Weighted Micro-Clustering: Application to Community Detection in Large-Scale Co-Purchasing Networks with User Attributes

Tomoya Yamazaki[1], Nobuyuki Shimizu[2], Hayato Kobayashi[2], Satoshi Yamauchi[2]
[1]Graduate School of Informatics, Kyoto University, [2]Yahoo Japan Corporation
t.yamazaki@iip.ist.i.kyoto-u.ac.jp, {nobushim,hakobaya,satyamau}@yahoo-corp.jp

ABSTRACT

We propose a simple and scalable method for soft community detection that makes use of both graph structures and vertex attributes. Our method is based on micro-clustering, which is a scalable and efficient clique-based method for detecting overlapping communities in unweighted graphs. We extend this method to graphs with vertex attributes so that we can make use of information supplied by vertex attributes. Our method still requires the same time complexity as micro-clustering. We confirm the validity and efficiency of our method by applying it to a large-scale co-purchasing network of real online auction data.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Network problems*

Keywords

Community detection, Micro-clustering, Online auction.

1. INTRODUCTION

With the rise of e-commerce services and SNSs, graph structure data have been constantly increasing. In such graphs, edges indicating friendship or co-purchasing relationships connect user vertices, and vertices often have attributes, for example ages or genders. To identify characteristic structures called *communities* in such data, many community detection algorithms have been proposed [1, 2].

While traditional community detection algorithms only focus on the graph structures, we propose a community detection method that takes account of both the graph structures and their vertex attributes. Our clique-based method is scalable, easy-to-implement, and enables us to extract overlapping communities.

As enumerating a huge number of cliques is hard, clique-based methods must address computability issues. To make the enumeration problem tractable, Uno et al. proposed *micro-clustering* [5] which is the latest clique-based method shown to be effective [4]. Since micro-clustering takes only unweighted graphs, we propose a weighted version of micro-clustering in order to apply it to community detection in graphs with user attributes, while keeping the same time complexity as its predecessor.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2889406.

Algorithm 1: Weighted micro-clustering.

Data: Graph $G = (V, E)$ with vertex attributes X, threshold θ, and the number τ of iterations.
Result: Clustering result of V

```
1 P ← G;
2 for i ← 1 to τ do    /* graph polishing */
3     E' ← ∅;
4     for (u, v) ∈ V^2 : u ≠ v ∧ N(u) ∩ N(v) ≠ ∅ do
5         if f(u, v) ≥ θ then E' ← E' ∪ (u, v);
6     P ← (V, E');
7 return all maximal cliques in the polished graph P;
```

2. OUR METHOD

We propose a weighted micro-clustering algorithm as shown in Algorithm 1, which is an extension of micro-clustering for treating vertex attributes. For a given graph $G = (V, E)$ with vertex attributes $X : V \to \mathbb{R}^d$, where X is a function that returns the d-dimensional attribute vector of a vertex $v \in V$, this algorithm extracts communities or dense structures from graph G through the following two procedures: graph polishing and clique enumeration.

In the graph polishing procedure (lines 2-6), the algorithm cleans the original graph by removing and adding edges on the basis of the following similarity:

$$f(u,v) = \frac{\sum_{n \in N(u) \cap N(v)} \min\{J(\mathbf{u},\mathbf{n}), J(\mathbf{v},\mathbf{n})\}}{\sum_{n \in N(u) \cup N(v)} \max\{J(\mathbf{u},\mathbf{n}), J(\mathbf{v},\mathbf{n})\}},$$

where $N(v)$ is the set of neighbors of a vertex v on a polished graph $P = (V_P, E_P)$, i.e., $N(v) = \{u \mid (u, v) \in E_P\}$, J is the generalized Jaccard similarity, i.e., $J(\mathbf{u}, \mathbf{v}) = \sum_i \min\{u_i, v_i\} / \sum_i \max\{u_i, v_i\}$, and the bold style $\mathbf{v}$ of v is the attribute vector of v, i.e., $\mathbf{v} = X(v)$. Note that f depends on P and X as well as u and v, though they are omitted for simplicity.

Our graph polishing procedure is very efficient, since it is enough to enumerate intersections $N(u) \cap N(v)$ of neighbors for calculating the above similarity. The next theorem shows the time complexity of composing a polished graph, which is the same as the original micro-clustering (cf. Theorem 6

Table 1: Statistics of graphs.

Data set	CHARITY	PET	FOOD	BOOKS	SPORTS
Vertices	99	4,249	22,398	39,469	120,741
Edges	1,645	60,110	492,158	525,873	2,185,242
True Class	2	12	119	151	141

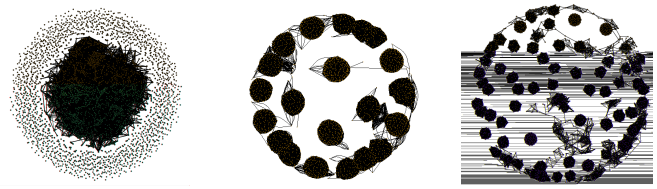

Figure 1: The left-hand side is an original graph, the middle is the result of micro-clustering, and the right is that of weighted micro-clustering.

in [5]). We proved this theorem by using the fact that the number of traversing vertices in our similarity f is the same as that in the Jaccard index in the original one.

THEOREM 1. *If the degree of a given graph $G = (V, E)$ follows a power law, that is, the degree of i-th vertex (in descending order of degree) is α/i^{Δ} for some constant α and Δ, the time complexity of composing the polished graph P in Algorithm 1 is $\mathrm{O}(A\alpha^2 \log |V|)$ for $\Delta = 1/2$ and $\mathrm{O}(A\alpha^2)$ for $\Delta > 1/2$, where A is the dimension of attribute vectors.*

In the clique enumeration procedure (line 7), the algorithm enumerates all maximal cliques in the polished graph P. We can use any enumeration algorithms such as MACE [3], and the practical performance should be good since the number of maximal cliques becomes very small after graph polishing. Uno et al. showed that the worst case number of maximal cliques in a polished graph is bounded by a polynomial with respect to the number of vertices $|V|$, though that in an original graph is exponential.

3. EXPERIMENTS

We evaluate the algorithm with co-purchasing graph data from YAHUOKU![1], the largest online auction site in Japan, operated by Yahoo Japan Corporation. In our co-purchasing graph, vertices represent buyers and if two buyers buy products from the same seller, their vertices are adjacent. We show the statistics of graphs in Table 1.

Figure 1 shows the graphs obtained from BOOKS data as well as the results of original/weighted micro-clustering. We observe that the weighted version extracts smaller clusters compared with the original micro-clustering. Therefore, we know more detailed user relationships.

In order to evaluate the algorithm, we partition vertices by associating users with their most frequently purchased product categories. Products are placed in a predefined hierarchical taxonomy. For example, a Harry Potter novel is first placed in `Books/Magazines` (1st level), then `Literature/Novels` (2nd level), and finally in `Fantasy` (3rd level). We use 3rd level categories to induce the gold standard communities so that a user belongs to a true class $c \in C = \{c_1, \ldots, c_L\}$.

[1]http://auctions.yahoo.co.jp/

Table 2: NMI and purity of our method and other methods.

	CHARITY	PET	FOOD	BOOKS	SPORTS
	Normalized Mutual Information				
Our method	0.6723	0.5368	**0.5899**	**0.6463**	**0.5251**
Micro-clustering [5]	0.6689	0.5272	0.5779	0.6020	0.5212
Louvain method [1]	0.6083	**0.6614**	0.5707	0.5110	0.3213
K-means method	**0.7782**	0.2898	0.4008	N/A	N/A
	Purity				
Our method	**1.0**	**0.9582**	**0.8953**	**0.7917**	**0.8101**
Micro-clustering [5]	0.9914	0.9482	0.8286	0.7619	0.4298
Louvain method [1]	0.9759	0.8409	0.6688	0.5635	0.3740
K-means method	0.8989	0.3656	0.6845	N/A	N/A

We use purchase histories in the 2nd level categories as user attributes. For example, if a girl buys products in `Literature/Novel` twice, her attribute vector has the value 2 for the dimension representing `Literature/Novel`.

We measure the normalized mutual information (NMI) between the set of users in communities $\Omega = \{w_1, \ldots w_M\}$ and C. The NMI between Ω and C is defined as follows:

$$I(\Omega, C) = \frac{-2\sum_{m=1}^{M}\sum_{l=1}^{L}\frac{|w_m \cap c_l|}{S}\log\frac{S|w_m \cap c_l|}{|w_m||c_l|}}{\sum_{m=1}^{M}\frac{|w_m|}{S}\log\frac{|w_m|}{S} + \sum_{l=1}^{L}\frac{|c_l|}{S}\log\frac{|c_l|}{S}},$$

where $S = |\bigcup_{\omega\in\Omega} w|$. The purity between them is defined as $p(\Omega, C) = \frac{1}{S}\sum_m \max_l |w_m \cap c_l|$.

We compare our method with popular community detection and clustering methods in terms of the NMI and purity in Table 2. The number of clusters of k-means method is set to that of the true classes. The run time of our method is comparable to that of the Louvain method which is one of the fastest community detection methods, while the k-means method timed-out for BOOKS and SPORTS. We fix the threshold θ to 0.6 and τ to 5. The NMI of our method is higher than other methods except for the cases where the number of true classes or vertices is excessively small. In terms of purity, our method is consistently higher than other methods.

Interestingly, we observe users in some communities with low purity who bid on similar items not included in the same category; for example, in reality, there is only one community for cookbooks; however, cookbooks could be under three different categories such as `Living`, `Kids` or `Health`. In this case, we find a real community despite low purity. These results confirm the validity of our community detection method, which takes account of both the graph structures and their vertex attributes.

4. REFERENCES

[1] V. Blondel et al. Fast unfolding of communities in large networks. *J. Stat. Mech*, page P10008, 2008.

[2] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99(12):7821–7826, 2002.

[3] K. Makino and T. Uno. New algorithms for enumerating all maximal cliques. In *Algorithm Theory-SWAT 2004*, pages 260–272. Springer, 2004.

[4] T. Nakahara et al. Prediction model using micro-clustering. *Procedia Computer Science*, 35:1488–1494, 2014.

[5] T. Uno et al. Micro-clustering: Finding small clusters in large diversity. *CoRR*, pages 1–23, 2015.