Fast community detection in social graphs

Fast unfolding of communities in large networks

Laya Fakher

Danial Chekani

K. N. Toosi University of Technology

September 23, 2023

1928

K. N.Toosi University of Technology

A social network graph is a graph where the nodes represent people and the lines between nodes, called edges, represent social connections between them, such as friendship or working together on a project.

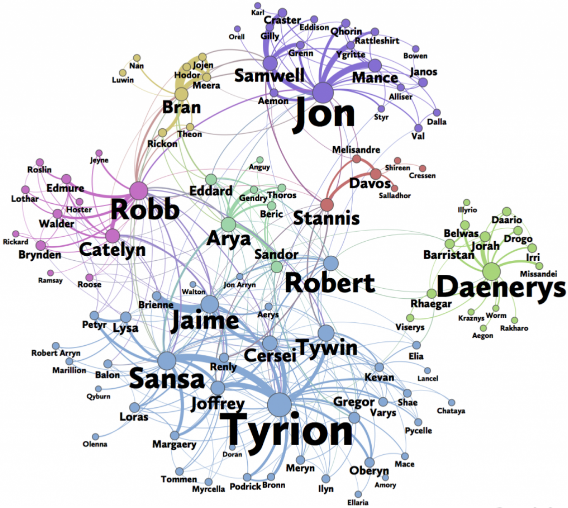

Figure 1: Social network graph example

- Finding key people
- Identifying clusters of people and communities
- Determining the closeness between two users
- Recommending new friends
- Pinpointing important content using common neighbors
- Push information flow based on friend relationships and geographic location

- communities are sets of highly inter-connected nodes.
- community detection algorithms can partition the network into multiple communities.
- The problem of community detection requires the partition of a network into communities of densely connected nodes, with the nodes belonging to different communities being only sparsely connected.

Several types of community detection algorithms:

- divisive algorithms
- agglomerative algorithms
- optimization methods

- The quality of the partitions resulting from these methods is often measured by the so-called modularity of the partition.
- The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities

In the case of weighted networks, modularity is defined as :

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

- Exact modularity optimization is a problem that is computationally hard.
- approximation algorithms are necessary when dealing with large networks.
- The fastest approximation algorithm for optimizing modularity on large networks was proposed by Clauset et al.
- That method consists in recurrently merging communities that optimize the production of modularity.

- This greedy algorithm may produce values of modularity that are significantly lower than what can be found by using simulated annealing.
- This algorithm has a tendency to produce super-communities that contain a large fraction of the nodes, even on synthetic networks that have no significant community structure
- This algorithm slows down the algorithm considerably and makes it inapplicable to networks of more than a million nodes

Today, the social networking service Facebook has about 64 million active users, the mobile network operator Vodaphone has about 200 million customers and Google indexes several billion web pages.

- The algorithm finds high modularity partitions of large networks in a short time.
- Unlike other algorithms, the network size limits are due to limited storage capacity rather than limited computation time.

- The algorithm is divided into two phases that are repeated iteratively.
- When the process starts, there are as many communities as there are nodes.
- for each node **i** and neighbors **j** of **i**, we evaluate the gain of modularity by removing **i** from its community and by placing it in the community of **j**.
- The ordering of the nodes does not have a significant influence on the final result, but it does affect the computation time.

- We aim for the maximum gain only if the gain is positive.
- The first phase ends when a local maxima of the modularity is attained.
- The gain in modularity can be obtained by the following formula :

$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \frac{\sum_{tot} + k_i}{2m}^2\right] - \left[\frac{\sum_{in}}{2m} - \frac{\sum_{tot}}{2m}^2 - \frac{k_i}{2m}^2\right]$$

- In the second phase, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities.
- Combination of these two phases is called a ”pass”.
- The passes are iterated until a maximum of modularity is attained.
- Most of the computation happens in the first pass.

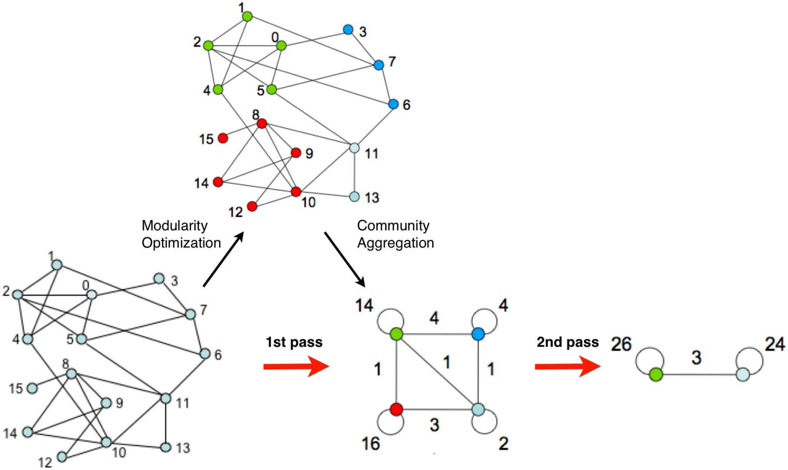

Figure 2: Visualization of the fast unfolding algorithm

- The steps are intuitive and easy to implement.
- Time complexity is linear on typical and sparse data since the gain in modularity is easier to compute.

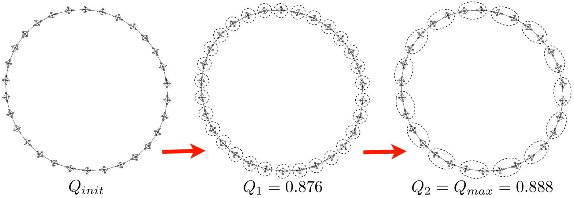

Figure 3: Visualization of the fast unfolding algorithm

1 Social network graph

2 Community detection

3 Fast unfolding method

4 **Application to large networks**
   - Testing against other algorithms
   - Belgian phone company

5 Conclusion

6 References

	Karate	Arxiv	Internet	nd.edu
Nodes/links:	34/77	9k/24k	70k/351k	325k/1M
CNM	.38/0s	.772/3.6s	.692/799s	.927/5034s
PL	.42/0s	.757/3.3s	.729/575s	.895/6666s
WT	.42/0s	.761/0.7s	.667/62s	.898/248s
Our algorithm	.42/0s	.813/0s	.781/1s	.935/3s

	Phone	uk-2005	WebBase 2001
Nodes/links:	2.6M/6.3M	39M/783M	118M/1B
CNM	-/-	-/-	-/-
PL	-/-	-/-	-/-
WT	.56/464s	-/-	-/-
Our algorithm	.769/134s	.979/738s	.984/152min

- Our algorithm outperforms all the other methods to which it is compared.
- The computation time is also relatively small.
- In the above examples, the number of passes is always smaller than 5.

- The network is composed of 2.6 million customers, and their total number of phone calls during 6 months period.
- This large social network is exceptional due to the fact that two main linguistic communities (French and Dutch) coexist.
- When processed by the algorithm, we identified a hierarchy of six levels in the network.
- At the top level there are 261 communities that have more than 100 customers which accounts for 75% of the customers.

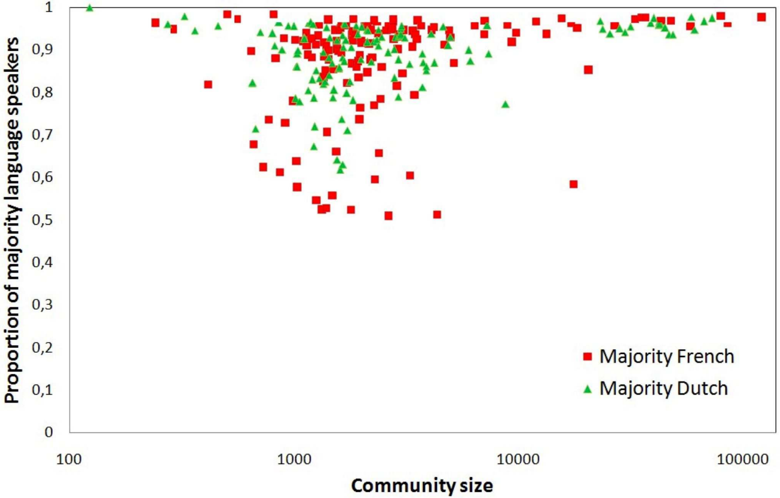

Figure 4: proportion of majority language speakers per community size

- The network is strongly segregated.
- There are 36 communities with more than 10000 customers and, except for one community at the interface between the two language clusters, all these communities have more than 85% of their members speaking the same language.

- Our algorithm unfolds a complete hierarchical community structure for the network.
- As mentioned before, the limitation factor in our algorithm is memory size rather than the computation time; which change of scales, i.e., from around 5 million nodes for previous methods to more than 100 million.
- The speed of the algorithm can still be substantially improved by using some simple heuristics; for instance by stopping the first phase of our algorithm when the gain of modularity is below a given threshold or by carefully ordering the nodes during the first phase.

References

[D. Blondel, 2008] Vincent D. Blondel (2008)
Fast unfolding of communities in large networks
physics.soc-ph

Thanks!