Recovering communities in the general stochastic block model

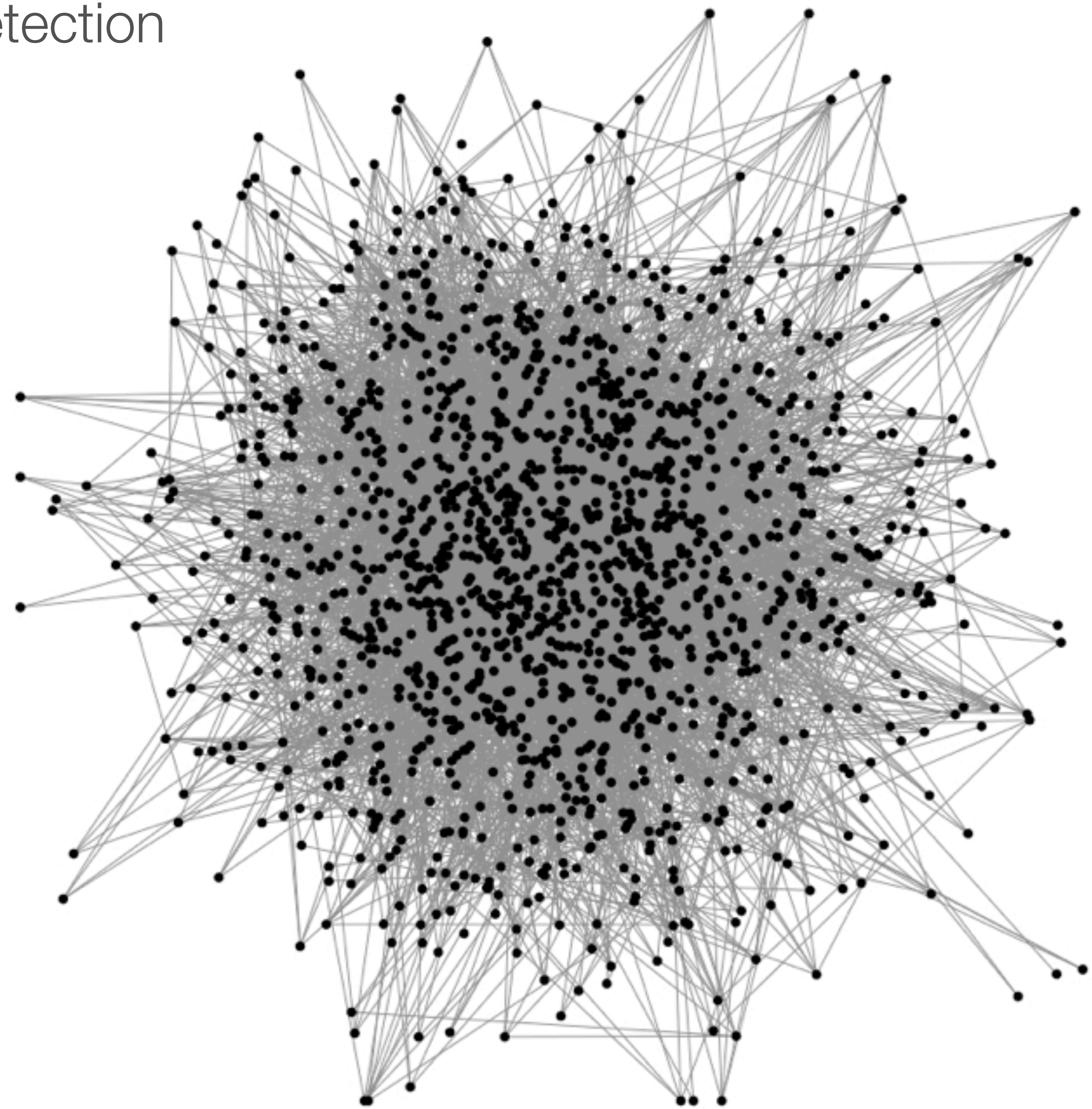
Emmanuel Abbe and Colin Sandon
Princeton University

http://arxiv.org/abs/1503.00609

Simons Institute, 03.16.15

community detection

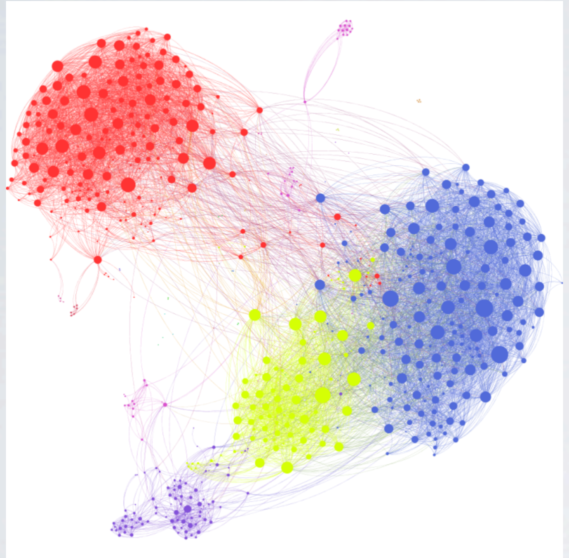
community detection



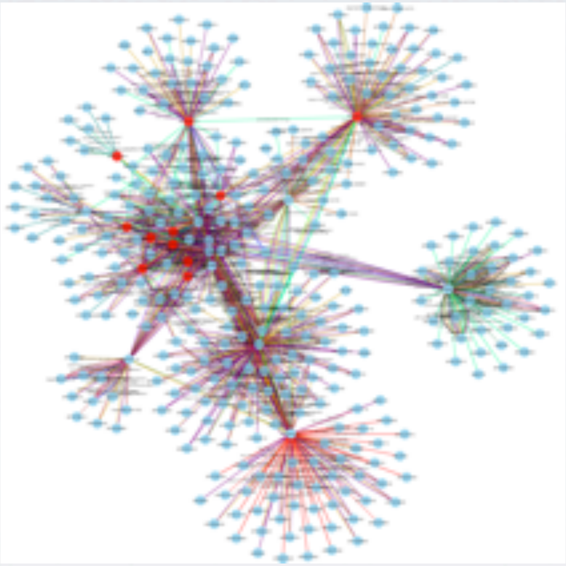
community detection

community detection

community detection: applications



social networks

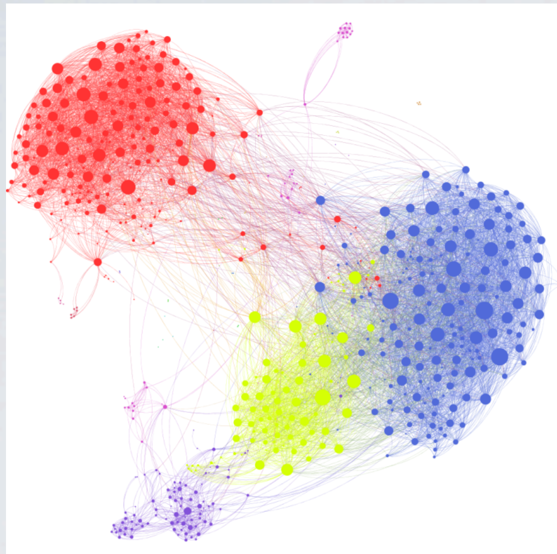


biological networks

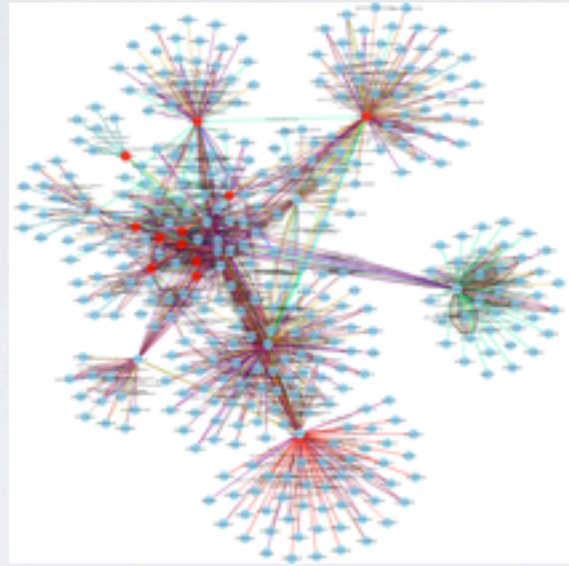
communication networks

Also: image segmentation, classification, recommendation systems, advertisement, information retrieval, ...

community detection: applications



social networks



biological networks

communication networks

Also: image segmentation, classification, recommendation systems, advertisement, information retrieval, ...

Identify groups that are alike from similarity relationships in data sets

The stochastic block model: a random graph model with communities

The stochastic block model: a random graph model with communities

$$\mathrm{SBM}(n, p, Q)$$

The stochastic block model: a random graph model with communities

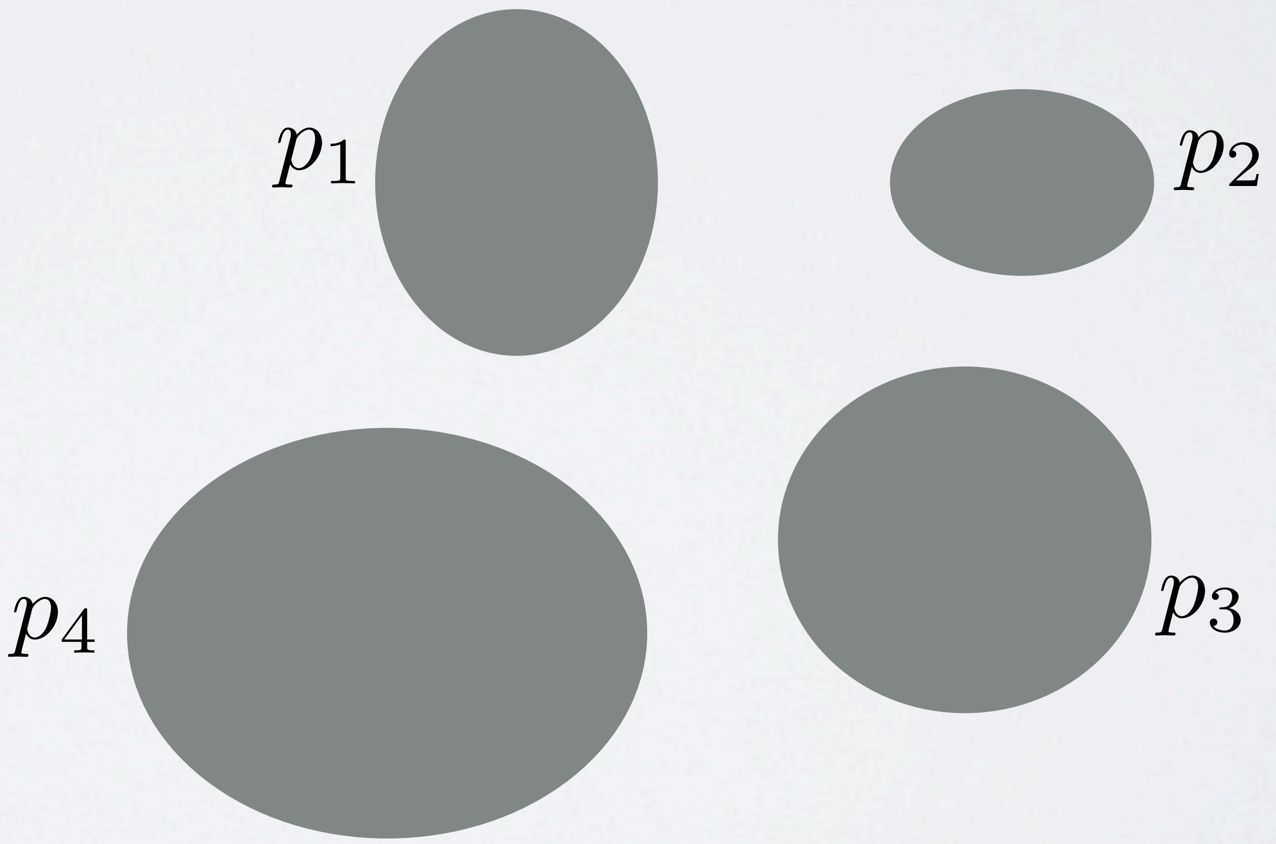
$$\mathrm{SBM}(n, p, Q)$$

$p = (p_1, \ldots, p_k)$ <- probability vector = relative size of the communities

The stochastic block model: a random graph model with communities

$$\mathrm{SBM}(n, p, Q)$$

$p = (p_1, \ldots, p_k)$ <- probability vector = relative size of the communities



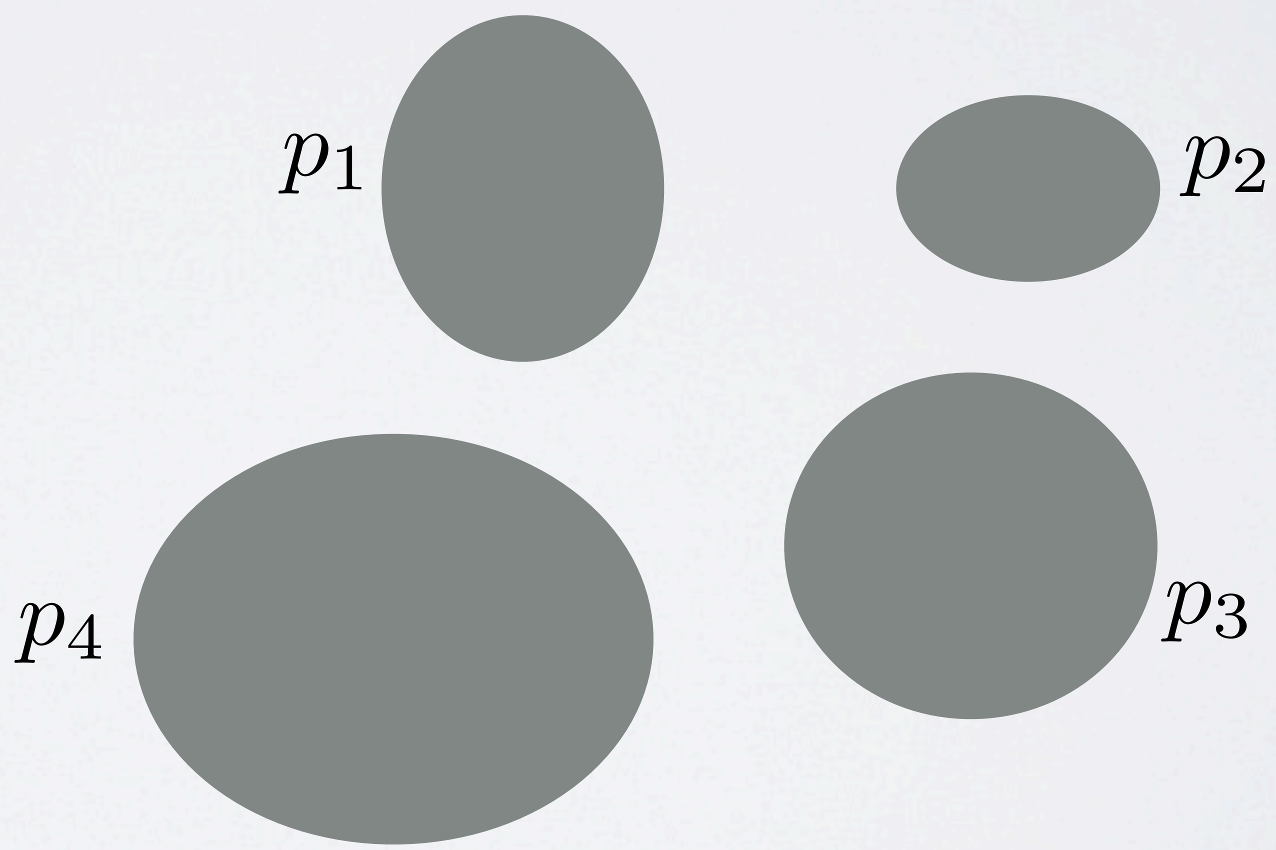
$$k = 4$$

The stochastic block model: a random graph model with communities

$$\mathrm{SBM}(n, p, Q)$$

$$P = \mathrm{diag}(p)$$

$p = (p_1, \ldots, p_k)$ <- probability vector = relative size of the communities



$$k = 4$$

The stochastic block model: a random graph model with communities

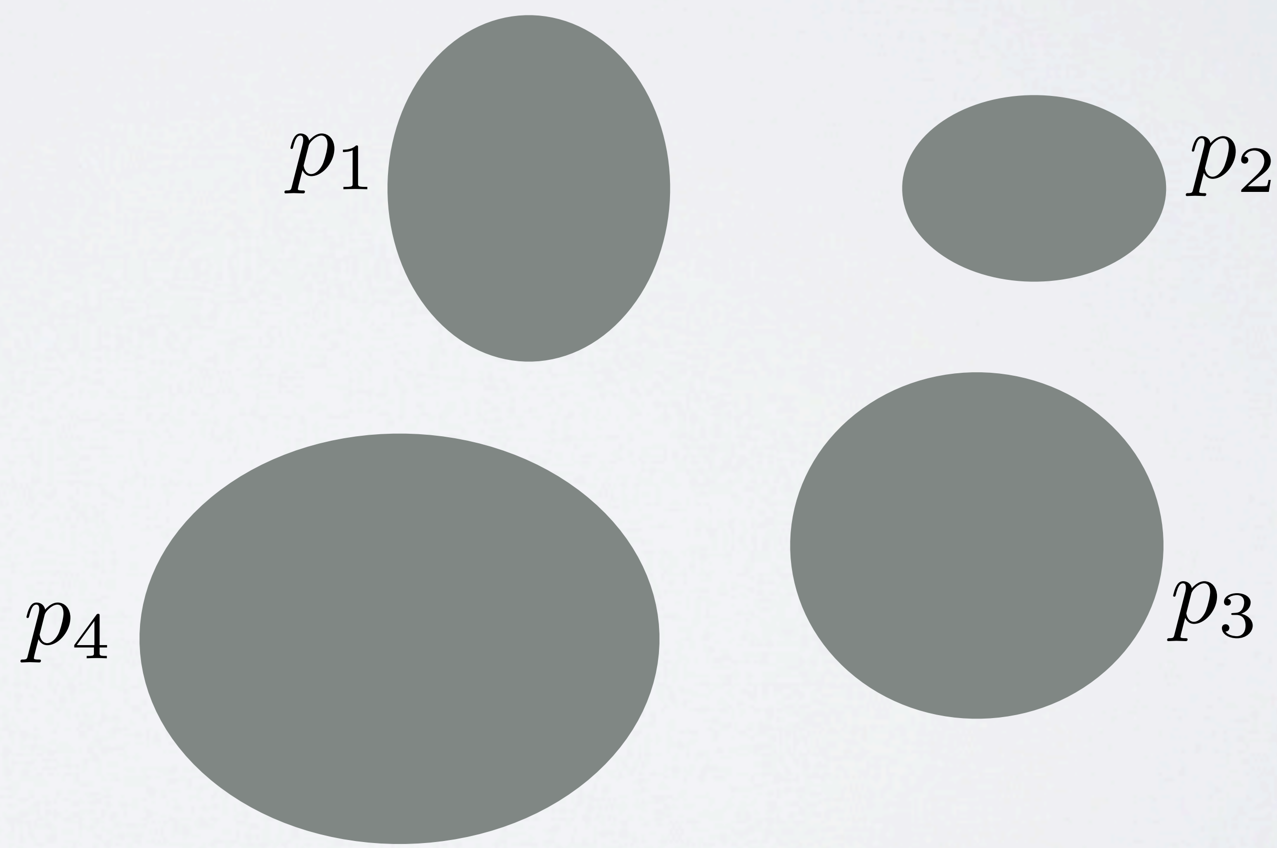
$$\mathrm{SBM}(n, p, Q)$$

$$P = \mathrm{diag}(p)$$

$$p = (p_1, \ldots, p_k)$$ <- probability vector = relative size of the communities

$$Q = \begin{pmatrix} Q_{11} & \cdots & Q_{1k} \\ \vdots & \ddots & \vdots \\ Q_{k1} & \cdots & Q_{kk} \end{pmatrix}$$

<- symmetric matrix with entries in [0,1]
= connectivity among communities



$$k = 4$$

The stochastic block model: a random graph model with communities

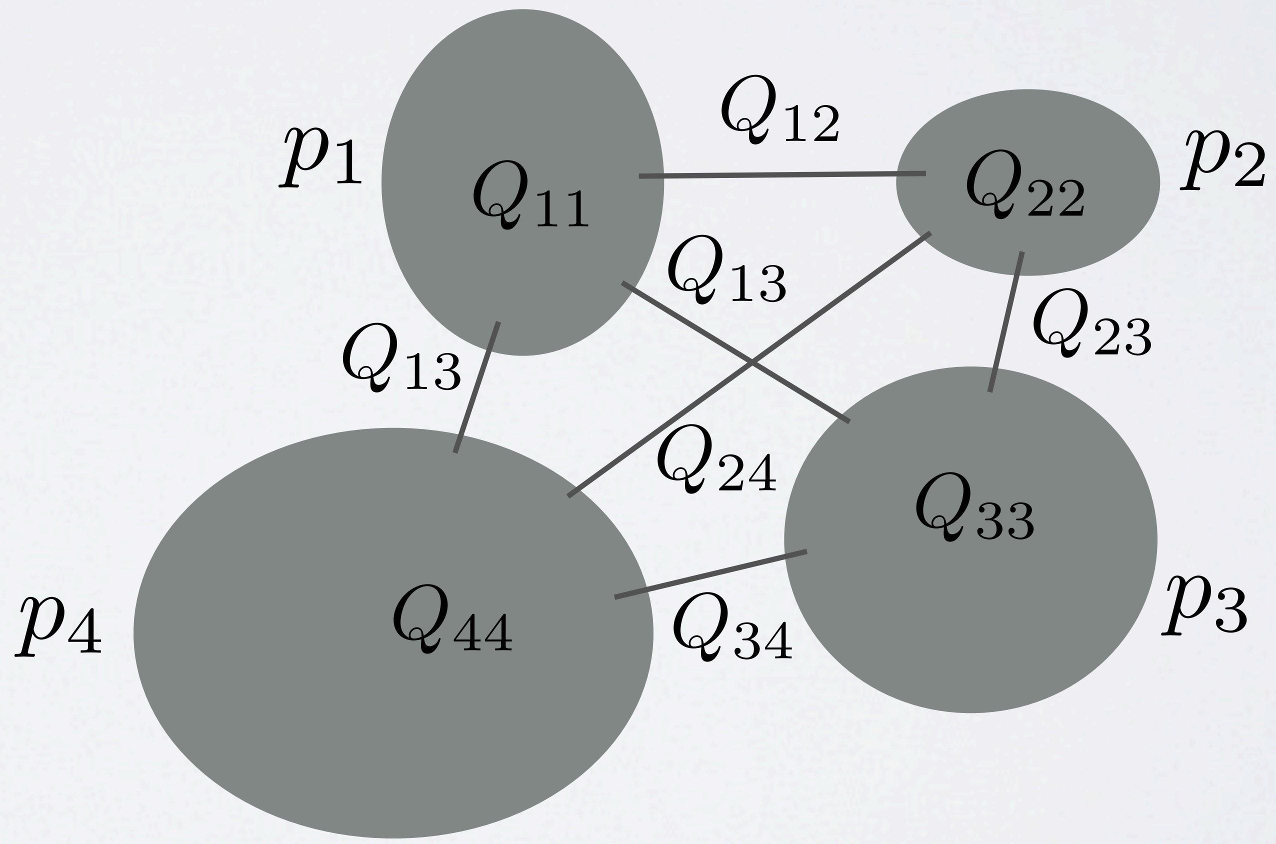
$$\mathrm{SBM}(n, p, Q)$$

$$P = \mathrm{diag}(p)$$

$p = (p_1, \ldots, p_k)$ <- probability vector = relative size of the communities

$$Q = \begin{pmatrix} Q_{11} & \cdots & Q_{1k} \\ \vdots & \ddots & \vdots \\ Q_{k1} & \cdots & Q_{kk} \end{pmatrix}$$

<- symmetric matrix with entries in [0,1]
= connectivity among communities



$$k = 4$$

The stochastic block model: a random graph model with communities

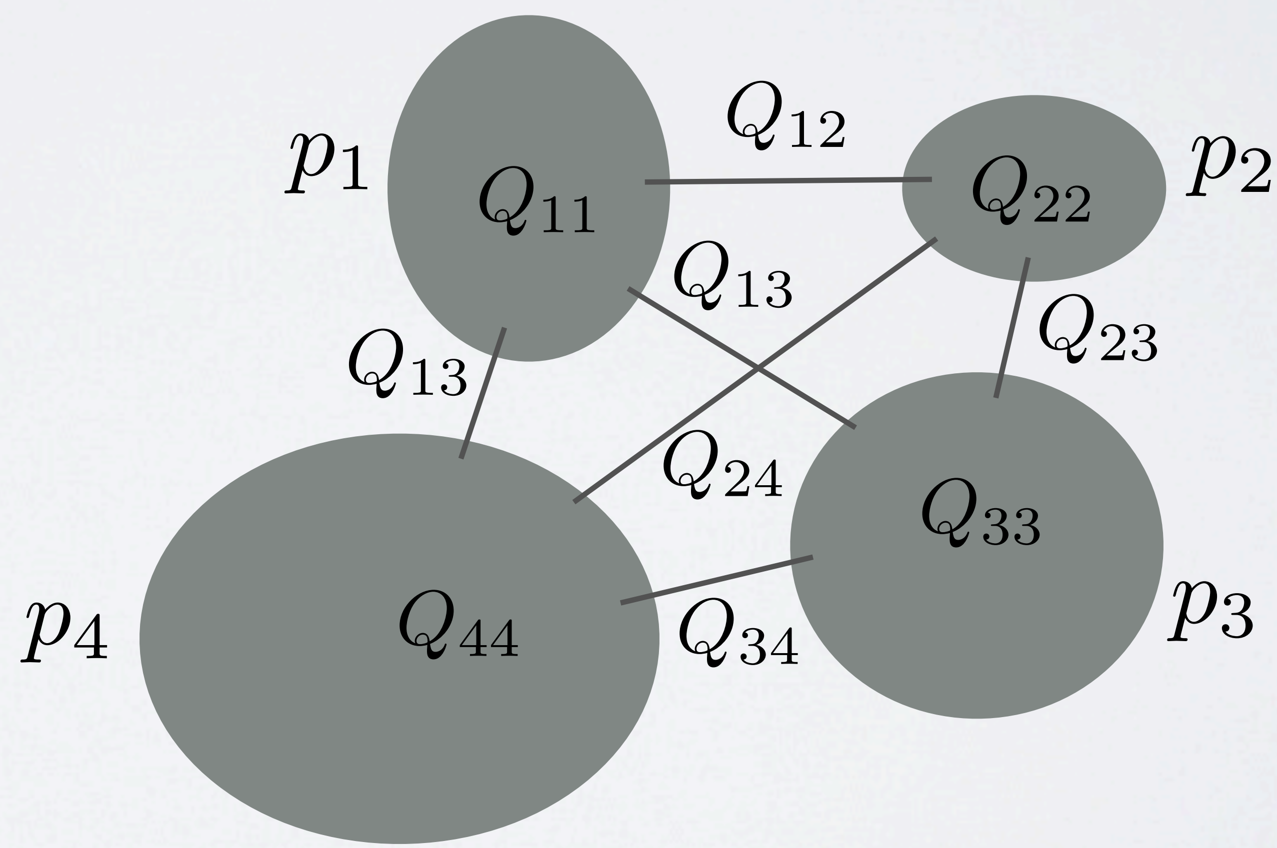
$$\mathrm{SBM}(n, p, Q)$$

$P = \mathrm{diag}(p)$

$p = (p_1, \ldots, p_k)$ <- probability vector = relative size of the communities

$$Q = \begin{pmatrix} Q_{11} & \cdots & Q_{1k} \\ \vdots & \ddots & \vdots \\ Q_{k1} & \cdots & Q_{kk} \end{pmatrix}$$

<- symmetric matrix with entries in [0,1]
= connectivity among communities



$k = 4$

The DMC of clustering..?

The stochastic block model: a random graph model with communities

$$\mathrm{SBM}(n, p, Q)$$

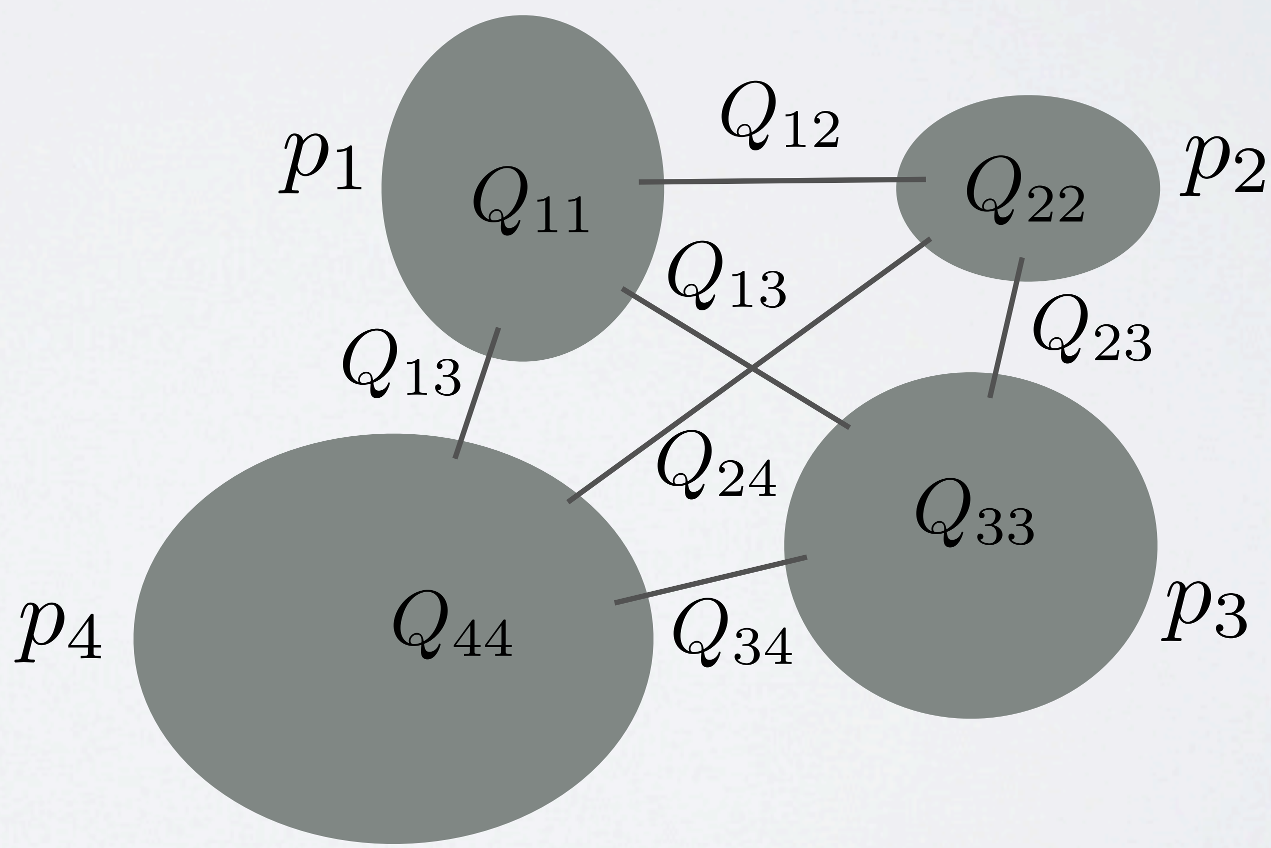
$P = \mathrm{diag}(p)$

$p = (p_1, \ldots, p_k)$ <- probability vector = relative size of the communities

$$Q = \begin{pmatrix} Q_{11} & \cdots & Q_{1k} \\ \vdots & \ddots & \vdots \\ Q_{k1} & \cdots & Q_{kk} \end{pmatrix}$$

<- symmetric matrix with entries in [0,1]
= connectivity among communities

The DMC of clustering..?
nice and reasonable model



$k = 4$

Quiz:

If a node is in community i, how many neighbors does it have in expectation in community j ?

Quiz:

If a node is in community i, how many neighbors does it have in expectation in community j ?

$$np_jQ_{ij}$$

Quiz:

If a node is in community i, how many neighbors does it have in expectation in community j ?

$$np_j Q_{ij}$$

$$\left(\quad PQ \quad \right)$$

Problem: when and how can we recover the clusters from the graph?

Problem: when and how can we recover the clusters from the graph?

for which p and Q
(w.h.p. in n)

Problem: when and how can we recover the clusters from the graph?

for which p and Q
(w.h.p. in n)

efficient
algorithms

Problem: when and how can we recover the clusters from the graph?

for which p and Q
(w.h.p. in n)

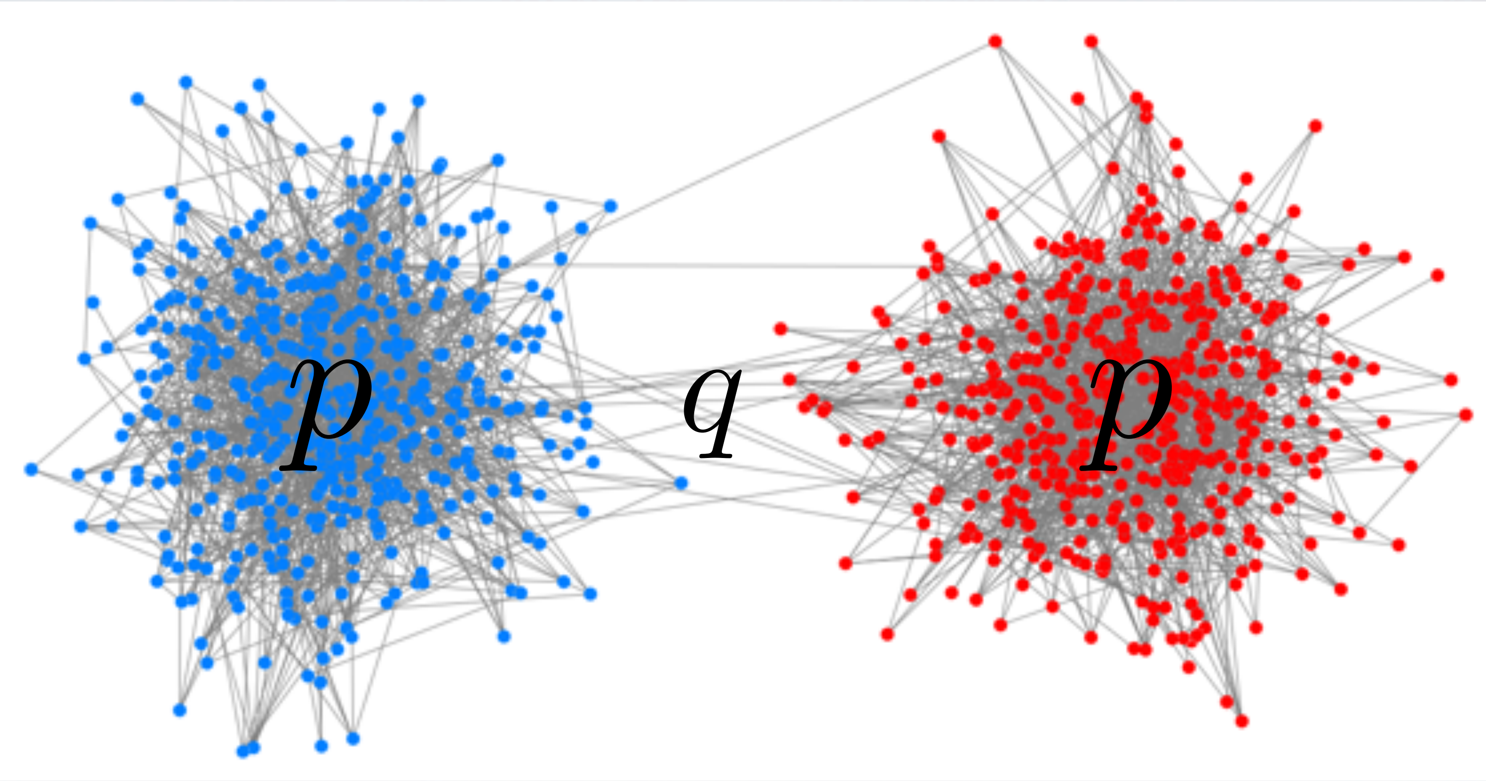
efficient
algorithms

Next:

- warm up: two symmetric communities
- new results: partial and exact recovery in the general SBM
- analogy with the channel coding theorem
- some real data

SBM with two symmetric communities (planted bisection model)

SBM with two symmetric communities (planted bisection model)



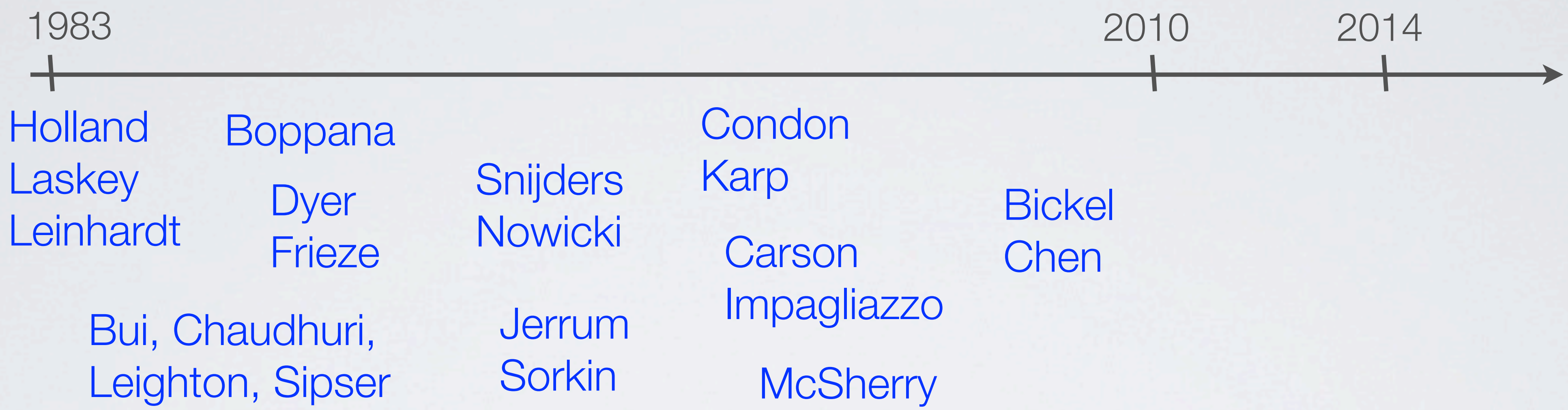
$\frac{n}{2}$ $\frac{n}{2}$

$$p_1 = p_2 = 1/2$$

$$Q_{11} = Q_{22} = p \qquad Q_{12} = q$$

Some history

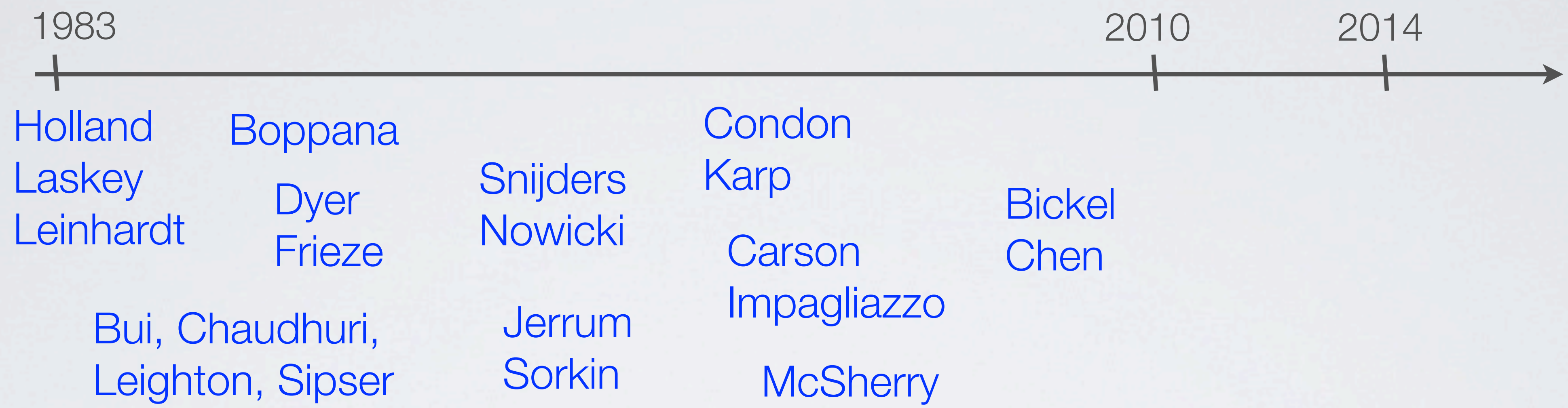
Recovery



Some history

$$\mathbb{P}(\hat{X}^n = X^n) \to 1$$

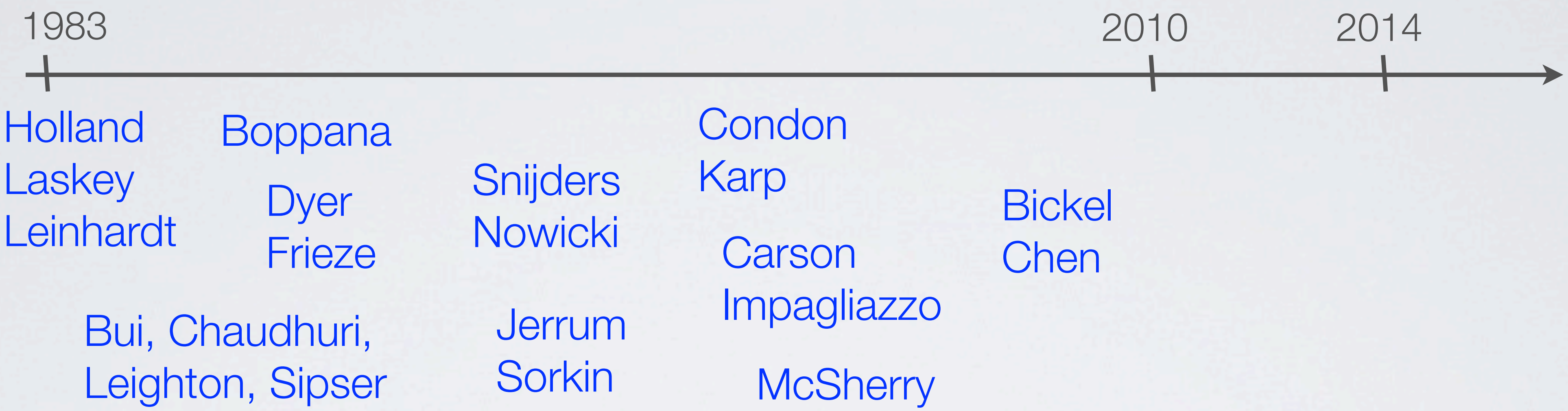
Recovery



Some history

$$\mathbb{P}(\hat{X}^n = X^n) \to 1$$

Recovery



Bui, Chaudhuri, Leighton, Sipser '84	min-cut method	$p = \Omega(1/n), q = o(n^{-1-4/((p+q)n)})$
Boppana '87	spectral meth.	$(p-q)/\sqrt{p+q} = \Omega(\sqrt{\log(n)/n})$
Dyer, Frieze '89	min-cut via degrees	$p - q = \Omega(1)$
Snijders, Nowicki '97	EM algo.	$p - q = \Omega(1)$
Jerrum, Sorkin '98	Metropolis aglo.	$p - q = \Omega(n^{-1/6+\epsilon})$
Condon, Karp '99	augmentation algo.	$p - q = \Omega(n^{-1/2+\epsilon})$
Carson, Impagliazzo '01	hill-climbing algo.	$p - q = \Omega(n^{-1/2}\log^4(n))$
Mcsherry '01	spectral meth.	$(p-q)/\sqrt{p} \geq \Omega(\sqrt{\log(n)/n})$
Bickel, Chen '09	N-G modularity	$(p-q)/\sqrt{p+q} = \Omega(\log(n)/\sqrt{n})$
Rohe, Chatterjee, Yu '11	spectral meth.	$p - q = \Omega(1)$

Some history

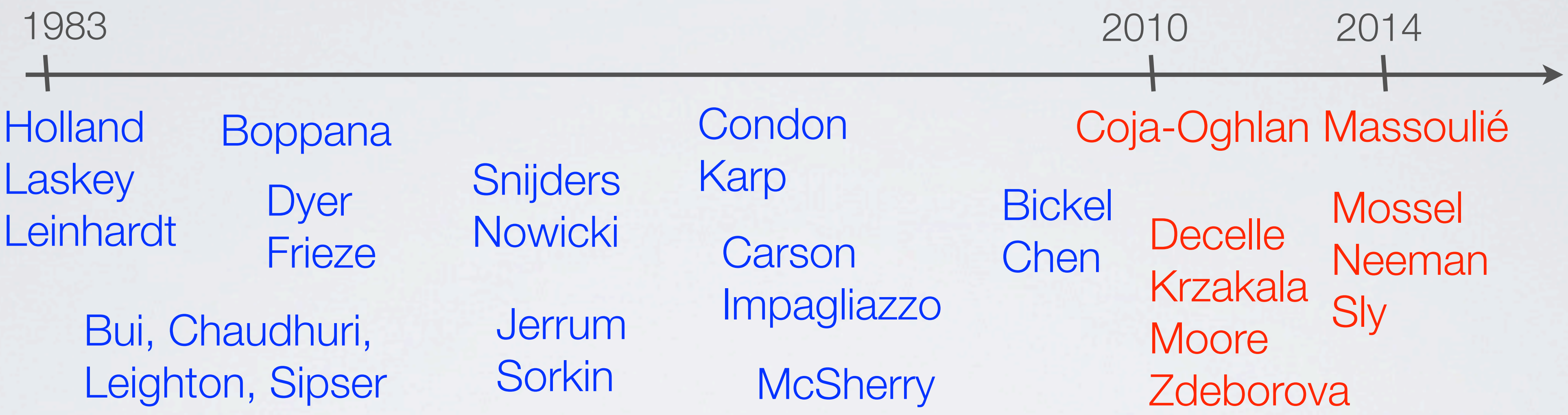
$$\mathbb{P}(\hat{X}^n = X^n) \to 1$$

Bui, Chaudhuri, Leighton, Sipser '84	min-cut method	$p = \Omega(1/n), q = o(n^{-1-4/((p+q)n)})$
Boppana '87	spectral meth.	$(p-q)/\sqrt{p+q} = \Omega(\sqrt{\log(n)/n})$
Dyer, Frieze '89	min-cut via degrees	$p - q = \Omega(1)$
Snijders, Nowicki '97	EM algo.	$p - q = \Omega(1)$
Jerrum, Sorkin '98	Metropolis aglo.	$p - q = \Omega(n^{-1/6+\epsilon})$
Condon, Karp '99	augmentation algo.	$p - q = \Omega(n^{-1/2+\epsilon})$
Carson, Impagliazzo '01	hill-climbing algo.	$p - q = \Omega(n^{-1/2}\log^4(n))$
Mcsherry '01	spectral meth.	$(p-q)/\sqrt{p} \geq \Omega(\sqrt{\log(n)/n})$
Bickel, Chen '09	N-G modularity	$(p-q)/\sqrt{p+q} = \Omega(\log(n)/\sqrt{n})$
Rohe, Chatterjee, Yu '11	spectral meth.	$p - q = \Omega(1)$

Some history

$$\mathbb{P}(\hat{X}^n = X^n) \to 1 \qquad \mathbb{P}\left(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon\right) \to 1$$

Recovery | **Detection**



Bui, Chaudhuri, Leighton, Sipser '84	min-cut method	$p = \Omega(1/n), q = o(n^{-1-4/((p+q)n)})$
Boppana '87	spectral meth.	$(p-q)/\sqrt{p+q} = \Omega(\sqrt{\log(n)/n})$
Dyer, Frieze '89	min-cut via degrees	$p - q = \Omega(1)$
Snijders, Nowicki '97	EM algo.	$p - q = \Omega(1)$
Jerrum, Sorkin '98	Metropolis aglo.	$p - q = \Omega(n^{-1/6+\epsilon})$
Condon, Karp '99	augmentation algo.	$p - q = \Omega(n^{-1/2+\epsilon})$
Carson, Impagliazzo '01	hill-climbing algo.	$p - q = \Omega(n^{-1/2}\log^4(n))$
Mcsherry '01	spectral meth.	$(p-q)/\sqrt{p} \geq \Omega(\sqrt{\log(n)/n})$
Bickel, Chen '09	N-G modularity	$(p-q)/\sqrt{p+q} = \Omega(\log(n)/\sqrt{n})$
Rohe, Chatterjee, Yu '11	spectral meth.	$p - q = \Omega(1)$

Some history

$\mathbb{P}(\hat{X}^n = X^n) \to 1$ $\qquad$ $\mathbb{P}(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon) \to 1$

Recovery $\qquad$ **Detection**

1983 — 2010 — 2014

Holland
Laskey
Leinhardt

Boppana

Dyer
Frieze

Bui, Chaudhuri,
Leighton, Sipser

Snijders
Nowicki

Jerrum
Sorkin

Condon
Karp

Carson
Impagliazzo

McSherry

Bickel
Chen

Coja-Oghlan

Decelle
Krzakala
Moore
Zdeborova

Massoulié

Mossel
Neeman
Sly

Bui, Chaudhuri, Leighton, Sipser '84	min-cut method	$p = \Omega(1/n), q = o(n^{-1-4/((p+q)n)})$
Boppana '87	spectral meth.	$(p-q)/\sqrt{p+q} = \Omega(\sqrt{\log(n)/n})$
Dyer, Frieze '89	min-cut via degrees	$p - q = \Omega(1)$
Snijders, Nowicki '97	EM algo.	$p - q = \Omega(1)$
Jerrum, Sorkin '98	Metropolis aglo.	$p - q = \Omega(n^{-1/6+\epsilon})$
Condon, Karp '99	augmentation algo.	$p - q = \Omega(n^{-1/2+\epsilon})$
Carson, Impagliazzo '01	hill-climbing algo.	$p - q = \Omega(n^{-1/2}\log^4(n))$
Mcsherry '01	spectral meth.	$(p-q)/\sqrt{p} \geq \Omega(\sqrt{\log(n)/n})$
Bickel, Chen '09	N-G modularity	$(p-q)/\sqrt{p+q} = \Omega(\log(n)/\sqrt{n})$
Rohe, Chatterjee, Yu '11	spectral meth.	$p - q = \Omega(1)$

$$p = \frac{a}{n}, q = \frac{b}{n}$$

Detection iff $\;(a-b)^2 > 2(a+b)$

Some history

$$\mathbb{P}(\hat{X}^n = X^n) \to 1 \qquad \mathbb{P}\left(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon\right) \to 1$$

Recovery

Detection

1983 — 2010 — 2014

Holland, Laskey, Leinhardt

Boppana

Dyer, Frieze

Bui, Chaudhuri, Leighton, Sipser

Snijders, Nowicki

Jerrum, Sorkin

Condon, Karp

Carson, Impagliazzo

McSherry

Bickel, Chen

Coja-Oghlan

Decelle, Krzakala, Moore, Zdeborova

Massoulié

Mossel, Neeman, Sly

Bui, Chaudhuri, Leighton, Sipser '84	min-cut method	$p = \Omega(1/n), q = o(n^{-1-4/((p+q)n)})$
Boppana '87	spectral meth.	$(p-q)/\sqrt{p+q} = \Omega(\sqrt{\log(n)/n})$
Dyer, Frieze '89	min-cut via degrees	$p - q = \Omega(1)$
Snijders, Nowicki '97	EM algo.	$p - q = \Omega(1)$
Jerrum, Sorkin '98	Metropolis aglo.	$p - q = \Omega(n^{-1/6+\epsilon})$
Condon, Karp '99	augmentation algo.	$p - q = \Omega(n^{-1/2+\epsilon})$
Carson, Impagliazzo '01	hill-climbing algo.	$p - q = \Omega(n^{-1/2}\log^4(n))$
Mcsherry '01	spectral meth.	$(p-q)/\sqrt{p} \geq \Omega(\sqrt{\log(n)/n})$
Bickel, Chen '09	N-G modularity	$(p-q)/\sqrt{p+q} = \Omega(\log(n)/\sqrt{n})$
Rohe, Chatterjee, Yu '11	spectral meth.	$p - q = \Omega(1)$

$$p = \frac{a}{n}, q = \frac{b}{n}$$

Detection iff $(a-b)^2 > 2(a+b)$

Some history

$\mathbb{P}(\hat{X}^n = X^n) \to 1$

$\mathbb{P}(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon) \to 1$

Recovery | **Detection**

1983 | 2010 | 2014

Holland
Laskey
Leinhardt

Boppana

Dyer
Frieze

Bui, Chaudhuri,
Leighton, Sipser

Snijders
Nowicki

Jerrum
Sorkin

Condon
Karp

Carson
Impagliazzo

McSherry

Bickel
Chen

Coja-Oghlan

Decelle
Krzakala
Moore
Zdeborova

Massoulié

Mossel
Neeman
Sly

$$p = \frac{a}{n}, q = \frac{b}{n}$$

Detection iff $(a-b)^2 > 2(a+b)$

Some history

$\mathbb{P}(\hat{X}^n = X^n) \to 1$

$\mathbb{P}\left(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon\right) \to 1$

Recovery

Detection

1983 — 2010 — 2014

Holland
Laskey
Leinhardt

Boppana

Dyer
Frieze

Bui, Chaudhuri,
Leighton, Sipser

Snijders
Nowicki

Jerrum
Sorkin

Condon
Karp

Carson
Impagliazzo

McSherry

Bickel
Chen

Coja-Oghlan

Decelle
Krzakala
Moore
Zdeborova

Massoulié

Mossel
Neeman
Sly

Abbe-Bandeira-Hall '14

$$p = \frac{a \log(n)}{n}, q = \frac{b \log(n)}{n}$$

Recovery iff $\frac{a+b}{2} \geq 1 + \sqrt{ab}$

$$p = \frac{a}{n}, q = \frac{b}{n}$$

Detection iff $(a-b)^2 > 2(a+b)$

Some history

$$\mathbb{P}(\hat{X}^n = X^n) \to 1 \qquad \mathbb{P}\left(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon\right) \to 1$$

Recovery

Detection

1983 — 2010 — 2014

Holland, Laskey, Leinhardt

Boppana

Dyer, Frieze

Bui, Chaudhuri, Leighton, Sipser

Snijders, Nowicki

Jerrum, Sorkin

Condon, Karp

Carson, Impagliazzo

McSherry

Bickel, Chen

Coja-Oghlan

Decelle, Krzakala, Moore, Zdeborova

Massoulié

Mossel, Neeman, Sly

Abbe-Bandeira-Hall '14

Mossel-Neeman-Sly '14

$a_n, b_n = \Theta(1)$

$$p = \frac{a\log(n)}{n}, q = \frac{b\log(n)}{n}$$

Recovery iff $\frac{a+b}{2} \geq 1 + \sqrt{ab}$

$$p = \frac{a}{n}, q = \frac{b}{n}$$

Detection iff $(a-b)^2 > 2(a+b)$

Some history

$$\mathbb{P}(\hat{X}^n = X^n) \to 1 \qquad \mathbb{P}\left(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon\right) \to 1$$

Recovery **Detection**

1983 2010 2014

Holland, Laskey, Leinhardt

Boppana

Dyer, Frieze

Bui, Chaudhuri, Leighton, Sipser

Snijders, Nowicki

Jerrum, Sorkin

Condon, Karp

Carson, Impagliazzo

McSherry

Bickel, Chen

Coja-Oghlan

Decelle, Krzakala, Moore, Zdeborova

Massoulié

Mossel, Neeman, Sly

Abbe-Bandeira-Hall '14

Mossel-Neeman-Sly '14

$a_n, b_n = \Theta(1)$

$$p = \frac{a\log(n)}{n}, q = \frac{b\log(n)}{n}$$

Recovery iff $\frac{a+b}{2} \geq 1 + \sqrt{ab}$

$$p = \frac{a}{n}, q = \frac{b}{n}$$

Detection iff $(a-b)^2 > 2(a+b)$

In both cases we have efficient algorithms achieving the thresholds

Some history

$\mathbb{P}(\hat{X}^n = X^n) \to 1$ $\qquad$ $\mathbb{P}\left(\frac{d(\hat{X}^n, X^n)}{n} < \frac{1}{2} - \epsilon\right) \to 1$

Recovery $\qquad$ **Detection**

1983 — 2010 — 2014

Holland, Laskey, Leinhardt

Boppana

Dyer, Frieze

Bui, Chaudhuri, Leighton, Sipser

Snijders, Nowicki

Jerrum, Sorkin

Condon, Karp

Carson, Impagliazzo

McSherry

Bickel, Chen

Coja-Oghlan

Decelle, Krzakala, Moore, Zdeborova

Massoulié

Mossel, Neeman, Sly

Abbe-Bandeira-Hall '14

Mossel-Neeman-Sly '14

$a_n, b_n = \Theta(1)$

$$p = \frac{a \log(n)}{n}, q = \frac{b \log(n)}{n}$$

$$\text{Recovery iff } \frac{a+b}{2} \geq 1 + \sqrt{ab}$$

$$p = \frac{a}{n}, q = \frac{b}{n}$$

$$\text{Detection iff } (a-b)^2 > 2(a+b)$$

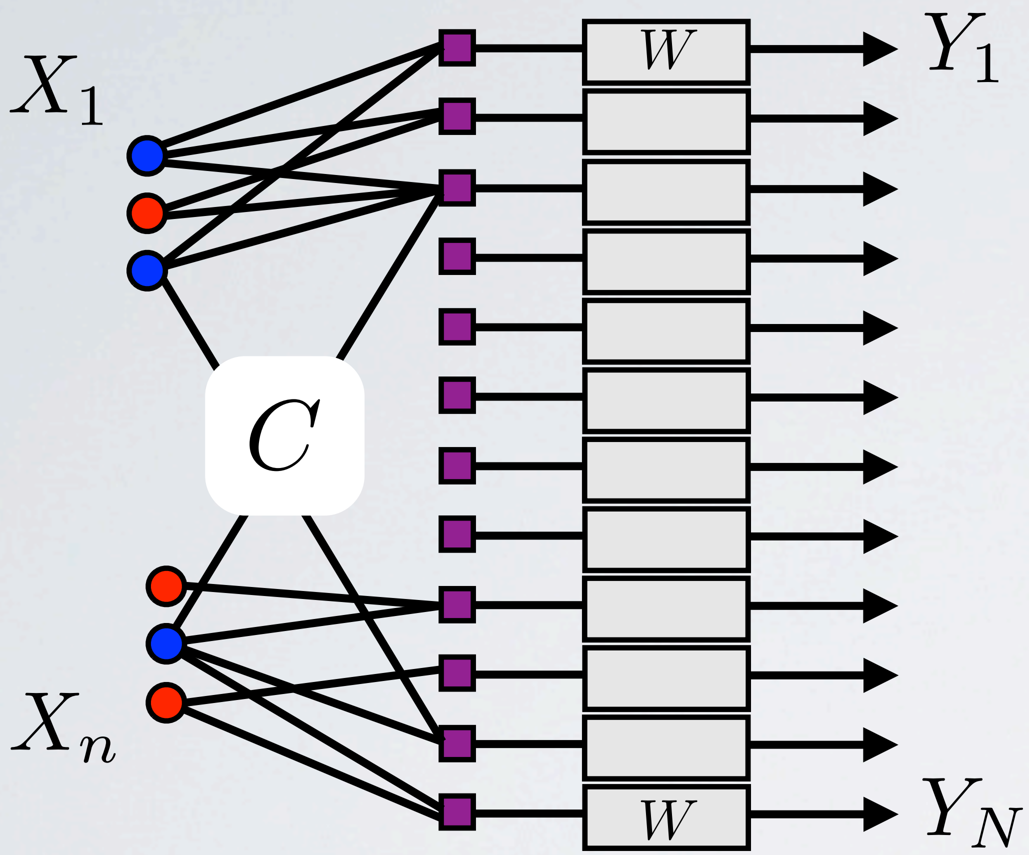
In both cases we have efficient algorithms achieving the thresholds

→ not clear for multiple communities

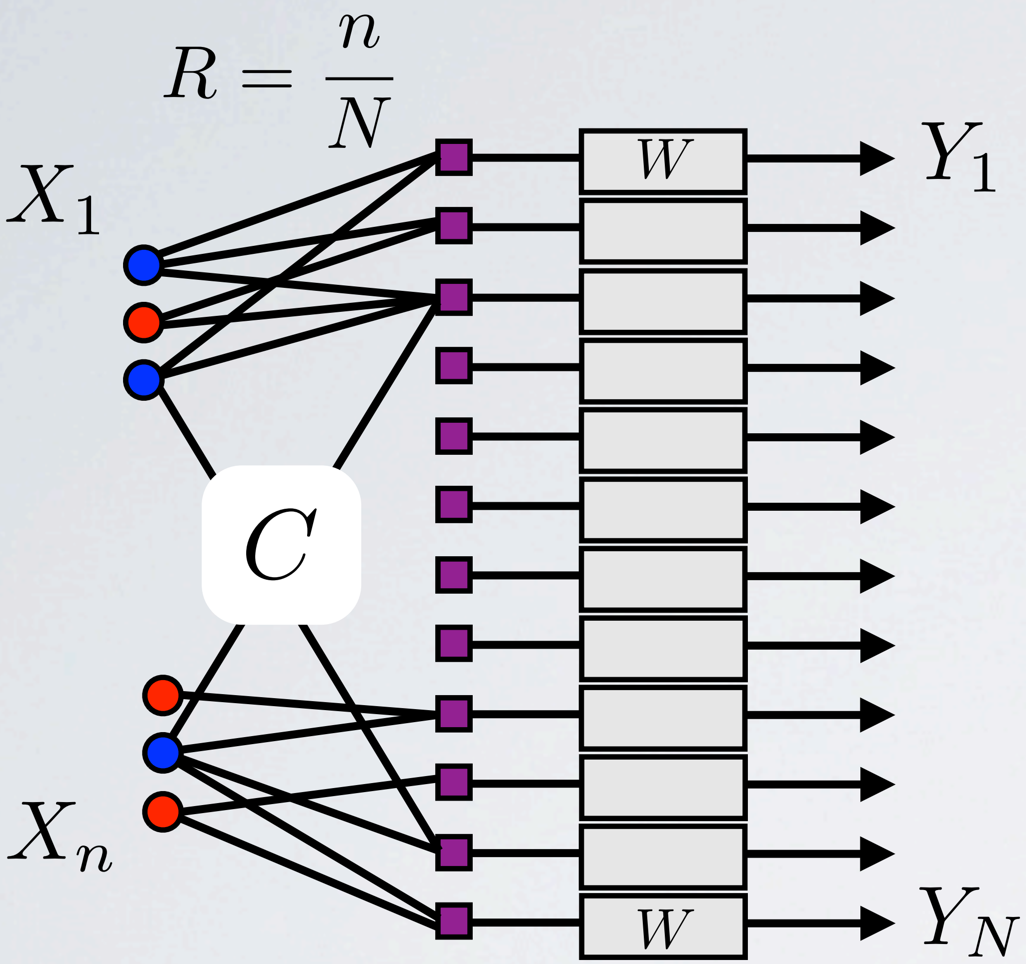
Recovery in the general SBM

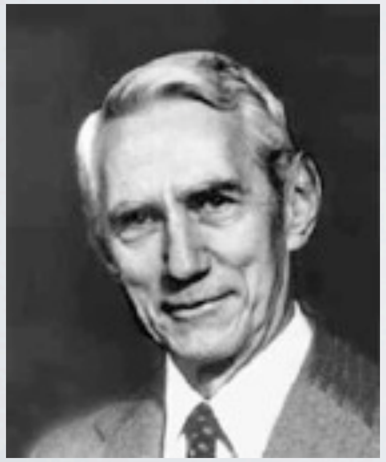
Recovery in the general SBM -> an information theoretic motivation

Recovery in the general SBM -> an information theoretic motivation



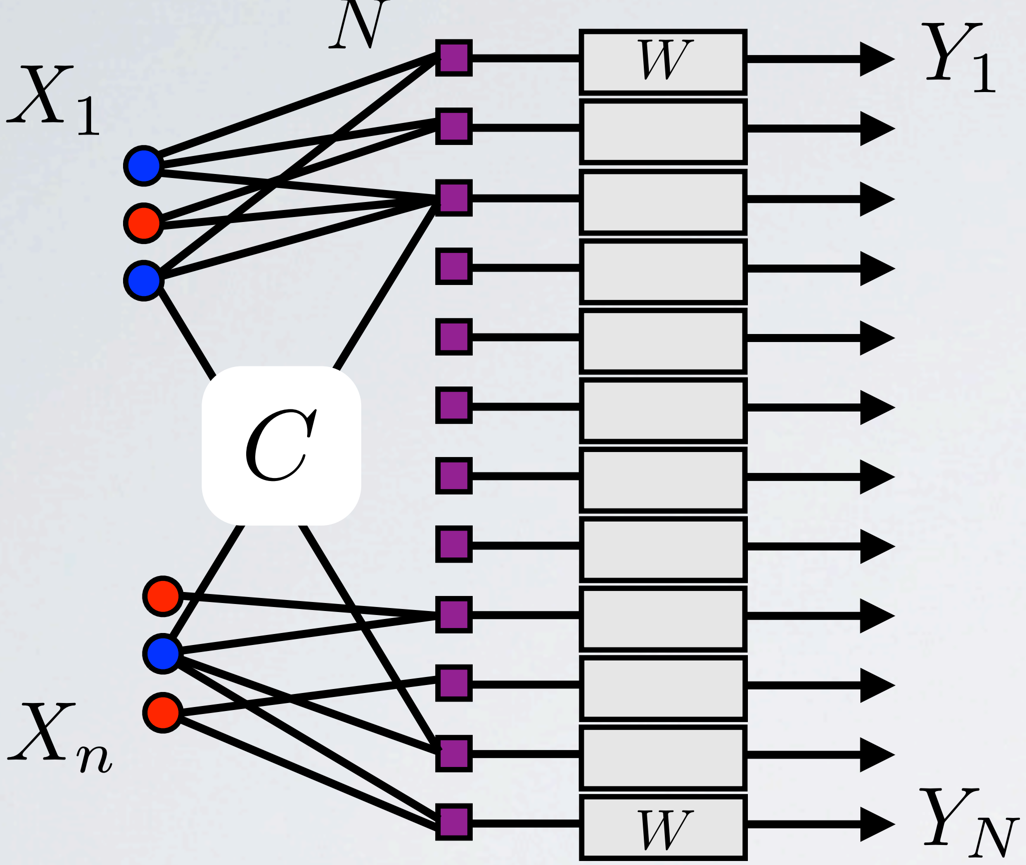
Recovery in the general SBM -> an information theoretic motivation

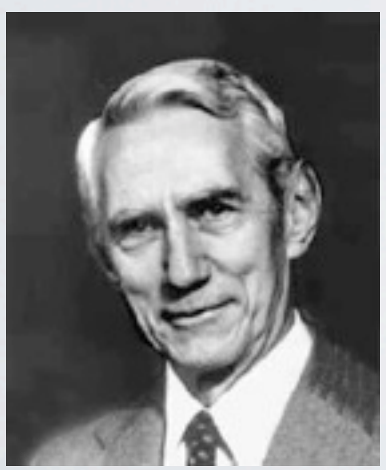




Recovery in the general SBM -> an information theoretic motivation

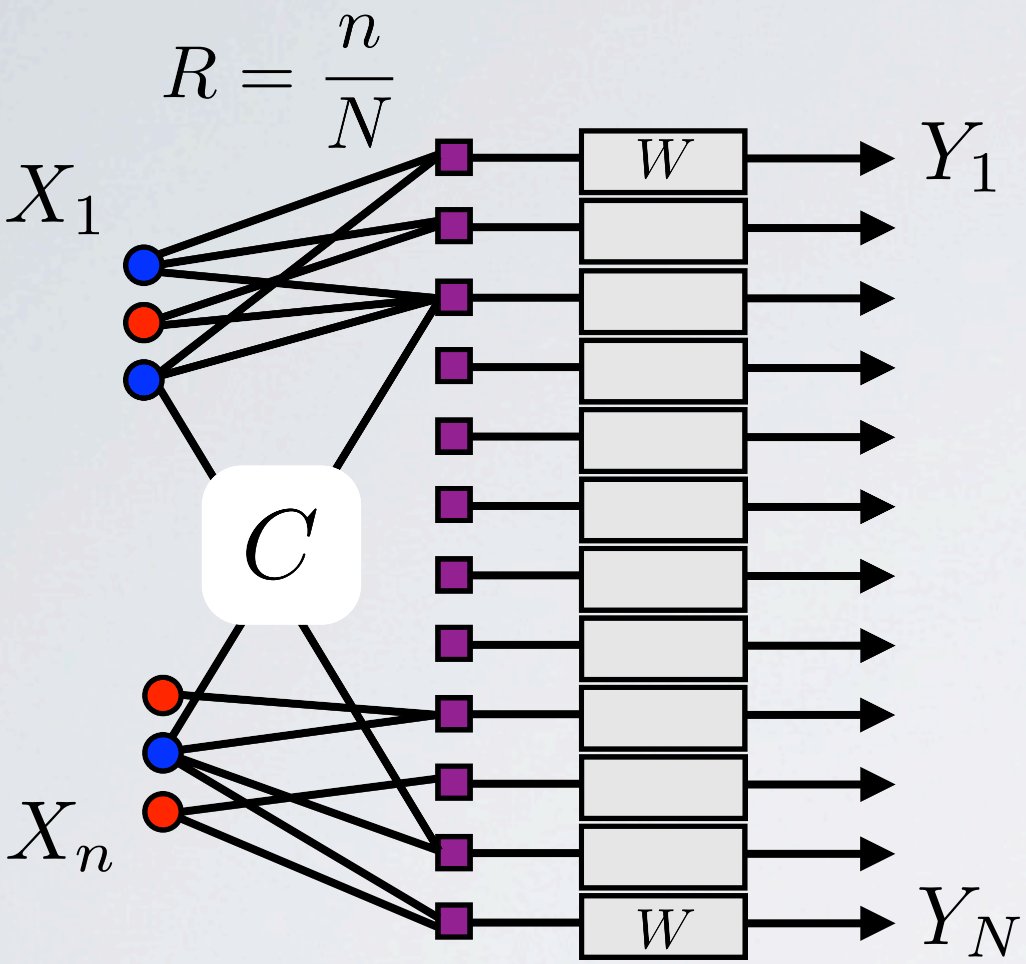
$$R = \frac{n}{N}$$

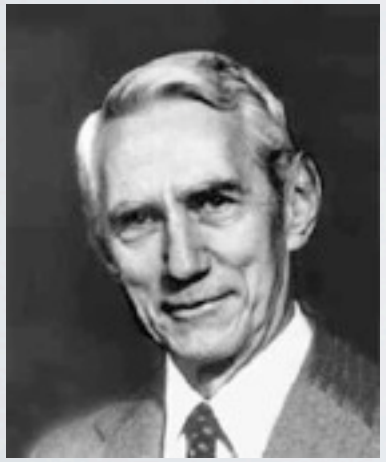




$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

Recovery in the general SBM -> an information theoretic motivation



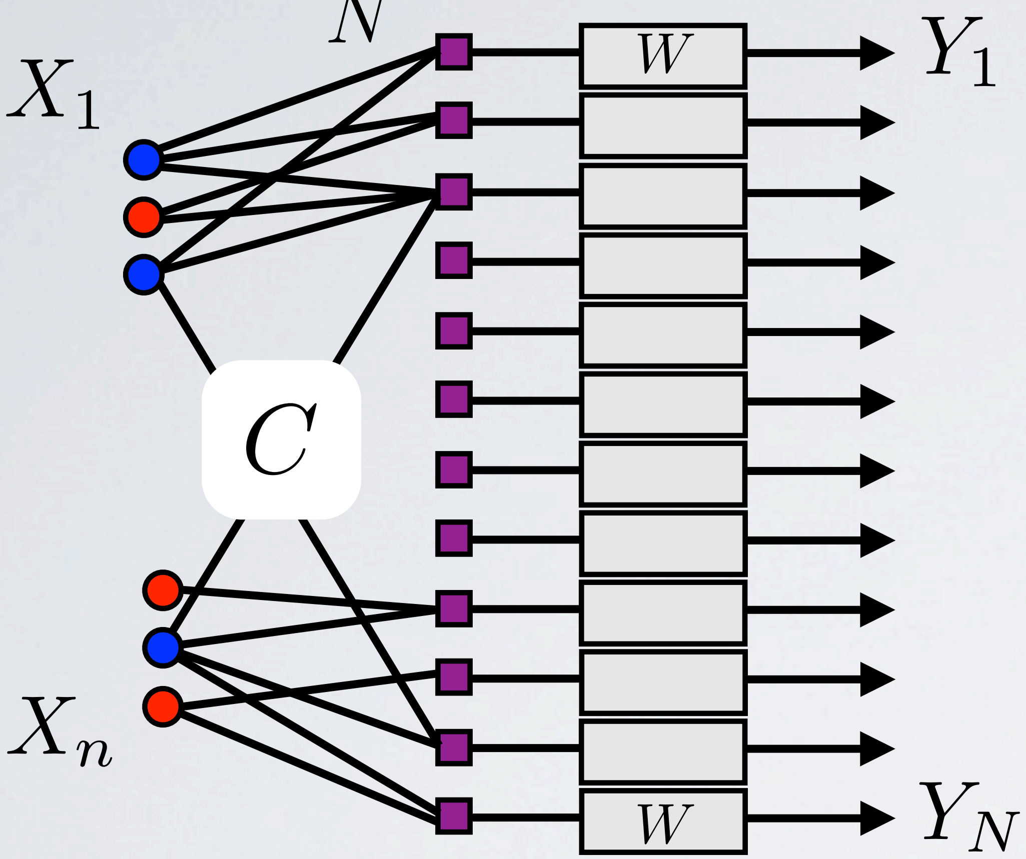


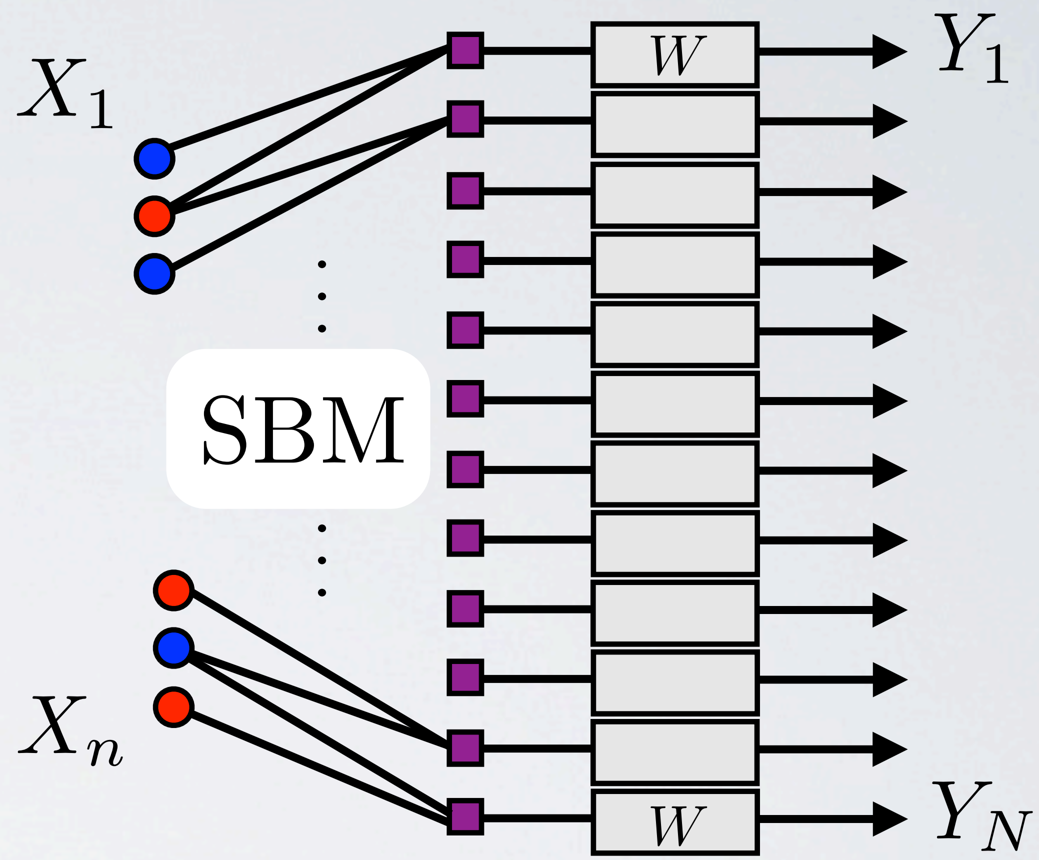
$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

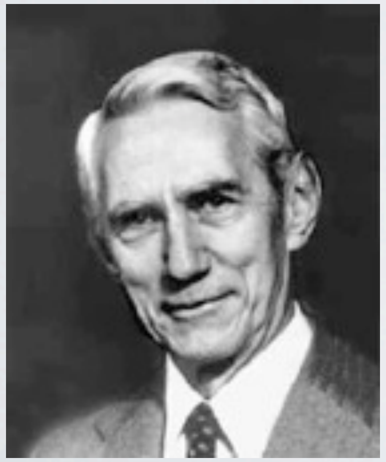
reliable comm. iff R < 1-H($\boldsymbol{\varepsilon}$)

Recovery in the general SBM -> an information theoretic motivation

$$R = \frac{n}{N}$$





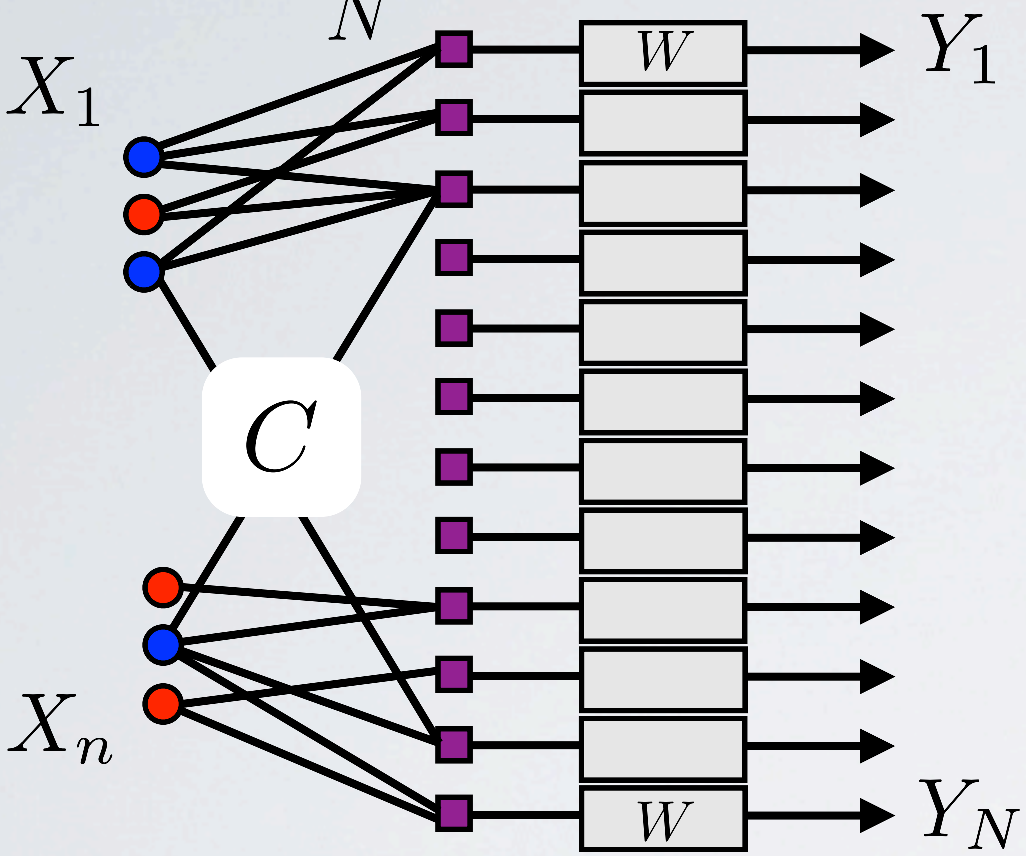


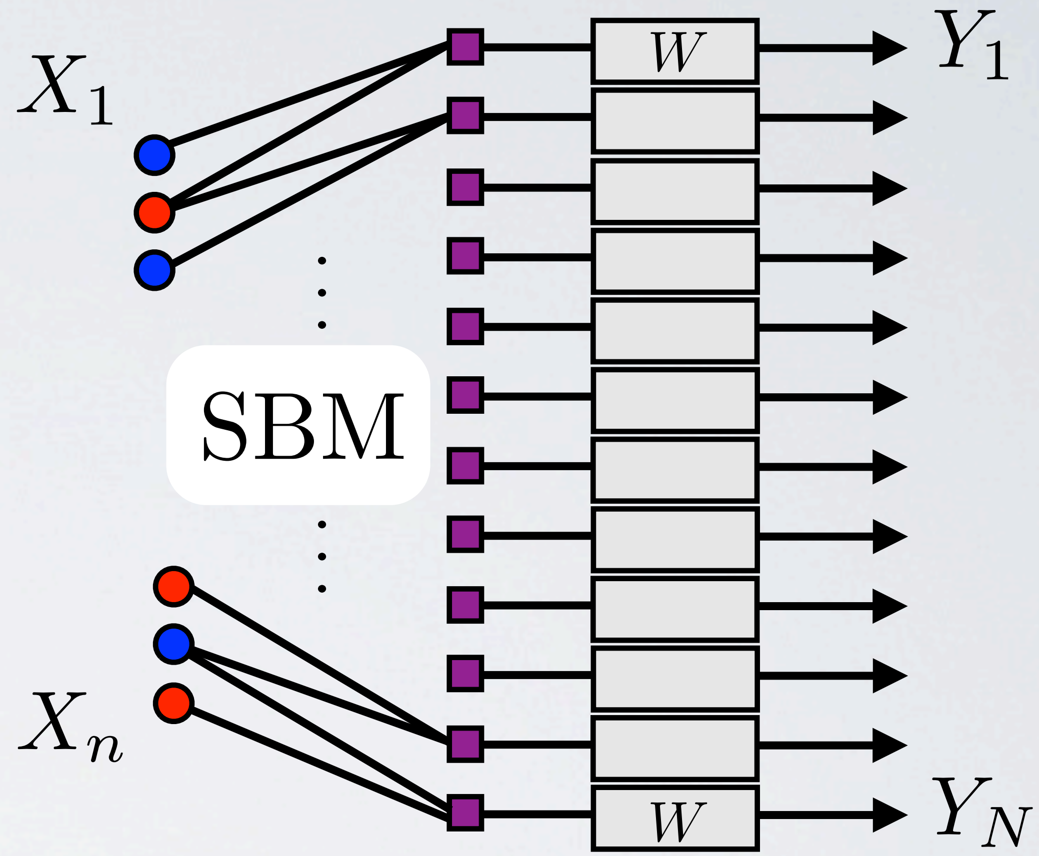
$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

reliable comm. iff R < 1-H(**ε**)

Recovery in the general SBM -> an information theoretic motivation

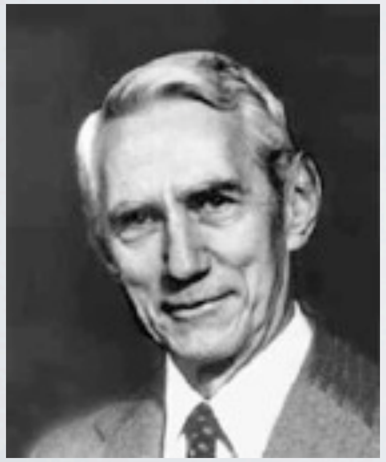
$$R = \frac{n}{N}$$





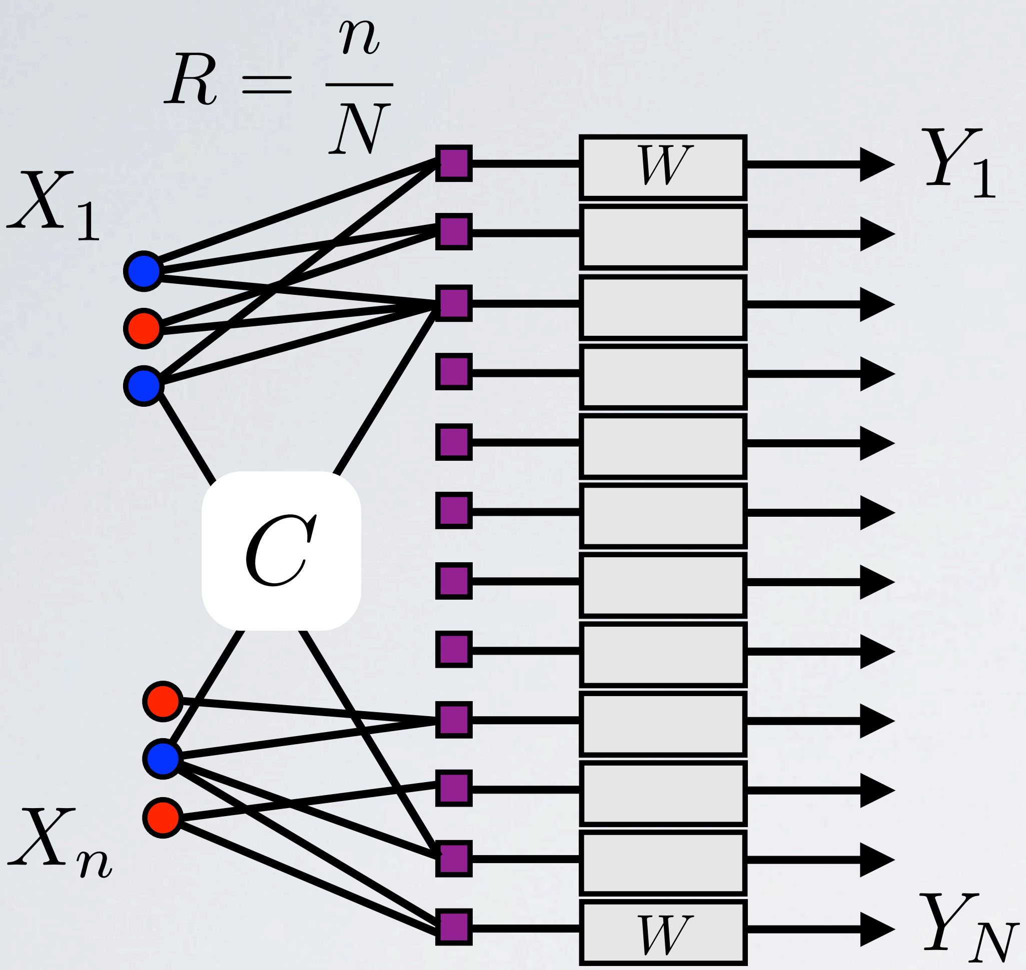
$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

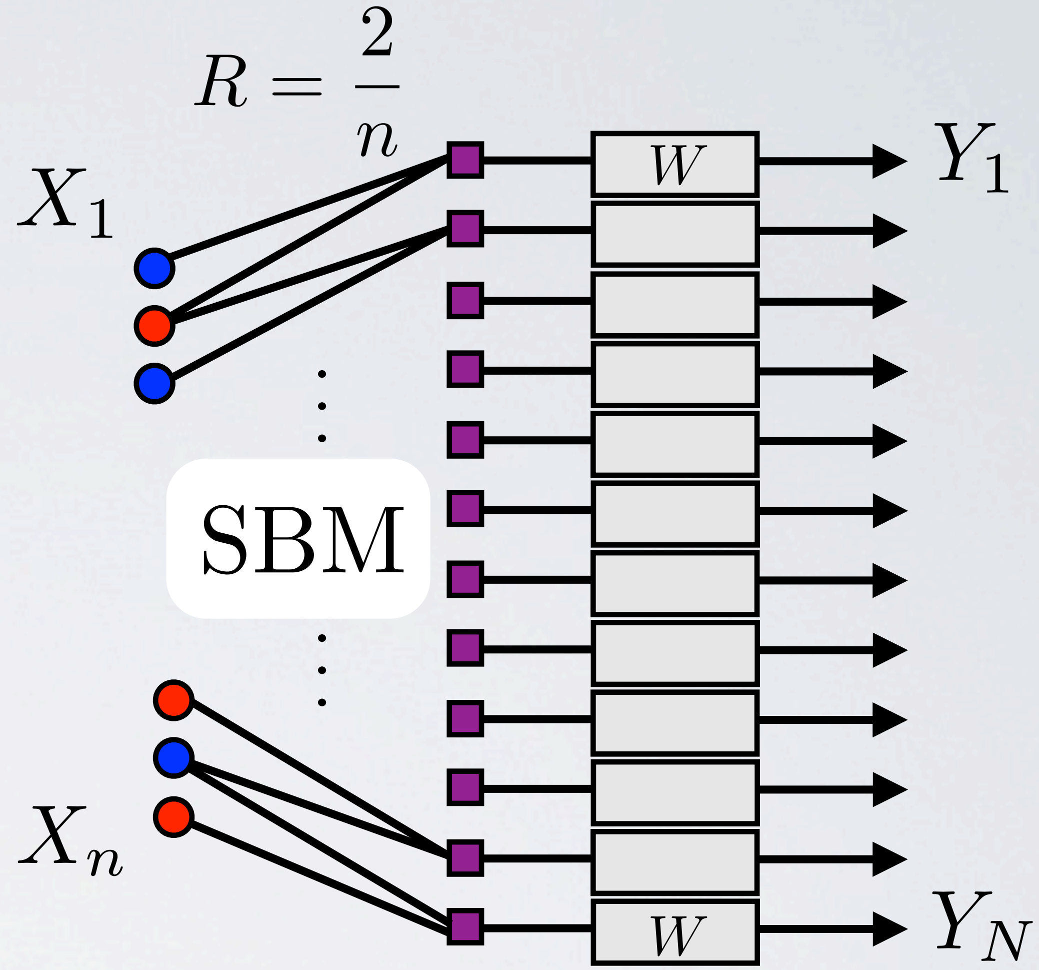
$$W = \begin{pmatrix} 1 - a\log(n)/n & a\log(n)/n \\ 1 - b\log(n)/n & b\log(n)/n \end{pmatrix}$$

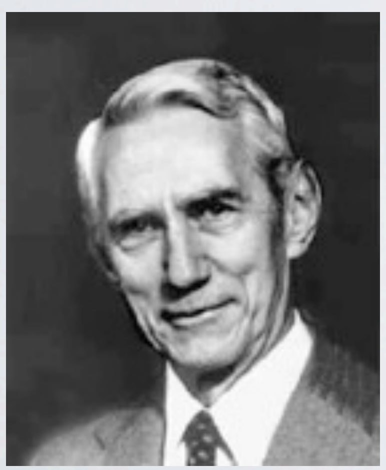


reliable comm. iff R < 1-H(**ε**)

Recovery in the general SBM -> an information theoretic motivation





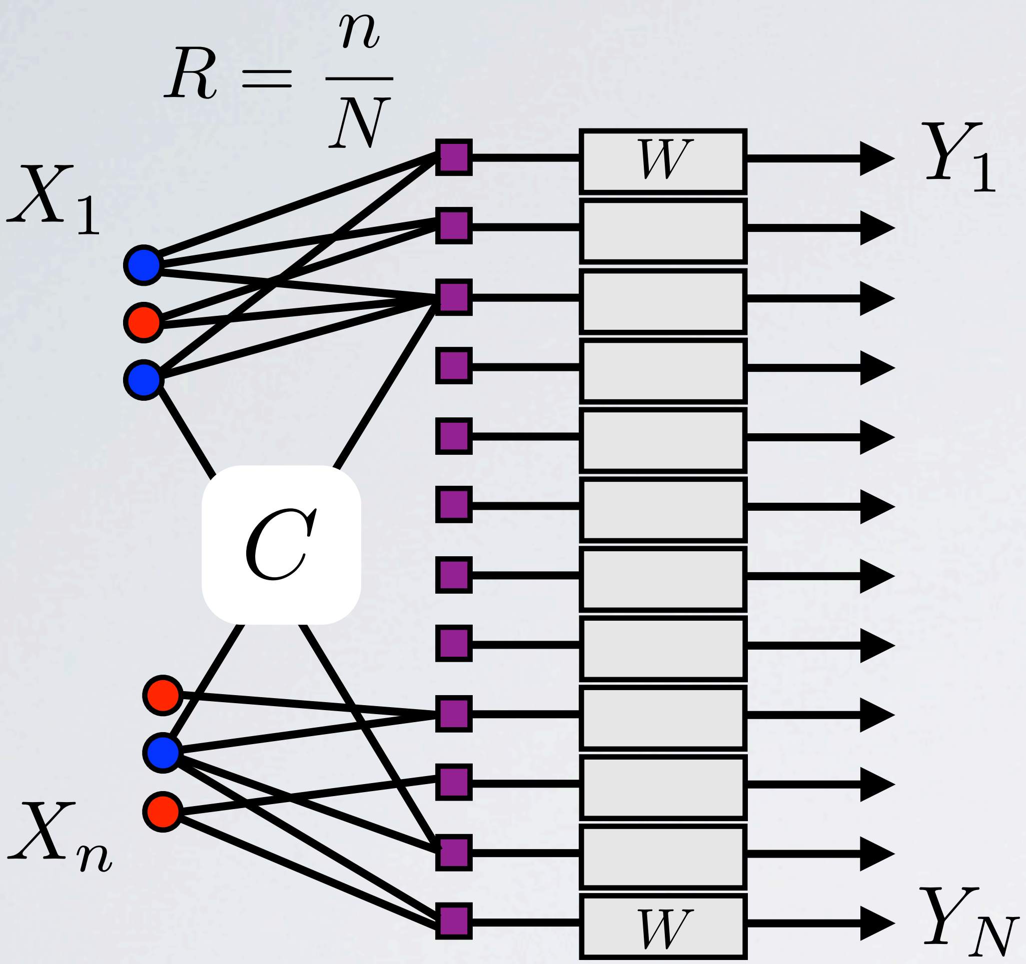


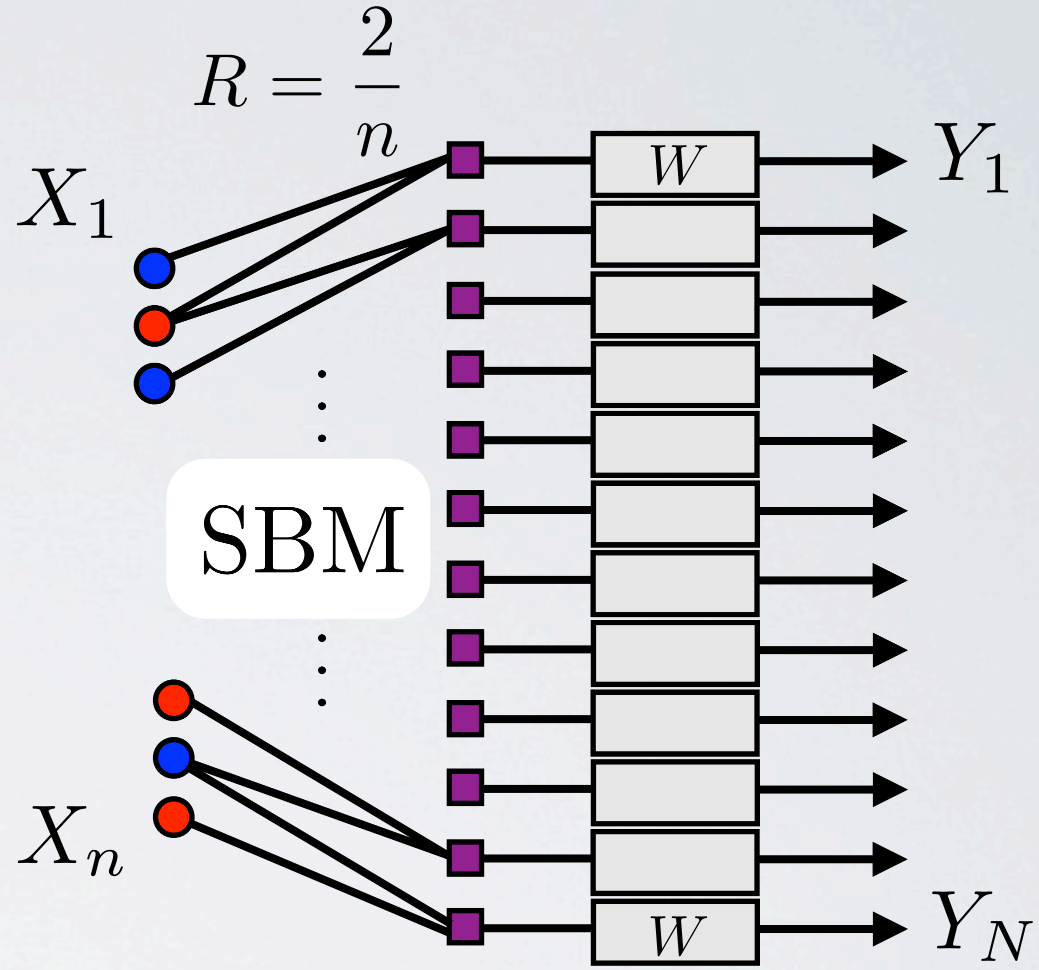
$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

reliable comm. iff R < 1-H($\boldsymbol{\varepsilon}$)

$$W = \begin{pmatrix} 1 - a\log(n)/n & a\log(n)/n \\ 1 - b\log(n)/n & b\log(n)/n \end{pmatrix}$$

Recovery in the general SBM -> an information theoretic motivation



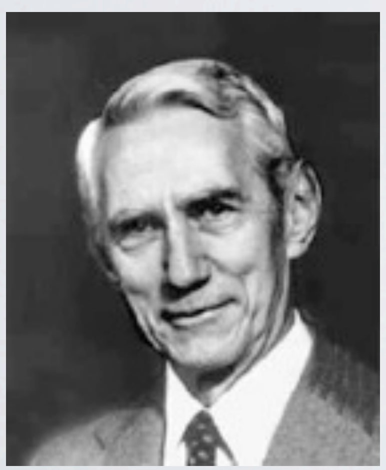


$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

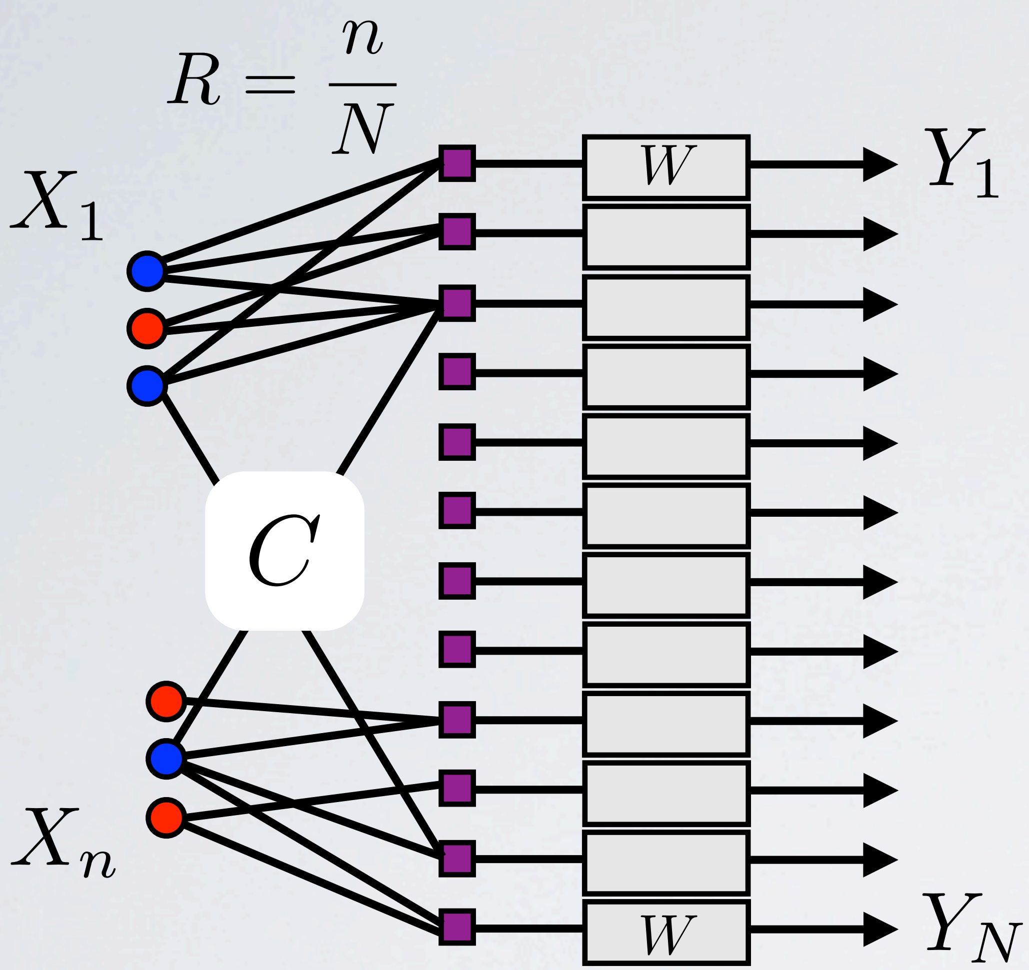
reliable comm. iff R < 1-H(**ε**)

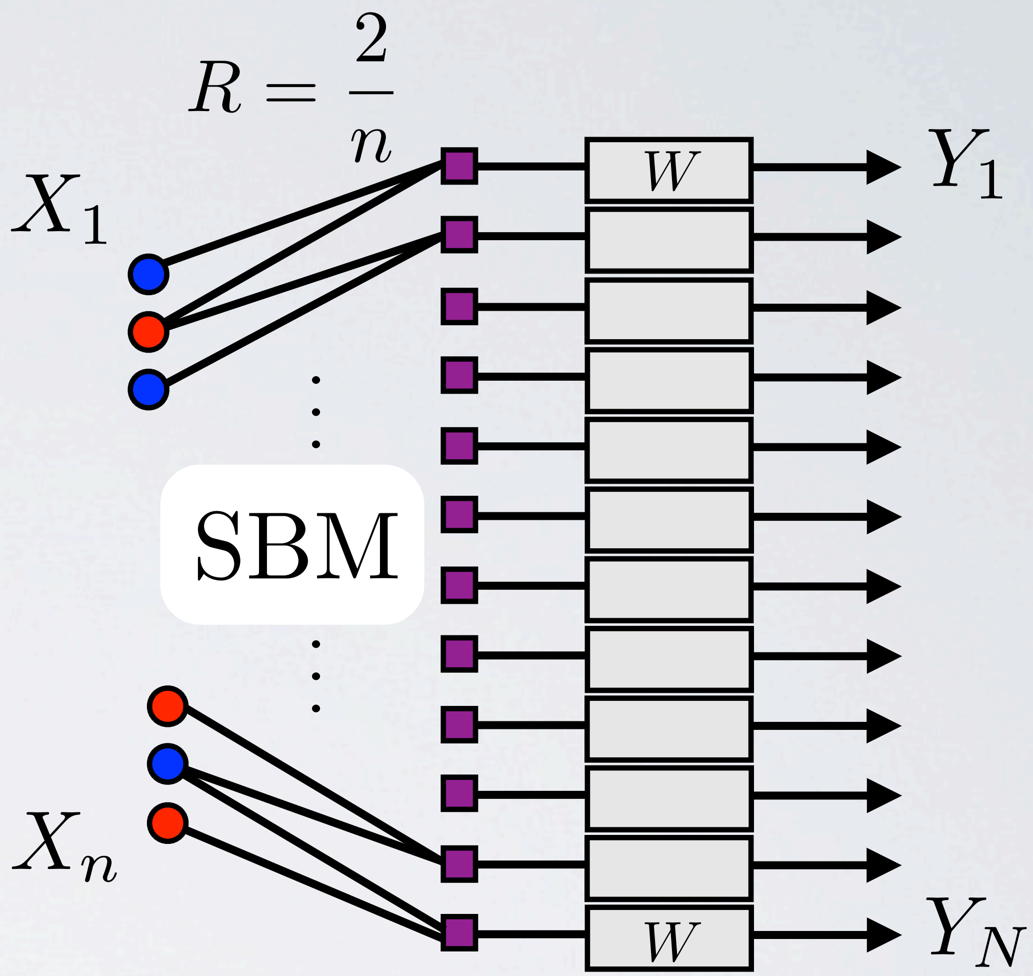
$$W = \begin{pmatrix} 1-a\log(n)/n & a\log(n)/n \\ 1-b\log(n)/n & b\log(n)/n \end{pmatrix}$$

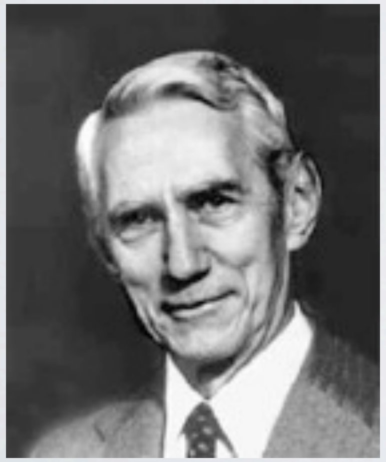
reliable comm. iff 1 < (a+b)/2-$\sqrt{ab}$



Recovery in the general SBM -> an information theoretic motivation







$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

reliable comm. iff R < 1-H(ε)

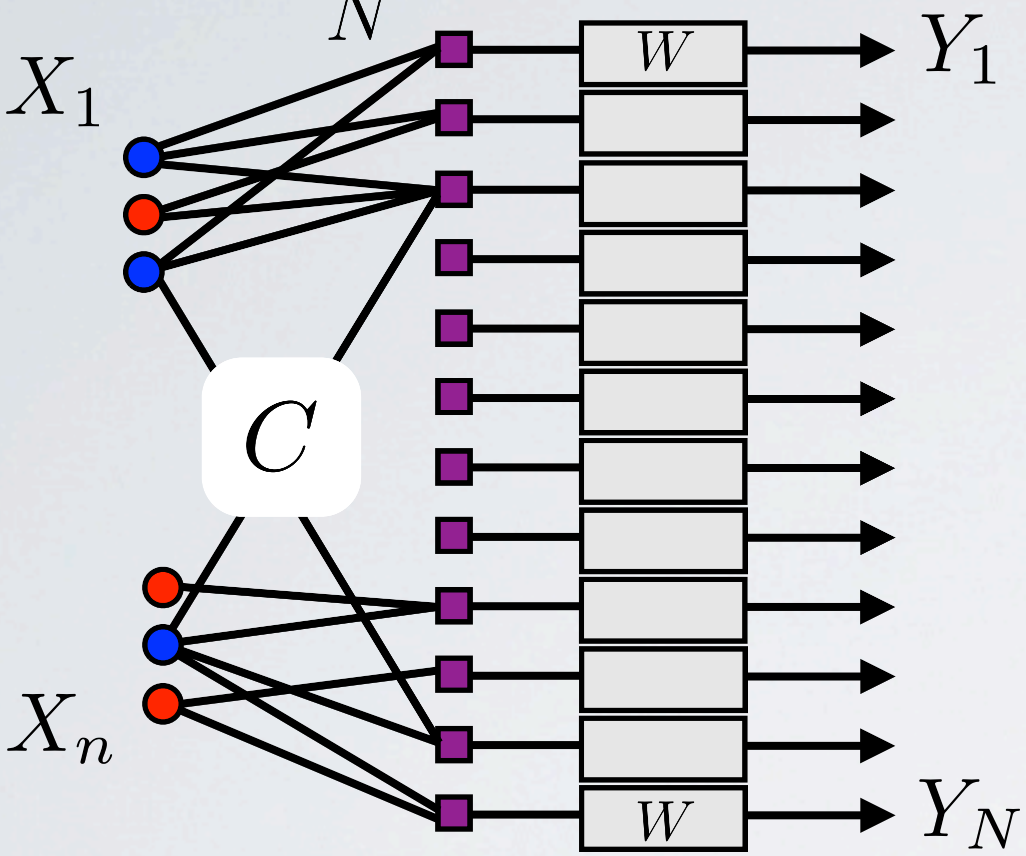
reliable comm. iff R < $\max_p$ I(p,W)

$$W = \begin{pmatrix} 1-a\log(n)/n & a\log(n)/n \\ 1-b\log(n)/n & b\log(n)/n \end{pmatrix}$$

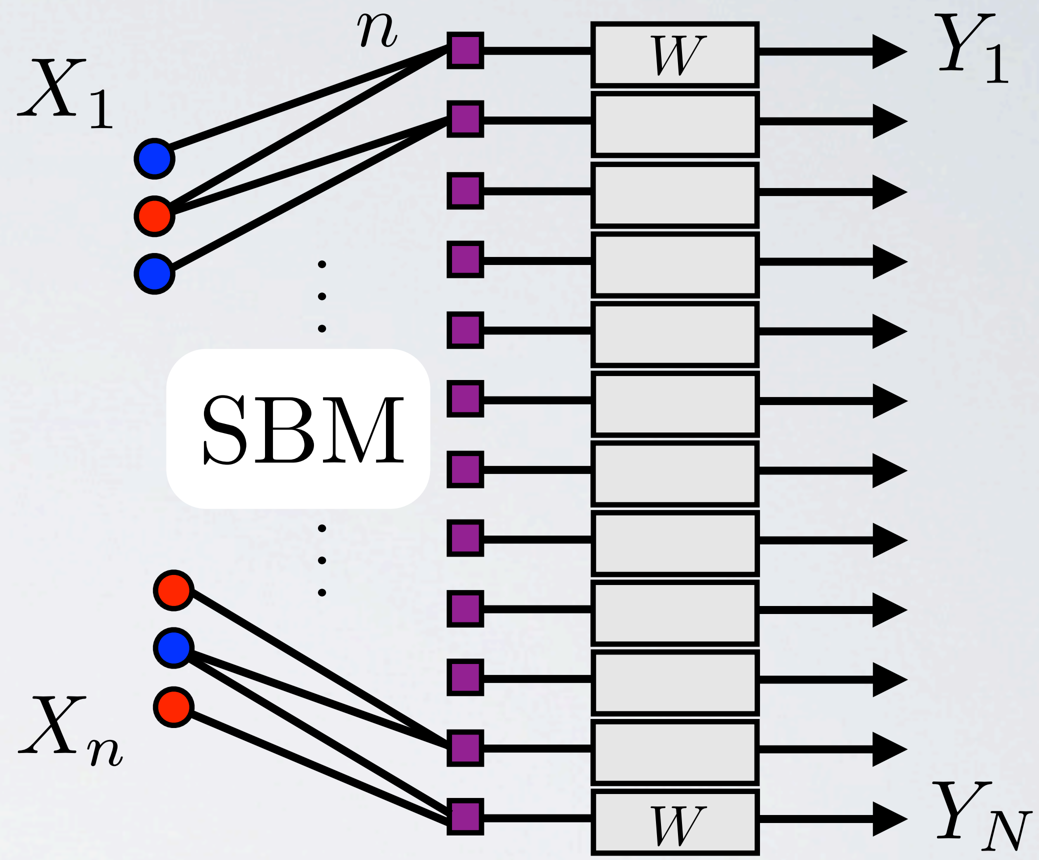
reliable comm. iff 1 < (a+b)/2-$\sqrt{ab}$

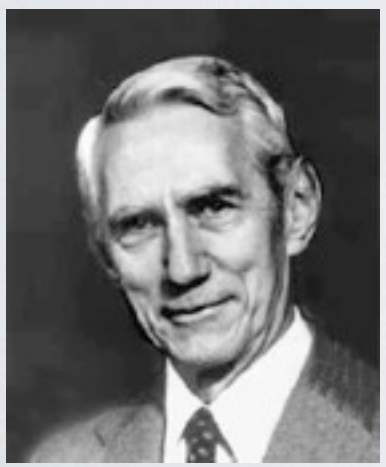
Recovery in the general SBM -> an information theoretic motivation

$R = \frac{n}{N}$



$R = \frac{2}{n}$





$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

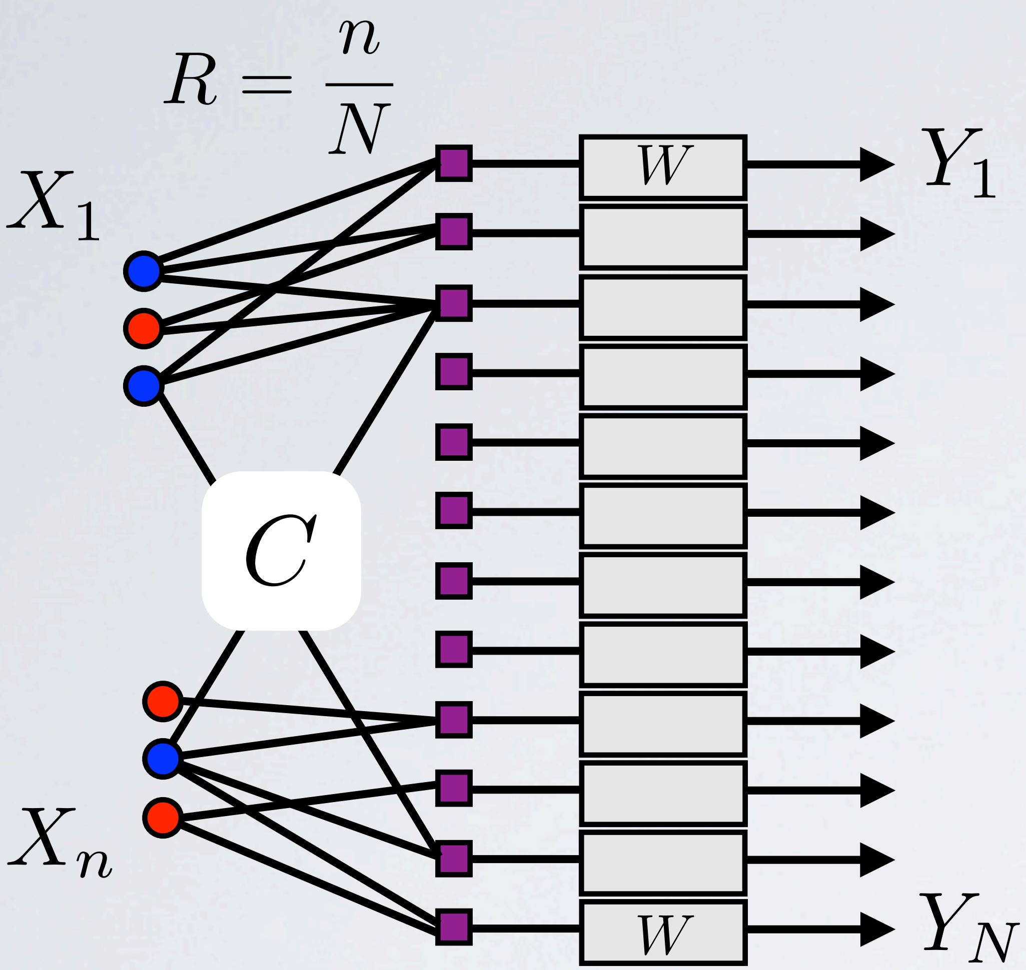
reliable comm. iff R < 1-H(ε)

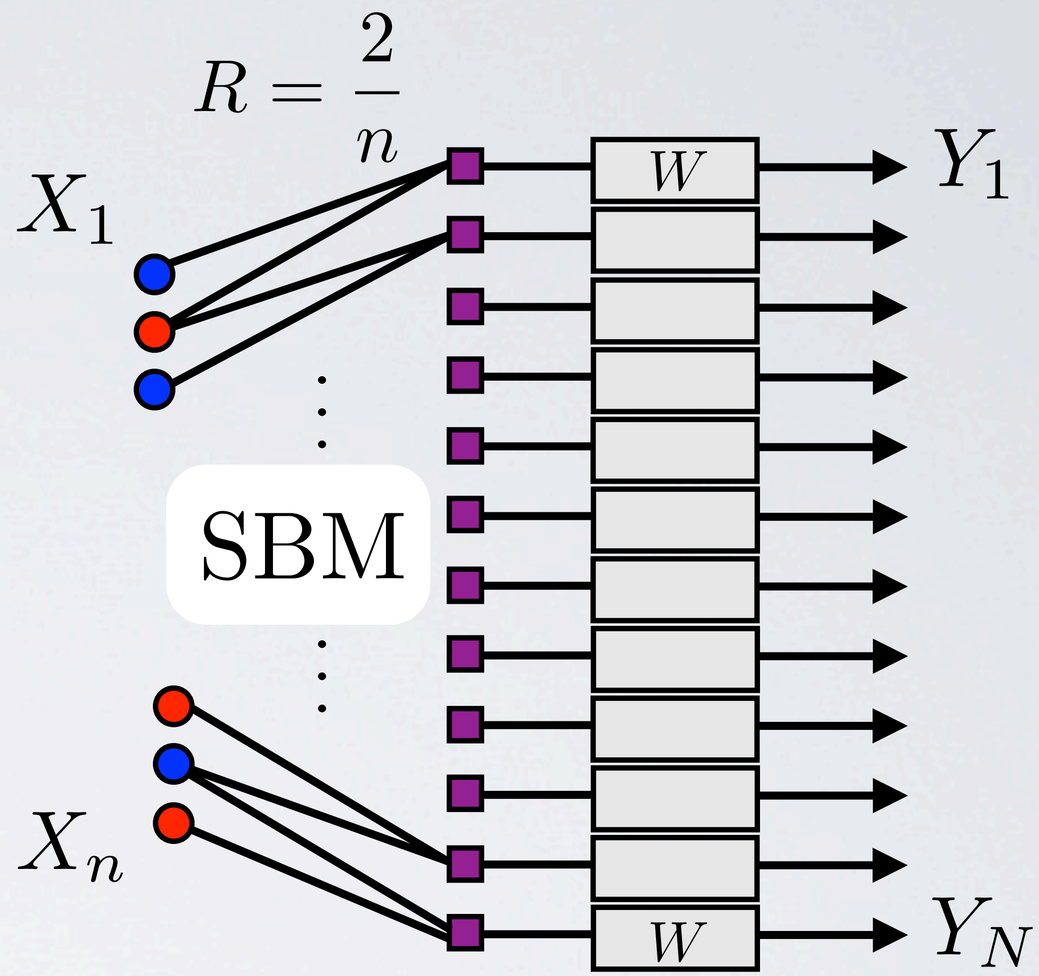
$$W = \begin{pmatrix} 1-a\log(n)/n & a\log(n)/n \\ 1-b\log(n)/n & b\log(n)/n \end{pmatrix}$$

reliable comm. iff $1 < (a+b)/2-\sqrt{ab}$

reliable comm. iff $R < \max_p \underbrace{I(p,W)}_{\text{KL-divergence}}$

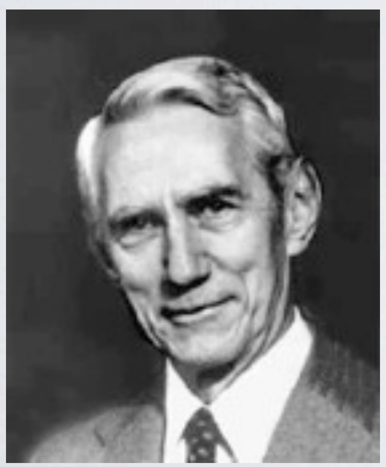
Recovery in the general SBM -> an information theoretic motivation





$$W = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}$$

$$W = \begin{pmatrix} 1 - a\log(n)/n & a\log(n)/n \\ 1 - b\log(n)/n & b\log(n)/n \end{pmatrix}$$



reliable comm. iff R < 1-H($\boldsymbol{\varepsilon}$)

reliable comm. iff 1 < (a+b)/2-$\sqrt{ab}$

reliable comm. iff $R < \max_p \underbrace{I(p,W)}_{\text{KL-divergence}}$

reliable comm. iff 1 < J(p,W) ???

Results

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$J(p, Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$J(p,Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu,\nu) := \max_{t\in[0,1]} \sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)$$

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q \log(n)/n)$ if and only if

$$J(p,Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu,\nu) := \max_{t\in[0,1]} \underbrace{\sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)}_{D_t(\mu,\nu)}$$

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$J(p,Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu,\nu) := \max_{t\in[0,1]} \underbrace{\sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)}_{D_t(\mu,\nu)}$$

- D_t is an f-divergence

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q \log(n)/n)$ if and only if

$$J(p, Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu, \nu) := \max_{t \in [0,1]} \underbrace{\sum_{i \in [k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)}_{D_t(\mu,\nu)}$$

- D_t is an f-divergence
- $D_{1/2}(\mu, \nu) = \frac{1}{2}\|\sqrt{\mu} - \sqrt{\nu}\|_2^2$ is the Hellinger divergence (distance)

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$J(p,Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu,\nu) := \max_{t\in[0,1]} \underbrace{\sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)}_{D_t(\mu,\nu)}$$

- D_t is an f-divergence
- $D_{1/2}(\mu,\nu) = \frac{1}{2}\|\sqrt{\mu} - \sqrt{\nu}\|_2^2$ is the Hellinger divergence (distance)

$\frac{1}{2}(\sqrt{a} - \sqrt{b})^2 \geq 1 \longleftarrow$

Abbe-Bandeira-Hall '14

Mossel-Neeman-Sly '14

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$J(p,Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu,\nu) := \max_{t\in[0,1]} \underbrace{\sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)}_{D_t(\mu,\nu)}$$

- D_t is an f-divergence
- $D_{1/2}(\mu,\nu) = \frac{1}{2}\|\sqrt{\mu} - \sqrt{\nu}\|_2^2$ is the Hellinger divergence (distance)
- $-\log \max_t \sum_i \mu_i^t \nu_i^t$ is the Chernoff divergence

$$\frac{1}{2}(\sqrt{a} - \sqrt{b})^2 \geq 1 \longleftarrow$$

Abbe-Bandeira-Hall '14

Mossel-Neeman-Sly '14

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q \log(n)/n)$ if and only if

$$J(p, Q) := \min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu, \nu) := \max_{t \in [0,1]} \underbrace{\sum_{i \in [k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)}_{D_t(\mu,\nu)}$$

- D_t is an f-divergence
- $D_{1/2}(\mu, \nu) = \frac{1}{2}\|\sqrt{\mu} - \sqrt{\nu}\|_2^2$ is the Hellinger divergence (distance)
- $-\log \max_t \sum_i \mu_i^t \nu_i^t$ is the Chernoff divergence

$\frac{1}{2}(\sqrt{a} - \sqrt{b})^2 \geq 1 \longleftarrow$

Abbe-Bandeira-Hall ’14

Mossel-Neeman-Sly ’14

We call D_+ the CH-divergence.

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$\min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu, \nu) := \max_{t\in[0,1]} \sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)$$

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$\min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu, \nu) := \max_{t\in[0,1]} \sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)$$

Theorem 2. The `degree-profiling` algorithm achieves the threshold and runs in quasi-linear time.

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$\min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

$$D_+(\mu, \nu) := \max_{t\in[0,1]} \sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)$$

Theorem 2. The `degree-profiling` algorithm achieves the threshold and runs in quasi-linear time. 

Q_{ij} non-zero

Theorem 1. Recovery is solvable in $\text{SBM}(n, p, Q\log(n)/n)$ if and only if

$$\min_{i<j} D_+((PQ)_i, (PQ)_j) \geq 1$$

where

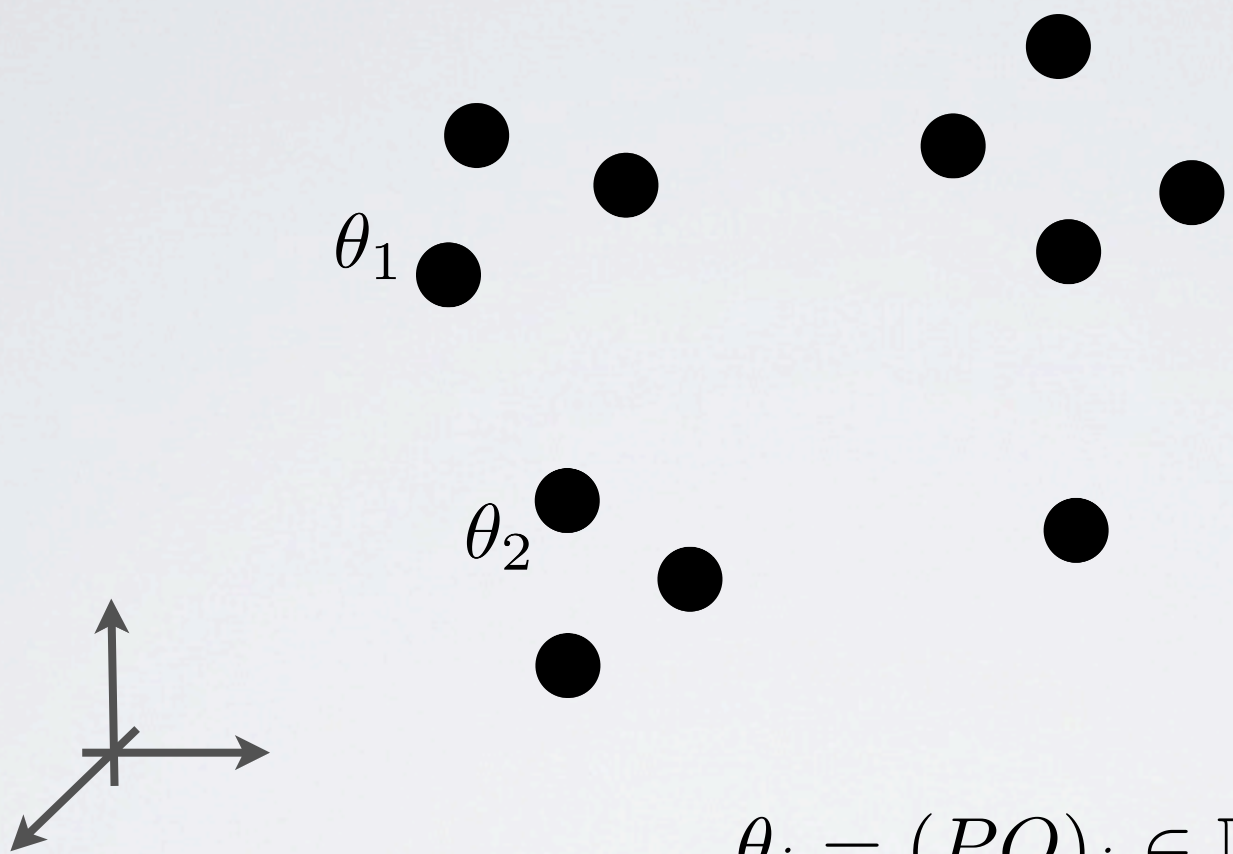
$$D_+(\mu, \nu) := \max_{t\in[0,1]} \sum_{i\in[k]} \left(t\mu_i + (1-t)\nu_i - \mu_i^t \nu_i^{1-t}\right)$$

Theorem 2. The `degree-profiling` algorithm achieves the threshold and runs in quasi-linear time.

Exact recovery in the general SBM is solvable efficiently whenever it is solvable information theoretically

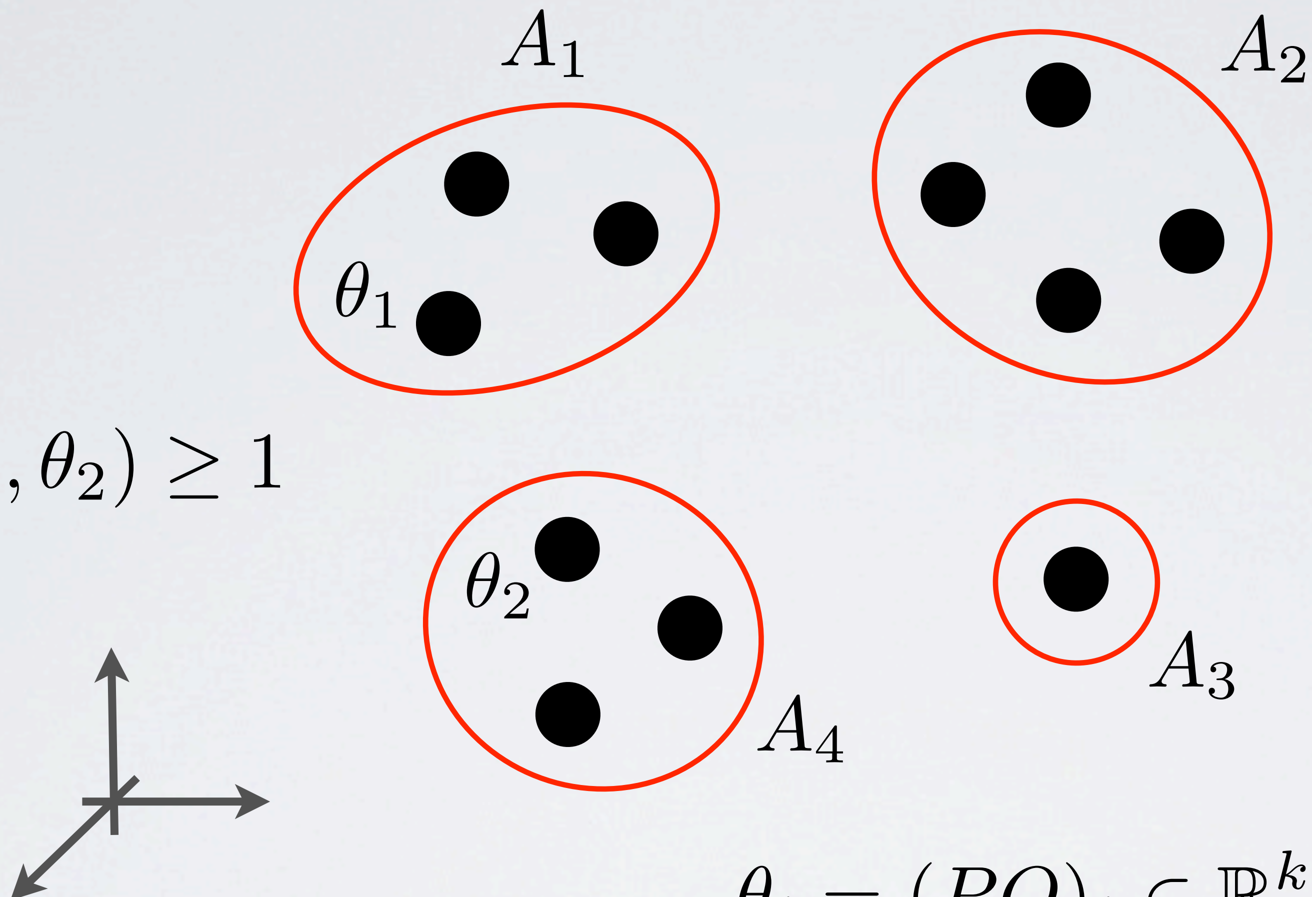
What about recovering a subgroup of the communities?

What about recovering a subgroup of the communities?



$$\theta_i = (PQ)_i \in \mathbb{R}_+^k \ \ \text{(local community profile)}$$

What about recovering a subgroup of the communities?

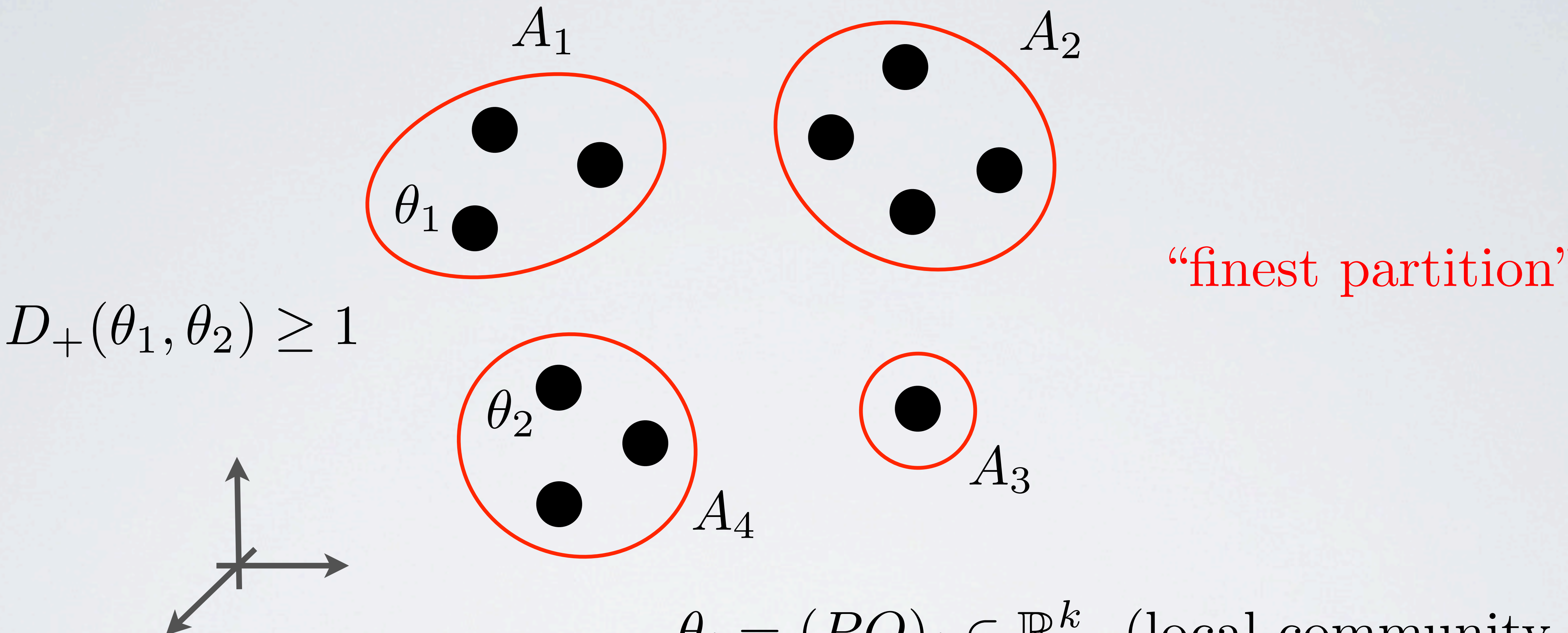


"finest partition"

$$D_+(\theta_1, \theta_2) \geq 1$$

$$\theta_i = (PQ)_i \in \mathbb{R}_+^k \ \text{(local community profile)}$$

What about recovering a subgroup of the communities?



$\theta_i = (PQ)_i \in \mathbb{R}_+^k$ (local community profile)

Theorem 3. Exact recovery for a partition $[k] = \sqcup_{i=1}^{s} A_i$ is solvable in $\mathrm{SBM}(n, p, Q\log(n)/n)$ if and only if

$$\min_{x<y} \underbrace{D_+(A_x, A_y)}_{\min\limits_{i \in A_x, j \in A_y} D_+((PQ)_i, (PQ)_j)} \geq 1$$

$$\min_{i \in A_x, j \in A_y} D_+((PQ)_i, (PQ)_j) \geq 1$$

Proof idea and partial recovery

Message: recover first most of the nodes and then finish differently

Message: recover first most of the nodes and then finish differently

How to recover a fraction of the nodes (partial recovery)?

Message: recover first most of the nodes and then finish differently

How to recover a fraction of the nodes (partial recovery)?

Theorem (informal). In the sparse SBM$(n, p, Q/n)$, the `Sphere-comparison` algorithm recovers a fraction of nodes which gets close to 1 when a prescribed SNR tends to infinity, in particular, when Q scales.

Message: recover first most of the nodes and then finish differently

How to recover a fraction of the nodes (partial recovery)?

Theorem (informal). In the sparse $\mathrm{SBM}(n, p, Q/n)$, the `Sphere-comparison` algorithm recovers a fraction of nodes which gets close to 1 when a prescribed SNR tends to infinity, in particular, when Q scales.

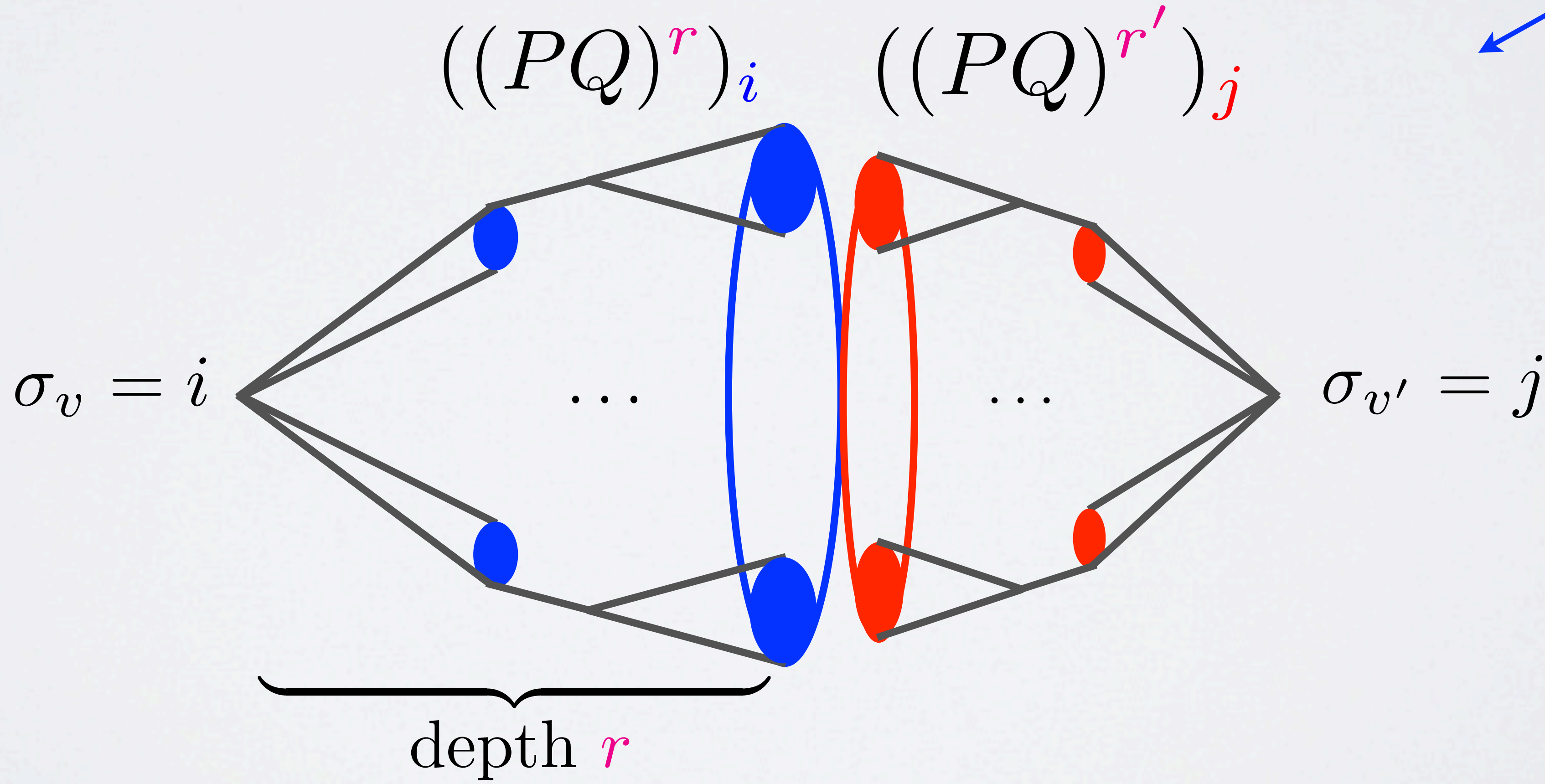
defined in terms of the spectrum of PQ, given by $\frac{(a-b)^2}{2(a+b)}$ in the 2-symmetric case

Message: recover first most of the nodes and then finish differently

How to recover a fraction of the nodes (partial recovery)?

Theorem (informal). In the sparse SBM$(n, p, Q/n)$, the `Sphere-comparison` algorithm recovers a fraction of nodes which gets close to 1 when a prescribed SNR tends to infinity, in particular, when Q scales.

defined in terms of the spectrum of PQ, given by $\frac{(a-b)^2}{2(a+b)}$ in the 2-symmetric case

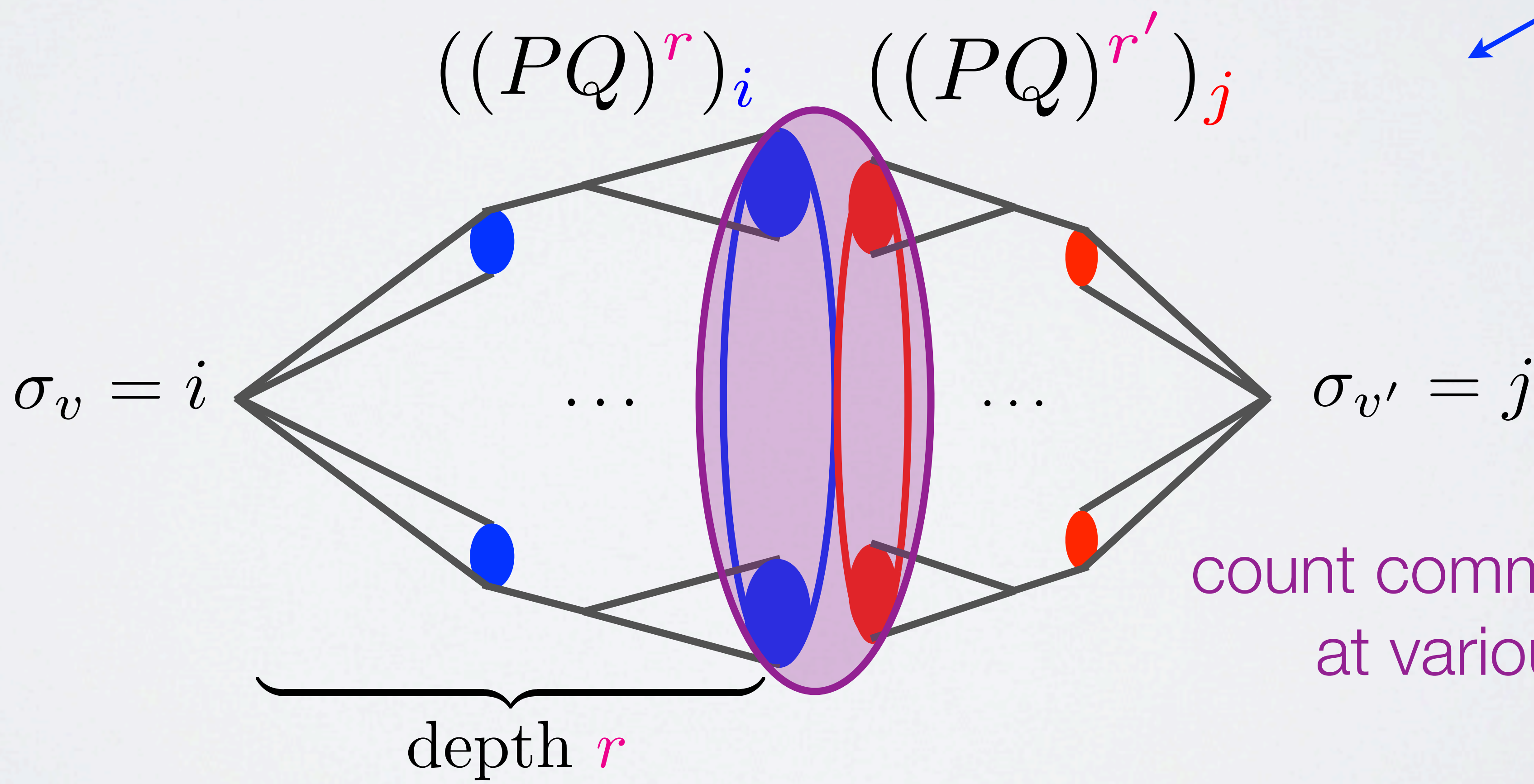


Message: recover first most of the nodes and then finish differently

How to recover a fraction of the nodes (partial recovery)?

Theorem (informal). In the sparse SBM$(n, p, Q/n)$, the `Sphere-comparison` algorithm recovers a fraction of nodes which gets close to 1 when a prescribed SNR tends to infinity, in particular, when Q scales.

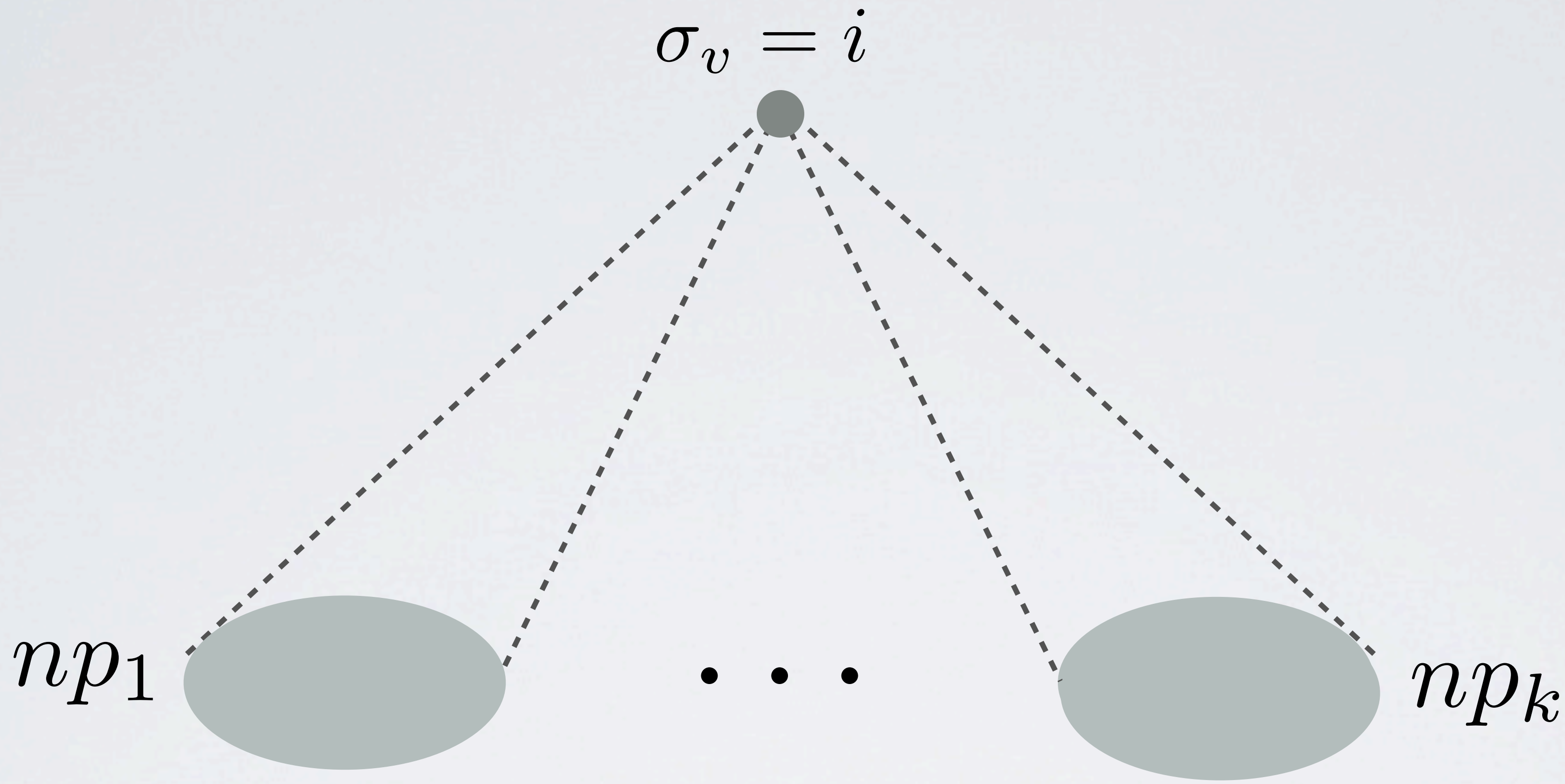
defined in terms of the spectrum of PQ, given by $\frac{(a-b)^2}{2(a+b)}$ in the 2-symmetric case



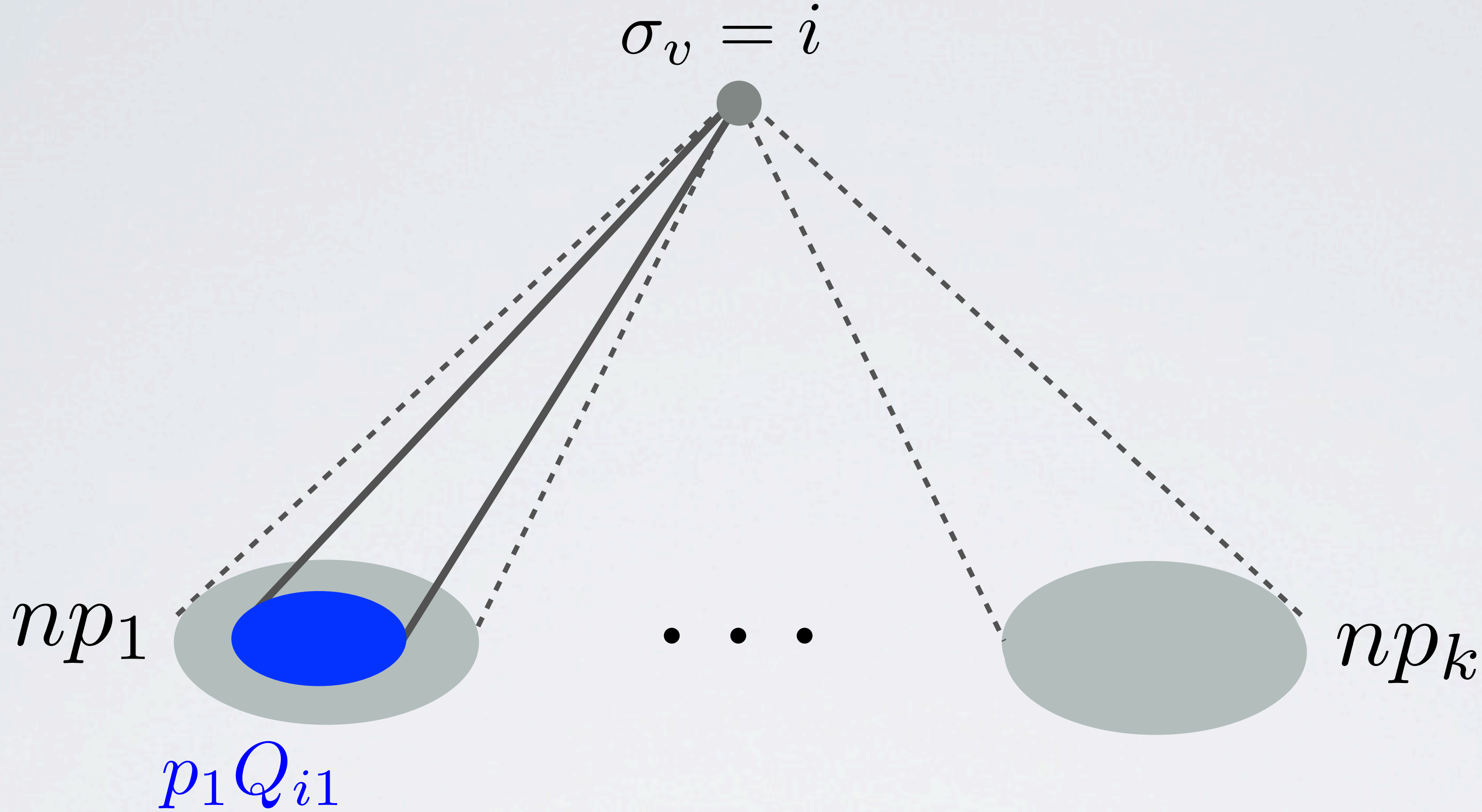
count common neighbors at various depths

What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$

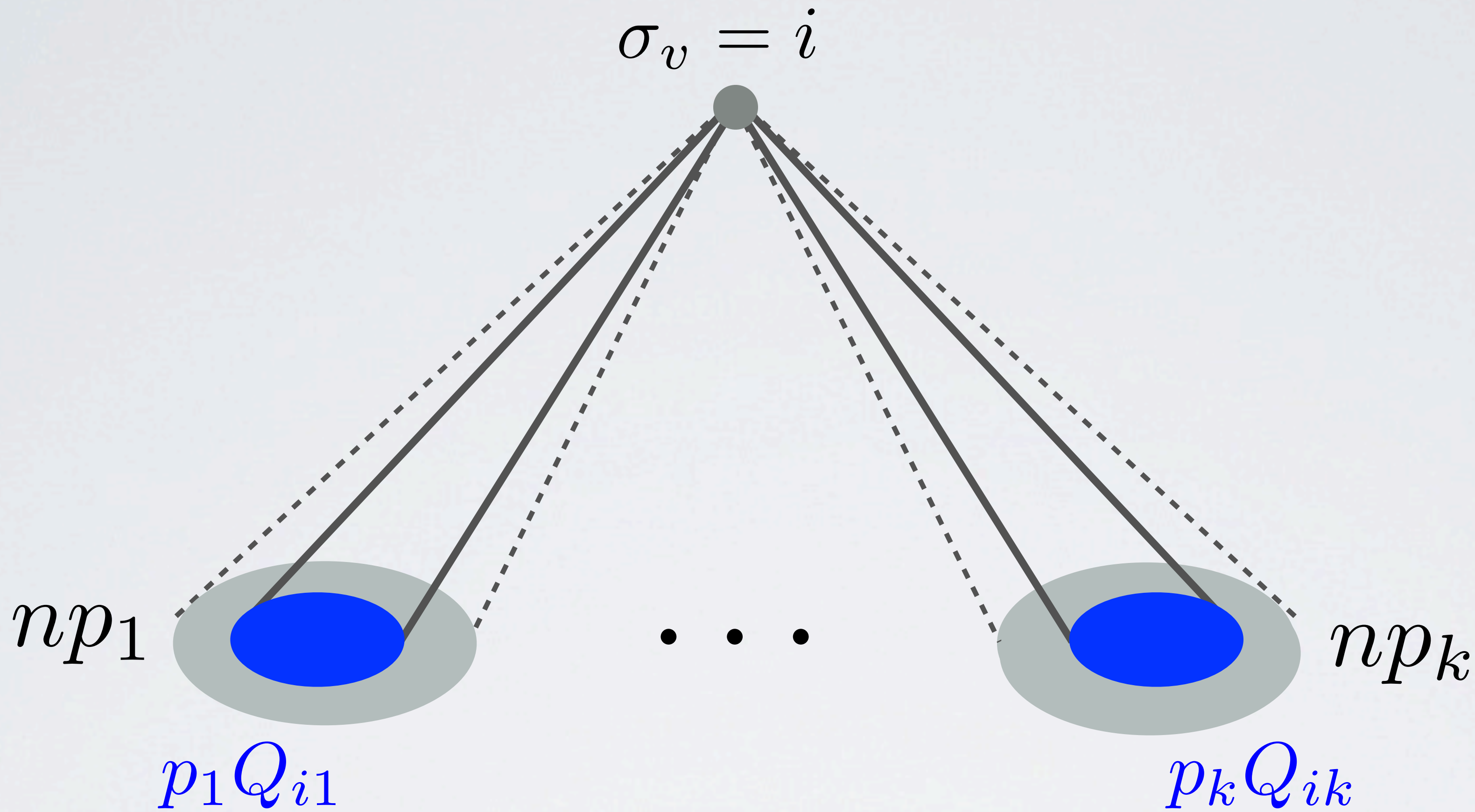
What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$



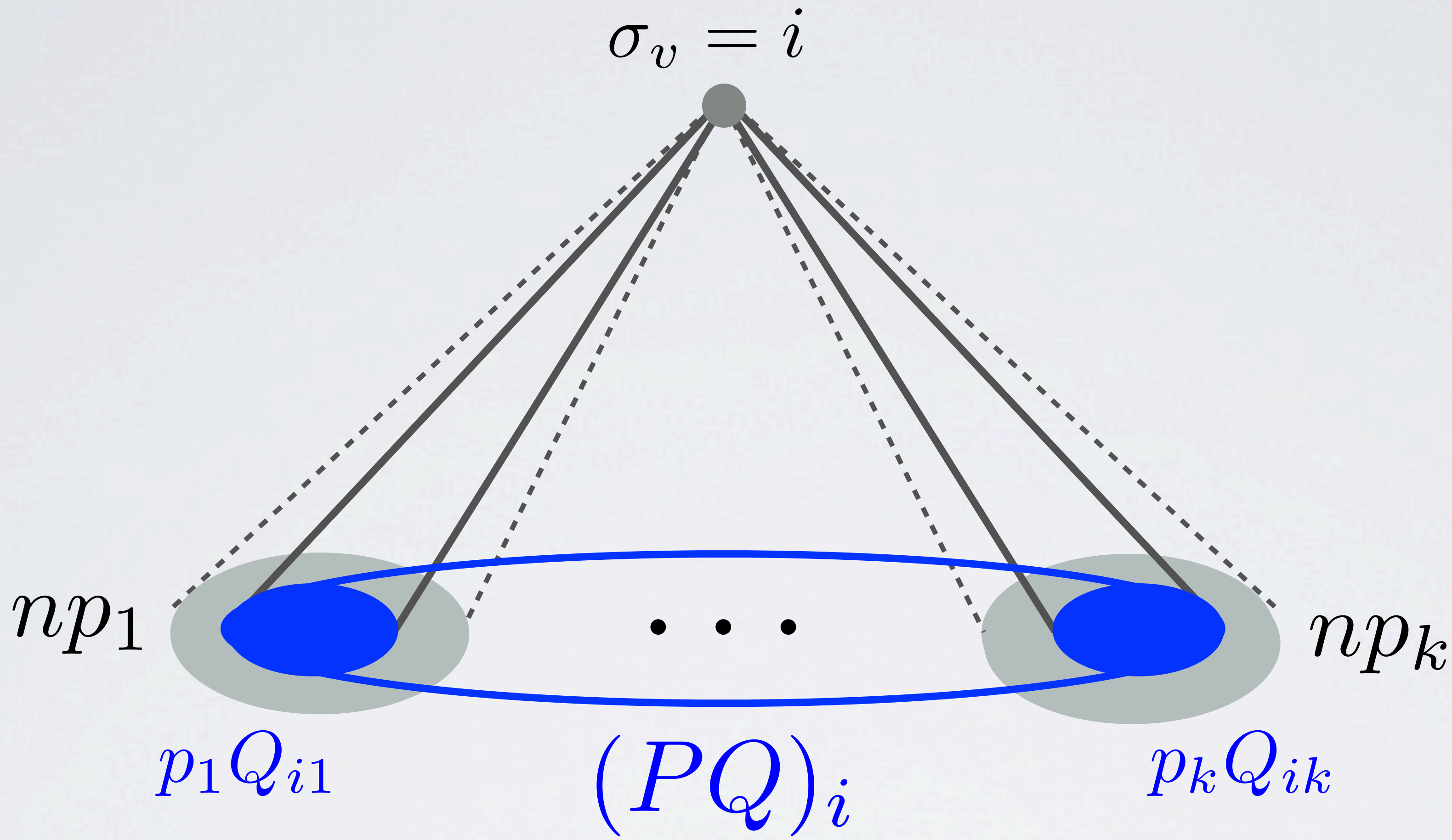
What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$



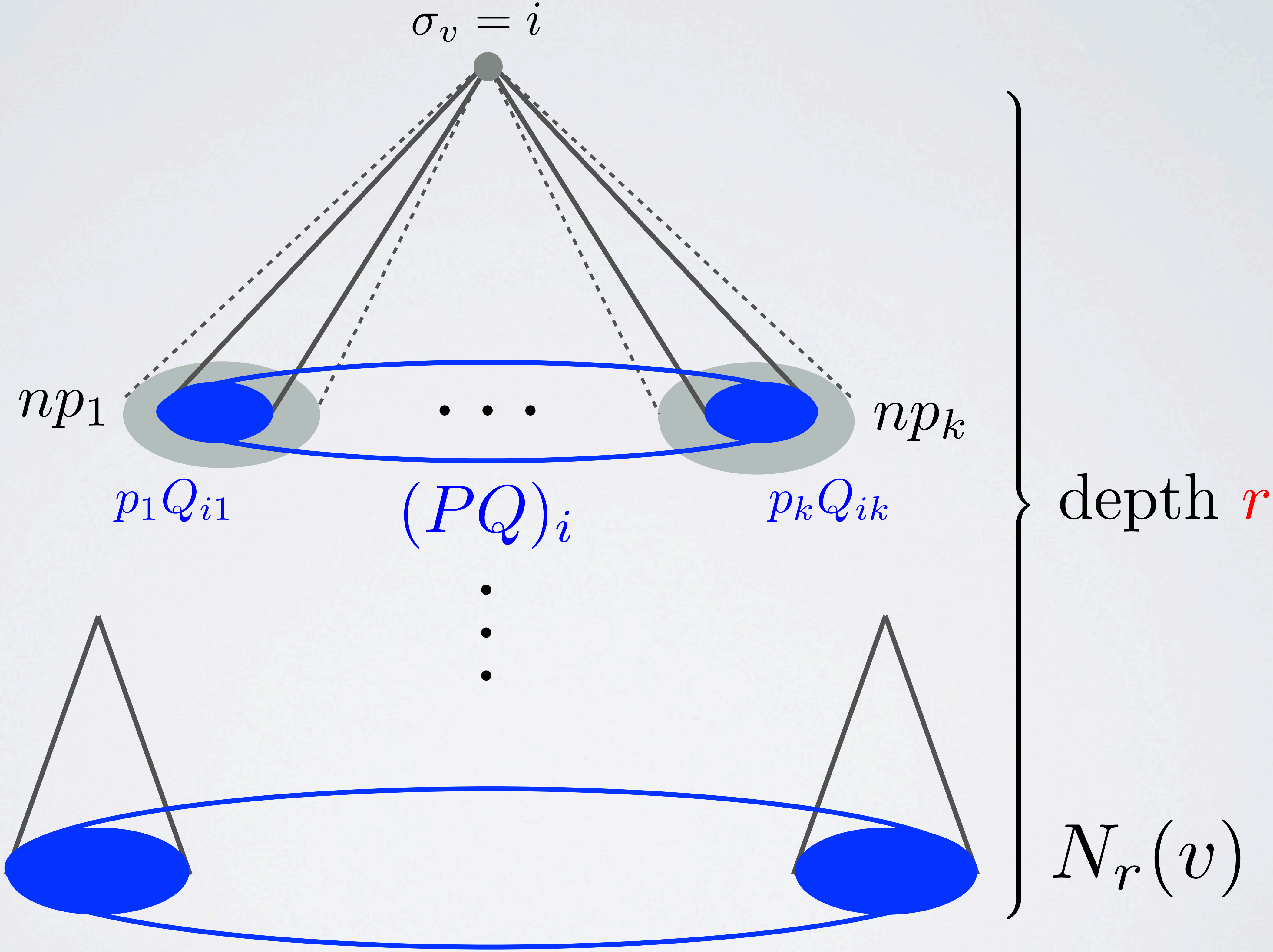
What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$



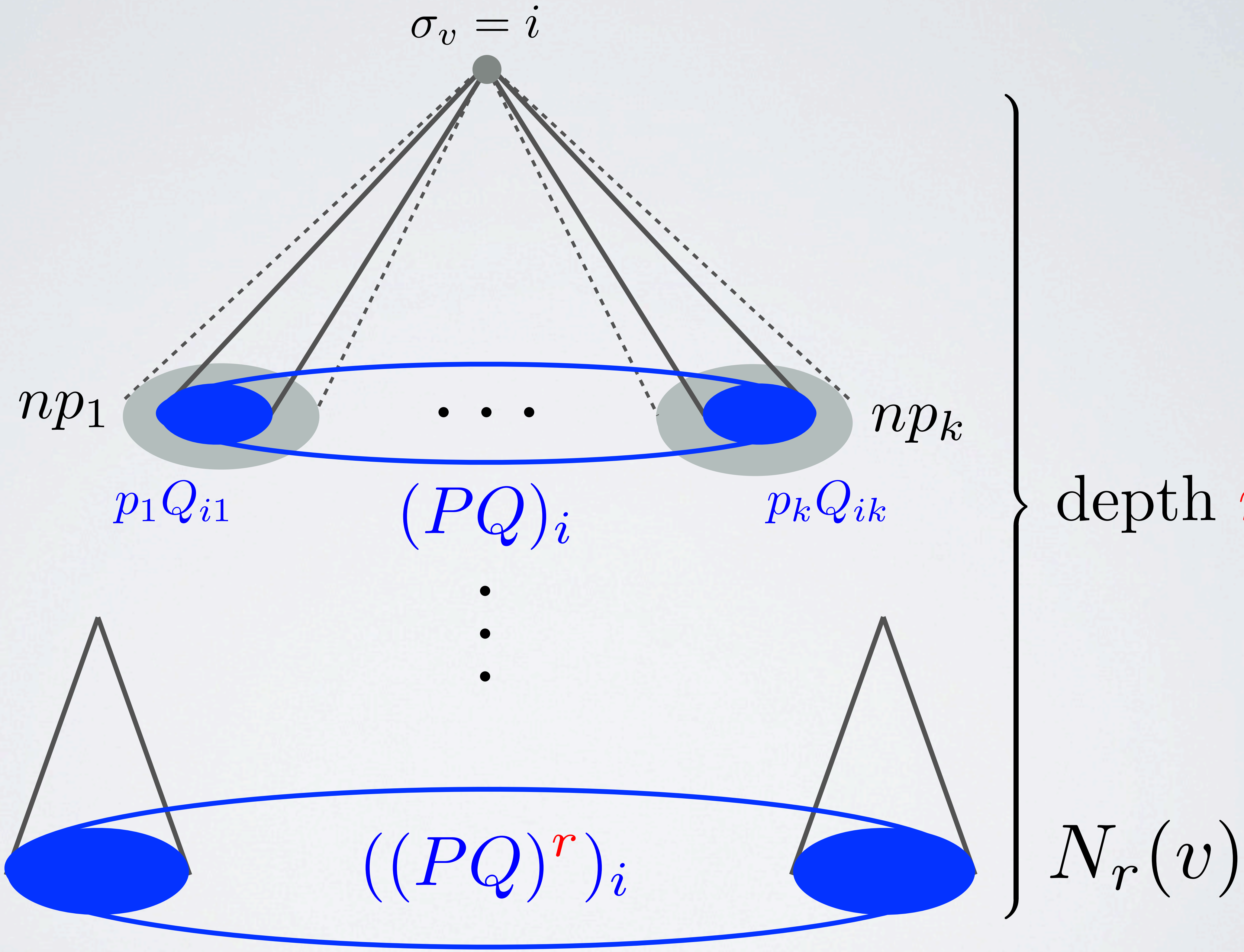
What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$



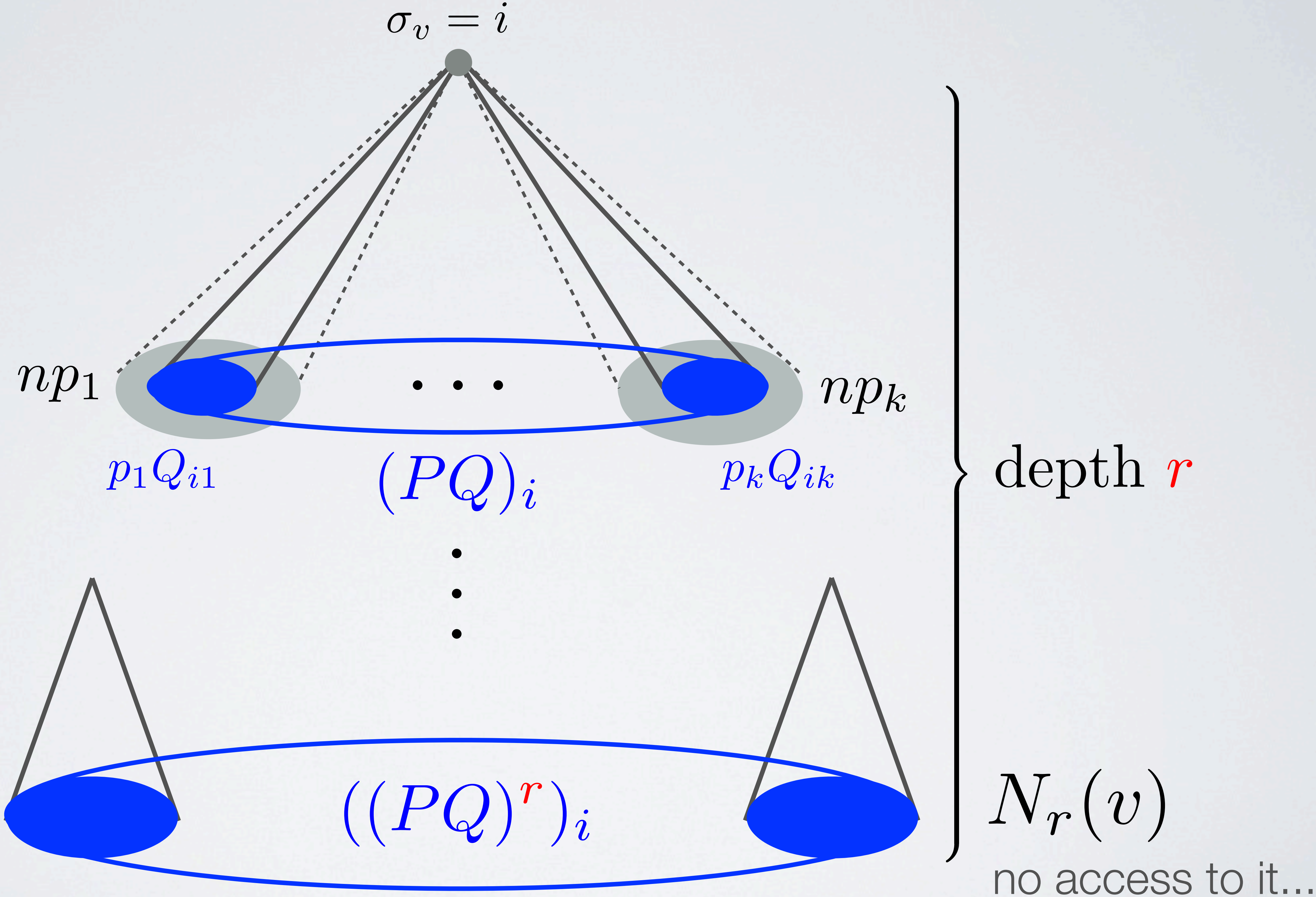
What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$



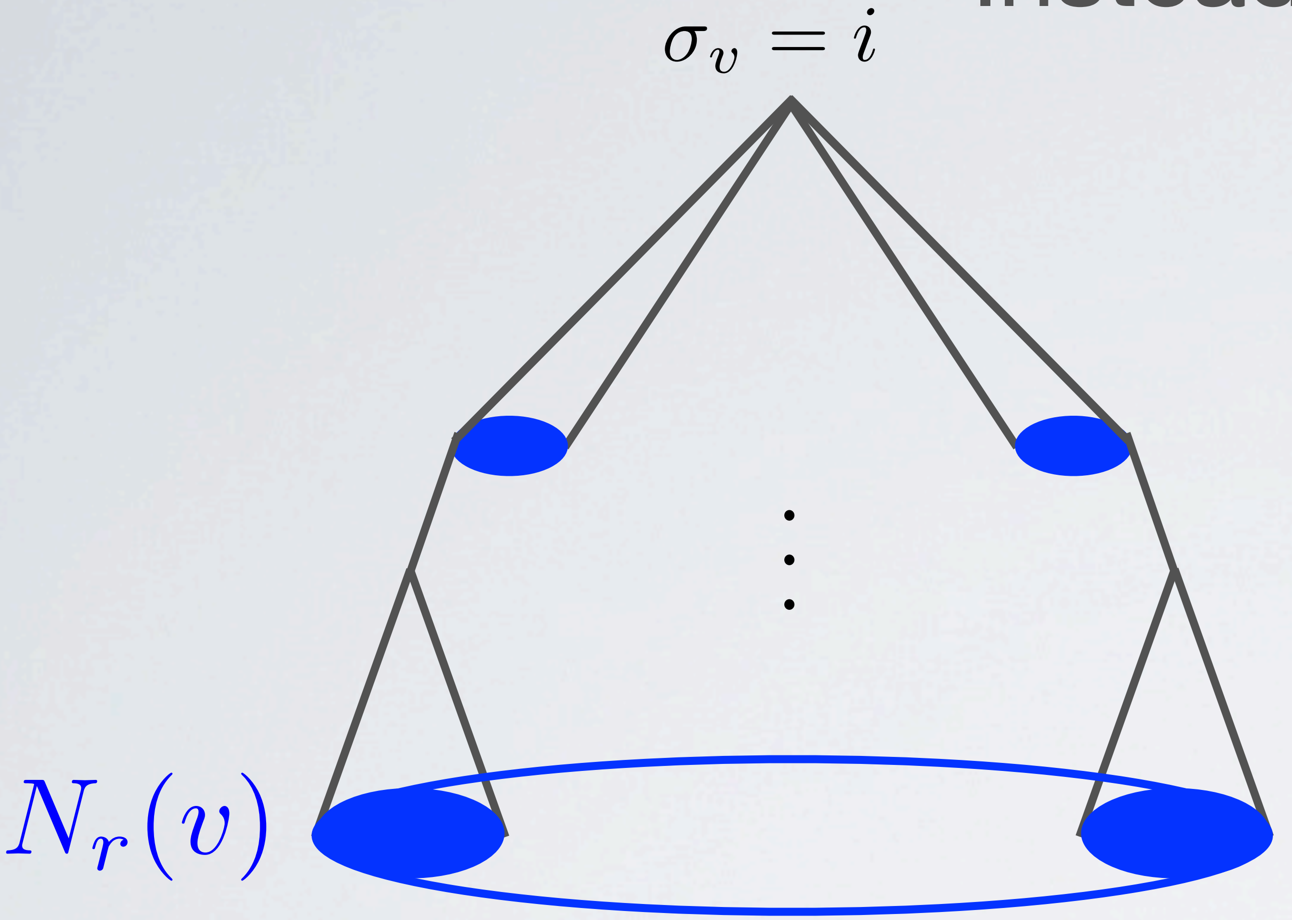
What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$



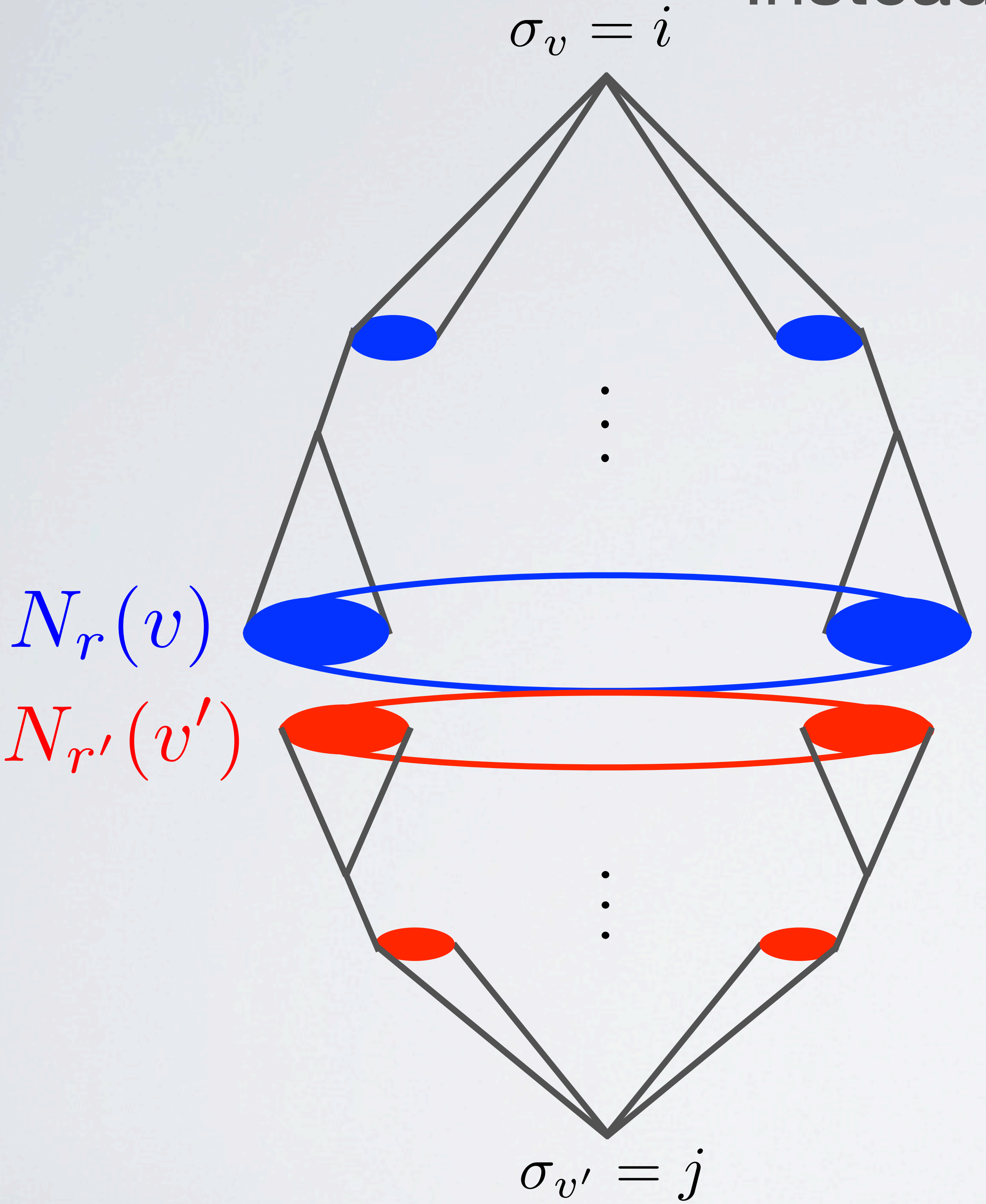
What is a vertex neighborhood like? $\mathrm{SBM}(n, p, Q/n)$



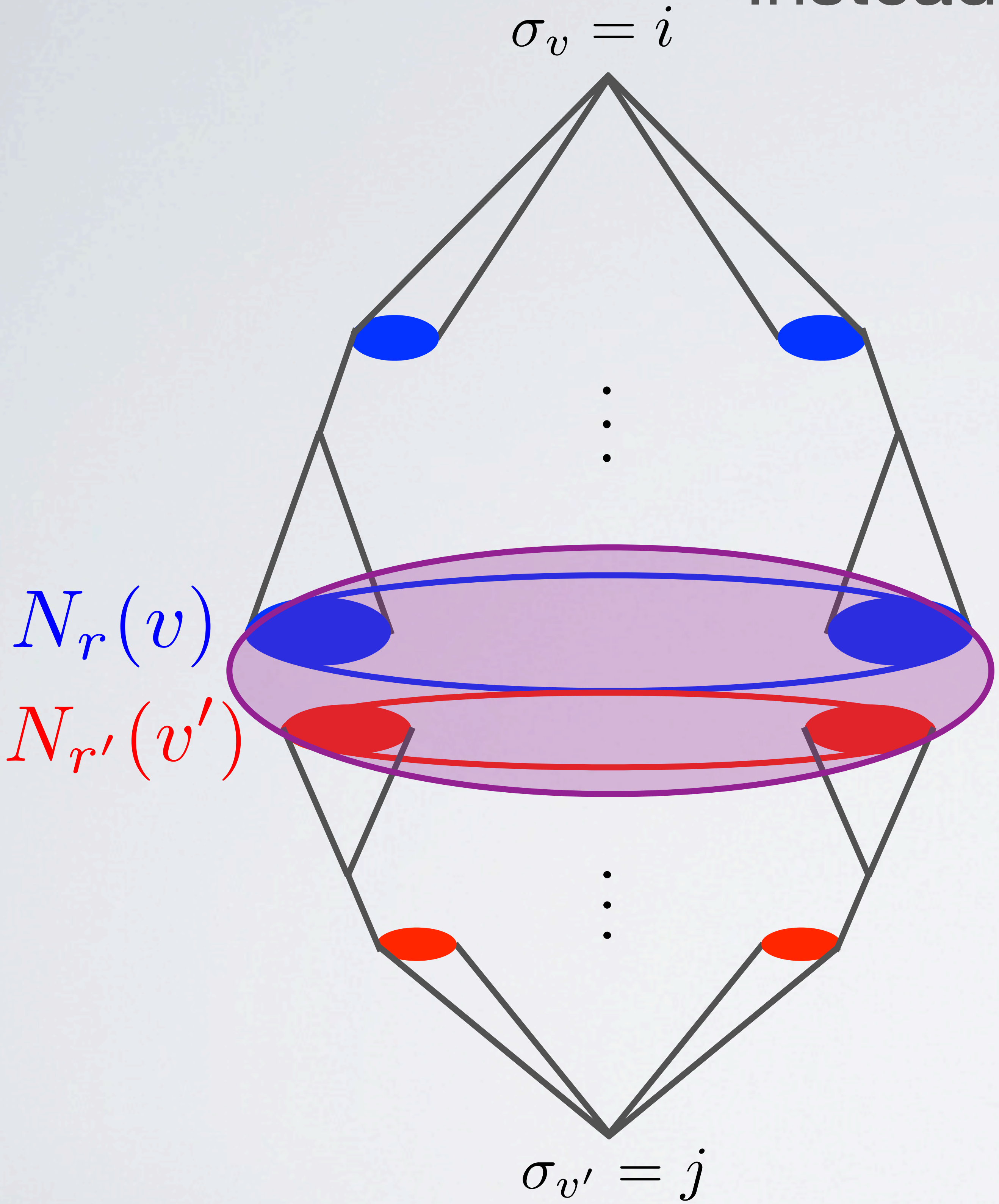
Instead: compare



Instead: compare



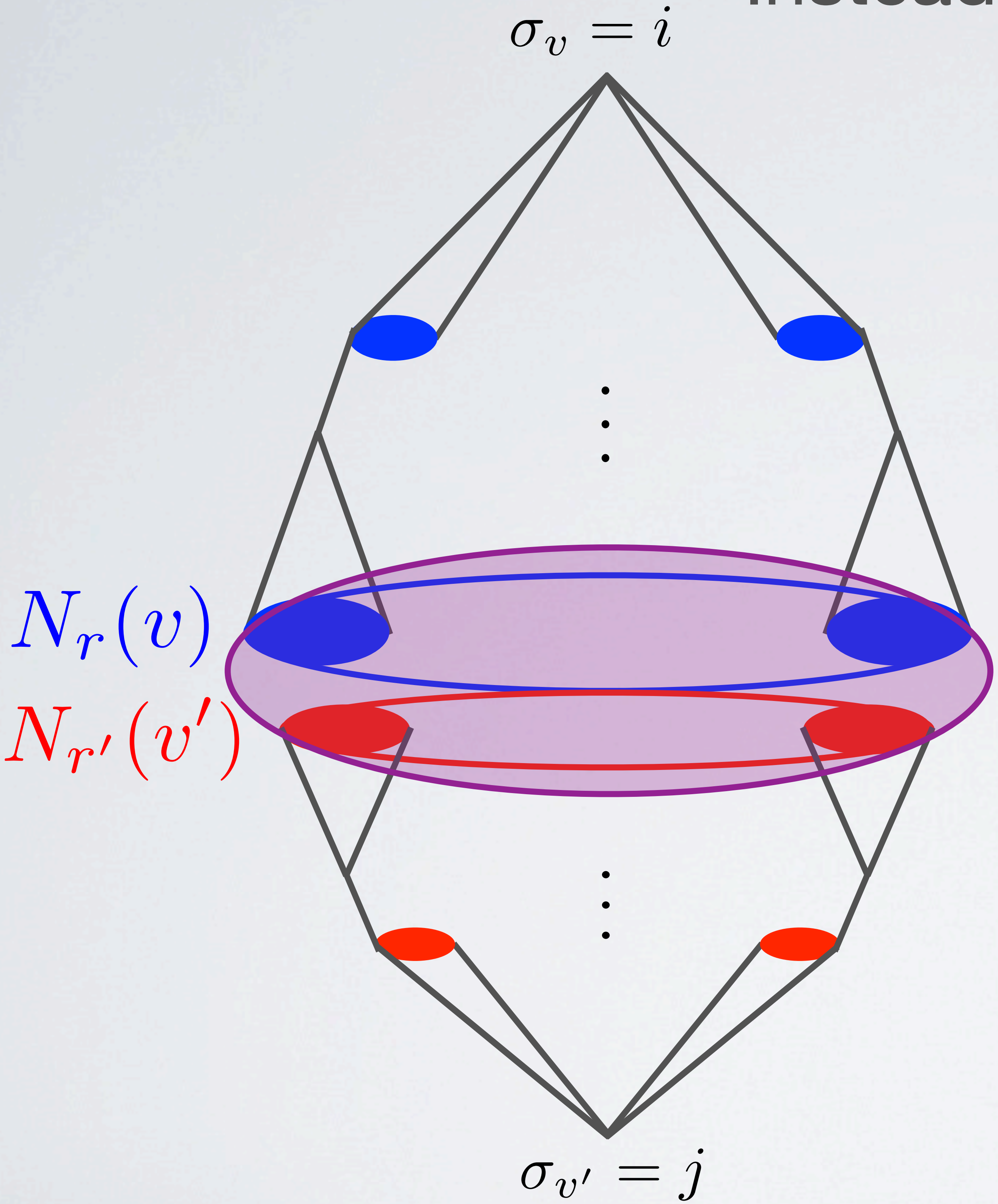
Instead: compare



Compare v and v' from:

$$|N_r(v) \cap N_{r'}(v')|$$

Instead: compare

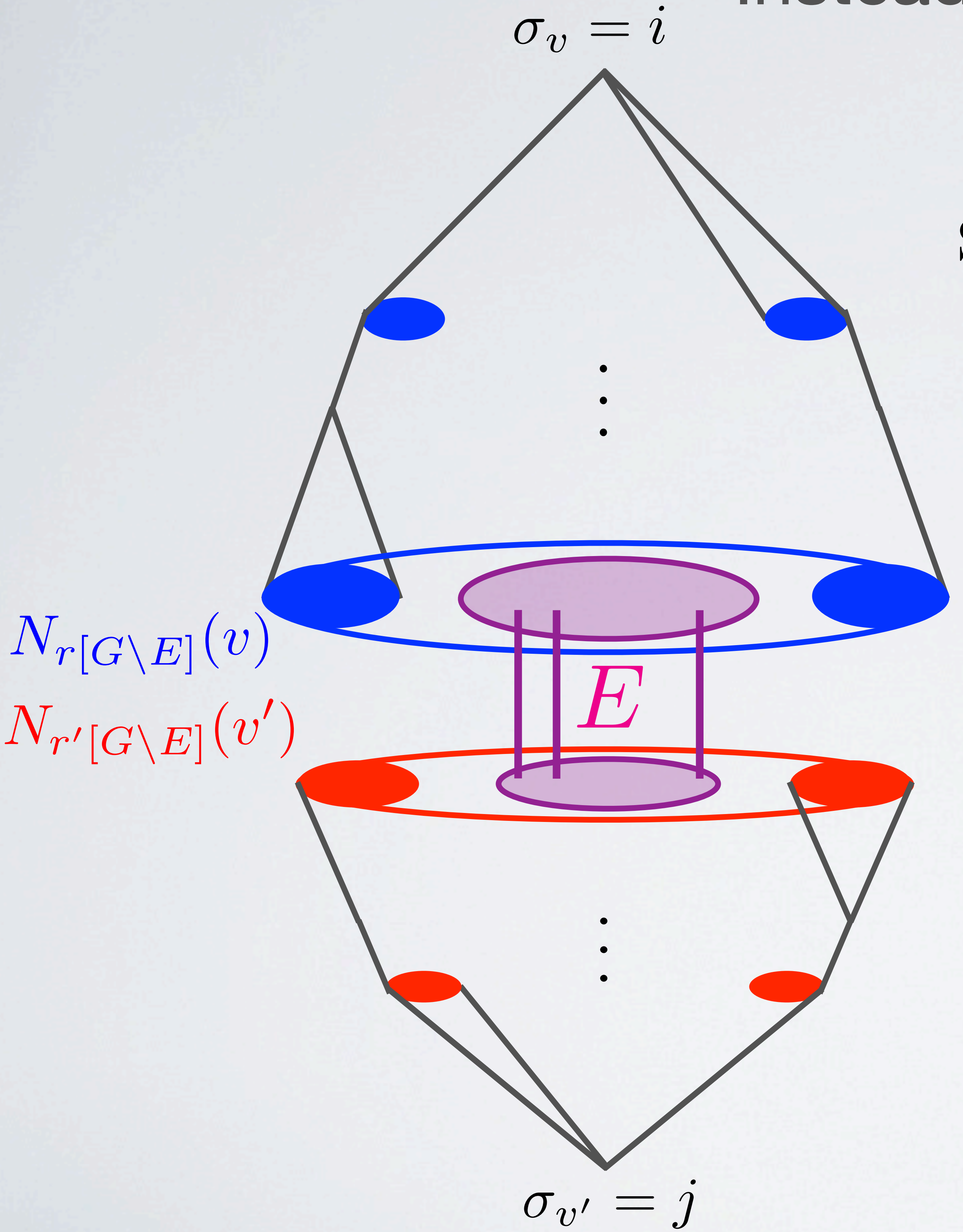


Compare v and v' from:

$$|N_r(v) \cap N_{r'}(v')|$$

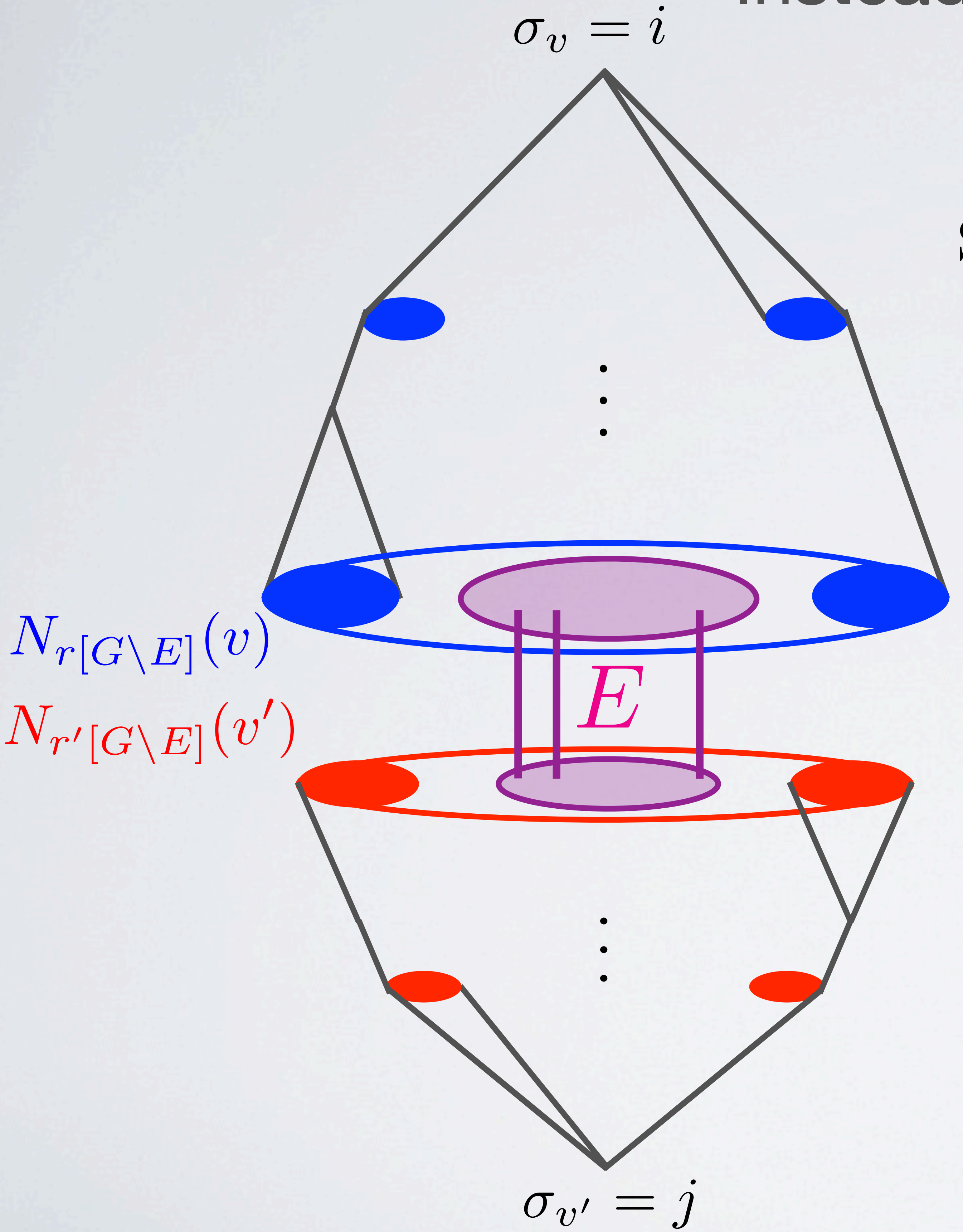
Not enough independence...

Instead: compare



Subsample G with prob. c to get E

Instead: compare



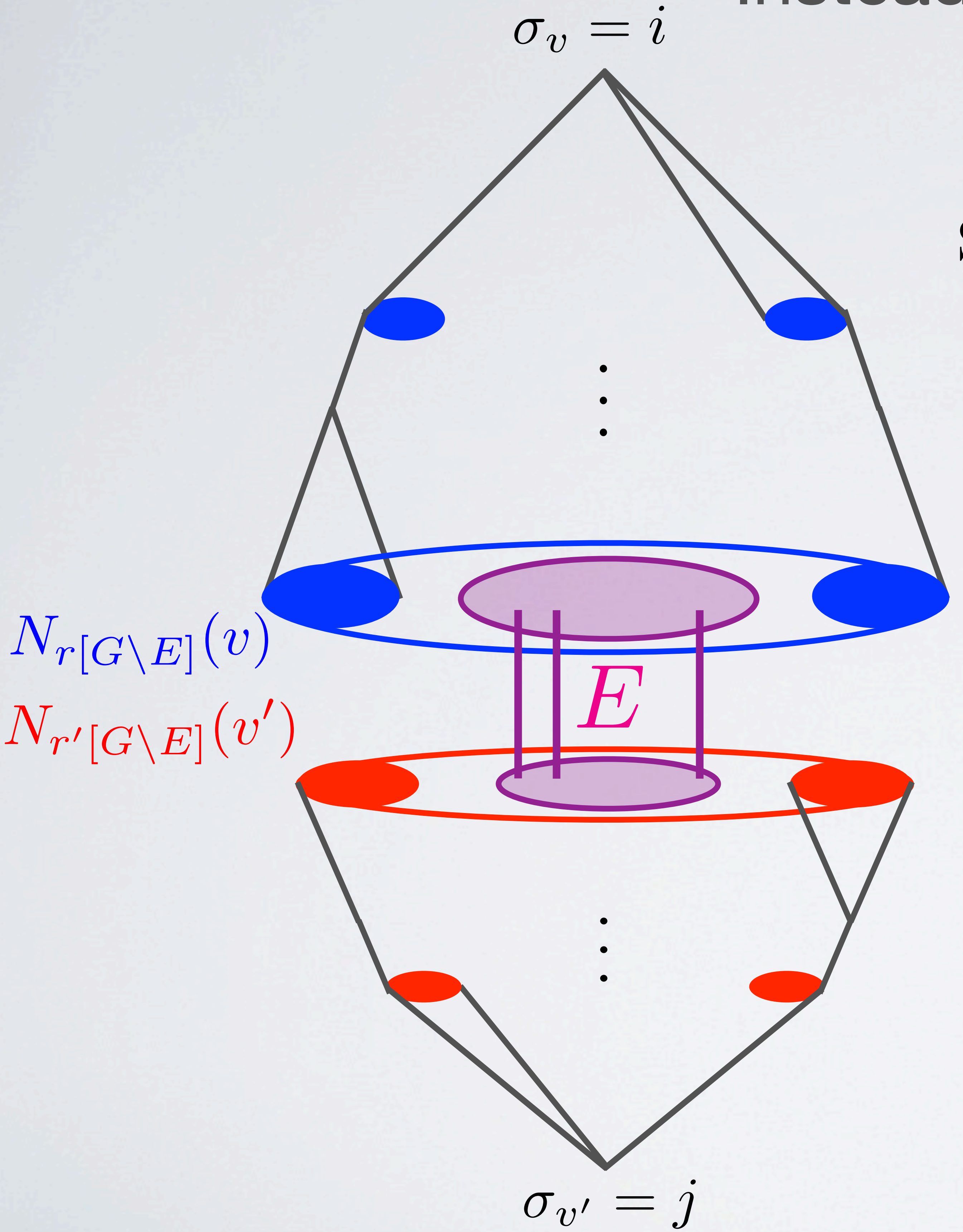
Subsample G with prob. c to get E

Compare v and v' from:

$$N_{r,r'[E]}(v \cdot v')$$

= number of such pairs of vertices

Instead: compare



Subsample G with prob. c to get E

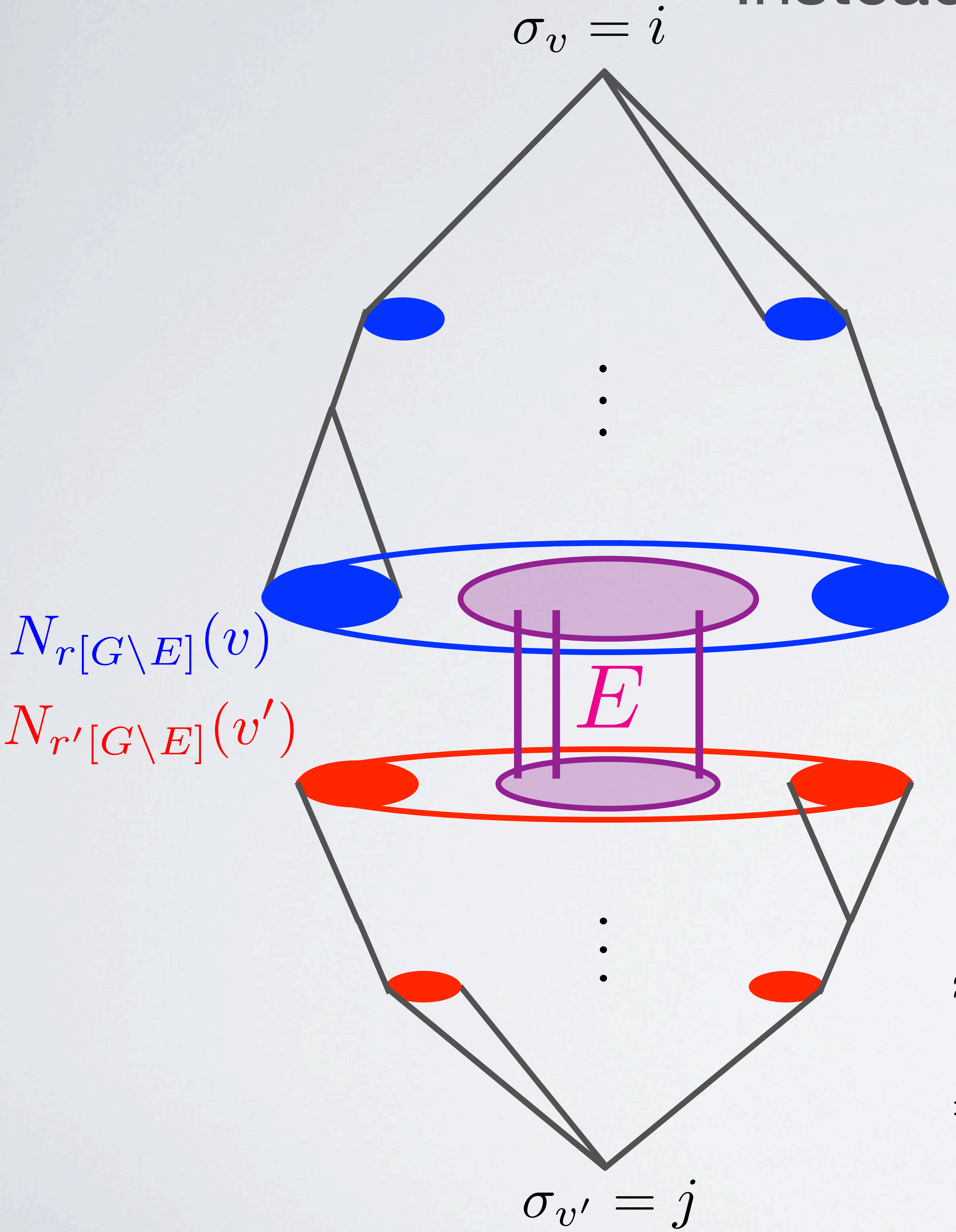
Compare v and v' from:

$$N_{r,r'[E]}(v \cdot v')$$

= number of such pairs of vertices

$$\approx N_{r[G\setminus E]}(v) \cdot \frac{cQ}{n} N_{r'[G\setminus E]}(v')$$

Instead: compare



Subsample G with prob. c to get E

Compare v and v' from:

$$N_{r,r'[E]}(v \cdot v')$$

= number of such pairs of vertices

$$\approx N_{r[G\setminus E]}(v) \cdot \frac{cQ}{n} N_{r'[G\setminus E]}(v')$$

$$\approx ((1-c)PQ)^r e_{\sigma_v} \cdot \frac{cQ}{n}((1-c)PQ)^{r'} e_{\sigma_{v'}}$$

$$= c(1-c)^{r+r'} e_{\sigma_v} \cdot Q(PQ)^{r+r'} e_{\sigma_{v'}}/n$$

The `degree-profiling` algorithm

The `degree-profiling` algorithm

(1) Split G into two graphs $\to$ G' sparse but large degree; G'' log-degree

The `degree-profiling` algorithm

(1) Split G into two graphs

- G' sparse but large degree
- G'' log-degree

(2) Run `Sphere-comparison` on G'

-> gets a fraction 1-o(1) with quasi-linear complexity

The `degree-profiling` algorithm

(1) Split G into two graphs → G' sparse but large degree; → G'' log-degree

(2) Run `Sphere-comparison` on G'

-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now G'' with the clustering of G'

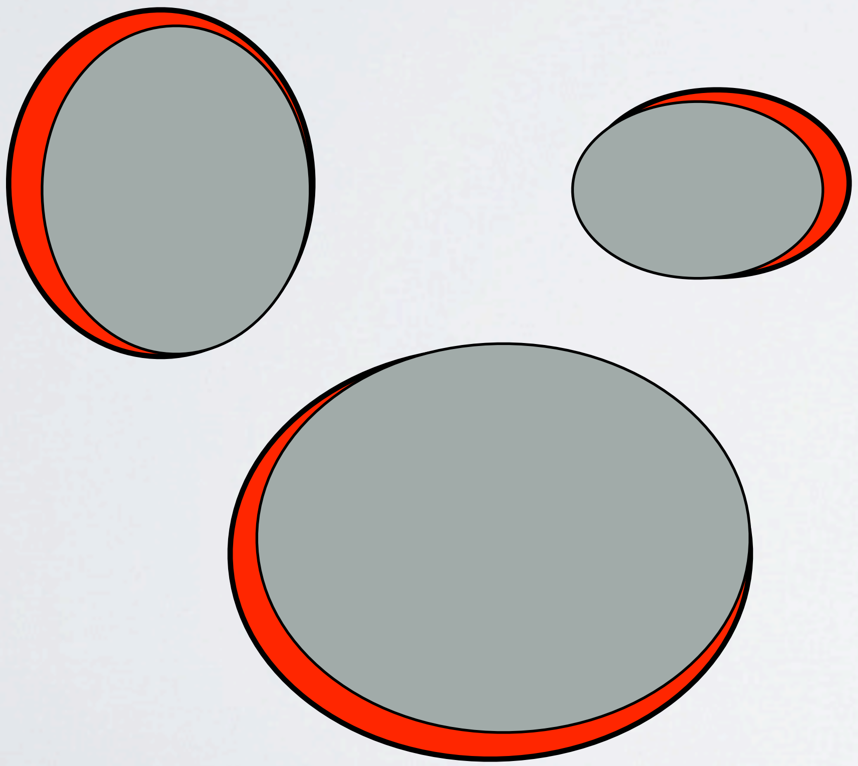
The degree-profiling algorithm

(1) Split G into two graphs → G' sparse but large degree; G'' log-degree

(2) Run `Sphere-comparison` on G'

-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now G'' with the clustering of G'



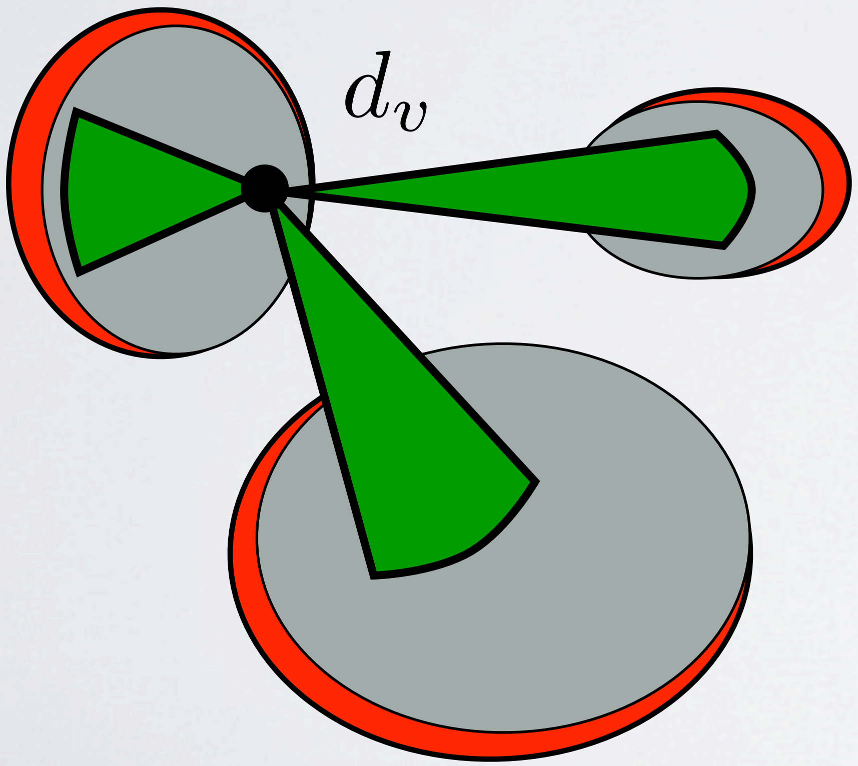
The `degree-profiling` algorithm

(1) Split G into two graphs → G' sparse but large degree; G'' log-degree

(2) Run `Sphere-comparison` on G'

-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now G'' with the clustering of G'



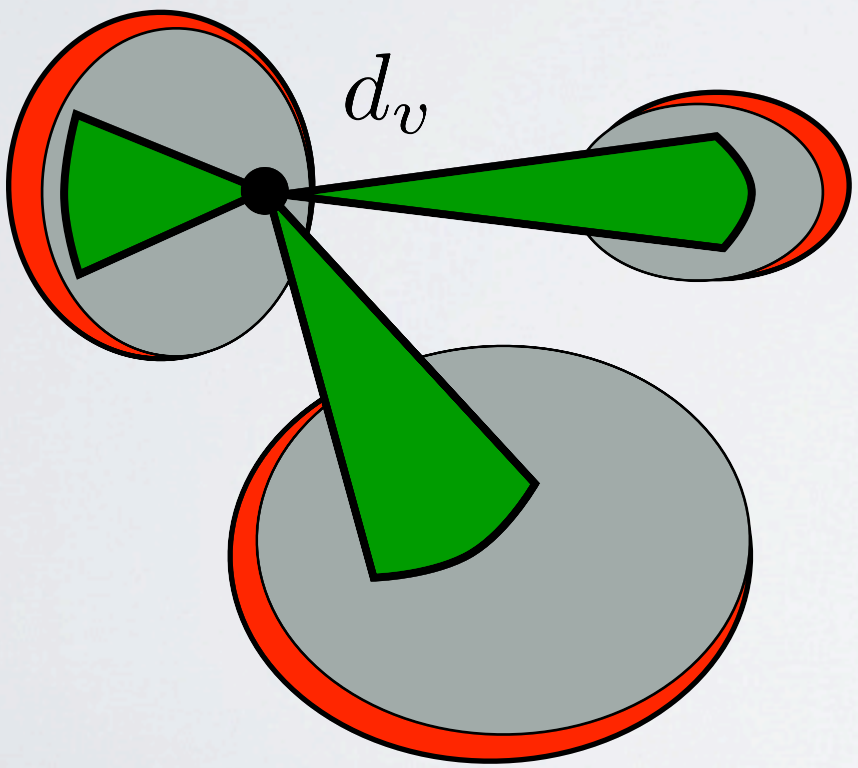
The `degree-profiling` algorithm

(1) Split G into two graphs → G' sparse but large degree
→ G" log-degree

(2) Run `Sphere-comparison` on G'

-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now G" with the clustering of G'



$$d_v \dddot{\sim} \mathcal{P}(\log(n)(PQ)_1)$$

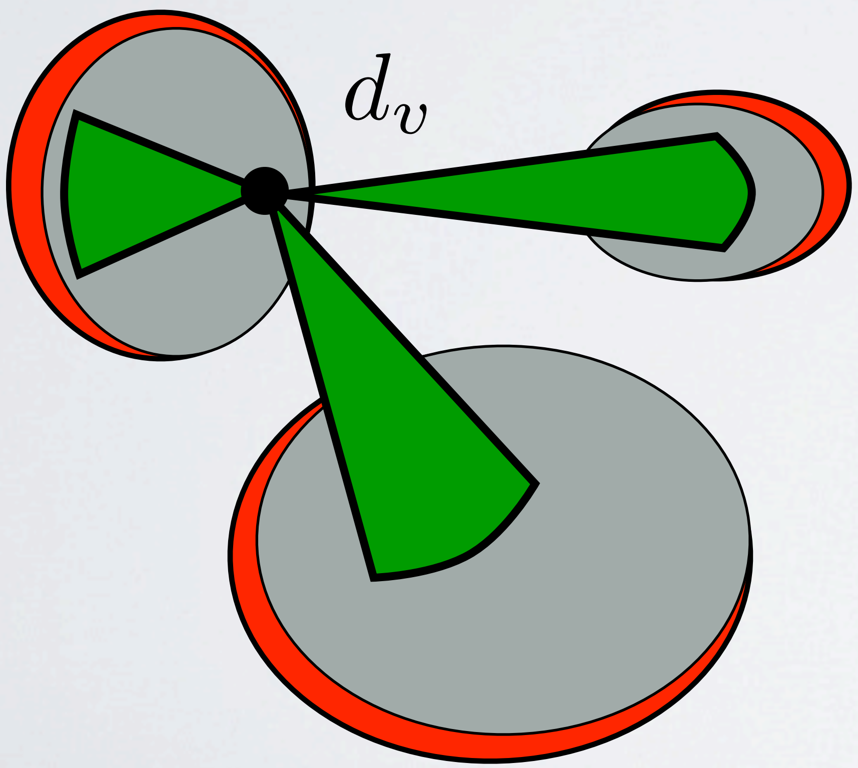
The degree-profiling algorithm

(1) Split G into two graphs
- G’ sparse but large degree
- G” log-degree

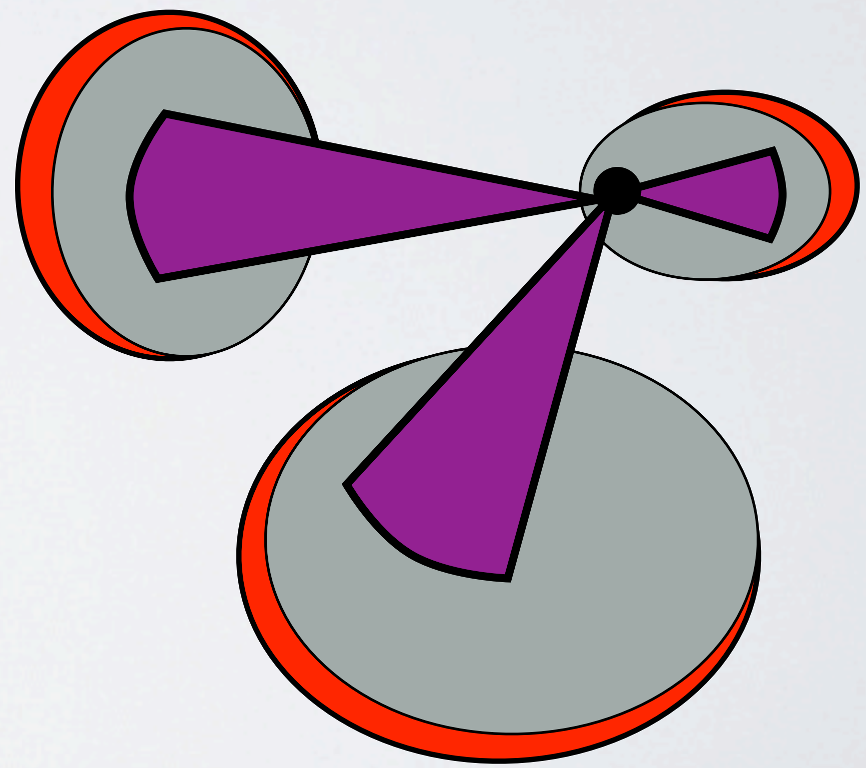
(2) Run `Sphere-comparison` on G’

-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now G” with the clustering of G’



$$d_v \ddot{\sim} \mathcal{P}(\log(n)(PQ)_1)$$



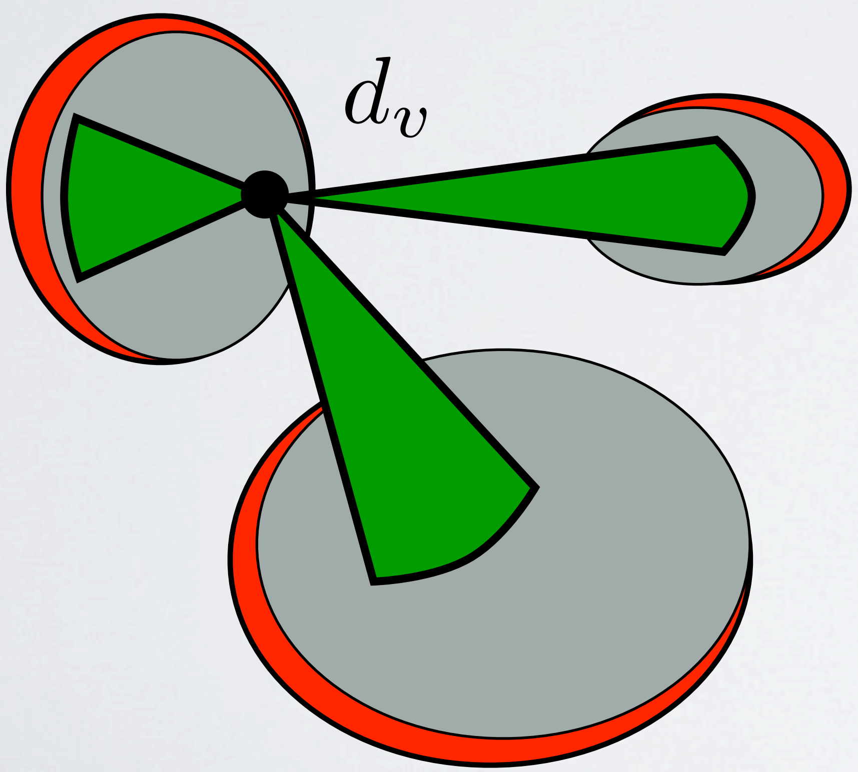
The `degree-profiling` algorithm

(1) Split G into two graphs → G' sparse but large degree
→ G'' log-degree

(2) Run `Sphere-comparison` on G'

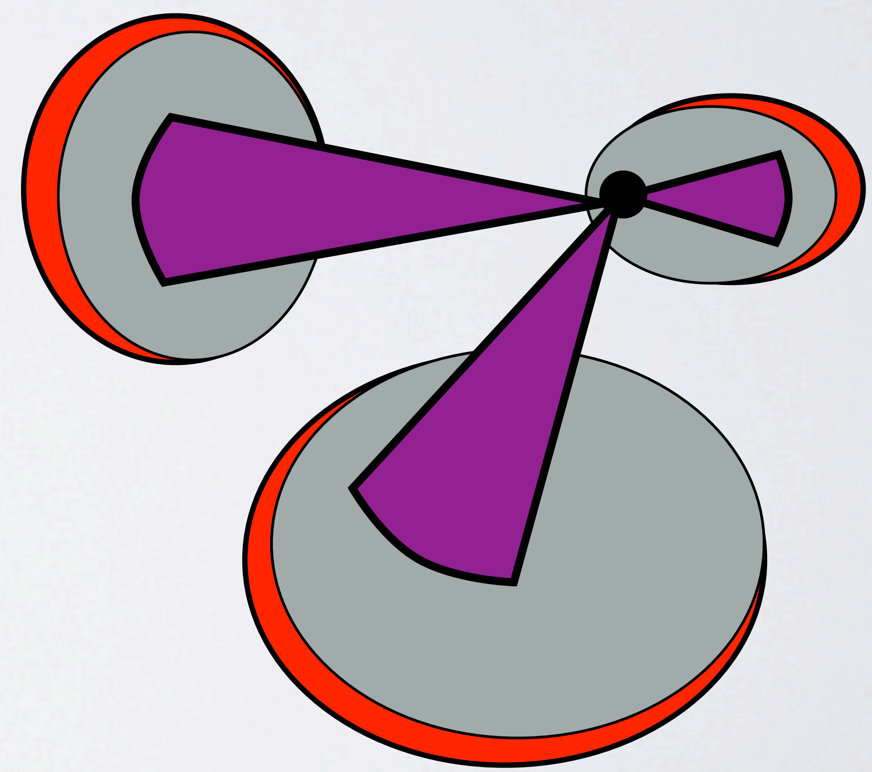
-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now G'' with the clustering of G'



$$d_v \overset{..}{\sim} \mathcal{P}(\log(n)(PQ)_1)$$

$$d_v \overset{..}{\sim} \mathcal{P}(\log(n)(PQ)_2)$$



The degree-profiling algorithm

(1) Split G into two graphs

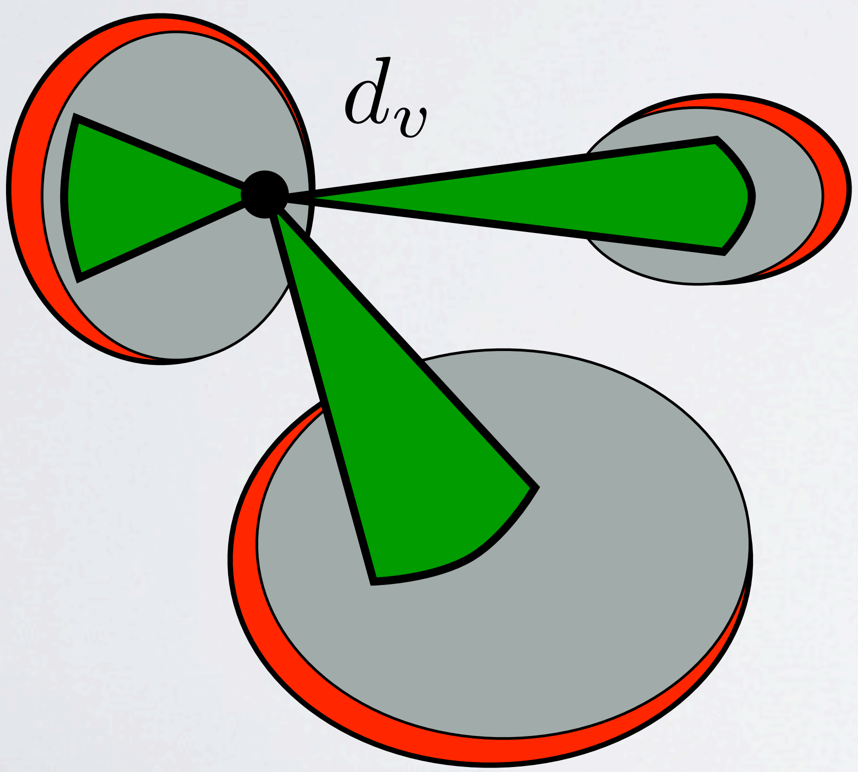
G' sparse but large degree

G'' log-degree

(2) Run `Sphere-comparison` on G'

-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now G'' with the clustering of G'

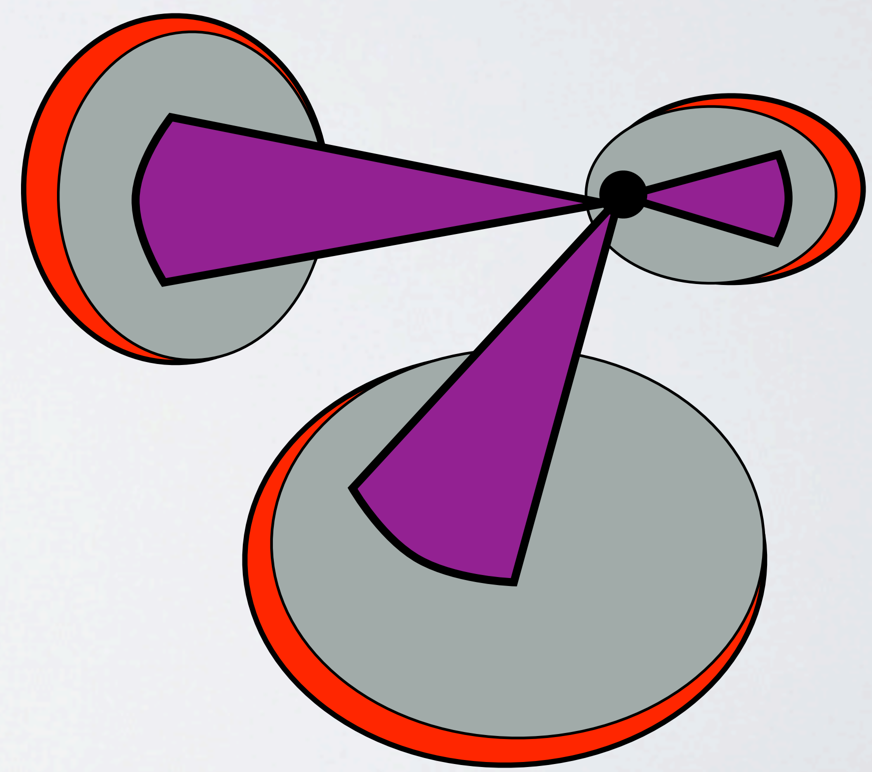


Hypothesis 1

$$d_v \ddot{\sim} \mathcal{P}(\log(n)(PQ)_1)$$

Hypothesis 2

$$d_v \ddot{\sim} \mathcal{P}(\log(n)(PQ)_2)$$



The degree-profiling algorithm

(1) Split G into two graphs

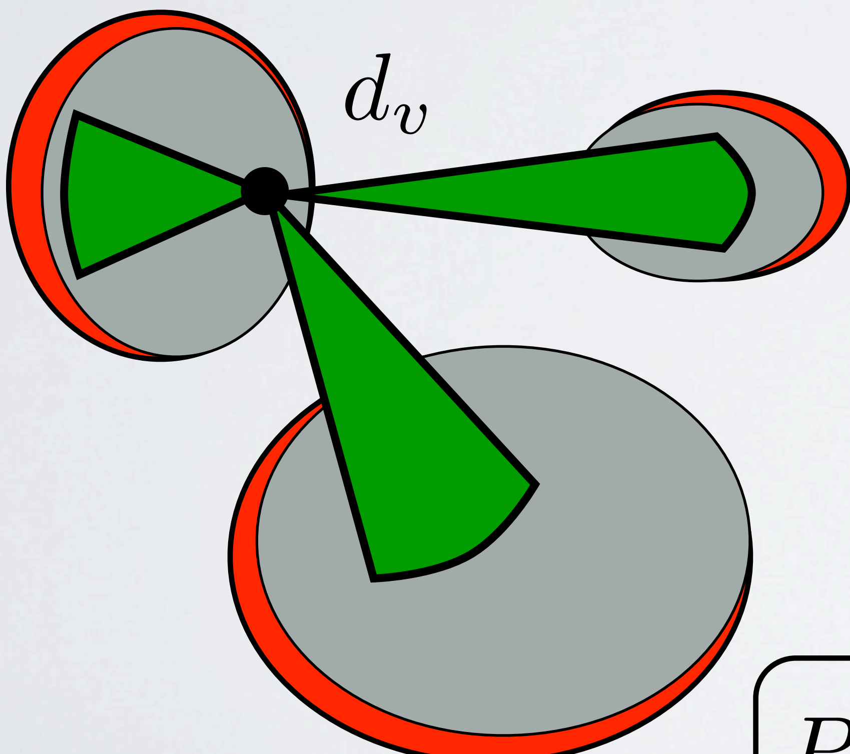
$\mathrm{G'}$ sparse but large degree

$\mathrm{G''}$ log-degree

(2) Run `Sphere-comparison` on $\mathrm{G'}$

-> gets a fraction 1-o(1) with quasi-linear complexity

(3) Take now $\mathrm{G''}$ with the clustering of $\mathrm{G'}$

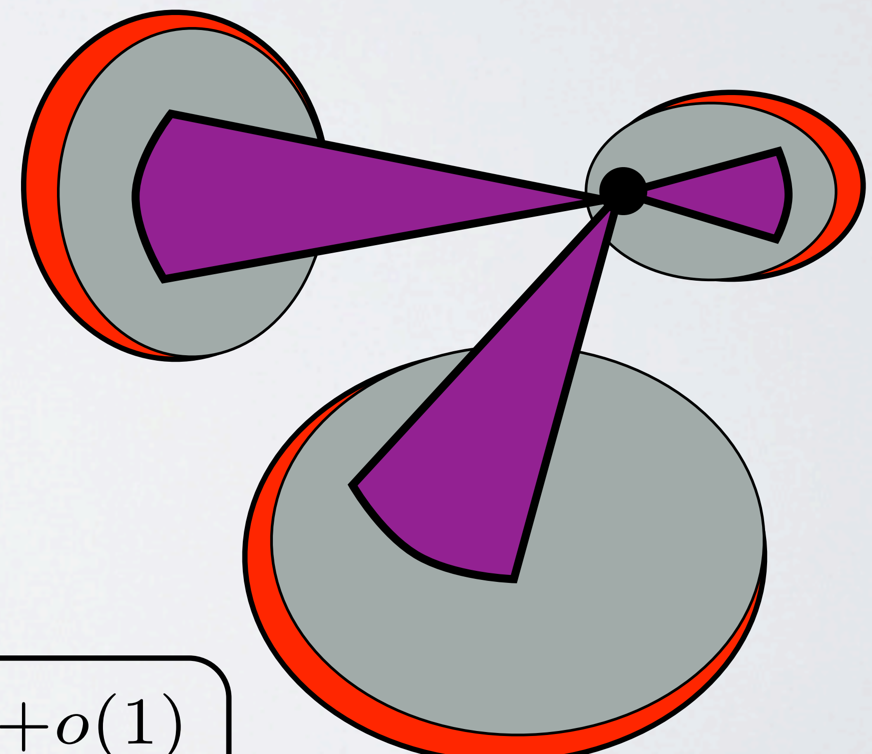


Hypothesis 1

$$d_v \overset{\cdot\cdot\cdot}{\sim} \mathcal{P}(\log(n)(PQ)_1)$$

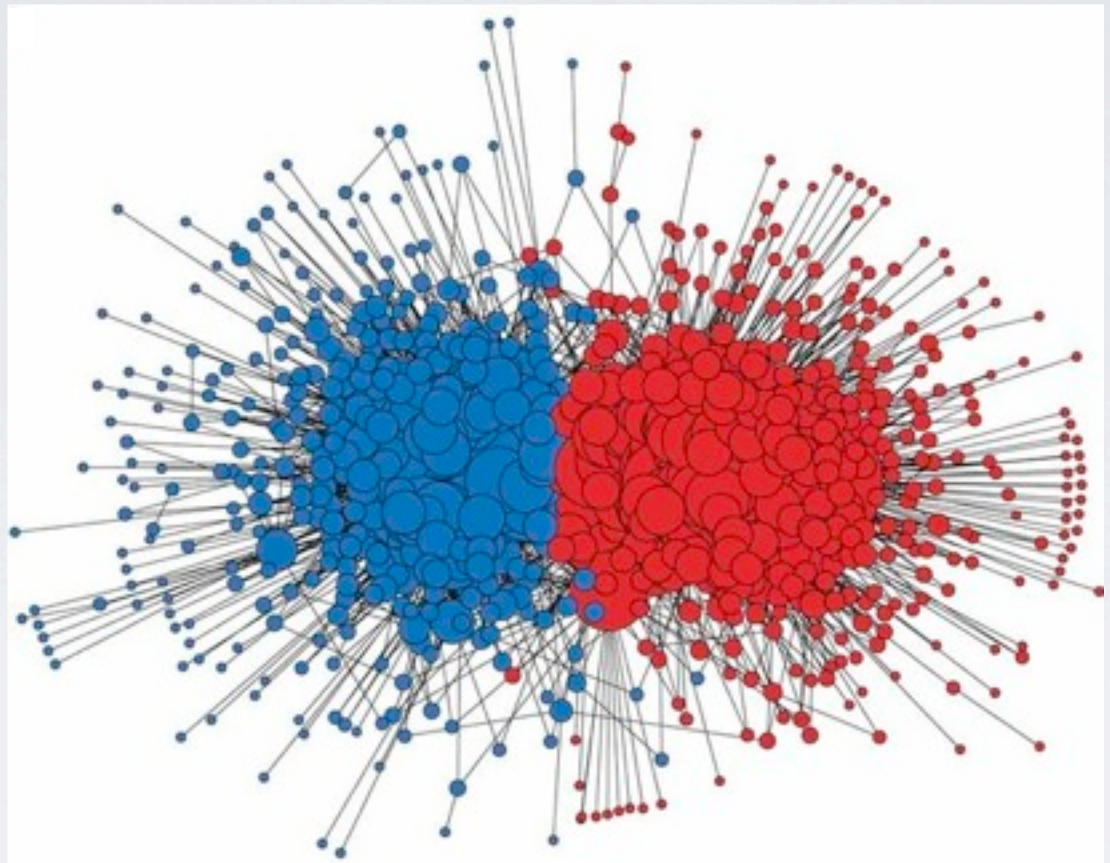
Hypothesis 2

$$d_v \overset{\cdot\cdot\cdot}{\sim} \mathcal{P}(\log(n)(PQ)_2)$$



$$\boxed{P_e = n^{-D_+((PQ)_1,(PQ)_2)+o(1)}}$$

Some data: the blog network



1490 blogs
(left- and right-leaning)
[Adamic and Glance ’05]

$$Q_{11} \approx Q_{22} \approx 5.5 \log(n)/n$$
$$Q_{12} \approx 0.5 \log(n)/n$$

$$\sqrt{a} - \sqrt{b} \approx 1.6 > 1.41$$
$$95\%$$

Open problems

- exact distortion curve for partial recovery
- other models
- universal results
- detection with multiple symmetric clusters

Advertisement

- Tutorial on Information Theory and Machine Learning, ISIT 2015, Hong Kong