Message-Passing Algorithms for Network Analysis

Cristopher Moore (UNM / Santa Fe Institute)
joint work with
Lenka Zdeborová, Florent Krzakała, Aurelien Decelle (CNRS)
Xiaoran Yan, Yaojia Zhu (UNM)
Mark Newman (Michigan)
and a brief mention of joint work with
Varsha Dani (UNM)
Anna Olson (Chicago)

Functional communities

assortative communities: vertices connect with others of the same type

but food webs, word adjacency networks, even some social networks have a more general kind of structure

a *functional* community or "module" is a set of vertices that connect to the rest of the network in similar ways

how do we find them? modularity, spectral approaches...

statistical inference: choose a class of generative models, and find the one most likely to generate the data

The stochastic block model

each vertex i has a type $t_i \in \{1,...,k\}$, with prior distribution $q_1,...,q_k$

$k \times k$ matrix p

if $t_i = r$ and $t_j = s$, there is an edge $i \to j$ with probability p_{rs}

p is not necessarily symmetric

we don't assume that $p_{rr} > p_{rs}$

given G, we want to infer the type assignment $t : V \to \{1,...,k\}$ and the matrix p

how do we get off the ground?

The likelihood

the probability of *G* given the types *t* and parameters θ=(*p*,*q*) is

$$P(G \mid t, \theta) = \prod_{(i,j)\in E} p_{t_i,t_j} \prod_{(i,j)\notin E} (1 - p_{t_i,t_j})$$

so the probability of *t* given *G* is

$$P(t \mid G, \theta) = \frac{P(t \mid \theta)\, P(G \mid t, \theta)}{\sum_{t' \in \{1,\ldots,k\}^n} P(G \mid t', \theta)}$$

$$\propto \prod_{i\in V} q_{t_i} \prod_{(i,j)\in E} p_{t_i,t_j} \prod_{(i,j)\notin E} (1 - p_{t_i,t_j})$$

call this the Gibbs distribution on *t*. How do we maximize it, or sample from it?

Maximizing the likelihood

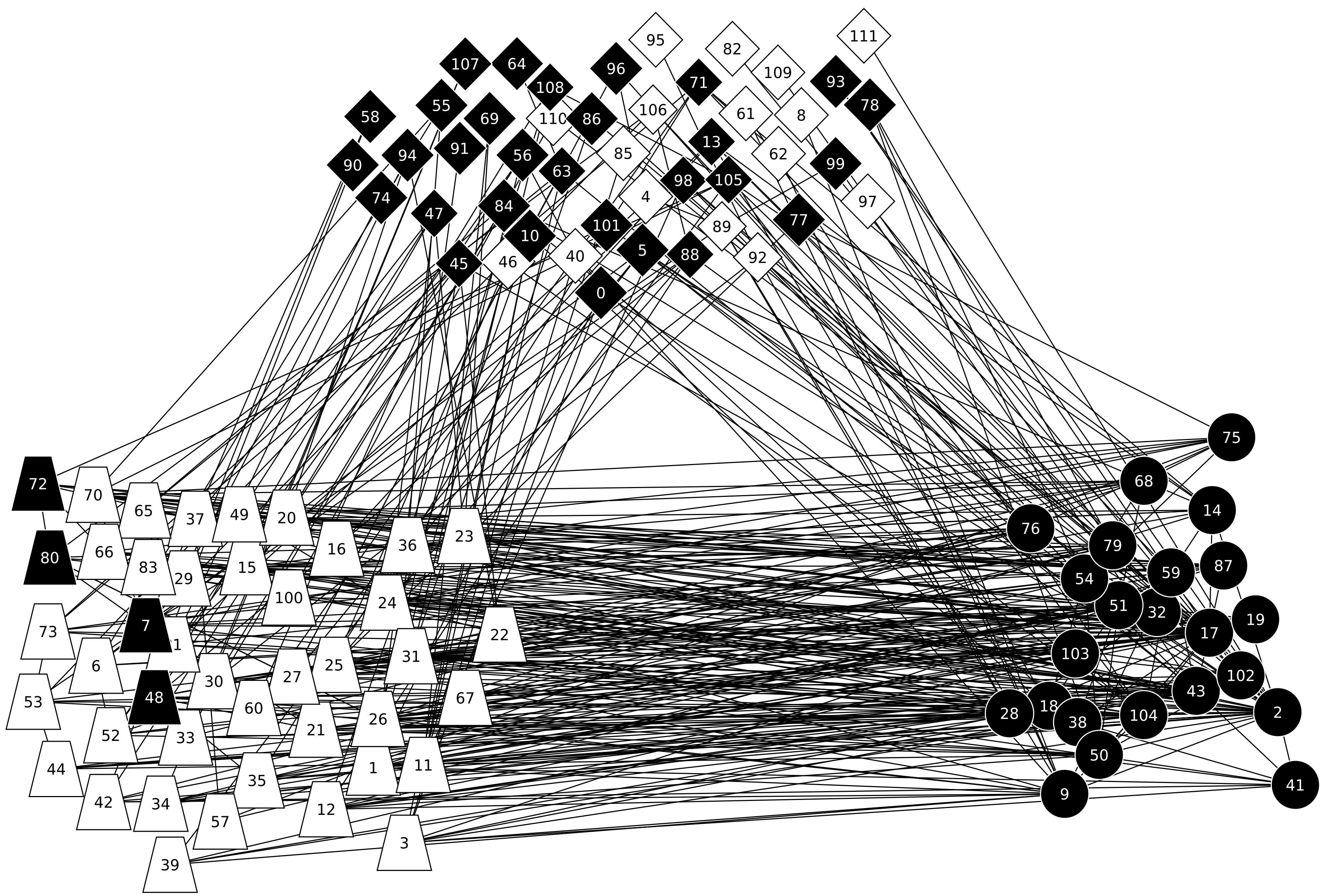
single-site heat-bath dynamics: choose a random vertex and update its type

if we like, we can jointly maximize $P(G|t,\theta)$ as a function of t and p by setting

$$p_{rs} = \frac{e_{rs}}{n_r n_s}, \quad q_r = \frac{n_r}{n}$$

this works reasonably well on small networks...

I record that I was born on a Friday



Maximizing the likelihood

single-site heat-bath dynamics: choose a random vertex and update its type

if we like, we can jointly maximize $P(G|t,p)$ as a function of t and p by setting

$$p_{rs} = \frac{e_{rs}}{n_r n_s}\,, \quad q_r = \frac{n_r}{n}$$

this works reasonably well on small networks... but it isn't really what we want

the probability of θ given G is a proportional to a partition function

$$P(G\,|\,\theta) = \sum_{t \in \{1,\ldots,k\}^n} P(G\,|\,t,\theta)$$

and –log $P(G|\theta)$ is a free energy, not a ground state energy

Maximizing the free energy

a several-line derivation shows that

$$\nabla_\theta \log P(G \mid \theta) = \sum_t P(t \mid G, \theta) \nabla_\theta \log P(t, G \mid \theta).$$

expectation-maximization (EM): given the current estimate $\hat{\theta}$, find the new θ that maximizes the expected log-likelihood

$$\sum_t P(t \mid G, \hat{\theta}) \log P(t, G \mid \theta)$$

then set $\hat{\theta} = \theta$ and iterate

but how to compute this expectation?

Marginals

to do the maximization, we don't need the entire Gibbs distribution

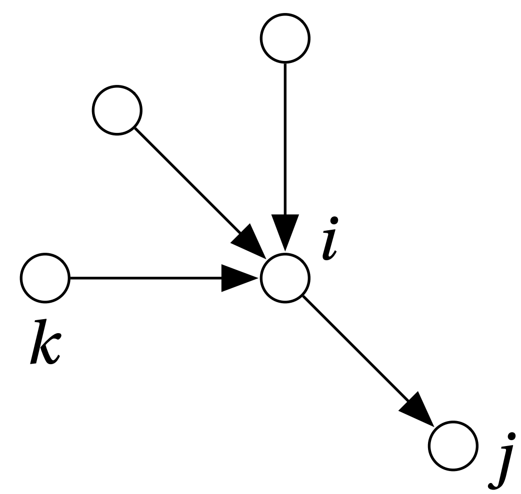
suppose we can estimate the one- and two-point marginals

$$\mu_r^i = \Pr[t_i = r]\,, \quad \mu_{rs}^{ij} = \Pr[t_i = r \text{ and } t_j = s]$$

then the expected log-likelihood is maximized by θ=(*p*,*q*) where

$$q_r = \frac{\sum_i \mu_r^i}{n}\,, \quad p_{rs} = \frac{\sum_{(i,j)\in E} \mu_{rs}^{ij}}{\sum_{i,j} \mu_{rs}^{ij}}$$

Belief propagation (a.k.a. the cavity method)

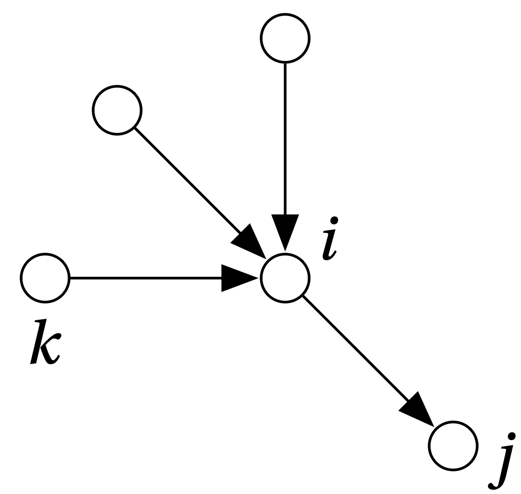


each vertex *i* sends a “message” to each of its neighbors j, giving *i*’s marginal distribution based on its other neighbors *k*

denote this message $\mu_r^{i\to j} = \text{estimate of } \Pr[t_i = r] \text{ if } j \text{ were absent}$

how do we update it?

Belief propagation (a.k.a. the cavity method)



conditional independence

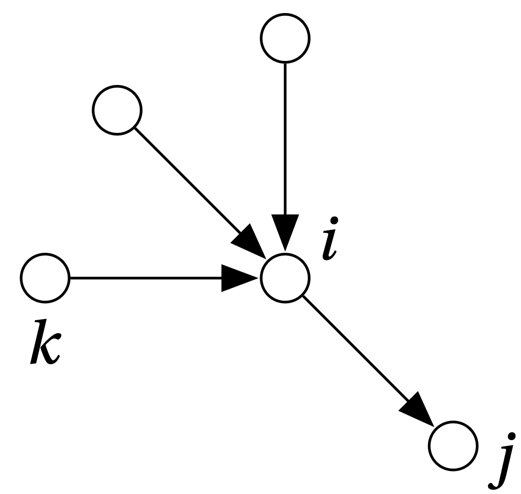
$$\mu_s^{i\to j} = \frac{1}{Z^{i\to j}}\, q_s \prod_{\substack{k\neq j\\(i,k)\in E}} \sum_r \mu_r^{k\to i} p_{rs} \times \prod_{\substack{k\neq j\\(i,k)\notin E}} \sum_r \mu_r^{k\to i}(1-p_{rs})$$

BP on a complete graph — takes O(n^2) time to update

can simplify by assuming that $\mu_r^{k\to i} = \mu_r^k$ for all non-neighbors *i*

each vertex *k* applies an "external field" $\sum_r \mu_r^k (1-p_{rs})$ to all vertices of type *s*

Belief propagation (a.k.a. the cavity method)



$$\mu_s^{i\to j} = \frac{1}{Z^{i\to j}}\, q_s \prod_{\substack{k\neq j\\(i,k)\in E}} \sum_r \mu_r^{k\to i} p_{rs} \times \frac{\prod_k \sum_r \mu_r^k (1-p_{rs})}{\prod_{k:(i,k)\in E} \sum_r \mu_r^k (1-p_{rs})}$$

each update now takes O(n+m) time

update until the messages reach a fixed point

From expectation to maximization

after the messages $\mu_r^{k\to i}$ reach a fixed point,

the two-point BP marginals are

$$\mu_{rs}^{ij} \propto \mu_r^{i\to j} \mu_s^{j\to i} \times \begin{cases} p_{rs} & (i,j) \in E \\ 1-p_{rs} & (i,j) \notin E \end{cases}$$

and we update θ=(p,q) to

$$q_r = \frac{\sum_i \mu_r^i}{n}\,, \quad p_{rs} = \frac{\sum_{(i,j)\in E} \mu_{st}^{ij}}{\sum_{i,j} \mu_{st}^{ij}}$$

EM: alternate expectation (through BP) and maximization to find θ and μ

The Bethe free energy

Bayes' rule implies

$$\log P(G\,|\,\theta) = \sum_t P(t\,|\,G,\theta)\ln P(G\,|\,t,\theta) - \sum_t P(t\,|\,G,\theta)\ln P(t\,|\,G,\theta)$$

physically, the free energy has an energy and entropy term, $F = U - TS$

the average energy *U*=E[-log *P*(*G*|*t*,θ)] is a function of the marginals μ_{rs}^{ij}

to compute the entropy *S* of the Gibbs distribution, we assume an approximate joint distribution based on the marginals for which we can compute *S* exactly,

$$P_{\text{Bethe}}(t\,|\,G,\theta) = \frac{\prod_{ij}\mu_{t_i,t_j}^{ij}}{\prod_i (\mu_{t_i}^{i})^{d_i-1}}$$

yields a surprisingly good approximation of *F*, even on finite graphs with loops; can compare with exact calculations and MCMC calorimetry

Performance on large synthetic networks

fix θ=(p,q) and take n=10^5 or so

choose type assignment t randomly according to q

generate edges randomly according to p

run the algorithm — how well does it do? given θ, does it find the right t? can it find θ using EM?

given the marginals $\mu_r^{k\to i}$, guess that $t_i = \mathrm{argmax}_r \, \mu_r^{k\to i}$

note: not the ground state!

define the *overlap* as the fraction of vertices labeled correctly...

...minus the size of the largest group, and normalized

Sparse benchmarks

set $q_i = 1/k$ for all i

let $p_{ij} = c_{ij}/n$ where $c_{ij} = c_{\text{in}}$ if $i=j$ and c_{out} if $i \neq j$

vary the ratio

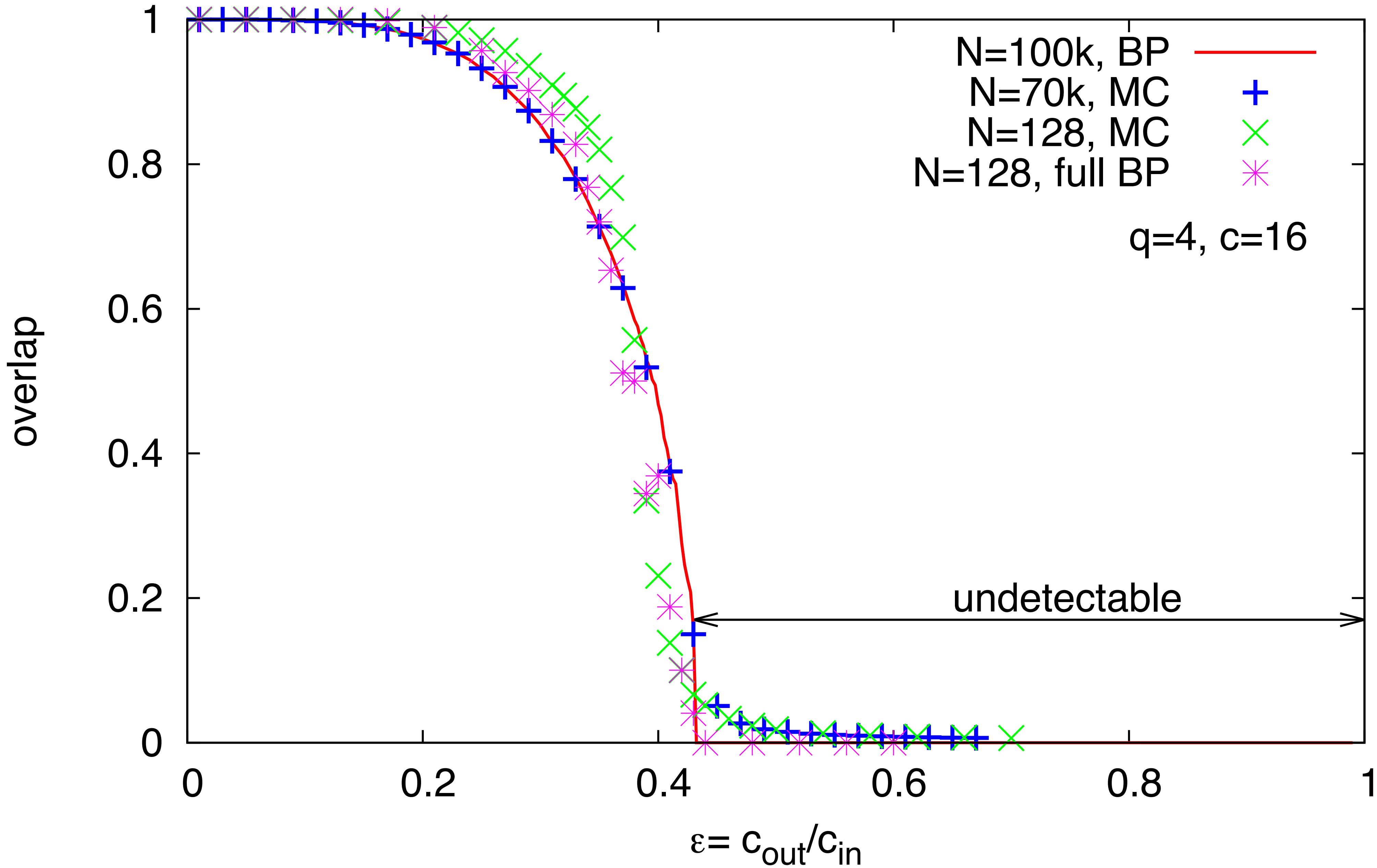
$$\varepsilon = \frac{c_{\text{out}}}{c_{\text{in}}}$$

while keeping the average degree $c = c_{\text{in}}/k + (1-1/k)c_{\text{out}}$ fixed

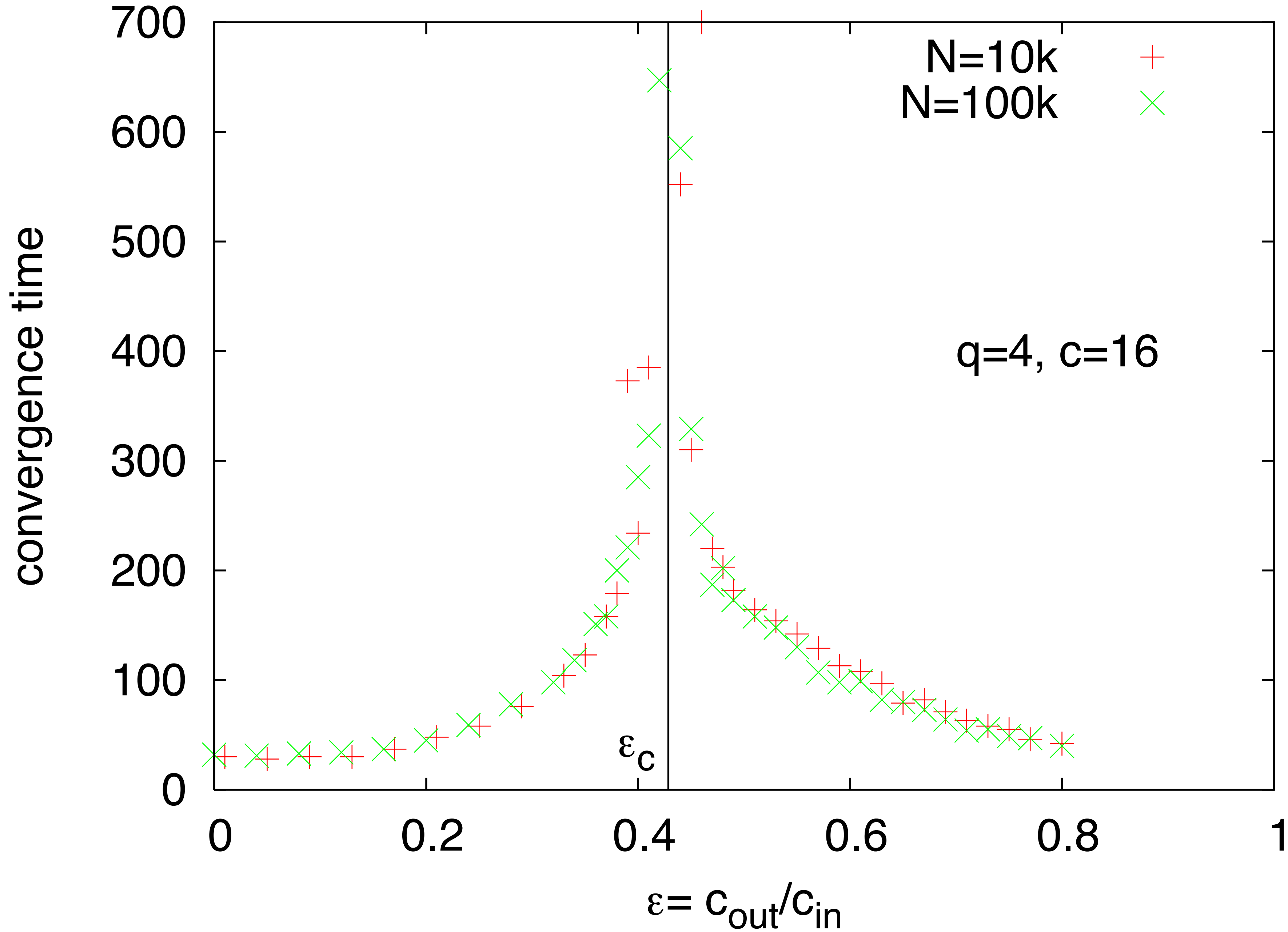
if ε is too close to 1, BP converges to the uniform fixed point

$$\mu_r^{i \to j} = \frac{1}{k}$$

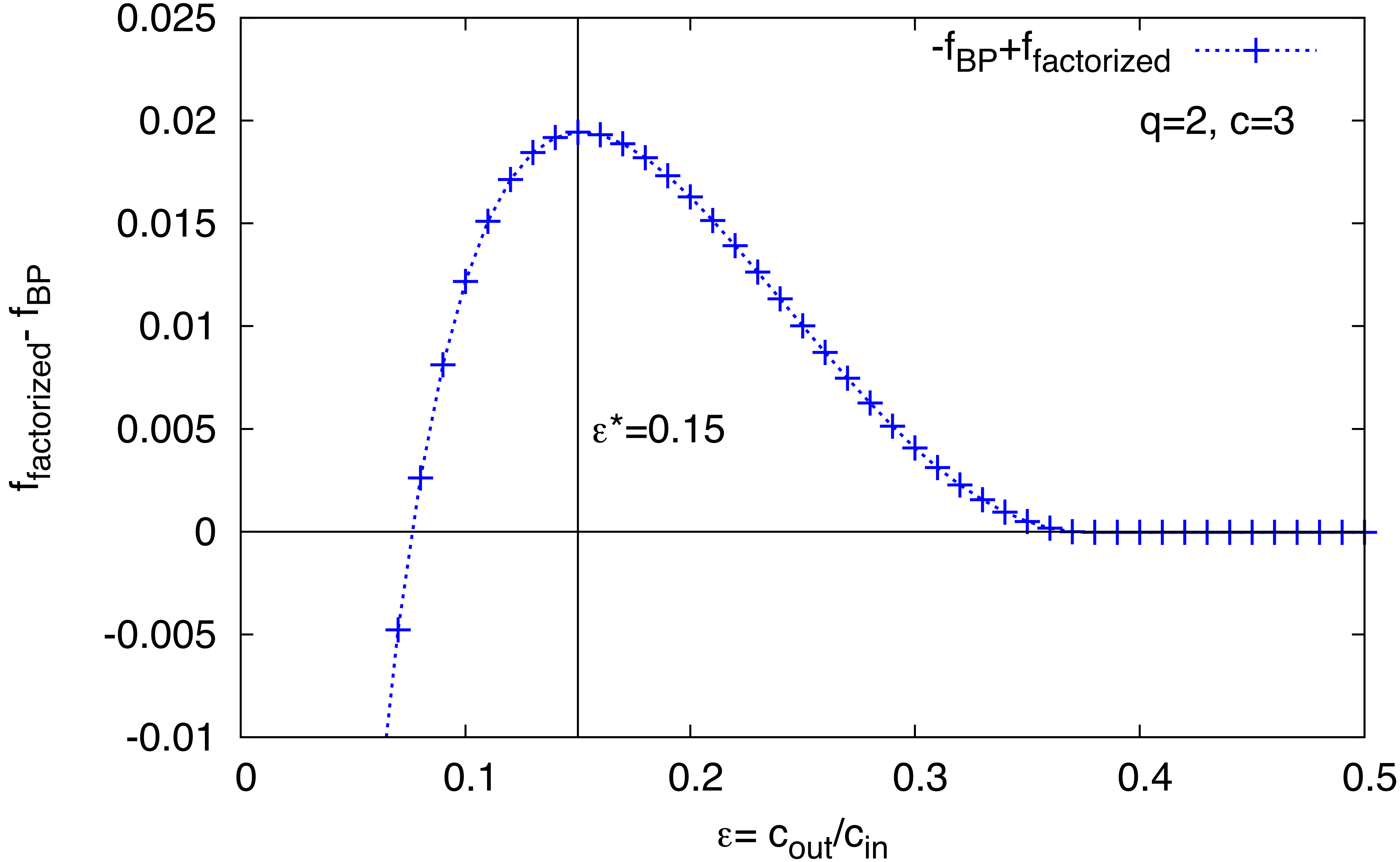
A phase transition from detectable to undetectable communities



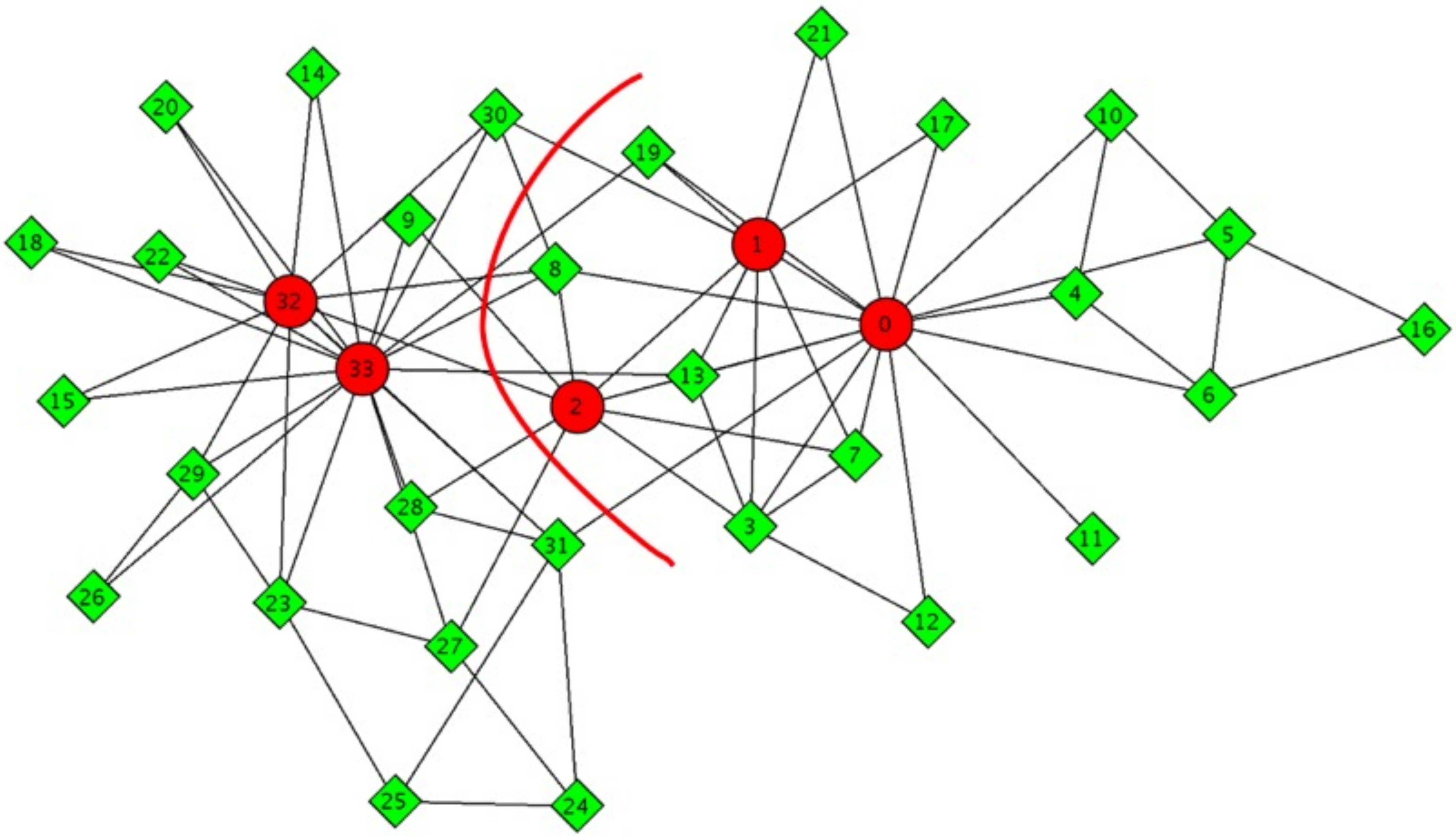
BP converges in a constant number of iterations



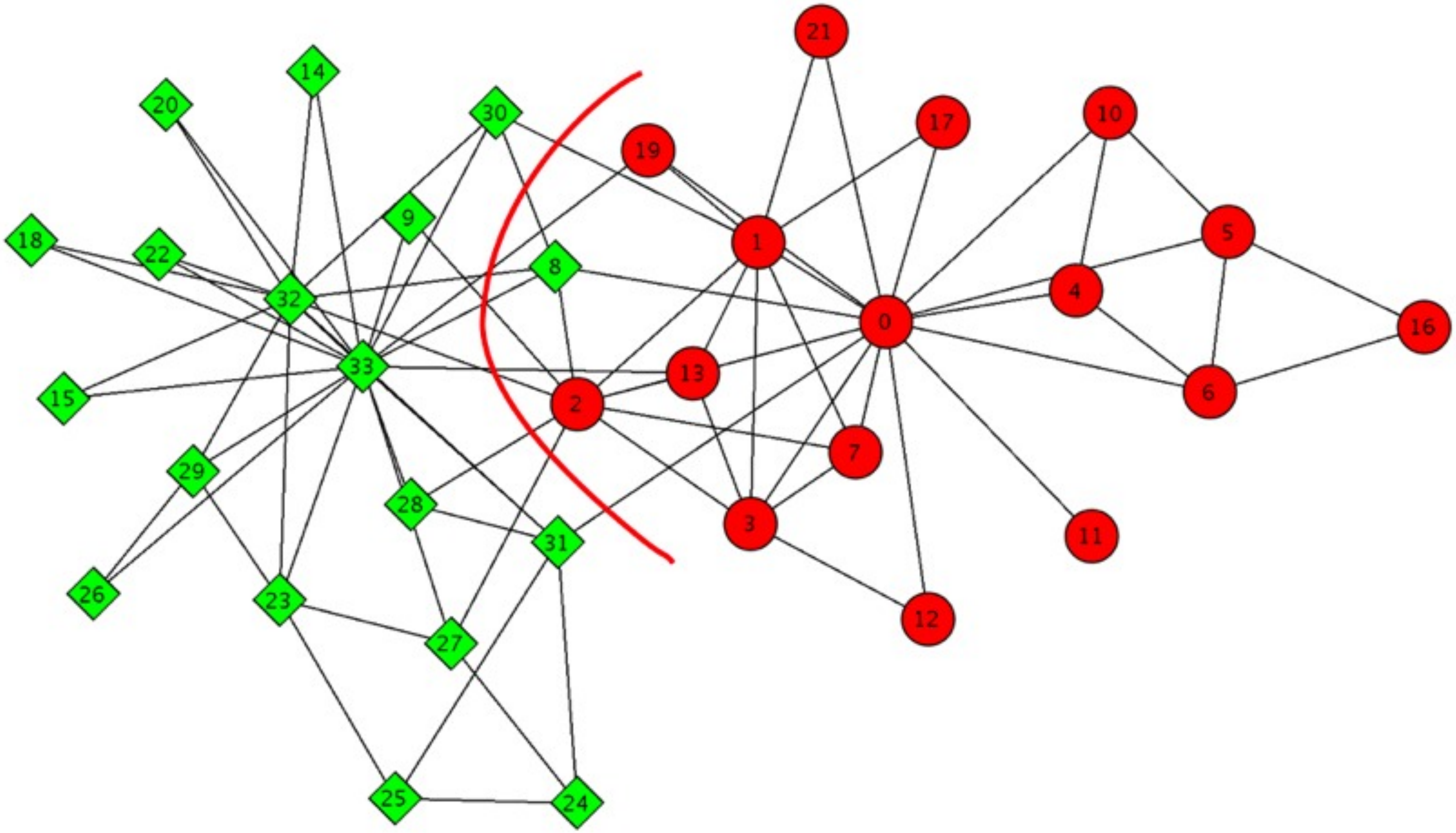
The free energy landscape



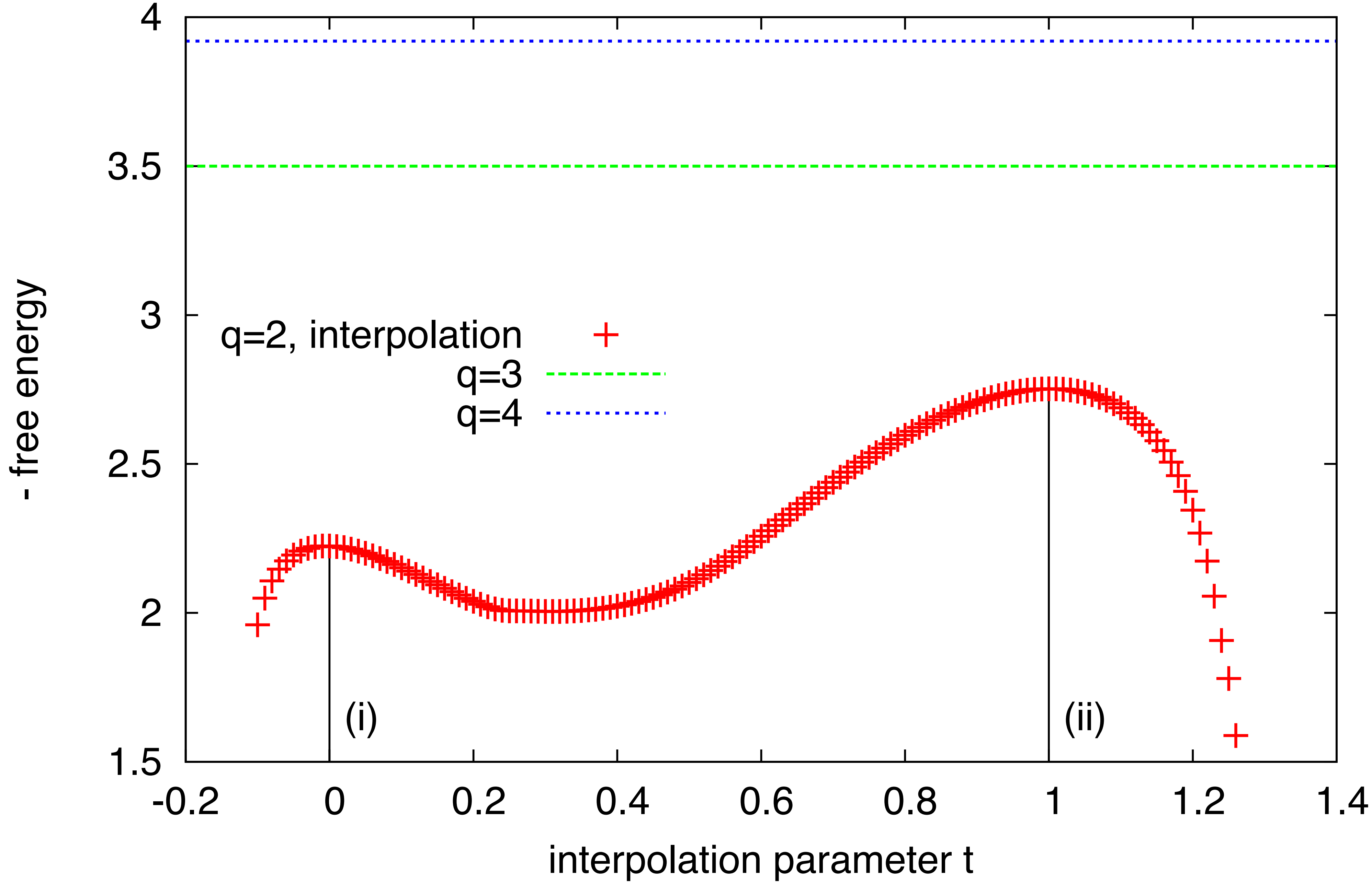
Which kind of community do you want?



Which kind of community do you want?



Two local optima



Degree-corrected block models

the “vanilla” block model expects vertices of the same type to have roughly the same degree

a random multigraph [Karrer & Newman, 2010]

each vertex i has an expected degree d_i

analogous to p_{ij}, a $k \times k$ matrix w_{ij}

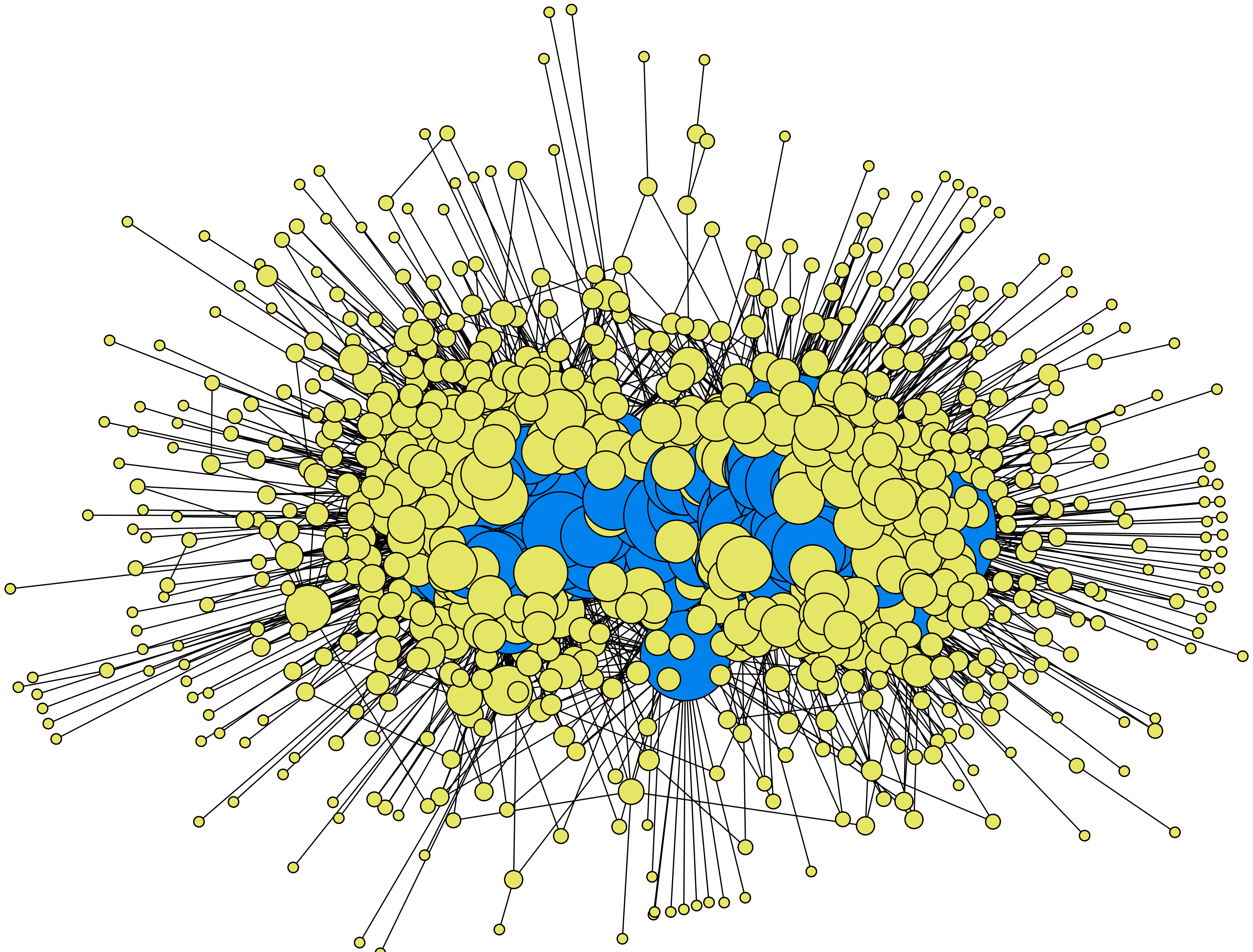
for each pair i, j with $t_i = r$ and $t_j = s$, the number of edges between them is

$$m_{ij} = \mathrm{Poi}(d_i d_j w_{rs})$$

now the degrees are parameters, not data to be explained

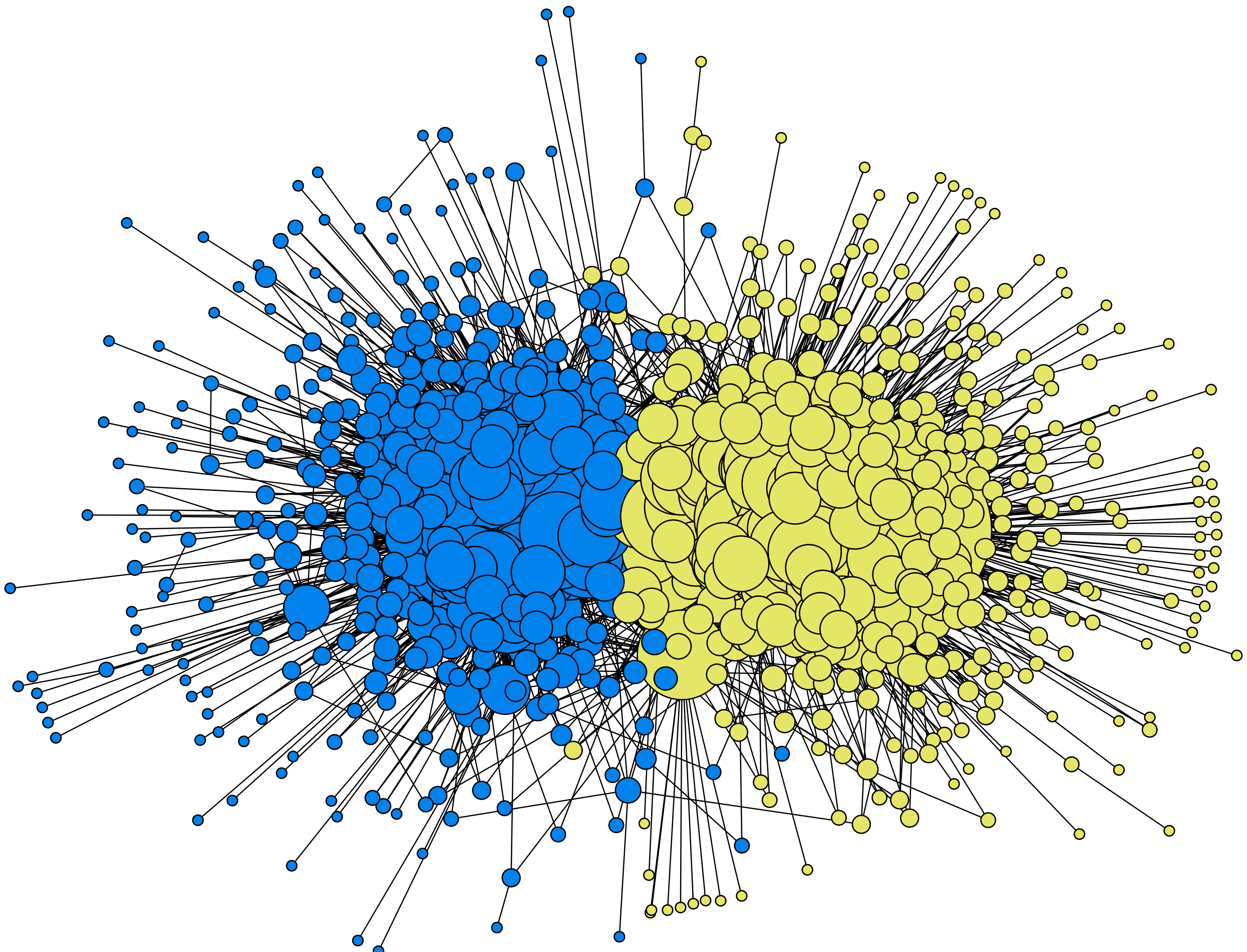
can again write down the BP/EM algorithm

Blogs: vanilla block model



[Karrer & Newman, 2010]

Blogs: degree-corrected block model



[Karrer & Newman, 2010]

Strengths and weaknesses

degree-corrected models don't mind inhomogeneous degree distributions

but they also can't use the degrees to help them label the nodes

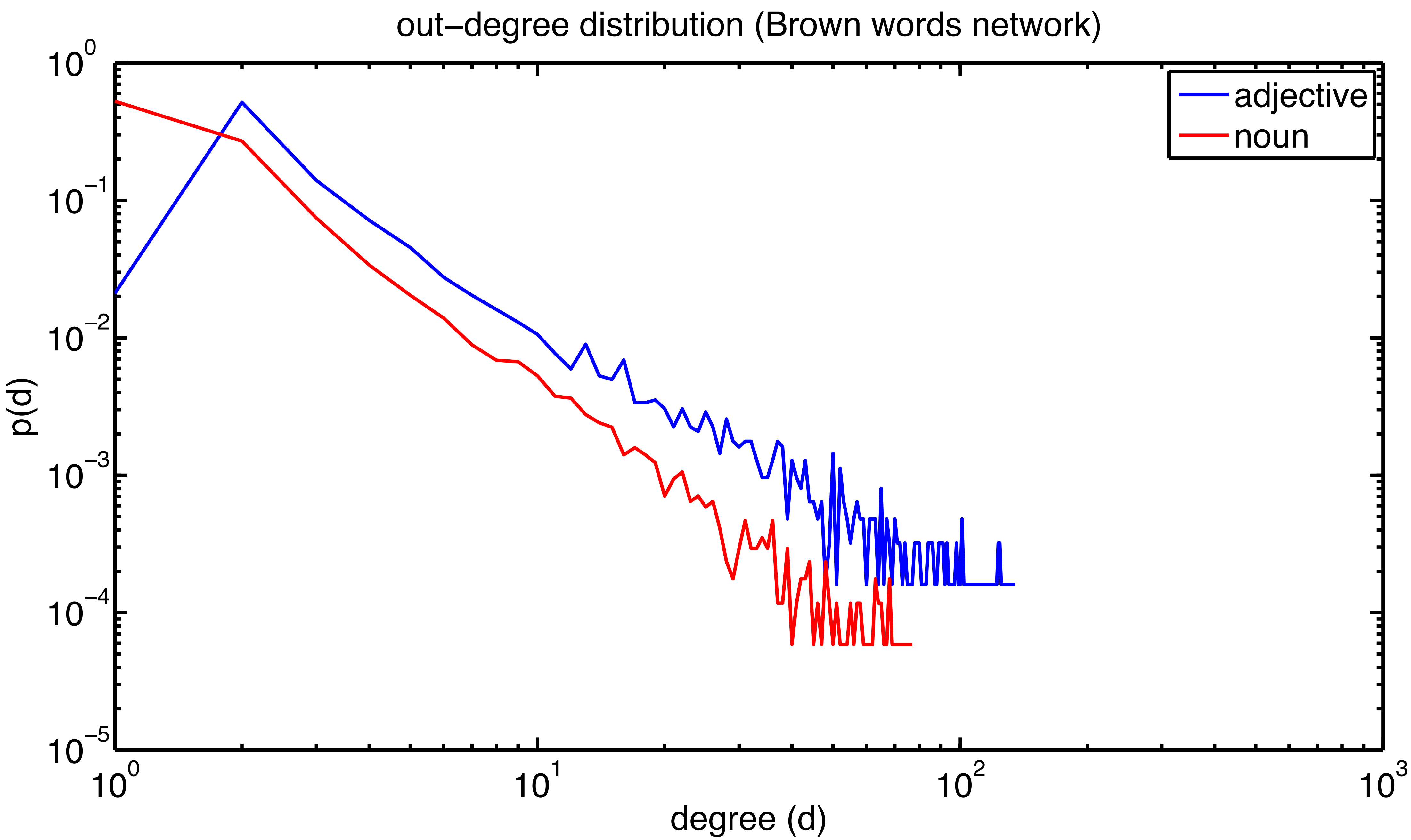
on some networks, they perform worse than the vanilla model

yet another model: first generate vertex degrees d_i according to some distribution whose parameters depend on t_i (e.g. power law)

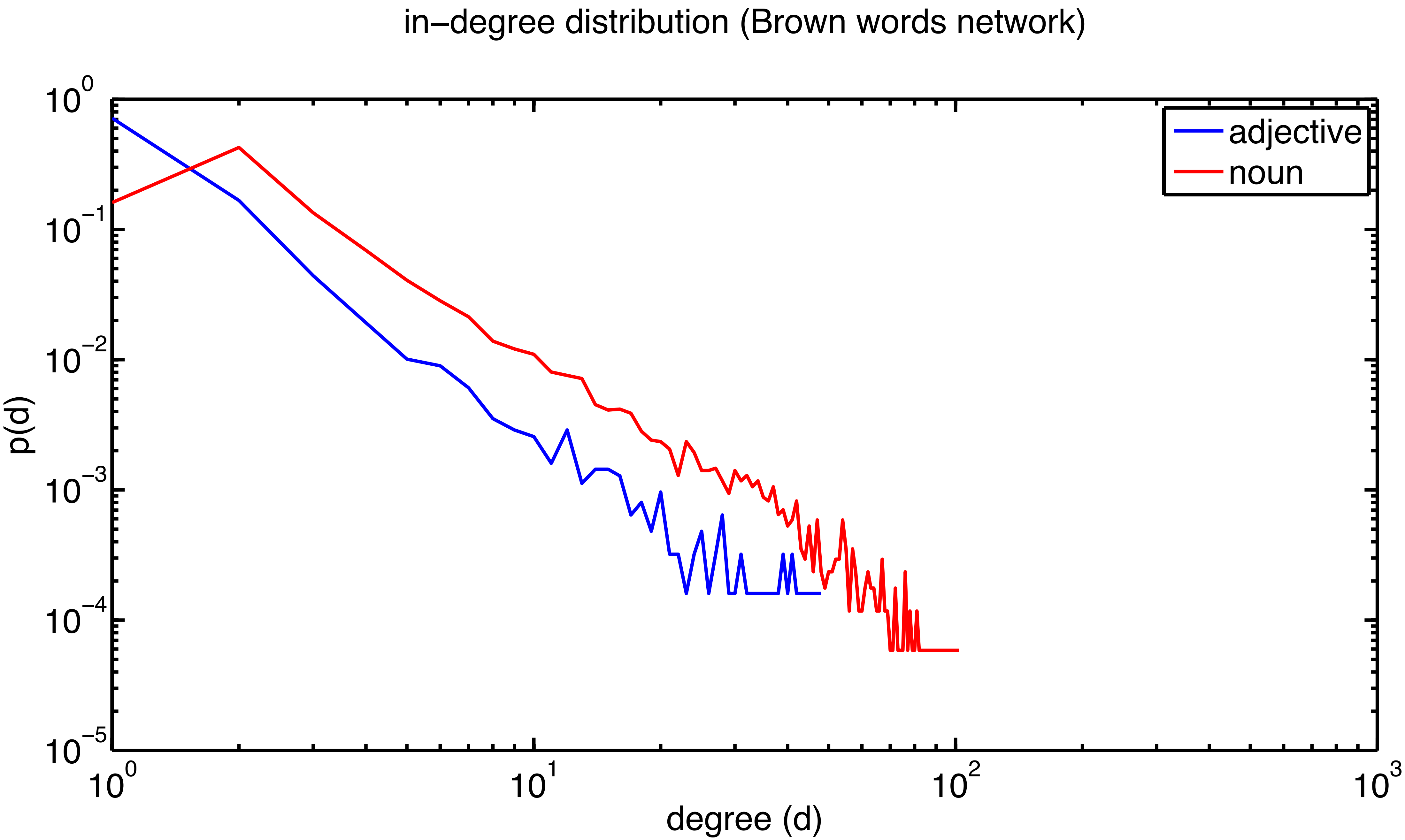
then generate edges according to the degree-corrected model

for some networks (e.g. large word networks) works better than either vanilla or degree-corrected model

Degree-generated model



Degree-generated model



Colorings with permutations

Threshold conjecture for *k*-colorability:

$$\lim_{n\to\infty} \Pr[G(n, p = d/n) \text{ is } k\text{-colorable}] = \begin{cases} 1 & \text{if } d < d_k \\ 0 & \text{if } d > d_k \end{cases}$$

Achlioptas and Naor determined d_k to within O(log k) (and determined *k* as a function of *d* to two integers)

Conjecture: the threshold d_k stays the same if we put a random permutation $\pi \in S_k$ on each edge, and demand that $c(u) \neq \pi(c(v))$ instead of $c(u) \neq c(v)$

Justification: correlation decay, reconstruction, survey propagation

Second moment calculations are much easier, letting us bound d_c within an additive constant [Dani, Moore, Olson]: for any ε and sufficiently large *k*,

$$2k \ln k - \ln k - 2 - \varepsilon \leq d_k \leq 2k \ln k - \ln k - 1 + \varepsilon$$

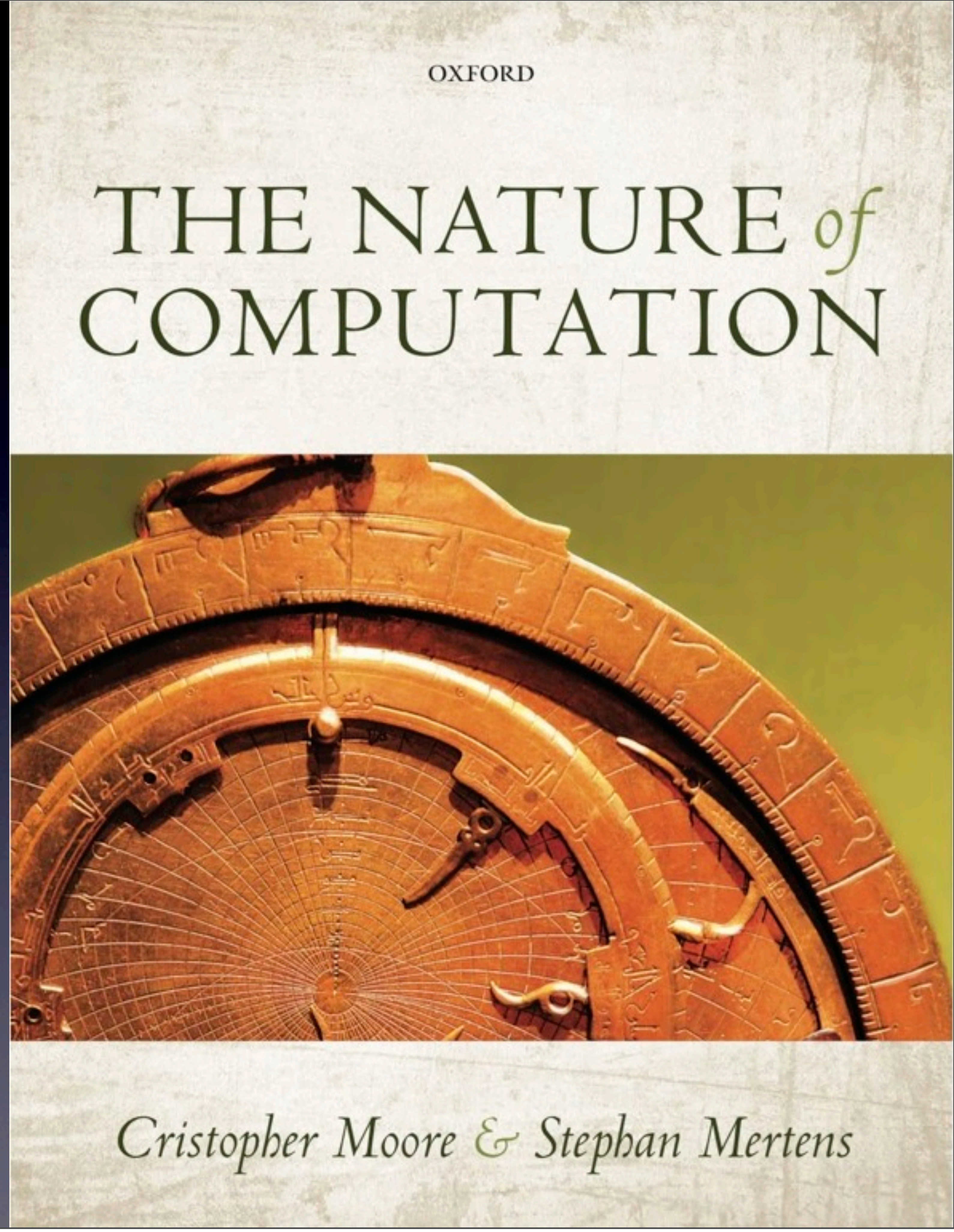
Shameless Plug

This book rocks! You somehow manage to combine the fun of a popular book with the intellectual heft of a textbook.
— Scott Aaronson

A treasure trove of information on algorithms and complexity, presented in the most delightful way.
— Vijay Vazirani

A creative, insightful, and accessible introduction to the theory of computing, written with a keen eye toward the frontiers of the field and a vivid enthusiasm for the subject matter.
— Jon Kleinberg

Oxford University Press, 2011



Acknowledgments

and the McDonnell Foundation