On Finding Fine-Granularity User Communities by Profile Decomposition

Seulki Lee, Minsam Ko, Keejun Han, Jae-Gil Lee*
Department of Knowledge Service Engineering
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
{seulki15, minsam.ko, brianhan87}@gmail.com, jaegil@kaist.ac.kr

***Abstract*—The social network represents various relationships between users, and community discovery is one of the most popular tasks analyzing these relationships. The relationships are either explicit (e.g., friends) or *implicit*, and we focus on community discovery with implicit relationships. Here, the key issue is how to extract the relationships between users. A user is typically represented by his/her *profile*, and the similarity between user profiles is measured. In most algorithms, a user has a single profile aggregating all the information about the user. For example, a profile for a researcher is a list of papers he/she wrote. This setting, however, oversimplifies the multiple characteristics of a man since individual characteristics are mixed up. In this paper, we propose the notion and method of *profile decomposition*, which divides a profile into a set of sub-profiles so that they represent individual characteristics precisely. Then, we develop a community discovery algorithm, which we call *DecompClus,* based on profile decomposition. Using a real data set of CiteULike, we show that our proposed algorithm can precisely distinguish multiple research interests of a user and discover communities corresponding to each interest, whereas previous algorithms cannot. Overall, profile decomposition enables us to find fine-granularity user communities, thus improving the accuracy of community discovery.**

Keywords-community discovery; social network; user profile; profile decomposition

I. Introduction

As interactions between users are growing every day on social network services, a considerable amount of attention has been devoted to analysis of social networks. In particular, community discovery is one of the most popular tasks in social network analysis [10] and has many real-world applications such as advertisement to common interest groups and recommendation of potential collaborators in workplaces. Community discovery is being widely used in other disciplines such as politics [3] and sociology [1].

A social network is modeled as a huge graph, where a node is a user and an edge is a relationship between users. The edge can be added using explicit links specified by users themselves, e.g., friends in Facebook and followers in Twitter. On the other hand, the edge can be added using implicit relationships between users. For example, in researcher networks, the researchers who have similar research interests but have not known each other can be connected. We note that implicit relationships are more common than explicit ones since users do *not* know all of other related users. Thus, we focus on community discovery with *implicit* relationships.

When dealing with implicit relationships, the key issue is how to extract the relationships between users. A user is typically represented by his/her *profile*, and the similarity between user profiles is measured. The form of the profile depends on the social network and application. For example, in researcher databases such as DBLP, the profile is a list of papers he/she wrote; in Twitter, the profile can be a list of tweets he/she posted.

In most algorithms for extracting implicit relationships, a user is described by only a *single* profile aggregating all the information about the user. For example, a user is simply described by the entire list of his/her papers or tweets. This setting, however, oversimplifies the multiple characteristics of a man since individual characteristics are mixed up. Although some groups of users share a common characteristic, they might not be connected to one another if the *overall* similarities are not very high. This problem results in loss of meaningful communities.

Example 1. Suppose there are two Twitter users as in Figure 1. User A has written tweets about "photography" and "hiking," but User B about "photography" and "art." The representative terms for each user are shown as well. Although they share an interest "photography," they are not connected since the overall similarity between them is low.

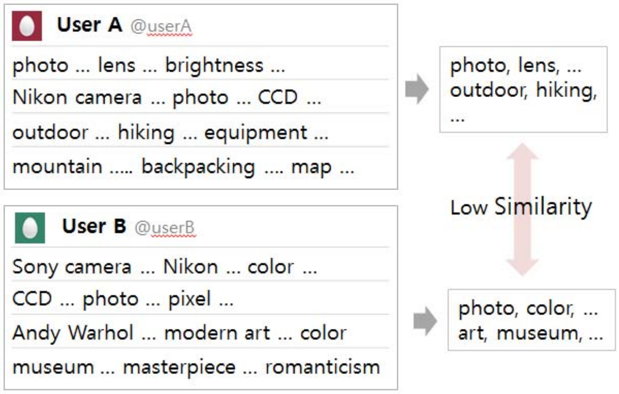
User A @userA
photo ... lens ... brightness ...
Nikon camera ... photo ... CCD ...
outdoor ... hiking ... equipment ...
mountain ..... backpacking .... map ...
photo, lens, ...
outdoor, hiking,
...
Low Similarity
User B @userB
Sony camera ... Nikon ... color ...
CCD ... photo ... pixel ...
Andy Warhol ... modern art ... color
museum ... masterpiece ... romanticism
photo, color, ...
art, museum, ...

Figure 1. Example of two Twitter users

[*] Jae-Gil Lee is the corresponding author.

978-0-7695-4799-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ASONAM.2012.106

In this paper, we propose the notion and method of *profile decomposition*, which divides a profile into a set of *sub-profiles* so that they represent individual characteristics precisely. Here, a sub-profile corresponds to each individual characteristic (e.g., interest). Then, we develop a community discovery algorithm, which we call *DecompClus*, based on profile decomposition. The algorithm consists of two steps: profile decomposition and sub-profile clustering. In Example 1, profile decomposition finds the two sub-profiles for each user, and sub-profile clustering groups User A and User B through the common sub-profile "photography."

The two steps are designed in a consistent manner and are both performed using a network clustering technique. In both steps, a network is constructed, and then a clustering algorithm is applied to the network. In the first step, a network of unit items (e.g., papers or tweets) is constructed *for each user*, and then clustering is performed on the small network. Here, each cluster becomes a sub-profile. In the second step, a network of sub-profiles is constructed by accumulating sub-profiles from every user, and then clustering is performed on the large network. Now, each cluster becomes a user community.

The final results are *overlapping* communities. Since, in the sub-profile network, multiple nodes can be generated from one user, *non*-overlapping communities of sub-profiles, in fact, imply overlapping communities of users. This is a good point in that we can adopt any of non-overlapping community discovery algorithms which are more prevalent than overlapping community discovery algorithms. In addition, it is known that overlapping community discovery algorithms [6][12] have higher computational cost.

Using a real data set of CiteULike, we evaluate the effectiveness of DecompClus by comparing with a baseline algorithm that does not perform profile decomposition. Our algorithm is shown to discover additional communities which cannot be discovered by the baseline algorithm owing to the mix of multiple research interests. This means that a user's minor interests are *not* assimilated into his/her major interests. Moreover, the users within the communities discovered by our algorithm are more tightly related than those grouped by the baseline algorithm.

In summary, the contributions of this paper are as follows:

- We propose a novel concept of profile decomposition, which enables us to detect fine-granularity user communities with implicit relationships.
- We present a new approach to discovering overlapping communities with non-overlapping community discovery algorithms.
- We demonstrate, by using a real data set, that our algorithm effectively discovers user communities from social media data.

The rest of the paper is organized as follows. Section 2 overviews our approach. Section 3 explains our community detection algorithm DecompClus. Section 4 evaluates the effectiveness of our algorithm. Section 5 discusses related work. Finally, Section 6 concludes this study.

II. Overview

A. Problem Statement

In this paper, we address the problem of community discovery in social networks which consist of users who have multiple characteristics and implicit relationships with each other. We propose a novel algorithm for community discovery that reserves multiple characteristics of users by profile decomposition. Our algorithm receives a set of users attached with their profile as input and returns relevant communities to each user.

The definitions of the main concepts for our algorithm are as follows. TABLE I. summarizes the notation used throughout this paper.

- A user's *profile* is a set of items and is denoted by P_u. Most of existing algorithms *directly* use the profile for community discovery, whereas our algorithm does not.
- An *item* reflects a user's preferences and interests. For example, a user's tweet or his/her bookmarking web page can be regarded as an item. This item in a user's profile is represented by a term vector which consists of representative terms about the item and their weight. The term vector for the item *i* is denoted by $\overrightarrow{T_\iota}$.
- A *sub-profile* is a subset of a profile, i.e., a set of items. A profile is divided into multiple sub-profiles representing each of specific aspects of the user's characteristics. A sub-profile is represented by a term vector in the same way as an item is.
- A *community* is a set of users and is denoted by C_k. Every community should have at least two users as members. The final output is the set of communities.

TABLE I. NOTATION

Symbol	Description
$\overrightarrow{T_\iota}$	Term vector for an item *i* $\overrightarrow{T_\iota} = (w_{i,1}, w_{i,2}, \dots , w_{i,t})$ $w_{i,t}$ is the weight of a term *t* in an item *i*
P_u	User *u*'s profile $P_u = \{ i_1, i_2, \dots, i_{\lvert P_u \rvert} \} = \{ p_{u,1}, p_{u,2}, \dots, p_{u,s} \}$
$p_{u,s}$	User *u*'s *s*th sub-profile $p_{u,s} = \{ i_1, i_2, \dots, i_{\lvert p_{u,s} \rvert} \}$ $\overrightarrow{p_{u,s}} = (w_{u,s,1}, w_{u,s,2}, \dots , w_{u,s,t})$ $w_{u,s,t}$ is the weight of a term *t* in a user *u*'s *s*th sub-profile
C_k	*k*th discovered community $C_k = \{ u_1, u_2, \dots, u_{\lvert C_k \rvert} \}$ $\lvert C_k \rvert > 1$

B. Overall Procedure

Our algorithm consists of two steps: *profile decomposition* and *sub-profile clustering*. The first step is, for each user, to group items in accordance with their topic and to extract representative terms of each group. These groups are the sub-profiles of a user. The second step is, given all the sub-profiles, to measure similarities between them and to group similar sub-profiles. If two different users' sub-profiles are similar with each other, the two sub-profiles (i.e., users) are likely to belong to the same group. Finally, our algorithm completes with finding communities of each user by replacing users' sub-profiles with users (owners).

Figure 2 describes the overall procedure of our algorithm. In this figure, the user A has seven items in his/her profile. After profile decomposition, the user is split into two sub-profiles that consist of similar items. The sub-profile A-1 consists of i_1, i_2, i_3, and i_4, and the sub-profile A-2 consists of i_5, i_6, and i_7. Given these sub-profiles of all users, through sub-profile clustering, the user A's two sub-profiles are respectively grouped with other users' similar sub-profiles. Community discovery is finished by placing a user on his/her sub-profiles. Finally, the user A is a member of two communities which well match with his/her interests.

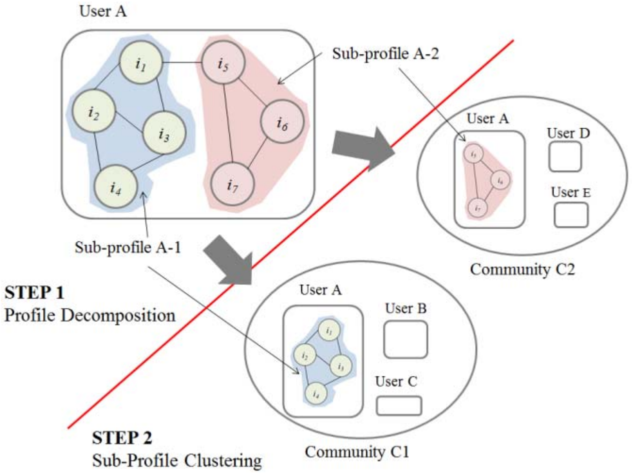

Figure 2. Overall procedure of our algorithm

III. DECOMPCLUS ALGORITHM

A. Step1: Profile Decomposition

In this step, similar items in a user's profile are grouped together according to their topic, and sub-profiles of the user are built by extracting representative terms of each sub-profile.

First, in order to group a user's items by the topic, it is necessary to calculate the similarity between items. An item can be any textual data including a tweet and a web page. The similarity between two items is calculated by comparing their term vectors. A term vector consists of terms about an item and their weight. The terms are weighted by TF-IDF measure. The TF-IDF value of a term t in an item i is calculated by Equation (1):

$$w_{i,t} = tf(i,t) \times \log(\frac{|I|}{df(t)}) \quad (1\,)$$

where *tf(i, t)* is term frequency, i.e., how often the term t appears in the item i, *df(t)* is document frequency, i.e., how many items contain the term t, and $|I|$ is the total number of items.

As a measure of the similarity, we use the cosine similarity widely used in information retrieval. It is calculated by Equation (2). The cosine similarity value ranges from 0 to 1, and a higher value means that two items are more similar. If two items do not have any common term, the cosine similarity value is zero.

$$\text{Similarity}(\overrightarrow{T_1}, \overrightarrow{T_2}) = \frac{\sum w_{1,t} w_{2,t}}{\sqrt{\sum w_{1,t}^2} \sqrt{\sum w_{2,t}^2}} \quad (2)$$

Next, a network clustering technique groups similar items. Any state-of-the-art clustering algorithm can be adopted. We adopt a clustering algorithm based on modularity optimization [2], which tries to detect high modularity partitions of networks. This algorithm is widely used to detect non-overlapping communities. In comparison with other algorithms, it is fast and guarantees good quality of discovered communities. TABLE II. shows the pseudo code of this algorithm.

TABLE II. PSEUDO CODE OF MODULARITY-BASED CLUSTERING

Input	N	a set of nodes for grouping $N = \{n_1, n_2, \ldots, n_{\lvert N\rvert-1}, n_{\lvert N\rvert}\}$
	M	a matrix of similarity values between nodes $m_{i,j}$= similarity between a node i and a node j
Output	G	a set of groups of nodes $G = \{g_1, g_2, \ldots, g_{\lvert G\rvert-1}, g_{\lvert G\rvert}\}$
Function		
$neighbors(n)$:		return the neighbors of a node n
$group(n)$:		return the group of a node n
$\Delta Q(n, g, M)$:		return the gain in modularity of M obtained by moving a node n into a group g
$sumOfSim(g, h)$:		return the sum of similarities between nodes in a group g and nodes in a group h

```
Procedure
$G = \emptyset$
FOR $n$ in $N$
        put { $n$ } into $G$ // Each node is one group
REPEAT
        /* 1st step */
      FOR $n$ in $N$
                $ne^* = argmax_{ne \in neighbors(n)}\, \Delta Q(n, group(ne), M)$
                IF $\Delta Q(n, group(ne^*), M) > 0$
                        move $n$ to $group(ne^*)$
        /* 2nd step */
        $N = \{g_1, g_2, \ldots, g_{|G|-1}, g_{|G|}\}$        // Set groups as nodes
        $M$ = a new $|N|$ x $|N|$ matrix
        FOR $g$ in $G$
                For $h$ in $G$
                        $m_{g,h} = sumOfSim(g, h)$
UNTIL $G$ does not change
RETURN $G$
```

The modularity-based algorithm iteratively attempts to find community structures having the highest modularity. First, it starts with placing every node into a different group. When putting a node into the groups of its neighbor nodes, the algorithm calculates the gain in modularity. If there is any positive change in the modularity value, the node moves into the group with the largest gain in modularity. Otherwise, the node remains in its current group. The gain in modularity which is calculated by adding a node n into a group g can be computed by Equation (3).

$$\Delta Q(n, \mathrm{g}, M) = \left[\frac{\sum_{g,in} m + 2S_{g,n,in}}{2\sum_{tot} m} - \left(\frac{\sum_{g,tot} m + 2S_{g,n}}{2\sum_{tot} m}\right)^2\right] - \left[\frac{\sum_{g,in} m}{2\sum_{tot} m} - \left(\frac{\sum_{g,tot} m}{2\sum_{tot} m}\right)^2 - \left(\frac{S_{g,n}}{2\sum_{tot} m}\right)^2\right] \quad (3)$$

where $\sum_{g,in} m$ is the sum of the weights of the links inside the group g, $\sum_{g,tot} m$ is the sum of the weights of the links incident to the nodes in the group g, $S_{g,n}$ is the sum of the weights of the links incident to the node , $S_{g,n,in}$ is the sum of the weights of the links from the node n *t*o the nodes in the group g, and $\sum_{tot} m$ is the sum of the weights of all the links in the network.

After clustering the items, a user has multiple groups of items. Each group of items represents a sub-profile reflecting a user's interest on a specific topic. There is only one exception. If a group has only one item as a member, it is not considered as the user's sub-profile and eliminated since one item is not enough to reflect his/her interest.

B. Step2: Sub-Profile Clustering

Our algorithm groups all users' sub-profiles and completes community discovery by transforming sub-profiles to users. This step is performed in a similar way to the previous step.

A term vector of each sub-profile is first constructed. The list of terms about a sub-profile is obtained by fetching all the textual information of all the items in the sub-profile. We also use TF-IDF to calculate the term weights of the sub-profile and calculate similarity between sub-profiles by the cosine similarity. Next, the same modularity-based clustering algorithm is used for grouping these sub-profiles.

Finally, in order to complete community discovery, it transforms sub-profiles to users for each group (i.e., community). TABLE III. shows the pseudo code of transforming sub-profiles to users. It simply replaces sub-profiles with their owner and removes duplicate users. Since a user has multiple sub-profiles, a user can belong to multiple communities.

TABLE III. PSEUDO CODE OF TRANSFORMING SUB-PROFILES TO USERS

```
Input     PG    a set of groups of all users' sub-profiles
                $PG = \{PG_1, PG_2, \ldots, PG_{k-1}, PG_k\}$
Output    C     a set of communities
                $C = \{C_1, C_2, \ldots, C_{k-1}, C_k\}$
Procedure
$C = \emptyset$
FOR $k$=1 to $|PG|$
    $C_k = \emptyset$
    FOR $p_{u,s}$ in $PG_k$
        $C_k = C_k \cup \{u\}$
    $C = C \cup \{C_k\}$
RETURN $C$
```

IV. EVALUATION

In this section, we explain the results of our experiments to evaluate the performance of DecompClus. Our evaluation is two-fold. First, we *quantitatively* verify that DecompClus finds more tightly and well-connected communities. Second, we *qualitatively* verify that DecompClus discovers more natural communities by investigating representative terms of each community.

A. Experimental Set-Up

1) Data Set

CiteULike[1] is a website that offers social bookmarking service for scholarly papers. Like other social bookmarking websites, it allows users to create a bookmark for an academic reference and to attach tags to their bookmarks. In this web service, a user's bookmark list represents the user's research interests although the user did not write the paper. Thus, this bookmark list is used as a profile.

CiteULike offers their database for research purposes. We downloaded tagging data which reflects who posted what items with which tags. To simplify the experiments, we selected the data for the years from 2010 to 2011. For the purpose of easier interpretation of user interests, we extracted the users who have specific interests related to "social network," "data mining," or "recommendation" by identifying users who frequently use related tags to those areas. The sets of related tags we used are shown in TABLE IV. . The first set is a list of tags related to "data mining." The second set is about "social network." The third set is about "recommendation."

TABLE IV. TAGS USED FOR USER SAMPLING

Topic	**Set 1**	**Set 2**	**Set 3**
	Data mining	***Social network***	***Recommendation***
Tags	tag like 'data_mining%' or tag like 'mining%' or tag like 'knowledge_discovery%'	tag like 'social_network%' or tag like 'socialnetwork%'	tag like 'recommend%'

We selected 50 users who most frequently use those tags for each set. Here, we did not include the users who had posted more than 500 articles. If some users have written too many articles, they are likely to be spammers. The heaviest user from 2010 to 2011 had posted 34,254 articles, which is obviously an abnormal usage pattern. Theses heaviest users account for 1 percent of the total number of users in our data set.

Last, we merged the three sets of users. The total number of users in the data set is 122. The total number of articles is 25,089; and the total number of unique stemmed tags is 16,161. Half of the users used the tags related to at least two topics among the above topics. This indicates that half of the users have more than one interest. Figure 3 is the diagram showing the distribution of the users in our data set.

2) Implementation

We used Gephi [18] which is open-source software for visualizing and analyzing large network graphs. The modularity-based clustering function provided by Gephi detects communities in a given network and returns the modularity values of the communities. We used this function for profile decomposition and sub-profile clustering.

[1] http://www.citeulike.org

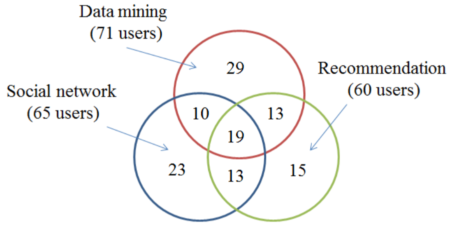

Figure 3. Distribution of users who used tags related to each topic

3) Baseline

For the comparison of our algorithm, we implemented a conventional algorithm using only one overall profile for a user. We make this algorithm follow almost the same procedures only except that it does not perform profile decomposition. As the result, a group of profiles forms a community. By doing so, we expect to see how effective profile decomposition is. We call the baseline algorithm *Baseline*.

B. Discovered Communities

TABLE V. shows the number of communities discovered by the two algorithms and the number of members that belong to each community. DecompClus finds more communities than Baseline does. In total, 2 communities are discovered by Baseline, and 4 communities by DecompClus. In addition, the allocation of users to the communities is different in the two algorithms. Because our algorithm divides a user node into multiple sub-profile nodes, it is possible that a user belongs to multiple communities at the same time. Therefore, the discovered communities by our algorithm have higher numbers of members.

TABLE V. NUMBER OF USERS IN DISCOVERED COMMUNITIES

Community ID	Baseline	Community ID	DecompClus
Bc1	57	Dc1	80
Bc2	65	Dc2	53
		Dc3	91
		Dc4	84

C. Quantitative Evaluation

We use three criteria for quantitative evaluation as follows:

- Modularity: It evaluates the structure of communities. If the modularity value is higher, it indicates that the members in that community are structurally well-grouped.
- Intra-similarity: It is the average value of edge weights in a community. If the value of this measure is high, it indicates that the members in the community are densely located.
- Inter-similarity: It is the average value of similarities between communities. If the value of this measure is low, it indicates that the communities are far from each other.

DecompClus achieves much higher modularity than Baseline (see TABLE VI.). This indicates that the members in that community are structurally well-grouped.

TABLE VI. MODULARITY VALUE

	Baseline	DecompClus
Modularity	0.0035	**0.0734**

The strength of DecompClus is also shown in the comparison results of intra-similarity and inter-similarity. TABLE VII. shows the intra-similarity of the users within each community, and TABLE VIII. shows the inter-similarity between the communities. DecompClus shows higher intra-similarity and lower inter-similarity in comparison with Baseline. This indicates that, in DecompClus, the connections between the members within a community are denser; in contrast, the connections between the members in different communities are sparser.

TABLE VII. INTRA-SIMILARITY

	Baseline		DecompClus
Avg.	**0.0133**	**Avg.**	**0.0279**
Bc1	0.0168	Dc1	0.0261
Bc2	0.0097	Dc2	0.0343
		Dc3	0.0216
		Dc4	0.0297

TABLE VIII. INTER-SIMILARITY

	Baseline		DecompClus
Avg.	**0.4534**	**Avg.**	**0.3604**
Bc1 – Bc2	0.4534	Dc1-Dc2	0.3929
		Dc1-Dc3	0.1454
		Dc1-Dc4	0.3482
		Dc2-Dc3	0.3586
		Dc2-Dc4	0.2722
		Dc3-Dc4	0.3749

D. Qualitative Evaluation

As well as the quantitative analysis, the quality of the communities found by DecompClus also needs to be evaluated. To this end, the 10 most representative stemmed tags are chosen from each community in the decreasing order of *tf* values (see TABLE IX. and TABLE X.); and we manually defined the common theme of those terms. Then, we can figure out how topics are distributed among the communities and explain how the communities discovered by DecompClus and those by Baseline are different semantically. In TABLE IX. and TABLE X. , the terms in square bracket show the possible original tags before stemming is applied.

All of the communities found by Baseline are also found by DecompClus. This implies that DecompClus preserves the themes defined by Baseline. Bc1 corresponds to Dc1; Bc2 is similar to Dc4. Please note that these pairs share several common tags marked in bold.

On the other hand, Dc2 and Dc3 are new communities that are not found by Baseline. The theme of Dc2 is "semantic web." The theme of Dc3 is "data mining & bioinformatics." The 10 most representative stemmed tags of Dc3 were general terms. For clarifying the theme of this community, 20 more stemmed tags are extracted from Dc3 (see TABLE XI.). The specific terms which can define the theme of Dc3 are related to biology and marked in bold, e.g., "genom," "biology," "sequence," and "protein."

TABLE IX. REPRESENTATIVE TERMS OF DISCOVERED COMMUNITIES BY BASELINE

ID	Theme	Representative Terms
Bc1	Data mining & Recommendation	stat [statistics], **model** [modeling], **dat min** [data-mining], **clust** [clustering], **survey** [survey], network [network], **algorithm** [algorithm], **recommend** [recommender], analys [analysis], ontolog [ontology]
Bc2	Social network	**soc network** [social_networks], stat [statistics], **network** [network], **clust** [clustering], search [search], model [modeling], **commun** [community], web [web], vis [visualization], mot [motifs]

TABLE X. REPRESENTATIVE TERMS OF DISCOVERED COMMUNITIES BY DECOMPCLUS

ID	Theme	Representative Terms
Dc1	Data mining & Recommendation	**survey** [survey], **dat min** [data-mining], machin learn [machine_learning], **recommend** [recommender], **clust** [clustering], class [classification], recommend system [recommender_systems], collab filt [collaborative-filtering], **model** [modeling], **algorithm** [algorithm]
Dc2	Semantic web	ontolog [ontology], sem [semantics], sem web [semantic-web], web [web], inform retriev [information_retrieval], context [context], tag [tagging], inform [information], evalu [evaluation], recommend [recommender]
Dc3	Data mining & Bioinformatics	stat [statistics], model [modeling], method [methods], analys [analysis], research [research], scy [scientific], dat [data], gen [generation], algorithm [algorithm], the [theory]
Dc4	Social network	**network** [network], **soc network** [social_networks], soc [social], graph [graph], **commun** [community], recommend [recommender], evolv [evolution], **clust** [clustering], soc network analys [social_network_analysis], diffus [diffusion]

TABLE XI. REPRESENTATIVE TERMS OF DC3

Representative Terms
stat [statistics], model [modeling], method [methods], analys [analysis], research [research], scy [scientific], dat [data], gen [generation], algorithm [algorithm], the [theory], educ [education], vary [variable], inform [information], hum [human], evolv [evolution], comput [computation], **databas [databases]**, of [of], interact [interactions], system [system], rat [rational], met [met], **genom [genomics], and [and], sequ [sequencing]**, web [web], hist [historical], vis [visualization], **predict [predictive]**, **genet [genetic]**

Now, let's look into why Dc2 and Dc3 are not discovered by Baseline. To this end, we retrieved the articles having the tags related with "semantic web" and "bioinformatics," respectively. Then, we distributed these articles to the discovered communities according to the membership of the users who wrote tags to the articles. Figures 4 and 5 show their distribution.[2] Please note that, in Baseline, the articles are evenly distributed to the two communities. This is because the topics "semantic web" and "bioinformatics" are not properly considered in measuring user similarity due to the limitation of community discovery using Baseline: it only allows a single user profile. In contrast, our algorithm, DecompClus, is able correctly to identify these topics. We can observe that the corresponding articles are concentrated in one community in DecompClus.

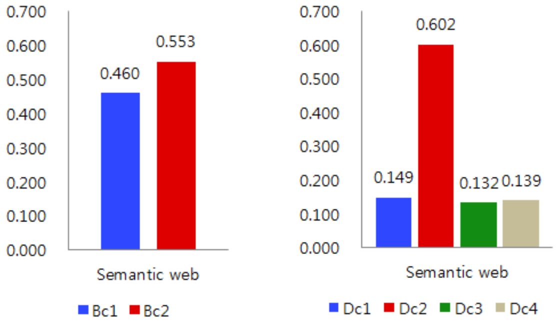

Figure 4. Distribution of articles related to "semantic web"

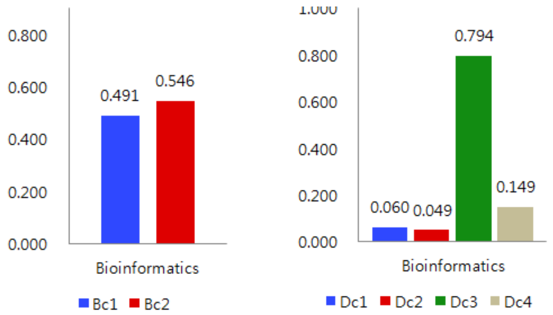

Figure 5. Distribution of articles related to "bioinformatics"

E. Case Studies

Next, we investigated two cases of users whose communities are detected differently in our algorithm. These cases strongly support our statement that DecompClus discovered fine-granularity user communities.

1) Case 1: Users who become a member in multiple communitiies by profile decomposition

The users in this case have at least two interests. In Baseline, these users are grouped only with their single biggest interest. Other interests of the users are overlooked. In contrast, DecompClus allows a user to be in more than one community by splitting their single profile into several sub-profiles. For example, a user *A*'s profile contains the terms related to "data mining & recommendation," "semantic web," and "social network" (see TABLE XII.).

[2] The sum of the portions is greater than 1 since an article can be tagged by multiple users from different communities.

TABLE XII. USER A'S PROFILE CONTAINING TERMS RELATED TO MULTIPLE TOPICS

Topic	Terms
Data mining & Recommendation	user model, recommender, personalization, user profiling, knn, data mining …
Semantic web	semantics, semantic web, rdf, ontology, social semantic web …
Social network	social network analysis, social search, graphs, …

The user *A* belongs to only one community Bc1 (data mining) in Baseline. However, through DecompClus, the user *A*'s profile is decomposed into the three sub-profiles. Hence, the user *A* can become a member of three different communities: Dc1 (data mining & recommendation), Dc2 (semantic web), and Dc4 (social network). In our data set, there are total 99 users (81.1%) like the user *A*.

2) Case 2: Users who become a member in the communities newly discovered by DecompClus

The users in this case have an interest which is not common to other users. In our algorithm, a user's minor interests are not assimilated into his/her major interests, so new communities which consist of users' minor interests can be discovered. For example, a user *B*'s profile contains the terms related to "bioinformatics" (see TABLE XIII.).

TABLE XIII. USER B'S PROFILE CONTAINING TERMS RELATED TO NEW TOPIC

Topic	Terms
Bioinformatics	statistics, genomics, sequencing, microarray, cancer, structure, bacteria, database, gene, classification, virus, proteins …

Bioinformatics is the major interest of the user *B*, whereas it is a minor interest and assimilated to "data mining" in most other users. With Baseline, the user *B* is involved in the community Bc1, i.e., "data mining." With DecompClus, the user *B* and others' sub-profiles which are related to the topic of "bioinformatics" are grouped as one community. Like the user *B*, the number of users who are a member of one community in Baseline and are only a member of one of the new communities in DecompClus is 9 (7.3%).

F. Visualization

Finally, we visualize the community structures discovered by Baseline and DecompClus. For visualization, we used ForceAtlas2 which is one of layouts provided by Gephi.

The community structure discovered by Baseline is shown in Figure 6. The nodes in the network represent users, and each node is colored differently according to its community. Overall, the network is shown as a set of weak community structures that are not easily distinguishable. Each user has multiple interests so that there are many relationships between members in one community and those in the other community.

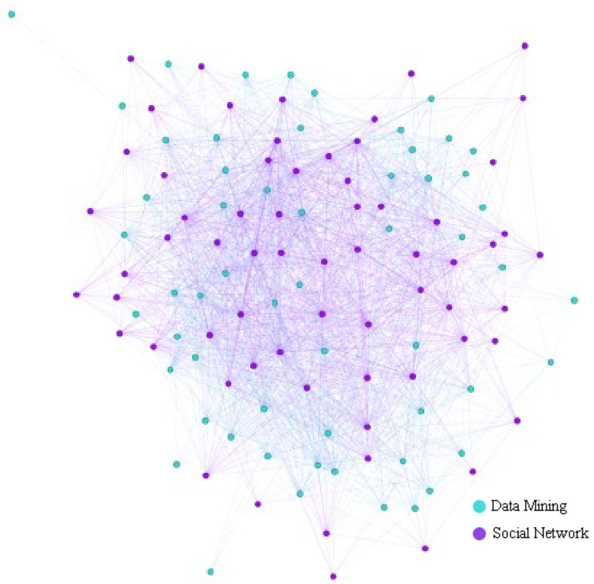

Figure 6. Community structure produced by Baseline

On the other hand, the community structure produced by DecompClus is more clearly distinguishable (see Figure 7). In Figure 7, unlike Baseline, the nodes represent users' sub-profiles. The communities about "bioinformatics" on the left, those about "social network" on the right, and those about "semantic web" at the bottom are easily identified in the network. As for the "data mining" community, its members tend to spread over the entire network. This is because data mining has many overlapping sub-topics with the other fields.

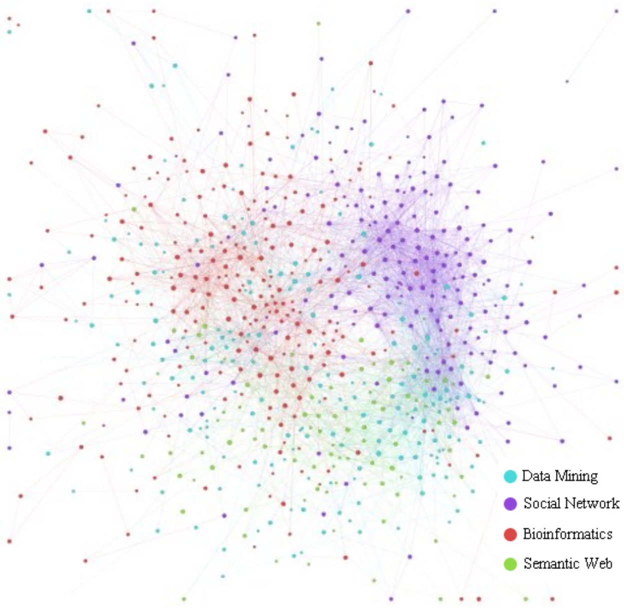

Figure 7. Community structure produced by DecompClus

V. Related Work

A. Community Discovery

Attempts to discover communities from networks have been continually made in various disciplines.

Some of the researches use the topological characteristics in the network to cluster nodes. Newman et al. [11] suggested a popular community detection method that removes inter-community edges which connect many nodes between different communities. Their method is a type of hierarchical clustering which is easy to implement. However, the computational cost becomes prohibitive if the adjacency matrix is large. Karypis et al. [8] proposed a multi-level graph partitioning method to reduce the computational cost by working on a smaller graph. In their method, the original graph is first coarsened down to a few hundred nodes.

On the other hand, there have been researches trying to combine structural relationships and content information. Zhou et al. [17] suggested an attribute augmented graph, where new nodes are introduced for each attribute value, to combine both types of information. To compute similarity between nodes, a distance measure based on random walk is computed over the augmented graph. Thus, each cluster is assumed to consist of the nodes that have similar labels. Wang [16] and Pathak et al. [13] use Bayesian generative models, incorporating textual contents into certain relationships.

However, these conventional approaches do not consider users' multiple interests. There is a limitation that a user must be a member of only one community. In contrast, in our algorithm, a user can belong to diverse communities which well match the user's interests.

B. Overlapping Communities

Researches on overlapping community discovery have been conducted in order to deal with users' multiple interests. Pallal et al. [12] proposed a community discovery method by utilizing cliques in a network. The method starts with finding cliques in a network and tries to add relevant nodes to each clique. Goldberg et al. [6] presented two community axioms: connectedness and local optimality. These two axioms are used to define communities less strict than cliques. These methods discover communities based on structural features of a network. However, sometimes, it is difficult to observe structural features such as cliques. Finding a particular structure in a network is also not an easy task. It usually requires much computation time.

Our approach is more flexible because we do not require any structural feature. In our approach, a community is defined by content features not by structural features. It can be readily applied in various networks and give better explanation about discovered communities. In addition to that, unlike traditional overlapping community discovery methods, our approach can adopt any general community discovery algorithms.

C. User Profiling

The user profile reflects a user's interests, preferences, and contexts [5]. Building a user profile is one of the essential tasks for detecting communities in implicit social networks. Because there is no explicit relationship in such networks, it is necessary to infer the relationships between users. User profiles are used to identify the relationships and determine whether two users have the same interests or not. As user profiles are more sophisticated, it can be possible to find better communities.

Researches on building user profiles have been conducted in the field of recommendation and personalization. Diverse sources are used, including users' ratings [9], users' search queries [15], click-through data [7], and user tags [14]. User profiles can be manually constructed by asking users to input their own information or can be automatically built by referring users' previous activities. Our algorithm automatically constructs user profiles by analyzing the textual items that reflect users' interests.

A few of researches tried to divide a user's overall profile into specific profiles. Dou et al. [4] proposed a personalized search method utilizing different user profiles for each search query. According to the query, each user has a different profile reflecting his/her search context.

VI. Conclusion

In this paper, we proposed a novel notion and method of *profile decomposition*. Then, we developed a community discovery algorithm *DecompClus*. Profile decomposition divides a user's profile into multiple sub-profiles, thus enabling us to find fine-granularity user communities. Using a real data set of CiteULike, we verified the benefits of fine-granularity user communities. Another interesting point is that overlapping communities can be obtained using *non*-overlapping community discovery algorithms. In addition, our algorithm is flexible in that it can adopt any state-of-the-art network clustering algorithms. Overall, we believe that our algorithm is helpful to take advantage of the potential value of implicit relationships in social networks.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2011-0029185).

References

[1] J. Baumes, M. Goldberg, M. Magdon-Ismail, and W. Wallace, "Discovering hidden groups in communication networks," in *Intelligence and Security Informatics*, vol. 3073, H. Chen, R. Moore, D. Zeng, and J. Leavitt, Eds. Springer Berlin / Heidelberg, 2004, pp. 378–389.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, vol. 2008, no. 10, p. 10008, Oct. 2008.

[3] M. D. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer, "Predicting the political alignment of twitter users," in *SocialCom*, 2011, pp. 192–199.

[4] Z. Dou, R. Song, J.-R. Wen, and X. Yuan, "Evaluating the effectiveness of personalized web search," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 8, pp. 1178 –1190, Aug. 2009.

[5] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User profiles for personalized information access," in *The Adaptive Web*,

vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer Berlin / Heidelberg, 2007, pp. 54–89.

[6] M. Goldberg, S. Kelley, M. Magdon-Ismail, K. Mertsalov, and A. Wallace, “Finding overlapping communities in social networks,” in *SocialCom*, 2010, pp. 104 –113.

[7] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, “Accurately interpreting clickthrough data as implicit feedback,” in *SIGIR*, 2005, pp. 154–161.

[8] G. Karypis and V. Kumar, “METIS-unstructured graph partitioning and sparse matrix ordering system, Version 2.0,” 1995.

[9] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, “GroupLens: applying collaborative filtering to Usenet news,” *Commun. ACM*, vol. 40, no. 3, pp. 77–87, Mar. 1997.

[10] M. E. J. Newman, “Modularity and community structure in networks,” *PNAS*, vol. 103, no. 23, pp. 8577–8582, Jun. 2006.

[11] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[12] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.

[13] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson, “Social topic models for community extraction,” in *SNA-KDD Workshop*, vol. 8, 2008.

[14] S. Sen, J. Vig, and J. Riedl, “Tagommenders: connecting users to items through tags,” in *WWW*, 2009, pp. 671–680.

[15] B. Tan, X. Shen, and C. Zhai, “Mining long-term search history to improve search accuracy,” in *KDD*, 2006, pp. 718–723.

[16] X. Wang, “Group and topic discovery from relations and their attributes,” *DTIC Document*, 2006.

[17] Y. Zhou, H. Cheng, and J. X. Yu, “Graph clustering based on structural/attribute similarities,” *PVLDB*, vol. 2, no. 1, pp. 718–729, Aug. 2009.

[18] Gephi, http://gephi.org