

HOMEWORK 1 SOLUTIONS

Brendan Meeder¹

1 Signed Graphs

Part 1

The essence of our solution is that if there were two different clusterings, we could find vertices u, v such that in one clustering u is in the same cluster as v , and in the other clustering they are in different clusters. What is the sign of the edge $\{u, v\}$? According to one clustering it's positive, while according to the other it's negative — this is clearly a contradiction.

How do we know that we can find such vertices u and v ? Suppose that G is a complete signed graph and let $C_1 = \{A_1, \dots, A_m\}$ and $C_2 = \{B_1, \dots, B_n\}$ be two different clusterings (two different partitionings of the vertex set). We know that some $A_i \in C_1$ is different from all of the B_j , otherwise the two partitionings would be the same. Without loss of generality, assume that A_1 is distinct from all of the B_j . Since $\{B_j\}_j$ is a partitioning of $V(G)$, we know that every vertex of A_1 is in exactly one of the B_j . Either A_1 contains some $u \in B_i$ and $v \in B_j \neq B_i$, or A_1 is properly contained in some cluster B_i . In the first case, we can use those u and v directly— they are in the same cluster according to C_1 and are in different clusters in C_2 . Otherwise, as $A_1 \subsetneq B_i$, we can choose $u \in A_1$ and $v \in B_i \setminus A_1$.

Part 2

Lemma: If u is connected to v via a path of positive edges, then in any clustering of G , u is in the same cluster as v . This is easy to prove with contradiction: Suppose C is a clustering in which u and v are in different clusters. Walking from u to v along the positive path, there is some first node w not in the same cluster containing u . The edge that we just took to get to w has endpoints in different clusters and should therefore have a negative sign. This is our contradiction to the assumption of a positive edge path.

The insight of the lemma is that it is necessary to look at positive-edge connected components. Our algorithm is simply to find the connected components of the graph after all negative edges are removed; the clusters will simply be the resulting connected components. We do some $O(|E|)$ preprocessing to remove all of the negative edges, and finding connected components can be done easily using a breadth-first search. If you want to be fancy, you can use a method such as union-find. We also check each negative edge and make sure that its endpoints are in different clusters. If we find a negative edge with endpoints in the same cluster, we output “NOT-CLUSTERABLE”.

Notice that the correctness of the algorithm follows from the lemma. If the algorithm outputs a clustering the check makes sure that it is a valid clustering. Suppose that our algorithm says “NOT-CLUSTERABLE” due to some edge $\{u, v\}$. Then any supposedly valid clustering C' would have to have u and v in separate clusters; however, because we have taken connected components with respect to positive edges, there is a positive $u - v$ path in G . By the lemma, C' cannot be a valid clustering.

¹**Collaboration notice:** I did not collaborate with anyone on this homework.

Part 3

We prove that ‘ G contains no cycle with exactly one negative edge $\Rightarrow G$ is clusterable’ by the contrapositive. Suppose that the algorithm outputs “NOT-CLUSTERABLE” because some edge $e = \{u, v\}$ is negative and both u and v are in the same cluster C . As the algorithm finds connected components with respect to positive edges, there is a path P of positive edges with endpoints u and v . If we add e to this path, we get a cycle with exactly one negative edge.

We prove that ‘ G is clusterable $\Rightarrow G$ contains no cycle with exactly one negative edge’ as follows: Suppose the algorithm returns a clustering $\{C_1, \dots, C_k\}$ and doesn’t output “NOT-CLUSTERABLE”. Let $C = (v_1, e_1, v_2, e_2, \dots, v_k, e_k, v_1)$ be an arbitrary cycle in G . If the vertices of C all belong to one of the clusters, C_i , then we know that all of the edges in the cycle are positive. Otherwise, at least two clusters contain vertices in C . Walking along the cycle starting at v_1 , we must change the cluster we are currently in at least twice (once when we leave the cluster containing v_1 , and once when we reenter the cluster containing v_1). Thus, the cycle contains at least two negative edges with endpoints in different clusters. These cases are exhaustive, and in each case, the cycle C doesn’t contain exactly one negative edge.

2 Normalized Betweenness Centrality

Part 1

The star graph is the only graph which contains a vertex with NBC of 1. Let G be any graph that contains a vertex w with NBC of 1. First, we will show that every vertex is incident to w , and then show that any two vertices distinct from w are not connected to each other. Let $u \neq w$ be an arbitrary vertex and suppose for a contradiction that u isn’t adjacent to w . Then the shortest path from u to w contains at least one intermediate node v that is adjacent to u . But in this case, $short_{uv}(w) = 0$, so the NBC of w cannot be 1. Contradiction.

Now that we have shown every vertex is adjacent to w , the same argument can be used to show that any $u, v \neq w$ cannot be adjacent. If there were two vertices u, v that were adjacent, then $short_{uv}(w) = 0$ and the NBC of w would be less than 1. Thus, the characterization of G is exactly that of a star graph: every vertex is connected to a central vertex w , and none of the other nodes are connected to each other.

Part 2

We recall that in a tree there is exactly one path between every pair of nodes; therefore, there is a unique shortest path between any pair of nodes. If the path from u to v contains a node w , then rooting T at w will result in u and v being in different subtrees. Finally, we recall that a perfect binary tree of height k has $2^{k+1} - 1$ vertices in it. We let $M = \binom{2^{n+1}-2}{2}$ be the normalization factor for the NBC calculation of vertices in T_n . In the case of the root node, there are $2^n - 1$ many nodes in each subtree, so the normalized betweenness centrality is

$$C_{NB}(v_{root}) = (2^n - 1)^2 / M.$$

The leaf nodes (at depth n) have NBC of 0, and we’ve already already calculated the NBC of the root node, so let v be at depth $0 < k < n$ in the tree. v has two subtrees of height $n - k - 1$ containing $2^{n-k} - 1$ vertices each. There are $2^{n+1} - 1 - (2^{n-k+1} - 1)$ many nodes that aren’t below v in T_n . Rooting T_n at v we see that there are three subtrees, two with $2^{n-k} - 1$ vertices and one with $2^{n+1} - 2^{n-k+1}$ many nodes. If a and b are in different subtrees (with respect to the rooting at

v), the term $short_{ab}(v)/short_{ab}$ will be 1 in the summation, otherwise it will be 0. Therefore, we get that

$$C_{NB}(v) = \frac{2(2^{n-k} - 1)(2^{n+1} - 2^{n-k+1}) + (2^{n-k} - 1)^2}{M}$$

Part 3

Again, we use the property that in any tree there is exactly one path between every pair of vertices. Let T be an arbitrary tree rooted at v with subtrees T_1, \dots, T_k . If $a, b \in T$ are in the same subtree, $short_{ab}(v) = 0$. Otherwise, a and b are in different subtrees and the unique (shortest) path between them passes through v . Thus $short_{ab}(v)/short_{ab} = 1$. Considering the definition of betweenness centrality of v :

$$C_B(v) = \sum_{a < b, a, b \neq v} \frac{short_{ab}(v)}{short_{ab}},$$

we see that the only nonzero terms are those for which a and b are in different subtrees, in which case the corresponding term in the summation is 1. The total number of such (a, b) pairs is $\sum_{1 \leq i < j \leq k} |T_i||T_j|$, and we get that

$$C_B(v) = \sum_{1 \leq i < j \leq k} |T_i||T_j|.$$

3 Clustering Coefficient and NBC

Let $T(v)$ be the set of unordered pairs $\{i, j\}$ such that $i, j \neq v$ and the edges $\{i, v\}, \{j, v\}, \{i, j\}$ all exist. These are simply the pairs over vertices which, when combined with v , induce a triangle in the graph. The key insight for this problem is that if two nodes i and j are in the neighborhood of v and have an edge between them (that is, $\{i, j\} \in T(v)$), then the term

$$short_{ij}(v)/short_{ij}$$

in the calculation of the betweenness centrality will be zero. As every term $short_{ab}(v)/short_{ab}$ in the NBC definition is at most 1, the NBC is maximizes by assuming all of the other $\binom{n-1}{2} - |T(v)|$ terms are 1. For brevity, we will write S instead of $short$. We use the definition of clustering coefficient of v to calculate $|T(v)|$:

$$CC(v) = \frac{|T(v)|}{\binom{deg(v)}{2}} \Rightarrow |T(v)| = CC(v) \binom{deg(v)}{2}.$$

Therefore,

$$\begin{aligned} C_B(v) &= \sum_{i, j \neq v} \frac{S_{ij}(v)}{S_{ij}} = \sum_{\{i, j\} \in T(v)} \frac{S_{ij}(v)}{S_{ij}} + \sum_{\{i, j\} \notin T(v)} \frac{S_{ij}(v)}{S_{ij}} \\ &\leq \sum_{\{i, j\} \in T(v)} 0 + \sum_{\{i, j\} \notin T(v)} 1 \\ &\leq 0 + \left(\binom{n-1}{2} - |T(v)| \right) \cdot 1 \\ &= \binom{n-1}{2} - CC(V) \binom{deg(v)}{2}. \end{aligned}$$

Normalizing by $\binom{n-1}{2}$ and expanding the binomial coefficients, we get the desired result.

Some example for which the bound is tight include the star graph and the 3-cycle triangle. For a ‘nontrivial’ solution (one in which the $C_{NB}(v) \notin \{0, 1\}$) we can take a star graph with at least three leaf nodes and connect two of the leaf nodes.

4 Gephi

Property	Karate	Political Books
Highest degree node (ID + Label)	ID 34, label 34	A National Party No More (8), Off With Their Heads (12)
Degree of highest deg. nodes	17	25
Average node degree	4.588	8.4
Most common node degree	2 (appears 11 times)	5 (appears 22 times)
Number of triangles	45	560
Average CC	0.571	0.488

Table 1: Information for parts one and two.

Node ID	Closeness Centrality	Betweenness Centrality
1	0.569	0.438
3	0.559	0.144
34	0.55	0.304
32	0.541	0.138

Table 2: Closeness and Betweenness centrality statistics for the Karate Network

Node ID	Closeness Centrality	Betweenness Centrality
30	0.414	0.139
58	0.413	0.075
7	0.408	0.069
49	0.408	0.104

Table 3: Closeness and Betweenness centrality statistics for the Political Books Network

Trends

We see that the number of triangles appears to be growing linearly with the degree of the node. However, the number of potential triangles grows quadratically with the degree (in particular, there are $\binom{deg(v)}{2}$ potential triangles). Thus, we would expect in any model where each edge has some positive probability of being present that the number of triangles in which a node participates increases quadratically with its degree.