Theory and Applications of Complex Networks

Classes Nine–Eleven

College of the Atlantic

David P. Feldman

10 October 2008

`http://hornacek.coa.edu/dave/`

Community discovery and higher-order structure in networks

Summary Thus Far

Basic Network Properties:

- Path lengths
- Degree distribution
- Cluster coefficient

Three Models:

- Erdős-Rényi random graphs
- Watts-Strogatz small-world model
- Preferential attachment models & scale-free degree distributions

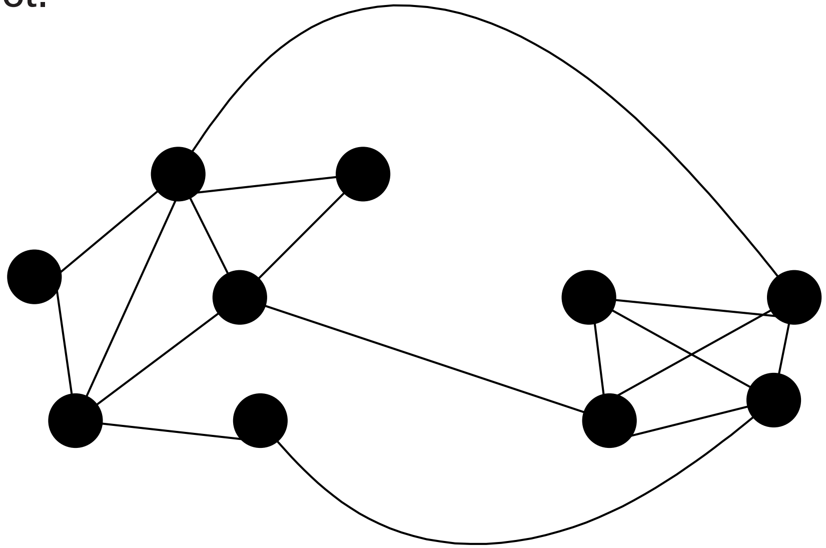
Note: “Scale-free network” is an improper term. It is only the degree distribution which is scale free, not the entire network.

Community Discovery and Higher Order Structure

- How can we say more about the structure of large networks?
- Are the nodes clustered in any way?
- What might these clusters mean?
- How can we discover these clusters?
- More generally, how can we think about higher order structures in networks? I.e., structures involving more than a single node.
- How might the presence or absence of these higher order structures influence the dynamics on a network or give clues about how the network formed?
- And what might these higher order structures mean physically or biologically or economically, etc.?
- All of these questions are areas of active research.

What Makes a Community?

- Suppose we suspect that a network is made of two communities. Can we test this?
- A group is a community if there are more within-community connections than one would expect.



- How can we quantify this?
- Note: Communities are also sometimes referred to as modules.

How can we Specify Communities?

- We are interested in discovering community structure.
- A community can be thought of as a *partition* of the network. Each node is assigned to one and only one community.
- Usually, we don't want to specify the number of communities in advance.
- That is, we want to discover the optimal number of communities *and* the optimal placement of individual nodes into communities.
- This is an extremely difficult computational task. Trying out all possible community specifications wildly unfeasible.

Data Mining

- Of particular interest will be figuring out how to address these questions for large networks with thousands or even millions of nodes.
- Reliable visualization is essentially impossible.
- Large data sizes require efficient algorithms.
- “Data mining is the art of extracting useful patterns from large bodies of data; finding seams of actionable knowledge in the raw ore of information. The rapid growth of computerized data, and the computer power available to analyze it, creates great opportunities for data mining in business, medicine, science, government, etc.”
- “Data mining is related to statistics and to machine learning, but has its own aims and scope. Statistics is a mathematical science, studying how reliable inferences can be drawn from imperfect data. Machine learning is a branch of engineering, developing a technology of automated induction. [Data mining] use[s] tools from statistics and from machine learning.
- Quotes from Shalizi `http://www.stat.cmu.edu/~cshalizi/350/`.

(Analysis of Algorithms)

- When considering algorithms, it is important to think carefully about how the run time depends on the size of the problem.
- Usually, we are interested in how the run time of an algorithm depends on n, the size of the input.
- This relationship is expressed using “Big O” notation.
- If an algorithm’s run time depends on the square of the input size n, we would write that it is an $\mathcal{O}(n^2)$ algorithm.
- Formally, $f(x) = \mathcal{O}(g(x))$ means that there is some M and x_0 such that

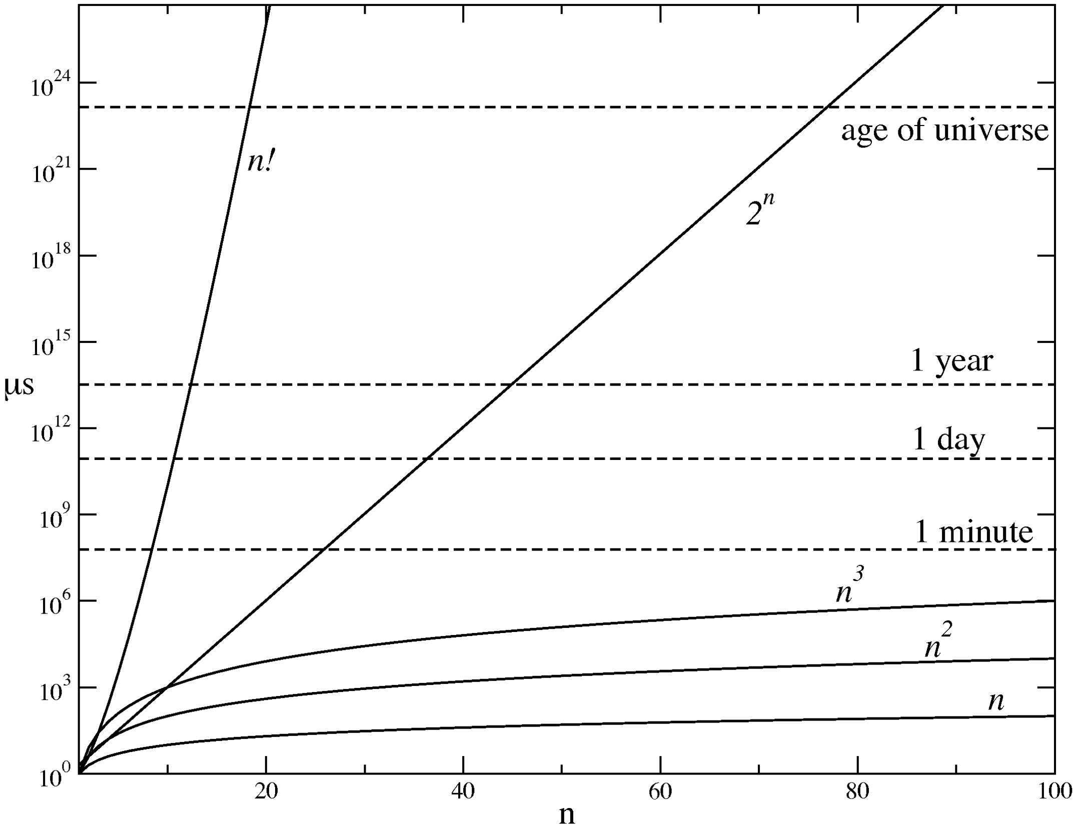
$$|f(x)| < M|g(x)| \text{ for all } x > x_0 \; . \tag{1}$$

- If $g(x) = x^3$ we would say that $f(x)$ is order x^3 or scales as x^3.
- In addition to runtime, the memory usage of an algorithm is also important.

(Analysis of Algorithms)

- The constant M doesn't matter for large x, but it could make a big difference for small x.
- The order of an algorithm is a statement about its large-x, asymptotic behavior.
- The order of an algorithm is given for the *worse-case* scenario: i.e., the longest possible runtime.
- Very often, an average runtime is more relevant.
- The average runtime can be experimentally determined, but this isn't always reliable.
- In some instances, analytically figuring out an average run time can be very hard.

(Analysis of Algorithms)



- The order of the algorithm makes a big difference!

- Figure: S. Mertens and C. Moore, *The Nature of Computation*, Oxford University Press, 2008.

Community Discovery

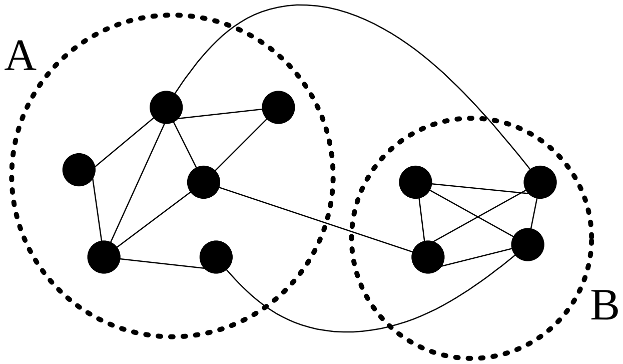
- This is an area of active research. There is not a standard algorithm, nor is there general agreement about which algorithm is the best.
- Nevertheless, there are some well understood and reliable(?) methods for community discovery.
- There are two components to any community discovery algorithm:
  1. A metric: some way of measuring how good a potential community partition is. This, is the thing that the algorithm tries to maximize.
  2. A search method: some way of generating candidate community partitions which are then evaluated according to the metric.

Modularity

- One commonly used metric for the quality of a community partition is the *modularity* Q.
- The larger the Q-value, the better the community partition.
- Let A be the adjacency matrix for the network.
- Let n_c be the number of communities in our partition.
- Define an $n_c \times n_c$ matrix E whose elements e_{ij} are the fraction of total links starting at a node in community i and ending in community j
- Let $a_i = \sum_j e_{ij}$ be the fraction of links connected to i.
- Then

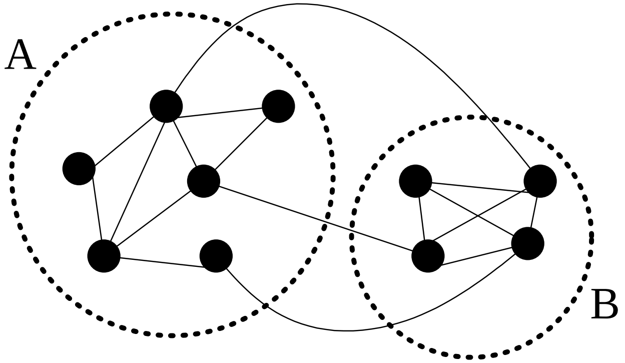
$$Q = \sum_i (e_{ii} - a_i^2) \ . \tag{2}$$

Modularity: A Measure of Community-ness



- Suppose we think there are two communities, A and B.
- Divide the links into two types: between-community and within-community.
- For this network, there are 8 links within A, 6 within B, and 3 between A and B.
- There are 17 total links.
- So $\frac{8}{17}$ of the links are within community A.
- Is this a lot? How many would we expect?

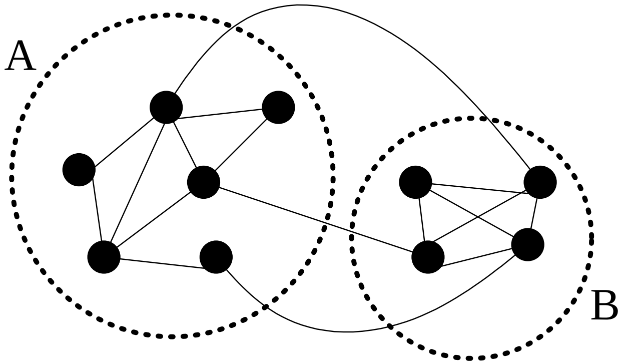
Modularity: Continued



- 8 links within A, 6 within B, and 3 between A and B, and 17 total links.
- $\frac{8}{17}$ of the links are within community A. Is this a lot?
- Of the 17 total links, 11 connect to A.
- If no community structure, then the communities edges link to are independent.
- So, if we draw a link at random, what is the chance it connects A to A?

$$\text{Prob of connection to A} \times \text{Prob of connecting to A} = \frac{11}{17} \times \frac{11}{17}$$

Modularity: Continued

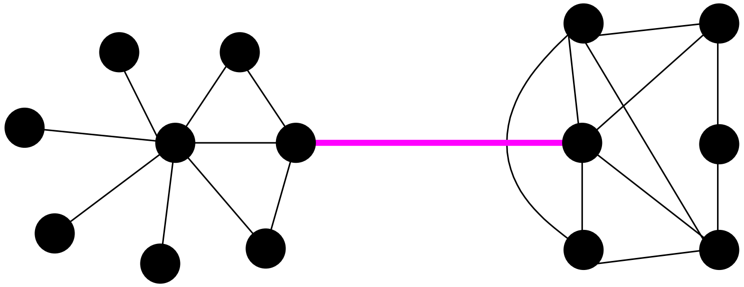


- **Modularity** is defined as the fraction of within-community links minus the number of within community links one would expect if the links were random.
- For community A: $\frac{8}{17} - \frac{11^2}{17^2}$.
- For community B: $\frac{6}{17} - \frac{9^2}{17^2}$.
- Adding these together, we get the modularity of the network. In this case, modularity $= 0.12$.
- **Modularity is a measure of the strength of a set of communities. The bigger the number, the stronger the community structure.**

Modularity: Conclusion

- There are a number of community discovery algorithms which are based on the modularity Q.
- These algorithms generate a series of candidate community partitions and evaluates the Q for each.
- The partition that has the largest Q is then chosen.
- I'm not sure the statistical properties of Q are well understood, or if this is the ideal metric for community-ness.
- These algorithms are conceptually simple, have been widely used, and have produced reasonable and interesting results.

Girvan-Newman Betweenness Algorithm



- The betweenness is a property of an edge.
- Betweenness measures how important an edge is in connecting other members of the network.
- To calculate betweenness, consider all possible pairs of edges.
- Find the shortest path connecting each pair.
- The betweenness of an edge is the number of shortest paths running along that edge
- See, e.g., Finding and evaluating community structure in networks, M. E. J. Newman and M. Girvan, Phys. Rev. E 69, 026113 (2004). for discussion of betweenness and modularity.

Girvan-Newman Betweenness Algorithm

- Central Idea: Edges with high betweenness separate communities.

  1. Calculate all betwennesses
  2. Remove the edge with the highest betweenness
  3. Repeat until all nodes are in their own community

- As one does this process, the network fractures into successively smaller and smaller, disconnected components
- Consider each disconnected component as a community
- Each successive splitting generates a new candidate community partition, one with one more community than before.
- Evaluate the modularity Q for each partition
- Choose the partition with the largest Q

Girvan-Newman Betweenness Algorithm

- Note: This is an example of a divisive algorithm.
- This algorithm gives good results for networks with known community structure: e.g., college American football conferences, Zachary's karate club.
- This is an $\mathcal{O}(m^2n)$ algorithm. $n =$ number of nodes, $m =$ number of links.
- For sparse networks, $m \sim n$, so this is $\prime(n^3)$.
- Not feasible for very large networks.
- This algorithm gives not just a single, optimal-Q community structure, but a full hierarchy, or dendogram.

Newman Fast Algorithm

- See M. E. J. Newman, Phys. Rev. E 69, 066133 (2004), and Aaron Clauset, M. E. J. Newman, and Cristopher Moore, Phys. Rev. E 70, 066111 (2004).
- This is an agglomerative algorithm.
- Initially, every node is its own community.
- Merge the two communities that lead to the largest increase in Q.
- Repeat, until all have been merged together.
- Like the previous algorithm, this yields a dendogram.
- The partition with the highest Q is chosen as optimal.
- This algorithm is $\mathcal{O}(md \log n)$, where d is the dendogram depth.
- For sparse, hierarchical networks, the order is $\mathcal{O}(n \log^2 n)$.
- This is much faster than the Girvan-Newman algorithm.

Community Discovery/Data Mining Thoughts

- Discovering communities when you have good reason to believe that communities are present is a hard problem.
- But what if you're not certain communities are present?
- Most algorithms will still find communities.
- Is there some notion of statistical significance for communities?
- There are at least two issues: the significance of the overall community structure, and the significance of the placement of individual nodes.