# A New Hierarchical Clustering Algorithm to Identify Non-overlapping Like-minded Communities

Talasila Sai Deepak
Google, Mountain View
t.deepak.iitg@gmail.com

Hindol Adhya
Indian Institute of Technology, Guwahati
hindol.adhya@gmail.com

Shyamal Kejriwal
Indian Institute of Technology, Guwahati
shyamalkejriwal@gmail.com

Bhanuteja Gullapalli
Indian Institute of Technology, Guwahati
bhanutejaiit@gmail.com

Saswata Shannigrahi
Indian Institute of Technology, Guwahati
saswata.sh@iitg.ernet.in

## ABSTRACT

In this paper, we present a new algorithm to identify non-overlapping like-minded communities in a social network and compare its performance with Girvan-Newman algorithm, Lovain method and some well-known hierarchical clustering algorithms on Twitter and Filmtipset datasets.

## Keywords

Community detection; Modularity; Like-mindedness

## 1. INTRODUCTION

A social network is denoted by an undirected and unweighted sparse graph $G = (V, E)$ with vertex set $V = \{1, 2, \ldots, |V|\}$ such that $|E| = O(|V|)$. Each vertex $v \in V$ is associated with a behavioral vector $X_v$ of dimension $d$. For example, the ratings given by a user on a movie rating website (with some default rating being given to those movies he has not rated) can be his behavioral vector, the dimension of which is the number of movies available for rating. A similarity metric $sim(u, v)$ is a distance measure between the vectors $X_u$ and $X_v$ representing the behavior of the vertices $u, v \in V$, respectively. In this paper, we use cosine similarity $\frac{X_u \cdot X_v}{\|X_u\| \|X_v\|}$ as the distance measure. Let $C = \{C_1, C_2, \ldots, C_k\}$ be a partition of $V$, each representing a set of vertices (community) in a community structure having $k$ non-overlapping communities. The **modularity** [7] $Q(C)$ of the set of communities $C$ is defined as $Q(C) = \sum_{i=1}^{k} \left(a_i - b_i^2\right)$, where $a_i$ is the fraction of $|E|$ edges with both its vertices in the same community $C_i$ and $b_i$ is the fraction of $|E|$ edges with at least one vertex in community $C_i$. The **like-mindedness** [6] $L(C)$ of the set of communities $C$ is defined as the average of all intra-community vertex pair similarities. In other words, $L(C) = \frac{1}{\sum_{u,v \in V: u \leq v} \delta(u,v)} \sum_{u,v \in V: u \leq v} sim(u, v)\delta(u, v)$, where the boolean function $\delta(u, v)$ is 1 if and only if there exists a community $C_i \in C$ such that $u, v \in C_i$.



*HT '16 July 10-13, 2016, Halifax, NS, Canada*



ACM ISBN 978-1-4503-4247-6/16/07.

DOI: http://dx.doi.org/10.1145/2914586.2914613

## 2. LIKE-MINDEDNESS MAXIMIZATION

In this section, we present a bottom-up hierarchical clustering approach in which one starts with each vertex belonging to $V$ as its own community. In each subsequent step, pairs of communities are merged till there is only one community left. The pair of communities that gives the minimum value of a pre-defined linkage criterion (defined below) is selected for being merged in each step. In this agglomerative approach, we get a hierarchy of communities, often visualized as a dendrogram. We observe from the dendograms of several hierarchical clustering algorithms that an algorithm produces higher like-mindedness if small clusters are merged in the early iterations in order to avoid the creation of large heterogeneous communities. Motivated by this fact, we design this algorithm in which we discourage the merging of two large communities in an iteration. At every iteration, we identify the pair $\{C_i, C_j\}$ of communities that has the highest score $S(C_i, C_j) = \frac{1}{max\{|C_i|,|C_j|\}} + \frac{1}{|C_i||C_j|} \sum_{u \in C_i, v \in C_j} sim(u, v)$. The left term is used to discourage the merging of two large communities, and the right term accounts for the average like-mindedness of the inter-community pairs from $\{C_i, C_j\}$. A high value of the right term ensures that the like-mindedness of the set of communities after merging is high, since $\sum_{u \in C_i, v \in C_j} sim(u, v)$ is the sum of the similarities of $|C_i||C_j|$ pairs of inter-community vertices belonging to $C_i$ and $C_j$.

## 3. DATASET & EXPERIMENTAL SETUP

### 3.1 Filmtipset

Filmtipset is Sweden's largest movie rating website, in which a user has the option to rate a movie on a scale of 1 to 5. Apart from this, there is a social network element of the website where a user can *follow* another user in the network. We have 86,725 such following relationships between the users. We designate two users $u, v \in V$ as friends if $u$ is following $v$, and vice-versa.

To use the dataset in our experiments, we apply a couple of filters. The first filter is applied on the number of times a

**Table 1: Filmtipset and Twitter datasets**

| Parameter | Filmtipset Unfiltered Network | Filmtipset Filtered Network 1 | Filmtipset Filtered Network 2 | Twitter Unfiltered Network | Twitter Filtered Network |
|---|---|---|---|---|---|
| Number of nodes | 91530 | 4305 | 983 | 40096646 | 5013 |
| Number of isolated nodes | 61211 | 168 | 152 | 17522652 | 1 |
| Edge count (friendships) | 56387 | 10940 | 1807 | 232157703 | 1636971 |
| Avg. clustering coefficient | 0.467 | 0.434 | 0.338 | 11.5799 | 653.09 |
| Avg. degree | 1.232 | 5.082 | 3.676 | | 0.2063612 |
| Diameter | 20 | 18 | 16 | 18 [5] | 4 |
| Avg. path length | 7.508 | 5.796 | 4.817 | 4.12 [5] | 1.874073 |
| Size of giant component | 29.54% | 90.77% | 72.94% | | 5012 |
| Homophily ratio | | 26.44 | 62.07 | | 1.18 |

**Table 2: Symbols used in Figure 1**

| Symbol | Full form | Running time |
|---|---|---|
| LMM/LMMS | LMM Algorithm using (un)interested vector | $O(\|V\|^2 \log \|V\|)$ |
| LMMR | LMM Algorithm using rating vector | $O(\|V\|^2 \log \|V\|)$ |
| L | Louvain method [1] | $O(\|V\| \log \|V\|)$ (estimated) |
| ML/MLS | Modified Louvain method using (un)interested vector | $O(\|V\|^2 \log \|V\|)$ (estimated) |
| MLR | Modified Louvain method using rating vector | $O(\|V\|^2 \log \|V\|)$ (estimated) |
| GN | Girvan-Newman algorithm [4] | $O(\|V\|^3)$ |
| S/SS | Single-linkage Clustering using (un)interested vector [8] | $O(\|V\|^2)$ |
| A/AS | Average-linkage Clustering using (un)interested vector [2] | $O(\|V\|^2 \log \|V\|)$ |
| C/CS | Complete-linkage Clustering using (un)interested vector [3] | $O(\|V\|^2)$ |
| SR | Single-linkage Clustering using rating vector [8] | $O(\|V\|^2)$ |
| AR | Average-linkage Clustering using rating vector [2] | $O(\|V\|^2 \log \|V\|)$ |
| CR | Complete-linkage Clustering using rating vector [3] | $O(\|V\|^2)$ |

movie has been rated. Some movies are *popular* since they are rated by many. We observe that removing the most popular movies from being considered results in a higher ratio of the average similarity (w.r.t. rating vectors defined below) of the pairs of friends to the average similarity of the non-friend pairs. We denote this ratio for $G$ as its *homophily ratio*

$$H(G) = \left( \frac{\sum_{u,v \in V:(u,v) \in E} sim(u,v)}{|E|} \right) / \left( \frac{\sum_{u,v \in V:(u,v) \notin E} sim(u,v)}{\binom{|V|}{2} - |E|} \right).$$

It can be noted that aiming for too high a homophily ratio reduces the number of movies a lot. Since the number of movies left is used for filtering out inactive users (see below), a high homophily ratio implies a reduction in the number of active users as well. Therefore, we remove movies that are rated at least 50 times in filter 1 since it ensures a large number of movies left after filtering. To see the performance of our algorithm on networks having high homophily ratio, we remove movies that are rated at least 5 times in filter 2.

For each of the movie filters 1 and 2, we define a user filtering criterion as follows. For movie filter 1 (filter 2), we say a user to be *active* if he rates at least 5 movies among the movies left after removing all movies rated more than 50 (5, respectively) times. A user is called *social* if he has at least 5 friends in the network. For each of movie filter 1 and 2, we create an induced subgraph such that each user in this subgraph is active and social. In Table 1, we summarize the properties of these datasets, which we would denote by Filmtipset Filtered Networks 1 and 2. We also note that the degrees of vertices in the Unfiltered Network as well as the Filtered Networks follow a *power law distribution*.

For either of Filmtipset Filtered Network 1 and 2, we consider the movies in an order and each user $u$ is assigned a *rating vector* $R_u$, each entry of which is either the rating given by him to that particular movie or 0 if he has not rated it. We also create another *(un)interested vector* $S_u$ for each user $u$, each entry of which is 1 or 0 depending on whether a user has rated that movie or not, respectively. In order to implement the behavioral property based community finding algorithms, either $R_u$ or $S_u$ is used as the behavioral vector $X_u$ of $u \in V$.

## 3.2 Twitter

Using the publicly available dataset [5] which has about 40 million users including users having more than 10,000 followers designated as *celebrities*, we create a friendship graph

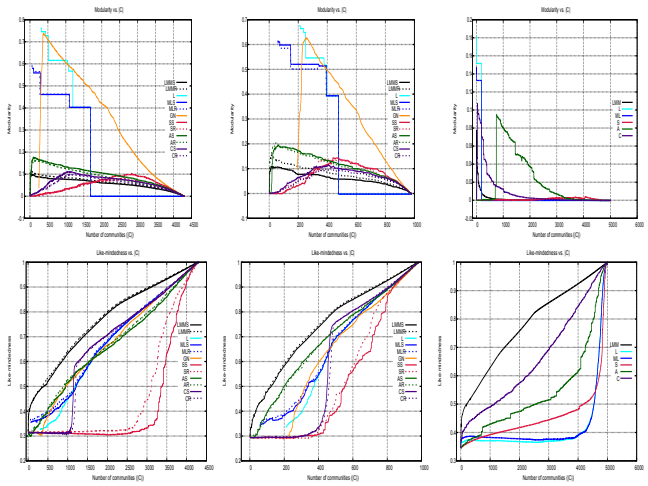

**Figure 1: Modularity and like-mindedness scores in Filmtipset Filtered Networks 1 (left) and 2 (middle), and Twitter Filtered Network (right)**

called Twitter Filtered Network of the non-celebrity users having at least 5000 non-celebrity friends. We summarize the dataset in Table 1 and also note that the degrees of vertices in the Filtered Network follow a *power law distribution*.

As before, two users $u$ and $v$ are said to be friends if $u$ is following $v$, and vice-versa. We create a 0/1 vector $F_u$ (the $i$-th entry of which corresponds to the $i$-th celebrity) for each user $u \in V$ and use it as his behavioral vector $X_v$. The entry in $i$-th position of $F_u$ for $u$ is 1 or 0 depending on whether $u$ is following the $i$-th celebrity or not, respectively.

# 4. RESULTS AND CONCLUSIONS

In Figure 1, the like-mindedness and modularity scores achieved by different algorithms are compared by plotting their values against $|C|$, the number of communities identified. In Table 2, we summarize the running time of the algorithms considered in this paper. The key observation from Figure 1 is that our algorithm Like-mindedness Maximization outperforms all other algorithms (including Modified Louvain method where we add pairs of vertices having higher similarity than the current like-mindedness score as an edge after every iteration of the Louvain Method) on *like-mindedness* metric. Moreover, when the number of identified communities is large, we observe from Figure 1 that our algorithm obtains a community structure with comparable (to other algorithms considered here) modularity. The running time of our algorithm is $O(|V|^2 \log |V|)$, which is much faster than the Girvan-Newman algorithm but slightly slower than other hierarchical clustering algorithms. We also note that the actual ratings (not just the data about whether a user has rated a movie or not) given by Filmtipset users does not give any significant advantage to the performance of the community detection algorithms. This is due to the fact that the similarity matrices of rating and (un)interested vectors of the user pairs are quite similar, e.g., having a cosine similarity of 0.9420 and 0.9062 for Filmtipset Filtered Network 1 and 2, respectively. Another interesting observation is that all the algorithms obtain higher like-mindedness and modularity scores in Filmtipset Filtered Network 2 compared to Network 1, due to it having higher homophily ratio.

## 5. REFERENCES


[1] V. D. Blondel, J. L. Guillaume, R. Lambiotte and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 10:P10008, 2008.

[2] W. H. Day and H. Edelsbrunner. Efficient Algorithms for Agglomerative Hierarchical Clustering Methods. *Journal of Classification*, Volume 1, pp. 1-24, 1984

[3] D. Defays. An Efficient Algorithm for a Complete Linkage Method. *The Computer Journal (British Computer Society)*, 20 (4): 364-366, 1977.

[4] M. Girvan and M.E.J. Newman. Community Structure in Social and Biological Networks. *Proc. National Academy of Sciences*, 99(12): 7821-7826, 2002.

[5] H. Kwak, C. Lee, H. Park and S. Moon. What is Twitter, a Social Network or a News Media? *Proc. 19th international conference on World Wide Web*, 591-600, 2010.

[6] N. Modani, R. Gupta, S. Nagar, S. Shannigrahi, S. Goyal and K. Dey. Like-minded Communities: Bringing the Familiarity and Similarity Together. *World Wide Web*, 17 (5), 899-919, 2014.

[7] M.E.J. Newman and M. Girvan. Finding and Evaluating Community Structure in Networks. *Physical review E*, 69(2): 026113, 2004.

[8] R. Sibson. SLINK: an Optimally Efficient Algorithm for the Single-link Cluster Method. *The Computer Journal (British Computer Society)* 16 (1): 30-34, 1972.