①

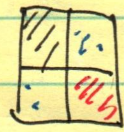
March 5, 2024

Lecture 8 a

§ Modular networks, inference

G $\{z\}$ M

G



clusters of nodes bundles of edges

$$\underset{\text{model}}{\mathbb{P}(G \mid \theta)} \underset{\xleftarrow[\text{inference}]{}}{\xrightarrow{\text{generate}}} \underset{\text{data}}{G = (V, E)}$$

» There are 100s of algm's for community detection now.
(not all are useful)

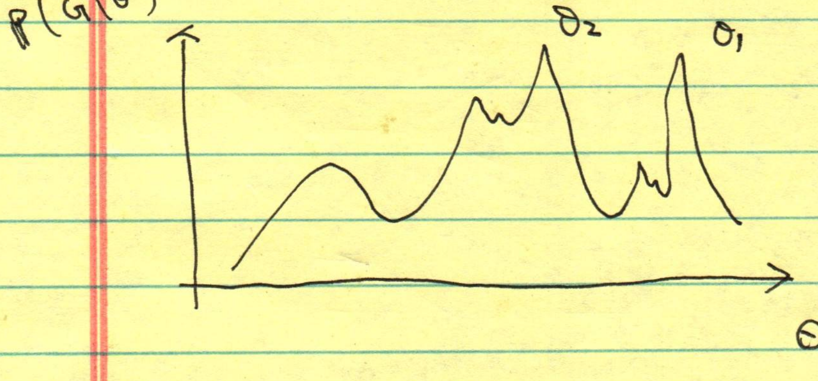
» our approach : prob. generative models (structured random graphs)

1) generate synthetic data
2) compare different models $\mathbb{P}(G \mid \theta_1)$ vs. $\mathbb{P}(G \mid \theta_2)$
3) predict missing links (things) $\mathbb{P}(i \to j \mid \theta)$
4) score thing that are spurious
5) others in see Lecture Notes.
6) ...

» costs : models often expensive (global fitting)

②

To optimize or to sample ?



optimize : fast EM (no guarantee of best model)

sample : MCMC / gibbs slow, but guaranteed to find global

($\lim_{t \to \infty}$)

" simulated annealing "

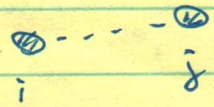
The likelihood of the SBM

$$\Theta = \left(\{z\},\ M \right)$$

Given a choice of Θ, the probability of generating G is $\mathbb{P}(G \mid \Theta)$

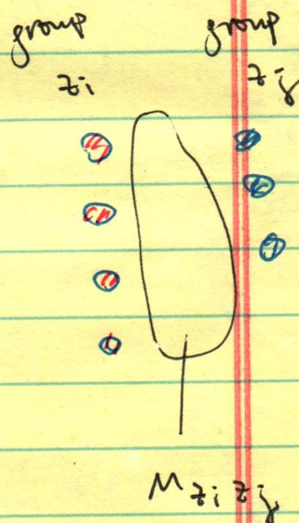
(called "likelihood")

$$\mathcal{L}(G \mid z, M) = \prod_{ij} \mathbb{P}(i \to j \mid z, M)$$



$$= \left(\prod_{(i,j) \in E} \mathbb{P}(i \to j \mid z, M) \right) \left(\prod_{(i,j) \notin E} \left(1 - \mathbb{P}(i \to j \mid z, M)\right) \right)$$

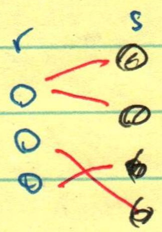
$$= \prod_{(i,j)} M_{z_i z_j} \prod_{(i,j) \notin E} \left(1 - M_{z_i z_j}\right)$$



» How many terms are there in the equation?

Observation: Lots of these terms are identical, and can be combined.

For example:



$$n_{rs} = n_r \cdot n_s \quad (\text{if } r \neq s)$$

$$\binom{n_r}{2} \quad (\text{if } r = s)$$

4

why is $\arg\max_\theta \mathcal{L} = \arg\max_\theta \log \mathcal{L}$?

Auxiliary variables

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

counts : # edges from $r \to s$

whereas n_{rs} counts # "possible" edges $r \to s$.

now :

$$\mathcal{L}(G \mid z, M) = \prod_{r,s} \left(M_{rs}\right)^{e_{rs}} \left(1 - M_{rs}\right)^{n_{rs} - e_{rs}} .$$

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

} The maximum likelihood choice

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

- note : the SBM likelihood is a bunch of Bernoulli trials (aka coin flips).
- hence $\hat{M}_{rs} = e_{rs} / n_{rs}$
- plugin & take logs.

$$\log \mathcal{L} = \sum_{rs} e_{rs}\left(\log\left(\frac{e_{rs}}{n_{rs}}\right)\right) + (n_{rs} - e_{rs}) \log\left(\frac{n_{rs} - e_{rs}}{n_{rs}}\right)$$

Highlight :

subject to mild "regularity" conditions,
the max. like. estimate is asymptotically consistent.
i.e., $\lim_{n \to \infty} \hat{\theta} = \theta^*$.

⑤

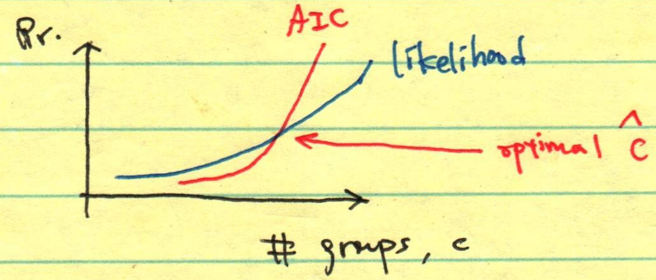
» See Lecture Notes for examples.

» DC-SBM (see notes, we skipped!)

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

⇝ in so far, <u>SBM</u> and <u>DC-SBM</u> both have fixed c inside $\mathcal{L}$.

⇝ General idea: Use "regularization" to learn the best c.



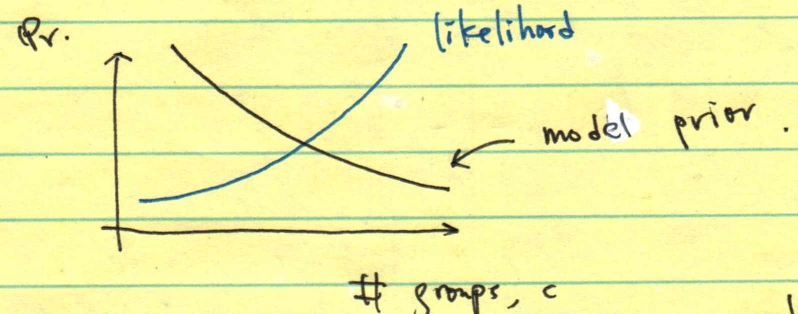
Akaike information criterion.

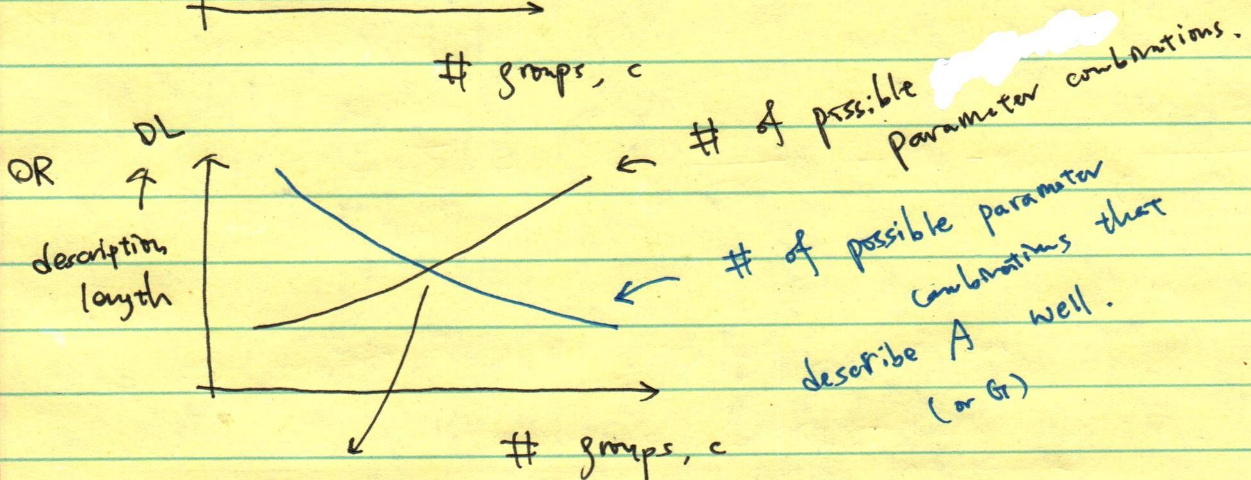
Other methods include: BIC, MDL.

LRT (likelihood ratio tests)

Bayes Factors

" We will go into some details by using <u>MDL</u> "





MDL estimate $\Sigma = S + K$

(-1)

$$\Sigma = S + K$$

$$S = -\log_2 P(A \mid k, e, b)$$

bits to precisely describe the network.

$$K = -\log_2 P(k, e, b)$$

bits to necessary describe the model parameters

"finding the choise of parameters that best compresses the data"

>> Minimum Description Length approach.

$\updownarrow$ equivalence! "occam's razor"

Bayesian inference

7 ①

Lecture 8b

March 7, 2024

§ Microcanonical degree-corrected SBM | Peixoto 2017

parameters :

partition $\mathbb{b} = \{b_i\}$ of N nodes into B groups.

group membership $b_i \in \{1, 2, \dots, B\}$ of node i

degree sequence $\mathbb{k} = \{k_i\}$

matrix of edge counts $e = \{e_{rs}\}$.

edges between groups r and s

($e_{rr} \equiv$ twice # edges inside group r).

Assume that :

1) we generate networks as in the configuration model.

2) half-edges are distinguishable.

$$\Omega(e) = \frac{\prod e_r!}{\prod_{r<s} e_{rs}! \prod_r e_{rr}!!}$$

$$e_r = \sum_s e_{rs} \quad \text{and} \quad (2m)!! = 2^m m!$$

However, many different pairings correspond to the same graph.

Given adj mat A,

$$\Xi(A) = \frac{\prod_i k_i!}{\prod_{i<j} A_{ij}! \prod_i A_{ii}!!}$$

Therefore,

$$\Pr(A \mid k, e, b) = \frac{1}{\left(\begin{array}{c}\text{\# of parameter combinations} \\ \text{that correspond to the observed} \\ \text{graph } A\end{array}\right)}$$

$$= \frac{1}{\left(\frac{\Omega(e)}{\Xi(A)}\right)}$$

$$= \frac{\Xi(A)}{\Omega(e)} = \underline{\qquad\qquad}$$

This probability is non-zero only when the graph A meets the hard-constraints, i.e., $e_{rs} = \sum_{ij} A_{ij} \delta_{b_i,r} \delta_{b_j,s}$.

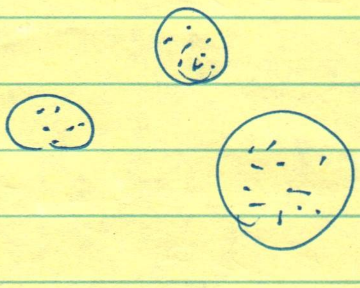
$k_i = \sum_j A_{ij}$.

3

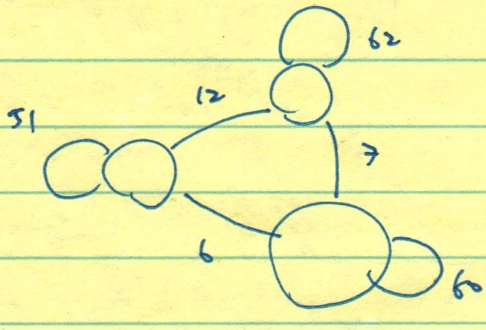
Peixoto (2017) || Python graph-tool

Priors — what do we do with the $\mathbb{P}(e, k, b)$?

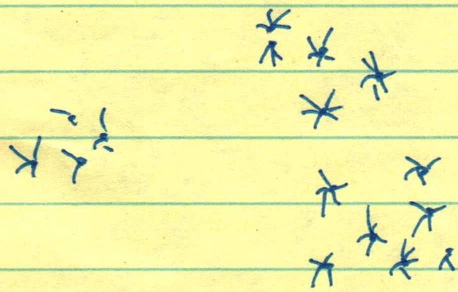
Notice how we generate an SBM ...



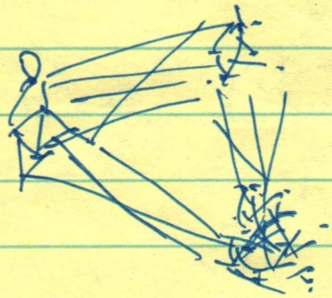
node partitions, $\mathbb{P}(b)$



edge counts, $\mathbb{P}(e|b)$



degrees, $\mathbb{P}(k|e, b)$



network, $\mathbb{P}(A|k, e, b)$

$$\mathbb{P}(e, k, b) = \mathbb{P}(b)\,\mathbb{P}(e|b)\,\mathbb{P}(k|e, b)$$

"generative model for parameters"

Guideline:

1) We want "uniform" priors ;
2) We want it uniform in a way that's still "flexible" (thru, possibly, auxiliary hyperpriors)

>> choice for $\mathbb{P}(b) = P(b|B, N)$

① $P(b|B, N) = B^{-N}$

not a good choice

a) assumes that group sizes will be the same

b) "not flexible" — we cannot further compress the data

② $\mathbb{P}(b|n)\,\mathbb{P}(n|B,N)$

group sizes (hyperparameter / hyperprior)

$$= \left(\prod_r n_r! \Big/ N! \right) \left(\left(\!\binom{B}{N}\!\right)^{-1} \right)$$

$\left(\!\binom{n}{m}\!\right) = \binom{n+m-1}{m}$ counts the number of m combinations from a set of size n.

>> better option : we can show it conforms to known expression of the canonical model.

>> small detail : empty group is not allowed.

③ Finally

$$\mathbb{P}(b) = \mathbb{P}(b|n)\,\mathbb{P}(n|B)\,\mathbb{P}(B)$$

$$= \left(\frac{\prod_r n_r!}{N!} \right) \binom{N-1}{B-1}^{-1} \frac{1}{N}.$$

⑤ To Class, we only reached to this page in Lecture 8b; then we used 10 min to explain the three "caveats."

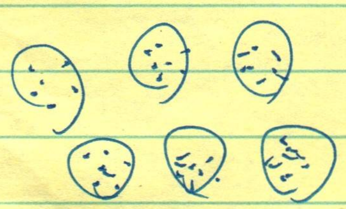
» choice for $P(e|b)$

$$P(e) = \left(\!\!\left(\begin{matrix} \left(\!\binom{B}{2}\!\right) \\ E \end{matrix}\right)\!\!\right)^{-1}$$

counts # symmetric e_{rs} matrices with a constrained sum $\sum_{rs} e_{rs} = 2E$.

Good assumption (uniform), but it limit our capacity to detect small groups in a very large network.

i.e., planted partition ($P_{out} = 0$)



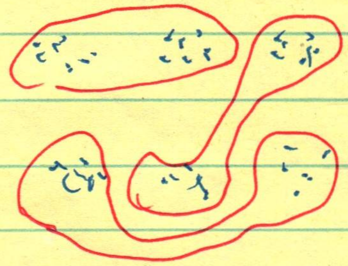
each clique contains ~ 30 nodes

"Resolution limit"

$B_{max} \sim O(\sqrt{N})$

inference

similar problem exists in modularity maximization



Three groups.

⑥

Rain check :

we know how to improve $\mathbb{P}(e|b)$!

but let's see what a complete model would look like.

(also, see Peixoto (2017) for $\mathbb{P}(k|e,b)$)

$$\mathbb{P}(A,k,e,b) = P(A|k,e,b) \cdot P(k|e,b)\, P(e|b) \cdot P(b)$$

$$= \frac{\prod_i k_i! \prod_{r<s} e_{rs}! \prod_r e_{rr}!!}{\prod_r e_r! \prod_{i<j} A_{ij}! \prod_i A_{ii}!!} \Bigg/ \prod_r \frac{\prod_k \eta_k^r!}{n_r!}\, q(e_r, n_r)^{-1}$$

$$\left(\!\!\binom{\left(\!\binom{B}{2}\!\right)}{E}\!\!\right)^{-1} \cdot \frac{\prod_r n_r!}{N!} \binom{N-1}{B-1}^{-1} \frac{1}{N} .$$

η_k^r : # nodes w/ degree k that belong to group r.

$q(m,n)$: # different degree counts with the sum of degrees being exactly m, that have at most n nonzero counts.

" restricted partitions " of integer m into at most n parts.

Daunting ! But — $\mathbb{P}(A,k,e,b) = \overset{k,e}{\mathbb{P}}(A,\underline{b})$

↑

explorable by MCMC !

⑦

3) see also Fig 13 of Peixoto (arXiv: 1705.10225)
(page 30)

Rash check — Second highlight :

$P(e|b)$ can be improved by ... imposing a deeper Bayesian hierarchy !

$$P(e|b) \equiv P(\{e_\ell\}|\{b_\ell\}) = \prod_{\ell=1}^{L} P(e_\ell | e_{\ell+1}, b_\ell) .$$

see

1) graph-tool cook book
"nested SBM"

2) see Fig. 5, 6, 8 of Peixoto (2017)

Finally, three caveats for community detection

1) Competitive local optima.

» exponentially many optima
(but super-exp bad partitions)

» many near-optimal algms exist .

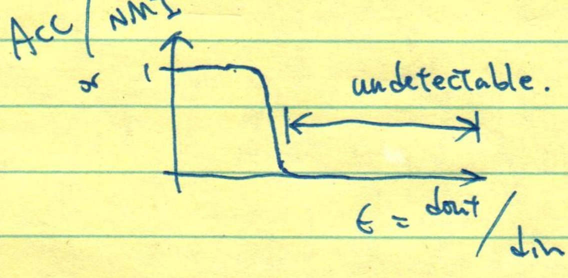
» many run several times, take avg, etc.
/algm
(overlapping communities)

missing :

How to evaluate goodness of node partition ?

⟹ use "mutual / normalized information".
(NMI) .

2) Detectability limits.



planted-2 → but in say planted-5
(richer structure)

See also Fig 1 of Zdeborová – Krzakala (AiP, 2016)

⑧

3) No ground truth & no free lunch (see Lecture Notes)

partition $\Leftrightarrow$ network

$\uparrow$

not a bijection.