Identifying Types of Nodes in a Network

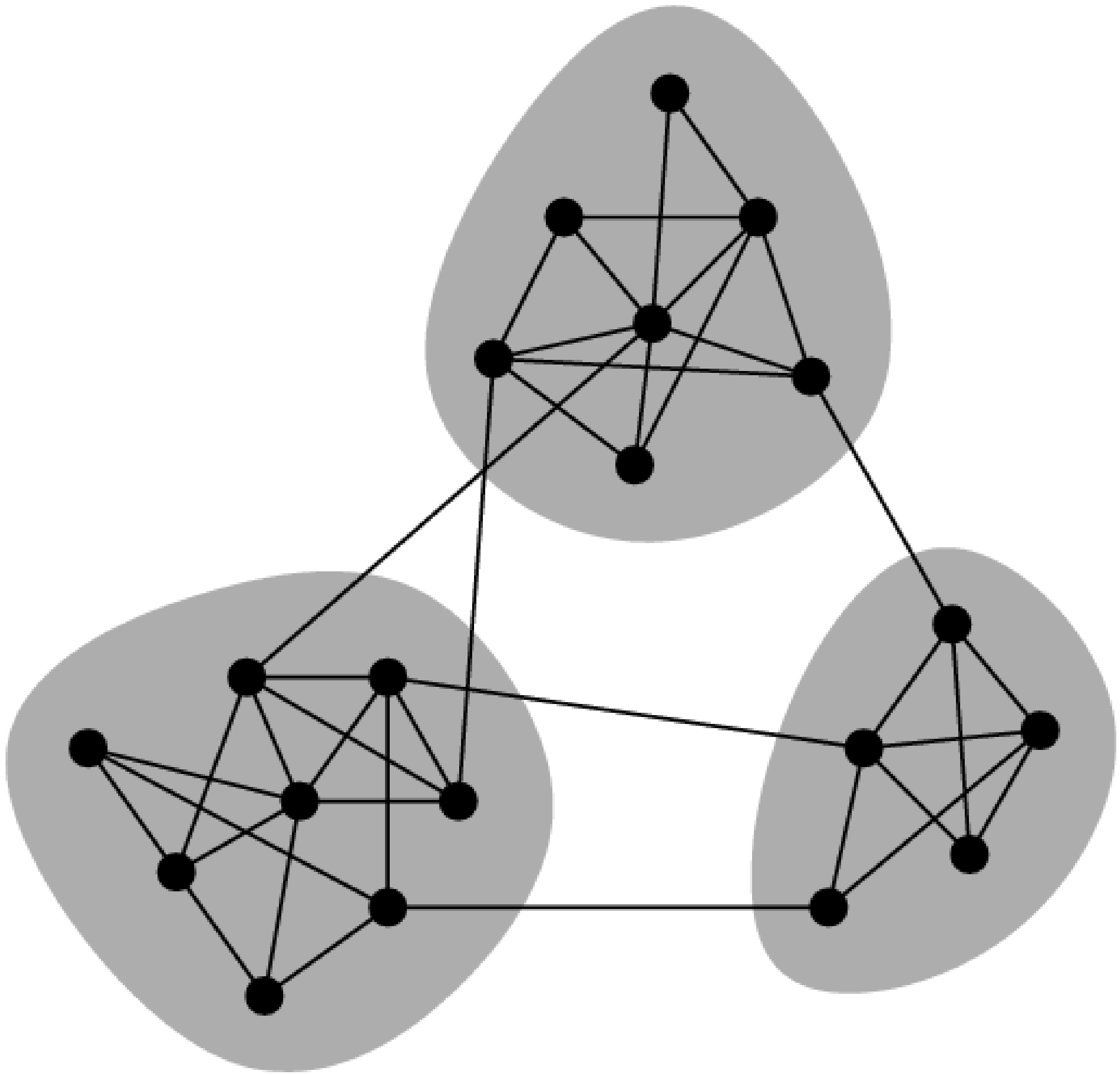
Mark Newman
University of Michigan and SFI

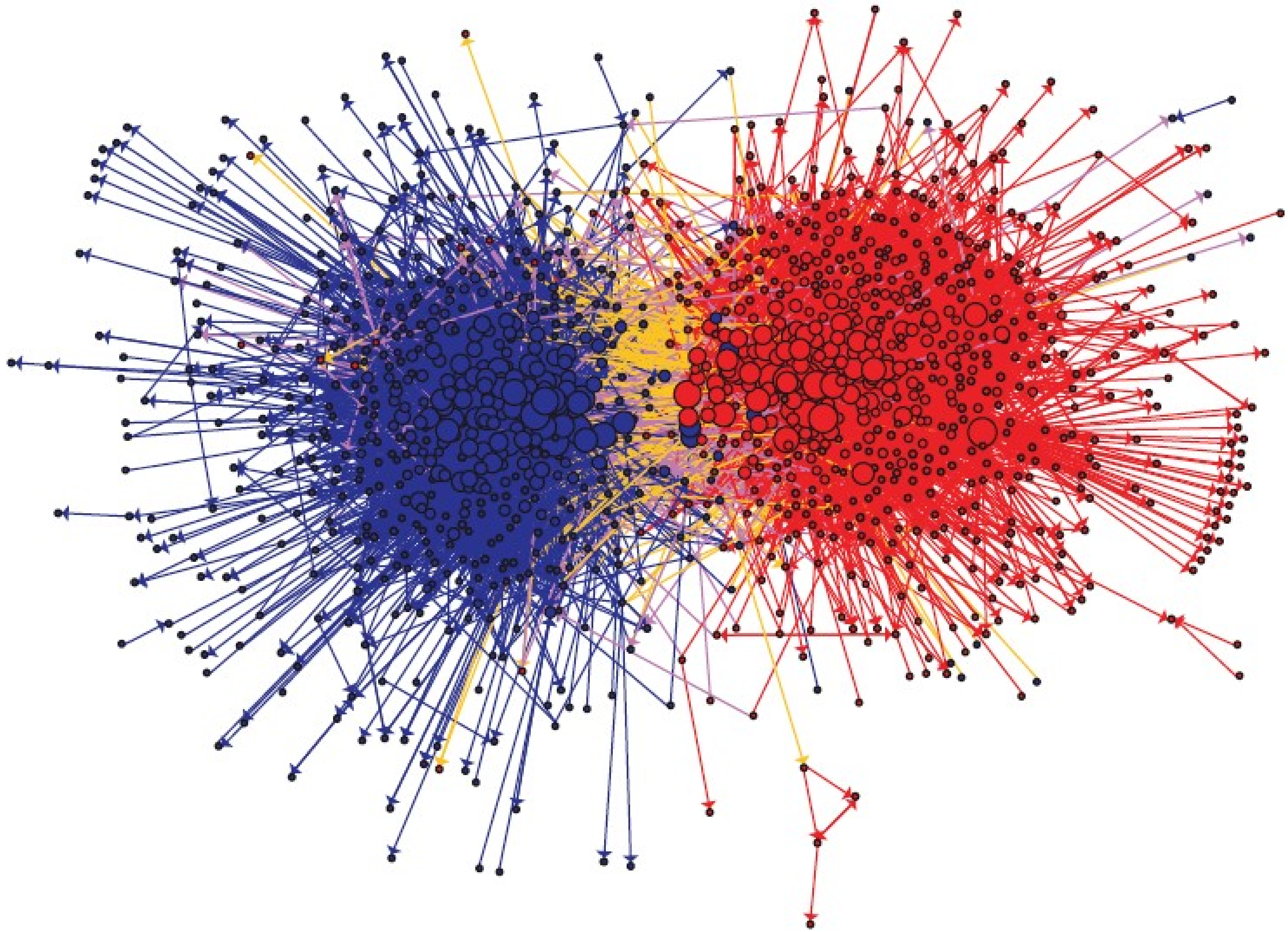
Joint work with Elizabeth Leicht (UC Davis)
and Gavin Clarkson & Kerby Shedden (Michigan)

Simple Network Statistics

- Numbers of vertices and edges
- Degree sequences or degree distributions
- Degree correlations
- Path lengths, diameter
- Transitivity, reciprocity
- Motif (subgraph) counts
- Centrality measures (eigenvector centralities, betweenness, etc.) and their distributions

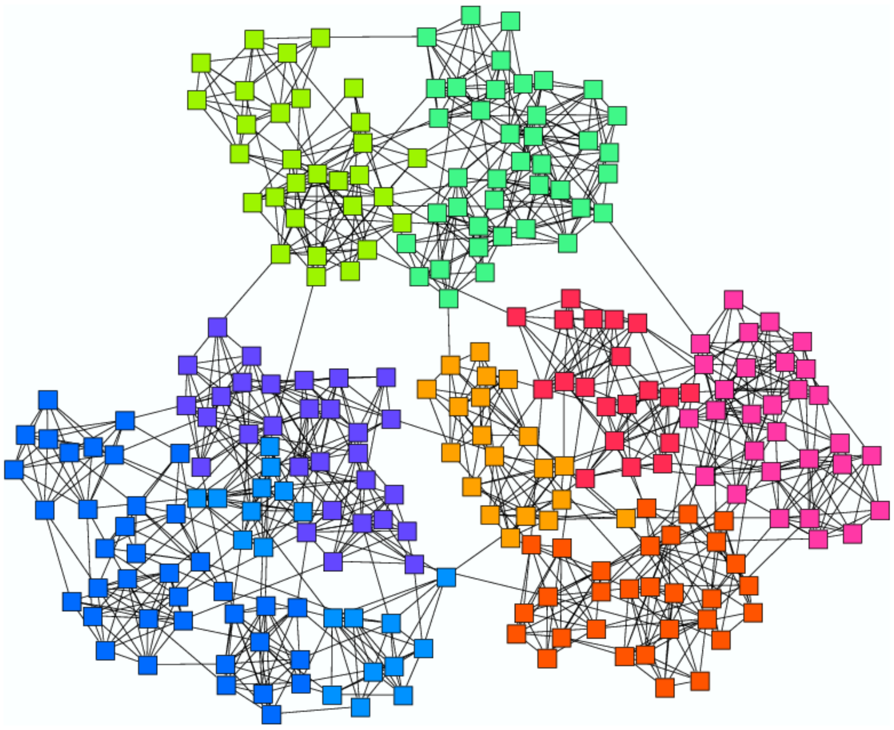
Modules, groups, or communities

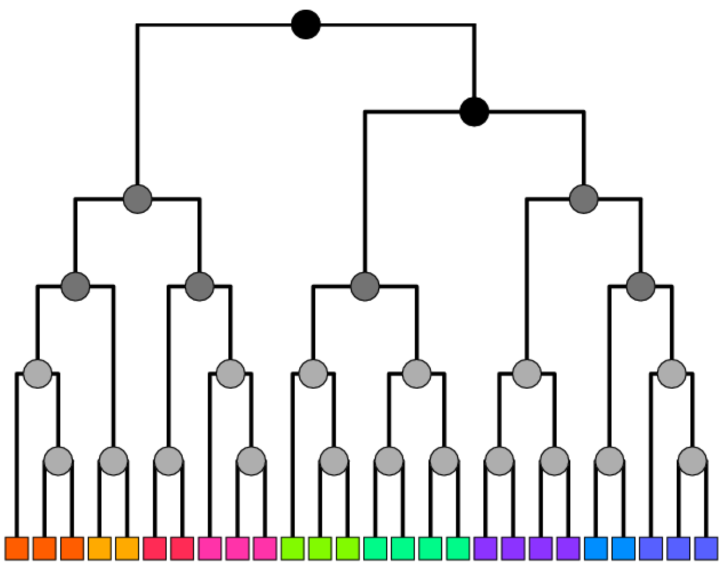




Adamic & Glance 2005

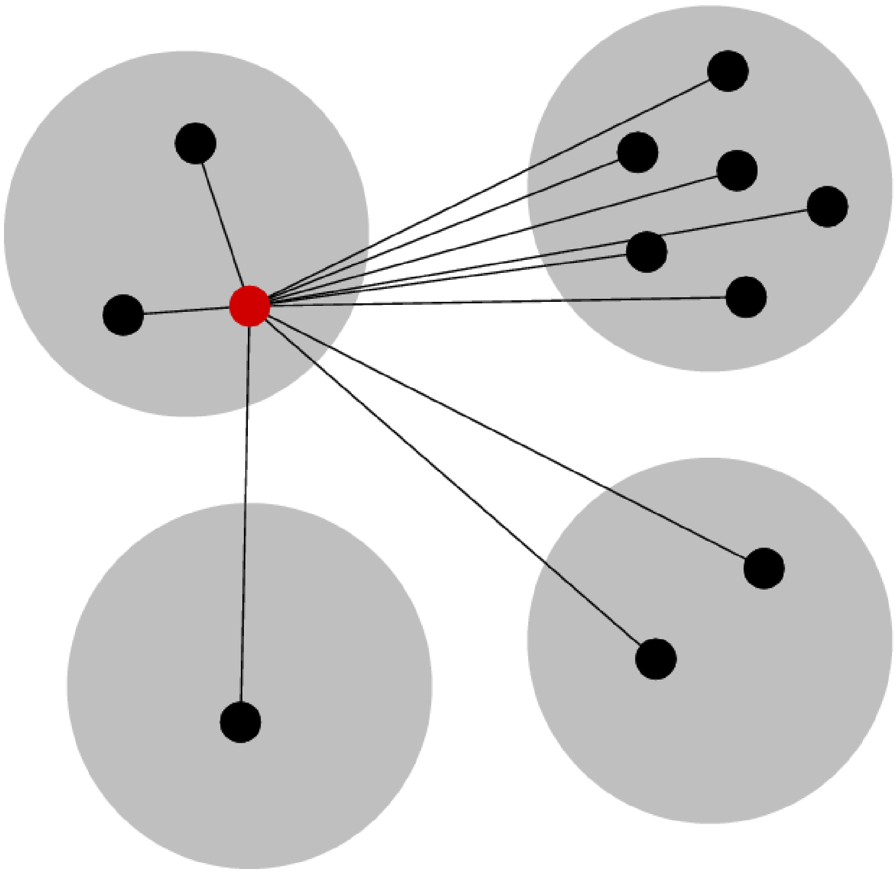
Network hierarchy





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

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

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

- 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 $\Pr(A, g | \pi, \theta)$ 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:

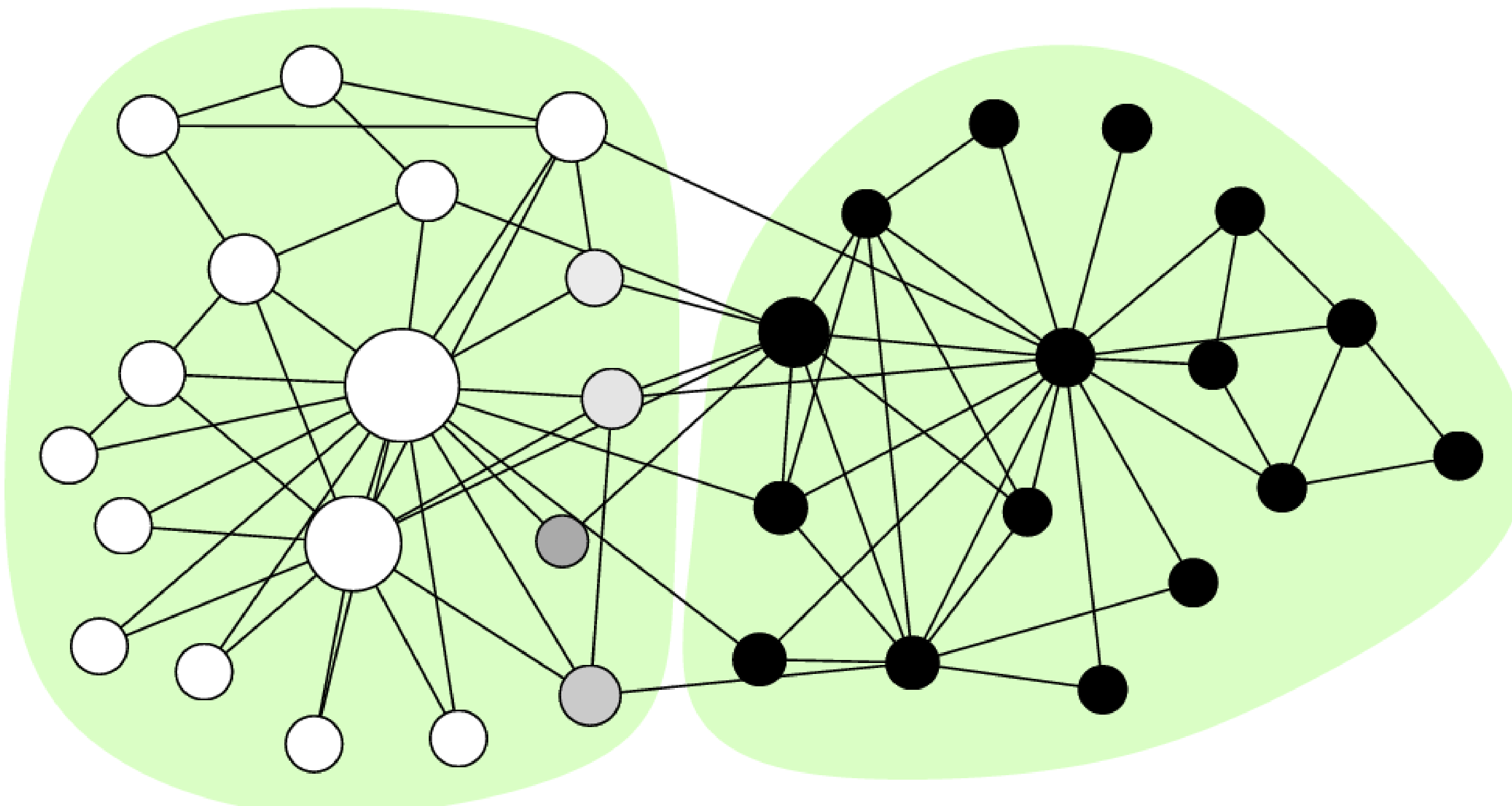
$$\pi_r = \frac{1}{n}\sum_i q_{ir}, \qquad \theta_{rj} = \frac{\sum_i A_{ij} q_{ir}}{\sum_i k_i q_{ir}},$$

- 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

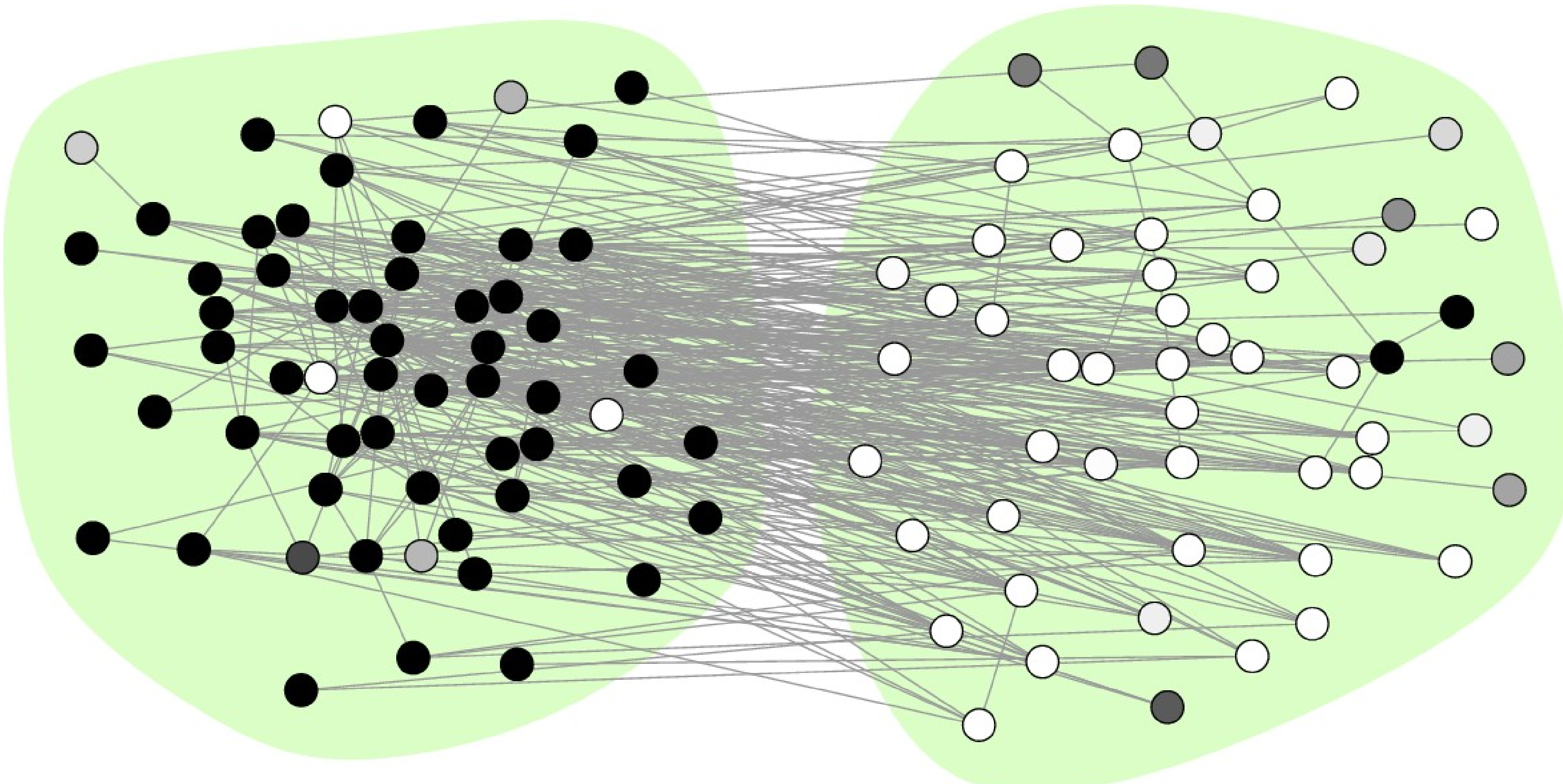
Expectation-Maximization Algorithm

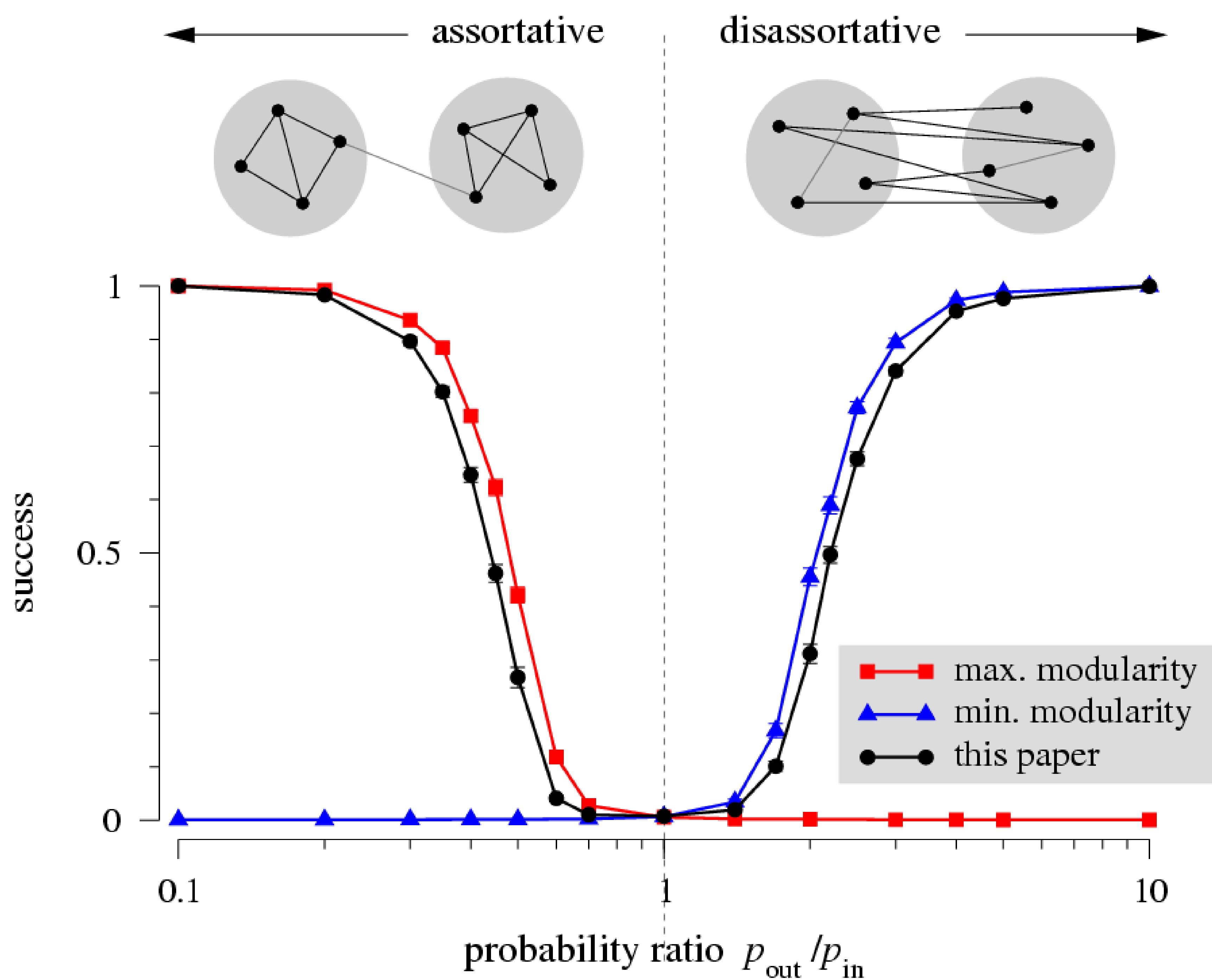
- Has a number of clear advantages:
  - Very simple: just a few lines of computer code to implement the method
  - Fast: typically only a few seconds to analyze even a large network
  - Simultaneously tells us how to group the vertices in the network and what the appropriate definition is for the groups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

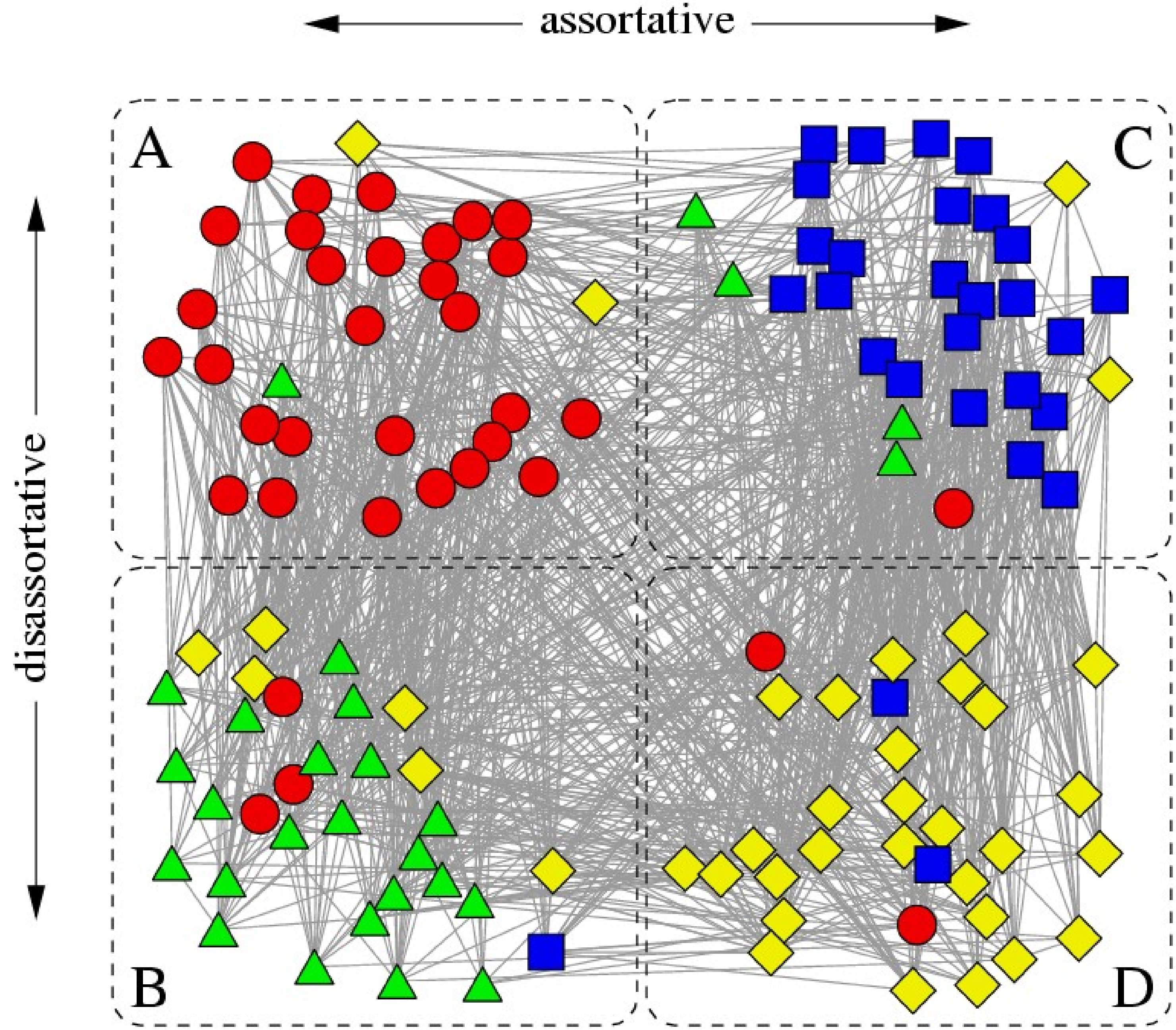
Example: Social network

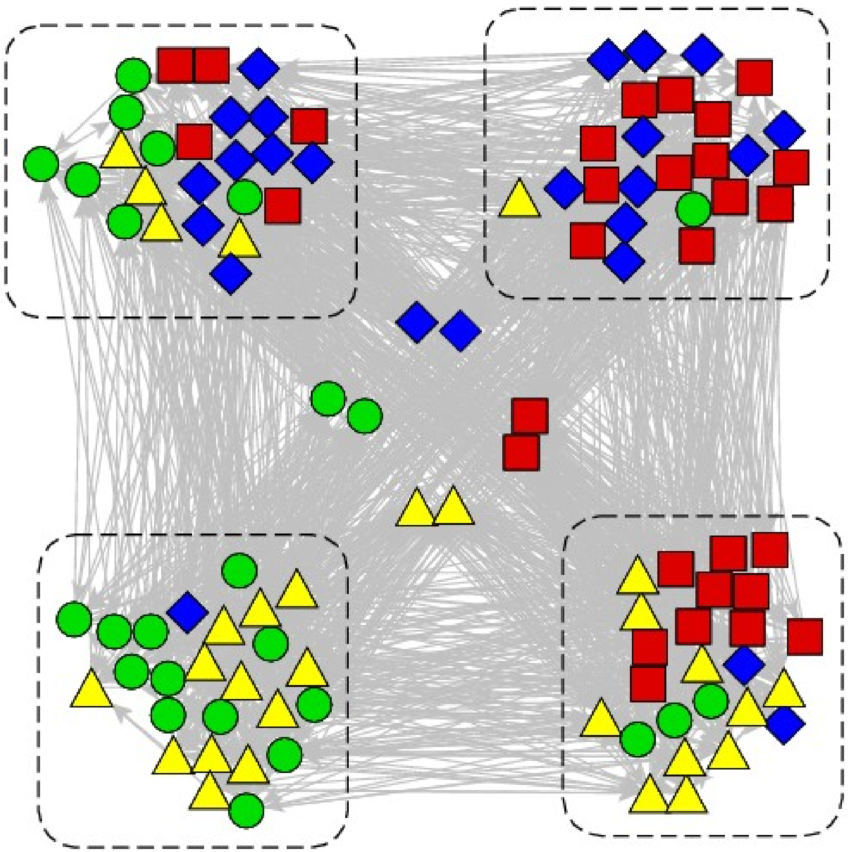


Example: Lexical network

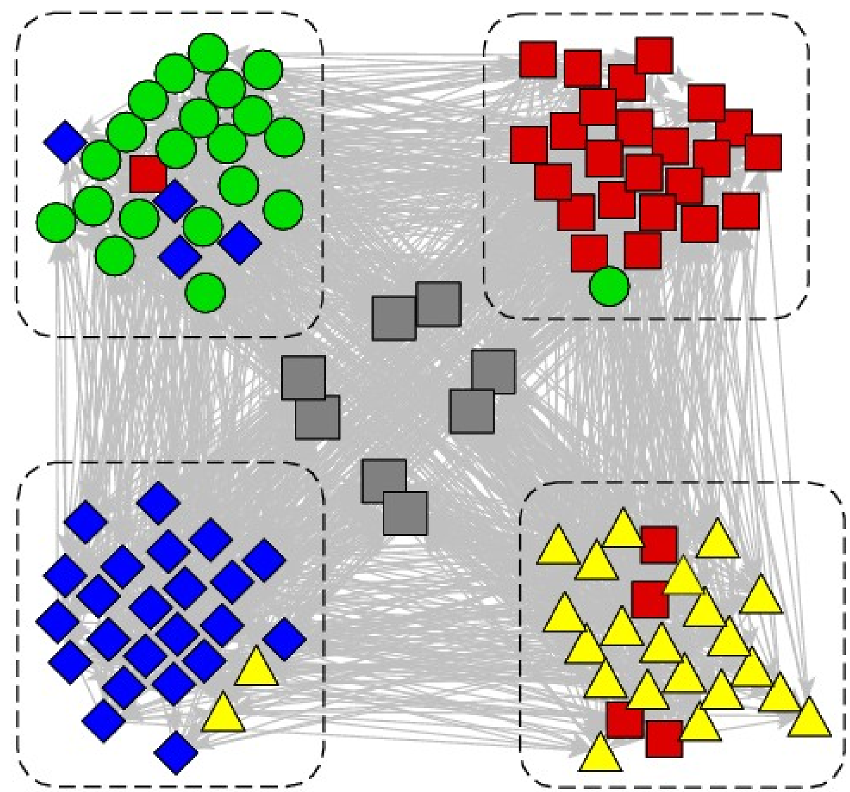



assortative
disassortative
1
0.5
0
success
0.1
1
10
probability ratio p_{out}/p_{in}
max. modularity
min. modularity
this paper


assortative
disassortative
A
B
C
D

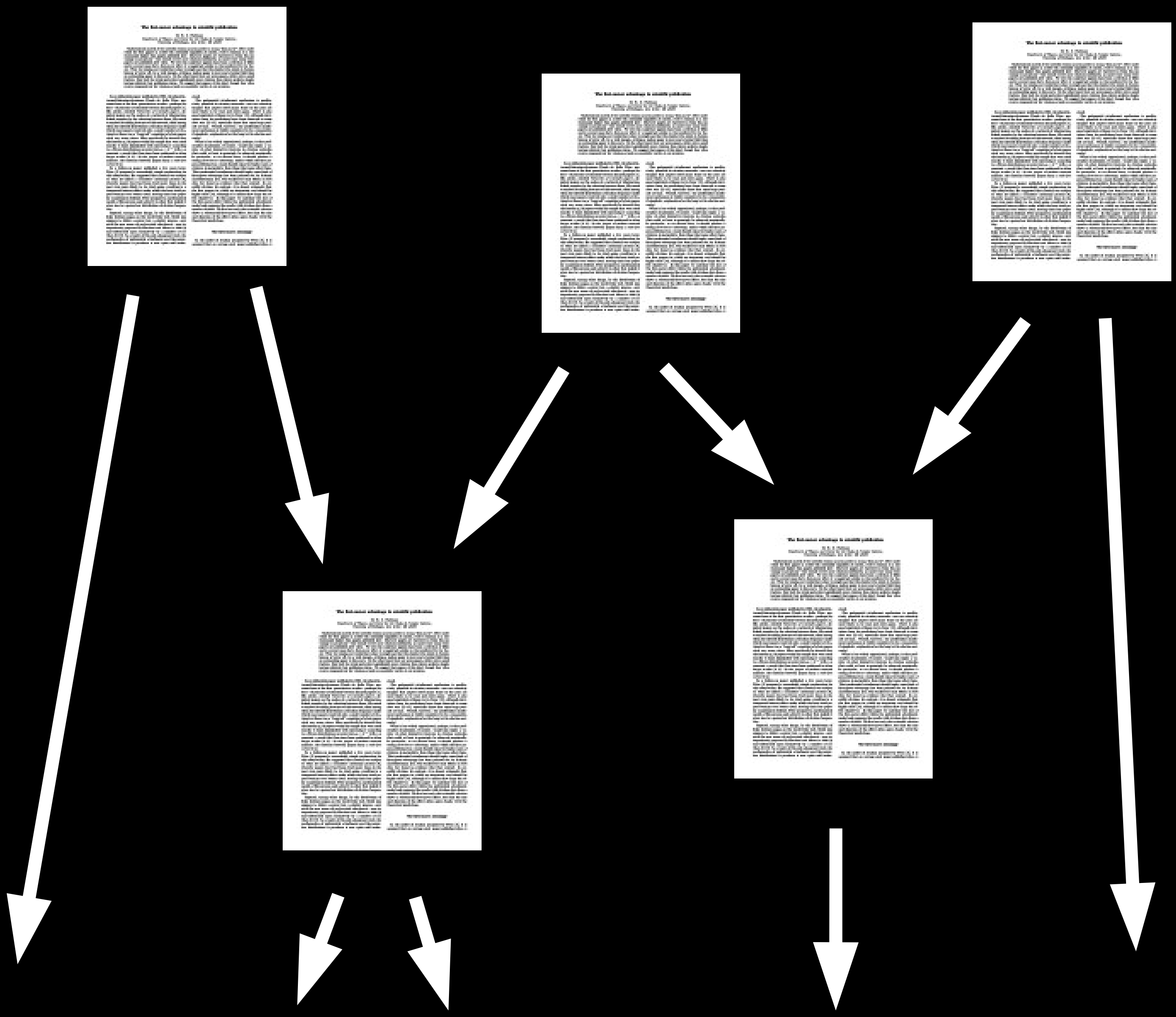


Ordinary community detection



EM algorithm

Citation networks



- The links in a citation network contain information about relations between subject matter (and possibly other connections, such as social connections)
  - Similar to World Wide Web, where these connections have successfully been mined by search engines such as Google
- Example:
  - Network of citations between cases heard by the US Supreme Court
  - About 30,000 decisions
  - Spans over 200 years, from 1789 to present day

EM algorithm

- Divide cases up into groups denoted by $r = 1, 2, 3 \ldots$
  - π_r = fraction of cases in group r
  - $\theta_r(t)$ = probability that a case in group r cites an opinion at time t
  - q_{ir} = probability that case i belongs to group r
- If we know π_r and $\theta_r(t)$, then

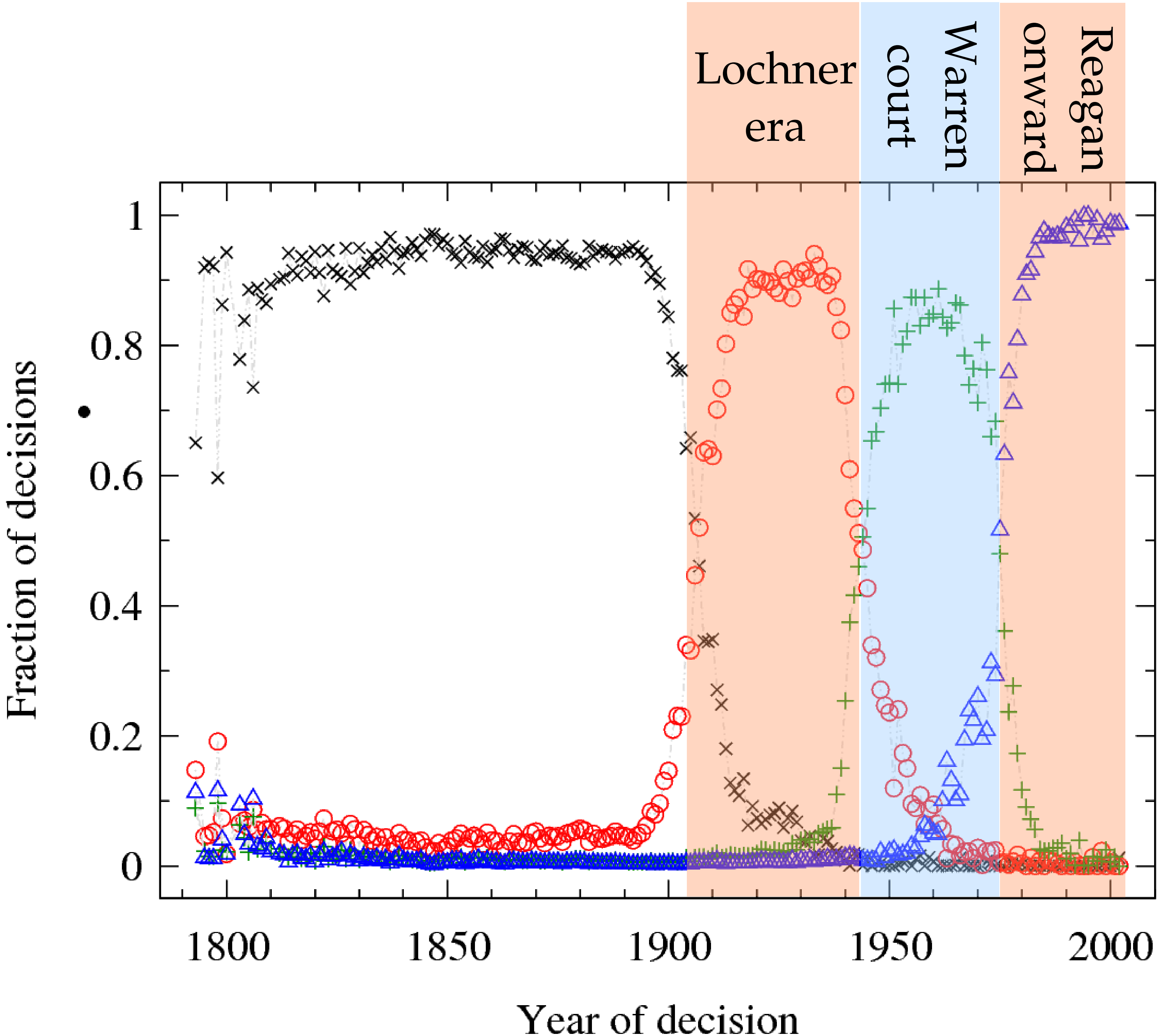
$$q_{ir} = \frac{\pi_r \prod_t \left[\theta_r(t)\right]^{z_i(t)}}{\sum_k \pi_k \prod_t \left[\theta_k(t)\right]^{z_i(t)}}$$

EM algorithm

- From maximum likelihood:

$$\pi_r = \frac{1}{n}\sum_i q_{ir}, \qquad \theta_r(t) = \frac{\sum_i q_{ir} z_i(t)}{\sum_i q_{ir} k_i}.$$

- Iterate from a random starting point until the equations converge
- End result:
  - Division of the equations into group
  - A definition of what the groups are



Lochner era
Warren court
Reagan onward
Fraction of decisions
Year of decision
0
0.2
0.4
0.6
0.8
1
1800
1850
1900
1950
2000

- References:

  - M. E. J. Newman and E. A. Leicht, *Proc. Natl. Acad. Sci.* **104**, 9564–9569 (2007)
  - E. A. Leicht, G. Clarkson, K. Shedden, and M. E. J. Newman, *Eur. Phys. J. B* **59**, 75–83 (2007)

- Thanks to:

  - Marian Boguña for useful input
  - NSF Applied Math and McDonnell Foundation for funding