Study on Cooperator Recommendation of Virtual Collaborative Community

Xiang Chen
School of Management and Economics, Beijing Institute of Technology, Beijing, P.R. China
Email: bjchenxiang@163.com

***Abstract*—Previous researches on virtual collaborative communities emphasize more on the interactive behavior but ignore the collaborative behavior. To address this issue, this paper focuses on how to apply the tags which express the technical need of projects and the interest of cooperators to cooperator recommendation. This paper proposes a method of measuring the tags similarity from both text semantics perspective and relational semantics perspective based on items and cooperators, and constructs the work preference of cooperator in virtual collaborative communities from a new way. By defining a cooperator recommendation model, this paper also introduces the method of measuring the similarity of work preferences between cooperators and calculating the matching degree between cooperators and projects. The acquired information can then be used in cooperator recommendation algorithm. Furthermore, this paper investigates the popularity of tags and proposes the method of recommending project to community newcomer using tag popularity. Finally, by using the open-source community data from www.codeplex.com and comparing with other algorithm, the behavior of the proposed recommendation algorithm is verified. Results show that it gets a good recommendation effect in virtual collaborative communities and solves the problem of sparse matrix and cold start.**

***Index Terms*—first term, second term, third term, fourth term, fifth term, sixth term[1]**

I. INTRODUCTION

With the development of computer science and network technology, more and more virtual communities emerge and keep growing, such as e-commerce sites, learning communities, and online dating sites. Virtual communities with collaborative behaviors, such as various open-source websites, have also been developing. Also the vast majority of companies have a virtual collaborative system, through which their members interact. A lot of high-tech work is completed via the virtual community, and many high-tech people cooperate with each other through virtual communities. The members of virtual communities are distributed in different regions and not familiar with each other. The communication and cooperation among them is different from traditional communities. There must be a recommendation mechanism that involves and represents the relationship between members and the interests of members in the virtual community.

Given the particularity of the relations between cooperators and the work flexibility of the virtual collaborative community, many researchers have done a lot of researches on virtual collaborative communities. Concas et al. applied Social Network Analysis (SNA) to the interactive behavior among community cooperators. The acquired information describes the communion and interaction well [1]. Toral et al. did the similar work and find that the middle-man role dilutes the core-edge structure and promote the development of the community [2]. Korba et al. constructs social network from the cognitive perspective and analysis its affect to collaborative work [3]. Massa et al. built trust relationship by the discussion information from the discussion groups and use Linear attenuation method to measure the trust relationship [4]. Matthew et al. study the projects from www.SourceForge.com by using SNA. The result proves that former connections of developers play an important part in the future process of projects [5].

Most of the existing researches of virtual collaborative communities use communication relationship and cooperation relationship of cooperators to build the social network model, and then study the relationship of cooperators and its impact on the communities by the model [6]. There are also many researches focus on domain ontologies model, which abstracts the impersonal existences in social network information domain into some primary ontologies [7]. In fact, project tags contain a very important message. Tags in e-commerce systems can be freely added by any user. But the tags of virtual collaborative communities are marked by the promulgators of a project, and therefore are more standardized, rational, and aggregated. The tags of a project not only tell us the technical requirements of the project, but also indicate the technical capacity and interests of the cooperators involved in the project. There are few researches focusing on using tag information to analyze collaborative virtual communities.

In other domains, tag is widely used for recommendation, mainly in two ways. First, tag-aware recommender systems use tag-based user-user or item-item similarities as a additional information to improve CF algorithms, like the literature [8]. Second, tag-based recommender systems are completely based on tags. Hung et al. take eco-occurrence of tags in the user-tag matrix to

[1] Manuscript received March 1, 2013; revised April 20, 2013; accepted May 1, 2013.
Supported by the Natural Science Fund of China (No. 71102111).

doi:10.4304/jsw.8.11.2908-2916

compute tag similarity and base recommendation on the similarity between the set of tags used by a user and the tags of an item. They built user profile - user tag vector, according to the sequence of the tag user labeled projects. For a pair of user and item, they find the most similar user tag for each item tag and sum up the maximal similarities [9].

Based on the theory Hung et al. built, this paper investigates the tag-based recommendation in virtual collaborative communities. We introduce a new way to built user profile (call it cooperator work preference bellow) for the virtual collaborative communities, and a new method of measuring the tags similarity from both text semantics perspective and relational semantics perspective based on items and cooperators. Then, we use the above result to calculate the similarity between cooperators and the matching degree between cooperators and projects. Moreover, this paper proposes a method of analyzing the popularity of tags and defining the subject degree of project in popular tag set. The acquired information is then used in the cooperative recommendation.

The rest of the paper is organized as follows: section II describes a new method to compute tag similarity; section III presents a cooperator recommendation mode based on tag similarity; section IV describes tag popularity and uses it to recommend projects to newcomer; section V examine the proposed mode; section VI gets a conclusion. At the end of the paper, we also provide a notations table for easy reading.

II. TAG SIMILARITY

Since the tags of the virtual collaborative community are manually added, there are inevitable errors caused by spelling mistakes, case difference, semantic ambiguity, polysemy, and so on. In addition, tags representing the technical requirements of the project are interdependent. It is necessary to measure the relationship among tags, so that the work preferences of cooperators and the technical needs of projects can be expressed more integrated. Meanwhile, it can reduce the negative impact caused by the noise and the sparsity of the tags on the cooperator recommendation.

There are many methods of speculating the relationship among tags [10], such as the Levenshtein Metric method, the cosine similarity based on co-occurrence distributing.

Markines *et al.* build an evaluation framework to compare various general tag-based similarity measures with different aggregation methods [11]. Their experiment proves most similarity measure of the collaborative aggregation outperforms all of the other aggregations.

Calculating tag similarity using co-occurrence distributing is a mature method. Learning from the collaborative aggregation method, this paper takes both of project-based co-occurrence and cooperator-based co-occurrence into account. They reflect some relationships of tags in the same way. Tags associated with the same project or cooperator relate with each other in the following ways: similar, correlated, complementary and so forth. Taking the two aspects into account will better reflect the relationship between the tags.

From another perspective, co-occurrence relationship can't fully reflect the relationships among tags, and some tags are not clearly different with each other because of few different letters. For example, it is not much difference between ".NET 3.0" and ".NET 3.5". Calculating tag similarity by the traditional co-occurrence distribution may lose some information. The Levenshtein distance (edit distance) is widely used to define the similarity of two strings. It is the minimum number of insertion, deletion and substitution operations needed to turn one string or sentence into another [12]. In this paper, we combine the normalized Edit distance with co-occurrence information to calculate tag similarity, measuring the tags similarity from both text semantics perspective and relational semantics perspective.

[Definition 1] Tag Relationship of Collaborative Network (TRCN).

A TRCN is 6-tuple $TR = (C, P, T, TLev, TPS, TCS, TS)$, where:

① $C = \{c_1, c_2, ..., c_z\}$ is a set of cooperators, $P = \{p_1, p_2, ..., p_m\}$ is a set of projects, $T = \{t_1, t_2, ..., t_n\}$ is a set of tags.

② $TLev$ is tag similarity based on Edit distance, called edit distance similarity; TPS is tag co-occurrence similarity based on projects; TCS is tag co-occurrence similarity based on cooperators;

③ TS is tag integrated similarity, and:

$$TS_{ij} = TS(t_i, t_j) = w_1 * TLev_{ij} + w_2 * TPS_{ij} + w_3 * TCS_{ij} \quad (1)$$

where, w_1、w_2、w_3 is the weight of $TLev$、TPS、TCS in TS, according to unitary condition.

A. Edit Distance Similarity

The edit distance can be calculated by dynamic programming formulation [13, 14].

We define $Lev(t_1, t_2)$ as the edit distance of string t_1 and t_2. According to the characteristics of the tag, tag edit distance similarity is defined as follows:

$$TLev_{ij} = 1 - \frac{Lev(t_i, t_j)}{Max(Length(t_i), Length(t_j))} \quad (2)$$

B. Project-based Co-occurrence Similarity

For any collaborative community, we can obtain Project-Tag matrix:

$$PT_{mn} = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{pmatrix}$$

We can get project-based tag co-occurrence matrix

$$Tp_{nn} = (tp_1, tp_2, , , tp_n)^T = \begin{pmatrix} tp_{11} & \cdots & tp_{1n} \\ \vdots & \ddots & \vdots \\ tp_{n1} & \cdots & tp_{nn} \end{pmatrix} = PT_{mn}{}^T PT_{mn}$$

.

Tp_{nn} indicates the project-based co-occurrence relationship of tags. tp_{ij} denotes the co-occurrence frequency of tags t_i and t_j. In the matrix Tp_{nn}, every row and column vector is a tag vector. We use the row vector $tp_i = (tp_{i1}, tp_{i2}, \cdots, tp_{in})$ as tag vector. The tag co-occurrence similarity is the similarity of tag vectors. There are some common similarity measure method, such as Euclidean distance, Jaccard coefficient, cosine similarity and Pearson correlation coefficient, and so on. Cosine similarity can well express the similarity between tags, so we use cosine similarity to calculate the project-based co-occurrence similarity of tags t_i and t_j, as Equation 3.

$$TPS_{ij} = cos(tp_{i,} tp_{j}) \tag{3}$$

C. Cooperator-based Co-occurrence Similarity

Projects involved with the same cooperator associate with each other in the perspective of technical, and each cooperator has a relative fixed work preference. Therefore, there are semantic relations among the corresponding tags.

For any collaborative community, we can obtain Cooperator-Project matrix:

$$CP_{zm} = \begin{pmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{z1} & \cdots & s_{zm} \end{pmatrix}$$

We can get cooperator-based project co-occurrence matrix Pc_{mm} by $Pc_{mm} = CP_{zm}{}^{T} CP_{zm}$. We set the diagonal value of Pc_{mm} to zero to eliminate the effect of project-based tag co-occurrence relations, and acquire the matrix Pc^{*}_{mm}. Then we use the Equation 4 to calculate the cooperator-based tag co-occurrence matrix:

$$Tc_{nn} = (tc_1, tc_2, , , tc_n)^{T} = PT_{mn}{}^{T} Pc^{*}_{mm} PT_{mn} \tag{4}$$

Finally, we define the cooperator-based co-occurrence similarity of tags t_i and t_j, as Equation 5.

$$TCS_{ij} = cos(tc_{i,} tc_{j}) \tag{5}$$

III. Cooperator Recommendation Model

Cooperator recommendation researches in the open network environment are mainly about the relationship between cooperators as well as the relationship between cooperators and projects. In addition to tags, there are other characteristics can express the collaborative relations and collaborative behaviors. We first define the virtual collaboration model as follows.

[Definition 2] Virtual Collaboration Model (VCM). A VCM is 6-tuple $VC = (C, P, T, TS, CS, CPS)$, where

① C, P, T, TP_s are the same with definition 1.

② CS is the work preference similarity of cooperators; CPS is the matching degree between work preference of cooperator and technical needs of project.

Cooperators pay dissimilar attention to different projects. The interest and work capability of each cooperator is different and will change with the time. The role, participant frequency, duration of a cooperator in a project reflects the preference of the cooperator in the project. The time of a cooperator recently involved in a project to a certain extent reveals that if the preference of the cooperator has changed. The method advanced to define user profile by tags is not suitable in the virtual collaborative communities, as tags are added by promulgator when the project is built, not by cooperator anon [9].

In the literature [15], the author proposed a method of using mark time, mark frequency and mark duration as the indicators for assessing the user long-term and short-term interest. Based on this basis, the author introduced a mixed-recommendation method. This method combines the advantages of content-based and collaborative filtering-based method. This paper expands this application, and uses the role of a cooperator in a project, the latest time to submit code, submit frequency and participant duration to assess the preference of the cooperator on the project. They also indicate the preference of the cooperator on the project-related tags.

[Definition 3] Preference Weight of Cooperator on Project (PWCP).

A PPWC is decided by participant role (P), recently participant time (R), participant frequency (F), and participant duration (D). P is the role of a cooperator in a project. The more important the role is, the more the interest is. R is the distance of the recent time of submitting code to current time. The shorter the distance is, the more the interest is. F is the times of submitting code in a certain period. The more the times is, the more the interest is. D is the duration of a cooperator participating in a project. The bigger the duration is, the more the interest is.

We divide the value of P, R, F, D into five equal portions and give them a score of 1 to 5 [15], Consistently with Table1.

TABLE I.

PRFD SCORE

Value section	P score	R score	F score	D score
[max,4/5(max－min)]	5	1	5	5
(4/5(max－min),3/5(max－min)]	4	2	4	4
(3/5(max－min),2/5(max－min)	3	3	3	3
(2/5(max－min),1/5(max－min)]	2	4	2	2
(1/5(max－min),min]	1	5	1	1

Then we define the PPWC of cooperator c_x to p_s as Equation 6:

$$pw_{xs} = \frac{P_{xs} + R_{xs} + F_{xs} + D_{xs}}{20} \quad (6)$$

Let $tw^*_{xi} = \sum_{s \in Dxi} pw_{xs}$, where Dxi is the set of projects that cooperator c_x participanted in and are involved with the tag t_i.

Then the preference weight of cooperator c_x to tag t_i is $tw_{xi} = \frac{tw^*_{xi}}{\sum_{i=1}^{n} tw^*_{xi}}$. And the cooperator tag vector is $ct_x = (tw_{x1}, tw_{x2}, \cdots, tw_{xn})$.

In the virtual collaborative community, Cooperator - Project Matrix and Project - Tag Matrix is sparse, leading to the sparse Cooperator - Tag Matrix $CT_{zn} = (ct_1, ct_2, \cdots, ct_z)^T$. If we calculate the similarity of cooperators by Cooperator - Project Matrix and Cooperator - Tag Matrix, the acquired information of relations between cooperators will be very limited. So in this paper, we calculate the similarity of cooperators based on the tag similarity and the similarity of cooperator tag vectors. This proposed method gets the relationship of cooperators via the relationship of tags, captures the information among cooperators as far as possible.

We redefine cooperator tag vector by removing the tag component of zero weight, then obtain the cooperator work preference vector as **Definition 4**.

[Definition 4] Work Preference Vector of Cooperator (WPVC).

A **WPVC** is a vector as $c_x = ((t_{cx_1}, w_{cx_1}), (t_{cx_2}, w_{cx_2}), \cdots, (t_{cx_k}, w_{cx_k}))$. It contains k tag components of non-zero weight, where $t_{cx_i} \in T(1 \le i \le k)$ is tags cooperator c_x was involved in, $w_{cx_i} \in ct_x(1 \le i \le k)$ is the weight of the tag t_{cx_i} in the work preference vector c_x.

The cooperator work preference similarity CS can be measured by the similarity of **WPVC**. It reflects not only the similarity of collaborators' work ability but also their cooperative relationship.

The cooperators c_x and c_y are not in the same vector space, and there are various relativities between the components of the two vectors. So we define the CS of cooperator c_x and c_y as multiplying the similarity between any two components of the vectors and the corresponding weights, consistently with Equation 7.

$$CS_{xy} = \sum_{j=1}^{h} \sum_{i=1}^{k} (w_{cx_i} w_{cy_j} TS(t_{cx_i}, t_{cy_j})) \quad (7)$$

The matching degree of cooperator and project CPS can be measured by the similarity between **WPVC** and the tag vector of project.

[Definition 5] Tag Vector of Project (TVP). A **TVP** is a vector as $p_s = (t_{ps_1}, t_{ps_2}, \cdots, t_{ps_h})$, where $t_{ps_i} \in T(1 \le i \le h)$ is tags that the project p_s contains, and the weight of t_{ps_i} in p_s is $1/h$.

The cooperator c_x and the project p_s are not in the same vector space, and there are various relativities between the components of the two vectors. Like the definition of CS, we define the CPS of c_x and p_s as multiplying the similarity between any two components of the vectors and the corresponding weights, consistently with Equation8.

$$CPS_{xs} = (\sum_{j=1}^{h} \sum_{i=1}^{k} (w_{cx_i} TS(t_{cx_i}, t_{ps_j}))) / h \quad (8)$$

Whether a cooperator will participate in a project, is decided not only by the matching degree of the cooperator with the project, but also by the similarity between the cooperator and the original cooperators of the project. And these two factors are mutually reinforcing. We define the cooperator-project recommendation index as the greater of the two, consistently with Equation 9.

$$RP_{xs} = Max(CPS_{xs}, Max(CS_{xi})) \quad (9)$$

CS_{xi} is the CS of cooperator c_x and the original cooperators c_i of project p_s.

We extract $TopN$ cooperators with highest RP_{xs} as the recommendations to a project.

IV. Tag Popularity And Project Recommendation For Newcomer

Like the traditional collaborative recommendation, cooperator recommendation also has the problem of cold start. In order to solve the problem of cold start, collaborative recommendations commonly use content-based nearest neighbor search method [16]. This method finds the nearest neighbors of new project from the property perspective of the project, and predicts the ratings of new project by the average score of all the nearest neighbors. The cooperator recommendation model in the Section III of this paper fully uses the project attribute information by the use of project tag vector and cooperator work preference vector. This model can recommend cooperators for a new project, but ignore the newcomers of the community. In this section we will investigate the popularity of project tags, and recommend projects for the newcomer by applying the popularity of project tags.

Gotardo *et al.* define the resource popularity by the addition of Most Frequently Used, Most Recently Used and Total Access Time of all users [17,18], and recommend resource for users according to the resource popularity. This paper uses the number of cooperators in a project, the latest time to submit code, submit frequency

and participant duration to assess the project popularity, and measures the tag popularity by the project popularity.

Project popularity is decided by the number of cooperators (U), the latest time to submit code (L), submit frequency (F), and participant duration (D). U is the number of cooperators in a project. The more the cooperators are, the more the project is popular. L is the distance of the latest time of submitting code to current time. The short the distance is, the more popular the project is. F is the times of submitting code in a certain period. The more the times is, the more popular the project is. D is the duration of submitting code in a project. The bigger the duration is, the more popular the project is.

We divide the value of U, L, F, D into five equal portions and give them a score from 1 to 5. We define the PP of p_s as follows:

$$pp_s = \frac{U_s + L_s + F_s + D_s}{20} \tag{10}$$

The project popularity only denotes the activity of a single project, but ignores the mutually reinforcing relationship of project popularity. The relations of project tags to some extent show the relationship of the projects. The popularity of tags will reflect the development direction of the community and the hobby of the members. Tags that associate with more projects are more popular, so we can define the tag popularity as Equation 11:

$$tp_i = \sum_{s \in Di} pp_s \tag{11}$$

Di is the set of projects associated with tag t_i.

The total tag set is divided into popular tag set $PPT = (t_{p_1}, t_{p_2}, \ldots, t_{p_p})$ and non-popular tag set $NPT = (t_{np_1}, t_{np_2}, \ldots, t_{np_q})$, according to the popularity tp_i ,where $t_{p_i} \in T(1 \le i \le p)$, $t_{np_i} \in T(1 \le i \le q)$, $p + q = n$.

We define the subject degree of project p_s in any tag set A as follows:

$$PS_s^{\,A} = TagIN / TagALL \tag{12}$$

$TagALL$ is the number of tags that the project p_s contains, $TagIN$ is the number of tags that the project p_s contains and are in the tag set A.

PS_s^{PPT} is the subject degree of project p_s in the popular tag set PPT. It not only shows the popularity of a project as an individual, but also impliedly shows the mutually reinforcing of project popularity. And it can also forecast the popularity of new projects.

We recommend the projects with high PS_s^{PPT} to the new members of the community to solve the recommendation problem due to the lack of participant records of new members.

V. Demonstration

A. Data Processing

This article uses the Microsoft's open source community www.codeplex.com website as the experimental data resource. We select 2269 random projects from the site, as well as the corresponding tags, cooperators, and the record of submitting code. We delete the inactive projects (the number of submitting code is fewer than 6 and inactive cooperators (only participated in one project) from the acquired data. We finally get the total data set of 474 cooperators, 587 projects, 1505 tags. And the total data set is divided into two groups by the date of 1 January 2010, the former as an experimental data and the later as a validation data set.

In the similarity computation stage, the tag co-occurrence similarity based on project is calculated in the total data set. But the tag co-occurrence similarity based on cooperator is calculated in the experimental data sets. Some cooperators only act as "Followers" in the community, it is said that they only take part in the project discussion but not in project development. They bring little contribution to community development, so we do not consider the data of the "Followers" in calculating the tag co-occurrence similarity based on cooperator.

B. Evaluate Criteria

There are two main methods to evaluate the recommendation quality: the statistical accuracy measurement and the decision-support accuracy measurement. The Recall [19] and Precision [20] in the decision-support accuracy measurement are common evaluation criteria.

We use the Recall and Precision as the evaluation criteria of the recommendation quality:

$$Precision = \frac{Hits}{TopN}, \quad Recall = \frac{Hits}{N}$$

$Hits$ is the number of correct recommendation (excluding the original cooperators), N is the number of the actual new cooperators, $TopN$ is the number of recommendation.

The Recall and Precision contain a contradiction. Recall increases with the increasing of $TopN$, but Precision declines. Recall and Precision are equally important to evaluate recommendation system quality. Therefore, Recall and Precision are combined together as $F1$, to find the best balance between the two [21]:

$$F1 = \frac{2 * Precision * Recall}{(Precision + Recall)}.$$

C. Recommendation Result

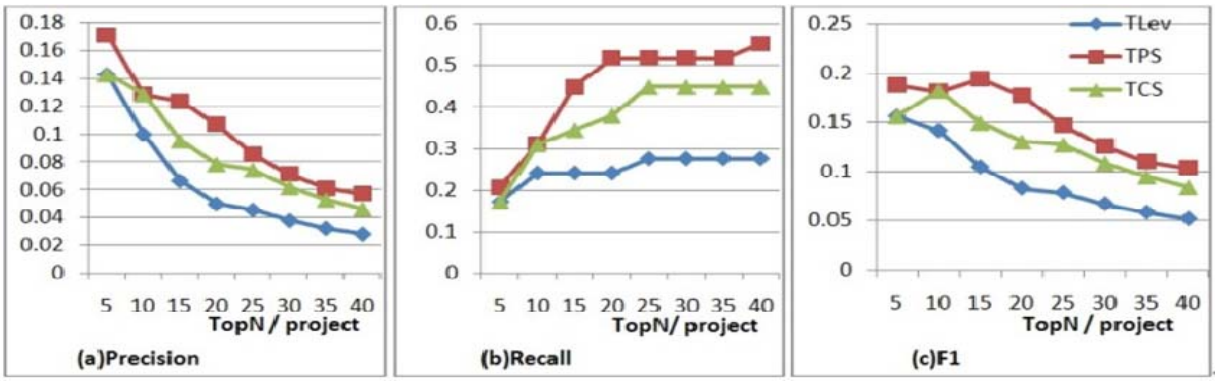

Figure 1. Separate recommendation result of *TLev,TPS,TCS* with ascending number of recommenders per project

We first validate the recommendation result of $TLev$、TPS、TCS respectively, by setting the weight in Equation1 with $(w_1 = 1, w_2 = 0, w_3 = 0)$、$(w_1 = 0, w_2 = 1, w_3 = 0)$、$(w_1 = 0, w_2 = 0, w_3 = 1)$. The result is shown in Figure 1. It can be seen that the recommendation quality of TPS is the best, and the $TLev$'s is the worst. TPS plays a dominant contribution to the recommendation algorithm, TCS follows it and the contribution of $TLev$ is the least.

In the subsequent experiment, we set different weight in Equation1. The results show that different values of w_1, w_2, w_3 all acquire a certain recommendation effect.

In order to validate the effectiveness of the proposed recommendation algorithm (call it TSCR follow), we apply the recommendation algorithm (call it TBUPR follow) provided by Hung [9] to the virtual collaborative community for cooperator recommendation. For the way to define user profile can be used in this domain, we use the Simple Cooperator-Tag Matrix and proposed WPVC as the user profile in the TBUPR, and call then TBURP-SCTM and TBUPR-WPVC. The Precision and Recall for projects with 2 new cooperators at least is given in Figure 2.

We can see that the way to define user profile strongly affect the algorithms quality. WPVC can better capture the user profile than Simple Cooperator-Tag Matrix. The recommendation results of TBUPR-WPVC and TSCR show that TSCR will perform better than TBUPR-WPVC with the increasing of recommend numbers. The TSCR's method to define tag similarity and matching degree between cooperator and project is more effective to avoid the impact of sparseness and gets an effective cooperator recommendation.

D. Verification Of Tag Popularity

To verify the tag popularity, we first calculate the popularity tp_i of every tag in the experimental data and arrange tags according to tp_i in decreasing order. Then we divide tags into N parts, respectively labeled as TC_i.

We obtain the new members set CN and the corresponding projects set PCN from the validation data set. Then we calculate the subject degree of PCN in TC_i: $PCNS_i = \sum_{j \in PCN} PS_j^{TC_i}$. Results are shown in Figure 3.

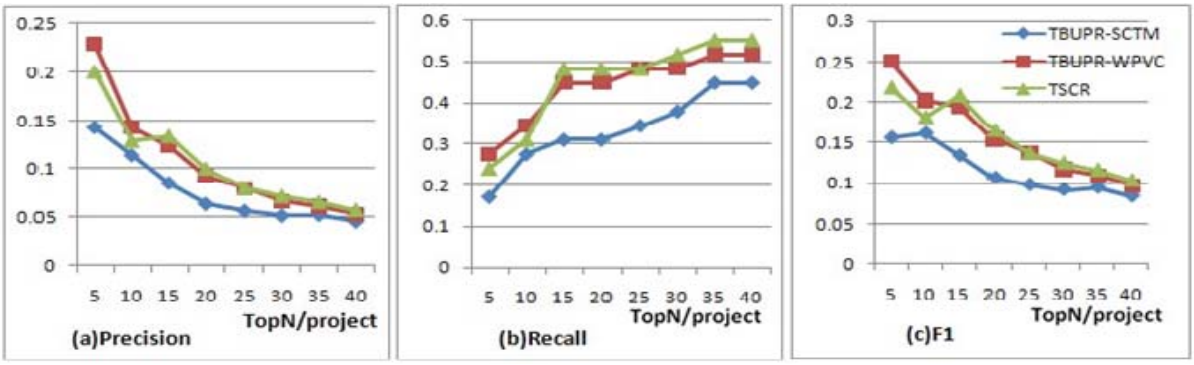

Figure 2. Recommendation result of three algorithms with ascending number of recommenders per project

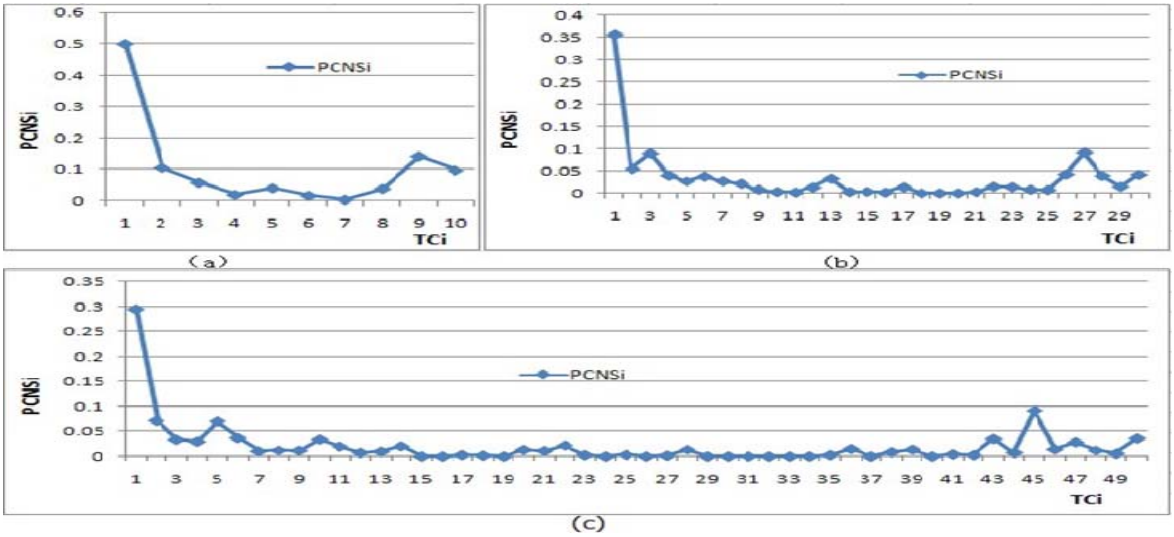

Figure 3. *PCNSi* distributing (N=10、30、50)

The distribution of $PCNS_i$ shows that new members mostly participate in projects associated with popular tags. The tail of the distribution reveals that new tags (new technologies) to a certain sense attract the new members.

In view of the above results, we choose the most popular $p = n/10$ tags as PPT set, and the others as NPT set. We recommend the highest subject degree $TopN$ projects in the set of PPT and NPT to new members separately, and compare the results with three criteria:

$$pPrecision = \frac{PHits}{TopN}, \quad cRecall = \frac{CHits}{CN},$$

$$jValidity = \frac{JHits}{JN}.$$

$TopN$ is the number of recommended projects. $PHits$ is the number of recommended projects accepted by the new members. CN is the number of new members of the community. $CHits$ is the number of new members who participated in the recommended projects. JN is the participation number of new members(if a member participates in n projects, then his/her participation number is n). $JHits$ is the participation number of successful recommendation.

$pPrecision$ is the probability of recommended projects to be accepted by new members. $cRecall$ is the probability of new members participating in recommended projects. $jValidity$ is the participation rate of new members in the recommended project.The results are shown in Figure4.

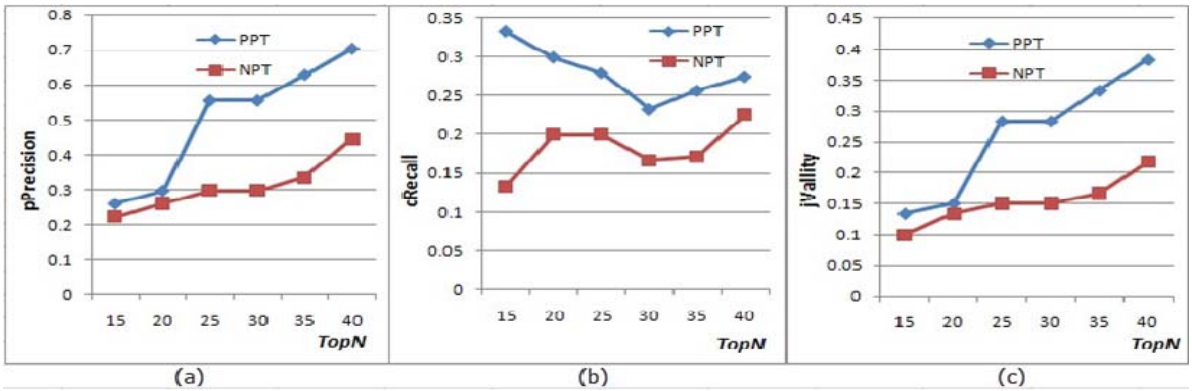

Figure 4. Recommendation result of *PPT* and *NPT*

We can see that new members of the community participate more in projects recommended from PPT than NPT. The subject degree of project in popular tag set predicts the project choice of new members to a certain extern.

In sum, the tag popularity plays a guiding role in the project selection of the new members of the community. It can be used in cooperator recommendation.

VI. Conclusions And Future Works

This paper introduces the method of measuring the similarity of work preference between cooperators and calculating the matching degree between cooperators and projects by analyzing the similarity of tags. The acquired information is then used for cooperator recommendation,

avoiding the influence of the sparsity of the Project-Cooperator Matrix in recommendation. Moreover, this paper proposes the method of analyzing the popularity of tags and defining the subject degree of project in popular tag set. Then we recommend projects to new members of the community by the subject degree.

Experiment results show that the proposed recommendation algorithm gets a good recommendation effect. So it can be used in the cooperator recommendation system. Traditional collaborative recommendation can't make recommendation for the new projects and members of the community due to the lack of the participant records. The proposed algorithm makes full use of the tag information, so it can make recommendation for all of the projects and members of the community.

During the experiment, we found that a lot cooperator with high RP_{xs} are not involved in the project in the later work. It reveals that the original work preferences did not play a key role in the choice of projects for some cooperators. Our follow-up research will further study the effect of work preferences on the choice of projects and the evolution of the work preferences of cooperators.

ACKNOWLEDGMENT

This research was supported by the Natural Science Fund of China (No. 71102111).

REFERENCES

[1] G. Concas, M. Lisci, S. Pinna, "Open Source Communities as Social Networks: an analysis of some peculiar characteristics," *19th Australian Conference on Software Engineering*, IEEE Press, 2008,pp. 387-391.

[2] S.L. Toral, M.R. Martínez-Torres, F. Barrero, "Analysis of virtual communities supporting OSS projects using social network analysis," *Information and Software Technology*, vol. 53,2010, pp. 296-303.

[3] L. Korba, R. Song, G. Yee, A. Patrick, "Automated social network analysis for collaborative work," *The Third International Conference on Cooperative Design, Visualization, and Engineering(CDVE)*. Springer, 2006, pp.1-8.

[4] P. Massa, P. Avesani, "Trust-aware collabortative filtering for recommender system," *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, 2004, pp. 492-508.

[5] M. Van Antwerp, G. Madey, "The Importance of Social Network Structure in the Open Source Software Developer Community," *Proceedings of the 43rd Hawaii International Conference on System Sciences*, IEEE Press, 2010, pp.1-10.

[6] P. Wu, S.K. Li, "Social Network Analysis Layout Algorithm under Ontology Model," *Journal of Software*, vol. 6, no. 7,2011, pp.1321-1328.

[7] K. Xu, W. Cui, J. Tie, X.F. Zhang, "An Algorithm for Detecting Group in Mobile Social Network," *Journal of Networks*,Vol 7, No 10,2012,1584-1591.

[8] K.H.L. Tso-Sutter, L.B. Marinho, L.Schmidt-Thieme, "Tag-aware recommender systems by fusion of collaborative filtering algorithms," *Proceedings of the 2008 ACM symposium on Applied computing*, ACM, 2008, pp. 1995-1999.

[9] C.C. Hung, Y.C. Huang, J.Y. Hsu, et al., "Tag-Based User Profiling for Social Media Recommendation," *In Workshop on Intelligent Techniques for Web Personalization & Recommender Systems at AAAI2008*, 2008.

[10] W.F. Pan, S. Li, "Tag Ontology Automatic Building for Semantic Searching of Services: a Case Study on Mashup Services," *Journal of computers*, ,vol. 7, no. 12, 2012, pp.2979-2986.

[11] B. Markines, C. Cattuto, D. Benz, et al., "Evaluating Similarity Measures for Emergent Semantics of Social Tagging," *Proceedings of the 18th International Conference on World Wide Web*, ACM, 2009, pp. 641-650.

[12] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals ," *Soviet Physics Doklady*, vol. 10, no. 8, Feb , 1966, pp.707–710.

[13] D. Gusfield, *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997.

[14] S. Rane and W. Sun, " Privacy Preserving String Comparisons Based on Levenshtein Distance," *IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE Press, 2010, pp.1-6.

[15] S.M. Pi, H.L. Liao, S.H. Liu, C.W. Lin, "Content Classification by folksonomies: Framework of Social Bookmarking System," *Journal of software*, vol.7, no.4, 2012, pp.741-744.

[16] B.M. Kim, Q. Li, C.S. Park, et al., "A new approach for combining content-based and collaborative filters," *Journal of Intelligent Information Systems*, vol.27, no.1, 2006, pp.79-91.

[17] R.A. Gotardo, C.A.C.Teixeira, S.D. Zorzo, "Ip2 model—content recommendation in web-based educational systems using user's interests and preferences and resources' popularity," *Proceedings of the 2008 32nd annual IEEE international computer software and applications conference*, IEEE Computer Society, 2008, pp.460–463

[18] R. A. Gotardo, "An Approach to Recommender System in Web-based Educational Systems using Usage Mining to Predict User s Interests," *The 15th International Conference on Systems, Signals and Image Processing*, 2008, pp.571-580.

[19] Z. Chedrawy, S.S.R. Abidi, "An adaptive personalized recommendation strategy featuring context sensitive content adaptation," *Proceedings of Adaptive Hypermedia and Adaptive Web-based Systems*, Springer, 2006, pp.61-70.

[20] K. Goldberg, T. Roeder, D. Gupta, "Eigentaste: a constant time collaborative filtering algorithm," *Information Retrieval*, vol.4, no.1, 2011, pp.133-151.

[21] L.T. Weng, Y. Xu, Y.F. L, et al., " An improvement to collaborative filtering for recommender systems," *International Conference on Computational Intelligence for Modelling, Control &Automation Jointly with International Conference on Intelligent Agents, Web Technologies &Internet Commerce*, IEEE, 2006, PP.792-795.

Xiang Chen was born in Jiangxi, China, in March 1976. He received the B.S. degree in computer science and application in 1997 from Changchun Institute of Optics, Fine Mechanics and Physics, Changchun, China. He got the M.S. degree in computer application in 2000 from Kunming University of Science and Technology, Kunming, China, and the Ph.D. degree in management science & engineering in 2003 from Beihang University, Beijing, China.

In 2003 he joined the School of Management & Economics, Being Institute of Technology, where he is presently an Associate Professor, Director of Master of Engineering Center in Being Institute of Technology. From November 2009-May 2010, he was a visiting scientist in Karlsruhe Institute of Technology, German, supported by Federal Ministry for Research and Technology. He has published over 40 journal papers and is the author or co-author of 4 books, including Database System Concepts, Electronic Business and Project Management. As a mater advisor, he has graduated 42 Masters since 2005.

Dr. Chen is also a member of the China Computer Federation. He served on the Program Committees of the Tenth Workshop on E-Business, 2011 International Conference on Information Systems (ICIS 2011), Shanghai, China, and the 6th China Summer Workshop on Information Management (CSWIM 2012), Beijing, China.