Clustering Graphs based on Net Reactivity: An intuitive approach

Author: Morad A.

Date: June 2023

Introduction

There exists a lot of clustering algorithms, such as MCL, k-means, and DBSCAN to name a few. This document takes an intuitive approach, supported by minimal assumptions for our use-case, and mathematical formulation, to reach a satisfiable result.

Firstly, more information is proportional to easier algorithms. For example, undirected unweighted graphs are harder to cluster ‘at least theoretically’ than directed weighted graphs, because we have more information contained in the graph that we can use to reach a solution. In PPI networks’ case, it is normal to see weighted ‘and sometimes directed’ graphs. We will assume directed weighted graphs for the sake of simplicity, and that we can achieve satisfiable results anyway.

Secondly, there exists a theoretical complexity threshold for any clustered graph. That threshold is the most optimal clustering solution for the graph, and we cannot go beyond it. This threshold might not be satisfiable enough to give modular-enough clusters, which is a bummer. However, reaching that threshold is our goal.

Thirdly, we will discuss intuition, formalize it, then optimize our algorithm for the edge case handling. This will be a lot easier ‘and more effective’ than designing the optimal algorithm and landing on the optimal solution pinpoint. This will allow us to develop a good mathematical basis as we go.

Fourthly, we do not have any reliable method to check the correctness of any clustering algorithm quantitatively. Although we will discuss high connectivity and other topics, these are not the correct way to evaluate algorithms. Instead, we will look at many cases and see if our human manual clustering process matches the algorithm.

The assumptions are:

1. Any element in the graph can join multiple clusters.
2. There can be unclustered elements in the graph.
3. We won’t take into account the bidirectional connections ‘aka. Self loops’.
4. A graph is weighted and directed.

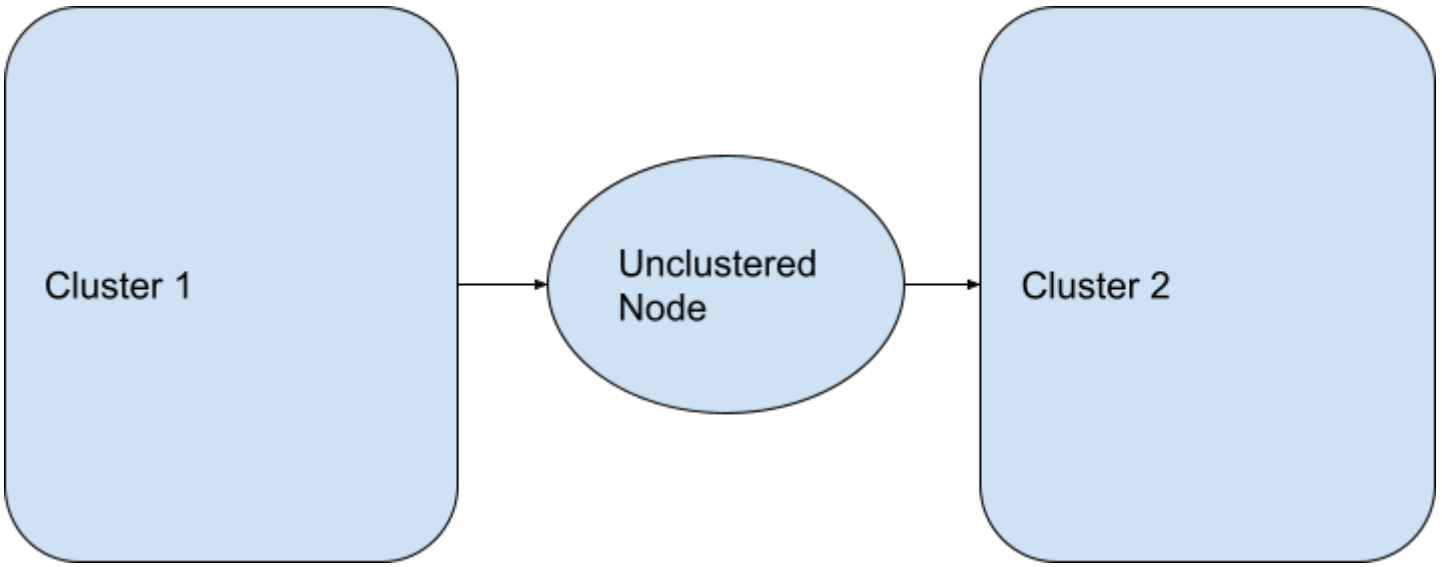
We shall begin.

Intuition, and general definitions

A graph can be called a ‘network’, nodes can be referred to as ‘elements’, and edges can be called ‘connections’.

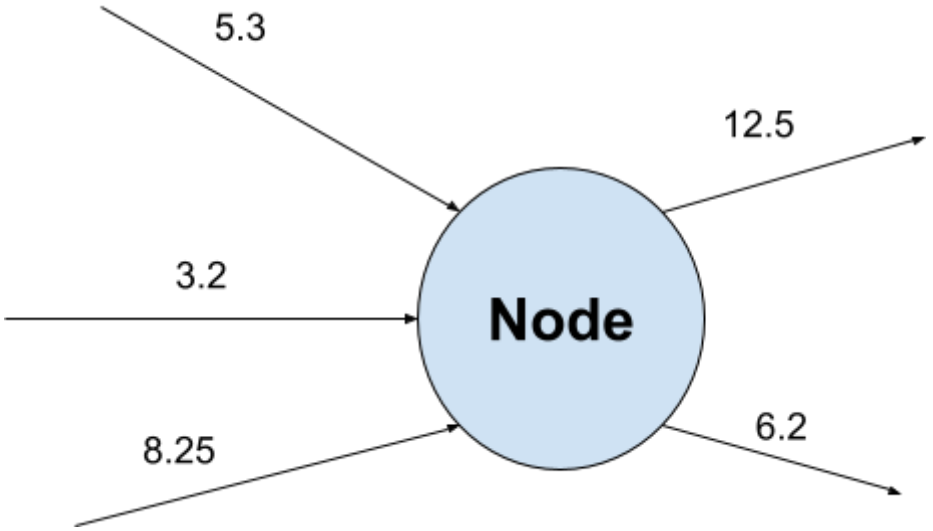
The clustering algorithm works on nodes, not edges. This is important to note, since we will classify nodes based on a property that we will construct later. Also, we will focus on weights and directions rather than general connectivity based on edge number for some nodes, as this will be more logical for the PPI Networks case.

Let’s look at node ‘N’ with some inbound edges and outbound edges. How can we classify N such that it can be clustered or nonclustered? It would be intuitive to say that node N that contributes or takes by sum of the edges values over the network must be clustered, or that if node N takes many edge values and transmits many edge values with little or no difference it must be nonclustered.



As we can see, the unclustered node connects between highly-connected clusters. Therefore, we can say that the unclustered node is hard to classify based on the directions and edge values the node has. We can directly conclude that the unclustered node has almost indifferent inbound and outbound edges. Put formally, we can say that $R = \sum OEV - \sum IEV$, where ‘R’ is the reactivity coefficient for the node N, ‘OEV’ is the sum of all outbound edge values, and ‘IEV’ is the sum of all inbound edge values. For example,

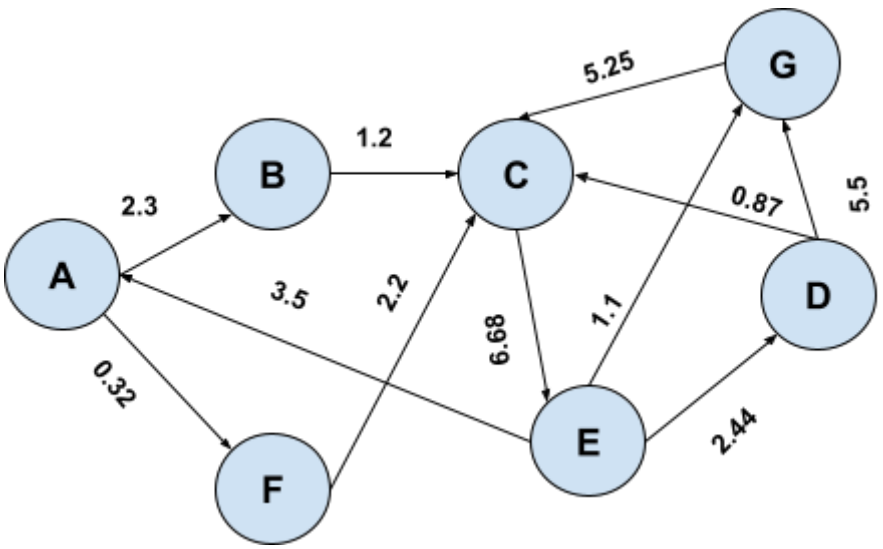
The Reactivity of the node shown is



$R = (12.5 + 6.2) - (5.3 + 3.2 + 8.25) = 18.7 - 16.75 = 1.95.$

As you can notice, the Reactivity coefficient does not inform us about the number of Inbound edges, or Outbound edges. Instead, it gives us a scalar value about the net reactivity of the node. In this case, R=1.95 means that the node adds 1.95 of value to the network as a whole. For complex representation, we can use 3-tuples for node representation. For example, the previous node tuple is $< O, I, R > = < 2, 3, 1.95 >$. The first value is the number of outbound edges, the second value is the number of inbound edges, and the third value is the reactivity coefficient. Although this will be an accurate way to classify nodes, the rest of the paper will go towards classification by reactivity coefficient only for the sake of simplicity.

Based on what we have established, we can say that as the reactivity coefficient goes to zero, the more probable it will be unclustered. We can construct a nonclustered network to demonstrate some properties. The next sections will formalize the representation and clustering algorithm.



Statistical Analysis for Satisfiability, and node properties

We will begin to construct a table of the Nodes' properties.

Node Name	Node Indegree	Node Outdegree	Node Reactivity
A	1	2	0.32
B	1	1	-1.1
C	4	1	-2.84
D	1	2	3.93
E	1	3	0.36
F	1	1	1.88
G	2	1	-1.35

In order to measure the effectivity of our algorithm beforehand, we can measure the coefficient of variation for all the nodes' reactivity coefficients, that is, dividing the standard deviation of the nodes' reactivities by the arithmetic mean. The higher the coefficient of variation, the higher the validity and satisfiability of the clusterization output. For example, the standard deviation in our example is nearly 2.073771, and the arithmetic mean is nearly 0.171428. Therefore, the coefficient of variation is 12.097. Therefore, the output might be satisfactory enough. Note that a satisfactory coefficient of variation is generally an integer about 5, but this is an arbitrary threshold.

Let's exclude two nodes, and make two clusters. Note that the numbers are totally subjective and can be set based on preference. In our case, the cutoff percentage for unclustered nodes is $2/7 \simeq 28.57\%$, same for the cluster cutoff percentage.

The two excluded nodes are nodes whose reactivity coefficients are closest to zero. In this case, they're nodes A,E. Furthermore, every cluster must have a starting point, hereinafter called a cluster head. They are chosen based on the highest reactivity coefficient, which are nodes D, F. The cluster is constructed by taking each cluster head and constructing the cluster from the outdegree nodes, excluding unclustered nodes.

We will execute our algorithm and visualize it in the next section.

Algorithm Execution and Formalization

Let's construct the D cluster. Node 'D' has 'C,G' as outbound connections, we will cluster these three nodes.

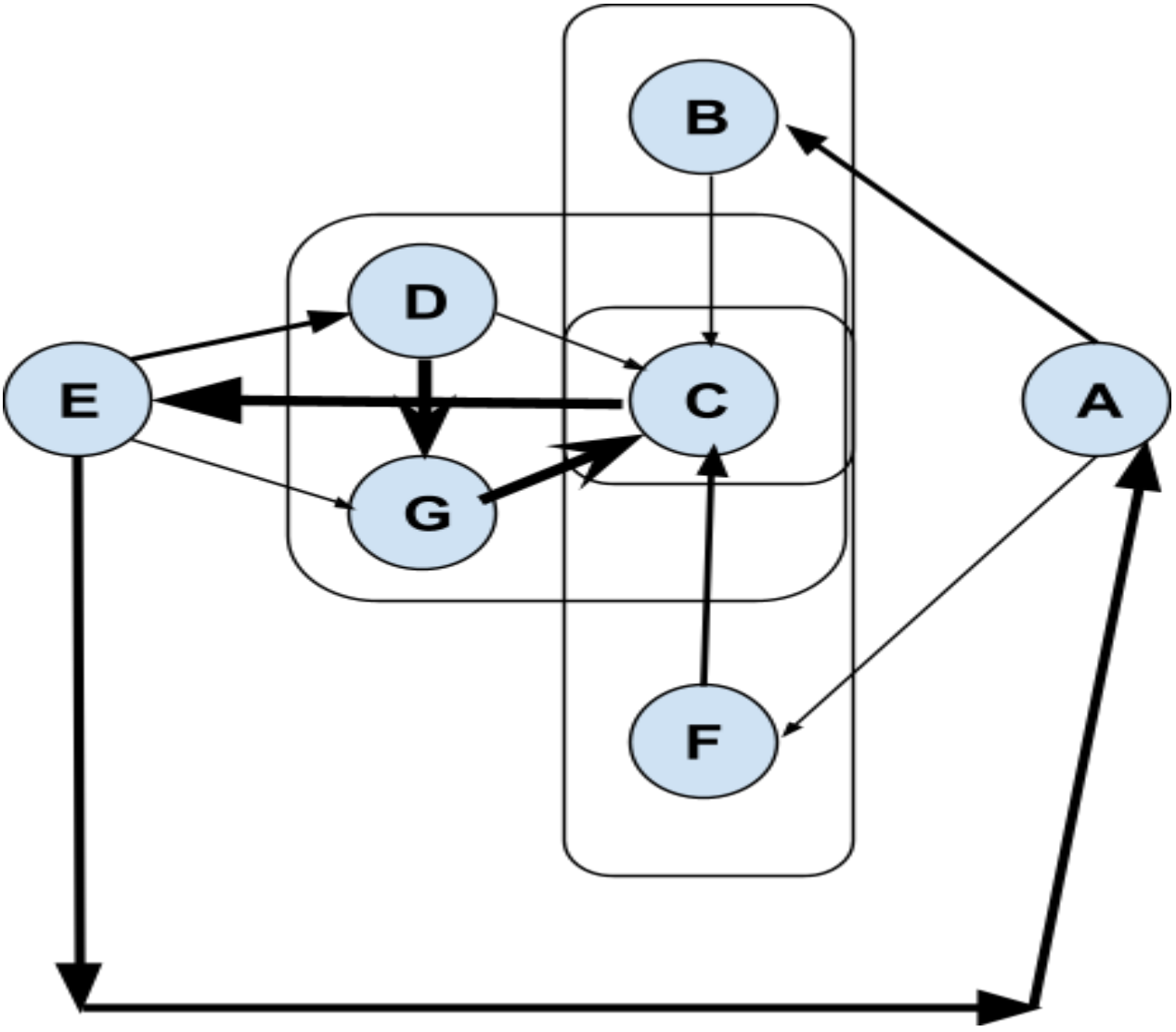
For the F cluster. Node 'F' has 'C' as an outbound connection, we will cluster these two nodes.

Now that we have nodes D,F,C,G are clustered and A,E are unclustered, we have one node we did not consider 'B', we will make it as a cluster head and will cluster 'C' with it.

We have 3 clusters now,

1. $C_F = \{C\}$
2. $C_D = \{C, G\}$
3. $C_B = \{C\}$

Recall that A, E are unclustered nodes, let's visualize what we have done.



Different arrow sizes represent edge values.