Fast unfolding of communities in large networks

R. Lambiotte
Institute for Mathematical Sciences
Imperial College London

with V.D. Blondel, J.-L. Guillaume and E. Lefebvre
Contents

1. Community detection
2. Our algorithm
3. Advantages
4. Belgian Mobile phone network
5. Conclusion
Community detection

Larger and larger networks, e.g. mobile phone networks of millions of users

Need for algorithm for unraveling useful information
Community detection

Larger and larger networks, e.g. mobile phone networks of millions of users

Need for algorithm for unraveling useful information

Community detection
communities of highly connected nodes while nodes belonging to different communities are weakly connected

=> discover functions
=> visualization
Modularity

Several ways to uncover such communities

Modularity: function which measures the quality of a partition

Let us focus on a weighted network and attribute to each node a community $C_i$

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

$A_{ij}$ weight of the edge between $i$ and $j$

$k_i = \sum_j A_{ij}$ degree of $i$

$m = \frac{1}{2} \sum_{ij} A_{ij}$ total weight

$Q \in [-1, 1]$
Modularity

Community detection by modularity optimization:

Given a network, what is the partition with the highest value of $Q$

Finding an exact answer is not possible in large networks

=> need for algorithms which find *good partitions* in short times

The largest network studied so far is a Japanese social networking systems of about 5.5 million (Wakita and Tsurumi) to be compared with:

Facebook has about 64 million active users

Vodaphone has about 200 million customers and Google indexes several billion web-pages.
Our algorithm

The algorithm is based on two steps that are repeated iteratively.

Initially, all the nodes belong to their own community (N nodes and N communities)

One looks through all the nodes (from 1 to N) in an ordered way. The nodes look among their neighbours and adopt the community of their neighbour if there is an increase of modularity. This step is performed iteratively until a local maximum of modularity is reached (each node may be considered several times).

Node 0 moves to the community of Node 3

After N nodes have been considered

After each nodes has been considered 4 times
Our algorithm

Once a local maximum has been attained, we build a new network whose nodes are the communities. The weight of the links between communities is the total weight of the links between the nodes of these communities.

New network of 4 nodes!
Note the self-loops

In typical realizations, the number of nodes diminishes drastically at this step.

This ensures the rapidity of the algorithm for large networks
Our algorithm

The two steps are repeated iteratively, thereby leading to a hierarchical decomposition of the network.
Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve
Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve

Very fast!

Largest network: 5.5 million nodes

<table>
<thead>
<tr>
<th></th>
<th>Karate</th>
<th>Arxiv</th>
<th>Internet</th>
<th>Web nd.edu</th>
<th>Phone</th>
<th>Web uk-2005</th>
<th>WebBase 2001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes/links</td>
<td>34/77</td>
<td>9k/24k</td>
<td>70k/351k</td>
<td>325k/1M</td>
<td>2.6M/6.3M</td>
<td>39M/783M</td>
<td>118M/1B</td>
</tr>
<tr>
<td>CNM</td>
<td>.38/0s</td>
<td>.772/3.6s</td>
<td>.692/799s</td>
<td>.927/5034s</td>
<td>-/-</td>
<td>-/-</td>
<td>-/-</td>
</tr>
<tr>
<td>PL</td>
<td>.42/0s</td>
<td>.757/3.3s</td>
<td>.729/575s</td>
<td>.895/6666s</td>
<td>-/-</td>
<td>-/-</td>
<td>-/-</td>
</tr>
<tr>
<td>Our algorithm</td>
<td>.42/0s</td>
<td>.813/0s</td>
<td>.781/1s</td>
<td>.935/3s</td>
<td>.769/134s</td>
<td>.979/738s</td>
<td>.984/152mn</td>
</tr>
</tbody>
</table>

Our only limitation being the storage of the network in main memory

Computer simulations on large ad-hoc modular networks suggest that its complexity is linear on typical and sparse data
Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve

Very fast!

Good accuracy
Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve

Very fast!

Good accuracy

<table>
<thead>
<tr>
<th></th>
<th>Karate</th>
<th>Arxiv</th>
<th>Internet</th>
<th>Web nd.edu</th>
<th>Phone</th>
<th>Web uk-2005</th>
<th>Web</th>
<th>WebBase 2001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes/links</td>
<td>34/77</td>
<td>9k/24k</td>
<td>70k/351k</td>
<td>325k/1M</td>
<td>2.6M/6.3M</td>
<td>39M/783M</td>
<td></td>
<td>118M/1B</td>
</tr>
<tr>
<td>CNM</td>
<td>.38/0s</td>
<td>.772/3.6s</td>
<td>.692/799s</td>
<td>.927/5034s</td>
<td>-/-</td>
<td>-/-</td>
<td></td>
<td>-/-</td>
</tr>
<tr>
<td>PL</td>
<td>.42/0s</td>
<td>.757/3.3s</td>
<td>.729/575s</td>
<td>.895/6666s</td>
<td>-/-</td>
<td>-/-</td>
<td></td>
<td>-/-</td>
</tr>
</tbody>
</table>
| Our algorithm  | .42/0s | .813/0s | .781/1s  | .935/3s    | .769/134s | .979/738s   | .984/152mn
Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve

Very fast!

Good accuracy

On ad hoc networks with a well-known community structure:

Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve

Very fast!

Good accuracy

Resolution limit

Modularity Optimization fails to uncover communities under a certain scale

\[ n > \sqrt{L} \quad \Rightarrow \quad Q_{single} < Q_{pair} \]

Santo Fortunato and Marc Barthélemy, Resolution limit in community detection, PNAS 2007 104: 36-41
Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve

Very fast!

Good accuracy

Resolution limit

Our algorithm performs a local optimization and each step corresponds to a certain resolution
Advantages

Incredibly simple (local greedy approach) => easy to implement and to improve

Very fast!

Good accuracy

Multiresolution
Belgian Mobile Phone network

6 months of communications; One Belgian major operator.

2.6 millions customers;
language information (Dutch, English, French or German);
6.3 millions links (after filtering):
weight : number of calls + sms;
Belgian Mobile Phone network

6 months of communications; One Belgian major operator.

2.6 millions customers;
language information (Dutch, English, French or German);
6.3 millions links (after filtering):
weight: number of calls + sms;

Algorithm runs in 2 minutes
The more red (green), the higher (lower) is the percentage of french speaking people in the community
Belgian Mobile Phone network

6 months of communications; One Belgian major operator.

2.6 millions customers;
language information (Dutch, English, French or German);
6.3 millions links (after filtering):
weight : number of calls + sms;

All but two communities of size >10k are >93% segregated.
One (center) is made of people living in Brussels
Conclusion

- Can deal with millions/billions nodes/links
- Achieves very good modularity
- Directly produces a hierarchy structure;
- Is strikingly simple;
- Can use other local quality functions (instead of modularity);
- C++ code available on demand