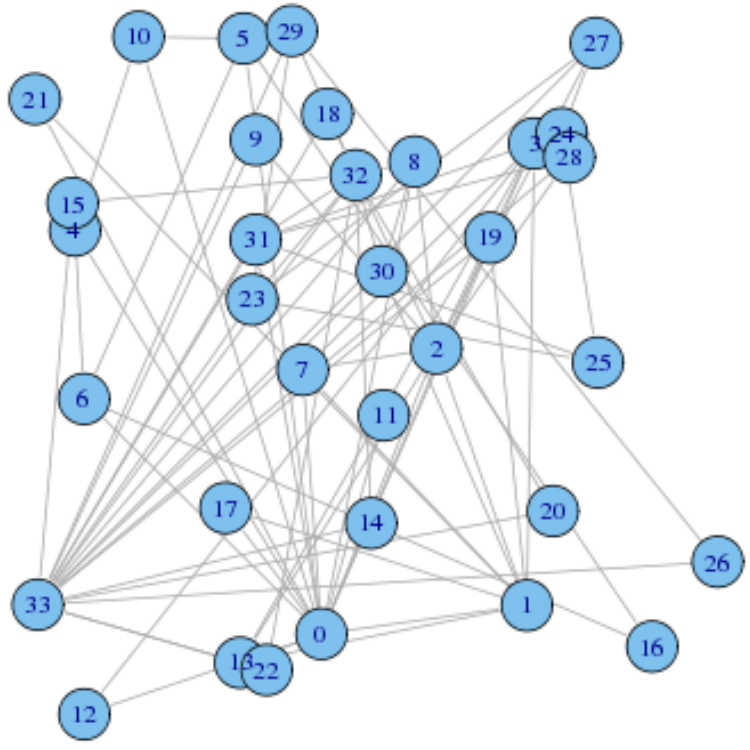
Community Detection

Data Analytics - Community Detection Module

Zachary's karate club

- Members of a karate club (observed for 3 years). Edges represent interactions outside the activities of the club.



Community detection in Zachary's karate club

- At some point, a fissure developed, and the group split into two factions.
  Can you predict the factions?
- Picture (Fortunato, p.5, Fig. 2.a)
- Discussion.
  Some links have been pruned (not sure why).
  Two clusters. One around '1' who was the Instructor.
  One around '33' and '34', the latter was president of the club.

Collaborations of scientists at Santa Fe Institute

- 118 vertices of resident scientists and their collaborations.
- An edge if they have published at least one paper together.
- Picture (Fortunato, p.5, Fig. 2.b)
- Discussion:
  Lay-out to portray subject-wise classification.
  Many cliques. Why?
  Reasonably well-separated with very few cross-links.

Dolphins at Doubtful Sound (Lusseau 2003)

- A network of 62 bottlenose dolphins living around Doubtful Sound (New Zealand).
- Nodes: Dolphins. Edge: if seen together at more often than random chance meetings.
- Picture (Fortunato, p.5, Fig. 2.c)
- Discussion:
  One of the dolphins was away for some time, and the group split into two.
  Lusseau provided a biological classification (squares vs. circles).
  Colours indicate a classification by one algorithm; matched the biological classification
  Small number of cross-linkages.
  Again, very useful as a benchmark.

Protein-protein interactions

- PPI in cancerous cells of rats.
- Nodes: proteins. Edge: if they are contact during biochemical events.
- Picture (Fortunato, p.7, Fig. 3)
- Discussion:
  Communities = functional group involved in similar processes.
  Most communities associated with cancer and metastasis.

Abstraction

Given a graph (nodes and edges), partition the graph into components, subsets of nodes, such that each subset is strongly interconnected with comparatively fewer edges across subsets.

Complications

- Edges may be directed.
- World Wide Web
  - Hyperlinks may be asymmetric.
  - Page A may provide a link to Page B. But Page B may not have a link to Page A.
  - Less than 10% of links are reciprocal.
- Picture (Fortunato, p.7, Fig. 4).
- We could neglect directedness since a link implies related content. But we are perhaps losing something.

Complications

- Nodes could belong to multiple communities.
- Example: Word association.
  - Give a word 'bright'. Ask subjects to identify associated words.
    Give these words to subjects, identify associated words, ...
    Build the network.
  - Is there a community structure in the words?
- Nodes: Words. Edge: If people associate two words.
- Picture (Fortunato, p.7, Fig. 5).
- Discussion:
  Four categories: Astronomy, Colours, Intelligence, Light.
  'Bright' belongs to all four of them.
  We are looking for a 'cover' instead of a partition.

Complications continued.

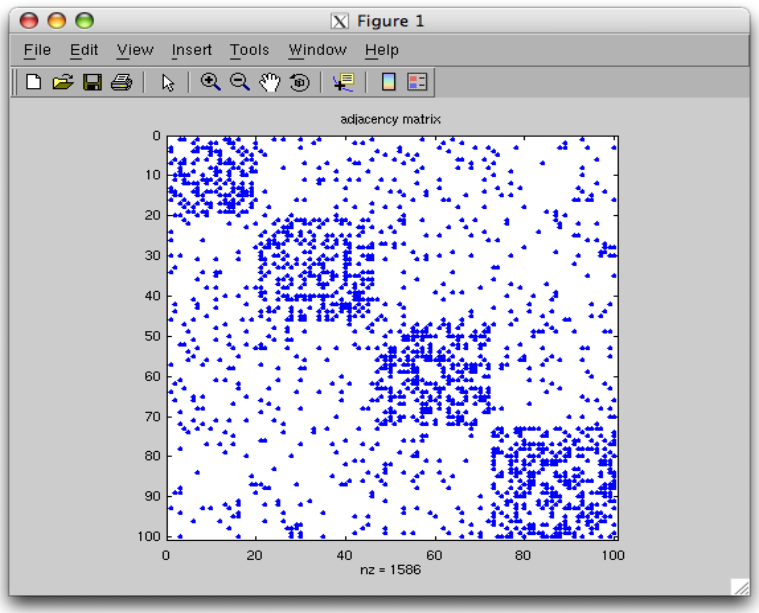
- Bipartite or multipartite graph.
- Example: Southern Women Event Participation (Davis et al., 1941).
  - 18 women from Natchez, Mississippi, attended 14 social events.
  - Nodes: Open nodes – women. Closed nodes – social events.
  - Edge if a woman attended an event.
- Picture: (Fortunato, p.8, Fig. 6).
- Discussion:
  Four groups identified by one algorithm.
  Can consider a different graph with only women nodes, edge if they attended the same event.
  But a lot of contextual information, that many attended the same event, is lost.

Plan

- Today we will cover the spectral method.
- Next two lectures:
  - Performance metrics.
  - Other algorithm classes.

Towards spectral method: Adjacency matrix

- For a graph $G = (V, E)$, $A =$ adjacency graph defined by $A_{i,j} = 1$ if i and j are connected. Symmetric.
- If the graph has two components, there is a permutation of the vertices such that A is block diagonal with two blocks.
- If some additional cross-linkages, we would see sparse nonzeros outside the blocks.



- Of course, we see only a permutation. Is there a way invert the permutation to get back to the (nearly) block diagonal A?

A probabilistic model and ML detection

- Stochastic block model: Consider two equal-sized communities. Suppose that within community edges are i.i.d. with probability p. Outside community edges are i.i.d. with probability $q < p$. Erase community labels and show only edges.
- Problem: Identify the two communities.
- Approach: Maximum likelihood detection.

$$\max_{v \in \{-1,1\}^n : \langle \mathbf{1}, v \rangle = 0} v^T A v$$

- Proof on the board.
- $v = \mathbf{1}_S - \mathbf{1}_{S^c}$. So $\langle \mathbf{1}, v \rangle = 0$ ensures that the partition is balanced.

Laplacian

- d_i = degree of node i. D = diagonal matrix of degrees.
- Laplacian $L = D - A$.
- L is symmetric and positive semidefinite.
- 0 is an eigenvalue with eigenvector $\mathbf{1}$.
- Generalisation: Edge weights and associated Laplacian.
- If $v = \mathbf{1}_S - \mathbf{1}_{S^c}$, then

  $v^T D v = \sum_i v_i^2 d_i$ = sum of degrees = twice number of edges.
- So we can write an equivalent objective:

$$\min_{v \in \{-1,1\}^n : \langle \mathbf{1}, v \rangle = 0} v^T L v = \min_{(S, S^c),\ \textit{balanced}} 4\ \textit{cut}(S, S^c).$$

- NP-hard optimisation problem. Heuristics.

Relaxation and lift approaches

- Spectral: Relax $v \in \{-1, 1\}^n$ to $v \in \mathbb{R}^n$

$$\begin{aligned} & \min && v^T L v \\ \text{subject to} \quad & && ||v|| = 1 \\ & && \langle \mathbf{1}, v \rangle = 0. \end{aligned}$$

- SDP: $v^T A v = \text{trace}(A v v^T)$. So let $V = v v^T$.

$$\begin{aligned} & \max && \text{trace}(AV) \\ \text{subject to} \quad & && \mathbf{1}^T V = 0 \qquad \text{(Relax to trace}(\mathbf{1}\mathbf{1}^T V) = 0) \\ & && V \succeq 0 \\ & && V_{ii} = 1 \quad \text{for all } i \\ & && \text{rank}(V) = 1. \qquad \text{(Relax this).} \end{aligned}$$

Spectral method solution: Fiedler vector

- We could look for vectors v that minimise $v^T L v$ among all unit vectors v orthogonal to $\mathbf{1}$.
- Write $v = \sum_{i=1}^{n} a_i u_i$, where u_i are the eigenvectors of L. So

$$v^T L v = \sum_{i=1}^{n} \lambda_i a_i^2.$$

  with $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.
- The minimising value is λ_2. The corresponding vector u_2 (called Fiedler vector) solves

$$\min_{v:\langle v,\mathbf{1}\rangle=0,||v||=1} v^T L v = \min_{v:\langle v,\mathbf{1}\rangle=0,||v||=1} \sum_{(i,j)\in E} (v_i - v_j)^2.$$

- Use u_2 as a surrogate for $\frac{1}{\sqrt{n}}(\mathbf{1}_S - \mathbf{1}_{S^c})$.
- Use sign of u_2, or cluster its entries into two groups, or rank order and pick the top k (if number is known).

Normalised Laplacian

- One could also consider the normalised Laplacian:

$$L_{norm} = D^{-1}L = I - D^{-1}A.$$

- This has the advantage that $D^{-1}A$ is a stochastic matrix, and one can construct an associated random walk (a Markov process). L_{norm} is the generator for the Markov process.

- 0 is an eigenvalue of both L and L_{norm} with eigenvector $\mathbf{1}$.

- What if there are 2 (or more) components?

Spectrum of the Laplacian and components

Theorem

Let G be an undirected (possibly weighted) graph. Let L_{norm} be its normalised Laplacian. Let k be the multiplicity of the eigenvalue 0. Then

- *The number of connected components is k.*
- *The eigenspace of 0 is spanned by the indicators on the components.*

Idea:

- If the graph has k components, then perfectly identified by clustering.
- If A has cross-linkages, but relatively small in number, the eigenvalues get perturbed, but perhaps not by much.
- Exploit it.

A more general spectral algorithm

Input: Adjacency matrix A and number of components k.

- Compute the normalised Laplacian $I - D^{-1}A$.
- Find the k smallest eigenvalues and eigenvectors.

$$X = [u_1 \ u_2 \ \ldots \ u_k].$$

- Identify node i with the ith row of X.
- Cluster the n points in R^k using a 'data clustering' algorithm. (Say via k-means algorithm (Lloyd's algorithm, Lloyd-Max algorithm).)
- Output : Clusters of the 'data clustering' algorithm.

Data clustering

- Suppose we are given points $x_1, x_2, \ldots, x_v \in \mathbb{R}^r$.
- Points in a metric space, with a notion of distance. r was k in the previous page.
- Cluster the points into k groups.
- Many related algorithms.

Data clustering algorithms

- Minimum k-clustering: Objective is to minimise over partitions $S_1, \ldots, S_k$:

$$\max_{1 \leq i \leq k} diam(S_i).$$

- k-clustering sum:

$$\max_{1 \leq i \leq k} d_{ave}(S_i)$$

- k-centre clustering:

$$\max_{1 \leq i \leq k} \max_{l \in S_i} d(x_l, c_i),$$

  where c_i is the centre of S_i.

- k-median clustering:

$$\max_{1 \leq i \leq k} \frac{1}{|S_i|} \sum_{l \in S_i} d(x_l, c_i^*),$$

  where c_i^* is the median of S_i.

A popular variant: k-means clustering

- k-means clustering:

$$\sum_{i=1}^{k} \sum_{l \in S_i} d(x_l, \overline{c}_i)^2.$$

  where $\overline{c}_i$ is the centroid of S_i.

- A natural iterative block coordinate descent approach:

  Start with some initial candidate centroids.

  - Given the centroids, find the best partition.
  - For each partition, find new centroids.
  - Repeat until convergence or max number of iterations.

Each of the individual steps is easy

- Given the centroids, find the best partition $S_1, \ldots, S_k$.

$$\sum_{i=1}^{k} \sum_{l \in S_i} d(x_l, \overline{c}_i)^2 = \sum_{l} \sum_{i=1}^{k} \mathbf{1}_{S_i}(l) d(x_l, \overline{c}_i)^2.$$

- Which cluster to associate an l with? Nearest neighbour.
- Given the clusters $S_1, \ldots, S_k$, where should the centroids be?

$$\min_{c_i} \sum_{l \in S_i} d(x_l, c_i)^2 = \min_{c_i} \sum_{l \in S_i} ||x_l - c_i||^2$$

$\overline{c}_i$, the centroid, is the best choice.

Issues

- Objective function always goes down. Lower bounded by zero. So convergence of the objective function is clear.
- Could be a local minimum.
- Multiple restarts alleviates the problem to some extent.

Hierarchical clustering: agglomeration

- $A_{i,j} = 1$ indicates that i and j are loosely speaking 'similar'.
- Let us say we have a measure of 'similarity' $s(i, j)$ instead of $A_{i,j} = 1$.
- Meta-algorithm for agglomerative clustering:
  Initialise: Each vertex is a separate cluster.
  - Identify a measure of "similarity between clusters" $\tilde{s}(C_l, C_k)$.
  - Merge two clusters with the greatest similarity.
  - Repeat until we arrive at a single cluster.
- Picture: Dendrogram (Fortunato, p. 14, Fig. 8).
- Number of clusters determined based on large drops in similarity at mergings.

Measures of similarity of clusters

- Single linkage:

$$\tilde{s}(C_l, C_k) = min_{i \in C_l, j \in C_k} s(i, j)$$

- Complete linkage: Maximum

- Average linkage

Hierarchical clustering: division

- Picture (Fortunato p.23, Fig. 10)
- Meta-algorithm for divisive clustering:

  Initialise: $G^{(0)} = G$. Iterate $t = 0$.

  - Compute "centrality" of all edges in $G^{(t)}$.
  - Set $G^{(t+1)}$ to be the graph $G^{(t)}$, but with the edge having the greatest centrality removed.
  - Increment t and repeat until all vertices belong to separate clusters.

Measures of edge centrality

- Geodesic edge betweenness:
  - $\sum_{(i,j)} w_{i,j}(e)$
  - $w_{i,j}(e)$ is the weight of e for shortest paths from i to j.
  - If there are many shortest paths from i to j, the fraction that passes through e.
- Computation time is $O(mn)$ via breadth-first search.

More measures of edge centrality

- Random walk betweenness:
  - Fix s and t. Start a random walk at s.
  - From each node, pick one of the edges uniformly at random, and traverse that edge.
  - Stop as soon as you reach t.
  - Random walk betweenness of $e = \sum_{s,t} \Pr\{\textit{Walk from } s \textit{ to } t \textit{ traverses } e\}$.
- More computationally intensive. $O(mn^2)$.
- Complexity of approximations?

More measures of edge centrality

- Current flow betweenness:
  - Fix s and t. Apply a 1 volt difference between s and t.
  - Current flow betweenness of $e = \sum_{s,t} |\textit{Current through } e|$.
- Same as random walk betweenness since the equations satisfied by the probability of traversal through e and the current through e are the same.

More measures of edge centrality

- Effective resistance of edge $e = (k, l)$.
  - Inject 1 A of current through k and extract through l.
  - The voltage difference $v(k) - v(l)$ is the effective resistance.
  - Higher the resistance, more central is that edge.

An expression for effective resistance

- Write $B = m \times n$ matrix of edge-vertex incidence matrix.
- Row $B(e, \cdot) = (\mathbf{1}_k - \mathbf{1}_l)^T$.
- $L = B^T W B$: Laplacian for graph with edge-weights $w(e)$.
  W is the diagonal matrix of edge weights.
- i : currents through edges, i_{ext} : external currents into nodes.
- KCL: $i_{ext} = B^T i$.
  KVL: $i = WBv$.
  $i_{ext} = B^T WBv = Lv$.
- Suppose we have a connected graph, since $\langle i_{ext}, \mathbf{1} \rangle = 0$, injected current is extracted, then $v = L^\dagger i_{ext} = L^\dagger(\mathbf{1}_k - \mathbf{1}_l)$.
- $v(k) - v(l) = (\mathbf{1}_k - \mathbf{1}_l)^T v = (\mathbf{1}_k - \mathbf{1}_l)^T L^\dagger i_{ext} = (\mathbf{1}_k - \mathbf{1}_l)^T L^\dagger (\mathbf{1}_k - \mathbf{1}_l)$.
  Interpret: Euclidean distance between points in an embedding of the graph in $\mathbb{R}^n$.
- There are algorithms that compute effective resistance to within a factor $(1 \pm \varepsilon)$ in time $O(m(logn)/\varepsilon^2)$ steps.

Modularity of Girvan and Newman 2004

- Modularity measures the goodness of a partition.

$$Q := \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \mathbf{1}(C_i = C_j)$$

where C_i is i's cluster,
m is the number of edges in the graph, and
P_{ij} is the expected no. of edges between i and j in a 'null model'.

- Example null models: random graph, random graph under the 'configuration model' (prescribed degree sequence), $P_{ij} = d_i d_j / 2m$.

- Alternative expressions for Q under the configuration model:

$$Q = \sum_{c:\ cluster} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right],$$

where l_c/m is the fraction of edges within cluster, and $d_c/2m$ is the fraction of edges involving vertices in the cluster.
(d_c is the sum of degrees of vertices in the cluster c).

Another agglomerative approach

- Choose a partition that maximises modularity.
- Example: GREEDY algorithm:
  - Initialise: Each vertex a community by itself. $Q(0) < 0$.
  - At each stage: Choose an edge, to merge communities, that maximises ΔQ.
- Remarks:
  - When two communities along an edge are merged, number of communities may change. $Q(t)$ is computed on the original graph for the clustering at time t.
  - Internal edge does not change Q, since clusters don't change.
  - External edge reduces number of clusters by 1. Need to recompute $Q(t+1)$.
  - A naive implementation requires $O(m)$ for which edge $+O(n)$ for updating d_c. This is done for $O(n)$ iterations yielding $O((m+n)n)$.
  - Better algorithms available $O(md \log n)$ where $d =$ depth of dendrogram.

Louvain method

(0) Each node in its own community.

(1) For each node, identify the $(\Delta Q)_{ij}$ when i is removed from its current community and added to the community of a neighbour j. Move i to the community providing the largest modularity increase. Stop when no such increase is possible.

(2) Create a new network.
Merge nodes within a cluster. (Self loops for within community edges, weighted links across clusters.)

(3) Repeat Step 1.

A useful benchmark: back to stochastic block model

Stochastic block model or Planted partition model:

- Mark each vertex with label 0 or 1 independently and uniformly at random.
- Include each edge independently:
  - with probability p if between vertices with the same label,
  - with probability q if the vertices have different labels.
- Exactly solvable if fraction of recovered nodes is 1 with high probability (probability tending to 1 as $n \to \infty$).

Some striking results

- Fix p and q with $p > q$. Let $n \to \infty$.
  - Exactly solvable via min-bisection – two equal sized graphs with minimum cut (Dyer and Frieze).
    Average running time is $O(n^3)$.
  - Or use the ML or EM algorithm (Snijders and Nowicki).

- $p - q$ can shrink with n, and yet we can recover the partition!
  - Take $p = (a \log n)/n$ and $q = (b \log n)/n$.
  - Exactly solvable if and only if $|\sqrt{a} - \sqrt{b}| \geq \sqrt{2}$.
    (Mossel et al.; Massoulie; Bordenave et al.)
  - Spectral method, on the so-called "non-backtracking" matrix.

Even more striking ...

Consider the sparser regime $p = a/n$ and $q = b/n$. Here we ask for weak recovery - accuracy must exceed $0.5 + \varepsilon$.

- If $(a-b)^2 < 2(a+b)$, clustering problem not solvable. (Mossel, Neeman, Sly.)
  - Indeed, fix two vertices. Suppose we see the graph and know the first vertex's community.
    The probability that the second vertex belongs to the same community approaches $1/2$.
  - Cannot even estimate a and b consistently.
  - Connection to multi-type branching process, and label recovery.
- If $(a-b)^2 > 2(a+b)$, weak recovery possible with probability approaching 1 as $n \to \infty$.
  (Mossel et al.; Massoulie; Bordenave et al.; Abbe and Sandon).
  - Acyclic belief propagation, relation to spectrum of "non-backtracking" matrix.
  - BP on the board.
- Sharp threshold: If $\text{SNR} = (a-b)^2/(2(a+b)) > 1$, then easy to solve $O(n \log n)$ algorithms. Otherwise, impossible.
- For $k \geq 4$, gap between what's impossible and what's easy to solve.

References

(1) W. W. Zachary. An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452-473 (1977).

(2) Fortunato, Santo. Community detection in graphs. Physics Reports 486.3 (2010): 75-174.

(3) Spielman, Daniel A., and Nikhil Srivastava. Graph sparsification by effective resistances. SIAM Journal on Computing 40.6 (2011): 1913-1926.

(4) Abbe, Emmanuel and Sandon, Colin. Detection in the stochastic block model arXiv:1512.09080

(5) Mossel, Elchanan, Joe Neeman, and Allan Sly. Reconstruction and estimation in the planted partition model. Probability Theory and Related Fields (2014): 1-31.

(6) Mossel, Elchanan, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. arXiv:1311.4115 (2013).