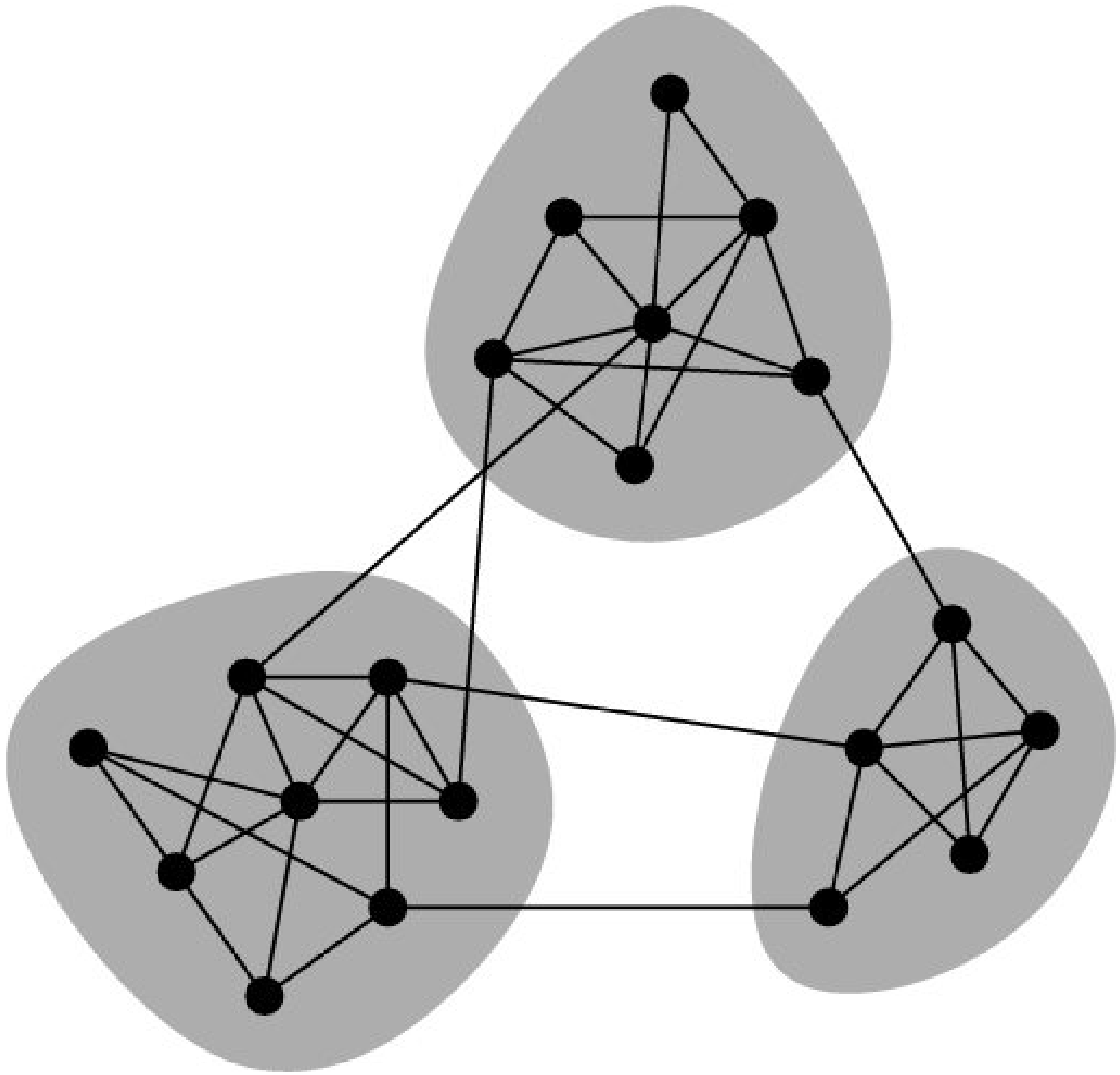
Large-scale Structure in Networks

Mark Newman
University of Michigan

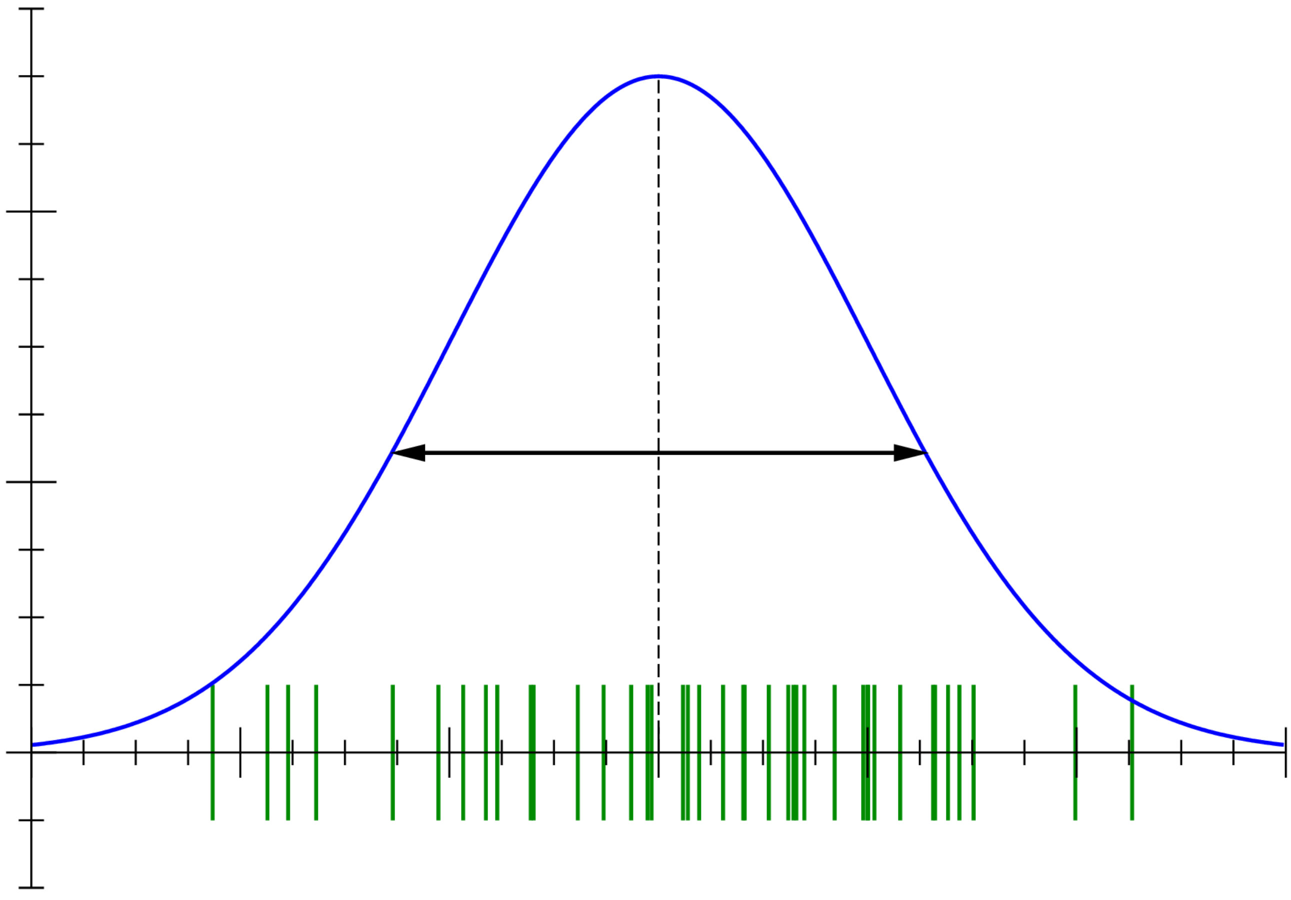
Work in collaboration with
Brian Ball, Aaron Clauset, Brian Karrer, Cristopher Moore and Lenka Zdeborová

Modules, groups, or communities



Statistical inference

- Suppose we have measured a set of n numbers x_i which we believe to be drawn from a normal distribution:



- The probability of making a measurement is

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- The probability of measuring the whole set is

$$\prod_{i=1}^{n} P(x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$$

$$= \left(2\pi\sigma^2\right)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2\right)$$

- This is the *likelihood* of the data. Its logarithm is

$$\mathscr{L} = -\tfrac{1}{2}n\log 2\pi - \tfrac{1}{2}n\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i-\mu)^2$$

- We take the log-likelihood

$$\mathscr{L} = -\tfrac{1}{2} n \log 2\pi - \tfrac{1}{2} n \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2$$

and differentiate with respect to μ:

$$\frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu) = \frac{1}{\sigma^2} \left[-n\mu + \sum_{i=1}^{n} x_i \right] = 0$$

- Hence

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

- Similarly, take

$$\mathscr{L} = -\tfrac{1}{2} n \log 2\pi - \tfrac{1}{2} n \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2$$

and differentiate with respect to σ^2:

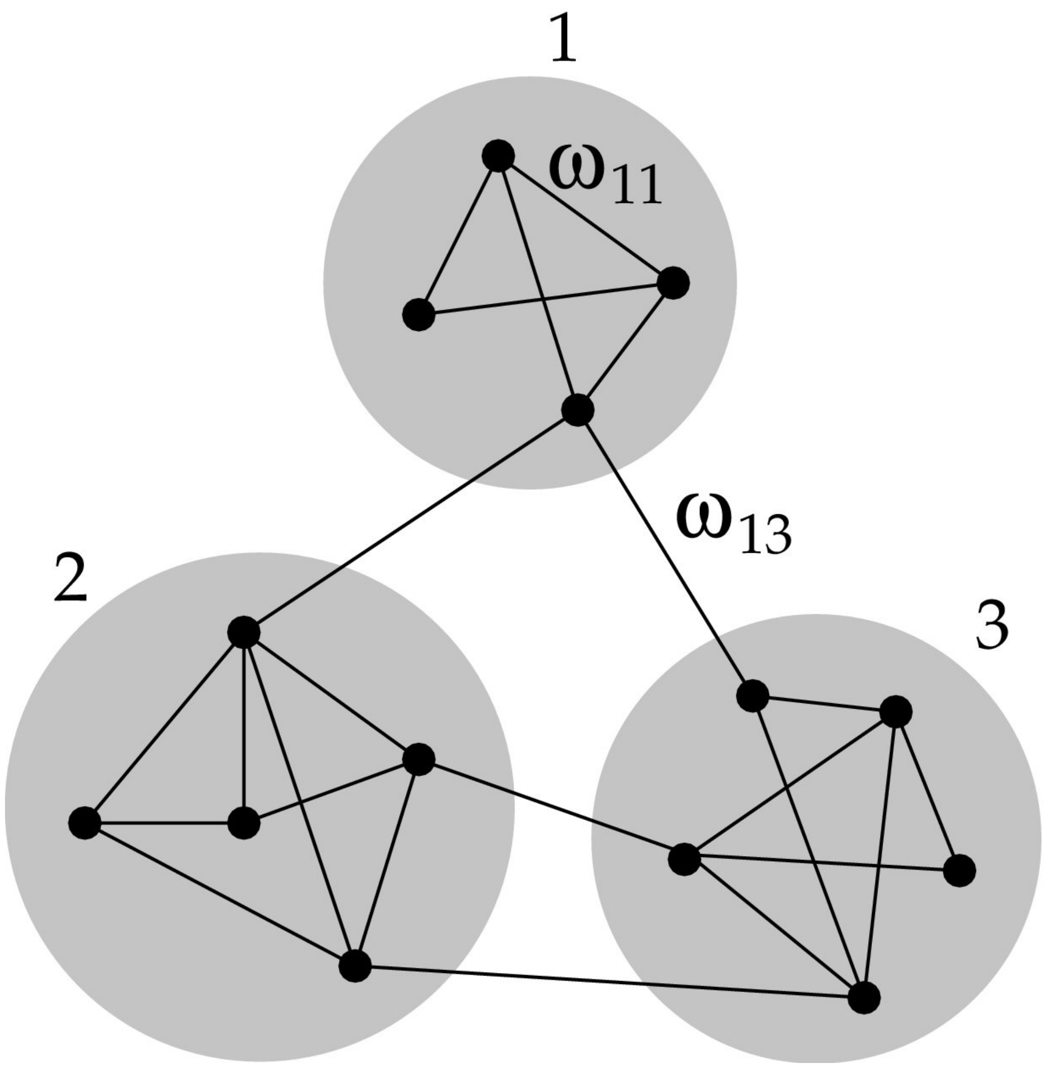
$$-\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{n} (x_i - \mu)^2 = 0$$

or

$$\boxed{\sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)^2}$$

Block models and inference

- We treat the problem as one of statistical inference
- We create a model of community structure then fit it to the observed data. Such models are called *block models* in the literature.



- The probability that this model generates a given observed network is

$$P(G|\omega, g) = \prod_{ij} \frac{(\omega_{g_i g_j})^{A_{ij}}}{A_{ij}!} \exp(-\omega_{g_i g_j})$$

- We want to find the set of parameters that maximizes this, or equivalently maximizes the logarithm. Neglecting constants the logarithm is

$$\log P(G|\omega, g) = \sum_{rs} (m_{rs} \log \omega_{rs} - n_r n_s \omega_{rs}).$$

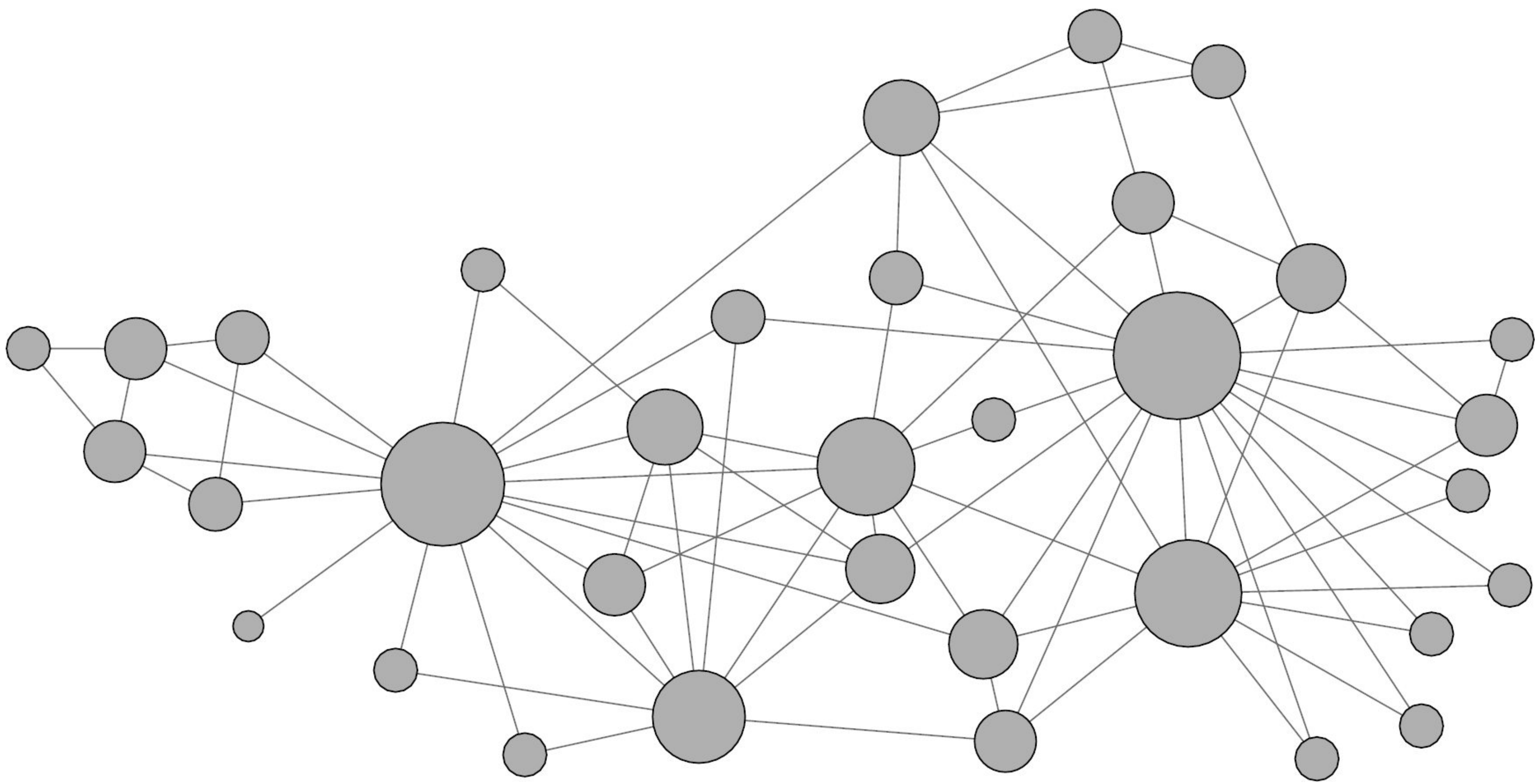
- Here m_{rs} is the number of edges between groups r and s and n_r is the numbers of vertices in group r

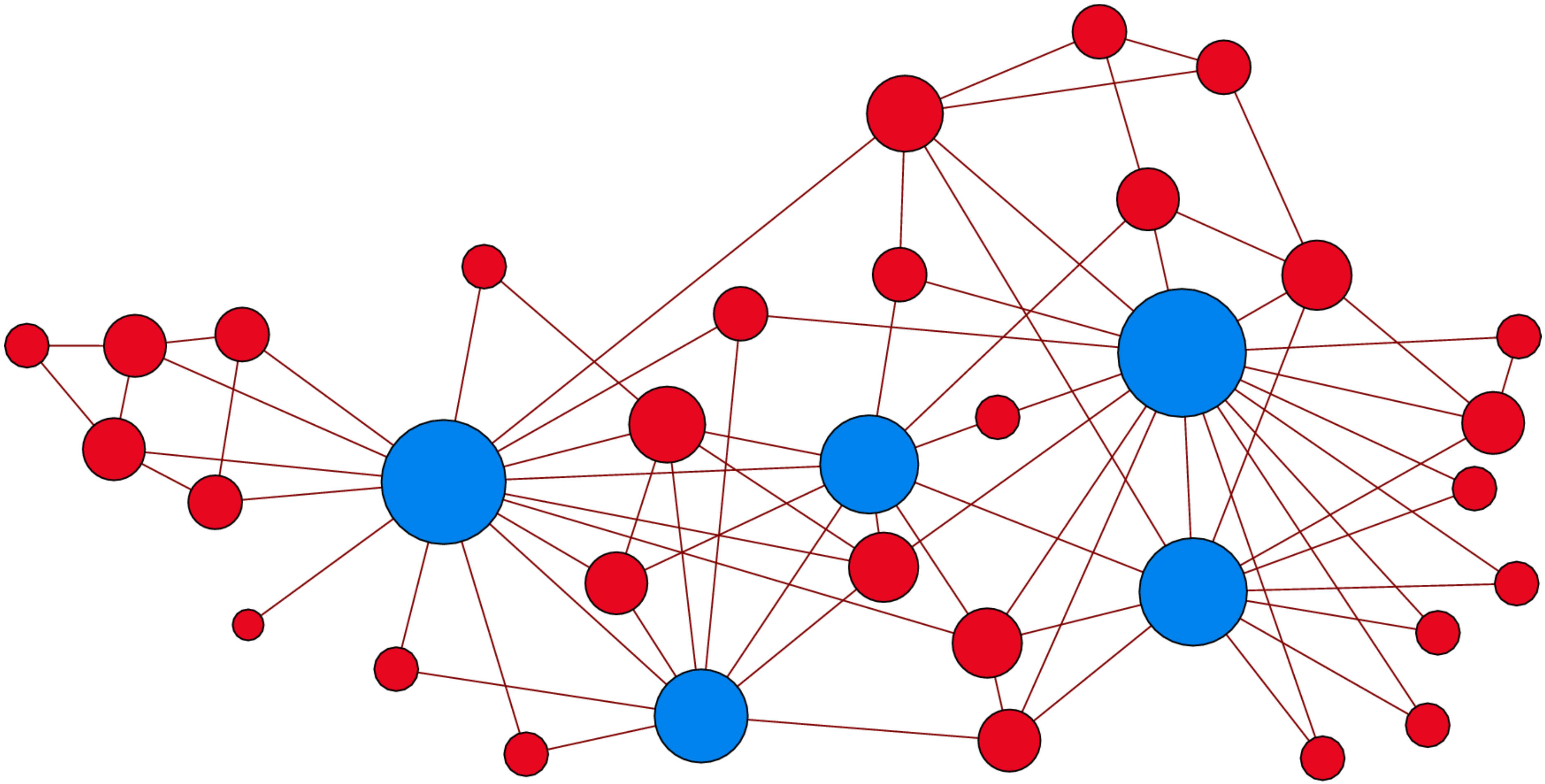
- Maximizing this expression first with respect to ω_{rs} we get $\omega_{rs} = m_{rs}/n_r n_s$ and substituting back into the log-likelihood gives

$$\mathcal{L}(G|g) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{n_r n_s}.$$

- Now we just have to maximize this expression with respect to the group memberships, and we have our answer
- Actually, we need to do a little more – it only works right if you also correct for degree
- This turns the problem of detecting communities into an optimization problem. A simple vertex-moving heuristic works well for small networks.

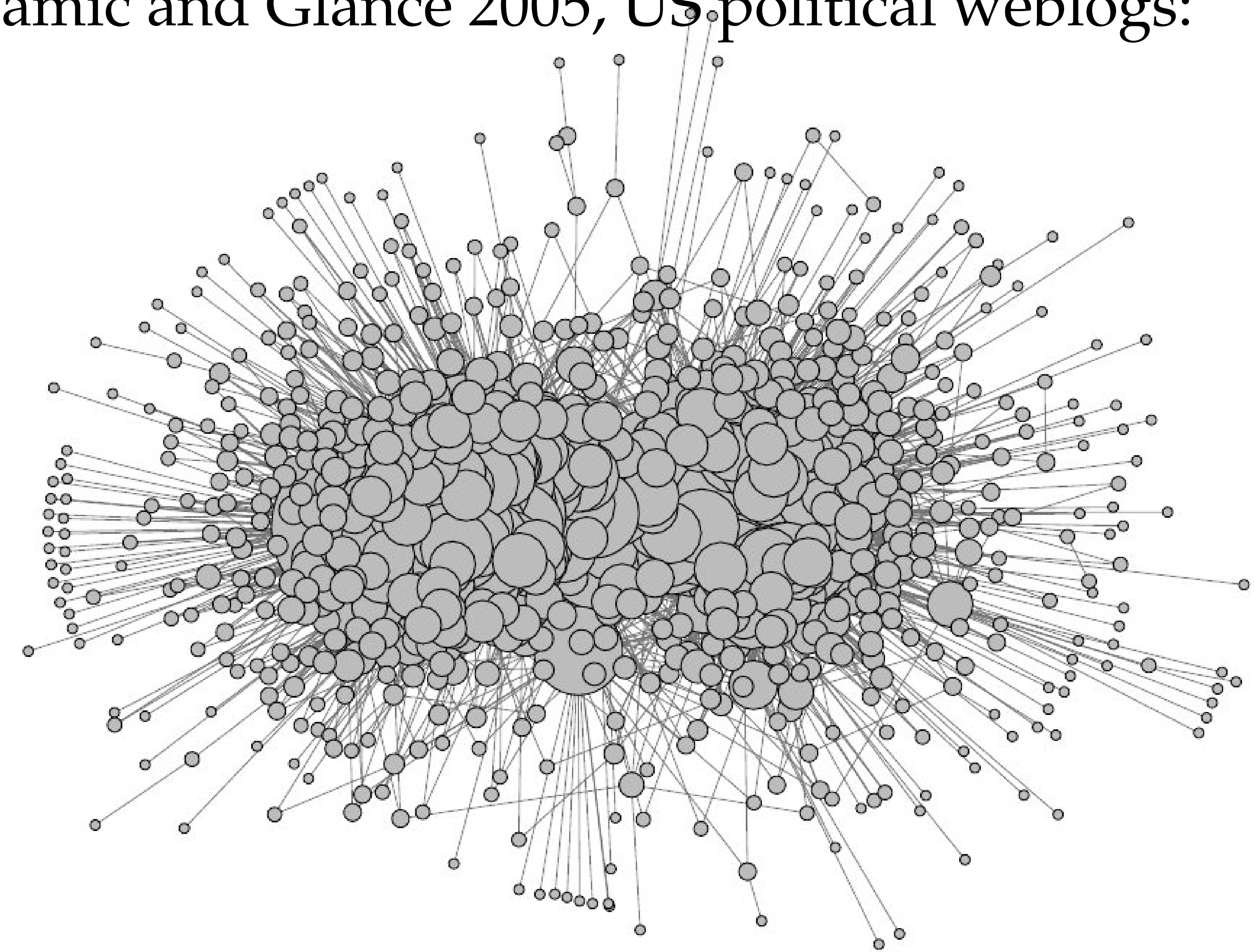
Example: Student club

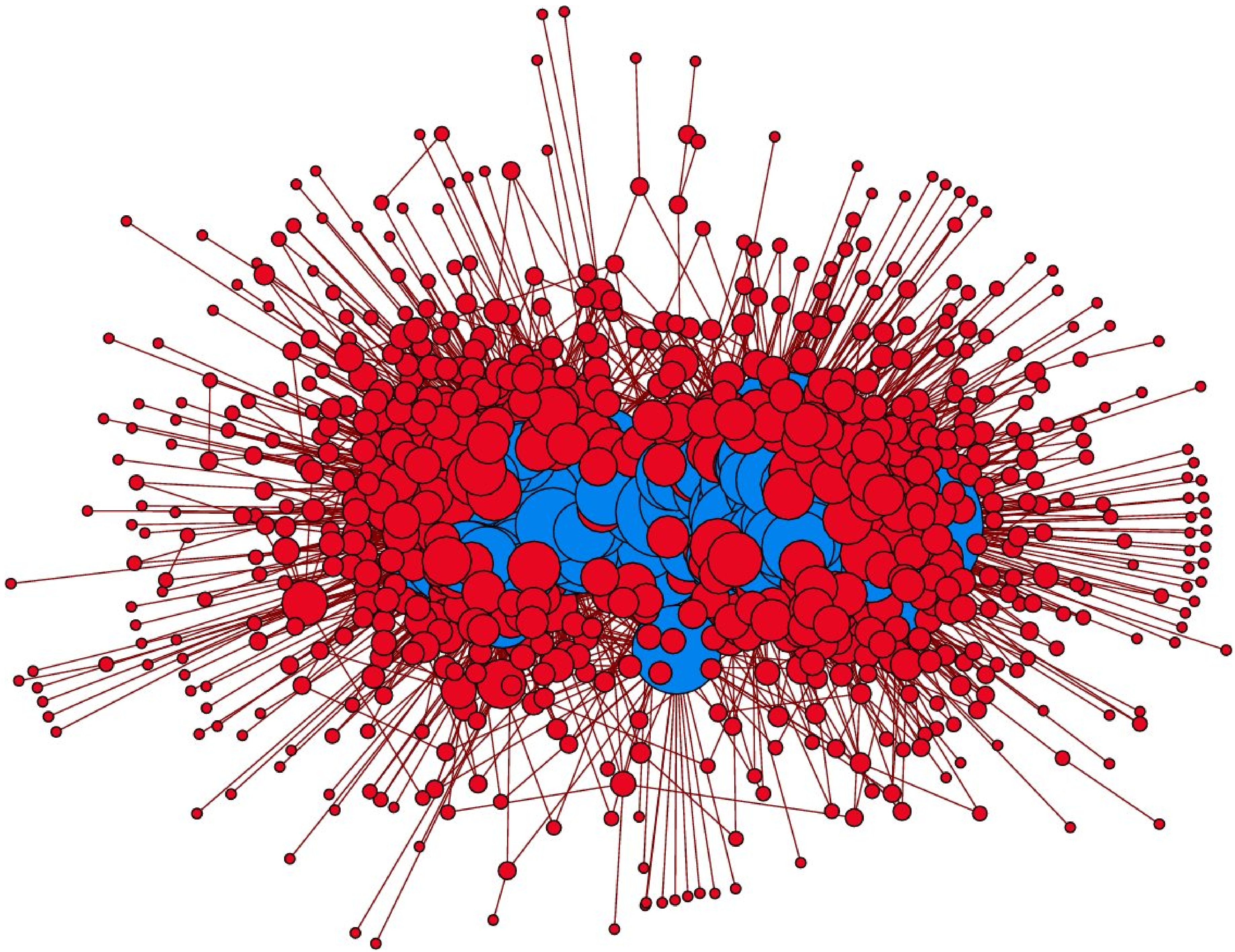




Blog network

- Adamic and Glance 2005, US political weblogs:





Correcting for degree

(Karrer and Newman 2011)

- The solution is to build the correct dependence on degree into the block model:

$$P(G|\theta,\omega,g) = \prod_{ij} \frac{(\theta_i\theta_j\omega_{g_ig_j})^{A_{ij}}}{A_{ij}!} \exp(-\theta_i\theta_j\omega_{g_ig_j})$$

- The overall constant is fixed by the normalization condition:

$$\sum_i \theta_i \delta_{g_i,r} = 1$$

Correcting for degree

- The log-likelihood, ignoring constants, is then

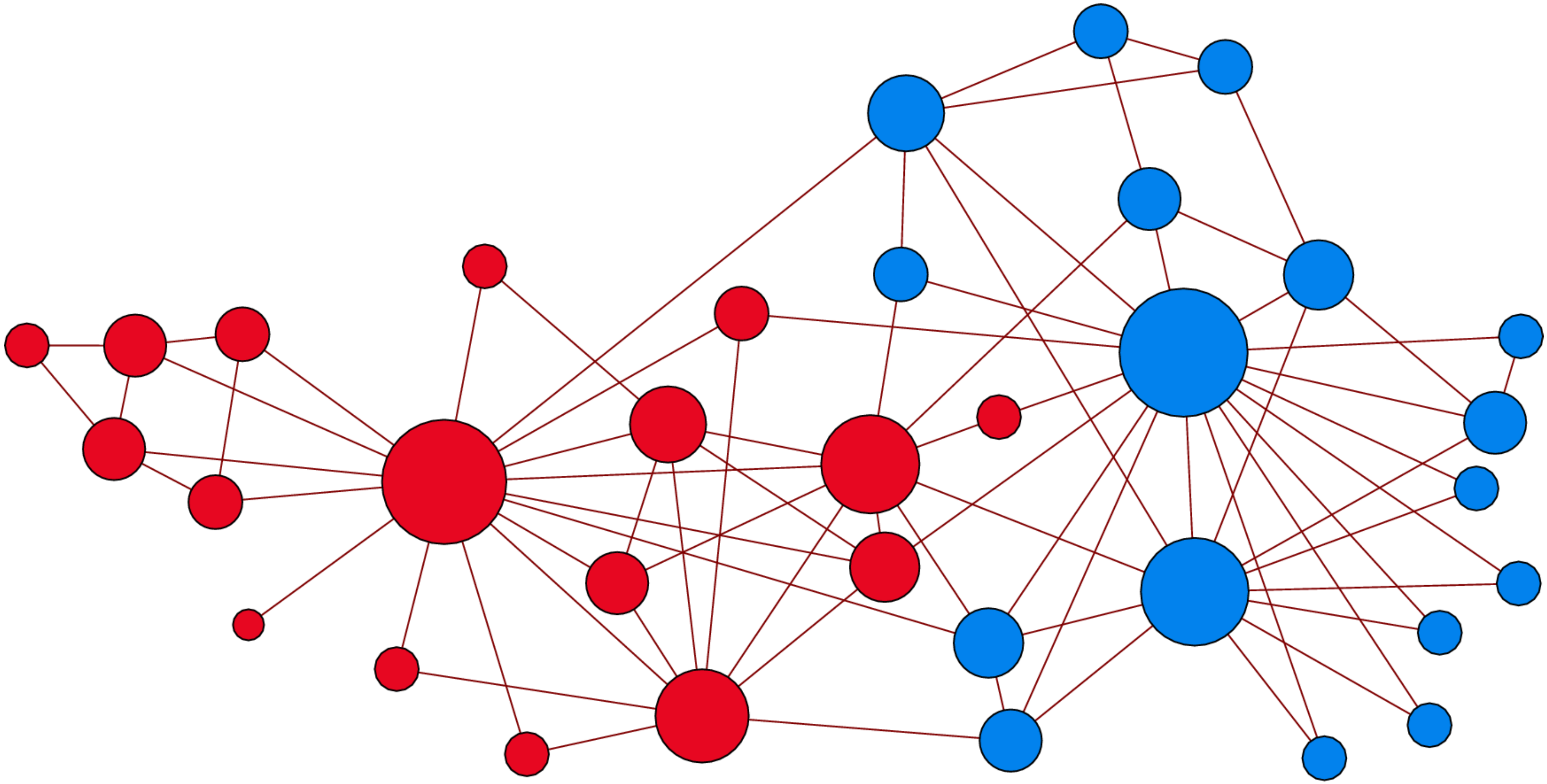
$$\log P(G|\theta, \omega, g) = 2\sum_i k_i \log \theta_i + \sum_{rs}(m_{rs} \log \omega_{rs} - \omega_{rs}).$$

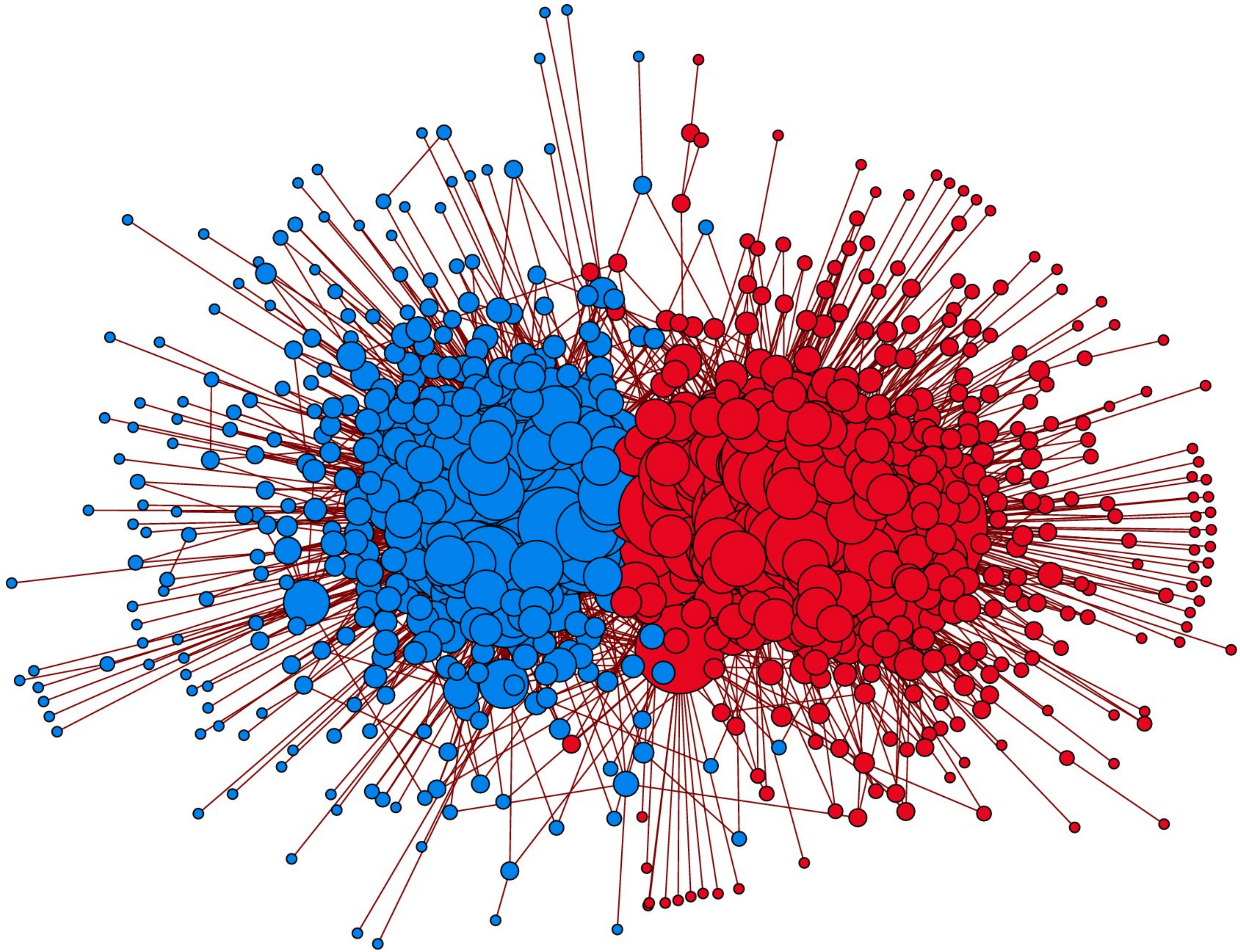
- The maximum likelihood parameter values are:

$$\hat{\theta}_i = \frac{k_i}{\sum_l k_l \delta_{g_l, g_i}}, \qquad \hat{\omega}_{rs} = m_{rs}$$

- Which gives a log-likelihood objective function thus:

$$\mathcal{L}(G|g) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{\kappa_r \kappa_s}, \qquad \kappa_r = \sum_i k_i \delta_{r, g_i}$$





Overlapping groups

(Ball, Karrer, and Newman 2011)

- A vertex can belong to more than one group
  - Family
  - Coworkers
  - Friends you know now
  - Friends from school or university
  - “Friends” on Facebook
- We can extend the θ_i parameters in our previous model to θ_{ir} which is i's degree in group r

- Our log-likelihood now looks like:

$$\log P(G|\theta,\omega) = \sum_{ij} A_{ij} \log\left(\sum_{rs} \theta_{ir}\theta_{js}\omega_{rs}\right) - \sum_{ijrs} \theta_{ir}\theta_{js}\omega_{rs}$$

- In principle, we can now just differentiate this to maximize, but we can do better than that
- Suppose you know the values of the parameters. Then:

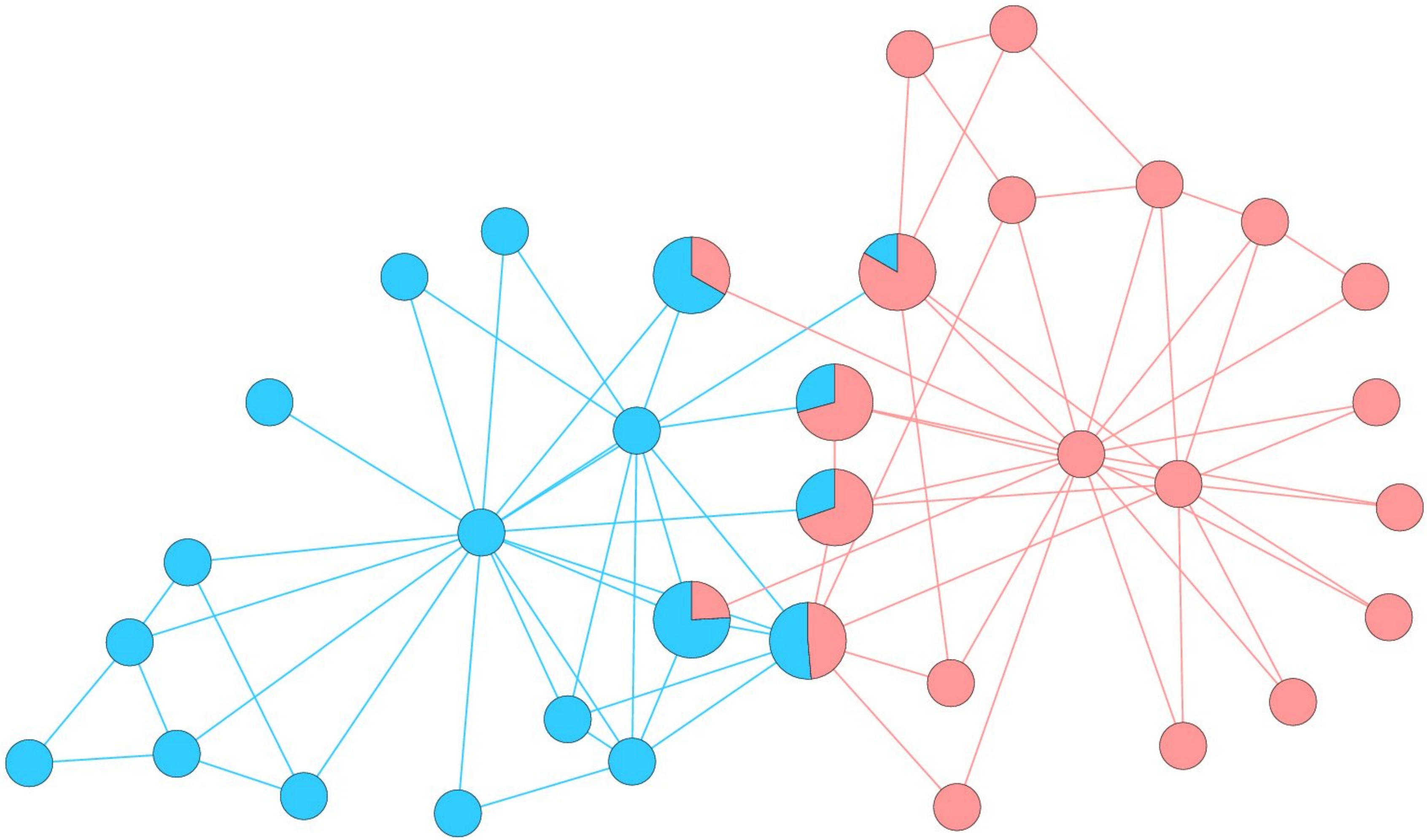
$$q_{ij}(r,s) = \frac{\theta_{ir}\theta_{js}\omega_{rs}}{\sum_{rs} \theta_{ir}\theta_{js}\omega_{rs}}$$

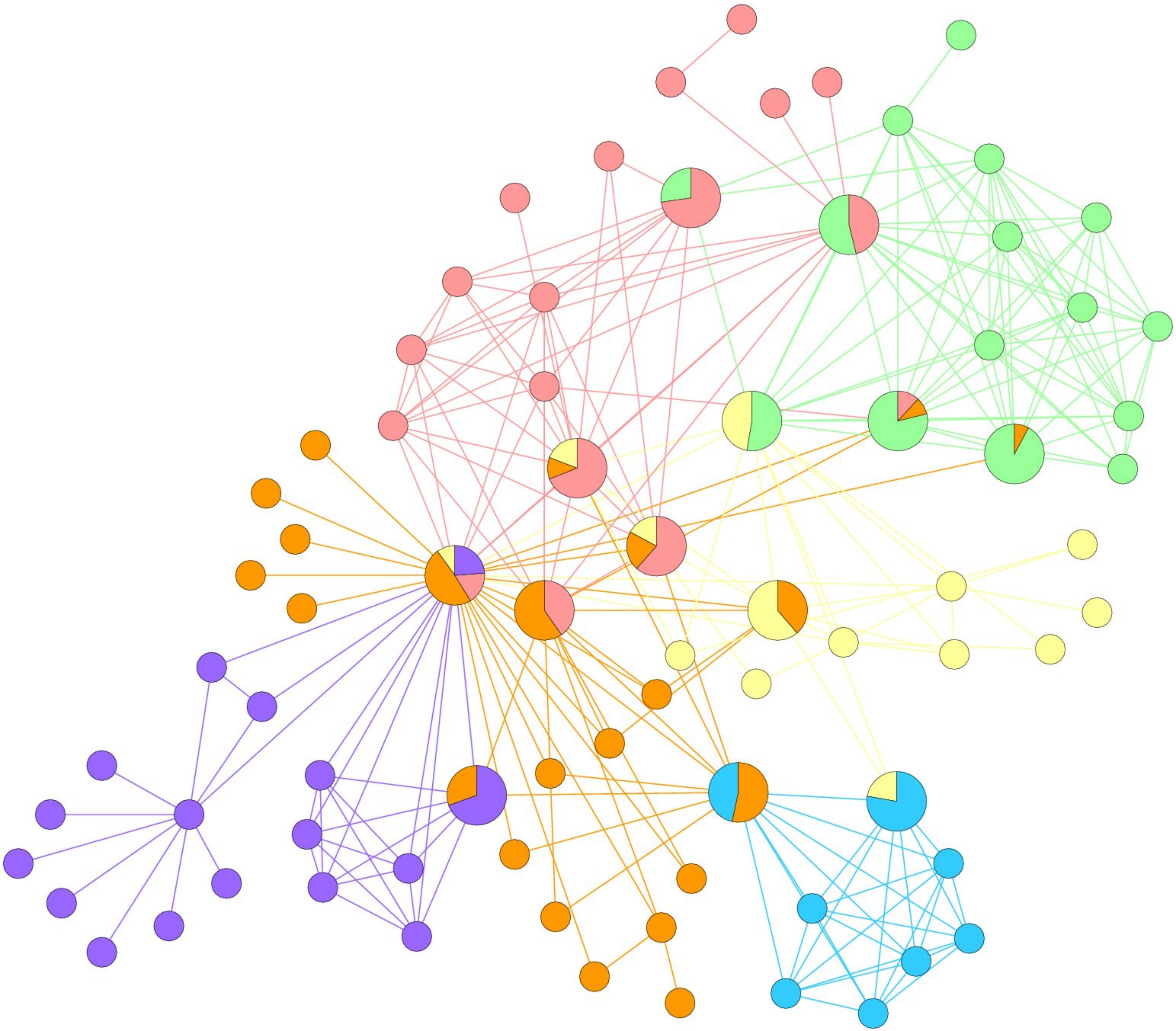
- And given this we can calculate the values of the parameters:

$$\theta_{ir} = \frac{\sum_{js} A_{ij} q_{ij}(r,s)}{\sum_{ijs} A_{ij} q_{ij}(r,s)}, \qquad \omega_{rs} = \sum_{ij} A_{ij} q_{ij}(r,s)$$

$$q_{ij}(r,s) = \frac{\theta_{ir}\theta_{js}\omega_{rs}}{\sum_{rs} \theta_{ir}\theta_{js}\omega_{rs}}$$

- This gives a classic expectation-maximization (EM) algorithm: choose a random starting condition and iterate to convergence.
- Each iteration takes O(m) time
- Scales to millions of nodes

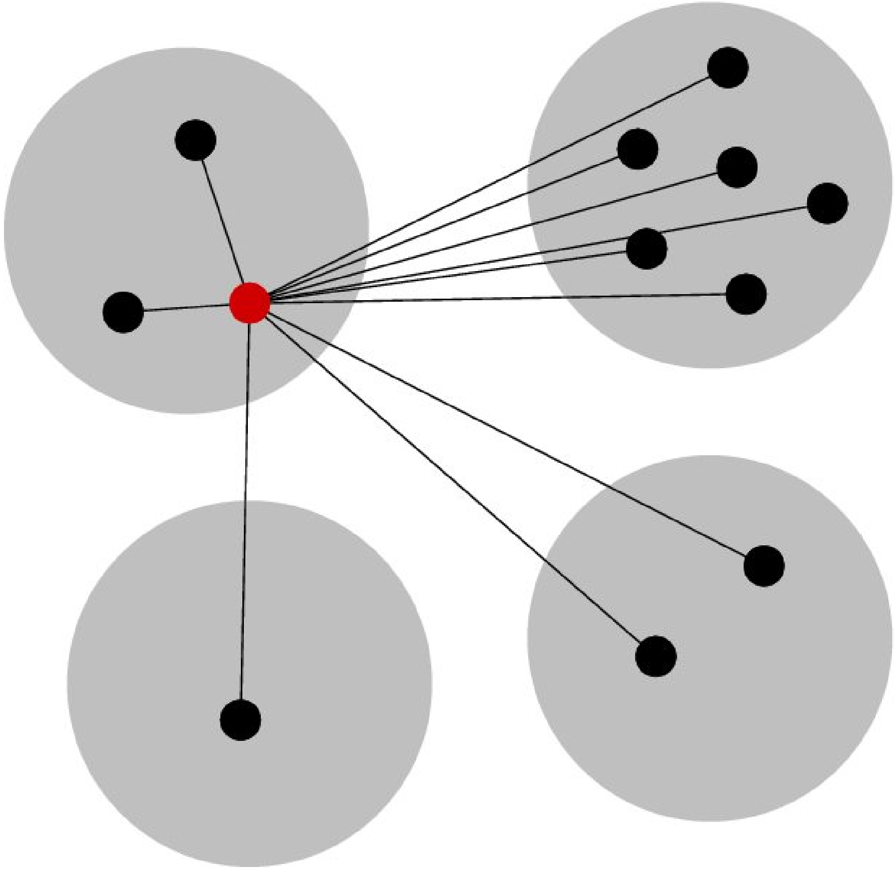




Vertex classification

(Newman and Leicht 2008)

We can define a very broad set of possible group structures for networks:



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

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

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

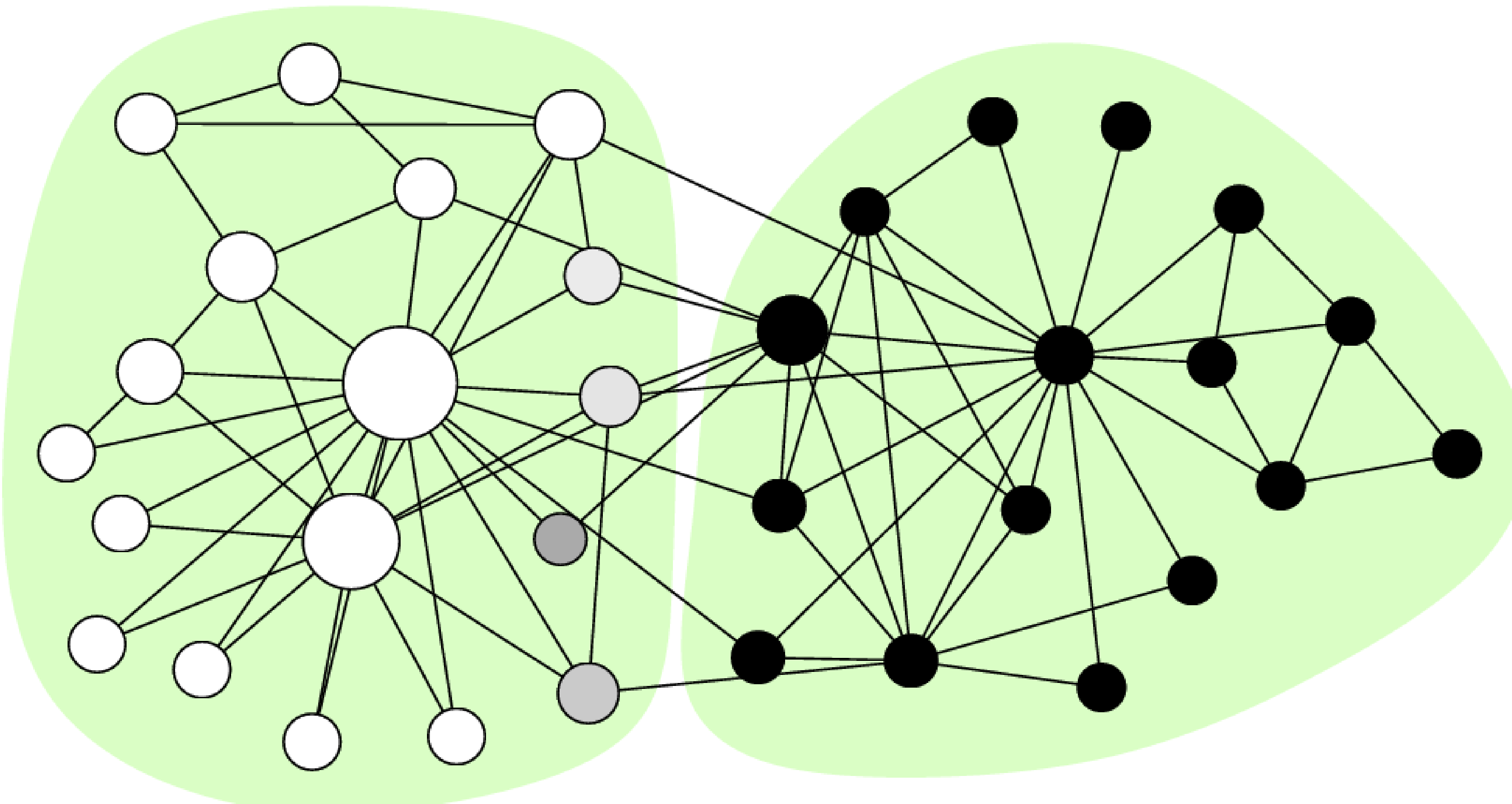
EM algorithm

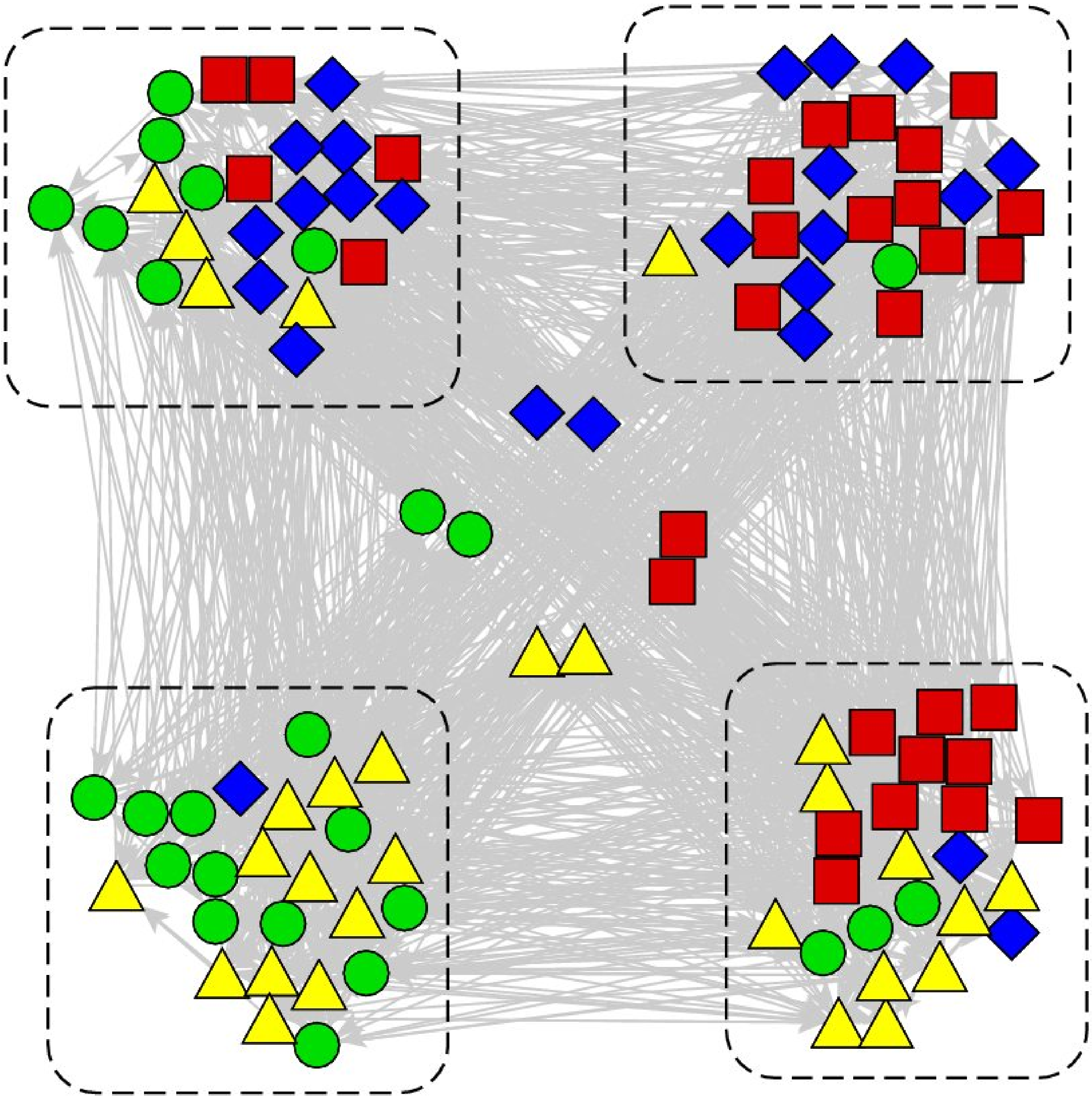
- The EM equations now look like this:

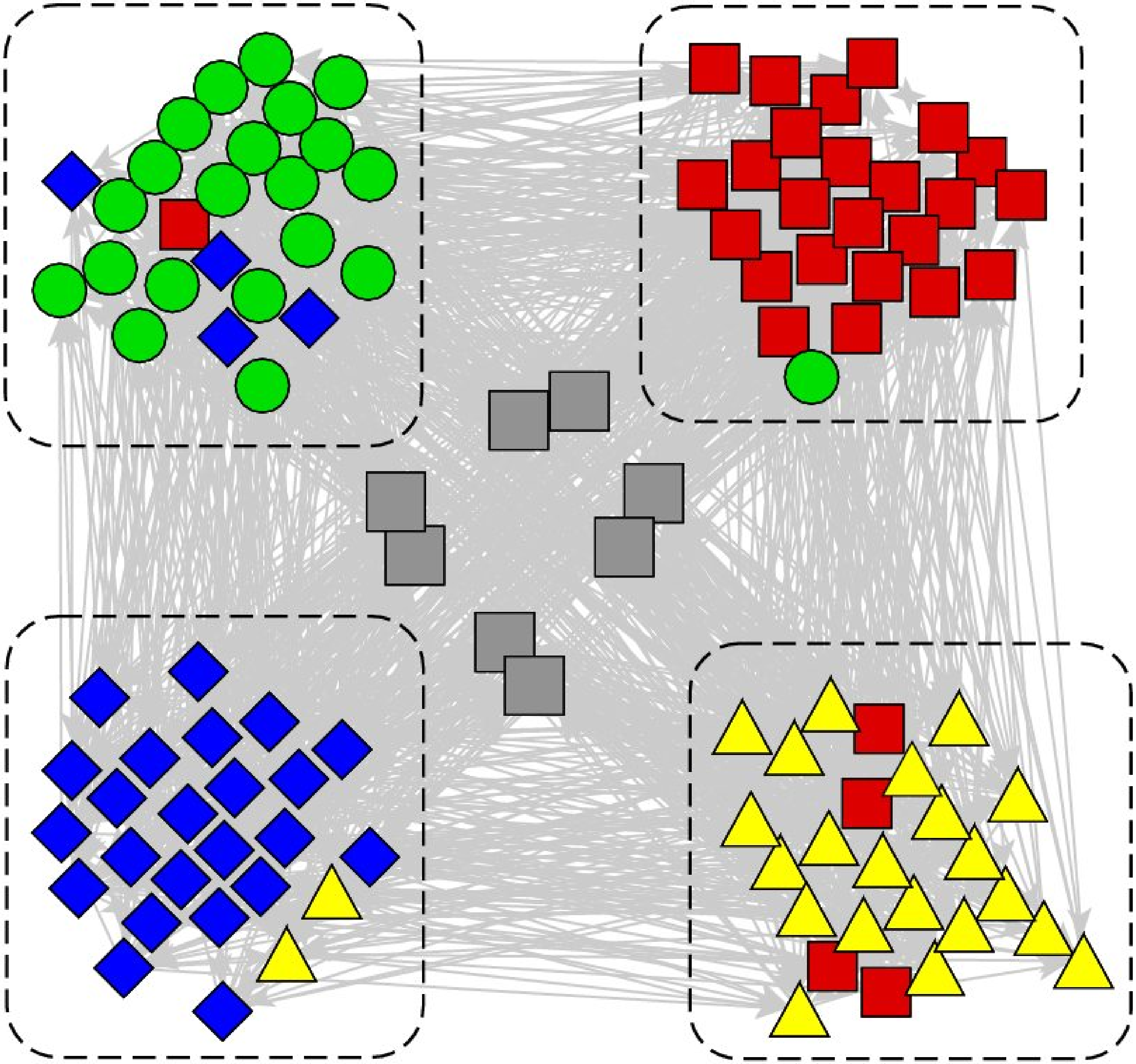
$$\pi_r = \frac{1}{n} \sum_i q_{ir}, \qquad \theta_{rj} = \frac{\sum_i A_{ij} q_{ir}}{\sum_i k_i q_{ir}},$$

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

- The derivation is more complicated for the undirected case, but the equations end up the same

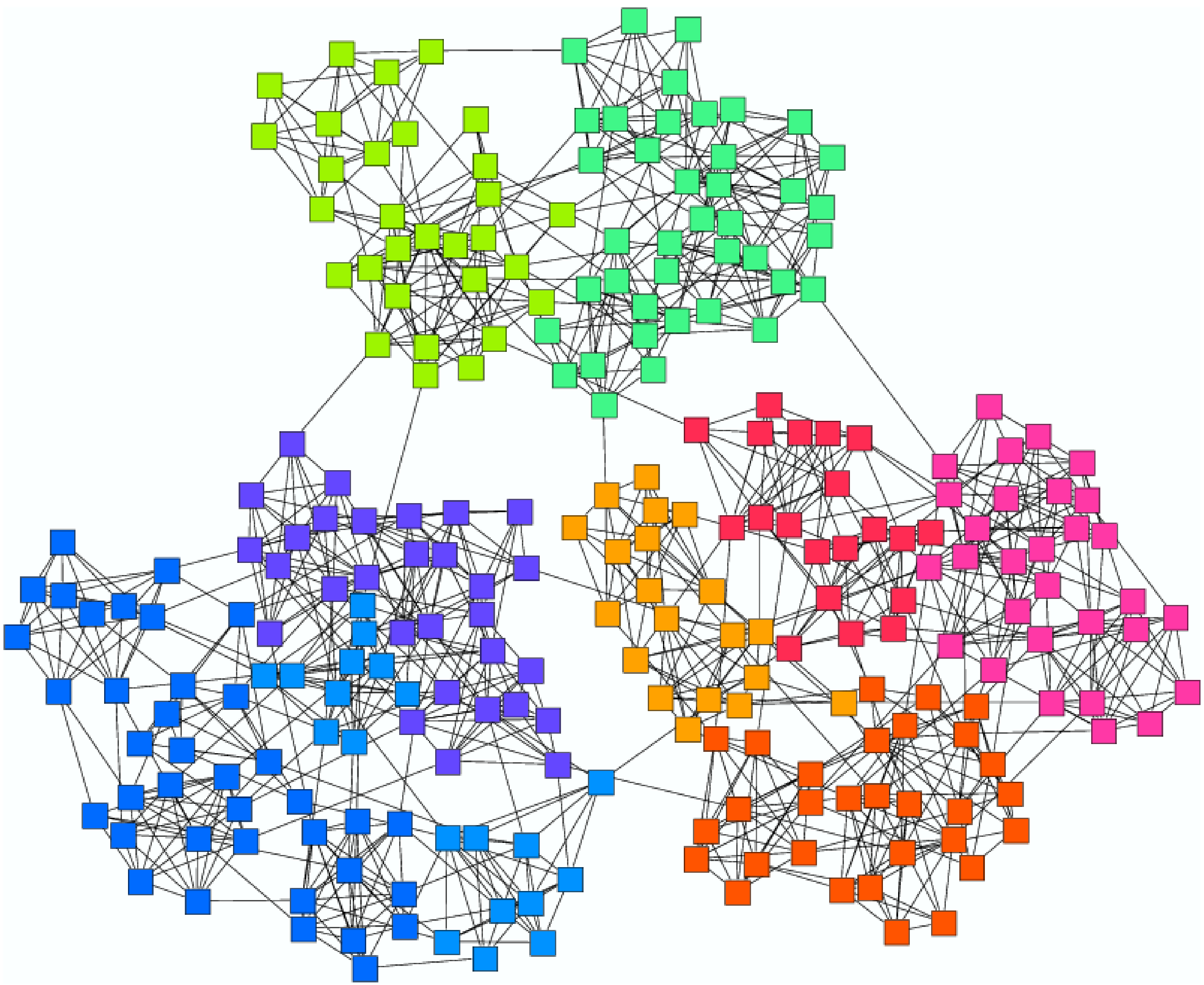




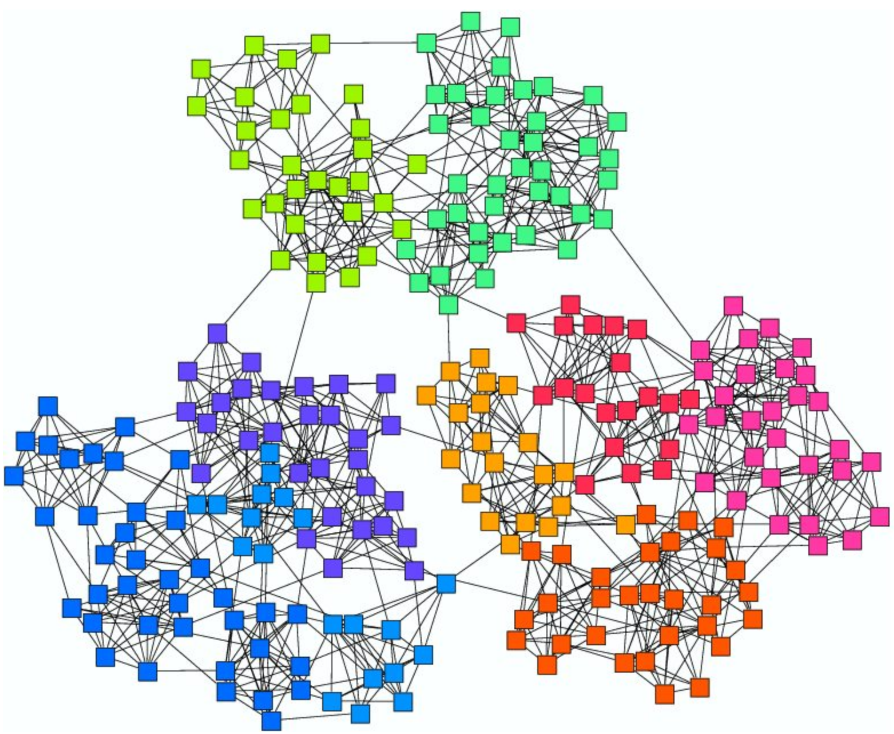


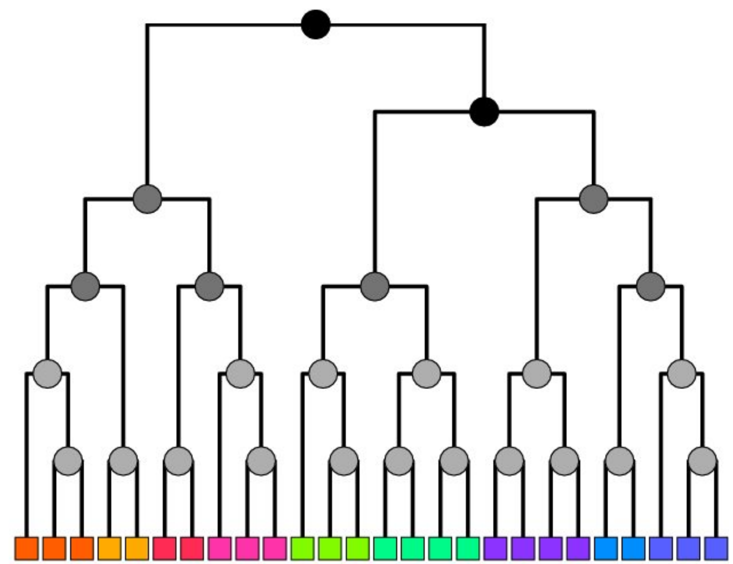
Network hierarchy

(Clauset, Moore, and Newman 2008)

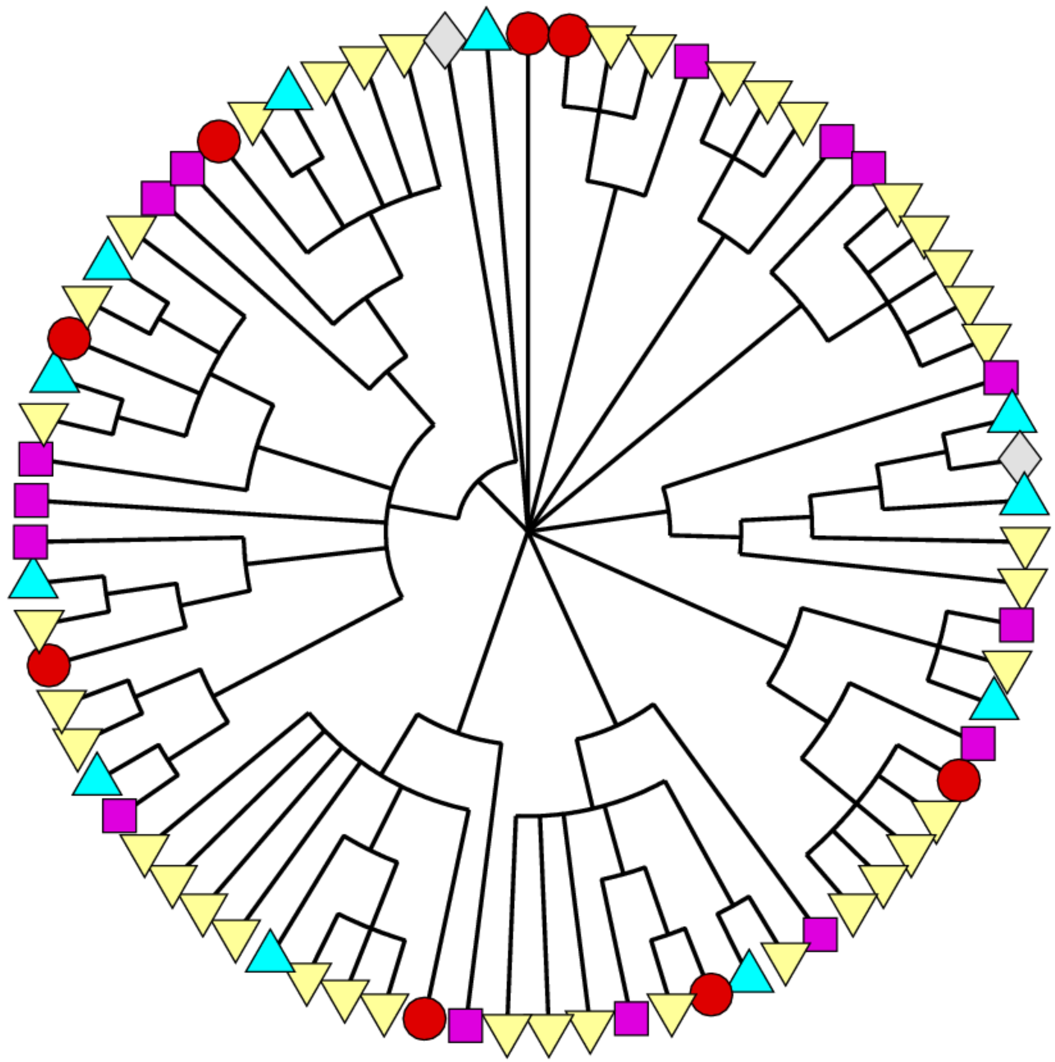


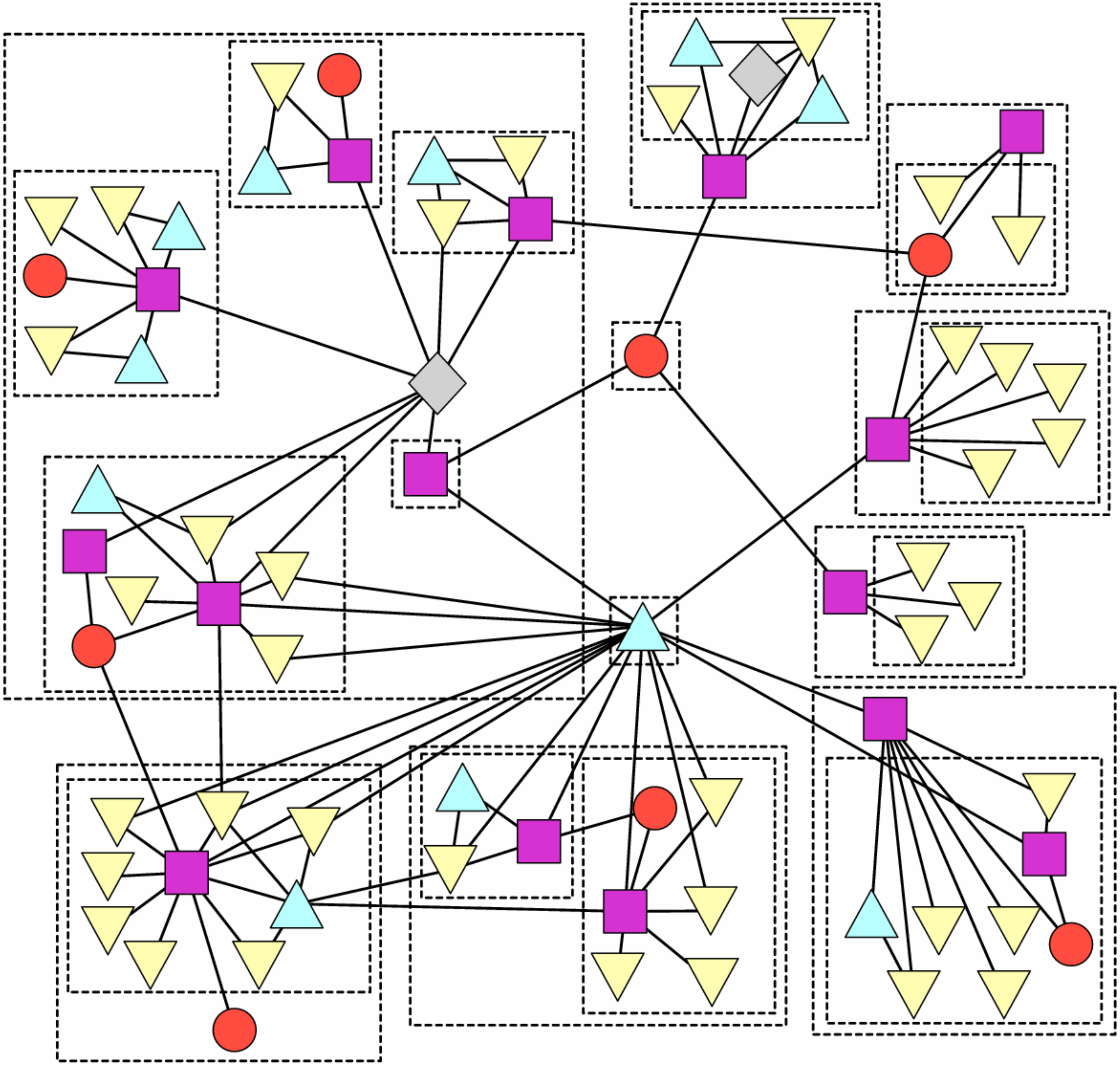
Network hierarchy





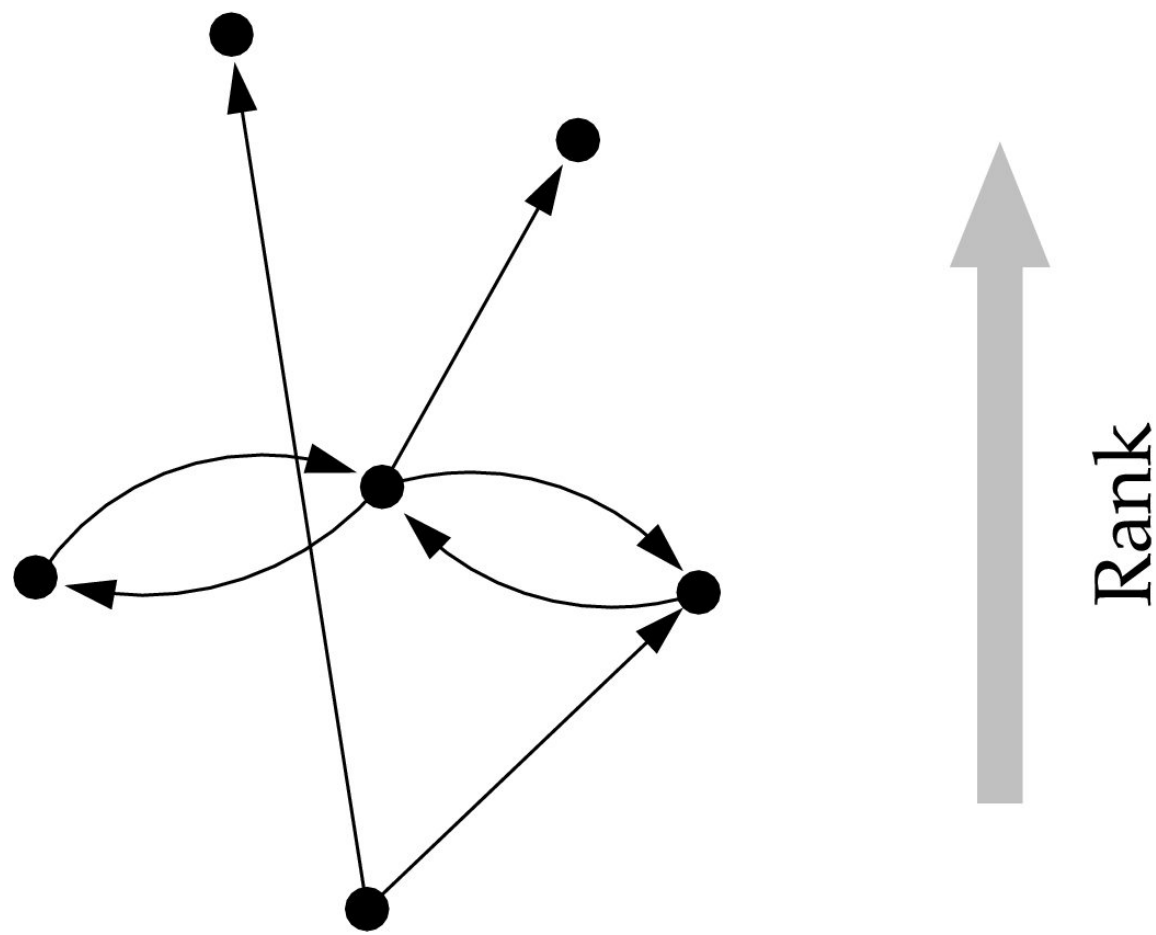
- Generate consensus hierarchies:





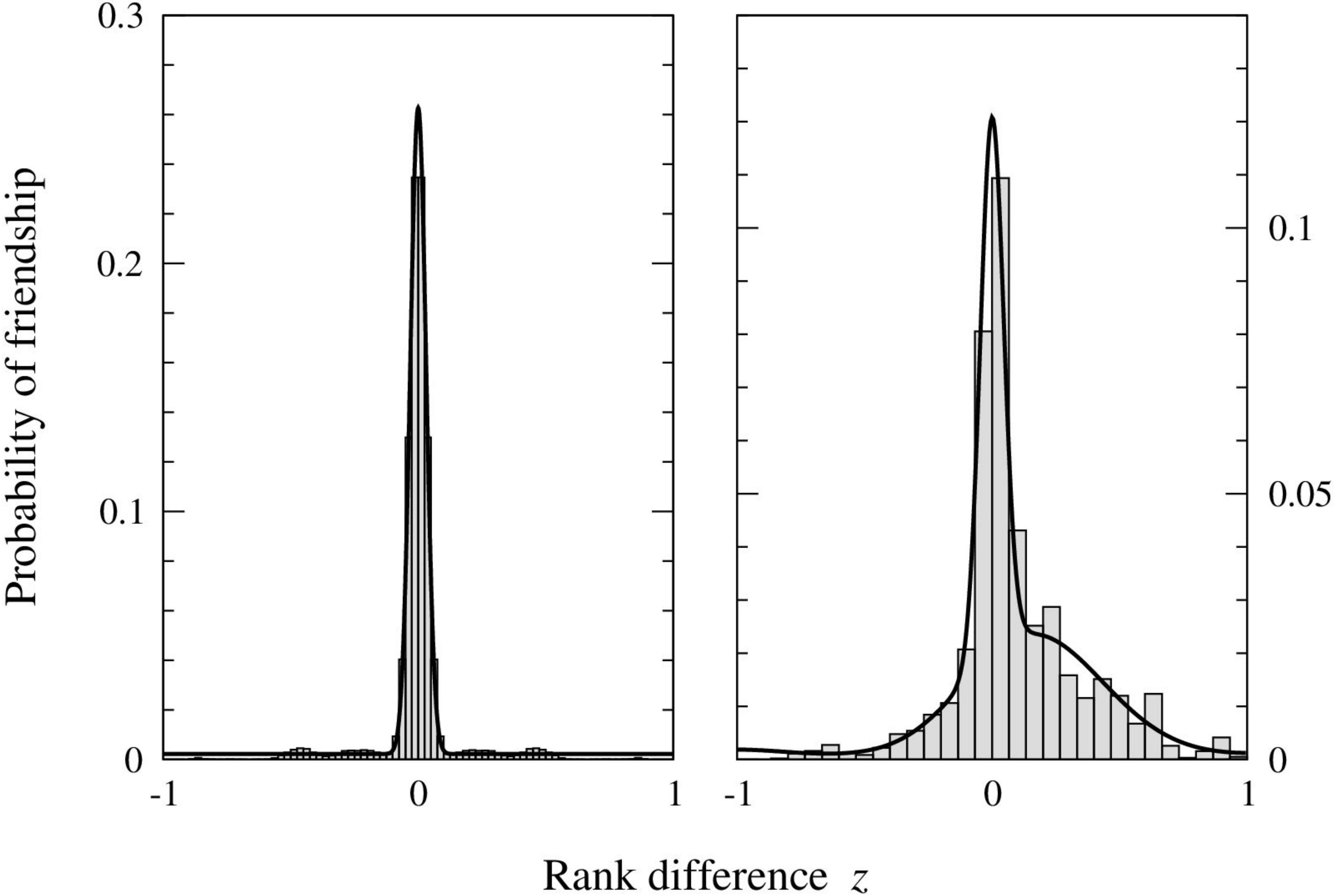
Ranking and status

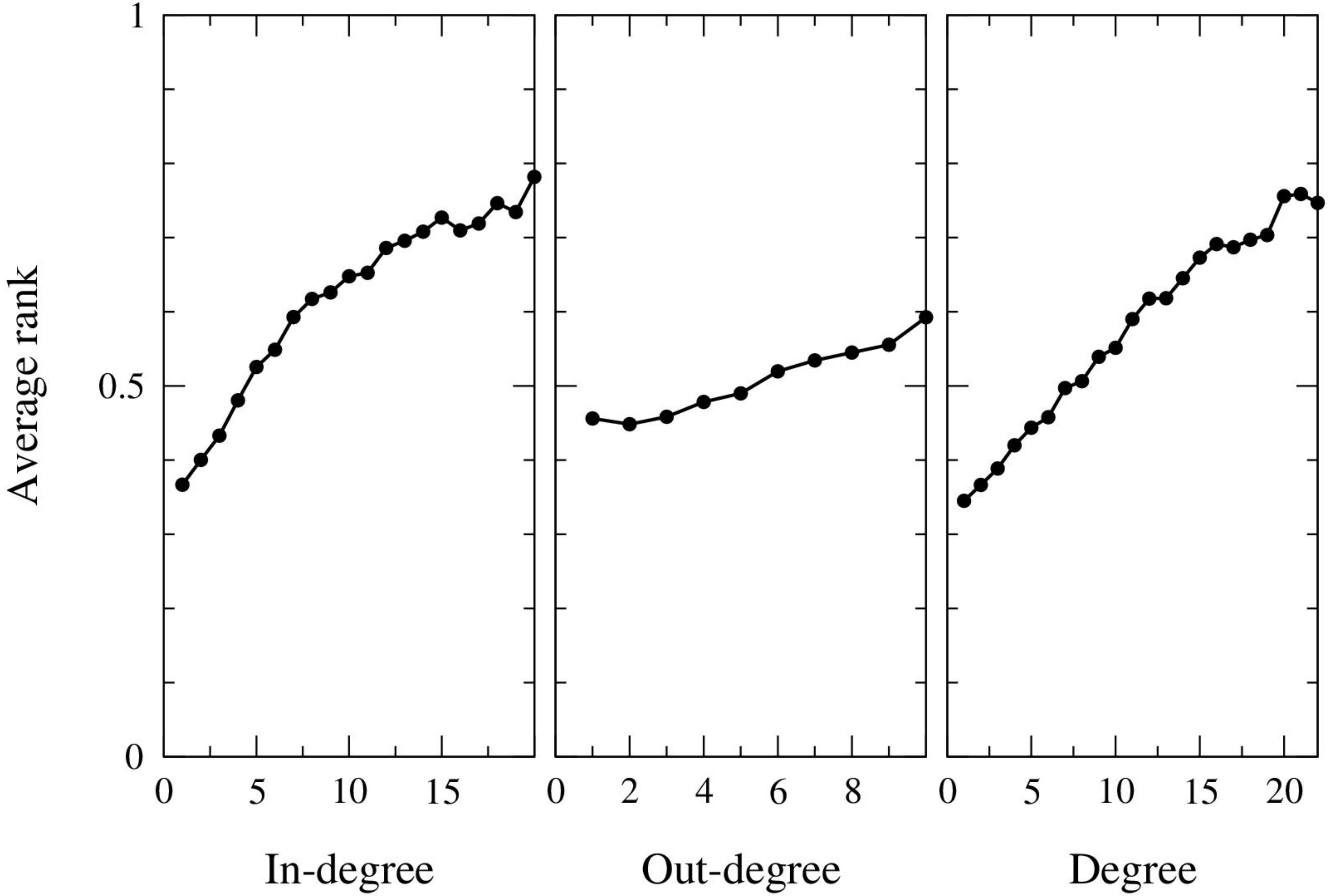
(Ball and Newman 2013)



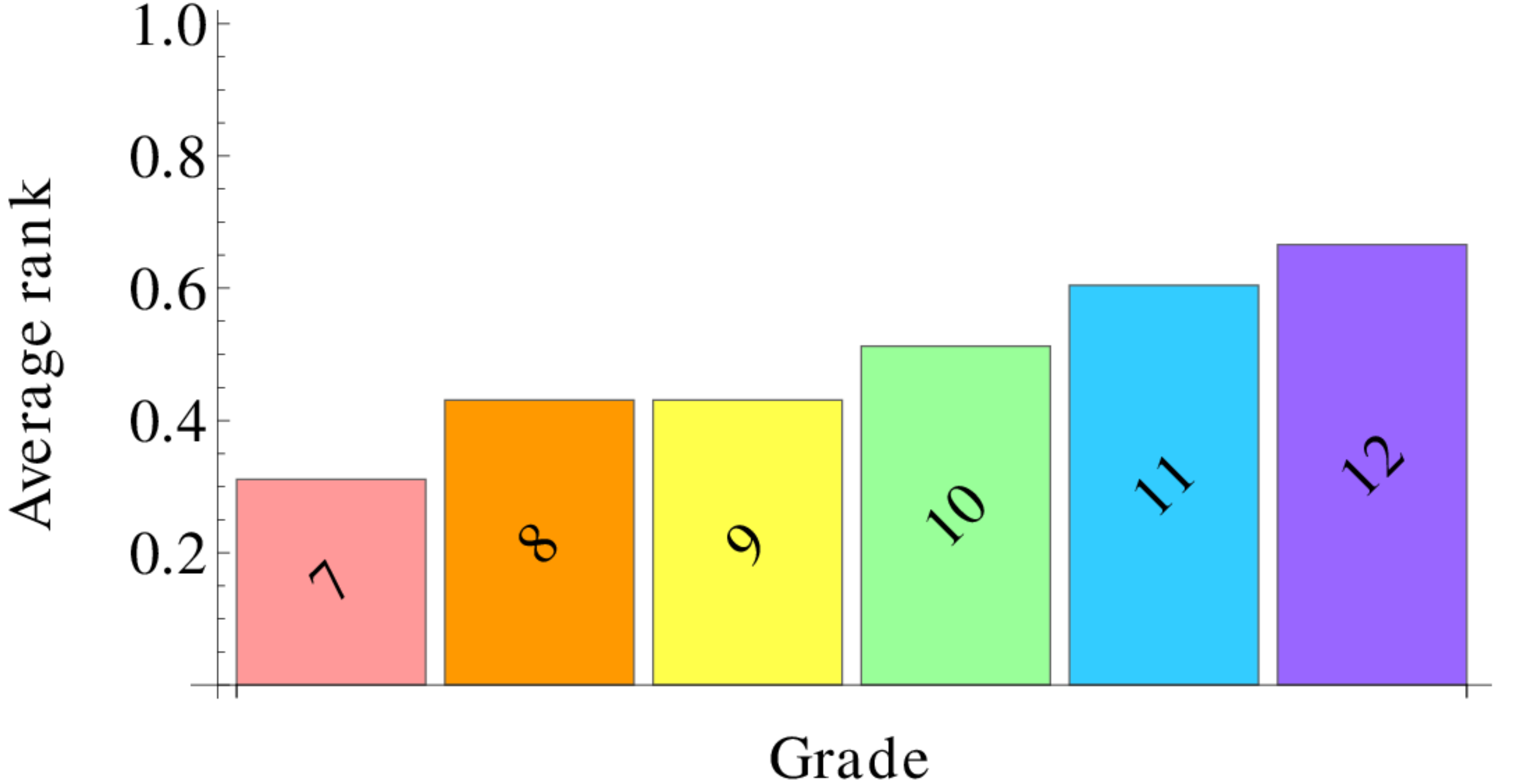
Rankings from network inference

- Assume a ranking and different probabilities for the directed and bidirectional edges
- We use an EM algorithm to calculate both self-consistently
- From this we learn:
  - The ranking of the nodes
  - The separate probability functions for directed and undirected edges

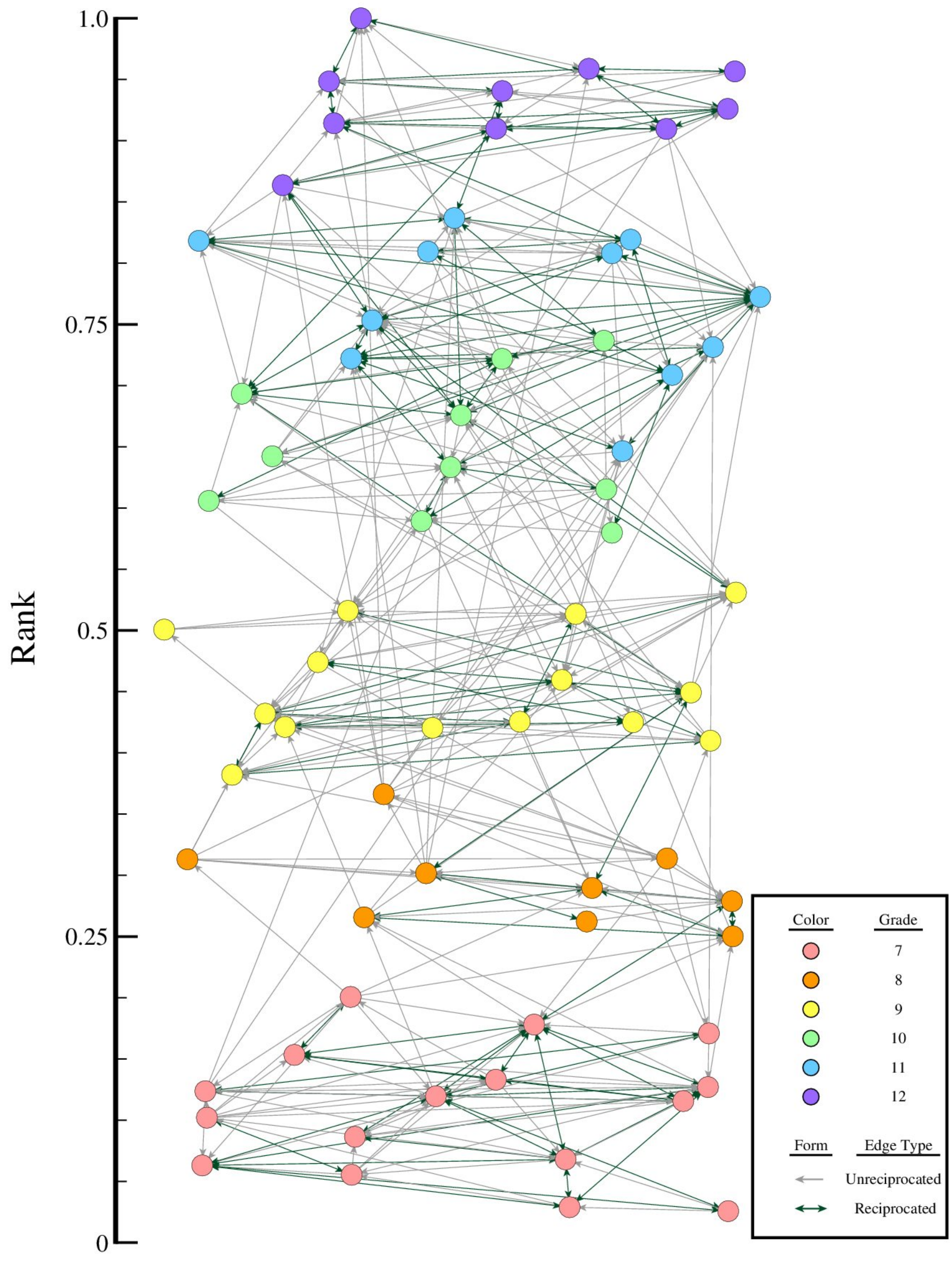

0.3
0.2
0.1
0
Probability of friendship
0.1
0.05
0
-1
0
1
-1
0
1
Rank difference z



Average rank
1
0.5
0
0
5
10
15
In-degree
0
2
4
6
8
Out-degree
0
5
10
15
20
Degree



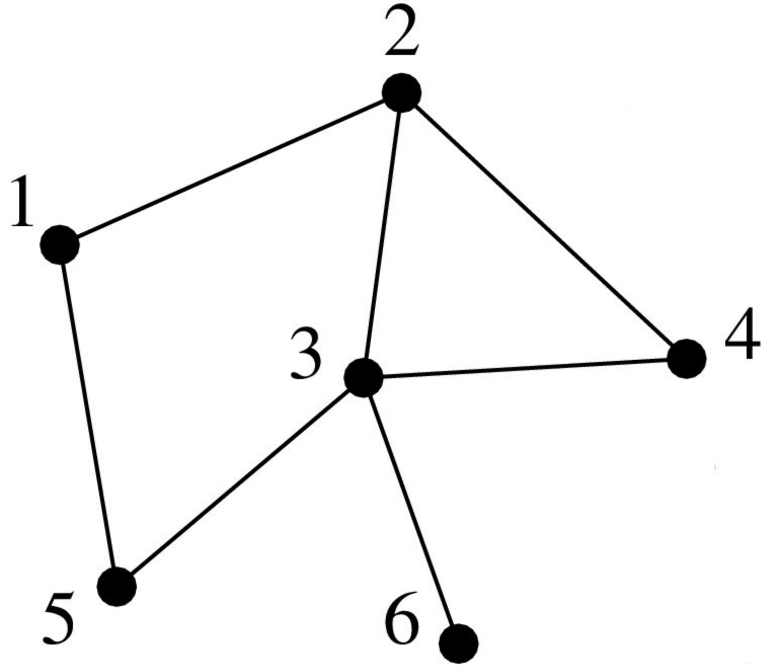
1.0
0.8
0.6
0.4
0.2
Average rank
7
8
9
10
11
12
Grade



Rank
1.0
0.75
0.5
0.25
0
Color
Grade
7
8
9
10
11
12
Form
Edge Type
Unreciprocated
Reciprocated

Graph spectra

- A network (or graph) can be represented by an *adjacency matrix* **A**:



$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

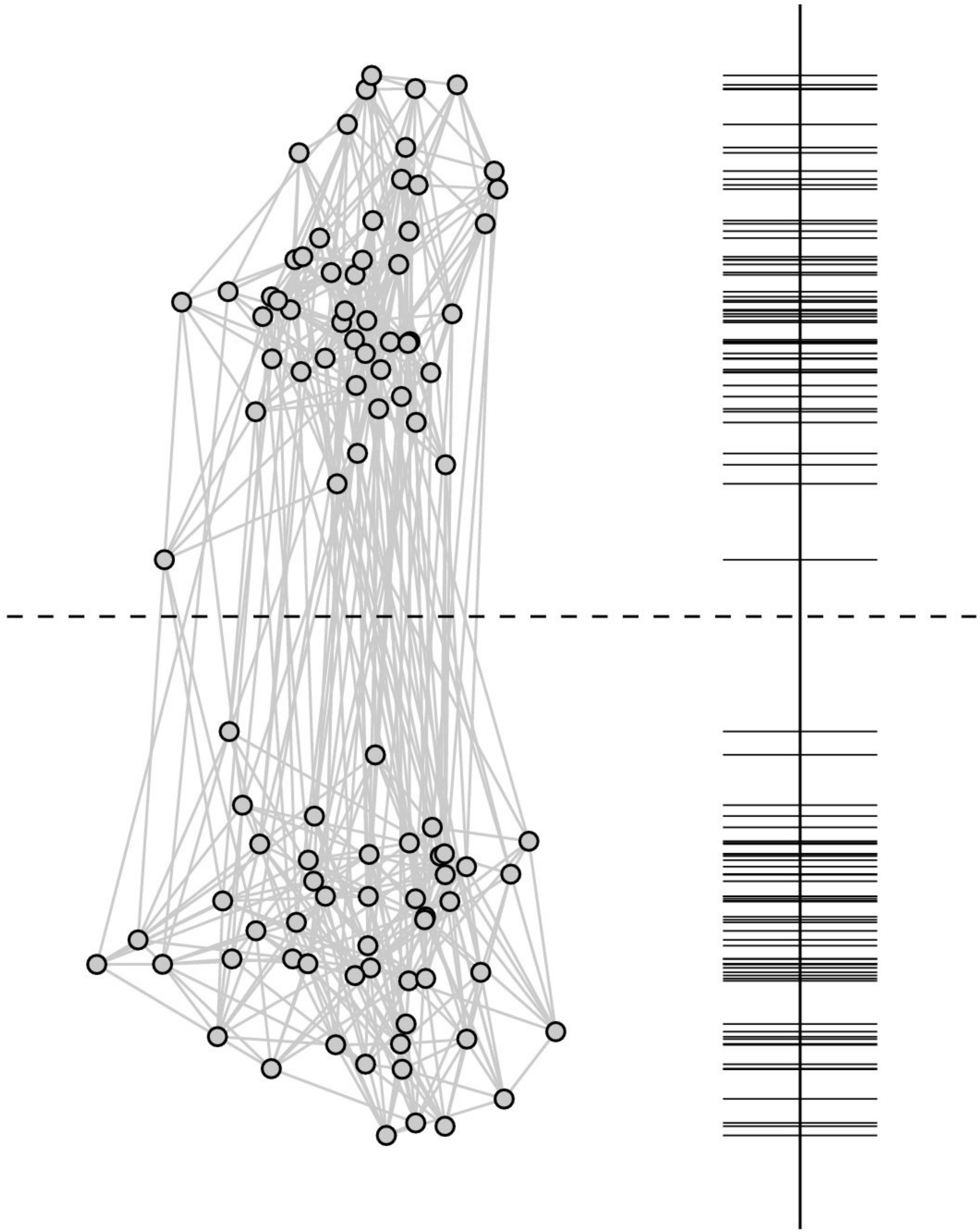
Centrality

- Which is the most important node in a network?
  - Degree centrality: you get one point for each neighbor
  - Better: you get points in proportion to the sum of your neighbor's points (Bonacich 1987)

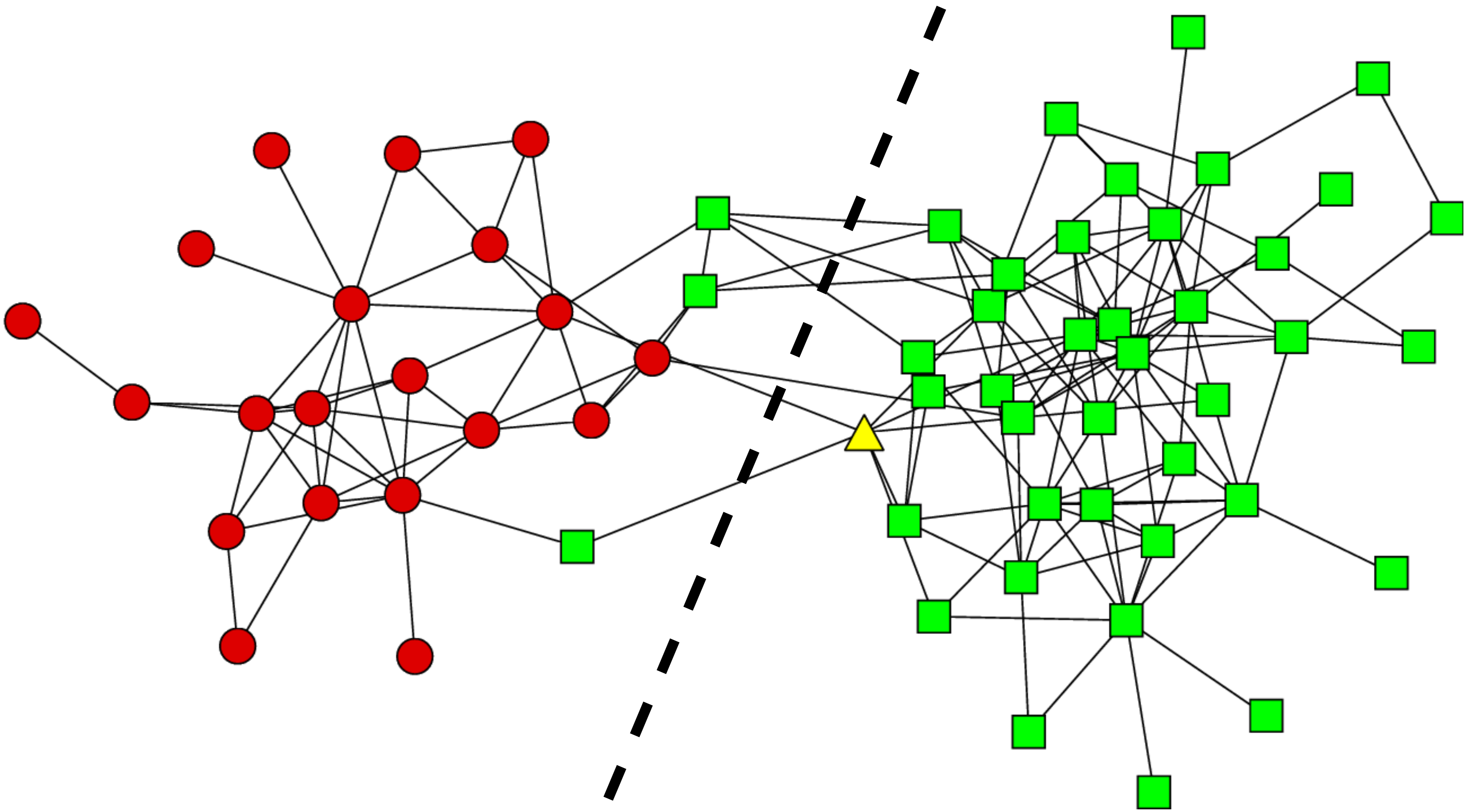
$$x_i = \lambda^{-1} \sum_{j \in \mathcal{N}(i)} x_j = \lambda^{-1} \sum_{j=1}^{n} A_{ij} x_j$$

or

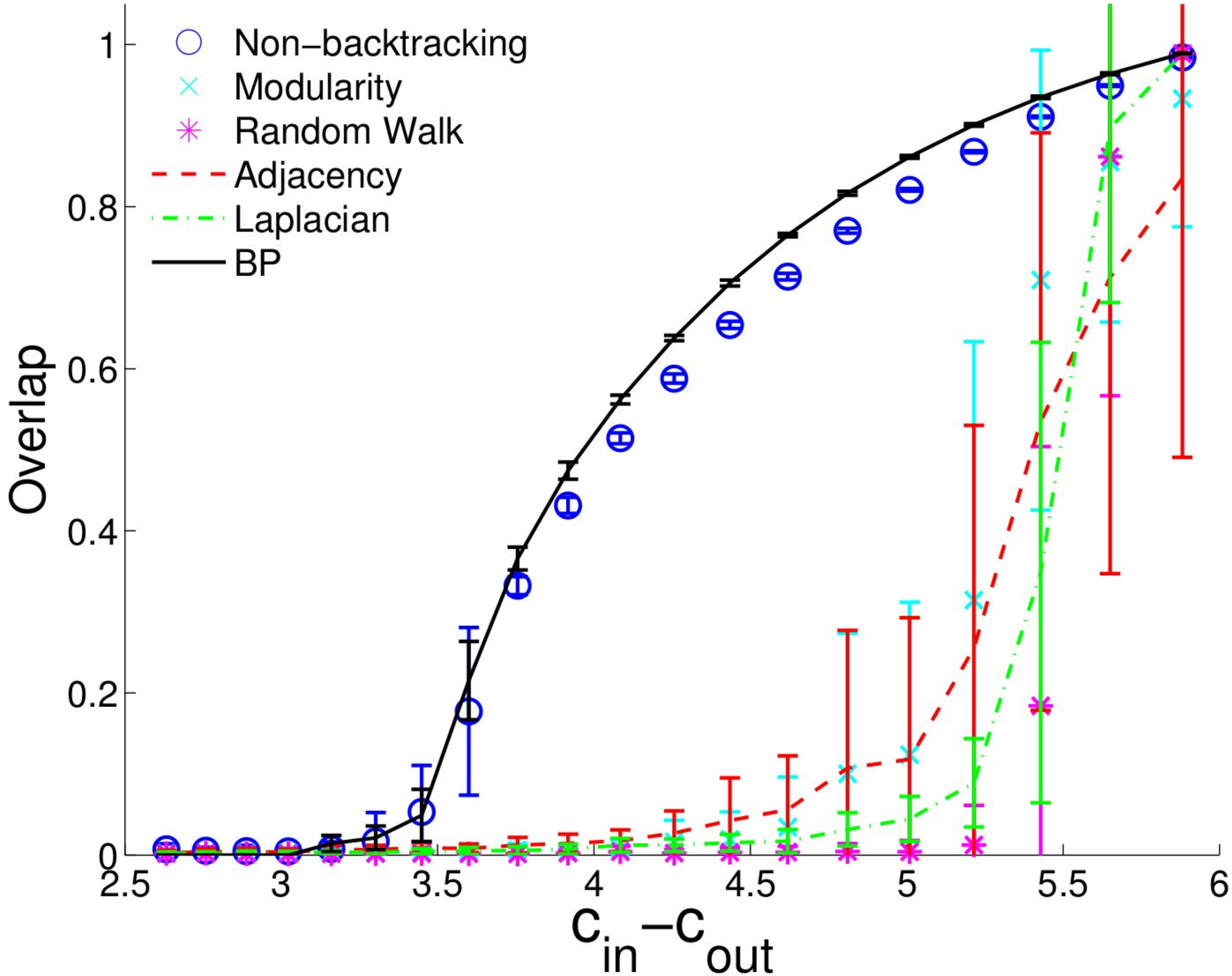
$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$$



Example: Animal network



Stochastic block model



Krzakala *et al.* 2013

Belief propagation for block models

- Decelle, Krzakala, Moore, and Zdeborová (2010) developed belief propagation for the maximum likelihood fit:

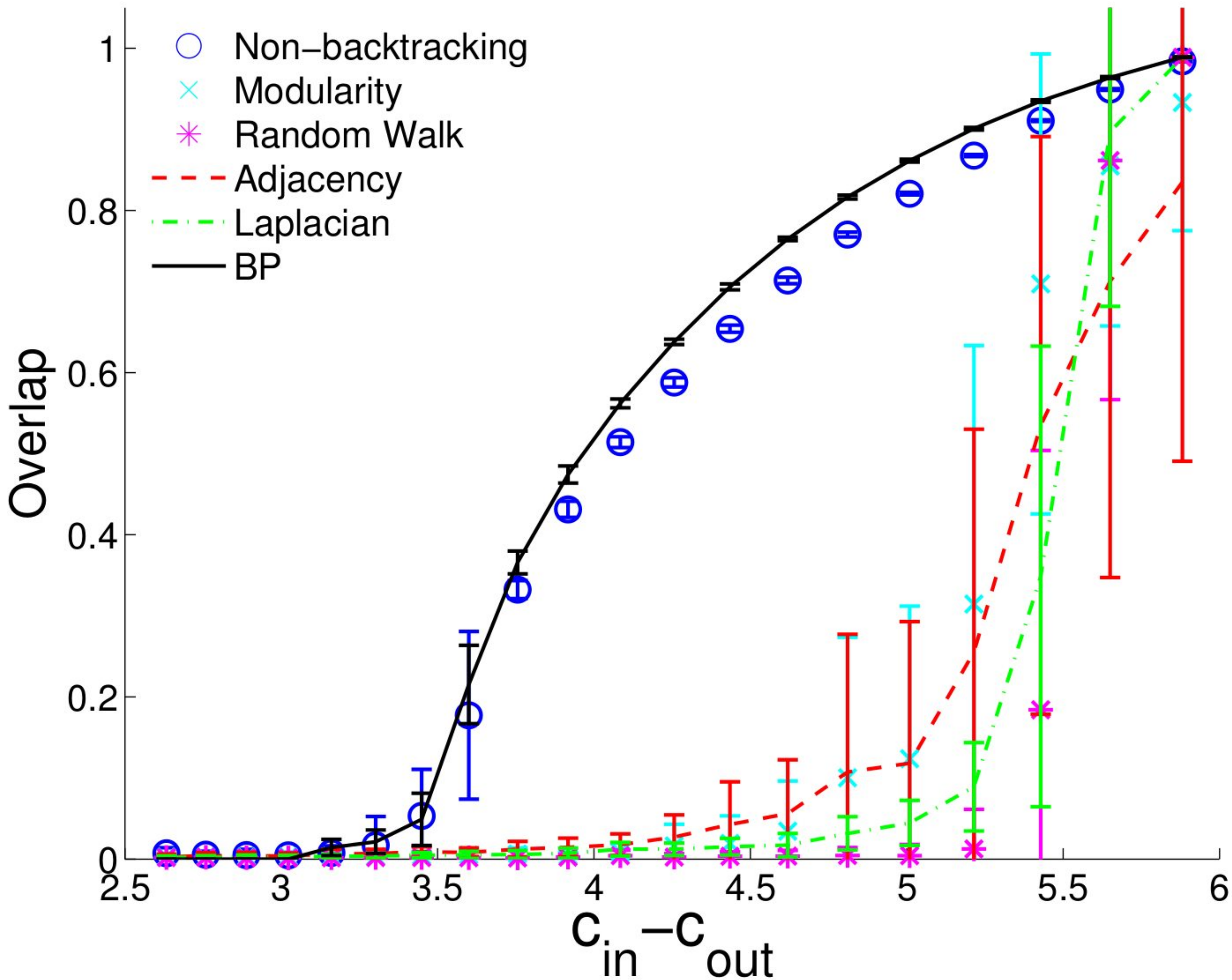
$$\mu_r^{i\to j} = \frac{1}{Z} \prod_{\substack{k\in\mathcal{N}(i)\\ k\neq j}} \sum_s \mu_s^{k\to i} \frac{\omega_{rs}^{A_{ik}}}{A_{ik}!} \mathrm{e}^{-\omega_{rs}}$$

- Each node assesses its own probabilities $\mu_r^{i\to j}$ to belong to each group based on the probabilities of its neighbors

Belief propagation for block models

- Efficient algorithm for the stochastic block model
  - Scales to millions of nodes
- Can be *linearized* to give a simpler algorithm, faster still
  - Equivalent to finding the leading eigenvector of a new matrix, the *non-backtracking matrix*
  - Gives better results for sparse networks
  - Appears to work all the way down to the limit of detectability

Non-backtracking matrix



Krzakala *et al.* 2013

Non-backtracking matrix

- Second eigenvector gives a good estimate of community structure
- First eigenvector gives an improved estimate of eigenvector centrality
- First eigenvalue gives the percolation threshold
- Also appears in the pair approximation for epidemic models on networks
- Also appears in iterative methods for calculating network spectra

- Thanks to:

Brian Ball

Aaron Clauset

Brian Karrer

Cris Moore

Lenka Zdeborová

- Funding from the National Science Foundation, the McDonnell Foundation, AFOSR, and DARPA