Finding Community Structure in Mega-scale Social Networks [Extended Abstract][*]

Ken Wakita
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku
Tokyo 152-8552, Japan
wakita@is.titech.ac.jp

Toshiyuki Tsurumi
Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku
Tokyo 152-8552, Japan
tsurumi2@is.titech.ac.jp

ABSTRACT

Community analysis algorithm proposed by Clauset, Newman, and Moore (CNM algorithm) finds community structure in social networks. Unfortunately, CNM algorithm does not scale well and its use is practically limited to networks whose sizes are up to 500,000 nodes. We show that this inefficiency is caused from merging communities in unbalanced manner and that a simple heuristics that attempts to merge community structures in a balanced manner can dramatically improve community structure analysis. The proposed techniques are tested using data sets obtained from existing social networking service that hosts 5.5 million users. We have tested three three variations of the heuristics. The fastest method processes a SNS friendship network with 1 million users in 5 minutes (70 times faster than CNM) and another friendship network with 4 million users in 35 minutes, respectively. Another one processes a network with 500,000 nodes in 50 minutes (7 times faster than CNM), finds community structures that has improved modularity, and scales to a network with 5.5 million. Further detail is reported in [3].

Categories and Subject Descriptors: H.2.8 [Database applications]: Data mining; G.2.2 [Graph Theory]: Graph algorithms

General Terms: Algorithms, Performance

Keywords: Community analysis, clustering, social networking service

1. INTRODUCTION

Finding community structure in complex networks is an important first step to grasp inherent complex structure of social networks. Many research activities attempt to define the notion of communities and propose community analysis algorithms[2, 1].

We implemented a fast community analysis algorithm proposed by Clauset, Newman, and Moore [1] (CNM algorithm) and applied it to analyze various subsets of an acquaintance relationship network obtained from a social networking system (SNS). It performs well for a mid-scale subset of the network that consists of less than 500,000 users it was incapable to analyze larger networks.

We observed that merging in coupling pair of community structures, balancedness of the pair has great impact on computational efficiency of CNM algorithm. Based on this observation, we have implemented a slightly modified versions of CNM algorithm and observed remarkable speedup and slight improvement of the modularity.

2. CNM ALGORITHM

Newman and Girvan attempt to measure the quality of network clustering by means of *modularity* [2]. Their algorithm (CNM algorithm) is a bottom-up agglomerative clustering which continuously finds and merges pairs of clusters trying to maximize modularity of the community structure in a greedy manner [1].

The authors have programed CNM algorithm and attempted to analyze an friendship network of an SNS called "mixi[1]" that hosted about one million users in October 2005. In spite of the good scalability as advertised in [1], we had to stop the experiment after a week when less than 10% of the whole analysis was finished.

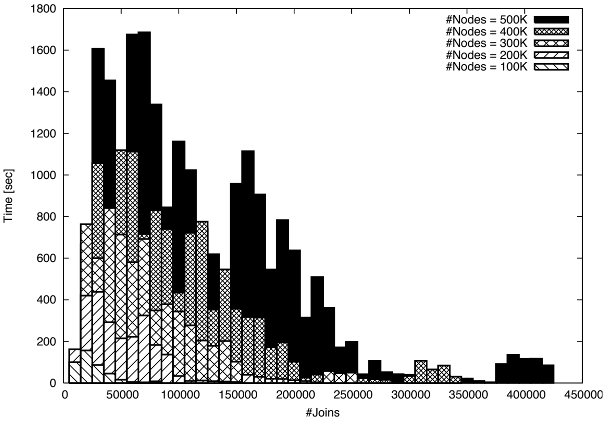

Figure 1: Analysis time.

To figure out the performance bottleneck, we conducted community analysis on a various subsets of mixi SNS network. The mixi SNS assigns each user an incrementa serial registration ID . Let us represent the mixi SNS network by a

[*] A long version of this paper is available as [3].

This work is supported in part by the Grant-in Aid (No. 18300041) of MEXT, Japan.

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

[1] mixi (http://mixi.jp/) is the largest invitation-based SNS in Japan and it hosts about 7 million users (Feb. 2007).

graph $G_{\rm mixi} = (U, F)$, where $U = \{1, 2, \ldots\}$ is the set of user IDs and $F \subset U \times U$ is a set of friendship relationship. A subset of this graph can be given as follows:

$$G^{n}_{\rm mixi} = (U_n, F \cap (U_n \times U_n))\text{where } U_n = \{u \in U | u \leq n\}$$

Figure 1 illustrates time required for analysis of various subsets: $G^{N}_{\rm mixi}$. Each bar of the graph depicts time required to merge 10,000 community pairs. For most of the computation time is consumed for the first half of the merging process which decreases dramatically for the latter half.

[1] discusses the computational complexity of CNM algorithm can be approximated by $O(n \log^2 n)$, for n nodes but this approximation contracicts super quadratic cost illustrated in Figure 1.

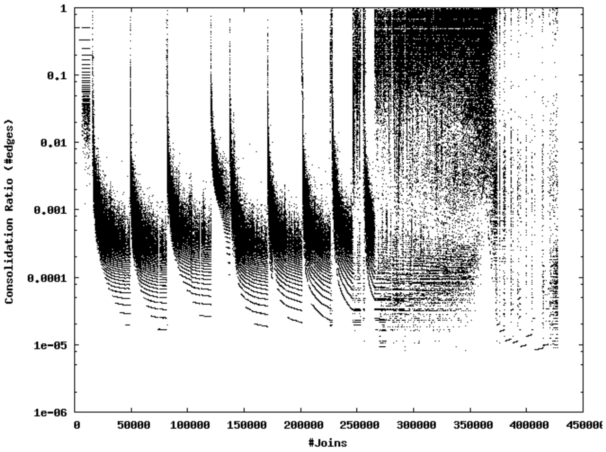

Figure 2: Consolidation ratio of each merge step.

From the merge logs that record how community pairs are merged into larger ones, it was observed that only a few large communities are growing fast, merging in many tiny communities. Figure 2 presents unbalancedness of merge steps in the course of community analysis for $G^{500K}_{\rm mixi}$. For this purpose, we have defined the notion of *consolidation ratio*, which plots, for n-th merge step, $c_k :=$ join(c_i, c_j), $(n, \mathit{ratio}(c_i, c_j))$, where the size ($s$) of a community is measured in terms of the number of its links to other communities.

$$\mathit{ratio}(c_i, c_j) = \min(s(c_i)/s(c_j), s(c_j)/s(c_i)).$$

The approximation of computational complexity ($n \log^2 n$) of [1] is based on an anticipation that the merges are performed in a balanced manner, which in practice is not the case and thus leads to super quadratic computational complexity.

3. HEURISTICS

We have seen that if we could control the growth of communities so that they grow in a balanced manner, the performance of the algorithm will improve remarkably. The structure of the algorithm remains the same except for the way the commuity pairs are chosen: we use both $\Delta Q^{\mathcal{C}}_{c_i,c_j}$ as well as $\mathit{ratio}(c_i, c_j)$, which the original algorithm uses only the former.

We can build variants of the algorithm choosing different metrics for computing the community sizes, $s(c)$. We tested three variants *HN*, *HE*, and *HE′*: HN measures a community by the number of its members, and HE measures a community by the number of edges that link this community with others, and HE′ is a hybrid of CNM algorithm and HE.

Algorithm 1 Outline of the proposed algorithm.

```
    $\mathit{ratio}(c_i, c_j) \equiv \min(s(c_i)/s(c_j), s(c_j)/s(c_i))$;
    while (true) {
        updateDeltaQ();
        Find $(c_i, c_j) \in \mathcal{C}^2$
        that has maximum $\Delta Q^{\mathcal{C}}_{c_i,c_j} \cdot \mathit{ratio}(c_i, c_j)$.
        if $(\max(\Delta Q^{\mathcal{C}}_{c_i,c_j} < 0)$ break;
        $\mathcal{C}$:= join$(c_i, c_j)$;
    }
```

Table 1: Elapsed time (minutes):
Java 5.0, 3.2GB Heap, Intel Xeon 2.80GHz.

	$G^{200K}_{\rm mixi}$	$G^{400K}_{\rm mixi}$	$G^{600K}_{\rm mixi}$	$G^{800K}_{\rm mixi}$	$G^{1M}_{\rm mixi}$	$G^{4M}_{\rm mixi}$
CNM	42.2	197	NA	NA	NA	NA
HE	2.15	6.80	13.6	24.5	36.2	243
HE′	8.52	35.5	68.2	124	173	3,400
HN	0.43	1.16	2.05	3.17	4.47	33.3

4. EVALUATION

Use of heuristics dramatically accelerates execution of community analysis (Table 1, Figure 3). The largest data set the original algorithm (Clauset+ (2004)) was possible to analyse is $G^{500K}_{\rm mixi}$. It took about 5.9 hours. The fastest heuristics (HN) processes $G^{1M}_{\rm mixi}$ in less than five minutes. The slower on HE′ takes 3 hours but offers imporoved modularity [3].

5. REFERENCES

[1] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[2] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[3] Ken Wakita and Toshiyuki Tsurumi. Finding community structure in mega-scale social networks, February 2007, cs.CY/0702048. http://arxiv.org/abs/cs.CY/0702048v1.

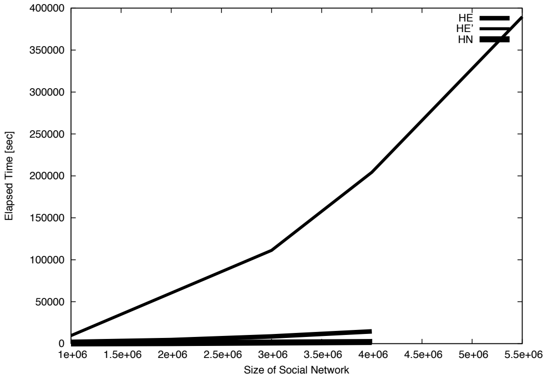

Figure 3: Scalability