L21_SNA

- Girvan–Newman algorithm (named after Michelle Girvan and Mark Newman) is a hierarchical method used to detect communities in complex systems
- Top Down approach –Hence divisive in nature

- The Girvan–Newman algorithm detects communities by progressively removing edges from the original network. The connected components of the remaining network are the communities.
- Instead of trying to construct a measure that tells us which edges are the most central to communities, the Girvan–Newman algorithm focuses on edges that are most likely "between" communities.

- Vertex betweenness-how central the nodes are?
    - as the number of shortest paths between pairs of nodes that run through it
    - relevant to models where the network modulates transfer of goods between known start and end points, under the assumption that such transfer seeks the shortest available route.

- Edge betweeness
    - the number of shortest paths between pairs of nodes that run along it
    - If there is more than one shortest path between a pair of nodes, each path is assigned equal weight such that the total weight of all of the paths is equal to unity.
    - If a network contains communities or groups that are only loosely connected by a few inter-group edges, then all shortest paths between different communities must go along one of these few edges.
    - Thus, the edges connecting communities will have high edge betweenness (at least **one** of them).
    - By removing these edges, the groups are separated from one another and so the underlying community structure of the network is revealed.

Girvan_Newmann Algorithm

The general form of their algorithms is as follows:

1 Calculate betweenness score for all edges in the network using any measure.

2 Find the edge with the highest score and remove it from the network.

3 Recalculate betweenness for all remaining edges.

4 Repeat from step 2.

The procedure is continued until a sufficiently small number of communities are obtained, and a hierarchical nesting of the communities is also obtained as a natural by-product.

- The main disadvantage of this approach
- is the high computational cost: simply computing the betweenness for all edges takes $O(|V||E|)$ time, and the entire algorithm requires $O(|V|^3)$ time.

Divisive method: starts with the full graph and breaks it up to find communities.

Too slow for many large networks (unless they are very sparse), and it tends to give relatively poor results for dense networks.

- The method gives us only a succession of splits of the network into smaller and smaller communities, but it gives no indication of which splits are best.
- One way to find the best split is via the modularity concept .

Newman’s greedy optimization of modularity

- What is Modularity?
- greedy agglomerative clustering algorithm for optimizing modularity.
- The basic idea of the algorithm is that at each stage, groups of vertices are successively merged to form larger communities such that the modularity of the resulting division of the network increases after each merge.

- At the start, each node in the network is in its own community, and at each step one chooses the two communities whose merger leads to the biggest increase in the modularity.
- We only need to consider those communities which share at least one

edge, since merging communities which do not share any edges cannot result in an increase in modularity - hence this step takes $O(|E|)$ time.

- Additional data structure which maintains the fraction of shared edges between each pair of communities in the current partition is also maintained, and updating this data structure takes worst-case $O(|V|)$ time

- There are a total of $|V| - 1$ iterations
- (i.e. mergers), hence the algorithm requires $O(|V|^2)$ time.
- Clauset et al improved this by making use of efficient data structures such as max-heaps, with thefinal complexity coming to

$O(|E| d \log |V|)$, where d is the depth of the dendrogram describing the successive partitions found during the execution of the algorithm.