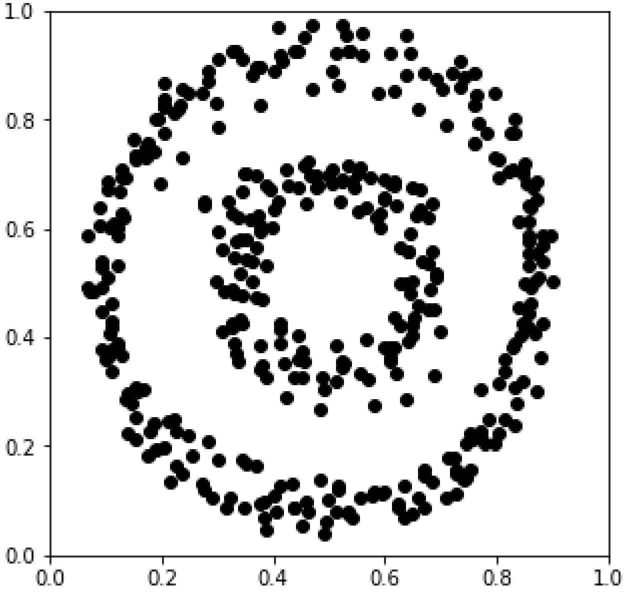
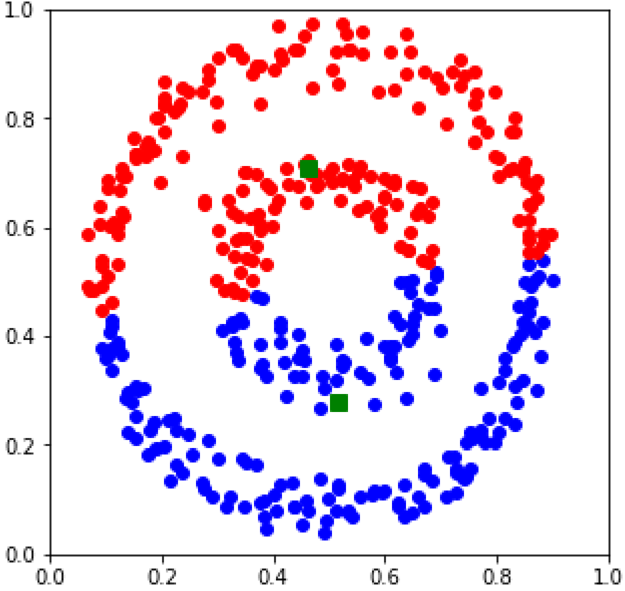
Community Detection

Rajesh Sundaresan
Indian Institute of Science
Data Analytics

Can machines see what we see? We see two clusters

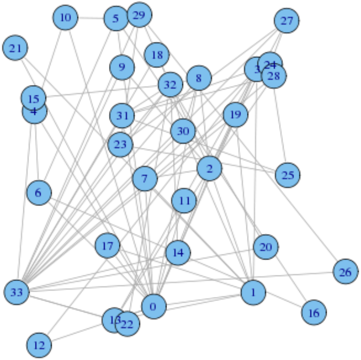


The standard k-means algorithm fails



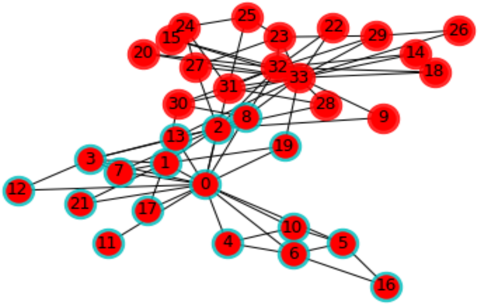
Zachary's karate club

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



Can machines see beyond what we can?

- At some point, a fissure developed, and the group split into two. Can you predict the factions?



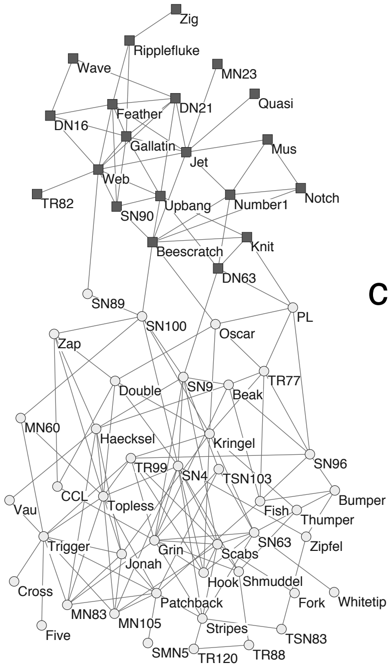
- Two clusters. One around '0' who was the Instructor. One around '32' and '33', the latter was president of the club.

Bottlenose dolphins at Doubtful Sound, New Zealand

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.
- One of the dolphins was away for some time, and the group split into two.

Two groups



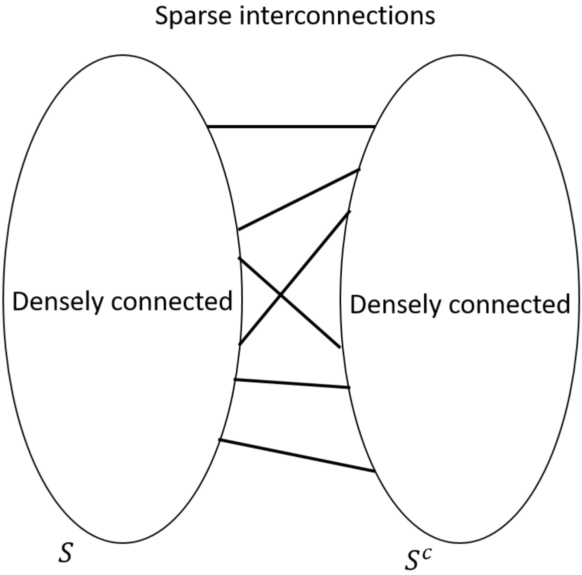
Zig
Ripplefluke
Wave
MN23
Quasi
DN21
Feather
DN16
Gallatin
Mus
Jet
Web
Notch
TR82
Number1
Upbang
SN90
Knit
Beescratch
DN63
C
SN89
PL
SN100
Oscar
Zap
TR77
SN9
Double
Beak
MN60
Haecksel
Kringel
SN96
TR99
SN4
TSN103
CCL
Bumper
Vau
Topless
Fish
Thumper
SN63
Zipfel
Trigger
Grin
Scabs
Jonah
Hook
Shmuddel
Cross
Fork
Whitetip
Patchback
MN83
MN105
Stripes
TSN83
Five
SMN5
TR120
TR88

Other examples

- Collaborations of scientists
- Protein-protein interaction network and its change in cancerous rats
- Word networks and categorisation, experiment with the word ‘bright’

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.



Sparse interconnections
Densely connected
Densely connected
S
S^c

Why study?

- Fast isolation of communities in case of epidemics
- Targeted advertisements, better recommendations
- Detection of vulnerabilities in the network

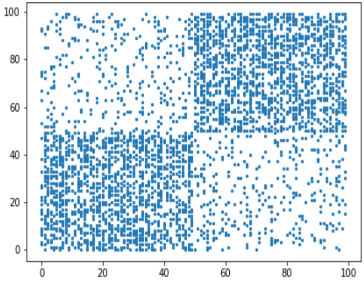
- Main difference from previous settings
  - Unsupervised, no training samples

The generative perspective

- Suppose you were to generate an instance of a graph with a few communities, and challenge your colleague's algorithm, how would you go about it?

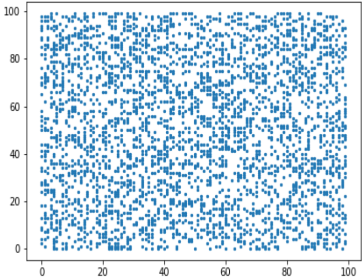
The Stochastic Block Model SBM$(p, q, (n/2, n/2))$

- For a graph $G = (V, E)$, A = adjacency graph defined by $A_{i,j} = 1$ if i and j are connected. Symmetric.
- Generate A with two communities.
  - Links within community w.p. $p = 1/2$
  - Links across community w.p. $q = 1/8$, note $q < p$. (Noise)



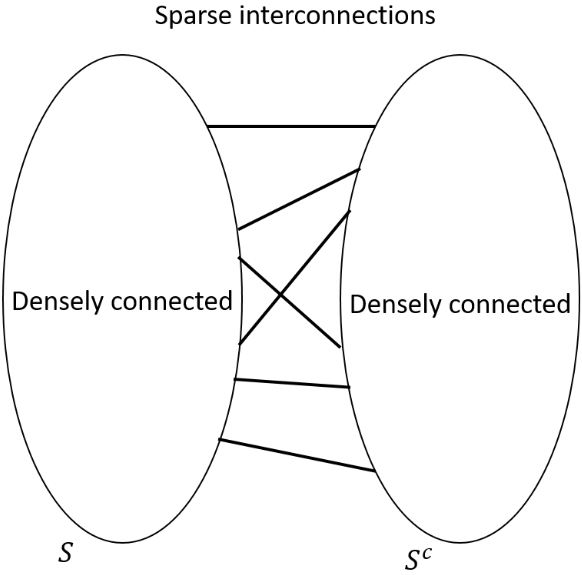
The Stochastic Block Model SBM$(p, q, (n/2, n/2))$

- Permute, erase community labels, and send graph to your colleague.



Two equal communities

- SBM$(p, q, (50, 50))$, two equal-sized communities.
- This is a $+1, -1$ classification problem for each node
- A candidate labelling is $v = (-1, -1, \ldots, -1, +1, +1, \ldots, +1)^T$
- For any such balanced labelling, we know $\langle \mathbf{1}, v \rangle = 0$
  where $\mathbf{1}$ is the vector of all 1s.
- Since you generated using a statistical model,
  your colleague could use the maximum likelihood principle.


Sparse interconnections
Densely connected
Densely connected
S
S^c

Maximum likelihood principle

- Given a graph generated by the stochastic block model $\mathrm{SBM}(p, q, (50, 50))$:

- If S and S^c are the two communities, we can write

$$v = \mathbf{1}_S - \mathbf{1}_{S^c}.$$

- Balanced: $\langle \mathbf{1}, v \rangle = 0$.

- Assign labels $+1$ to 50% of the nodes and -1 to 50% of the nodes to maximise likelihood of the observed graph:

$$\Pr\left\{ G \;\middle|\; v = \mathbf{1}_S - \mathbf{1}_{S^c} \text{ with } \langle \mathbf{1}, v \rangle = 0 \right\}$$

The outcome

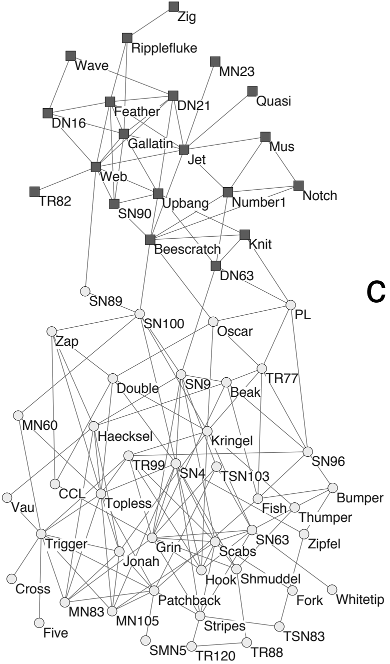
Theorem

The maximum likelihood assignment v solves

$$\begin{aligned} &\max_{v\in\{-1,1\}^n:\langle\mathbf{1},v\rangle=0} v^T A v \\ \equiv &\min_{v\in\{-1,1\}^n:\langle\mathbf{1},v\rangle=0} v^T L v \\ \equiv &\min_{(S,S^c),\ \textit{balanced}} cut(S, S^c) \end{aligned}$$

- $L = D - A$, Laplacian
  $D = diag(d_1, \ldots, d_n)$
  d_i = degree of node i.
- $cut(S, S^c)$ = number of cross-linkages.
- Works for any $0 < q < p < 1$!

Min-cut



Zig
Ripplefluke
Wave
MN23
Quasi
DN21
Feather
DN16
Gallatin
Mus
Jet
Web
Notch
TR82
Upbang
Number1
SN90
Knit
Beescratch
DN63
C
SN89
PL
SN100
Oscar
Zap
TR77
SN9
Double
Beak
MN60
Haecksel
Kringel
SN96
TR99
SN4
TSN103
CCL
Vau
Topless
Bumper
Fish
Thumper
Trigger
SN63
Zipfel
Grin
Scabs
Jonah
Hook
Shmuddel
Cross
Fork
Whitetip
Patchback
MN83
MN105
Stripes
TSN83
Five
SMN5
TR120
TR88

Tough nut to crack, and a settlement for less

- Computer scientists know that this is a hard optimisation problem to solve.
- Relax $v \in \{-1, 1\}^n$ to $v \in \mathbb{R}^n$.
  Since only sign matters, normalise v to have unit norm.

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

- Look for vectors v that minimise $v^T L v$ among all unit vectors v orthogonal to $\mathbf{1}$.

Properties of L

- Facts:
  - L is symmetric with all eigenvalues real and nonnegative.

$$Lu_i = \lambda_i u_i$$

  - $\{u_1, u_2, \ldots, u_n\}$ are orthogonal and span $\mathbb{R}^n$.
  - Order the eigenvalues as $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.
    The lowest eigenvalue is $\lambda_1 = 0$, with $u_1 = \mathbf{1}$.
- Write $v = \sum_{i=1}^{n} a_i u_i$, where u_i are the eigenvectors of L. So

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

- What is the smallest value of $v^T L v$ when $\langle v, \mathbf{1} \rangle = 0$?
  The corresponding eigenvector?

Solution: Fiedler vector

- The minimising value is λ_2.
- The corresponding vector is u_2 and is called Fiedler vector
- Use u_2 as a surrogate for $\frac{1}{\sqrt{n}}(\mathbf{1}_S - \mathbf{1}_{S^c})$.
- Order and pick the top half.
- If two communities of different sizes, use sign of u_2, or cluster its entries into two groups, or pick the top k (if number is known).

Normalised Laplacian

- One could also consider the normalised Laplacian:

$$L_{norm} = I - D^{-1/2} A D^{-1/2}.$$

- 0 is an eigenvalue of both L and L_{norm}. The corresponding eigenvectors are $\mathbf{1}$ and $D^{1/2}\mathbf{1}$, respectively.

- 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 be its 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, see second part of theorem.
- If A has cross-linkages, but relatively small in number, the eigenvalues get perturbed, but perhaps not by much.
- Eigenvectors also get perturbed, but perhaps not by much.
- Exploit these regularities.

A more general spectral algorithm

Input: Adjacency matrix A and number of components k.

- Compute the Laplacian or the normalised Laplacian L_{norm}.
- Find the k smallest eigenvalues and eigenvectors.

$$X = [u_1 \;\; u_2 \;\; \dots \;\; u_k].$$

- 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.)
- Output : Clusters of the 'data clustering' algorithm.

Data clustering

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

One example: k-means clustering

- Find a partition $S_1, S_2, \ldots, S_k$ of the points so that the following is minimised:
$$\sum_{i=1}^{k} \sum_{l \in S_i} d(x_l, \overline{c}_i)^2.$$
where $\overline{c}_i$ is the best representative (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

Draw picture on board

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.

The two circles problem

- So, how does it solve the two circles problem?
- Generate a complete graph with weights:

$$A(i,j) = \exp\left\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}\right\}$$

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).

One approach : GREEDY

- Want 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.

Relaxation, lift, and BP approaches

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

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

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

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

- Belief Propagation: On the board

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
- 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.

(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).