Overlapping Communities & Models of Community Structure in Large Networks

CS224W: Social and Information Network Analysis
Jure Leskovec, Stanford University
http://cs224w.stanford.edu
Non-overlapping vs. overlapping communities
Overlaps of social circles

- A node belongs to many social circles

[Palla et al., '05]
Two nodes belong to the same community if they can be connected through adjacent $k$-cliques:

- **$k$-clique:**
  - Fully connected graph on $k$ nodes

- **Adjacent $k$-cliques:**
  - overlap in $k-1$ nodes

- **$k$-clique community**
  - Set of nodes that can be reached through a sequence of adjacent $k$-cliques
Method: CPM: Steps

- **Clique Percolation Method:**
  - Find maximal-cliques (not \(k\)-cliques!)
  - 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

- **How to set \(k\)?**
  - Set \(k\) so that we get the “richest” (most widely distributed cluster sizes) community structure
CPM method: Example

- Start with graph and find maximal cliques
- Create clique overlap matrix
- Threshold the matrix at value $k-1$
  - If $a_{ij} < k-1$ set 0
- Communities are the connected components of the thresholded matrix

(1) Graph

(2) Clique overlap matrix

(3) Thresholded matrix at $k=4$

(4) Communities (connected components)
Example: Phone call network

Communities in a “tiny” part of a phone calls network of 4 million users
[Barabasi-Palla, 2007]
Example (2)

- Each node is a community
- Nodes are weighted for community size
- Links are weighted for overlap size
- DIP “core” data base of protein interactions (S. cerevisiase, yeast)
How to find max-cliques?

- No nice way, NP-hard combinatorial problem
- Simple Algorithm:
  - Start with max-clique size $s$
  - Choose node $u$, extract cliques of size $s$ node $u$ is member of
  - Delete $u$ and its edges
  - When graph is empty, $s = s - 1$, restart on original graph
Finding cliques around $u$ of size $s$:

- **2 sets $A$ and $B$:**
  - Each node in $B$ links to all nodes in $A$
  - Set $A$ grows by moving nodes from $B$ to it

- **Start with** $A = \{u\}$, $B = \{v: (u,v) \in E\}$

- **Recursively move each possible** $v \in B$ to $A$ and prune $B$
  - If $B$ runs out of nodes before $A$ reaches size $s$,
    - backtrack the recursion and try a different $v$
Let’s rethink what we are doing...

- Given a network
- Want to find clusters!

Need to:

- Formalize the notion of a cluster
- Need to design an algorithm that will find sets of nodes that are “good” clusters

More generally:

- How to think about clusters in large networks?
What is a good cluster?

- Many edges internally
- Few pointing outside

Formally, conductance:

$$\phi(S') = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min\{A(S), A(S')\}}$$

Where: $A(S)\ldots$ volume

$$A(S) = \sum_{i \in S} \sum_{j \in V} A_{ij}$$

Small $\Phi(S)$ corresponds to good clusters
Define:

Network community profile (NCP) plot

Plot the score of best community of size $k$

$$\Phi(k) = \min_{S \subseteq V, |S| = k} \phi(S)$$

Log $\Phi(k)$ vs. Community size, log $k$
NCP plot: Meshes

- Meshes, grids, dense random graphs:

\[ \phi(k) = \frac{1}{k^d} \]  

- d-dimensional meshes

- California road network
Collaborations between scientists in networks [Newman, 2005]
Natural hypothesis about NCP:
- NCP of real networks slopes downward
- Slope of the NCP corresponds to the dimensionality of the network

What about large networks?

Examine more than 100 large networks
Typical example: General Relativity collaborations (n=4,158, m=13,422)
More NCP plots of networks

(a) LIVEJOURNAL01

(b) MESSENGER-DE

(c) AT&T-DBLP

(d) CIT-HEP-TH

(e) WEB-GOOGLE

(f) AMAZONALL
NCP: LiveJournal \((n=5m, m=42m)\)

- Better and better communities
- Communities get worse and worse
- Best community has \(~100\) nodes
Each successive edge inside the community costs more cut-edges.

- \( \Phi = \frac{1}{3} = 0.33 \)
- \( \Phi = \frac{2}{4} = 0.5 \)
- \( \Phi = \frac{8}{6} = 1.3 \)
- \( \Phi = \frac{64}{14} = 4.5 \)

Each node has twice as many children.
Empirically we note that best clusters (call them whiskers) are barely connected to the network.

⇒ Core-periphery structure

If we remove whiskers.. How does NCP look like?

11/10/2010
What if we remove whiskers?

Nothing happens! \( \Rightarrow \) Nestedness of the core-periphery structure
Nested core-periphery

Denser and denser network core

Small good communities
Cluster size is independent of the network size

Each dot is a different network

Practically constant!
Comparison with “Ground truth”

LiveJournal

DBLP

Amazon

IMDB

Rewired

Network

Ground truth
Some issues with community detection:

- Many different formalizations of clustering objective functions
- Objectives are NP-hard to optimize exactly
- Methods can find clusters that are systematically “biased”
- Methods can perform well/poorly on some kinds of graphs

Questions:

- How well do algorithms optimize objectives?
- What clusters do different methods find?
Many algorithms to extract clusters:

- **Heuristics:**
  - Girvan-Newman, Modularity optimization: popular heuristics
  - Metis (multi-resolution heuristic): common in practice [Karypis-Kumar ‘98]

- **Theoretical approximation algorithms:**
  - Spectral partitioning: most practical but confuses “long paths” with “deep cuts”
    - Local Spectral [Andersen-Chung ‘07]
  - Leighton-Rao: based on multi-commodity flow
Spectral vs. Flow methods

- Spectral embeddings stretch along directions in which the random-walk mixes slowly
  - Resulting hyperplane cuts have "good" conductance cuts, but may not yield the optimal cuts

spectral embedding  flow based embedding
NCP: Live Journal

Rewired network

Local spectral

Metis

\[ \varphi \text{ (conductance)} \]

\[ n \text{ (number of nodes in the cluster)} \]
Properties of clusters (1)

500 node communities from **Local Spectral**:

500 node communities from **Metis**:
Properties of clusters (2)

- **Metis** (red) gives sets with better conductance
- **Local Spectral** (blue) gives tighter and more well-rounded sets
Other clustering methods

- LeightonRao: based on multi-commodity flow
  - Disconnected clusters vs. Connected clusters
- Graclus prefers larger clusters
- Newman’s modularity optimization similar to Local Spectral
Many different objective functions

- **Single-criterion:**
  - Modularity: $m-E(m)$
  - Modularity Ratio: $m-E(m)$
  - Volume: $\sum_u d(u) = 2m + c$
  - Edges cut: $c$

- **Multi-criterion:**
  - Conductance: $c/(2m+c)$
  - Expansion: $c/n$
  - Density: $1-m/n^2$
  - CutRatio: $c/n(N-n)$
  - Normalized Cut: $c/(2m+c) + c/2(M-m)+c$
  - Max ODF: $\text{max frac. of edges of a node pointing outside } S$
  - Average-ODF: $\text{avg. frac. of edges of a node pointing outside}$
  - Flake-ODF: $\text{frac. of nodes with mode than } \frac{1}{2} \text{ edges inside}$

$n$: nodes in $S$
$m$: edges in $S$
c$: edges pointing outside $S$
Multi-criterion objectives

- Qualitatively similar
- Observations:
  - Conductance, Expansion, Norm-cut, Cut-ratio and Avg-ODF are similar
  - Max-ODF prefers smaller clusters
  - Flake-ODF prefers larger clusters
  - Internal density is bad
  - Cut-ratio has high variance
Single-criterion objectives

Observations:
- All measures are monotonic
- Modularity
  - prefers large clusters
  - ignores small clusters
Lower and upper bounds

- **Lower bounds** on conductance can be computed from:
  - Spectral embedding (independent of balance)
  - SDP-based methods (for volume-balanced partitions)

- Algorithms find clusters close to theoretical lower bounds
Nested core-periphery

Denser and denser network core

So, what’s a good model?

Small good communities

Nested core-periphery
Modeling nested core-periphery

- None of the common models works!
Forest Fire model works!

- **Forest Fire**: connections spread like a fire
  - New node joins the network
  - Selects a seed node
  - Connects to some of its neighbors
  - Continue recursively

**Model ingredients:**
- **Preferential attachment**: second neighbor is not uniform at random
- **Copying flavor**: since we burn seed’s neighbors
- **Hierarchical flavor**: burn around the seed node
- “Local” flavor: burn “near” the seed node

As cluster grows it blends into the core of the network
Forest Fire NCP plot

\[ \phi \text{ (conductance)} \]

\[ 10^0 \quad 10^{-1} \quad 10^{-2} \quad 10^{-3} \]

\[ 10^0 \quad 10^1 \quad 10^2 \quad 10^3 \quad 10^4 \quad 10^5 \quad 10^6 \]

\( k \) (number of nodes in the cluster)