Social and Technological Network Analysis

Lecture 4: Community Detection and Overlapping Communities

Prof. Cecilia Mascolo
Communities

• Weak ties (Lecture 2) seemed to bridge groups of tightly coupled nodes (communities)
• How do we find these communities?
In This Lecture

- We will describe a Community Detection method based on betweenness centrality.
- We will describe the concept of Modularity and Modularity Optimization.
- We will describe methods for overlapping community detection.
What is a Community?

Ideally such automatically detected clusters would then correspond to real groups.

For example: Communities, clusters, groups, modules.

4

11/10/2009 Jure Leskovec, Stanford CS322: Network Analysis
Why do we want to find partitions/communities?

• Clustering online customers with similar interests or geographically near can improve performance
  – Customers with similar interests could be clustered to help recommendation systems
• Clusters in large graphs can be used to create data structures for efficient storage of graph data to handle queries or path searches
• Study the relationship/mediation among nodes
  – Hierarchical organization study
Example

Zachary’s Karate club: 34 members of a club over 3 years. Edges: interaction outside the club.

WWW: pages and hyperlinks
Identification of clusters can improve page ranking
Remove weak ties

- Local bridges connect weakly interacting parts of the network.
- What if we have many bridges: which do we remove first? Or there might be no bridges.

- Note: *Without those bridges paths between nodes would be longer.*
Edge Betweenness

- Edge Betweenness: the number of shortest paths between pairs of nodes that run along the edge.
Algorithm of Girvan-Newmann (PNAS 2002)

- Calculate the betweenness of all edges
- Cut the edge with highest betweenness
- Recalculate edge betweenness
Edge: deletion

When do we stop?

• How do we know when to stop?

• When X communities have been detected?
• When the level of cohesion inside a community has reached Y?

• There is no prescriptive way for every case
• There are also many other ways of detecting communities.
• Perhaps a good measure of when to stop is when for each community the “cohesion” within the community is higher than what would be at random...

• $Q = (\text{edges inside the community}) - (\text{expected number of edges inside the community for a random graph with same node degree distribution as the given network})$
Modularity (2)

• Number of edges **inside** a community:

\[ \frac{1}{2} \sum_{a,b} A_{a,b} \delta(c_a, c_b) \]

• Where:
  • \( A_{a,b} \) is 1 if there is an edge \( a \rightarrow b \),
  • \( \delta(c_a, c_b) \) is the Kronecker Delta (1 if \( c_a \) is equal to \( c_b \))
Modularity on a randomized graph calculation

The expected number of edges in the randomized version of the graph where nodes are rewired:

\[ \frac{k_a k_b}{2m} \]

m is the number of edges of the graph = \( \frac{1}{2} \sum_{i} k_i \)
Modularity (3)

\[ Q_1 = \frac{1}{2} \sum_{a,b} A_{a,b} \delta(c_a, c_b) - \frac{1}{2} \sum_{a,b} \frac{k_a k_b}{2m} \delta(c_a, c_b) \]

\[ Q_1 = \frac{1}{2} \sum_{a,b} (A_{a,b} - \frac{k_a k_b}{2m}) \delta(c_a, c_b) \]

\[ Q = \frac{1}{2m} \sum_{a,b} (A_{a,b} - \frac{k_a k_b}{2m}) \delta(c_a, c_b) \]

Fraction of edges over all edges m
Modularity (4)

• Modularity ranges from -1 to 1.
  – It is positive if the number of edges inside the group are more than the expected number.
  – Variation from 0 indicate difference with random case.

• Modularity can be used at each round of the Girvan-Newmann algorithm to check if it is time to stop. However the complexity of this is $O(m^2n)$. 
Example of Dendrogram

FIG. 2: Dendrogram of the communities found by our algorithm in the “karate club” network of Zachary [5, 17]. The shapes of the vertices represent the two groups into which the club split as the result of an internal dispute.
Modularity Optimization

• Why not optimize modularity directly?

• Finding the configuration with maximum modularity in a graph is an NP complete problem.

• However there are good approximation algorithms.
Fast Modularity

• Start with a network of n communities of 1 node
• Merge the communities that lead to largest increase in Q
• Repeat previous step until one community remains
• Cross cut the dendrogram where Q is maximum.
• This runs in $O((m + n)n)$.

• A further optimization runs in $O(m \cdot d \cdot \log n)$ [d depth of dendrogram].
• Most networks are sparse so $m \sim n$ and $d \sim \log n$
Application to Amazon Recommendations

- Network of products.
- A link between product a and product b if b was frequently purchased by buyers of a.
- 200000 nodes and 2M edges.
- Max when 1684 communities
- Mean size of 243 products

FIG. 1: The modularity $Q$ over the course of the algorithm (the $x$ axis shows the number of joins). Its maximum value is $Q = 0.745$, where the partition consists of 1684 communities.
Amazon: Top Communities (87% of nodes)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>114538</td>
<td>General interest: politics; art/literature; general fiction; human nature; technical books; how things, people, computers, societies work, etc.</td>
</tr>
<tr>
<td>2</td>
<td>92276</td>
<td>The arts: videos, books, DVDs about the creative and performing arts</td>
</tr>
<tr>
<td>3</td>
<td>78661</td>
<td>Hobbies and interests I: self-help; self-education; popular science fiction, popular fantasy; leisure; etc.</td>
</tr>
<tr>
<td>4</td>
<td>54582</td>
<td>Hobbies and interests II: adventure books; video games/comics; some sports; some humor; some classic fiction; some western religious material; etc.</td>
</tr>
<tr>
<td>5</td>
<td>9872</td>
<td>classical music and related items</td>
</tr>
<tr>
<td>6</td>
<td>1904</td>
<td>children’s videos, movies, music and books</td>
</tr>
<tr>
<td>7</td>
<td>1493</td>
<td>church/religious music; African-descent cultural books; homoerotic imagery</td>
</tr>
<tr>
<td>8</td>
<td>1101</td>
<td>pop horror; mystery/adventure fiction</td>
</tr>
<tr>
<td>9</td>
<td>1083</td>
<td>jazz; orchestral music; easy listening</td>
</tr>
<tr>
<td>10</td>
<td>947</td>
<td>engineering; practical fashion</td>
</tr>
</tbody>
</table>

TABLE I: The 10 largest communities in the Amazon.com network, which account for 87% of the vertices in the network.
Amazon:
Community Size Distribution

- A power law distribution of community size
- (more on power laws in later lectures)
Limitations of Modularity

• Modularity is not a perfect measures
• It appears to depend on the number of links in the network (L).
• Problems for modules with a number of internal links of the order of $\sqrt{2L}$ or smaller.

• Intuition: modularity depends on links of a community to the “outside”, ie the rest of the network.

Louvain Method

- The Louvain method is more efficient and more accurate.
- Step 1: for each node $i$ consider neighbours ($j$) and evaluate gain in modularity of community if $i$ moves to $j$’s community. Do this for all nodes. Stop when no improvement can be achieved.
- Step 2: see each created community as a node and connect them with edges (could be weighted) and repeat step 1 on the network of communities (which are now the nodes). Stop when no modularity increase is obtained.
Efficiency

• Extremely faster than other algorithms
• Complexity is linear on typical and sparse data.
  – Possible gains in modularity are easy to compute and number of communities decreases drastically after a few steps.
Performance and Modularity results for various networks and approaches

<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>WT</td>
<td>.42/0s</td>
<td>.761/0.7s</td>
<td>.667/62s</td>
<td>.898/248s</td>
<td>.56/464s</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>
Louvain over a telecom network in Belgium

The colours are different languages spoken by people. The intermediate node is one with a lot of language mixing.

Edges are calls. Each of these communities are more than 100 people.
Overlapping Communities

- Community membership could overlap: a node could be part of more than 1 community.
Nodes can belong to more than 1 social circle!

- Scientists
- Mathematicians
- Biologists
- Department of Biological Physics
- Physicists
- "zoom"
- Scientific Community
- Hobby
- Family
- Friends
- Schoolmates
Clique Percolation Method: the idea (Palla 2005)

- Two nodes belong to the same community if they can be connected through adjacent $k$-cliques.
- A $k$-clique is a fully connected graph of $k$ nodes.
- $K$-cliques are adjacent if they have $k-1$ overlapping nodes.
- $K$-clique community: nodes which can be reached through a sequence of adjacent $k$-cliques.

University of Cambridge
Clique Percolation Method: The algorithm

• Find the maximal cliques
  – A maximal clique is a clique that cannot be extended by including one more adjacent vertex
  – This is complex but real networks are relatively sparse.

• Build clique overlap matrix
  – Each clique is a node
  – Connect two cliques if they overlap in at least $k-1$ nodes

• Communities:
  – Connected components of the clique overlap matrix
Maximal cliques

Erase elements less than 4 on diagonal and less than 3 elsewhere

Overlap Matrix: elements are n. of overlapping nodes

Example

k=4

K-cliques
Application

Overlapping networks:
1) Parisi’s coauthorship networks
2) Networks of “bright” in the word association network
3) Protein to protein interaction network
Application:
Phone Call Network

From Leskovec
Community Detection and Weak Ties

- Twitter was analyzed trying to identify if the static network of followers gives information about the dynamics of retweeting and mentioning.
- Dataset: follower network (undirected), 2M users, and network of tweets, mention and retweets for 1 month.
- Some community detection methods are used to find clusters in the follower network.
Sample

- Gray: followers
- Red: mentions
- Green: retweet
- 3 groups, one user between groups.
Some statistics

92,000 groups
Largest group: 10,000 users
37% users: no group

Mentions are double the followers in internal and bridging
Internal Links

Internal mentions are more than follower links with groups around 100.

The distribution of mentions over links is quite wide.

C: The dashed curves are the total for the follower network (black) and for the links with mentions (red). Others (from bottom to top): fractions of links with: 1 non-reciprocated mentions (diamonds), 3 mentions (circles), 6 mentions (triangle up) and more than 6 reciprocated mentions (triangle down).
Links between groups

Occur between groups of <200 nodes

\[
sim(A, B) = \frac{|\cap \text{links } A \cap \text{ links } B|}{|\cup \text{links } A \cap \text{ links } B|}
\]

Retweets seem to occur more between groups than within! Weak ties!!!!! Retweets also seem to happen between less similar groups!
Bridge Links

A x10^{-2}

B

link ratio

x10^{-4}

mentions retweets

between groups bridging internal

0 2 4 6 8 10 12

0 2 4 6 8 10

P(n\text{-}sg)

0 0.1 0.2 0.3 0.4 0.5

0 2 4 6 8 10 12

0 1 2 3

ratio

0 5 10 15 20

non-shared groups

Retweets on a bridge increase with the number of groups assigned to the bridging nodes

IV. MATERIALS AND METHODS

A. Description of the dataset

The data analyzed in this paper was collected in a two step process: the first stage corresponds to the collection of the follower network of followers and followees, while the second consists in the retrieval of the user activity from the stream of Twitter online tweets, mentions and retweets. In the first stages the directed unweighted network is obtained from the information on the followers and followees of each user. The data was collected using a breadth-first search technique: Starting from several seeds, followers and followees of the seeds were retrieved. Then the same procedure was repeated for the newly discovered users obtaining a so-called snowball sampling of the follower network. The procedure is stopped after several steps when the number of newly discovered users in the nth breadth is small compared with the total number of users already discovered in the n_{\text{step}}. The process was run in November 2008 gathering information for a total of 134,574 users. Due to the internal exploration of the networks one can anticipate that this method tends to detect the users with the highest in or out degree that belong to the largest connected cluster of the network.

The second stage consists in searching for all the tweets of the users found in the follower network for a period of time from November 2008 to December 2009. The activity dataset was constructed from these gathered tweets. The tweets containing usernames with an '@username' functional syntax were used for the mentions. Tweets that contain the following retweets:

\begin{itemize}
  \item mentions
  \item retweets
\end{itemize}
Discussion on findings

• There seems to be a correlation with the role of weak ties and the clustering done on the followers network
• Weak ties seem to be carrier of information (retweets) while internal group links seem to be more about mentions and communication
Summary

• We have discussed modularity based community detection as well as overlapping community detection.
• Many methods exist...

• We have shown cluster and weak ties analysis on an online social network dataset.
References

• S. Fortunato. **Community detection in graphs**, Arxiv 2009.