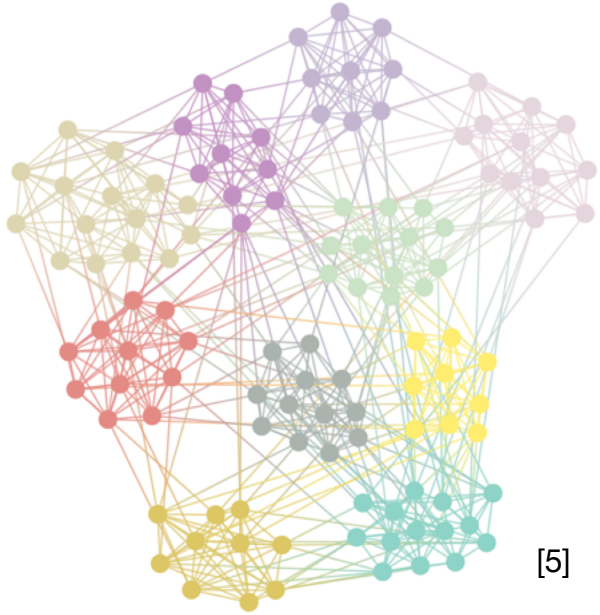
Overview of Peixoto's Monte Carlo Method for Inference of SBMs

SBM THEORY | RICARDO BATISTA

TABLE OF CONTENTS



[5]

EXECUTIVE SUMMARY

EXECUTIVE SUMMARY

We provide an overview of a Markov chain Monte Carlo (MCMC)** [1] **method for inference of stochastic block models (SBMs)**[(1)]**.

Goal

- Determine which partition $\boldsymbol{b} = \{b_i\}$ generated an observed network $\boldsymbol{A}$, assuming this was done via the SBM.
- Moreover, we would like to detect such a partition in complex networks and in a computationally efficient manner.

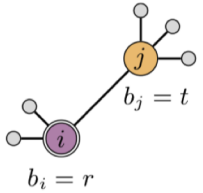
The Algorithm

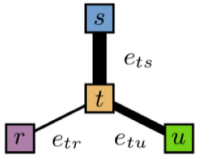
- The MCMC method we present seeks to identify the partition that maximizes the posterior probability $P(\boldsymbol{b}|\boldsymbol{A})$.
- This greedy algorithm has an almost linear $O(N \ln^2 N)$ complexity and works on a wide variety of network setups.

(1) Inference of modular network structure using MCMC methods, among other approximation heuristics, is referred to as a semidefinite programming (SDP) relaxation [2]

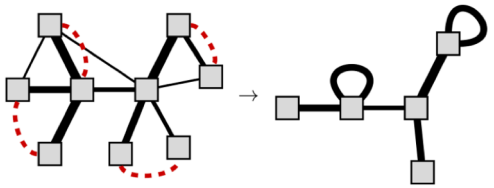
ALGORITHM OVERVIEW

Graph G
(i.e., adjacency matrix A)





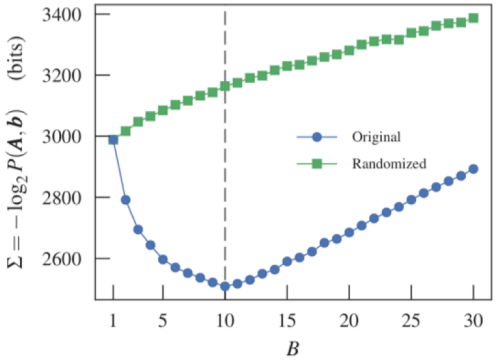
Build a multigraph (i.e., each block is a node) and find the merger that produces the best partition for B blocks where $B < B'$



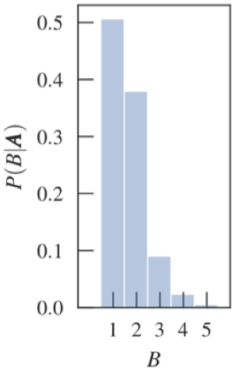
For a given number of blocks B', sample new partitions $\boldsymbol{b}$ by inspecting each node's neighbor until MCMC chain equilibrates

while $B > 0$

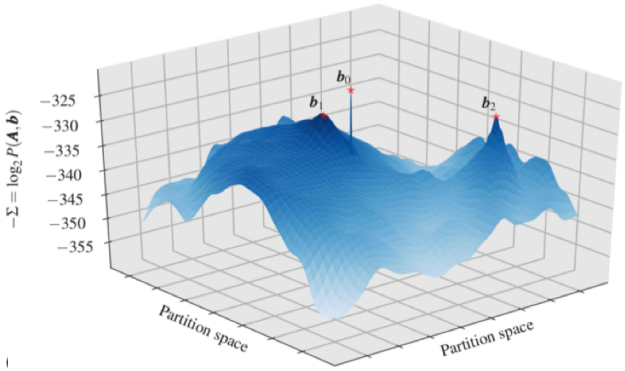
Search for the model corresponding to the B that maximizes the posterior distribution (i.e., smallest loss)



Get another sample from the posterior (until we reach N iterations)



Perform model selection



LIMITATIONS

i. Detectability threshold

Even planted structures cannot be recovered for values of $e = N(\lambda_{in} - \lambda_{out})$ below

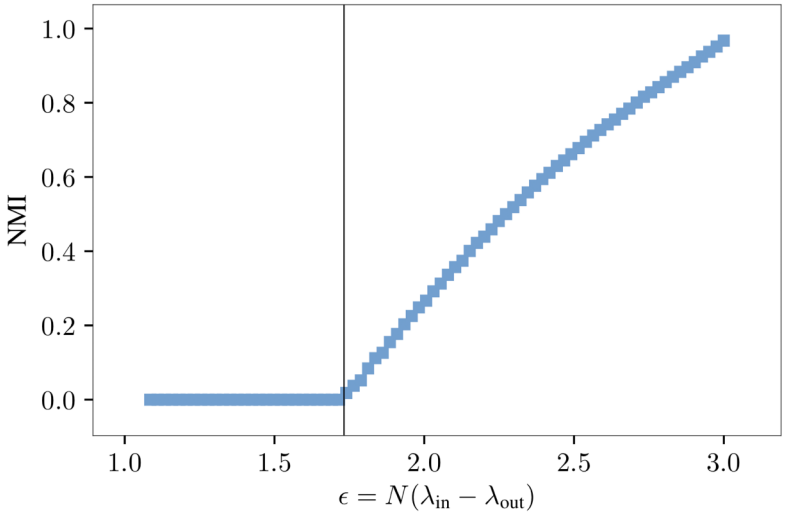
$$e^* = B\sqrt{\langle k \rangle}\,,$$

where λ_{in} and λ_{out} are the expected number of edges between nodes of the same groups and of different groups, respectively [4].

ii. Limit on B

This method is unable to uncover a number of groups that is larger than

$$B_{max} \propto N/\log N \text{ [4].}$$



Normalized mutual information (NMI)(1) between the planted and inferred partitions of a (Planted partition) PP model with $N = 105$, $B = 3$ and $\langle k \rangle = 3$ and $\varepsilon = N(\lambda_{in} - \lambda_{out})$. The vertical line marks the detectability threshold $\varepsilon^* = B\langle k \rangle$ [4].

(1) Normalized mutual information (NMI) is defined as $2I(X,Y)/(H(X) + H(Y))$, where $I(X,Y)$ is the mutual information between X and Y, and $H(X)$ is the entropy of X [3]. It is a measure of inference accuracy

EXAMPLES

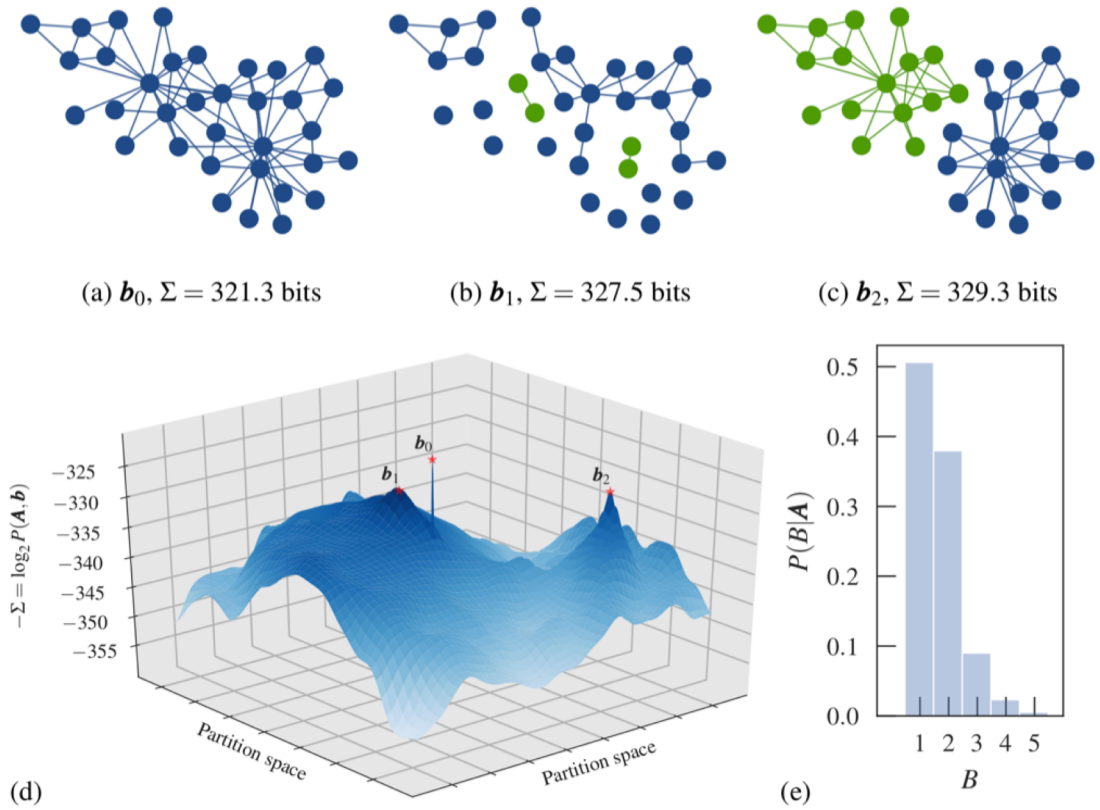
The following are meant to highlight (1) core challenges in SBM inference inherent both in network complexity and inference methods, as well as (2) mechanisms to meet said demands.

PRELIMINARIES

- The algorithm is implemented in the graph-tool library (a Python module). The package also produced the following visualizations.

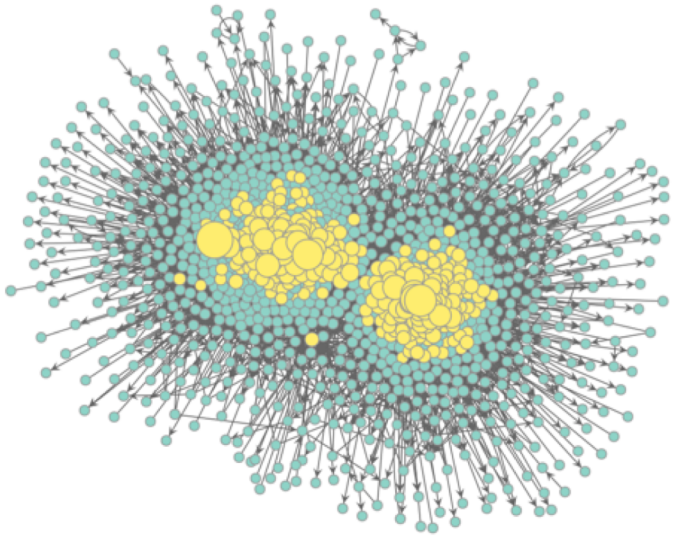
Key terms

- **Description length** is defined as $\Sigma = -\log_2 P(\boldsymbol{A}, \boldsymbol{b})$. Selecting the partition with the minimum description length (MDL) is equivalent to selecting the partition with the largest posterior probability. (For more, see slide.)
- **Degree-corrected** SBM (DC-SBM) is defined just like the traditional model but considers degree homogeneity among members of a same group. (For more, see slide.)

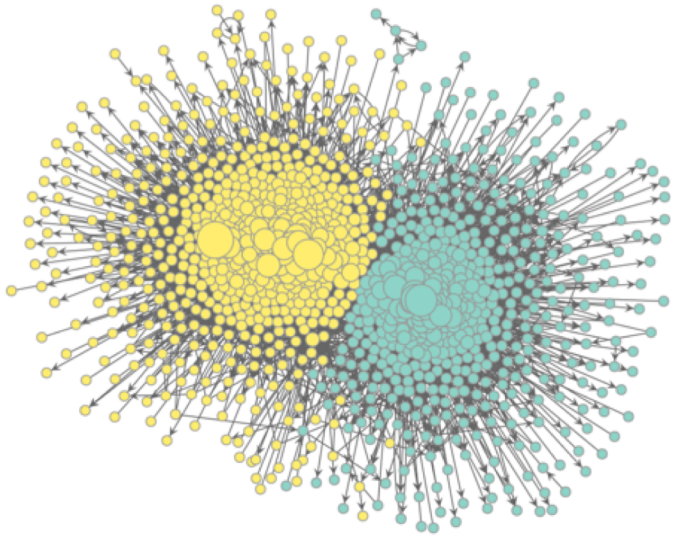


EXAMPLE | MULTI-MODAL POSTERIOR DISTRIBUTION

Posterior distribution of partitions of Zachary's karate club network using the degree-corrected SBM (DC-SBM). Panels (a) to (c) show three modes of the distribution and their respective description lengths[1] ; (d) 2D projection of the posterior obtained using multidimensional scaling [89]; (e) Marginal posterior distribution of the number of groups B [4].



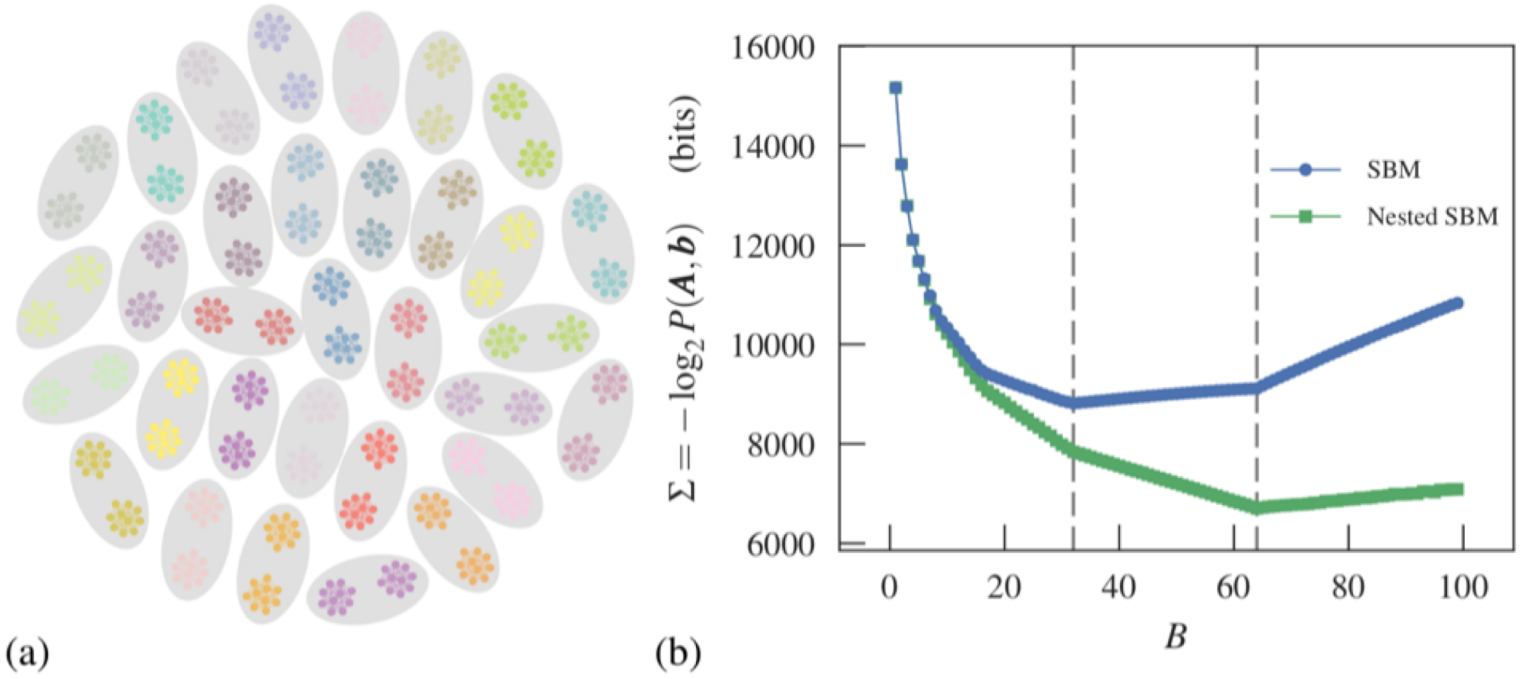
(a)



(b)

EXAMPLE | HETEROGENEOUS DEGREES WITHIN A GROUP

Inferred partition for a network of political blogs [61] using (a) the SBM and (b) the DC-SBM, in both cases forcing B = 2 groups. The node sizes are proportional to the node degrees. The SBM divides the network into low and high-degree groups, whereas the DC-SBM prefers the division into political factions [4].



EXAMPLE | RESOLUTION LIMIT

Inference of the SBM on a simple artificial network composed of 64 cliques of size 10, illustrating the underfitting problem: (a) The partition that maximizes the posterior probability of Eq. 10, or equivalently, minimizes the description length of Eq. 25. The 64 cliques are grouped into 32 groups composed of two cliques each. (b) Minimum description length as a function of the number of groups of the corresponding partition, both for the SBM and its nested variant, which is less susceptible to underfitting, and puts all 64 cliques in their own groups [4].

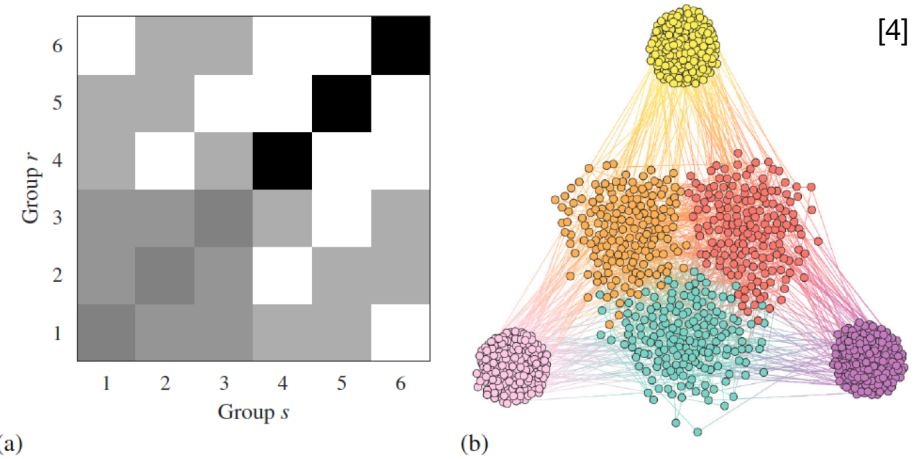
THEORY

GENERATIVE PROCESS

TRADITIONAL SBM

The matrix of probabilities between groups p_{rs} defines the large-scale structure of generated networks

- Let A be an undirected graph (i.e., symmetric adjacency matrix) with N nodes
- Each node belongs to a group (i.e., cluster). That is,
  - node i has group membership $b_i \in \{1, \dots, B\}$ and,
  - vector $\boldsymbol{b}$ represents a partition of said network
- The probability that a member of group r is connected to a member in group s is p_{rs}



[4]

Figure 2. The stochastic block model (SBM): (a) The matrix of probabilities between groups p_{rs} defines the large-scale structure of generated networks; (b) a sampled network corresponding to (a), where the node colors indicate the group membership.

COMPLEX SBM

- The following slides cover more complex structures of networks, all of which graph-tool can handle
- We will emphasize those structures observed often in empirical networks, namely
  - Degree heterogeneity among nodes of a same group
  - Nested networks and/or very small groups
- Accounting for said complexity typically improves inference significantly, as you may recall from examples 1 and 2

COMPLEX SBM

Degree-corrected SBM

- The underlying assumption of the traditional generative process is that all nodes that belong to the same group receive on average the same number of edges [4]
- As it turns out, this fundamental aspect results in a very unrealistic property (i.e., this is often a poor model for many networks) [4]
- A better model is called the ***degree-corrected*** SBM, and it is defined just like the traditional model, with the addition of the degree sequence

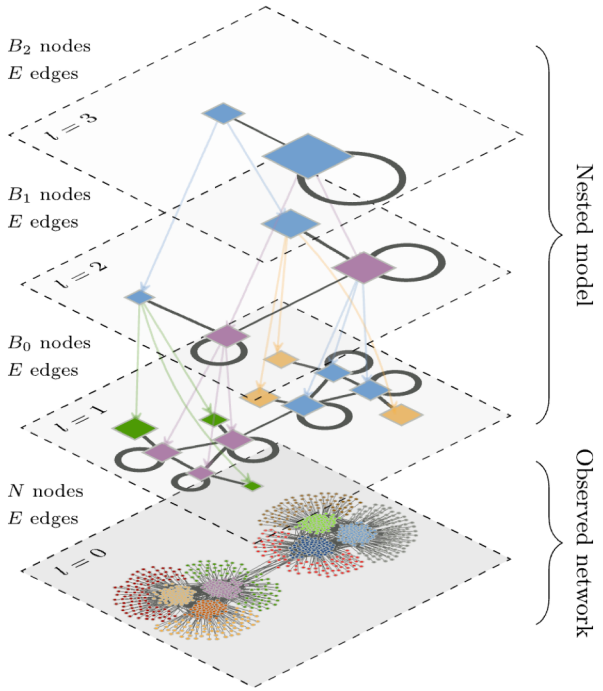
$$\boldsymbol{k} = \{k_i\}$$

of the graph as an additional set of parameters [4]

COMPLEX SBM

Nested network

- Systematic underfitting (i.e., not finding relatively small groups) is observed for a wide variety of network datasets when using the regular SBM [4]
- This underfitting often disappears with the nested model [4]
- In a nested SBM, the groups themselves are clustered into groups [4]

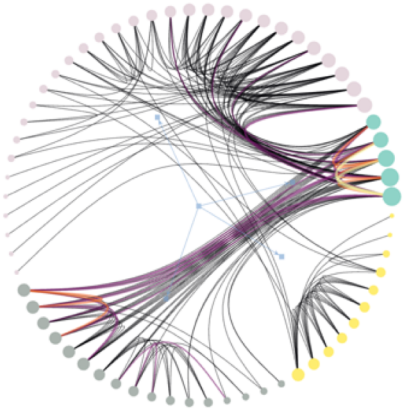


Example of a nested SBM with three levels. [4]

COMPLEX SBM

Edge weights

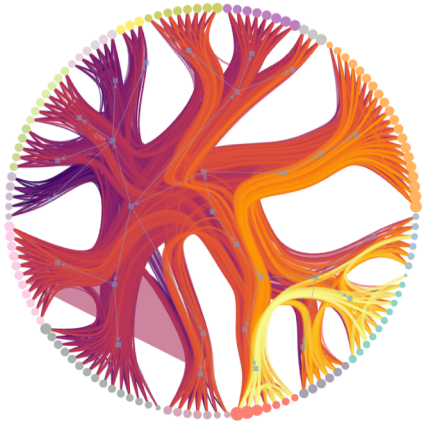
Very often networks cannot be completely represented by simple graphs, but instead have arbitrary "weights" xij on the edges.



Best fit of the Binomial-weighted degree-corrected SBM for a network of terror suspects, using the strength of connection as edge covariates. The edge colors and widths correspond to the strengths.

Directed edges

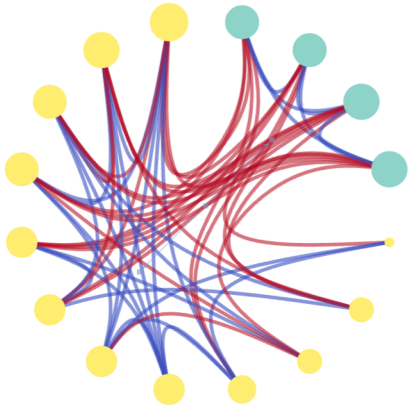
Example: A directed link exists from species i to j if a biomass flow exists between them, and a weight x_{ij} on this edge indicates the magnitude of biomass flow.



Best fit of the exponential-weighted degree-corrected SBM for a food web, using the biomass flow as edge covariates (indicated by the edge colors and widths).

Layered networks [5]

The edges of the network may be distributed in discrete "**layers**", representing distinct types of interactions

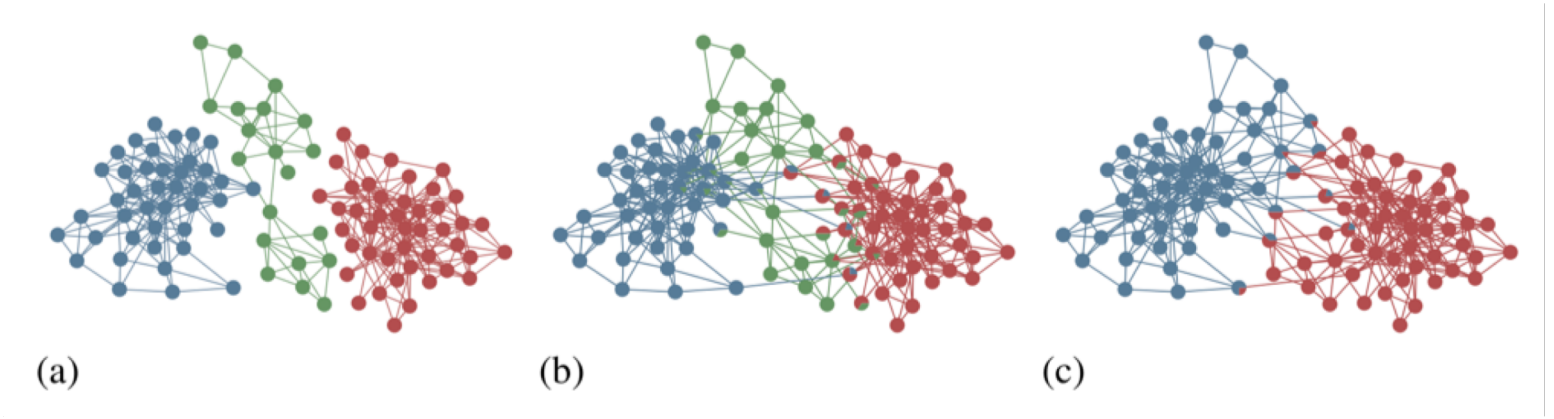


Best fit of the DC-SBM with edge layers for a network of tribes, with edge layers shown as colors. The groups show two enemy tribes.

COMPLEX SBM

Group overlaps

Another way we can change the internal structure of the model is to allow the groups to overlap, i.e. we allow a node to belong to more than one group at the same time [4].



Network of co-purchase of books about US politics [66], with groups inferred using (a) the non- overlapping DC-SBM, with description length $\Sigma \approx 1,938$ bits, (b) the overlapping SBM with description length $\Sigma \approx 1,931$ bits and (c) the overlapping SBM forcing only $B = 2$ groups, with description length $\Sigma \approx 1,946$ bits [4].

THEORY

INFERENCE

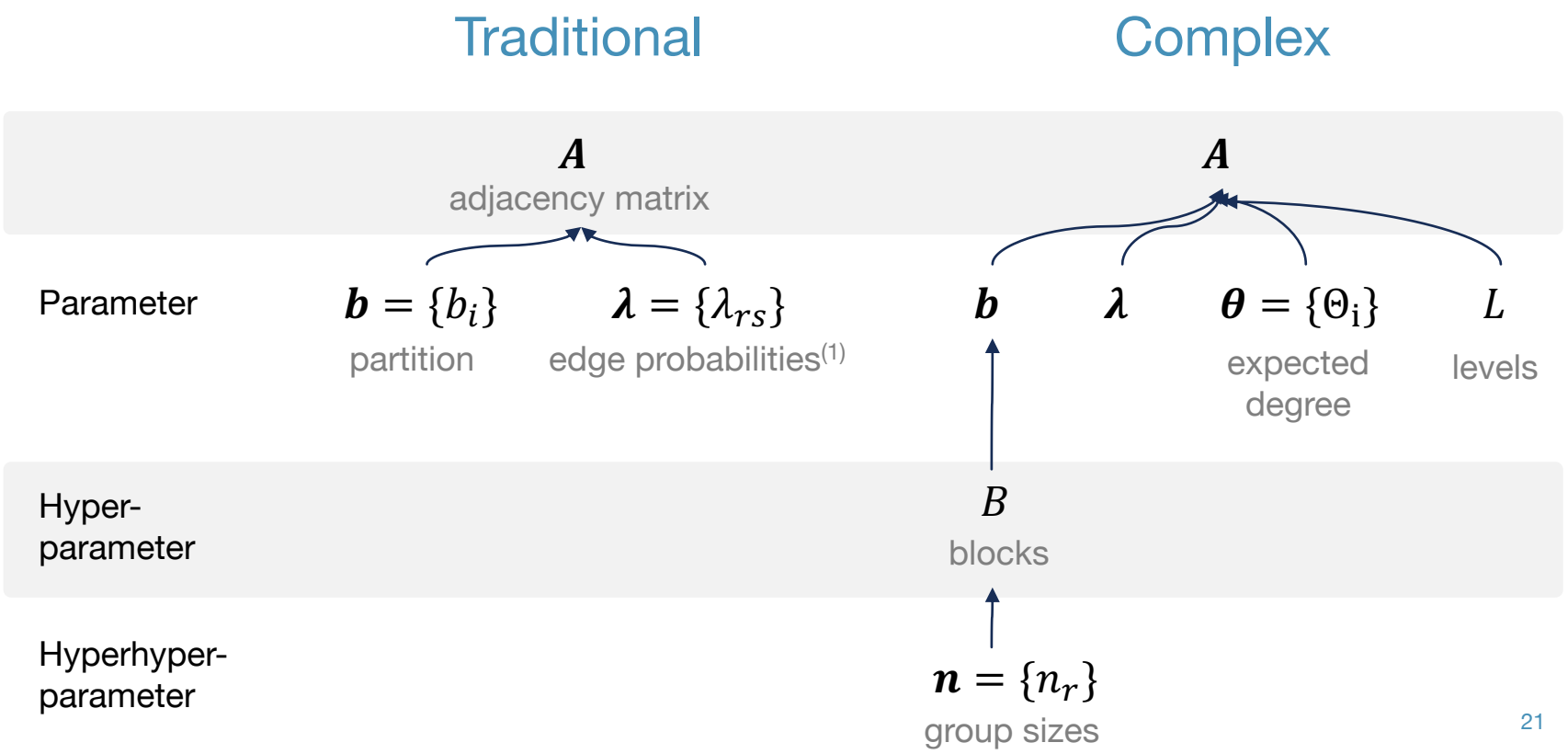
PRELIMINARIES

- We split this section in two:
  - **Canonical**: traditional Bayesian interpretation of the SBM
  - **Microcanonical**: The term "microcanonical" — borrowed from statistical physics — means that model parameters correspond to "hard" constraints that are strictly imposed on the ensemble, as opposed to "soft" constraints that are obeyed only on average. [4]

- Why microcanonical?
  - ***Canonical and microcanonical cannot be distinguished from data(1), since their marginal likelihoods (and hence the posterior probability) are identical [4]***
  - The algorithm uses the microcanonical interpretation (i.e., when doing MCMC, we do not primarily sample from the priors in the canonical model) since it's more powerful (in many respects). More on this later.

(1) at least for the basic priors we use [4]

CANONICAL | PARAMETER DIAGRAM



(1) For clarification, λ is the vector of probabilities of an edge existing between any two nodes belonging to group r and s, respectively.

CANONICAL | DISTRIBUTIONS

Traditional

$$P(\boldsymbol{b}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b})}{P(\boldsymbol{A})}$$

Posterior distribution. Probability $P(b|A)$ that a node partition b was responsible for a network A

$$P(\boldsymbol{A}|\boldsymbol{b}) = \int P(\boldsymbol{A}|\boldsymbol{\lambda},\boldsymbol{b})P(\boldsymbol{\lambda}|\boldsymbol{b})\,\mathrm{d}\boldsymbol{\lambda}$$

marginal likelihood integrated over the remaining model parameters

$$P(\boldsymbol{b}) = \frac{1}{\sum_{\boldsymbol{b}'} 1} = \frac{1}{a_N}$$

"flat" distribution where all partitions into at most B = N groups are equally likely, where a_N are the ordered Bell numbers [4]

$$a_N = \sum_{B=1}^{N} \left\{ {N \atop B} \right\} B!$$

where $\left\{ {N \atop B} \right\}$ are the Stirling numbers of the second kind [4]

$$P(\boldsymbol{\lambda}|\boldsymbol{b}) = \prod_{r \le s} e^{-n_r n_s \lambda_{rs}/(1+\delta_{rs})\bar{\lambda}} n_r n_s/(1+\delta_{rs})\bar{\lambda}$$

- $\bar{\lambda} = 2E\,/B(B+1)$ is the expected total number of edges
- $\langle \lambda_{rs} \rangle = \bar{\lambda}(1+\delta_{rs})/n_r n_S$, is local average such that that the expected number of edges $e_{rs} = \lambda_{rs} n_r n_s/(1+\delta_{rs})$ will be equal to $\bar{\lambda}$, irrespective of the group sizes n_r and n_s [4]

$$P(\boldsymbol{A}|\boldsymbol{b}) = \frac{\bar{\lambda}^E}{(\bar{\lambda}+1)^{E+B(B+1)/2}} \times \frac{\prod_{r<s} e_{rs}! \prod_r e_{rr}!!}{\prod_r n_r^{e_r} \prod_{i<j} A_{ij}! \prod_i A_{ii}!!}$$

CANONICAL | DISTRIBUTIONS – P($\boldsymbol{b}$)

Complex

$$P(\boldsymbol{b}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b})}{P(\boldsymbol{A})}$$

Posterior distribution. Probability $P(b|A)$ that a node partition b was responsible for a network A

$$P(B) = 1/N_r$$

$$P(\boldsymbol{n}|B) = \binom{N-1}{B-1}^{-1}$$

since $N-1$ is the number of ways to divide N nonzero counts into B nonempty bins [4]

$$P(\boldsymbol{b}|\boldsymbol{n}) = \frac{\prod_r n_r!}{N!}.$$

Given the randomly sampled sizes as a constraint, we sample the partition randomly [4]

$$P(\boldsymbol{b}) = P(\boldsymbol{b}|\boldsymbol{n})P(\boldsymbol{n}|B)P(B) = \frac{\prod_r n_r!}{N!}\binom{N-1}{B-1}^{-1} N^{-1}$$

CANONICAL | DISTRIBUTIONS – P($A|b$)

Complex

$$P(\boldsymbol{b}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b})}{P(\boldsymbol{A})}$$

Posterior distribution. Probability $P(b|A)$ that a node partition b was responsible for a network A

Each node i is attributed with a parameter θ_i that controls its expected degree, independently of its group membership. [4]

$$P(\boldsymbol{A}|\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{b}) = \prod_{i<j} \frac{e^{-\theta_i\theta_j\lambda_{b_i,b_j}}(\theta_i\theta_j\lambda_{b_i,b_j})^{A_{ij}}}{A_{ij}!} \times \prod_i \frac{e^{-\theta_i^2\lambda_{b_i,b_i}/2}(\theta_i^2\lambda_{b_i,b_i}/2)^{A_{ii}/2}}{(A_{ii}/2)!}$$

We us uninformative priors for both the node propensities θ and for λ [4]

$$P(\boldsymbol{\lambda}|\boldsymbol{b}) = \prod_{r\leq s} e^{-\lambda_{rs}/(1+\delta_{rs})\bar{\lambda}}/(1+\delta_{rs})\bar{\lambda} \qquad P(\boldsymbol{\theta}|\boldsymbol{b}) = \prod_r (n_r-1)!\delta(\textstyle\sum_i \theta_i\delta_{b_i,r}-1)$$

$$P(\boldsymbol{A}|\boldsymbol{b}) = \int P(\boldsymbol{A}|\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{b})P(\boldsymbol{\lambda}|\boldsymbol{b})P(\boldsymbol{\theta}|\boldsymbol{b})\,\mathrm{d}\boldsymbol{\lambda}\mathrm{d}\boldsymbol{\theta}$$

$$= \frac{\bar{\lambda}^E}{(\bar{\lambda}+1)^{E+B(B+1)/2}} \times \frac{\prod_{r<s} e_{rs}!\prod_r e_{rr}!!}{\prod_{i<j} A_{ij}!\prod_i A_{ii}!!} \times \prod_r \frac{(n_r-1)!}{(e_r+n_r-1)!} \times \prod_i k_i!$$

MICROCANONICAL | OVERVIEW

- In this section we will
  1. Describe why Peixoto likes it
  2. Define microcanonical
  3. Show the distributions for the traditional SBM using the microcanonical model
  4. Draw connection to canonical
  5. Introduce the statistics behind the algorithm

WHY MICROCANONICAL MODELS FOR SBM?

This approach can be used to sample modular hierarchies from the posterior distribution, as well as to perform model selection. It allows simultaneously for two important improvements over more traditional inference approaches:

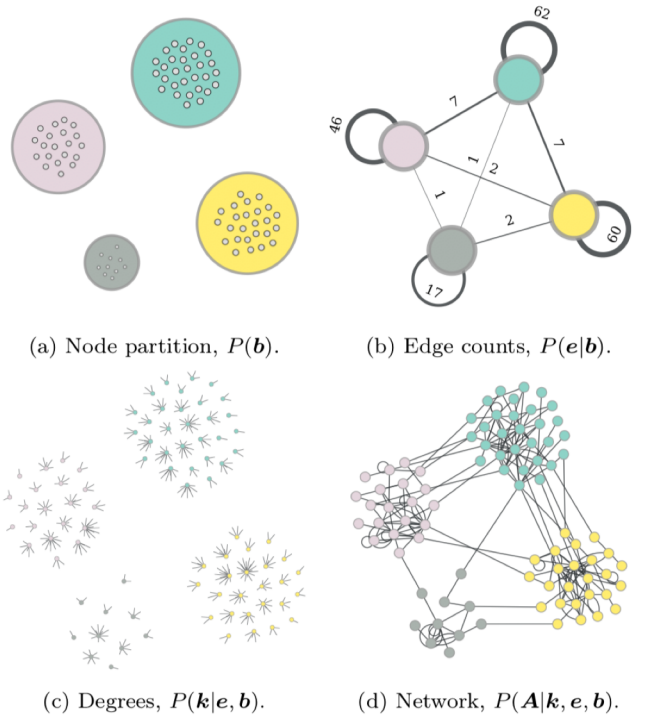
1. Deeper Bayesian hierarchies, with noninformative priors replaced by sequences of priors and hyperpriors, that not only remove limitations that seriously degrade the inference on large networks, but also reveal structures at multiple scales;
2. A very efficient inference algorithm that scales well not only for networks with a large number of nodes and edges, but also with an unlimited number of modules [7].

MICROCANONICAL MODELS

- We can re-interpret the integrated marginal likelihood as the joint likelihood of a microcanonical model given by

$$P(\boldsymbol{A}|\boldsymbol{b}) = P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b})$$

- The term "**microcanonical**" means that model parameters correspond to "hard" constraints that are strictly imposed on the ensemble of graphs, as opposed to "soft" constraints that are obeyed only on average [4].
- In this particular case, $P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})$ is the probability of generating a multigraph A where the total number of edges between groups r and s is always exactly e_{rs} without any fluctuation allowed between samples [4]



(a) Node partition, $P(\boldsymbol{b})$. (b) Edge counts, $P(\boldsymbol{e}|\boldsymbol{b})$.

(c) Degrees, $P(\boldsymbol{k}|\boldsymbol{e},\boldsymbol{b})$. (d) Network, $P(\boldsymbol{A}|\boldsymbol{k},\boldsymbol{e},\boldsymbol{b})$.

Figure 1. Illustration of the complete nonparametric generative process for the DC-SBM considered in this work. First the partition of the nodes is sampled (a), followed by the edge counts between groups (b), the degrees of the nodes (c) and finally the network itself (d).

MICROCANONICAL | DISTRIBUTIONS

Microcanonical | Traditional

$$P(\boldsymbol{b}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b})}{P(\boldsymbol{A})}$$

Posterior distribution. Probability $P(b|A)$ that a node partition b was responsible for a network A

$$P(\boldsymbol{A}|\boldsymbol{b}) = P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b})$$

- $\boldsymbol{e} = \{e_{rs}\}$ is the matrix of edge counts between groups
- $e_{rs} = \sum_{ij} A_{ij}\delta_{b_i,r}\delta_{b_j,s}$

  where δ , I presume, is the delta function

$$P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b}) = \frac{\prod_{r<s} e_{rs}! \prod_r e_{rr}!!}{\prod_r n_r^{e_r} \prod_{i<j} A_{ij}! \prod_i A_{ii}!!},$$

$$P(\boldsymbol{e}|\boldsymbol{b}) = \prod_{r<s} \frac{\bar{\lambda}^{e_{rs}}}{(\bar{\lambda}+1)^{e_{rs}+1}} \prod_r \frac{\bar{\lambda}^{e_{rs}/2}}{(\bar{\lambda}+1)^{e_{rs}/2+1}} = \frac{\bar{\lambda}^{E}}{(\bar{\lambda}+1)^{E+B(B+1)/2}}$$

- *E is edges*
- Recall that $\bar{\lambda} = 2E/B(B+1)$ is the expected total number of edges

MICROCANONICAL VS CANONICAL

- If we wish to impose that nodes that belong to the same group are statistically indistinguishable, our ensemble of networks (i.e., the networks A given a partition b) should be fully characterized by the number of edges that connects nodes of two groups r and s [4],

$$e_{rs} = \sum_{ij} A_{ij} \delta_{b_i,r} \delta_{b_j,s} \qquad (1)$$

  where δ (I think) is the delta function (i.e., acts as an indicator function).

- If we relax somewhat our requirements, such that Eq. (1) is obeyed only on expectation, and if we assume that the placement of edges are conditionally independent,

$$P(\boldsymbol{A}|\boldsymbol{b}) = \prod_{i \leq j} P_{ij}(A_{ij}),$$

  Then we obtain the setup for the the canonical formulation of the SBM model.

Important: do not confuse the probability that the edge ij exists P_{ij} with the average number of edges existing between any two nodes belonging to group r and s $\boldsymbol{p} = \{p_{rs}\} = \boldsymbol{\lambda} = \{\lambda_{rs}\}$.

MICROCANONICAL VS CANONICAL

- We can re-interpret the integrated marginal likelihood as the joint likelihood of a microcanonical model given by

$$P(\boldsymbol{A}|\boldsymbol{b}) = P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b})$$

where $\boldsymbol{e} = \{e_{rs}\}$ is the matrix of edge counts between groups [4].

- So $P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})$ is the probability of generating a multigraph $\boldsymbol{A}$ where Eq. (1) is always fulfilled, i.e. the total number of edges between groups r and s is always exactly e_{rs} without any fluctuation allowed between samples.
- This contrasts with the parameter λ_{rs}, which determines only the average number of edges between groups, which fluctuates between samples [4].

MICROCANONICAL VS CANONICAL

Canonical and microcanonical cannot be distinguished from data(1), since their marginal likelihoods (and hence the posterior probability) are identical [4]

- Notice that this equation $P(\boldsymbol{A}|\boldsymbol{b}) = P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b})$ does not contain the sum $P(\boldsymbol{A}|\boldsymbol{b}) = \sum_{\boldsymbol{e}} P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e}|\boldsymbol{b})$.
  - Indeed, that is the proper way to write a marginal likelihood.
  - However, for the microcanonical model there is only one element of the sum that fulfills the constraint of equation (1) and thus yields a nonzero probability, making the marginal likelihood identical to the joint. The same is true for the partition prior $P(\boldsymbol{b})$ [4].

- Conversely, the prior for the edge counts $P(\boldsymbol{e}|\boldsymbol{b})$ is a mixture of geometric distributions with average $\bar{\lambda}$, which *does allow the edge counts to fluctuate*, guaranteeing the overall **equivalence** (between canonical and microcanocial) [4].

(1) at least for the basic priors we use [4]

DESCRIPTION LENGTH

With the microcanonical interpretation in mind, we may frame the posterior probability as follows:

- If a variable x occurs with a probability mass $P(x)$, the amount of information necessary to describe it is $-\log_2 P(x)$. Thus, we may write

$$P(\boldsymbol{A}|\boldsymbol{b})P(\boldsymbol{b}) = P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})P(\boldsymbol{e},\boldsymbol{b}) = 2^{-\Sigma}$$

where

$$\begin{aligned}\Sigma &= -\log_2 P(\boldsymbol{A},\boldsymbol{e},\boldsymbol{b}) \\ &= -\log_2 P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b}) - \log_2 P(\boldsymbol{e},\boldsymbol{b})\end{aligned}$$

- is called the ***description length*** of the data. It corresponds to the amount of information necessary to encode the data $\boldsymbol{A}$ together with the model parameters $\boldsymbol{e}$ and $\boldsymbol{b}$.
- Therefore, if we find a network partition that maximizes the posterior distribution, we are also automatically finding one which minimizes the description length. [4]

BIAS-VARIANCE TRADEOFF

With this, we can see how the Bayesian approach just outlined prevents overfitting: As the size of the model *increases* (via a larger number of occupied groups),

- it will constrain itself better to the data, and the amount of information necessary to describe it when the model is known, $-\log_2 P(\boldsymbol{A}|\boldsymbol{e},\boldsymbol{b})$, will *decrease*.
- At the same time, the amount of information necessary to describe the model itself, $-\log_2 P(\boldsymbol{e},\boldsymbol{b})$, will *increase* as it becomes more complex.

Therefore, the latter will function as a penalty that prevents the model from becoming overly complex, and the optimal choice will amount to a proper balance between both terms. Among other things, this approach will allow us to properly estimate the dimension of the model — represented by the number of groups B — in an efficient way [4].

ENTROPY AND MODEL SIZE

- Description length more commonly goes by $\Sigma = \mathcal{L} + S$ where
  - $\mathcal{L} = -\log_2 P(\boldsymbol{e}, \boldsymbol{b})$ is the amount of information necessary to fully describe the model, and
  - $S = -\log_2 P(\boldsymbol{A}|\boldsymbol{e}, \boldsymbol{b})$ corresponds to **entropy** of the lowest level $l = 0$ of the hierarchy.

- Notice that although minimizing S allows one to find the most likely partition into B blocks, it cannot be used to find the best value of B itself. This is because the minimum of S is a strictly decreasing function of B, since larger models can always incorporate more details of the observed data, providing a better fit [4].

ALGORITHM

For point estimate that maximizes posterior distribution

ALGORITHM

For point estimate that maximizes posterior distribution

The pseudocode in the next slide outlines the main features of the following function:

`graph_tool.inference.`**`minimize_blockmodel_dl`**(*g, B_min=None, B_max=None, b_min=None, b_max=None, deg_corr=True, overlap=False, nonoverlap_init=True, layers=False, state_args={}, bisection_args={}, mcmc_args={}, anneal_args={}, mcmc_equilibrate_args={}, shrink_args={}, mcmc_multilevel_args={}, verbose=False*) [source] [5]

Fit the stochastic block model.

Parameters:

g : **Graph**

The graph.

B_min : `int` (optional, default: `None`)

The minimum number of blocks.

B_max : `int` (optional, default: `None`)

The maximum number of blocks.

b_min : **PropertyMap** (optional, default: `None`)

The partition to be used with the minimum number of blocks.

b_max : **PropertyMap** (optional, default: `None`)

The partition to be used with the maximum number of blocks.

deg_corr : `bool` (optional, default: `True`)

If `True`, the degree-corrected version of the model will be used.

overlap : `bool` (optional, default: `False`)

If `True`, the overlapping version of the model will be used.

nonoverlap_init : `bool` (optional, default: `True`)

If `True`, and `overlap == True` a non-overlapping initial state will be used.

layers : `bool` (optional, default: `False`)

If `True`, the layered version of the model will be used.

state_args : `dict` (optional, default: `{}`)

Arguments to be passed to appropriate state constructor (e.g. **BlockState**, **OverlapBlockState** or **LayeredBlockState**)

bisection_args : `dict` (optional, default: `{}`)

Arguments to be passed to **bisection_minimize()**.

mcmc_args : `dict` (optional, default: `{}`)

Arguments to be passed to **graph_tool.inference.BlockState.mcmc_sweep()**, **graph_tool.inference.OverlapBlockState.mcmc_sweep()** or **graph_tool.inference.LayeredBlockState.mcmc_sweep()**.

mcmc_equilibrate_args : `dict` (optional, default: `{}`)

Arguments to be passed to **mcmc_equilibrate()**.

shrink_args : `dict` (optional, default: `{}`)

Arguments to be passed to **graph_tool.inference.BlockState.shrink()**, **graph_tool.inference.OverlapBlockState.shrink()** or **graph_tool.inference.LayeredBlockState.shrink()**.

mcmc_multilevel_args : `dict` (optional, default: `{}`)

Arguments to be passed to **mcmc_multilevel()**.

verbose : `bool` or `tuple` (optional, default: `False`)

If `True`, progress information will be shown. Optionally, this accepts arguments of the type `tuple` of the form `(level, prefix)` where `level` is a positive integer that specifies the level of detail, and `prefix` is a string that is prepended to the all output messages.

Returns:

min_state : **BlockState** or **OverlapBlockState** or **LayeredBlockState**

State with minimal description length.

ALGORITHM | PSEUDOCODE

For point estimate that maximizes posterior distribution

1. We're given a graph G (i.e., adjacency matrix); set B = N
2. Initialize the graph's partition vector G.b equal to b_0 (since B = N, each node is in its own block)
3. While B > 0
4. While chain has not equilibrated
5. For each node n_i in G:
6. Metropolis Hastings Routine 1 — attempt to change n_i's membership (i.e., b_i)
7. Calculate the description length for b' (and save b' if it has the lowest so far)
8. Let B' = B; B = B/σ
9. Build a multigraph: the B' blocks themselves are nodes
10. For j ∈ {1, … B'}
11. For k ∈ {1,…, n_m}
12. Metropolis Hastings Routine 2 — attempt to merge block j and block s ∈ {1, …, B}
13. Keep track of the best merger (based on description length)
14. Select the best $B' - B$ merges to obtain the desired partition into B blocks — update b accordingly and save b
15. Model selection: having calculated the optimal b for each B, we select the one with minimum description length

Agglomerative step
Obtain the best partition from a larger partition B' > B

METROPOLIS HASTINGS ROUTINE 1

- Given a value of B, directly obtaining the partition $\{bi\}$ which minimizes description length is in general not tractable, since it requires testing all possible partitions [1].

- Instead one must rely on approximate, or stochastic procedures
  - The MCMC approach consists in modifying the block membership of each node in a random fashion and accepting or rejecting each move with a probability given as a function of the entropy difference ΔS
  - The simplest approach one can take is to attempt to move each vertex into one of the B blocks with equal probability. However, this can be very inefficient [1].

METROPOLIS HASTINGS ROUTINE 1

- A better approach consists in attempting to move a vertex from block r to s with a probability given by

$$p(r \to s|t) = \frac{e_{ts} + \epsilon}{e_t + \epsilon B} \qquad (2)$$

where

- t is the block label of a randomly chosen neighbor, and
- $\varepsilon > 0$ is a free parameter (note that by making $\varepsilon \to \infty$ we recover the fully random moves described in the previous slide) [1]

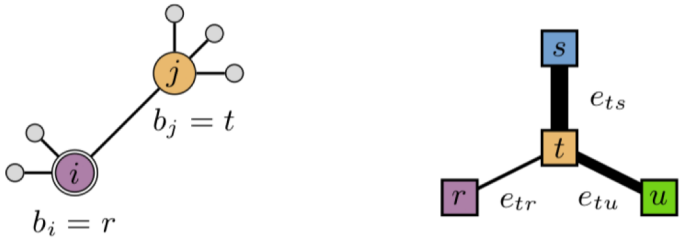


FIG. 1. *Left:* Local neighborhood of node i belonging to block r, and a randomly chosen neighbor j belonging to block t. *Right:* Block multigraph, indicating the number of edges between blocks, represented as the edge thickness. In this example, the attempted move $b_i \to s$ is made with a larger probability than either $b_i \to u$ or $b_i \to r$ (no movement), since $e_{ts} > e_{tu}$ and $e_{ts} > e_{tr}$. [1]

- The Eq. (2) above means that we attempt to guess the block membership of a given node by inspecting the block membership of its **neighbors** and by using the **currently inferred model parameters** to choose the most likely blocks to which the original node belongs (see Fig. 1) [1].
- It should be observed that this move imposes **no inherent bias**; in particular, it does not attempt to find assortative structures in preference to any other, since it depends fully on the matrix e_{rs} currently inferred [1].

METROPOLIS HASTINGS ROUTINE 1

The moves with probabilities given by Eq. (1) can be implemented efficiently. We simply write $p(r \to s|t) = (1 - Rt)e_{ts}/e_t + R_t/B$, with $R_t = \varepsilon B/(e_t + \varepsilon B)$ [1].

1. Sample s

i. A random neighbor j of the node i being moved is selected, and its block membership $t = b_j$ is obtained;

ii. The value s is randomly selected from all B choices with equal probability;

iii. With probability R_t it is accepted;

iv. If it is rejected, a randomly chosen edge adjacent to block t is selected, and the block label s is taken from its opposite endpoint [1].

2. Accept move with probability a

$$a = \min\left\{e^{-\beta\Delta\mathcal{S}_{t/c}}\frac{\sum_t p_t^i p(s \to r|t)}{\sum_t p_t^i p(r \to s|t)}, 1\right\}$$

where

- p_t^i is the fraction of neighbors of node i which belong to block t, and
- $p(s \to r|t)$ is computed after the proposed $r \to s$ move (i.e., with the new values of e_{rt}), whereas $p(r \to s|t)$ is computed before.
- The parameter β in Eq. 4 is an inverse temperature, which can be used to escape local minima [1]

AGGLOMERATIVE STEP

In order to avoid the metastable states, we attempt to find the best configuration for some B′ > B, and then use that configuration to obtain a better estimate for one with B blocks *[1]*

1. We implement this by constructing a block (multi)graph, where the blocks themselves are the nodes (weighted by the block sizes) and the edge counts e_{rs} are the edge multiplicities between each block node [1].

   In this representation, a block merge is simply a block membership move of a block node, where initially each node is in its own block [1].

2. The choice of moves is done with same probability as before, i.e. via Eq. (1). In order to select the best merges, we attempt n_m moves for each block node, and collectively rank the best moves for all nodes according to ΔS. From this global ranking, we select the best $B' - B$ merges to obtain the desired partition into B blocks [1].

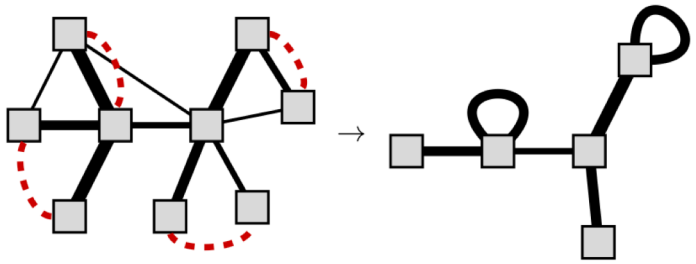


FIG. 5. Representation of the block merges used in the agglomerative heuristic. Each square node is a block in the original graph, and the merges (represented as red dashed lines) correspond simply to block membership moves. [1]

OBTAIN THE BEST VALUE OF B

Having obtained the minimum of S for each B, we simply pick the model with the lowest description length $(\sum_b)$ [3]

- Instead of description length, we could also consider BIC or AIC [6]
- Efficient search for the best model:

  the best value of B is obtained via an independent one-dimensional minimization of $\sum_b$ (we could use) using a Fibonacci search based on subsequent bisections of an initial interval which brackets the minimum [3]

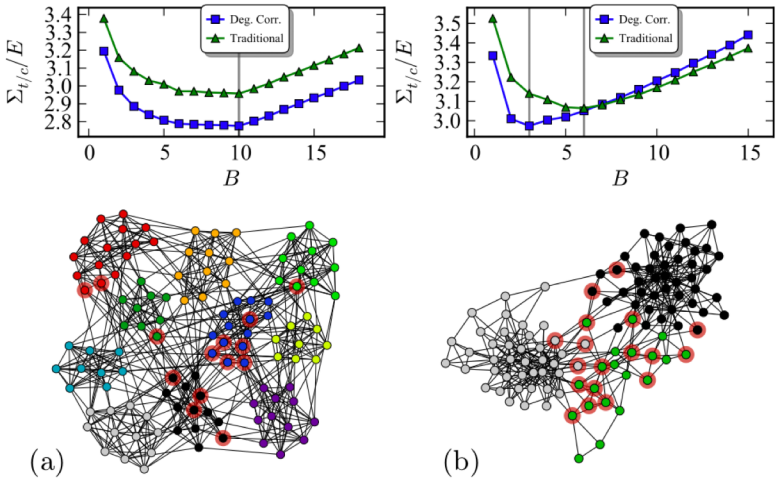
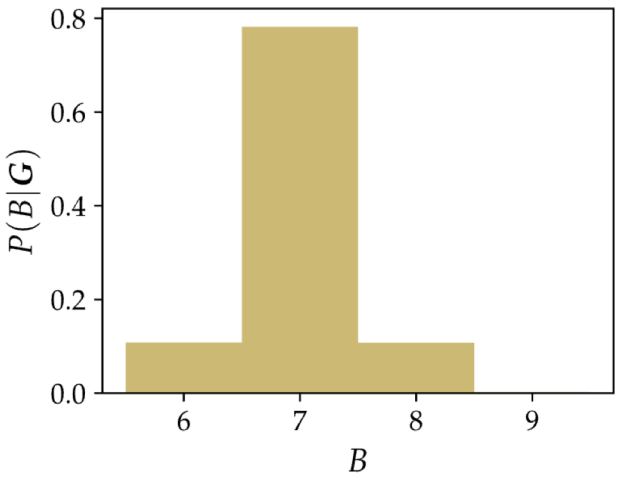


FIG. 3. *Top:* Value of Σ_b/E for both blockmodel variants as a function of B for (a) the American football network of [47] (with the corrections described in [48, 49]) and (b) the political books network of [50]. *Bottom:* Inferred partitions with the smallest Σ_b. Nodes circled in red do not match the known partitions. [3]

MODEL SELECTION

- Since the inference algorithm is stochastic by nature, we will benefit from running it many times and inspecting the resulting empirical posterior distribution [5]
- In particular we interested in evaluating which model classes (i.e., models with a different internal structure and set of parameters) provide a better fit to the data
- To this end we calculate, for instance, the marginal posterior probability of the number of groups (see right)
- This type of analysis helps us determine whether we should
  - select the partition with the largest posterior probability, or
  - average over may alternative fits [5].



Marginal posterior probability of the number of nonempty groups for the network of characters in the novel Les Misérables, according to the degree-corrected SBM.

Go back to the first example for an empirical instance of this bias-variance tradeoff

RESULTS

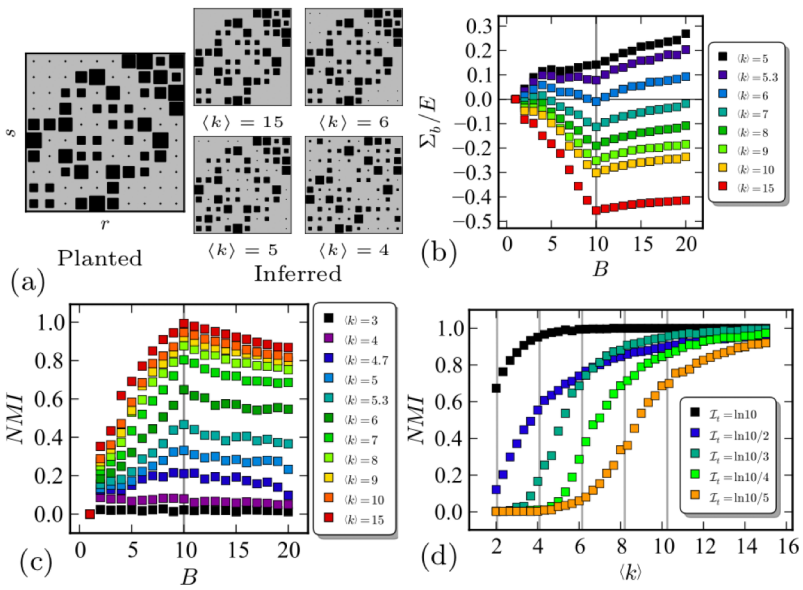


FIG. 1. (a) Prescribed block structure with $B = 10$ and $\mathcal{I}_t = \ln B/6$, together with inferred parameters for different $\langle k\rangle$; (b) Description length Σ_b/E for different B and $\langle k\rangle$, for networks sampled from (a). The vertical line marks the position of the global minimum; (c) NMI between the true and inferred partitions, for the same networks as in (b); (d) Same as (b) for different $\langle k\rangle$ and prescribed block structures. The grey lines correspond to the threshold of Eq. 7. In all cases we have $N = 10^4$. [3]

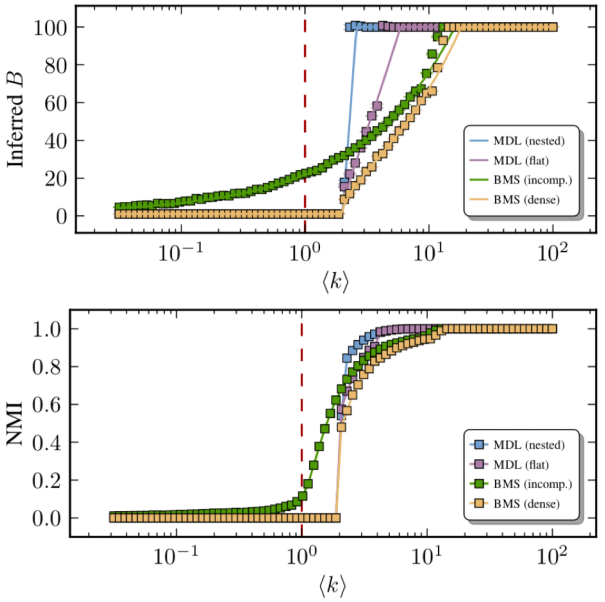


Figure 2. Model selection results for a PP model with $N = 10^4$, $B_{\text{true}} = 100$ and fully isolated blocks ($c = 1$), using the model selection criteria described in the text. The top panel shows the inferred value of B versus the average degree $\langle k\rangle$ in the network. The solid lines show the theoretical value according to each criterion, and the data points are direct optimization of the corresponding quantities for actual generated network, averaged over 40 independent realizations. The bottom panel shows the normalized mutual information (NMI) between the inferred and planted partitions. The dashed line marks the threshold $\langle k\rangle = 1$ where inference becomes impossible for $N \to \infty$. [6]

GRAPH DENSITY SENSITIVITY ANALYSIS

Planted partition (PP) Models

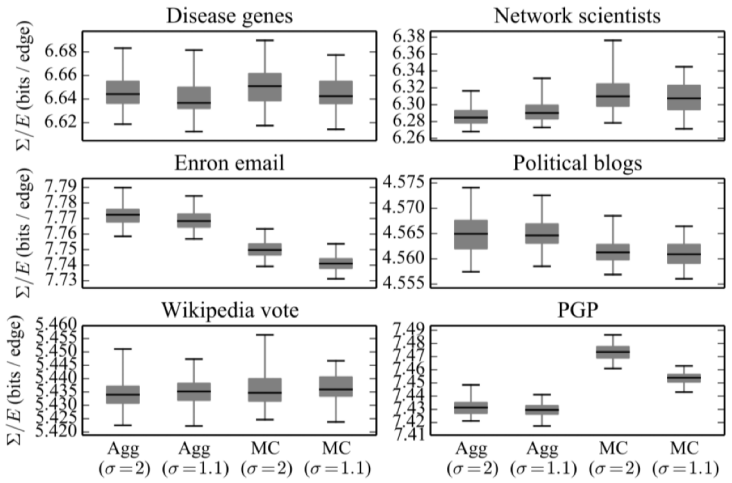


FIG. 8. Description length Σ for different empirical networks, collected for 100 independent runs of the MCMC algorithm (MC) and the agglomerative heuristic (Agg), for different agglomeration ratios σ.

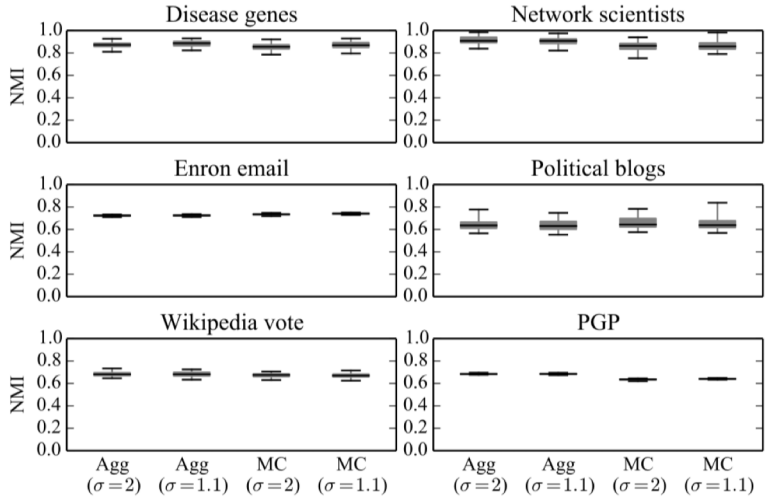
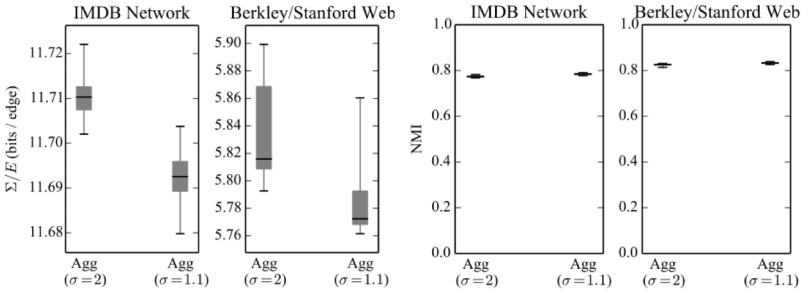


FIG. 9. Normalized mutual information (NMI) between the best overall partition and each one collected for 100 independent runs of the MCMC algorithm (MC) and the agglomerative heuristic (Agg), for different agglomeration ratios σ.

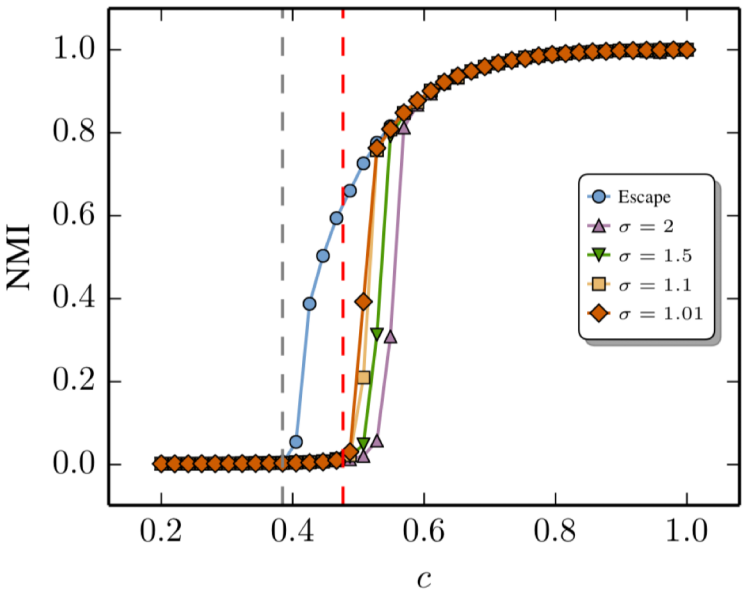


[1]

FIG. 10. Description length Σ for different empirical networks, as well the Normalized mutual information (NMI) between the best overall partition and each one, collected for 100 independent runs of the agglomerative heuristic, for different agglomeration ratios σ.

AGGLOMERATION RATIO SENSITIVITY ANALYSIS

Empirical Networks



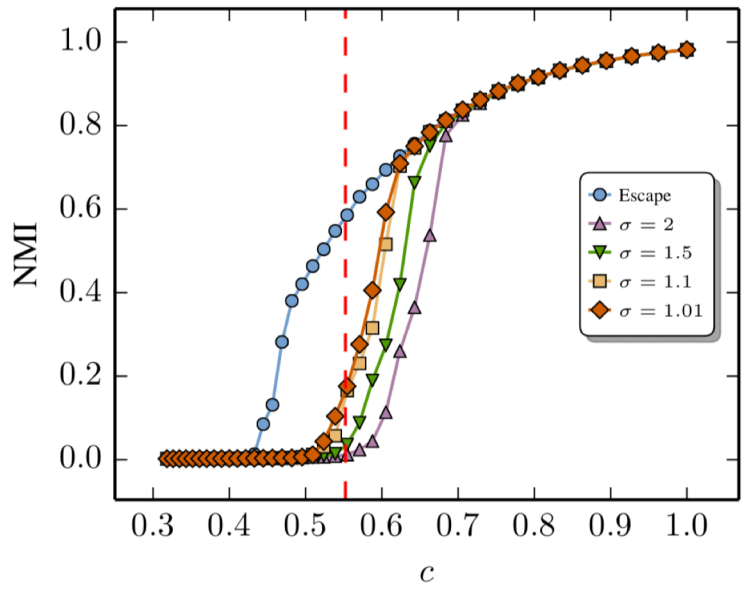


FIG. 7. Normalized mutual information (NMI) (see footnote 6) between the planted and the inferred partitions for (top) the PP model and (bottom) the circular multipartite model described in the text, as a function of the modular strength c, for $N = 10^4$ and $B = 10$. The "Escape" curves correspond to MCMC equilibrations starting from the planted partition, and the remaining curves to the greedy agglomerative heuristic with ratio σ shown in the legend, and $n_m = 10$. All curves are averaged over 20 independent network realizations. The grey vertical dashed line corresponds to the detectability threshold c^* for the PP model, and the red dashed line to the MDL model selection threshold of Eq. 7. [1]

MODULAR STRENGTH SENSITIVITY ANALYSIS

PP models

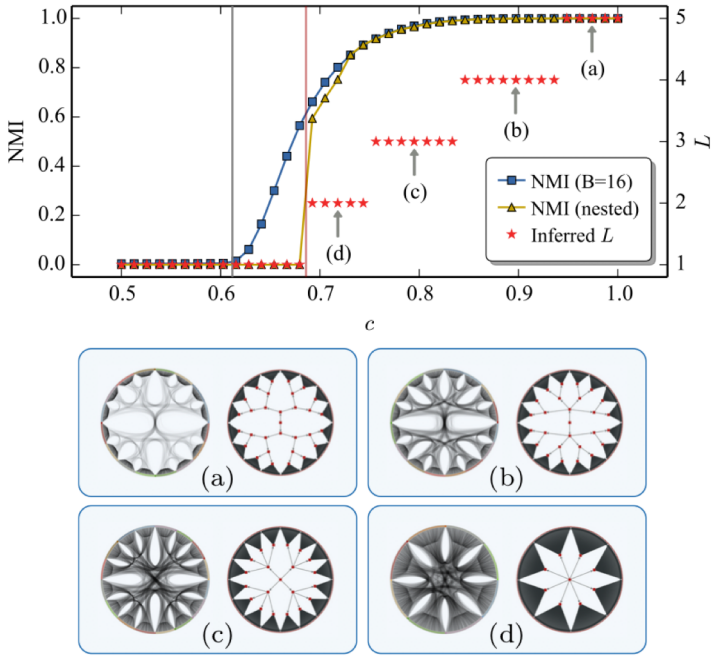


Figure 4. Top: Normalized mutual information (NMI) between the inferred and true partitions for network realizations of the nested PP model described in the text with $B_1 = 2$, $L = 5$, $\langle k \rangle = 20$ and $N = 10^4$, as a function of the assortativity strength c, both via the standard stochastic block model with $B = 16$, and the nested variant with unspecified parameters. The star symbols ($\star$) show the value of L for the inferred hierarchy. All points are averaged over 20 independent realizations. The gray vertical line marks the detectability threshold c^* when B is predetermined, and the red line when the nested model fails to detect any structure. Bottom: Example hierarchies inferred for the values of c indicated in the top panel. The left image shows the network realization itself, and the right one the hierarchical structure [the planted hierarchy corresponds to the one in (a)]. [6]

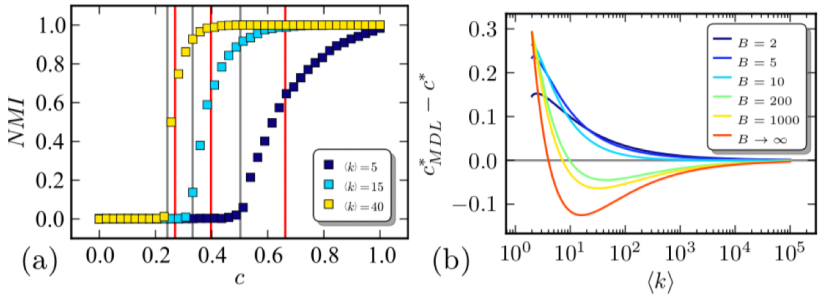


FIG. 2. (a) NMI between the true and inferred partitions for PP samples with $B = 10$ as a function of c for different $\langle k \rangle$. The grey (red) lines correspond to the threshold c^* of Ref. [17] (c^*_{MDL} given by Eq. 7); (b) Difference between c^*_{MDL} and c^*, for different $\langle k \rangle$ and B.

[3]

ASSORTATIVITY SENSITIVITY ANALYSIS

PP models

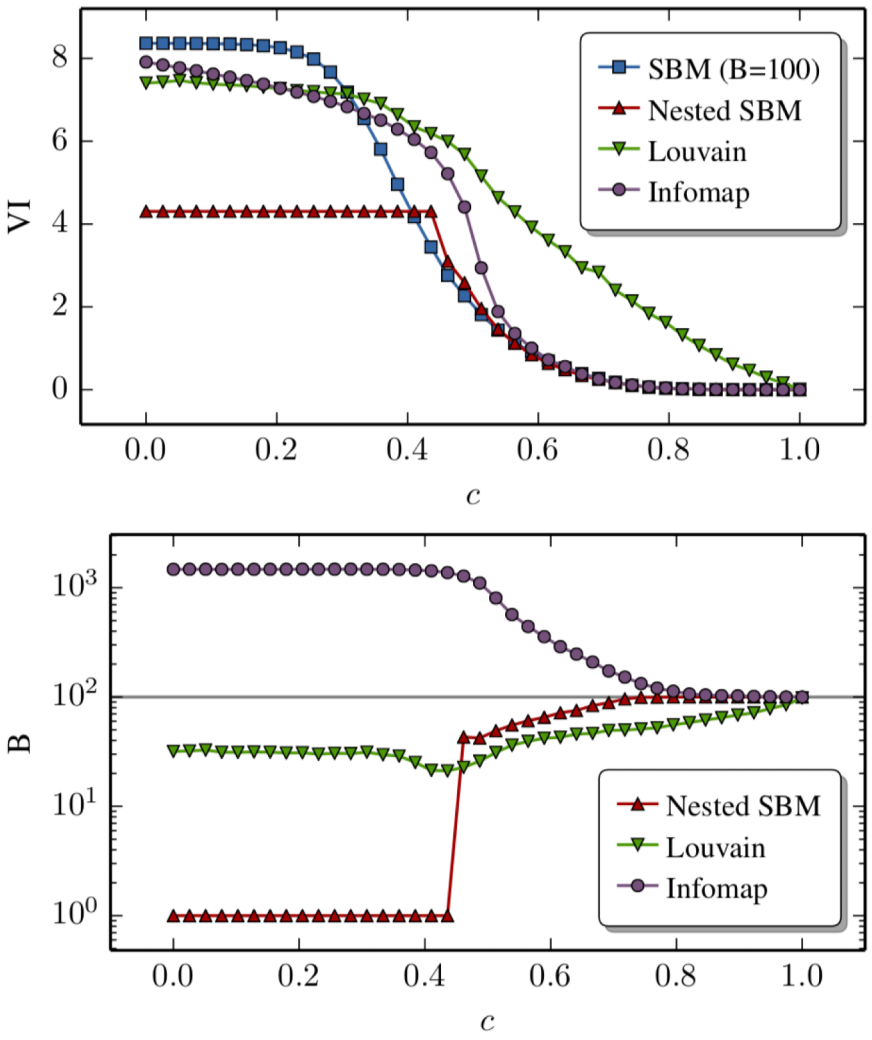


Figure 9. *Top:* Variation of information (VI) between the planted and obtained partitions as a function of the assortativity parameter c, for networks with $N = 2 \times 10^4$, generated as described in the text. The legend indicates results obtained with different methods: Fitting the degree-corrected stochastic block model with a fixed number of blocks $B = 100$ (SBM), performing model selection with the nested stochastic block model (Nested SBM), the Louvain modularity maximization method [12], and the Infomod method [45, 92, 93]. *Bottom:* The obtained number of blocks B as a function of c, for the same methods as in the top panel. The gray horizontal line marks the planted $B = 100$ value. All results were obtained by averaging over 20 network realizations. [6]

METHOD COMPARISON

PP models

APPENDIX

MICROCANONICAL | DISTRIBUTIONS

Microcanonical | Complex

NESTED and NOT DEGREE-CORRECTED

$$P(\{\boldsymbol{b}_l\}|\boldsymbol{A}) = \frac{P(\boldsymbol{A},\{\boldsymbol{b}_l\})}{P(\boldsymbol{A})} = \frac{P(\boldsymbol{A},\{\boldsymbol{e}_l\},\{\boldsymbol{b}_l\})}{P(\boldsymbol{A})}$$

Posterior distribution of the hierarchical partition

We can treat the hierarchy depth L as a latent variable as well, by placing a prior on it $P(L) = 1/L_{max}$, where L_{max} is the maximum value allowed. But since this only contributes to an *overall multiplicative constant* it has no effect on the posterior distribution, and **thus can be omitted**.

$$P(\boldsymbol{e}_l|\boldsymbol{b}_{l-1},\boldsymbol{e}_{l+1},\boldsymbol{b}_l) = \prod_{r<s}\left(\!\!\binom{n_r^l n_s^l}{e_{rs}^{l+1}}\!\!\right)^{-1}\prod_r\left(\!\!\binom{n_r^l(n_r^l+1)/2}{e_{rs}^{l+1}/2}\!\!\right)^{-1}$$

This is the likelihood of a maximum-entropy multigraph SBM, i.e. every multigraph occurs with the same probability, provided they fulfill the imposed constraints

$$P(\boldsymbol{b}_l) = \frac{\prod_r n_r^l!}{B_{l-1}!}\binom{B_{l-1}-1}{B_l-1}^{-1}B_{l-1}^{-1}$$

Same as in the L = 1 case

$$P(\boldsymbol{A},\{\boldsymbol{e}_l\},\{\boldsymbol{b}_l\}|L) = P(\boldsymbol{A}|\boldsymbol{e}_1,\boldsymbol{b}_0)P(\boldsymbol{b}_0)\prod_{l=1}^{L}P(\boldsymbol{e}_l|\boldsymbol{b}_{l-1},\boldsymbol{e}_{l+1},\boldsymbol{b}_l)P(\boldsymbol{b}_l)$$

The joint probability of the data, edge counts and the hierarchical partition $\{\boldsymbol{b}_l\}$

ENTROPY CALCULATION

Entropy

Traditional

$$\mathcal{S}_t = \frac{1}{2}\sum_{rs} n_r n_s H_{\mathrm{b}}\left(\frac{e_{rs}}{n_r n_s}\right)$$

Degree-corrected

$$\mathcal{S}_c \simeq -E - \sum_k N_k \ln k! - \frac{1}{2}\sum_{rs} e_{rs} \ln\left(\frac{e_{rs}}{e_r e_s}\right)$$

- $E = \sum_{rs} e_{rs}/2$ is the total number of edges,
- N_k is the total number of nodes with degree k,
- $e_r = \sum_s e_{rs}$ is the number of half-edges incident on block r, and
- $H_b\,(x) = -x \ln x - (1-x)\ln(1-x)$ is the binary entropy function [1].

REFERENCES

REFERENCES

- [1] T. P. Peixoto, *Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models*, arXiv:1310.4378 (2014).
- [2] A. Javanmard, A. Montanari, and F. Ricci-Tersenghi, *Phase Transitions in Semidefinite Relaxations*, arXiv:1511.08769 (2016).
- [3] T. P. Peixoto, *Parsimonious module inference in large networks*, arXiv:1212.4794 (2013)
- [4] T. P. Peixoto, *Bayesian stochastic blockmodeling*, arXiv:1705.10225 (2018)
- [5] T. P. Peixoto, graph-tool Documentation, https://graph-tool.skewed.de/static/doc/demos/inference/inference.html
- [6] T. P. Peixoto, *Hierarchical block structures and high-resolution model selection in large networks*, arXiv:1310.4377 (2014)
- [7] T. P. Peixoto, Nonparametric Bayesian inference of the microcanonical stochastic block model, arXiv:1610.02703 (2016)