects on different users. That is to say, different users value different features and feature values to decide to whether they would participate in a social-collaboration task. Some feature values have a greater impact on users, while other feature values have a smaller impact. To provide each user with in-depth personalized and fine-grained recommendations, we assign different weights to different feature values according to the degree of interest, which can improve the performance of the recommendation. In accordance with the social-collaboration preference model, we calculate the weight for each feature value. The weight $w_k^{p_k=v_j}$ on the *j*-th feature value of *k*-th data feature is calculated as follow:

$$w_k^{p_k=v_j} = \frac{NF(p_k = v_j)}{SF(p_k) + ns(p_k)} \cdot BC(p_k = v_j) \tag{4}$$

Where $NF(p_k = v_j)$ denotes the total number that data feature p_k is equal to the *j*-th feature value v_j, $SF(p_k)$ denotes the total times that user interacts with data feature p_k (with any feature values of p_k), $ns(p_k)$ denotes the number of all different feature values of data feature p_k. At the same time, $ns(p_k)$ can avoid the division by zero, $BC(p_k = v_j)$ denotes balanced coefficient of feature value $p_k = v_j$.

$$BC(p_k = v_j) = \frac{AF(p_k = v_j)}{SF(p_k)} \tag{5}$$

Where $AF(p_k = v_j)$ denotes average times that the feature p_k is equal to the *j*-th feature value v_j from all users, $SF(p_k)$ denotes average times that each user interacts with data feature p_k. Balanced coefficient $BC(p_k = v_j)$ is to balance the influence of user's habits on real degree of interest. The following gives a simple example that intuitively illustrates the extracting process of personal social preferences.

In the example, the user-set defined by $U = \{u_1, u_2, u_3, u_4, u_5\}$ that represents all members in a group. The set $SocialTask_{100} = \{st_1, st_2, \ldots, st_{100}\}$ denotes all social-collaboration tasks that five users participate in, and the user's SPT $= \{\langle df_1^{ctp}, v_1^{ctp}, w_1^{ctp}\rangle, \ldots, \langle df_{k1}^{ctp}, v_{k1}^{ctp}, w_{k1}^{ctp}\rangle,$

TABLE 1. Examples of social-collaboration task logs.

ST	U	Behaviors	Ability	Role	Task-Type	Severity	...	Intensity
st_1	u_1	Comment	Knowledge	Release	A:online knowledge	Critical	...	Strong
st_1	u_2	Update	Program	Collaborator	A:online knowledge	Critical	...	Normal
st_1	u_5	Discuss	Knowledge	Collaborator	A:online knowledge	Critical	...	Weak
st_2	u_1	Update	Knowledge	Member	D:research	Normal	...	Medium
st_2	u_4	Comment	Program	Collaborator	D:research	Normal	...	Strong
st_2	u_5	Discuss	Language	Member	D:research	Normal	...	Strong
...	...	...	...	...	...	..	...	...
st_{100}	u_5	Comment	Knowledge	Release	B:software	Major	...	Medium

$\ldots\langle df_{k2}^{crp}, v_{k2}^{crp}, w_{k2}^{crp}\rangle, \ldots, \langle df_{k3}^{pbp}, v_{k3}^{pbp}, w_{k3}^{pbp}\rangle\}$. Table 1 shows some examples of social-collaboration task logs. We describe the processes of the preferences extraction algorithm as follows.

Step 1 (Aggregating Each User Tasks): From all users' social-collaboration logs, we aggregate all the social-collaboration tasks that each user participates in. The reason why we aggregate all the tasks of each user is to mining each user's different preferences. So that we can provide user deep personalized recommendations, rather than provide a standardized recommendations. For example, for the user u_4, all social-collaboration tasks that he participates in are represented by $ST_4 = \{st_2, st_5, \ldots, st_{100}\}$, for the user u_1, all social collaboration tasks that he participates in are represented by $ST_1 = \{st_1, st_2, \ldots, st_{98}\}$.

Step 2 (Calculating the Correlation): Since we believe that different data features have different effects on different users, we need to find out the features with strong correlation. So we calculate the correlation between different features and collaborative intensity. This step is performed for each user. If the feature and the collaborative intensity are positively correlated, we assume that this data feature to be valid, otherwise we remove the feature from the user's SPT_i. Specifically, we estimate the correlation by calculating the Pearson Product Coefficient [9] that between data features and collaborative intensity.

For example, for user u_1, the correlation that between the behaviors data feature and the collaborative intensity is 0.56, which shows that the behaviors data features have a positive impact on participating in social-collaboration tasks. For user u_2, the correlation that between task-type data feature and collaborative intensity is 0.03, which shows that the task-type data feature has little impact on participating in social collaboration tasks. So this task-type feature can be ignored.

Step 3 (Calculating the Weight): In terms of the equation (4) and (5), we calculate the weight of each data feature. Our goal is to quantify the impact of different feature values of data features on participation in social-collaboration task. This can improve the performance of recommendation. After that, the calculated weight are filled into u_i's SPT_i.

For instance, if we want to calculate the $w_3^{severity=Major}$ that is the weight value of the severity data feature. The number of tasks with severity (data feature) is equal to Major (feature value) is 25, the total number of tasks that user participates is 82, the average number that each user participates in the same severity's feature value (Major) is 16, the average number of each user participates in social task is 80. There are 3 different feature values in the severity data feature. Then, the $w_3^{severity=Major} = [25/(82+3)] * 16/80 = 0.0588$.

Step 4 (Standardizing Weights): All the weight values in SPT are normalized so that the sum of all weight values is 1, i.e., $w_1^{ctp}+\ldots+w_{k1}^{ctp}+w_1^{crp}+\ldots+w_{k2}^{crp}+w_1^{pbp}+\ldots+w_{k3}^{pbp} = 1$. Then, all normalized weight values are updated into user's SPT.

Step 5 (Removing Uninterested Feature Value): The feature value with weight value is less than the weight threshold parameter v_s is removed from his SPT. Finally, it outputs user u_i's SPT_i. It should be noted that each user has a different SPTs due to the different personalization of each user.

For example, if the weight threshold parameter v_s is 0.1, and it outputs the SPT_3.

The Preferences Extraction algorithm's pseudo code is shown in Algorithm 1.

B. SOCIAL-COLLABORATION TASK RECOMMENDATION ALGORITHM

If new social-collaboration tasks $SocialTask_w = \{st_1, st_2, \ldots, st_w\}, (w > 2)$ are generated in the future $(T+1)$ period, our goal is to recommend these social collaborative tasks $SocialTask_w$ to interested users set: $\{u_i, u_j, etc.\}$. This paper proposes a universal recommendation method based on personal social preferences and behavioral preferences. This method establishes each user's SPT_i, that is, a preferences model that we learned from user collaborative history data. This model utilizes feature values with high weights to describe user's preferences. In the process of recommendation, we calculate the similarity between personal data features and the data features of the new tasks. Then, we recommend the most similar social tasks to users. We describe the processes of the Recommendation Social-collaboration Task (RST) algorithm as follows.

Step 1 (Matching Data Features): First of all, we represent each new social-collaboration task as a data feature collection. For example, the data feature collection of a new task: $Set\_st_h = \{type : C : collaborate, status : reopen, severity : normal, role : collaborator, \ldots\}$. In Set_st_h, we initially

Algorithm 1 PreferencesExtraction

Input: Social-collaboration tasks log *SCTLog*, weight threshold parameter v_s;
Output: u_i's SPT_i;

Initialize: v_s, $SPT_i \leftarrow \varnothing$, $ST_i \leftarrow \varnothing$;
for *each* u_i *in* U : **do**
  $ST_i \leftarrow$ GroupBy u_i
for *each* st *in* ST_i : **do**
  for *pap*, *ctp*, *crp* *in* SPT *:* **do**
    $R \leftarrow$ Pearson Coefficient (p, collaborative intensity)
    if $R \leq 0$ **then**
      Remove p from SPT_i
for *each* p_k *in* SPT_i : **do**
  calculating w_k^p
  w_k^p add into SPT_i
Standardizing w_k^p to be $w_1^{ctp}+\ldots+w_{k1}^{ctp}+w_1^{crp}+\ldots+w_{k2}^{crp}+w_1^{pbp}+\ldots+w_{k3}^{pbp}=1$
Update w_k^p from SPT_i
if $w_k^p < v_s$: **then**
  Remove w_k^p from SPT_i
return SPT_i ;

assign each data feature with the same weight value: 1. Next, since the feature value in each user's SPT_i is not the same as task st_h. We remove those feature values that are not the same from each user's SPT_i, leaving the feature values that overlap the new task st_h. Finally, each feature in SPT_i is mapped to st_h, and their weights are represented as vectors. For example, the Social Feature Value Vector (SFVV) is [0.132, 0.15, 0.23, 0.13, 0.12], the new Task's Feature Value Vector (TFVV) is [1, 1, 1, 1, 1].

Step 2 (Calculating Similarity): We calculate the similarity between the TFVV_h and different users' SFVV_i. We represent them as feature value vectors, and we can use these weight vectors to measure similarities between new tasks and users' social preferences. We use the cosine similarity [9] to measure. The similarity formula is as follows:

$$sim(SFVV_i, TFVV_h) = \frac{SFVV_i \cdot TFVV_h}{\sqrt{SFVV_i^2} \cdot \sqrt{TFVV_h^2}} \tag{6}$$

where $SFVV_i$ denotes user u_i's Social Feature Value Vector, $TFVV_h$ denotes the Task's Feature Value Vector of the h-th ($h \subset w$) new social task.

Step 3 (Recommending to Top K Users): For each new social-collaboration task st_h, we choose Top K users with the highest similarity as our target user set $\{u_i, u_j, u_z\}_h$ and recommend this task st_h to these users.

Algorithm 2 RecommendationSocialTask (RST)

Input: A set of all users' SPT:
$SPT\_Set = \{SPT_1, SPT_2, \ldots, SPT_n\}$, A set of all new Social-Collaboration tasks:
$SocialTask_w = \{st_1, st_2, \ldots, st_w\}$;
Output: The target user set $\{u_i, u_j, u_z\}_h$ for each task $st_h (1 \leq h \leq w)$;

Initialize: K, $\{u\}_h \leftarrow \varnothing$;
for *each* st_h *in* $SocialTask_w$: **do**
  Transform st_h into Set_st_h
  for SPT_i *in* SPT_Set **do**
    Overlap $Set\_st_h withSPT_i$
    Transfer Overlap to $SFVV_i$ and $TFVV_h$
    $sim(SFVV_i, TFVV_h)$ add into SIM_SET
  $\{u\}_h \leftarrow$ Pick up Top K similarity from SIM_SET
  Return $\{u\}_h$

The Recommendation Social Task (RST) algorithm's pseudo code is shown in Algorithm 2.

V. EXPERIMENT

In this section, we perform experiments to verify the effectiveness of our proposed algorithm. First of all, we introduce the experimental data set. Then, we represent the experimental setup, environment, and evaluation metrics. Next, the experimental results and analysis are shown. Finally, parameter tuning and performance analysis are given.

A. DATA SET

We perform the experiments by using two real-world data sets. By crawling the web-pages, we collect two real-world data sets which are from Bugzilla and Github. In these two service scenarios, the user is the developer, the social-collaboration tasks in the Bugzilla data set is the bug, the social-collaboration tasks in the Github data set is the issue. A social-collaboration scenario is where multiple developers in a group/team address program bugs/issues by discussing, learning, editing, and updating code. Therefore, these two data sets fit our scenario that multiple users participate in social-collaboration tasks. In addition, the developer's data is rich and publicly available in both web-sites. The crawled web-pages of the two web-sites are shown in Figure 3.

The collected Bugzilla data set contains a total of 41,686 modification bugs and the Github data set contains

$$SPT_3 = \left\{ \begin{array}{cccccc} Behaviors & Ability & Role & Type & Severity & Intensity \\ update & knowledge & release & B & major & strong \\ 0.132 & 0.175 & 0.151 & 0.23 & 0.148 & 0.13 \end{array} \right\}$$

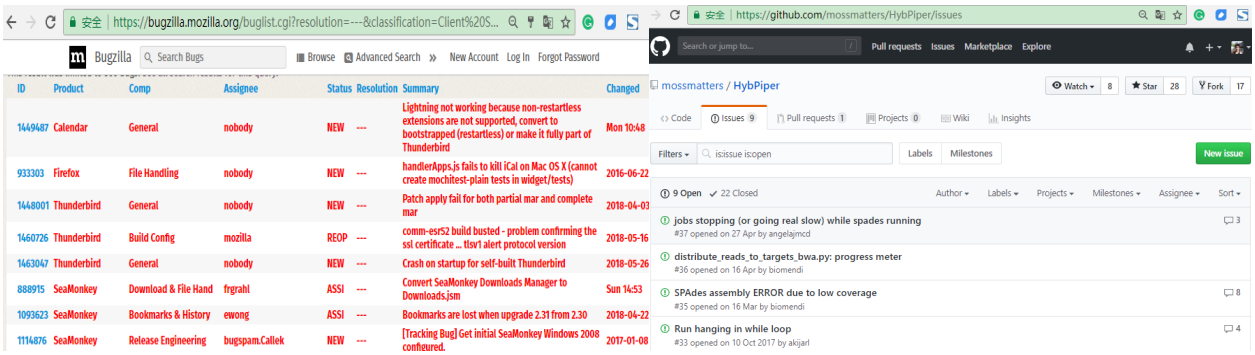


FIGURE 3. The Crawled Web-pages from Bugzilla and Github.

TABLE 2. Some samples of collected Bugzilla data set.

u	BugID	Activity	Role	Type	Severity	Status	Intensity	...
Nel	45773	Describe	Reproter	C	Critical	Reopen	Strong	...
Tim	45773	Comment	Worker	C	Critical	Reopen	Medium	...
Wee	14361	Comment	Worker	B	Major	New	Weak	...
Time	14361	Update	Collaborator	A	Major	Fixed	Medium	...
...	...	...	...	...	...	..	...	...

TABLE 3. Some samples of collected Github data set.

u	issueID	Activity	Role	Type	Status	Intensity
sensi	#576	Edited	Release	D	Closed	Strong
Alvis	#576	Comment	Collaborator	D	Open	Strong
Alvis	#109	Comment	Release	A	Open	Medium
sensi	#2986	Update	Collaborator	B	Closed	Weak
...	...	...	...	...	...	...

41,878 revision issues. The data contained in each task (bug/issue) is the collaborated content by multiple developers. Examples of two data sets are shown in Tables 2 and 3. The features/attributes of the two data sets are not exactly the same. We upload the pre-processed data sets to our web disk.[1]

B. EXPERIMENTAL SETUP AND ENVIRONMENT

1) EXPERIMENTAL SETUP

In order to evaluate the effectiveness of the proposed RST algorithm, we divide the two data sets into 80% of the training data and the remaining 20% of the test data, respectively. First of all, we run the PreferencesExtraction (Algorithm 1) on the training data to obtain *SPT* that can describe each user's social preferences. We then implement the RecommendationSocialTask (Algorithm 2) on test data to recommend a set of interested users set $\{u_i, u_j, u_z\}_h$ for each new social-collaboration task. At the same time, we also perform other comparison methods (PBM) that only based on the user's personal behavior preferences (without social preference features). In addition, user-task matrix-based collaborative filtering methods (UCF and BCF) [6], [19] can also help users recommend social collaboration tasks. All methods are running on two data sets. We compare their recommended performance under the evaluation metrics.

[1]https://pan.baidu.com/s/1cVyoX9PIGowSk7ZyGKADuw

On another hand, in order to ensure the stability of the recommendation results, and the contingency of results can be avoided. We perform a 5-fold cross validation [10] of all the algorithms (including our algorithm and the comparison methods). In particular, We divide the data set into 5 equal parts. Four of them are taken as training set and the remaining one is used as testing set. Finally, it takes the average of all cross-validations as the final results. In addition, we also perform parameter tuning and performance evaluation experiments for the algorithm.

2) EXPERIMENTAL ENVIRONMENT

The programming environment is Python 3.5.2 and MongoDB v3.2.7, the Integrated Development Environment (IDE) is Pycharm 2017. The computing configuration environment is Intel Core(TM) i3-2100 CPU 3.10GHz, 8 GB RAM, 500 GB hard disk space and Window 10 64-bit operating system.

3) EVALUATION METRICS

We adopt precision ($precision@K$), recall ($recall@K$) and $F-measure$ to evaluate the recommendation performance. In our experiments, we recommend K users for each social-collaboration task. The precision is defined as the ratio of the number of social-collaboration tasks that user actually

TABLE 4. The comparison results on Bugzilla dataset.

Method	$precision@4$	$recall@4$	$precision@8$	recall@8	$F_1@4$	$F_1@8$
UCF	0.174	0.085	0.212	0.093	0.114	0.131
BCF	0.255	0.149	0.287	0.156	0.188	0.202
PBM	0.261	0.153	0.314	0.168	0.193	0.22
RST	**0.385**	**0.156**	**0.409**	**0.188**	**0.222**	**0.258**

TABLE 5. The comparison results on Github dataset.

Method	$precision@4$	$recall@4$	$precision@8$	recall@8	$F_1@4$	$F_1@8$
UCF	0.187	0.105	0.23	0.116	0.134	0.154
BCF	0.264	0.154	0.294	0.163	0.195	0.21
PBM	0.257	0.146	0.306	0.159	0.186	0.209
RST	**0.362**	**0.151**	**0.424**	**0.211**	**0.213**	**0.282**

participates to the number of all recommended collaboration-tasks by the method. The formula of $precision@K$ is shown as follows:

$$precision@K = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{T_i}{R_i} \tag{7}$$

Where T_i is the number of social-collaboration tasks that the user u_i actually participates in, R_i is the number of all recommended social-collaboration tasks to user u_i by the method, and n is the number of all users. The $recall@K$ is defined as the probability of participating in any social-collaboration task, that is, the number of social-collaboration tasks that each user actually participates in is the proportion of all social-collaborative tasks (including recommended and not recommended tasks). The formula of $recall@K$ is shown as follows:

$$recall@K = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{T_i}{AN} \tag{8}$$

Where AN is the number of all social collaboration tasks in the test set. $F-measure$ is defined as the harmonic mean of the $precision@K$ and $recall@K$. Its formula is shown as follows:

$$F_1@K = \frac{2 \cdot precision@K \cdot recall@K}{precision@K + recall@K} \tag{9}$$

C. EXPERIMENTAL RESULTS AND ANALYSIS

We compare Recommendations SocialTask (RST) algorithm with PBM, UCF and BCF algorithm on the evaluation metrics. Tables 4 and 5 compare the results of the various methods in the Bugzilla and Github data sets, respectively. We use $precision@4$, $precision@8$, $recall@4$ and $recall@8$ as evaluation metrics. The experimental results are shown in Table 4 and Table 5, which indicate that:

(1) PBM algorithm based on personal behavior preferences is superior to UCF and BCF which are based on collaborative filtering, indicating that personalized social-collaboration task recommendation is very necessary.

(2) The precision of UCF, PBM and RST algorithm increase as the K value increases, and the BCF algorithm increase slowly. The main reason is that the BCF algorithm has nothing to do with the K number of recommendations when it calculates the similarity between different tasks. This indicates that the recommendation performance which calculates the similarity between the task feature value vector and the user preference feature value vector is better than the recommendation performance which calculates the similarity between different tasks.

(3) The RST algorithm shows better results in both *precision* and *recall*, which indicates that the algorithm/model based on the integration of personal social preferences is more effective than the algorithm/model which only based on personal behavioral preferences. The RST algorithm can more effectively describe personal social preferences by calculating the correlation between feature values and preferences and quantifying the weight values of each feature values. This avoids the influence of uninteresting/unimportant feature values on the user's participation in social-collaboration tasks, thereby RST improves the recommendation precision.

D. PARAMETER TUNING AND PERFORMANCE ANALYSIS

1) THE EFFECT OF PARAMETERS ON THE EXPERIMENTAL RESULTS

We examine the effect of the weight threshold parameter v_s on the precision of the recommendation results. The weight threshold parameter v_s is used to determine the retained feature values within SPT. We fix other experimental environment/variable, and the experimental results are shown in Figure 4 and Figure 5. We try different values of the weight threshold parameters v_s: 0.05, 0.08, 0.1, 0.12, 0.15, 0.18, 0.2, and 0.25.

The experimental results indicate that the value of weight threshold parameter v_s that between 0.12 and 0.18 on the two data sets can achieve better results. When the value of v_s is greater than 0.18, the recommended precision begins to decrease quickly. This is mainly due to the fact that most of the frequent/interested feature values can be obtained when the value of v_s is between 0.12 and 0.18, in the meanwhile feature values that are seldom involved by the user are removed.

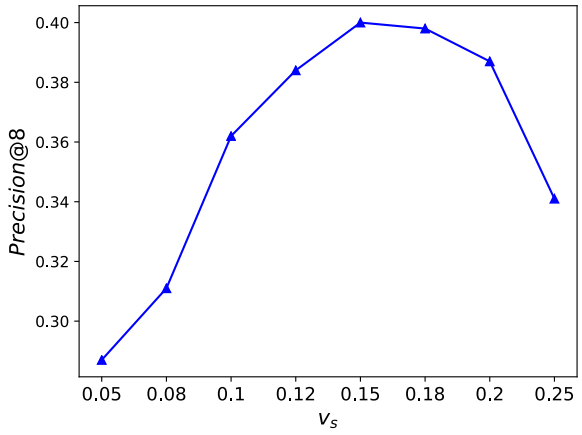


FIGURE 4. *Precision*@8 result with varied v_s on Bugzilla dataset.

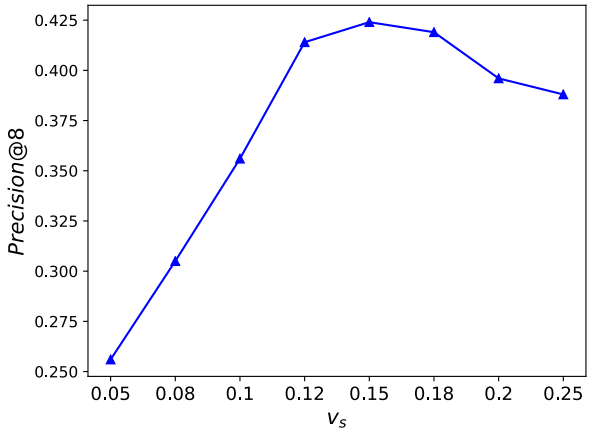


FIGURE 5. *Precision*@8 result with varied v_s on Github dataset

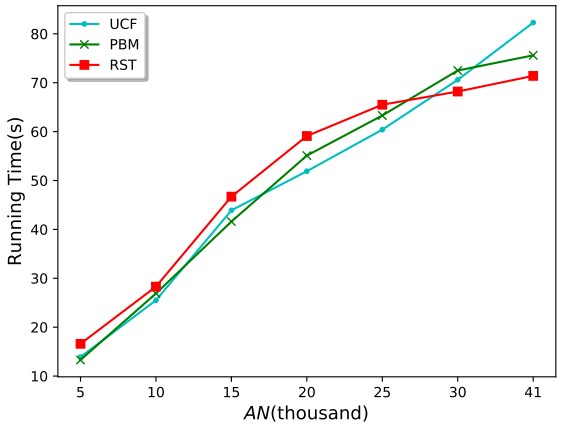


FIGURE 6. *RunningTime* with varied *AN* on Bugzilla dataset

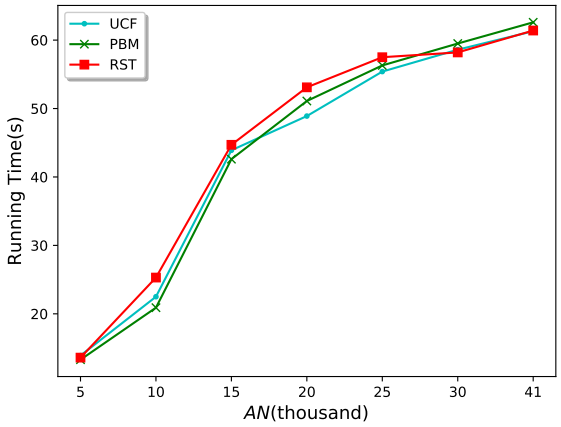


FIGURE 7. *RunningTime* with varied *AN* on Github dataset

2) RUNNING TIME WITH VARYING NUMBER OF TASKS

We further investigate the effect of the number of social-collaboration tasks on the running time of the algorithm. We fix the other experimental environments and vary the number of social-collaboration tasks. At the same time, we calculate the running time of all methods. The experimental results on the two data sets are shown in Figure 6 and Figure 7. The experimental results show that with the increase of the number of tasks, the running time of the three algorithms is gradually increasing. When the number of tasks, i.e., $AN \leq 15,000$, the running time of the RST algorithm increases more slowly. When $AN \leq 30,000$, the running time of the RST algorithm is lower than other comparison algorithms. This is mainly because the UCF algorithm based on collaborative filtering needs to repeatedly calculate the similarity between a large number of different users in each recommended process, whereas the RST algorithm only needs to calculate the similarity between the SFVV and TFVV once. The RST algorithm runs faster on the Github dataset than on the Bugzilla dataset. This is mainly due to the fact that the Bugzilla dataset has more data features than the Github dataset. The RST algorithm needs to calculate the weight value of each data feature when establishing the *SPT*.

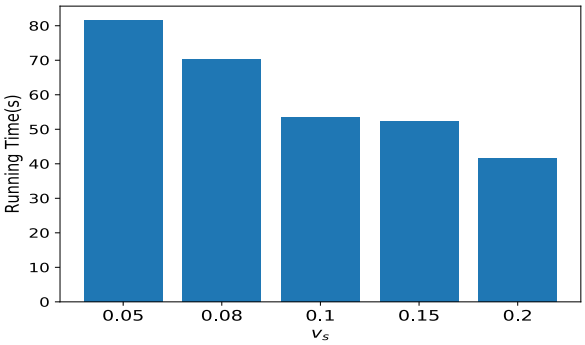


FIGURE 8. *RunningTime* with varing v_s.

3) RUNNING TIME WITH VARYING WEIGHT THRESHOLD PARAMETER v_S

We also study the effect of weight threshold parameter v_s on the running time. We fix the other experimental environments and vary v_s. At the same time, we calculate the running time of RST. The experimental result is shown in Figure 8. We observe that as the weight threshold parameter v_s increases, the running time of the RST algorithm gradually decreases. When v_s is between 0.1 and 0.15, the running time tends to be stable, when the v_s exceeds 0.15, it drops rapidly. This is mainly because when the v_s is less than

0.1, there are a large number of weighted terms of feature value in SPT, each weighted terms needs to be matched and the weights are calculated during the recommendation. However, with the increase of the v_s, feature values that are not user-interested are removed, and the feature values that the algorithm needs to match are reduced.

VI. RELATED WORK

As far as our investigations are concerned, there is currently litter research on the universal recommendation for personal social-collaboration tasks. We compare the following two researching aspects to reflect the significance of our work.

A. PERSONAL SOCIAL PREFERENCE MODEL

Regarding the preference model for individuals participating in social-collaboration tasks, most of the related work focuses on modeling general user preferences. At present, there are mainly three methods for modeling user preferences: spatial boolean model, vector space model, and latent semantic indexing model. The spatial boolean model is given a series of characteristic variables with binary logic. For example, the literature [11] extracts these variables from user's historical data and uses Boolean operations to represent user portraits as Boolean expressions. Regarding the vector space model, Salton *et al.* [12] utilize user preference documents with high weight keywords to establish the vector, where the weights are calculated by using the TF-IDF (term frequency-inverse document frequency) method. Chen *et al.* [13] establish a user preference document by utilizing user browsing behaviors and user-active feedback. Latent semantic indexing model [14] utilizes the relationship between data features and user data to establish the semantic structure of information. This model can reflect the most important correlative patterns between data.

In our work, we extend the vector space model so that user's social preferences, task preferences, and personal behavioral preferences can be expressed in terms of keywords and weights. Moreover, we consider that different feature values have different degrees of influence on the user, then we assign different weights to different feature values.

B. UNIVERSAL SOCIAL-COLLABORATION TASKS RECOMMENDATION ALGORITHM

Regarding the recommendation algorithm for social-collaborative tasks, most of the related work is concerned with the accuracy of the recommended algorithm. These recommendation methods include Content-based [15], [16], Collaborating Filtering [17]–[19], Demographic [20], and Hybrid [21]. For example, Orii and Naoki [22] propose a recommendation method based on user collaborative filtering to recommend software libraries that may be interested to developers. Brch *et al.* [23] propose a content-based recommendation method to recommend related software libraries to developers by mining their behavioral data. Xuan *et al.* [24] model the developer prioritization in a bug repository and assists in predictive tasks with his model.

The differences between our proposed recommendation algorithm and related work: (1) Related work is always restricted to a specific scenario for the recommendation. However, we propose a universal social collaborative task recommendation algorithm, which can be applied to the universal scenarios (e.g., software library recommendation and internal collaboration of company and so on). (2) Our method integrates the social preferences, personal behavioral preferences and task preferences that individuals involved in. Our method also extracts a variety of data features from social preferences, while most related work only considers personal behavioral preferences. (3) Our method considers that different feature value have different effects on individual participation in social-collaboration tasks. Therefore, we assign different weights to different feature value to indicate the degree of interest/importance.

VII. CONCLUSIONS AND FUTURE WORK

This paper proposes a recommendation method based on personal social preferences in the multiple users social collaborative scenario. This method is used universally to recommend social-collaboration tasks that users can participate/intereste. This method integrates the personal social preferences and personal behavioral preferences that users participate. Furthermore, this method utilizes the weights of the feature values to represent the degree of interest/influence on users' participation. Finally, we carry out comprehensive experiments by using the two collected real-world data sets. The experimental results verify the effectiveness of the proposed method.

Future work: We intend to design and implement a feedback learning mechanism to further improve the accuracy of recommendation. We also consider that user satisfaction and recommendation agility would affect the accuracy of recommendations. User agility refers to how long it takes the user to respond to the recommendation, and the quicker the response is, the higher the agility is. In addition, the efficiency of the RST algorithm will be further improved to accommodate more social collaborative tasks.

ACKNOWLEDGMENT

The authors thank the editors and the anonymous reviewers for their helpful comments and suggestions that have led to this improved version of the paper.

REFERENCES

[1] R. Arora, S. Goel, and R. K. Mittal, "Supporting collaborative software development over GitHub," *Softw. Pract. Exper.*, vol. 47, no. 10, pp. 1393–1416, Oct. 2017.

[2] A. Davoust, A. Craig, B. Esfandiari, and V. Kazmierski, "P2Pedia: A peer-to-peer Wiki for decentralized collaboration," *Concurrency Comput. Pract. Exper.*, vol. 27, no. 11, pp. 2778–2798, Oct. 2015.

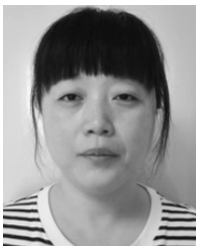
[3] A. Kuswara, A. Cram, and D. Richards, "Web 2.0 supported collaborative learning activities: Towards an affordance perspective," in *Proc. Int. Conf. Learn. Design*. 2008, pp. 70–80.

[4] C. Alario-Hoyos *et al.*, "Enhancing learning environments by integrating external applications," *Bull. IEEE Techn. Committee Learn. Technol.*, vol. 15, no. 1, pp. 21–24, Jan. 2013.

[5] A. Klašnja-Milićević, M. Ivanović, B. Vesin, and Z. Budimac, "Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques," *Appl. Intell.*, vol. 48, no. 6, pp. 1519–1535, Jun. 2018.
[6] H. K. Kim, J. K. Kim, and Y. U. Ryu, "Personalized recommendation over a customer network for ubiquitous shopping," *IEEE Trans. Services Comput.*, vol. 2, no. 2, pp. 140–151, Apr./Jul. 2009.
[7] C. Gianni and O. Riccardo, "Model-based collaborative personalized recommendation on signed social rating networks," *ACM Trans. Internet Technol.*, vol. 16, no. 3, p. 20, Jun. 2016.
[8] M. Ahmad, Z. Aneesh, A. Grilo, and R. Jardim-Goncalves, "Matching heterogeneous e-catalogues in B2B marketplaces using vector space model," *Int. J. Comput. Integr. Manuf.*, vol. 30, no. 1, pp. 134–146, Jan. 2017.
[9] P. Pirasteh, D. Hwang, and J. E. Jung, "Weighted similarity schemes for high scalability in user-based collaborative filtering," *Mobile Netw. Appl.*, vol. 20, no. 4, pp. 497–507, Aug. 2015.
[10] B. Giulio and P. Gianluigi, "The generalized cross validation filter," *Automatica*, vol. 90, no. 1, pp. 130–137, Apr. 2018.
[11] C. Danilowicz and A. Indyka-Piasecka, "Dynamic user profiles based on Boolean formulas," in *Proc. Int. Conf. Ind., Eng. Appl. Appl. Intell. Syst.*, May 2004, pp. 779–787.
[12] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.
[13] L. Chen and K. Sycara, "WebMate: A personal agent for browsing and searching," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 1998, pp. 132–139.
[14] K. Kesorn, Z. Liang, and S. Poslad, "Use of granularity and coverage in a user profile model to personalise visual content retrieval," in *Proc. Int. Conf. Adv. Hum.-Oriented Pers. Mech., Technol., Services*, Sep. 2009, pp. 79–87.
[15] J. Bu *et al.*, "Music recommendation by unified hypergraph: Combining social media information and music content," in *Proc. Int. Conf. Multimedia*, 2010, pp. 391–400.
[16] J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proc. Int. Conf. Digit. Libraries*, 2010, pp. 195–204.
[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. World Wide Web*, 2001, pp. 285–295.
[18] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.
[19] L. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proc. Int. Conf. Inf. Retr.*, 2009, pp. 195–202.
[20] T. Mahmood and F. Ricci, "Towards learning user-adaptive state models in a conversational recommender system," in *Proc. Int. Conf. Adapt. User Modeling Interact. Syst.*, 2007, pp. 373–378.
[21] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences," in *Proc. Int. Conf. Music Inf. Retri.*, 2006, pp. 296–301.
[22] N. Orii, "Collaborative topic modeling for recommending GitHub repositories," *Inf. Softw. Technol.*, vol. 83, no. 2, pp. 110–121, Apr. 2012.
[23] N. X. Bach, N. G. Hai, and T. M. Phuong, "Personalized recommendation of stories for commenting in forum-based social media," *Inf. Sci.*, vol. 352, no. 1, pp. 48–60, Jul. 2016.
[24] J. Xuan, H. Jiang, H. Ren, and W. Zou, "Developer prioritization in bug repositories," in *Proc. Int. Conf. Softw. Eng. (ICSE)*, Jun. 2012, pp. 25–35.
[25] N. B. Nazar and R. Senthilkumar, "An online approach for feature selection for classification in big data," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 25, no. 1, pp. 163–171, 2017.

JIAQIU WANG received the B.S. degree from the School of Computer Science and Technology, Harbin University of Commerce, Harbin, China, in 2011, and the M.S. degree from the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, in 2013, where he is currently pursuing the Ph.D. degree in software engineering. His current research interests focus on service computing, software engineering, and recommendation service collaboration based on personal data.

ZHONGJIE WANG (M'08) received the B.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology (HIT) in 2000 and 2006, respectively. He is currently a Professor with the School of Computer Science and Technology, HIT. He has authored over 40 publications, such as the IEEE ICWS and IEEE TSC. His research interests include service computing, mobile and social networking services, and software architecture.



JIN LI received the Ph.D. degree in computer science from the Harbin Institute of Technology in 2014. She is currently a Lecturer with Harbin Engineering University. Her research interests in software service engineering, model-driven engineering, MDA, model transformation, cloud services, and crowdsourcing.