Modularity and Graph Algorithms

David Bader
Georgia Institute of Technology

Joe McCloskey
National Security Agency

12 July 2010

Outline

- **Modularity Optimization and the Clauset, Newman, and Moore Algorithm**
- **The Resolution Limit in Community Detection**
- **A Modified CNM Algorithm**
- **Final Comments**

Modularity Optimization

Let G represent a directed, unweighted graph with adjacency matrix A. Assume that L is the total number of edges in the network and that G has been partitioned into m disjoint sets of nodes $M_s, s = 1, 2, ..., m$.

Consider the set M_s and let l_{ss} represent the number of observed transitions from M_s to M_s. Now define $l_{s\bar{s}}$ to be the total number of transitions from within to outside the node set M_s.

The quantities $l_{\bar{s}s}$ and $l_{\bar{s}\bar{s}}$ are defined analogously.

Note that we have "collapsed" the adjacency matrix into a 2×2 table of transition counts where $L = l_{ss} + l_{s\bar{s}} + l_{\bar{s}s} + l_{\bar{s}\bar{s}}$.

Modularity Optimization

Let the row and column marginal sums of the 2×2 transition table be defined as $l_{+s} = l_{ss} + l_{\bar{s}s}$, and $l_{s+} = l_{ss} + l_{s\bar{s}}$.

We observe l_{ss} transitions within node set M_s. Under the hypothesis of independence the **estimated number** of transitions is as follows:

$$\hat{l}_{ss} = L \left(\frac{l_{+s}}{L} \right) \left(\frac{l_{s+}}{L} \right) = \frac{l_{s+} l_{+s}}{L}.$$

The **residual** is defined as $R_{ss} = l_{ss} - \hat{l}_{ss}$. If $R_{ss} > 0$ then we observe more transitions than expected. If R_{ss} is significantly larger than zero then we might suspect that M_s is a module (cluster).

Modularity Optimization

Following CNM we let $Q_s = R_{ss}/L$. The set of nodes M_s to said to be a **module** if and only if $Q_s > 0$.

Next we define the **modularity function** Q as follows:

$$Q = \sum_{s=1}^{m} Q_s = \frac{1}{L} \sum_{s=1}^{m} R_{ss} = \frac{1}{L} \sum_{s=1}^{m} \left(l_{ss} - \hat{l}_{ss} \right).$$

The goal of modularity optimization is to efficiently find a decomposition of G into a set of m modules $M_s, s = 1, 2, ..., m$ that maximizes the modularity function Q.

Cross-Product Formulation of Q

It follows that the residual $R_{ss} = I_{ss} - \hat{I}_{ss} = I_{ss} - \frac{I_{s+}I_{+s}}{L} > 0$ if and only if $I_{ss}I_{\bar{s}\bar{s}} - I_{s\bar{s}}I_{\bar{s}s} > 0$. Moreover,

$$Q = \frac{1}{L^2}\sum_{s=1}^{m}\left(I_{ss}I_{\bar{s}\bar{s}} - I_{s\bar{s}}I_{\bar{s}s}\right),$$

and M_s is a module if and only if $c_{12} = \frac{I_{ss}I_{\bar{s}\bar{s}}}{I_{\bar{s}s}I_{s\bar{s}}} > 1$, where c_{12} is the **cross-product ratio** of a 2×2 contingency table.

The expected value of c_{12} equals one if and only if the rows and columns of the table are statistically independent. $c_{12} > 1$ indicates positive dependence.

Modularity Optimization

Now let G represent an undirected graph. The adjacency matrix A is now symmetric and it follows that

$$l_{s\bar{s}} = l_{\bar{s}s}, \text{ and } L = l_{ss} + l_{\bar{s}s} + l_{\bar{s}\bar{s}}.$$

Since there are L edges in the graph, the sum of all of the entries of the adjacency matrix is equal to $N = 2L$.

Let X represent the 2×2 contingency table obtained by collapsing the adjacency matrix with respect to module M_s. Because of symmetry $x_{ss} = 2l_{ss}$, $x_{\bar{s}\bar{s}} = 2l_{\bar{s}\bar{s}}$, and $x_{\bar{s}s} = x_{s\bar{s}} = l_{\bar{s}s} = l_{s\bar{s}}$.

Modularity Optimization

Let $\hat{m}_{ij}$ represent the estimated entries of table X under independence. The marginal sums of X with respect to module M_s are $x_s = x_{+s} = x_{s+} = x_{ss} + x_{s\bar{s}} = 2l_{ss} + l_{s\bar{s}}$.

The **expected number** of transitions within M_s is seen to be

$$\hat{l}_{ss} = \frac{\hat{m}_{ss}}{2} = (2L)\left(\frac{x_{+s}}{2L}\right)\left(\frac{x_{s+}}{2L}\right)\left(\frac{1}{2}\right) = \frac{(x_s)^2}{4L}.$$

$$\text{As before } Q = \sum_{s=1}^{m} Q_s = \frac{1}{L}\sum_{s=1}^{m} R_{ss} = \frac{1}{L}\sum_{s=1}^{m}\left(l_{ss} - \hat{l}_{ss}\right).$$

Cross-Product Formulation of Q

The residual $R_{ss} = l_{ss} - \hat{l}_{ss} = l_{ss} - \frac{x_s^2}{4L}$, and M_s is a module if and only if $4L_{ss} > x_s^2$. Equivalently, M_s is a module if and only if $4l_{ss}l_{\bar{s}\bar{s}} - l_{s\bar{s}}^2 > 0$.

The cross-product ratio becomes

$$c_{12} = \left(\frac{x_{ss}}{x_{s\bar{s}}}\right)\left(\frac{x_{\bar{s}\bar{s}}}{x_{s\bar{s}}}\right) = \left(\frac{2l_{ss}}{l_{s\bar{s}}}\right)\left(\frac{2l_{\bar{s}\bar{s}}}{l_{s\bar{s}}}\right) = \frac{4l_{ss}l_{\bar{s}\bar{s}}}{l_{s\bar{s}}^2}.$$

Cross-Product Formulation of Q

The **modularity function** Q can be rewritten in the following useful form where the cross-product ratio is more apparent:

$$Q = \sum_{s=1}^{m} Q_s = \frac{1}{L} \sum_{s=1}^{m} R_{ss} = \frac{1}{4L^2} \sum_{s=1}^{m} \left(4 l_{ss} l_{\bar{s}\bar{s}} - l_{s\bar{s}}^2 \right).$$

Clauset, Newman, and Moore (2004) have developed a scalable, agglomerative algorithm to maximize the modularity function.

Their greedy algorithm makes use of "efficient data structures" and for certain networks can run in time $O\left(L \, log^2 L\right)$.

The Resolution Limit in Community Detection

"It is a-priori impossible to tell whether a module (large or small), detected through modularity optimization, is indeed a single module or a cluster of smaller modules."

Fortunato and Barthelemy (2007)

The Most Modular Connected Network

Let G be composed of n cliques, n even, where each clique has m nodes. Each clique is connected by a single edge to an adjacent clique. The number of edges in this graph is equal to

$$L = \frac{1}{2}nm(m-1) + n = \frac{n}{2}(m(m-1)+2).$$

Consider two partitions of G into hypothesized modules. In the first partition every module is a clique, while in the second partition every module is a pair of adjacent cliques. Then

$$Q_1 = 1 - \frac{2}{m(m-1)+2} - \frac{1}{n}, \; Q_2 = 1 - \frac{1}{m(m-1)+2} - \frac{2}{n}.$$

The Most Modular Connected Network

It follows that $Q_2 > Q_1$ if $n > m(m-1)+2$.

Since $L = \frac{n}{2}(m(m-1)+2)$ this further implies that $Q_2 > Q_1$ if $n^2 > 2L$.

If $n = 25$, and $m = 5$, then $Q_2 = 0.8745 > Q_1 = 0.8691$ and modularity optimization has not determined the "best" partition.

Fortunato and Barthelemy have analyzed real world data sets that exhibit the property that the "best" partition does not have the maximum modularity score.

An Improved Resolution Limit

Berry, Hendrickson, Laviolette, and Phillips (2010) use the local topology of the graph to infer edge weights. Then they generalize the results of Fortunato and Barthelemy to show it is possible to resolve much smaller communities.

They also suggest that the dendrogram constructed by the CNM algorithm can be mined to resolve smaller communities. This might be even more effective with their modified CNM algorithm.

A Different Resolution?

Claim: The failure of the CNM algorithm to resolve communities is due to the properties of the modularity function Q and not the information that it uses.

Can CNM be salvaged?

Is it possible to optimize a different objective function that uses the same information and resolves communities?

The CNM Algorithm Revisited

Consider a graph with L edges and three modules. Let l_{ij} represent the number of transitions from module M_i to module M_j. Then

$$L = l_{11} + l_{12} + l_{13} + l_{22} + l_{23} + l_{33}.$$

What happens to the modularity function if modules M_1 and M_2 are merged together?

We find two equivalent representations of the difference $Q_{1,2,3} - Q_{1\cup 2,3}$ of the modularity functions due to the merge of M_1 and M_2. If $Q_{1\cup 2,3} > Q_{1,2,3}$ then CNM will merge M_1 and M_2.

The CNM Algorithm Revisited

Use the three modules M_1, M_2 and M_3 to collapse the undirected graph into a 3×3 table $X_{1,2,3}$ of counts x_{ij}. The x_{ij} are determined from the transition counts l_{ij} as follows:

$$x_{jj} = 2l_{jj},$$

$$x_{ij} = x_{ji} = l_{ij} \text{ for } i \neq j,$$

$$x_1 = x_{1+} = x_{+1} = 2l_{11} + l_{12} + l_{13},$$

$$x_2 = x_{2+} = x_{+2} = l_{12} + 2l_{22} + l_{23}.$$

Because of symmetry the total sample size of the x_{ij}'s is $N = 2L$.

The CNM Algorithm Revisited

If we further merge M_1 and M_2 together then the collapsed graph has transition counts of the form

$$\begin{aligned} l'_{11} &= l_{11} + l_{12} + l_{22}, \\ l'_{12} = l'_{21} &= l_{13} + l_{23}, \\ l'_{22} &= l_{33}. \end{aligned}$$

The resulting 2×2 symmetric table $X_{1\cup 2,3}$ of counts x'_{ij} has entries

$$\begin{aligned} x'_{11} &= 2l_{11} + 2l_{12} + 2l_{22}, \\ x'_{12} = x'_{21} &= l_{13} + l_{23}, \\ x'_{22} &= 2l_{33}. \end{aligned}$$

Properties of the Modularity Function

Let $Q_{1,2,3}$ be the modularity function when modules M_1, M_2, M_3 are considered to be separate modules, and $Q_{1\cup 2,3}$ the modularity function when modules M_1 and M_2 are merged together.

Let $\Delta Q = Q_{1,2,3} - Q_{1\cup 2,3}$ be the difference of the two modularity functions. If $Q_{1\cup 2,3} > Q_{1,2,3}$ then $\Delta Q < 0$, and CNM will merge M_1 and M_2.

To understand the properties of the modularity function we find two equivalent representations of $\Delta Q = Q_{1,2,3} - Q_{1\cup 2,3}$.

Properties of the Modularity Function

Module M_3 makes the same contribution $Q_3 = R_{33}/L$ to each of the modularity functions $Q_{1,2,3}$ and $Q_{1\cup 2,3}$. Thus,

$$\begin{aligned} L\Delta Q &= L\left(Q_{1,2,3} - Q_{1\cup 2,3}\right) \\ &= LQ_{1,2,3} - LQ_{1\cup 2,3} \\ &= \left(R_{11} + R_{22} + R_{33}\right) - \left(R_{1\cup 2,1\cup 2} + R_{33}\right) \\ &= R_{11} + R_{22} - R_{1\cup 2,1\cup 2}. \end{aligned}$$

We have left to compute $R_{1\cup 2,1\cup 2} = I'_{11} - \hat{I}'_{11}$.

Properties of the Modularity Function

Now $\hat{l}'_{11} = \frac{1}{4L} {x'_1}^2$, where $x'_1 = 2l_{11} + 2l_{12} + 2l_{22} + l_{13} + l_{23}$.
A straightforward calculation yields

$$2L^2 \Delta Q = c_0 + c_1 + c_2 + c_3,$$

where we define

$$\begin{aligned} c_0 &= 4l_{11}l_{22} - l_{12}^2, \\ c_1 &= 2l_{11}l_{23} - l_{12}l_{13}, \\ c_2 &= 2l_{22}l_{13} - l_{12}l_{23}, \\ c_3 &= l_{13}l_{23} - 2l_{12}l_{33}. \end{aligned}$$

The $c'_j s$ have cross-product interpretations as 2×2 subtables of $X_{1,2,3}$.

Properties of the Modularity Function

With this representation it is easy to "manufacture" examples where $\Delta Q < 0$ and modules M_1 and M_2 will be merged in error by the CNM algorithm.

Note that l_{33} is the number of transitions in the rest of the graph and recall that

$$c_3 = l_{13} l_{23} - l_{12} l_{33}.$$

If $l_{12} > 0$ and l_{33} is "large" then $c_3 << 0$ and $\Delta Q < 0$.

The most modular connected network is a simple example where this happens.

Properties of the Modularity Function

A second representation of ΔQ leads to a simple observation that addresses several flaws in the current CNM algorithm.

Standard properties of symmetric contingency tables can be used to show that the residuals of the 3×3 contingency table $X_{1,2,3}$ satisfy

$$2R_{11} + R_{12} + R_{13} = 0,$$
$$R_{12} + 2R_{22} + R_{23} = 0,$$
$$R_{13} + R_{23} + 2R_{33} = 0.$$

Analogously, the residuals of the 2×2 collapsed contingency table $X_{1\cup 2,3}$ satisfy $R_{1\cup 2,1\cup 2} = R_{33}$.

Properties of the Modularity Function

These equations lead to another representation of $L\Delta Q$.

$$\begin{aligned} L\Delta Q &= L\left(Q_{1,2,3} - Q_{1\cup 2,3}\right) \\ &= R_{11} + R_{22} - R_{1\cup 2,1\cup 2} \\ &= R_{11} + R_{22} - R_{33} \\ &= \frac{1}{2}(-R_{12} - R_{13}) + \frac{1}{2}(-R_{12} - R_{23}) - \frac{1}{2}(-R_{13} - R_{23}) \\ &= -R_{12}. \end{aligned}$$

Hence,

$$R_{12} > 0 \text{ if and only if } \Delta Q < 0.$$

A Modified CNM Algorithm

This last inequality means that two modules M_1 and M_2 are candidates to be merged by the CNM algorithm if their residual $R_{12} > 0$.

- The greedy CNM algorithm merges the two modules M_i and M_j with the largest residual $R_{ij} > 0$.
- **CNM Modification:** Before two modules M_i and M_j are merged their residual $R_{ij} > 0$ should be deemed "statistically significant."
- The residuals R_{ij} are "not comparable" since their variances are not necessarily equal. Let $S_{ij} = \frac{R_{ij}}{std(R_{ij})}$ be standardized residuals.

A Modified CNM Algorithm

- **CNM Modification:** Before two modules M_i and M_j are merged their standardized residual $S_{ij} > 0$ should be deemed "statistically significant."
- The greedy modified CNM algorithm will merge the two modules M_i and M_j with the largest statistically significant standardized residual $S_{ij} > 0$.
- This adjustment to the CNM algorithm will alleviate the resolution limit problem, and the tendency for CNM to favor the merging of "large" modules.

Final Comments

Modularity optimization can fail to resolve communities.

- This failure is due to the properties of the modularity function.
- The CNM algorithm can be modified to avoid the resolution limit problem.
- Algorithms that use spectral information from the modularity matrix can detect modules when CNM "fails."