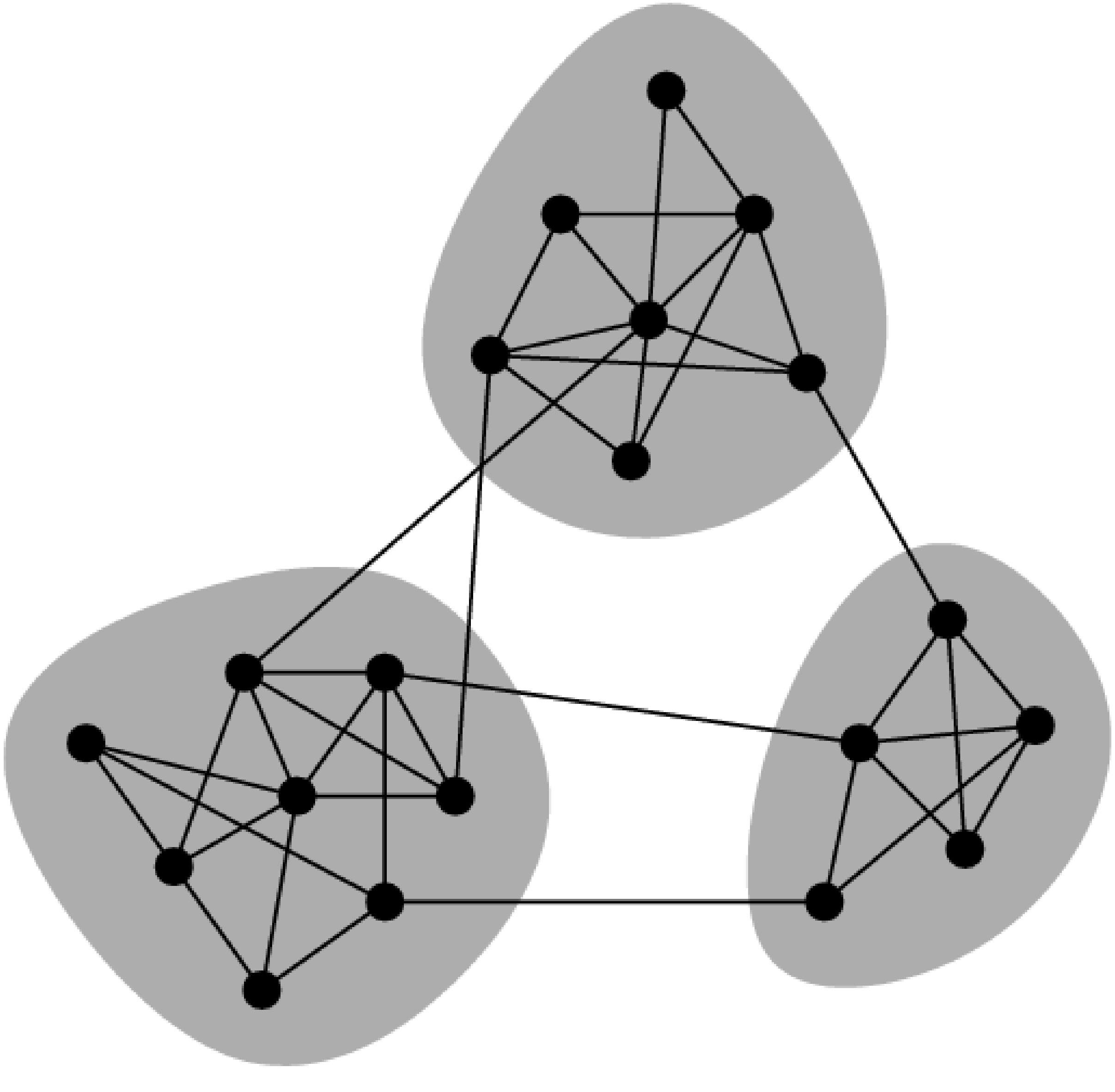
Complex Structures in Complex Networks

Mark Newman
University of Michigan

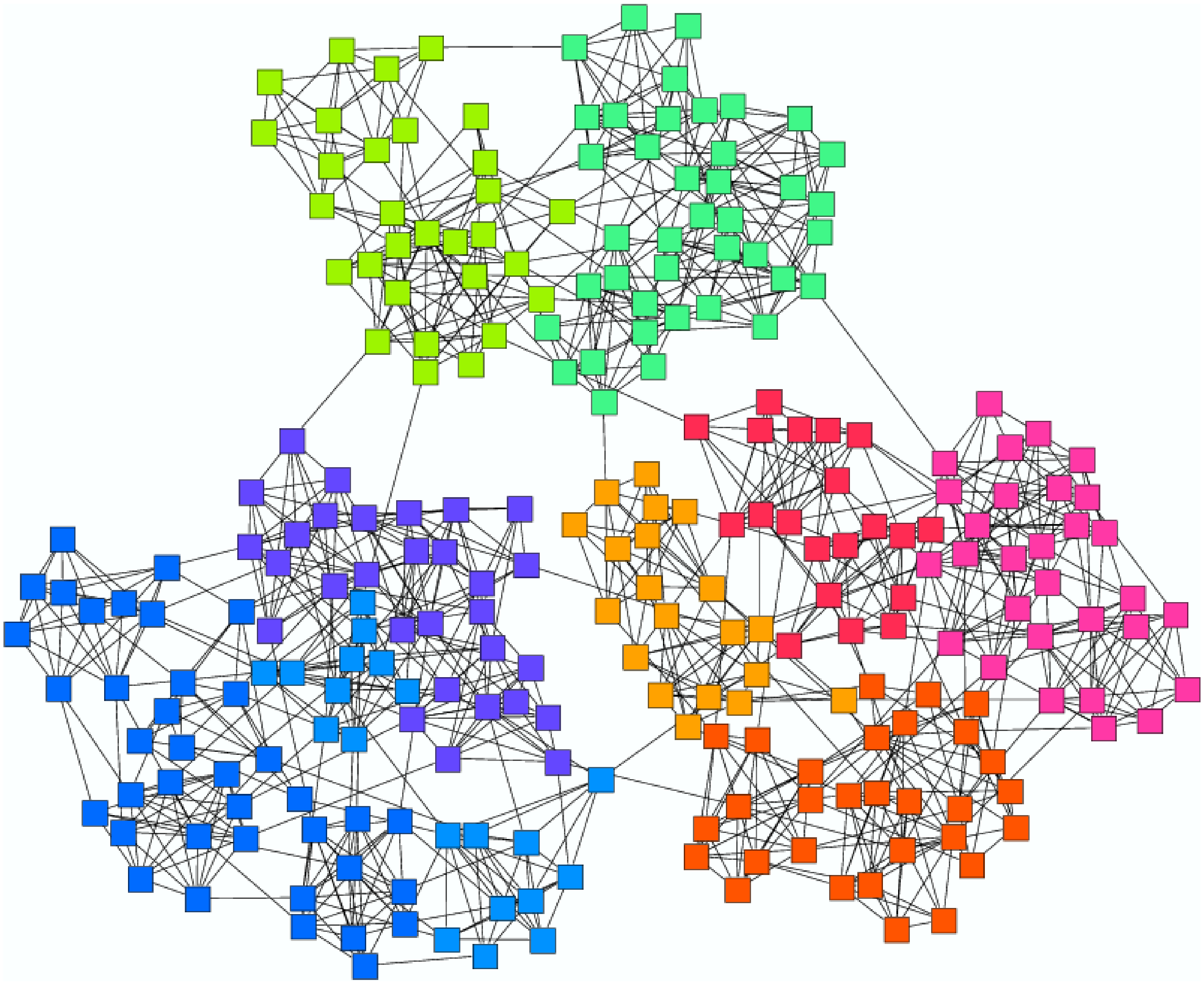
Joint work with Aaron Clauset & Cris Moore (SFI)
and Elizabeth Leicht (UC Davis)

Modules, groups, or communities

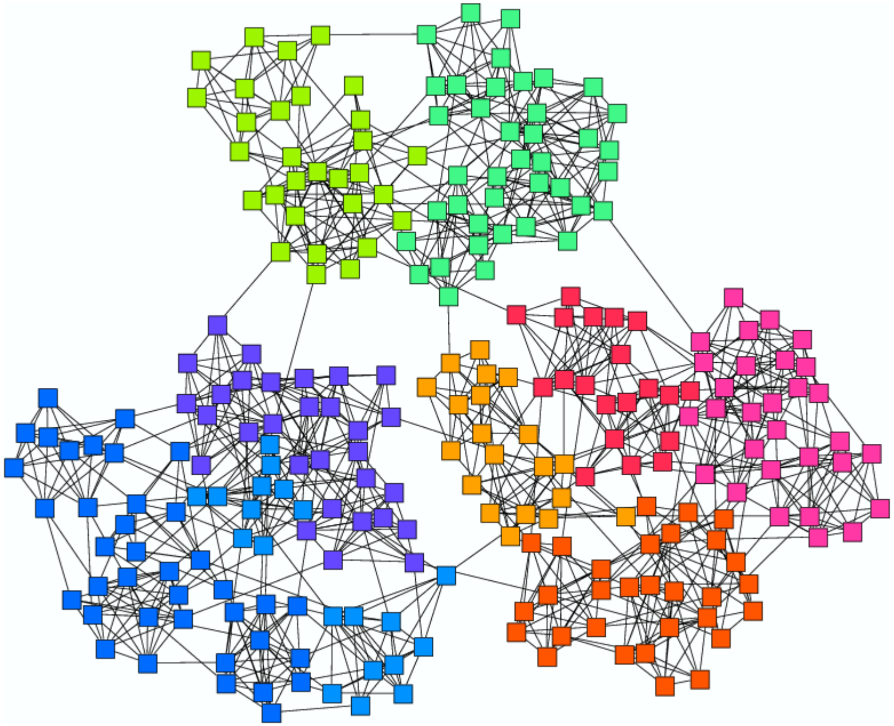


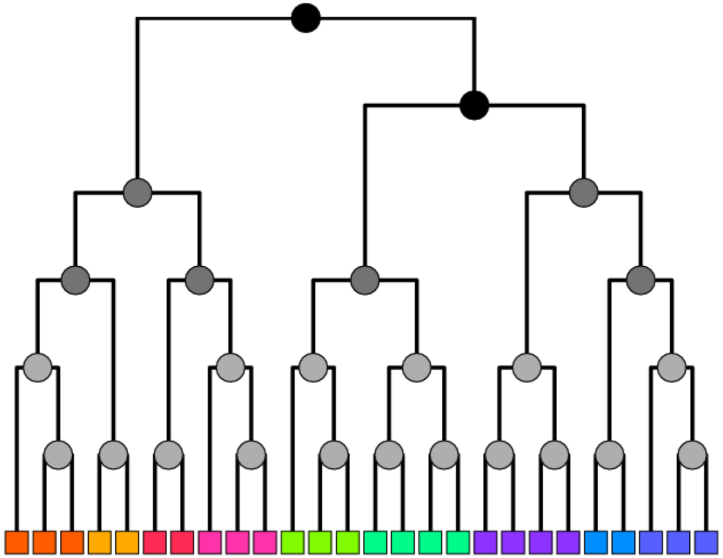
Network hierarchy

(Clauset, Moore, and Newman 2006, 2008)

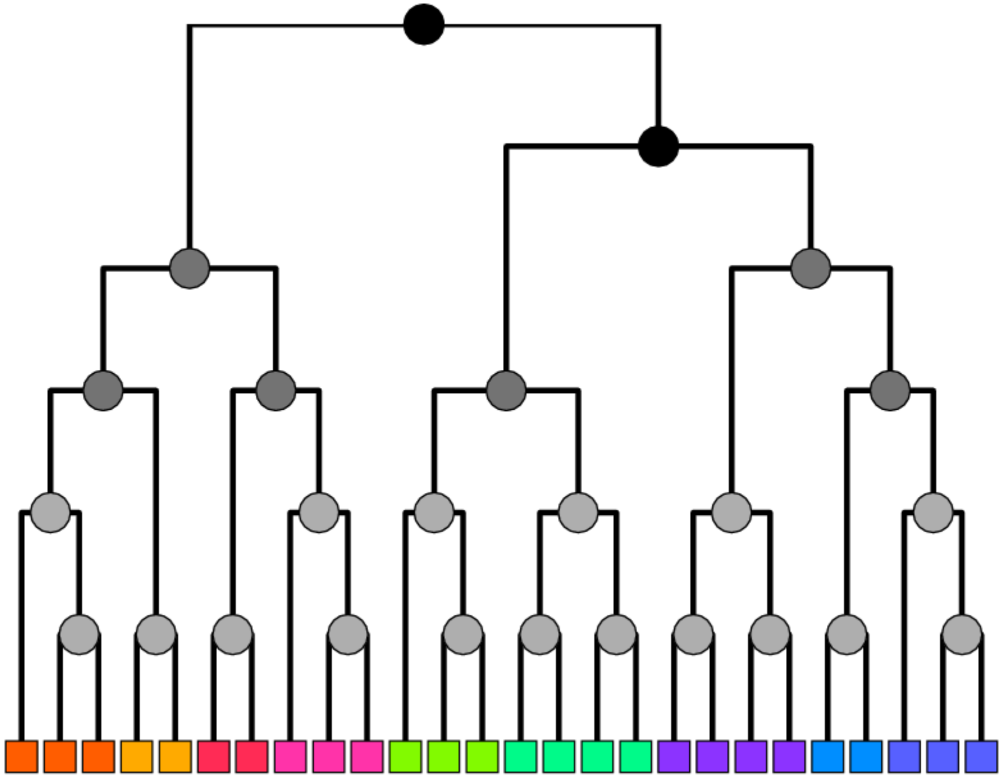


Network hierarchy





Network hierarchy



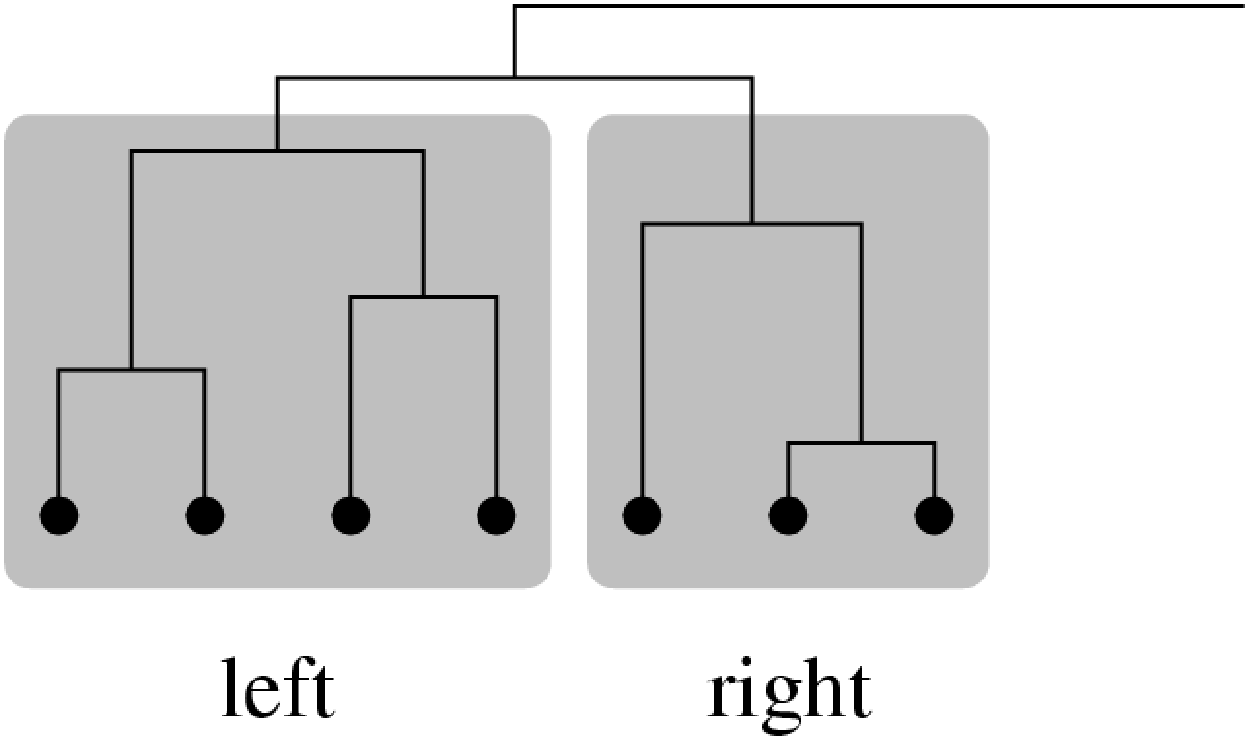
Let:

θ_i = probability of an edge

L_i = number of vertices in left subtree

R_i = number of vertices in right subtree

E_i = actual number of edges in between two subtrees

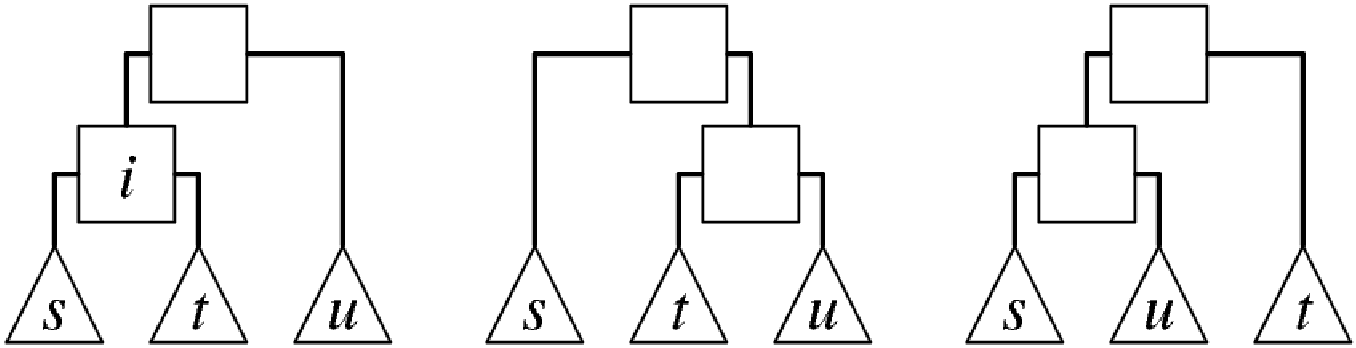


Likelihood of a network given a dendrogram and a set of probabilities is:

$$\mathcal{L}(\mathcal{D},\theta) = \prod_{i=1}^{n-1} \theta_i^{E_i} \left(1-\theta_i\right)^{L_i R_i - E_i} .$$

The maximum with respect to θ gives simply $\theta_i = E_i \,/(L_i\, R_i)$. The maximum with respect to the dendrogram structure is harder: we use Markov chain Monte Carlo to sample the configuration space.

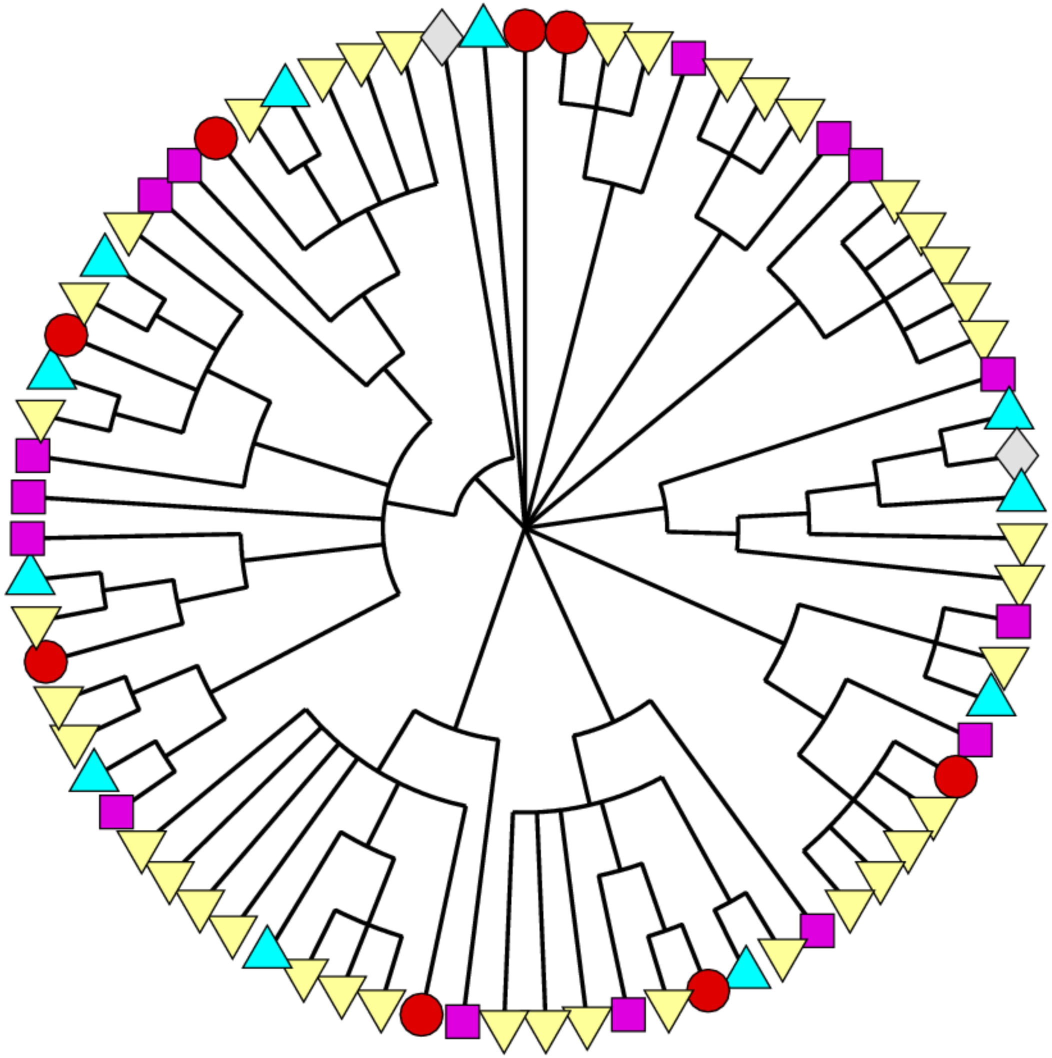
- Use standard methods borrowed from phylogenetic reconstruction:

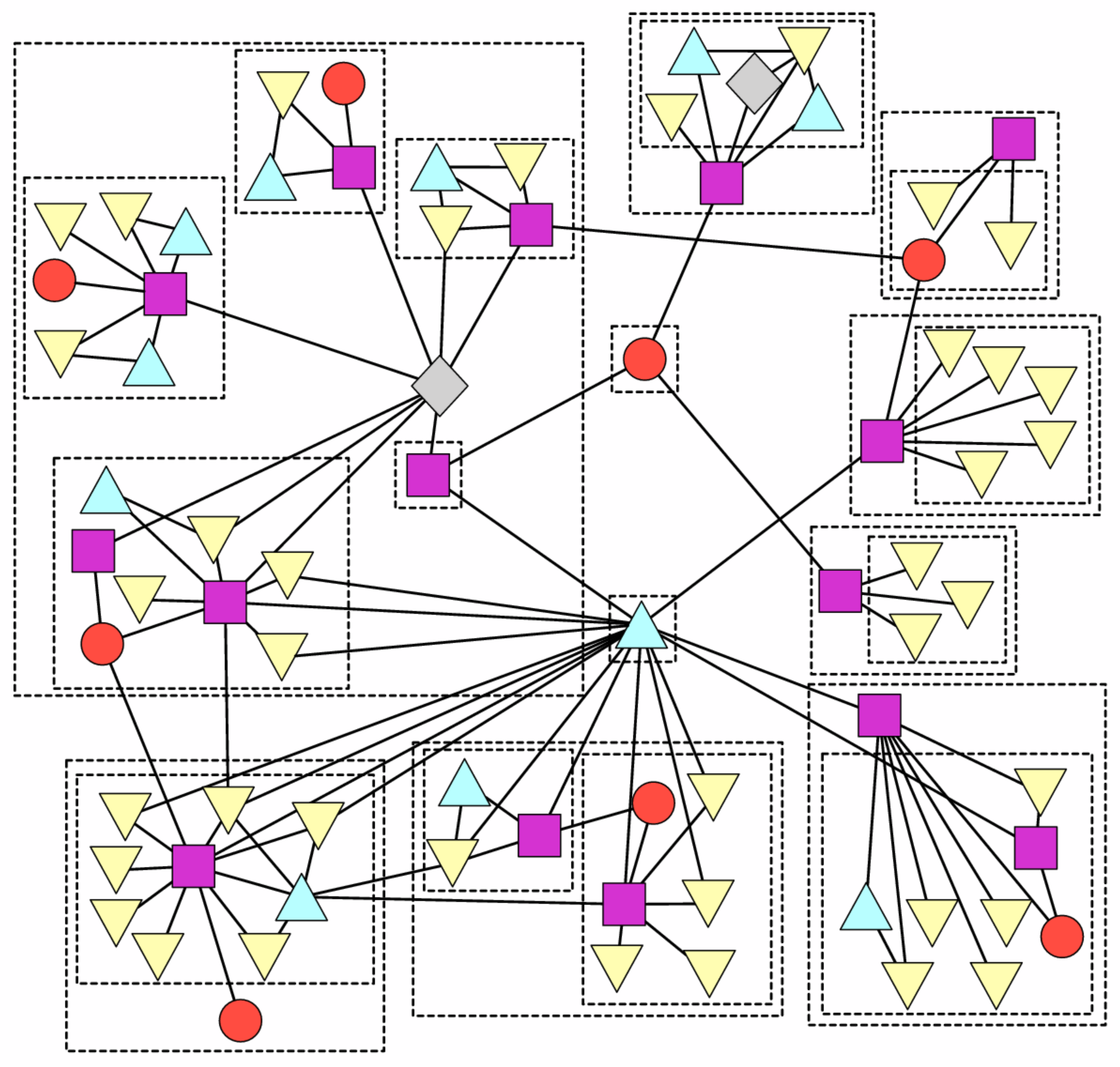


- Repeat as necessary:
  - Exchange subtrees
  - Calculate ratio of likelihoods
  - Accept/reject using the usual Metropolis-Hastings probability
- Reduce “temperature” to find max-likelihood tree

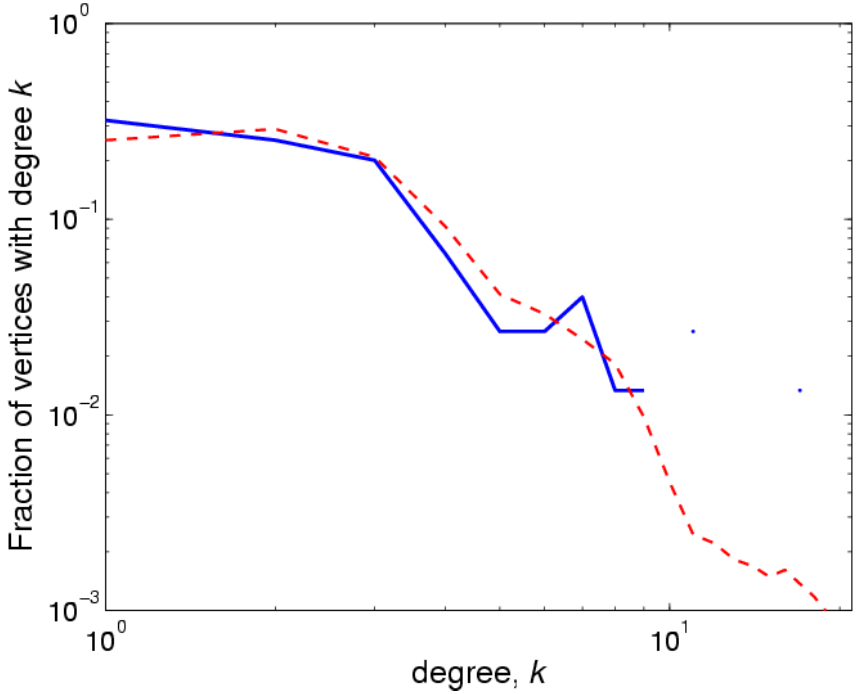
- But the real interest in the method is when we don't just look at the maximum likelihood tree
    - Many trees are competitive with the maximum likelihood tree
    - Real structure is captured not by one tree, but by the *distribution* over possible trees
    - The Monte Carlo method automatically generates this distribution and with this we can do many things. . .

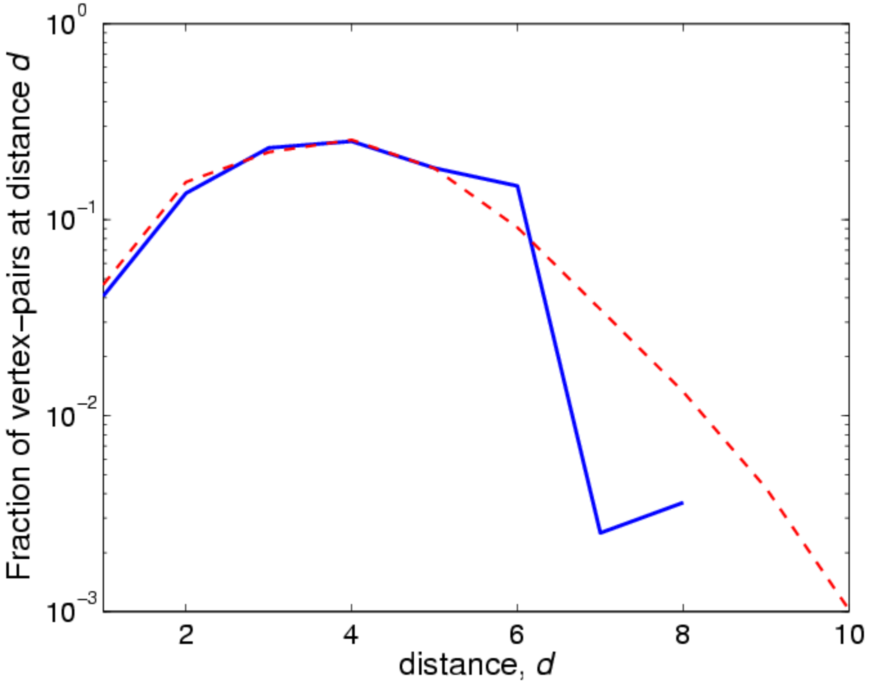
- Generate consensus hierarchies:





- Perform network "generalization", i.e., generate new networks from the model that are not the same as the original but are statistically similar

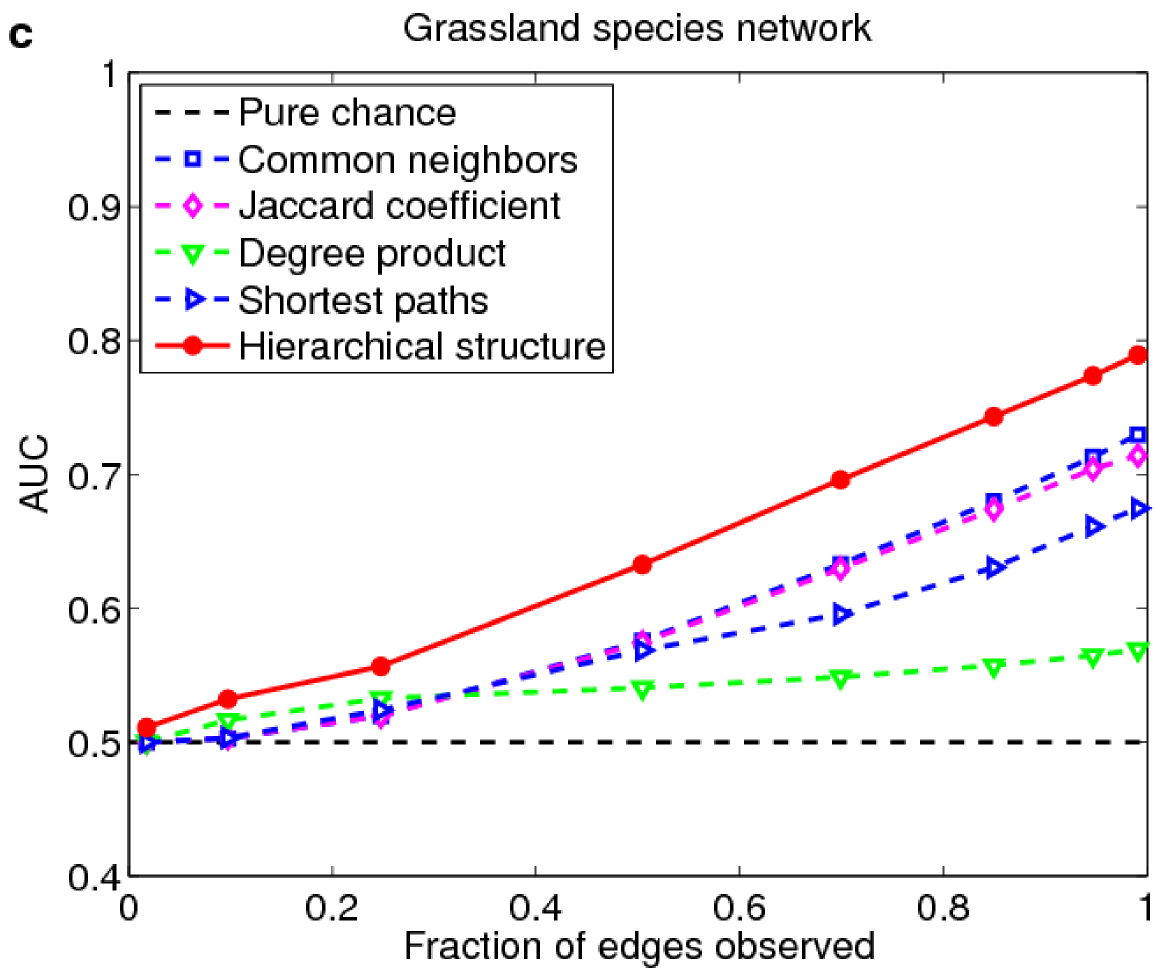


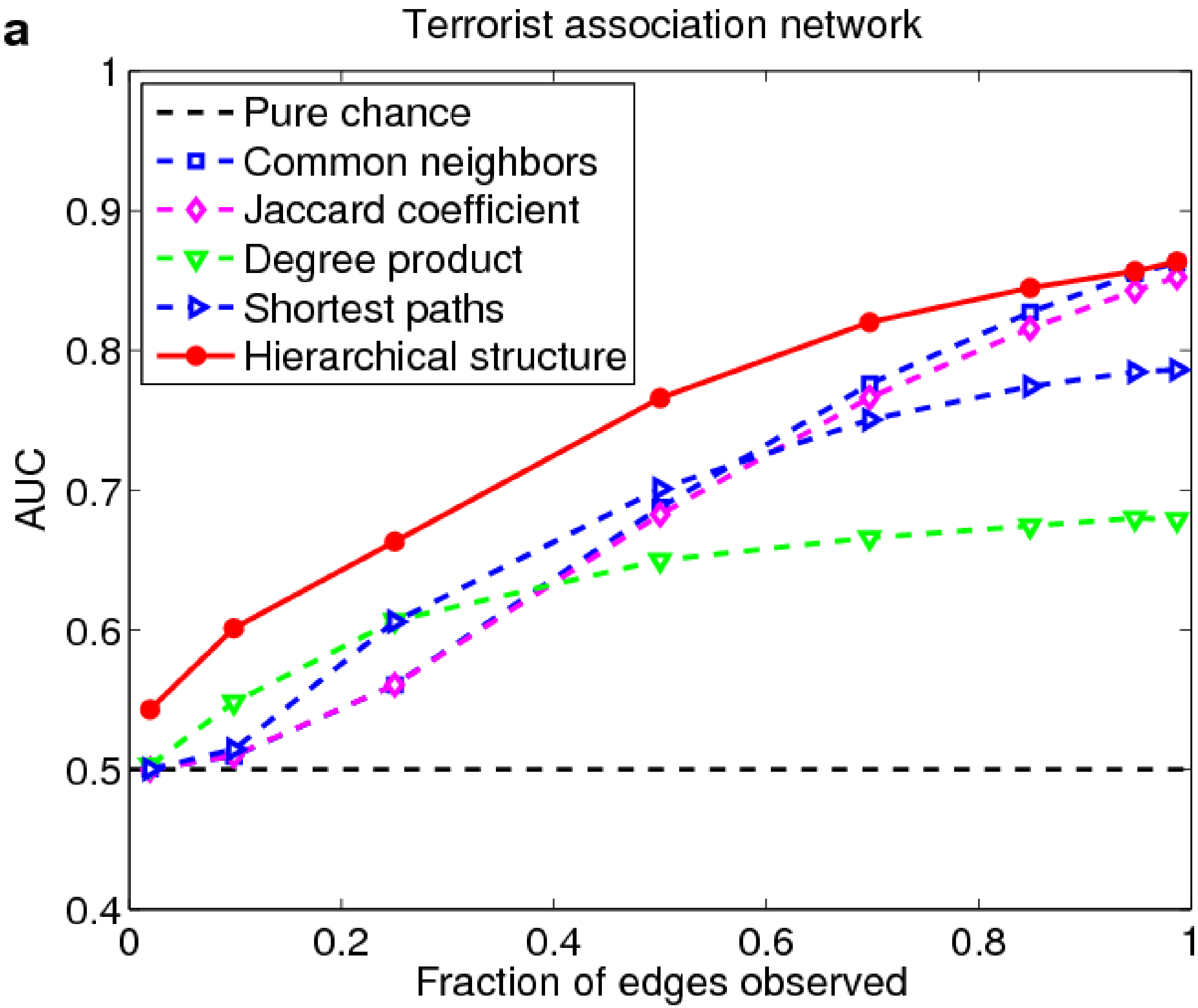


- Learn which edges are probable and which are improbable, which are "surprising"

Link prediction

- Find vertex pairs that have high probability of connection, but that are not actually connected:

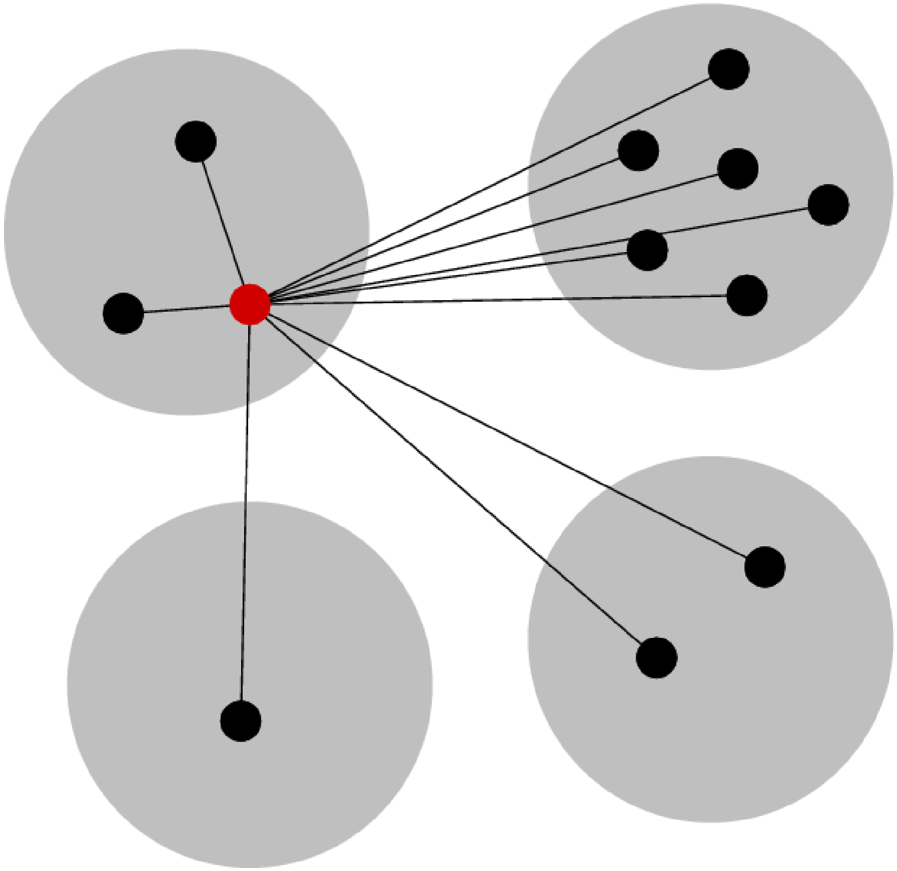



a
Terrorist association network
Pure chance
Common neighbors
Jaccard coefficient
Degree product
Shortest paths
Hierarchical structure
AUC
1
0.9
0.8
0.7
0.6
0.5
0.4
0
0.2
0.4
0.6
0.8
1
Fraction of edges observed

Vertex classification

(Newman and Leicht 2007)

- We specify a very broad set of possible structures that we are interested in:



Definition of the model

- There are three kinds of quantities in this approach:
  - Observed data: the pattern of edges observed between the vertices. These are given to us by the experimenter.
  - Missing data: We assume that the vertices divide into c groups. We denote the group to which vertex i belongs by g_i. These are missing data.
  - Model parameters: these describe the patterns of connection between vertices in different groups.

Definition of the model

Directed case:

$$\pi_r = \text{probability of being in group } r$$

and

$$\theta_{ri} = \text{probability of a link to vertex } i$$

These satisfy

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

Likelihood and log-likelihood

- The likelihood is

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

- Here

$$\Pr(A|g,\pi,\theta) = \prod_{ij} \theta_{g_i,j}^{A_{ij}}, \quad \Pr(g|\pi,\theta) = \prod_i \pi_{g_i}$$

- So

$$\Pr(A,g|\pi,\theta) = \prod_i \left[\pi_{g_i} \prod_j \theta_{g_i,j}^{A_{ij}} \right]$$

$$\mathcal{L} = \ln \Pr(A,g|\pi,\theta) = \sum_i \left[\ln \pi_{g_i} + \sum_j A_{ij} \ln \theta_{g_i,j} \right]$$

- Unfortunately, we don't know the values of the missing data, so we can't evaluate this expression
- However, we can make a pretty good guess at the values of the missing data if we know A, π, and θ. More specifically, we can calculate the probability that g_i takes a particular value r thus:

$$q_{ir} = \Pr(g_i = r | A, \pi, \theta) = \frac{\Pr(A, g_i = r | \pi, \theta)}{\Pr(A | \pi, \theta)}.$$

- The numerator we can calculate by summing Pr(A,g | π,θ) over all the gs except g_i
- The denominator is fixed by the normalization

- The result is:

$$q_{ir} = \frac{\pi_r \prod_j \theta_{rj}^{A_{ij}}}{\sum_s \pi_s \prod_j \theta_{sj}^{A_{ij}}}.$$

- This looks odd: we're saying you can calculate q_{ir} given the model and the data, and then we're going to calculate the model from q_{ir} and the data?
- Yes, but we have to do it self-consistently. . .

Expected likelihood

- We can now make a guess about the value of the log-likelihood. Our best guess is just the expectation value:

$$
\begin{aligned}
\overline{\mathcal{L}} &= \sum_{g_1=1}^{c} \ldots \sum_{g_n=1}^{c} \Pr(g|A,\pi,\theta) \sum_i \Big[\ln \pi_{g_i} + \sum_j A_{ij} \ln \theta_{g_i,j}\Big] \\
&= \sum_{ir} \Pr(g_i = r|A,\pi,\theta) \Big[\ln \pi_r + \sum_j A_{ij} \ln \theta_{rj}\Big] \\
&= \sum_{ir} q_{ir} \Big[\ln \pi_r + \sum_j A_{ij} \ln \theta_{rj}\Big].
\end{aligned}
$$

- Now it's a straightforward matter to maximize this with respect to π and θ to find the best values. The result is:

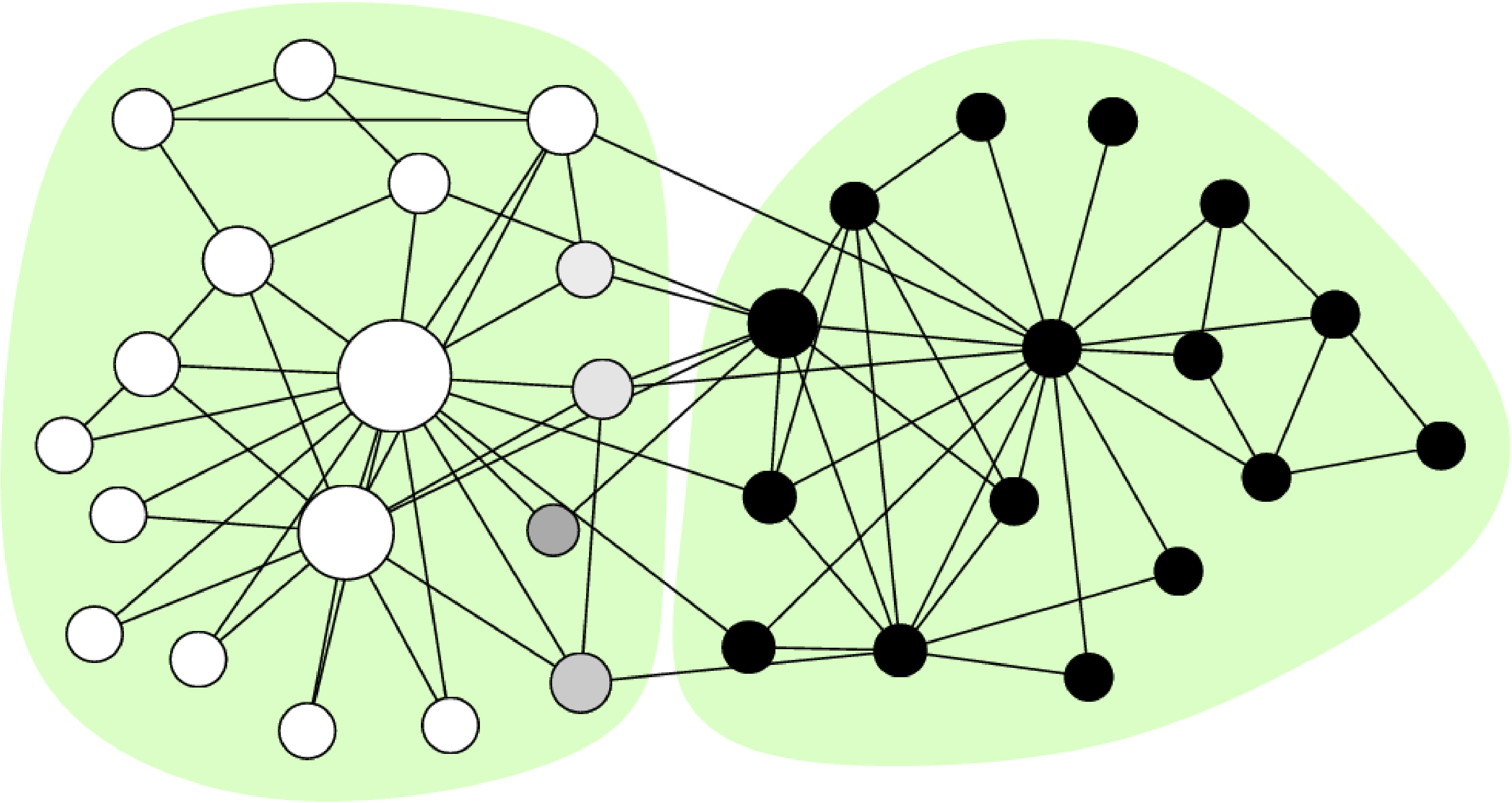
$$\pi_r = \frac{1}{n}\sum_i q_{ir}, \qquad \theta_{rj} = \frac{\sum_i A_{ij} q_{ir}}{\sum_i k_i q_{ir}},$$

- So we have π and θ in terms of *q* and we have *q* in terms of π and θ
- To find a self-consistent solution to both sets of equations, we iterate from a suitable set of starting values

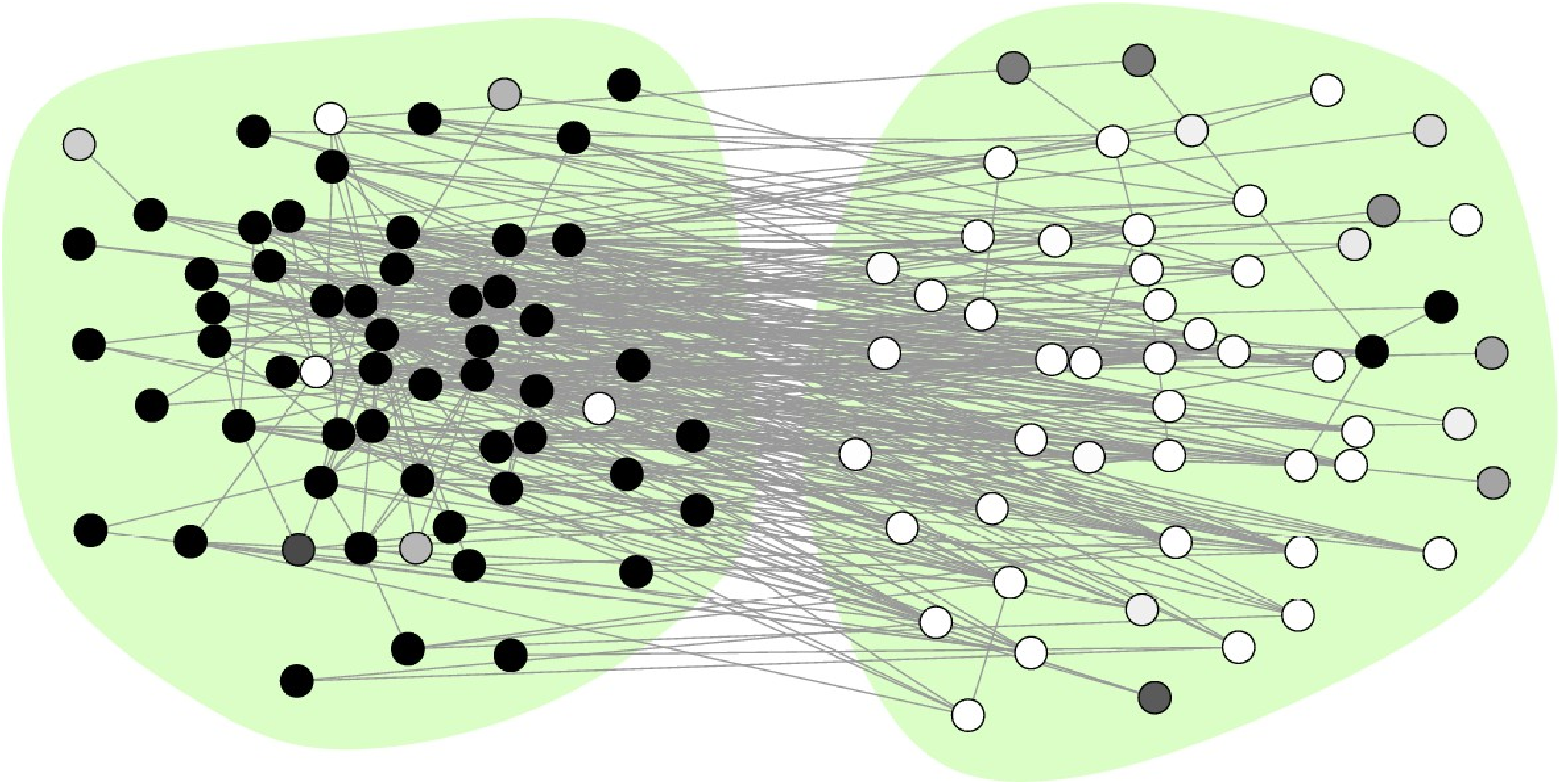
Expectation-Maximization Algorithm

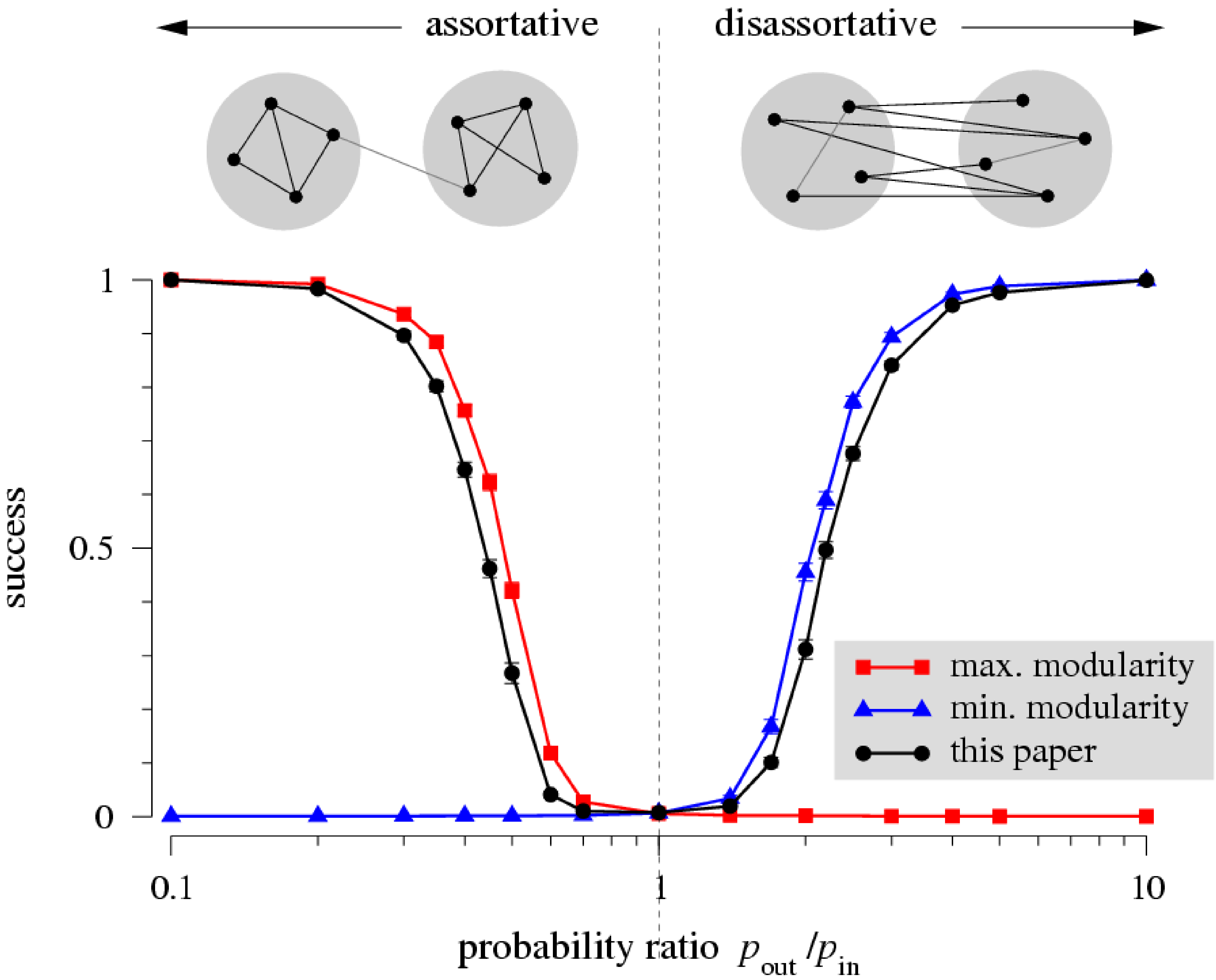
- Has a number of clear advantages:
  - Very simple: just a few lines of computer code to implement the method
  - Fast: typically only a few seconds to analyze even a large network
  - Simultaneously tells us how to group the vertices in the network and what the appropriate definition is for the groups
- Derivation is more complicated for undirected case, but the final equations are exactly the same

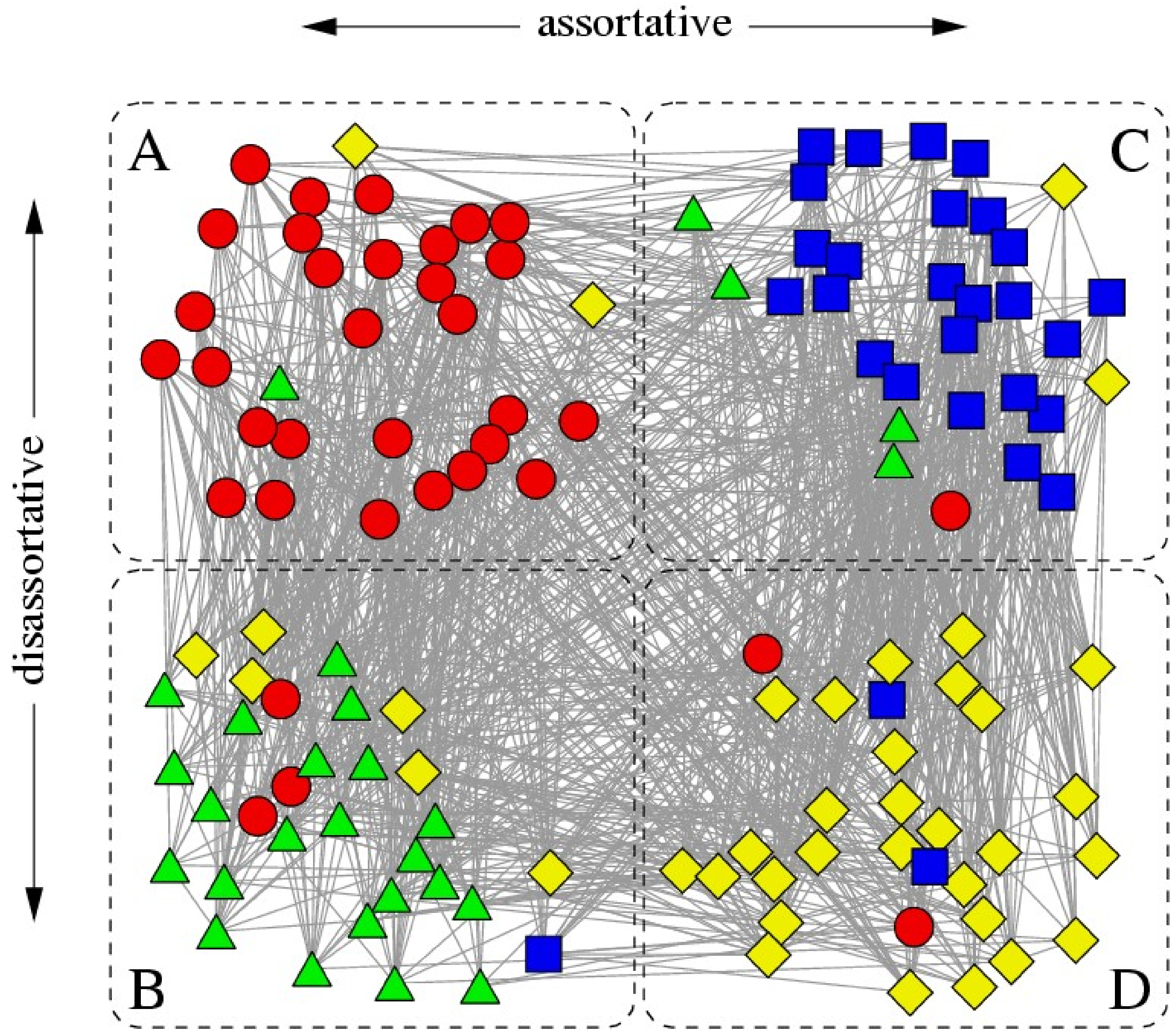
Example: Social network

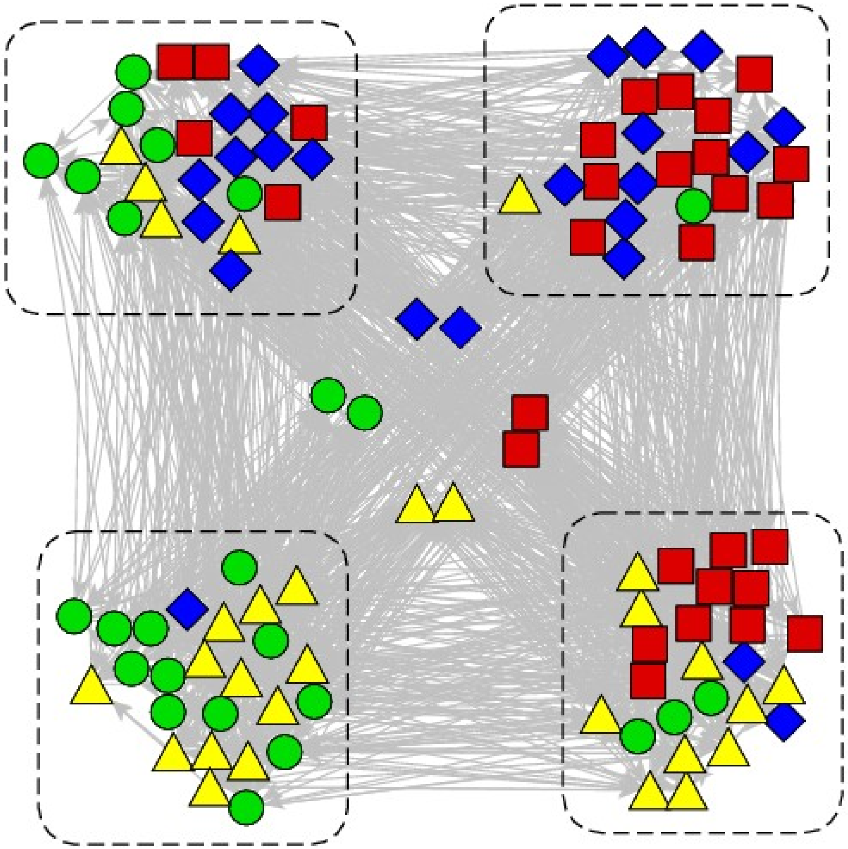


Example: Lexical network

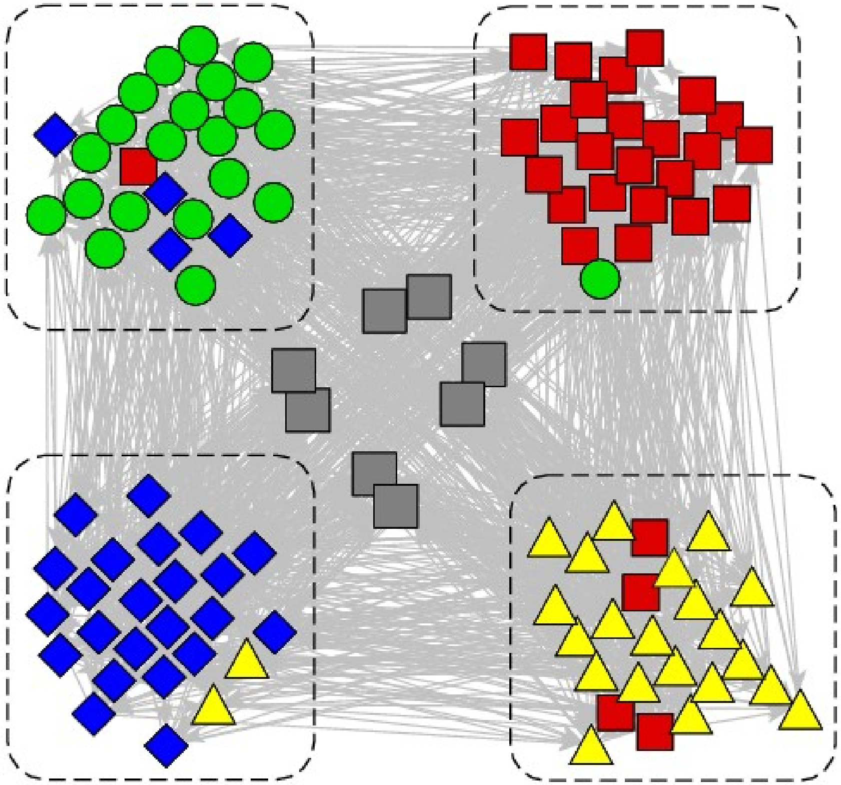



assortative
disassortative
1
0.5
0
success
0.1
1
10
probability ratio p_{out}/p_{in}
max. modularity
min. modularity
this paper


assortative
disassortative
A
C
B
D



Ordinary community detection



EM algorithm

- References:

  - A. Clauset, C. Moore, and M. E. J. Newman, *Nature* **453**, 98–101 (2008)
  - A. Clauset, C. Moore, and M. E. J. Newman in *Proceedings of the 23rd International Conference on Machine Learning*, ACM, New York (2006)
  - M. E. J. Newman and E. A. Leicht, *Proc. Natl. Acad. Sci.* **104**, 9564–9569 (2007)

- Thanks to:

  - NSF DMS and McDonnell Foundation for funding
  - Marian Boguña, Jennifer Dunne, and Kerby Shedden for useful input and for sharing data