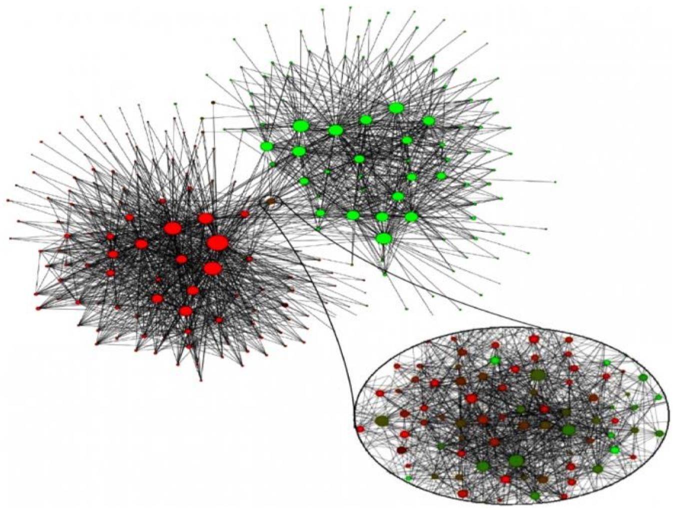
Community Detection in Social Networks

Reference: Network Science by Barabasi

Communities in Belgium

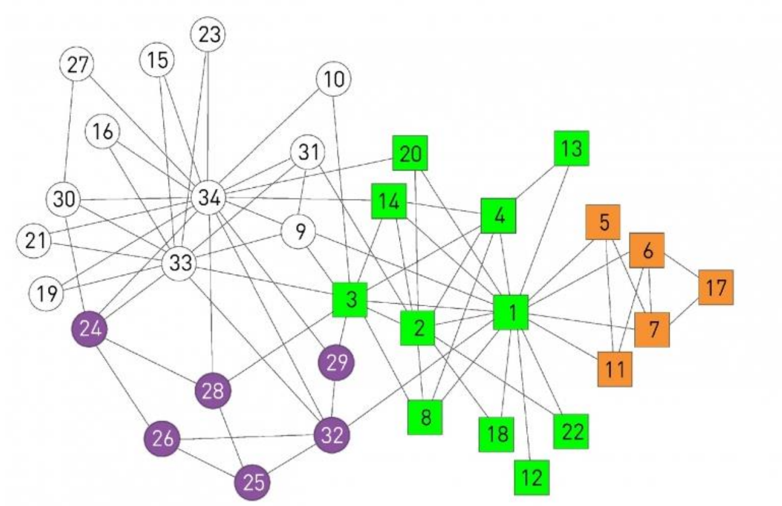
- Belgium, the model bicultural society: 59% of its citizens are Flemish, speaking Dutch and 40% are Walloons who speak French.
- What is the reason for the peaceful coexistence of these two ethnic groups since 1830 ?
  - Is it densely knitted society ?
  - Or we have two nations with the same borders, that learned to minimize contact with each other?
- Answer to the above question:
  - Research by Vincent Blondel and his students in 2007, who developed an algorithm to identify the country's community structure.

Communities extracted from the call pattern of the consumers of the largest Belgian mobile phone company



Community Detection : Major Application Areas

- Social Networks
  - E.g Zachary's Karate Club
- Biological Networks



Basics of Communities

- Fundamental Hypotheses
  - *A network's community structure is uniquely encoded in its wiring diagram.*
- What do we really mean by a community?
- How many communities are in a network?
- How many different ways can we partition a network into communities?

5

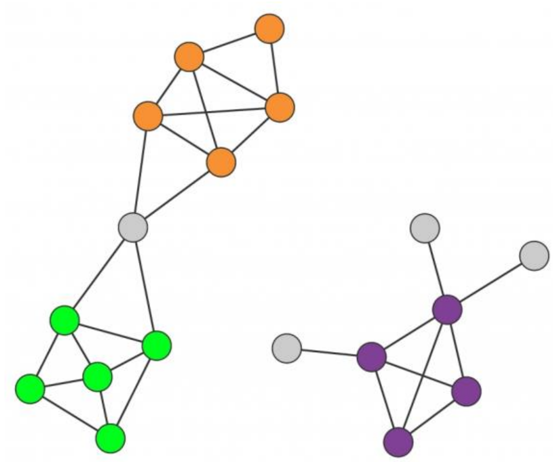
Defining Communities

- **Connectedness and Density Hypothesis**

  A community is a locally dense connected subgraph in a network.
- **Connectedness Hypothesis**
  - Each community corresponds to a connected subgraph
- **Density Hypothesis**
  - Nodes in a community are more likely to connect to other members of the same community than to nodes in other communities.

6

Connectedness and Density Hypothesis



Maximum Cliques

- Community as a group of individuals whose members all know each other
    - i.e. a complete graph or a Clique
- Does a Clique satisfy our hypothesis of connectedness and density ???
- However, there are drawbacks
    - While triangles are frequent in networks, larger cliques are rare.
    - Requiring a community to be a complete subgraph may be too restrictive, missing many other legitimate communities.

Strong and Weak Communities

- Consider a connected subgraph C of N_C nodes in a network
- k_i^{int} : *internal degree* of node *i*
  - number of links that connect *i* to other nodes in *C*.
- k_i^{ext} :external degree of node i
  - number of links that connect *i* to the rest of the network.
- If $k_i^{ext}=0$, each neighbor of *i* is within *C*, hence *C* is a good community for node *i*.
- If $k_i^{int}=0$, then node *i* should be assigned to a different community.

9

Strong and Weak Communities…

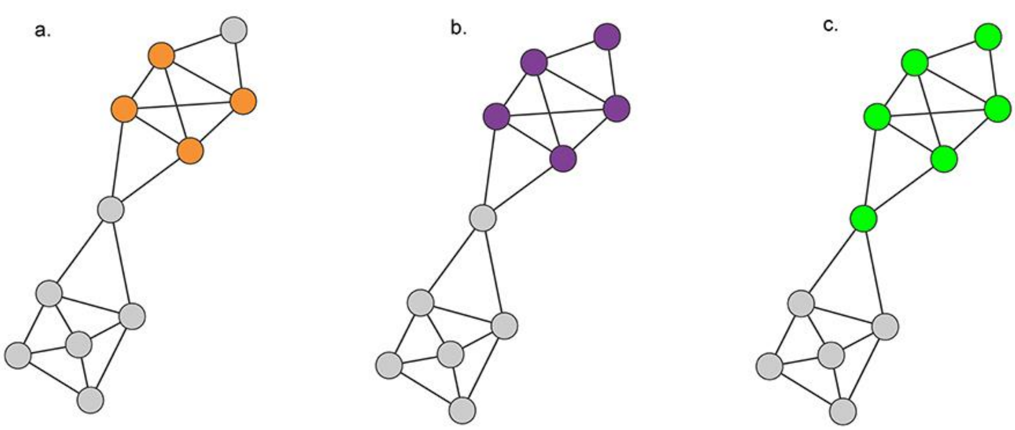
- **Strong Community**
  - *C* is a *strong community* if each node within *C* has more links within the community than with the rest of the graph
  - a subgraph *C* forms a strong community if for each node $i \in C$,
    - $k_i^{int}(C) > k_i^{ext}(C)$
- **Weak Community**
  - *C* is a *weak community* if the total internal degree of a subgraph exceeds its total external degree.
  - a subgraph *C* forms a weak community if,
  - $\sum_{i \in c} k_i^{int}(c) > \sum_{i \in c} k_i^{ext}(c)$

10

Strong and Weak Communities…

- Is clique a strong community?
- Is a strong community also a weak community?
- What about vice versa ?

Strong and Weak Communities…



Number of Communities

- How many ways can we group the nodes of a network into communities?
- Graph Bisection
  - Divide a network into two non-overlapping subgraphs, such that the number of links between the nodes in the two groups, called the *cut size*, is minimized
  - Graph Partitioning
    - inspecting all possible divisions into two groups and choosing the one with the smallest cut size
    - number of distinct ways we can partition a network of N nodes into groups of N_1 and N_2 nodes is,

13

Number of Communities…

Number of distinct ways we can partition a network of N nodes into groups of N_1 and N_2 nodes is,

$$\frac{N!}{N_1!\,N_2!}$$

Using Stirling's formula

$$n! \approx 2\pi n(n|\mathrm{e})^n$$

For two equal sizes of N_1 and N_2 ,

$$\frac{N!}{N_1!N_2!} = \mathrm{e}^{(N+1)\ln 2 - \frac{1}{2}\ln N}$$

The number of bisections increases exponentially with the size of the network.

14

Number of Communities…

- Consider a network with 10 nodes which we bisect into two subgraphs of size $N_1 = N_2 = 5$
  - What are the possible number of bisections?
- Now consider a network with 100 nodes with two subgraphs of size $N_1 = N_2 = 50$
  - What are the possible number of bisections?
- What are your observations from the above results???
- Is the Brute force approach feasible to compute graph bisection even for a modest size of network?

Difference between Graph Partitioning and Community Detection

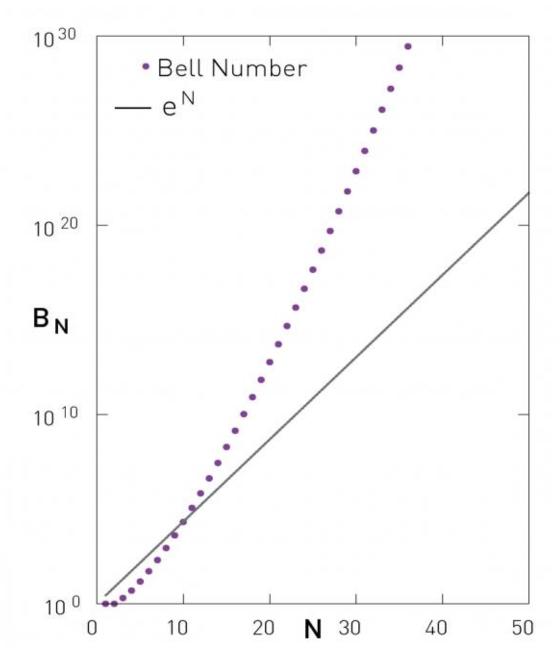
- Graph partitioning divides a network into a predefined number of smaller subgraphs.
- In contrast community detection aims to uncover the inherent community structure of a network.
- Consequently in most community detection algorithms the number and the size of the communities is not predefined, but needs to be discovered by inspecting the network's wiring diagram.

Community Detection

- Divide a network into an arbitrary number of groups, such that each node belongs to one and only one group
- The number of possible partitions are given by Bell Number as shown below:

$$B_N = \frac{1}{e}\sum_{j=0}^{\infty}\frac{j^N}{j!}$$

17



We therefore need polynomial time algorithms that can identify communities without inspecting all partitions

18

Hierarchical Clustering

1) Calculate *Similarity matrix*, whose elements x_{ij} indicate the distance of node *i* from node *j*.
2) Iteratively identify groups of nodes with high similarity
   1) ***Agglomerative algorithms*** merge nodes with high similarity into the same community
   2) ***Divisive algorithms*** isolate communities by removing low similarity links that tend to connect communities.
3) Outcome: a hierarchical tree, called a dendrogram, that predicts the possible community partitions

Agglomerative Procedures: the Ravasz Algorithm

- Step 1: Define the Similarity Matrix
- Step 2: Decide Group Similarity
- Step 3: Apply Hierarchical Clustering
- Step 4: Dendrogram

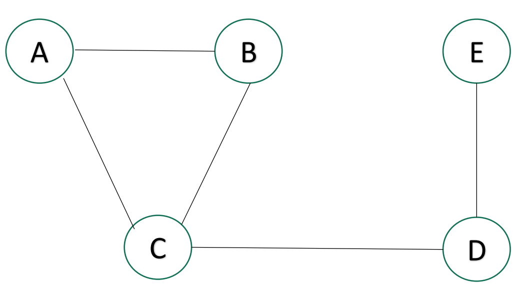
Ravasz Algorithm : Step 1-Similarity Matrix

- The topological overlap matrix,

$$x_{ij}^{0} = \frac{j(i,j)}{\min(k_i, k_j) + 1 - \theta(A_{ij})}$$

- *Θ(x)* is the Heaviside step function, which is zero for $x \leq 0$ and one for $x > 0$;
- *J(i, j)* is the number of common neighbors of node *i* and *j*, to which we add one (+1) if there is a direct link between *i* and *j*;
- $\min(k_i, k_j)$ is the smaller of the degrees k_i and k_j

Ravasz Algorithm : Step 1-Similarity Matrix…



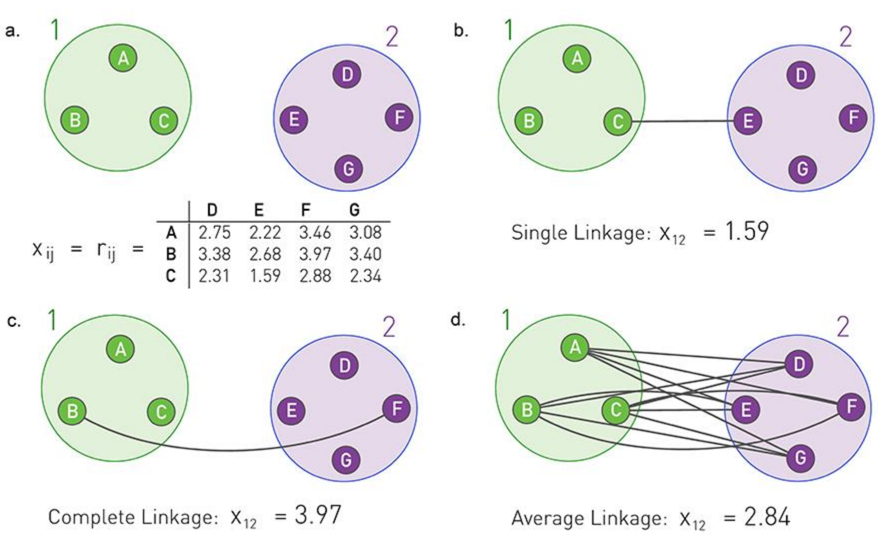
$$x_{ij}^{0} = \frac{j(i,j)}{\min(k_i, k_j) + 1 - \theta(A_{ij})}$$

$$\begin{bmatrix} - & & & & \\ 1 & - & & & \\ 1 & 1 & - & & \\ 1/3 & 1/3 & 1/2 & - & \\ 0 & 0 & 1/2 & 1 & - \end{bmatrix}$$

Ravasz Algorithm : Step 2-Decide Group Similarity

- We need to determine the similarity of two communities from the node similarity matrix x_{ij}
- Single Linkage Clustering
  - The similarity between communities 1 and 2 is the smallest of all x_{ij} , where *i* and *j* are in different communities.
- Complete Linkage Clustering
  - The similarity between two communities is the maximum of x_{ij} , where *i* and *j* are in distinct communities.
- Average Linkage Clustering
  - The similarity between two communities is the average of x_{ij} over all node pairs *i* and *j* that belong to different communities.
  - Ravasz algorithm uses this procedure

Ravasz Algorithm : Step 2-Decide Group Similarity



Ravasz Algorithm : Step 3: Apply Hierarchical Clustering

1. Assign each node to a community of its own and evaluate x_{ij} for all node pairs.
2. Find the community pair or the node pair with the highest similarity and merge them into a single community.
3. Calculate the similarity between the new community and all other communities.
4. Repeat Steps 2 and 3 until all nodes form a single community.

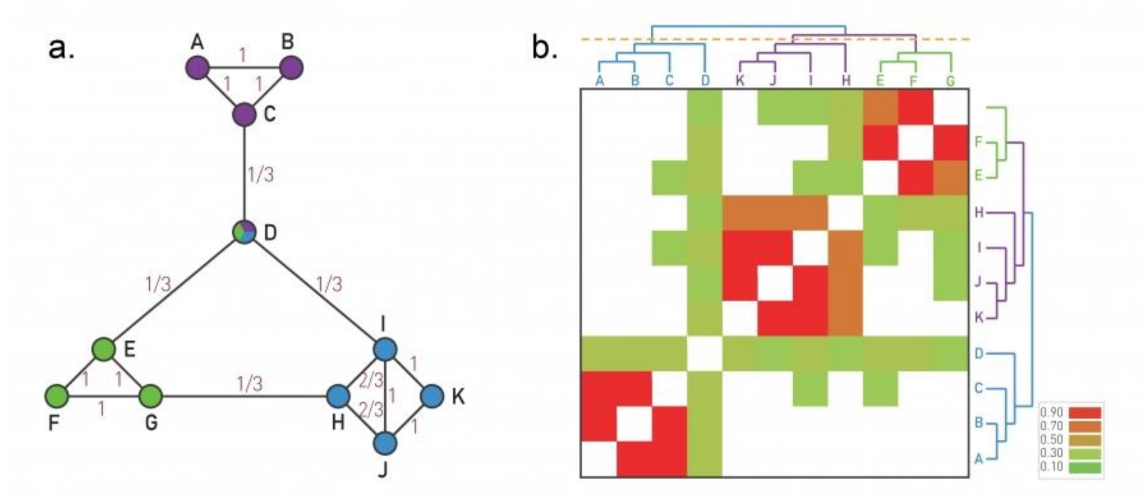
25

Ravasz Algorithm : Step 4-Dendrogram

- To extract the underlying community organization
  - By cutting the Dendrogram
- The dendrogram visualizes the order in which the nodes are assigned to specific communities.

26

Agglomerative Procedures: the Ravasz Algorithm



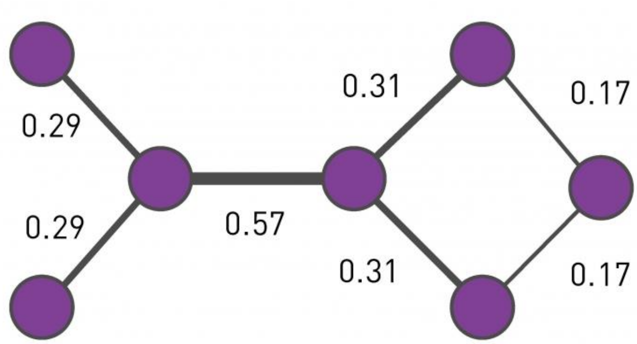
Ravasz Algorithm : Computational Complexity

- Exercise: Is it a polynomial time algorithm?

Divisive Procedures: The Girvan-Newman algorithm

- Idea: Systematically remove the links connecting nodes that belong to different communities, eventually breaking a network into isolated communities.
- Step 1: Define Centrality X_{ij} using Link Betweenness
  - i.e the number of shortest paths that pass through link (i, j)
  - Large X_{ij} for the links connecting nodes in different communities
  - Small X_{ij} for the links connecting nodes in same community

Divisive Procedures: The Girvan-Newman algorithm

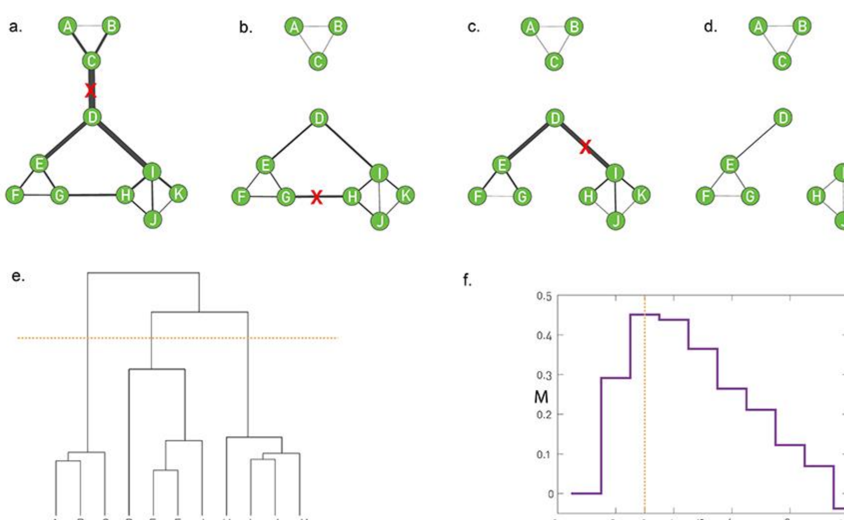


Divisive Procedures: The Girvan-Newman algorithm

- Step 2: Hierarchical Clustering
  - Compute the centrality xij of each link.
  - Remove the link with the largest centrality. In case of a tie, choose one link randomly.
  - Recalculate the centrality of each link for the altered network.
  - Repeat steps 2 and 3 until all links are removed.

31

Divisive Procedures: The Girvan-Newman algorithm



32

Modularity

- How do we decide which of the many partitions predicted by a hierarchical method offers the best community structure?
- Selecting the one for which modularity is maximal.
- Measures the quality of each partition.
- Allows us to decide if a particular community partition is better than some other one.
- Modularity optimization offers a novel approach to community detection.

Modularity…

- Consider the following scenario
- A network with N nodes and L links
- Network is partitioned into n_c number of communities
- Each community has N_c nodes and L_c links
- If L_c is larger than the expected number of links between the N_c nodes given the network's degree sequence, then the nodes of the subgraph C_c could indeed be part of a true community.
- We therefore measure the difference between the network's real wiring diagram (A_{ij}) and the expected number of links between i and j if the network is randomly wired (p_{ij})

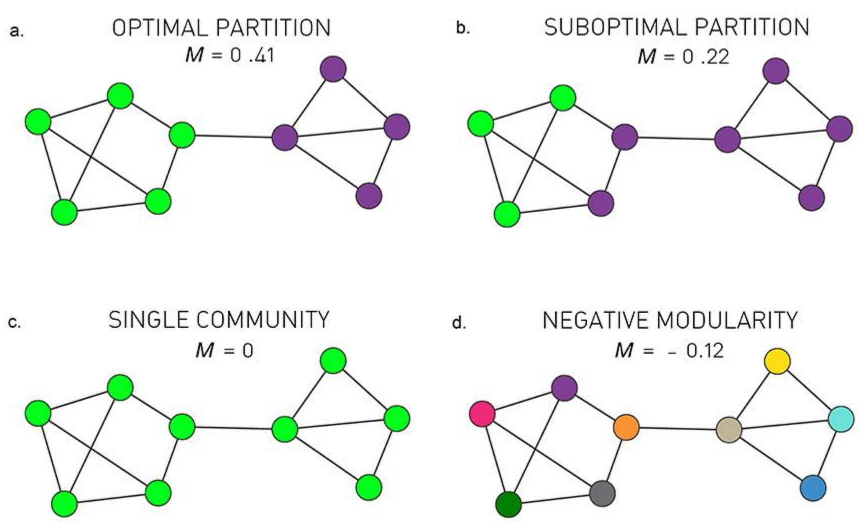
Modularity…

$$M = \sum_{c=1}^{n_c} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L} \right)^2 \right]$$

- L : total number of links in Graph
- n_c : total number of communities in the graph
- L_c : total number of links in community c
- k_c : total degree of nodes in community c

35

Modularity…



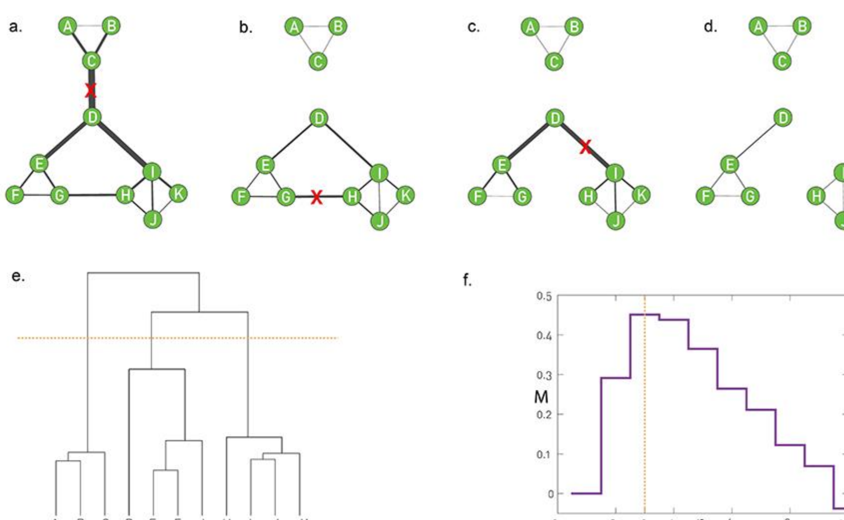
36

Modularity…

- Optimal Partition
  - The partition with maximal modularity M=0.41 closely matches the two distinct communities.
- Suboptimal Partition
  - A partition with a sub-optimal but positive modularity, M=0.22, fails to correctly identify the communities present in the network.
- Single Community
  - If we assign all nodes to the same community we obtain M=0, independent of the network structure.
- Negative Modularity
  - If we assign each node to a different community, modularity is negative, obtaining M=-0.12.

37

Divisive Procedures: The Girvan-Newman algorithm



38

The Louvain Algorithm

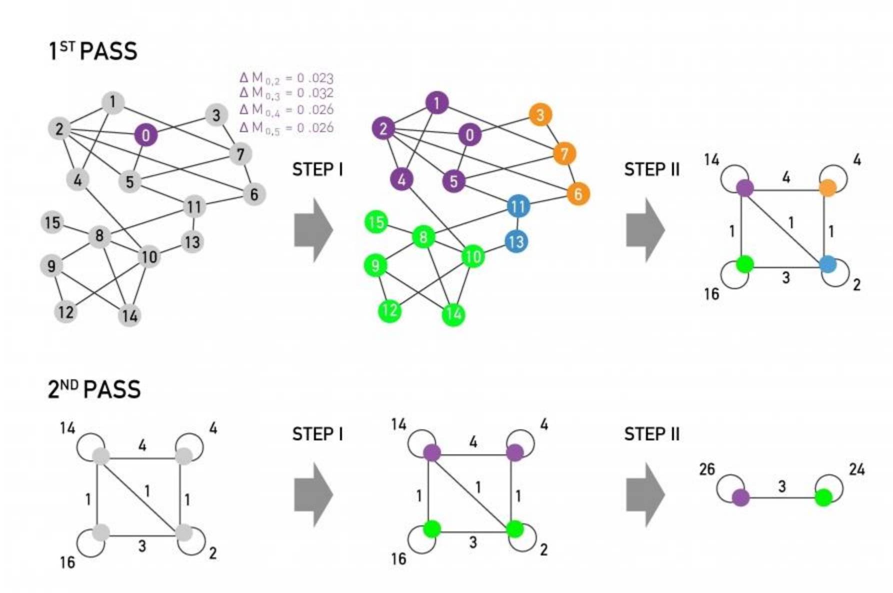
- Greedy algorithm for Community Detection
  - O(nlogn) run time
- Supports weighted graphs
- Provide hierarchical communities
- Widely utilized to study large networks because
  - Fast
  - Rapid Convergence
  - High modularity output (i.e. better communities)

39

The Louvain Algorithm…

- Operates in several iterations
- Ease iteration consists of 2 phases
  - **Phase 1:** Modularity is optimized by allowing only local changes to node-communities memberships
  - **Phase 2:** The identified communities are aggregated into super nodes to build a new network
  - Goto Phase 1 until no increase in modularity is possible

40



41

Community Detection through Modularity Maximization: Limitations

1) Resolution limit:

- well-connected smaller communities tend to get merged with larger communities even if the resultant communities are not that dense
- fails to detect those communities which are well-separated with densely connected intra-community nodes but only a single inter-community edge with the rest of the network

2) Degeneracy of solutions:

- the case when there is an exponential number of community structures with same (maximum) modularity value

42

Permanence and Community Detection

- Modularity is a network-centric global metric
  - Considers the entire network structure during maximization process
  - Not suitable for large and evolving networks
- Requires a method that looks at the local neighborhood while detecting communities
- Chakraborty et al. proposed a metric, named Permanence, which is a local metric for community detection
- A vertex-centric metric
- Two communities A and B are neighbouring communities if $\exists u \in A, v \in B$, and there is an edge between u and v

43

Permanence and Community Detection

- Hypothesis 1:
  - The number of internal connections of node v should be greater than the number of external connections of node v with any external community
- Hypothesis 2:
  - In a community, all the vertices should be highly inter-connected to each other
- Expression for Permanence for a vertex v is:

$$Perm(v) = \left[\frac{I(v)}{E_{max}(v)} \times \frac{1}{\deg(v)}\right] - [1 - c_{in}(v)]$$

- $I(v)$: Number of internal neighbours of v within its own community
- E_{max}: maximum number of connections of v to neighbors in an external community
- c_{in}: internal clustering coefficient of v

44

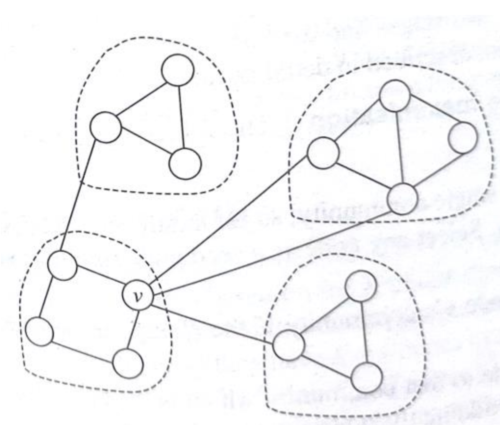
Permanence and Community Detection

- Permanence of the entire network:

$$Perm(G) = \frac{\sum_{v \in V} Perm(v)}{|V|}$$

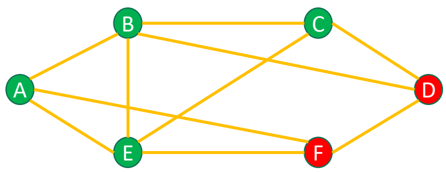
- Permanence value ranges between -1 to 1
  - when vertex v is a part of a clique, Permanence is 1
  - when there is no appropriate community structure of a network (like a grid network), Permanence is 0
  - when $I(v) \ll deg(v)$ and $c_{in}(v) \approx 0$, Permanence tends to -1

Permanence and Community Detection

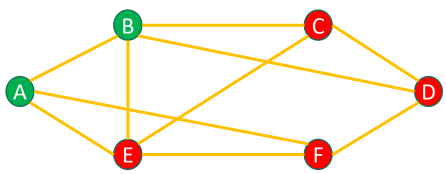


Vertex	$\deg(\cdot)$	$I(\cdot)$	$E_{max}(\cdot)$	$c_{in}(\cdot)$	$\mathrm{Perm}(\cdot)$
v	5	2	2	0	-0.8

Permanence and Community Detection: Illustration



Assignment A



Assignment B

To see how community membership alters permanence scores for vertices C and E

Assignment A

Vertex	$\deg(\cdot)$	$I(\cdot)$	$E_{max}(\cdot)$	$c_{in}(\cdot)$	$\mathrm{Perm}(\cdot)$
C	3	2	1	1	0.67
E	4	3	1	0.67	0.42

Assignment B

Vertex	$\deg(\cdot)$	$I(\cdot)$	$E_{max}(\cdot)$	$c_{in}(\cdot)$	$\mathrm{Perm}(\cdot)$
C	3	2	1	0	-0.33
E	4	2	2	0	-0.75

Therefore, Assignment A is preferable

Permanence Maximization for Community Detection: MaxPerm

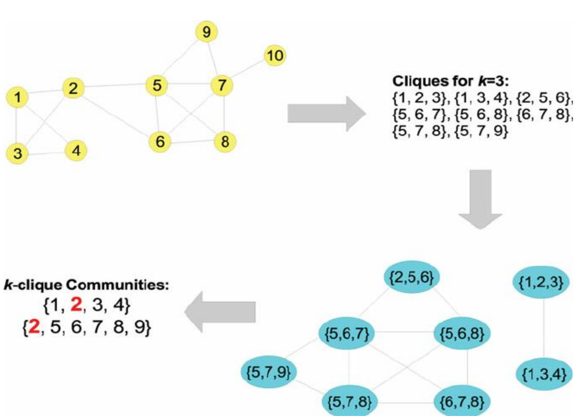
- ❑Uses greedy approach for producing high permanence partitions in the network
- ❑To join the small communities if and only if the permanence value of the network increases
- ❑Basic steps of the algorithm is same as Louvain method
- ❑Two Basic stages of the algorithm
  - ❑ First stage (Permanence maximization):
    - ❑ Merging of small communities greedily
    - ❑ Merging stops when the maximum permanence gain is attained
  - ❑ Second stage (Node aggregation)
    - ❑ Build the super-network whose nodes are the communities that are available in the final network of the first stage
    - ❑ Final nodes of super-network generated are the final communities of the initial network

Permanence Maximization for Community Detection: Limitations

- Permanence maximization reduces the problem of resolution limit and degeneracy of solutions
- If a vertex is connected to more than one neighboring communities and those communities overlap with each other, then Permanence maximization method fails to handle the resolution limit
- For real-world networks, permanence maximization tends to produce small communities

49

Overlapping Community Detection: Clique Percolation



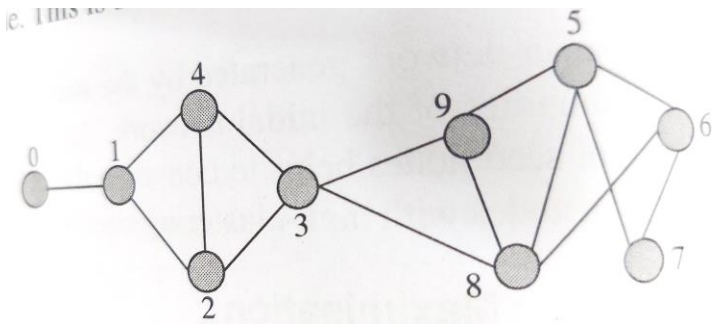
https://bit.ly/37uRQte

- Based on the K-clique (K-clique is the complete subgraph of size K).
- First, it finds all K-cliques present in the network.
- Then, it merges two K-cliques if they have (K-1) nodes in common
- The merging process stops when no more cliques are there to merge

50

Overlapping Community Detection: Clique Percolation

❑Find the communities with clique percolation for the below graph for K=3



Clique Percolation Method: Limitations

❑There is no fixed value of K, and it is not easy to find a correct value of K

❑Finding a clique in a network is computationally expensive.

❑Method is more like pattern matching applied to the network