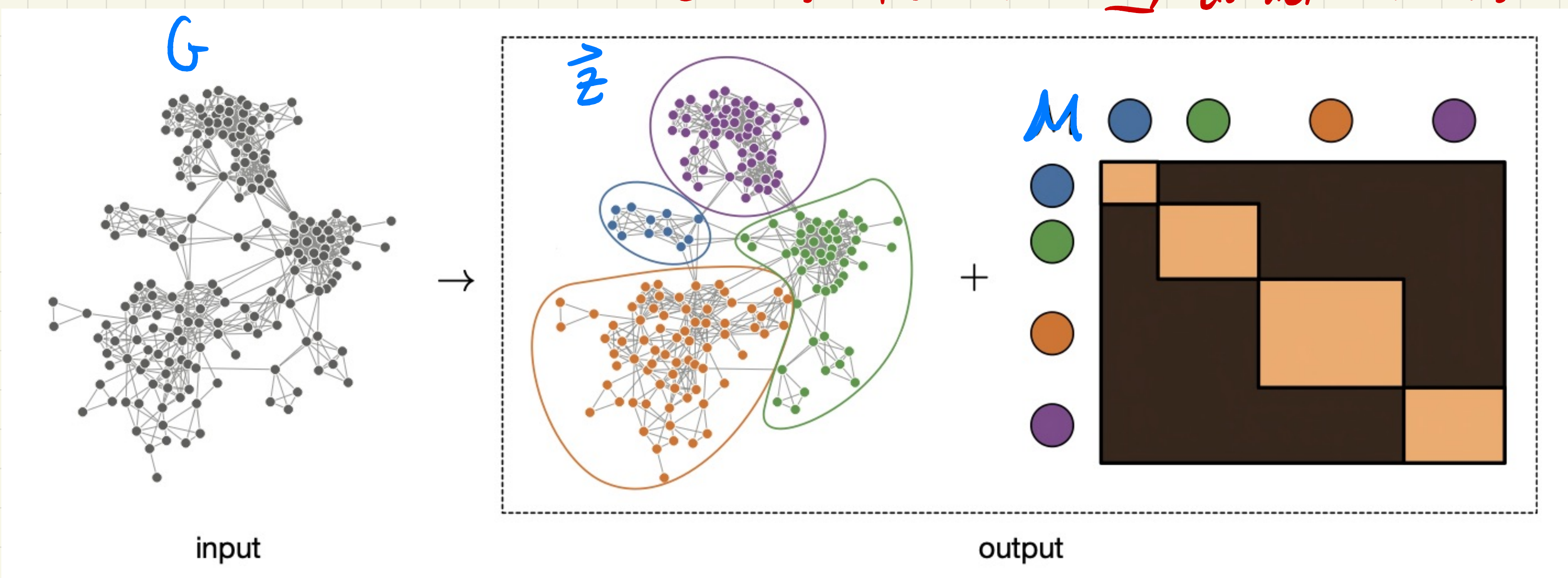
Lecture 6: Modular networks, inference

decomposing a $\underline{G}$ into "clusters" of nodes and "bundles" of edges

→ "community detection"

can use these for many downstream tasks



Statistical inference

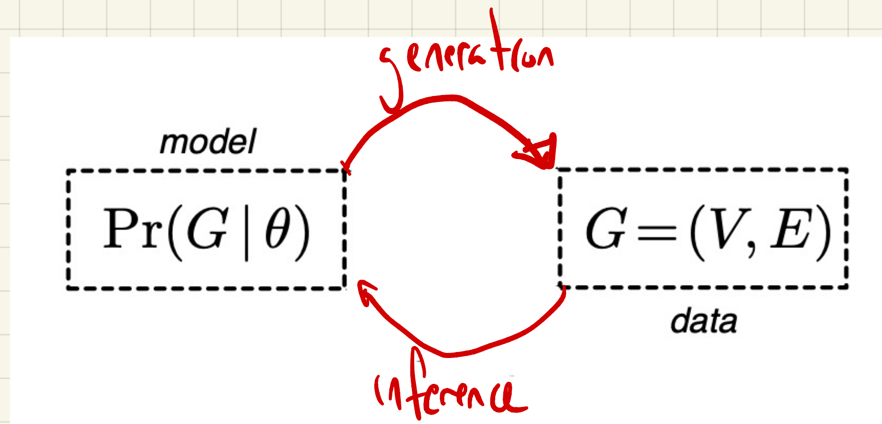
- there are 100s of algorithms for community detection now
  - * not all are useful *

- our approach: probabilistic generative models 2D structured random graphs

  $\Pr(G \mid \underline{\theta})$

  ↑ all structural assumptions

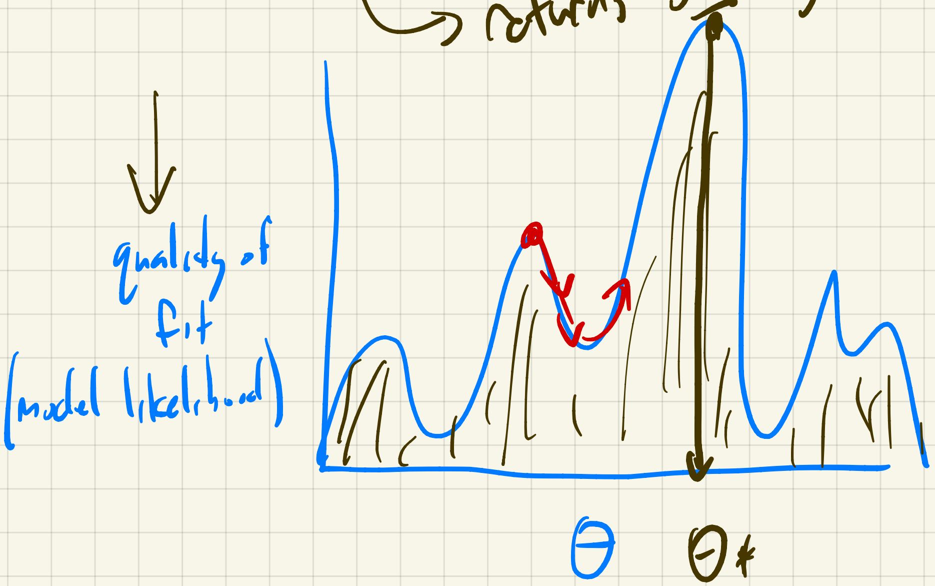
  1) assumptions are clearly stated
  2) defines model separately from estimation
  3) power Bayesian tools
  4) good at prediction $\Pr(i \to j \mid \hat{\theta})$
  5) $\Pr(G \mid \hat{\theta}) \rightarrow$ generate synthetic nets.
  6) compare models easily



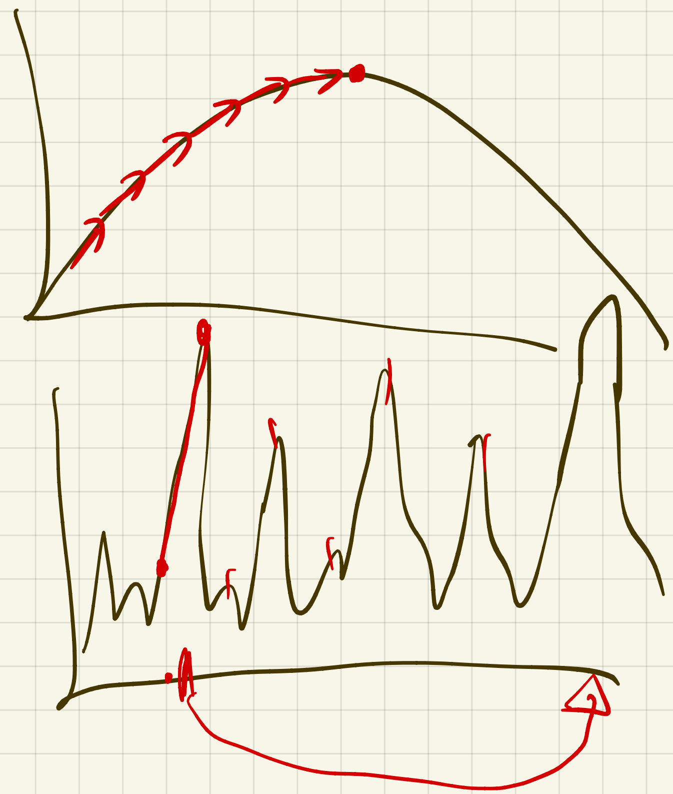
- Costs: models can be complicated

to optimize or to sample? $\rightarrow \Pr(\theta)$

returns one good θ



Markov chain Monte Carlo

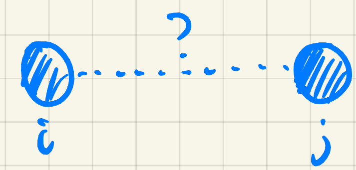


the likelihood of the SBM

$\Theta = (c, \vec{z}, M)$ # of groups c partition of nodes z mixing matrix M

given a choice of θ, the probability of generating G is $\Pr(G \mid \theta)$

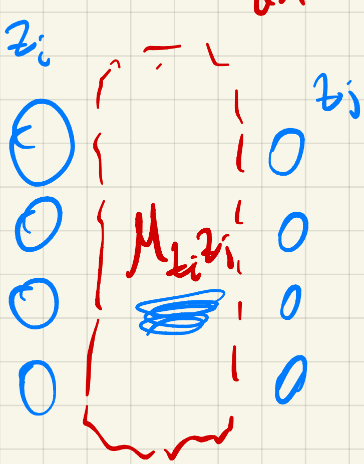
- call this the likelihood



$$\underline{\mathcal{L}}\Big(\underset{\uparrow \atop \text{data}}{G} \,\Big|\, \underbrace{z, M}_{\theta}\Big) = \prod_{i,j} \Pr(i \to j \mid z, M)$$

$$= \Big(\prod_{(i,j)\in E} \overbrace{\Pr(i \to j \mid z, M)}\Big)\Big(\prod_{(i,j)\notin E} 1 - \Pr(i \to j \mid z, M)\Big)$$

edges $(A_{ij} = 1)$ non-edges $(A_{ij} = 0)$



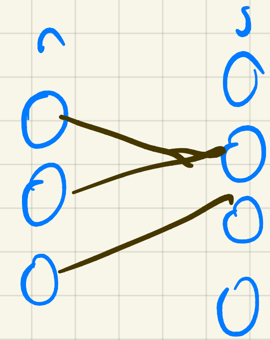
$$= \Big(\prod M_{z_i z_j}\Big)\Big(\prod 1 - M_{z_i z_j}\Big)$$

how many terms are there in this equation?

observation: lots of these terms are identical, and can be combined

→ which terms?

$n_r = 3$



$n_s = 4$

$n_{rs} = n_r n_s$

auxiliary variables:

$$e_{rs} = \sum_{i<j}^{n} A_{ij}\,\delta_{r,z_i}\,\delta_{s,z_j}$$

count actual # edges b/w groups r and s

$$n_{rs} = \sum_{i<j} \delta_{r,z_i}\,\delta_{s,z_j}$$

count # possible edges b/w groups r and s

now:

$$\mathcal{L}(G|z,M) = \prod_{r,s} \overbrace{(M_{rs})^{e_{rs}}}^{\text{edges}}\;\overbrace{(1-M_{rs})^{n_{rs}-e_{rs}}}^{\text{non-edges}}$$

is every i,j accounted for? how long does this take to compute?

the maximum likelihood choice

choose the θ^* that maximizes $\mathcal{L}(G \mid \theta)$

- note: the SBM likelihood is a bunch of Bernoulli trials (aka coin flips)

$\Rightarrow$ hence $\hat{\mu}_{rs} = \dfrac{e_{rs}}{n_{rs}}$

*subject to mild "regularity" conditions, the max. like. estimate is asymptotically consistent

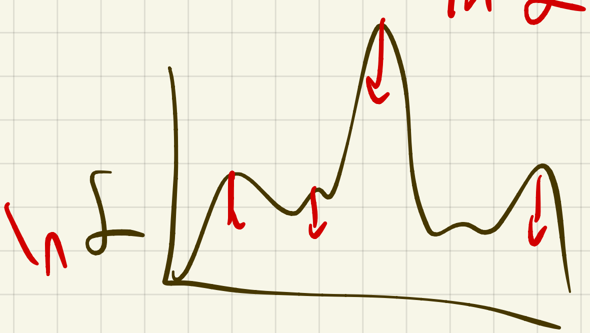
$\lim_{n\to\infty} \hat{\theta} \to \theta$

yay! :)

- plug & chug & taking logs:

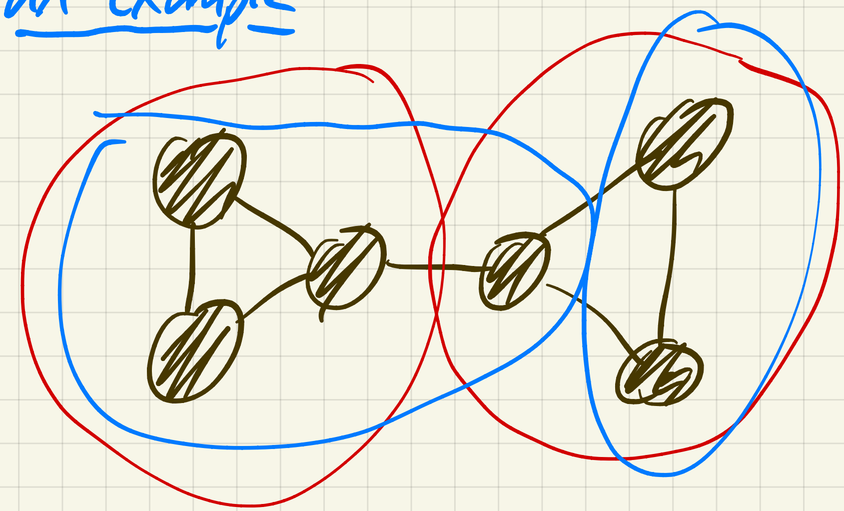
$$\ln \mathcal{L} = \sum_{r,s} \underbrace{e_{rs} \ln\left(\frac{e_{rs}}{n_{rs}}\right)}_{\text{edges}} + \underbrace{(n_{rs} - e_{rs}) \ln\left(\frac{n_{rs}-e_{rs}}{n_{rs}}\right)}_{\text{non-edges}}$$

*why is $\underset{\theta}{\operatorname{argmax}} \ln \mathcal{L} = \underset{\theta}{\operatorname{argmax}} \mathcal{L}$?

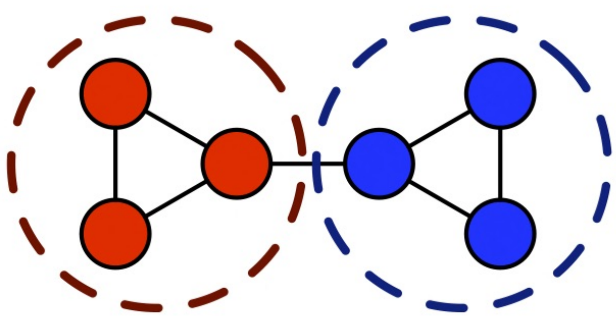


* define $0^0 = 1$

an example



"good" partition



$$\mathcal{L}_{\text{good}} = 0.043304\ldots$$
$$\ln \mathcal{L}_{\text{good}} = -3.1395\ldots$$

e_{rs}/n_{rs}	red	blue
red	3/3	1/9
blue	1/9	3/3

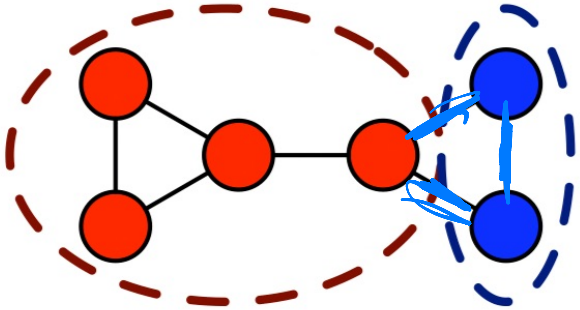
$$\ln \mathcal{L} = \sum_{r,s} e_{rs} \ln \frac{e_{rs}}{n_{rs}} + (n_{rs} - e_{rs}) \ln \left(\frac{n_{rs} - e_{rs}}{n_{rs}} \right)$$

$$\ln \mathcal{L}_{\text{good}} = \left(3 \ln \frac{3}{3}\right) + \left(1 \ln \frac{1}{9} + 8 \ln \frac{8}{9}\right) + \left(3 \ln \frac{3}{3}\right)$$

$$= -3.1395$$

$$e^{\ln \mathcal{L}} = 0.043\ldots$$

"bad" partition



$$\mathcal{L}_{\text{bad}} = 0.000244\ldots$$
$$\ln \mathcal{L}_{\text{bad}} = -8.3178\ldots$$

e_{rs}/n_{rs}	red	blue
red	4/6	2/8
blue	2/8	1/1

$$\ln \mathcal{L} = \sum_{r,s} e_{rs} \ln \frac{e_{rs}}{n_{rs}} + (n_{rs} - e_{rs}) \ln\left(\frac{n_{rs} - e_{rs}}{n_{rs}}\right)$$

$$\ln \mathcal{L}_{bad} = \left(4 \ln \frac{4}{6} + 2 \ln \frac{2}{6}\right) + \left(2 \ln \frac{2}{8} + 6 \ln \frac{6}{8}\right) + \left(1 \ln \frac{1}{1}\right) = -8.3178\ldots$$

$$e^{\ln \mathcal{L}} = 0.000244\ldots$$

$$\frac{\mathcal{L}_{good}}{\mathcal{L}_{bad}} = e^{\left(\ln \mathcal{L}_{good} - \ln \mathcal{L}_{bad}\right)} = 177$$

the likelihood of a DC-SBM

$$\Theta = (c, \vec{z}, \vec{k}, M)$$

in SBM : $\Pr(i \to j) \rightarrow \text{Bernoulli}(M_{z_i z_j})$

in DC-SBM : $\Pr(i \to j) \rightarrow \text{Poisson}(\gamma_i \gamma_j M_{z_i z_j})$

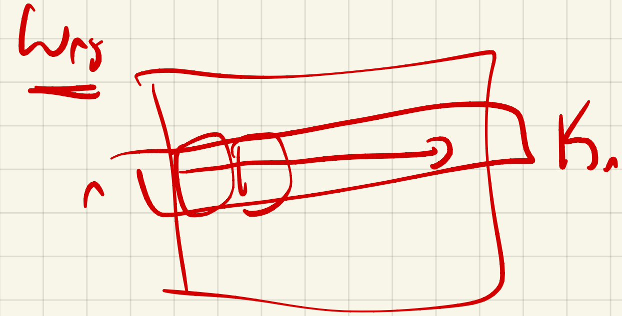
hence

$$\mathcal{L}(G \mid z, \gamma, M) = \prod_{i,j} \text{Poisson}(\gamma_i \gamma_j M_{z_i z_j})$$

$$= \prod_{i<j} \frac{(\gamma_i \gamma_j M_{z_i z_j})^{A_{ij}}}{A_{ij}!} \exp(-\gamma_i \gamma_j M_{z_i z_j}) \times \prod_i \frac{(\frac{1}{2}\gamma_i^2 M_{z_i z_i})^{A_{ii}/2}}{(A_{ii}/2)!} \exp\left(-\frac{1}{2}\gamma_i^2 M_{z_i z_i}\right)$$

just like SBM, likelihood has 2 parts: between group & within group edges just like SBM, we can simplify $\mathcal{L}(G \mid z, \gamma, \mu)$

- collect various constants
- substitute MLEs



$$\hat{\gamma}_i = \frac{k_i}{\kappa_{z_i}} \qquad \hat{\mu}_{rs} = \omega_{rs}$$

where

$$\kappa_r = \sum_{s} \omega_{rs} = \sum_{i} k_i \, \delta_{z_i r} \qquad \omega_{rs} = \sum_{i,j=1}^{n} A_{ij} \, \delta_{z_i r} \, \delta_{z_j s}$$

"group degrees" mixing matrix counts (of stubs)

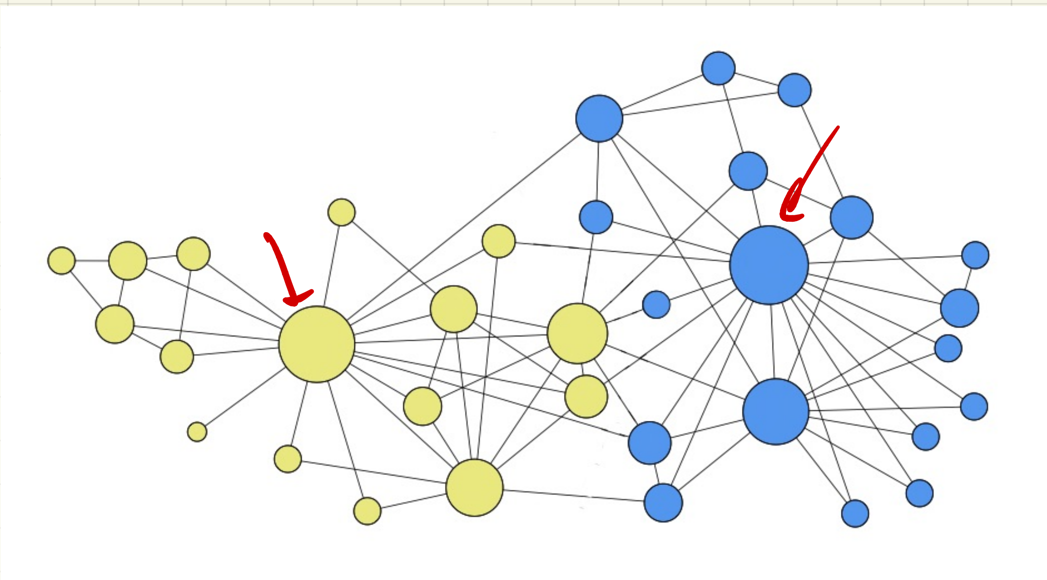
then

$$\ln \mathcal{L} = \sum_{r,s} \omega_{rs} \ln \frac{\omega_{rs}}{\kappa_r \kappa_s}$$

Key point: SBM $\underline{m_{rs}}$

DCSBM $\underline{\kappa_r \kappa_s}$

an example: the Zachary Karate Club

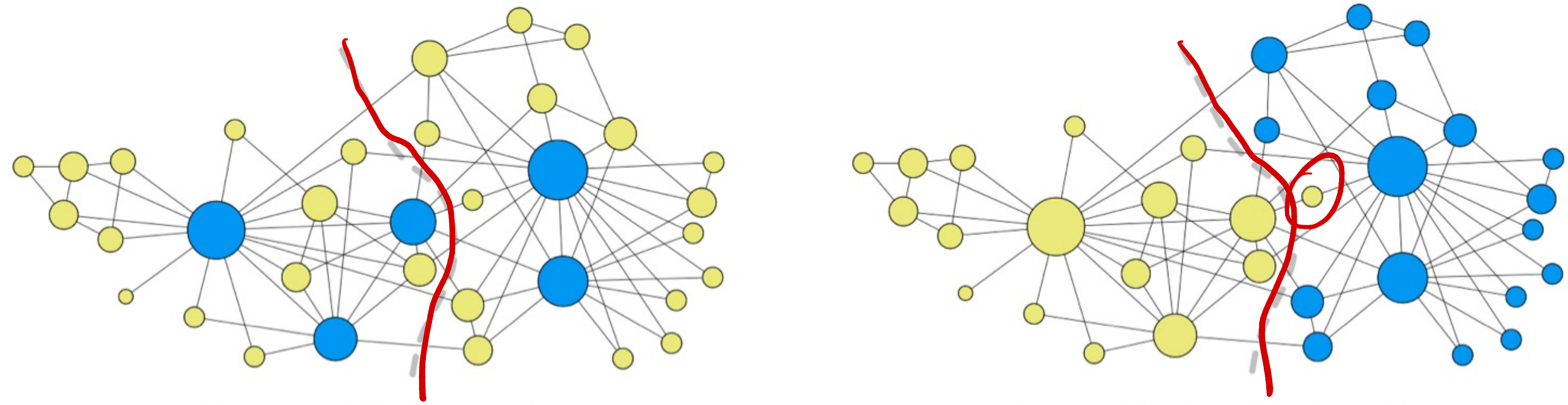


$n = 34$

$m = 78$

What happens if we SBM or DC-SBM this network?

the "social partition"



best $\hat{z}$, SBM ($c = 2$) — best $\hat{z}$, DC-SBM ($c = 2$)

"leader-follower" partition — almost the social partition

why? — why?

e_{rs}/n_{rs}	A (16)	B (18)
A (16)	33/120	10/288
B (18)	—	35/153

$\ln \mathcal{L}_{\text{SBM}} = -196.29$ (social)

e_{rs}/n_{rs}	A (5)	B (29)
A (5)	5/10	54/145
B (29)	—	19/406

$\ln \mathcal{L}_{\text{SBM}} = -179.39$ (leader-follower)

ω_{rs}	A (16)	B (18)	κ_r
A (16)	66	10	76
B (18)	10	70	80

$\ln \mathcal{L}_{\text{DC-SBM}} = -739.43$ (social)

ω_{rs}	A (5)	B (29)	κ_r
A (5)	10	54	64
B (29)	54	38	92

$\ln \mathcal{L}_{\text{DC-SBM}} = -772.28$ (leader-follower)

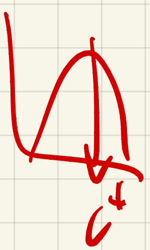
note: how do we choose c, the # of groups?

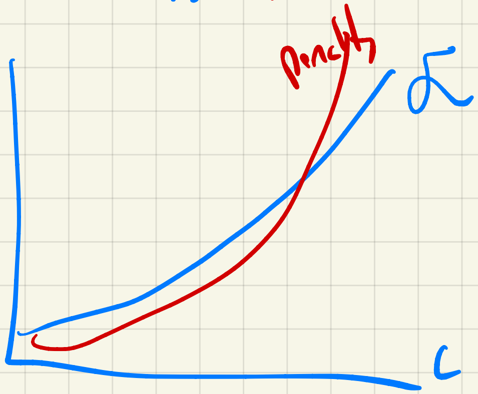
$\Theta = (c, z, \vec{k}, M)$

$\vec{k}$ ↑ DC-SBM

SBM & DC-SBM both have fixed c inside $\mathcal{L}$

→ can also learn the best c, but... → regularization





$c=2$ vs $c=3$

- AIC
- BIC
- MDL
- LRT
- ⋮ Bayesian marginalization

$M: O(c^2)$

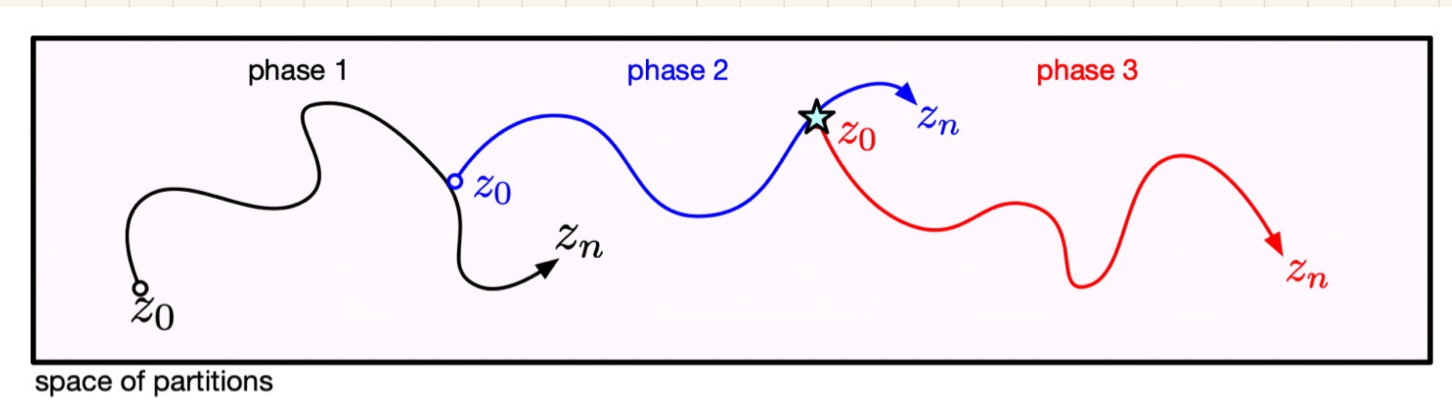
Finding good partitions

- search over the n^c possible partitions to find a good $\hat{z}$
- many, many ways
  - Markov chain Monte Carlo (MCMC) - simulated annealing (tempering)
  - expectation-maximization
  - Belief-Prop.

locally greedy heuristic

a generalization of the Kernighan-Lin heuristic (developed in 1970s for graph partitioning)

- constructs a sequence of partitions $z_i, z_{i+1}, z_{i+2} \dots$ in a series of phases
- continue phases until every partition in phase ℓ is no better than the best z_* (with the best $\ln \mathcal{L}_*$) in phase $\ell-1$
- if a new best z_* is found, start a new phase at that partition



high level: take greedy choice, for changing label of 1 node at a time

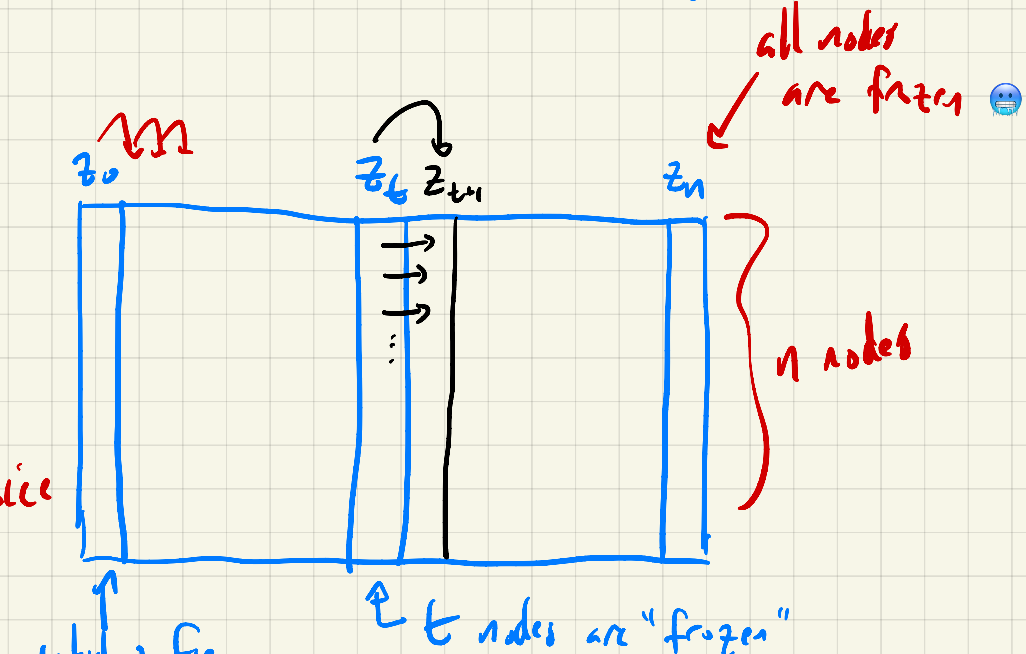
1) initially: choose a random z_0, set $z_* = z_0$, $L_* = \ln \mathcal{L}(z_0)$

2) within a phase:

- unfreeze all nodes
- loop: choose node label-change that is BEST * ; move that node & "freeze" it

$(c-1)\times(n-t)$

greedy choice



initial z for this phase

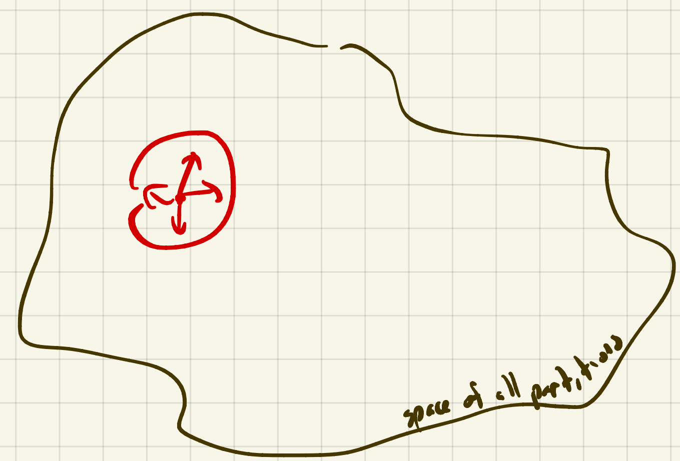
- how many entries change from $z_t \to z_{t+1}$?
- how can we efficiently store this info?

3) at end of phase:

if any $z_t > z_*$, then
update z_*, L_*
set $z_0 = z_*$
start new phase

update "best" partition

else: return z_*, L_*



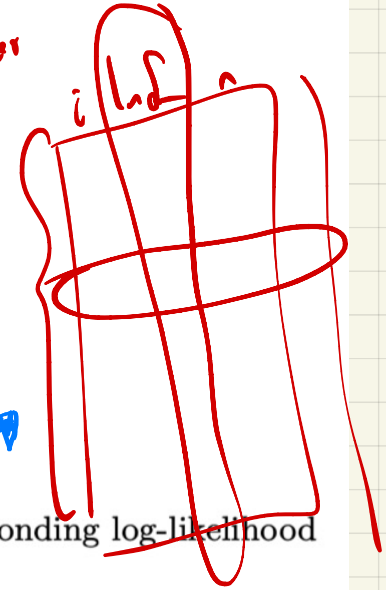
`function local-greedy(`G`):`

1. **`initialization`**
   (a) create z_0 by assigning $z_{0,i} = \text{Uniform}(\{1, \ldots, c\})$,
   (b) record its log-likelihood $L_0 = \ln \mathcal{L}(z_0)$
   (c) set $t = 1$ and set all nodes to be "unfrozen"

all label choices → $(c-1) \times (n-t)$ ← all unfrozen nodes

find max over these

$n-t$



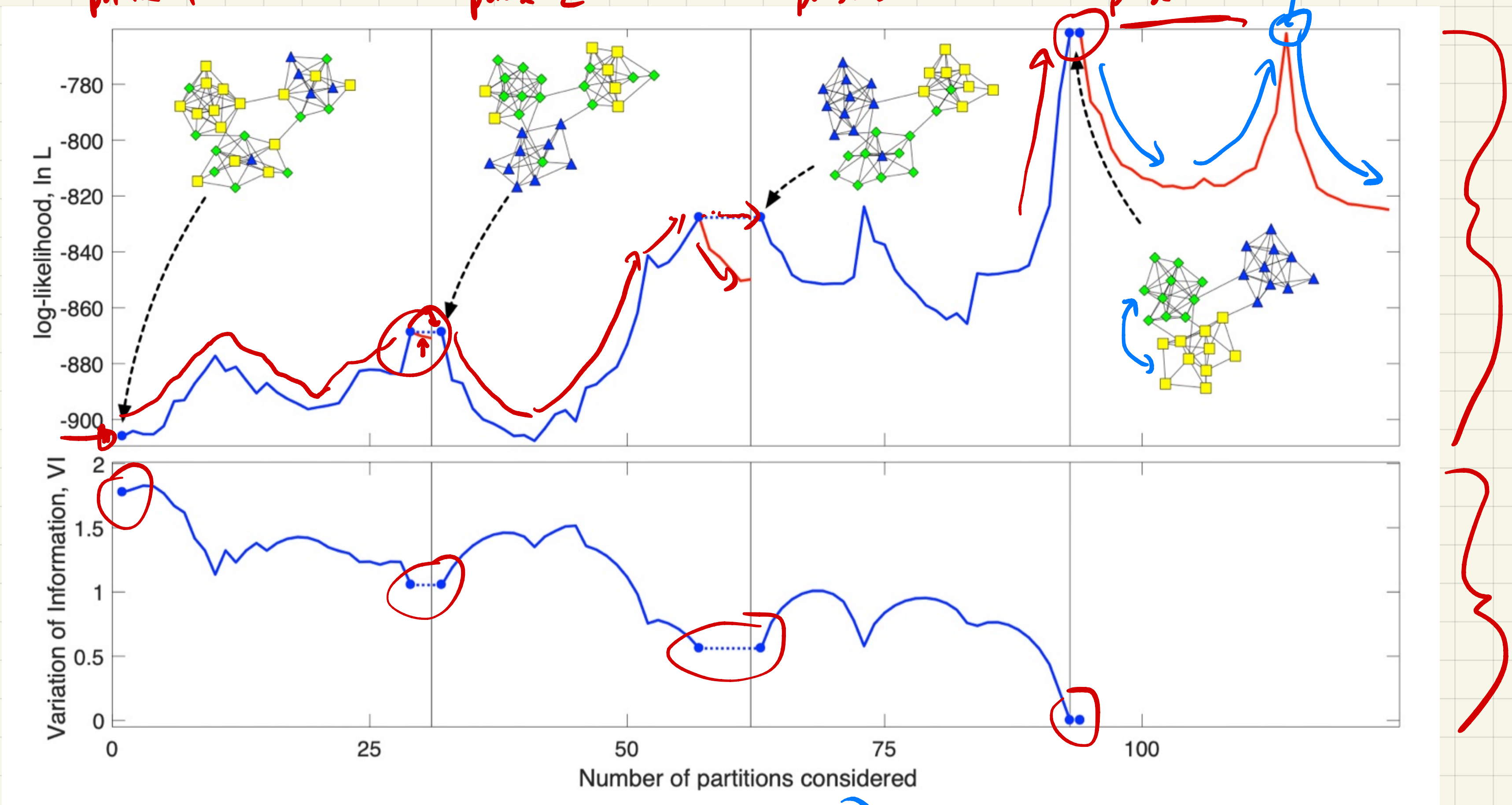
2. **`run a phase`**
   while $t < n$ (some node is unfrozen):
   (a) $z_t = z_{t-1}$ (copy the last partition)
   (b) **`try all the moves`**
   for each of the $(n - t)$ unfrozen nodes:
      i. let i be an unfrozen node, and let $s = z_{t,i}$ (i's current group is s)
      ii. for each choice of a group $r \in \{1, \ldots, c\} - s$, set $z_{t,i} = r$, compute the corresponding log-likelihood value, and store it as a tuple $(i, \ln \mathcal{L}, r)$
      iii. of these tuples, keep the one with the most positive log-likelihood $\ln \mathcal{L}$
      iv. reset $z_{t,i} = s$ * very important! put node i back in its original group
   (c) **`a greedy choice`**
   of the $(n - t)$ tuples $(i, \ln \mathcal{L}, r)$, one for each unfrozen node, select the one with the most positive $\ln \mathcal{L}$
   (d) **`freeze that node`**
   set $z_{t,i} = r$ (move one node)
   set $L_t = \ln \mathcal{L}$ (record its log-likelihood)
   freeze node i
   $t = t + 1$ (next partition in the sequence)

3. **`evaluate the phase`**
   (a) among the $(n + 1)$ log-likelihoods $L_0, \ldots, L_n$, identify the largest L_* and corresponding partition z_*
   (b) (halt) if $L_* \leq L_0$, then halt and return z_0
   (c) (new phase) else, set $z_0 = z_*$, $L_0 = L_*$, $t = 1$, unfreeze all nodes, and run a new phase

an example

$n=30$ $c=3$ (equal sized) assortative M

phase 1 phase 2 phase 3 phase 4



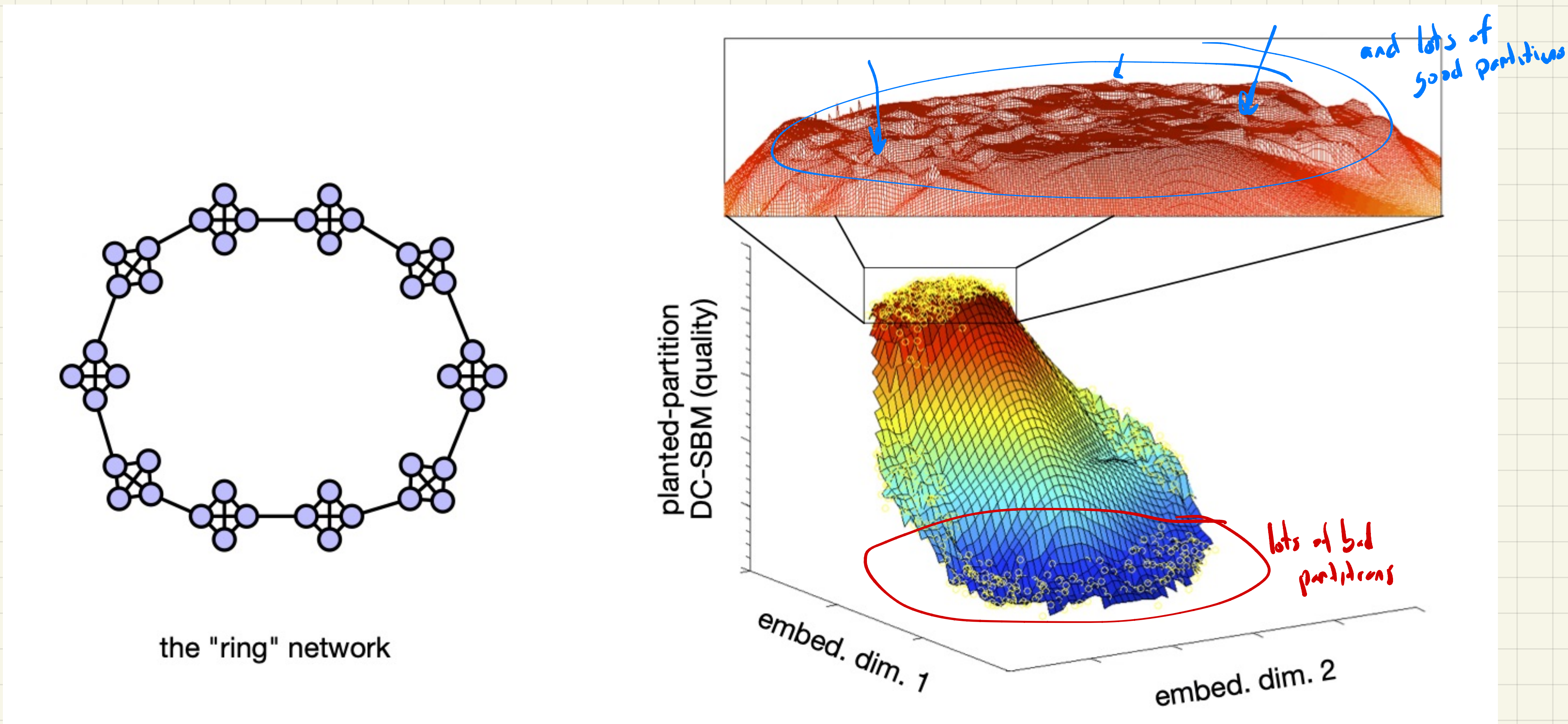
distance measure for partitions $VI(A, B)$

z ground truth

3 caveats for community detection

(1) Competitive local optima

- it's unsupervised ⇒ community detection just compresses the data
- many "good" compressions

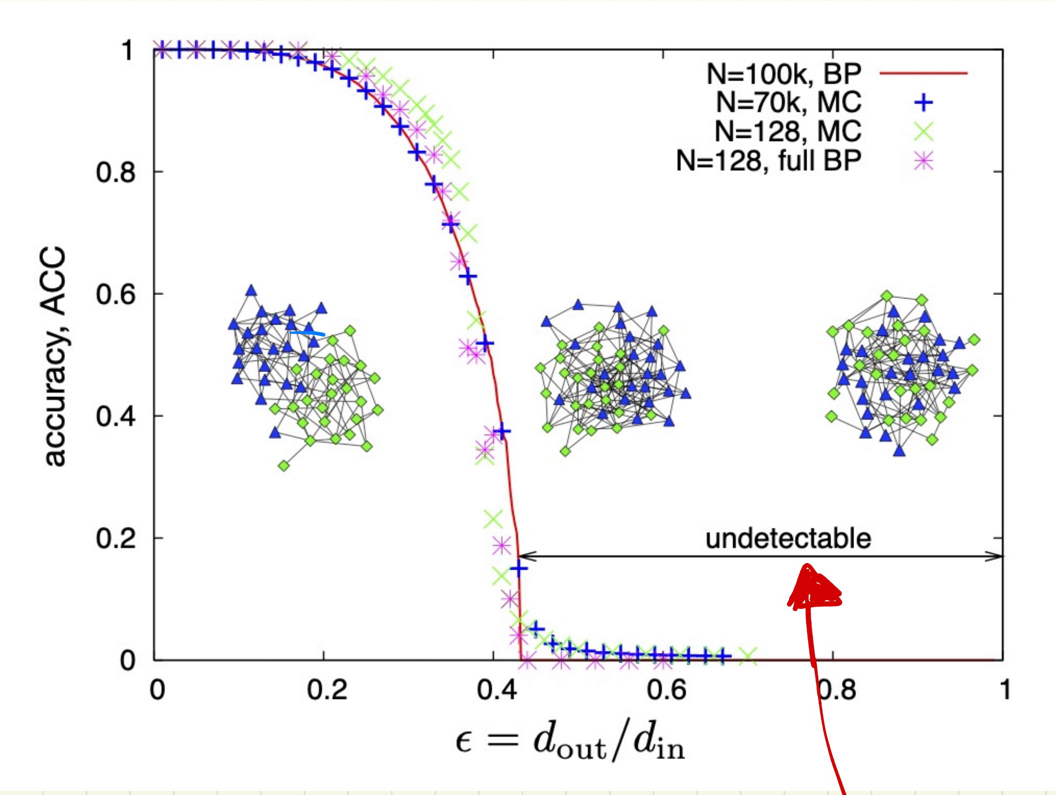


the "ring" network

② not all modules are detectable

SBM
planted partition $c=2$

stronger communities
weaker communities



no algorithm can work!

3 no ground truth and No Free Lunch

- no guarantee output is scientifically meaningful (no ground truth)
- no "best" algorithm for all data sets (No Free Lunch)

$$\underset{\uparrow \text{ SBM}}{g}\Big(\underset{\uparrow \text{ leader-follow}}{T}\Big) \rightarrow G_{ZKC} \leftarrow \underset{\uparrow \text{ DC-SBM}}{g'}\Big(\underset{\uparrow \text{ social partition}}{T'}\Big)$$