Hierarchical Block Models

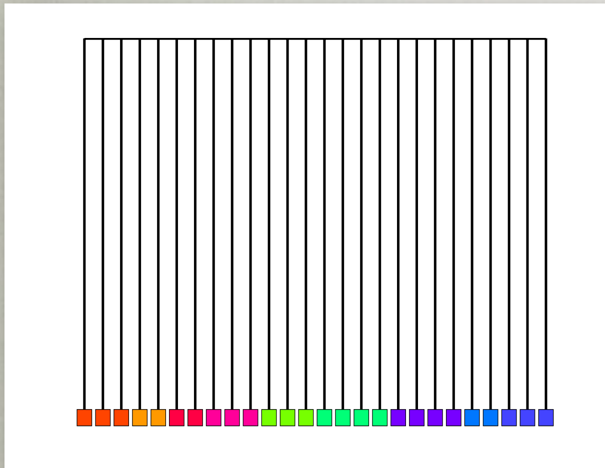
Lecture 7, Fall 2014
CSCI 5352, Network Analysis and Models

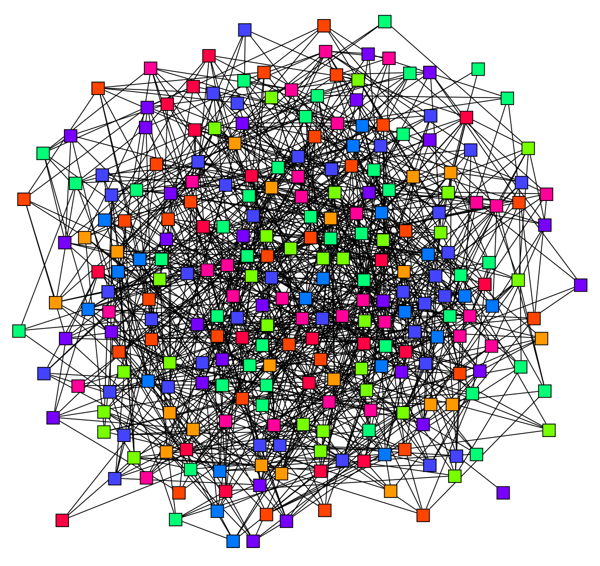
Prof. Aaron Clauset
University of Colorado, Boulder

Advanced Generative Models

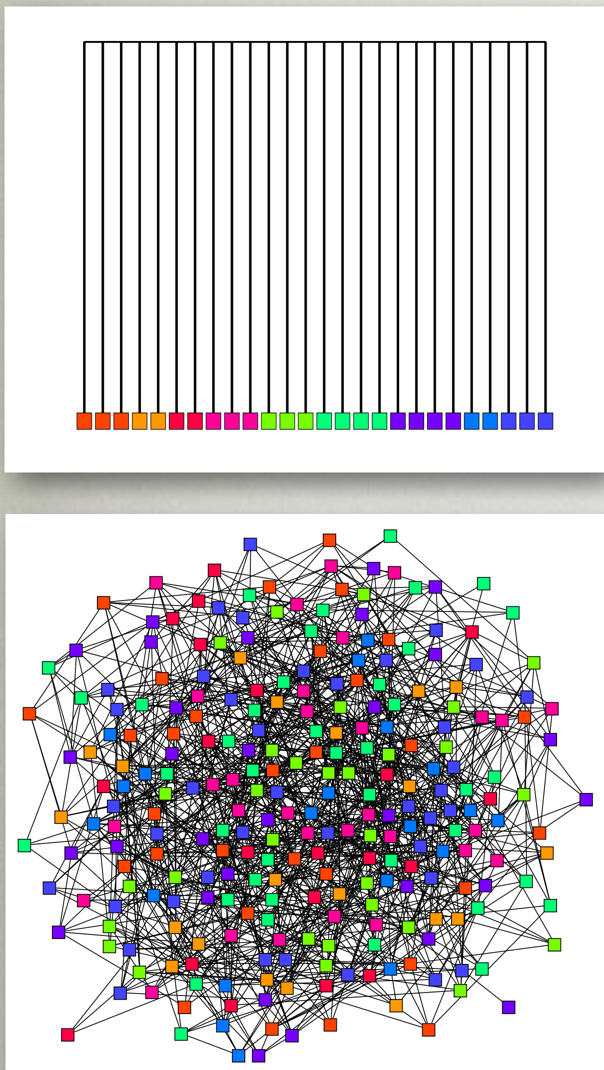
1. hierarchical random graph model example
2. examples of tasks generative models can do
3. application to some real-world data sets

no structure

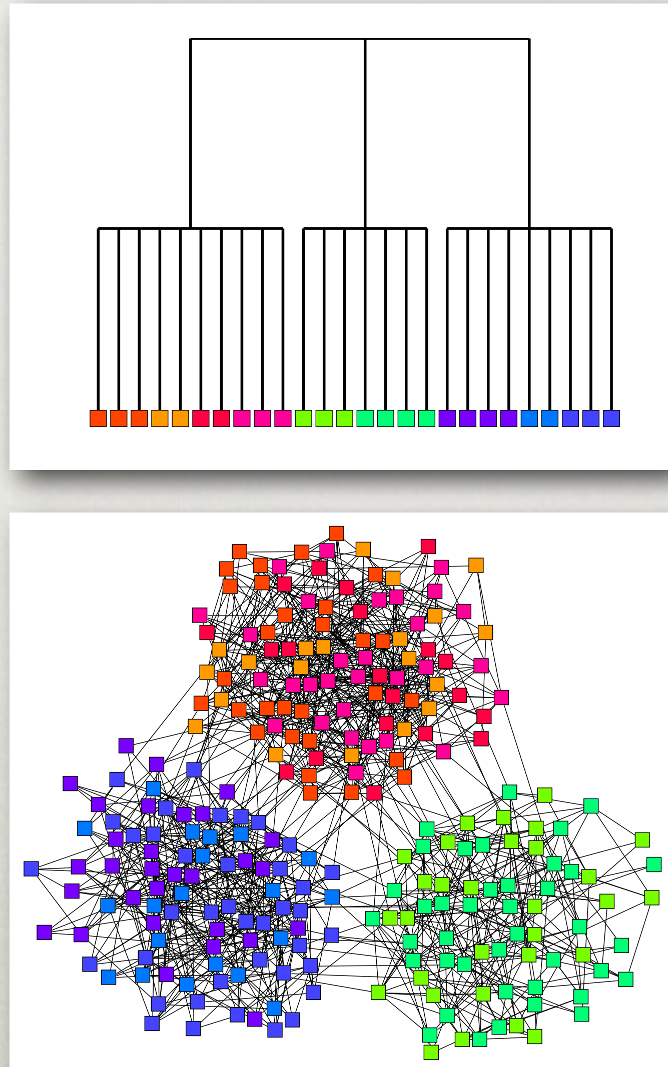




no structure

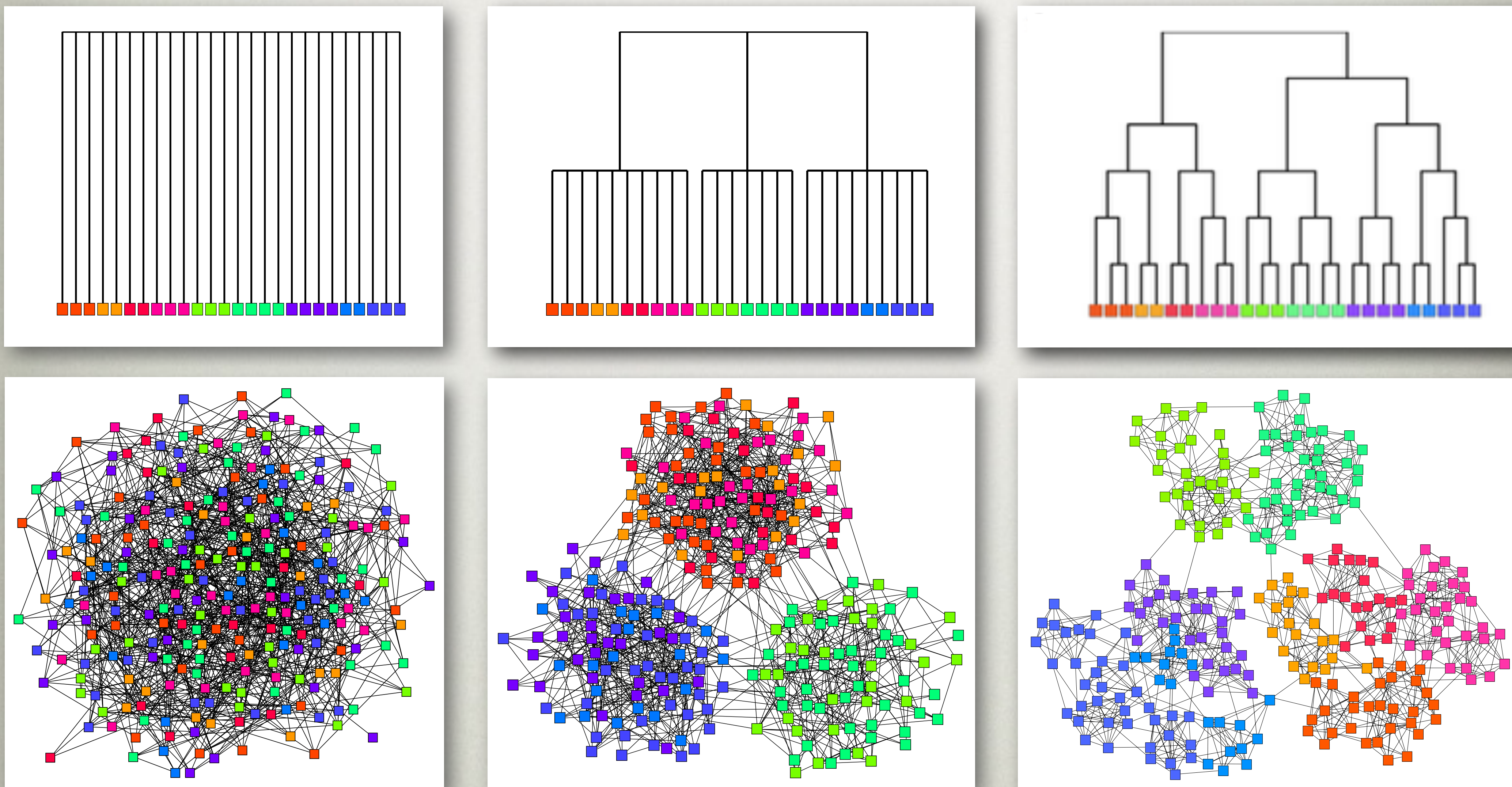


block structure



one scale

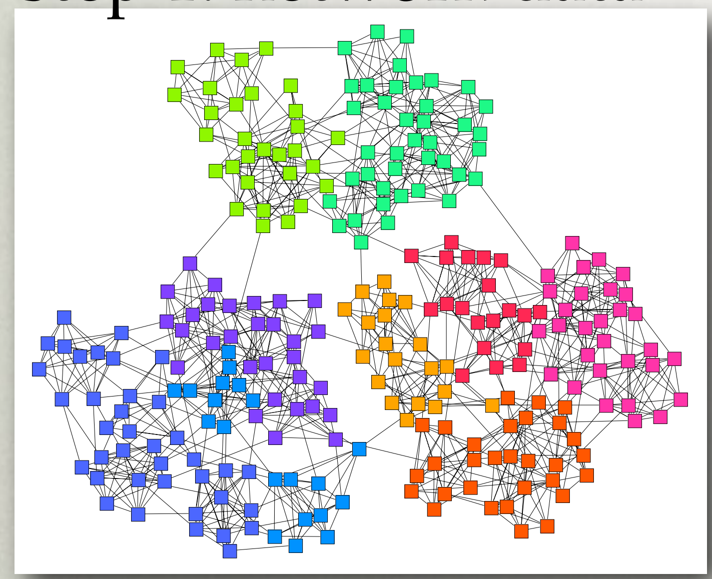
no structure **block structure** **hierarchical structure**



one scale **multi-scale**

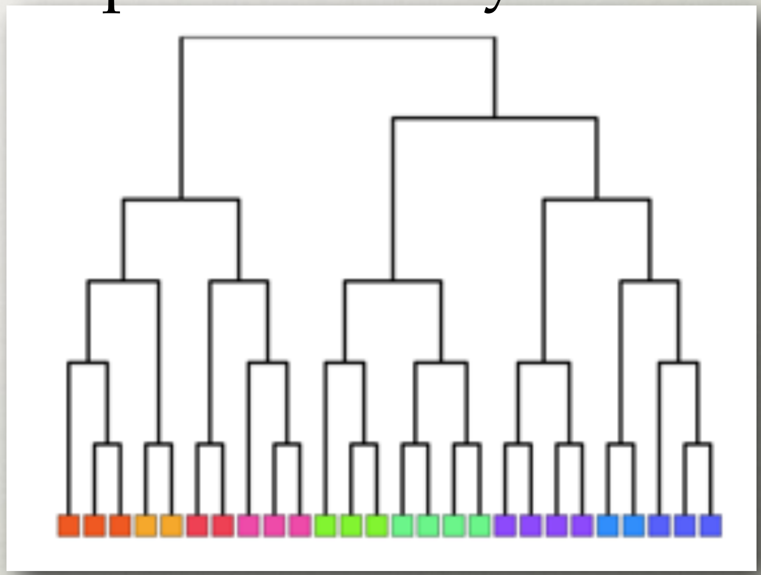
how can we extract a network's hierarchy?

step 1: network data



?

step 3: hierarchy



GENERATIVE MODELS

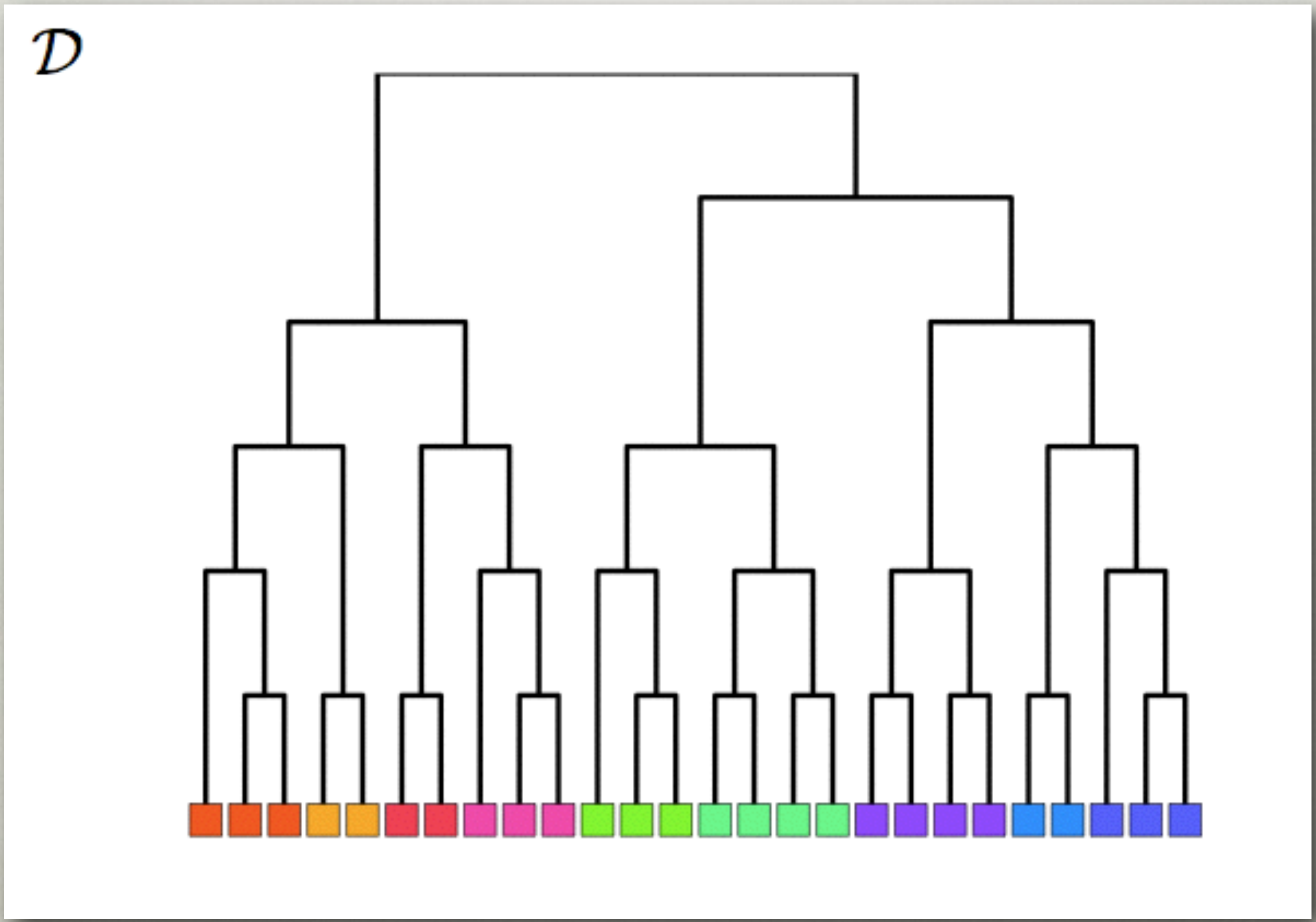
1. write down hierarchical stochastic block model

2. estimate / learn model from data

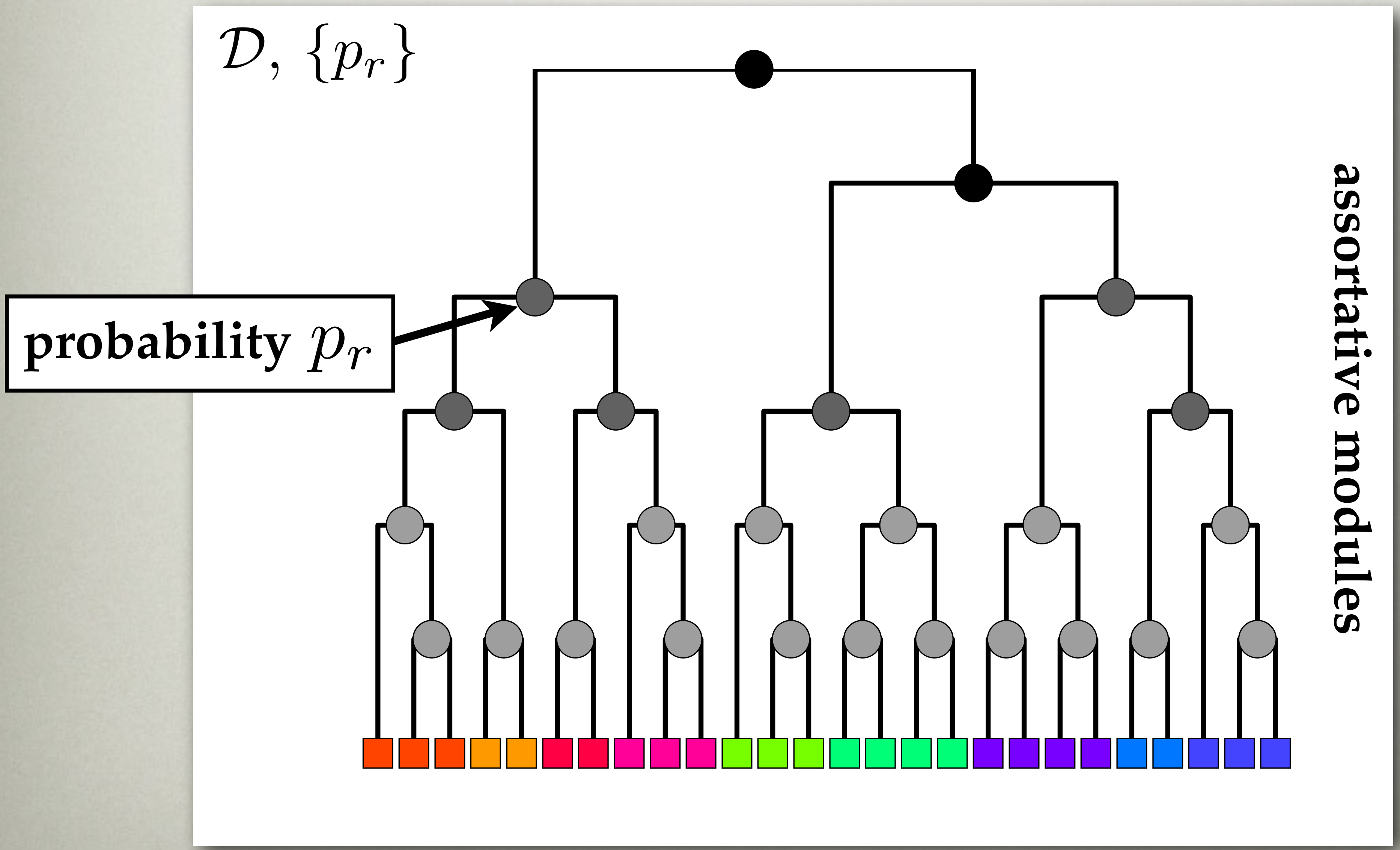
3. evaluate model goodness-of-fit

4. evaluate model predictions

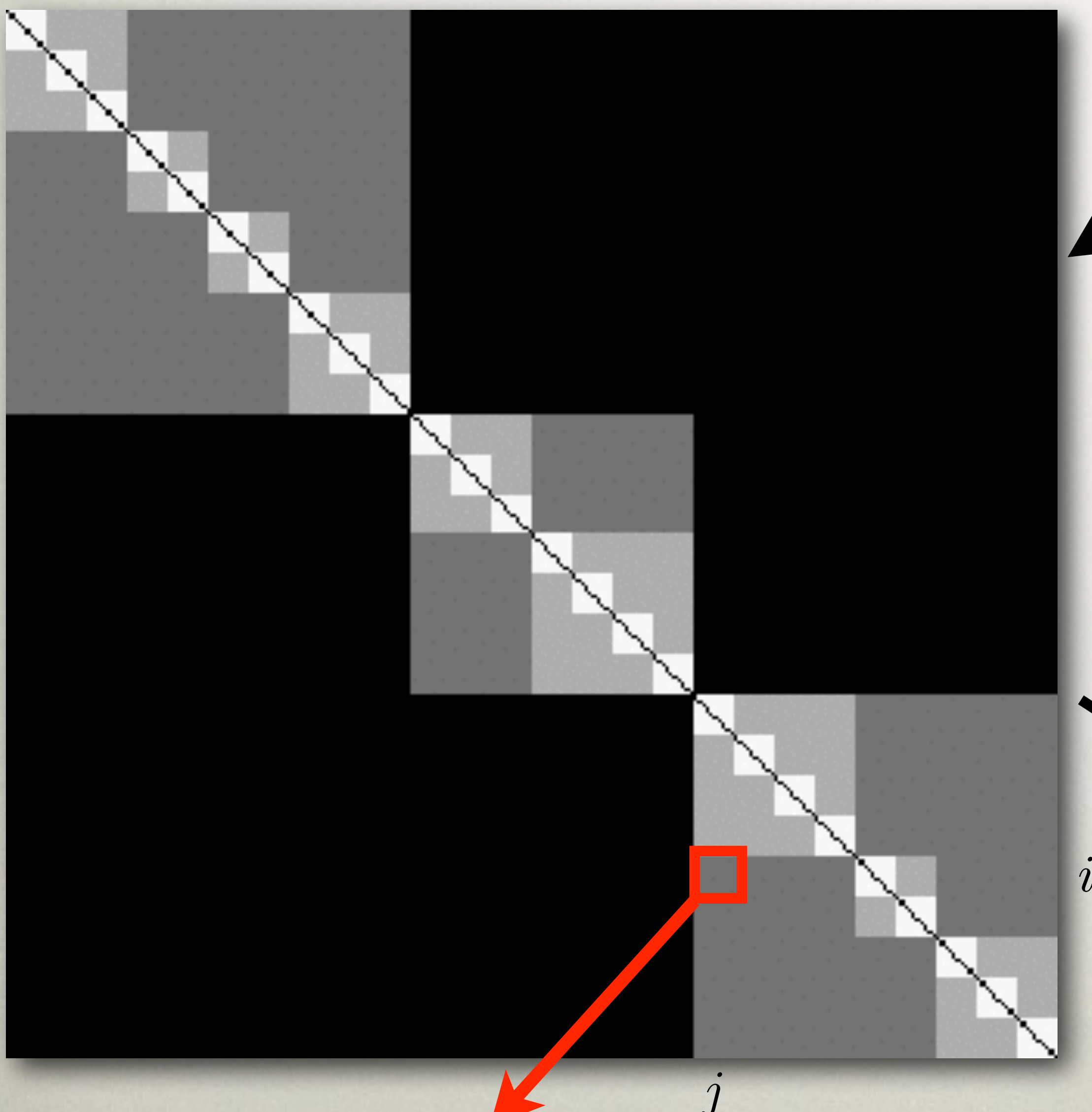
A Model of Hierarchy



A Model of Hierarchy

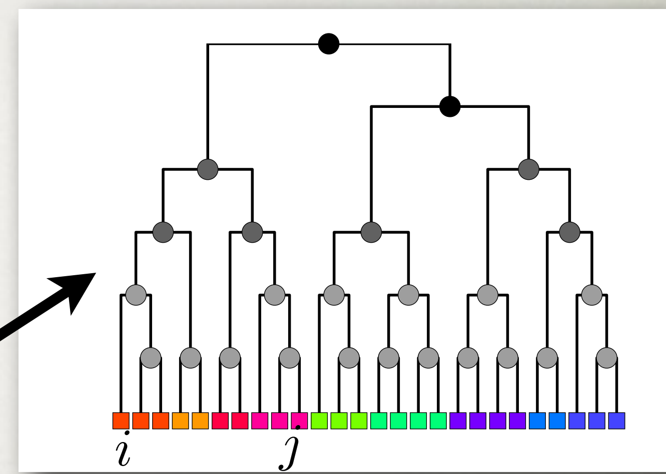


“inhomogeneous” random graph

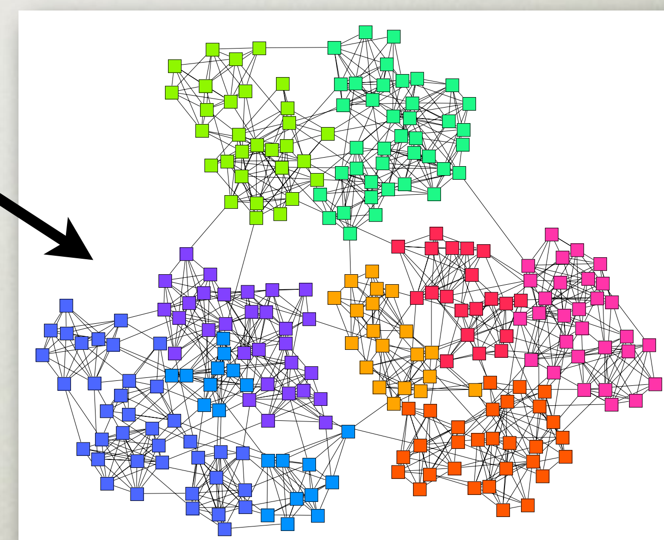


$$\Pr(i, j \text{ connected}) = p_r$$
$$= p_{(\text{lowest common ancestor of } i,j)}$$

model



instance



Hierarchical Random Graph

advantages

- explicit hierarchical structure
- flexible (2n parameters)
- captures structure at all scales
- captures mixtures of assortativity, disassortativity
- like SBM, decomposes adjacencies into "bundles" (each a random bipartite graph)
- learnable directly from data
- nice interpretable structure

disadvantages

- flexible (2n parameters)
- computationally slow
- can overfit to degree structure (like SBM)

Fitting the model

- **likelihood function** $\mathcal{L} = \Pr(\text{ data } | \text{ model })$

  ($\mathcal{L}$ scores **quality** of model)

- **sample all good models**

  via Markov chain Monte Carlo*
  over all dendrograms

- **technical details in**

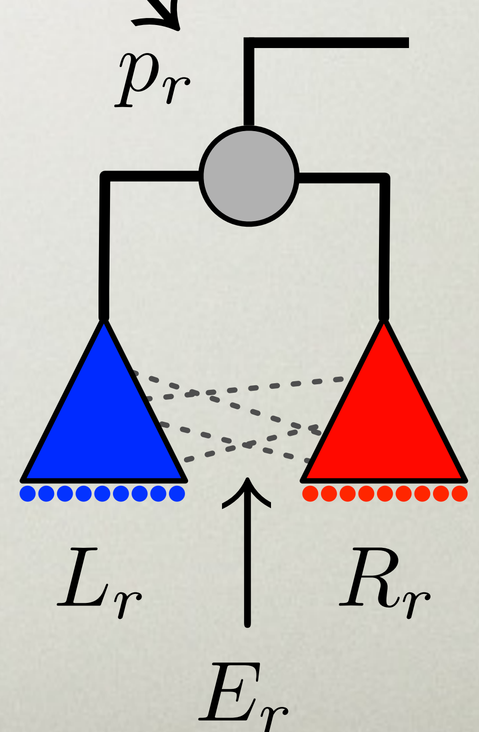
  Clauset, Moore and Newman, *Nature* **453**, 98-101 (2008) and

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

* other sampling or optimization methods possible

Likelihood Function

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

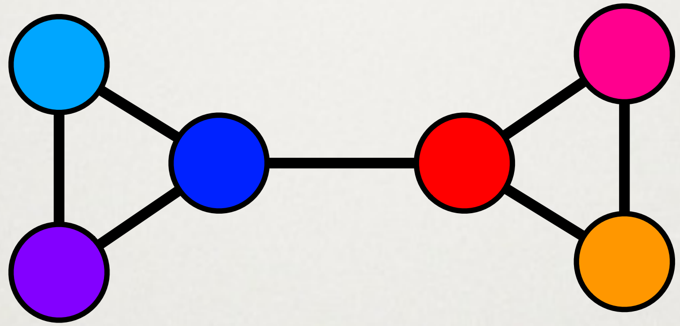


L_r = number nodes in left subtree

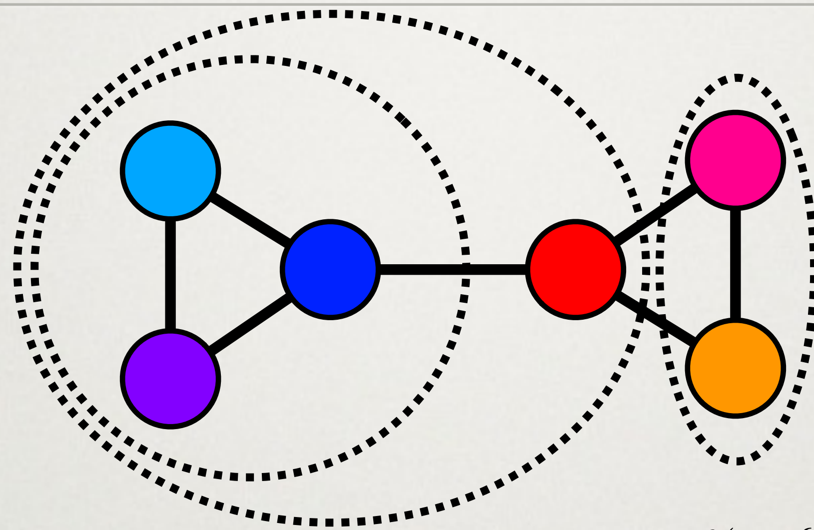
R_r = number nodes in right subtree

E_r = number edges with r as lowest common ancestor

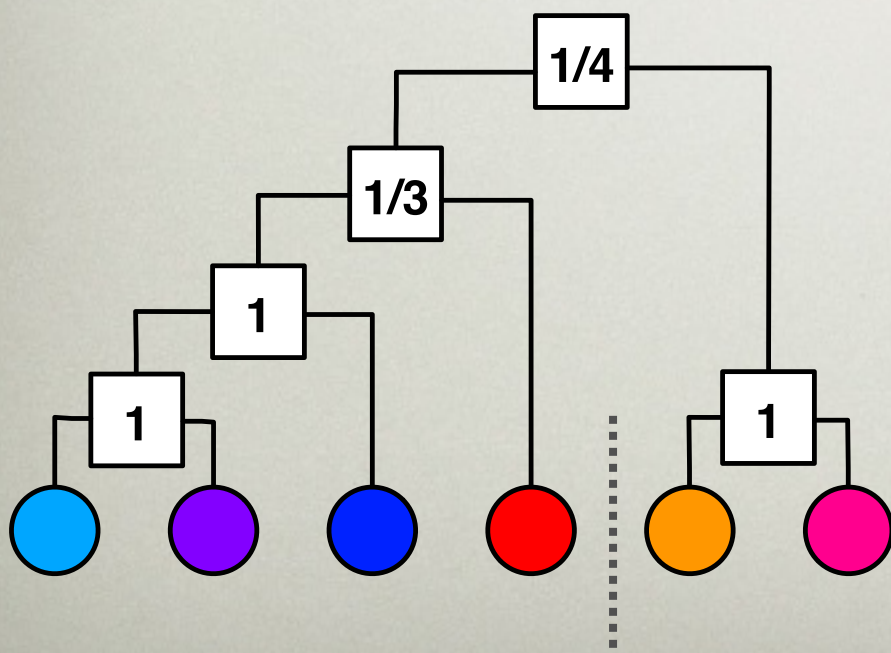
Example



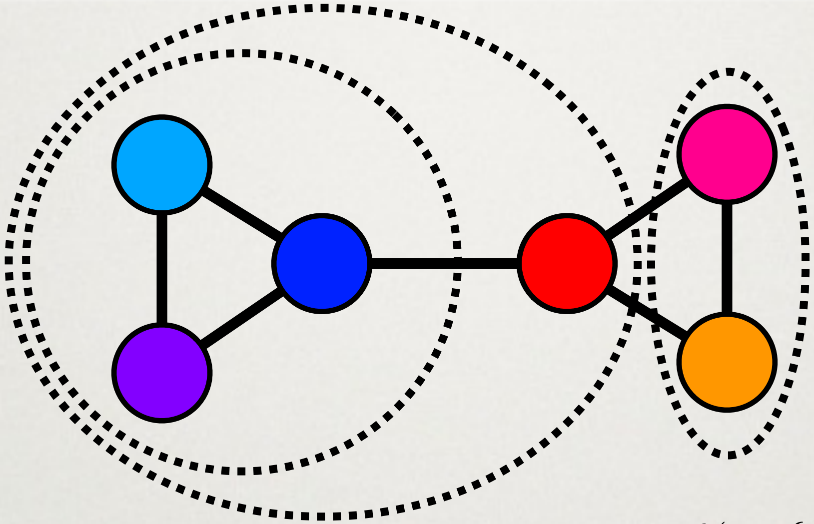
Bad Dendrogram



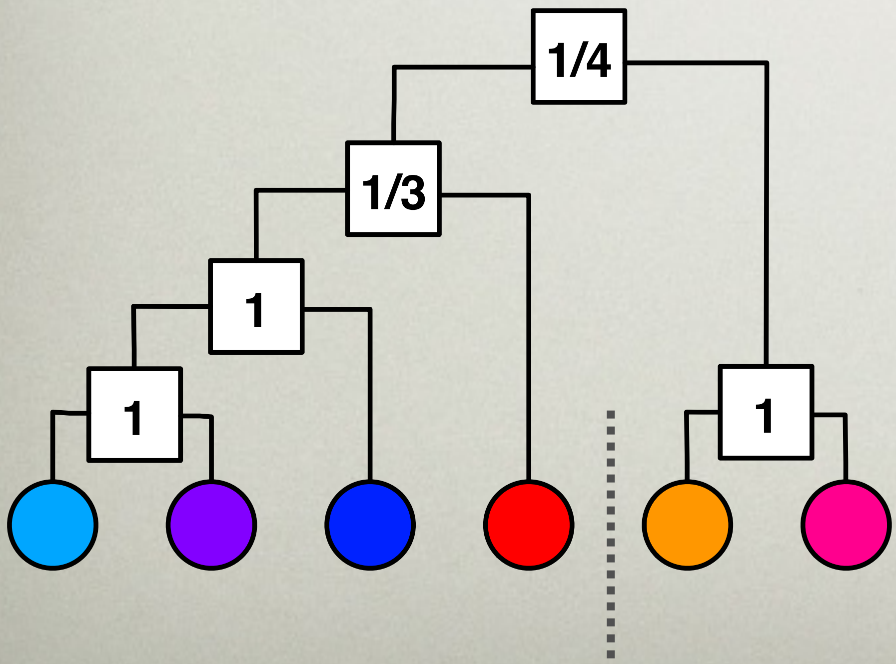
$$\mathcal{L}(\mathcal{D},\, \{p_r\}) = \prod_r p_r^{E_r}\, (1-p_r)^{L_r R_r - E_r}$$



Bad Dendrogram



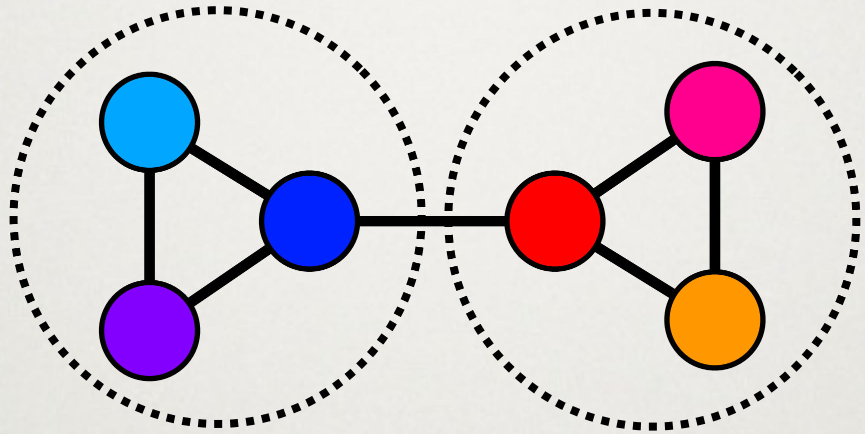
$$\mathcal{L}(\mathcal{D},\,\{p_r\}) = \prod_r p_r^{E_r}\,(1-p_r)^{L_r R_r - E_r}$$



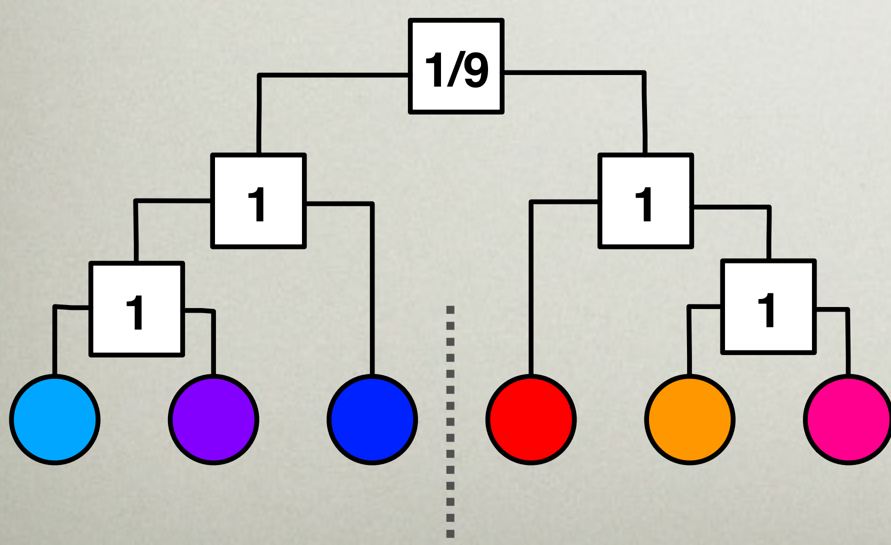
$$\mathcal{L} = \left[\left(\frac{1}{3}\right)^1\left(\frac{2}{3}\right)^2\right]\cdot\left[\left(\frac{1}{4}\right)^2\left(\frac{3}{4}\right)^6\right]$$

$$\mathcal{L} = 0.0016$$

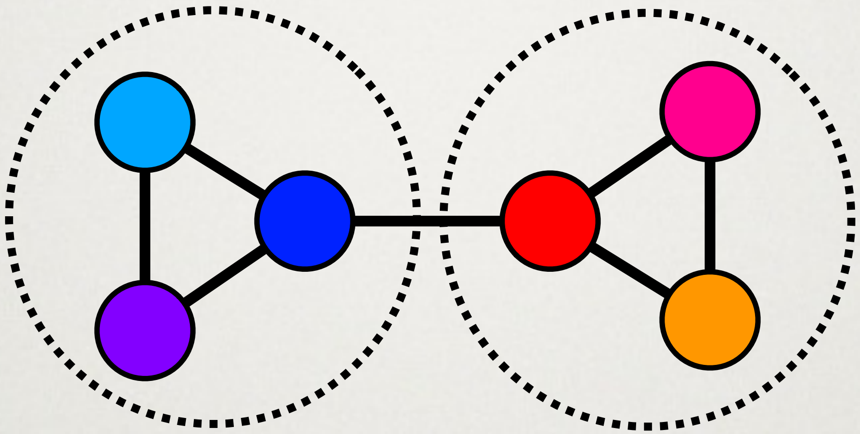
Good Dendrogram



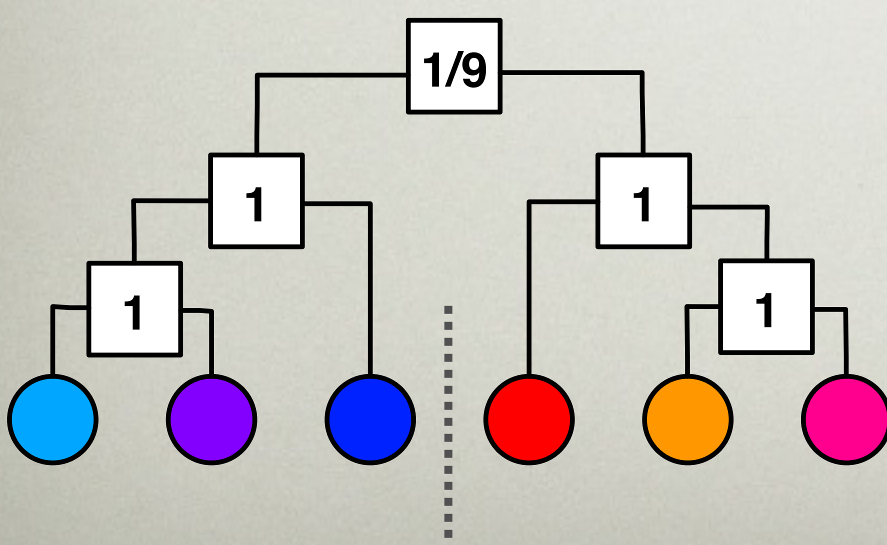
$$\mathcal{L}(\mathcal{D},\, \{p_r\}) = \prod_r p_r^{E_r}\, (1 - p_r)^{L_r R_r - E_r}$$



Good Dendrogram



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



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

$$\mathcal{L} = 0.0433$$

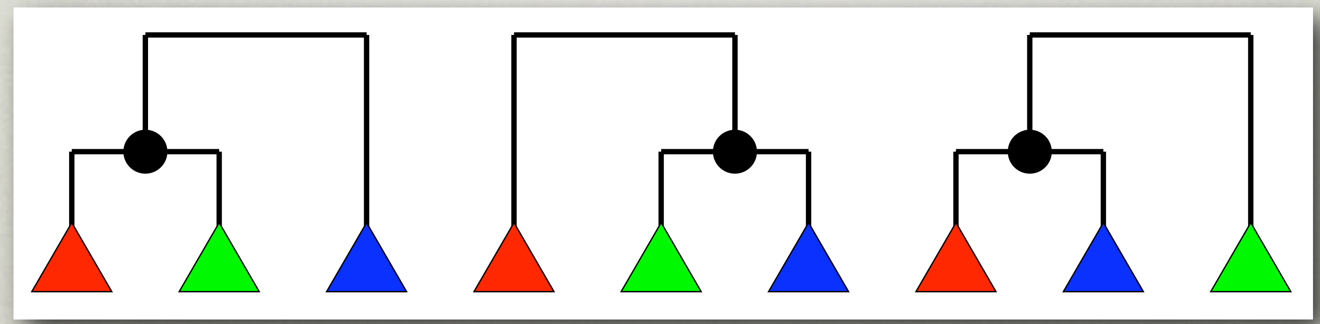
Markov chain Monte Carlo (MCMC)

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

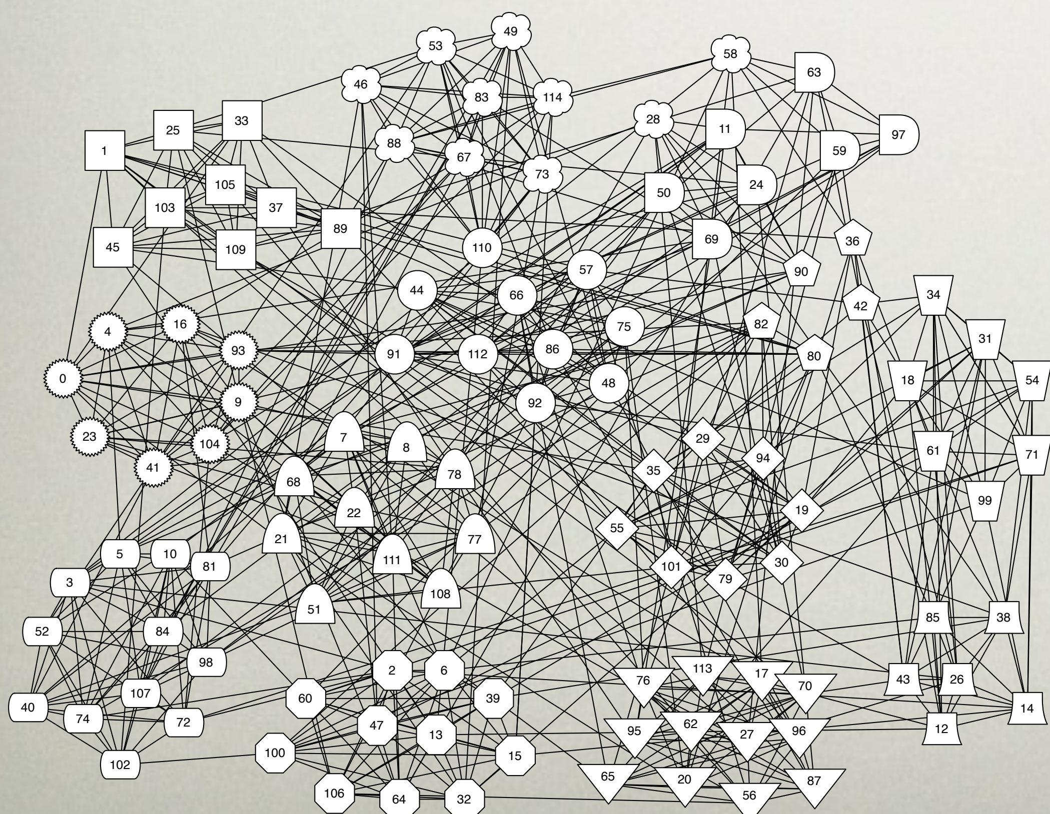
(up to relabeling)

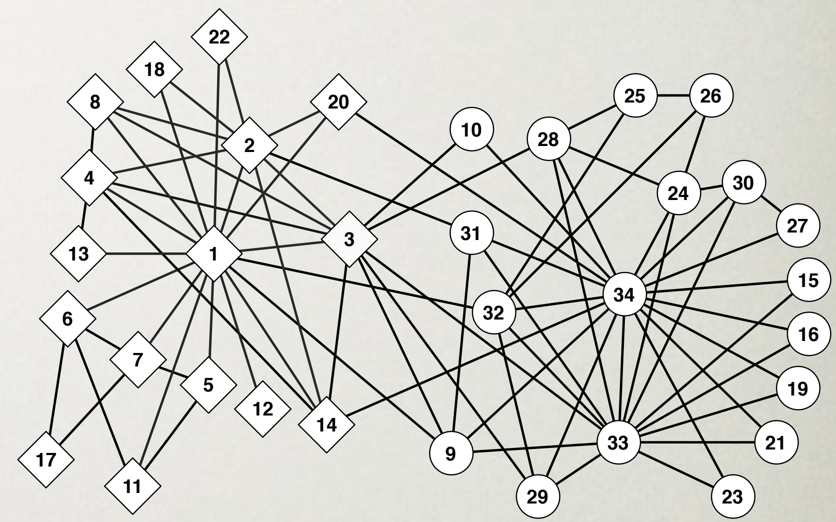
Some applications

Two Case Studies

NCAA Schedule 2000

$n = 115 \quad m = 613$





Zachary's Karate Club

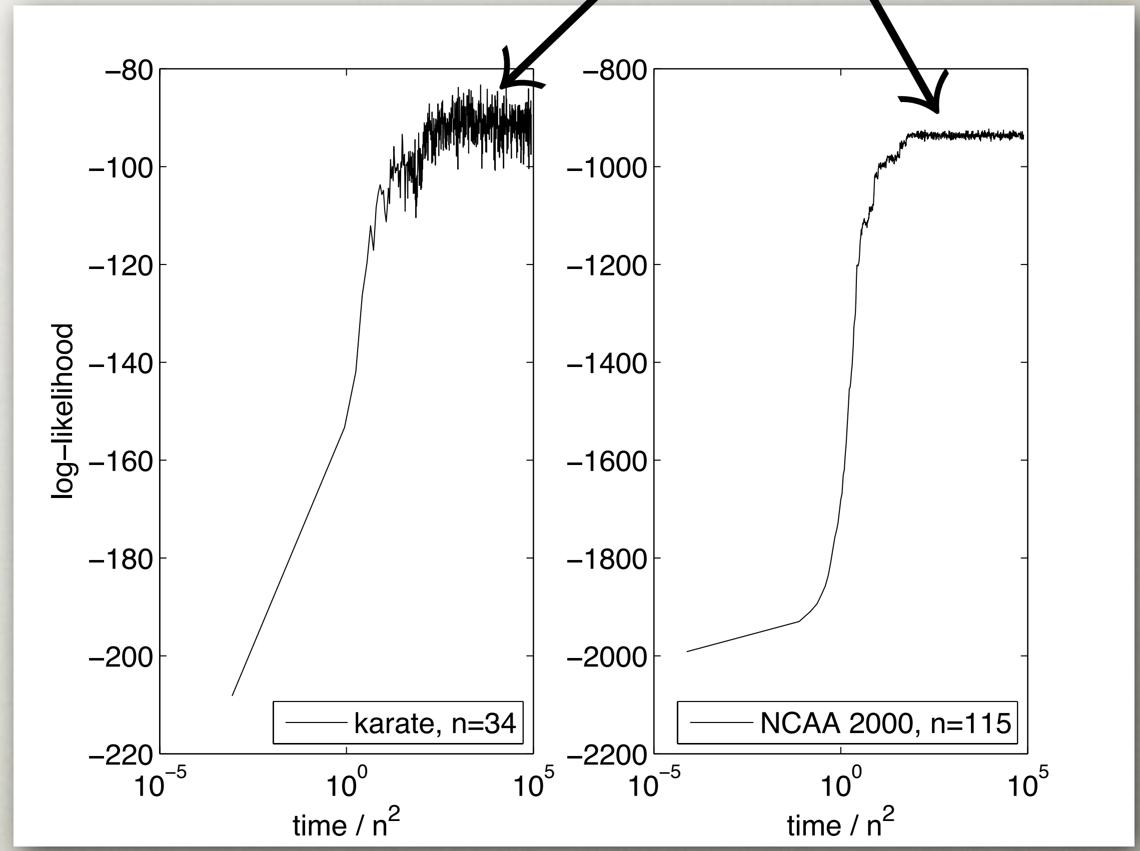
$n = 34 \quad m = 78$

Mixing Times

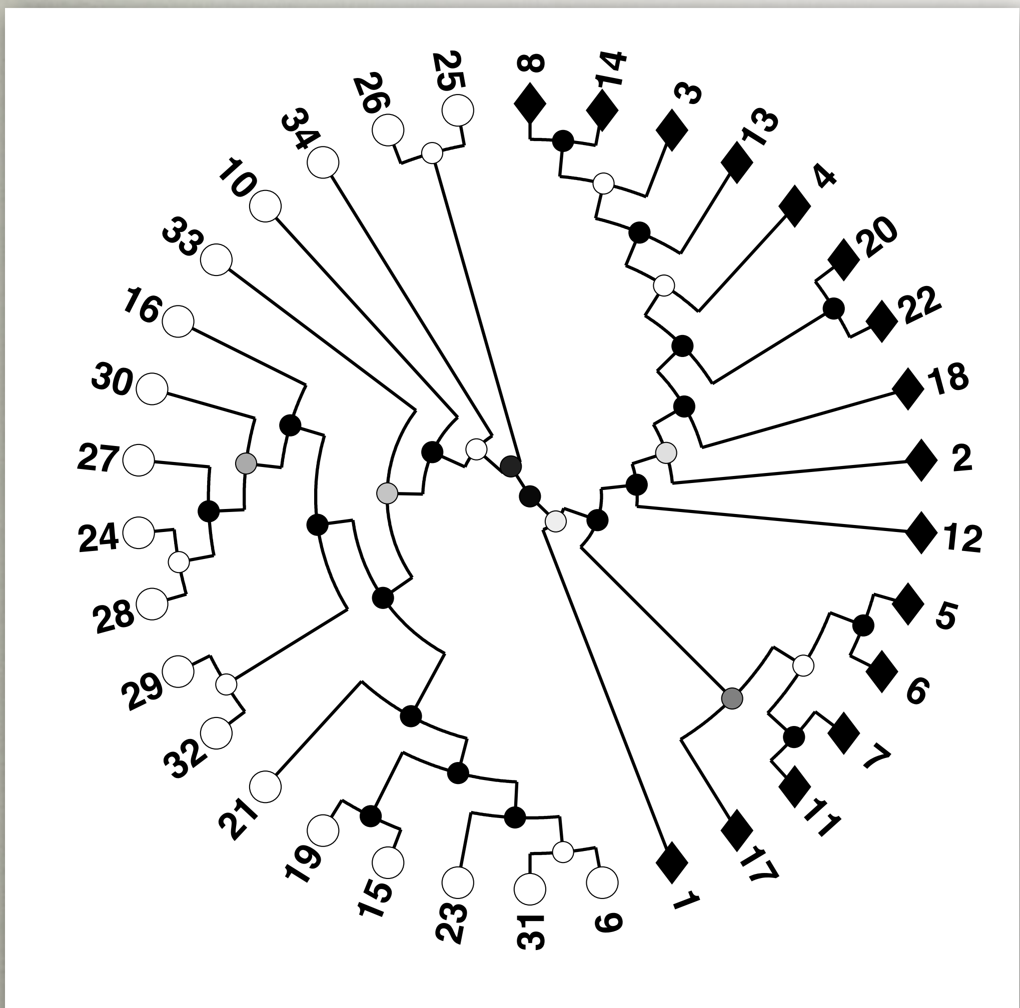
MCMC mixes relatively quickly

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

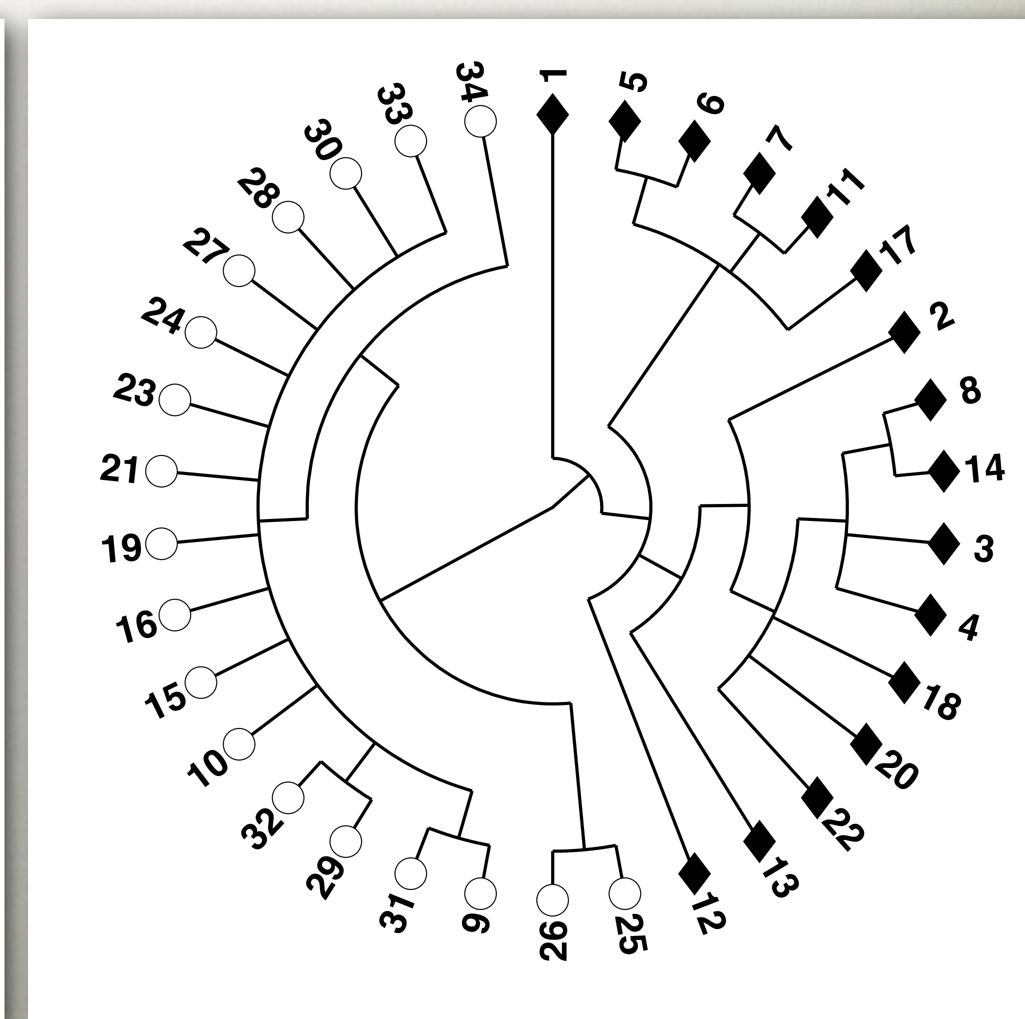
equilibrium



Hierarchies

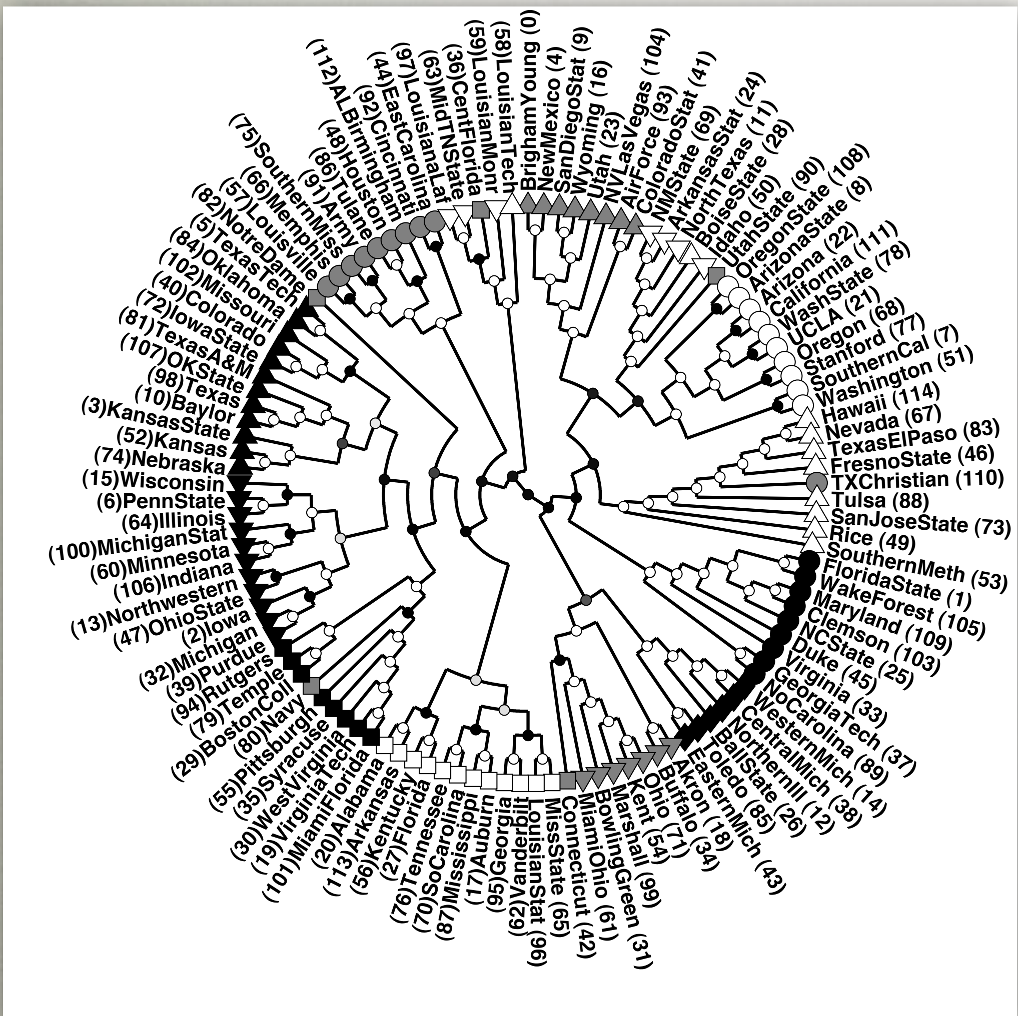


point estimate

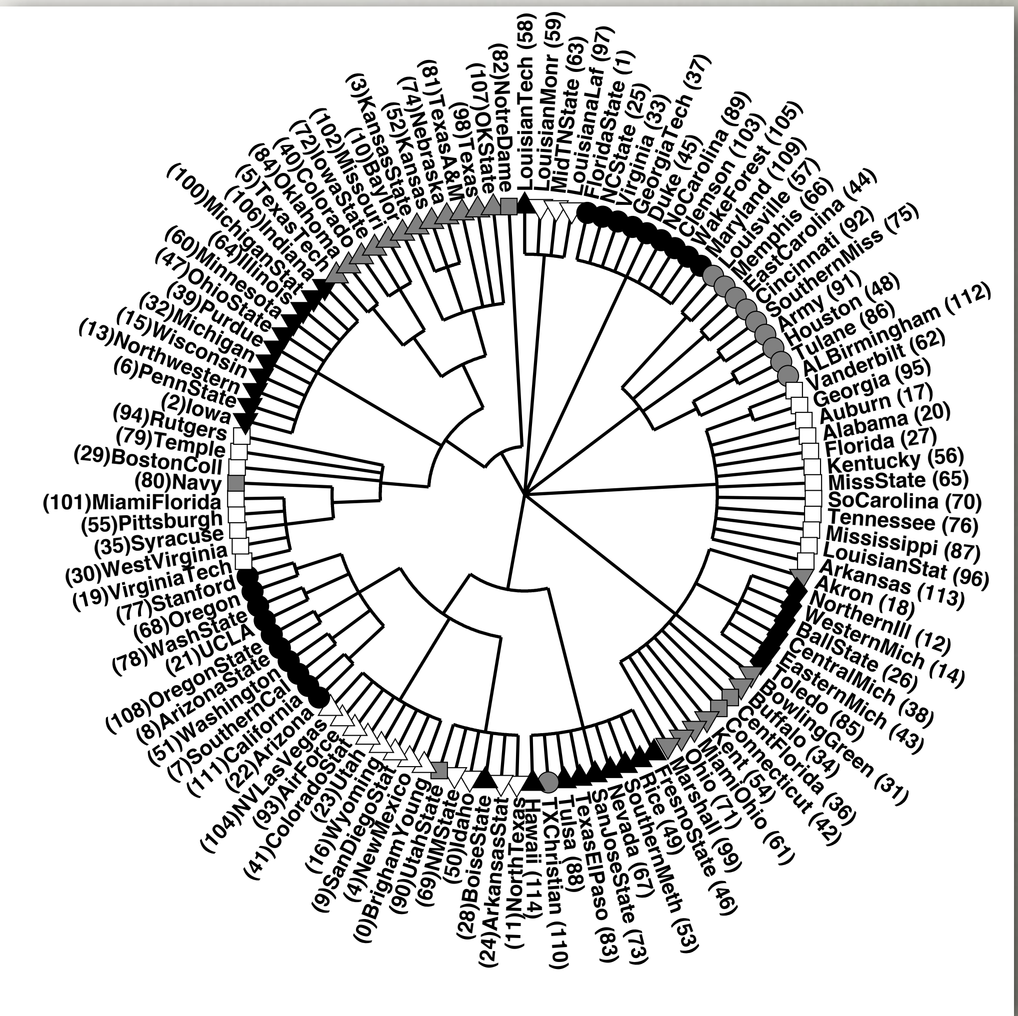


consensus hierarchy

Hierarchies



point estimate



consensus hierarchy

Edge Annotations

Average likelihood of edge existing

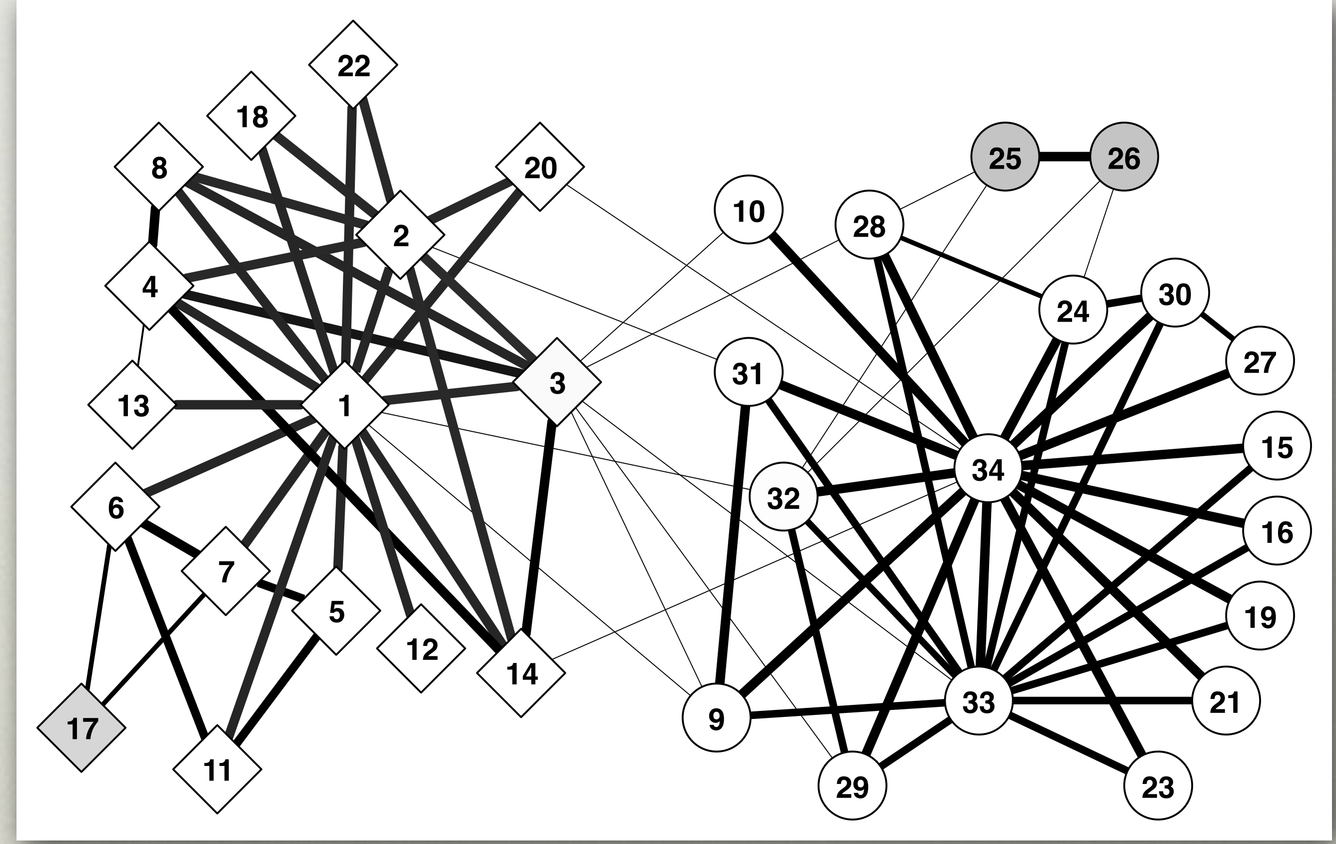
- For each edge (i, j) in *G*, compute average associated parameter $\langle \theta_r \rangle_{(i,j)}$ over sampled models
- $\langle \theta_r \rangle_{(i,j)}$ is edge annotation (weight)

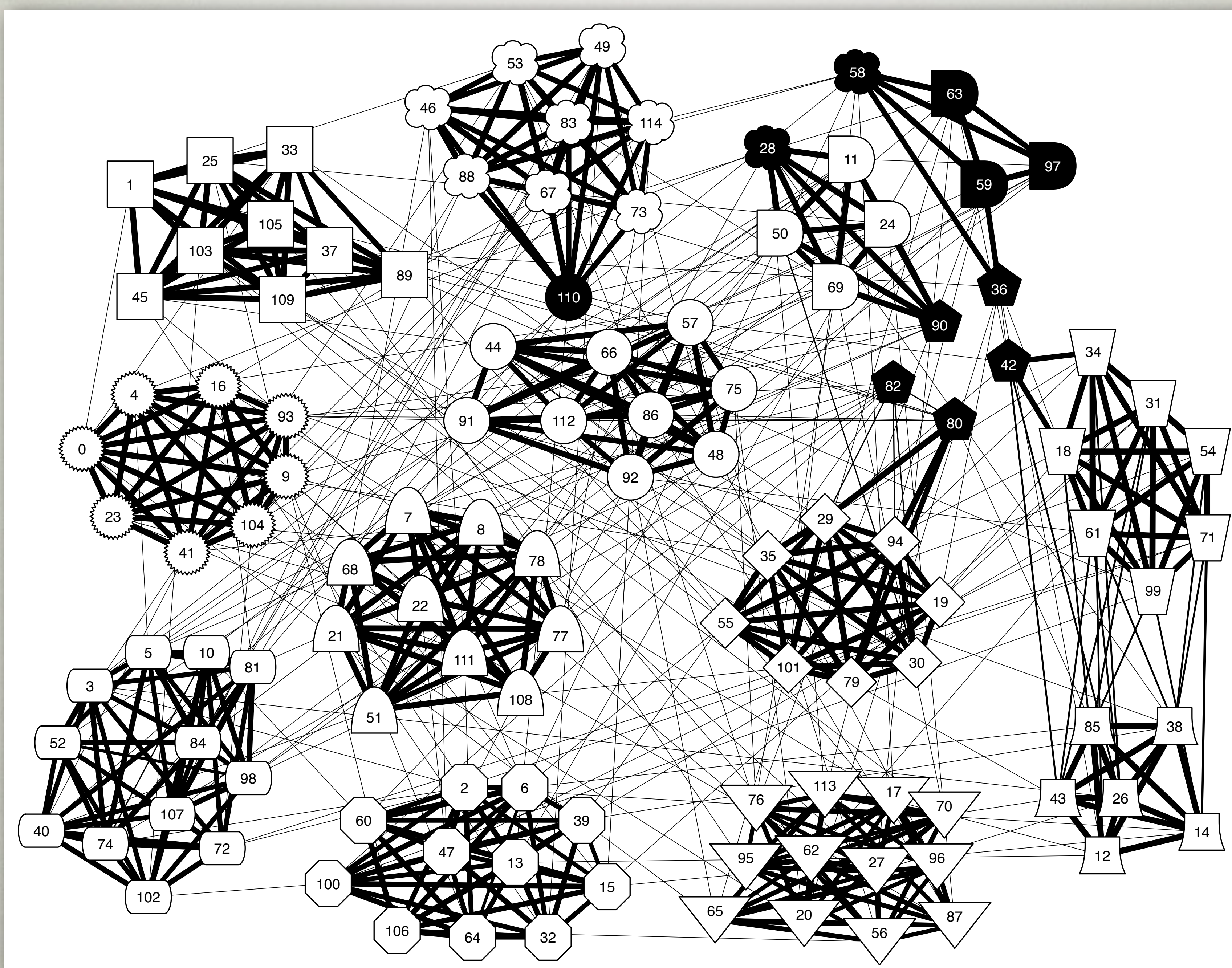
Vertex Annotations

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



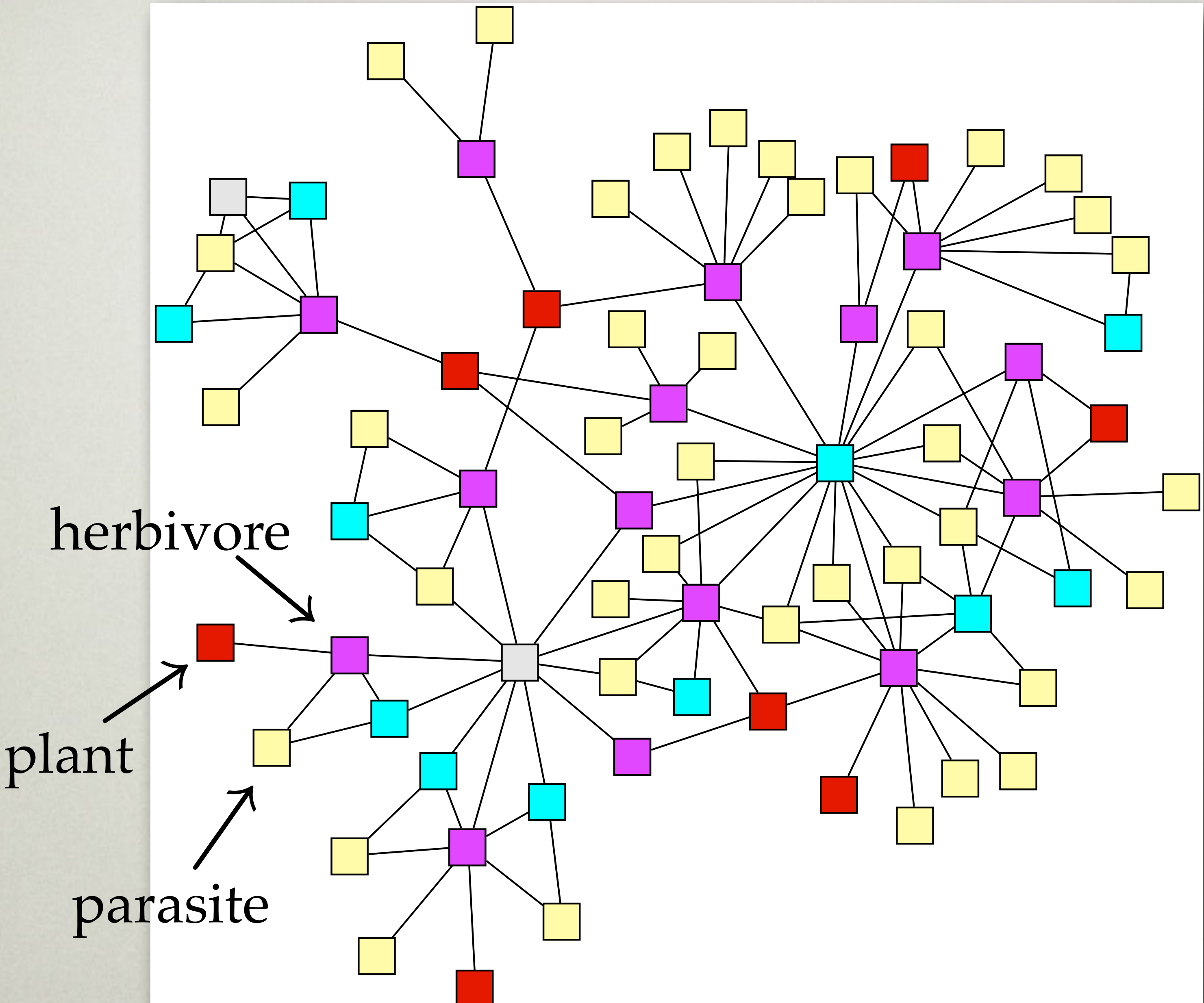

49
53
58
63
46
83
114
28
11
97
33
25
1
88
67
73
59
105
24
50
103
37
89
45
109
110
36
69
90
57
44
66
42
34
16
4
75
82
93
91
112
86
31
80
0
18
54
48
9
92
23
104
7
8
29
94
61
71
41
35
78
68
22
99
19
55
21
77
5
10
81
111
101
30
3
79
108
51
85
38
52
84
98
2
6
76
113
17
70
43
26
107
60
39
40
74
72
47
13
95
62
27
96
14
12
100
15
102
65
20
87
56
106
64
32

Model Checking

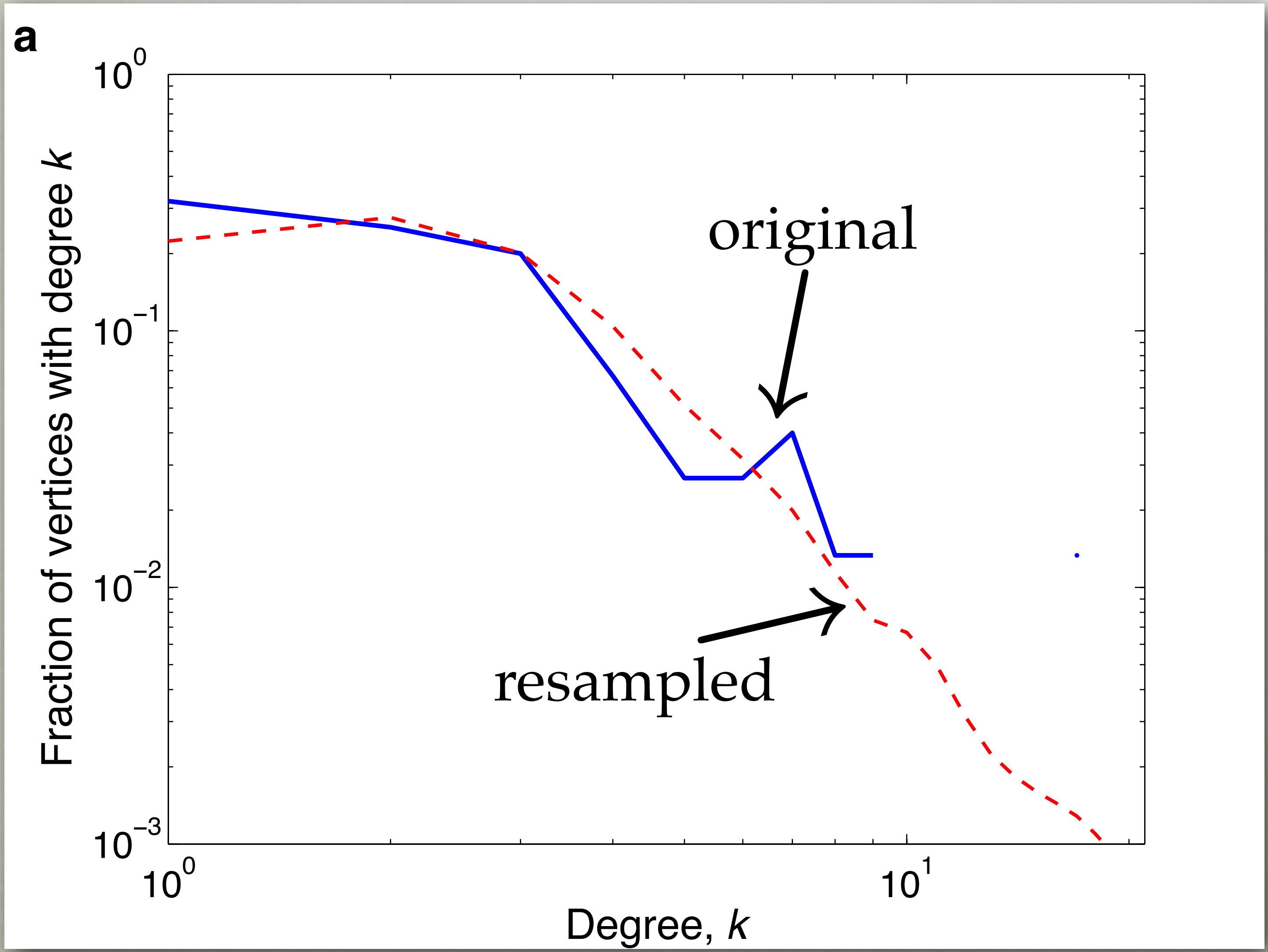
Model Checking

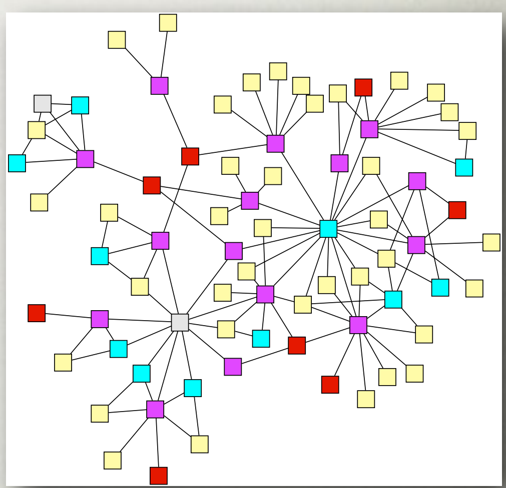
- Given graph G
- run MCMC to equilibrium
- then, for each sampled $\mathcal{D}$, draw a **resampled** graph G' from ensemble

Checking the model (goodness-of-fit):
do resampled graphs look like original?

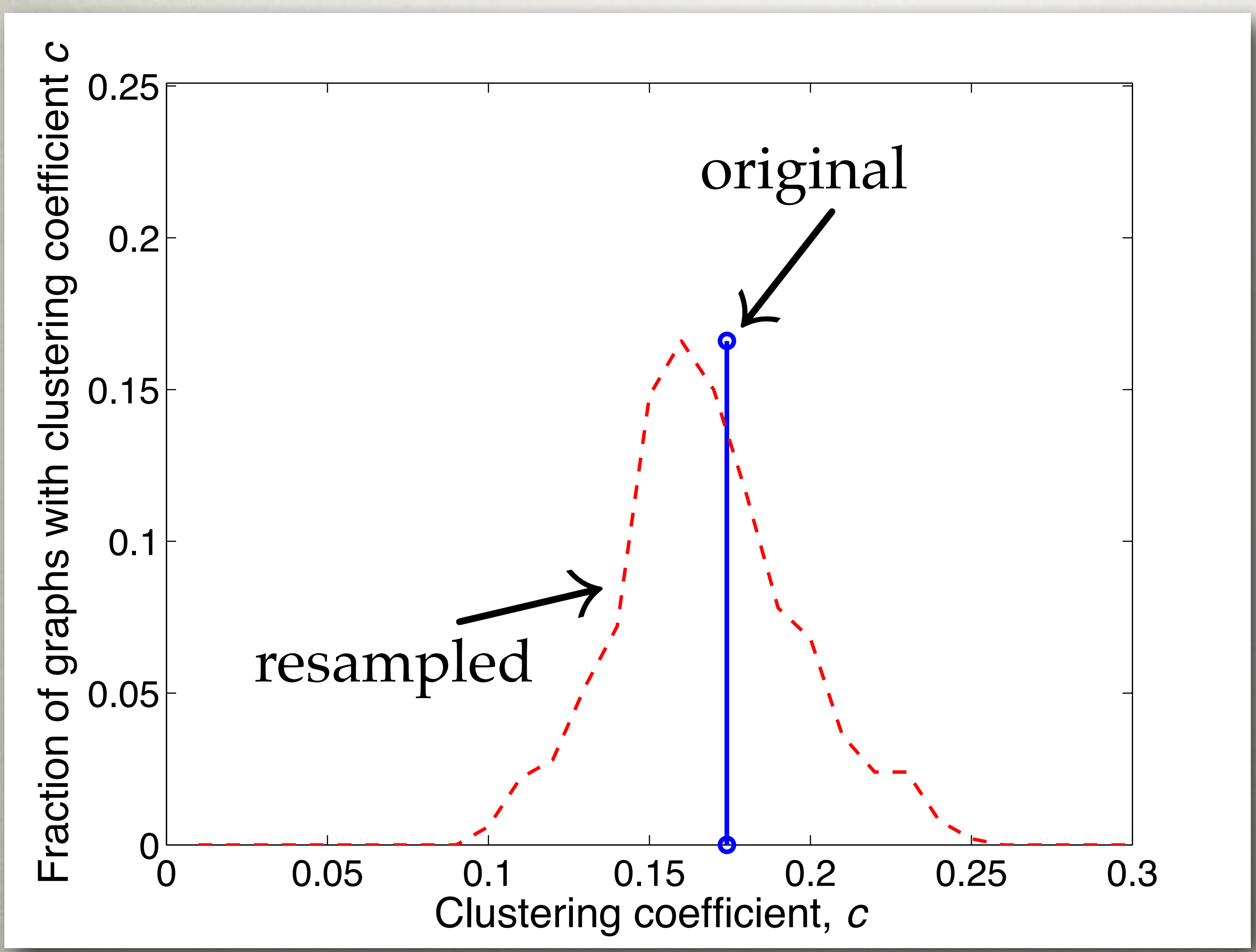

herbivore
plant
parasite

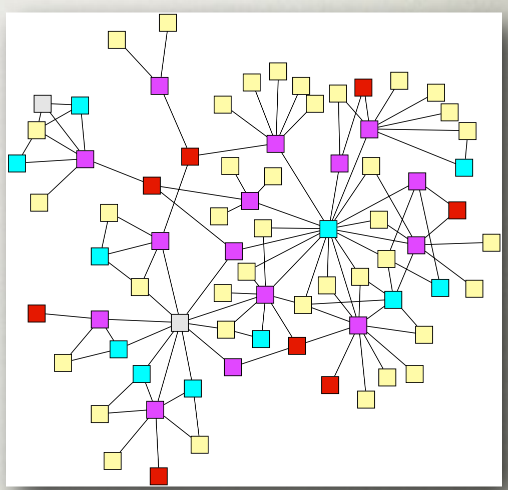
Degree Distribution



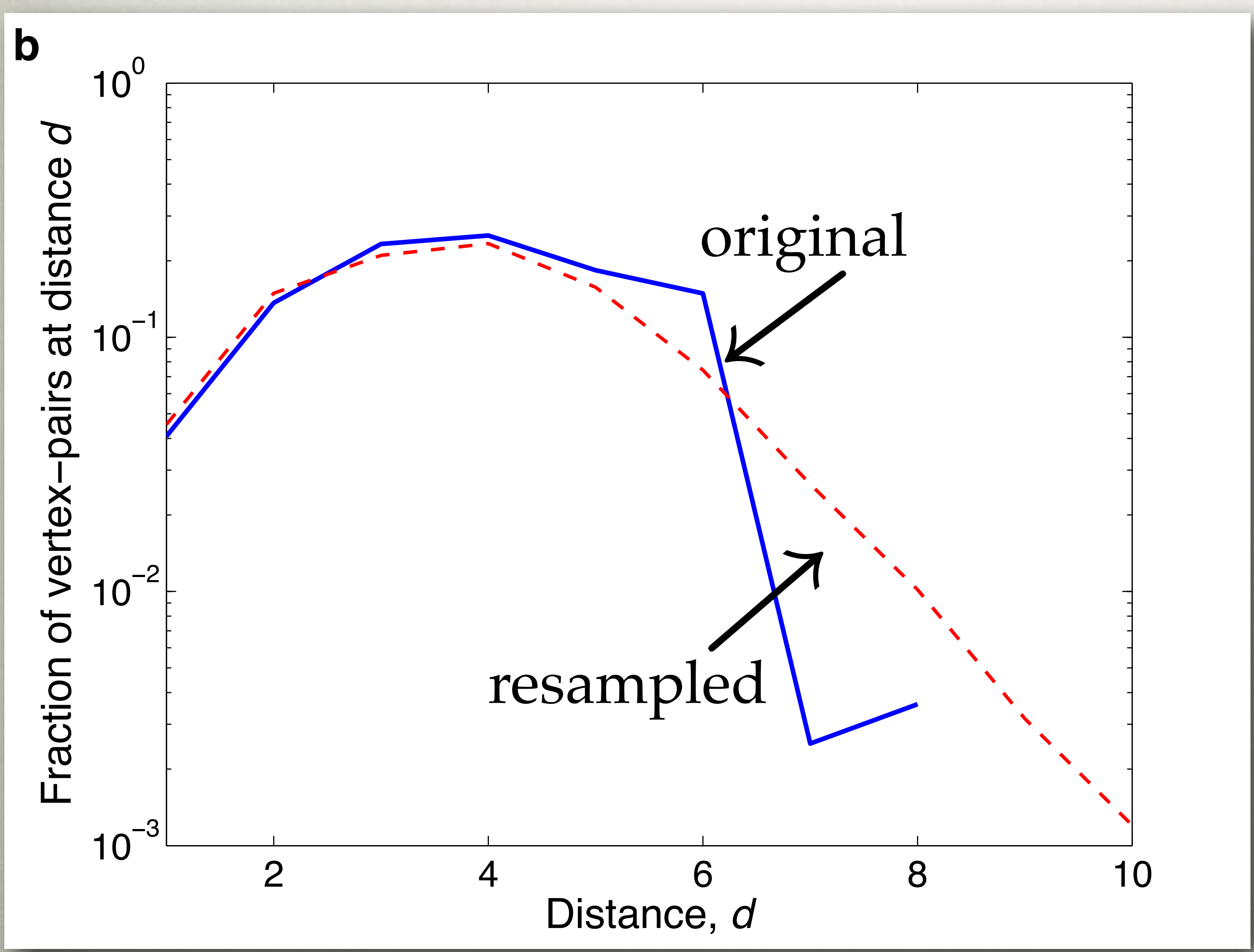


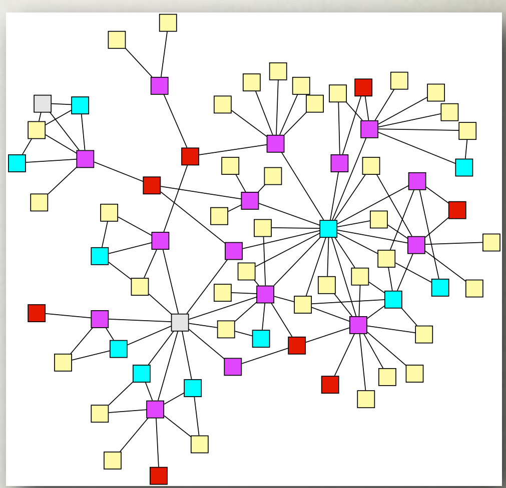
Clustering Coefficient





Distance Distribution





Predicting 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?

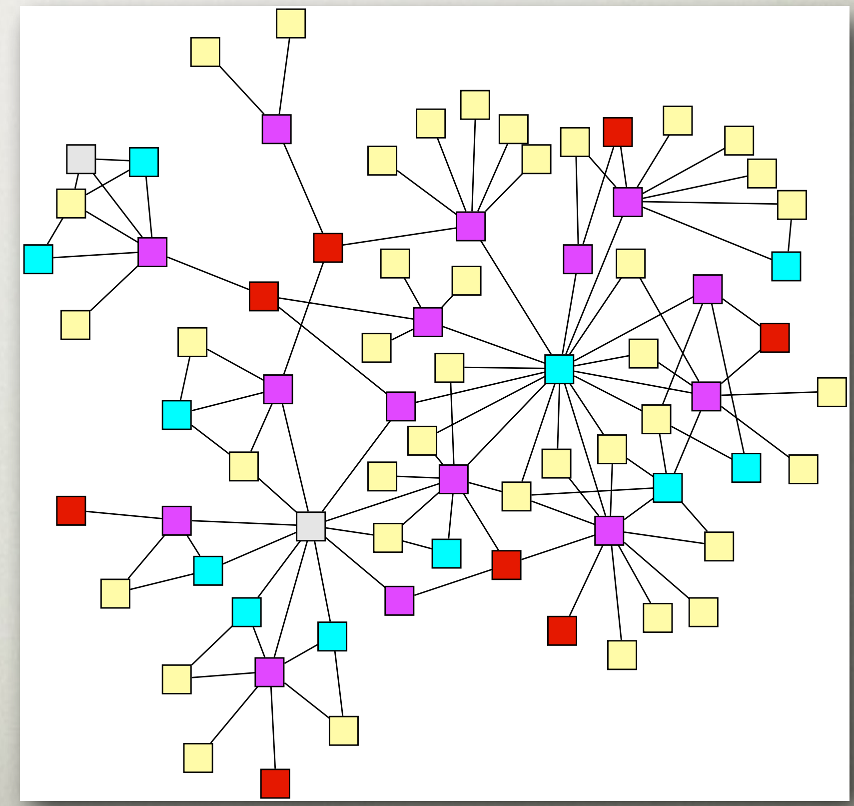
$$p_{\text{guess}} \approx \frac{k}{n^2 - m + k}$$

$$= O(n^{-2})$$

$$n = 75$$

$$m = 113$$

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



Generative Model Aproach

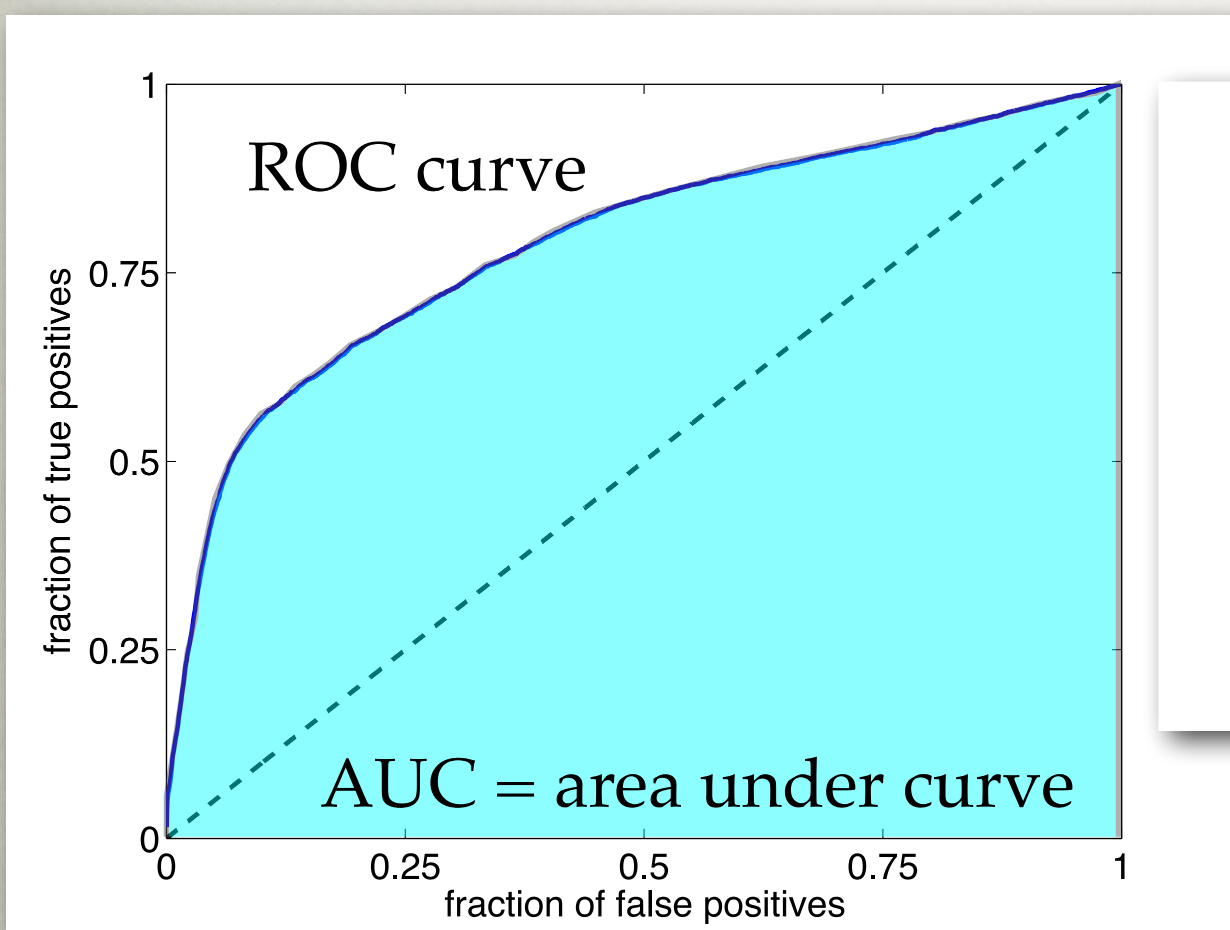
- Given incomplete graph G
- run MCMC to equilibrium
- then, over sampled $\mathcal{D}$, compute average $\langle p_r \rangle$ for links $(i, j) \notin G$
- predict links with high $\langle p_r \rangle$ values are missing

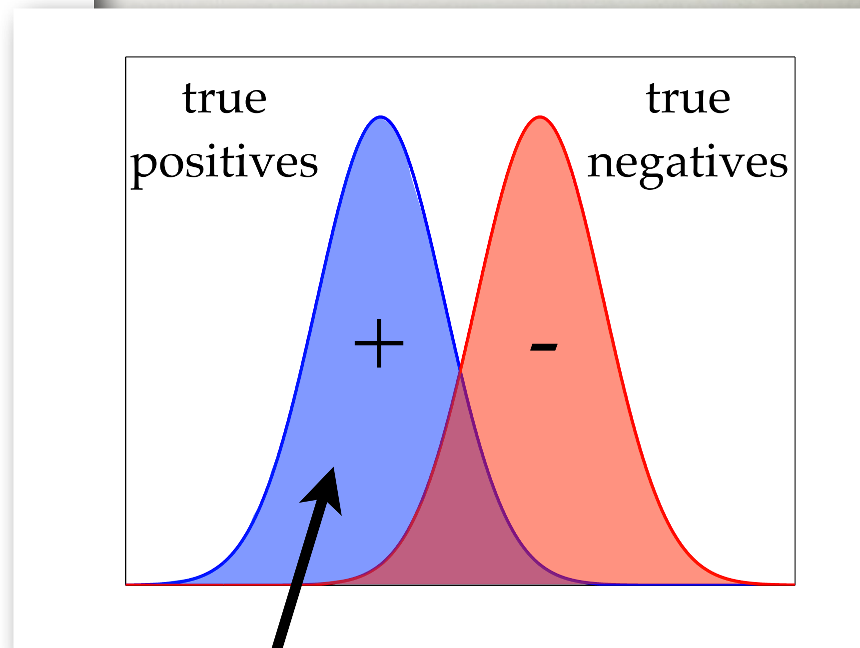
Test via leave-*k*-out cross-validation

perfect accuracy: AUC = 1

no better than chance: AUC = 1/2

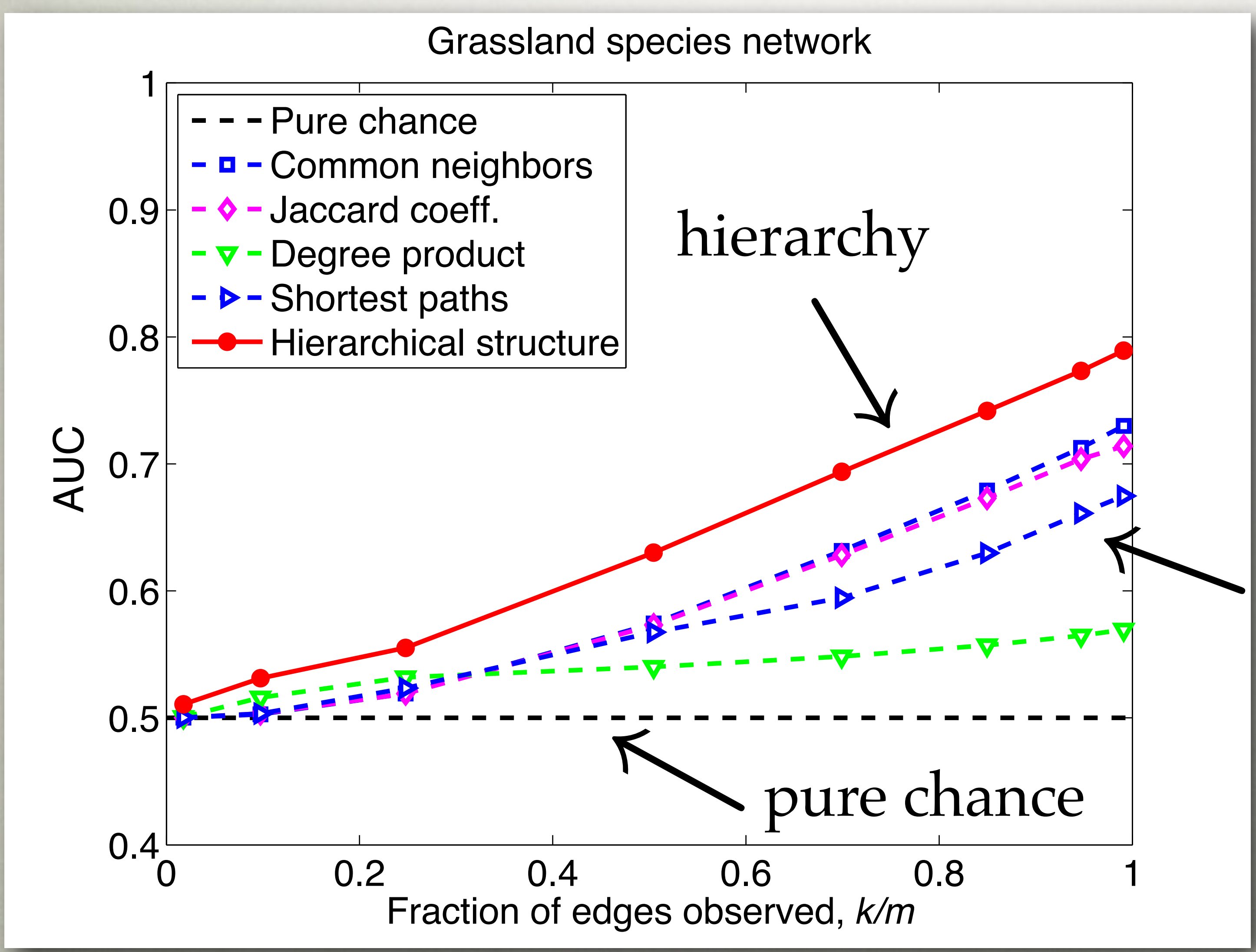
Scoring the Predictions

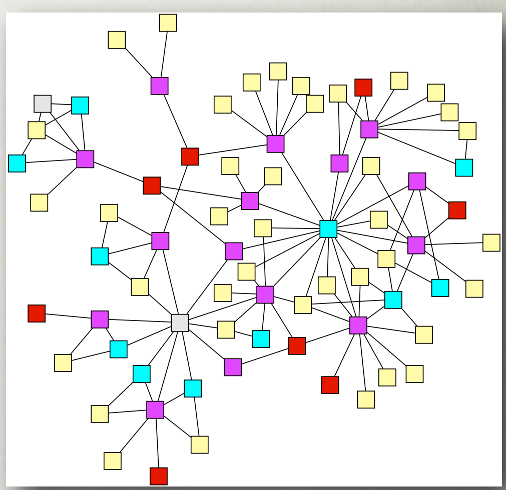




AUC =
Pr(distinguish
+ from -)

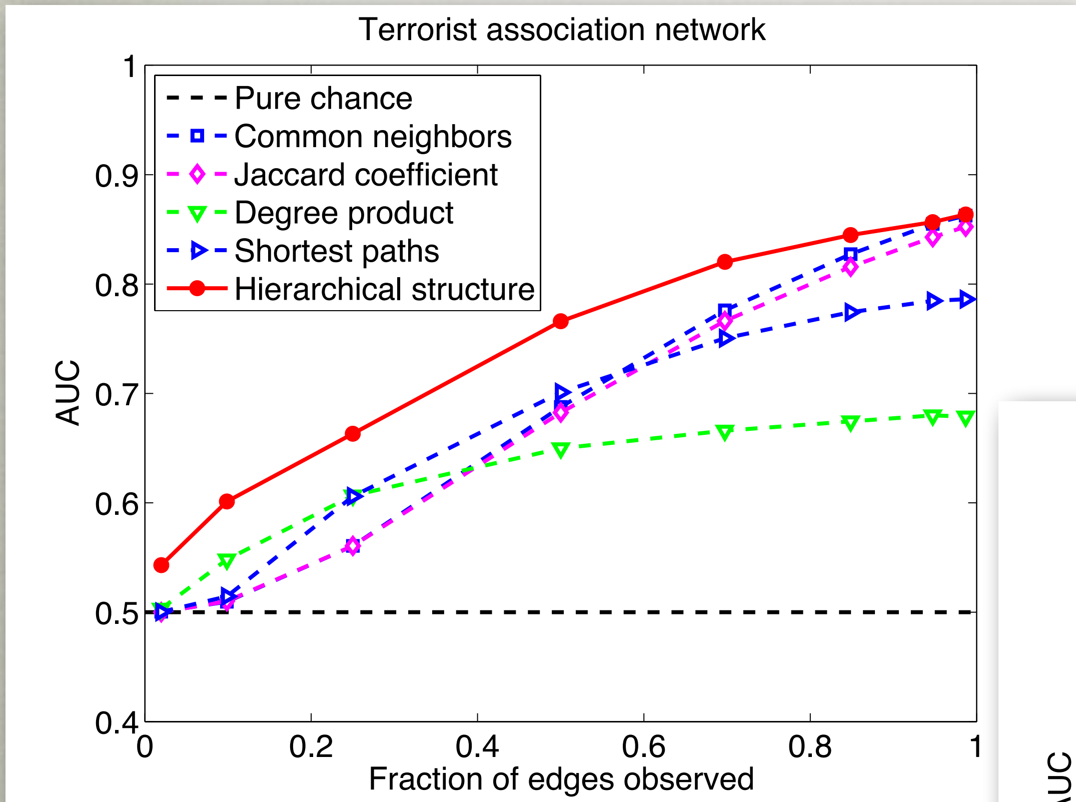
Performance 1

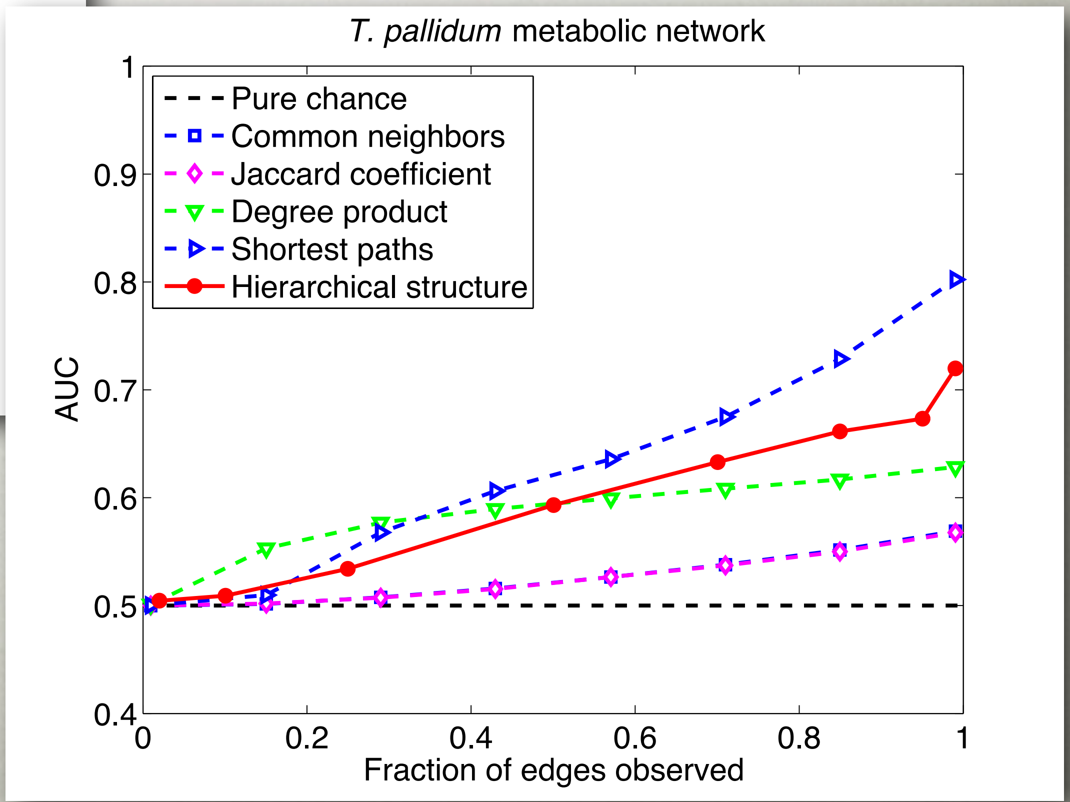




simple predictors

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