Louvains Algorithm

Trying to maximize Q

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \delta(c_i, c_j) \right]$$

- A_{ij} represents the edge weight between nodes i and j
- k_i and k_j is the sum of weights attached to nodes i and j (Degree of node)
- m is the sum of all edge weights in the graph
- c_i and c_j are the respective communities of i and j
- $\delta(c_i, c_j) = \begin{cases} 1 & \text{if } c_i = c_j \\ 0 & \text{if } c_i \neq c_j \end{cases}$ (Are nodes i and j in the same community)

Phase 1

- Each node in the network is assigned to its own community
- For each node, the change in modularity is found for removing i from its current community, and then moving it into the community of each neighbor j of i
- Node i is then moved to the community that resulted in the greatest increase of modularity
  - If no increase is possible, i remains in its original community
- After all nodes have been traversed, we start again at the first node, and repeat this process until no change in modularity is possible

$$\Delta Q = \left[\frac{\sum_{in} + 2k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

- $\sum_{in}$ is the sum of all weights of the edges inside the community i is moving to
- $\sum tot$ is the sum of all weights of the edges to nodes in the community i is moving to
- $k_{i,in}$ is the sum of the weights of edges between i and other nodes in the community i is moving to

Phase 2

- Nodes in the same community are grouped into a single node of a new graph
  - Edges between nodes of the same community are now represented as self loops
  - Edges from multiple nodes in the same community to a node in a different community are represented as weighted edges between the two communities
- Phase 1 is now ran with this new smaller graph
- This is repeated until no gain in modularity is possible

Output (Football data)