A Seed-Centric Community Detection Algorithm based on an Expanding Ring Search

Abstract :

Most community detection algorithms are designed to detect all communities in the entire network. This is computationally expensive to first detect all communities and later identify communities based on individual interest especially on large networks. So our proposed algorithm utilizes an expanding ring search starting from the individual of interest as the seed user. Following which, we iteratively include users at increasing number of hops from the seed user, based on our definition of a community. This iterative step continues until no further users can be added, thus resulting in the detected community comprising the list of added users.

Introduction :

Most definitions of a community are generally based on the concept that the community comprises individuals who are more densely connected to each other in the community than to those outside the community. Our proposed algorithm starts from a seed user (i.e. the individual of interest) and performs an expanding ring search to iteratively include users into that community. Users are included into the community based on a metric of their number of links to other users in the community. This iterative adding of users continues until no further users satisfy the metric and could be added

Radicchi et al. 2004 introduced concepts of strong and weak communities where strong communities comprise individual users who each has more links within this community than outside, while weak communities comprise users who collectively have more links within this community than outside

This algorithm implemented a modified version of Radicchi et al.'s definition of a strong community by introducing a community strength factor for adjusting the size and strength of the community detected.

Methodology :

This method model the social network as an undirected, unweighted graph, G = (N,E) where N and E respectively refer to the set of nodes/users and edges/links in the graph. Undirected links correspond to social links that are reciprocal and reflective of real life friendships, thus our choice of undirected links for the algorithm. While our paper uses unweighted links, the algorithm could cater for weighted links by implementing a simple filtering scheme based on the weight of links.

Each user $i \in N$ has k_i links. The number of links pointing to users within the community is denoted as k_i^{in} and those outside the community as k_i^{out}. Our definition of community defined by :

$$k_i^{in} > k_i^{out} * f \qquad (1)$$

f is community strength factor

Algorithm :

1. Identify a user of interest as the seed node and include this user as part of the community. This user could be most influential person like CEO, head master… or a node with more number of links

2. Retrieve all neighbouring nodes of the seed node. Include these 1st degree neighbours as part of the community.

3. Retrieve all the 2nd degree neighbours of the seed node. Include them as part of the community if they satisfy our definition of a community as stated in Equation 1.

4. Repeat Step 3 for the 3rd, 4th, nth degree neighbours until no further nodes can be added to the community.

5. The eventual list of included nodes would be the community centred at the seed node.

Evaluation on YouTube Dataset :

This algorithm is evaluated based on YouTube groups. Users belonging to same group deemed to same community. Evaluated by topological measures of average clustering coefficient, average path length, average degree and diameter

As a control group for comparing network statistics, we selected the largest connected component. Seed users are selected based on highest number of links.

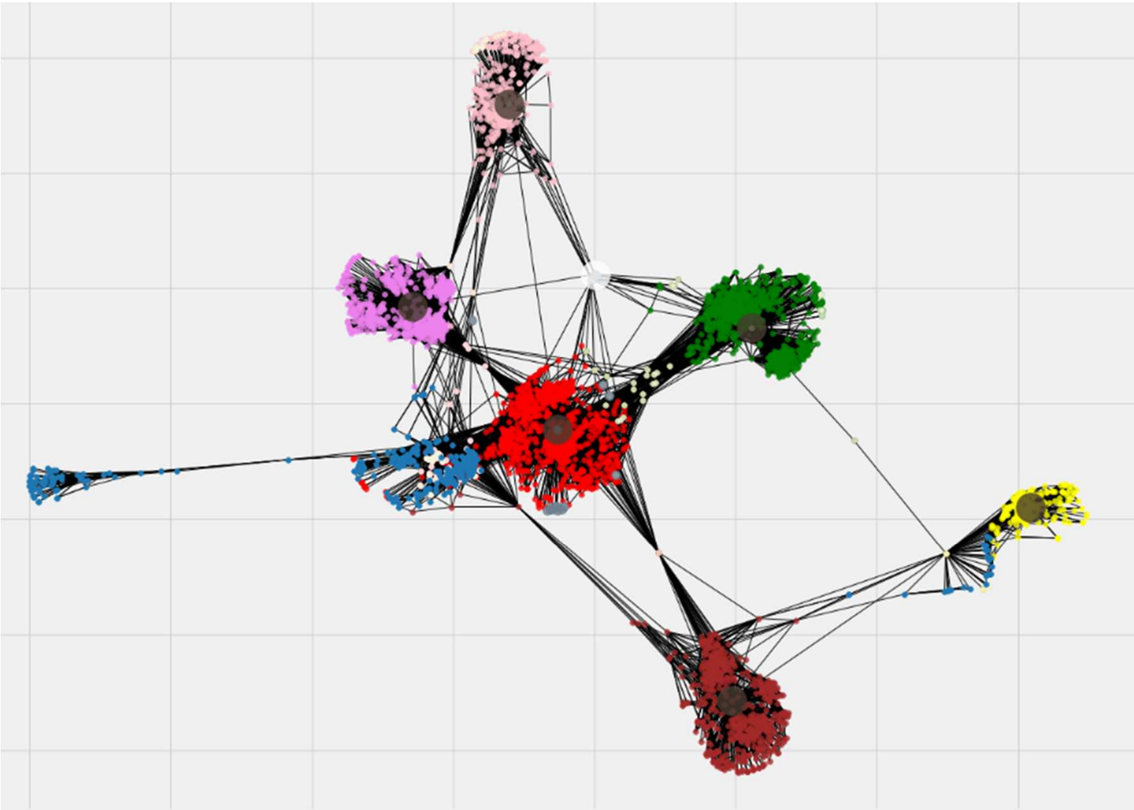

Fig 1 – Clusters formed by seed users

Network Property	Detected Community MIN	Detected Community MAX	Detected Community AVG	Control group
Avg degree links	77.52	16.02	43.7	36.6
Avg clustering coefficient	0.66	0.53	0.605	0.604
Avg Path length	2.11	1.92	2.01	3.70
Diameter	5	3	3.67	8

Conclusion :

Our evaluation of clustering coefficient, average path length, average degree of links and diameter also indicates that the detected communities are cohesive and well-connected. This is computationally less expensive if we want to detect community around a particular user.