Subgraphs and Community Structure of Networks

Mainack Mondal

CS 60017
Autumn 2021

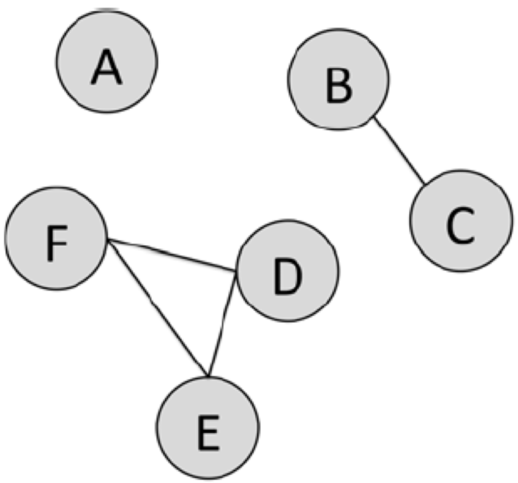
Subgraphs

- A subset of nodes and edges in a network
- Given a (social) network, what are some subgraphs of interest?

Subgraphs

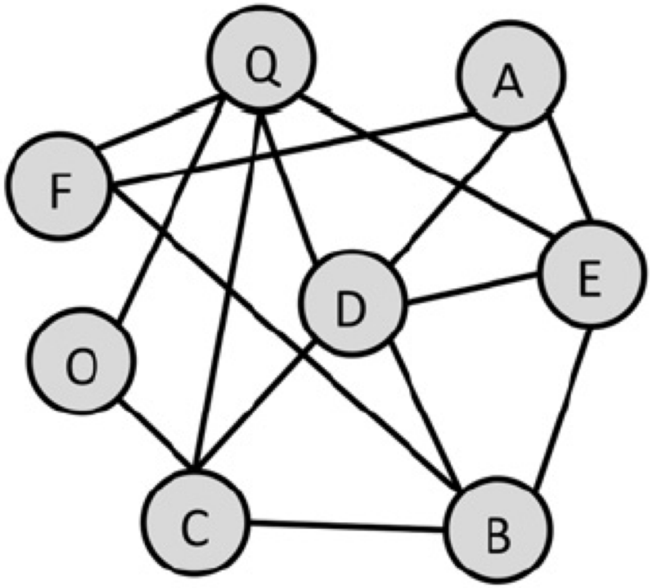
- A subset of nodes and edges in a network

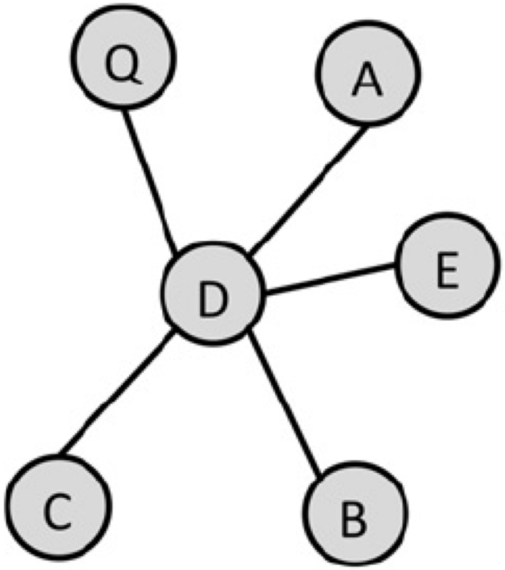
- Given a (social) network, what are some subgraphs of interest?
  - Singletons: Isolated nodes
  - Connected components
  - Triads or triangles
  - Larger cliques



Egocentric networks

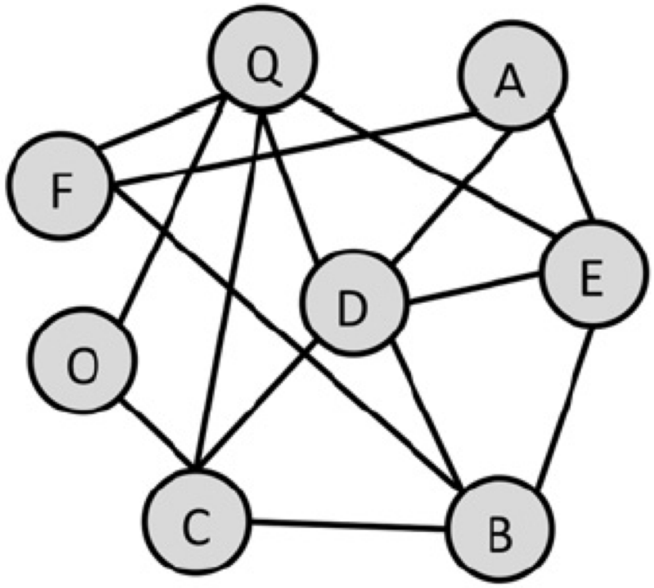
- From the perspective of a node (user)
- 1-degree egocentric network: a node and all its connections to its neighbors

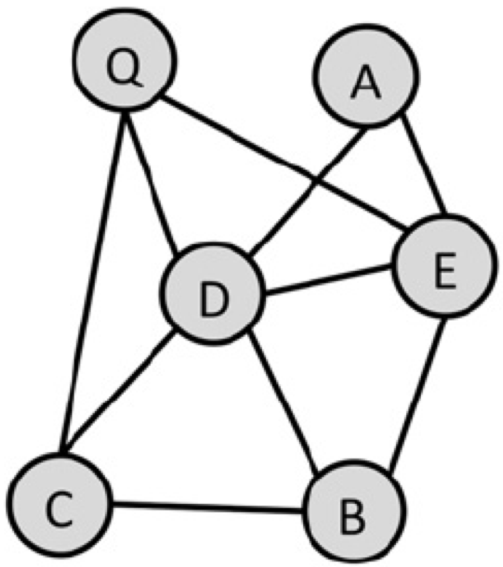




Egocentric networks

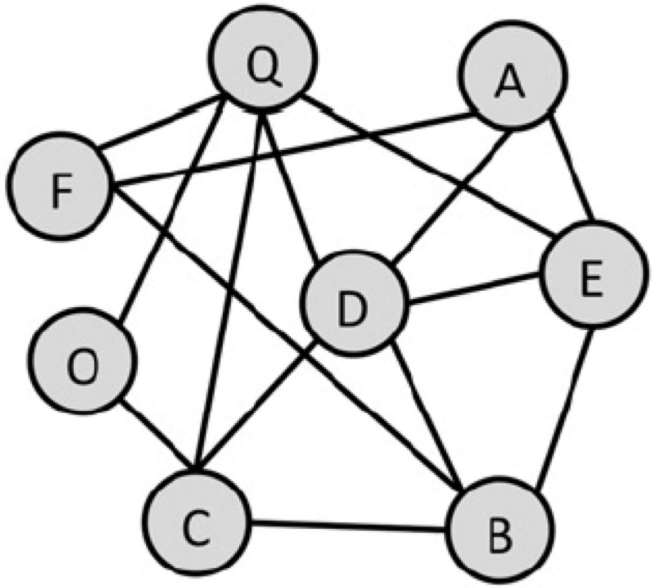
- 1.5-degree egocentric network: a node, all its connections to its neighbors, and the connections among the neighbors

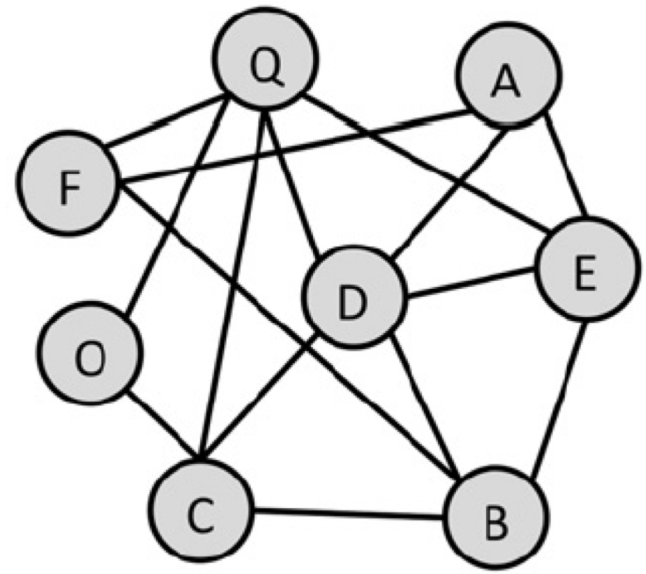




Egocentric networks

- 2-degree egocentric network: a node, all its neighbors, all neighbors of neighbors, and the connections among all these nodes





Communities

- Community or network cluster
  - Typically a group of nodes having more and / or better interactions among its members, than between its members and the rest of the network

- No unique formal definition

COMMUNITY DETECTION

Community detection algorithms

- Lot of applications – identifying similar nodes, close friends, recommendation, …

- Challenging
  - Communities are not well-defined
  - Number of communities in a network is not known

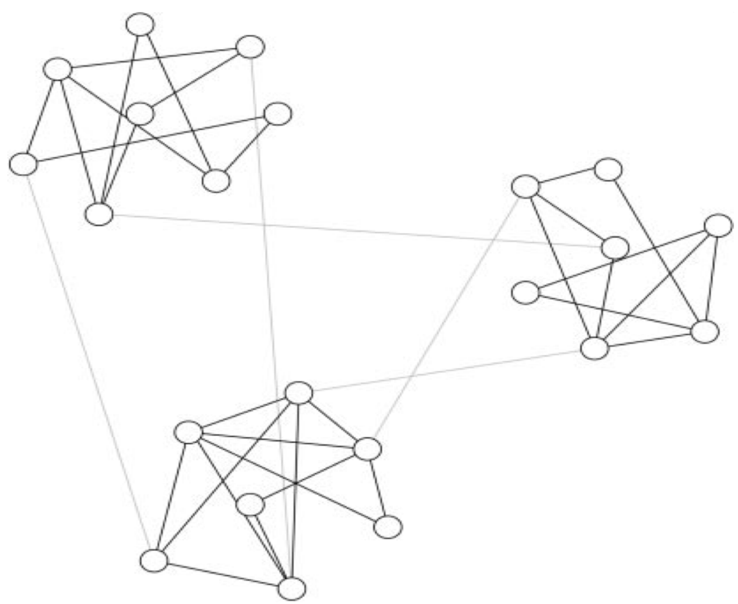
Two broad types of algorithms

- Detection of disjoint communities
    - Each community is a partition of the network

- Detection of overlapping communities
    - A node can be members of multiple communities

Algorithm by Girvan & Newman

- Community structure in social and biological networks, PNAS, 2002
- Focus on edges that are most “between” communities



Edge betweenness

- Edge betweenness of an edge *e*: fraction of shortest paths between all pairs of vertices, which run through *e*

- Edges between communities are likely to have high betweenness centrality

- Progressively remove edges having high betweenness centrality, to separate communities from one another

Girvan-Newman algorithm

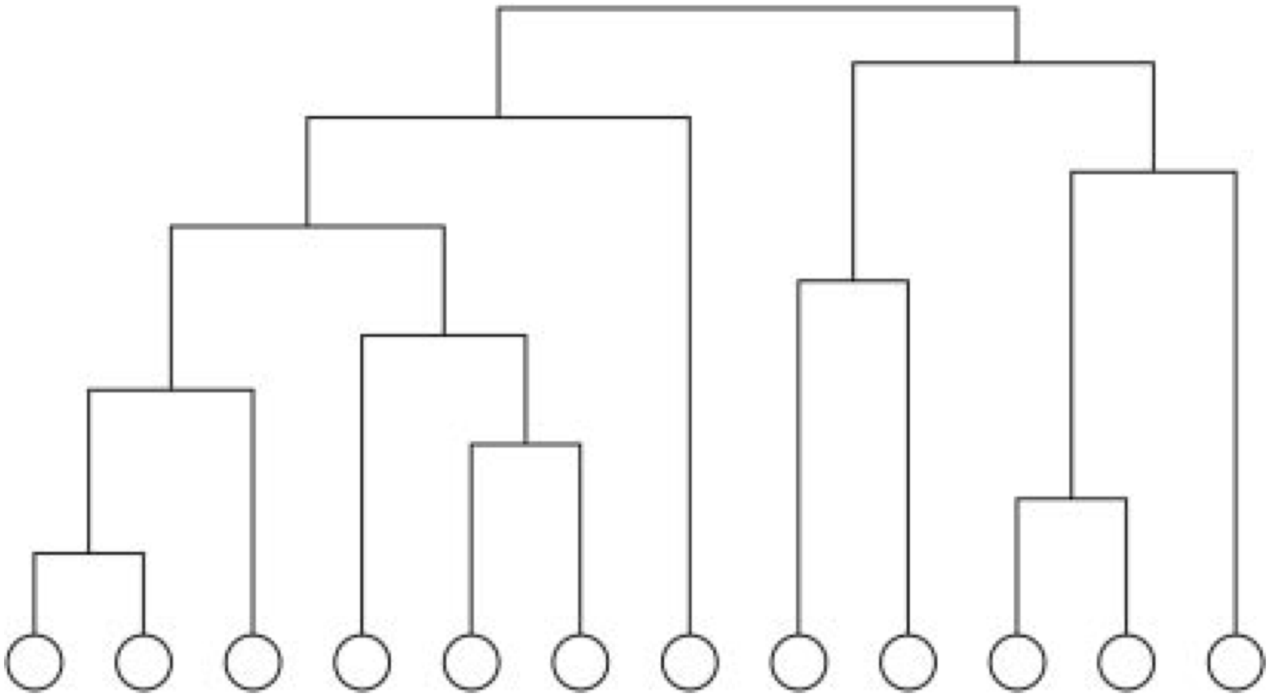
- Compute betweenness centrality for all edges
- Remove the edge with highest betweenness centrality
- Re-compute betweenness centrality for all edges affected by the removal
- Repeat steps 2 and 3 until no edges remain

- Time complexity
  - Graph of n vertices and m edges: betweenness centrality of all edges can be computed in $O(mn)$ time
  - Hence, worst case time complexity: $O(m^2n)$

How many communities?

- Community structure of a graph is hierarchical, with smaller communities nested within larger ones
- Represented as a hierarchical clustering tree: dendrogram
- A “slice” through the tree at any level gives a certain number of communities
- Which level to slice at?

An example dendrogram



Hierarchical clustering algorithms

- Agglomerative algorithms (bottom-up)
  - Clusters / communities iteratively merged if their similarity is sufficiently high
- Divisive algorithms (top-down)
  - Clusters / communities iteratively split by removing edges

- Both can be represented by dendrograms
- Need some way to decide at what level to slice the dendrogram – what is a good community structure?

What is a good community structure?

- A few large communities, or many small communities?
- Often depends on the end application

- Example: find communities in an OSN for
  - Application 1: personalized recommendation to users
  - Application 2: map user-accounts to data centers located in some places

Objective functions for Community Detection (CD)

- Community or network cluster
  - Typically a group of nodes having more and / or better interactions among its members, than between its members and the rest of the network

- Typical CD algorithms
  - Choose an objective function that captures the above intuition
  - Optimize the objective function using heuristics or approximation algorithms

OBJECTIVE FUNCTIONS FOR COMMUNITY DETECTION

Empirical Comparison of Algorithms for Network Community Detection, Leskovec et al., WWW 2010

Various objective functions

- Two criteria of interest for measuring how well a particular set S of nodes represents a community
  - Number of edges among the nodes within S
  - Number of edges between nodes in S and rest of network

- Two types of objective functions
  - Single criterion – considers any one of the above criteria
  - Multi criterion – considers both the above criteria

Multi-criterion scores

- Consider both the criteria for measuring quality of a set S of nodes
- Lower values of $f(S)$ signify a more community-like set of nodes

Notations

- $G = (V, E)$ is the network.
- $n = |V| =$ number of nodes
- $m = |E| =$ number of edges
- $d(u) = k_u =$ degree of node u

- S: set of nodes
- $n_s =$ number of nodes in S
- $m_s =$ number of edges within S (both nodes in S)
- $c_s =$ number of edges on the boundary of S

Expansion

$$f(S) = \frac{c_S}{n_S}$$

- Number of edges per node in S, that points outside the set S

Internal density

$$f(S) = 1 - \frac{m_S}{n_S(n_S - 1)/2}$$

- Internal edge density of the set S

Cut Ratio

$$f(S) = \frac{c_S}{n_S(n - n_S)}$$

- Fraction of all possible edges leaving the set S

Conductance

$$f(S) = \frac{c_S}{2m_S + c_S}$$

- Fraction of total edge volume that points outside the cluster
- Edge volume = sum of node-degrees
- Denominator: total connection from nodes in S to all nodes in graph G

Normalized Cut

$$f(S) = \frac{c_S}{2m_S + c_S} + \frac{c_S}{2(m - m_S) + c_S}$$

- Originally proposed in “Normalized cuts and Image Segmentation” by Shi et al, IEEE TPAMI, 2000
- Some doubts about the denominator of the second term

Normalized cut – original definition

- Partition graph G = (V, E) into two partitions A and B

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v).$$

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}, \qquad (2)$$

where $assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$ is the total connection from nodes in A to all nodes in the graph and $assoc(B,V)$ is similarly defined.

Maximum Out Degree Fraction (ODF)

$$\max_{u \in S} \frac{|\{(u,v) : v \notin S\}|}{d(u)}$$

- Maximum fraction of edges of a node in S, that points outside the set S

Average ODF

$$f(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{(u,v) : v \notin S\}|}{d(u)}$$

- Average fraction of edges of nodes in S, that points outside S

Flake ODF

$$f(S) = \frac{|\{u : u \in S, |\{(u,v) : v \in S\}| < d(u)/2\}|}{n_S}$$

- Fraction of nodes in S that have fewer edges pointing inside S, than to outside S

Observations by Leskovec et al.

- Internal density and Maximum-ODF are not good measures for community quality
  - Does not show much variation, except for very small communities
- Cut ratio has high variance
  - communities of similar sizes can have very different numbers of edges pointing outside

- Both very low variance and very high variance undesirable for objective functions for CD

Observations by Leskovec et al.

- Flake-ODF prefers larger communities

- Conductance, expansion, normalized cut, average-ODF all exhibit qualitatively similar behavior and give best scores to similar clusters

Single-criterion scores

- Consider only one of the two criteria for measuring quality of a set S of nodes

- Two simple single-criterion scores:
  - Volume: Sum of degrees of the nodes in *S*
  - Edges Cut: c_S: Number of edges needed to be removed to disconnect nodes in *S* from the rest of the network

Modularity-based measures

- A set of nodes is a good community if the number of edges within the set is significantly more than what can be expected by random chance

- Modularity $Q = 1/K * (m_S - E(m_S))$

  - Number of edges m_S within set S, minus expected number of edges within the set S
  - K is a constant, used for normalization

Modularity ratio

$$\frac{m_S}{E(m_S)}$$

- Alternative measure of how well set S represents a community
- Ratio of the number of edges among nodes in S, and expected number of such edges

Expected number of edges

- Null model: Erdos-Renyi random network having the same node degree sequence as given network

- Randomized realization of a given network, realized in practice using Configuration Model
  - Cut each edge into two half-edges or stubs
  - Randomly connect each stub to any stub
  - Expected to have no community structure

Mathematical definition of Modularity

- For two particular nodes *i* and *j* :
  - Number of edges between the nodes: A_{ij}
  - Degrees: k_i, k_j
  - Expected number of links between i and j: $k_i k_j /2m$

- Do the nodes *i* and *j* have more edges than expected by random chance?

$$A_{ij} - k_i k_j /2m$$

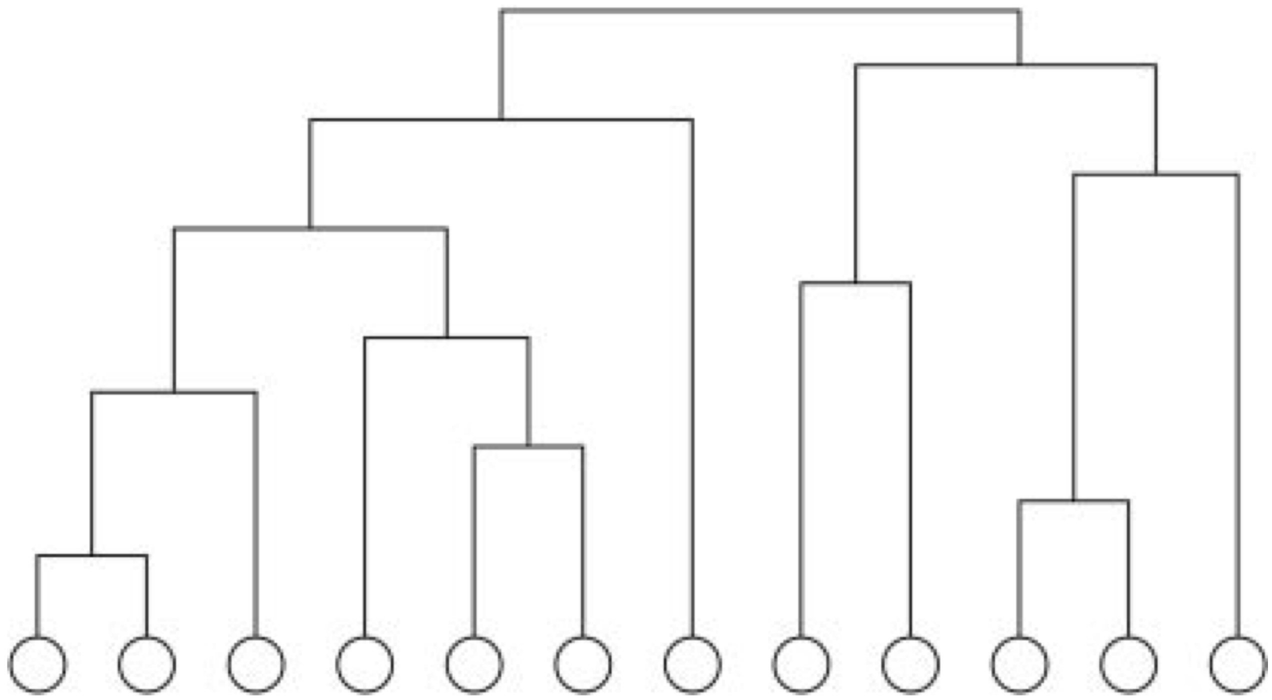
Modularity for a given network

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)$$

- The delta function is 1 if both nodes *i* and *j* are in the same community ($C_i = C_j$), 0 otherwise
- Consider a network with two communities c1, c2
  - Q is the fraction of edges that fall within c1 or c2, minus the expected number of edges within c1 and c2 for a random graph with the same node degree distribution as the given network

Using modularity for CD

- Approach 1: use Modularity to decide at which level to slice the dendrogram



Using modularity for CD

- Approach 1: use Modularity to decide at which level to slice the dendrogram

- Approach 2: Optimize modularity
  - Exhaustive maximization is NP-hard
  - Heuristics and approximations used

Greedy algorithm for maximizing Q

- Fast algorithm for detecting community structure in networks, Newman, PRE 69(6), 2004

- Greedy agglomerative hierarchical clustering
  - Start with n clusters, each containing a single node
  - Add edges such that the new partitioning gives the maximum increase (minimum decrease) of modularity wrt the previous partitioning
  - A total of *n* partitionings found, with number of clusters varying from *n* to 1
  - Select the partitioning having highest modularity

Most popular Q optimization algorithm

- Louvain algorithm:
    - https://perso.uclouvain.be/vincent.blondel/research/louvain.html

- Optimization in two steps
    - Step 1: look for small communities - optimizing Q locally
    - Step 2: aggregate nodes in the same community and build a new network whose nodes are the communities
    - Repeat iteratively until a maximum of modularity is attained and a hierarchy of communities is produced
    - Time: approx *O(n log n)*

For reading

- Many subsequent works have suggested improvements for maximizing modularity
    - Reducing time complexity
    - Normalizing with number of edges to minimize bias towards larger communities
    - …

- Read “Community detection in graphs” by Fortunato, Physics Reports, 2010.