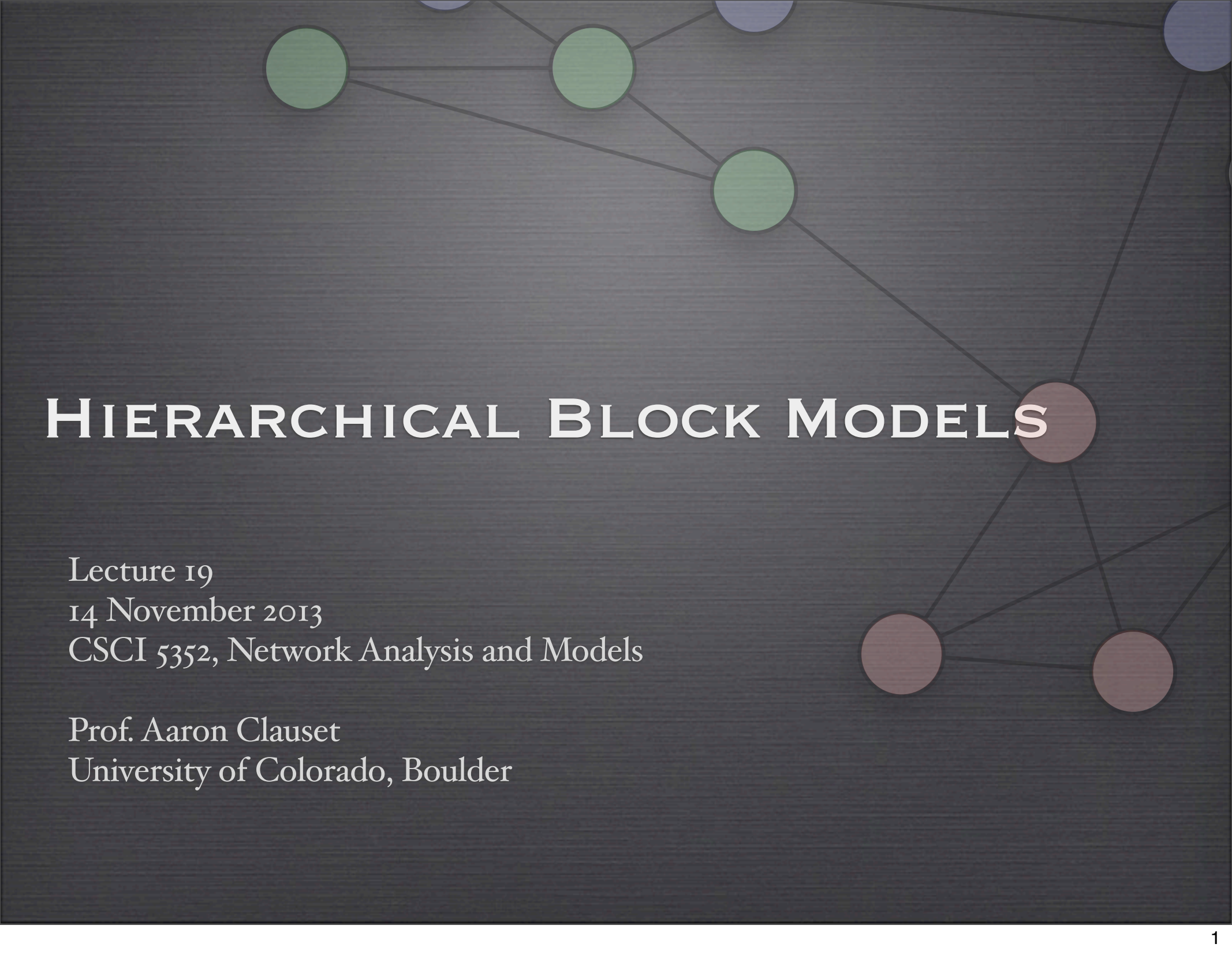


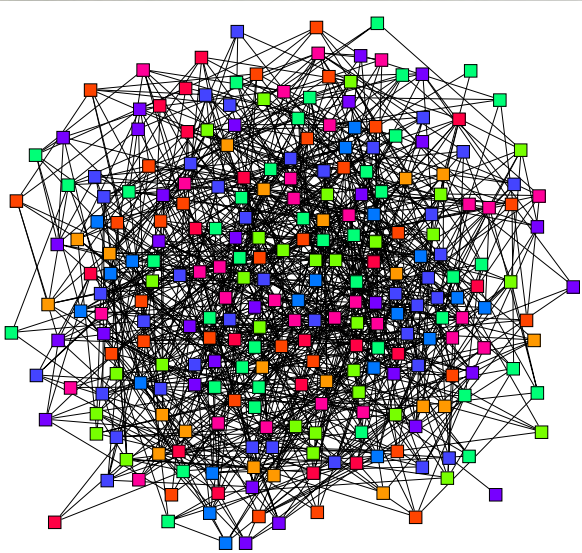
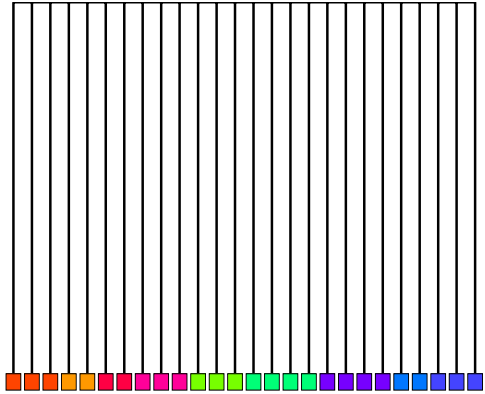


HIERARCHICAL BLOCK MODELS

Lecture 19
14 November 2013
CSCI 5352, Network Analysis and Models

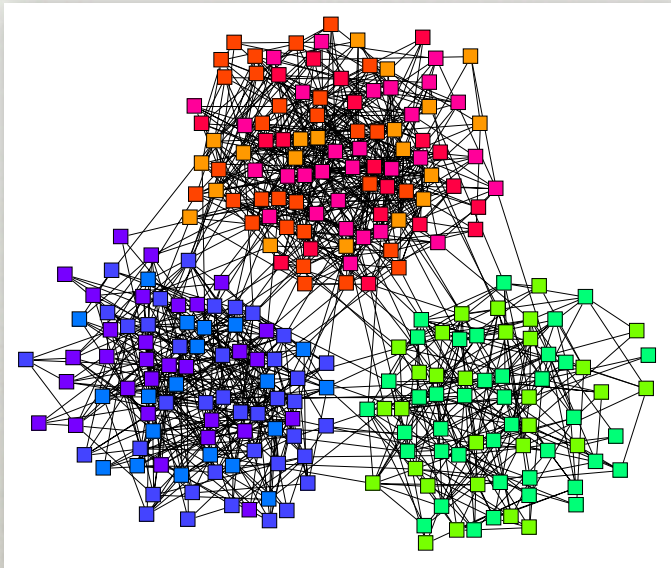
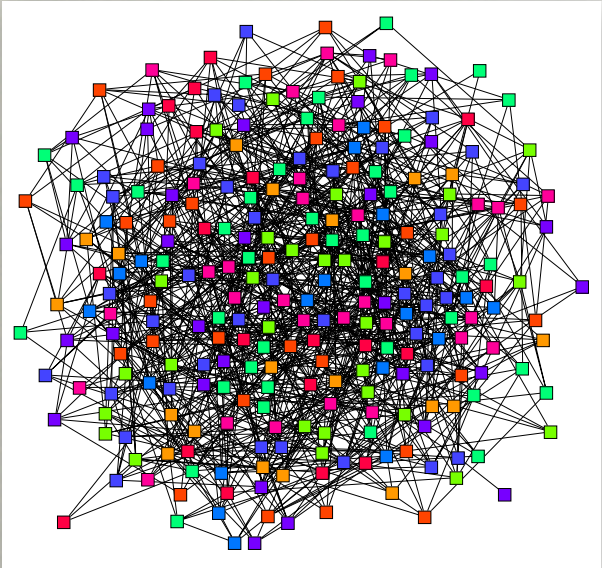
Prof. Aaron Clauset
University of Colorado, Boulder

no structure

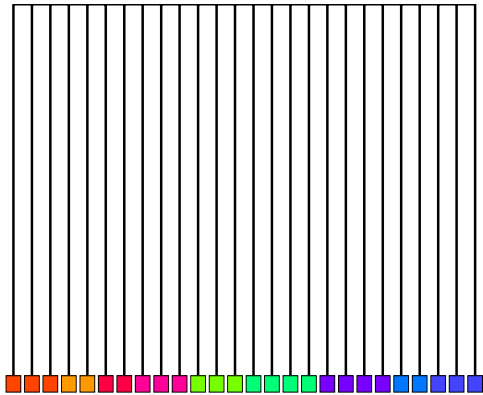


The diagram consists of 20 vertical lines. At the base of each line is a small square. The colors of these squares, from left to right, are: orange, orange, yellow, yellow, red, red, pink, pink, light green, light green, green, green, cyan, cyan, blue, blue, purple, purple, blue, and blue. This represents a sequence of 20 data points or categories.

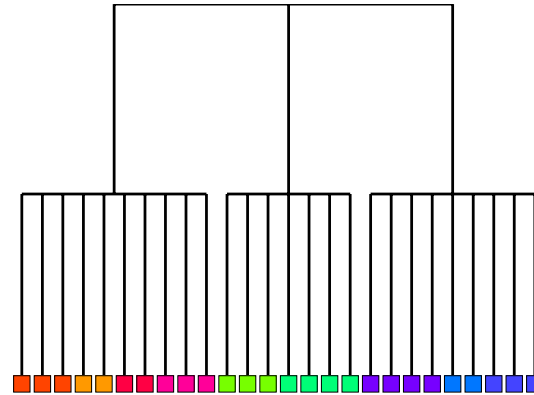
The diagram illustrates a hierarchical tree structure. At the top is a single root node (rectangle). This root node branches into three intermediate nodes. Each of these intermediate nodes further branches into four leaf nodes, resulting in a total of 12 leaf nodes at the bottom level. The leaf nodes are colored in a sequence: 4 orange, 4 pink, 4 green, 4 purple, and 4 blue.

[illegible]

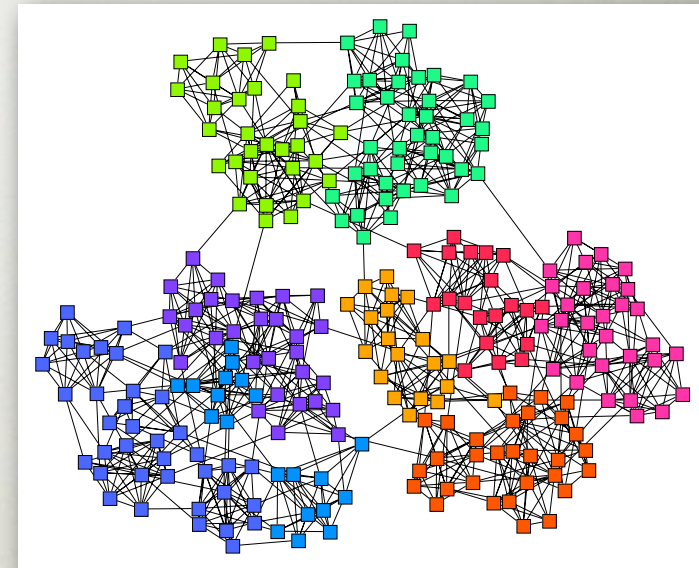
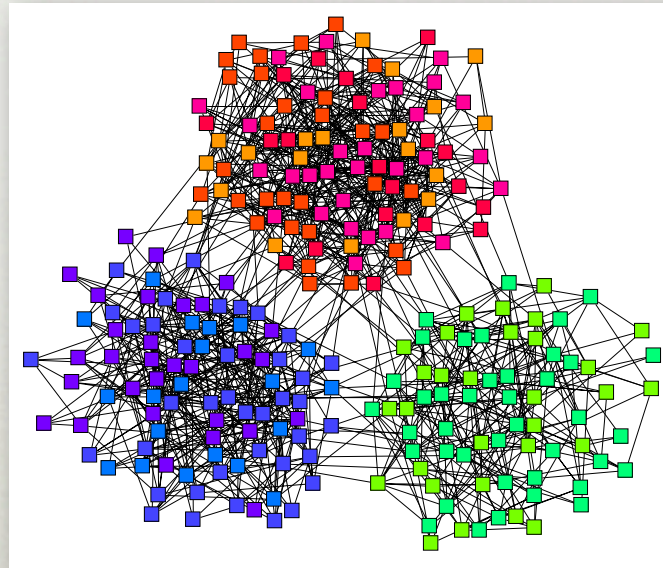
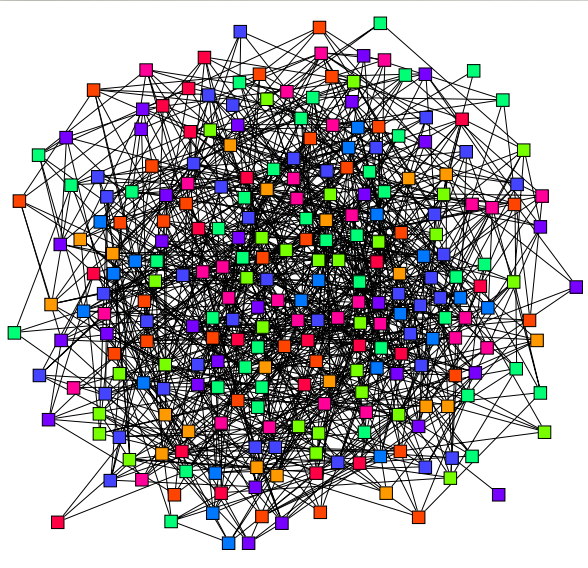
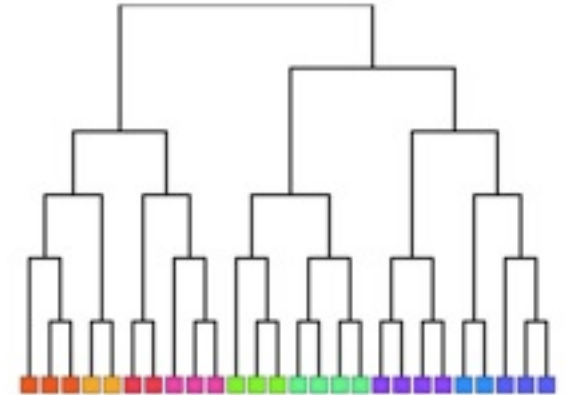
no structure



modular structure



hierarchical structure

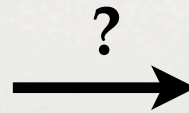
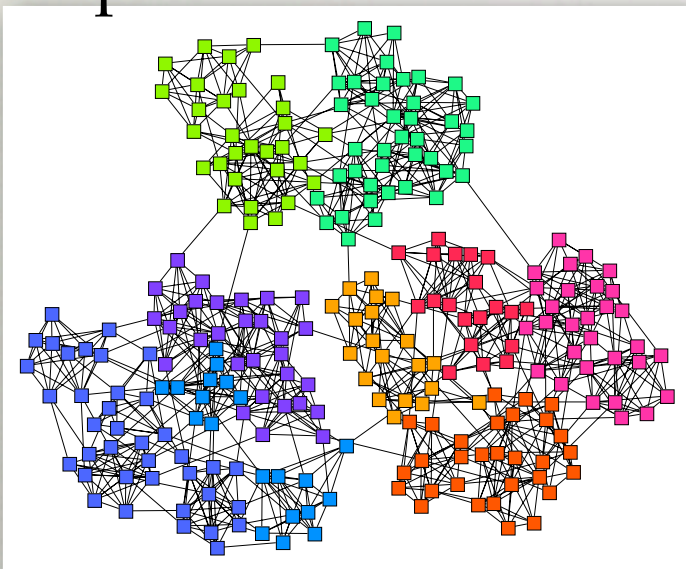


one scale

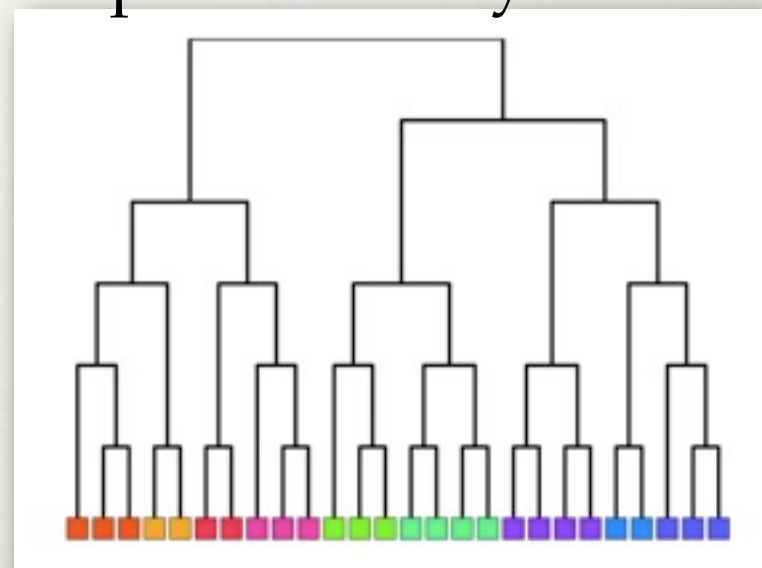
multi-scale

how can we measure a network's hierarchy?

step 1: network data



step 3: hierarchy



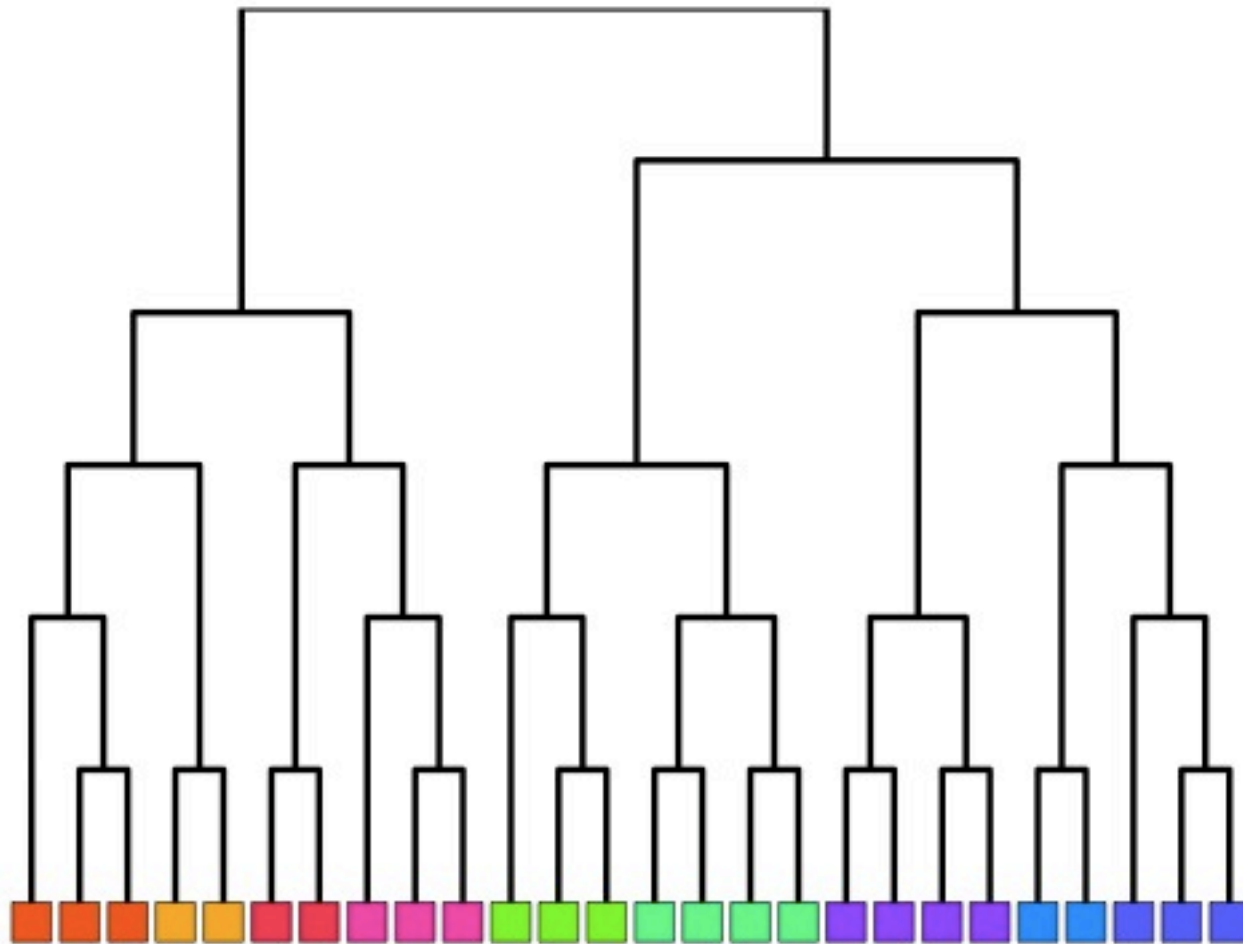
ONE APPROACH

model-based inference

1. describe how to generate hierarchies (a model)
2. estimate / learn model from data (algorithms)
3. test fitted model(s)
4. extract predictions, insight

A MODEL OF HIERARCHY

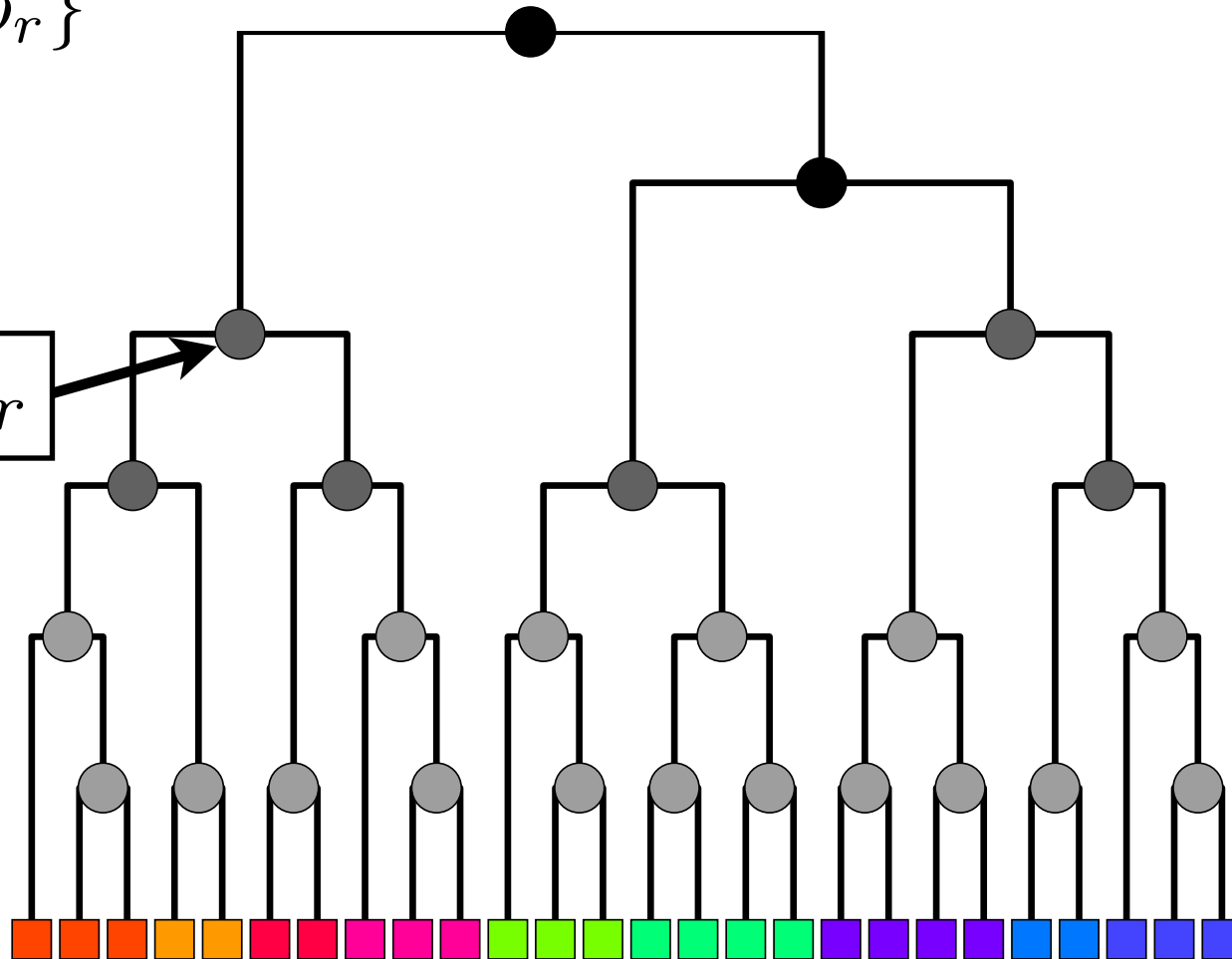
$\mathcal{D}$



A MODEL OF HIERARCHY

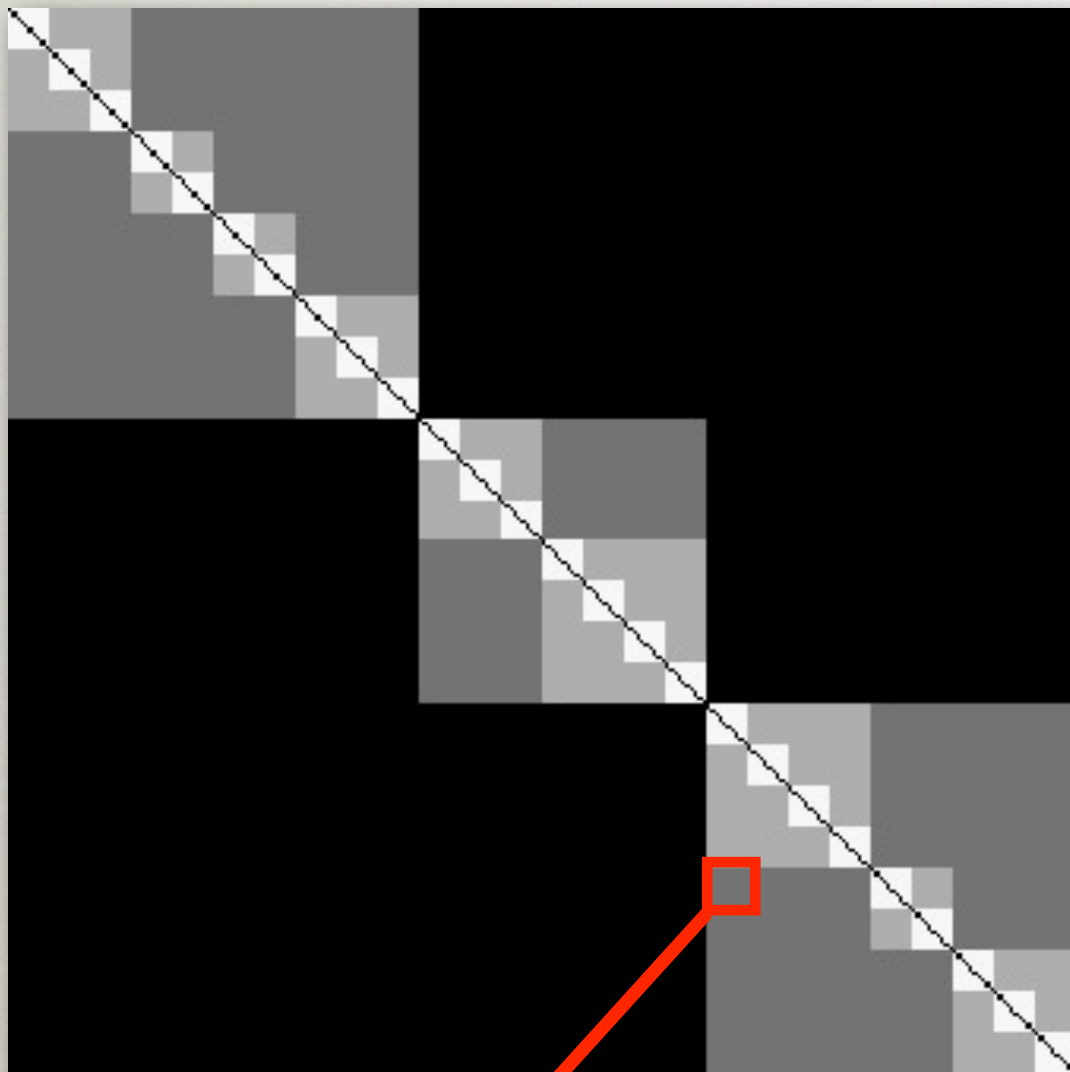
$\mathcal{D}, \{p_r\}$

probability p_r



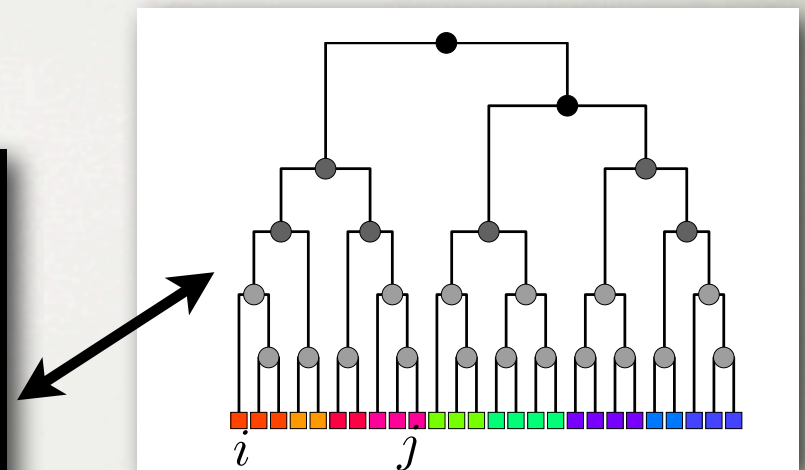
assortative modules

“inhomogeneous” random graph

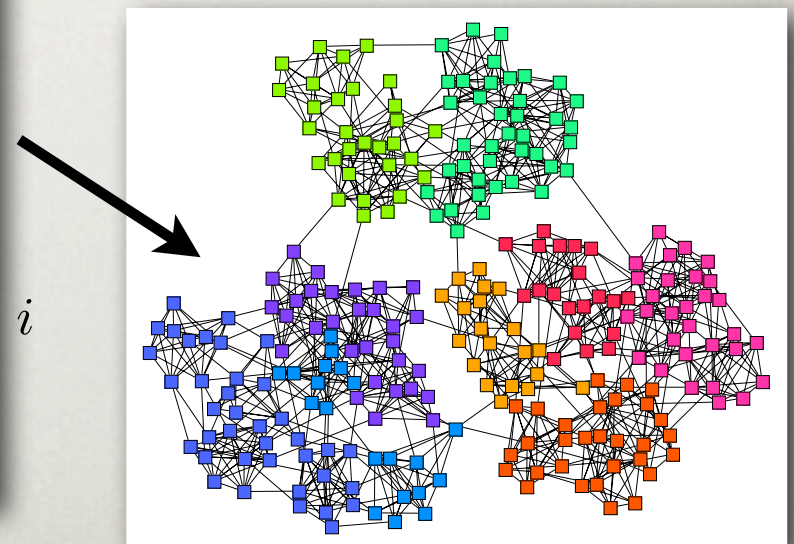


$$\begin{aligned} \Pr(i, j \text{ connected}) &= p_r \\ &= p_{(\text{lowest common ancestor of } i, j)} \end{aligned}$$

model



instance



HIERARCHICAL RANDOM GRAPH

- explicit model = explicit assumptions
- flexible ($2n$ parameters)
- captures structure at all scales
- mixtures of assortativity, disassortativity
- decomposition into set of random bipartite graphs
- learnable directly from data

LEARNING FROM DATA

a direct approach

- **likelihood function** $\mathcal{L} = \text{Pr}(\text{ data } | \text{ model })$
($\mathcal{L}$ scores **quality** of model)
- **sample all good models**
via Markov chain Monte Carlo*
over all dendrograms
- **technical details in**

Clauset, Moore and Newman, *Nature* **453**, 98-101 (2008) and

Clauset, Moore and Newman, *ICML* (2006)

* other sampling or optimization methods possible

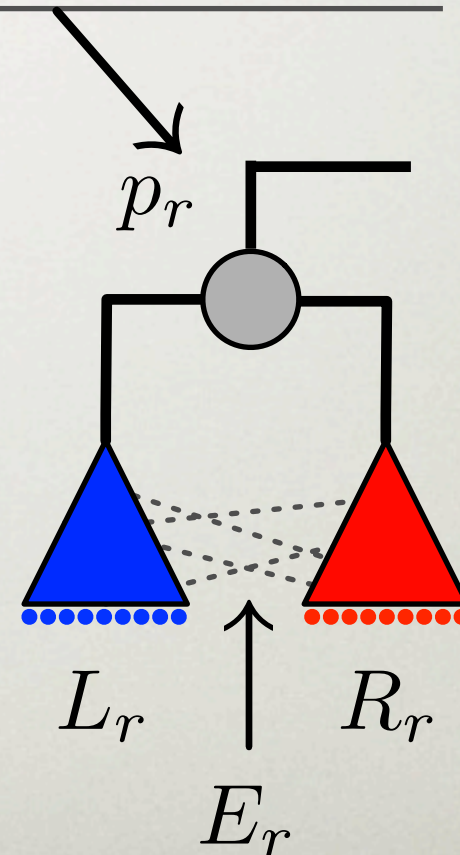
LIKELIHOOD FUNCTION

$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$

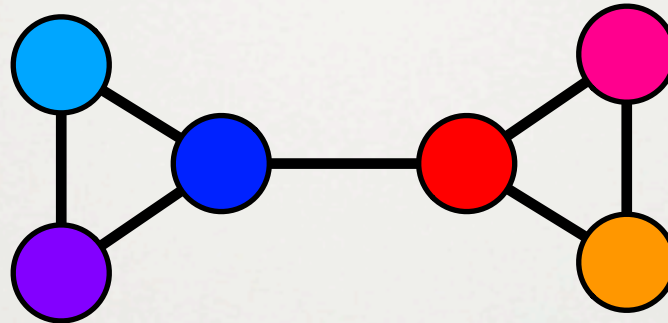
L_r = number nodes in left subtree

R_r = number nodes in right subtree

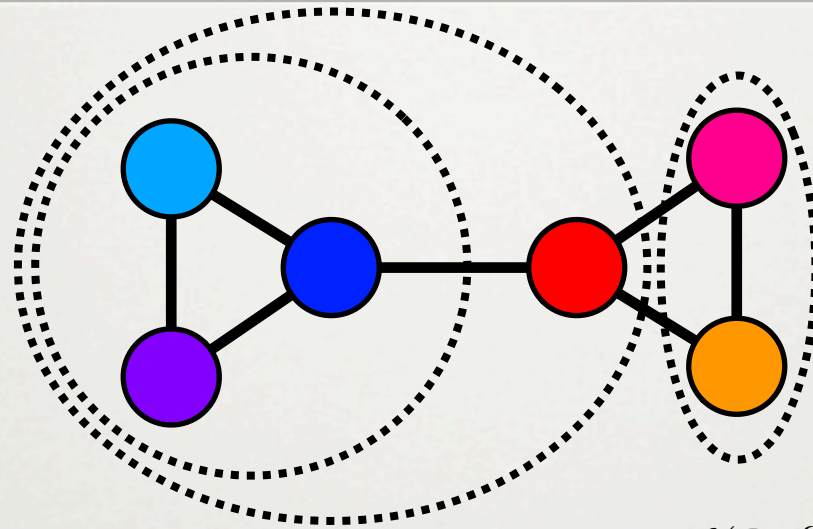
E_r = number edges with r as lowest common ancestor



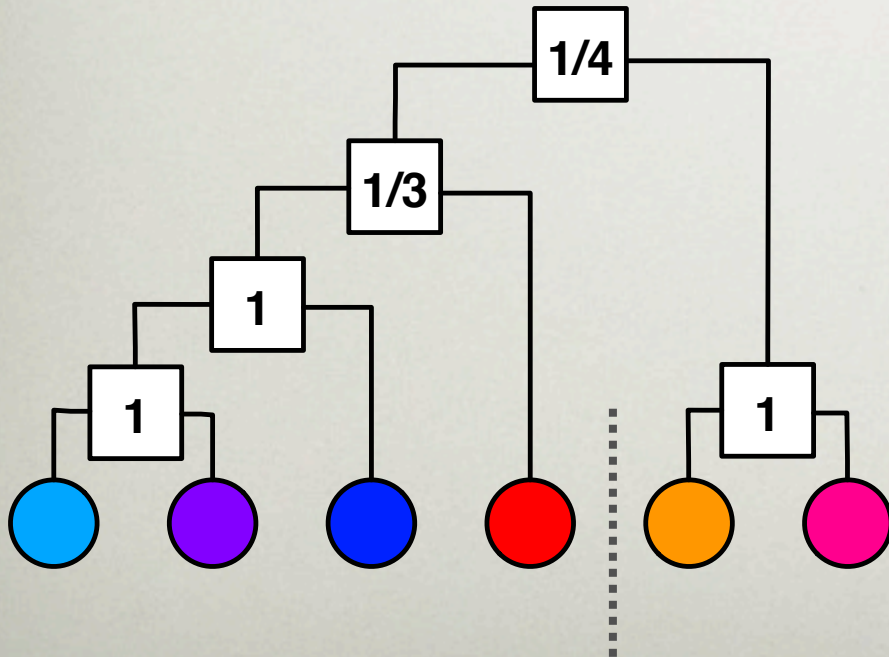
EXAMPLE



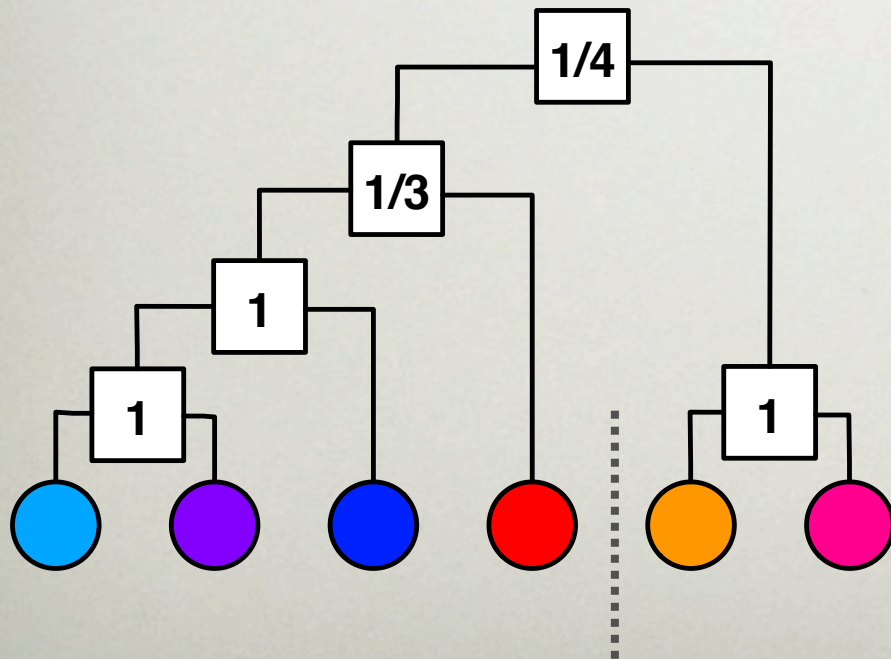
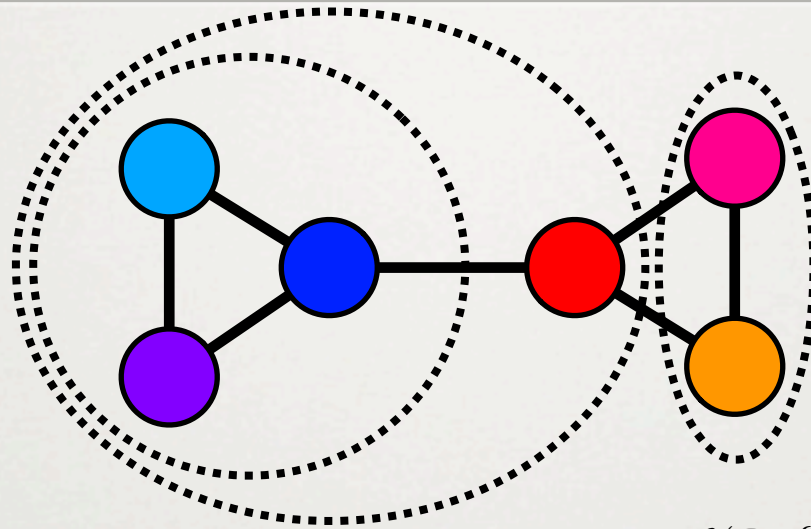
BAD DENDROGRAM



$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$



BAD DENDROGRAM

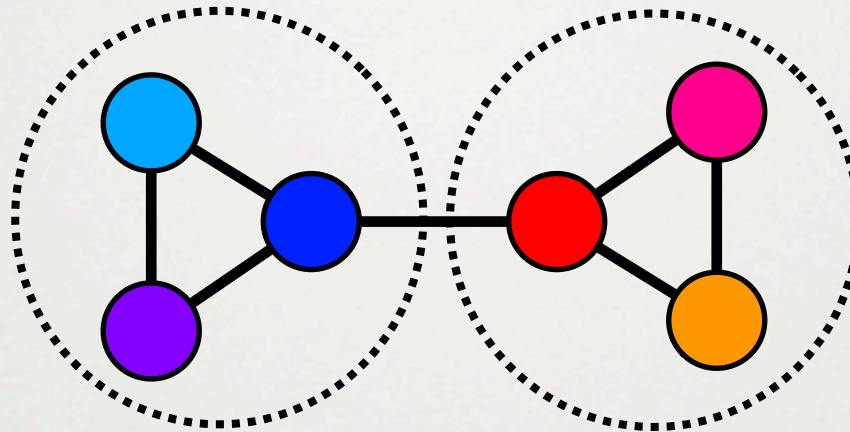


$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$

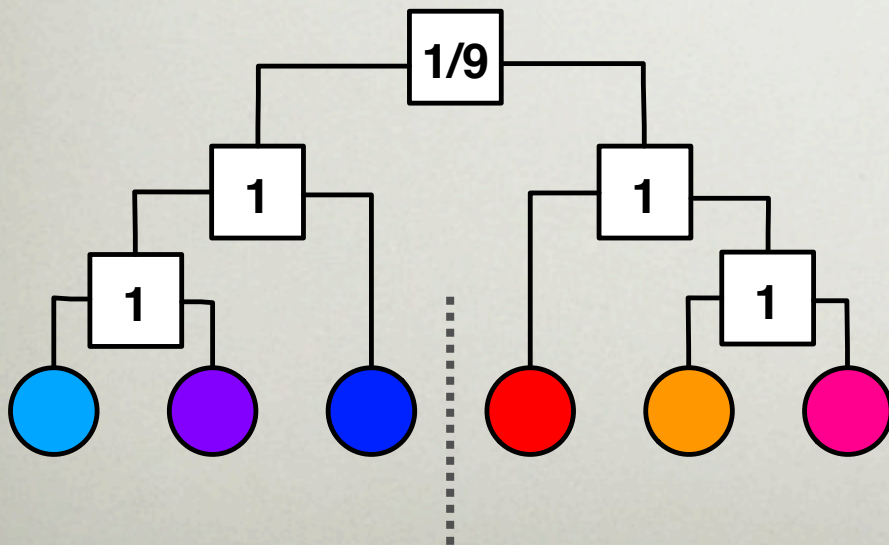
$$\mathcal{L} = \left[\left(\frac{1}{3} \right)^1 \left(\frac{2}{3} \right)^2 \right] \cdot \left[\left(\frac{1}{4} \right)^2 \left(\frac{3}{4} \right)^6 \right]$$

$$\mathcal{L} = 0.0016$$

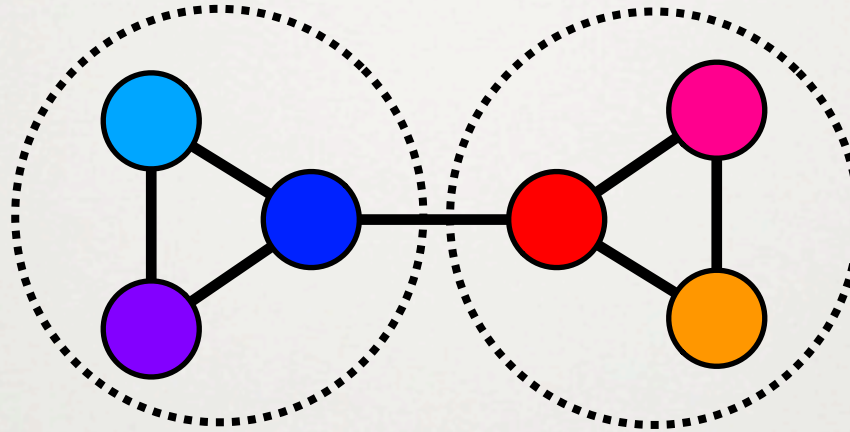
GOOD DENDROGRAM



$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$



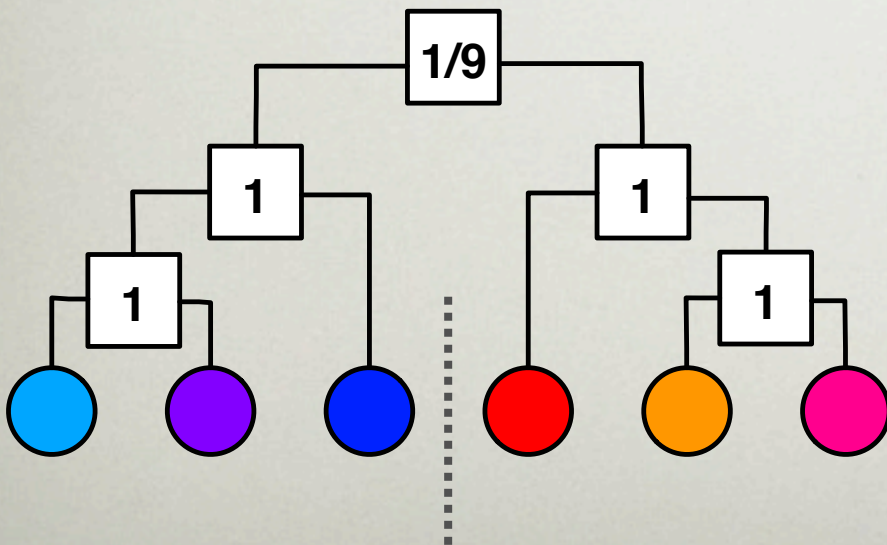
GOOD DENDROGRAM



$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$

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

$$\mathcal{L} = 0.0433$$



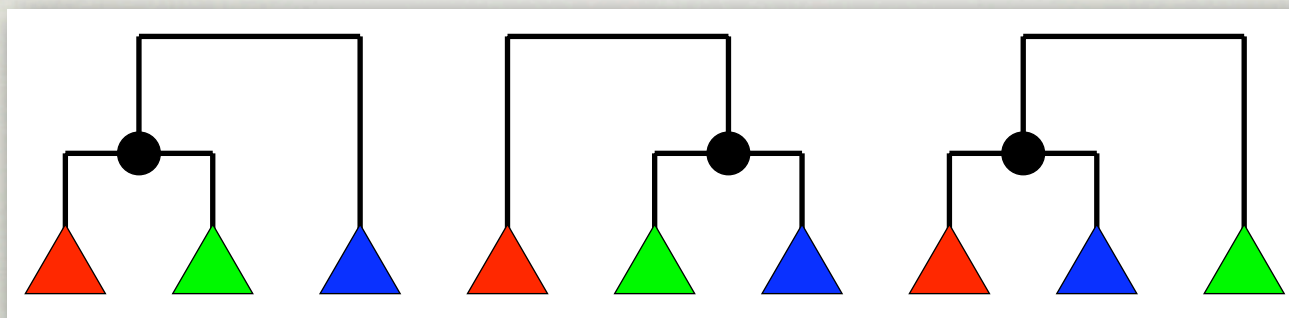
MARKOV CHAIN MONTE CARLO (MCMC)

Given $\mathcal{D}$, choose random internal node

Choose random reconfiguration of subtrees [ergodicity]

Recompute probabilities $\{p_r\}$ and likelihood $\mathcal{L}$

Sampling states according to their likelihood [detailed balance]



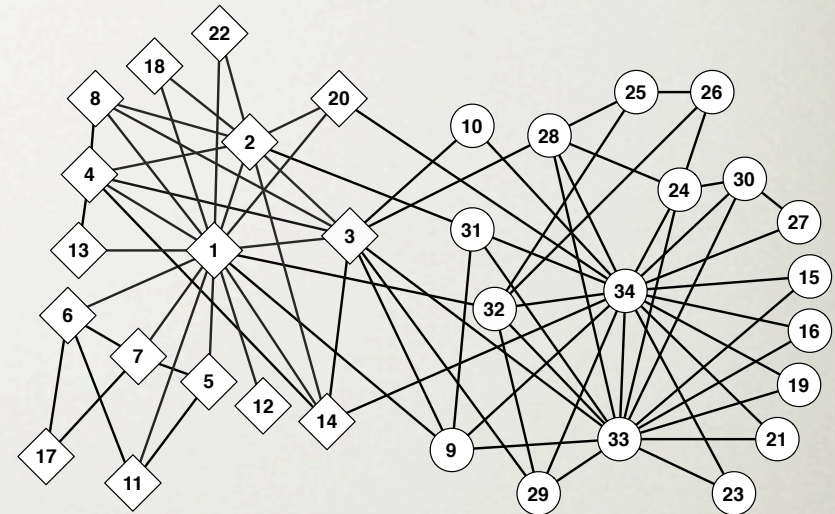
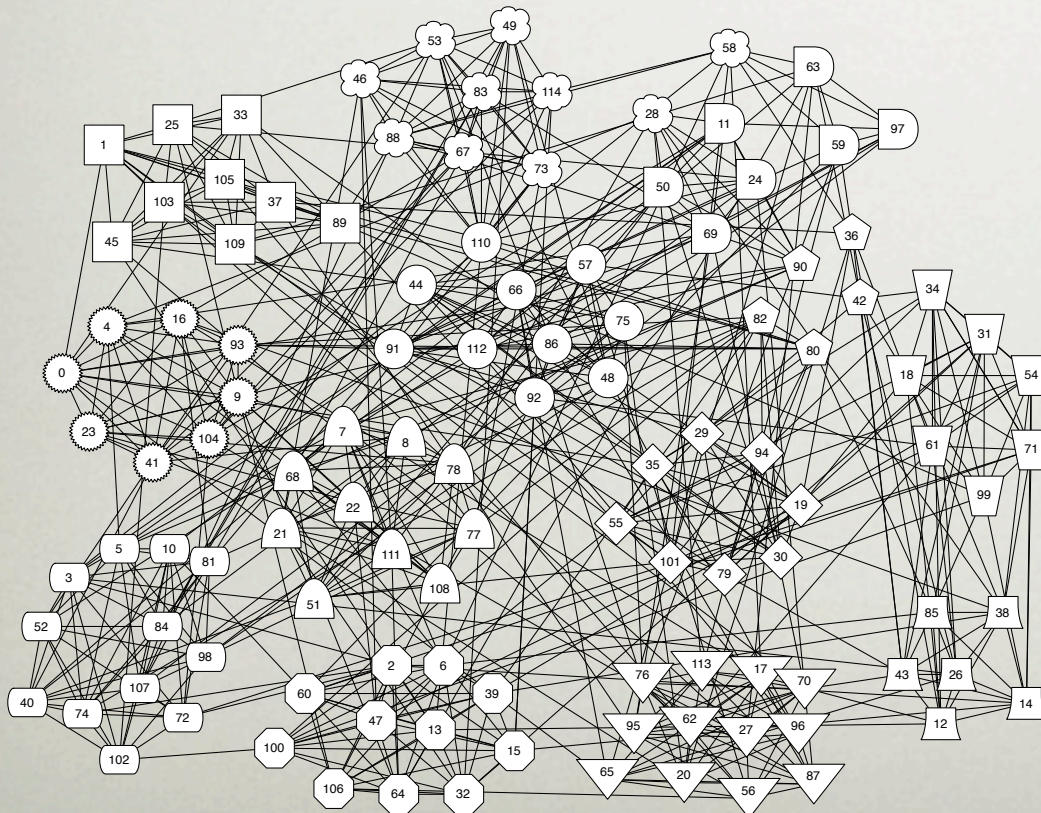
three subtree configurations
(up to relabeling)

SOME APPLICATIONS

TWO CASE STUDIES

NCAA Schedule 2000

$n = 115$ $m = 613$



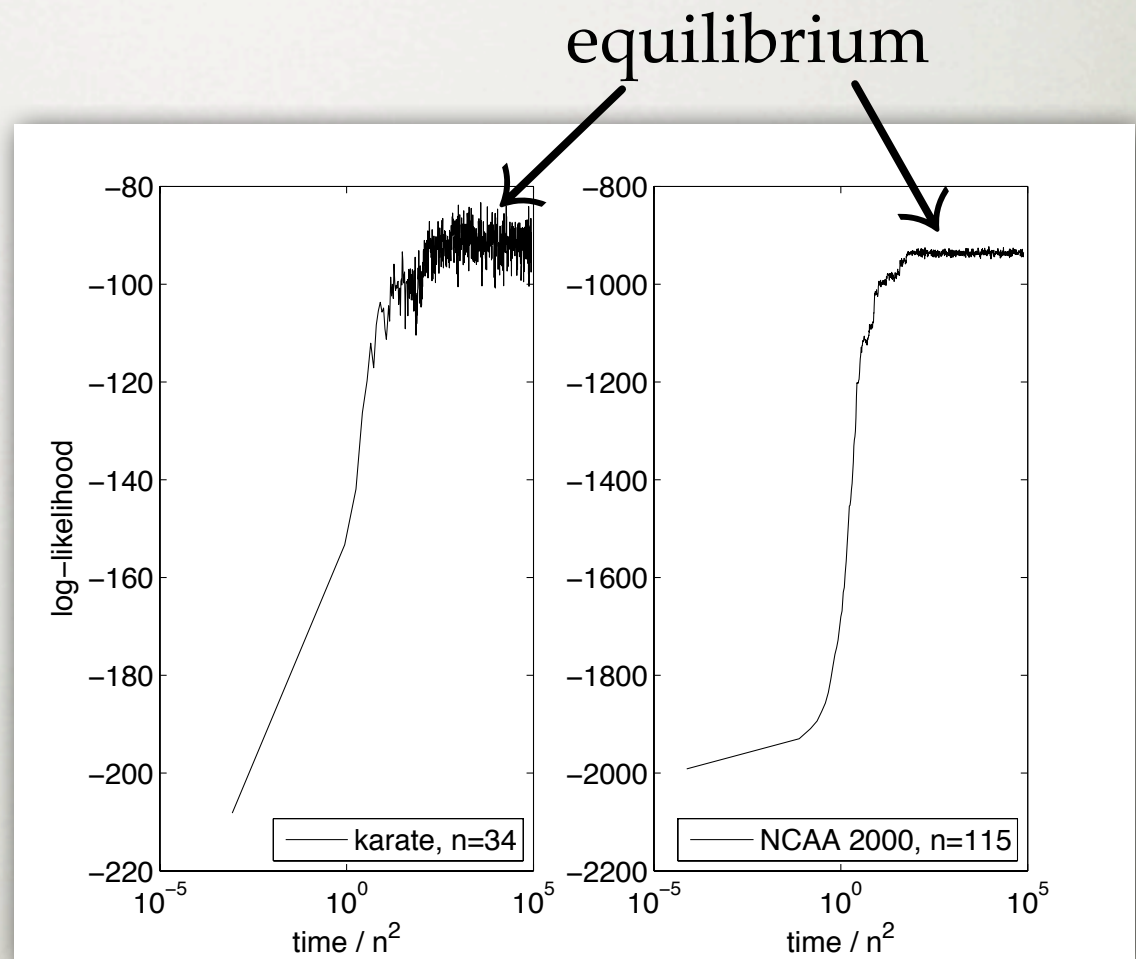
Zachary's Karate Club

$n = 34$ $m = 78$

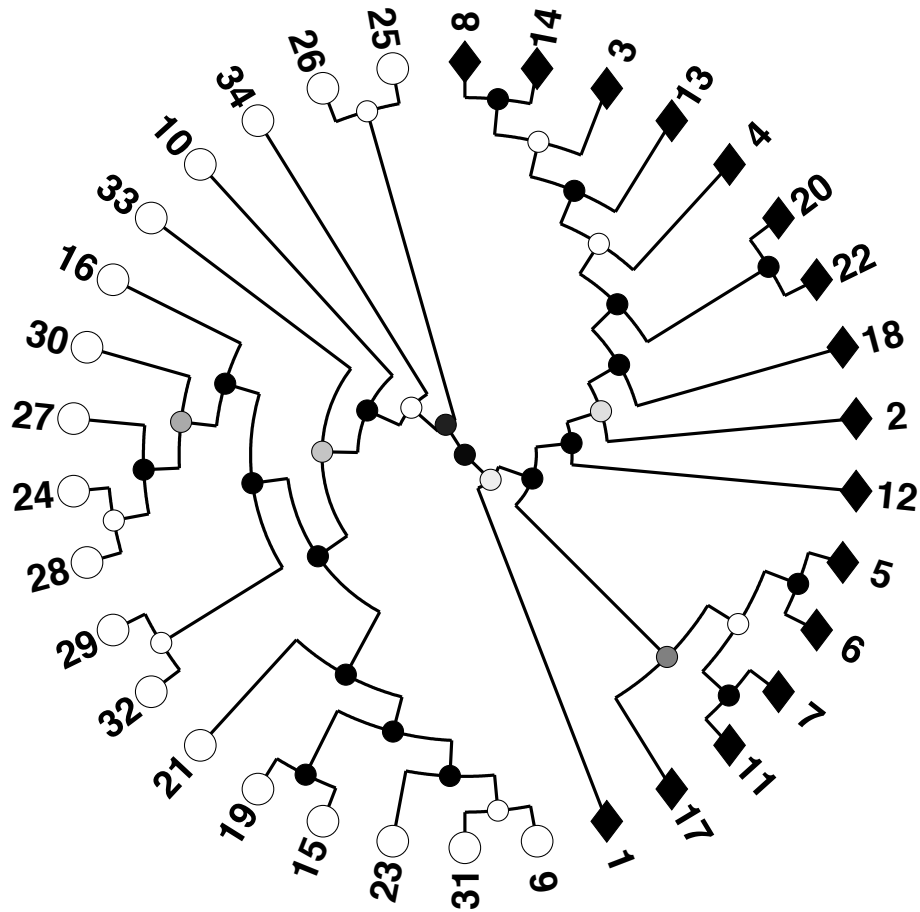
MIXING TIMES

MCMC mixes
relatively quickly

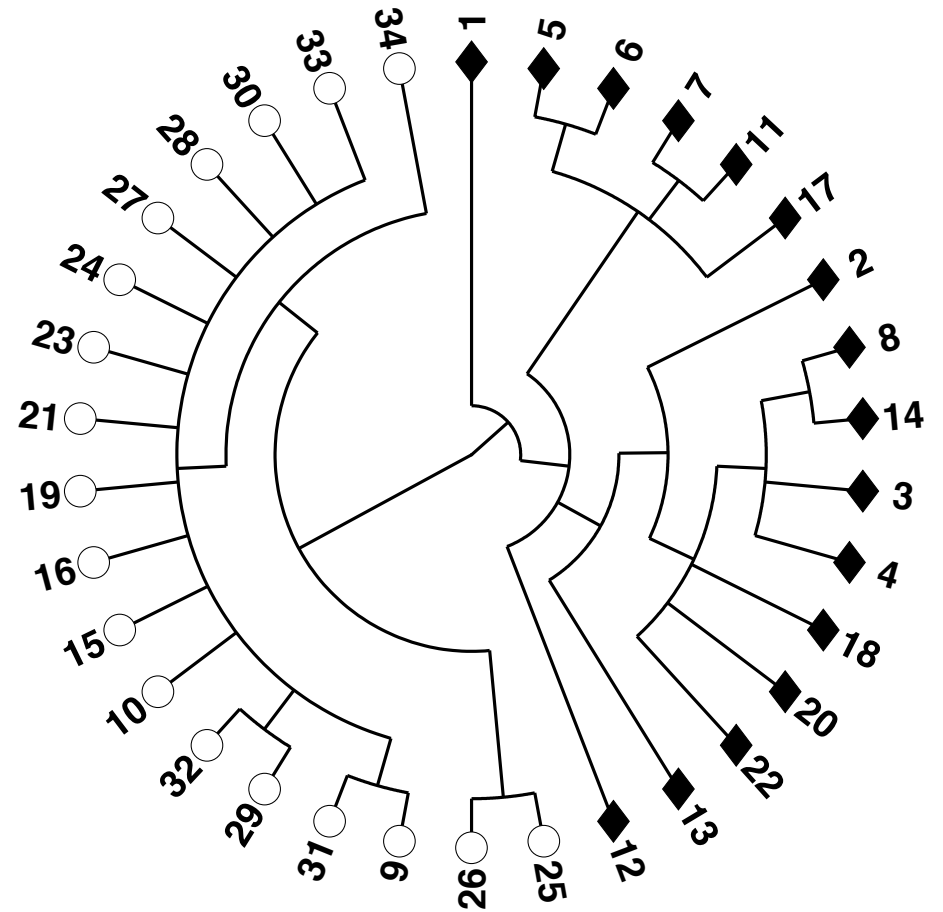
Equilibrium in
 $\sim O(n^2)$ steps



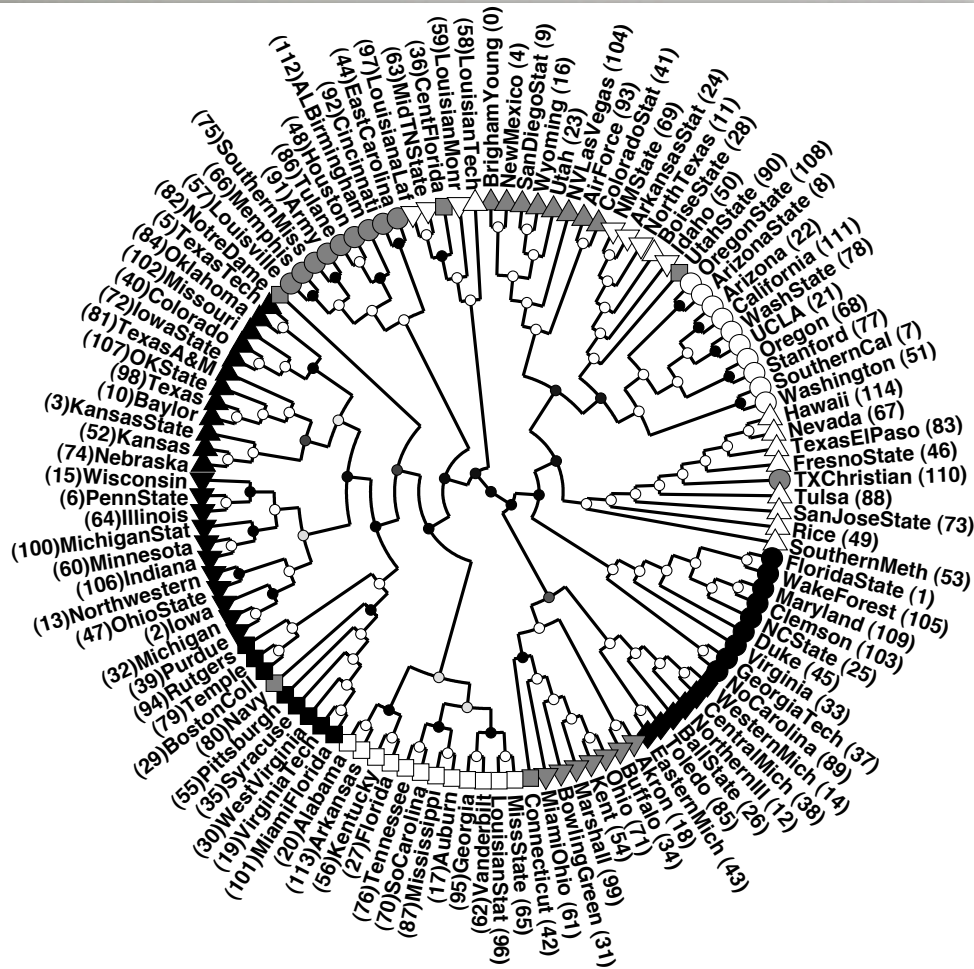
HIERARCHIES



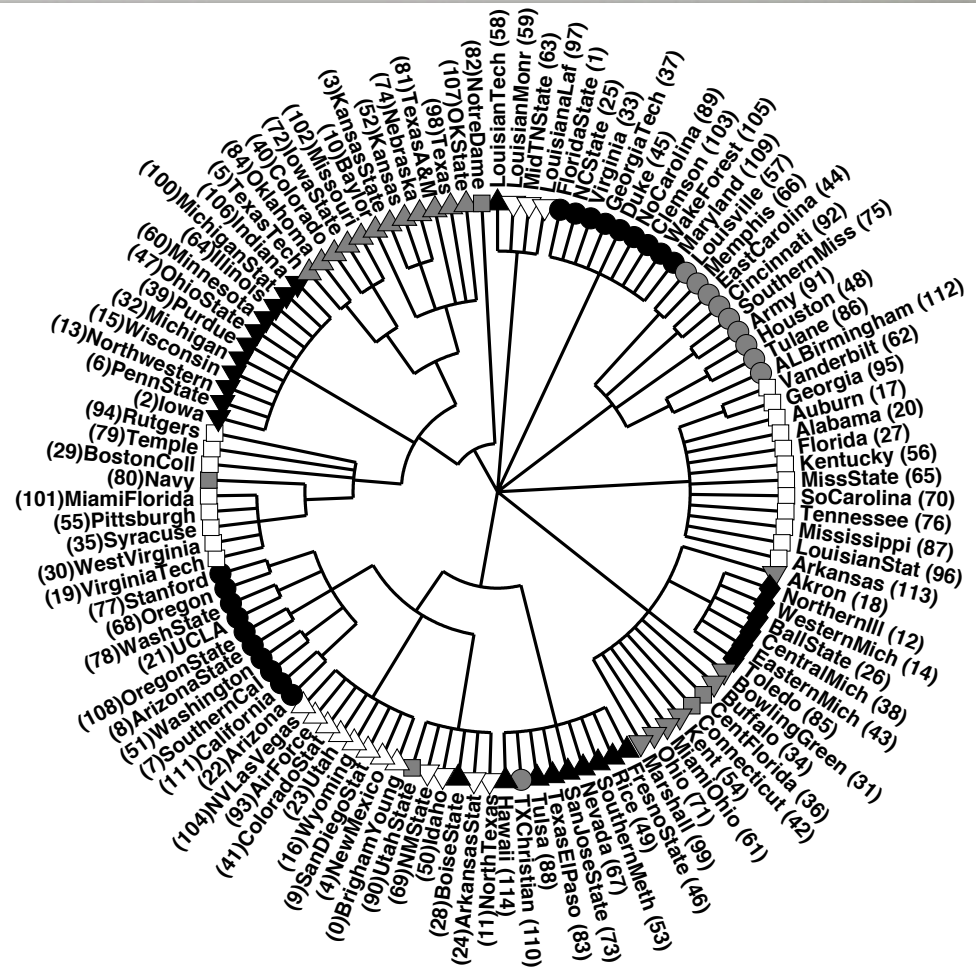
point estimate



consensus hierarchy



point estimate



consensus hierarchy

EDGE ANNOTATIONS

Average likelihood of edge existing

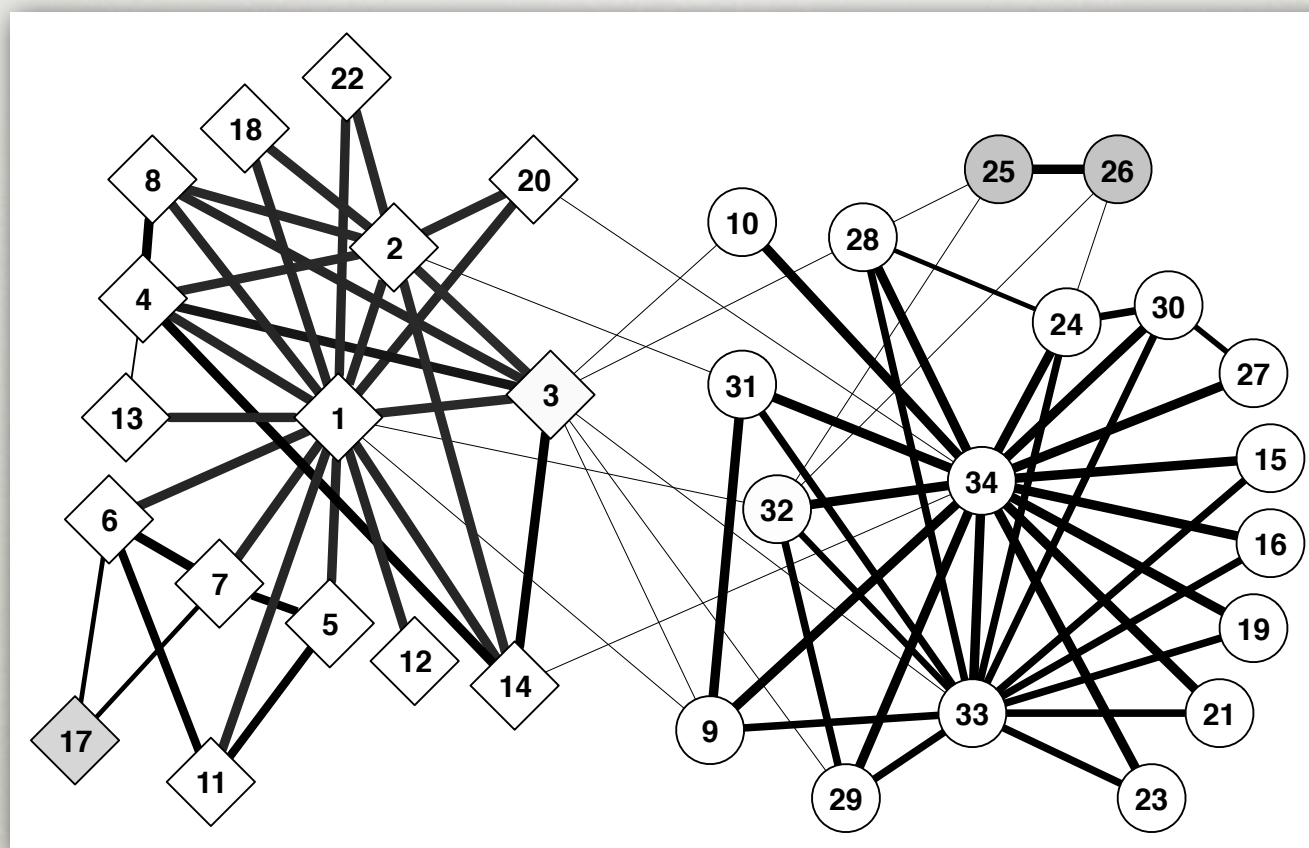
- For each edge (i, j) in G , compute average associated parameter $\langle \theta_r \rangle_{(i,j)}$ over sampled models
- $\langle \theta_r \rangle_{(i,j)}$ is edge annotation (weight)

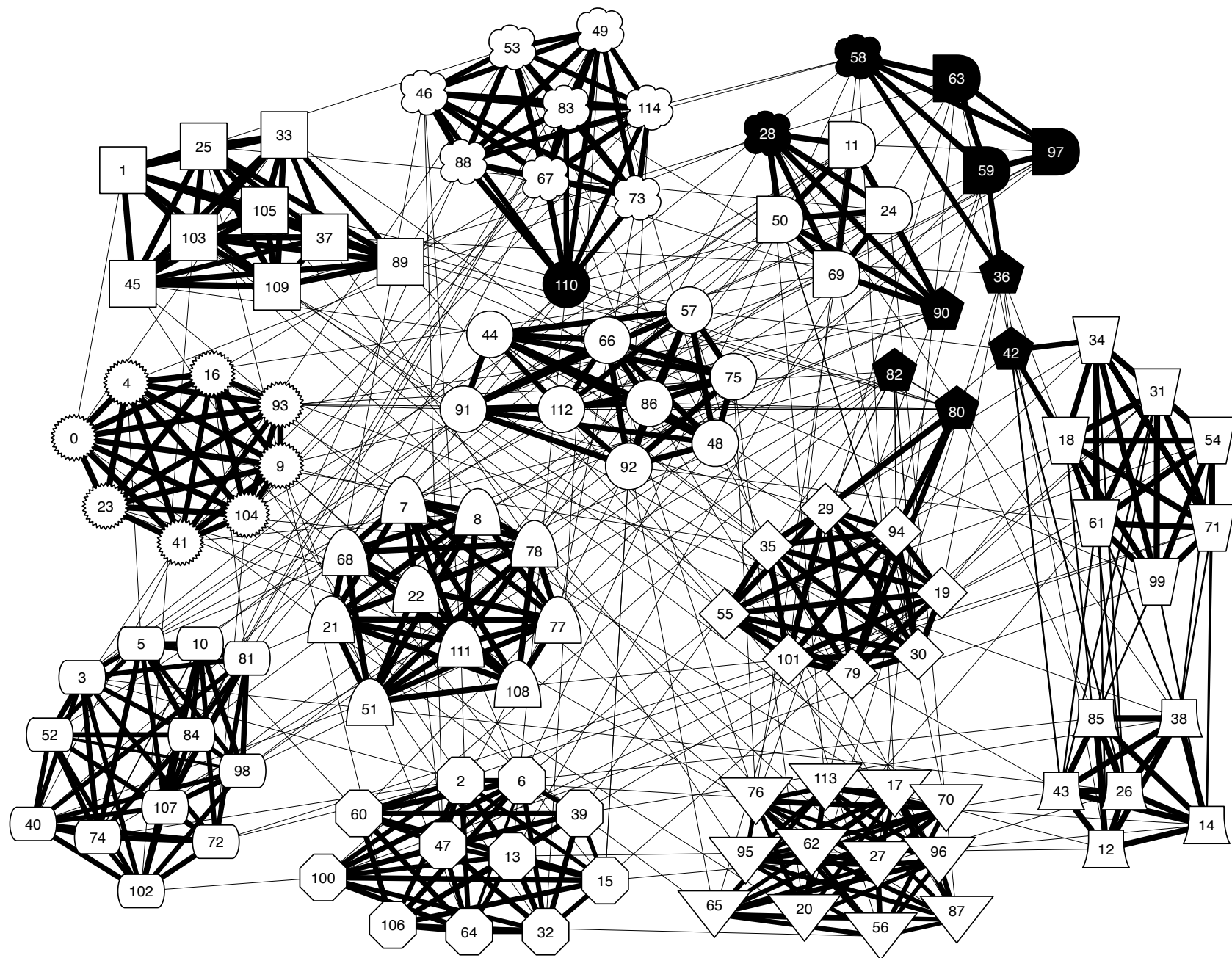
VERTEX ANNOTATIONS

Group-affiliation strengths

- If each vertex has known group label
- Ask, how often does vertex i appear in a subtree with majority of its fellows?
- Frequency is vertex annotation (strength)

EDGE, NOTE ANNOTATIONS



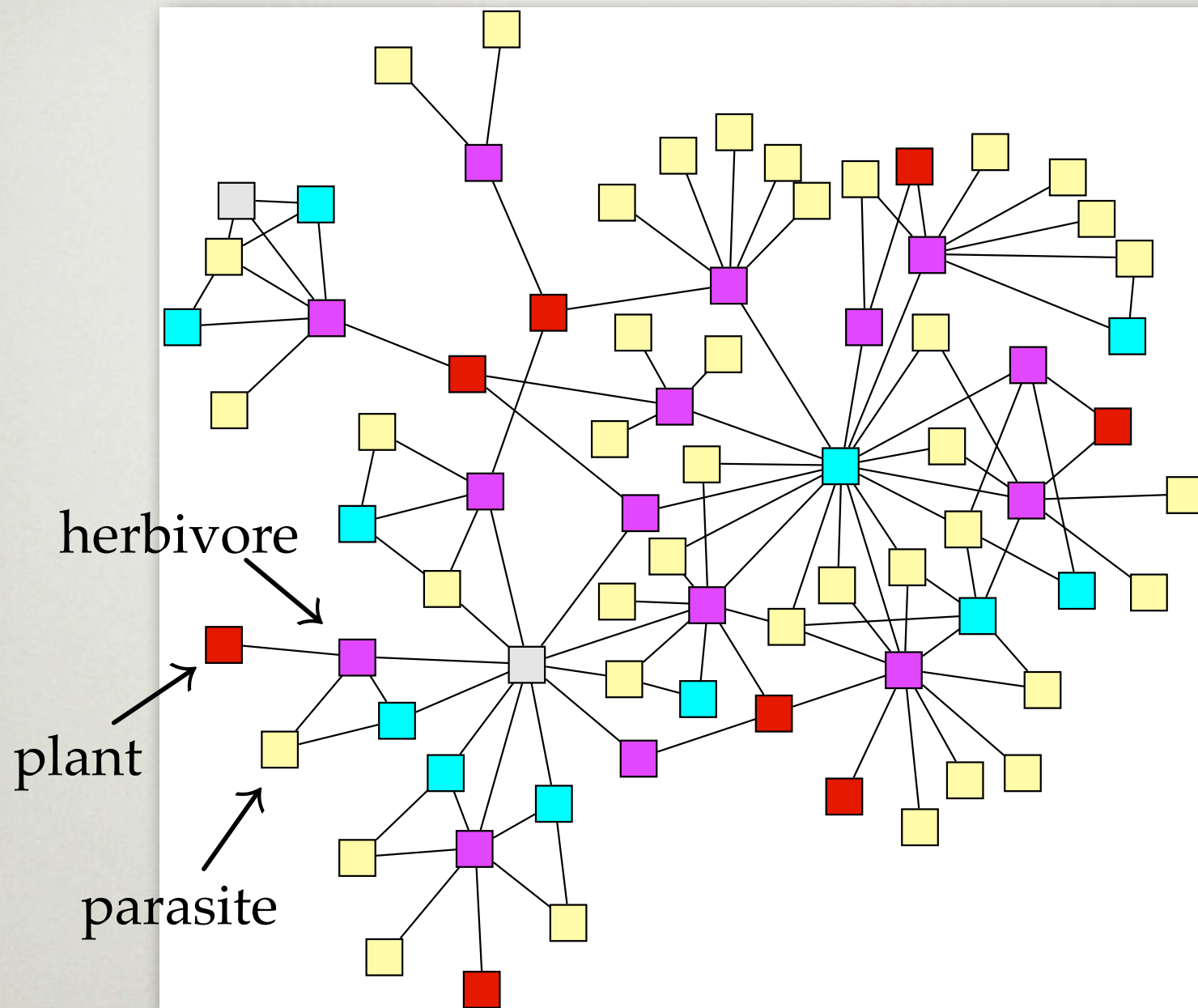


FROM GRAPH TO ENSEMBLE

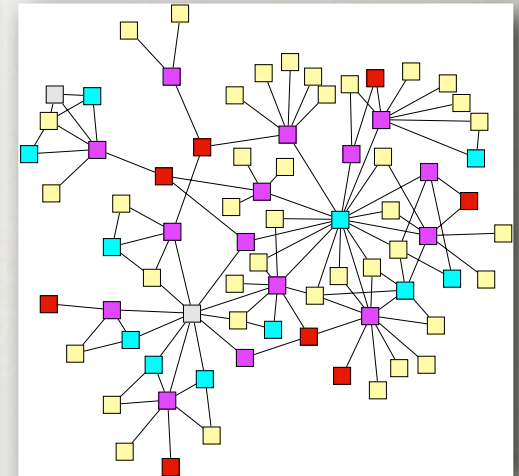
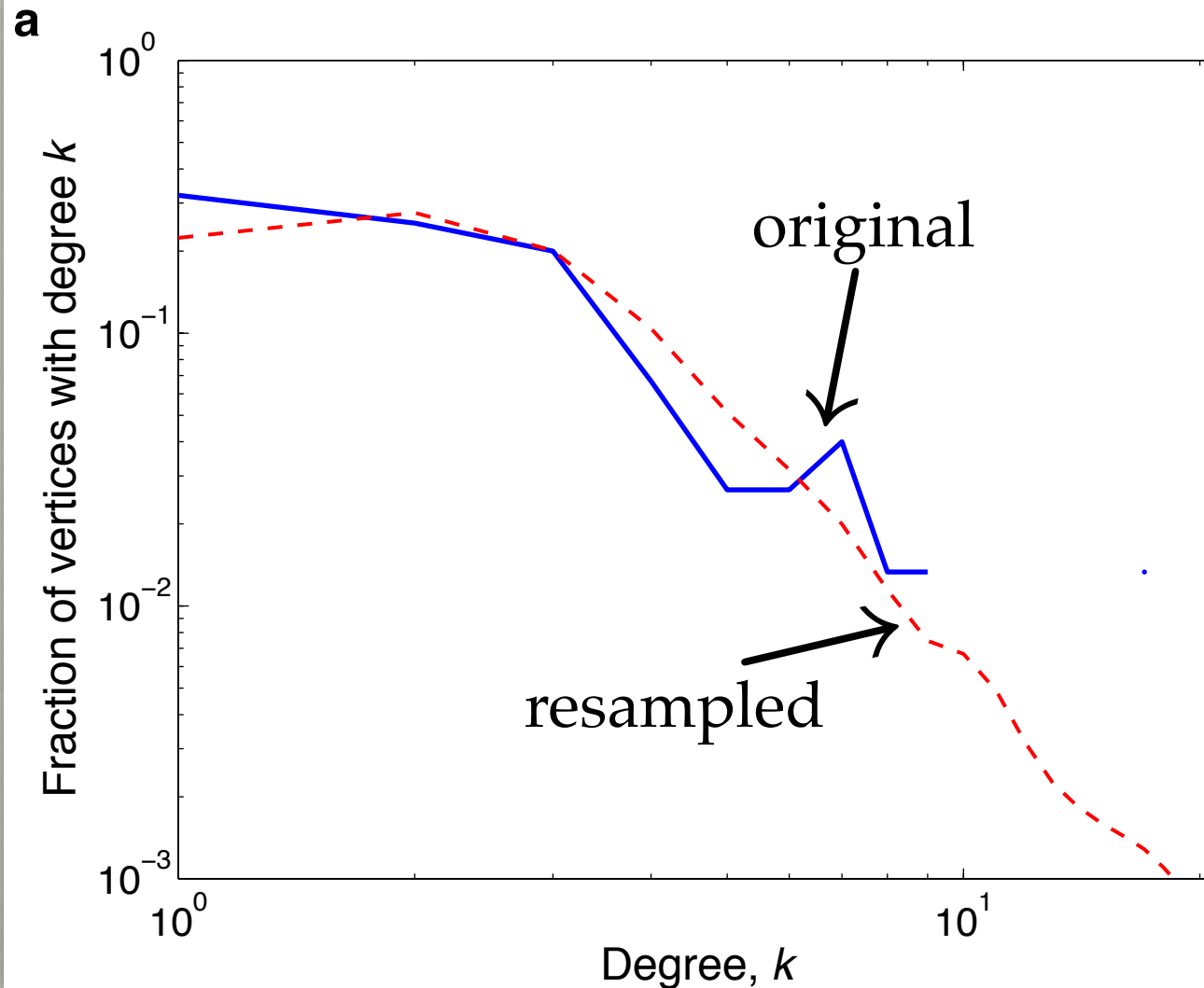
FROM GRAPH TO ENSEMBLE

- Given graph G
- run MCMC to equilibrium
- then, for each sampled $\mathcal{D}$, draw a **resampled** graph G' from ensemble

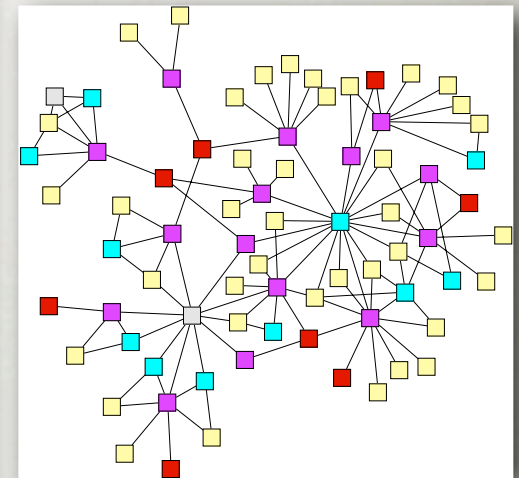
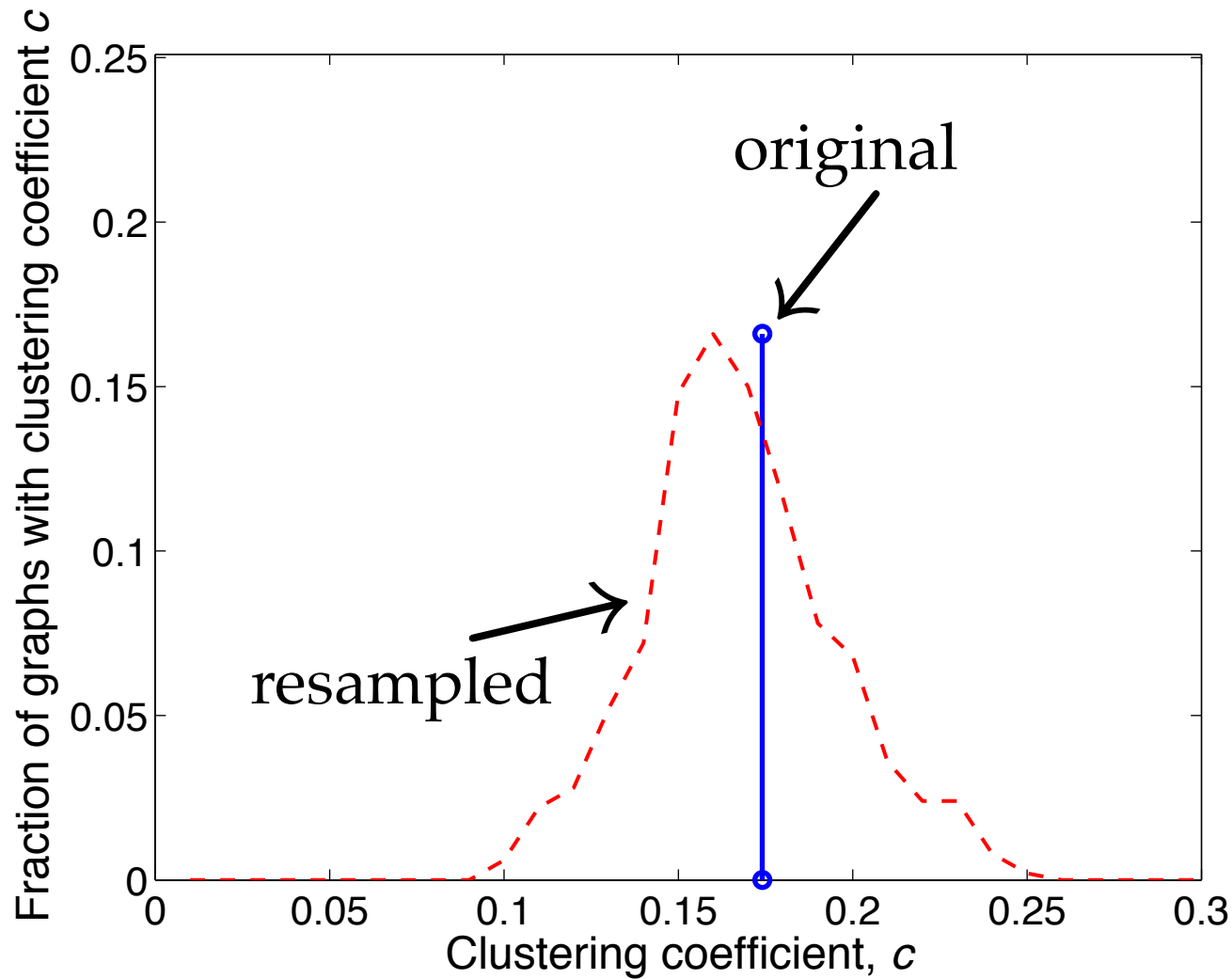
A test: do resampled graphs look like original?



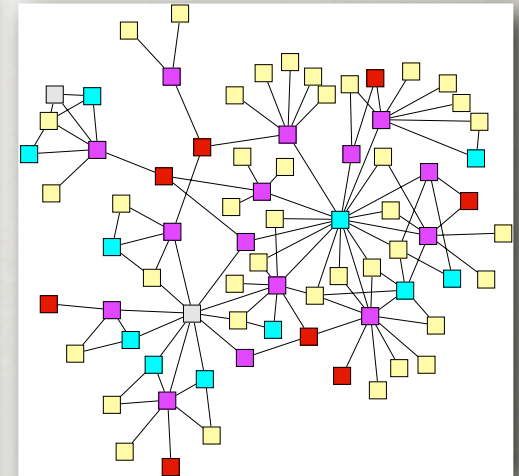
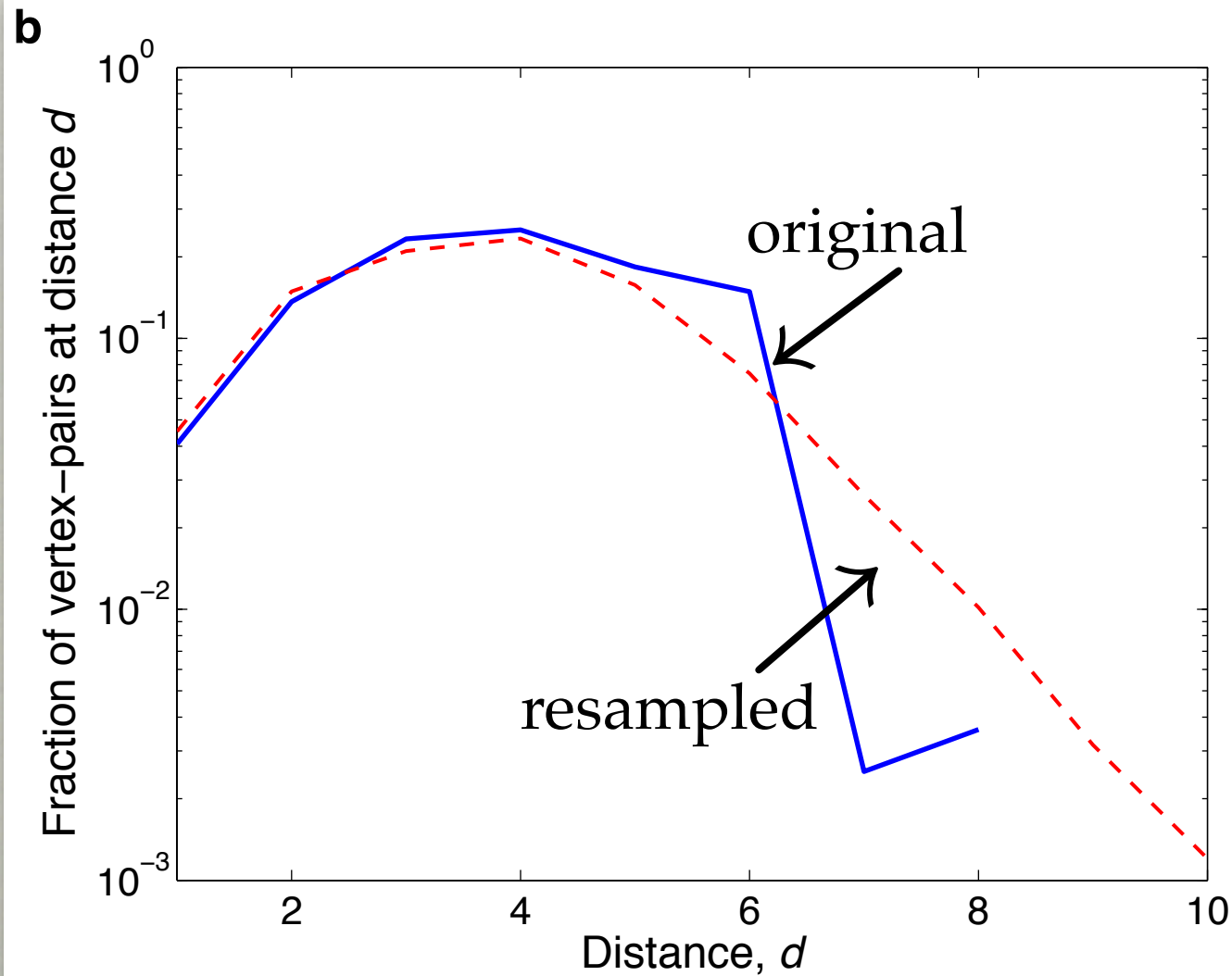
DEGREE DISTRIBUTION



CLUSTERING COEFFICIENT



DISTANCE DISTRIBUTION



MISSING LINKS

many networks partially known, noisy

- social nets, food webs, protein interactions, etc.

can hierarchies predict their **missing links**?

previous approaches

- Liben-Nowell & Kleinberg (2003)
- Goldberg & Roth (2003)
- Szilágyi et al. (2005)
- many more now

ACCURACY IS HARD

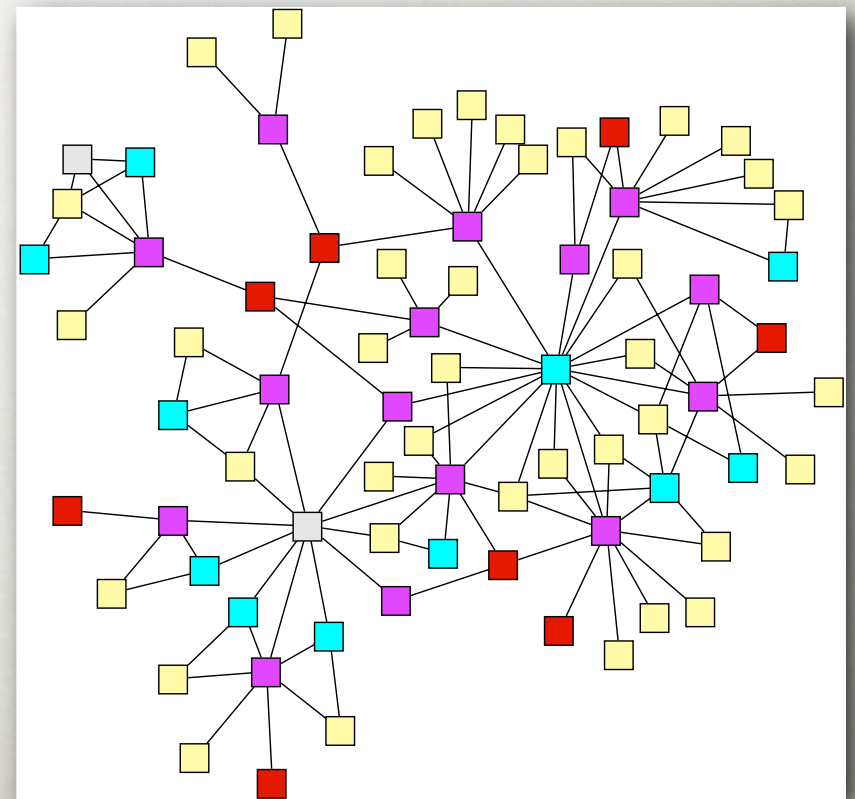
- remove k edges from G
- how easy to guess a missing link?

$$p_{\text{guess}} \approx \frac{k}{n^2 - m + k}$$
$$= O(n^{-2})$$

$$n = 75$$

$$m = 113$$

$$p_{\text{guess}} = k / (2662 + k)$$



AN HRG APPROACH

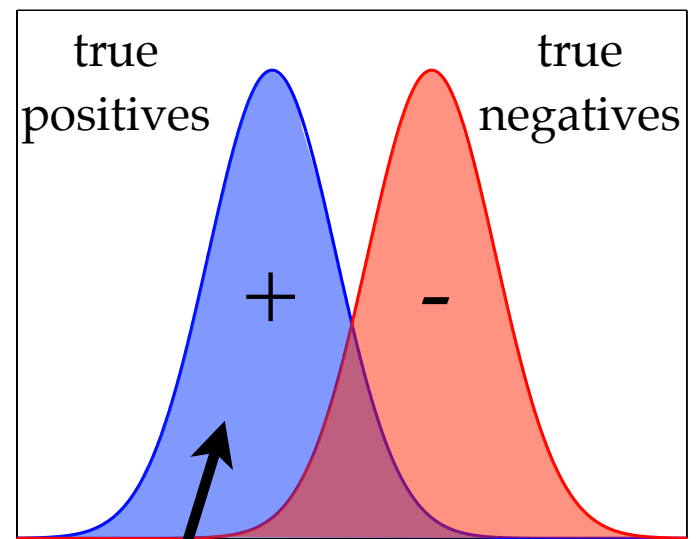
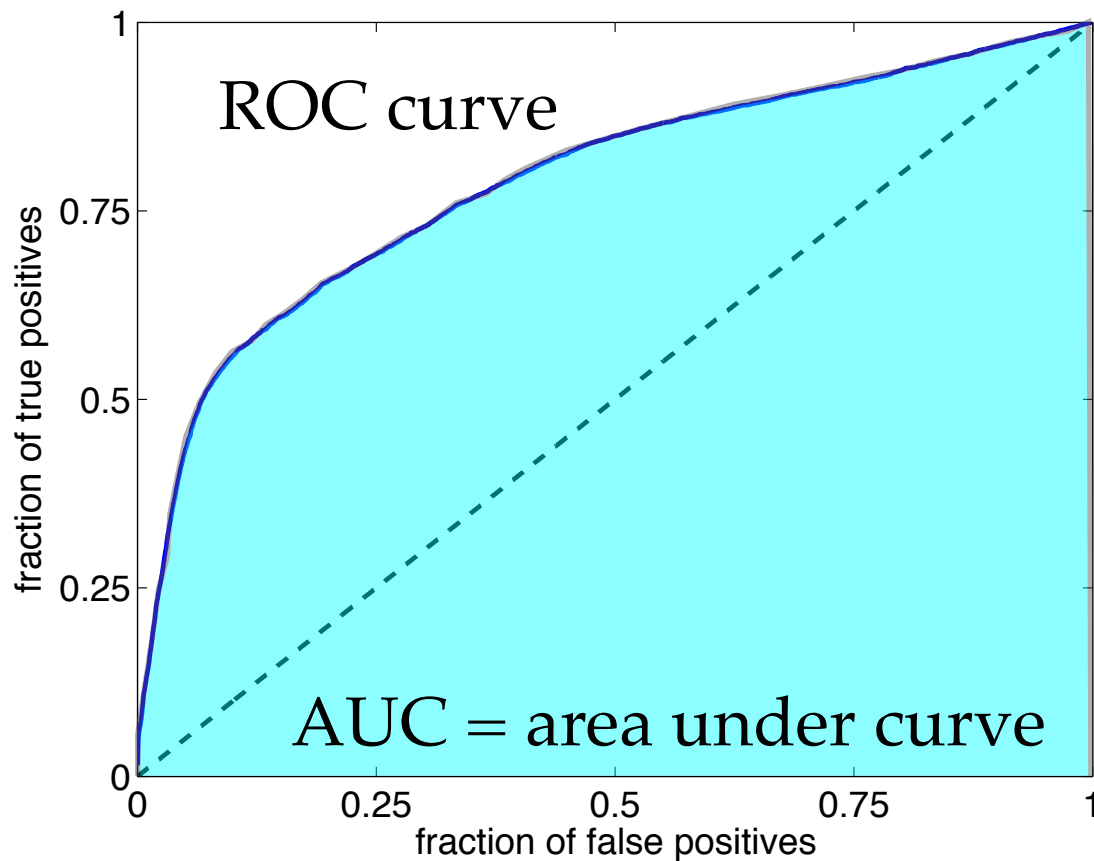
- Given incomplete graph G
- run MCMC to equilibrium
- then, over sampled $\mathcal{D}$, compute average $\langle p_r \rangle$ for links $(i, j) \notin G$
- predict links with high $\langle p_r \rangle$ values are missing

Test via leave- k -out cross-validation

perfect accuracy: $\text{AUC} = 1$

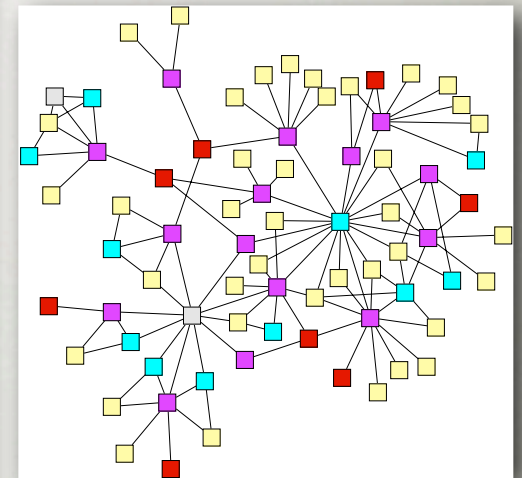
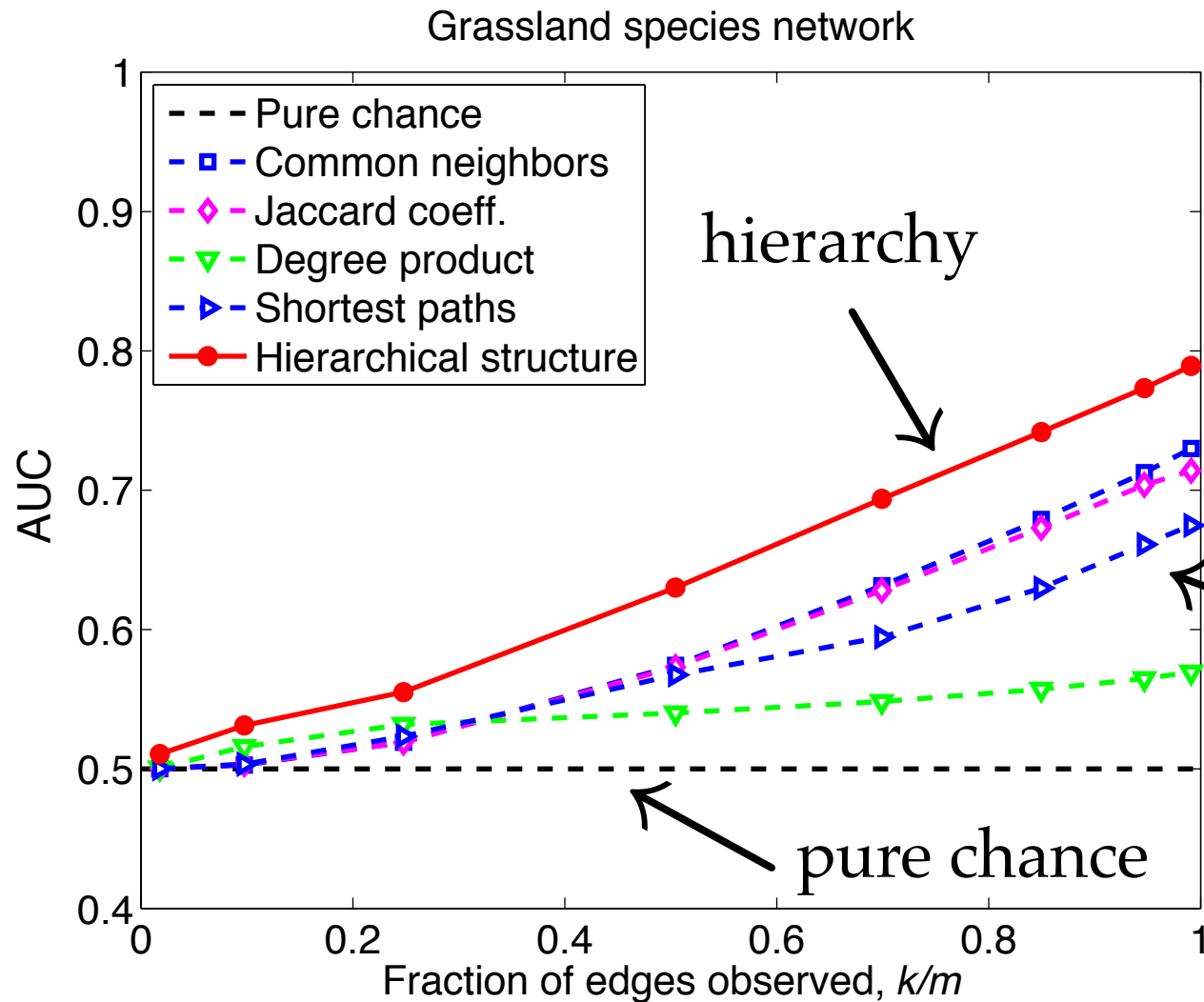
no better than chance: $\text{AUC} = 1/2$

SCORING THE PREDICTIONS



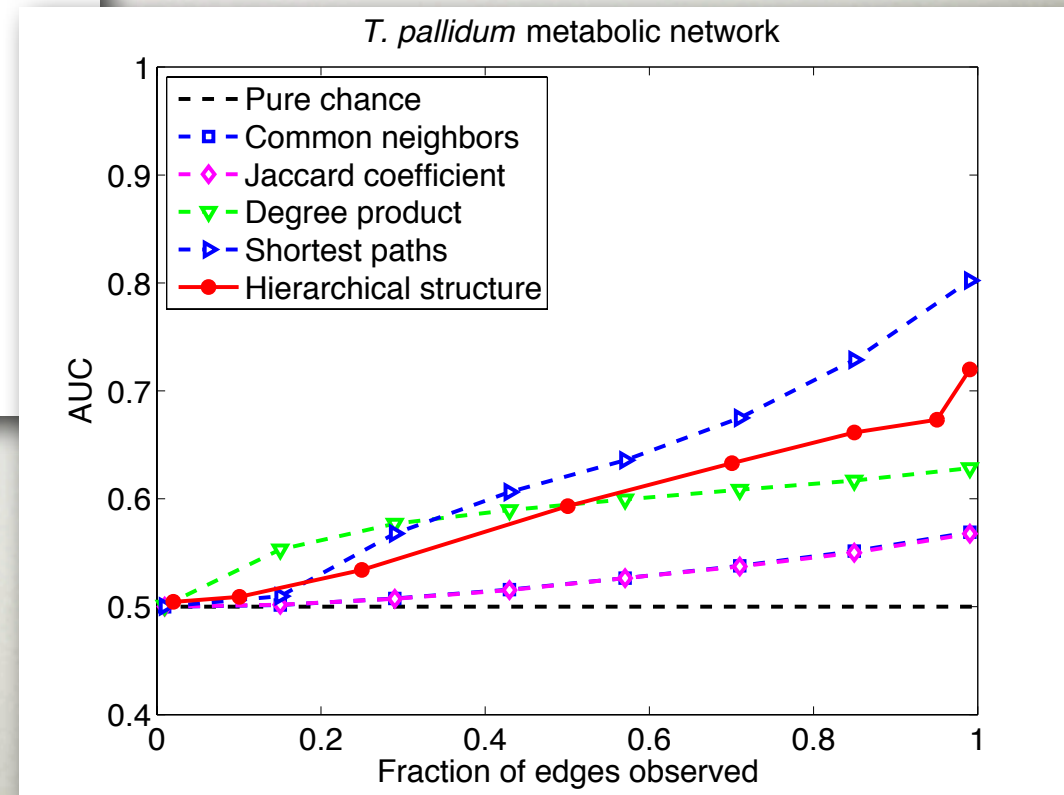
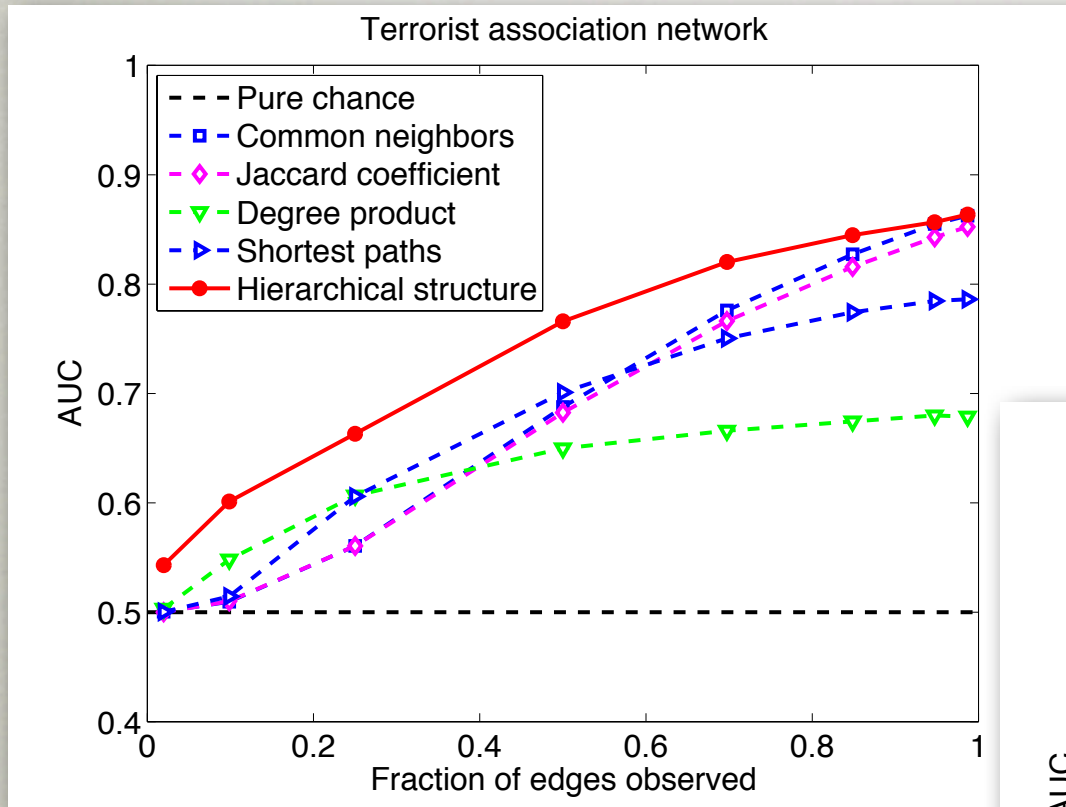
AUC =
Pr(distinguish
+ from -)

PERFORMANCE 1



simple predictors

PERFORMANCE 2



SOME FINAL THOUGHTS

- what processes create these hierarchical structures?
- scaling up the running time from $O(n^2)$?
- active learning
- generalization to weighted, directed edges
- generalization to non-Poisson distributions