IV: Section (vii): Community Detection in Complex Networks: "Betweenness" Centrality & The Girvan-Newman Algorithm.

In the study of graphs & networks it is commonly the case that we would like to know of a cogent decomposition or partition of the network into smaller, loosely compatible subsets - viz., communities. In the fields of data science and machine learning this task is analogously referred to as clustering.

Returning to our previous example, if our network represents a framework for research collaborations between faculty at a university, clusters or communities might be approximated by departments or formal research groups.

We begin with an additional definition of centrality.

Betweenness Centrality

Unlike our previous measures of centrality, betweenness has very little to do, ostensibly, with how well-connected (i.e. the degree of a vertex or its neighbors) a given vertex is. Instead, is best understood in relation to the (global) flow of a network.

~~[illegible]~~ Let's imagine, say, in the field of epidemiology

(The study of infectious diseases & their distribution/incidence), That we use a graph to represent neighboring cities in order to monitor the spread of a deadly "zombie bug." In this case, we might be concerned (quite literally) in the instance of vagabond zombies travelling from town to town. Perhaps, then, if we could determine which town receive the largest amount of zombie foot-traffic, on average, we could quarantine this town (or "delete" this vertex - say), potentially saving thousands of lives in the process.

Zombie references aside, this notion of centrality - put formally: The number proportional * of geodesic paths which contain a given vertex is called $\boxed{\text{betweenness}}$.

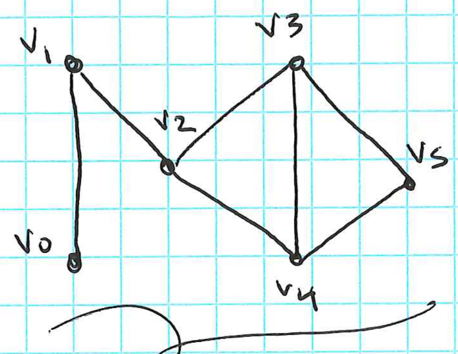
Betweenness Centrality

The betweenness centrality of a vertex $v \in V(G)$ is given by the expression:

$$\boxed{g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}}$$

where $\boxed{\sigma_{st}}$ is the total number of shortest paths from vertex s to t, and $\boxed{\sigma_{st}(v)}$ is the number of such paths passing through the specified vertex v.

$\boxed{\text{Ex.}}$ We compute the betweenness centrality of the vertices in the given graph.



G

Note that $\boxed{g(v_0)=0}$, since $\sigma_{st}(v_0)=0$ for all $s \neq v_0 \neq t$.

Next we calculate $\boxed{g(v_1)}$. Here we consider all shortest paths in the graph between vertices (other than v_1) that contain v_1, divided by the number of shortest paths total between these vertices; we then sum over all pairs.

shortest path from $\boxed{v_0 - v_2}$ (there is one & it contains v_1) $\to \boxed{\frac{1}{1}}$

" " $\boxed{v_0 - v_3} \to \boxed{\frac{1}{1}}$, shortest path: $\boxed{v_0 - v_4} \to \boxed{\frac{1}{1}}$

and, finally, $\boxed{v_0 - v_5} \to \left(\frac{2}{2}\right) = \boxed{1}$, since there are two such shortest paths, but each contains v_2. No other shortest paths b/w verts pass through v_2.

This yields: $g(v_1) = 1 + 1 + 1 + 1 = \boxed{4}$

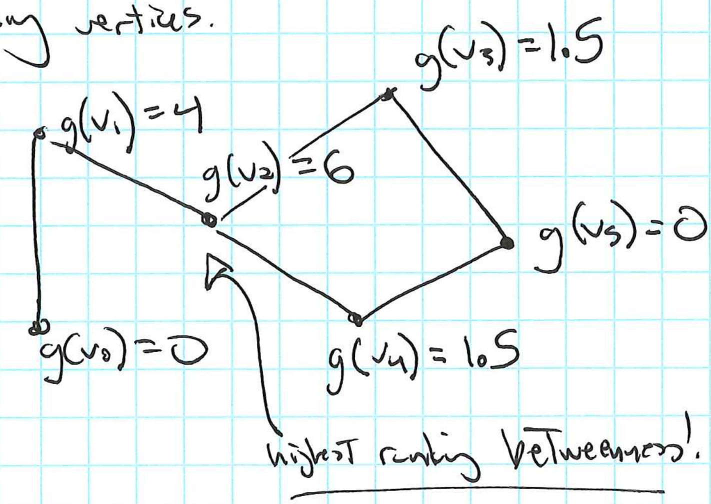
$\boxed{g(v_2)}$: $\underline{v_1 - v_3} \to \boxed{\frac{1}{1}}$, $\underline{v_1 - v_5} \to \boxed{\frac{2}{2}}$, $\underline{v_1 - v_4} \to \boxed{\frac{1}{1}}$

$\underline{v_0 - v_1} \to \boxed{\frac{0}{1}}$, $\underline{v_0 - v_3} \to \boxed{\frac{1}{1}}$, $\underline{v_0 - v_4} \to \boxed{\frac{1}{1}}$, $\underline{v_0 - v_5} \to \boxed{\frac{2}{2}}$

Note that no other shortest paths pass through v_2.

Hence, $g(v_2) = 1 + 1 + 1 + 1 + 1 + 1 = \boxed{6}$

Continuing in this way, we compute the betweenness centrality for the remaining vertices.



Community Detection & The Girvan-Newman Algorithm.

The Girvan-Newman Algorithm is a simple algorithm that begins with a graph G and iteratively "prunes" edges with the largest betweenness centrality (note that edge betweenness is determined comparably to vertex betweenness). The result at each stage of the algorithm is a successively refined community partition of the vertices of G. Note that this type of hierarchical partitioning/clustering is sometimes known as a $\boxed{\text{dendrogram}}$.

Girvan-Newman Algorithm Pseudo-code in 3 steps.

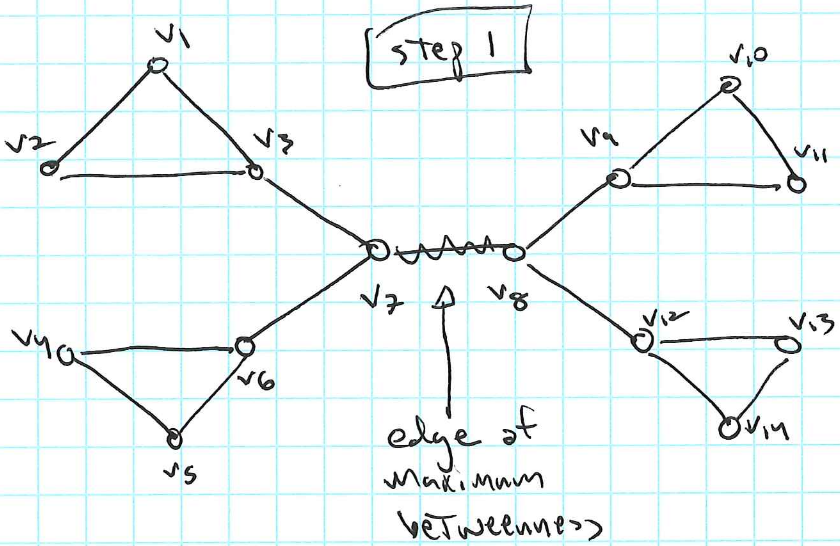
1. Remove the edge(s) of highest betweenness in the graph. If this edge(s) is a cut-edge, then the new components represent a partitioning of the graph.

2. Next re-calculate betweenness for the new, reduced graph. Remove, once again, the edge(s) of highest betweenness. As before, if this edge(s) is a cut-edge, the new, smaller components represent a fine-tuned partition.

3. Repeat this process until the stopping condition, or until the graph is totally disconnected.

Ex.

Step 1

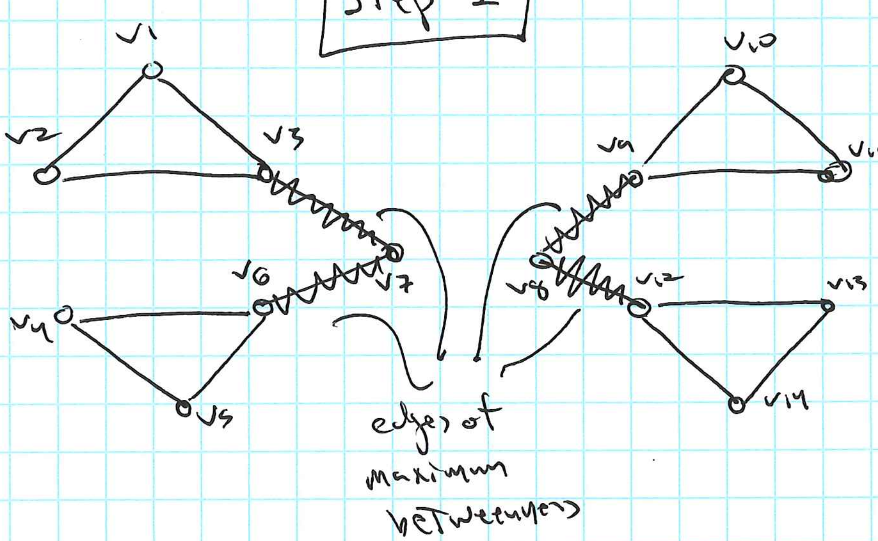


2 communities:

$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$,

$\{v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}\}$.

Step 2



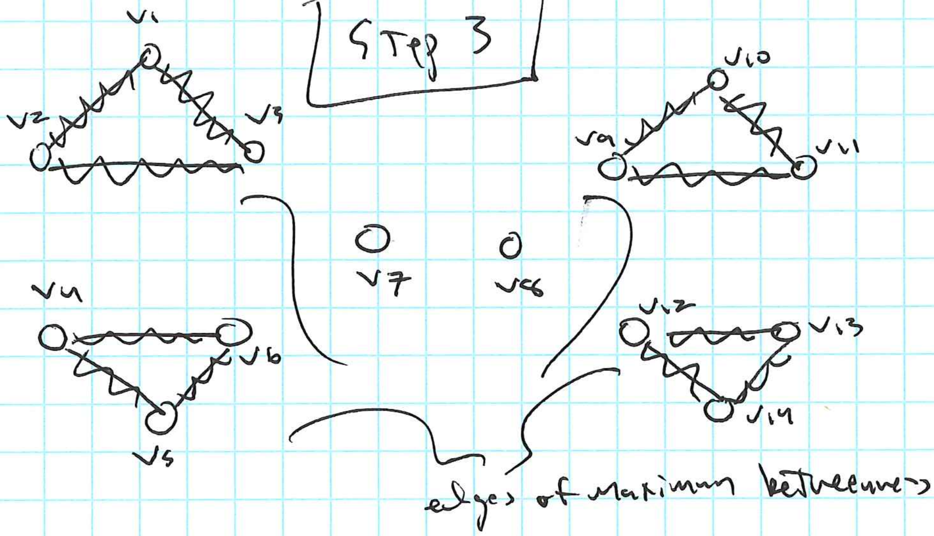
6 communities:

$\{v_1, v_2, v_3\}$, $\{v_4, v_5, v_6\}$,

$\{v_9, v_{10}, v_{11}\}$, $\{v_{12}, v_{13}, v_{14}\}$,

$\{v_7\}$, $\{v_8\}$.

Step 3



(14) Communities : $\{v_1\}, \{v_2\}, \{v_3\}, \ldots, \{v_{12}\}, \{v_{13}\}, \{v_{14}\}$.

all isolated vertices