CNM Community Detection for General Networks

Edward L. Platt
ed@elplatt.com

October 4, 2018

The Clauset-Newman-Moore (CNM) community detection algorithm [1] greedily maximizes the *modularity* Q by repeatedly merging smaller communities. The orginal presentation is given for an unweighted, undirected network with no self-loops. This paper derives analogoous equations for weighted networks with self-loops, both directed and undirected.

Throughout this paper, A_{vw} refers to an entry in the adjacency matrix of a network. Each *edge* connects two *vertices*, which can be the same vertex for both edges (in the case of a self-loop). A *stub* is one half of an edge, and is associated with single vertex. Each edge has a *weight*, with m refering to the total edge weight. In an *undirected* network, edges are symmetric with respect to vertices. In a *directed* network, the edge is outgoing from the *source* vertex and incident upon the *target* vertex. Respectively, the degree k_v, in-degree $k_v^{(in)}$, and out-degree $k_v^{(out)}$ refer to the sum of adjacent edge weights, sum of incident edge weights, and sum of outgoing edge weights. The term δ_{ij} is equal to 1 if $i = j$ and 0 otherwise.

The CNM algorithm takes the adjacency matrix A as input. Throughout the algorithm, a sparse matrix of modularity changes Δ and an array of stub fractions a are maintained. The output can be either a partition of vertices corresponding to maximum modularity or a hierarchical clustering linkage matrix.

1 Undirected

In an undirected network, modularity is defined as:

$$Q = \frac{1}{2m} \sum_{vw} \left[A_{vw}(1 + \delta_{vw}) - \frac{k_v k_w}{2m} \right] \delta_{c_v c_w} \tag{1}$$

$$m = \frac{1}{2} \sum_{vw} A_{vw}(1 + \delta_{vw}), \tag{2}$$

where m is the total weight of edges is the network and c_v is the community containing vertex v. Equation 25 differs from [1] by the inclusion of a δ_{vw} term.

This term is necessary when self-loops are present, because unlike off-diagonal entries, each diagonal entry represents two stubs.

The degree of vertex v is given by:

$$k_v = \sum_w A_{vw}(1+\delta_{vw}). \tag{3}$$

The fraction of edge weight between communities i and j is given by:

$$e_{ij} = \frac{1}{2m}\sum_{vw} A_{vw}(1+\delta_{vw})\delta_{c_v i}\delta_{c_w j}. \tag{4}$$

The fraction of stub weight within community i is given by:

$$\begin{aligned} a_i &= \frac{1}{2m}\sum_{vw} A_{vw}(1+\delta_{vw})\delta_{c_v i} & (5)\\ &= \frac{1}{2m}\sum_v k_v \delta_{c_v i}. & (6)\end{aligned}$$

Writing the modularity in terms of e_{ij} and a_i by using $\delta_{c_v c_w} = \sum_i \delta_{c_v i}\delta c_w i$:

$$\begin{aligned} Q &= \frac{1}{2m}\sum_{ivw}\left[A_{vw}(1+\delta_{vw}) - \frac{k_v k_w}{2m}\right]\delta_{c_v i}\delta_{c_w i} & (7)\\ &= \sum_i \frac{1}{2m}\sum_{vw} A_{vw}(1+\delta_{vw})\delta_{c_v i}\delta_{c_w i} \\ &\quad -\sum_i \frac{1}{(2m)^2}\sum_v k_v \sum_w k_w \delta_{c_v i}\delta_{c_w i} & (8)\\ &= \sum_i \frac{1}{2m}\sum_{vw} A_{vw}(1+\delta_{vw})\delta_{c_v i}\delta_{c_w i} \\ &\quad -\sum_i \left(\frac{1}{2m}\sum_v k_v \delta_{c_v i}\right)\left(\frac{1}{2m}\sum_w k_w \delta_{c_w i}\right) & (9)\\ &= \sum_i \left(e_{ii} - a_i^2\right). & (10)\end{aligned}$$

When community i is merged into community j, the edge fractions and stub fractions are updated according to:

$$\begin{aligned} e'_{ii} &= 0 & (11)\\ e'_{jj} &= e_{ii} + e_{jj} + e_{ij} & (12)\\ e'_{jk} &= e_{ik} + e_{jk} \text{ (if } j \neq k) & (13)\\ a'_i &= 0 & (14)\\ a'_j &= a_i + a_j. & (15)\end{aligned}$$

The change in modularity for merging communities i and j are given by:

$$\Delta_{ij} = \left(e'_{jj} - a'^2_j\right) - \left(e_{ii} + e_{jj} - a_i^2 - a_j^2\right) \tag{16}$$
$$= e_{ij} - 2a_i a_j. \tag{17}$$

Initially, each vertex is in its own community ($c_v = v$) and the values of Δ_{ij} and a_i are:

$$\Delta_{ij}^{(0)} = \frac{1}{2m}(A_{ij} + A_{ji})(1 + \delta_{ij}) - 2\frac{k_i}{2m}\frac{k_j}{2m} \tag{18}$$
$$= \frac{1}{m}\left[A_{ij}(1 + \delta_{ij}) - \frac{k_i k_j}{2m}\right] \tag{19}$$
$$a_i^{(0)} = \frac{k_i}{2m}. \tag{20}$$

When community i is merged into j, the values of Δ are updated as follows:

$$\Delta'_{jk} = e'_{jk} - 2a'_j a'_k \tag{21}$$
$$= e_{ik} + e_{jk} - 2a_k(a_i + a_j) \tag{22}$$
$$= e_{ik} - 2a_i a_k + e_{jk} - 2a_j a_k \tag{23}$$
$$= \Delta_{ik} + \Delta_{jk}. \tag{24}$$

Note that if there is no edge between i and k, the value of Δ_{ik} can be determined directly from a_i and a_k (the same applies to Δ_{jk}).

2 Directed

In a directed network, A_{vw} is the weight of the edge from v to w. The modularity is defined as:

$$Q = \frac{1}{m}\sum_{vw}\left[A_{vw} - \frac{k_v^{out} k_w^{in}}{m}\right]\delta_{c_v c_w} \tag{25}$$
$$m = \sum_{vw} A_{vw}, \tag{26}$$

where m is the total weight of edges is the network and c_v is the community containing vertex v.

The in and out degrees of vertex v are given by:

$$k_v^{in} = \sum_i A_{iv} \tag{27}$$
$$k_v^{out} = \sum_i A_{vi}. \tag{28}$$

The fraction of edge weight from community i to community j is given by:

$$e_{ij} = \frac{1}{m}\sum_{vw} A_{vw}\delta_{c_v i}\delta_{c_w j}. \tag{29}$$

The fraction of source weight within community i is given by:

$$a_i = \frac{1}{m}\sum_{vw} A_{vw}\delta_{c_v i} \quad (30)$$

$$= \frac{1}{m}\sum_{v} k_v^{out}\delta_{c_v i}. \quad (31)$$

The fraction of target weight within community i is given by:

$$b_i = \frac{1}{m}\sum_{vw} A_{vw}\delta_{c_w i} \quad (32)$$

$$= \frac{1}{m}\sum_{v} k_v^{in}\delta_{c_v i}. \quad (33)$$

Writing the modularity in terms of e_{ij}, a_i, and b_i by using $\delta_{c_v c_w} = \sum_i \delta_{c_v i}\delta c_w i$:

$$Q = \frac{1}{m}\sum_{ivw}\left[A_{vw} - \frac{k_v^{out}k_w^{in}}{m}\right]\delta_{c_v i}\delta_{c_w i} \quad (34)$$

$$= \sum_i \frac{1}{m}\sum_{vw} A_{vw}\delta_{c_v i}\delta_{c_w i} - \sum_i \left(\sum_{vw}\frac{k_v^{out}}{m}\delta_{c_v i}\right)\left(\sum_{vw}\frac{k_w^{in}}{m}\delta_{c_w i}\right) \quad (35)$$

$$= \sum_i \left(e_{ii} - a_i b_i\right). \quad (36)$$

When community i is merged into community j, the edge fractions and stub fractions are updated according to:

$$e'_{ii} = 0 \quad (37)$$

$$e'_{jj} = e_{ii} + e_{jj} + e_{ij} + e_{ji} \quad (38)$$

$$e'_{jk} = e_{ik} + e_{jk} \text{ (if } j \neq k) \quad (39)$$

$$a'_i = 0 \quad (40)$$

$$a'_j = a_i + a_j \quad (41)$$

$$b'_i = 0 \quad (42)$$

$$b'_j = b_i + b_j. \quad (43)$$

The change in modularity for merging communities i and j are given by:

$$\Delta_{ij} = \left(e'_{jj} - a'_j b'_j\right) - \left(e_{ii} + e_{jj} - a_i b_i - a_j b_j\right) \quad (44)$$

$$= e_{ij} + e_{ji} - a_i b_j - a_j b_i. \quad (45)$$

Initially, each vertex is in its own community ($c_v = v$) and the values of Δ_{ij}

and a_i are:

$$\Delta_{ij}^{(0)} = \frac{1}{m}\left[A_{ij} + A_{ji} - \frac{k_i^{out}k_j^{in} + k_i^{in}k_j^{out}}{m}\right] \tag{46}$$

$$a_i^{(0)} = \frac{k_i^{out}}{m} \tag{47}$$

$$b_i^{(0)} = \frac{k_i^{in}}{m}. \tag{48}$$

When community i is merged into j, the values of Δ are updated as follows:

$$\begin{aligned}
\Delta'_{jk} &= e'_{jk} + e'_{kj} - a'_j b'_k - a'_k b'_j & (49)\\
&= e_{ik} + e_{jk} + e_{ki} + e_{kj} - a_i b_k - a_j b_k - a_k b_i - a_k b_j & (50)\\
&= (e_{ik} + e_{ki} - a_i b_k - a_k b_i) + (e_{jk} + e_{kj} - a_j b_k - a_k b_j) & (51)\\
&= \Delta_{ik} + \Delta_{jk}. & (52)
\end{aligned}$$

Note that if there is no edge between i and k, the value of Δ_{ik} can be determined directly from a_i and a_k (the same applies to Δ_{jk}).

References

[1] Aaron Clauset, M. E. J. Newman, and Christopher Moore. *Finding community structure in very large networks*. Physical review E 70(6):066111, 2004.