Network Analysis

Wei Pan (& Xiaotong Shen)

Division of Biostatistics and Health Data Science, School of Public Health,
University of Minnesota, Minneapolis, MN 55455
Email: panxx014@umn.edu

PubH 8475/Stat 8056

Outline

- Network concepts/statistics
- Network community detection

Basics

- Network/graph $\mathcal{G} = (V, E)$:
  - A set of nodes/vertices $V = \{v_1, \ldots, v_p\}$.
  - A set of edges/links between nodes $E = \{e_1, \ldots, e_m\}$.
- $p \times p$ adjacency matrix A: $A_{ij} = 0$ or 1 (or w_i) for a binary (or weighted network).
  Undirected network: A is symmetric
- Density: the proportion of edges present in a graph, i.e. (# edges)/(the maximum possible number of edges).
- The number of edges observed is $|E|$.
- The number of possible edges is
  - $p(p-1)/2$ in an undirected graph
  - $p(p-1)$ in a directed graph

Nodal Degree

- Nodes vary in their involvement in the network. For binary relations, this heterogeneity can be summarized by the nodal degree.
  - Undirected relation:
    - The degree of a node is the node's number of edges.
  - Directed relation:
    - The outdegree of a node is the node's number of outgoing edges/links.
    - The indegree of a node is the node's number of incoming edges.

Nodal Degree

- The degrees are easy to calculate with adjacency matrix A:

$$\begin{aligned} d_i^o &= \sum_{j:j\neq i} A_{ij} \\ d_i^i &= \sum_{j:j\neq i} A_{ji} \end{aligned}$$

- This calculation works for both directed and undirected relations. Specifically, for an undirected relation,

$$d_i^o = d_i^i = d_i$$

Summary of Degrees

- Let $d = \{d_1, \ldots, d_p\}$ be a set of nodal degrees (either out-degrees, in-degrees, or undirected degrees).
  - The entries of d are often summarized further with
  - Mean: $\bar{d} = \sum d_i / p = (p-1)\bar{U}$
  - Variance: $s_d^2 = \sum (d_i - \bar{d})^2 / (p-1)$
  - Degree distribution is a set of counts $\{f_0, \ldots, f_{p-1}\}$ where

$$f_k = \#\{d_i = k\} = \text{ number of nodes with degree equal to } k$$

Some concepts

- Node centrality:
  - measures "importance" of a node in a network: e.g., deletion of which genes in a gene regulatory network is likely to be lethal to the corresponding organism; how critical is a given router in an Internet network to the flow of traffic...
  - some common centrality measures:
    - closeness: $1/\sum_{u \in V} \mathrm{dist}(u, v)$; "central" means the node is "close" to many other nodes;
    - betweenness: $\sum_{s,t \in V} \sigma(s, t|v)/\sigma(s, t)$, where $\sigma(s, t|v)$ is the total number of shortest paths between s and t that pass through v; measures the extent to which a vertex is located "between" other pairs of vertices;
    - ...

Some concepts

- Network cohesion:
  - measures the extent to which subsets of nodes are cohesive / stuck together; e.g., do friends of a given actor in a social network tend to be friends of one another as well; what collections of proteins in a cell appear to work closely together...
  - Some common cohesion measures:
    - clique: A clique is a complete subgraph.
    - maximal clique: is a clique but no other nodes can be added to make it a larger clique.
    - density of a subgraph
  - Connectivity
    - "small worlds" property: the average distance between nodes is small

Network community detection: Outline

- Introduction
- Spectral clustering
- Hierachical clustering
- Modularity-based methods
- Model-based methods
- Key refs:
  1.Newman MEJ
  2. Zhao Y, Levina E, Zhu J (2012, Ann Statist 40:2266-2292).
  3. Fortunato S (2010, Physics Reports 486:75-174).
- R package `igraph`: drawing networks, calculating some network statistics, some community detection algorithms, ...

Introduction

- Given a binary (undirected) network/graph: $G = (V, E)$, $V = \{1, 2, ..., n\}$, set of nodes; E, set of edges.
  Adjacency matrix $A = (A_{ij})$: $A_{ij} = 1$ if there is an edge/link b/w nodes i and j; $A_{ij} = 0$ o/w. $(A_{ii} = 0)$
- Goal: assign the nodes into K "homogeneous" groups.
  often means dense connections within groups, but sparse b/w groups.
- Why? Figs 1-4 in Fortunato (2010).
  Brain networks (Meunier et al, 2010).
  Gene networks (Langfelder and Horvath, 2008).

Spectral clustering

- Laplacian $L = D - A$, or ...
  $D = \text{Diag}(D_{11}, ..., D_{nn})$, $D_{ii} = \sum_j A_{ij}$.
- Intuition:
  If a network separates perfectly into K communities, then L (or A) is block diagonal (after some re-ordering of the rows/columns).
  If not perfectly but nearly, then the eigenvectors of L are (nearly) linear combinations of the indicator vectors.
- Apply K-means (or ..) to a few (K) eigenvectors corresponding to the smallest eigenvalues of L.
  Note: the smallest eigen value is 0, corresponding to eigenvector 1.
- Two clusters $\Longrightarrow$ spectral bisection: use the eigenvector of the second smallest eigen value; partition by its positive/negative elements.
  Generally, repeatedly apply the above to each cluster... vs apply SC once?
- Widely used; some theory (e.g consistency).

Modified spectral clustering

- SC may not work well for sparse networks.
- Regularized SC (Qin and Rohe): replace D with $D_\tau = D + \tau I$ for a small $\tau > 0$.
- SC with perturbations (Amini, Chen, Bickel, Levina, 2013, Ann Statist 41: 2097-2122):
  regularize A by adding a small positive number on a random subset of off-diagonals of A.

Hierarchical clustering

- Need to define some similarity or distance b/w nodes.
- Euclidean distance: $A_{i.} = (A_{i1}, A_{I2}, ..., A_{in})'$,

$$x_{ij} = ||A_{i.} - A_{j.}||_2$$

- Or, Pearson's corr,

$$x_{ij} = \text{corr}(A_{i.}, A_{j.})$$

- Then apply a hierarchical clustering.
  can be used to re-arrange the rows/columns of A to get a nearly block-diagonal A.
- Fig 3 in Neuman.
- Fig 2 in Meunier et al (2010).

Algorithms based on edge removal

- Divisive: edges are progressively removed.
- Which edges? "bottleneck" ones.
- *edge betweenness* is defined to be the number of shortest paths between all pairs of all nodes that run through the two nodes.
- Algorithm (Girvam and Neuman 2002, PNAS):
  1) calculate *edge betweenness* for each remaining edge in a network;
  2) remove the edge with the higest *edge betweenness*;
  3) repeat the above until ...
- A possible stopping critarion: *modularity*, to be discussed.
- Fig 4 in Neuman.
- Remarks: slow; some modifications, e.g. a Monte Carlo version in calculating *edge betweenness* using only a random subset of all pairs; or use a different criterion.
- R package `igraph`: `cluster_edge_betweenness()`

Modularity-based methods

- Notation:
  degree of node i: $d_i = D_{ii} = \sum_{j=1}^{n} A_{ij}$,
  (*twice*) total number of edges: $m = \sum_{i=1}^{n} d_i$,
  Community assignment: $C = (C_1, C_2, ..., C_n)$; **unknown**,
  $C_i \in \{1, 2, ..., K\}$: community containing node i.
- Modularity: given C,

$$Q = Q(C) = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{m} \right) I(C_i = C_j).$$

- Intuition: obs'ed - exp'ed
- Goal: $\hat{C} = \arg\max_C Q(C)$
  Assumption: good to maximize Q, reasonable but ...

- Key: a **combinatorial** optimization problem!
  seeking exact solution will be too slow $\Longrightarrow$ many *approximate* algorithms, such as greedy searches (e.g. genetic algorithms, simulated annealing), relaxed algorithms, ...
  Newman (2003): repeat: combining two nodes i and j with $A_{ij} = 1$ and the largest increase (or smallest decrease) in Q; until all nodes in one community.
  $\Longrightarrow$ hierarchical; choose one with the largest Q.
- Very nonparametric?!
- Problems: resolution limit; too many local solutions.
  cannot detect relatively small communities; why? an implicit null model for the *whole network* (Fortunato 2010, p.40).
- R package `igraph`:
  greedy search, approx./fast: `cluster_fast_(`);
  combinatorial search, exact/slow: `cluster_optimal()`;
  heuristic, hierarchical communities for large networks (e.g. millions of nodes); see Blondel et al (2008) in the manual: `cluster_louvain()`.

Model-based methods

- Stochastic block model SBM (Holland et al 1983):
  1) a $K \times K$ probability matrix P;
  2) $A_{ij} \sim \text{Bin}(1, P_{C_i, C_j})$ independently.
- Simple; can model dense/weak within-/between-community edges.
  But, treat all nodes/edges in a community equally; cannot model *hub* nodes!
  Scale-free network: node degree distribution $Pr(k)$ is heavy-tailed; a power law.
- SBM with $K = 1$: Erdos-Renyi Random Graph.
- Degree-corrected SBM (DCSBM) (Karrer and Newman 2011):
  1) P; each node i has a degree parameter θ_i (with some constraints for identifiability);
  2) $A_{ij} \sim \text{Bin}(1, \theta_i \theta_j P_{C_i, C_j})$ independently

- More notations:
  $n_k(C) = \sum_{i=1}^{n} I(C_i = k)$, number of nodes in community k;
  $O_{kl} = \sum_{i,j=1}^{n} A_{ij} I(C_i = k, C_j = l)$, number of edges b/w communities $k \neq l$;
  $O_{kk} = \sum_{i,j=1}^{n} A_{ij} I(C_i = k, C_j = k)$, (twice) number of edges within community k;
  $O_k = \sum_{l=1}^{K} O_{kl}$, sum of node degrees in community k;
  $m = \sum_{i=1}^{n} d_i$, (twice) the number fo edges in the network.
- Objective function: A profile likelihood (profiling out nuisance parameters P and θ's based on a Poisson approximation to a binomial).
  Given a likelihood $L(C, P)$,
  a profile likelihood $L^*(C) = \max_P L(C, P) = L(C, \hat{P}(C))$.

- SBM:

$$Q_{SB}(C) = \sum_{k,l=1}^{K} (O_{kl} \log \frac{O_{kl}}{n_k n_l}).$$

- DCSBM:

$$Q_{DC}(C) = \sum_{k,l=1}^{K} (O_{kl} \log \frac{O_{kl}}{O_k O_l}).$$

- Neuman-Girvan modularity:

$$Q_{NG}(C) = \frac{1}{2m} \sum_{k} (O_{kk} - \frac{O_k^2}{m}).$$

- Remarks: Still a combinatorial optimization problem; better theoretical properties.
- Numerical examples in Zhao et al (2012).

Other topics

- Weighted networks; with or without negative weights (e.g. Pearson's correlations).
- Overlapping communities.
- Time-varying (dynamic) networks.
- With covariates. How to model covariates?
- Fast (approximate) algorithms; theory.
- WGCNA (Weighted Gene Co-expression Network Analysis): Langfelder and Horvath (2008, BI).