

Modularity and community structure in networks

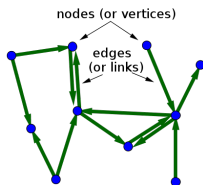
by David Stewart

May 2020

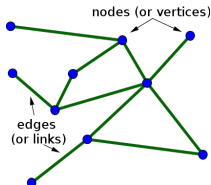
- Newman's leading eigenvector modularity algorithm:
<https://www.pnas.org/content/103/23/8577> [1]
- What are networks?
- What is modularity?
- Deriving Newman's algorithm
- Applying the algorithm
- Analysis of the algorithm

What are networks?

- Newman: “[networks are] *sets of nodes or vertices joined in pairs by lines or edges*”
- In math, they are studied as part of graph theory and topology
- Graphs: connections of vertices (nodes) by edges (links) $G = (V, E)$



(a) Example of a directed network



(b) Example of an undirected network

Figure: Directed (left) and undirected (right) From *mathinsight.org* [2]

- Increasingly important in modern applications: social networks, biology, computer networks...

What is modularity?

Why is it important to the study of networks?

- Network analysis is a powerful tool for understanding complex systems, but first you must have a network
- Newman's discussion assumes you have discovered a general network structure
 - E.g. Companies on the Stock Exchange, looking at correlation of returns...
- Given a network, can we find *communities* that exist within the network
 - Webpages in a community might all be related to the same topic
 - Users of a social media platform in a community might have the same interests

What is modularity? cont...

Why is it important to the study of networks?

- Modularity Q is a quality function: it tells you how well you have split your nodes up into communities
 - "true community structure in a network corresponds to a **statistically surprising arrangement of edges**, [and can] be quantified by using the measure known as modularity"
 - **N.B.** Some networks have *no* good division into communities. This is also a useful result.

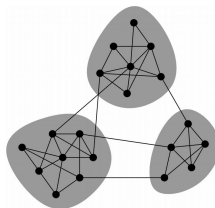


Figure: Communities in a network: small number of edges between the communities, lots of edges within the communities, from [1]

Newman's algorithm

Community detection and modularity maximisation through spectral analysis

*"...**modularity** can be expressed in terms of the **eigenvectors** of a characteristic matrix for the network, which I call the **modularity matrix**, and that this expression leads to a spectral algorithm for **community detection** that returns results of demonstrably **higher quality** than competing methods in **shorter running times**..."*

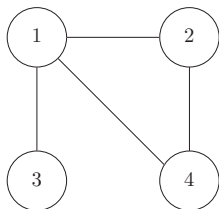
- Newman's algorithm seeks to maximise the modularity Q of a network
- Modularity can be positive or negative: positive values indicative the possibility of community structure
- Newman's approach is concerned with finding a "*statistically surprising*" arrangement of edges

Quickly deriving Newman's algorithm

How do you know if a division of the network is a good one?

- Suppose you have an adjacency matrix A of your network. $A_{ij} = 1$ if nodes i and j are connected, and 0 if they are not
- Imagine you took the network as described by A and “cut” all the edges in the network, so that no nodes are connected to one another anymore, but every node still has the same number of edges.

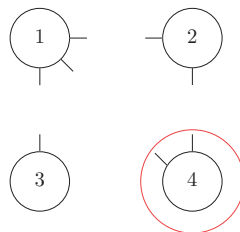
Our network, before cutting the edges



Adjacency matrix

0	1	1	1
1	0	0	1
1	0	0	0
1	1	0	0

After cutting the edges



Node 4 still has 4 edges,
they're just not connected
to anything

Quickly deriving Newman's algorithm cont...

How do you know if a division of the network is a good one?

- Given this disconnected network, we can ask “what is the probability that, if we were to reconnect the edges at random, node i and node j would end up connected?”
- If we denote the total number of edges in the network *before* cutting the edges as m , and the degree of a node x as k_x , the answer to this question is $\frac{k_i k_j}{2m}$
 - $k_i \cdot k_j$ = total number of ways you could reconnect two nodes (e.g. there are 6 ways you could reconnect nodes 1 and 4)
 - $2m$ = total number of edges after we cut. Technically, we should divide by $2m - 1$, however the -1 is often dropped as doing so introduces very little error in large networks.
 - E.g., the probability that Node 4 gets connected to Node 3 is $\frac{k_4 k_3}{2 \cdot 8} = \frac{2 \cdot 1}{16} = 0.125$
- Consider a binary adjacency matrix A , the difference between the actual number of connections between two nodes i and j and the number you'd expect on the basis of random chance is: $A_{ij} - \frac{k_i k_j}{2m}$
- The problem of modularity maximisation is NP-complete [3]

Quickly deriving Newman's algorithm cont...

- Recall that communities are characterised by small numbers of edges leading out of the community, and a greater number of intra-community edges
- One more piece of notation: we will first consider the case of splitting the network into two groups. We create a membership vector \mathbf{s} that contains 1 at index i if node i is in Group 1, and -1 if node i is in Group 2
 - By this definition, $\frac{1}{2}(s_i s_j + 1) = 1$ if nodes i and j are placed into the same group, and 0 if not
- With this in mind, we seek to maximise Q for a given community:

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \cdot (s_i s_j + 1) = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \cdot (s_i s_j) \quad (1)$$

Quickly deriving Newman's algorithm cont...

How do you know if a division of the network is a good one?

- This expression of modularity achieves 2 important outcomes:
 - 1 It is maximised by the difference between actual number of connections between 2 nodes and expected number of edges: i.e. it prioritises a *statistically surprising* arrangement of connections (good for dealing with noise);
 - 2 It is only concerned with intra-community edges (due to the membership vector \mathbf{s})

Scaling factor so result meets the convention of other modularity algorithms in this field

1 if s_i and s_j are in the same group,
-1 if not

$$\frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \cdot (s_i s_j)$$

Sum the difference between actual number of connections between two nodes, and the amount you'd expect based on random chance

Quickly deriving Newman's algorithm cont...

How do you know if a division of the network is a good one?

- Q can be written in matrix form as:

$$Q = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s} \quad (2)$$

where:

- \mathbf{s} is a column vector whose elements are the s_i discussed above;
- $\mathbf{B} = A_{ij} - \frac{k_i k_j}{2m}$
- **N.B** we will use a *slight* variation of equation 2 when dividing into more than 2 communities, read on!

Deriving Newman's algorithm cont...

How do you decide on the division?

- Q lets you know if you've made a good division, but how do we determine which nodes to put into Group 1 and which to put into Group 2?
- Use Newman's spectral analysis technique
- The idea is to rewrite the membership vector \mathbf{s} as a linear combination of the ortho-normalised eigenvectors (u_i) of \mathbf{B} :

$$\mathbf{s} = \sum_{i=1}^n a_i u_i \quad (3)$$

where

- $a_i = \mathbf{u}_i^T \cdot \mathbf{s}$ i.e the dot product of the transposed i^{th} eigenvector and the membership vector

Deriving Newman's algorithm cont...

How do you decide on the division?

- Thus we can rewrite Equation 1 as

$$Q = \frac{1}{4m} \sum_{i=1}^n a_i \mathbf{u}_i^T \mathbf{B} \sum_{j=1}^n a_j \mathbf{u}_j = \frac{1}{4m} \sum_{i=1}^n (\mathbf{u}_i^T \cdot \mathbf{s})^2 \beta_i \quad (4)$$

Where β_i is the i^{th} eigenvalue of \mathbf{B} (i.e. the eigenvalue that corresponds to the eigenvector \mathbf{u}_i^T)

- Assume we have the list of eigenvalues in β ordered from largest to smallest i.e. $\beta_1 \geq \beta_2 \geq \dots \geq \beta_n$. Since our goal is to maximise modularity, we want to choose our values of \mathbf{s} that maximise this sum. Since we assume β_1 is the largest eigenvalue, we can maximise modularity by maximising the first iteration of the sum

Deriving Newman's algorithm cont...

How do you decide on the division?

- Ideally, we would do this by making $\mathbf{s} = \mathbf{u}_i^T$ so that $(\mathbf{u}_i^T \cdot \mathbf{s})^2 = 1$, but that is probably not possible (if it were, you would only ever have to compute the first iteration, since \mathbf{u}^T is orthonormal, anything other than first iteration would be the dot product of two orthogonal vectors, which is 0);
- Instead, make \mathbf{s} as parallel as possible to \mathbf{u}_i^T
- If the i^{th} element of \mathbf{u}_1^T is positive, set the i^{th} element of \mathbf{s} to +1, else set it to -1.

Deriving Newman's algorithm cont...

Splitting into more than 2 communities?

- We just saw how to split a given network into 2 communities. How do we proceed to divide it further?
- Ans: Repeat the process! (with a slight difference)
- We now look at the *change* to the modularity score Q
- **N.B.** We cannot simply ignore the edges that connect the two communities: doing so is akin to “cutting” them out of the network, which would fundamentally alter the network structure for which we compute our modularity.

Deriving Newman's algorithm cont...

Splitting into more than 2 communities?

- The change in modularity ΔQ is then:

$$\begin{aligned}\Delta Q &= \frac{1}{4m} \sum_{i,j \in g} (B_{ij} - \delta_{ij} \sum_{k \in g} B_{ik}) s_i s_j \\ &= \frac{1}{4m} \mathbf{s}^T \mathbf{B}^{(g)} \mathbf{s}\end{aligned}\tag{5}$$

Where:

$$B_{ij}^{(g)} = B_{ij} - \delta_{ij} \sum_{k \in g} B_{ik}\tag{6}$$

- We stop the process then $\Delta Q \leq 0$: this indicates it is not a good division of the nodes

Community detection and modularity maximisation through spectral analysis

input : B : An $n \times n$ subgraph modularity matrix (equation 6 in Newman's paper)

: L : An incrementable label (implementation uses integers starting at 0)

: L : An incrementable label (implementation uses integers starting at 0)

: V : Indices of nodes. Nodes have their global index value e.g. a node of value 42 is the 42 node in the network

: L_v A vector of n labels corresponding to the community of each node

: Q : The modularity score of the network (Newman divides this by $4m$)

- ℓ : The number of nodes in the network (provided by the user)
- L : An incrementable label (implementation uses integers starting at 0)

Function assignCommunity(B, L, V):

$$B_{ij}^{(g)} \leftarrow B_{ij} - \delta_{ij} \sum_{k \in g} B_{ik}$$

▷ Compute subgraph modularity matrix (equation 6)

$$E_{val} \leftarrow \text{Largest eigenvalue of } B^{(g)}$$

$E_{vec} \leftarrow$ Eigenvector corresponding to E_{val}

$$s_i \leftarrow 1 \text{ if } E_{vec}(i) \geq 0, \text{ else } -1$$
$$\Delta Q \leftarrow s \cdot B^g \cdot s$$

if $\Delta Q < 0$ then

- | $L_v \leftarrow$ Label vertices in V with label L

Increment L
$$\Delta Q \leftarrow 0$$

else

$V_1 \leftarrow$ Vertices in V corresponding to $s_i == 1$

$V_2 \leftarrow$ Vertices in V corresponding to $s_i == -1$

$$[L_{v1}, L, \Delta Q_1] \leftarrow \text{assignCommunity}(s_i == 1, L, V_1)$$
$$[L_{p2}, L, \Delta Q_2] \leftarrow \text{assignCommunity}(s_i == -1, L, V_2)$$

end

$$\Delta Q \leftarrow \Delta Q + \Delta Q_1 + \Delta Q_2$$
$$L_v \leftarrow \text{concatenate/join } L_{v_1} \text{ and } L_{v_2}$$
return $L_v, L, \Delta Q$

End Function

Optimisation

Greedy optimisation

- Whilst the spectral analysis technique can achieve good results, Newman recommends an extra optimisation step
- Newman suggests a vertex moving approach (similar to the Kernighan-Lin algorithm);
- A refinement to this approach was suggested in 2009 by Sun. Et al [4]: computing ΔQ for each node movement would be expensive. Instead of calculating the modularity of a division after a node is moved, only the change in modularity caused by moving the node is calculated. The variation δQ obtained by moving node v_i is:

$$\delta Q = -\frac{s_i}{m} \mathbf{B}_i^T \mathbf{s} \quad (7)$$

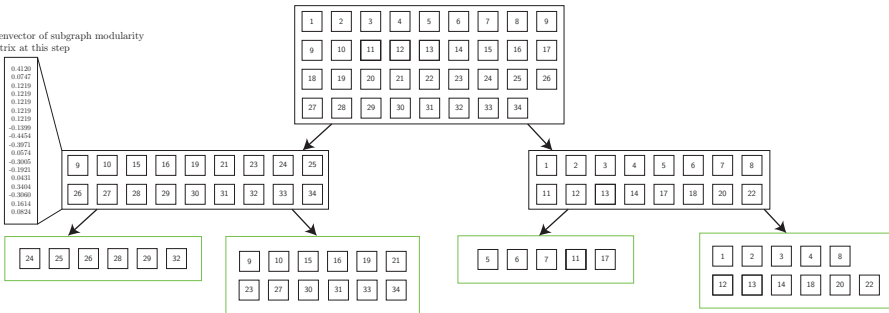
- Note: in the pseudocode given above, this optimisation step is not contained, it contains only the spectral analysis technique

Applying the algorithm

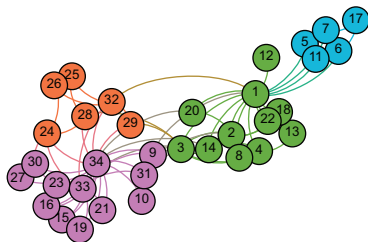
Divide and Conquer of Newman's algorithm on Zachary's Karate Network

eigenvector of subgraph modularity matrix at this step

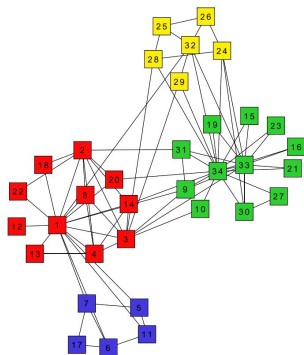
0.4120
0.0747
0.1219
0.1219
0.1219
0.1219
0.1219
-0.1399
-0.4454
-0.3971
0.0574
-0.3005
-0.1921
0.0431
0.3404
-0.3060
0.1614
0.0824



Applying the algorithm



(a) Community detection in the Karate network, as performed by the submitted MATLAB implementation



(b) Community detection performed by Agarwal and Kempe [?]

Figure: (a) Submitted MATLAB implementation, achieves same result as (b) Agarwal and Kempe's clustering of the Karate nodes

Analysis of the algorithm

Computational complexity

- Newman analyses the computational complexity of his algorithm in his paper, finding
 - The algorithm belongs to the efficiency class $O(n^2 \log n)$
 - **Better than** the $O(n^3)$ of the Newman-Girvan approach [5];
 - **Better (slightly) than** the $O(n^2 \log^2 n)$ of the Duch-Arenas extremal optimisation approach [6]
 - **Worse than** the $O(n \log^2 n)$ of the hierarchical agglomeration of Clauset, Newman and Moore [7] **but** provides much better results
 - This complexity is based on the two key elements of the algorithm:
 - 1 Division of the network: the worst case is that *every* node is in its own community. Dividing a starting network of n nodes until community size = 1 takes at most $\log n + 1$ divisions;
 - 2 Computing the eigenvectors at each step: We take this for granted in software packages such as `numpy` and `MATLAB`, but behind the scenes these need to be computed. Common methods are in $O(n^3)$ for dense matrices, however Newman recommends writing **B** in a form that leverages speedup of sparse graphs, reducing theoretical time to $O(n^2)$





Analysis of the algorithm

Comparing modularity results with other algorithms




Network	Size n	Modularity Q			
		GN	CNM	DA	This article
Karate	34	0.401	0.381	0.419	0.419
Jazz musicians	198	0.405	0.439	0.445	0.442
Metabolic	453	0.403	0.402	0.434	0.435
E-mail	1,133	0.532	0.494	0.574	0.572
Key signing	10,680	0.816	0.733	0.846	0.855
Physicists	27,519	—	0.668	0.679	0.723

Figure: Comparison of the leading eigenvector approach to existing algorithms at the time

References I

-  M. E. Newman, “Modularity and community structure in networks,” *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
-  M. Insight, “Network tutorial, may 2013,” URL http://mathinsight.org/thread/network_tutorial#introduction.
-  U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, “Maximizing modularity is hard,” *arXiv preprint physics/0608255*, 2006.
-  Y. Sun, B. Danila, K. Josić, and K. E. Bassler, “Improved community structure detection using a modified fine-tuning strategy,” *EPL (Europhysics Letters)*, vol. 86, no. 2, p. 28004, 2009.

References II

-  M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
-  J. Duch and A. Arenas, “Community detection in complex networks using extremal optimization,” *Physical review E*, vol. 72, no. 2, p. 027104, 2005.
-  A. Clauset, C. Moore, and M. E. Newman, “Hierarchical structure and the prediction of missing links in networks,” *Nature*, vol. 453, no. 7191, pp. 98–101, 2008.

Fin.