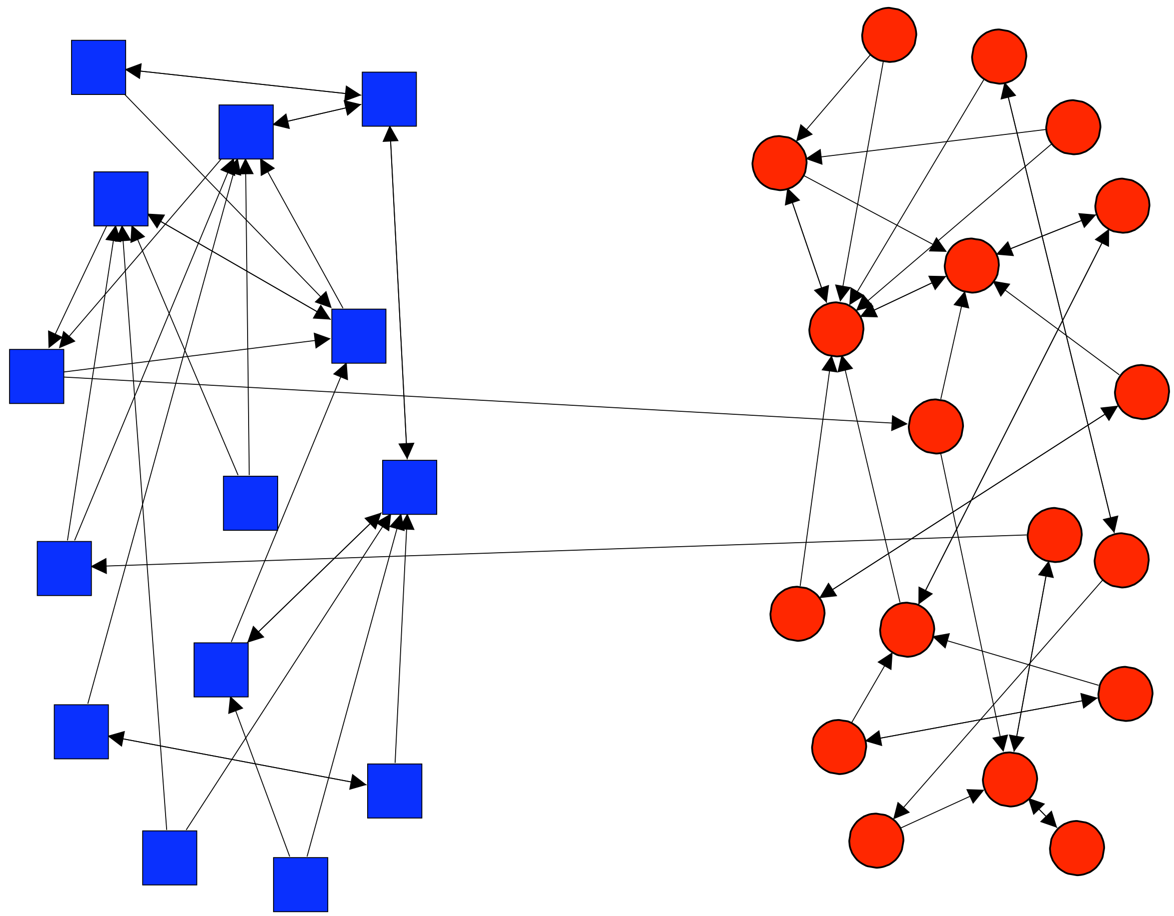
Community Structure and Beyond

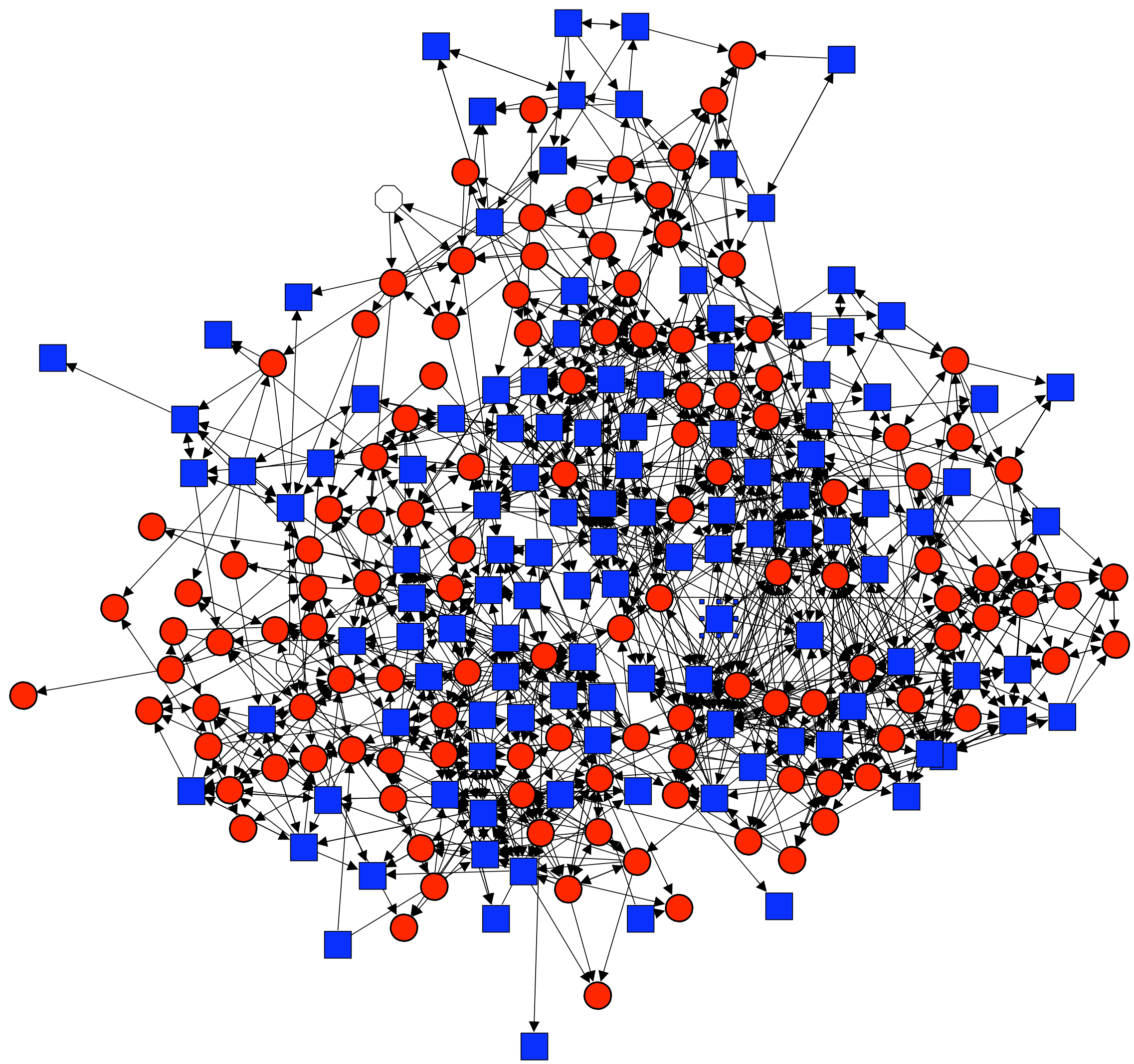
Elizabeth A. Leicht

MAE: 298
April 9, 2009

Why do we care about community structure?



Large Networks



Discussion Outline

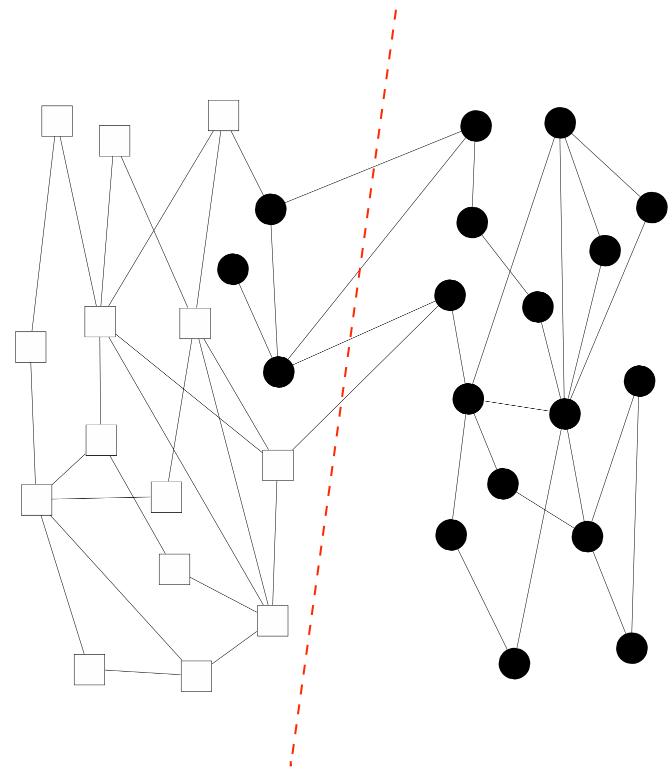
- Overview of past work on community structure.
- How to determine the “best” number of communities.
- Fast linear algebra based method.
- Bringing in statistics.

A Brief History of Methods

- **Spectral methods**, graph partitioning problems.
    - A well known example is **spectral bisection**, which uses the graph/ network **Laplacian**.

$$L_{ij} = \delta_{ij} k_i - A_{ij}$$

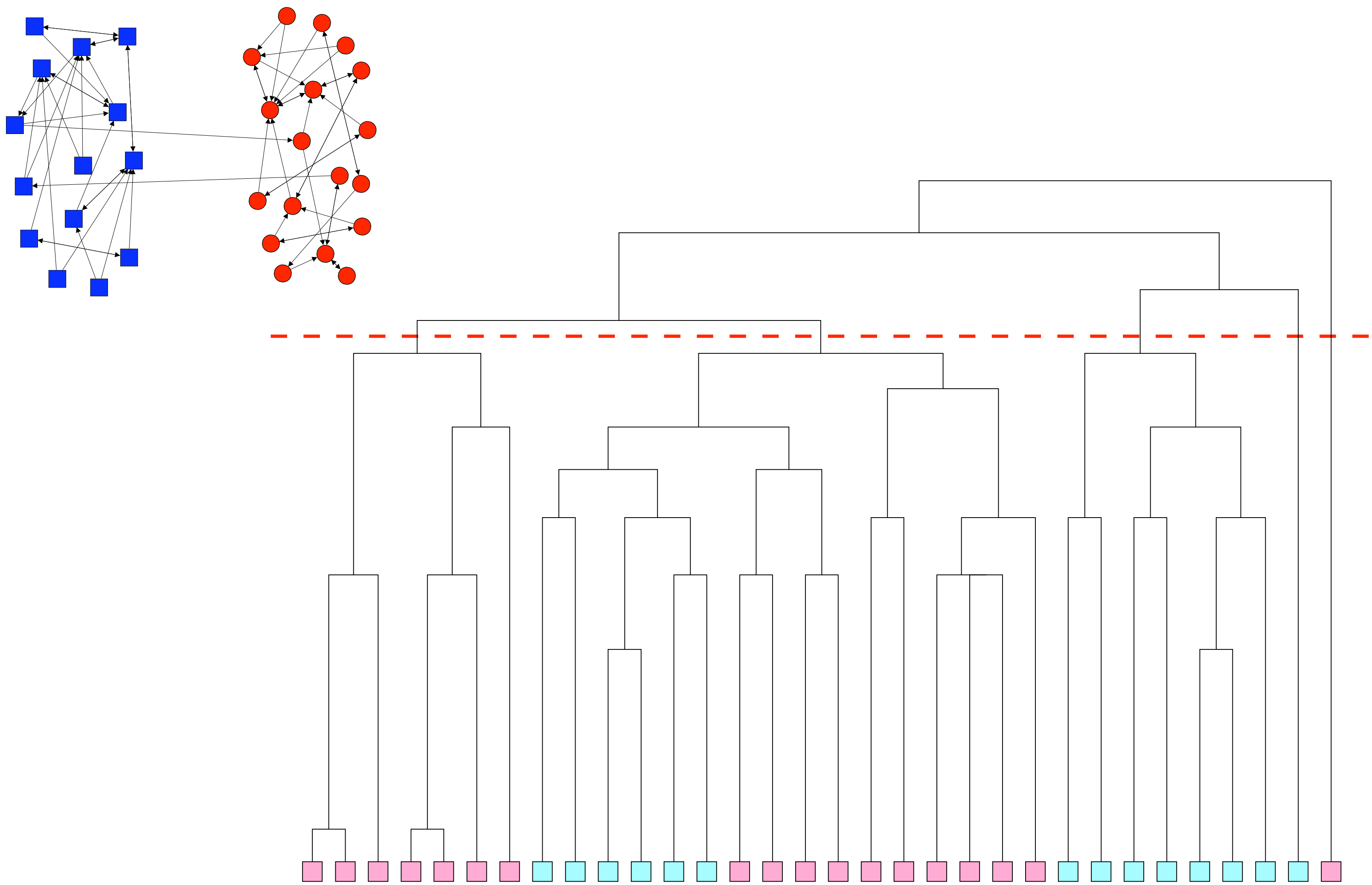
- In the special case of a network having only two communities, Fiedler proposed a method for identifying the the members nodes.



A Brief History of Methods

- **Hierarchical clustering**: groups nodes into communities such that nodes within a community are *similar* to each other in some sense; widely used in sociology.
  - Technique 1) calculate a weight, W_{ij} for every pair of nodes in the network 2) then take the n nodes with no edges between them an add edges between pairs one by one in order of their weights, from strongest to weakest.
  - Many ways exist for calculating the W_{ij} values.
  - The entire process is frequently represented as a **dendrogram**, a visualization of the vertices coalescing into communities.

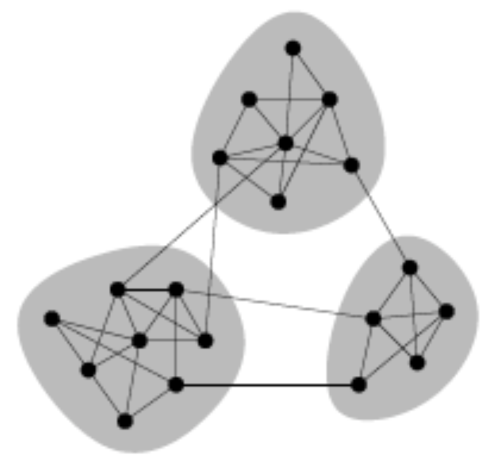
A Brief History of Methods



GN Algorithm

- **Girvan-Newman Algorithm:** a divisive method for determining community structure that focuses on the *betweenness* of edges.
  - **Edge betweenness:** the number of shortest paths between pairs of vertices that run along an edge.
  - Removing edges of high betweenness breaks up the connected network into communities.

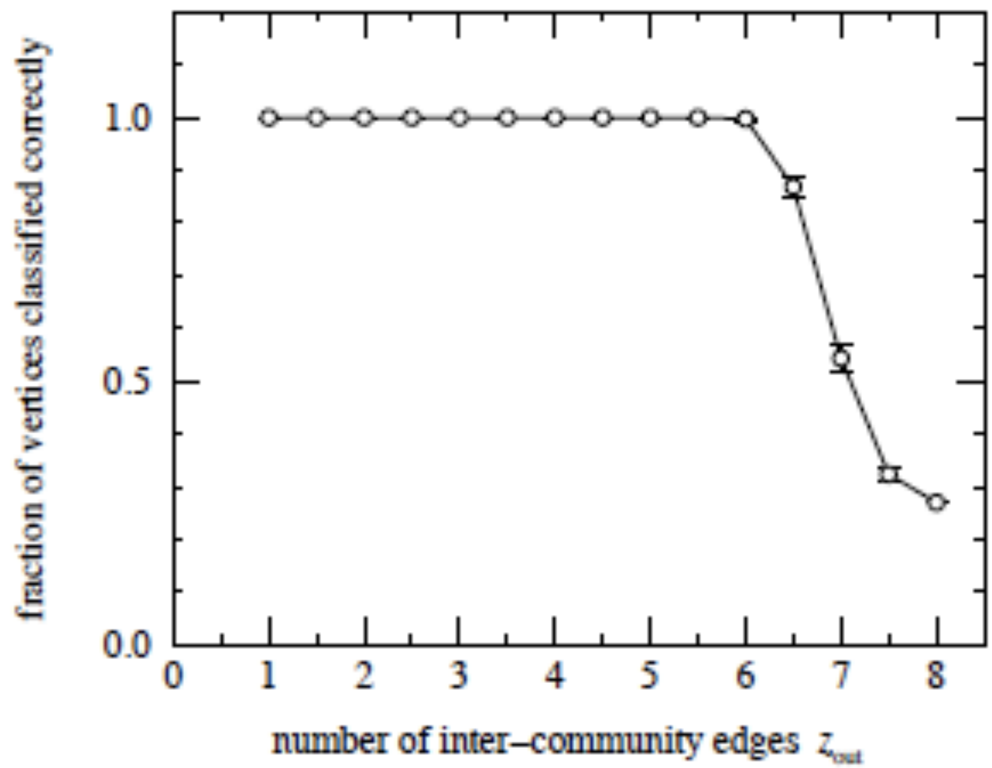
M. Girvan and M.E.J. Newman, "Community structure in social and biological networks" *Proc. Natl. Acad. Sci. USA* **99**, 7821–7826 (2002).



Algorithm

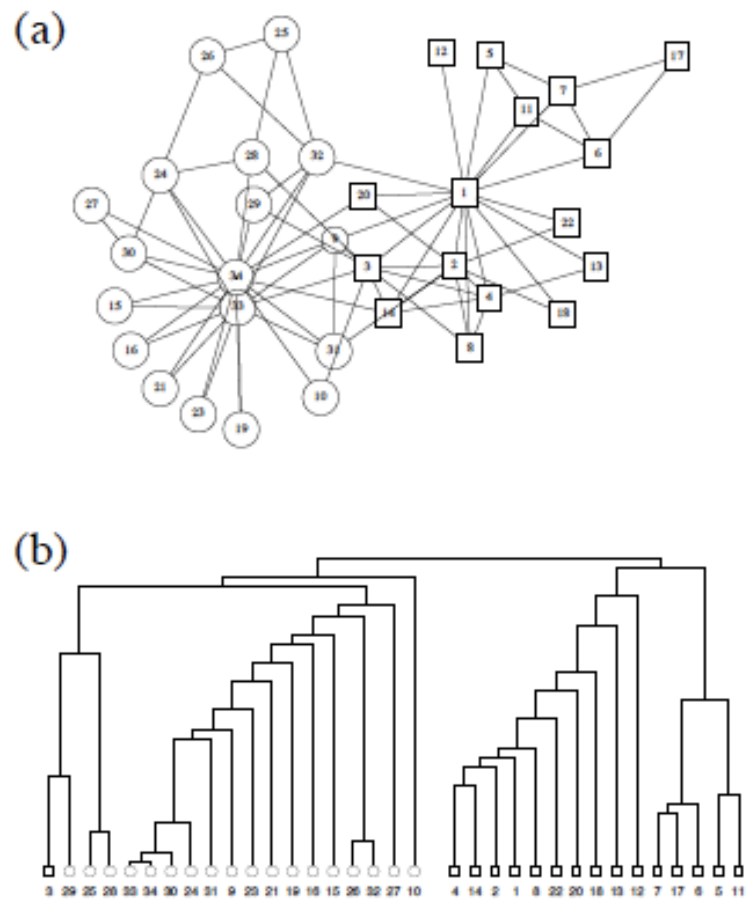
1. Calculate the betweenness for all edges in the network.
2. Remove the edge with the highest betweenness.
3. Recalculate betweenness for all edges affected by the removal.
4. Repeat from steps 2 until no edges remain.

GN Algorithm



GN Algorithm

The classic “Karate Club” example



Modularity

M.E.J. Newman and M. Girvan, “Finding and evaluating community structure in networks” *Phys. Rev. E* **69**, 026113 (2004)

- Introduced by Newman and Girvan to quantify which division of a network into communities/groups was the best.
  - Related to Newman’s work on assortativity in networks, “Mixing patterns in networks” *Phys. Rev. E* **67**, 026126 (2003)
- **Modularity**: the fraction of edges falling within communities minus the *expected fraction* of such edges.

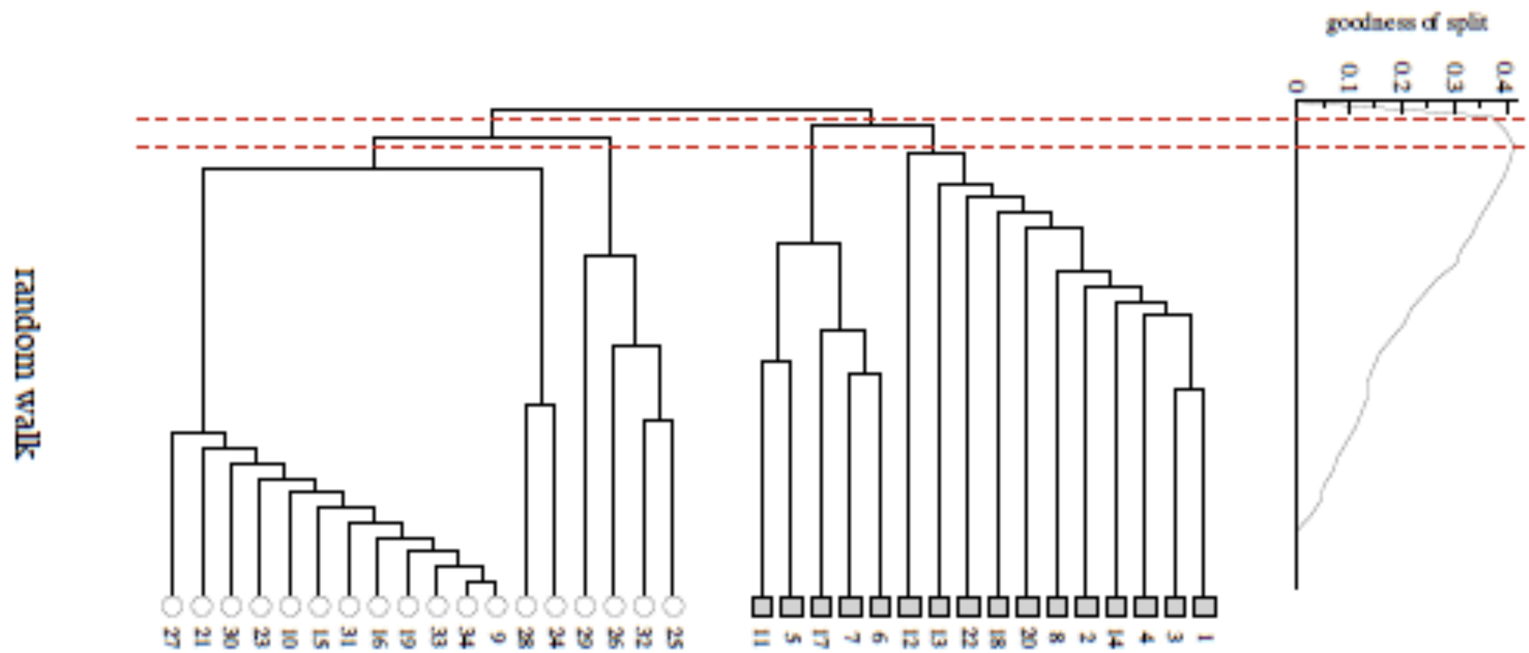
e_{ij} : the fraction of all edges in the network that link vertices in community i to vertices in community j.

$a_i = \sum_j e_{ij}$ the fraction of edges that connect to vertices in community i.

$$Q = \sum (e_{ii} - a_i^2) = \mathrm{Tr}\,\mathbf{e} - ||\mathbf{e}^2||$$

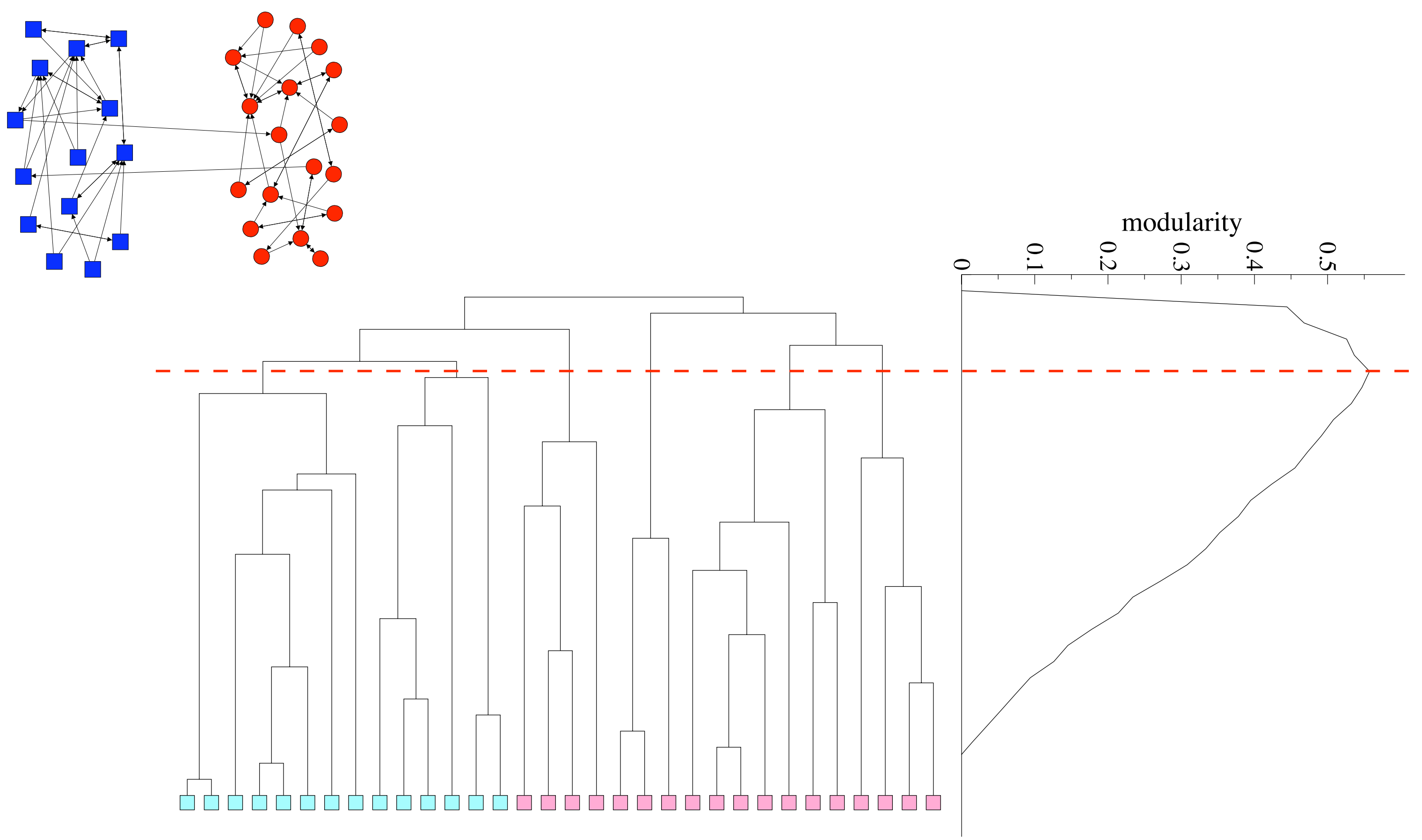
Modularity

M.E.J. Newman and M. Girvan, “Finding and evaluating community structure in networks” *Phys. Rev. E* **69**, 026113 (2004)

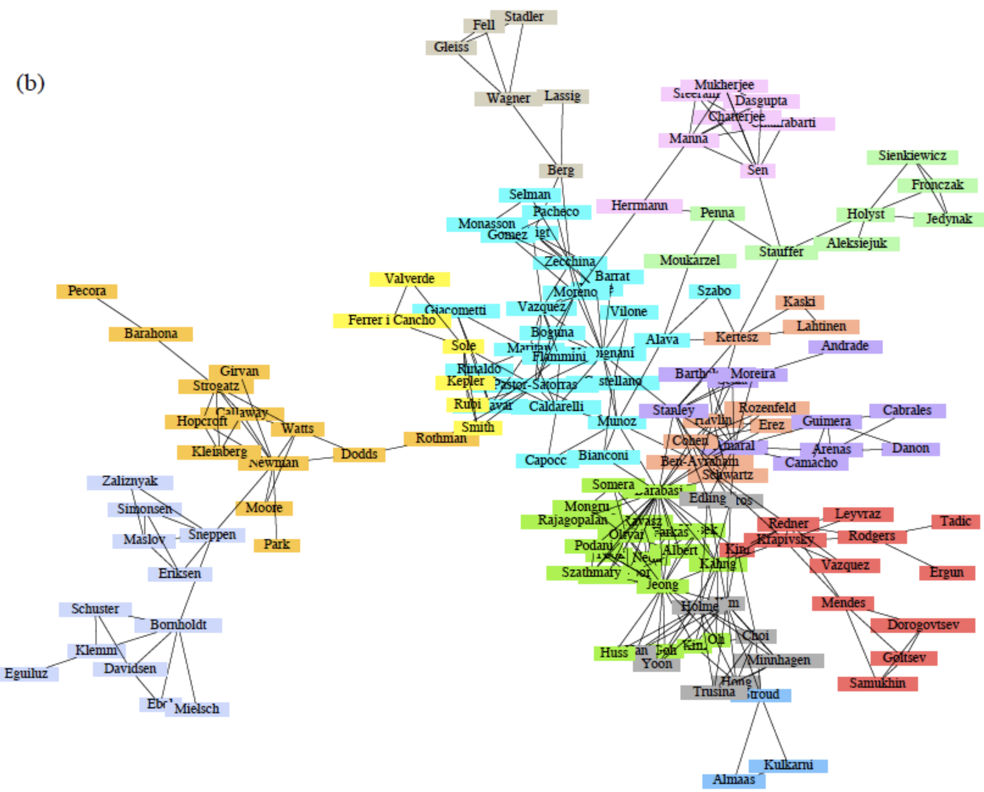


Again the “Karate Club”

Modularity



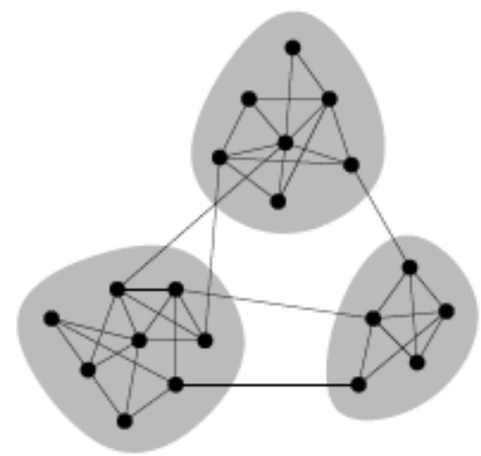
We love to study ourselves . . .



A New-New Approach

- Newman later returned to the subject of community structure and modularity with a new-new approach.
- Modularity maximization was the leading tool for determining optimal community structure.
  - Simulated annealing had been shown to be very successful, but slow.

M. E. J. Newman, *Proc. Natl. Acad. Sci. USA* **103**, 8577–8582 (2006).



Calculating Modularity

Rewrite modularity using the adjacency matrix.

$$Q = \frac{1}{2m} \sum_{ij=1}^{n} [A_{ij} - P_{ij}] \delta_{c_i, c_j}$$

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge from } j \text{ to } i \\ 0, & \text{otherwise} \end{cases} .$$

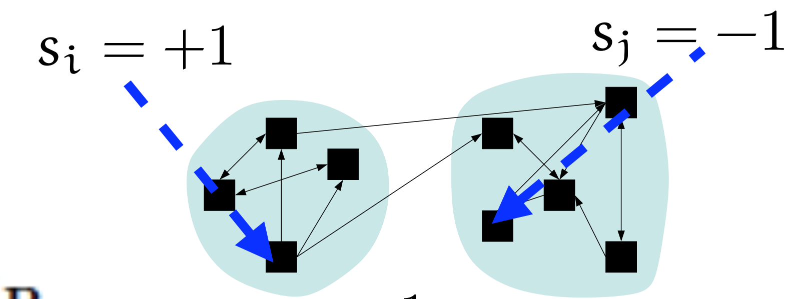
$P_{ij} =$ the expected number of edges from j to i.

$c_i =$ the community to which i belongs.

How do we determine the "expected" number of edges between two vertices?

$$P_{ij} = \frac{k_i k_j}{2m}$$

Division of a Network into Two Communities



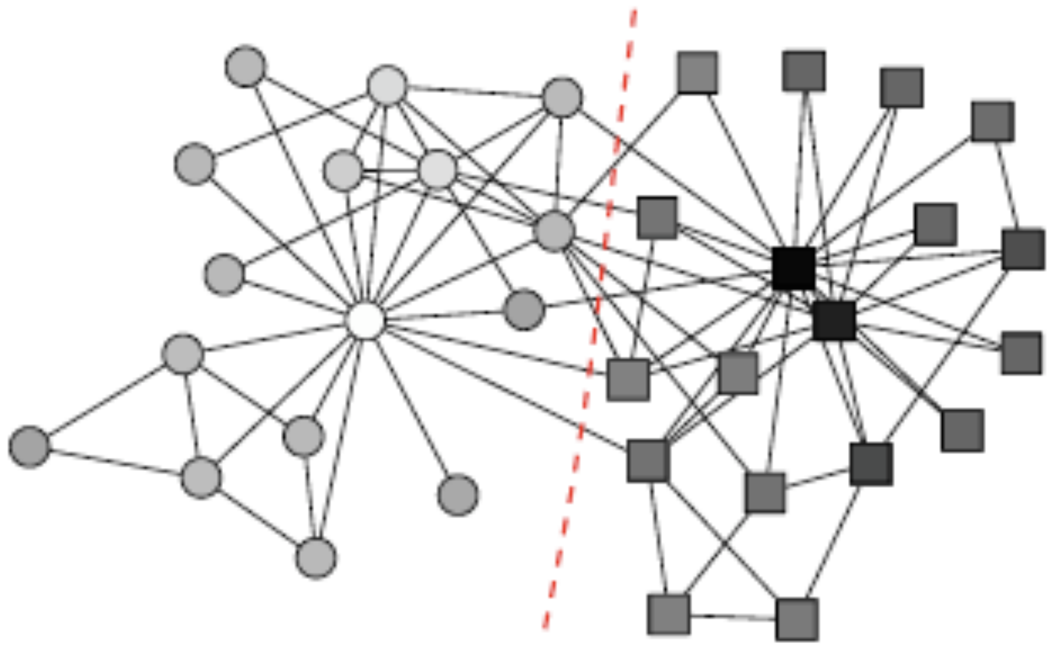
$$\delta_{ij} = \frac{1}{2}(s_i s_j + 1)$$

$$Q = \frac{1}{4m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)s_i s_j = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s},$$

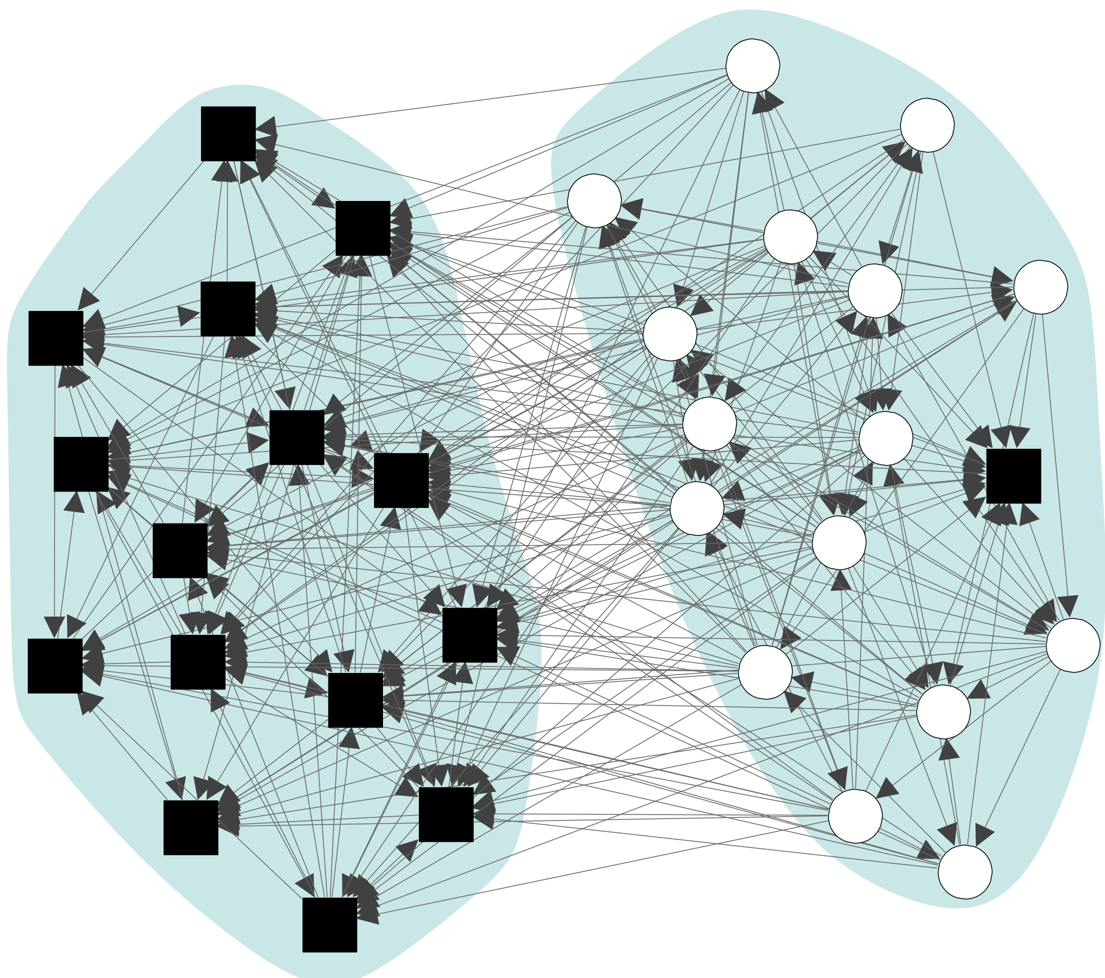
$$\mathbf{s} = \textstyle\sum_{i=1}^{n} a_i \mathbf{u}_i \text{ with } a_i = \mathbf{u}_i^T \cdot \mathbf{s}.$$

$$Q = \sum_i a_i \mathbf{u}_i^T \mathbf{B} \sum_j a_j \mathbf{u}_j = \sum_{i=1}^{n} (\mathbf{u}_i^T \cdot \mathbf{s})^2 \beta_i,$$

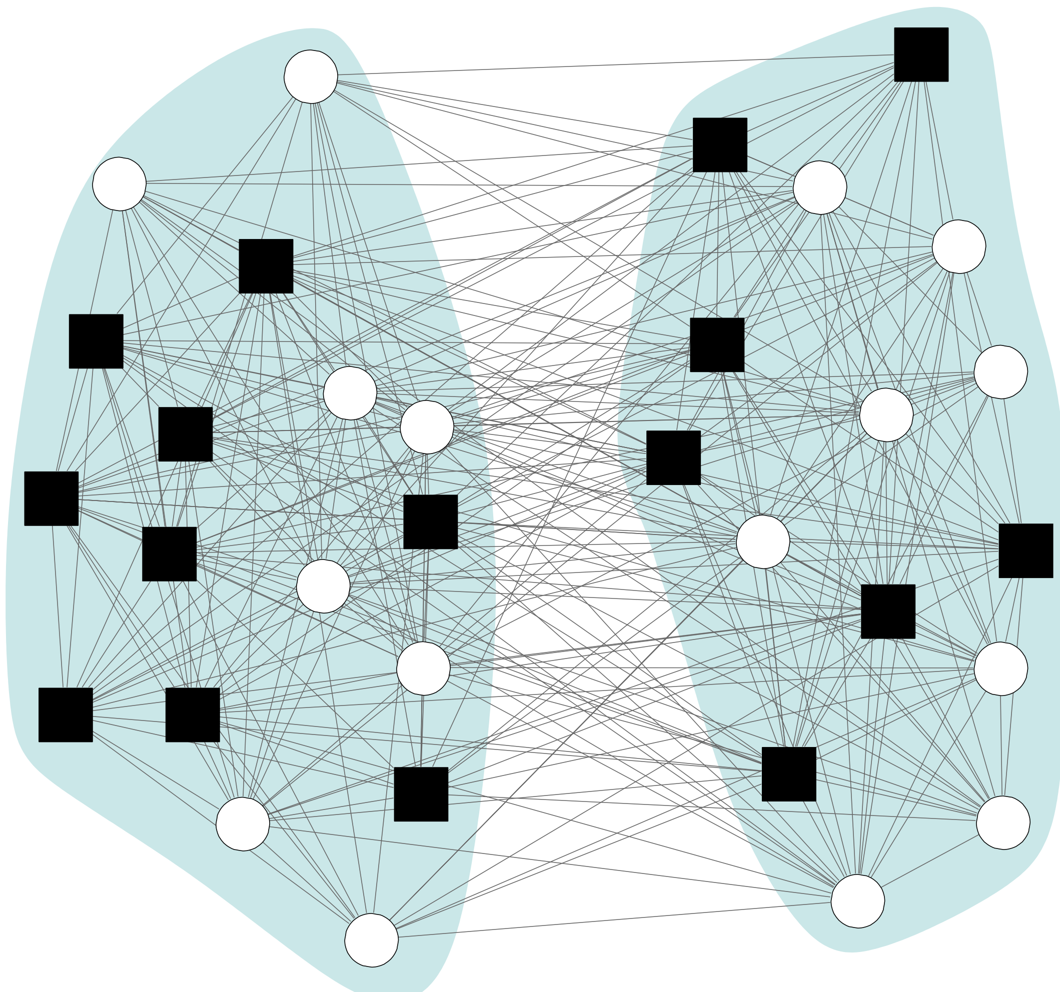
Again with the “Karate Club”



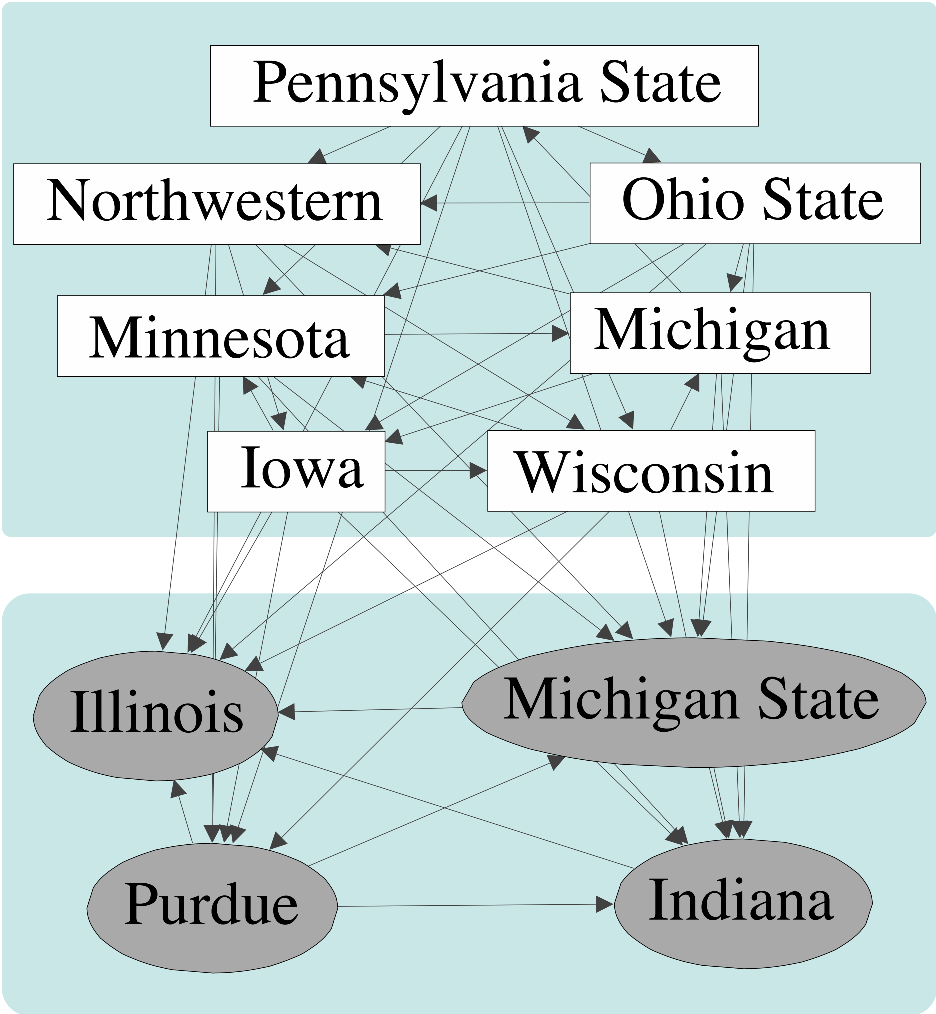
Communities with Edge Direction Bias



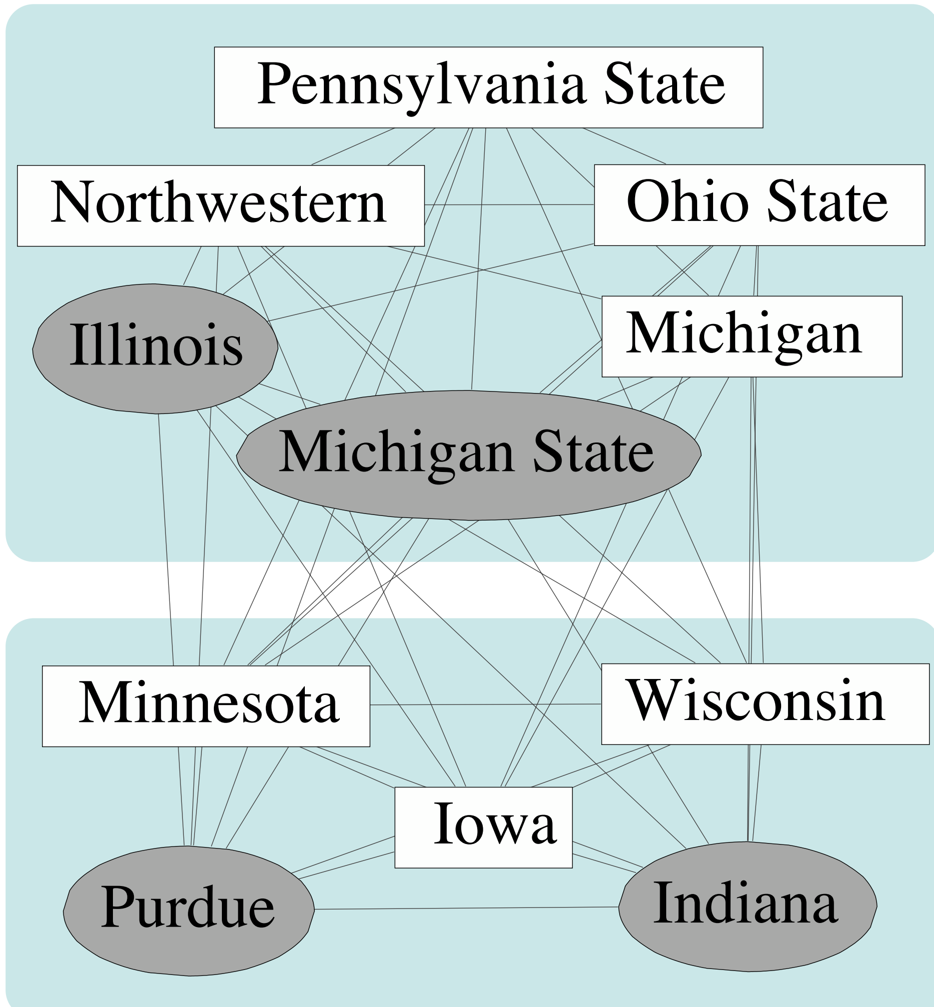
Communities with Edge Direction Bias



Edge Direction Bias in Real Networks



Edge Direction Bias in Real Networks

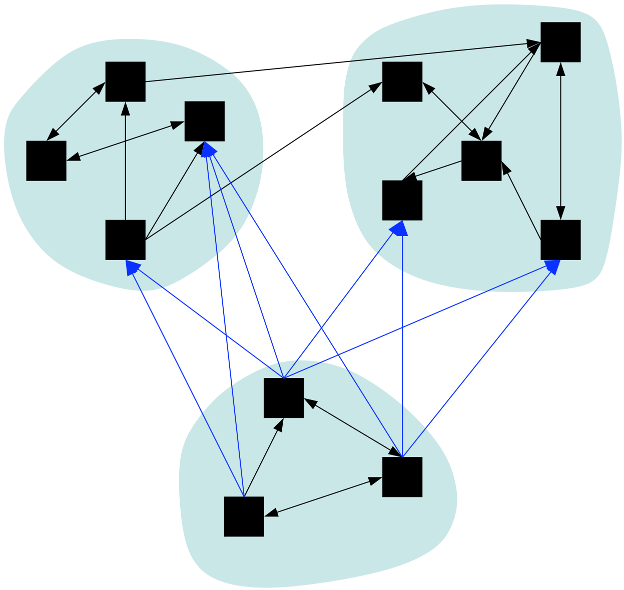


Exploratory Analysis of Structure in Networks

- Previously we identified communities in networks because we specifically sought a method to detect modules in networks.

- Reliance on specific measures of network structure where we are required to know the type of structure, for which we are looking, in advance can be limiting.

- We turn to probabilistic techniques and the Expectation Maximization (EM) Algorithm to identify general patterns of connection between vertices.



The Method

There are three types of quantities in this method of approach.

1. Observed data: the actual edges falling between pairs of vertices in a network (A).
2. Missing data: we assume that the vertices divide into c groups. We denote the group to which vertex i belongs as g_i and set of all missing data as g.
3. Model parameters: they describe the patterns of vertices in different groups (θ, π).

$\theta_{ri} =$ the probability that there exists an edge from a vertex in group r to a vertex i

$\pi_r =$ the probability of a vertex belonging to group r

$$\sum_{r=1}^{c} \pi_r = 1 \qquad\qquad \sum_{i=1}^{n} \theta_{ri} = 1$$

A Likelihood Problem

The likelihood of the data given the model is

$$\Pr(A, g|\pi, \theta) = \Pr(A|g, \pi, \theta) \Pr(g|\pi, \theta),$$

where

$$\Pr(A|g, \pi, \theta) = \prod_{ij} \theta_{g_j,i}^{A_{ij}} \text{ and } \Pr(g|\pi, \theta) = \prod_{j} \pi_{g_j}.$$

Frequently, one works not with the likelihood itself, but with the log-likelihood.

$$\mathcal{L} = \ln \Pr(A, g|\pi, \theta) = \sum_j \left[\ln \pi_{g_j} + \prod_i \theta_{g_j,i}^{A_{ij}}\right].$$

Dealing with the "Missing" Data

We cannot directly observe g.

It is, however, possible to calculate an expected value for the log-likelihood over all possible values of g.

$$\overline{\mathcal{L}} = \sum_{g_1=1}^{c} \cdots \sum_{g_n=1}^{c} \Pr(g|A,\theta,\pi) \sum_{j=1}^{n} \left[\ln \pi_{g_j} + \sum_{i=1}^{n} A_{ij} \ln \theta_{g_j,i} \right]$$

$$\overline{\mathcal{L}} = \sum_{r=1}^{c} \sum_{j=1}^{n} q_{rj} \left[\ln \pi_r + \sum_{i=1}^{n} A_{ij} \ln \theta_{ri} \right]$$

where

$$q_{jr} = \Pr(g_j = r|A,\pi,\theta) = \frac{\Pr(A, g_j = r|\pi,\theta)}{\Pr(A|\pi,\theta)} = \frac{\pi_r \prod_i \theta_{ri}^{A_{ij}}}{\sum_s \pi_s \prod_i \theta_{si}^{A_{ij}}}.$$

The EM Algorithm

- Initialize model parameters (θ, π) with random values.
- Find the probability a given vertex j is a member of group r (E-step).

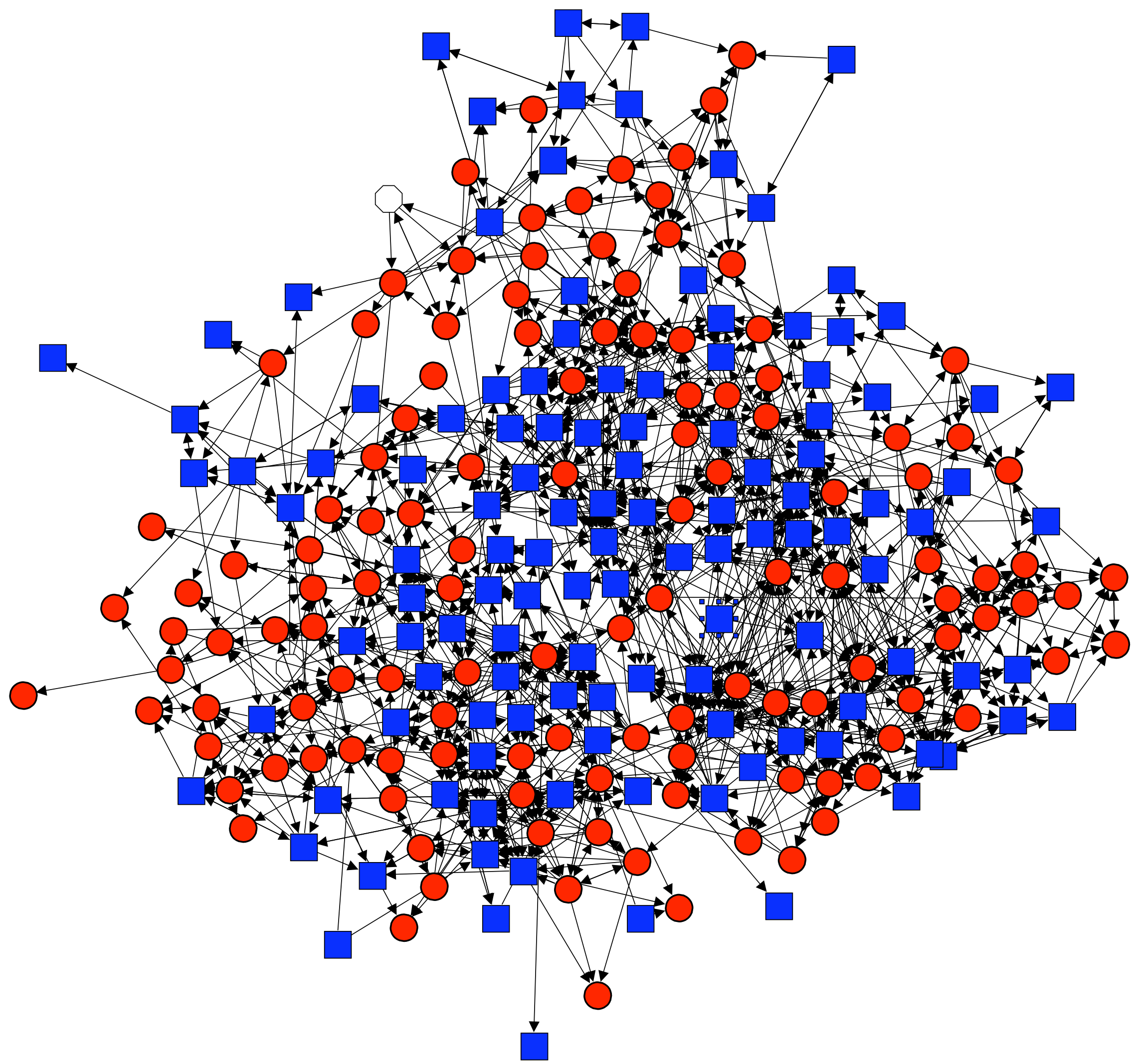
$$q_{jr} = \frac{\pi_r \prod_i \theta_{ri}^{A_{ij}}}{\sum_s \pi_s \prod_i \theta_{si}^{A_{ij}}}.$$

- Maximize the model parameters (M-step)

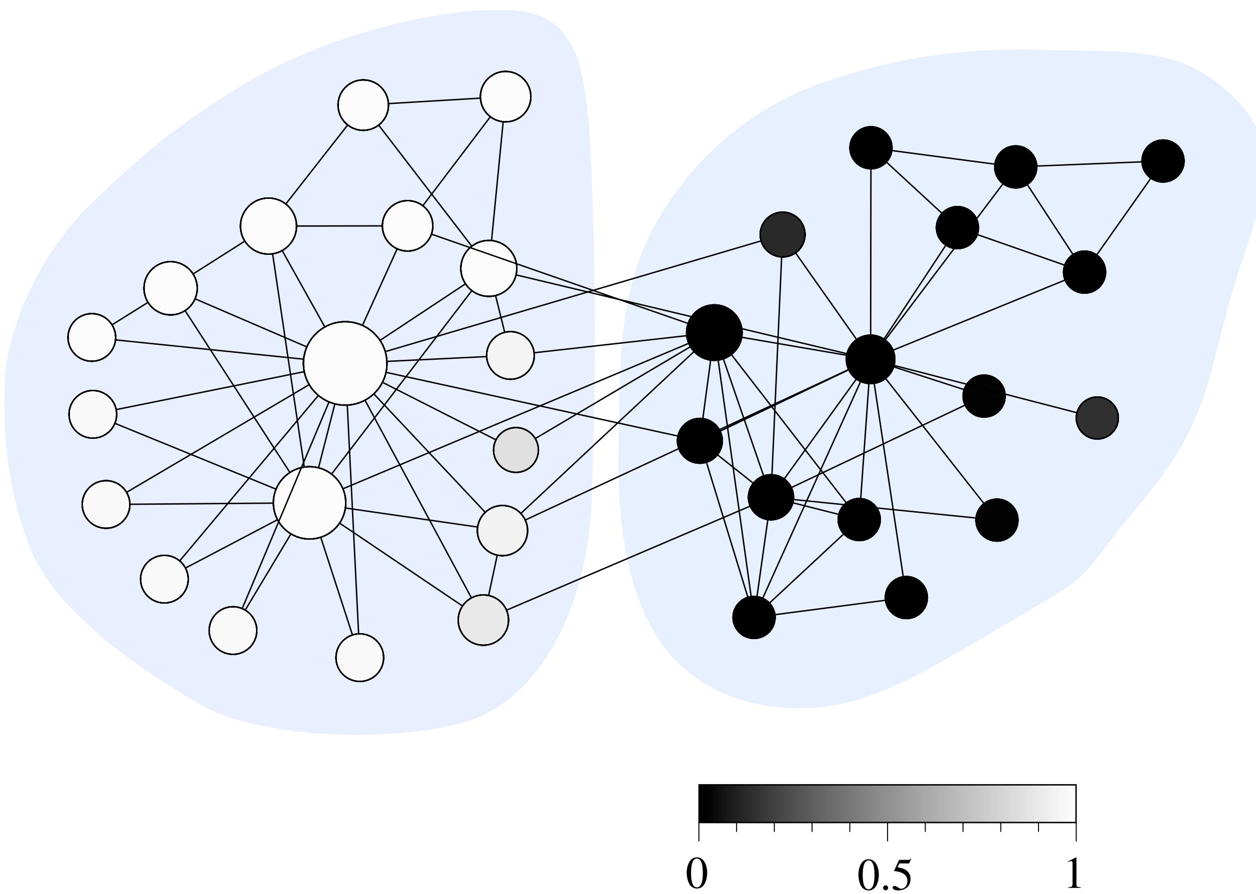
$$\pi_r = \frac{1}{n} \sum_i q_{ir} \qquad \theta_{ri} = \frac{\sum_j A_{ij} q_{jr}}{\sum_j k_j q_{jr}}.$$

- Iterate until convergence.

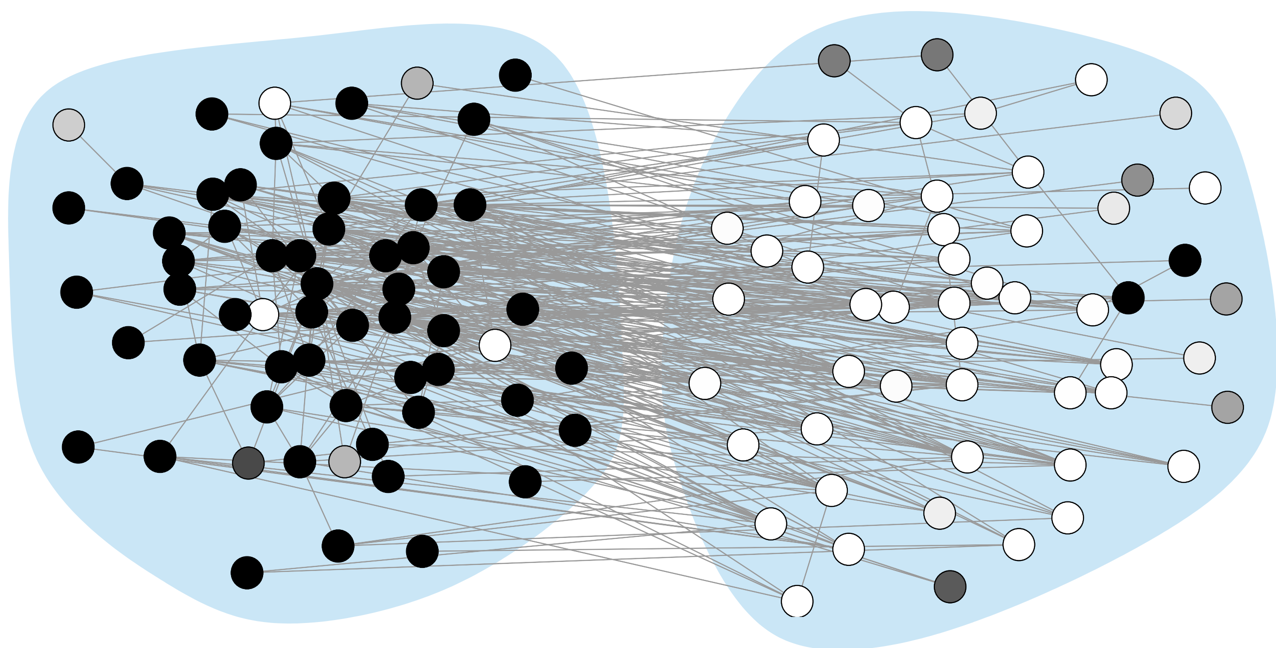
The AddHealth Network



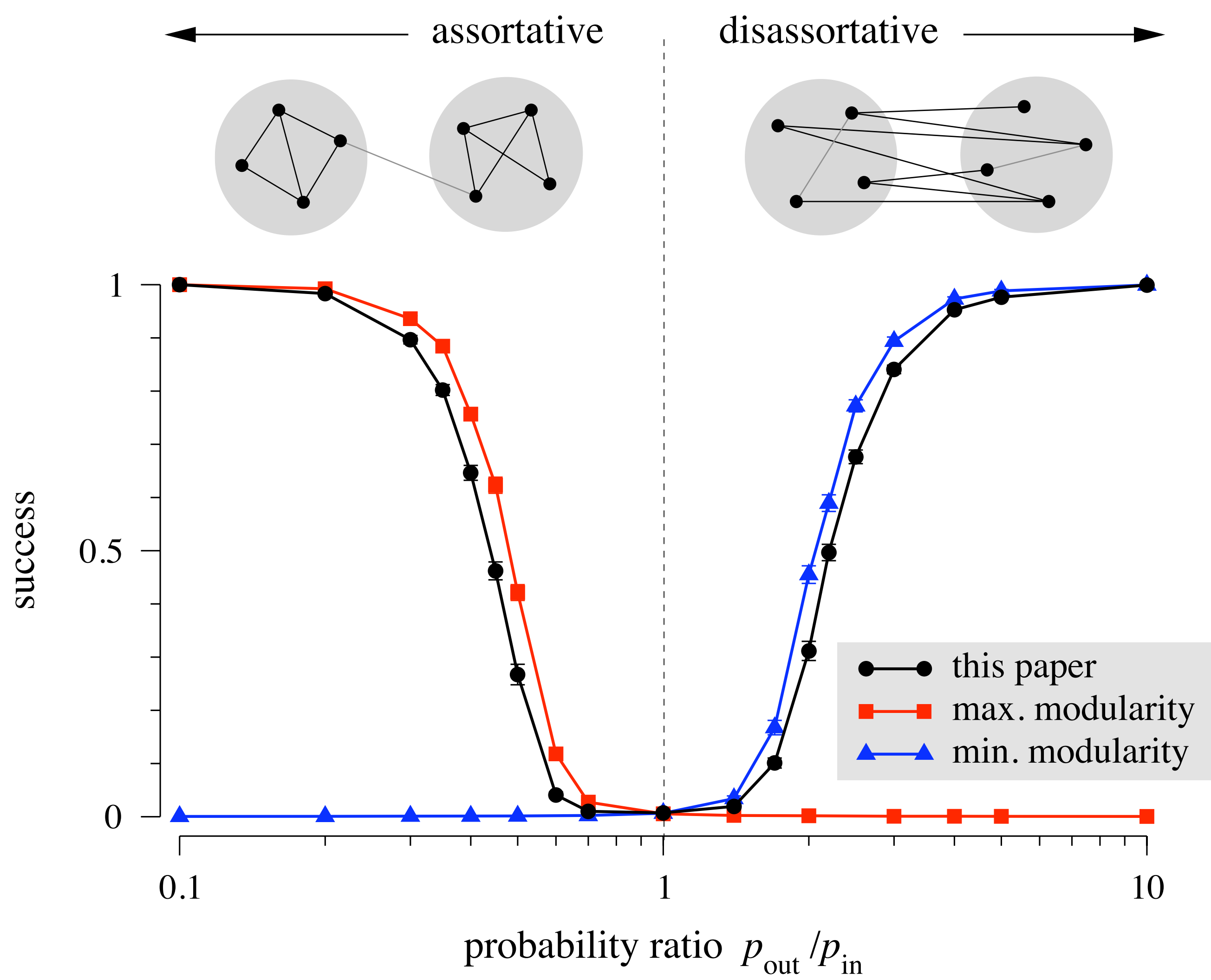
Example: Karate Club



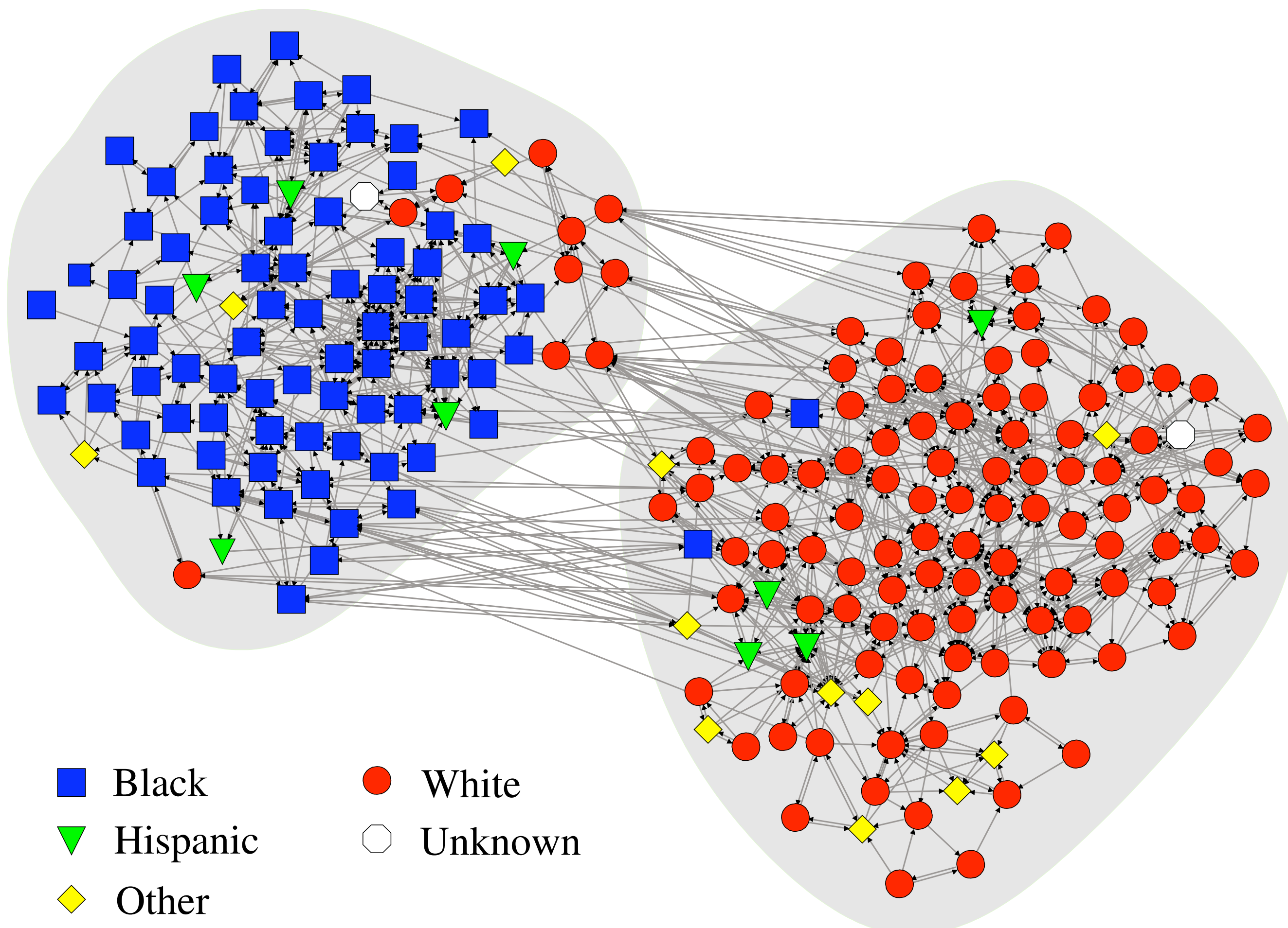
Example: Word Network



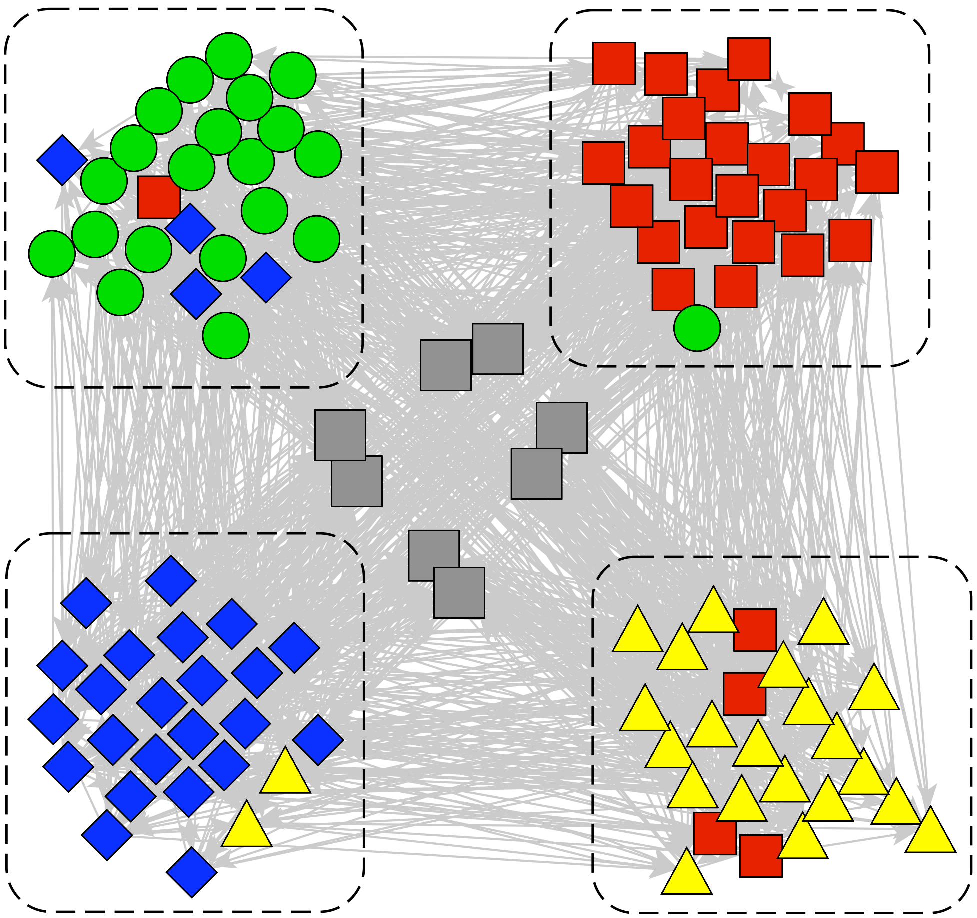
Example: Assortative/Dissassortative Network



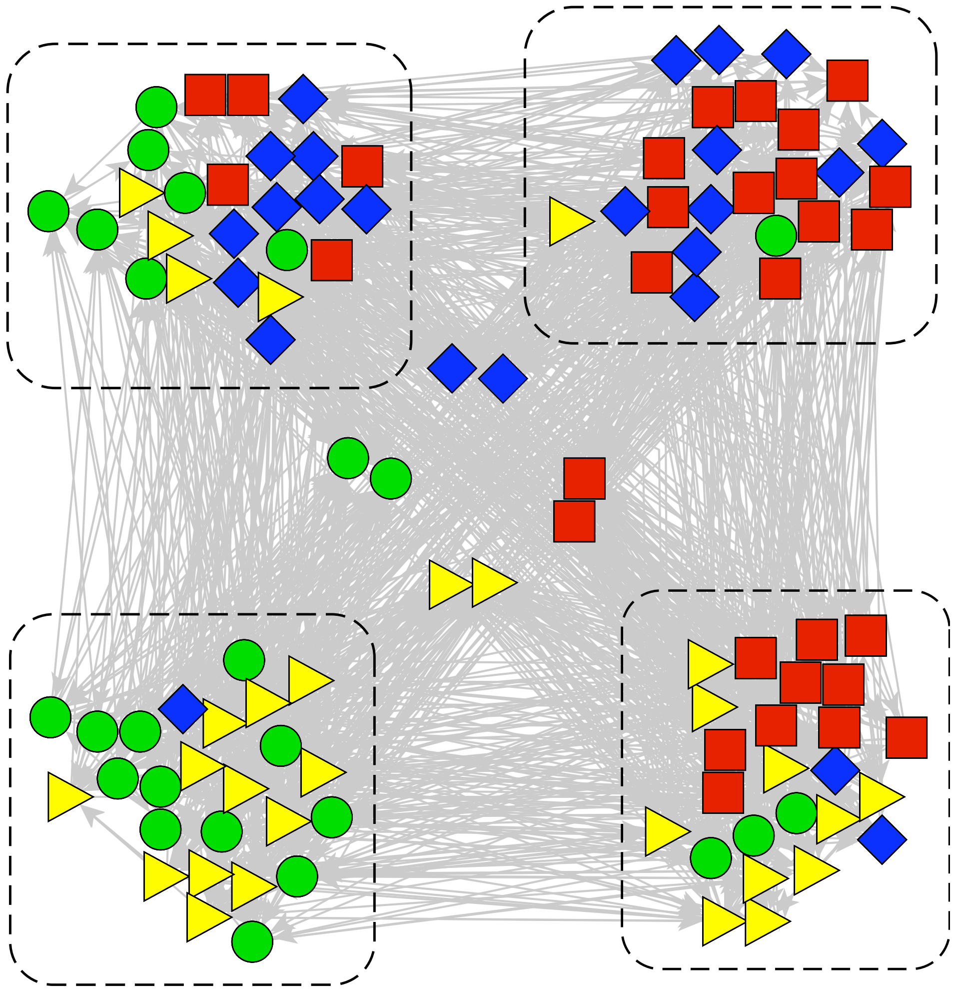
Example: AddHealth



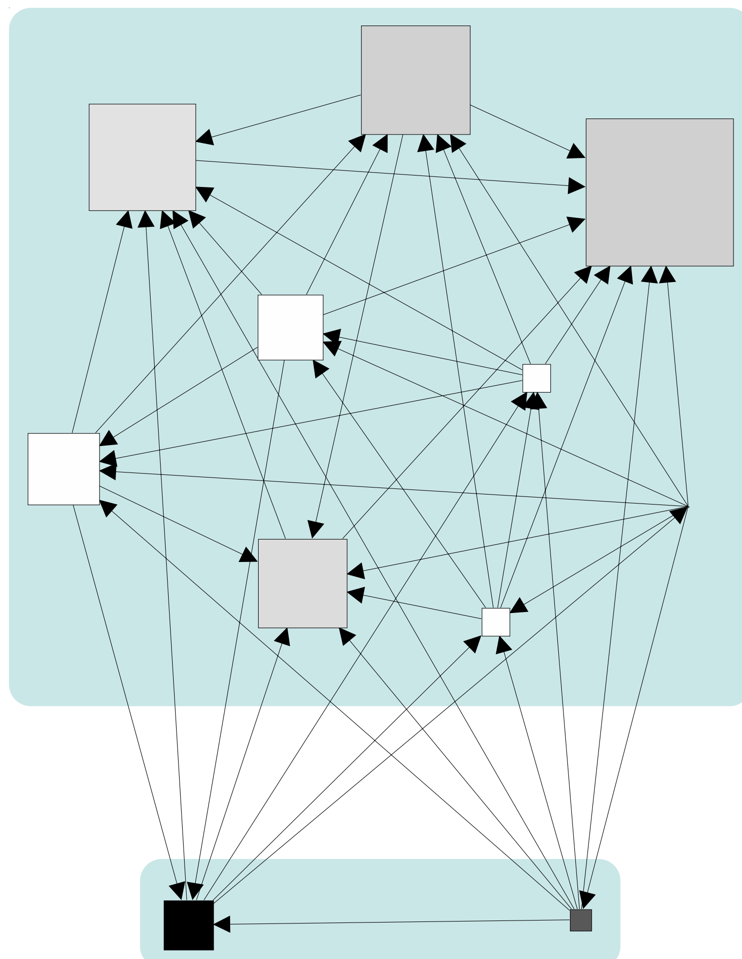
Example: A “Keystone” Network



Example: A “Keystone” Network

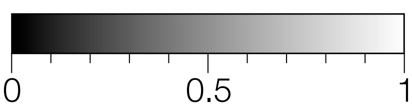


“Big Ten” Results with the EM Algorithm



Vertex size based on probability of being beaten by teams assigned to group 1.

$\mathbf{q}_{j1}$



Vertex shading based on probability of being assigned to group 1.

Example: The “Big Ten” Conference