Statistical Inference in Networks

Cristopher Moore, Santa Fe Institute

joint work (over the years) with
Xiaoran Yan, Yaojia Zhu, Lenka Zdeborová, Florent Krzakala, Aurelien Decelle, Pan Zhang,
Jean-Baptiste Rouquier, Tiffany Pierce, Cosma Shalizi, Jacob Jensen, Lise Getoor,
Aaron Clauset, Mark Newman, Elchanan Mossel, Joe Neeman, and Allan Sly

Big data needs big algorithms

We're swimming in data: the challenge is doing something with it

Finding simple trends isn't enough: we need to find structures and patterns in this data, that let us

understand it

predict it

generalize from what we know to what we don't

We need algorithms that do this automatically (often with a human in the loop)

These algorithms need to be scalable: on a data set of size n, taking 2^n time, or even n^3 time, is too slow

Not just computing power! Moore's law isn't enough—we need mathematical insight to avoid/simplify the search

What is structure?

Structure is that which...

makes data different from noise: makes a network different from a random graph, from a background “null model”

helps us compress the data: describe the network succinctly, giving a human-readable summary of important structures

helps us generalize from data we’ve seen from data we haven’t seen: e.g. predict missing links from the links we know about

helps us understand what multiple networks have in common: e.g. structure of food webs, from the Cambrian to today

helps us coarse-grain the dynamics, reducing the number of variables: e.g. compartmentalized models in epidemiology

Statistical inference

Imagine that the network is created by a *generative model*, and fit the parameters of this model to the data

Use whatever (partial, noisy) information we have to constrain the search...

- attributes of some nodes are known, or known with some confidence
- some links are known, others not observed yet (e.g. food webs)
- some links might be false positives (e.g. gene regulatory networks, protein interactions)

...and make good guesses about the information we don't have:

- label unknown nodes
- predict missing links
- identify anomalies

The stochastic block model

k types of nodes

we know the links between the nodes, but not their types

assumption: probability that two nodes are linked depends only on their types

assortativity / homophily: nodes connect more to others of the same type

disassortativity / heterophily: links between types instead of within

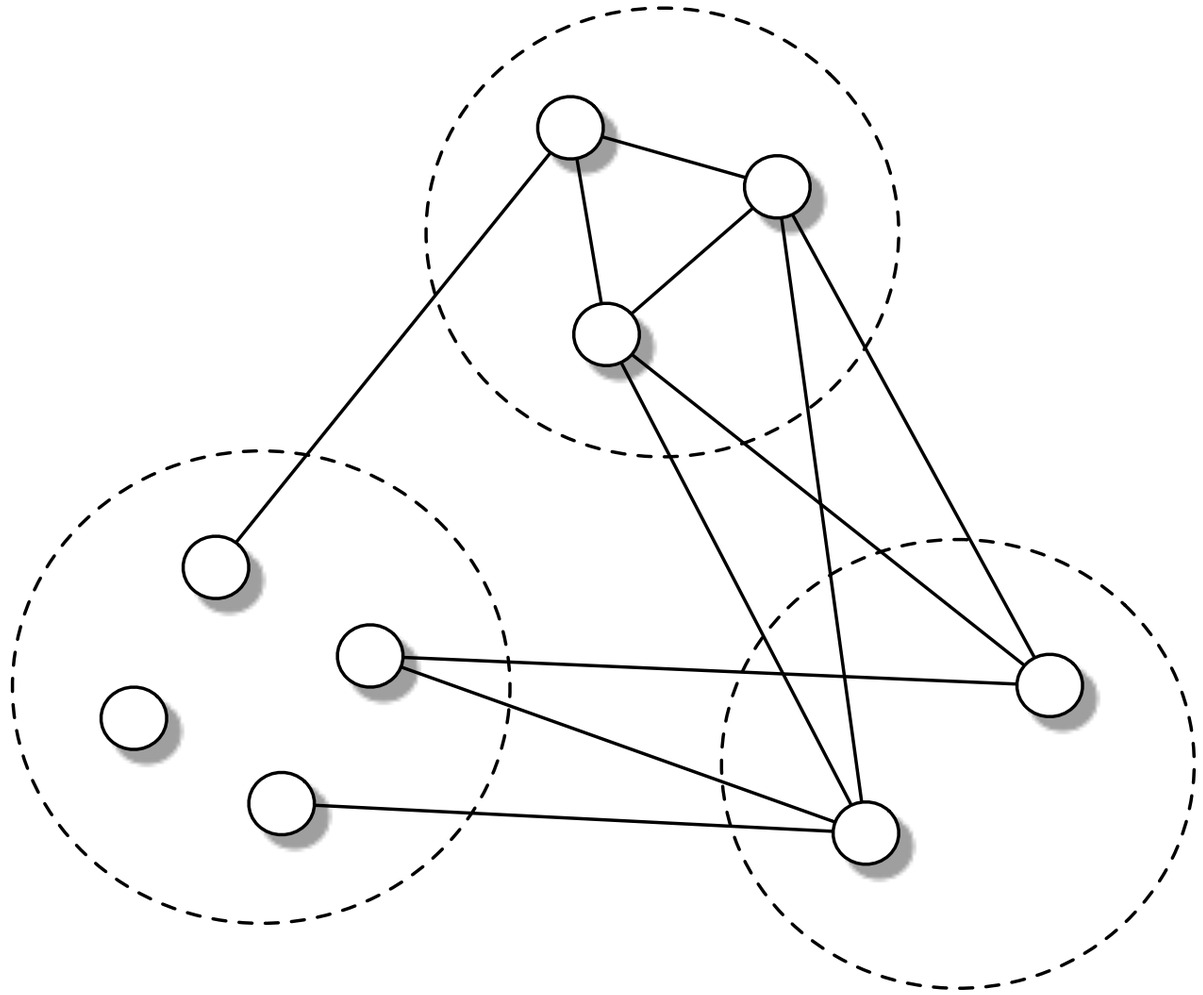
directed: links from i→j but not j→i

given a network, we want to simultaneously...

label the nodes with their types

learn the probability of a link between each pair of types

Assortative and disassortative



functional groups, not just clumps

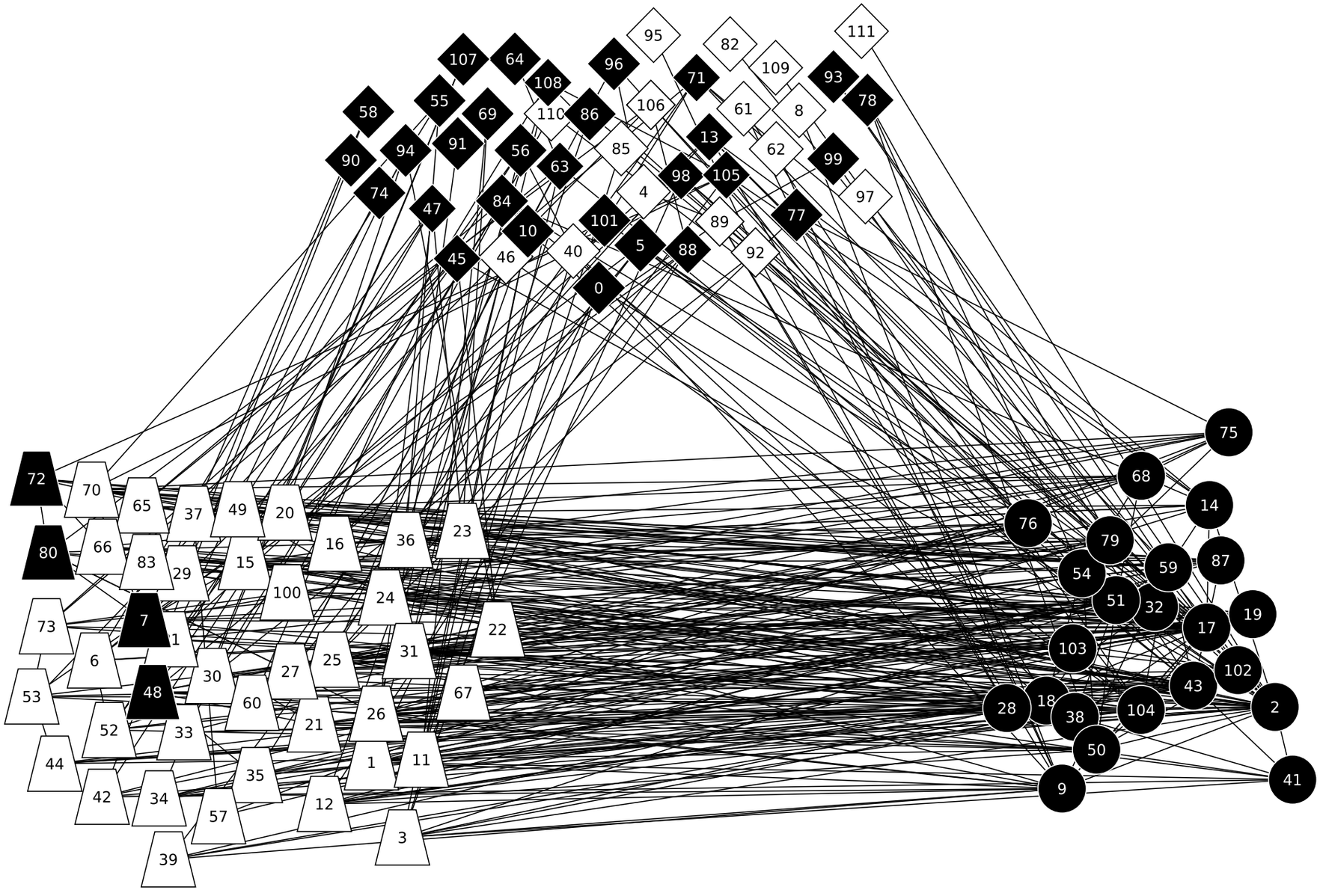
food webs: predators and prey

economics: suppliers and customers

word adjacencies: adjectives and nouns

social: leaders and followers

Classifying words with a ground state:
I record that I was born on a Friday



A little statistical physics

each labeling of the nodes is a “state”, like orientations of atoms in a magnet

2^n possible states

if a state has energy E, then its probability is proportional to

$$P \propto \mathrm{e}^{-E}$$

so the “energy” of a state in the block model is

$$E = -\log P$$

highest probability = lowest energy

Inferring the block model scalably

in the worst case, finding the most-likely labeling of the nodes is an exponentially hard (NP-hard) optimization problem: 2^n possibilities

happily, real networks are not diabolically designed

polynomial-time algorithms: $O(n^2)$, $O(n^3)$

if $n=10^6$, we need linear time, or nearly so: $O(n)$

several scalable methods that work in practice

how do we explore the space of possibilities?

Method #1:
Markov Chain Monte Carlo

---

update the labels one node at a time

choose a random node *v*

fix types of all other nodes

update *v*'s type according to its neighbors and the link probabilities

can speed up by introducing a temperature parameter:

simulated annealing

parallel tempering

but there's no free lunch: can get stuck in local optima

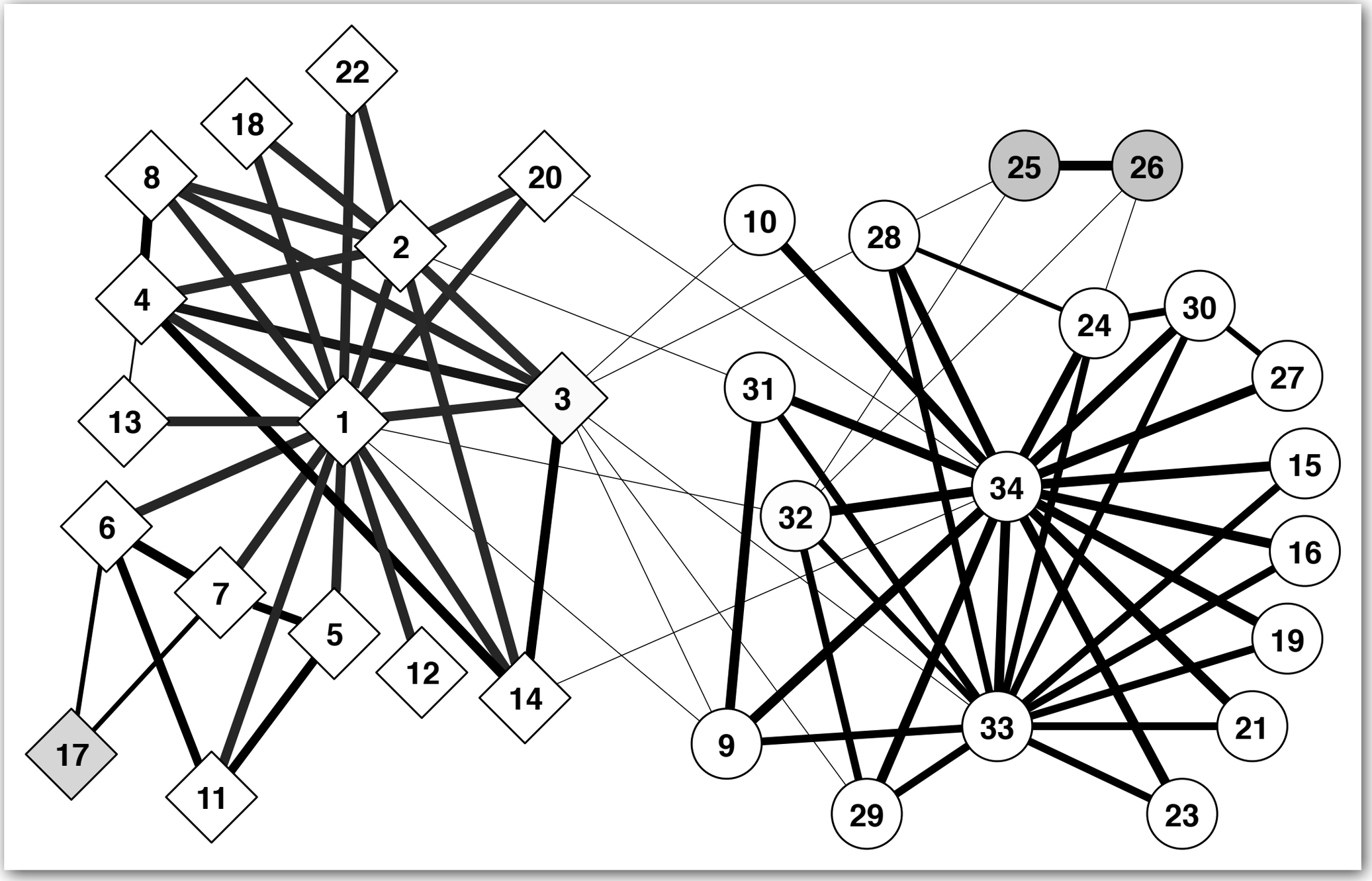
Ground states vs. equilibrium

we don’t just want the most likely labeling!

we want labels with confidence levels, “soft classifications”

how strongly does a node belong to a community?

Ground states vs. equilibrium



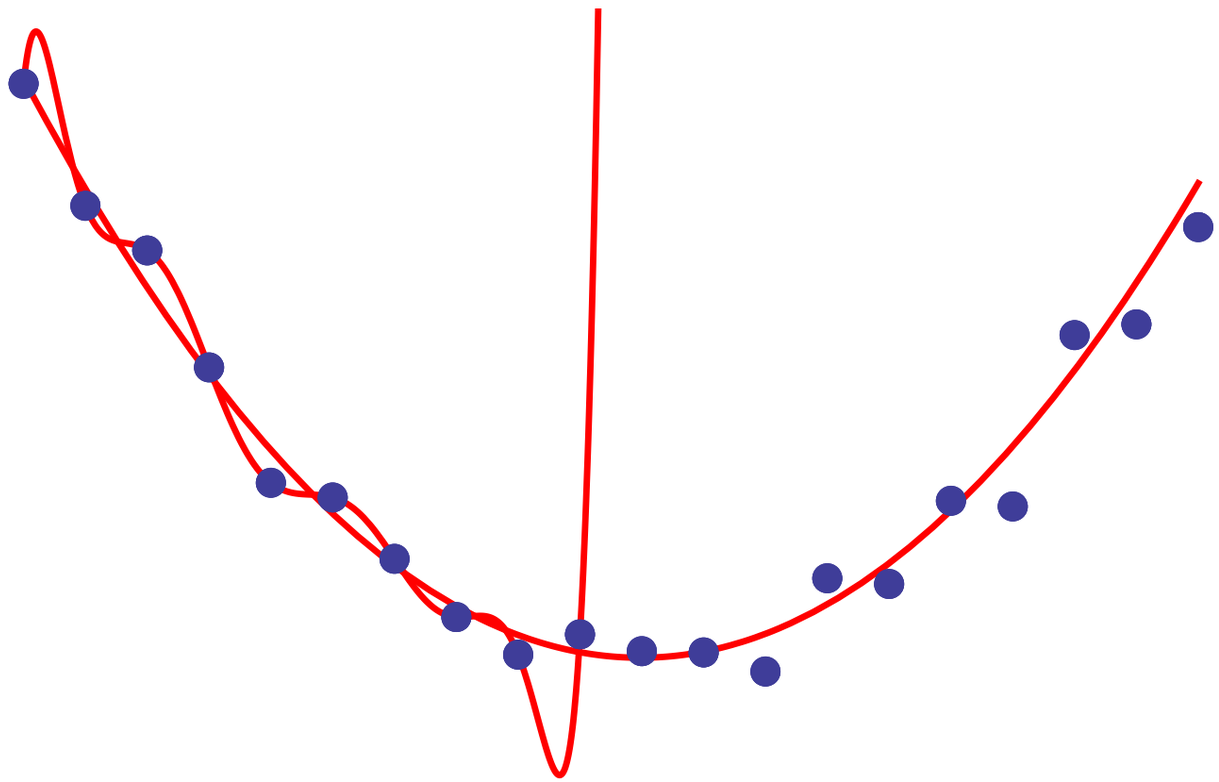
[Clauset, Moore, Newman]

Statistical significance

another reason we don't just want the most likely labeling:

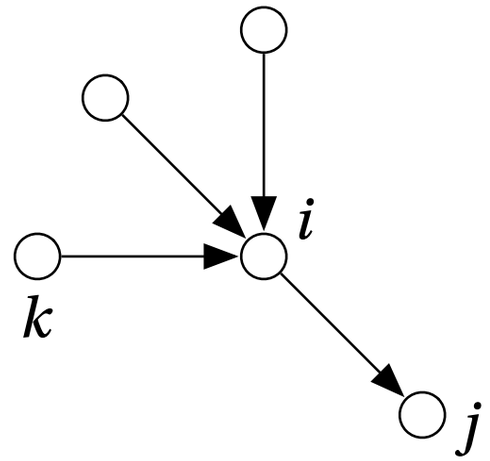
random graphs have illusory communities, that only exist because of noise

sometimes the patterns we find aren't really there:



we want to understand the coin, not the coin flips

Method #2:
Belief propagation (a.k.a. the cavity method)



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

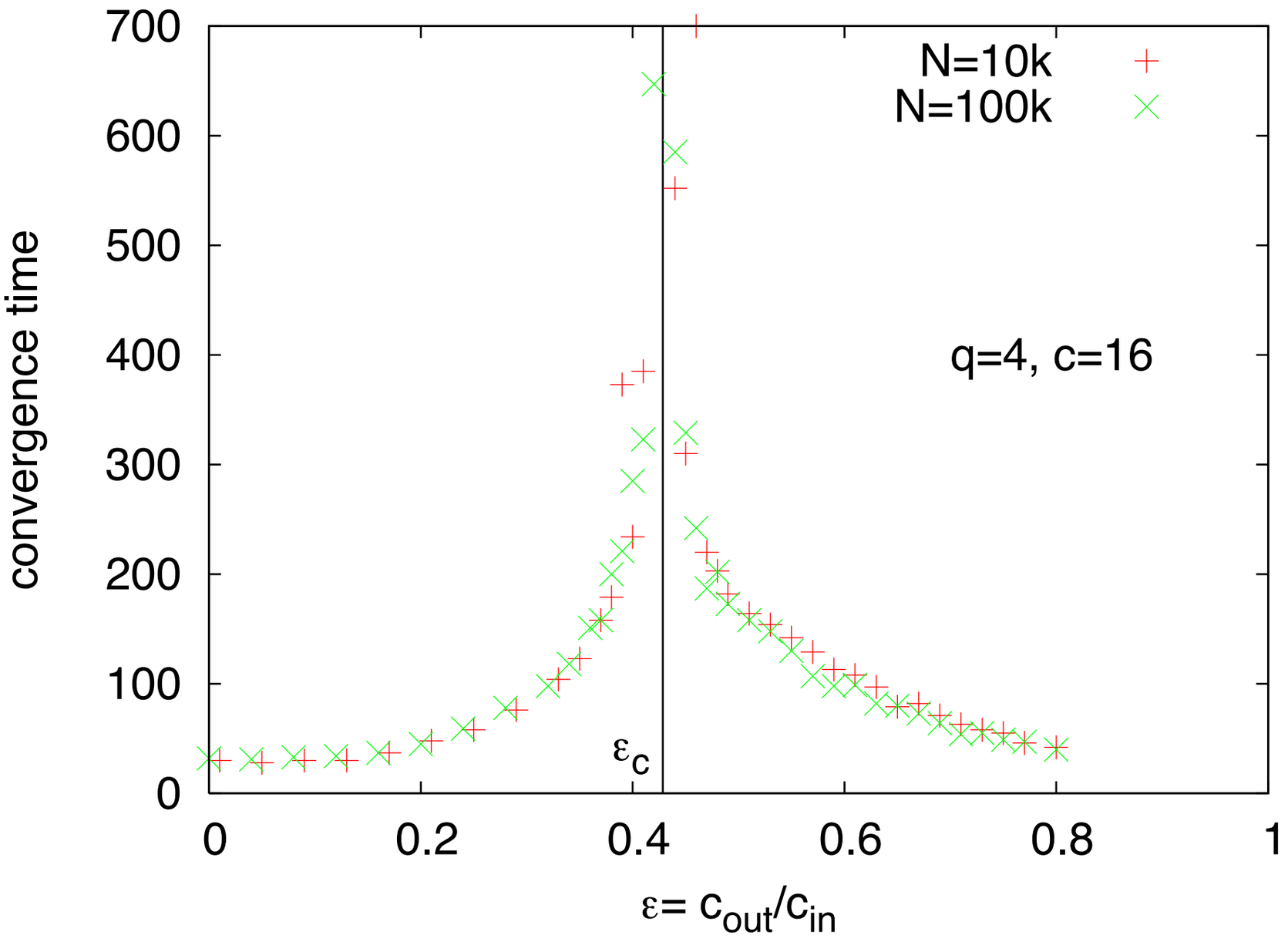
update messages, assuming that *i*’s neighbors are independent of each other...

true for trees, approximately true for real graphs

each update takes O(*n*+*m*) time: iterate until we reach a fixed point

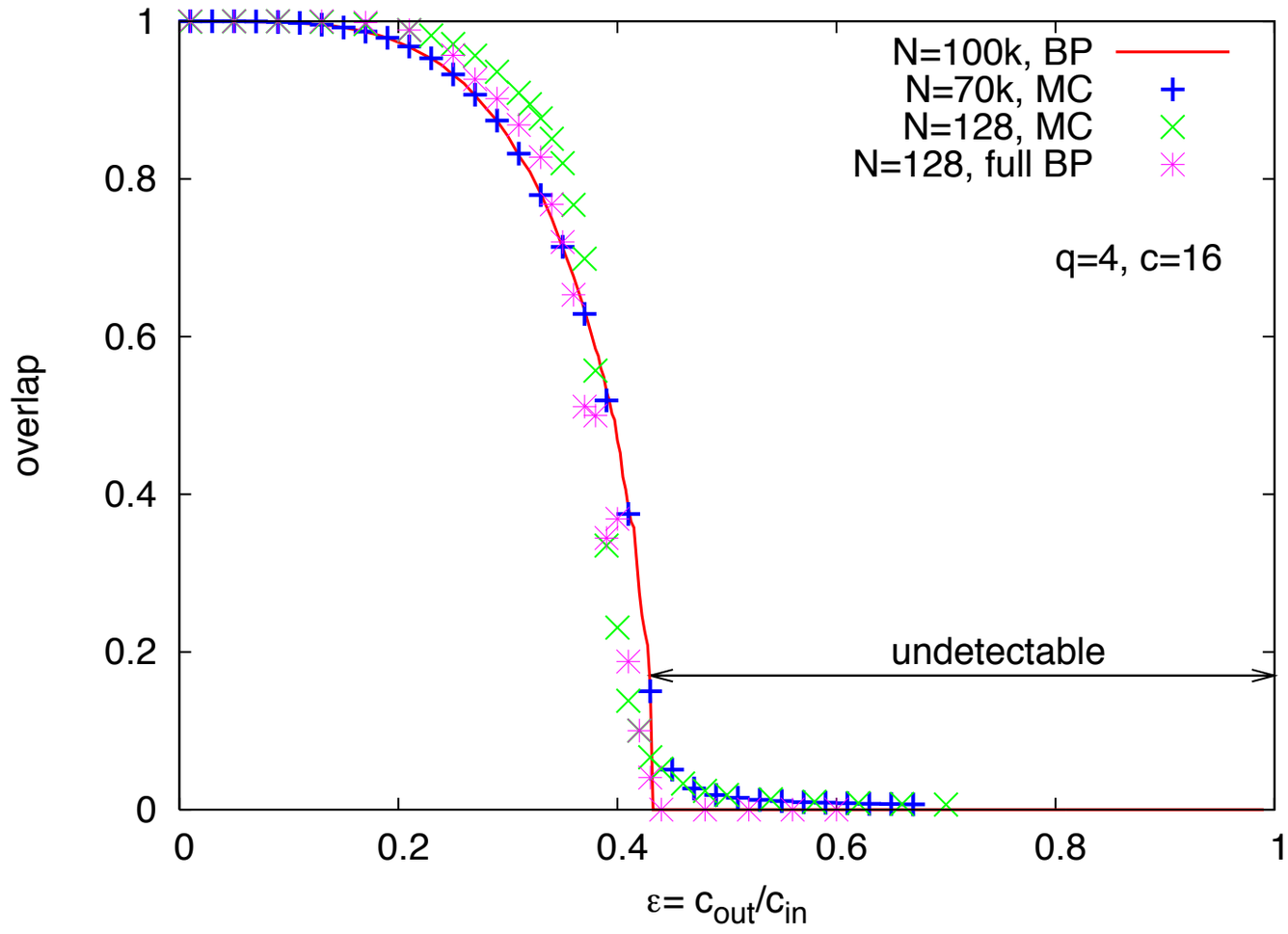
can explore the landscape of possible models

BP convergence: nearly size-independent, but with critical slowing down at a phase transition



[Decelle, Krzakala, Moore, Zdeborová]

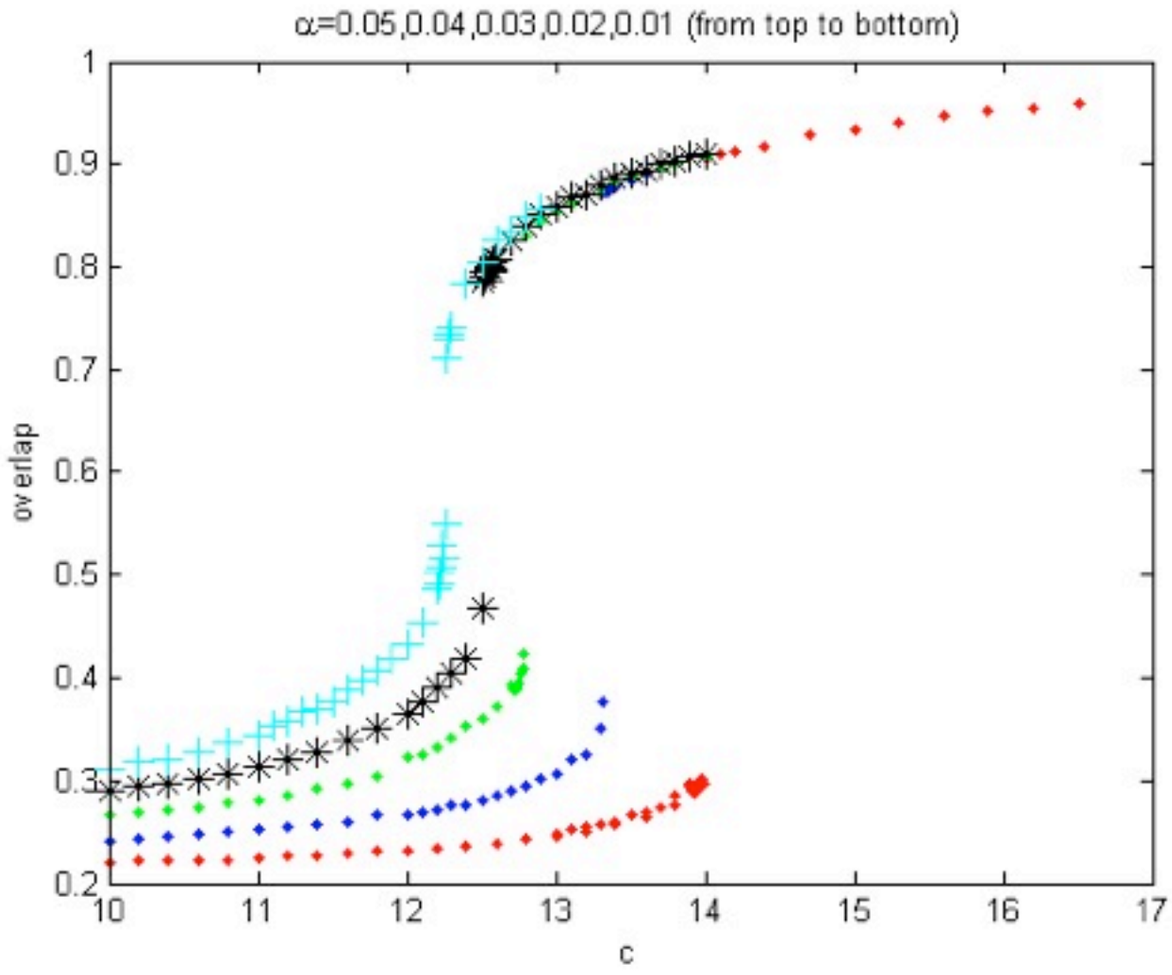
A phase transition: undetectable communities



if the link probabilities are different enough (e.g. more links within groups than between groups) then we can find the communities, efficiently and accurately...

but there is a point beyond which no algorithm can!

Another phase transition:
Generalizing from known nodes to unknown ones

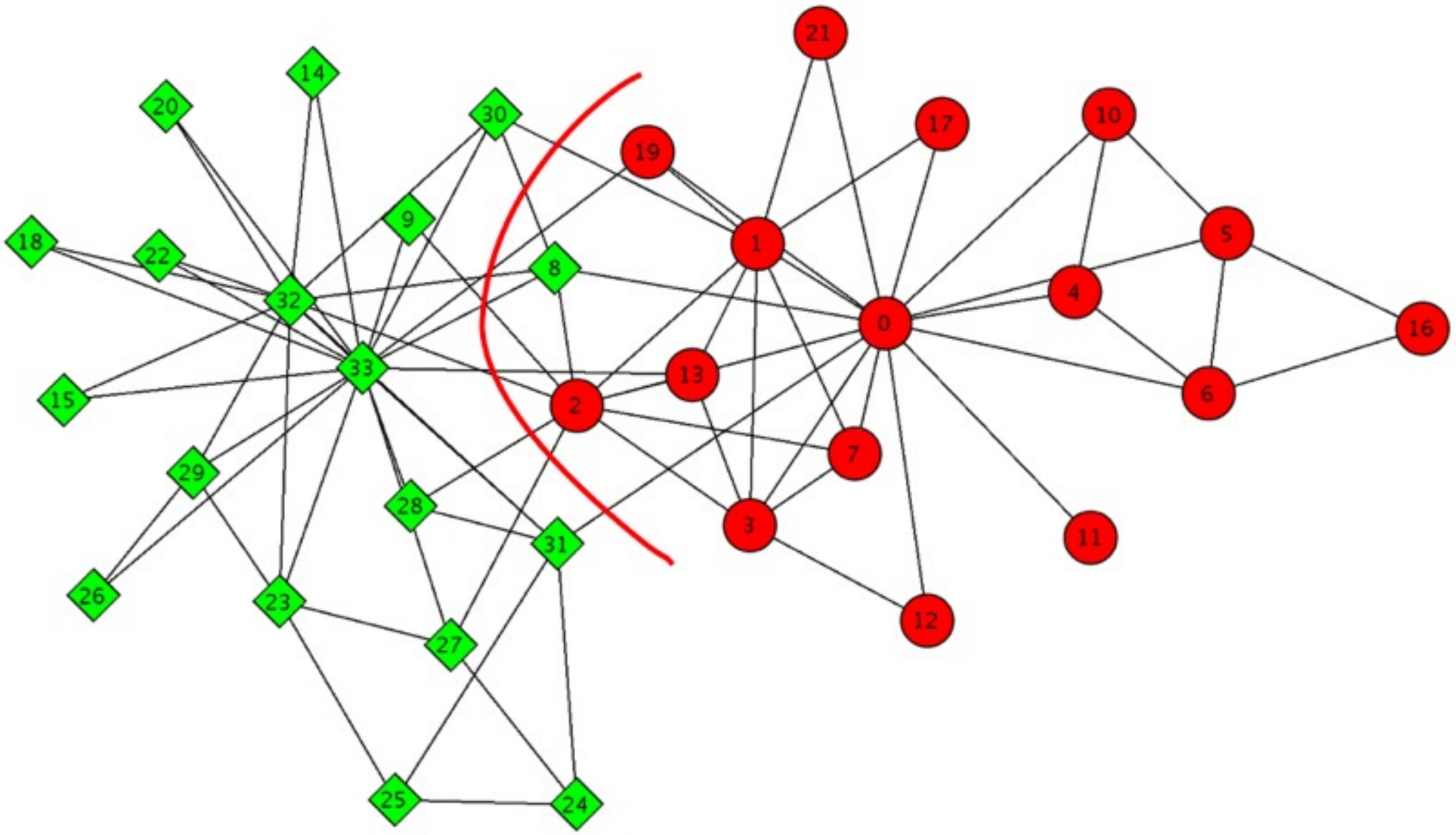


suppose we are initially given the correct types for a fraction α of the nodes.
can we use this information to label the rest of the nodes?

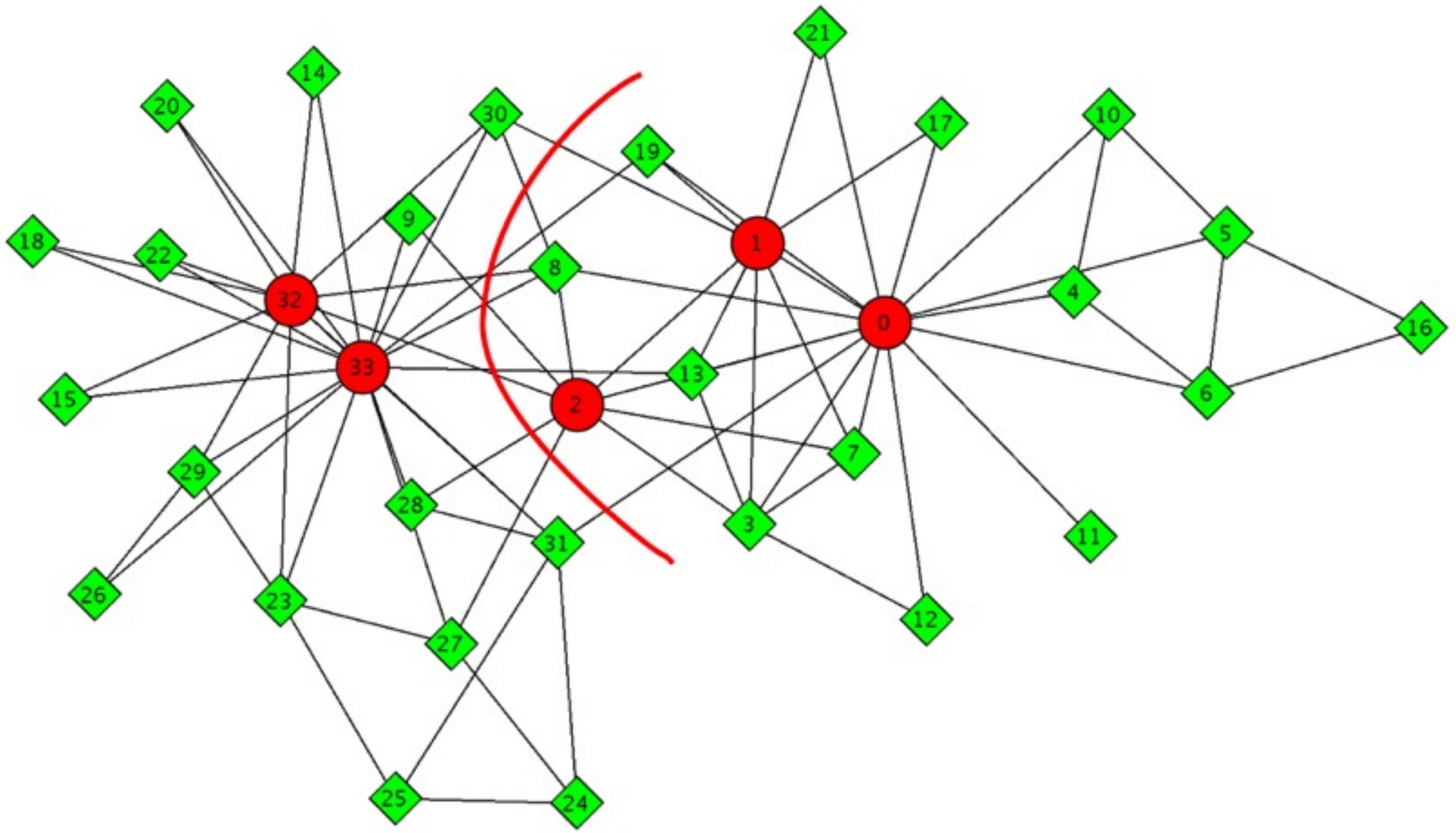
when α crosses a threshold, knowledge percolates throughout the network,
causing a discontinuous jump in the accuracy

[Moore, Zhang, Zdeborová]

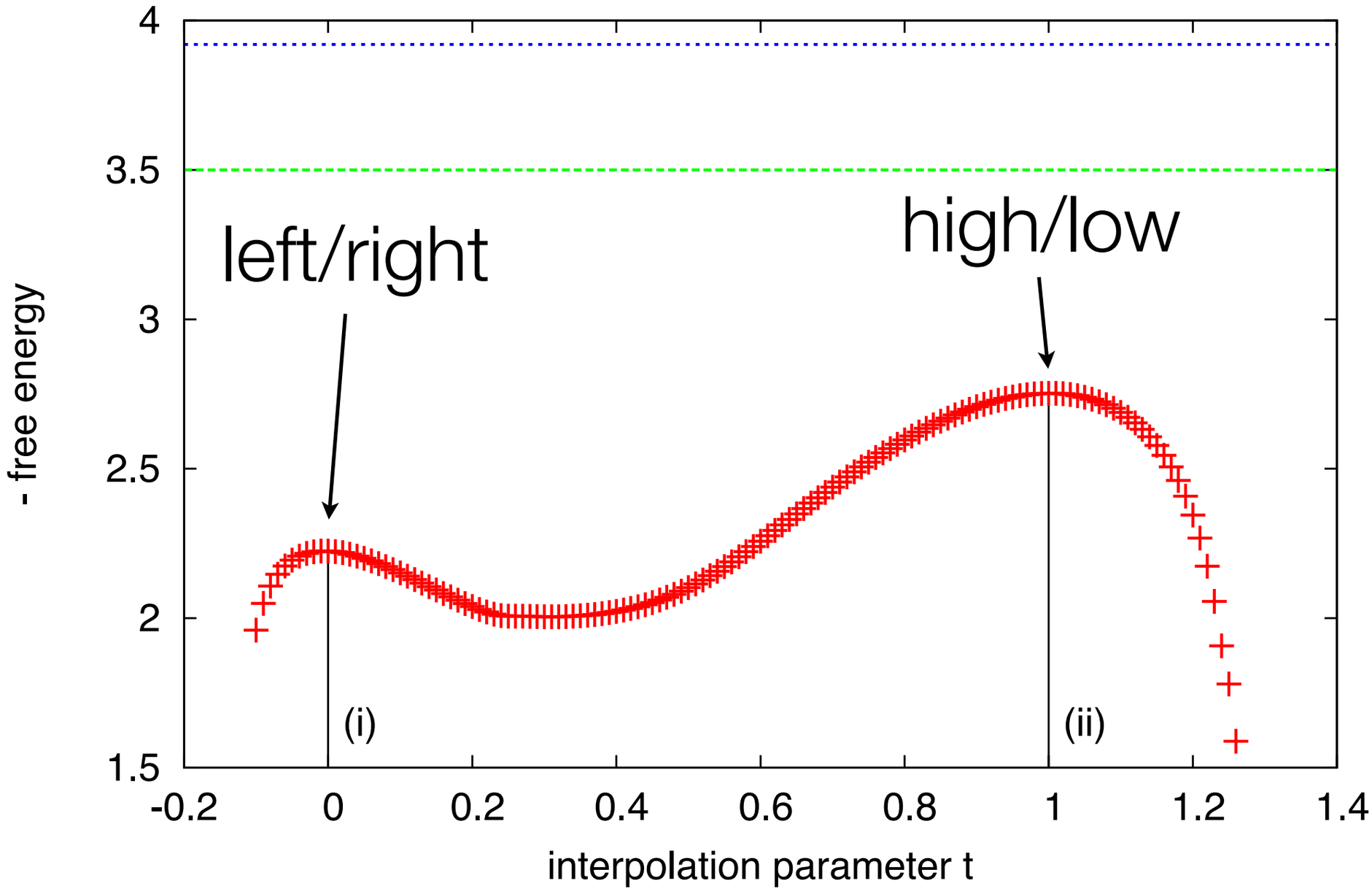
The Karate Club: two factions



The Karate Club: leaders vs. followers



Two local optima in free energy



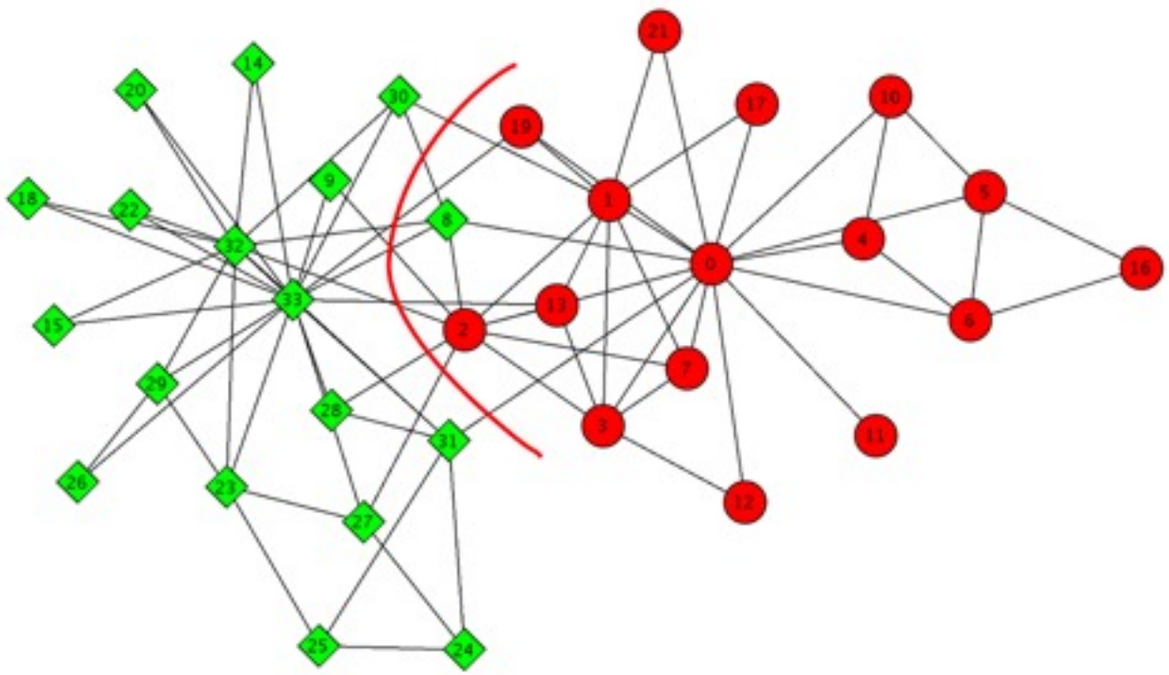
[Decelle, Krzakala, Moore, Zdeborová]

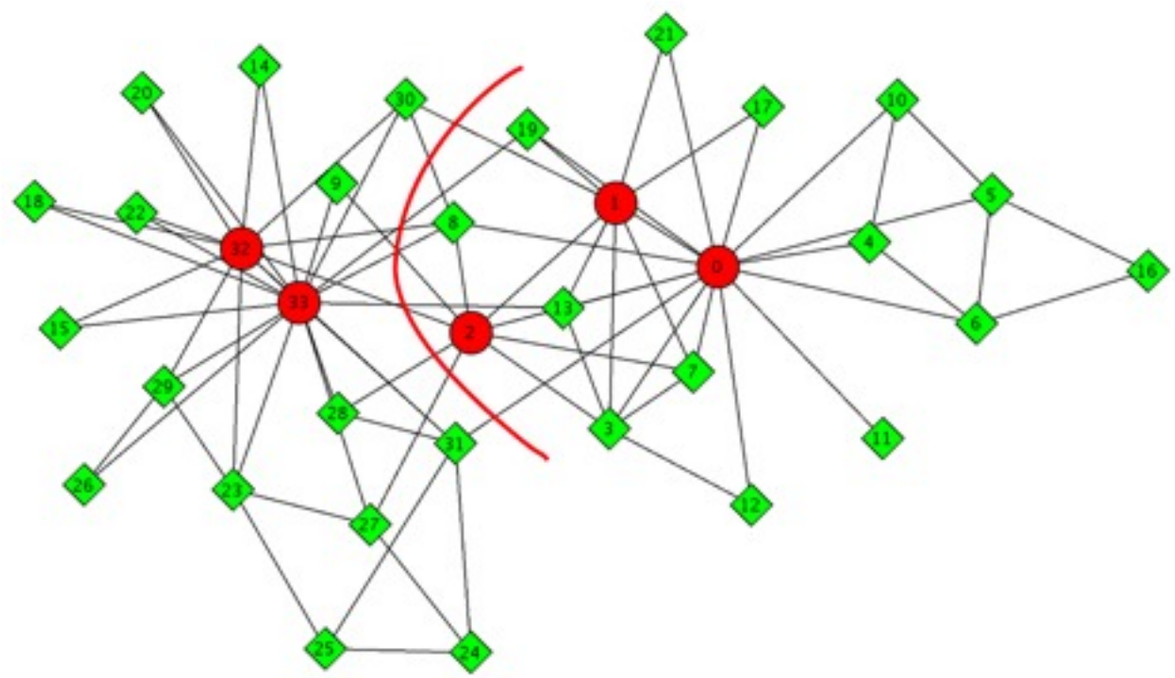
What kind of community do you want?

different models give different answers for the communities

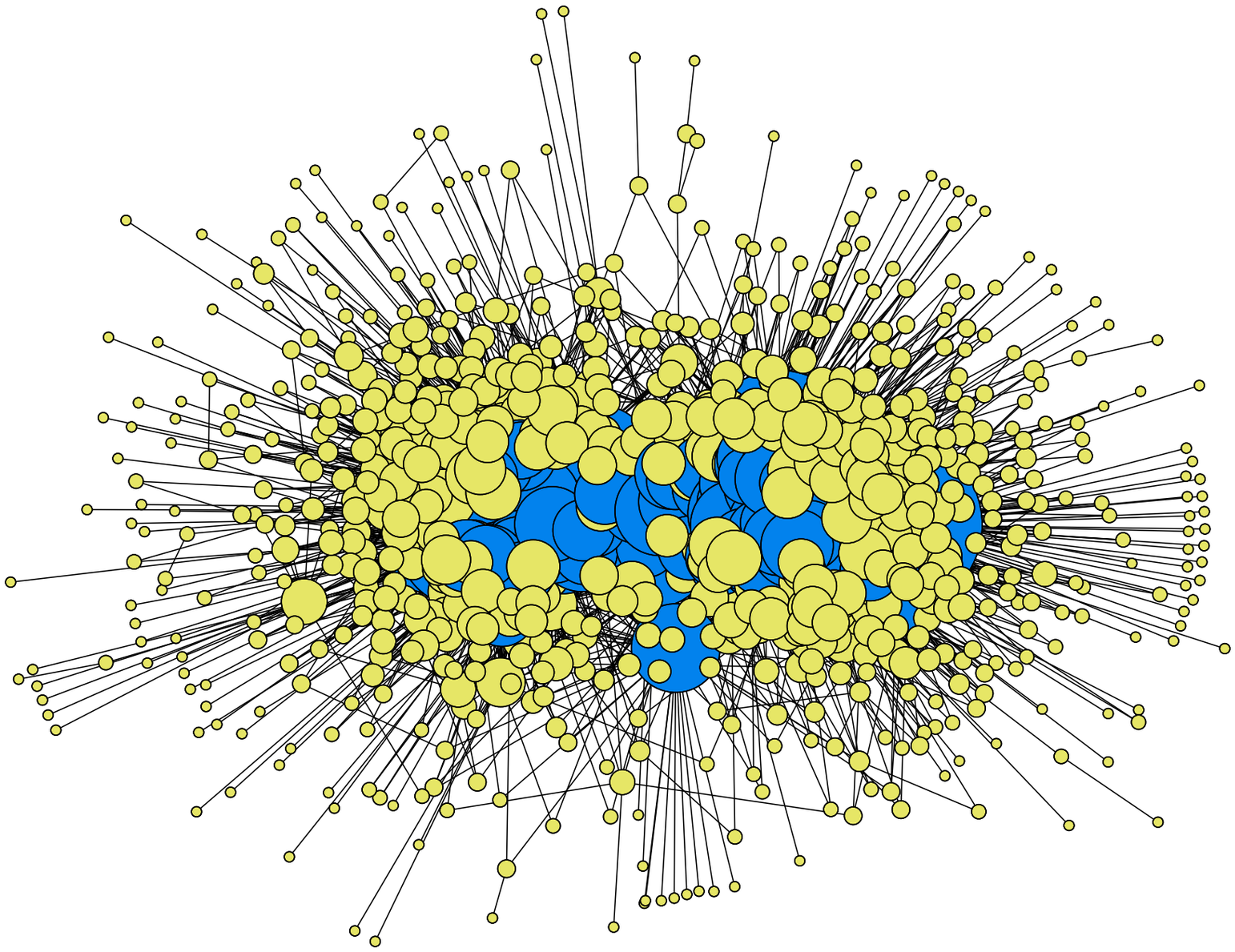
we can compare each one to “ground truth” and judge its accuracy...

...or embrace the fact that they are sensitive to different kinds of structure



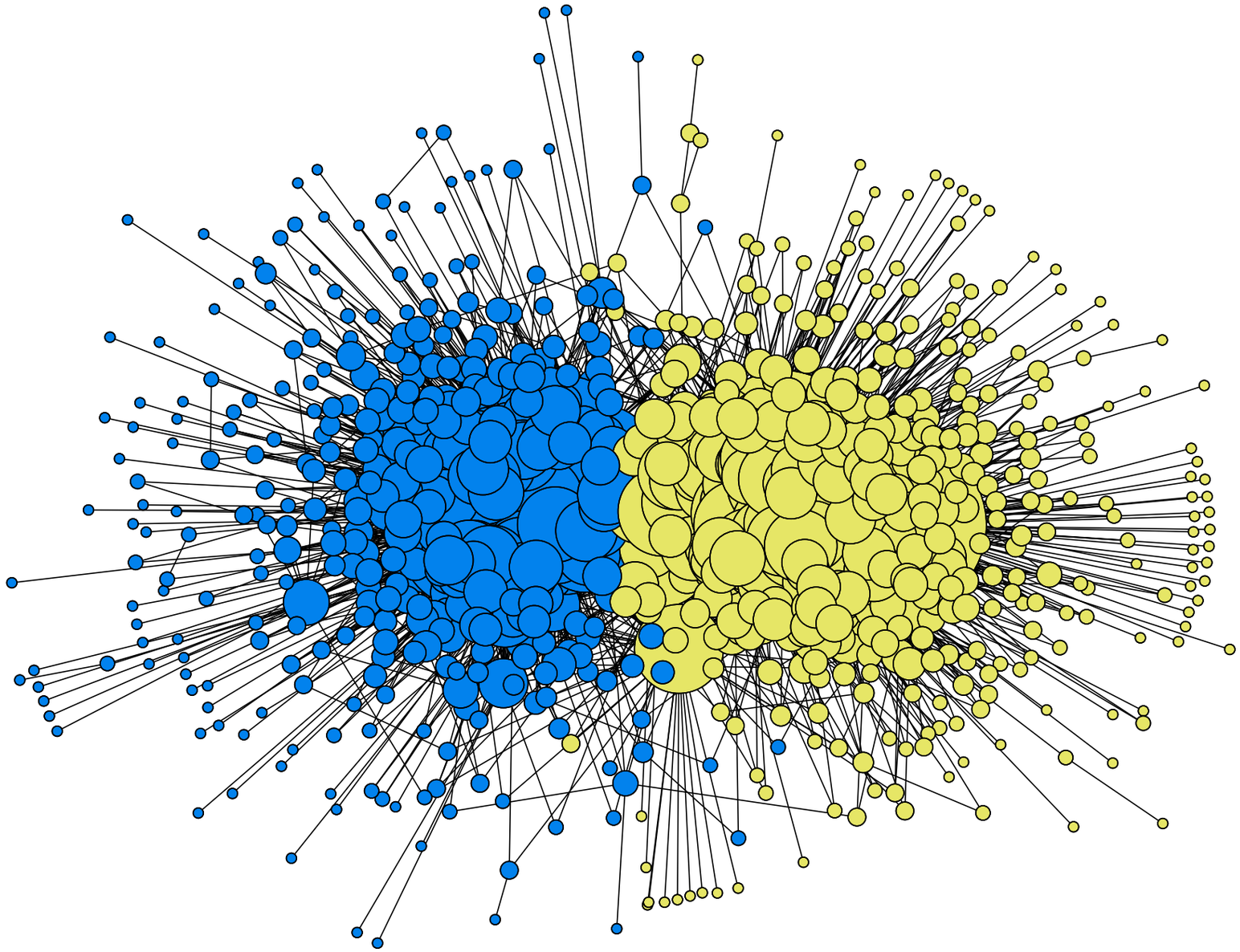


Blogs: vanilla block model



[Karrer & Newman]

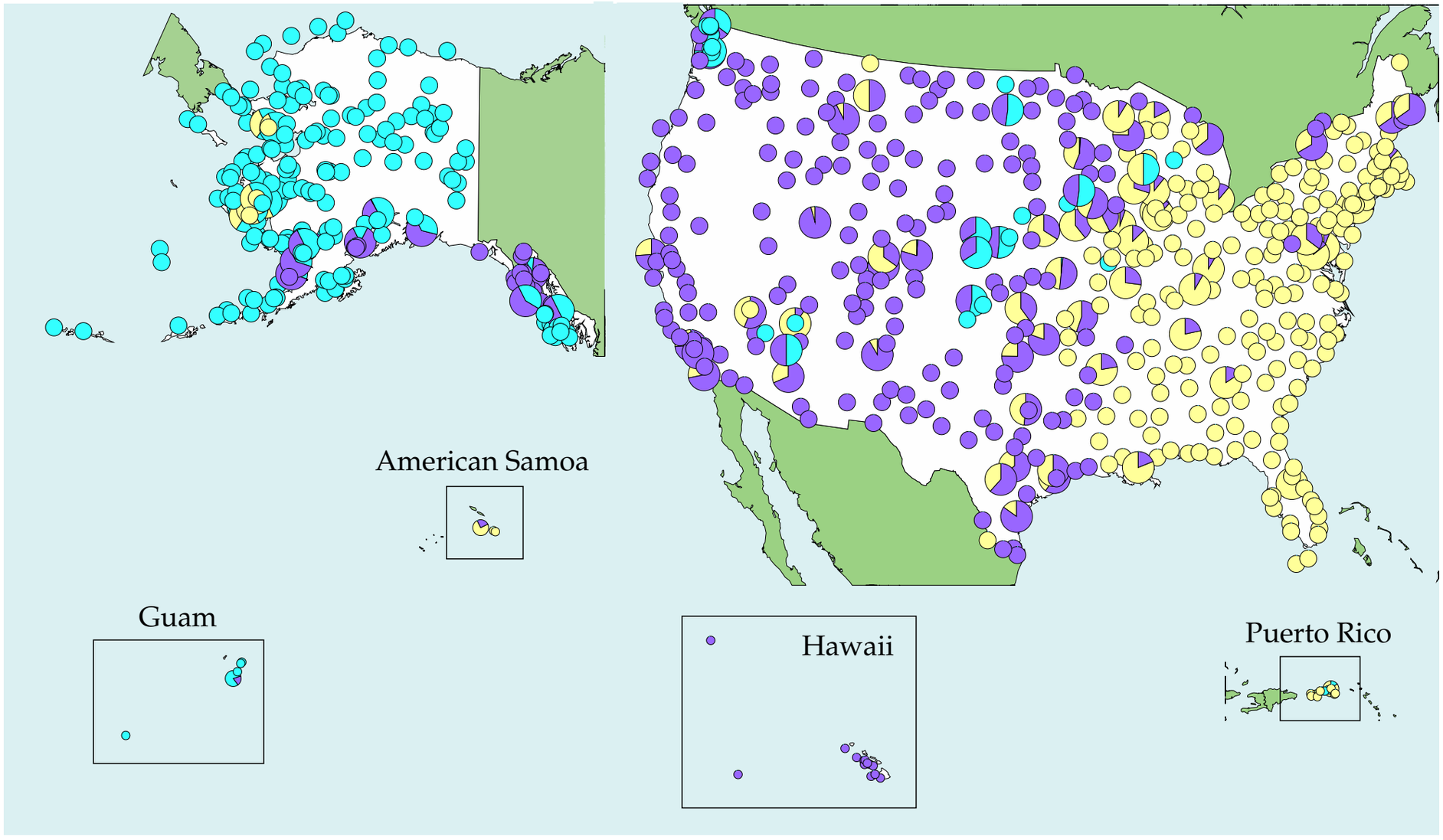
Blogs: degree-corrected block model



[Karrer & Newman]

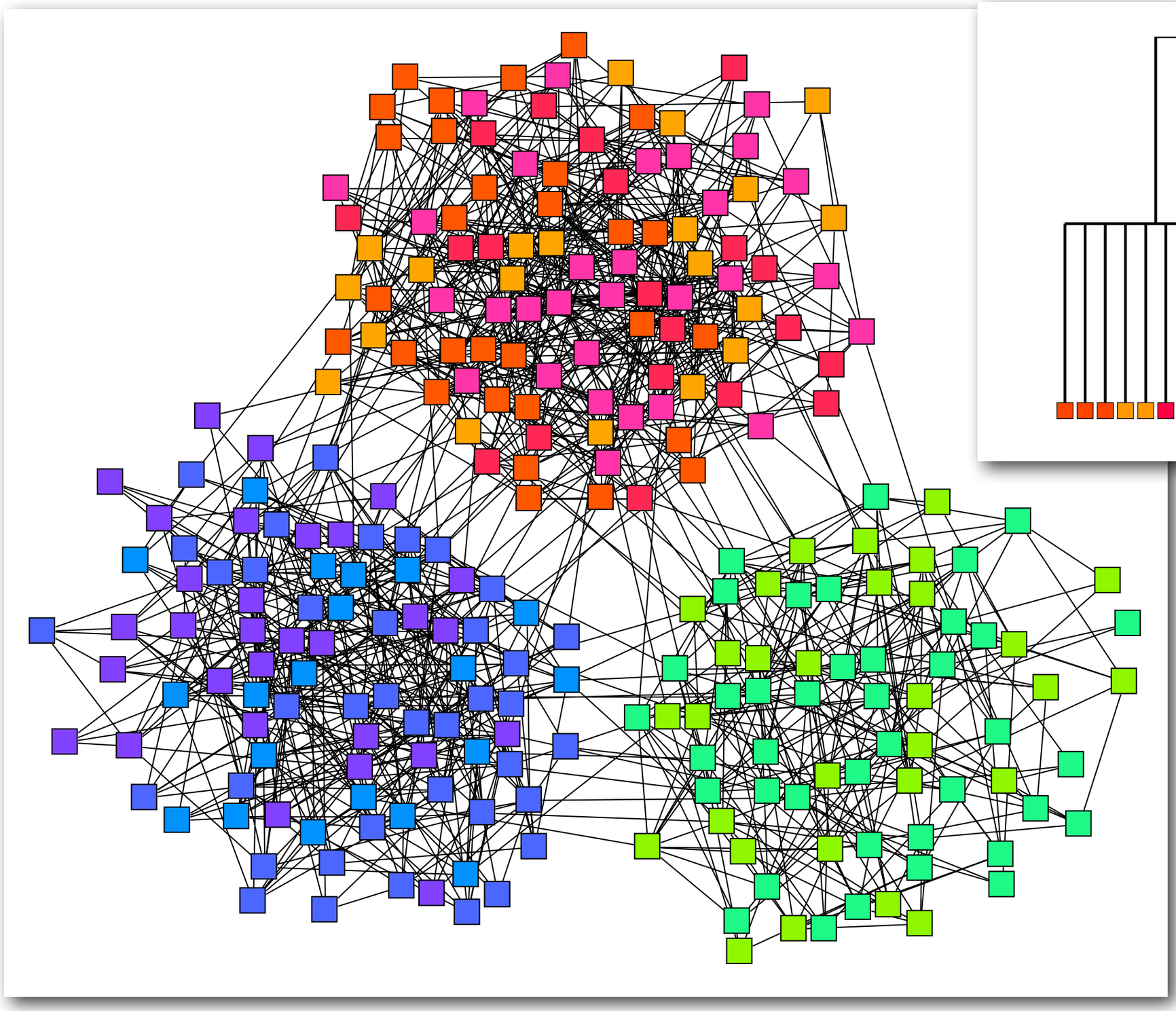
Overlapping communities

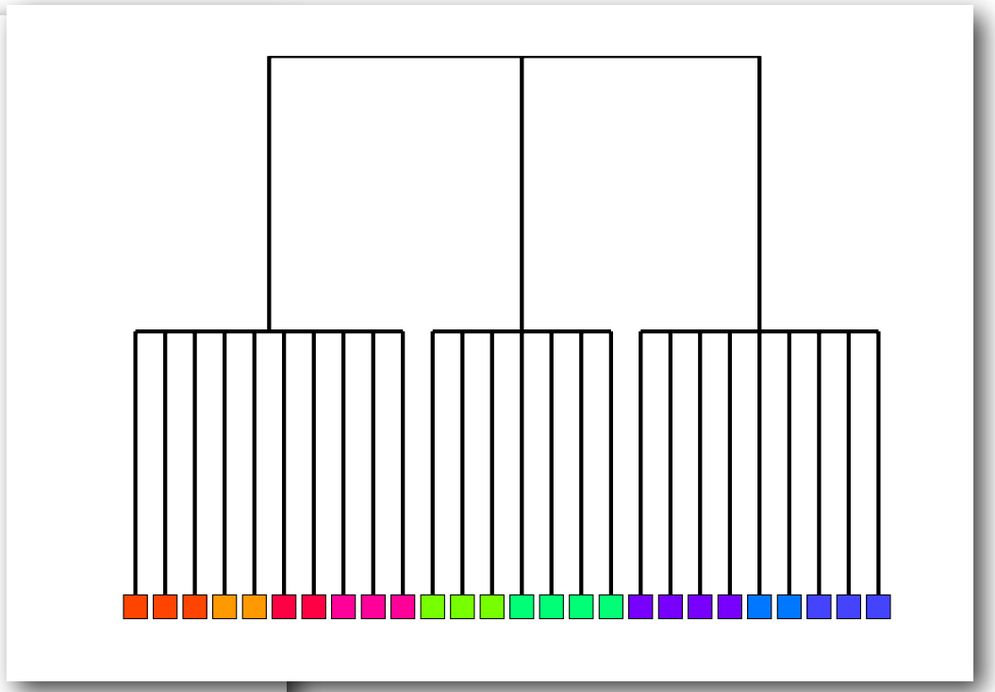
mixed-membership block model: each node has a mix of types, and can act like different types on different edges



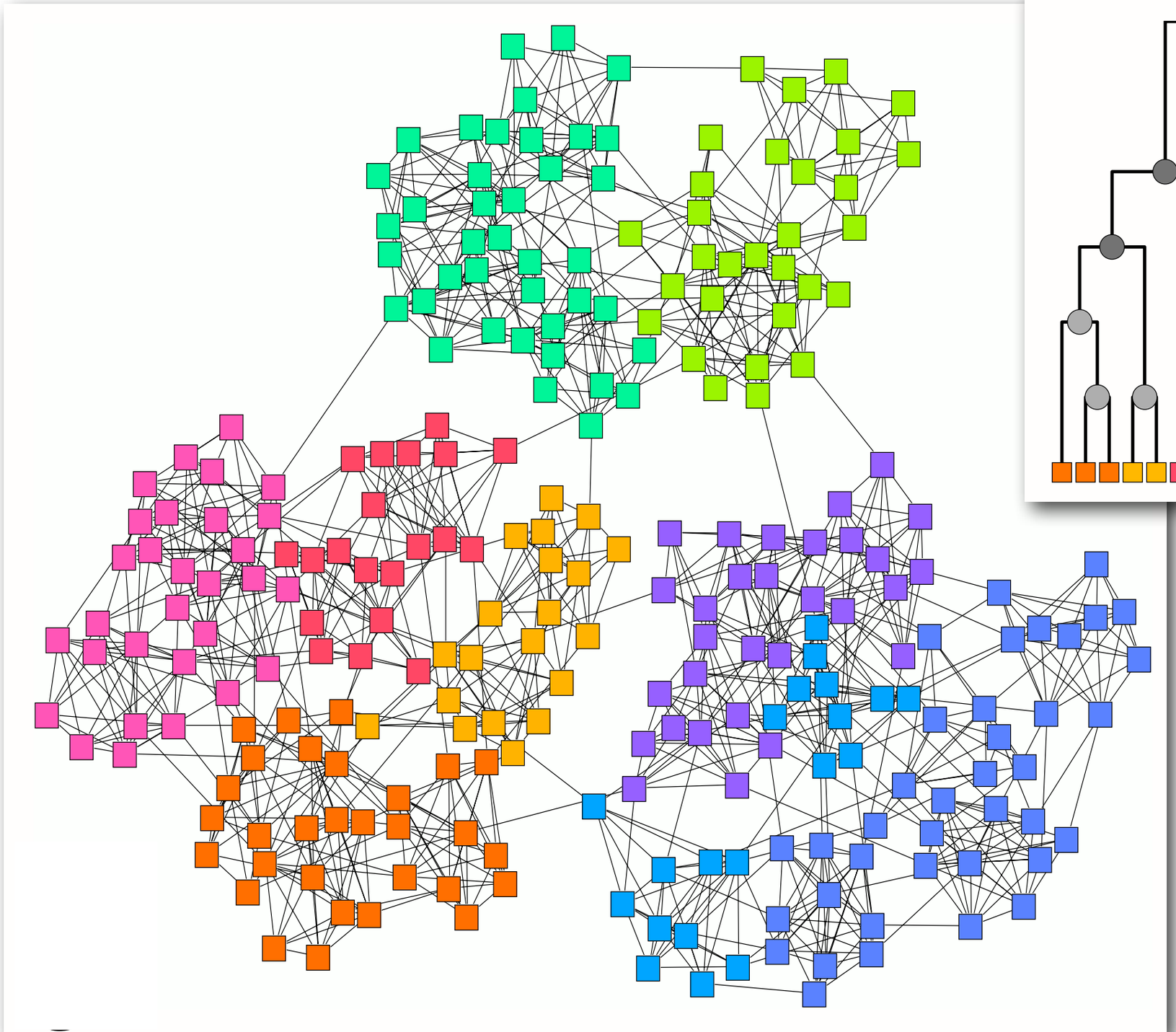
[Ball, Karrer, Newman]

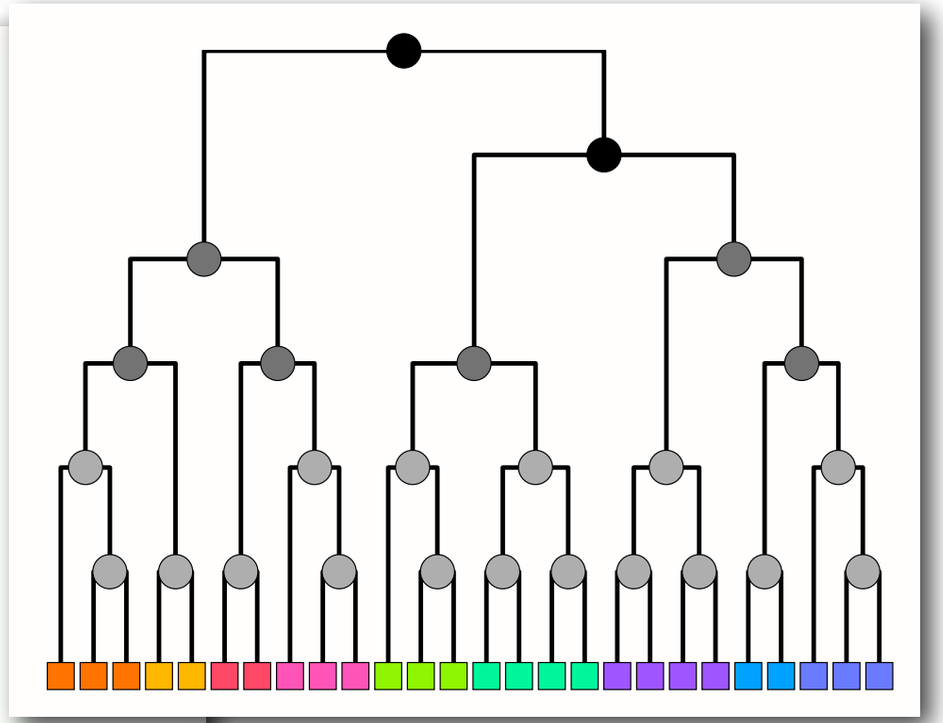
Hierarchy



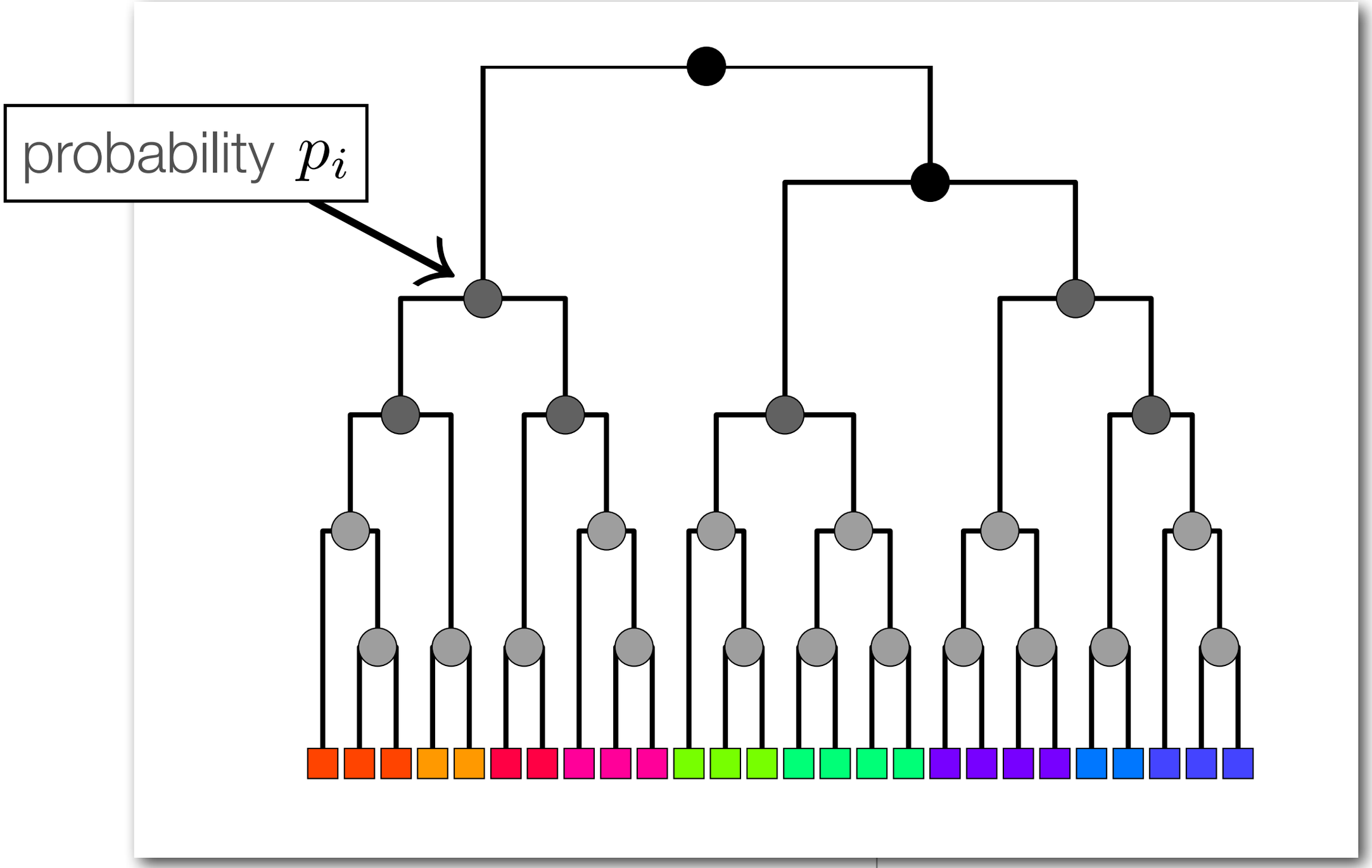


Hierarchy

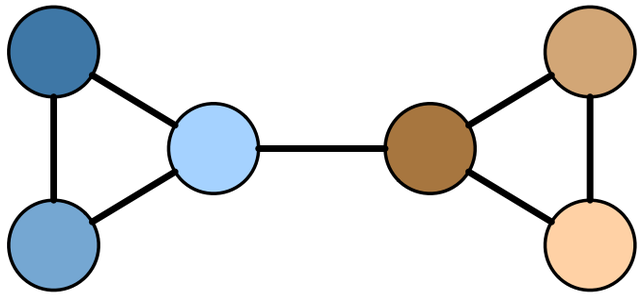


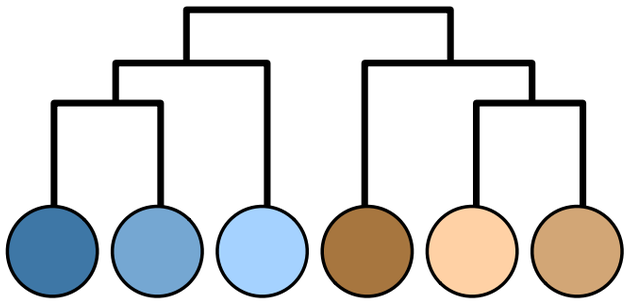


A probabilistic model



Maximum likelihood trees





$$\mathcal{L} = \left(\frac{1}{9}\right)\left(\frac{8}{9}\right)^{8}$$

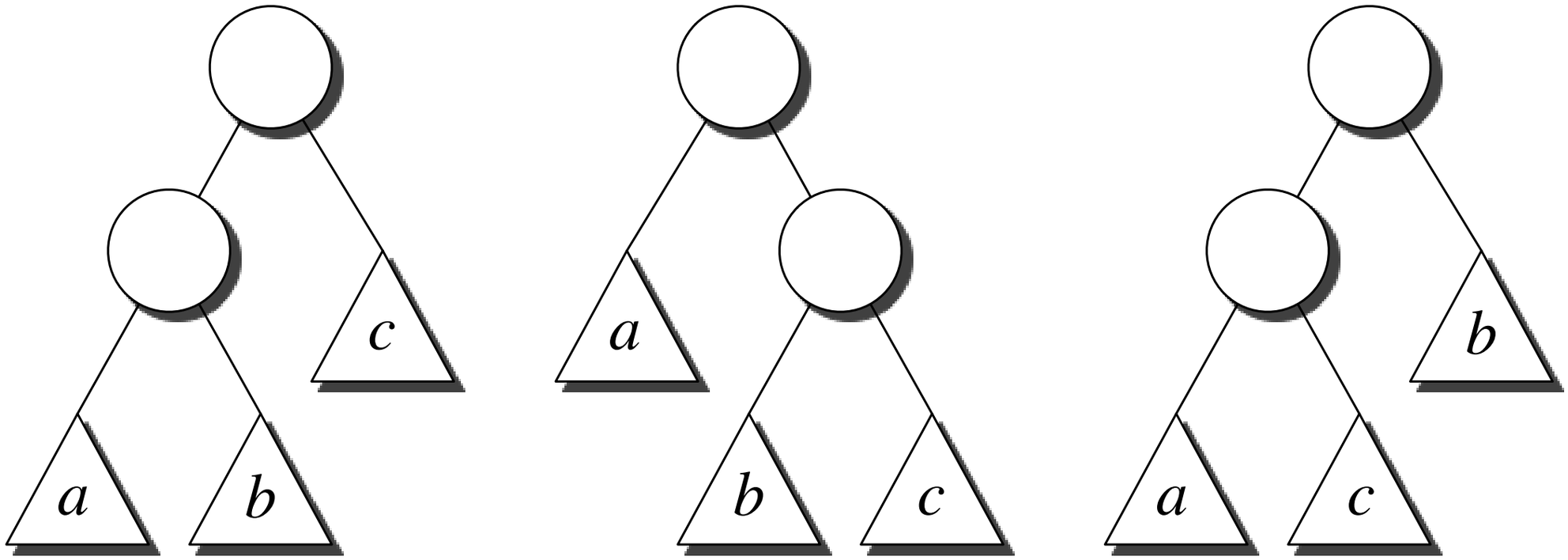
$$= 0.0433$$

$$\mathcal{L} = \left[\left(\frac{1}{3}\right)\left(\frac{2}{3}\right)^{2}\right] \cdot \left[\left(\frac{2}{8}\right)^{2}\left(\frac{6}{8}\right)^{6}\right]$$

$$= 0.0016$$

A Markov chain that explores the space of trees

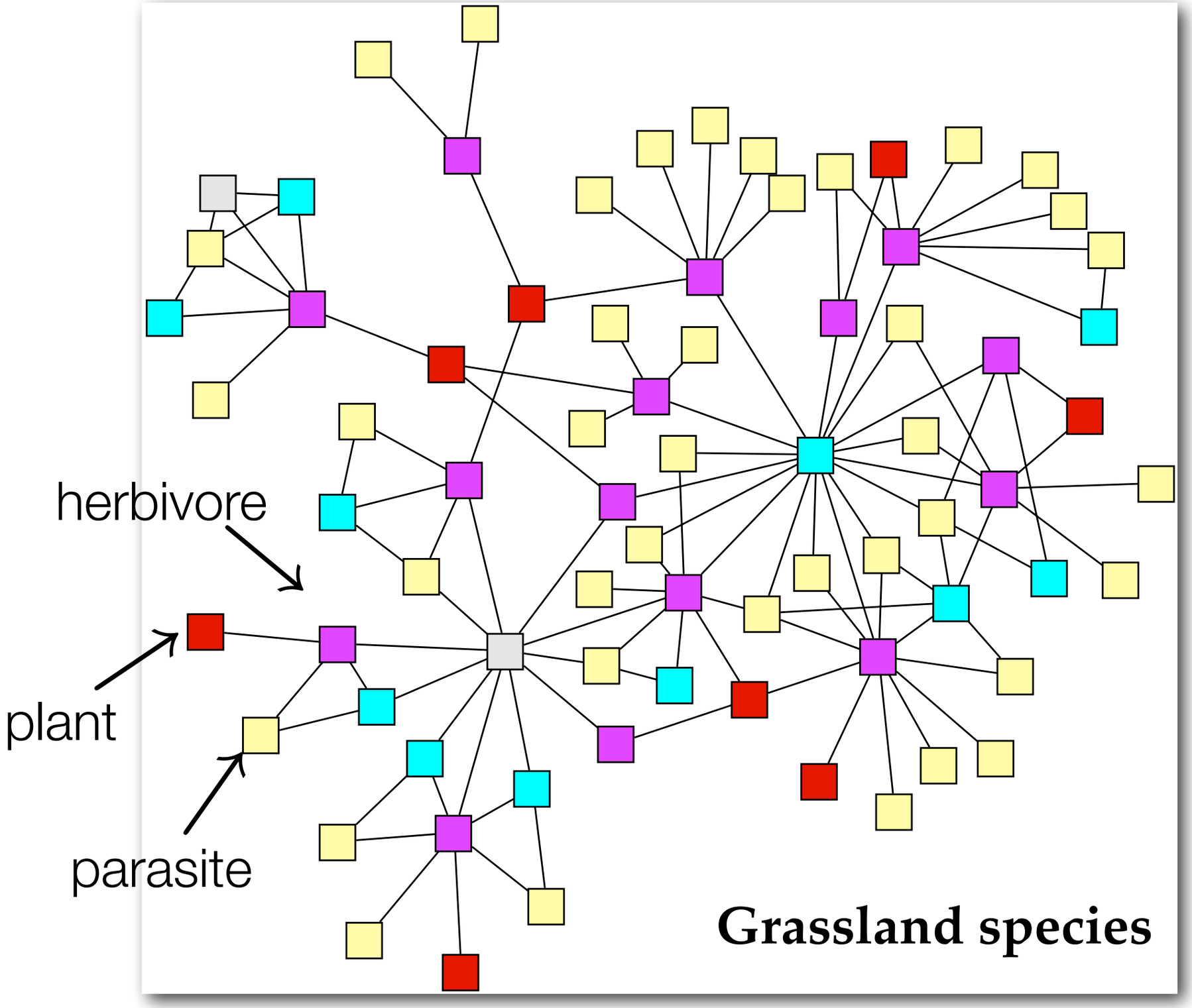
update the tree, changing the hierarchy of relationships



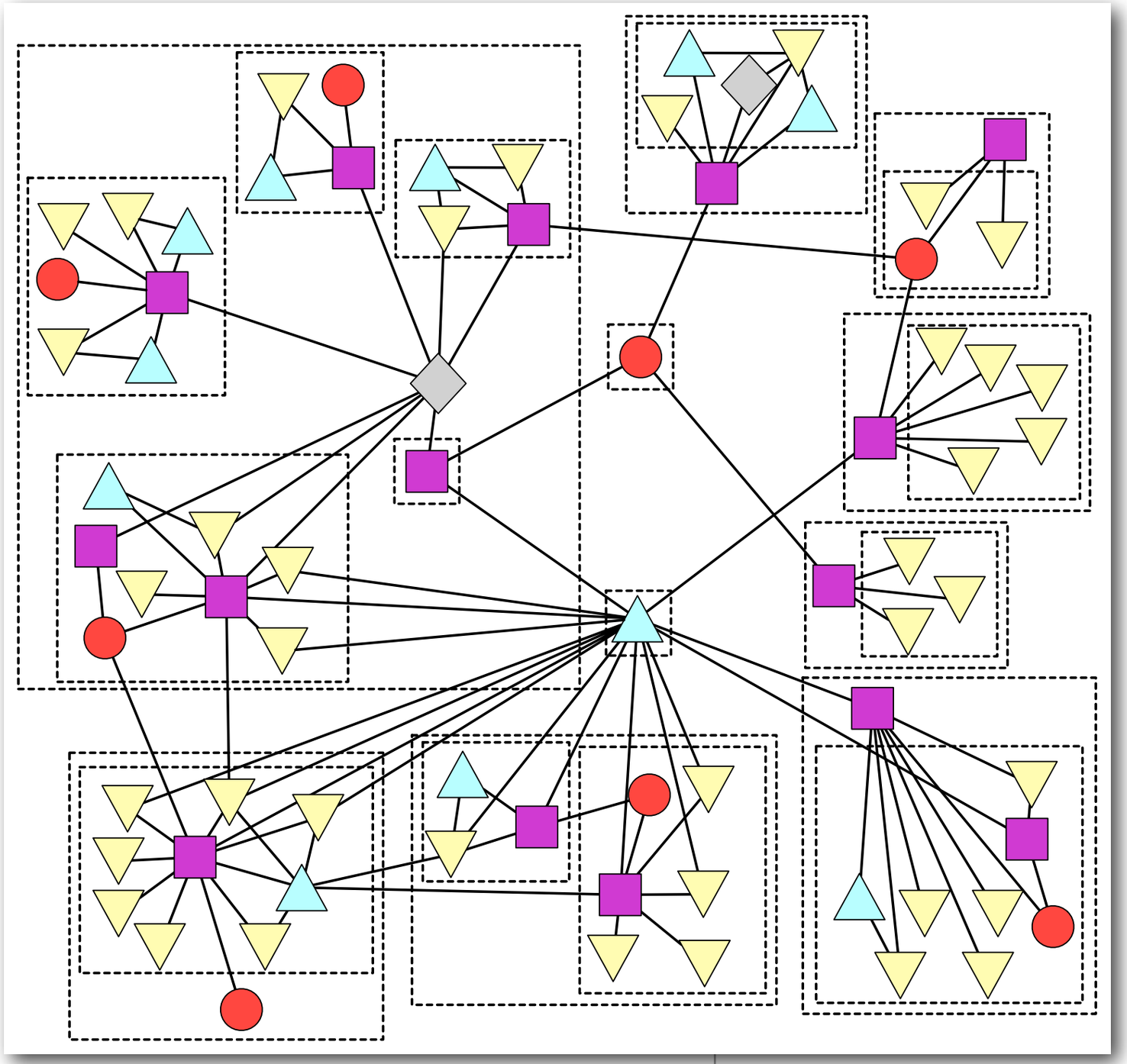
Move if it increases probability; otherwise move with probability p_{new}/p_{old}

[Clauset, Moore, Newman]

Functional roles in a food web



Functional roles in a food web



Dealing with uncertainty #1: Predicting missing links

for many networks, links are discovered one at a time, using difficult work and limited resources in the field or laboratory

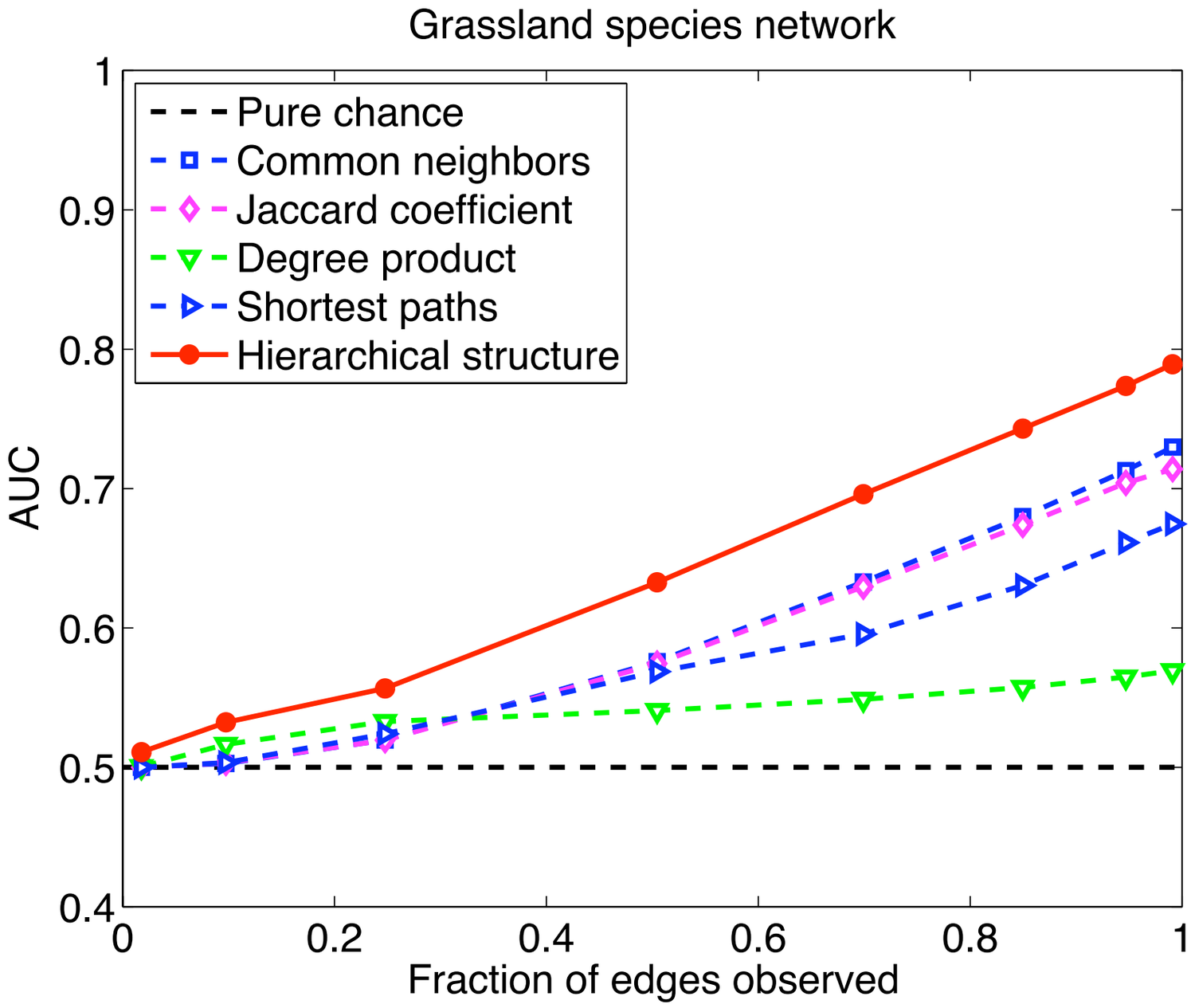
given the links observed so far, can we predict missing links?

if there are spurious edges (false positives), can we identify them?

test the algorithm by hiding a random subset of edges from it, and ask it to rank possible missing links according to their probability

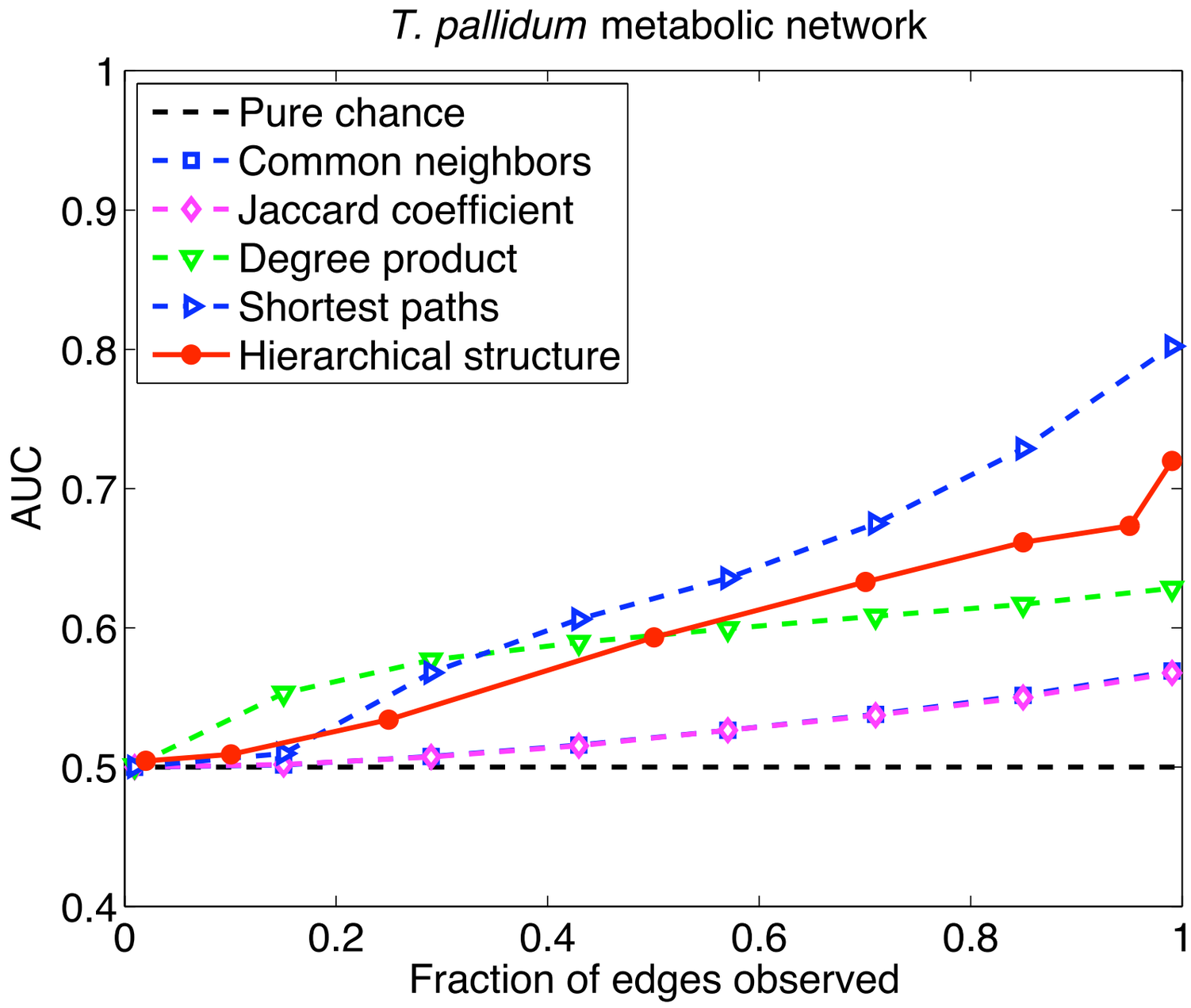
Predicting missing links: comparison with simple heuristics

AUC: probably a random true positive is ranked above a random true negative



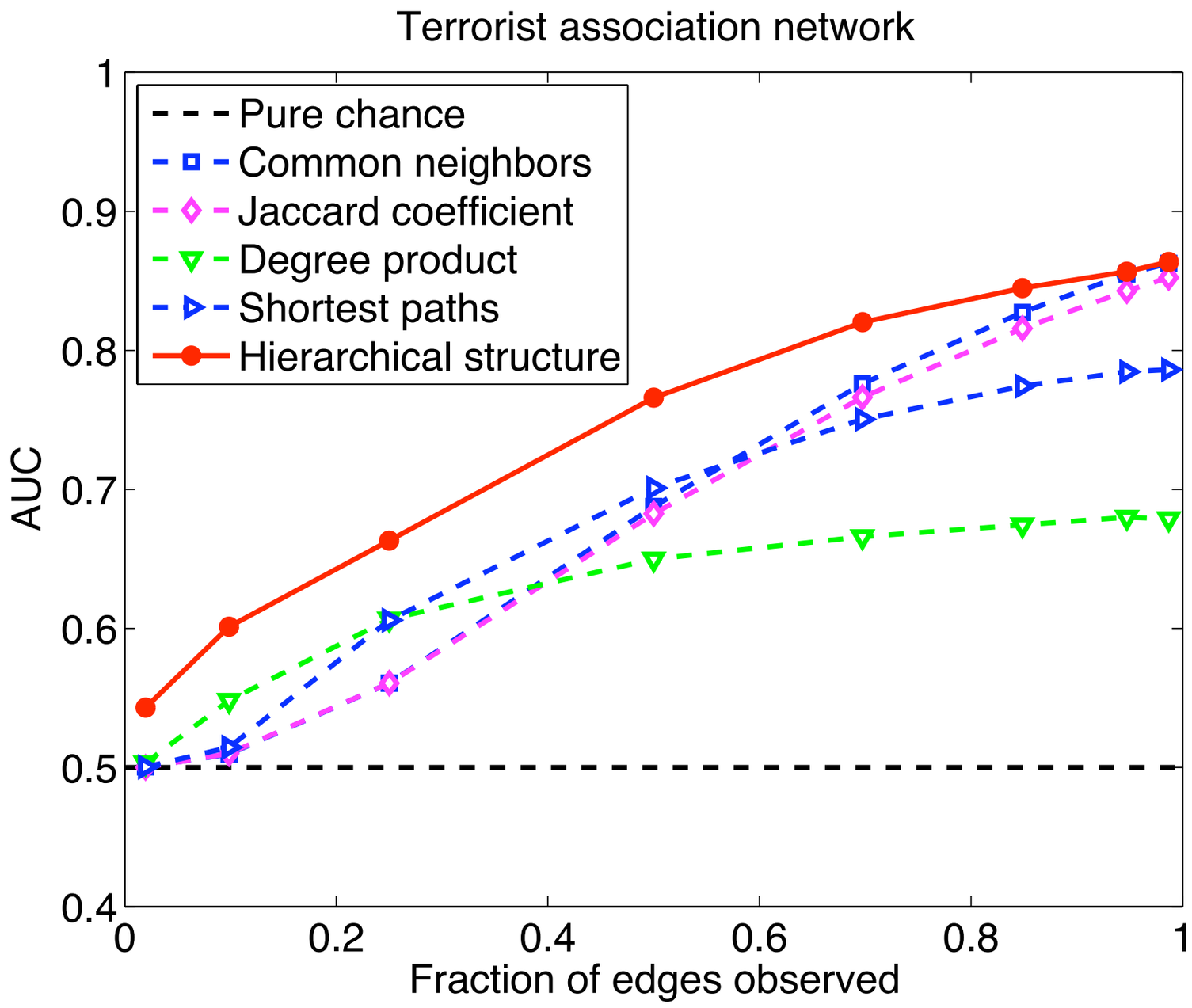
Predicting missing links: comparison with simple heuristics

AUC: probably a random true positive is ranked above a random true negative



Predicting missing links: comparison with simple heuristics

AUC: probably a random true positive is ranked above a random true negative



Dealing with uncertainty #2: Active exploration of networks

suppose we can learn a node's attributes, but at a cost: interviews, surveys, incentives, warrants

we want to make good guesses about most of the nodes, after querying just a few of them — but which which ones?

query the node with the largest *mutual information* between it and the others:

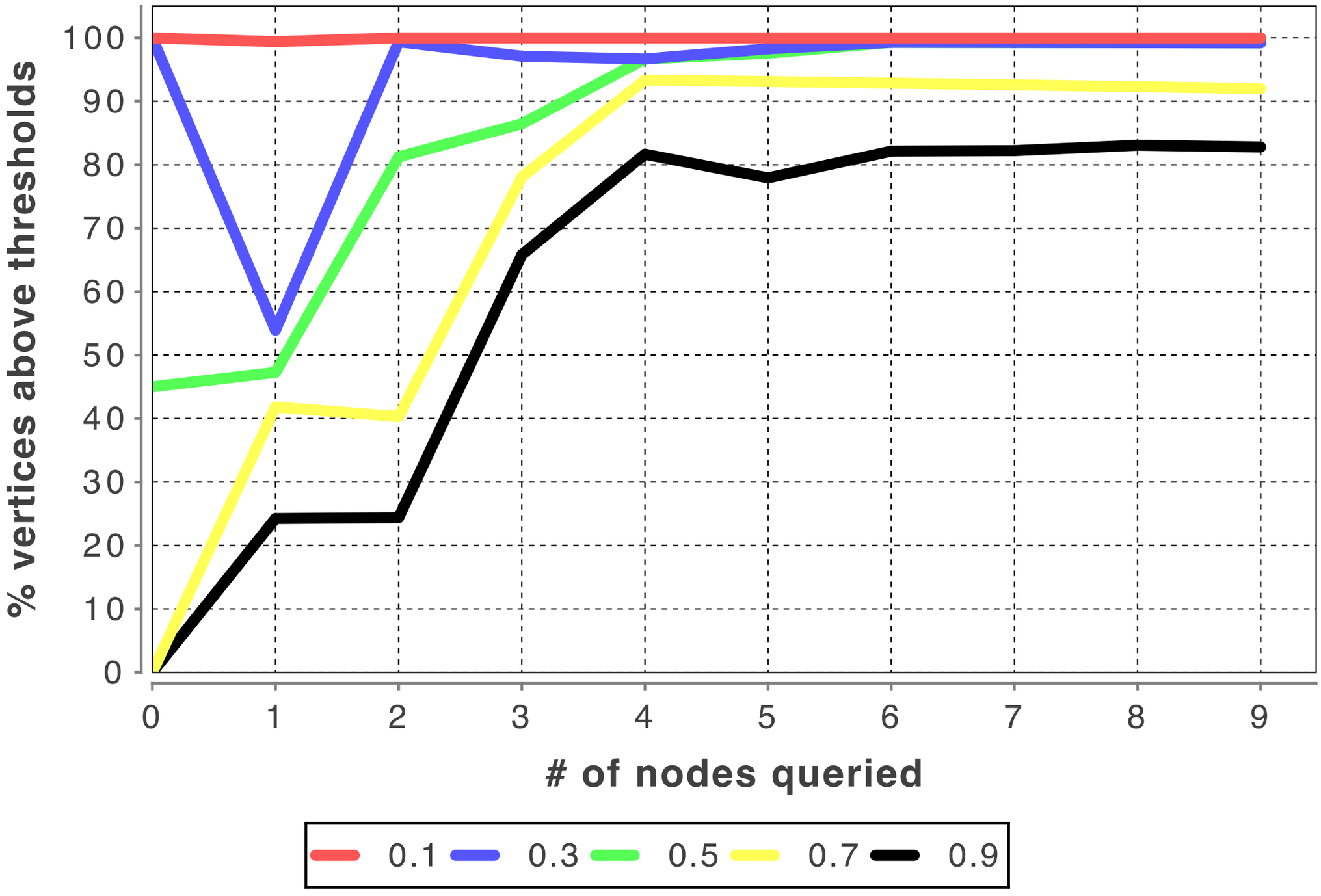
$$I(v, G - v) = H(v) - H(v \mid G - v)$$
$$= H(G - v) - H(G - v \mid v)$$

average amount of information we learn about *G–v* we learn by querying *v*

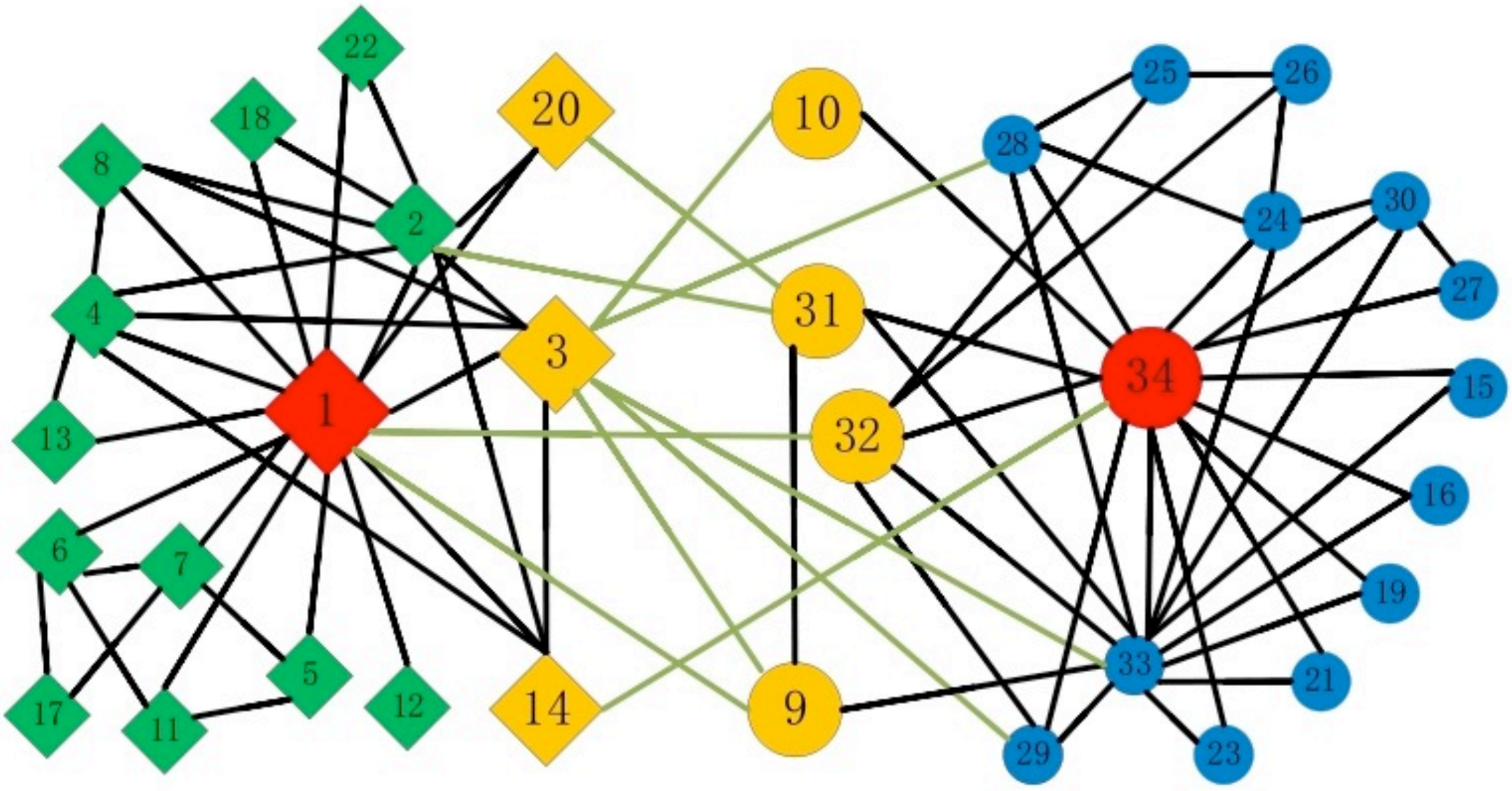
high when we're uncertain about *v*, and when *v* is highly correlated with others

[Moore, Yan, Zhu, Rouquier, Lane]

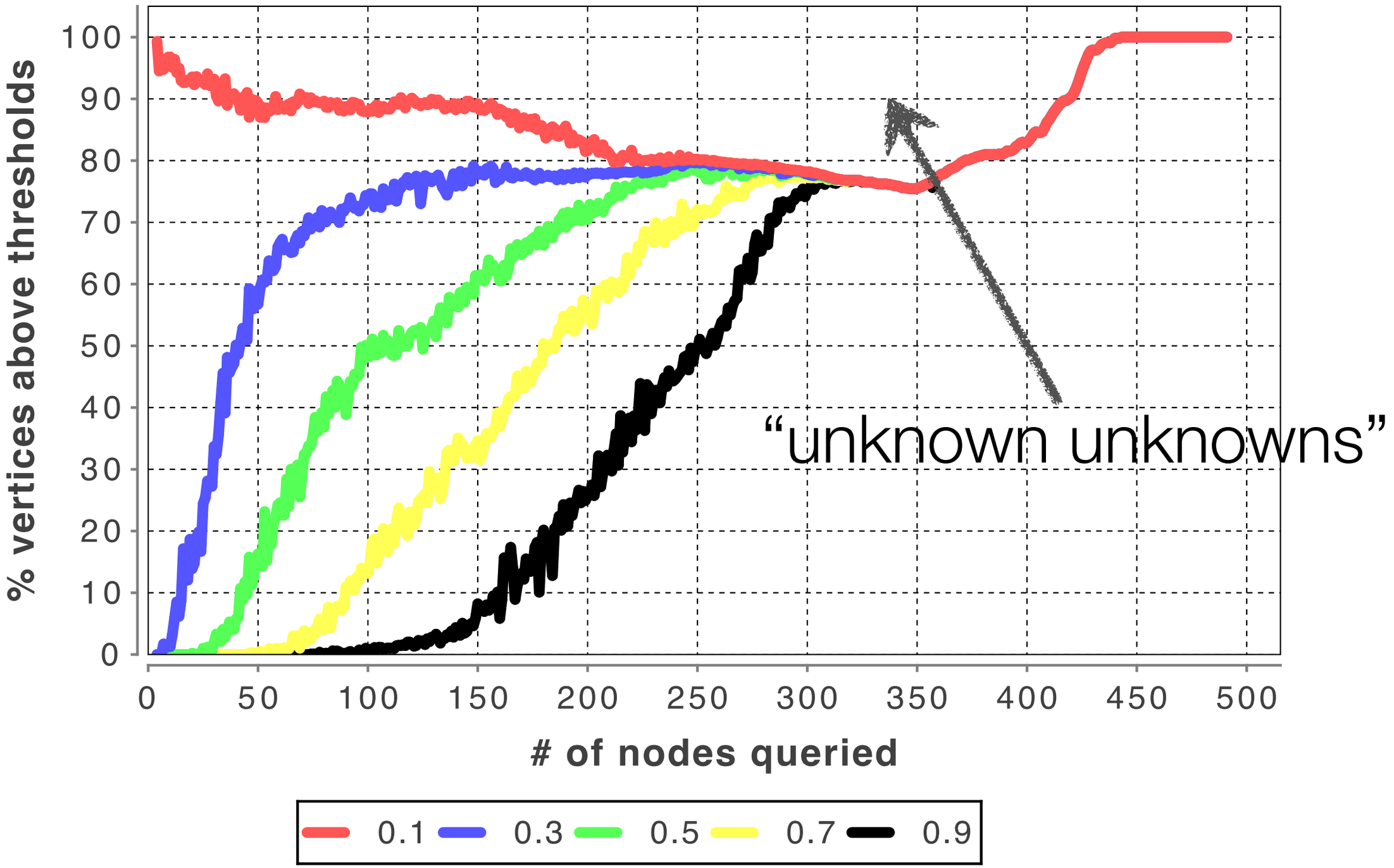
Learning factions in the Karate Club



Which vertices do we query first?



An antarctic food web



Dealing with uncertainty #3: Identifying groups of technologies

patents are documents, and have links (citations) between them

how can we identify groups of technologies, and understand how they depend on each other?

test case: 1,000 microprocessor patents

arithmetic	testing	power	protection	branching
multiplexer	debugging	reset	transparent	prediction
buses	emulator	frequencies	security	concurrence
microinstructions	error	pulses	multi-tasking	speculation
microprograms	traces	voltages	encryption	reordering
	embedding	sensing	restricting	
	jumps	driving		
	halting	oscillators		

using both text and links does better than using either one alone

[Zhu, Yan, Getoor, Moore]

The story so far

statistical inference, powered by ideas from physics, and carried out with highly scalable algorithms, lets us

detect communities

label nodes

predict missing links

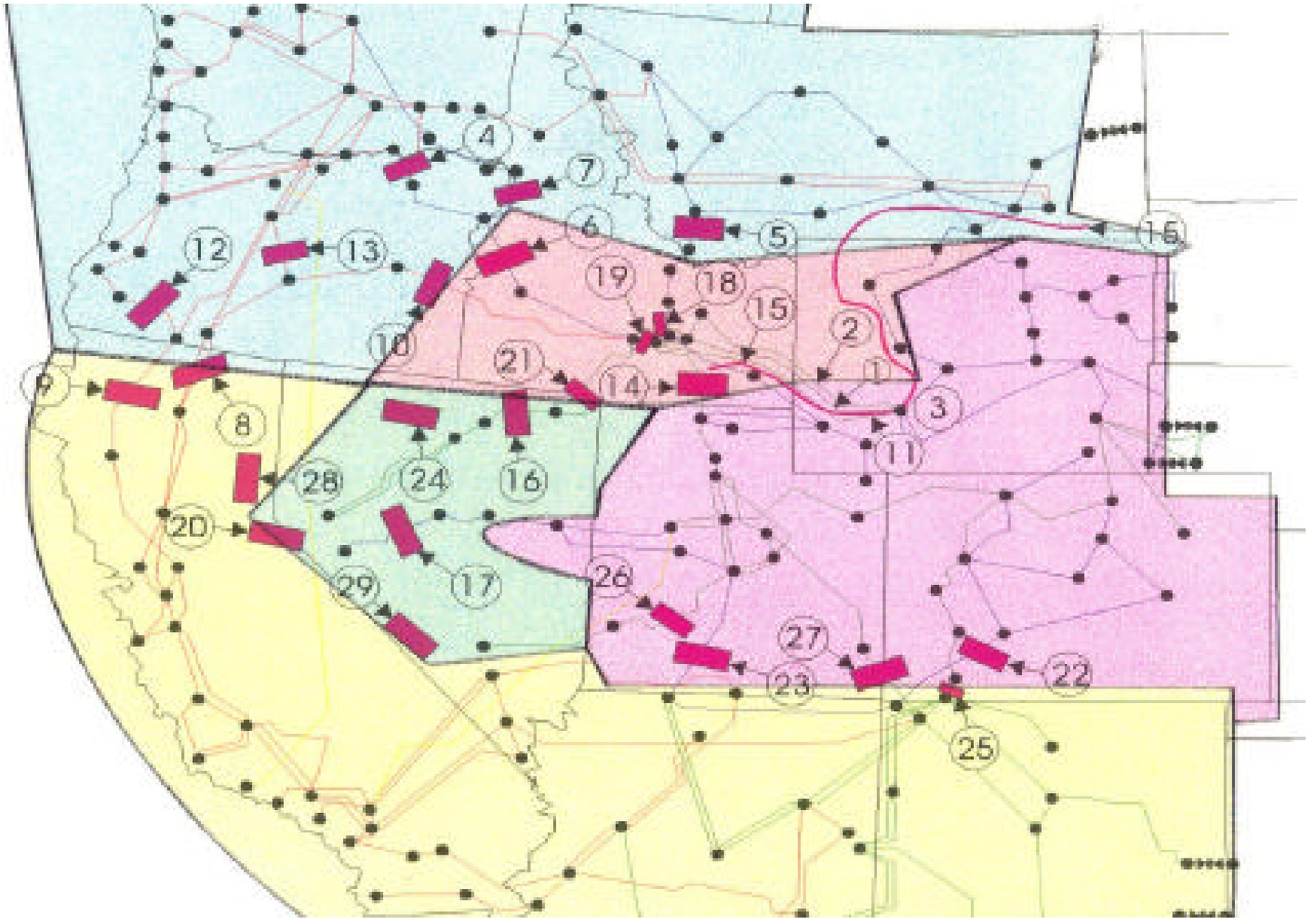
these models and algorithms reveal phase transitions where communities become detectable, or where knowledge suddenly spreads across the network

we can elaborate these models by adding discrete or continuous attributes: degree distributions, edge types, social status, overlapping communities, hierarchy, signed edges, document content...

but a cautionary note!

A real cascade of line and generator failures

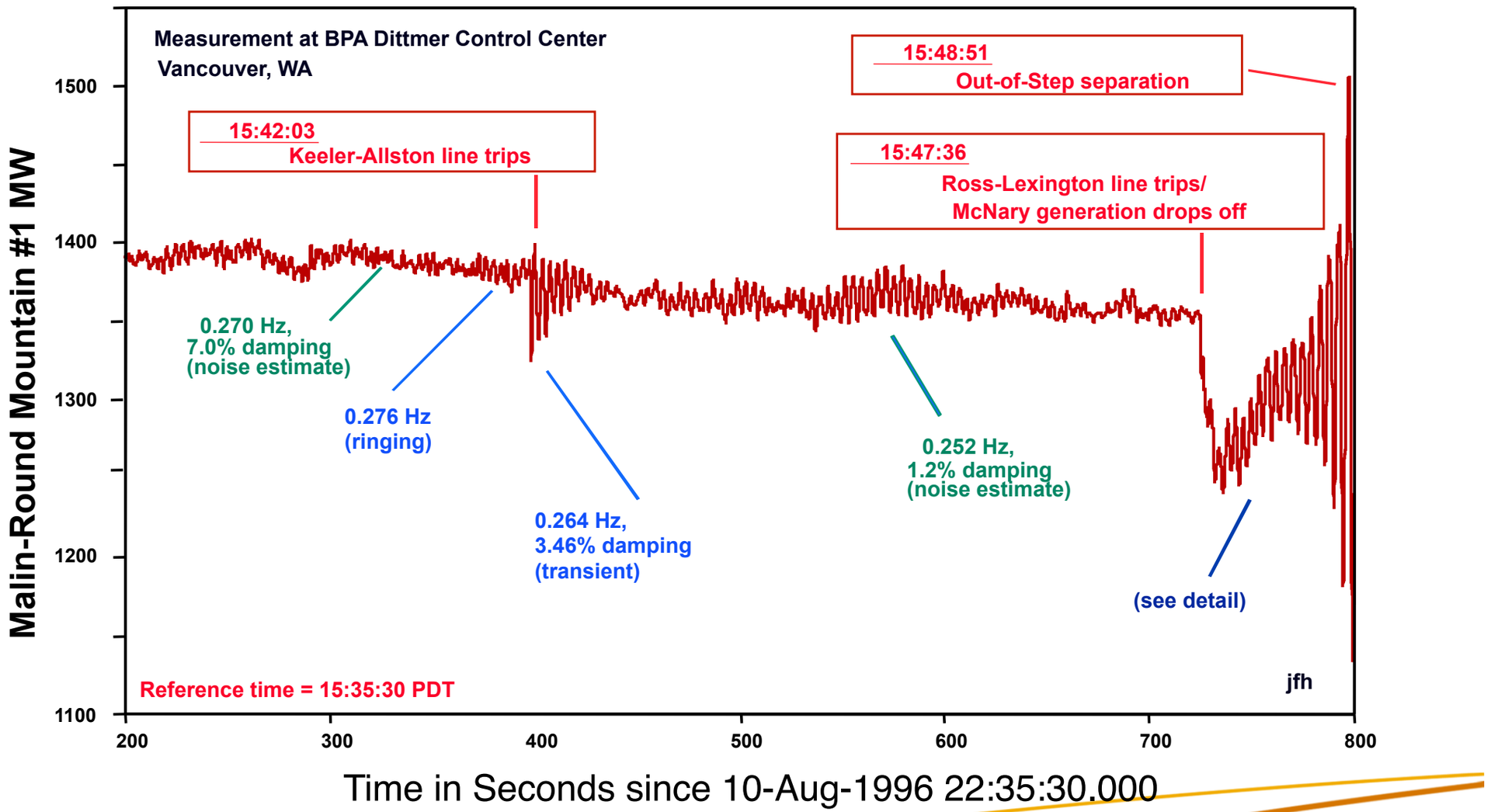
Sequence of outages in Western blackout, July 2 1996



from NERC 1996 blackout report

Rich dynamics of coupled, nonlinear oscillators

Sequence of Events



Pacific Northwest
NATIONAL LABORATORY

Beyond topology

we can’t understand networks with topology alone

networks are rich, dynamic data sets, not just lists of nodes and edges

nodes and edges have rich attributes:

power grid: generators have nonlinear dynamics at many time scales, transmission lines have capacities, users have fluctuating demands...

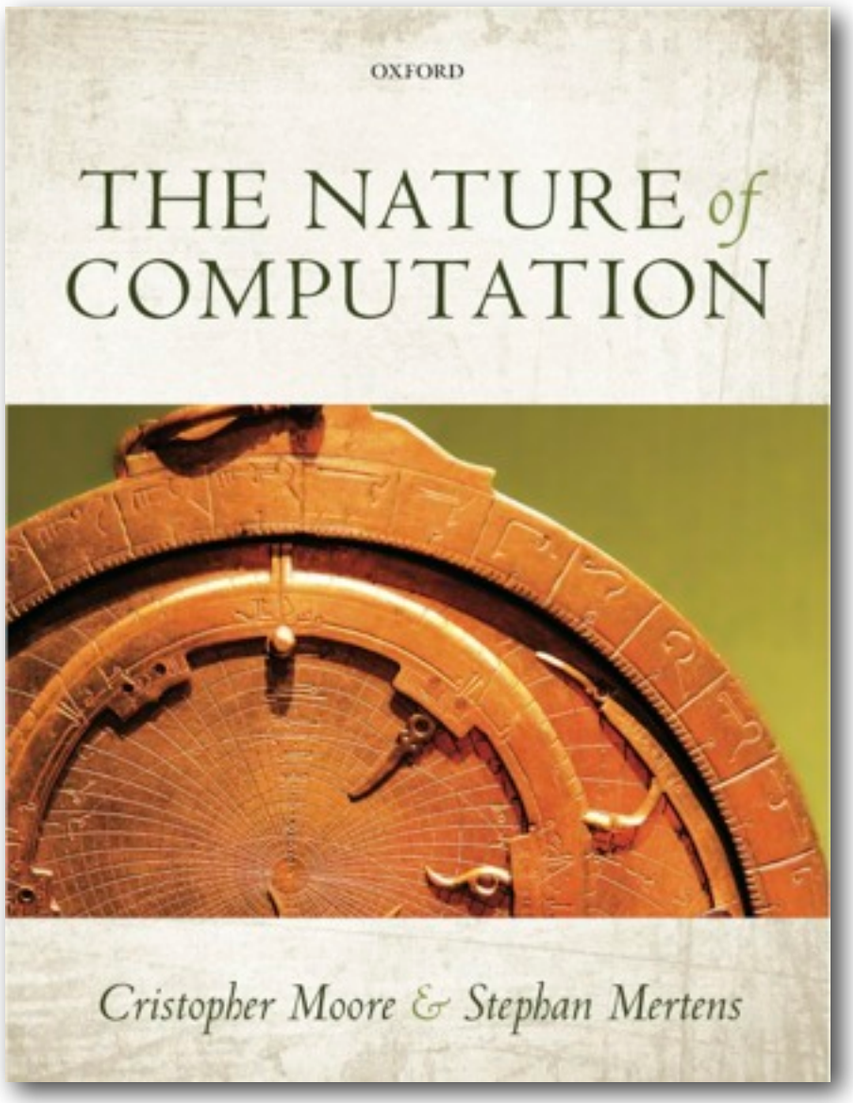
cybersecurity: multiple types of links between computers (web fetches, SSH links) with timing, duration, packet size... and many links are unique

food webs: species have populations, links have nutrient flows.... dynamic response to climate change, species loss, invasive species

extending statistical inference to richer data is possible, but challenging

the “best” model or algorithm is application dependent! do you want to label nodes? predict missing links? understand dynamics? or what?

Shameless Plug



To put it bluntly: this book rocks! It somehow manages to combine the fun of a popular book with the intellectual heft of a textbook.

Scott Aaronson, MIT

www.nature-of-computation.org