Link Prediction

CS224W: Social and Information Network Analysis

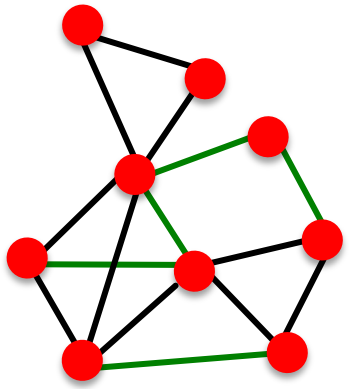
Jure Leskovec, Stanford University

http://cs224w.stanford.edu

[LibenNowell-Kleinberg '03]

Link Prediction in Networks

- **The link prediction task:**
  - Given $G[t_0, t_0']$ a graph on edges up to time t_0', **output a ranked list L** of links (not in $G[t_0, t_0']$) that are predicted to appear in $G[t_1, t_1']$



$G[t_0, t_0']$
$G[t_1, t_1']$

- **Evaluation:**
  - $n = |E_{new}|$: # new edges that appear during the test period $[t_1, t_1']$
  - Take top n elements of L and count correct edges

[LibenNowell-Kleinberg '03]

Link Prediction via Proximity

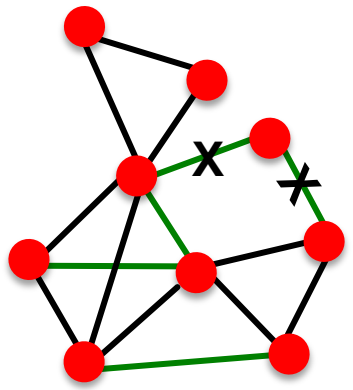
- **Predict links in a evolving collaboration network**

	training period			Core		
	authors	papers	collaborations[1]	authors	$\|E_{old}\|$	$\|E_{new}\|$
astro-ph	5343	5816	41852	1561	6178	5751
cond-mat	5469	6700	19881	1253	1899	1150
gr-qc	2122	3287	5724	486	519	400
hep-ph	5414	10254	47806	1790	6654	3294
hep-th	5241	9498	15842	1438	2311	1576

- **Core:** Because network data is very sparse
  - Consider only nodes with degree of at least 3
    - Because we don't know enough about nodes with less than 3 edges to make good inferences

Link Prediction via Proximity

- **Methodology:**
  - For each pair of nodes *(x,y)* compute score *c(x,y)*
    - For example, *c(x,y)* could be the # of common neighbors of *x* and *y*
  - Sort pairs *(x,y)* by the decreasing score *c(x,y)*
    - **Note:** Only consider/predict edges where both endpoints are in the core (*deg.* ≥ *3*)
  - **Predict top n pairs as new links**
  - **See which of these links actually appear in $G[t_1, t_1']$**



[LibenNowell-Kleinberg '03]

Link Prediction via Proximity

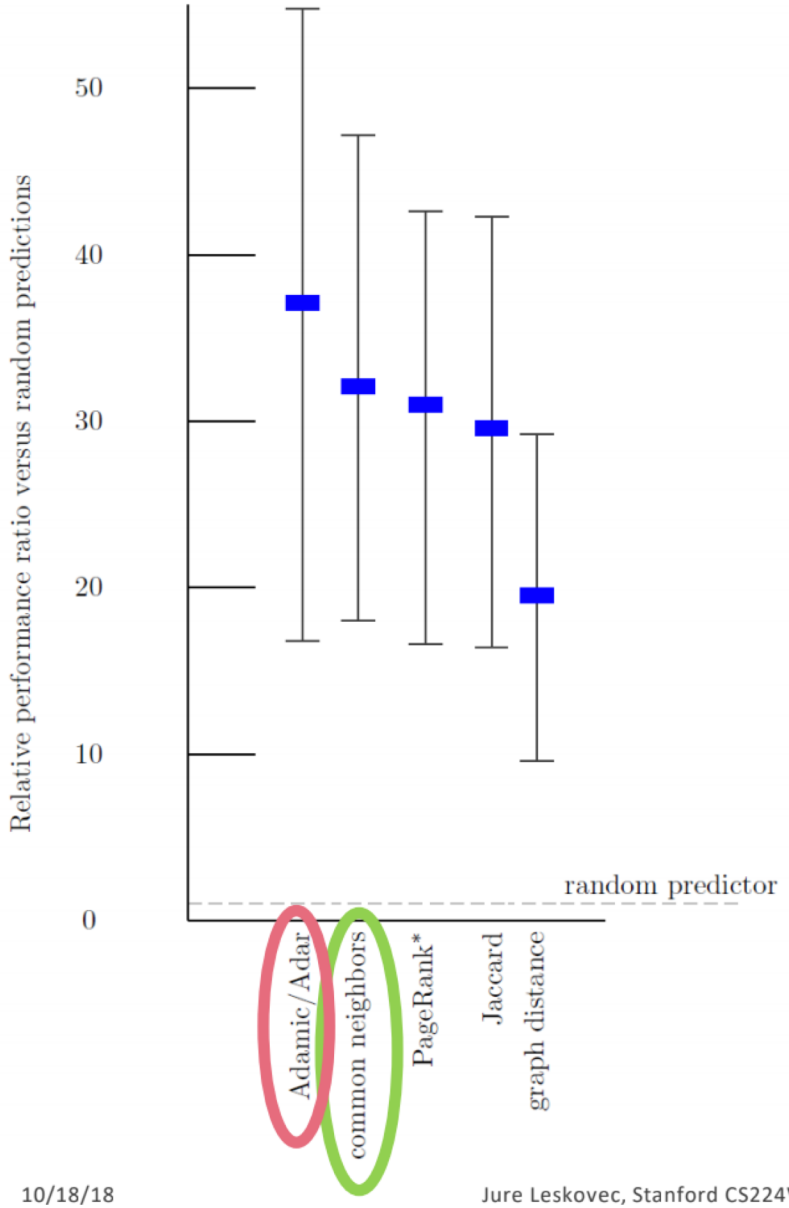
- **Different scoring functions $c(x, y) =$**
  - **Graph distance:** (negated) Shortest path length
  - **Common neighbors:** $|\Gamma(x) \cap \Gamma(y)|$
  - **Jaccard's coefficient:** $|\Gamma(x) \cap \Gamma(y)| / |\Gamma(x) \cup \Gamma(y)|$
  - **Adamic/Adar:** $\sum_{z \in \Gamma(x) \cap \Gamma(y)} 1/\log |\Gamma(z)|$
  - **Preferential attachment:** $|\Gamma(x)| \cdot |\Gamma(y)|$
  - **PageRank:** $r_x(y) + r_y(x)$
    - $r_x(y)$ … stationary distribution score of *y* under the random walk:
      - with prob. 0.15, jump to *x*
      - with prob. 0.85, go to random neighbor of current node

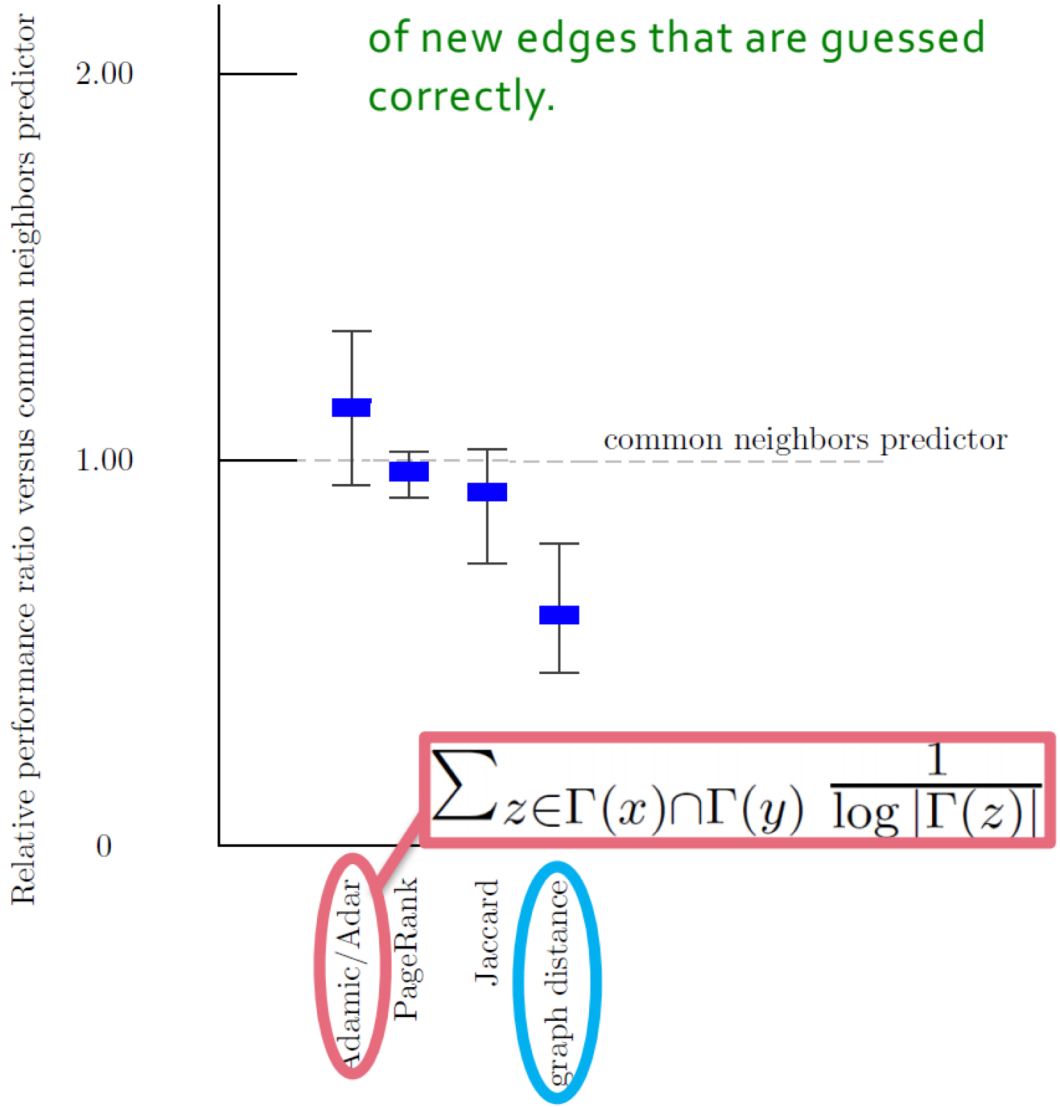
$\Gamma(x)$ … neighbors of node x

- **Then, for a particular choice of *c(·)***
  - For every pair of nodes *(x,y)* compute *c(x,y)*
  - Sort pairs *(x,y)* by the decreasing score *c(x,y)*
  - **Predict top n pairs as new links**

[LibenNowell-Kleinberg ' 03]

Results: Improvement



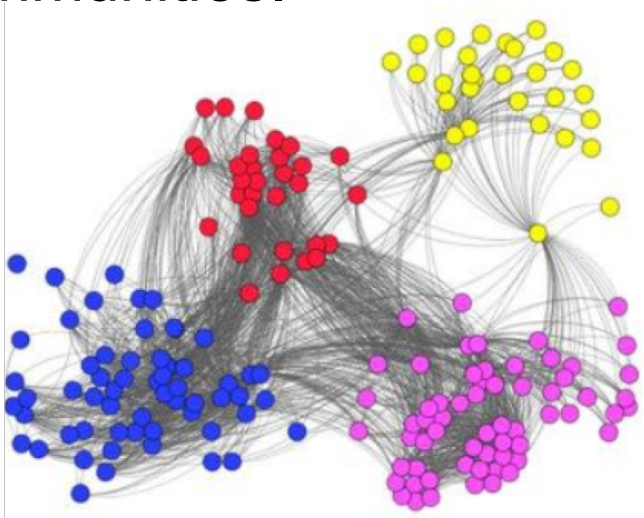


Performance score: Fraction of new edges that are guessed correctly.

Stochastic Blockmodels (SBM)

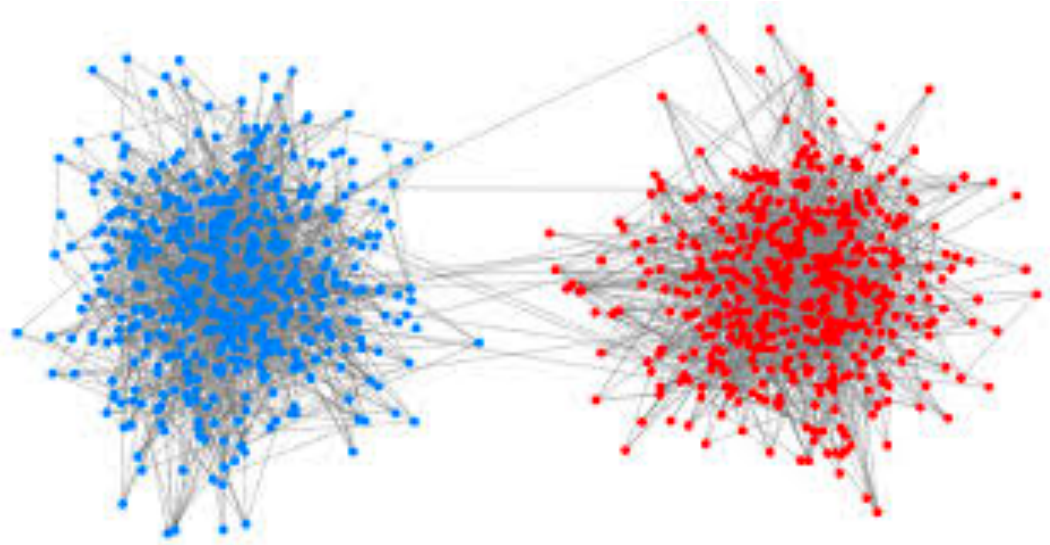
Link prediction via SBMs

- **Link prediction**
  - **Local structure**: Link prediction via proximity
  - **Global structure**: **Stochastic Blockmodels**
    - Another way to predict links is to identify communities.
    - We can then calculate link probabilities within and between communities.

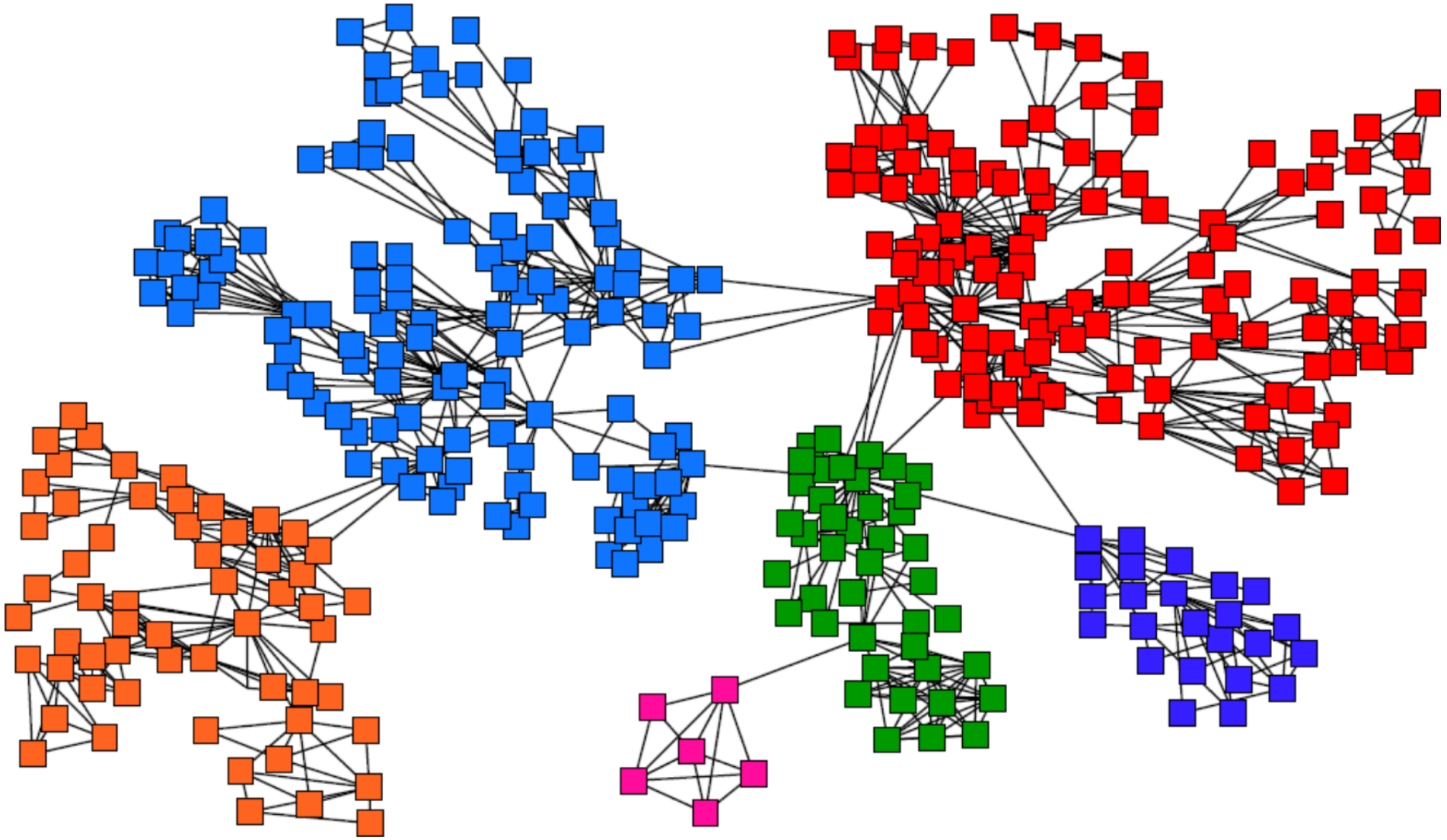


Block Models

- **We often think of networks being organized into modules, cluster, communities.**
- **Blockmodels:**
  - Divide the nodes of the network into distinct sets, or "blocks", where all nodes in the same block have the same pattern of connection to nodes in other blocks.

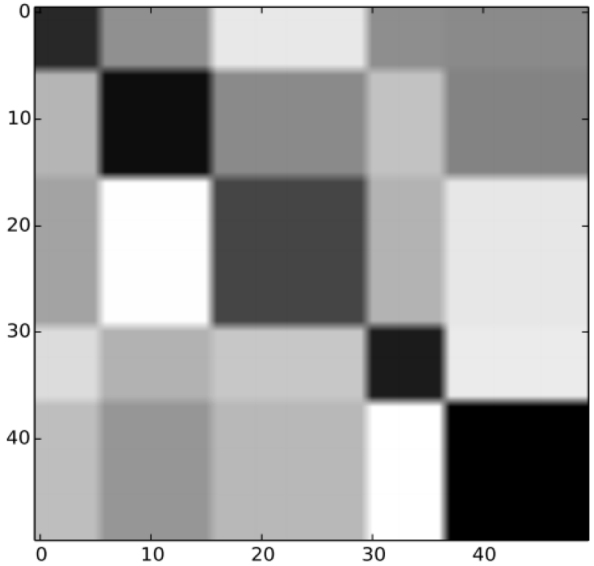


Goal: Model Densely Linked Clusters



Stochastic Blockmodel (SBM)

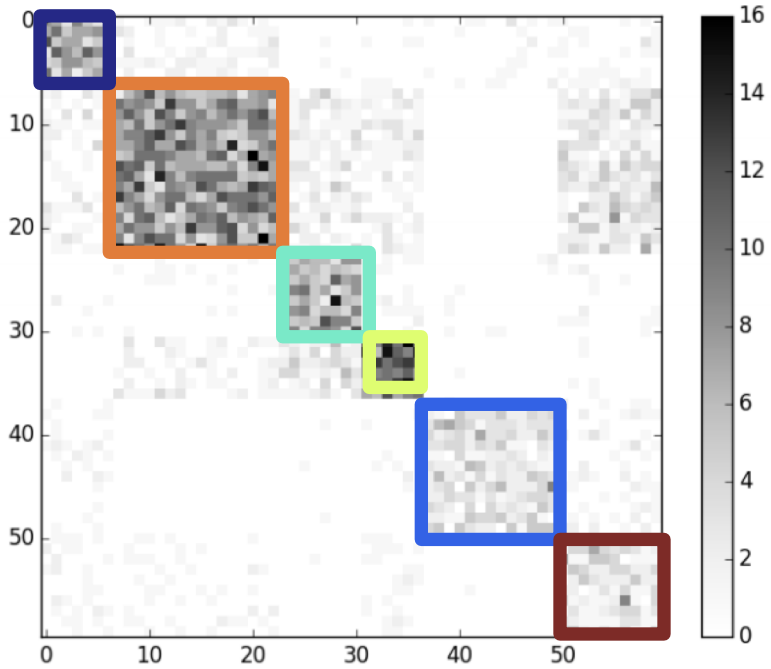
- **Stochastic Block Model** is a generative model for blocks, groups, or communities in networks.
- There are n vertices which belong to k groups
  - Each vertex i belongs to a group $c_i \in \{1, \ldots, k\}$;
- Given $\boldsymbol{c}$, the edges are generated independently,
  - Each pair of vertices is connected with a probability that depends only on their groups:
    - $\Pr[(i,j) \in E] = \theta_{c_i c_j}$.

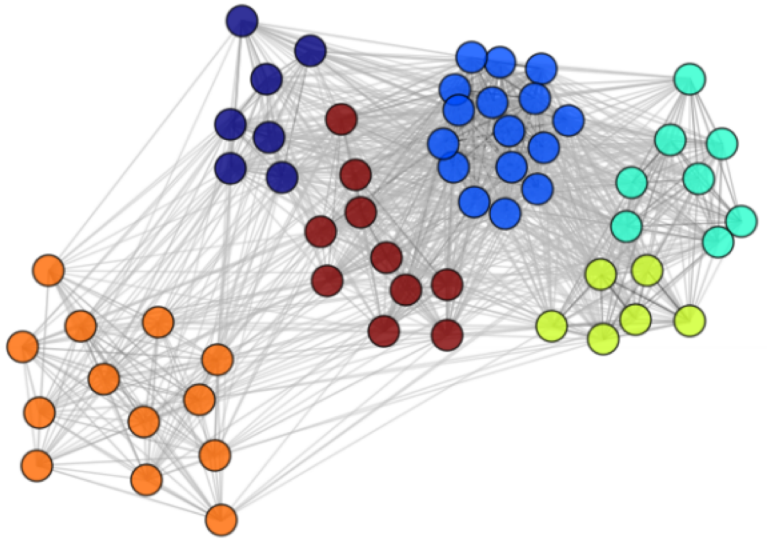


Adjacency matrix of a SBM. The probability of edges are different within and between blocks.

SBM Example Network

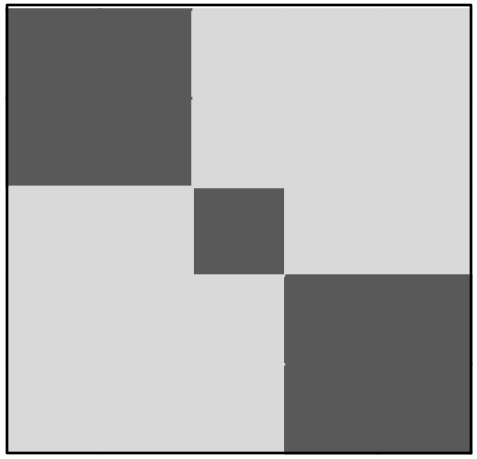
- Network of 60 nodes and 6 communities generated according to a stochastic blockmodel.





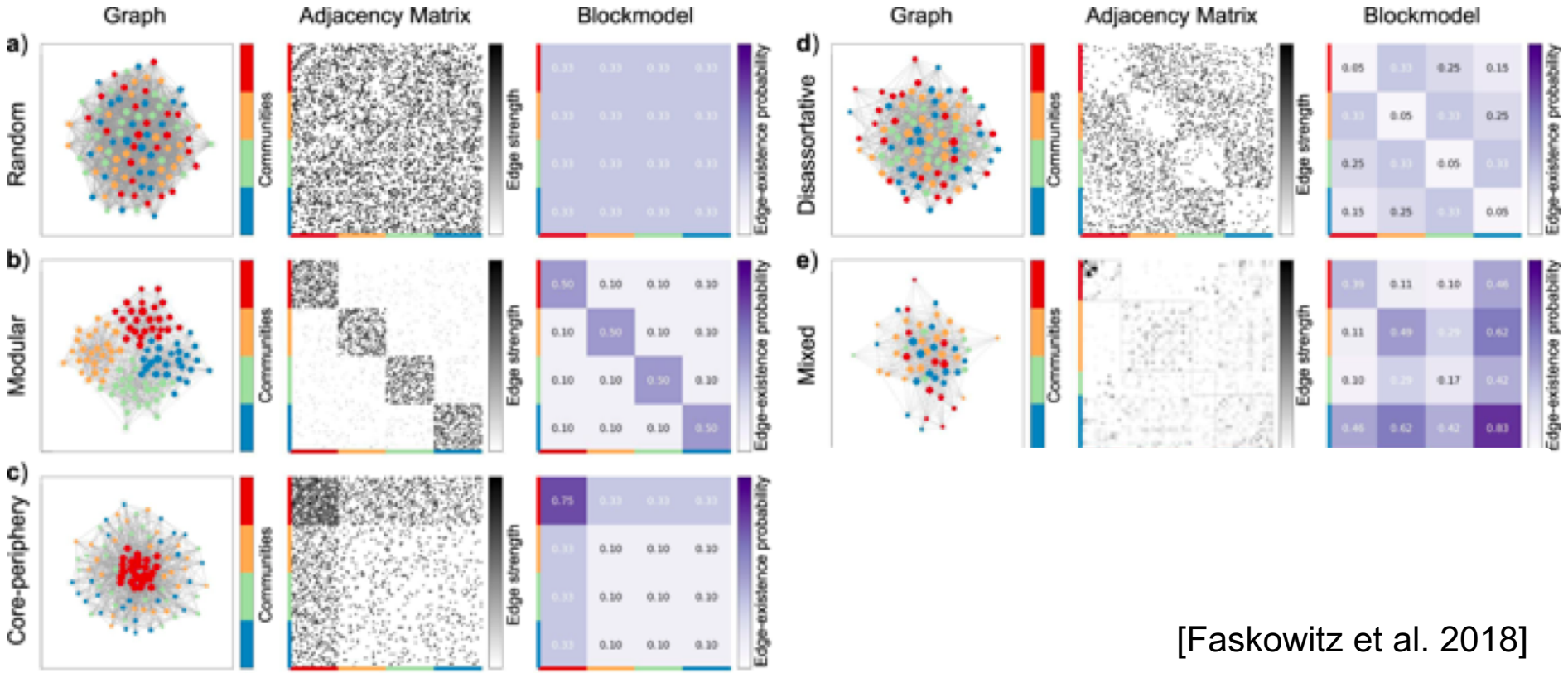
Symmetric SBM (SSBM)

- **In the symmetric case (SSBM)**:
  - c_i are chosen a priori independently and uniformly.
  - $\Pr[(i,j) \in E]$ depends only on whether i and j are in the same or different groups,
  - $$\Pr[(i,j) \in E] = \begin{cases} \theta_{in}. & \text{if } c_i = c_j \\ \theta_{out}. & \text{if } c_i \neq c_j \end{cases}$$



  - We often assume that $\theta_{in} > \theta_{out}$, i.e., that vertices are more likely to connect to others in the same group.
  - The case $\theta_{in} = \theta_{out}$ **is the classic Erdős-Rényi** random graph.

SBM Example Networks

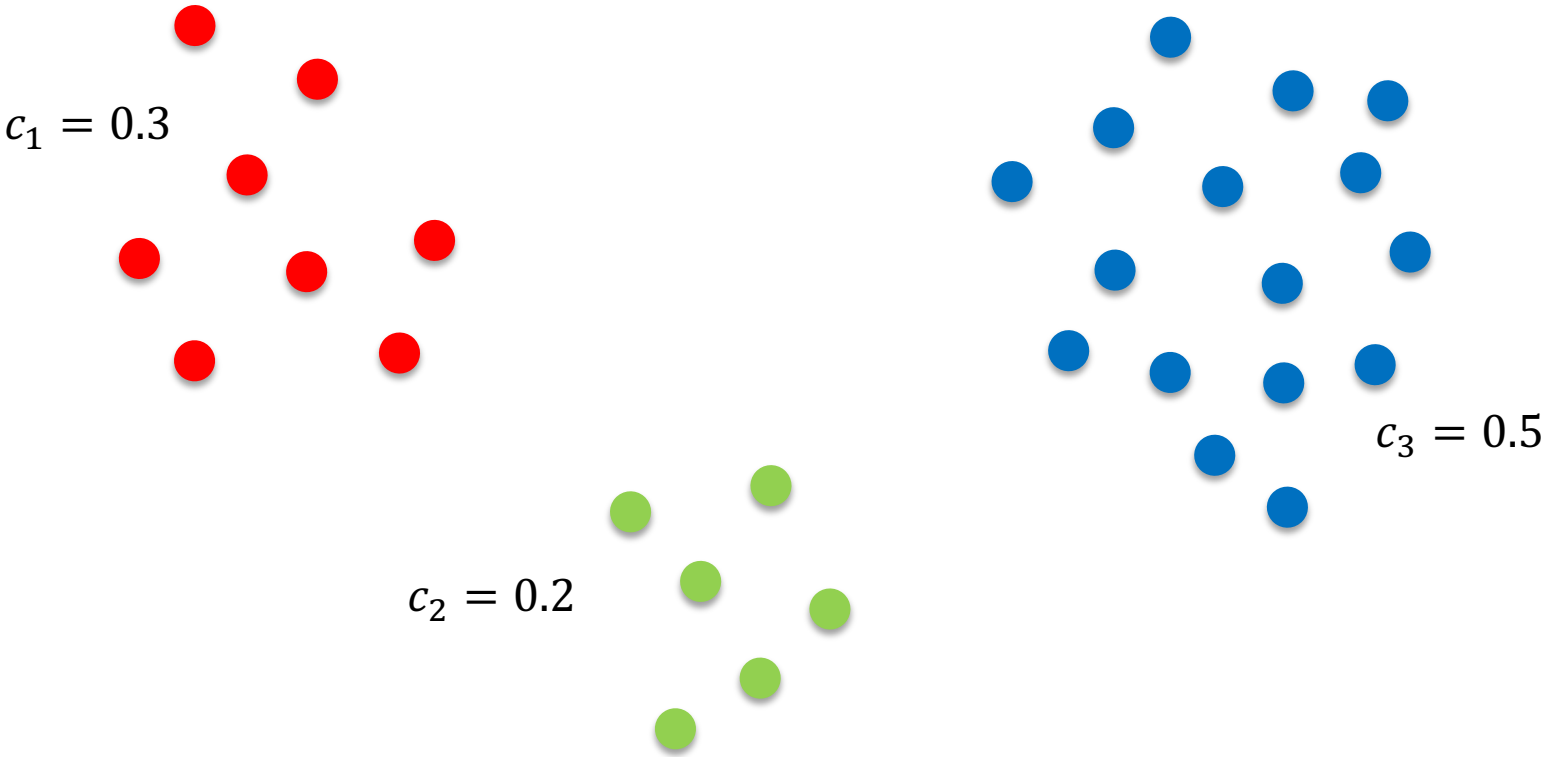


[Faskowitz et al. 2018]

Generating networks with SBM

- $G \sim \mathrm{SBM}(n, k, \theta)$
  - $n = 30, k = 3$
  - $c = [0.3, 0.2, 0.5]$

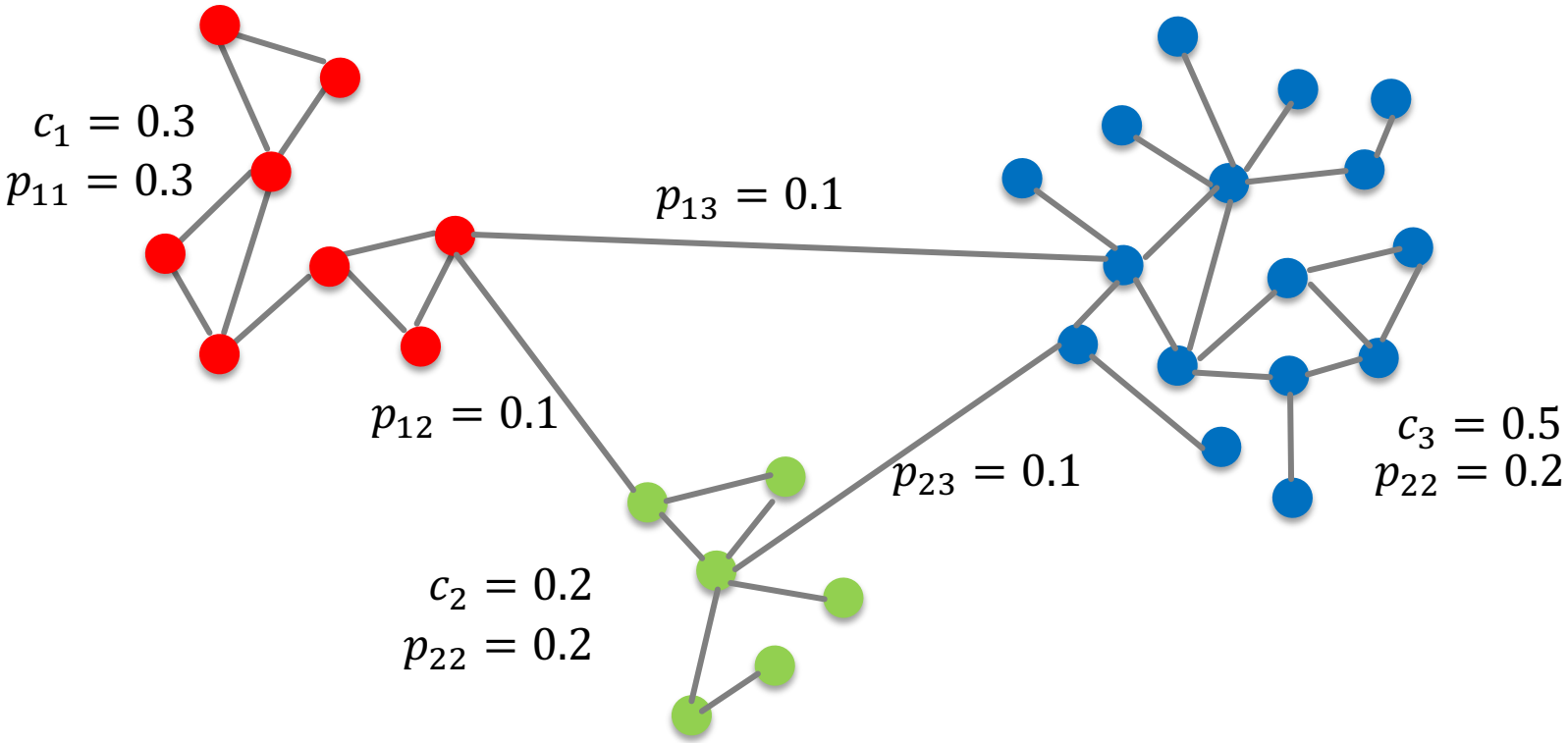
$$\theta = \begin{bmatrix} 0.3 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.2 \end{bmatrix}$$



Generating networks with SBM

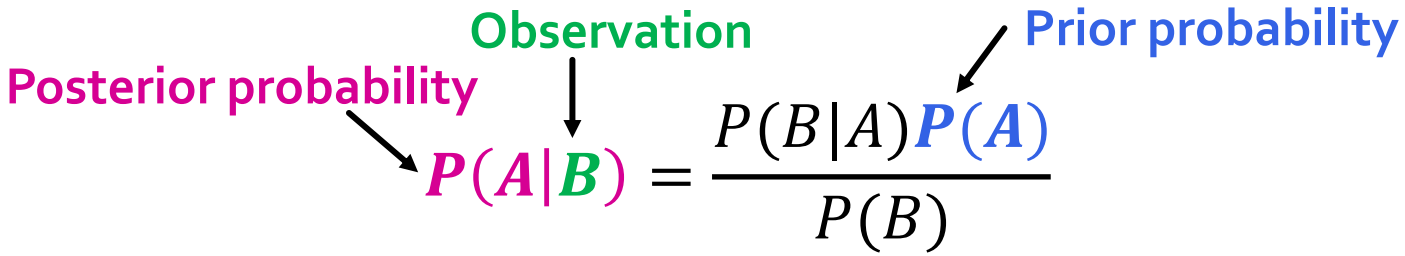
- $G \sim \mathrm{SBM}(n, k, \theta)$
  - $n = 30, k = 3$
  - $c = [0.3, 0.2, 0.5]$

$$\theta = \begin{bmatrix} 0.3 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.2 \end{bmatrix}$$



SBM in Bayesian context

- Some background:
  - **Bayes rule**: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$
  - **Prior probability**: our belief before observation.
  - **Posterior probability**: conditional probability after observation.
  - **Bayes rule can be used to update a prior belief based on some observation.**



SBM in Bayesian context

- More background:
  - **Bayes rule**:

Likelihood

Observation

Prior probability

Posterior probability

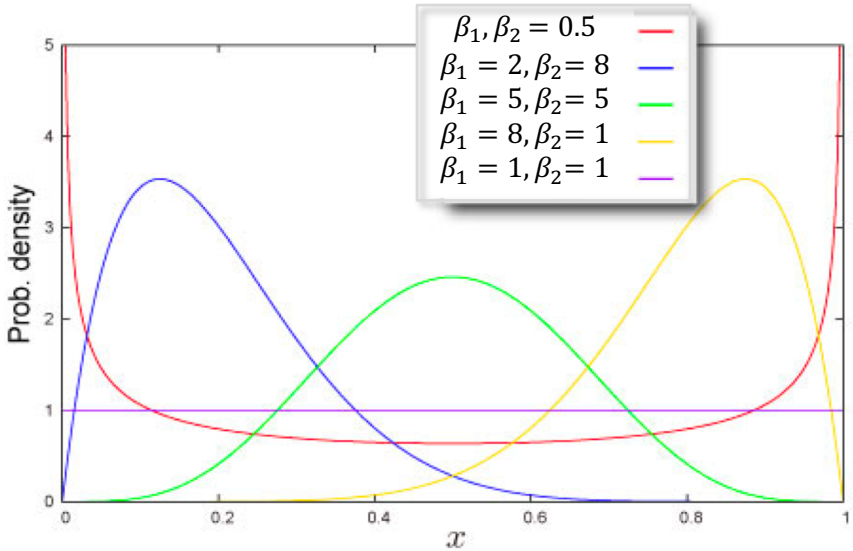
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

  - **Conjugate prior**: if prior and posterior probability distributions are in the same family, prior distribution is conjugate for the likelihood function.
    - **Using conjugate priors can simplify the inference.**

Likelihood	Conjugate prior
Bernoulli $f(k;p) = p^k(1-p)^{1-k}, k \in \{0,1\}$	**Beta** $f(x;\alpha,\beta) = \text{cnt.}\, x^{\alpha-1}(1-x)^{\beta-1}$
Categorical $f(x=i\|\boldsymbol{p}) = p_i, \quad \boldsymbol{p} = \{p_1,\dots,p_k\}$	**Dirichlet** $f(x_1,\dots,x_k;\alpha_1,\dots,\alpha_k) = \frac{1}{B(\alpha)}\prod_{i=1}^{K} x_i^{\alpha_i-1}$

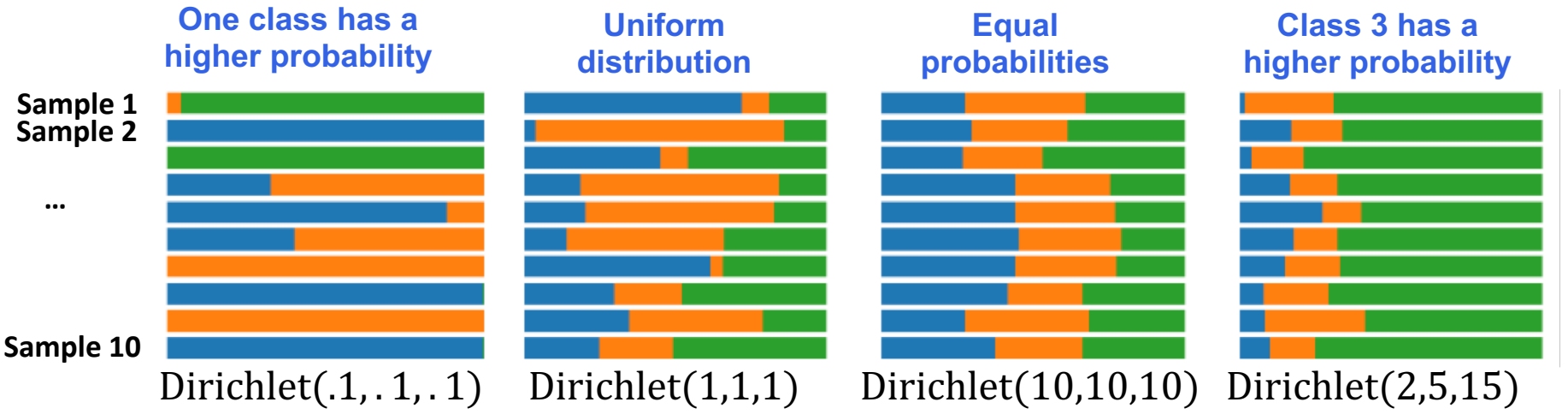
SBM in Bayesian context

- **Conjugate prior**: What does this mean?
  - β_1, β_2 = shape parameters ~ **number of nodes in c_i, c_j**
  - $\theta_{c_i c_j} \sim \text{Beta}(\beta_1, \beta_2)$ **Prior distribution**
  - $A_{ij} \sim \text{Bernoulli}(\theta_{c_i c_j})$ **Likelihood**
  - **Posterior distribution:**
  - $\theta_{c_i c_j} | n_{ij} \sim \text{Beta}\left(\beta_1 + n_{ij}, \beta_2 + m_{ij} - n_{ij}\right)$
    - n_{ij}= number of observed edges between c_i and c_j.
    - m_{ij}= number of possible edges between categories c_i and c_j.



SBM in Bayesian context

- **Conjugate prior**: What does this mean?
  - $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ = concentration hyperparameter
  - $\boldsymbol{\pi}|\boldsymbol{\alpha} = (\pi_1, \ldots, \pi_k) \sim \text{Dirichlet}(\boldsymbol{\alpha})$ **Prior distribution**
  - $\boldsymbol{c}|\boldsymbol{\pi} = (c_1, \ldots, c_k) \sim \text{Categorical}(\boldsymbol{\pi})$ **Likelihood**
  - **Posterior distribution:**
    - n_i = number of occurrences of category i.
    - $\boldsymbol{\pi}|\boldsymbol{n}, \boldsymbol{\alpha} \sim \text{Dirichlet}(\mathbf{n} + \boldsymbol{\alpha}) = \text{Dirichlet}(n_1 + \alpha_1, \ldots, n_k + \alpha_k)$



[Stringham]

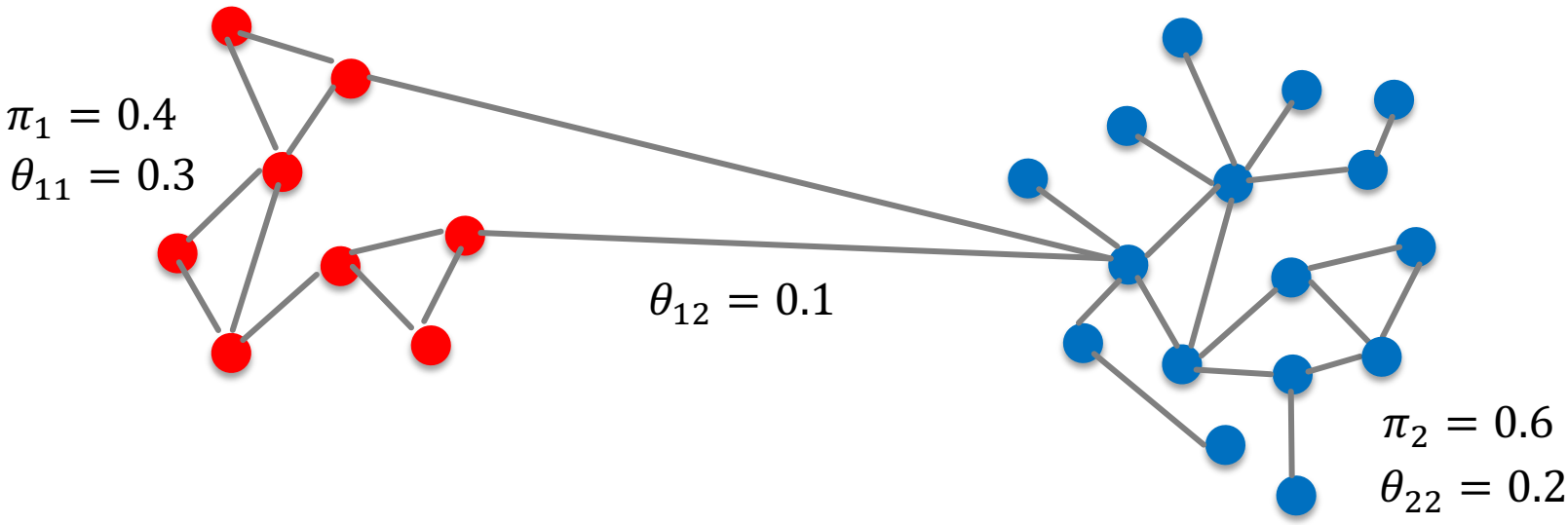
SBM in Bayesian context

- **Bayesian version of the $\mathbf{SBM}(n, k, \theta)$:**
  - **Probabilities of communities**
    - $\pi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k)$,
  - **Assignment of nodes to communities**
    - $c_i \sim \text{Categorical}(\pi)$, $\quad i = 1, \ldots, n$
  - **Edge probabilities within/between communities**
    - $\theta_{c_i c_j} \sim \text{Beta}(\beta_1, \beta_2)$, $\quad c_i, c_j = 1, \ldots, k$
      - If $\beta_1 = \beta_2 = 1 \implies \theta_{c_i c_j} \sim U(0,1)$.
  - **Adjacency matrix**
    - $A_{ij} \sim \text{Bernoulli}(\theta_{c_i c_j})$, $\quad A \in \{0,1\}^{n \times n}$

SBM in Bayesian context

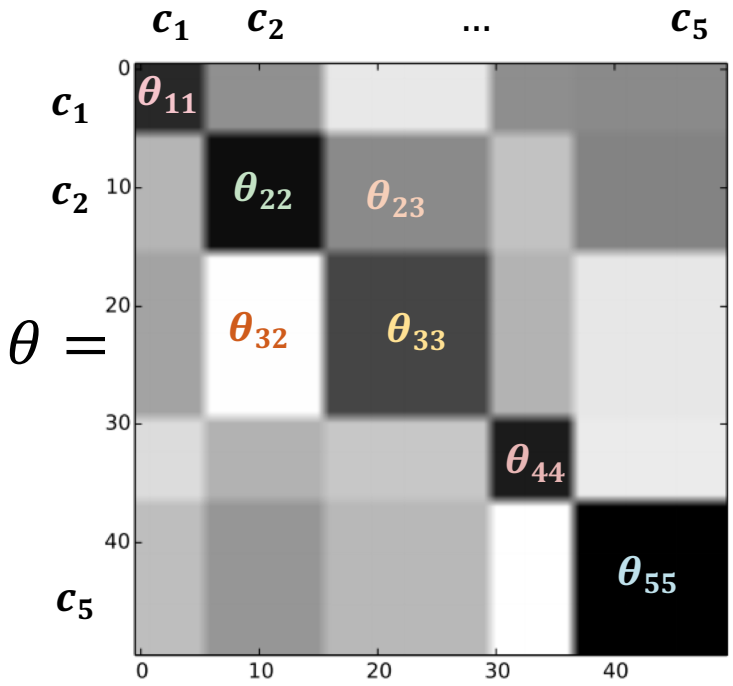
- Given any set of values for $\boldsymbol{\alpha}, \boldsymbol{\beta}$, we can generate random networks using the Bayesian SBM model.
  - Example:
    - n =25
    - k =2

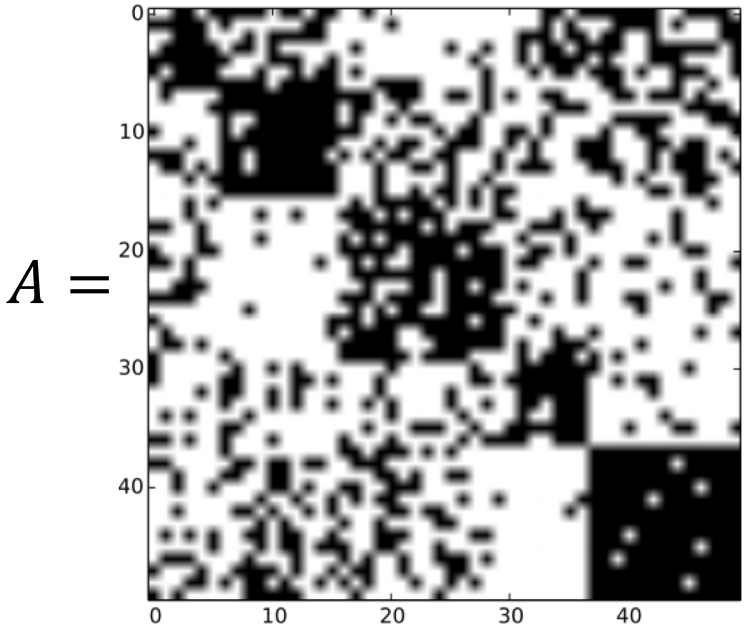
$$\alpha = [3, 5]$$
$$\pi = [0.4, 0.6]$$
$$\beta_1 = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix}$$
$$\beta_2 = \begin{bmatrix} 4 & 8 \\ 8 & 7 \end{bmatrix}$$
$$\theta = \begin{bmatrix} 0.3 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}$$



SBM in Bayesian context

- Given any set of values for α, β, we can generate random networks using the Bayesian SBM model.



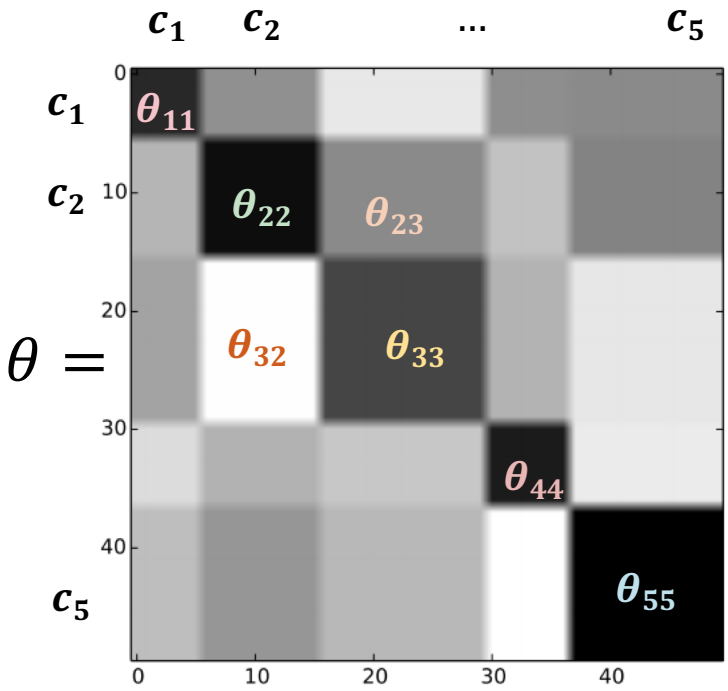


Latent structure of the stochastic blockmodel

Adjacency matrix of a graph generated with the stochastic blockmodel

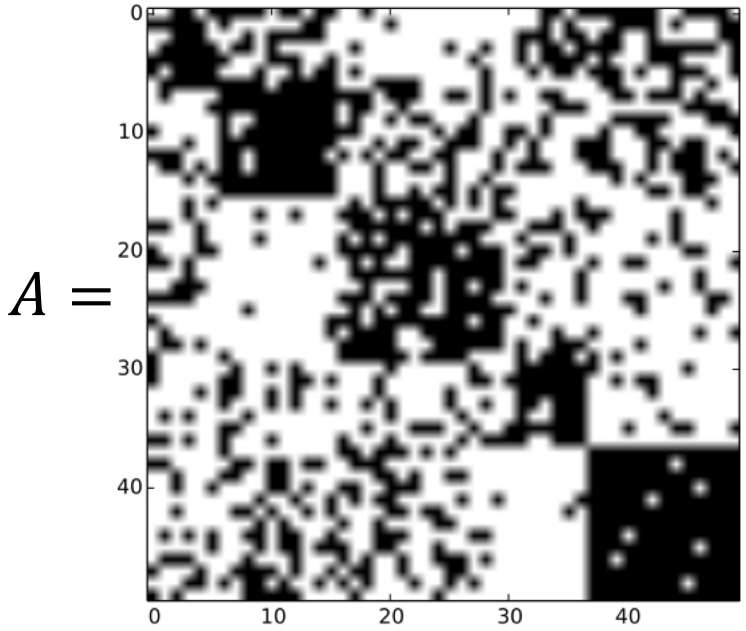
SBM in Bayesian context

- **How can we infer the latent structure (c, π, θ) for an observed network?**



Latent structure of the stochastic blockmodel

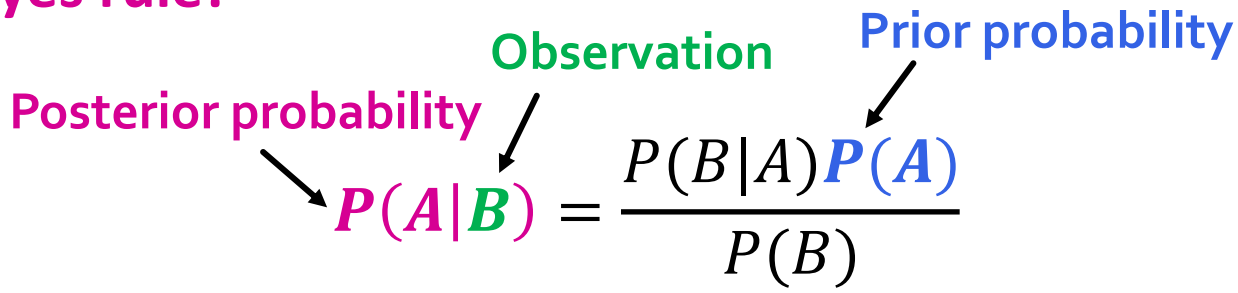
$\Leftarrow ?$



Adjacency matrix of an observed network

Inferring the latent structure

- How can we infer the latent structure (c, π, θ) for an observed network?
  - **Bayes rule!**



  - **Observations**: adjacency matrix A.
  - **Latent variables**: c, π, θ
    - $\boldsymbol{c}, \boldsymbol{\pi}$: **community detection**
    - $\boldsymbol{\theta}$: **link prediction**

Inferring the latent structure

- **Recall: Bayesian version of the SBM(n, k, θ):**
  - $\pi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k)$,
  - $c_i \sim \text{Categorical}(\pi)$, $\quad i = 1, \ldots, n$
  - $\theta_{c_i c_j} \sim \text{Beta}(\beta_1, \beta_2)$, $\quad c_i, c_j = 1, \ldots, k$
  - $A_{ij} \sim \text{Bernoulli}(\theta_{c_i c_j})$, $\quad A \in \{0,1\}^{n \times n}$

- **Bayes rule:** $P(c|A,\pi,\theta) = \dfrac{P(A|c,\pi,\theta)P(c,\pi,\theta)}{P(A,\pi,\theta)}$

$$= \frac{P(A|c,\pi,\theta)P(c|\pi)P(\pi)P(\theta)}{\underbrace{\int_{\pi,\theta} P(A|\pi,\theta)P(\pi,\theta)d\pi\, d\theta}}$$

Intractable to compute!

- **Solution: Approximate $P(c|A,\pi,\theta)$.**

Inference with MCMC

- **Approximate Inference**
  - **Markov Chain Monte Carlo (MCMC) methods**
    - Calculate numerical approximations of multi-dimensional integrals.
      - $E[f(z)] = \int f(z)p(z)dz.$
    - Sample N points $z^{(0)}, z^{(1)}, z^{(2)}, \ldots, z^{(N)}$ according to $p(z)$.
    - I.e., a random walk around the space —from $z^{(0)}$ to $z^{(N)}$— so that the likelihood of visiting any point z is $p(z)$.
      - **Exact**: $E_{p(z)}[f(z)] = \lim_{n\to\infty} \frac{1}{N}\sum_{t=1}^{N} f(z^{(t)})$,
      - **Estimate**: $E_{p(z)}[f(z)] \approx \frac{1}{T}\sum_{t=1}^{T} f\left(z^{(t)}\right)$.

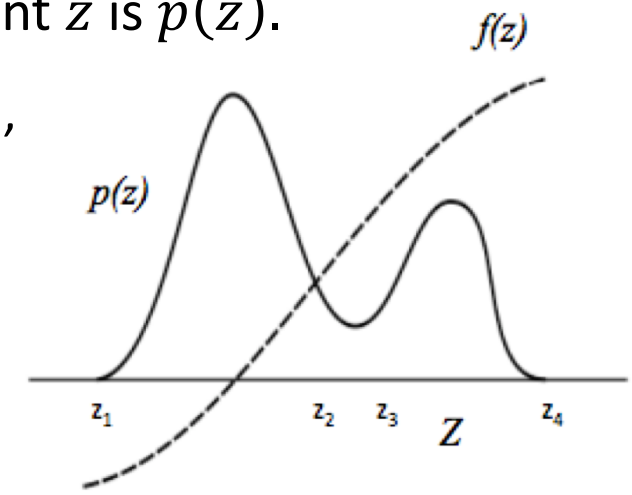


Figure [Bishop]. Walking around values for Z, we want to spend our time adding $f(z)$ to the sum when $p(z)$ is large, e.g. the space between z_1, z_2.

Inference with MCMC

- **Approximate Inference**
  - **Markov Chain Monte Carlo (MCMC) methods**
    - **Monte Carlo technique**: random sampling to obtain numerical results.
    - **Markov property**: next state $z^{(t+1)}$ only depends on the current state $z^{(t)}$.
      - $P(z^{(t+1)} | z^{(0)}, \ldots, z^{(t)}) = P(z^{(t+1)} | z^{(t)})$.
      - **We only need to know $P(z^{(t+1)} | z^{(t)})$.**
    - We need to make sure that the probability of visiting a state z will turn out to be $p(z)$.
      - I.e., Markov Chain has stationary distribution $p(z)$.
    - **Gibbs sampling** guarantees this!

Inferring the latent structure

- **Approximate Inference**
  - **Markov Chain Monte Carlo (MCMC) methods**
    - **Gibbs sampling**
      - Make a separate probabilistic choice for each of the k dimensions, where each choice depends on the other k-1 dimensions.
      - I.e., for each variable sample from its conditional distribution with the remaining variables fixed to their current values
      - That is, the probabilistic walk through the space proceeds as:

1: $z^{(0)} := \langle z_1^{(0)}, \ldots, z_k^{(0)} \rangle$
2: **for** $t = 1$ to T **do**
3: **for** $i = 1$ to k **do**
4: $z_i^{(t+1)} \sim P(Z_i | z_1^{(t+1)}, \ldots, z_{i-1}^{(t+1)}, z_{i+1}^{(t)}, \ldots, z_k^{(t)})$
5: **end for**
6: **end for**

Gibbs sampling for SBM with fixed k

- **Recall: Bayesian version of the $\mathbf{SBM}(n, k, \theta)$:**
  - $\pi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k)$,
  - $c_i \sim \text{Categorical}(\pi)$, $\quad i = 1, \ldots, n$
  - $\theta_{c_i c_j} \sim \text{Beta}(\beta_1, \beta_2)$, $\quad c_i, c_j = 1, \ldots, k$
  - $A_{ij} \sim \text{Bernoulli}(\theta_{c_i c_j})$, $\quad A \in \{0,1\}^{n \times n}$

- **Recall: How to update the posteriors with observations?**
  - n_{ij}= number of observed edges between categories c_i and c_j.
  - m_{ij}= number of possible edges between categories c_i and c_j.
  - $\theta_{c_i c_j} | n_{ij} \sim \text{Beta}\left(\beta_1 + n_{ij}, \beta_2 + m_{ij} - n_{ij}\right)$

  - n_i = number of occurrences of category i.
  - $\boldsymbol{\pi} | \boldsymbol{n}, \boldsymbol{\alpha} \sim \text{Dirichlet}(\mathbf{n} + \boldsymbol{\alpha}) = \text{Dirichlet}(n_1 + \alpha_1, \ldots, n_k + \alpha_k)$

Gibbs sampling for SBM with fixed k

- **Recall: Bayesian version of the SBM(n, k, θ):**
  - $\pi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_k)$,
  - $c_i \sim \text{Categorical}(\pi)$, $\quad i = 1, \ldots, n$
  - $\theta_{c_i c_j} \sim \text{Beta}(\beta_1, \beta_2)$, $\quad c_i, c_j = 1, \ldots, k$
  - $A_{ij} \sim \text{Bernoulli}(\theta_{c_i c_j})$, $\quad A \in \{0,1\}^{n \times n}$

- **Recall: How to update the posteriors with observations?**
  - **Bayes rule:**
  - $P(c|A, \pi, \theta) \propto P(A|c, \pi, \theta) P(c|\pi) P(\pi) P(\theta)$.
  - $P(A|c, \pi, \theta) \propto \Pi_{j \neq i} \theta_{c_i c_j}^{A_{ij}} \left(1 - \theta_{c_i c_j}\right)^{1 - A_{ij}} \times \Pi_{k \neq i} \theta_{c_k c_i}^{A_{ki}} \left(1 - \theta_{c_k c_i}\right)^{1 - A_{ki}}$,
  - $P(c|\pi) \propto \pi_{c_i}$

Gibbs sampling for SBM with fixed k

Gibbs Sampler for SBM(n, k, θ)

Initialize π, θ, c at random.

for $t = 1, 2, \ldots$ **do**

- **Collecting observations**

  Number of nodes in community c_r:

  $n_r = \sum_{i=1}^{n} I(c_i = r), \qquad r = 1, \ldots, k$

  Number of possible edges between communities c_r and c_s:

  $n_{rs} = \sum_{1 \le i \ne j \le n} I(c_i = r, c_j = s) = n_r n_s - n_r I(r = s).$

  Number of existing edges between c_r and c_s:

  $n_{rs} = \sum_{(i,j): c_i = r, c_j = s} A_{ij}, \quad r, s = 1, \ldots, k$

- **Link prediction**

  $\pi \sim \text{Dirichlet}(\alpha_1 + n_1, \ldots, \alpha_k + n_k)$

  $\theta_{c_i c_j} \sim \text{Beta}(\beta_1 + n_{ij}, \beta_2 + m_{ij} - n_{ij})$

- **Community detection**

  $P(c_i = l | c_{-i}, A, \pi, \theta) \propto \pi_{c_i} \times \Pi_{j \ne i} \theta_{c_i c_j}^{A_{ij}} \left(1 - \theta_{c_i c_j}\right)^{1 - A_{ij}} \times \Pi_{k \ne i} \theta_{c_k c_i}^{A_{ki}} \left(1 - \theta_{c_k c_i}\right)^{1 - A_{ki}}$

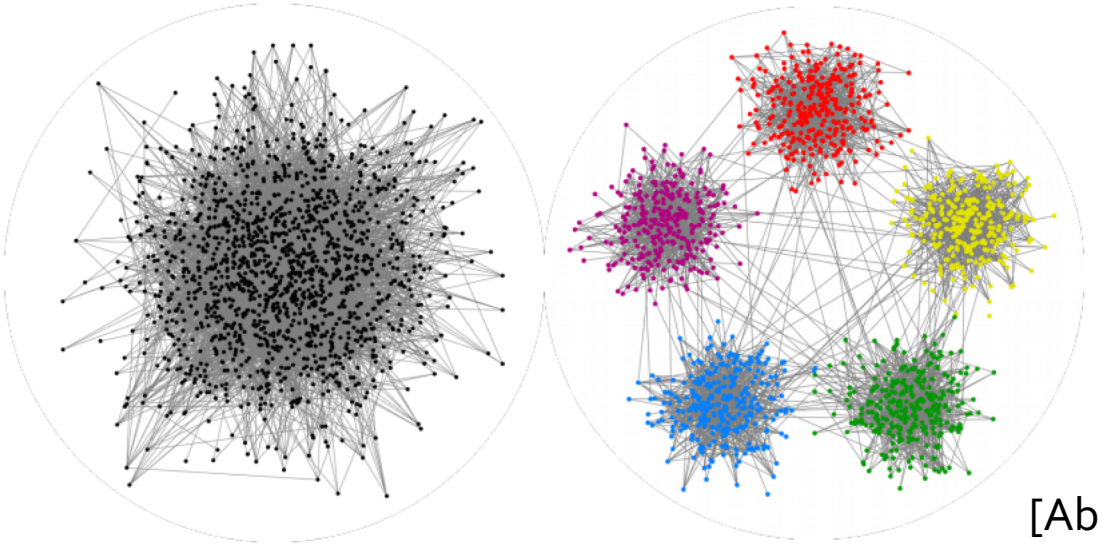
end for

c_{-i}: all community assignments except c_i.

Stochastic Blockmodel and Spectral Clustering

Recovery requirements

- **When can spectral clustering detects communities in SBM?**
  - If the community structure is too weak or the graph too sparse,
    - then no algorithm can label the vertices better than chance, or distinguish the graph from a purely random graph.



[Abbe'17]

Recovery requirements

[Abbe'17]

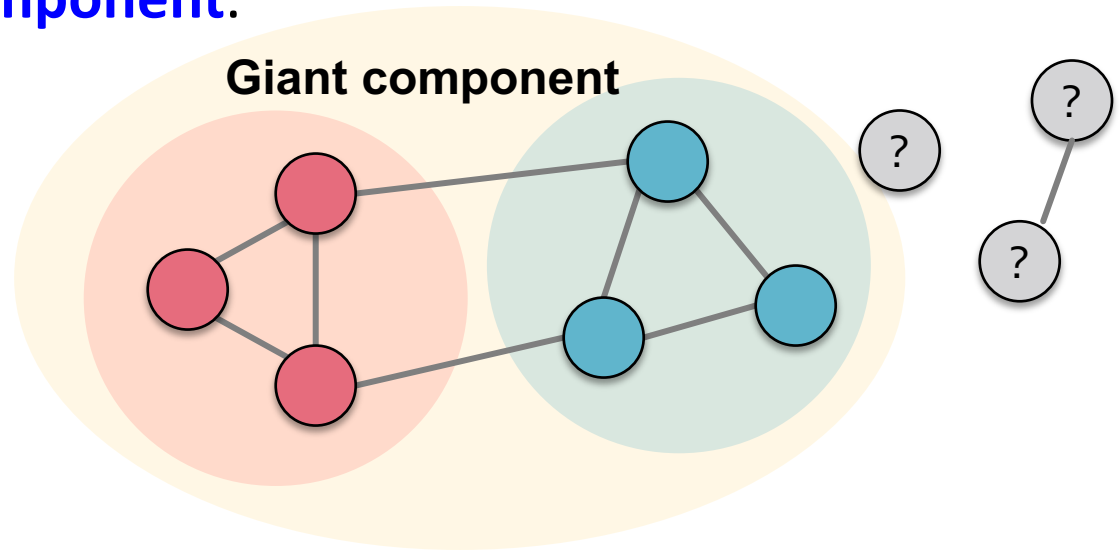
- **When can spectral clustering detects communities in SBM?**
  - Let $(X, G) \sim \mathrm{SBM}(n, \pi, \theta)$,
  - $\pi \in \mathbb{R}^k$: prior on the k communities,
  - $\theta \in [0,1]^{k \times k}$: connectivity probabilities.

- If there exists an algorithm that takes G as an input and outputs $X' = X'(G)$, then
  - **Exact recovery**: $P\{A(X, X') = 1\} = 1 - o(1)$,
    - **Requires the entire partition to be correctly recovered.**
  - **Weak recovery (detection)**: $P\{A(X, X') \geq \alpha\} = 1 - o(1)$, $\alpha \in (0, 1)$.
    - **Allows for a constant fraction of misclassified vertices.**

Recovery requirements

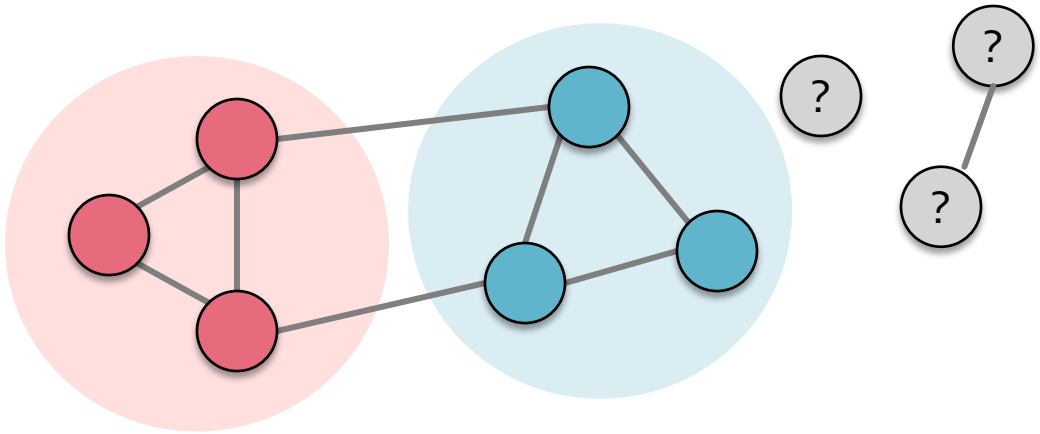
- **Exact recovery**:
  - If the SBM graph is **not connected**, exact recovery is not possible.
    - There is no hope to label disconnected components with higher chance than $1/2$.

- **Weak recovery**:
  - Weak recovery is not solvable if the graph **does not have a giant component**.



Recovery requirements

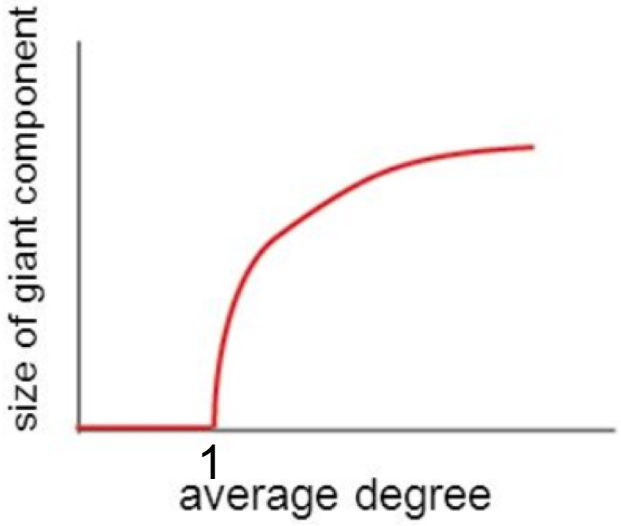
- **Exact recovery**:
  - If the graph is not **connected**, exact recovery is not possible.
    - Erdos-Renyi graph $G(n, p = z\ln(n)/n)$ **is connected** with high probability if and only **if** $z > 1$.
      - $p = \ln(n)/n$: **sharp threshold for the connectedness of** $G(n, p)$.



Recovery requirements

[Abbe'17]

- **Weak recovery**:
  - Weak recovery is not solvable if the graph does not have a **giant component**.
    - Erdos-Renyi graph $G(n, p = z/n)$ **has a giant component** (i.e., a component of size linear in n) if and only **if** $z > 1$.
      - For $z > 1$, $\mathrm{G}(n, z/n)$ will almost surely have a unique giant component containing a positive fraction of the vertices.
      - No other component will contain more than $O(\log n)$ vertices.



Recovery requirements

[Abbe'17]

- **In $\mathbf{SSBM}(n, \pi = 1/k, a/n, b/n)$**
  - The expected size of each group is n/k.
  - Each vertex has in expectation
    - a/k neighbors in its own group, and
    - b/k in each of the other groups.
  - Expected degree = $d = \frac{a+(k-1)b}{k}$
- **In $\mathbf{SBM}(n, \pi, \theta\, \log(n)/n)$**
  - Expected degree of group $i =$

$$\sum_{j \in [k]} (\pi_j \theta_{ji}) = \|(\mathrm{diag}(\pi)\theta)_i\|_1$$

Recovery requirements

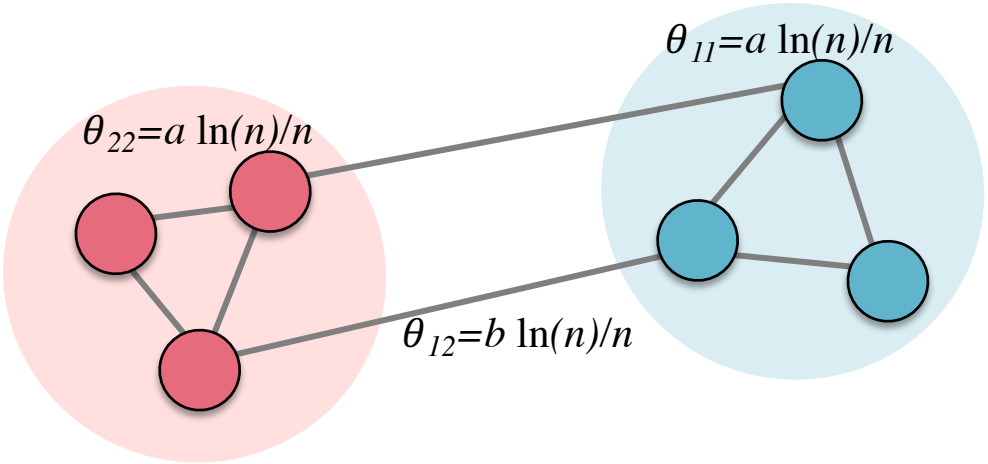
[Abbe'17]

- **Therefore for SBMs**:
  - Define $d :=$ expected degree
  - **Exact recovery**:
    - For $a, b > 0$, $\mathbf{SSBM}(n, k, a\log(n)/n, b\log(n)/n)$ is connected with high probability if and only if $d = (a + (k-1)b)/k > 1$.
    - $\mathbf{SBM}(n, k, Q\log(n)/n)$ is connected with high probability if $\min_{i\in[k]} d_i = \|(\mathrm{diag}(\pi)\theta)_i\|_1 > 1$.
  - **Weak recovery**:
    - $\mathbf{SSBM}(n, k, a/n, b/n)$ has a giant component (i.e., a component of size linear in n) if and only if $d := (a + (k-1)b)/k > 1$.

[Abbe'17]

Fundamental Limits for recovery

- **The fundamental limit for exact recovery**
  - **Exact recovery** in SSBM$(n, 1/2, a\ \ln(n)/n, b\ \ln(n)/n)$ is solvable and efficiently so if $\left|\sqrt{a}-\sqrt{b}\right|>\sqrt{2}$.
  - Note that $\left|\sqrt{a}-\sqrt{b}\right|>\sqrt{2} \Rightarrow \frac{a+b}{2}>1+\sqrt{ab}$.
  - Recall that $\frac{a+b}{2}>2$ is the connectivity requirement in SSBM.
  - $\sqrt{ab}$ is required to go from connectivity to exact recovery.



Exact recovery needs:

$$\frac{a+b}{2}>1+\sqrt{ab}$$

[Abbe'17]

Fundamental Limits for recovery

- **The fundamental limit for exact recovery**
  - **Exact recovery** in $\mathrm{SBM}(n, \pi, \ln(n)\theta/n)$ is solvable and efficiently so if
    - $\boldsymbol{I_+(\pi,\theta) := \min_{i<j} D_+((\mathbf{diag}(\pi)\theta)_i \| (\mathbf{diag}(\pi)\theta)_j) > 1,}$
    - $D_+ = \max_{t\in[0,1]} D_t$
  - **Chernoff-Hellinger (CH) divergence**
    - $D_t(\nu||\mu) = \max_{t\in[0,1]} \sum_x \nu(x) f_t(\mu(x)/\nu(x)), \quad f_t(y) := 1 - t + ty - y^t$
    - D_t is a **distance notion between communities**.
    - **Intuitively: the further "apart" the community profiles are, the easier it should be to distinguish the communities.**

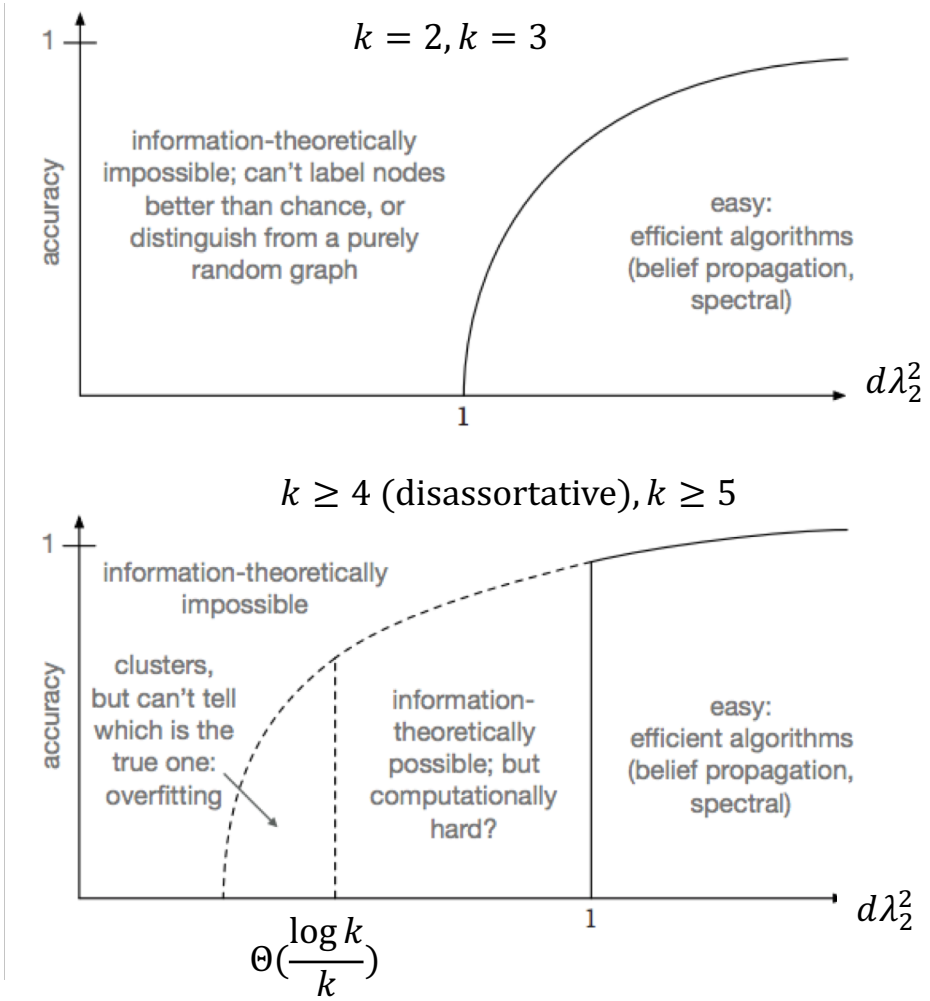
Fundamental Limits for recovery

[Abbe'17]

- **The fundamental limit for weak recovery**
  - **It is possible to detect communities if SNR > 1 (Kesten-Stigum (KS) threshold).**
    - **SNR:** expected number of in-block edges divided by the expected number of out-block edges.
    - **SSBM$(n, 1/k, a/n, b/n)$**
      - $\mathbf{SNR} = \frac{(a-b)^2}{k(a+(k-1)b)}$
    - **SBM$(n, \pi, \theta/n)$**
      - Let $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3|$ … be eigenvalues of $\mathrm{diag}(\pi)\theta$.
      - $\mathrm{SNR} = \lambda_2^2/\lambda_1$
      - When $(a-b)/d = O(1)$, $\mathbf{SNR} = \boldsymbol{\Theta}(\log(k)/k)$

[Moore'17]

Phase Transitions for Detection



A summary of known and conjectured phase transitions in the block model where the number of groups is $k = 2, 3$ (top) and $k \geq 4$ (bottom).

SBM and spectral clustering

[Rohe et al.'11]

- **How does spectral clustering perform on SBMs?**
  - Let $(X, G) \sim \mathrm{SBM}(n, \pi, \theta)$.
  - $Z \in \{0,1\}^{n \times k}$: block assignments (if $Z_{ij} = 1$, then node i is in block j).
  - $\theta \in [0,1]^{k \times k}$: connectivity probabilities (full rank and symmetric).
  - $\boldsymbol{\mathcal{W}} = Z\theta Z^T$

- Define the **population Laplacian** $\boldsymbol{\mathcal{L}}$,
  $$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{D}}^{-1/2} \boldsymbol{\mathcal{W}} \boldsymbol{\mathcal{D}}^{-1/2}, \quad \text{where } \boldsymbol{\mathcal{D}}_{ii} = \sum_k \boldsymbol{\mathcal{W}}_{ik}.$$

- **Spectral clustering can discover the block structure in the matrix Z.**

[Rohe et al.'11]

SBM and spectral clustering

- **How does spectral clustering perform on SBMs?**
  - If $(\log n)^2/\sqrt{n} = O(\lambda_k^2)$, then **the number of misclustered nodes $\mathcal{M}$ by spectral clustering is bounded by**

$$|\mathcal{M}| = o\left(\frac{P\,(\log n)^2}{\lambda_k^4 \tau^4 n}\right).$$

  - P is the size of the largest block in the network,
  - τ is the minimum expected degree, divided by the maximum possible degree, and
  - $|\overline{\lambda_1}| \geq |\overline{\lambda_2}| \geq \ldots \geq |\overline{\lambda_k}| > 0$ are the absolute values of the k nonzero eigenvalues of the population graph Laplacian $\mathcal{L}$.
  - **In SBM we expect all eigenvalues other than the largest k in absolute value are small.**

SBM and spectral clustering

[Rohe et al.'11]

- **How does spectral clustering perform on SBMs?**
  - If $(\log n)^2/\sqrt{n} = O(\lambda_k^2)$, then **the number of misclustered nodes $\mathcal{M}$ by spectral clustering is bounded by**

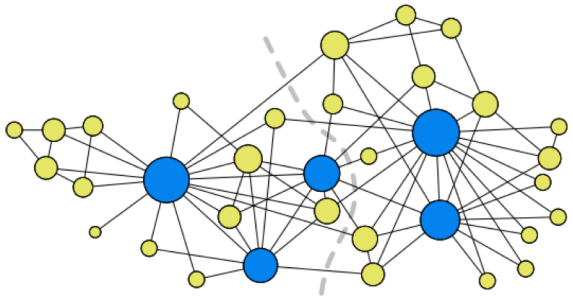
$$|\mathcal{M}| = o\left(\frac{P\ (\log n)^2}{\lambda_k^4 \tau^4 n}\right).$$

- **Spectral clustering can correctly partition most of the nodes in the SBM, even when the number of blocks grows with the number of nodes.**

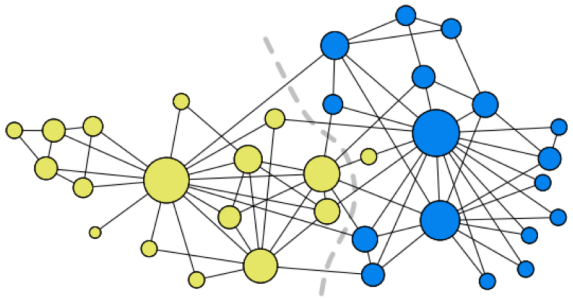
Stochastic Blockmodel Extensions

Degree-corrected SBMs

- Degree-corrected SBMs:
  - allowing for a degree parameter for each vertex that scales the edge probabilities in order to make expected degrees match the observed degrees.

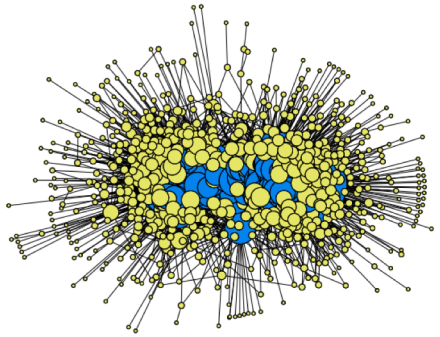


(a) Without degree correction

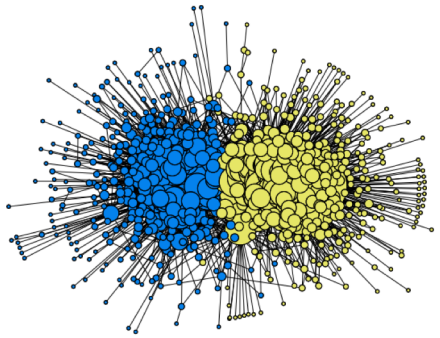


(b) With degree-correction

Divisions of the karate club network found using the (a) uncorrected and (b) corrected blockmodels.



(a) Without degree-correction



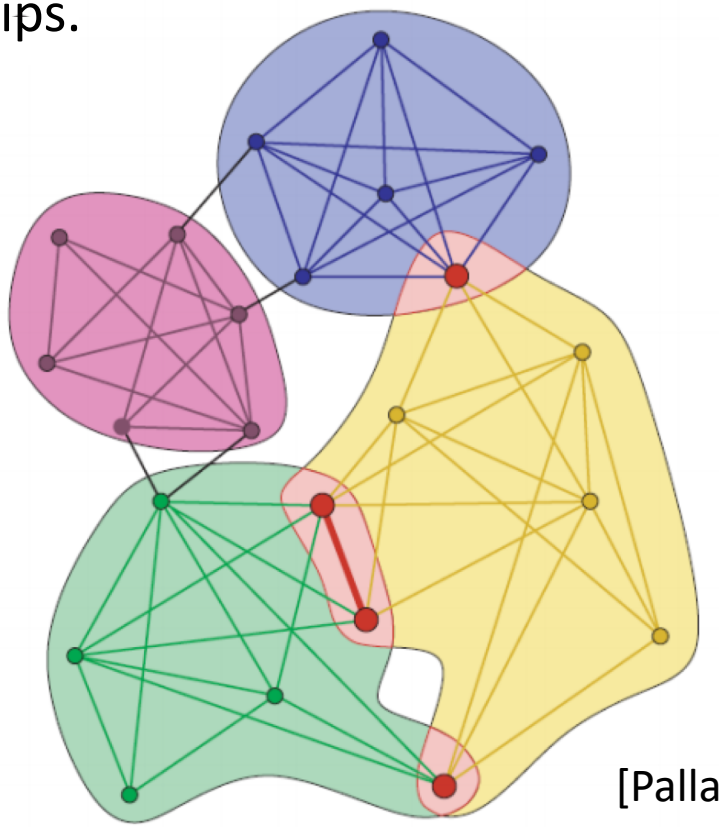
(b) With degree-correction

[Karrer et al. '10]

Divisions of the political blog network found using the (a) uncorrected and (b) corrected blockmodels

Overlapping SBMs

- Overlapping SBMs:
  - Allowing for the communities to overlap, such as in the mixed membership SBM [ABFX08], where each vertex has a profile of community memberships.

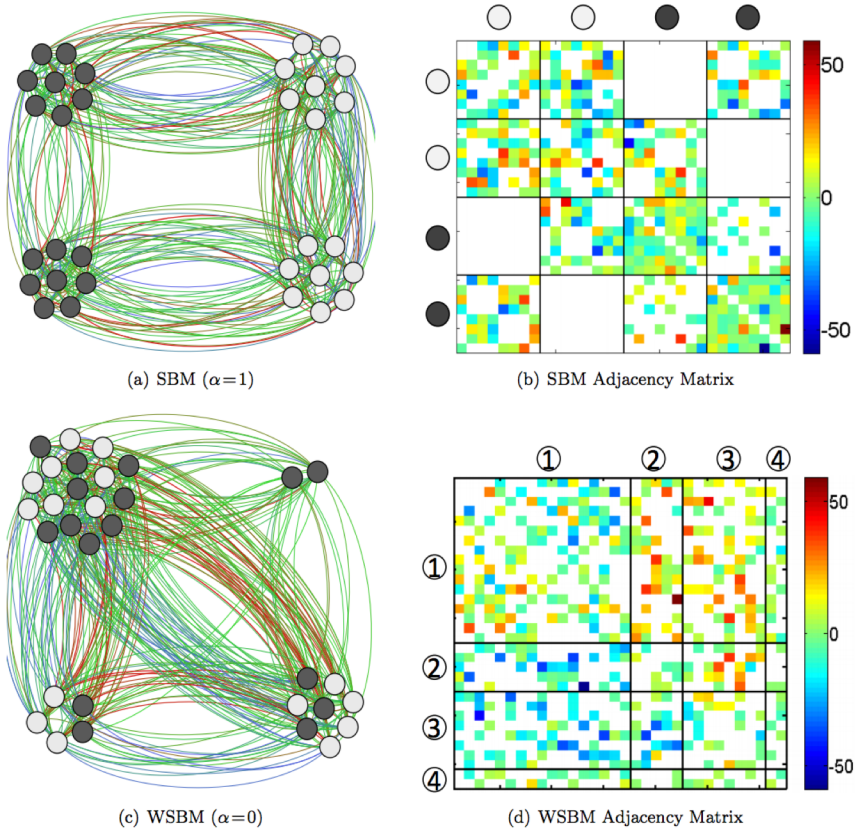


[Palla et al. '06]

Weighted (labeled) SBMs

- Labelled SBMs:
  - Allowing for edges to carry a label, which can model intensities of similarity functions between vertices.

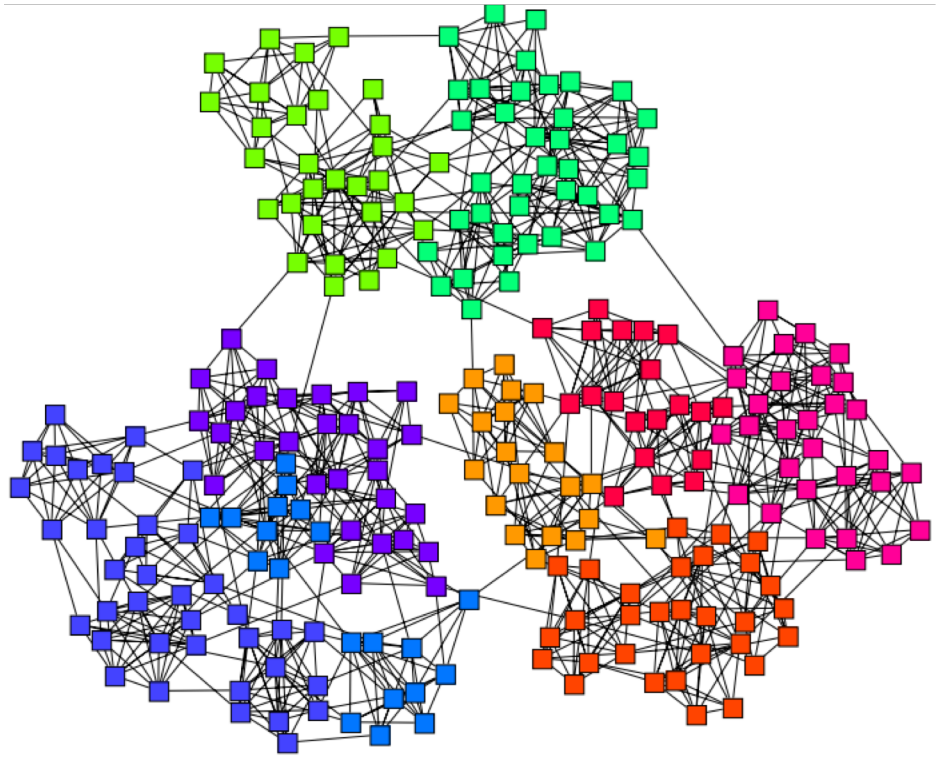
NFL-2009 network: black nodes are teams in conference 1 (NFC) and white nodes are teams in conference 2 (AFC). Edges are colored by score differential (red positive, green approximately zero, blue negative). (a, b) Network showing SBM communities (c, d) Network showing WSBM communities. The SBM (α=1) groups correspond to NFL conference structure, whereas the WSBM (α=0) corresponds to relative skills levels.



[Aicher et al. '14]

Hierarchical SBMs

- Hierarchical SBMs:
  - Allowing the stochastic blockmodel to have a hierarchical structure.



[Clauset '13]