Limits of Modularity and Greedy Algorithms

Alexander Gibson

Table of Contents

Modularity - Motivations

Modularity allows us to find communities in a graph in a quantifiable manner.

- What is a community?
- What do we mean by quantifiable?

Modularity - Motivations

- Community: A subset of nodes in a graph with a higher density of edges between them than the average density of edges in the entire network
- “Randomly wired networks lack an inherent community structure.” - Barabási
- For a given partition, we can compare the edge density of that partition with the edge density of a randomly rewired network of the same nodes.
- In this way we can quantify the quality of a partition.

Modularity - Calculation

For a given network with

- N nodes
- E edges
- n_c communities
  - N_c nodes
  - E_c edges
  - $c = 1 \ldots n_c$
  - k_c total degree of nodes in the community

we will derive the formula for modularity.

Modularity - Calculation

Modularity is the difference between the edge density of the real network A and the expected edge density of its corresponding randomly-rewired network p.

Thus the modularity of a given community is given by

$$M_c = (1 / 2E) * \Sigma(A_{ij} - p_{ij}) \text{ for } (i,j) \in C.$$

If p is BA scale-free, we have that

$$p_{ij} = k_i k_j / 2E.$$

Modularity - Calculation

By rewriting A_{ij}, we obtain

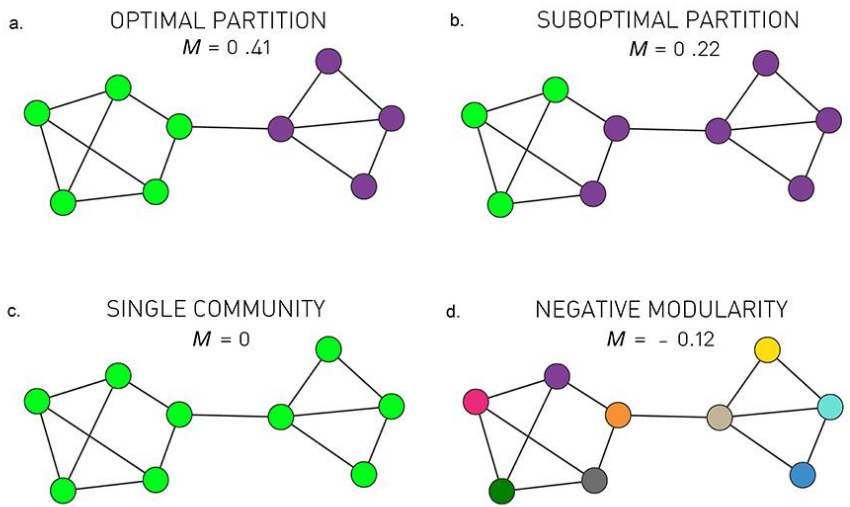
$$M_c = (E_c / E) - (k_c / 2E)^2.$$

We can then generalize to the modularity of the entire network:

$$M = \Sigma M_c \text{ for } c=1 \text{ to } n_c.$$

Modularity - Properties

- Maximum modularity M ➔ best partition
- M = 0 ➔ single community
- M < 0 ➔ N communities (each node is a community)



Modularity - Limits

- Resolution Limit
- Modularity Maxima

Modularity - Limits

Resolution Limit

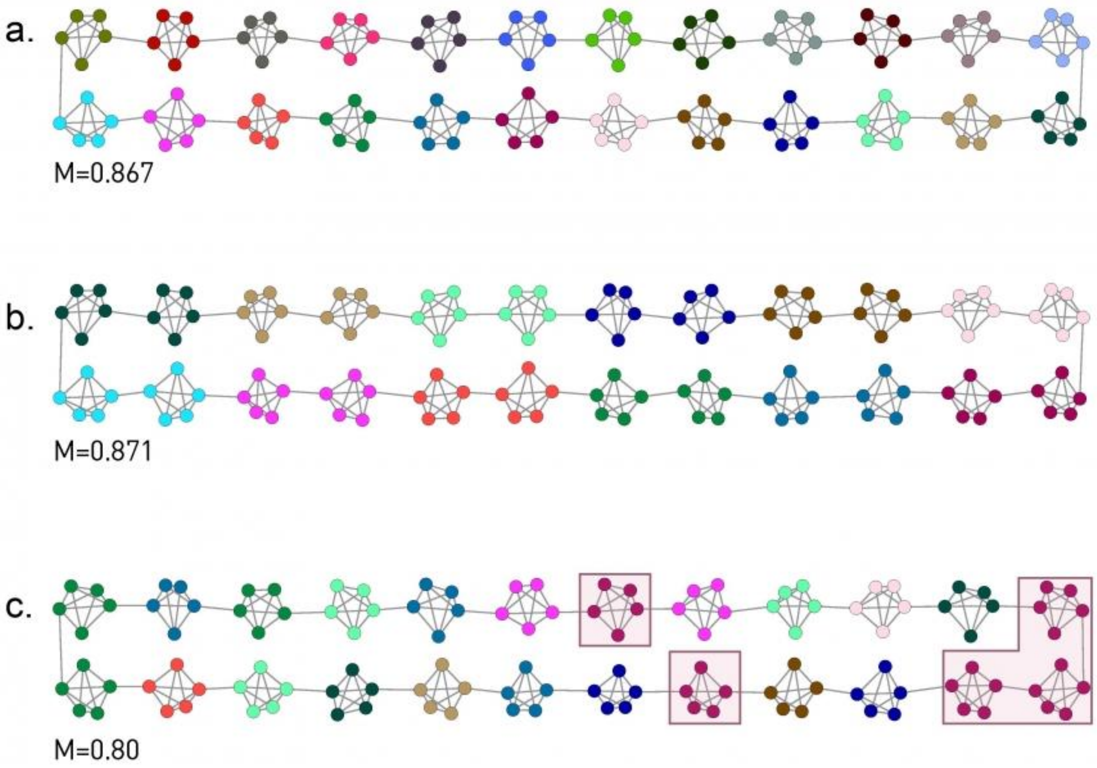
- Small communities are merged with larger communities
  - Assuming $k_A \backsimeq k_B = k$: If $k \leq \sqrt{(2E)}$ then communities A and B merge.
  - $\sqrt{(2E)}$ is referred to as the resolution limit.
- The modularity maximization algorithm cannot detect communities smaller than the resolution limit
  - Real networks often have many small communities – MM algorithm can mischaracterize them

Modularity - Limits

Modularity Maxima

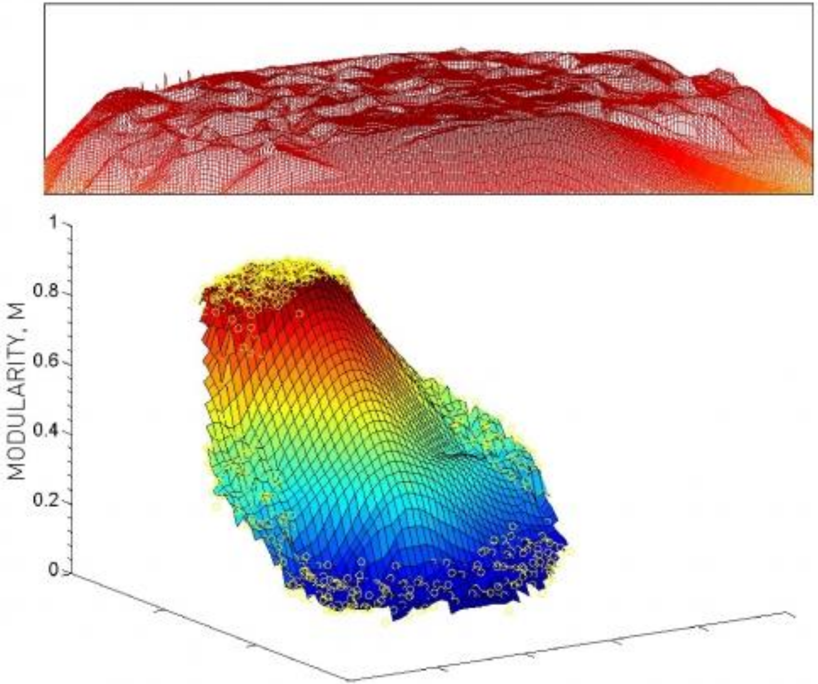
- Ideally we want to find an optimal modularity M_{max}
  - In practice on large networks, this is impossible!
- Merging two communities yields $\Delta M = (e_{AB} / E) - (2 / n_c^2)$
  - ΔM decreases as n_c increases ➔ goes to 0 for large numbers of groups
  - “Modularity Plateau”

Modularity - Limits



Modularity - Limits

d.



Modularity - Summary

- Question: How do we find the communities in a network?
- Answer: Maximizing modularity yields the partition that best captures the communities in a network.
  - $M = 0$ ➔ one community
  - $M < 0$ ➔ every node is its own community
- Limitations:
  - Cannot capture small communities (resolution limit)
  - Difficult to find true maximum modularity (modularity plateau)

Greedy Algorithms - Motivations

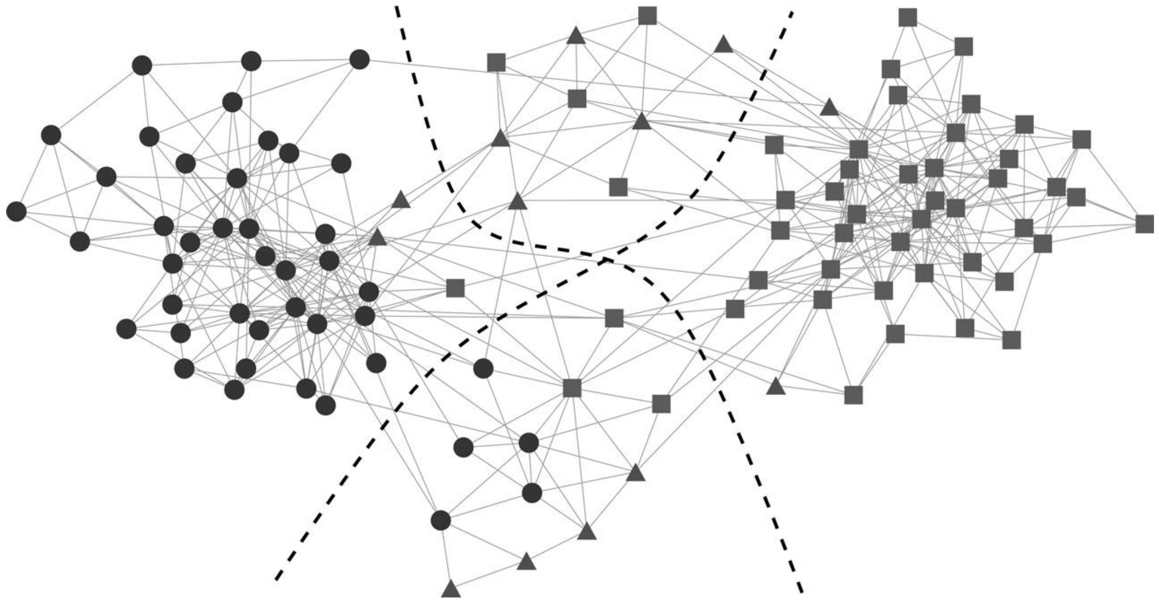
- Optimizing modularity is NP-hard.
  - Brute forcing M_{max} is computationally infeasible
- Question: Is there an algorithm that finds a partition with $M \approxeq M_{max}$ that doesn't have to check every possible partition?
- Answer: Yes - greedy algorithms.
- Modularity plateau to the rescue!
  - Easy to get close to M_{max} – and on average, the larger the network the easier it is.

Greedy Algorithms - Newman Algorithm

Very simple, but very powerful.

1. Assign each node to its own community
2. Merge the two communities with maximal ΔM and record M
3. Repeat 2. until the entire network is a single community
4. Return the partition with maximal M

Greedy Algorithms - Newman Algorithm

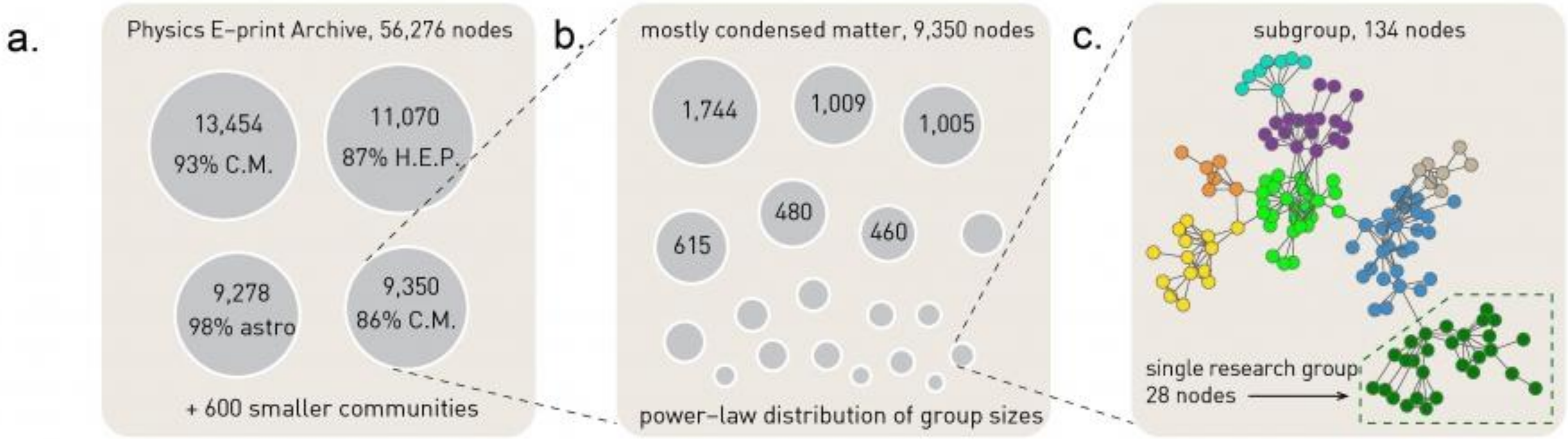


Network of books about American politics. Nodes are books classified by political alignment, edges represent books frequently purchased by the same readers.

Greedy Algorithms - Newman Algorithm

- Also solves the problem of the resolution limit – just run the NA on a subcommunity!
- Layering the NA can greatly increase the modularity of a given partition

Greedy Algorithms - Newman Algorithm



By running NA on the condensed matter community, M goes from 0.713 to 0.807 ➜ a 13% increase!

Greedy Algorithms - Newman Algorithm

Running Time Analysis:

- Calculating ΔM is O(1) ➔ step 2 is O(E)
- After merge in step 2, updating edges is O(N)
- Step 2 repeats N-1 times ➔ overall complexity: O[(E + N)N]

Still very computationally intensive!

Greedy Algorithms - Louvain Method

- On average, getting close to M_{max} is easier the larger your network
- The Louvain Method takes this to the extreme – approximates M_{max} in just O(E)
    - Can identify communities in networks with N > 2m in 2 minutes

Greedy Algorithms - Louvain Method

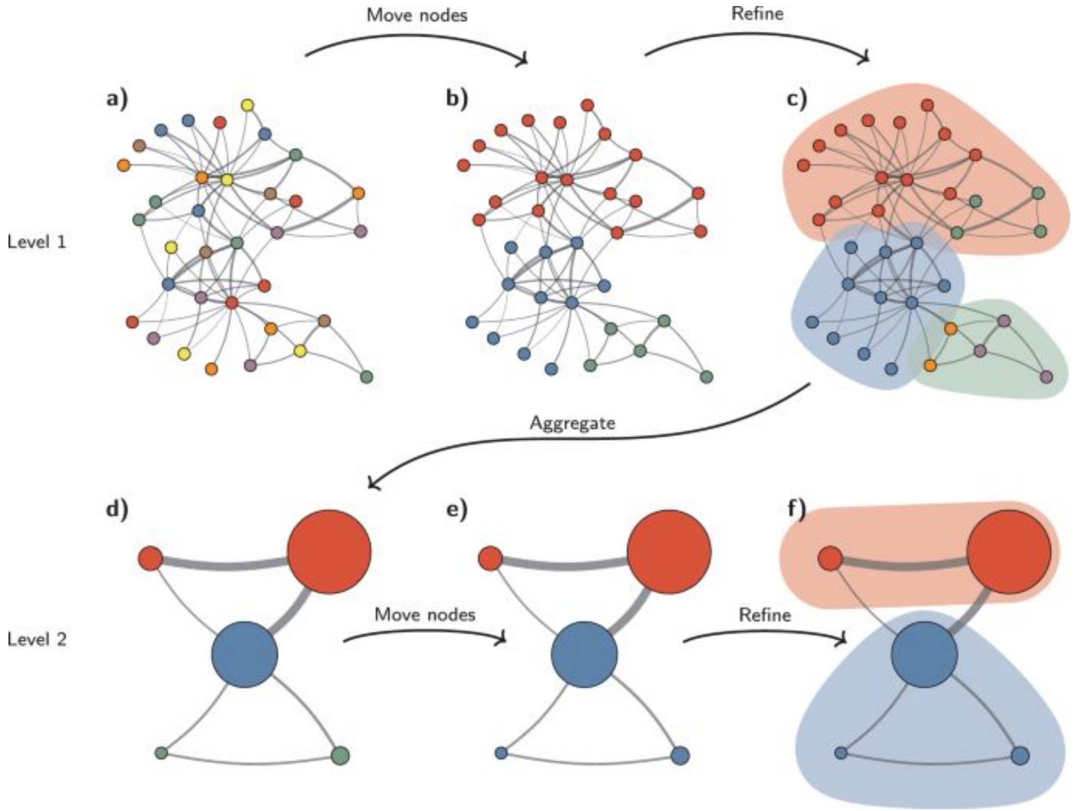
1. Assign each node to its own community
2. For each node A, calculate ΔM for moving A into the community of each of its neighbors B and move it into the community with maximum ΔM
3. Repeat 2. until no positive ΔM exists
4. Construct a new network whose nodes are the communities found in 3.
5. Repeat 1. – 4. until a maximum modularity is found

Greedy Algorithms - Louvain Method

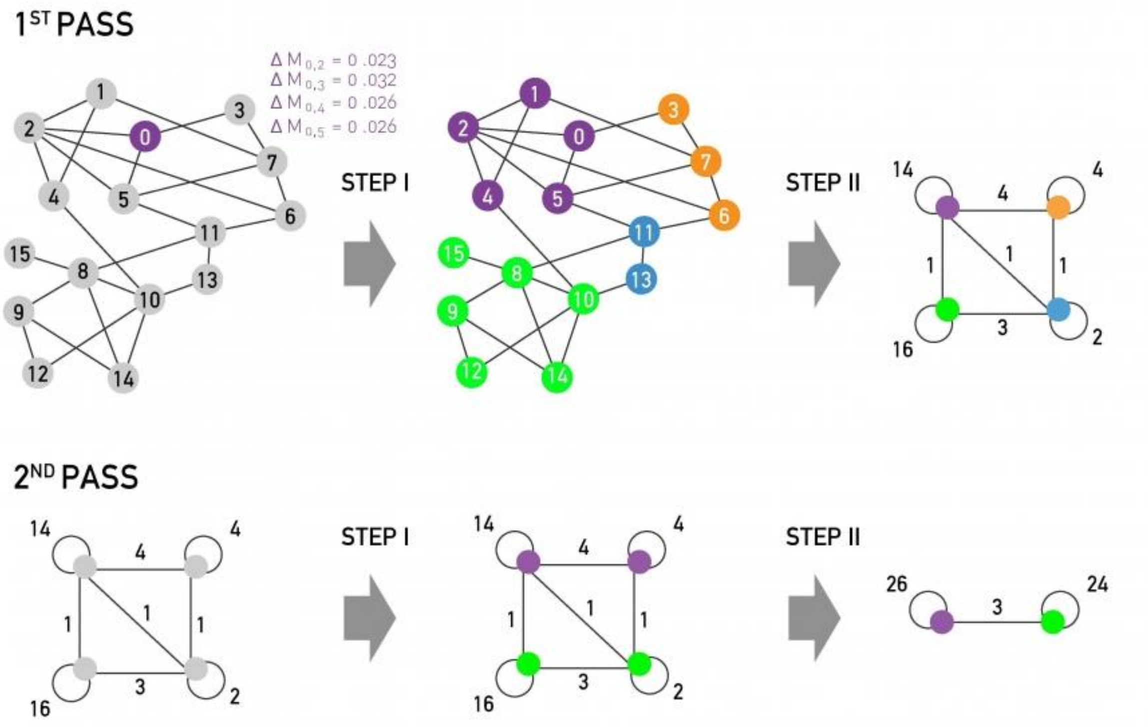
Running Time Analysis:

- Key insight: ΔM calculations are O(1).
  - How? The calculation for ΔM only depends on properties of the community as a whole and the node being moved – not its neighbors.
- Step 3. is O(E) ➔ the only step which isn't O(1).
  - No edges are gained or lost – existing edges are just "magnified" when constructing new graphs.
  - Edges can be visited multiple times, but that's okay.
- Overall complexity: O(E).

Greedy Algorithms - Louvain Method



Greedy Algorithms - Louvain Method



Works Cited

- http://networksciencebook.com/chapter/9#modularity
- https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/modularity.pdf
- https://doi.org/10.1088/1742-5468/2008/10/P10008
- https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.nature.com%2Farticles%2Fs41598-019-41695-z&psig=AOvVaw1eepRDIXJ1FwSRmMuOYdEU&ust=1701770146857000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCIiLsvrB9YIDFQAAAAAdAAAAABAa
- https://www.pnas.org/doi/10.1073/pnas.0601602103