Extracting Communities of Interests for Semantics-based Graph Searches

Makoto Nakatsuji
NTT Cyber Solutions Laboratories, NTT Corporation, Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847 Japan
nakatsuji.makoto@lab.ntt.co.jp

Akimichi Tanaka
NTT Cyber Solutions Laboratories, NTT Corporation, Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847 Japan
tanaka.akimichi@lab.ntt.co.jp

Toru Ishida
Department of Social Informatics, Kyoto University, Yoshida Honmachi, Sakyo, Kyoto 606-8501 Japan
ishida@i.kyoto-u.ac.jp

ABSTRACT

We propose an algorithm that divides a graph, whose vertices are users and whose edges represent shared interests, into subgraphs - communities of interests (COIs). Our algorithm differs from conventional modularity-based community extraction because it introduces a mechanism to harmonize the number of users in each COI. It also attaches semantic tags to the relationships between COIs or between users based on taxonomies of the content items of interest. Our algorithm allows users to identify new interests quickly and accurately when searching their interests in the graph.

1. INTRODUCTION

By analyzing shared user interests such as music and movies, the user can find, or be presented with, content items that are likely to be interesting. Many researchers use graphs that formalize the social relationships or similarity of interests of users to extract community structures. For extracting communities from a graph, many studies use the idea of modularity[2], the number of edges falling within subgraphs minus the expected number in an equivalent subgraph with edges placed at random; communities are extracted by maximizing the modularity.

Published modularity-based community extraction methods do not, unfortunately, allow a user to employ semantics, derived from taxonomies of content items, in understanding the relationships between communities. Thus, to identify suitable communities, and thus interesting content items, user u has to check the content items of all communities in detail. This task is a heavy burden.

Furthermore, modularity-generated community structures often contain communities that contain a lot of users like B in Figure. 1, and those that contain few users like C. If we extract D from B, and then F from D, the resulting community structure has a deep but narrow hierarchy. Furthermore, F may have a lot of users. This community structure is reasonable from the viewpoint of modularity[2]. However, u who has been assigned to F may have an extremely wide variety of interests, and although F is the "best" for him according to modularity, he will want to search outside this community to fully satisfy his interests. Modularity forces the user to search many small communities, which is very ineffective, to locate new information.

This paper focuses on communities of interests (COIs) that support graph searches; a COI is defined as a semantically tagged subgraph. For this, based on the NF (Newman Faster) algorithm[2], which is a modularity-based algorithm, we propose community formation by user number harmonization; the goal is to make it easier for the user to locate new interests.

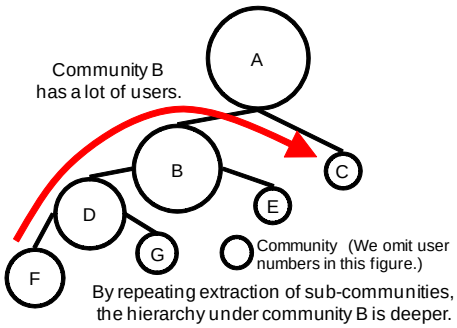

Figure 1: Problem of existing community structure.

2. METHOD

We first create a graph wherein users are vertices and edges between users represent similar interests. Several studies extract user interests by analyzing the blog entries of users and use the information to determine the similarity of user interests[1]. We assume that a graph has already been constructed based on above similarity measuring techniques.
(1) First, we apply the NF algorithm to the original graph and to any of the graphs newly created in step 2 below. We repeatedly pair subgraphs, choosing at each step the pair that yields the greatest increase in modularity function Q in NF algorithm[2]. If there are two or more pairs of subgraphs that maximize Q, we randomly choose one of them. This extracts a community structure from each graph.
(2) Identifying subgraph with the biggest user numbers.
(a) If this number is larger than N, we treat it as a new graph and return it to step 1. The remaining subgraphs are merged to restore them to the state they were in originally; we then return the resulting graph to step 1. This is because we avoid creating a subgraph with the small user numbers like C in Figure. 1. N is the maximum number of users permitted in an extracted subgraph. (b) If no subgraph has more than N users, we regenerate the original graph and return it to step 1. This process is repeated until it is impos-

sible to form the original graph with more than N users or we reach the iteration limit of T which is normally set much larger than the number of vertices in the original graph. If we reach the iteration limit of T, we accept only that subgraph that has the most users among the past T trials. The remaining subgraphs are merged to restore them to the state they were in originally; we then return the resulting graph to step 1. This process creates subgraphs that have fewer than N users. Each of the final subgraphs is taken as a COI. If we want to create a more hierarchical COI structure, we set the value of N smaller and repeat steps 1 and 2.

(3) We semantically tag the relationships between users in COIs and between COIs based on taxonomies of content items. We first express user interests according to a taxonomy of content items using the method presented in [1].

(a) Then, we assign similar interests to the relationship between users using instances with classes that both users are interested in. We also assign contra-interests according to the direction of the relationship between user a and user b, a neighbor of a in a graph. (b) In the same way, we assign similar interests and contra-interests between COI A and a neighboring COI of A. We also give parameter d, which is the permitted number of instances or classes assigned to the relationships between users or between COIs.

3. EVALUATION

Our experiments used the interests of 3,632 users extracted from the large-scale blog portal Doblog (http://www.doblog.-com) in our previous experiment presented in [1]. We used the taxonomy of music artists, which contains 325 classes as genres and 25,000 artists as instances, provided by Listen-Japan (http://listen.jp), a music service provider. We then created COIs by changing the value of N_1 and N_2, which is N explained in Section 2 in the first iteration, and N_2, which is N in the second iteration. We set T to 10,000, which exceeded the user number, 3,632.

Then, we evaluated the accuracy and efficiency of graph searches as follows. (1) We divided the instances of the interests of user u_e randomly into two datasets, $D1$ and $D2$. We then measured the similarity between users using $D1$ based on the similarity measure in [1], and created a graph as explained in Section 2. We consider $D2$ to be correct answers for u_e, and u_e tries to find instances in $D2$ by searching neighboring users in adjoining COIs. (2) User u_e checks d different classes (or instances) attached to the relationships set between u_e and neighbors of u_e, and selects the neighbor that has the most correct classes (or instances) for u_e among all neighbors. u_e then hops to the selected neighbor and checks all classes (instances) of the neighbor to find correct classes (instances). When there are no neighbors of COI A, to which u_e is assigned, u_e checks different classes (instances) assigned to the relationships between COI A and those of neighboring COIs. u_e hops to a user in COI B if B has the most correct classes (instances) in difference (classes) instances among all neighboring COIs.

We evaluated the accuracy of user interest searches using the metrics of recall and precision (averaged values) among the 3,632 users. Recall means the proportion of correct answers in $D2$ found by the latest hop from among all correct answers for each user. Precision means the proportion of correct answers in $D2$ found by the latest hop out of the total number of classes (instances) checked. We also evaluated the efficiency by investigating the number of classes

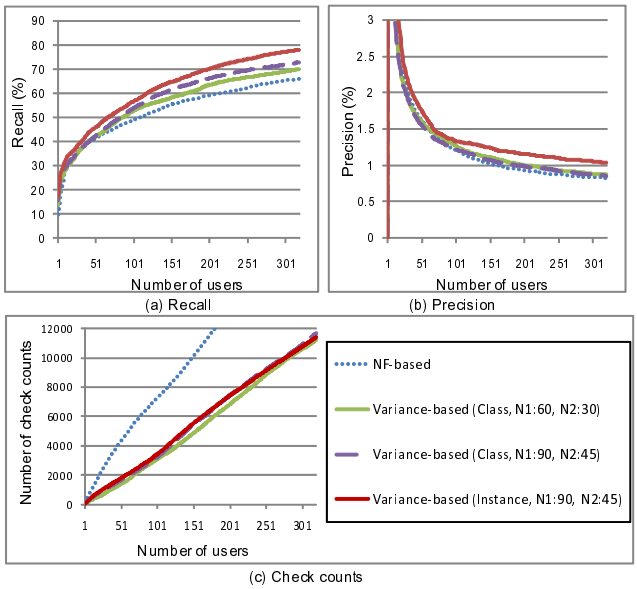

Figure 2: Result graphs of our evaluation.

(instances) users need to check in neighboring users or COIs.

We evaluated search performance when users referred to only classes such as "acid jazz". Figure. 2 compares the accuracy of NF-based COI search and variance-based COI search. We set d as 6 because the recall is improved as we increase d, but saturates above 7. Variance-based COI search offers lower checking frequency and better recall than NF-based COI search. We also compared the impact of basing searches on classes or on instances. Recall and precision are slightly lower with class search than with instance search. These results indicate that we can select suitable users or COIs by checking instances because instances reflect user interests in more detail than classes. However, COI search on classes is also useful since it allows users to acquire expert knowledge by referring to the taxonomy of instances; it is especially useful when users know only a few instances.

4. CONCLUSION

This paper extracts communities of interests (COIs) from a graph that well support graph searches with high recall and precision with low checking frequency. We attach semantic tags, based on taxonomies of content items, to the relationships between users and between COIs. Furthermore, we minimize the variance in the number of users in each COI and extract hierarchical COIs to yield more effective graph searches. Experiments showed that hierarchical COIs extracted by our algorithm outperform the hierarchical COIs extracted by the original NF algorithm in graph searches. Furthermore, we showed that users can select COIs or users in a graph effectively since the relationships between COIs or users are tagged with semantics based on taxonomies of content items.

5. REFERENCES

[1] Nakatsuji, M., Miyoshi, Y. and Otsuka, Y.: Innovation Detection Based on User-Interest Ontology of Blog Community., ISWC2006, pp. 515–528 (2006).

[2] Newman, M. E. J.: Fast algorithm for detecting community structure in networks, *Physical Review E*, Vol. 69, (2004).