Network clustering: 50 years and still going!
Information-theoretic criteria and efficient algorithms for a problem that thrives

Maximilien Dreveton[1] **Daniel R. Figueiredo**[2]

[1] Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
[2] Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

Tutorial at ACM SIGMETRICS/IFIP Performance Conference, June 10, 2024

EPFL UFRJ

Tutorial Organization

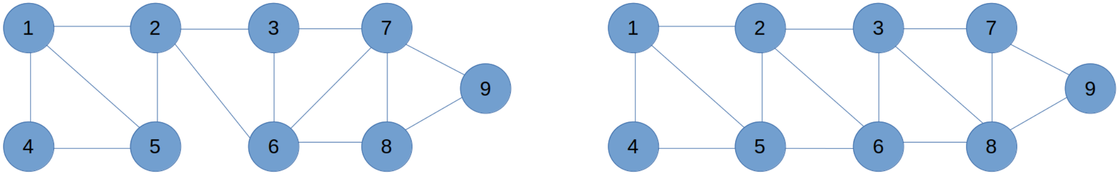
Part I: Introduction to network clustering *(Daniel)*

- what is network clustering and why it is important

Part II: How to detect network clusters? *(Daniel)*

- different algorithmic approaches to identify network clusters

Part III: When can network clusters be identified? *(Maximilien)*



(a) The easy case: clusters are clear

(b) The hard case: where are the clusters?

Part IV: What was not covered in this tutorial? *(Maximilien)*

- relevant topics not covered (time is the constraint)

Table of contents

What is Network Clustering

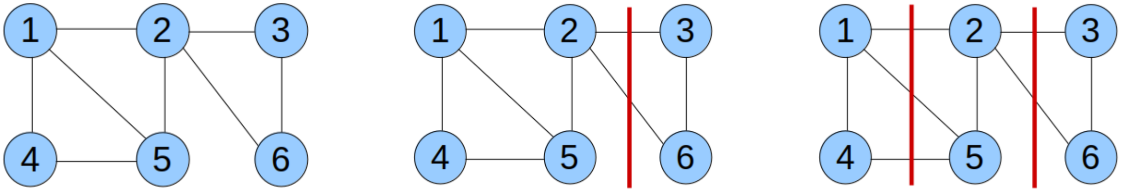
Partition the set of nodes of a network

- every node must be exclusively in one part (same definition of set partition)

Different names for the same thing (coming from different fields)

- graph partitioning, graph clustering, network clustering, community detection
- a subgraph, a cluster, a community

Examples



What makes for a good partition ?

- must quantify the quality of a partition
- to partition is easy, finding a good (optimal) partition can be hard

BUT WHY CLUSTER A NETWORK ?

Fun and hard combinatorial problem

- find the partition that minimizes the cut size (easy, polynomial time)
- find the balanced (equal sized) partition that minimize the cut size (difficult, NP-Hard)
- many problem variations and theoretical results

Cluster can reveal latent information about nodes

- real network structure is not random
- clusters reveal something about the nodes

Many applications

- networks are everywhere!
- network clustering is a fundamental tool in Data Science toolbox

Social Networks

Nodes represent individuals

- edges encode some pairwise relationship among individuals
- eg., friendship on FB, contact in Whatsapp, physical proximity, co-authorship, etc
- growing number of real datasets concerning all sorts of relationships

Social Network Analysis

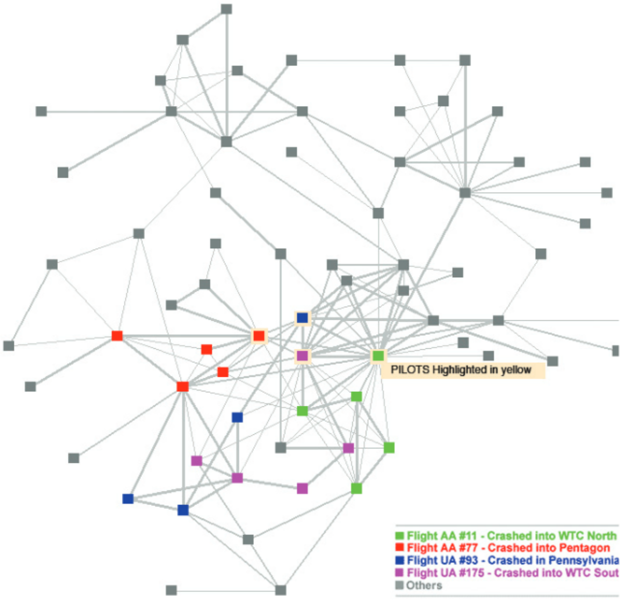
- focus on analyzing social networks to reveal information about individuals and the network
- reveal social structure, identify social behavior and influential individuals, quantify importance of relationships, etc
- much older than you think: first volume of "Social Networks" by Elsevier published in 1978!

Clustering in social networks

- clusters can reveal social structure, including influential groups
- Criminal networks: clusters can reveal individuals working for the same criminal organization

Criminal Network

Interactions between terrorists involved in September 11 attack (Xu & Chen, 2005)


PILOTS Highlighted in yellow
🟩 Flight AA #11 - Crashed into WTC North
🟥 Flight AA #77 - Crashed into Pentagon
🟦 Flight UA #93 - Crashed in Pennsylvania
🟪 Flight UA #175 - Crashed into WTC Sout
⬛ Others

Biological Networks

Nodes represent some biological entity

- eg., species, protein, gene, neuron
- edges encode some pairwise interaction among nodes
- eg., predator-prey, protein interaction, gene expression, neuron synapses
- growing number of real datasets concerning all sorts of relationships

Clustering in biological networks

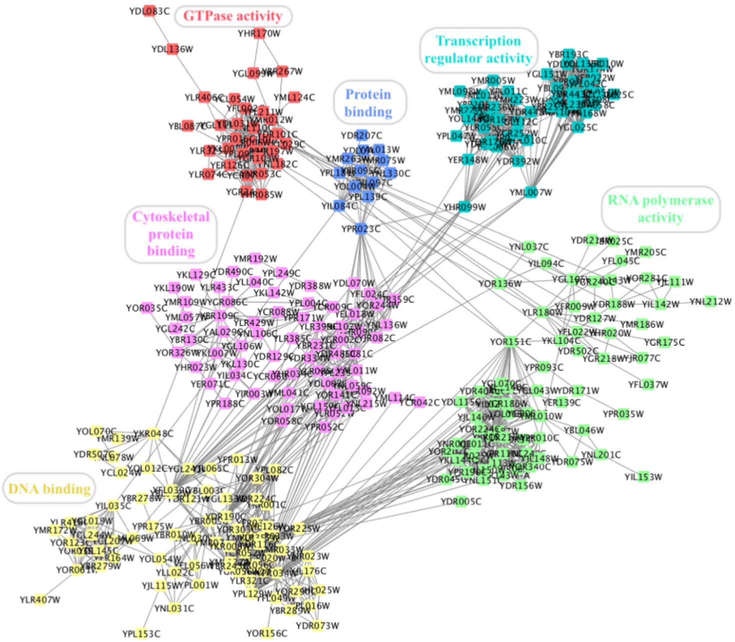
- clusters of species reveal their role and importance in an ecosystem
- clusters of gene or proteins reveal their functional role in the biological system
- clustering of PPI networks since 2000's (in Bioinformatics)

Examples

- clusters in protein-protein interaction (PPI) network are used to design more effective drugs
- clusters in species network used to refine the map of evolution of species across time

Protein-Protein Interaction Network

Six clusters identified in the PPI of the *Saccharomyces Cerevisiae* (Manipur et al., 2021)



Other Kinds of Networks

Information networks

- nodes represent some kind of information: words, documents, research papers, websites, etc
- edges encode some pairwise relationship: similar meaning, related topic, cited by, hyperlink to, etc
- knowledge graph: synthesis of kinds of information and relationships in a single network

Infrastructure networks

- nodes represent some artifact of an infrastructure: train stations, airports, power plants, datacenters, etc
- edges encode some kind of connectivity between these parts: train line, flight, transmission line, optic cable, etc

Clustering in these networks

- clusters can reveal properties of the network and are used to tackle different problems
- eg., clustering in knowledge graph reveals related topics to a user search
- eg., clustering in user-item graph used in recommendation systems

Bottom line: way too many networks and applications!

Good partition and how to find them

Intuitive definition of a good partition

- most edges are within each cluster, few edges are between different clusters
- clusters have relatively similar sizes

Formal definition (to be made precise later)

- a cost function for a given partition $\mathcal{P}$ of a graph, $c(\mathcal{P})$
- includes local or global information about $\mathcal{P}$ and the graph

Template to all network clustering algorithms

1. choose a cost function $c(\mathcal{P})$
2. run an algorithm to solve the combinatorial optimization problem

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} c(\mathcal{P})$$

3. algorithm is often a heuristic or approximation due to runtime complexity: does not necessarily returns $\mathcal{P}^*$
4. number of possible partitions is 2^n, where n is the number of nodes

Some fundamental issues

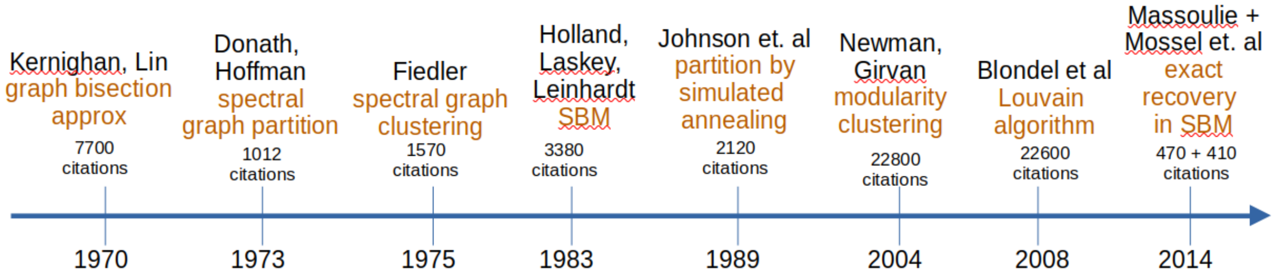
No silver bullet!

- no single best cost function $c(\mathcal{P})$, no single best algorithm
- different cost functions and different algorithms (even for the same cost function) can find different clusters
- what clusters should reveal (latent information) is application dependent (eg., influential group or similar functional behavior)
- tailored cost functions and algorithms for a domain can be more effective

The good news: network clustering algorithms can be clustered!

- most cost functions have a common feature
- most algorithms leverage a common approach
- relatively few approaches in the literature: cut-based, modularity-based, bayesian inference, label propagation, learning-based
- you will learn these approaches here

NETWORK CLUSTERING TIMELINE



Some influential papers proposing network clustering algorithms

- not representative of the thousands of algorithms and papers
- not drawn to scale

Network clustering is over 50 years old!

- interest exploded in the last 15 years
- more and larger real network available, more applications

Some Notation Before we Start

An arbitrary graph $G = (V, E)$

- V and E are the set of nodes and edges
- assumed to be undirected and unweighted, unless otherwise stated
- $n = |V|$ and $m = |E|$ are the number of nodes and edges of the graph
- A is the adjacency matrix of G, $A_{i,j} = 1 \Leftrightarrow (i,j) \in E$

An arbitrary partition $\mathcal{P} = \{C_1, \dots, C_k\}$ of V

- $C_\ell \cap C_{\ell'} = \emptyset$ for all $\ell \neq \ell'$, and $\bigcup_\ell C_\ell = V$ (definition of partition)
- k is the number of clusters (communities)
- $z \in [k]^n$ is the cluster assignment vector: $z_i = \ell$ means node $i \in V$ belongs to cluster $\ell \in \{1, \dots, k\}$

THE QUALITY OF A PARTITION, $c(\mathcal{P})$

Cut size

- number of edges with an endpoint in C_1 and another in C_2
- $c(C_1, C_2) = \sum_{(u,v)\in E} \mathbb{1}(z_u \neq z_v)$
- the most elementary cost function to assess the quality of a partition

Cut ratio: the normalized cut size

- cut size does not consider number of nodes in clusters. Often leads to very unbalanced clusters (almost all nodes in a single cluster)
- normalize the cut size by the number of possible edges in the cut
- cut ratio $c_r(C_1, C_2) = \frac{c(C_1,C_2)}{|C_1||C_2|}$

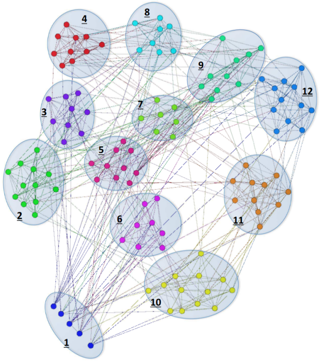
Cut size for fixed cluster sizes

- determine clusters with a given size, eg. n/k for k equally sized clusters
- only consider partitions within this constraint
- $c_b(C_1, C_2) = \sum_{(u,v)\in E} \mathbb{1}(z_u \neq z_v)$
- known as the balanced partition or planted partition problem

THE QUALITY OF A PARTITION: ANOTHER APPROACH

Assume knowledge of the latent information of given network

- latent information induces a network partition $C_1, \ldots, C_k$ where z is the cluster assignment vector
- z is assumed to be the desired partition (ground truth)



- American college football teams that played each other in 2000 and their 11 conferences in different clusters (Avrachenkov et al., 2014; Girvan & Newman, 2002)

THE QUALITY OF A PARTITION: ANOTHER APPROACH

Use the ground truth $C_1, \ldots, C_k$ with z

- suppose a network clustering algorithm returns a partition $\hat{C}_1, \ldots, \hat{C}_k$ with $\hat{z}$
- measure the quality of this partition with respect to the ground truth $C_1, \ldots, C_k$ with z

Accuracy for the case $k = 2$

- fraction of nodes whose cluster agree with the ground truth
- given by $1/n \sum_{u \in V} \mathbb{1}(\hat{z}_u = z_u)$

Problem: algorithms cannot identify the label of the cluster

- accuracy would be zero if $C_1 = \hat{C}_2$ and $C_2 = \hat{C}_1$?
- consider the largest accuracy across all permutations of cluster labels
- number of permutations grows as $k!$, and each permutation requires $O(n)$ time to compute the accuracy (not very practical)

The quality of a partition using ground truth

How to assess the quality of a partition with respect to the ground truth ?

- problem in data clustering (and not only network clustering)
- accuracy is just one approach but not the most adequate

Various metrics proposed in the literature

- Normalized Mutual Information (NMI) between the partition and the ground truth
- Rand-index: counts pairs of elements in the same or different clusters
- Adjusted Rand-index (ARI): adjust to remove influence of randomly clustering the elements
- Confusion Matrix: fraction of elements correct for each pair of clusters
- no clear preference as comparison between alternative partitions depends on the metric

Ground truth can be used to compare clustering algorithms

- how to obtain networks with ground truth information about its clusters?
- network models: models that randomly generate networks according to a ground truth

TABLE OF CONTENTS

NETWORK CLUSTERING ALGORITHMS

Over a thousand algorithms proposed in the last 50 years!

- different fields, different applications, different problem variations (input)
- a few fundamental ideas reappear in many algorithms

Algorithms can be roughly divided into categories

- cut-based algorithms: minimize cut size or cut ratio
- modularity-based algorithms: maximize modularity of the partition
- inference-based algorithms: maximize the likelihood function over the partitions under some statistical formulation
- learning-based algorithms: train a classification model to predict the cluster of nodes

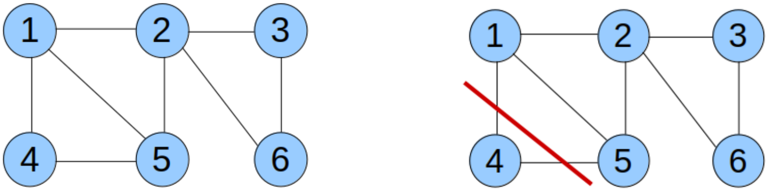
Main idea of each category will be presented

- along with an example of an algorithm in that category

The Very Beginning: Minimum Cut Problem

Minimum Cut Problem

- Given G find partition C_1 and C_2 such that $c(C_1, C_2)$ is minimum



Related to the Maximum Flow Problem

- Max-flow and Min-cut are equivalent for a given source/destination pair of nodes
- efficient algorithms and linear programming formulation in the 1950s (Ford & Fulkerson, 1957)

Min-cut is polynomial (near-linear time)

- randomized algorithm (1996) finds the minimum cut in time $O(n^2 \log n)$, (Karger, 2000)
- first deterministic near-linear time algorithm (best paper award in SODA'24), (Henzinger et al., 2024)

Graph Bisection Problem (aka. Planted Partition Problem)

Min-Cut Problem with identical cluster sizes and $k = 2$

- finding the optimal partition is NP-Hard
- several heuristics and approximation algorithms
- a corner stone is (Kernighan & Lin, 1970)

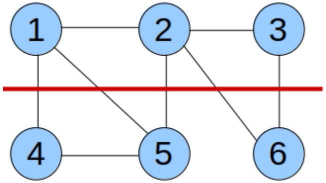
Kernighan-Lin algorithm (Kernighan & Lin, 1970)

1. start with a random bisection of the graph C_1, C_2 (eg., all even nodes in one cluster, odd nodes in the other
2. consider all pairs of nodes $u \in C_1$ and $v \in C_2$ and the reduction in the cut size when they are swapped clusters
3. select the pair that reduces the cut the most (greedy), and swap their clusters
4. return to step 2 or stop when the cut size cannot be reduced

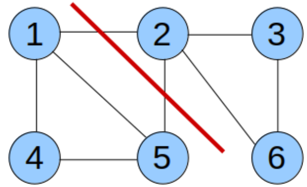
Some considerations

- each greedy iteration considers $O(n^2)$ pairs
- computing the change in the cut size when swapping a pair is proportional to the nodes' degrees
- number of iterations depends on initial bisection (and other factors)
- does not always return the optimal solution

Example of the Kernighan-Lin algorithm



- $C_1 = \{1, 2, 3\}$, $C_2 = \{4, 5, 6\}$, $c(C_1, C_2) = 5$
- node pair $(1, 6)$ reduces the cut the most, swap them



- $C_1 = \{1, 4, 5\}$, $C_2 = \{2, 3, 6\}$, $c(C_1, C_2) = 2$
- stop: no other pair reduces the cut

Spectral decomposition of a graph

The graph Laplacian matrix L

- $L = D - A$, where A is the adjacency matrix and D a diagonal matrix where D_{ii} is the degree of node i

Spectral decomposition of L

- let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of L, and $v_1, \ldots, v_n$ the corresponding basis of orthogonal eigenvectors, normalized so that $||v_i||_2^2 = n$

Eigenvalues and eigenvectors of L reflect the structure of G

- spectral graph theory: understand relationship between eigenvalues and eigenvectors and graph properties
- example: if G has k connected components, then L has k eigenvalues equal to zero and each corresponding eigenvector reveals (with non-zero entries) a connected component
- use spectral decomposition of L for network clustering!

Spectral network clustering

Connecting L with the cut size induced by z

- consider $k = 2$ and let $z_i \in \{-1, 1\}$ denote the assignment vector with -1 and 1 representing the two clusters
- given an assignment vector z the cut size induced by z is given by $z^T Lz$
- optimal solution to the min-cut problem is given by

$$\hat{z} = \arg\min_{z \in \{-1,1\}^n} z^T Lz \quad ,$$

Relaxation of $\hat{z}$ and connection to eigenvector

- the combinatorial problem above can be relaxed by having $z_i \in \mathbb{R}$
- exact optimal solution is given by

$$v_2 = \arg\min_{z \in \mathbb{R}^n} z^T Lz \quad ,$$

  where v_2 is the eigenvector associated with the second smallest eigenvalue of L

Spectral network clustering algorithm, $k = 2$

Simple algorithm

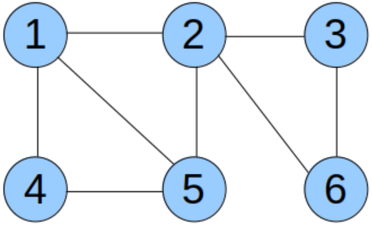
- determine L for a given graph G
- compute v_2 from L
- use the sign of $v_2(i)$ to determine the cluster of node i
- if $v_2(i) < 0$, then $\hat{z}_i = 1$ else $\hat{z}_i = 2$

SPECTRAL NETWORK CLUSTERING ALGORITHM, $k = 2$

Simple algorithm

- determine L for a given graph G
- compute v_2 from L
- use the sign of $v_2(i)$ to determine the cluster of node i
- if $v_2(i) < 0$, then $\hat{z}_i = 1$ else $\hat{z}_i = 2$

Example



- $\lambda_2 = 3 - \sqrt{5}$
- $v_2 = (-0.62, 0.23, 1.0, -1.0, -0.62, 1.0)$
- $C_1 = \{1, 4, 5\}$, $C_2 = \{2, 3, 6\}$
- a good job!

Spectral network clustering for cut ratio

Connecting L with the cut ratio induced by $C_1, \dots, C_k$

- consider a partition $C_1, \dots, C_k$
- let H be a $n \times k$ matrix where $H_{i\ell} = 1/\sqrt{|C_\ell|}$, if $i \in C_\ell$ and 0 otherwise
- cut ratio of $C_1, \dots, C_k$ is given by $\text{Tr}(H^T LH)$ where $\text{Tr}(\cdot)$ is the trace matrix operation

Minimum Cut Ratio Problem

- is given by
$$H^* = \arg\min_{C_1,\dots,C_k,H} \text{Tr}(H^T LH)$$
- this combinatorial problem is NP-Hard
- relaxation: allow H to have non-zero entries and impose the constraint $H^T H = I_k$ where I_k is the identity matrix with dimension k
- optimal solution H^* is the matrix whose columns are the first k orthogonal eigenvectors of L

$$H^* = \left[\begin{pmatrix} \\ v_1 \\ \\ \end{pmatrix} \quad \dots \quad \begin{pmatrix} \\ v_k \\ \\ \end{pmatrix} \right]$$

SPECTRAL NETWORK CLUSTERING FOR CUT RATIO

Recovering the partition

- matrix H^* does not directly provide the node clusters

$$H^* = \begin{bmatrix} \begin{pmatrix} v_1 \end{pmatrix} & \dots & \begin{pmatrix} v_k \end{pmatrix} \end{bmatrix}$$

- i-th row of matrix H^* is the "signature" vector of node i in a k dimensional space

$$s_i = (v_1(i), v_2(i), \dots, v_k(i))$$

- **idea:** nodes that have similar "signatures" should be in the same cluster
- use s_i to cluster nodes in this k-dimension space into k clusters (eg., use k-means algorithm)

Practical considerations for better results

- use normalized Laplacian (more stable when degrees have different scales)
- use more than k eigenvectors for generating signatures (for finding k clusters)

Network Cluster Modularity

- cut-based metrics only consider inter-cluster edges
- however, intra-cluster edges can also be important

Modularity

- **idea:** compare number of edges within a cluster to the expected number of edges in a random model
- larger modularity indicates intra-cluster edges are more present than in random model
- number of intra-cluster edges in C_ℓ is given by $X_\ell = \sum_{i,j \in C_\ell} A_{i,j}$
- expected number intra-cluster edges in C_ℓ is given by $Y_\ell = \sum_{i,j \in C_\ell} p_{i,j}$, where $p_{i,j}$ is the probability of observing the edge (i, j)

$$M(C_1, \dots, C_k) = \frac{1}{m} \sum_\ell (X_\ell - Y_\ell) = \frac{1}{2m} \sum_\ell \sum_{i,j \in C_\ell} A_{i,j} - p_{i,j} \quad ,$$

where $1/m$ is a normalization constant ($m = |E|$)

Network Cluster Modularity

Random network model

- to compute modularity, $p_{i,j}$ must be determined for any real network
- **idea:** use Configuration Model (CM) to determine $p_{i,j}$, in this case

$$p_{i,j} = \frac{d_i d_j}{2m}$$

  where d_i is the degree of node i
- substituting, yields

$$M(C_1, \ldots, C_k) = \frac{1}{2m} \sum_{\ell} \sum_{i,j \in C_\ell} A_{i,j} - \frac{d_i d_j}{2m} \quad ,$$

Properties

- $-1/2 \leq M(C_1, \ldots, C_k) \leq 1$ for any G and $C_1, \ldots, C_k$, modularity is bounded
- if $G \sim CM(d_1, \cdots, d_n)$ the $\mathbb{E}_G[M] = 0$ for any k, expected modularity is zero for the null model

Computing the Maximum Modularity

- determining the partition that maximizes modularity for a given network is NP-Hard

Greedy algorithm by Newman and Girvan, 2004

1. start with every node in its own cluster
2. for every pair of clusters a, b with at least one edge between them, compute the change in modularity when the two clusters are merged, $\Delta M_{a,b}$ (which can be negative)
3. merge the two clusters with the largest $\Delta M_{a,b}$
4. stop when number of clusters is 1
5. return the partition for which M was maximum

Properties

- algorithm induces a hierarchy by merging clusters bottom-up
- does not require an *a priori* value for k
- running time complexity is $O(nm)$, too slow for very large networks

Louvain Algorithm for Maximizing Modularity

Another greedy but faster approach by Blondel et al., 2008

- Louvain is because co-authors of the paper were from University of Louvain, Belgium

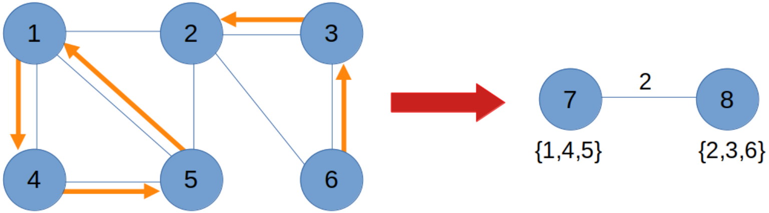
1. start with every node in its own cluster
2. for every cluster *a*, compute the change in modularity when merging with each neighboring cluster *b*, $\Delta M_{a,b}$ (which can be negative)
3. for every cluster *a*, merge *a* with neighboring cluster *b* with the largest $\Delta M_{a,b}$ (if positive)
4. using the new clusters, go to 1 until there are no more changes to cluster assignment
5. return the last partition

Key difference with prior approach

- merges multiple clusters in a single iteration
- reduces the number of clusters at least by half per iteration

EXAMPLE OF LOUVAIN ALGORITHM

One single iteration



- each node (cluster) identifies the best neighbor to merge with (if positive gains)
- node (cluster) 2 had no positive gains (no outgoing arrow)
- new graph is formed with edge weights corresponding to number of edges between clusters

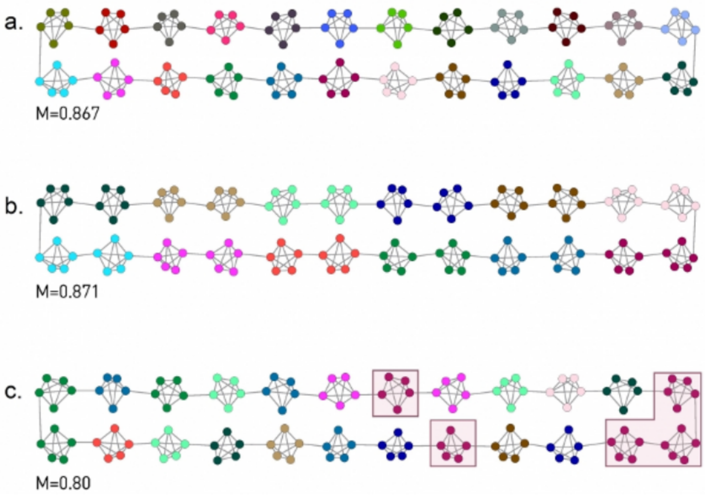
Properties

- algorithm induces a hierarchy by merging clusters bottom-up
- does not require an *a priori* value for k
- order in which clusters are merged matters
- running time complexity is $O(m \log n)$

Resolution Limit of Modularity Maximization

Modularity is biased towards large clusters

- small clusters will reduce modularity (Fortunato & Barthelemy, 2007)
- condition for clusters in the maximum modularity partition: $d_a \geq \sqrt{2m}$, where d_a is sum of the degrees of nodes in cluster a

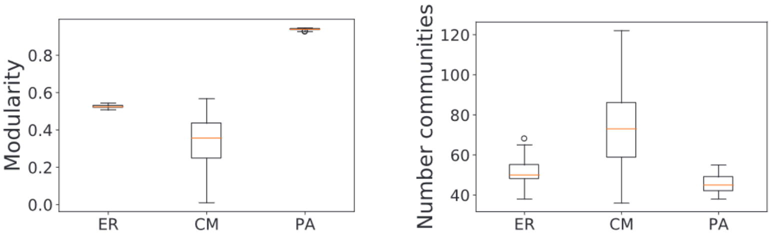


- (a) intuitive partition (every clique in a cluster), (b) optimal modularity (two cliques per cluster), (c) random assignment of cliques to clusters (Barabási, 2016)

Problem with Modularity Maximization

Algorithms find good partitions when no clear network clusters exist

- run Louvain algorithm on random graphs with no clusters



- ER = Erdos-Reyni model, PA = Preferential Attachment model, CM = Configuration Model with Zipf distribution (Avrachenkov & Dreveton, 2022)
- even for the null model (CM), Louvain finds relatively large modularity values

General problem with cut-based approach (not just modularity)

- can find clusters in pure randomness
- clusters do not reveal any latent information
- must interpret results with caution!

PRINCIPLED APPROACH: RANDOM MODELS FOR NETWORK CLUSTERS

Network clusters generated by some random process

- assume a model for how nodes are placed into clusters, $\mathbb{P}\left(z\right)$
- assume a model for how the network is generated given the assignment vector z, $\mathbb{P}\left(A \,|\, z\right)$
- A is a random network (adjacency matrix) conditioned on cluster assignment for the nodes

Use statistical inference to determine "best" node assignment

- given A and models $\mathbb{P}\left(A \,|\, z\right)$ and $\mathbb{P}\left(z\right)$, the posterior $\mathbb{P}\left(z \,|\, A\right)$ is given by applying Bayes rule,

$$\mathbb{P}\left(z \,|\, A\right) = \frac{\mathbb{P}\left(A \,|\, z\right)\mathbb{P}\left(z\right)}{\mathbb{P}\left(A\right)}$$

- $\mathbb{P}\left(z \,|\, A\right)$ is the likelihood function of the assignment z

Maximum Likelihood Estimation (MLE) of node assignment

- find z^* that maximizes the likelihood function

$$z^* = \arg\max_{z} \mathbb{P}\left(A \,|\, z\right)\mathbb{P}\left(z\right)$$

- maximum assignment does not depend on $\mathbb{P}\left(A\right)$

Challenges of inference-based approaches

Some main concerns

- determining z^* is often an NP-Hard problem
- requires knowledge of $\mathbb{P}\left(A \,|\, z\right)$ and $\mathbb{P}\left(z\right)$. What is the right model for your network data?
- must leverage heuristics to search for z^*
- heuristics and algorithms often depend on $\mathbb{P}\left(A \,|\, z\right)$ and $\mathbb{P}\left(z\right)$

Some common simplifying assumptions

- number of clusters k is fixed, a priori
- $\mathbb{P}\left(z\right)$ is assumed to be uniform (proportional to number of nodes in each cluster)
- $\mathbb{P}\left(A \,|\, z\right)$ is assumed to be conditionally independent, $\mathbb{P}\left(A_{ij}, A_{i'j'} \,|\, z\right) = \mathbb{P}\left(A_{ij} \,|\, z\right) \mathbb{P}\left(A_{i'j'} \,|\, z\right)$ for all edges

Models for $\mathbb{P}\left(A \,|\, z\right)$

- most common is Stochastic Block Model (SBM), to be discussed
- models for edge presence and weights that depend on clusters: density function $f_{\ell,\ell'}(X)$ for cluster pair (ℓ, ℓ')
- parametric density or probability functions: bernoulli, normal, exponential, etc

HARD CLUSTERING ALGORITHM

Greedy maximization the likelihood function

- let $L_{i,\ell}(z_{-i}, A)$ denote the likelihood function value obtained by placing node i in cluster ℓ
- $L_{i,\ell}(z_{-i}, A)$ can be computed using the parameters of the parametric distribution for edges

1. assume some initial assignment $z(0)$, and number of clusters k, set $t = 1$
2. using assignment $z(t-1)$ to compute empirical $\mathbb{P}(z)$ and parameters for parametric distributions $f_{\ell,\ell'}$ for all cluster pairs (ℓ, ℓ')
3. for each node i, determine $z_i(t) = \arg\max_\ell L_{i,\ell}(z_{-i}, A)$
4. set $t = t + 1$
5. go to 2 until convergence

Observations

- $\arg\max_\ell L_{i,\ell}(z_{-i}, A)$ can be computed by simply considering all clusters (brute force)
- convergence to maximum number of iterations or reaching a fixed point in assignment vector
- performance strongly depends on initial assignment $z(0)$
- how to determine $z(0)$? A random choice is likely to lead to poor performance

Markov Chain Monte Carlo Algorithm

Generate samples from the posterior distribution, $\mathbb{P}\left(z \mid A\right)$

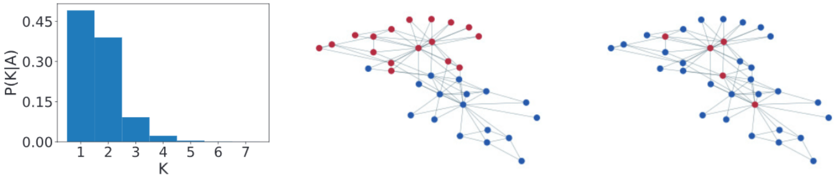
- consider Markov Chain (MC) with state space given by set of all possible assignments (all possible vectors z)
- determine some simple transition rule between states in the MC
- eg., choose a node i at random from current state, place i in cluster ℓ with probability proportional to number of neighbors of i that are in ℓ (Peixoto, 2014)
- use Metropolis-Hastings to induce the above MC to follow $\mathbb{P}\left(z \mid A\right)$ in steady state. Calculations require models for $\mathbb{P}\left(A \mid z\right)$ and $\mathbb{P}\left(z\right)$
- simulate the MC for many, many steps

Samples from $\mathbb{P}\left(z \mid A\right)$ used to compute cluster statistics

- use samples to determine the marginal distribution for the clusters for every node
- ie., $\hat{z}_i$ is now a probability vector: $\hat{z}_i(\ell)$ is the probability that node i belongs to cluster ℓ
- use maximum value to determine cluster for node i, value indicates the confidence of assignment

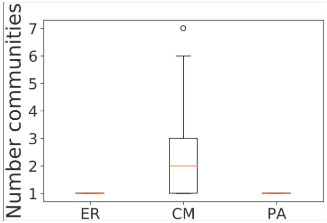
MCMC Network Clustering Example

Results for karate network dataset (Avrachenkov & Dreveton, 2022)



- predicts one or two clusters with higher probability. Examples of assignments with two clusters

Results for random graphs (same models as before)



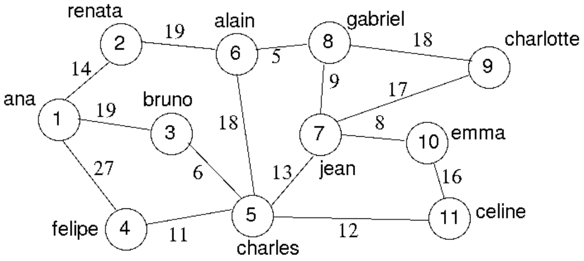
- predicts single cluster for ER and PA!

Clustering networks with node and edge attributes

Real network data is increasingly annotated

- node have attributes, edges have attributes
- attributes are often correlated with latent information

Social network with some students first name and number of messages exchanged



- latent information is the nationality: brazilian and french?
- homophily, name pattern, and communication pattern can be used to infer clusters

Most recent variation of network clustering!

- how to fuse network information with attribute information ?

PRINCIPLED APPROACH TO ATTRIBUTED-NETWORK CLUSTERING

MLE approach can be extended to this scenario

- A is the network matrix with edge attributes, X is the node vector with node attributes
- assume a model for generating A and X given the cluster assignment of nodes z, namely $\mathbb{P}(A, X \mid z)$
- compute the posteriori probability using Bayes rule

$$\mathbb{P}(z \mid A, X) = \frac{\mathbb{P}(A, X \mid z)\,\mathbb{P}(z)}{\mathbb{P}(A, X)} = \frac{\mathbb{P}(A \mid z)\,\mathbb{P}(X \mid z)\,\mathbb{P}(z)}{\mathbb{P}(A, X)}$$

where the last equality follows from the assumed conditional independence between A and X

Hard clustering algorithm

- idea identical to previous scenario with no attributes
- must consider both $\mathbb{P}(A \mid z)$ and $\mathbb{P}(X \mid z)$
- determining initial assignment z_0 is more challenging (must use network and attribute information)
- see (Dreveton et al., 2023), for example

Graph Neural Network (GNN) for network clustering

Neural network for learning on graphs using node and edge attributes

- node and edge attributes are the input to the neural network
- output is a vector for each node (with dimension much smaller than n): a representation for the node
- initially used as unsupervised technique for node classification and edge prediction
- recently adapted to other tasks, including network clustering
- idea: cluster the node representations (similar to spectral clustering)!

Challenges for network clustering with GNN

- design effective objective function for the neural network (in the unsupervised scenario)
- design effective mechanism to aggregate edge and node attributes from neighbors
- design GNNs that can find clusters in large networks

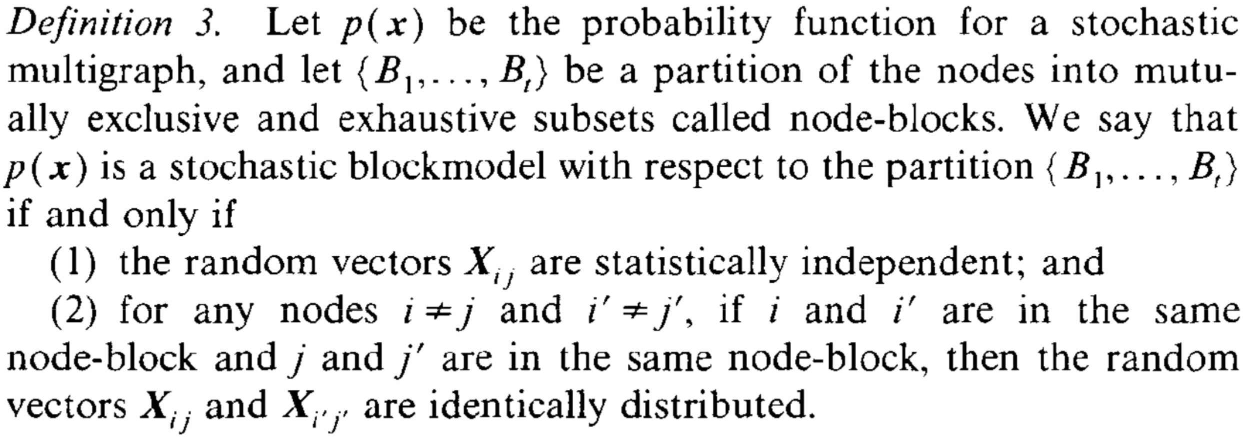
Growing recent literature on this topic

- see more details in Liu et al., 2023; Tsitsulin et al., 2023

Table of contents

Stochastic Block Model (SBM)

Original definition vs Modern definition



Definition 3. Let $p(\boldsymbol{x})$ be the probability function for a stochastic multigraph, and let $\{B_1,\ldots,B_t\}$ be a partition of the nodes into mutually exclusive and exhaustive subsets called node-blocks. We say that $p(\boldsymbol{x})$ is a stochastic blockmodel with respect to the partition $\{B_1,\ldots,B_t\}$ if and only if

(1) the random vectors $\boldsymbol{X}_{ij}$ are statistically independent; and

(2) for any nodes $i \neq j$ and $i' \neq j'$, if i and i' are in the same node-block and j and j' are in the same node-block, then the random vectors $\boldsymbol{X}_{ij}$ and $\boldsymbol{X}_{i'j'}$ are identically distributed.

Figure. Original definition of a SBM by (Holland et al., 1983).

Stochastic Block Model (SBM)

Original definition (Holland et al., 1983)

- n vertices partitioned into k clusters $C_1, \ldots, C_k$.
- Interaction between two vertices i and j is a **binary vector** $X_{ij} \in \{0,1\}^M$. Moreover,
  1. interactions $(X_{ij})_{i<j}$ are independent
  2. for any nodes $i \neq j$ and $i' \neq j'$, if i, i' are in a same cluster and j, j' are in a same cluster, then X_{ij} and $X_{i'j'}$ are identically distributed.

Originally: multiplex network, modelling the possibility of different types of edges.
Later: restricted to single layer ($M = 1$); $M \geq 2$ is called 'multilayer SBM'.

Important particular case: homogeneous SBM

- edges are decided randomly and independent for every pair of vertices
  - two vertices in the same cluster have an edge with probability p
  - two vertices in different clusters have an edge with probability q
- usually $p > q$ in order to reflect homophily.

HOMOGENEOUS SBM EXAMPLES

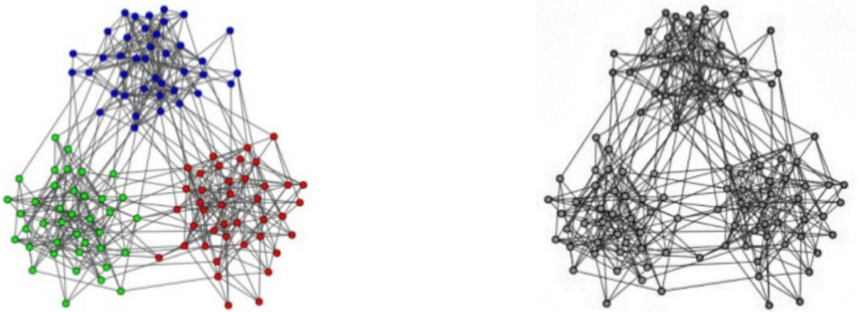


Figure. Homogeneous SBM with $n = 120$, $k = 3$, $p = 0.2$, $q = 0.01$

Statistical problem

- Generate the network (figure on the left);
- algorithm receives the edges (figure on the right);
- goal is to recover the ground truth partition.

Difficulty of the problem

- if $p \gg q$, it should be simple for most algorithms;
- if $p = q + \epsilon$, it should be difficult for all algorithms;
- if $p = q$ there is simply no information about the clusters.

Optimal misclassification rate in homogeneous SBM

For any $z \in [k]^n$, denote $n_a(z) = \sum_{u \in [n]} \mathbb{1}\{z_u = a\}$ the size of cluster $a \in [k]$. Let $\beta > 1$ and define

$$\mathcal{Z}_{n,k,\beta} \; = \; \left\{ z \in [k]^n \colon n_a(z) \in \left[\frac{n}{\beta k}, \beta \frac{n}{k} \right] \forall a \in [k] \right\}.$$

Let $\hat{z}$ be an estimator of z. We define the *loss* of $\hat{z}$ as

$$\mathrm{loss}(z, \hat{z}) \; = \; \min_{\tau \in \mathrm{Sym}(k)} \frac{1}{n} \sum_{u=1}^{n} \mathbb{1}\{z_u \neq \tau(\hat{z}_u)\},$$

where $\mathrm{Sym}(k)$ is the group of permutations of $[k]$ (we can only recover the *partition*, not the *labels*).

Aim: study the *expected loss* $\mathbb{E}_{G \sim \mathrm{SBM}(z,p,q)} [\mathrm{loss}(\hat{z}, z)]$ of an estimator $\hat{z}$.

Optimal rate in homogeneous SBM

Define the *Rényi divergence* of order 1/2 between two Bernoulli distributions

$$I \;=\; \mathrm{Ren}_{1/2}(\mathrm{Ber}(p), \mathrm{Ber}(q)) \;=\; -2\log\left(\sqrt{pq} + \sqrt{(1-p)(1-q)}\right).$$

Theorem 1 (Zhang and Zhou, 2016)

Suppose $\beta \in (1, \sqrt{2})$, and let $q < p$. If $\frac{nI}{k\log k} \gg 1$ we have

$$\inf_{\hat z} \sup_{z \in \mathcal{Z}_{n,k,\beta}} \mathbb{E}_{G \sim \mathrm{SBM}(z,p,q)}\left[\mathrm{loss}(\hat z, z)\right] \asymp \begin{cases} \exp\left(-(1+o(1))\frac{nI}{2}\right) & \text{if } k = 2, \\ \exp\left(-(1+o(1))\frac{nI}{\beta k}\right) & \text{if } k \geq 3. \end{cases}$$

Furthermore, if $\frac{nI}{k} = O(1)$ then $\inf_{\hat z} \sup_{z \in Z_\beta} \mathbb{E}\left[\mathrm{loss}(\hat z, z)\right] \geq c$ for some constant $c > 0$.

Understanding the theorem

The theorem hides two things

- A lower-bound for the expected loss of *any* algorithm;
- An upper-bound: there exist algorithms that achieve such expected loss. Which algorithms?
  - MLE (Zhang & Zhou, 2016); two-stage algorithms (Gao et al., 2017); semidefinite programs (Fei & Chen, 2018); VEM (Zhang & Zhou, 2020); spectral clustering (Zhang, 2023).

Optimal rate in homogeneous SBM

Two versus more than two communities

Two communities

- If the two communities are of different sizes (for example $n_1 > n_2$), then nodes in the community 1 have a higher expected degree than nodes in the community 2
- Hence the worst setting is when the two communities are of the same size
- The $n/2$ in the exponential error rate $e^{-(1+o(1))\frac{n}{2}I}$ represent the community sizes

Three (or more) communities

- One could think that having $k = 3$ communities of size n/k would be the worse, leading to an error rate of $e^{-(1+o(1))\frac{n}{k}I}$
- But, the worst case is two small communities of size $\frac{n}{\beta k}$ and one big of size $n - 2\frac{n}{\beta k}$. This leads to the minimax rate of $e^{-(1+o(1))\frac{n}{\beta k}I}$

Example: exact recovery in SBM

Setting: (approximately) equal-size communities ($\beta = 1 + o(1)$).

Exact recovery: recover the entire partition correctly.
More precisely: $\hat{z}$ solves exact recovery if $n \mathrm{loss}(z, \hat{z}) = 0$, which is equivalent to $n \mathrm{loss}(z, \hat{z}) < 1$.

Observations

- Minimax error rate: $\mathbb{E}\ [n \, \mathrm{loss}(\hat{z}, z)] \approx n e^{-(1+o(1))\frac{nI}{k}} \approx e^{-(1+o(1)) \log n \left(1 - \frac{nI}{k \log n}\right)}$.
- For $p = a \log n / n$ and $q = b \log n / n$, we have $I = (1 + o(1)) \left(\sqrt{a} - \sqrt{b}\right)^2 \frac{\log n}{n}$.

The two observations yields $\mathbb{E}\ [n \, \mathrm{loss}(\hat{z}, z)] \approx e^{-(1+o(1)) \log n \left(1 - \frac{(\sqrt{a} - \sqrt{b})^2}{k}\right)}$.

Theorem 2 (Abbe et al., 2016; Mossel et al., 2015)

Suppose $p = a \log n / n$, $q = b \log n / n$ and k constant. Exact recovery in homogeneous SBM with equal-size communities is:

1. *solvable and efficiently so if $\left(\sqrt{a} - \sqrt{b}\right)^2 > k$;*
2. *unsolvable if $\left(\sqrt{a} - \sqrt{b}\right)^2 < k$.*

Homogeneous SBM with edge covariates

'Modern' definition of SBM restricts interactions (edges) to belong to $\{0, 1\}$.

Generalisation: interactions take value in a space $\mathcal{S}$: multiplex networks ($\mathcal{S} = \{0, 1\}^M$), weighted networks ($\mathcal{S} = \mathbb{R}_+$), signed networks ($\mathcal{S} = \{0, -, +\}$), censored networks ($\mathcal{S} = \{\mathrm{unobserved}, \mathrm{observed\&present}, \mathrm{observed\&absent}\}$).

SBM with edge covariates : Let f and g be two pdf on $\mathcal{S}$. Conditionally on z, we observe $X \in \mathcal{S}^{n \times n}$ such that $X_{ij} = X_{ji}$ is sampled from f if $z_i = z_j$, and from g otherwise. We note $X \sim \mathrm{SBM}(z, f, g)$.

Define the *Rényi divergence* of order $1/2$ between f and g as

$$\mathrm{Ren}_{1/2}(f, g) \;=\; -2 \log \int \sqrt{\frac{df}{d\mu}} \sqrt{\frac{dg}{d\mu}} d\mu,$$

where μ is an arbitrary measure which dominates f and g.

Homogeneous SBM with edge covariates

Minimax rates

Theorem 3 (Avrachenkov et al., 2022; Xu et al., 2020)

Suppose $\beta \in (1,2)$*, and let* $I = \mathrm{Ren}_{1/2}(f,g)$*. If* $\frac{nI}{k \log k} \gg 1$*, we have*

$$\inf_{\hat{z}} \sup_{z \in \mathcal{Z}_\beta} \mathbb{E}_{X \sim \mathrm{SBM}(z,f,g)} \left[\mathrm{loss}(\hat{z}, z) \right] \asymp \begin{cases} \exp\left(-(1+o(1))\frac{nI}{2}\right) & \text{if } k = 2, \\ \exp\left(-(1+o(1))\frac{nI}{\beta k}\right) & \text{if } k \geq 3. \end{cases}$$

Furthermore, if $\frac{nI}{k} = O(1)$ *then* $\inf_{\hat{z}} \sup_{z \in Z_\beta} \mathbb{E}\left[\mathrm{loss}(\hat{z}, z)\right] \geq c$ *for some constant* $c > 0$*.*

Remarks

- Assumes f and g are known by the algorithm
- If f and g are unknown: results in (Xu et al., 2020) but with many additional technical conditions

Very similar to homogeneous SBM with binary interactions: Rényi-divergence is the key quantity. Why?

Homogeneous SBM with edge covariates

Why Rényi divergence? (1)

Setting: $n+1$ nodes, two communities of sizes $n/2$ and $n/2+1$; f and g denote the pdf for intra- and inter-cluster interactions.
Nodes $1,\cdots,n/2$ in community 1; nodes $n/2+1,\cdots,n$ in community 2. The last node $n+1$ belongs either to community 1 or 2.

Fundamental Testing Problem: A genie gives you $z = (\underbrace{1,\cdots,1}_{n/2},\underbrace{2,\cdots,2}_{n/2},?)$. You have to find z_{n+1}.

Denote $X = (A_{n+1,1}, A_{n+1,2},\cdots,A_{n+1,n}) \in \mathcal{S}^n$ (X_j denotes interaction between nodes $n+1$ and j) and the two hypothesis:

$$H_1 : z_{n+1} = 1 \quad \text{vs} \quad H_2 : z_{n+1} = 2.$$

Under H_1: $X \sim f^{\otimes n/2} \otimes g^{\otimes n/2} =: h_1$,
Under H_2: $X \sim g^{\otimes n/2} \otimes f^{\otimes n/2} =: h_2$.

$$\text{MLE: } \phi_{\mathrm{MLE}}(X) = \begin{cases} H_1 & \text{if } h_1(X) > h_2(X) \\ H_2 & \text{if } h_1(X) \le h_2(X). \end{cases}$$

Guarantee of MLE? Classic Chernoff–Stein theory of hypothesis testing applies for f and g independent of n (Cover & Thomas, 1999). But generalisation is possible.

HOMOGENEOUS SBM WITH EDGE COVARIATES

WHY RÉNYI DIVERGENCE? (2)

Under H_1: $X \sim f^{\otimes n/2} \otimes g^{\otimes n/2} =: h_1,$
Under H_2: $X \sim g^{\otimes n/2} \otimes f^{\otimes n/2} =: h_2.$

MLE: $\phi_{\mathrm{MLE}}(X) = \begin{cases} H_1 & \text{if } h_1(X) > h_2(X) \\ H_2 & \text{if } h_1(X) \leq h_2(X). \end{cases}$

Let $\mathrm{Ren}_t(f,g) = -(1-t)^{-1} \log \int f^t(x) g^{1-t}(x) dx$ be the Rényi divergence of order t between two pdf f and g, and define the *Chernoff information*

$$\mathrm{Chernoff}(h_1, h_2) = \sup_{t \in (0,1)} (1-t)\mathrm{Ren}_t(h_1, h_2).$$

Lemma 1 (Dreveton et al., 2024)

The worst-case error of $\phi \colon X \mapsto \phi(X) \in \{H_1, H_2\}$ *is* $r(\phi) = \max \left\{ \mathbb{P}_{H_1}(\phi(X) = H_2) ; \mathbb{P}_{H_2}(\phi(X) = H_1) \right\}.$
We have $\inf_\phi r(\phi) = r(\phi_{\mathrm{MLE}})$*. Moreover, if* $\mathrm{Chernoff}(h_1, h_2) \gg 1$ *we have*

$$r(\phi_{\mathrm{MLE}}) = e^{-(1+o(1)\mathrm{Chernoff}(h_1,h_2)}.$$

Remark: several particular cases of Lemma 1 appear in the literature (Abbe & Sandon, 2015; Gao et al., 2018).

Homogeneous SBM with edge covariates

Why Rényi divergence? (3)

Under H_1: $X \sim f^{\otimes n/2} \otimes g^{\otimes n/2} =: h_1$,
Under H_2: $X \sim g^{\otimes n/2} \otimes f^{\otimes n/2} =: h_2$.

MLE: $\phi_{\mathrm{MLE}}(X) = \begin{cases} H_1 & \text{if } h_1(X) > h_2(X) \\ H_2 & \text{if } h_1(X) \leq h_2(X). \end{cases}$

Final ingredient:

$$\begin{aligned} \mathrm{Chernoff}(h_1, h_2) &= \sup_{t\in(0,1)} (1-t)\mathrm{Ren}_t(\underbrace{f^{\otimes n/2} \otimes g^{\otimes n/2}}_{h_1}, \underbrace{g^{\otimes n/2} \otimes f^{\otimes n/2}}_{h_2}) \\ &= \sup_{t\in(0,1)} (1-t)\left[\sum_{i=1}^{n/2} \mathrm{Ren}_t(f,g) + \sum_{i=n/2+1}^{n} \mathrm{Ren}_t(g,f)\right] \quad \text{(linearity of Rényi divergence)} \\ &= \frac{n}{2} \sup_{t\in(0,1)} \left\{(1-t)\mathrm{Ren}_t(f,g) + t\,\mathrm{Ren}_{1-t}(f,g)\right\} \quad \text{using } (1-t)\mathrm{Ren}_t(g,f) = t\,\mathrm{Ren}_{1-t}(f,g) \\ &= \frac{n}{2}\mathrm{Ren}_{1/2}(f,g). \end{aligned}$$

Homogeneous SBM with edge covariates

Example: exact recovery in sparse SBM with edge covariates

Zero-inflated distribution : Suppose that the distributions f and g can be written as follows

$$f(x) \;=\; (1-a\rho_n)\delta_0(x) + a\rho_n\tilde{f}(x) \quad \text{and} \quad g(x) \;=\; (1-b\rho_n)\delta_0(x) + b\rho_n\tilde{g}(x), \tag{3.1}$$

When $\rho_n \ll 1$, the Rényi divergence $I = \mathrm{Ren}_{1/2}(f,g)$ between such zero-inflated distributions equals

$$I \;=\; (1+o(1))\rho_n \left[\left(\sqrt{a}-\sqrt{b}\right)^2 + 2\sqrt{ab}\,\mathrm{Hel}^2(\tilde{f},\tilde{g}) \right], \tag{3.2}$$

where $\mathrm{Hel}^2(\tilde{f},\tilde{g}) \in [0,1]$ is the *Hellinger divergence* defined by

$$\mathrm{Hel}^2(f,g) \;=\; \frac{1}{2}\int \left(\sqrt{\frac{df}{d\mu}} - \sqrt{\frac{dg}{d\mu}} \right)^2 d\mu.$$

Homogeneous SBM with edge covariates

Example: exact recovery in sparse SBM with edge covariates

Zero-inflated distribution : Suppose that the distributions f and g can be written as follows

$$f(x) \;=\; (1-a\rho_n)\delta_0(x) + a\rho_n \tilde{f}(x) \quad \text{and} \quad g(x) \;=\; (1-b\rho_n)\delta_0(x) + b\rho_n \tilde{g}(x), \tag{3.1}$$

When $\rho_n \ll 1$, the Rényi divergence $I = \mathrm{Ren}_{1/2}(f,g)$ between such zero-inflated distributions equals

$$I \;=\; (1+o(1))\rho_n \left[\left(\sqrt{a}-\sqrt{b}\right)^2 + 2\sqrt{ab}\,\mathrm{Hel}^2(\tilde{f},\tilde{g}) \right], \tag{3.2}$$

where $\mathrm{Hel}^2(\tilde{f},\tilde{g}) \in [0,1]$ is the *Hellinger divergence* defined by

$$\mathrm{Hel}^2(f,g) \;=\; \frac{1}{2}\int \left(\sqrt{\frac{df}{d\mu}} - \sqrt{\frac{dg}{d\mu}} \right)^2 d\mu.$$

Corollary [Exact recovery in sparse homogeneous SBM with edge covariates]

Consider an SBM with same-size communities and edge covariate distributions given in (3.1), where $\tilde{f},\tilde{g}$ are independent of n and $\rho_n = \log n/n$. Then, exact recovery is

- solvable if $\left(\sqrt{a}-\sqrt{b}\right)^2 + 2\sqrt{ab}\,\mathrm{Hel}^2(\tilde{f},\tilde{g}) > k$;
- unsolvable if $\left(\sqrt{a}-\sqrt{b}\right)^2 + 2\sqrt{ab}\,\mathrm{Hel}^2(\tilde{f},\tilde{g}) < k$.

$\mathrm{Hel}^2(\tilde{f},\tilde{g})$ characterises the additional information gained by observing the edge covariates.

TABLE OF CONTENTS

What was not covered?

Non-homogeneous models and node attributes

Observation : Pairwise interactions $(X_{ij})_{1\leq i,j\leq n}$ and node attributes $(Y_i)_{1\leq i\leq n}$

- $f_{ab}(X_{ij})$: probability of observing an interaction X_{ij} between a node i in block a and a node j in block b;
- $h_a(Y_i)$: probability of observing an attribute Y_i for a node i in a block a.

Conditional distribution of the data (X, Y) given block memberships z:

$$\mathbb{P}\left(X, Y \,|\, z\right) \;=\; \prod_{1\leq i<j\leq n} f_{z_i z_j}(X_{ij}) \prod_{i=1}^{n} h_{z_i}(Y_i).$$

How hard is it to recover z based on the observation of X and Y?
Denote $\pi_a = \frac{n_a(z)}{n}$ relative size of cluster a.
Key information-theoretic quantity is $\Delta \;=\; \min\limits_{\substack{a,b\in[K]\\ a\neq b}} \;\Delta(a,b)$ where

$$\Delta(a,b) \;=\; \sup_{t\in(0,1)} (1-t)\left[\; \underbrace{\sum_{c=1}^{K} \pi_c \mathrm{Ren}_t\left(f_{bc} \| f_{ac}\right)}_{\text{information from the network}} \;+\; \underbrace{\frac{1}{n}\mathrm{Ren}_t\left(h_b \| h_a\right)}_{\text{information from the attributes}} \;\right]. \qquad (4.1)$$

Results for exact recovery in (Dreveton et al., 2023).

What was not covered?

Non-homogeneous models and node attributes: Example

Contextual SBM: homogeneous SBM with Gaussian attributes

Suppose that $f_{\ell\ell'} = \begin{cases} \mathrm{Ber}\left(a\frac{\log n}{n}\right) & \text{if } \ell = \ell', \\ \mathrm{Ber}\left(b\frac{\log n}{n}\right) & \text{otherwise.} \end{cases}$ and $h_\ell = \mathcal{N}\left(\mu_\ell \log n, \sigma^2 I_d\right)$. Then,

$$\Delta \; = \; (1+o(1))\frac{\log n}{n}\left(\frac{\left(\sqrt{a}-\sqrt{b}\right)^2}{k} + \frac{\min_{a\neq b}\|\mu_a-\mu_b\|_2^2}{8\sigma^2}\right).$$

Non-homogeneous SBM with no node attributes

Consider $f_{\ell\ell'} = \mathrm{Ber}\left(\alpha_{\ell\ell'}\frac{\log n}{n}\right)$ and no attributes. Then,

$$\Delta \; = \; (1+o(1))\frac{\log n}{n}\min_{a\neq b}\sup_{t\in(0,1)}\sum_{c\in[k]}\pi_c\left(t\alpha_{bc}+(1-t)\alpha_{ac}-\alpha_{bc}^{t}\alpha_{ac}^{1-t}\right).$$

This last quantity is called Chernoff-Hellinger divergence in (Abbe & Sandon, 2015), and can also be interpreted as a Chernoff information (Leskelä, 2024).

What was not covered?

Degree-corrected SBM

- Under SBM: all nodes within the same community have the same degree distribution.
- Real-world networks: degree heterogeneity (even within communities).

Definition 4.1 (Degree-corrected SBM Karrer and Newman, 2011)

- ***cluster labels*** $z = (z_1, \cdots, z_n) \in [k]^n$;
- ***degree-correction parameters*** $\theta_1, \cdots, \theta_n$.

Generate a graph $G = ([n], E)$ such that

$$\mathbb{P}\left(\{i,j\} \in E \,|\, z_i, z_j\right) = \begin{cases} \min\{\theta_i\theta_j p, 1\} & \textit{if } z_i = z_j, \\ \min\{\theta_i\theta_j q, 1\} & \textit{otherwise.} \end{cases}$$

Theorem 4 (Minimax rates Gao et al., 2018)

(Under some technical conditions; in particular $p \asymp q \ll 1$) we have

$$\inf_{\hat{z}} \sup_{z,\theta \in \mathcal{P}_{n,k,\beta}} \mathbb{E}_{G\sim DCSBM(z,p,q,\theta)}\left[\mathrm{loss}(\hat{z}, z)\right] \asymp \begin{cases} \frac{1}{n}\sum_{i=1}^n \exp\left(-\theta_i \frac{n}{2}\left(\sqrt{p} - \sqrt{q}\right)^2\right) & \textit{if } k = 2, \\ \frac{1}{n}\sum_{i=1}^n \exp\left(-\theta_i \frac{n}{\beta k}\left(\sqrt{p} - \sqrt{q}\right)^2\right) & \textit{if } k \geq 3. \end{cases}$$

What was not covered?

Geometric extensions of the Stochastic Block Model

Definition 4.2 (Geometric Block Model)

- ***cluster labels*** *: $z_1, \cdots, z_n$ iid in $\mathbb{P}(z_i = 1) = \mathbb{P}(z_i = 2) = 1/2$;*
- ***geometric labels*** *: $x_1, \cdots, x_n$ uniformly distributed on the sphere $\mathcal{S}_{d-1} \subset \mathbb{R}^d$.*

Generate a graph $G = ([n], E)$ such that:

$$\mathbb{P}\left(\{i,j\} \in E \,|\, x_i, x_j, z_i, z_j\right) = \begin{cases} 0 & \text{if } \|x_i - x_j\| > \tau \\ p & \text{if } \|x_i - x_j\| \leq \tau \text{ and } z_i = z_j, \\ q & \text{if } \|x_i - x_j\| \leq \tau \text{ and } z_i \neq z_j. \end{cases}$$

Variant of this model: (Galhotra et al., 2018, 2023).

Recovery results

- Abbe et al., 2021; Gaudio et al., 2024: assumes the geometric labels $x_1, \cdots, x_n$ are known, only the clusters are latent.
- Recovery conditions with unknown geometric labels: open!
- Some standard clustering algorithms such as spectral clustering fail on geometric models (Avrachenkov et al., 2021).

What was not covered?

Semi-supervised community detection

Semi-supervised setting

- an oracle reveals the community labels of some nodes
- The oracle can be perfect (all community labels revealed are correct), or noisy (some labels are wrong)
- Goal: combine the graph *A* with the oracle information

Example of algorithms

- Label propagation (Zhu et al., 2003), Label Spreading (Zhou et al., 2003), Poisson learning (Calder et al., 2020)
- Constrained spectral clustering (Wang et al., 2014)
- Graph Neural Networks (Kipf & Welling, 2016)

Interesting fact

- In real networks, even a small amount of revealed labels helps a lot
- But in SBM, revealing a constant fraction $\eta \in (0,1)$ of community labels does not change the exact recovery threshold (Saad & Nosratinia, 2018)

Final Summary

How to detect network clusters?

- different algorithmic approaches to identify network clusters

When can network clusters be identified?

- different information-theoretic criteria for (exact) recovery

Variations and modern network clustering

- different models motivated by different applications

Much more information

- text-books, survey papers, and research papers in the appendix

References I

Abbe, E., Baccelli, F., & Sankararaman, A. (2021).**Community detection on euclidean random graphs.** *Information and Inference: A Journal of the IMA*, *10*(1), 109–160.

Abbe, E., Bandeira, A. S., & Hall, G. (2016).**Exact recovery in the stochastic block model.** *IEEE Transactions on Information Theory*, *62*(1), 471–487.

Abbe, E., & Sandon, C. (2015).**Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery.** *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, 670–688.

Avrachenkov, K., Bobu, A., & Dreveton, M. (2021).**Higher-order spectral clustering for geometric graphs.** *Journal of Fourier Analysis and Applications*, *27*(2), 22.

Avrachenkov, K., & Dreveton, M. (2022, October). ***Statistical analysis of networks*.** Boston-Delft: now publishers.

Avrachenkov, K., Dreveton, M., & Leskelä, L. (2022).**Community recovery in non-binary and temporal stochastic block models.** *arXiv preprint arXiv:2008.04790*.

Avrachenkov, K., El Chamie, M., & Neglia, G. (2014).**Graph clustering based on mixing time of random walks.** *IEEE International Conference on Communications (ICC)*, 4089–4094.

Barabási, A.-L. (2016). ***Network science*.** Cambridge University Press.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008).**Fast unfolding of communities in large networks.** *Journal of statistical mechanics: theory and experiment*, *2008*(10), P10008.

References II

- Calder, J., Cook, B., Thorpe, M., & Slepcev, D. (2020).**Poisson learning: Graph based semi-supervised learning at very low label rates.** *International Conference on Machine Learning*, 1306–1316.
- Cover, T., & Thomas, J. (1999). ***Elements of information theory*.** John Wiley & Sons.
- Dreveton, M., Fernandes, F., & Figueiredo, D. (2023).**Exact recovery and bregman hard clustering of node-attributed stochastic block model.** *Advances in Neural Information Processing Systems*, *36*.
- Dreveton, M., Gözeten, A., Grossglauser, M., & Thiran, P. (2024).**Universal lower bounds and optimal rates: Achieving minimax clustering error in sub-exponential mixture models.** *arXiv preprint arXiv:2402.15432*.
- Fei, Y., & Chen, Y. (2018).**Exponential error rates of sdp for block models: Beyond grothendieck's inequality.** *IEEE Transactions on Information Theory*, *65*(1), 551–571.
- Ford, L. R., & Fulkerson, D. R. (1957).**A simple algorithm for finding maximal network flows and an application to the hitchcock problem.** *Canadian journal of Mathematics*, *9*, 210–218.
- Fortunato, S., & Barthelemy, M. (2007).**Resolution limit in community detection.** *Proceedings of the national academy of sciences*, *104*(1), 36–41.
- Galhotra, S., Mazumdar, A., Pal, S., & Saha, B. (2018).**The geometric block model.** *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1).
- Galhotra, S., Mazumdar, A., Pal, S., & Saha, B. (2023).**Community recovery in the geometric block model.** *Journal of Machine Learning Research*, *24*(338), 1–53.

References III

Gao, C., Ma, Z., Zhang, A. Y., & Zhou, H. H. (2017).**Achieving optimal misclassification proportion in stochastic block models.** *Journal of Machine Learning Research*, *18*(60), 1–45.

Gao, C., Ma, Z., Zhang, A. Y., & Zhou, H. H. (2018).**Community detection in degree-corrected block models.** *The Annals of Statistics*, *46*(5), 2153–2185. https://doi.org/10.1214/17-AOS1615

Gaudio, J., Niu, X., & Wei, E. (2024).**Exact community recovery in the geometric sbm.** *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2158–2184.

Girvan, M., & Newman, M. E. (2002).**Community structure in social and biological networks.** *Proceedings of the national academy of sciences*, *99*(12), 7821–7826.

Henzinger, M., Li, J., Rao, S., & Wang, D. (2024).**Deterministic near-linear time minimum cut in weighted graphs.** *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 3089–3139.

Holland, P. W., Laskey, K. B., & Leinhardt, S. (1983).**Stochastic blockmodels: First steps.** *Social networks*, *5*(2), 109–137.

Karger, D. R. (2000).**Minimum cuts in near-linear time.** *Journal of the ACM (JACM)*, *47*(1), 46–76.

Karrer, B., & Newman, M. E. (2011).**Stochastic blockmodels and community structure in networks.** *Physical review E*, *83*(1), 016107.

Kernighan, B. W., & Lin, S. (1970).**An efficient heuristic procedure for partitioning graphs.** *The Bell system technical journal*, *49*(2), 291–307.

References IV

- Kipf, T. N., & Welling, M. (2016).**Semi-supervised classification with graph convolutional networks.** *arXiv preprint arXiv:1609.02907*.
- Leskelä, L. (2024).**Information divergences and likelihood ratios of poisson processes and point patterns.** *arXiv preprint arXiv:2404.00294*.
- Liu, Y., Liang, K., Xia, J., Zhou, S., Yang, X., Liu, X., & Li, S. Z. (2023).**Dink-net: Neural clustering on large graphs.** *International Conference on Machine Learning*, 21794–21812.
- Manipur, I., Giordano, M., Piccirillo, M., Parashuraman, S., & Maddalena, L. (2021).**Community detection in protein-protein interaction networks and applications.** *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *20*(1), 217–237.
- Mossel, E., Neeman, J., & Sly, A. (2015).**Consistency thresholds for the planted bisection model.** *Proc. 47th annual ACM Symposium on Theory of Computing*, 69–75.
- Newman, M. E., & Girvan, M. (2004).**Finding and evaluating community structure in networks.** *Physical review E*, *69*(2), 026113.
- Peixoto, T. P. (2014).**Efficient monte carlo and greedy heuristic for the inference of stochastic block models.** *Physical Review E*, *89*(1), 012804.
- Saad, H., & Nosratinia, A. (2018).**Community detection with side information: Exact recovery under the stochastic block model.** *IEEE Journal of Selected Topics in Signal Processing*, *12*(5), 944–958.
- Tsitsulin, A., Palowitch, J., Perozzi, B., & Müller, E. (2023).**Graph clustering with graph neural networks.** *Journal of Machine Learning Research*, *24*(127), 1–21.

References V

- Wang, X., Qian, B., & Davidson, I. (2014).**On constrained spectral clustering and its applications.** *Data Mining and Knowledge Discovery*, *28*, 1–30.
- Xu, J., & Chen, H. (2005).**Criminal network analysis and visualization.** *Communications of the ACM*, *48*(6), 100–107.
- Xu, M., Jog, V., & Loh, P.-L. (2020).**Optimal rates for community estimation in the weighted stochastic block model.** *Annals of Statistics*, *48*(1), 183–204.
- Zhang, A. Y., & Zhou, H. H. (2016).**Minimax rates of community detection in stochastic block models.** *The Annals of Statistics*, *44*(5), 2252–2280. https://doi.org/10.1214/15-AOS1428
- Zhang, A. Y., & Zhou, H. H. (2020).**Theoretical and computational guarantees of mean field variational inference for community detection.** *The Annals of Statistics*, *48*(5), 2575–2598.
- Zhang, A. Y. (2023).**Fundamental limits of spectral clustering in stochastic block models.** *arXiv preprint arXiv:2301.09289*.
- Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2003).**Learning with local and global consistency.** *Advances in neural information processing systems*, *16*.
- Zhu, X., Ghahramani, Z., & Lafferty, J. D. (2003).**Semi-supervised learning using gaussian fields and harmonic functions.** *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 912–919.