What can be Computed in a Simple Chaotic Way?

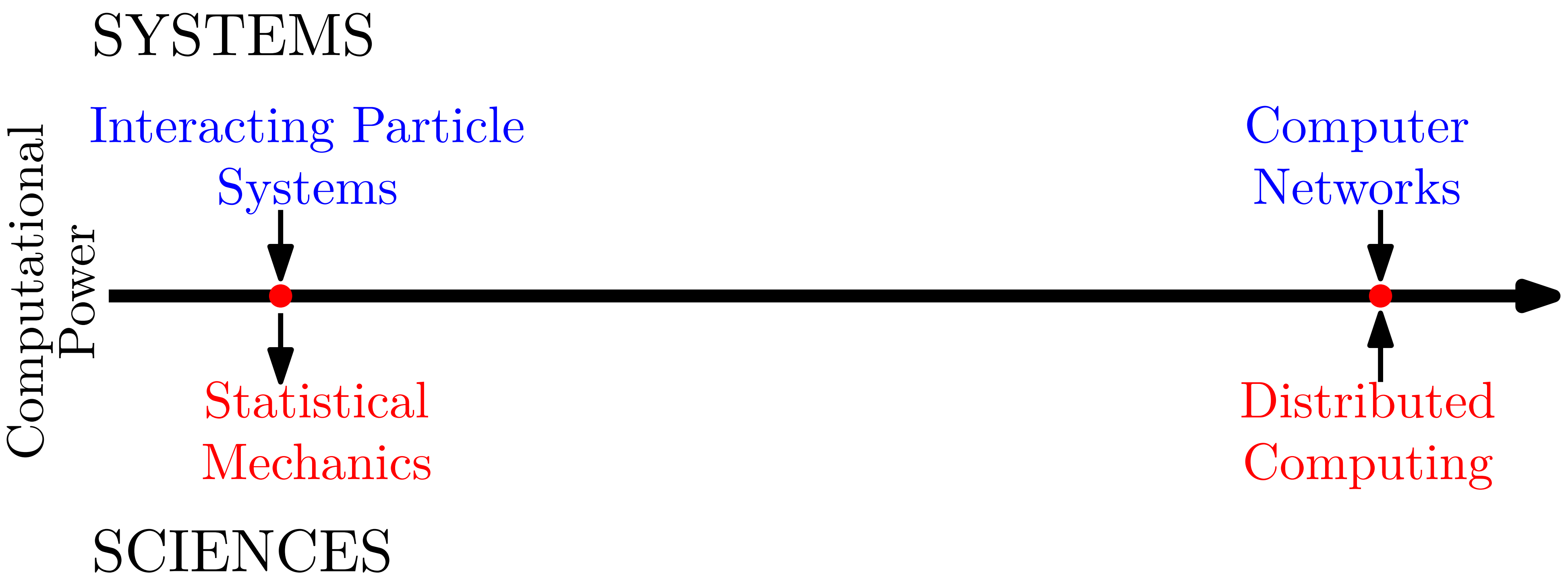
Emanuele Natale

max planck institut informatik

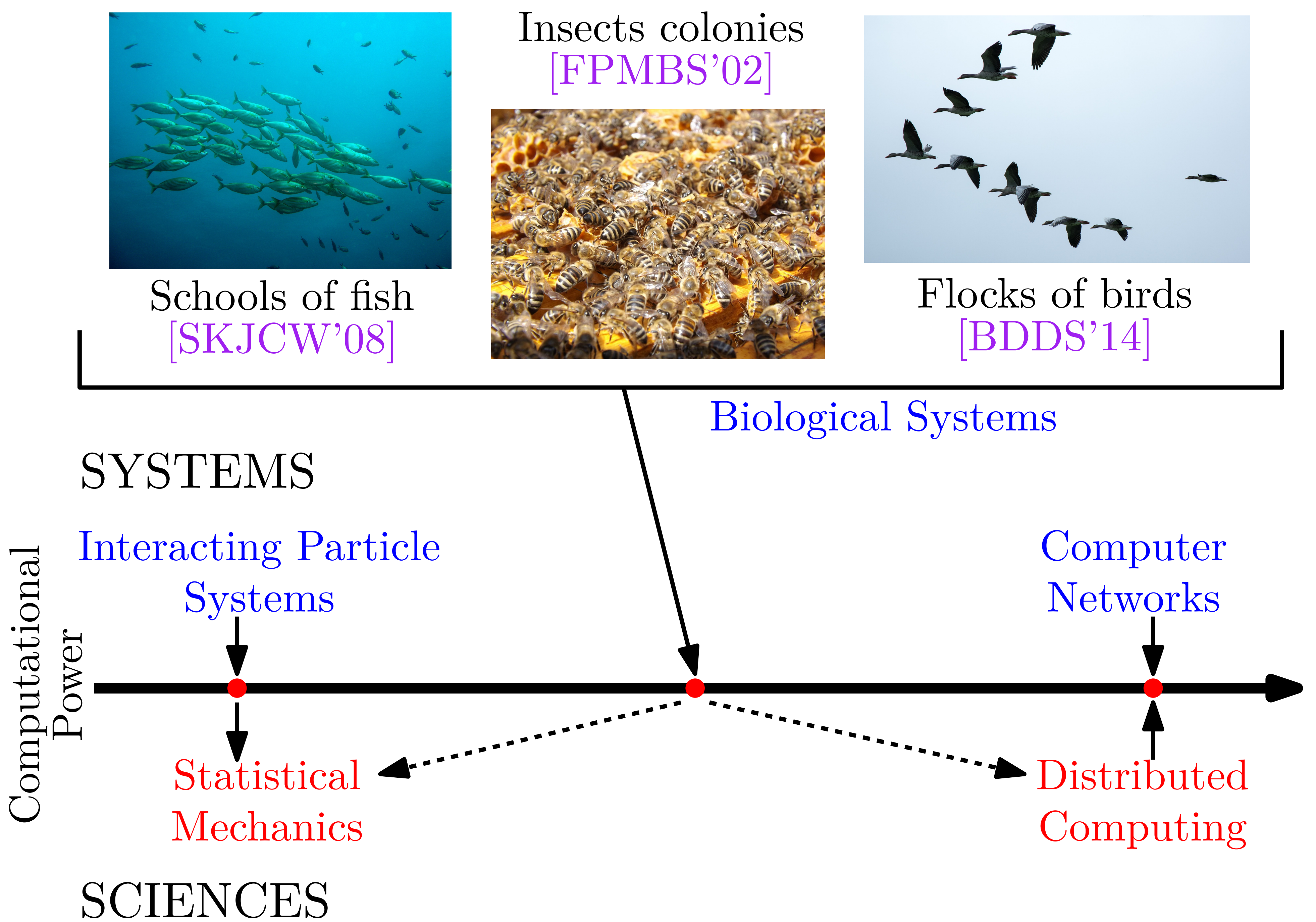
ADFOCS 2017

ADFOCS 25 August 2017,
MPII, Saarbrücken

What can *Simple* Systems do?



What can *Simple* Systems do?



Dynamics

(informal)

Very simple distributed algorithms:
For every graph, agent and round, states are updated according to fixed (random) rule of current state and symmetric function of states of neighbors.

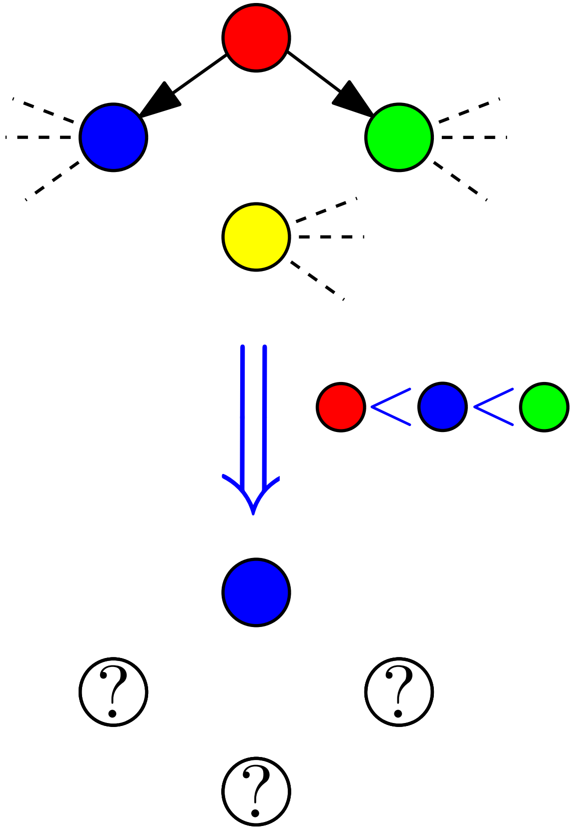
Dynamics

(informal)

Very simple distributed algorithms:
For every graph, agent and round, states are updated according to fixed (random) rule of current state and symmetric function of states of neighbors.

Examples of Dynamics

- 3-Median dynamics



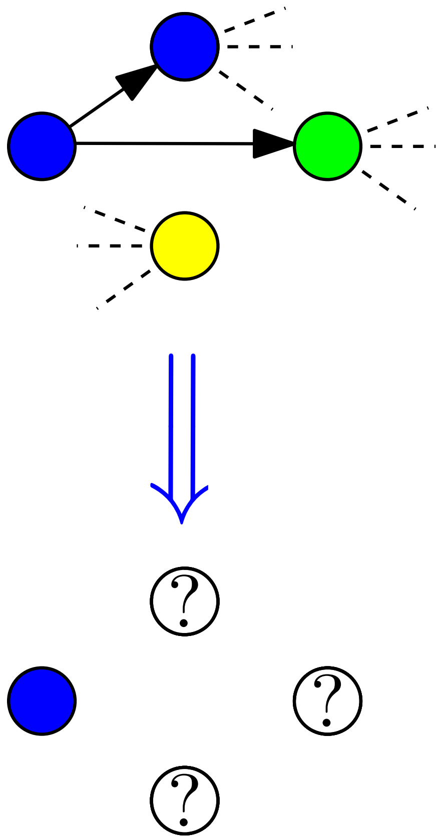
Dynamics

(informal)

Very simple distributed algorithms:
For every graph, agent and round, states are updated according to fixed (random) rule of current state and symmetric function of states of neighbors.

Examples of Dynamics

- 3-Median dynamics
- 2-Choice dynamics



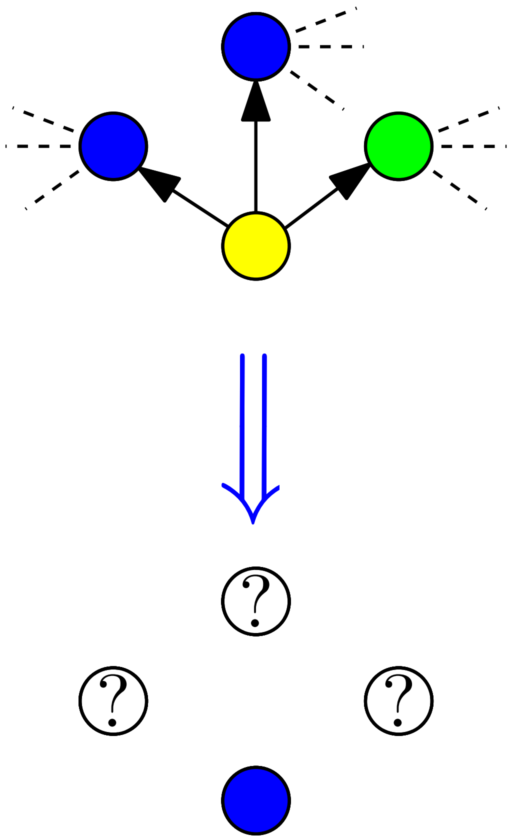
Dynamics

(informal)

Very simple distributed algorithms:
For every graph, agent and round, states are updated according to fixed (random) rule of current state and symmetric function of states of neighbors.

Examples of Dynamics

- 3-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics



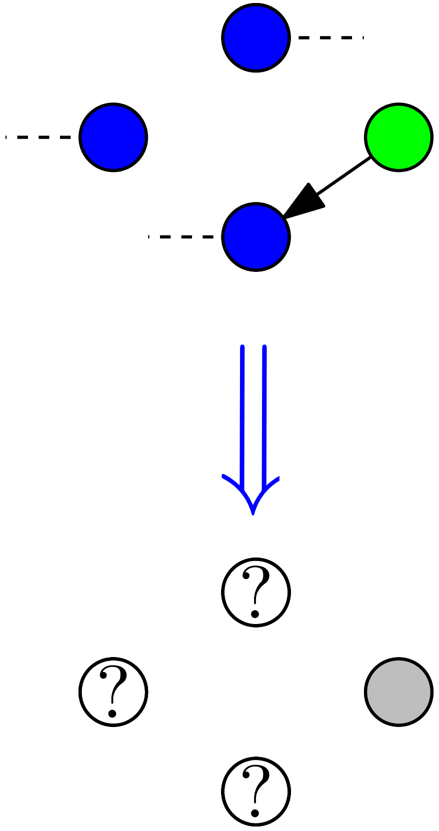
Dynamics

(informal)

Very simple distributed algorithms:
For every graph, agent and round, states are updated according to fixed (random) rule of current state and symmetric function of states of neighbors.

Examples of Dynamics

- 3-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics
- Undecided-state dynamics



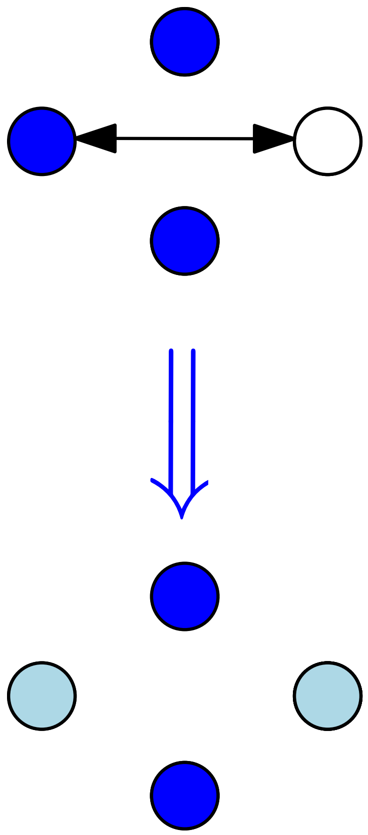
Dynamics

(informal)

Very simple distributed algorithms:
For every graph, agent and round, states are updated according to fixed (random) rule of current state and symmetric function of states of neighbors.

Examples of Dynamics

- 3-Median dynamics
- 2-Choice dynamics
- 3-Majority dynamics
- Undecided-state dynamics
- Averaging dynamics
  (asynchronous)



Some Results on Dynamics

On the complete graph:

3-Median dynamics [DGMSS '11]. Converge to $\mathcal{O}(\sqrt{n \log n})$ approximation of median of system in $\mathcal{O}(\log n)$ rounds w.h.p.

3-Majority dynamics [BCNPS '14, BCNPT '16, BCEKMN '17]. If plurality has **bias** $\mathcal{O}(\sqrt{kn \log n})$, converges to it in $\mathcal{O}(k \log n)$ rounds w.h.p., even against $o(\sqrt{n/k})$-bounded adversary. Without bias, converges in $\mathrm{poly}(k)$. When k is large, polynomial separation w.r.t. 2-Choice.

Undecided-State dynamics [BCNPST '15]. If majority/second-majority is at least $1+\epsilon$, system converges to plurality within $\tilde{\Theta}(\sum_i (\frac{\#\{\text{majority nodes}\}}{\#\{i\text{-colored nodes}\}})^2)$ rounds w.h.p.,

Some Results on Dynamics

On the complete graph:

3-Median dynamics [DGMSS '11]. Converge to $\mathcal{O}(\sqrt{n \log n})$ approximation of median of system in $\mathcal{O}(\log n)$ rounds w.h.p.

3-Majority dynamics [BCNPS '14, BCNPT '16, BCEKMN '17]. If plurality has **bias** $\mathcal{O}(\sqrt{kn \log n})$, converges to it in $\mathcal{O}(k \log n)$ rounds w.h.p., even against $o(\sqrt{n/k})$-bounded adversary. Without bias, converges in $\mathrm{poly}(k)$. When k is large, polynomial separation w.r.t. 2-Choice.

Undecided-State dynamics [BCNPST '15]. If majority/second-majority is at least $1 + \epsilon$, system converges to plurality within $\tilde{\Theta}(\sum_i (\frac{\#\{\text{majority nodes}\}}{\#\{i\text{–colored nodes}\}})^2)$ rounds w.h.p.,

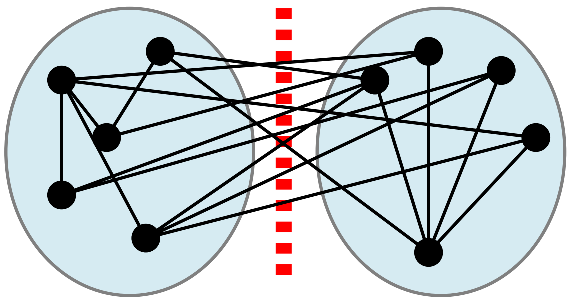
Dynamics can solve Consensus, Median, Majority, in a robust way, but this is trivial in centralized setting.. **Can they solve a problem non-trivial in centralized setting?**

Community Detection

Min. Bisection Problem.

Given a graph G with $2n$ nodes. Find

$$S = \arg\min_{\substack{S \subset V \\ |S| = n}} E(S, V - S).$$

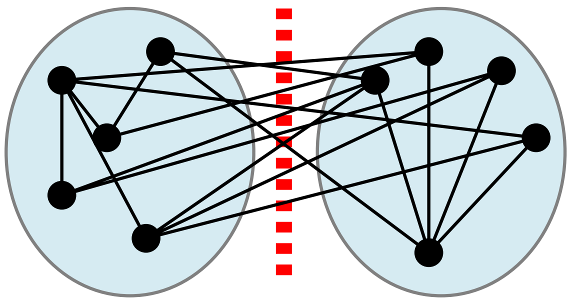


[GJS '76]: **Min. Bisection** is *NP-Complete*.

Community Detection

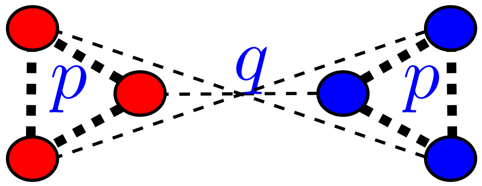
Min. Bisection Problem.
Given a graph G with $2n$ nodes. Find

$$S = \arg\min_{\substack{S \subset V \\ |S| = n}} E(S, V - S).$$



[GJS '76]: **Min. Bisection** is *NP-Complete*.

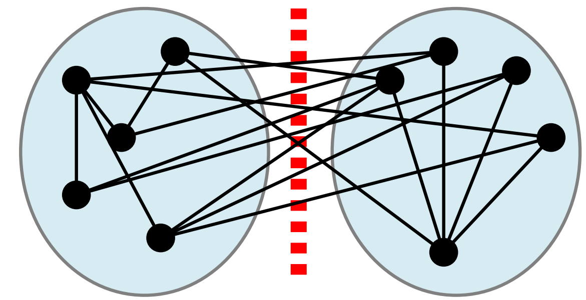
Stochastic Block Model. Two "communities" of equal size V_1 and V_2, each edge inside a community included with probability p, each edge across communities included with probability $q < p$.



Community Detection

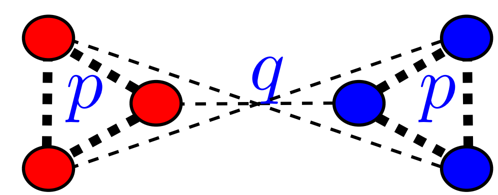
Min. Bisection Problem.
Given a graph G with $2n$ nodes. Find

$$S = \arg\min_{\substack{S \subset V \\ |S| = n}} E(S, V - S).$$

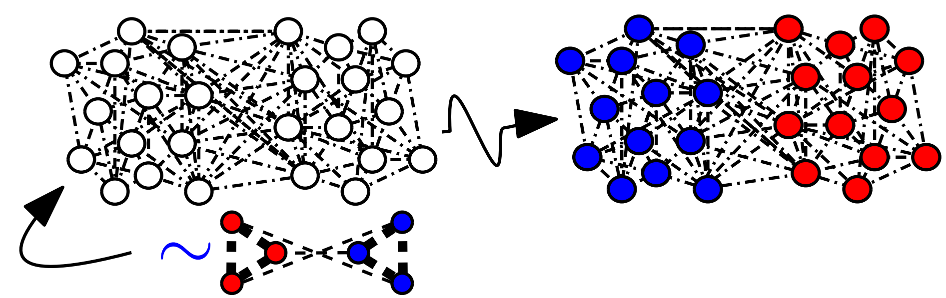


[GJS '76]: **Min. Bisection** is *NP-Complete*.

Stochastic Block Model. Two "communities" of equal size V_1 and V_2, each edge inside a community included with probability p, each edge across communities included with probability $q < p$.



Reconstruction problem. Given graph generated by SBM, find original partition.

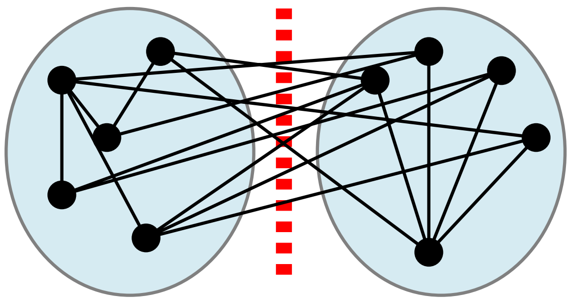


Community Detection

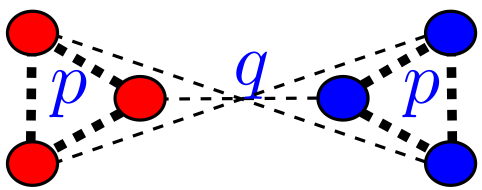
Min. Bisection Problem.
Given a graph G with $2n$ nodes. Find

$$S = \arg\min_{\substack{S \subset V \\ |S| = n}} E(S, V - S).$$

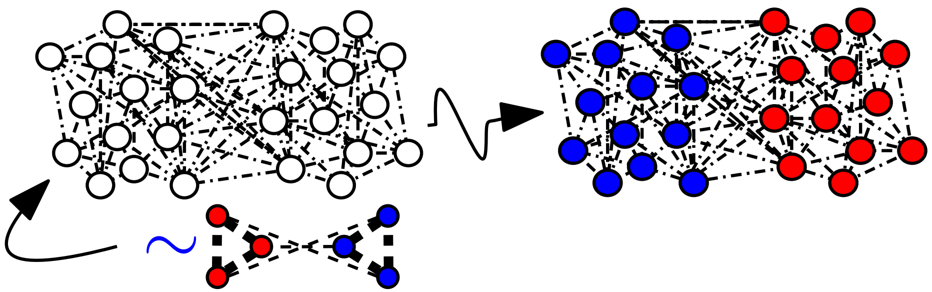
[GJS '76]: **Min. Bisection** is *NP-Complete*.



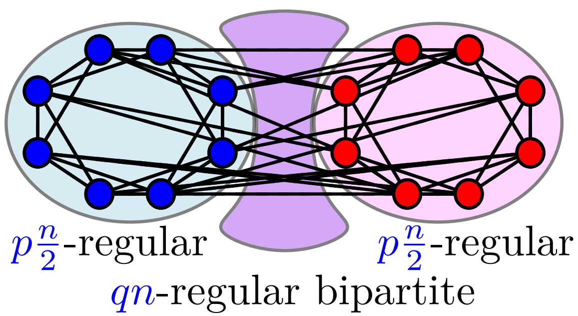
Stochastic Block Model. Two "communities" of equal size V_1 and V_2, each edge inside a community included with probability p, each edge across communities included with probability $q < p$.



Reconstruction problem. Given graph generated by SBM, find original partition.



Regular SBM [BDGHT '15]. Graph induced by communities are $p\frac{n}{2}$-regular random, graph induced by cut is qn-regular random.

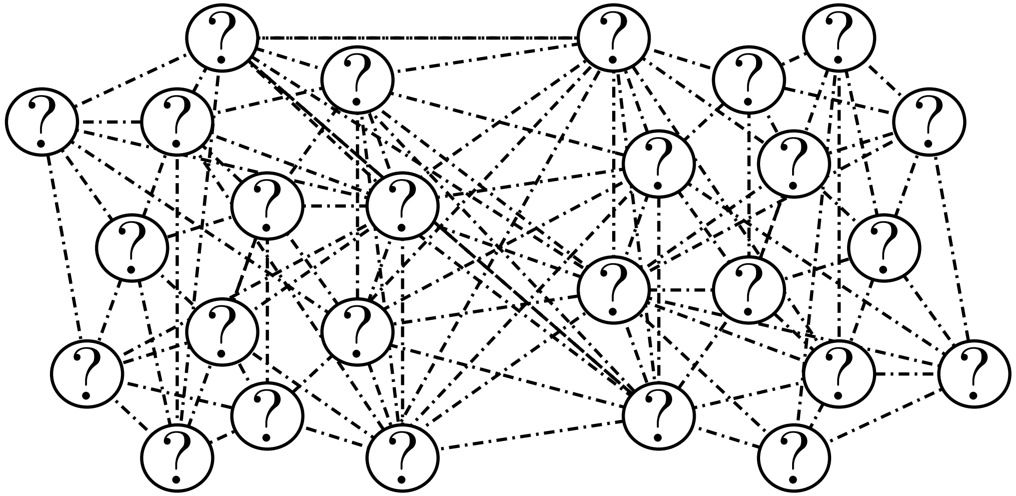


The Averaging Dynamics

Asynchronous Averaging Protocol:

At each round a random edge is chosen.

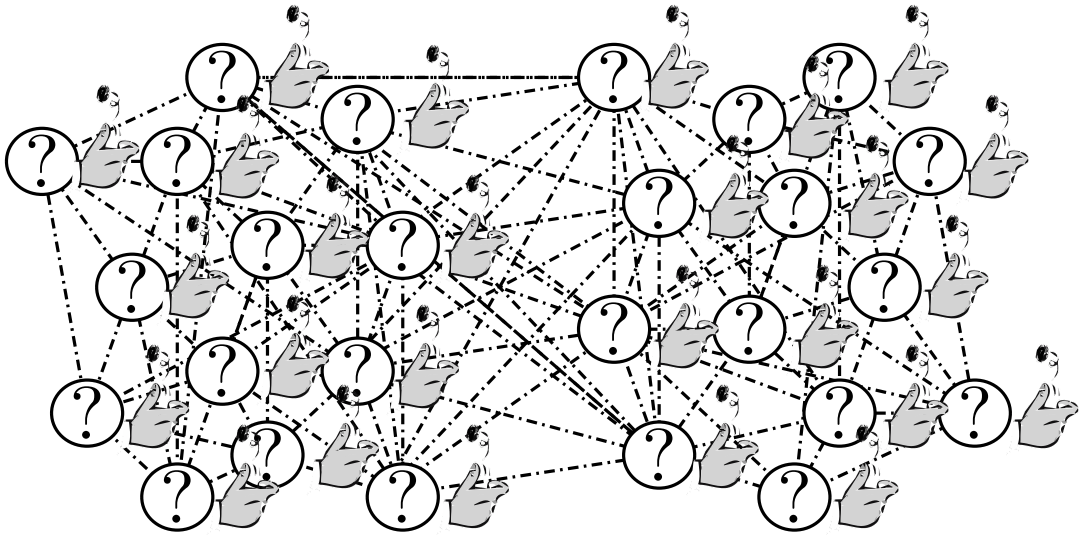
- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.



The Averaging Dynamics

Asynchronous Averaging Protocol:
At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.

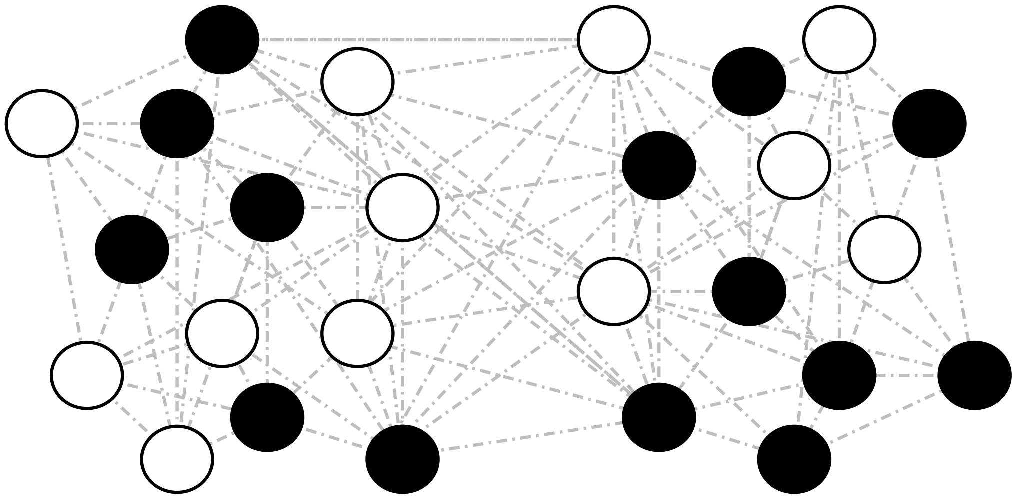


The Averaging Dynamics

Asynchronous Averaging Protocol:

At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.

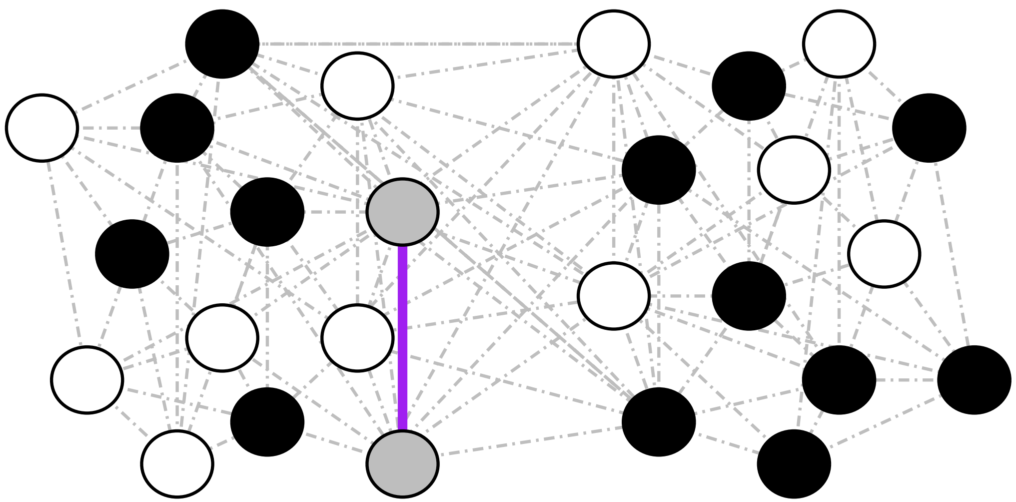


The Averaging Dynamics

Asynchronous Averaging Protocol:

At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.

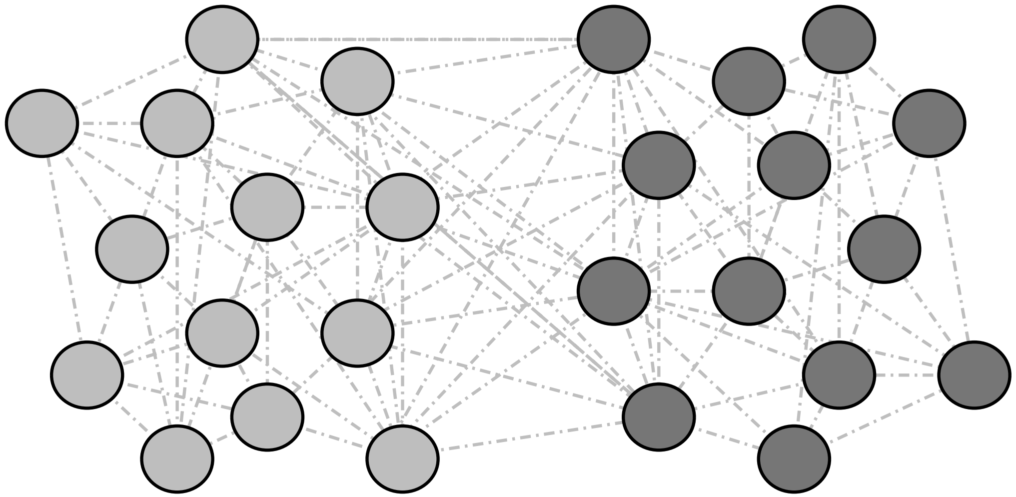


The Averaging Dynamics

Asynchronous Averaging Protocol:

At each round a random edge is chosen.

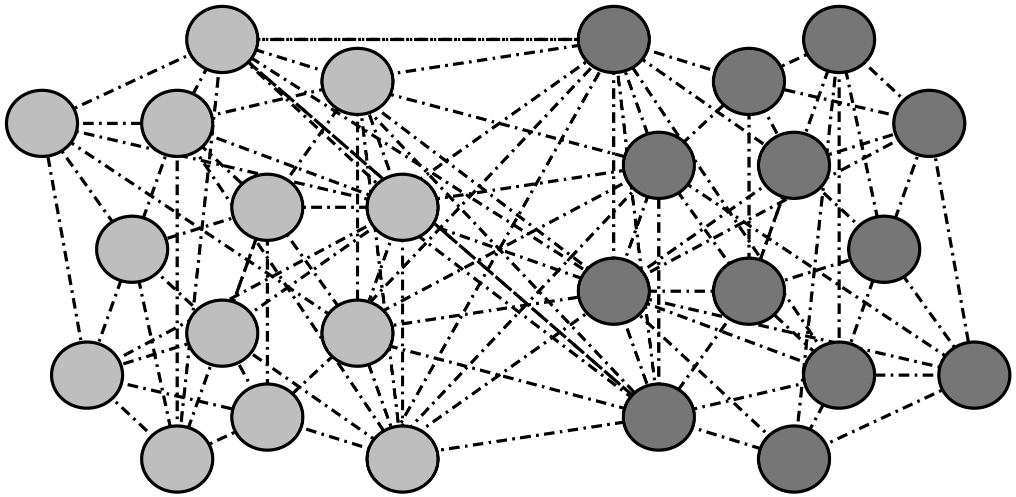
- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.



The Averaging Dynamics

Asynchronous Averaging Protocol:
At each round a random edge is chosen.

- At the first activation, each node picks at random $+1$ or -1.
- (Dynamics) At each activation, the nodes averages their values.

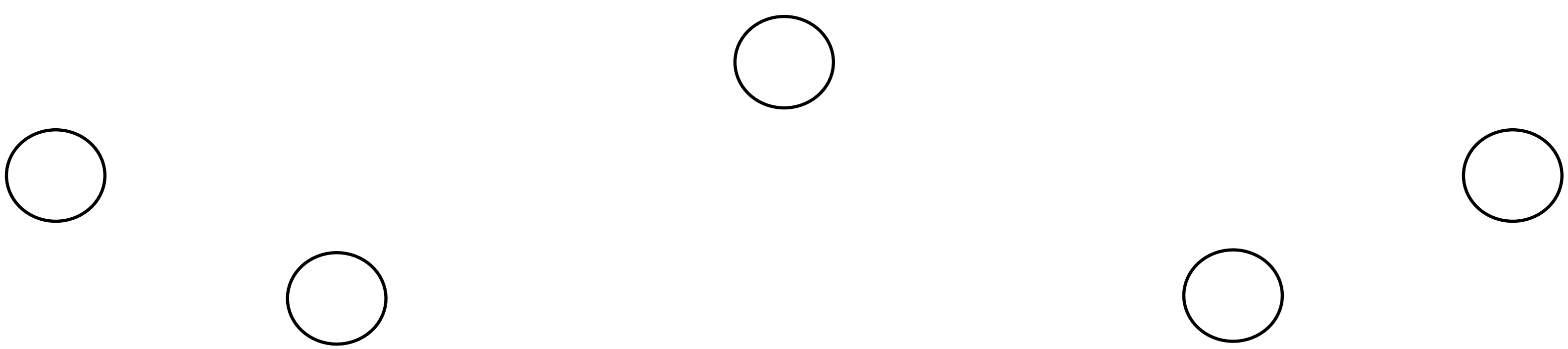


Theorem (Corollary of [BCMNPRT'17(Soon on Arxiv)]).
There exist τ_1, τ_2 s.t., if each node labels itself with the sign of the difference of its value at two activation times τ_1 and τ_2, then with prob. $1-\epsilon$, after $O_\varepsilon(n \log n + \frac{n}{\lambda_2})$ rounds, we get a correct reconstruction up to an ϵ-fraction of nodes.

"$\mathbb{E}$[Averaging Dynamics]"

Al nodes at the same time:

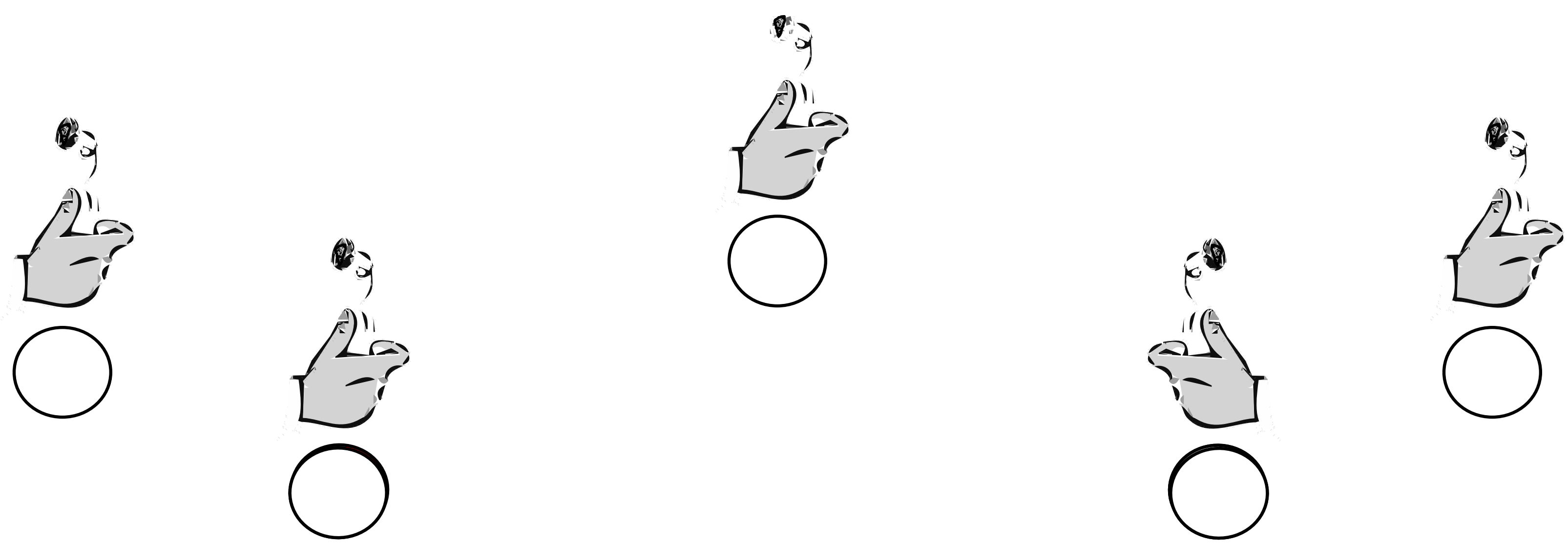
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



“$\mathbb{E}$[Averaging Dynamics]”

Al nodes at the same time:

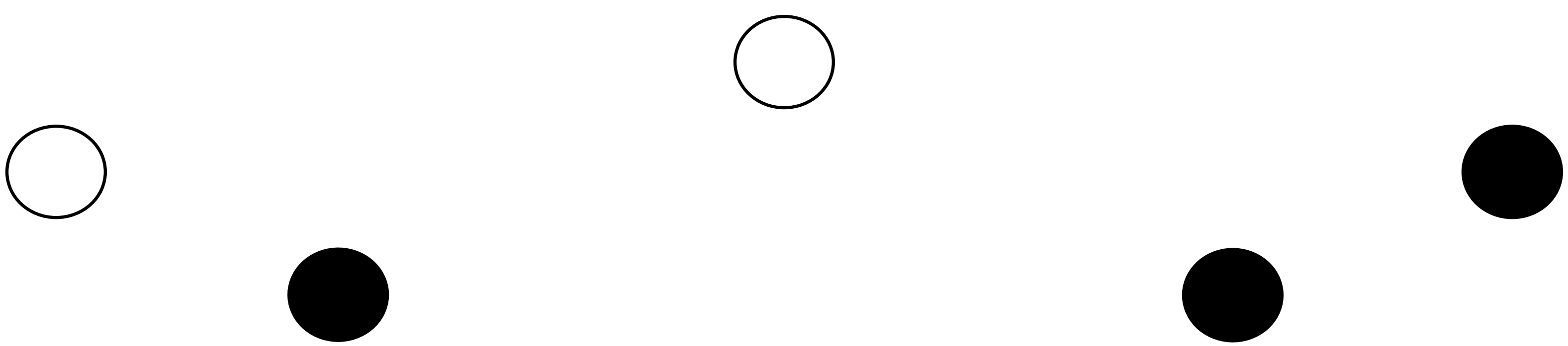
- At $t = 0$, randomly pick value $x^{(t)} \in \{+\mathbf{1}, \text{-}\mathbf{1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



"$\mathbb{E}$[Averaging Dynamics]"

Al nodes at the same time:

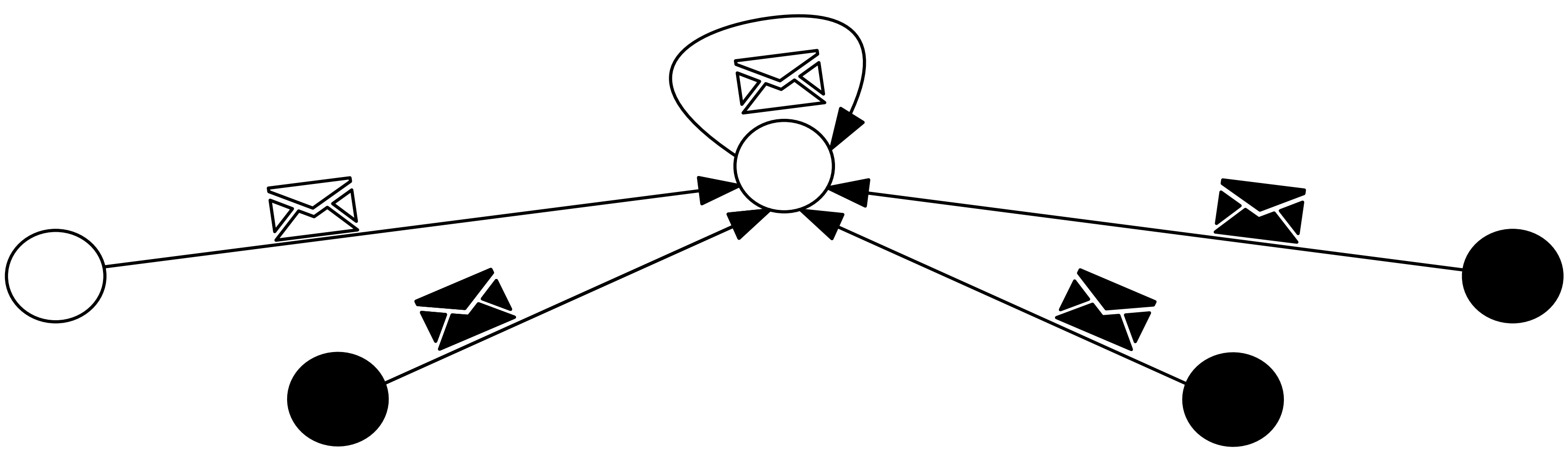
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



“$\mathbb{E}$[Averaging Dynamics]”

Al nodes at the same time:

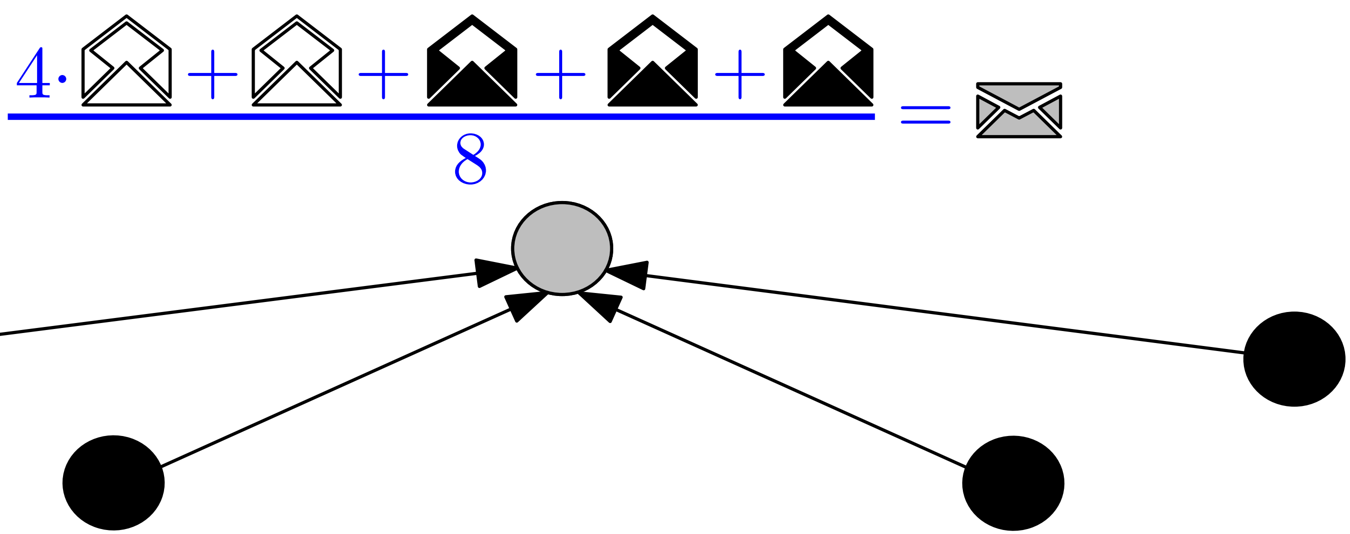
- At $t = 0$, randomly pick value $x^{(t)} \in \{+1, -1\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



"$\mathbb{E}$[Averaging Dynamics]"

Al nodes at the same time:

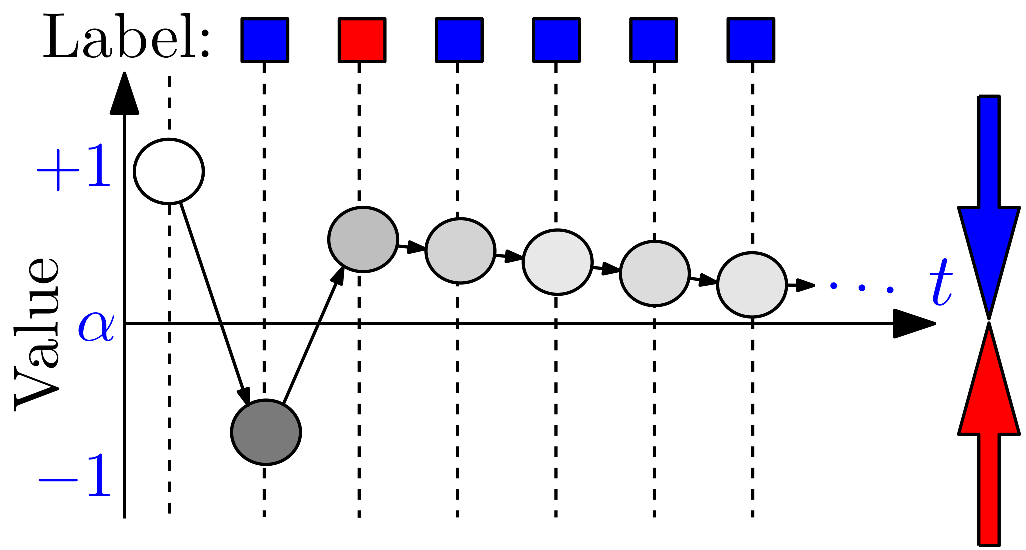
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

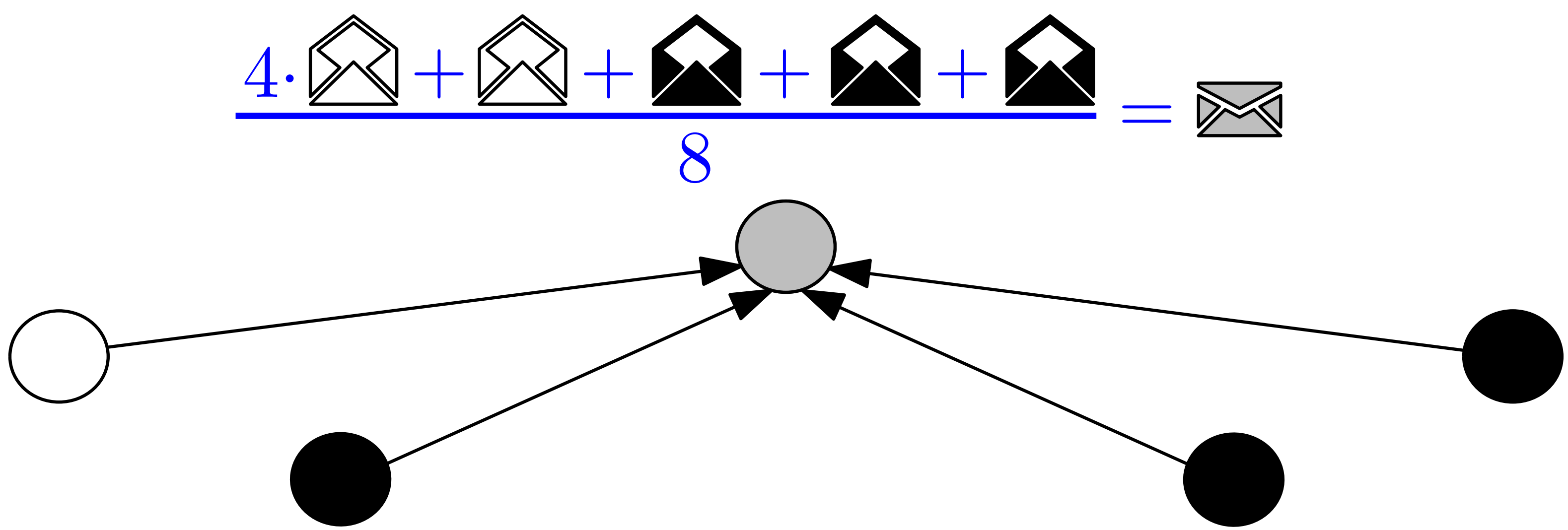


“$\mathbb{E}$[Averaging Dynamics]”

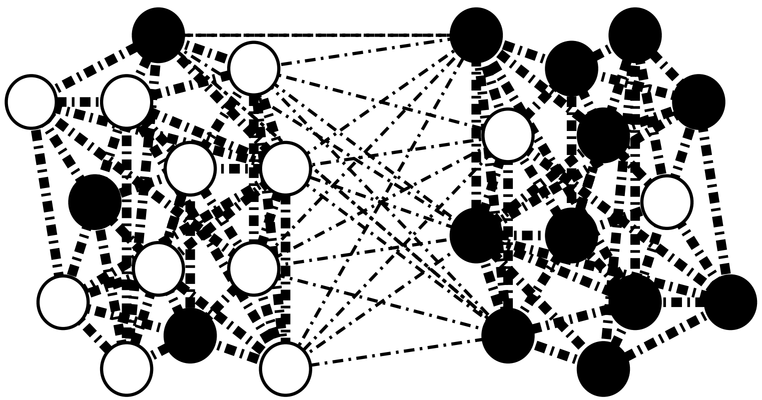
Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to lazy average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

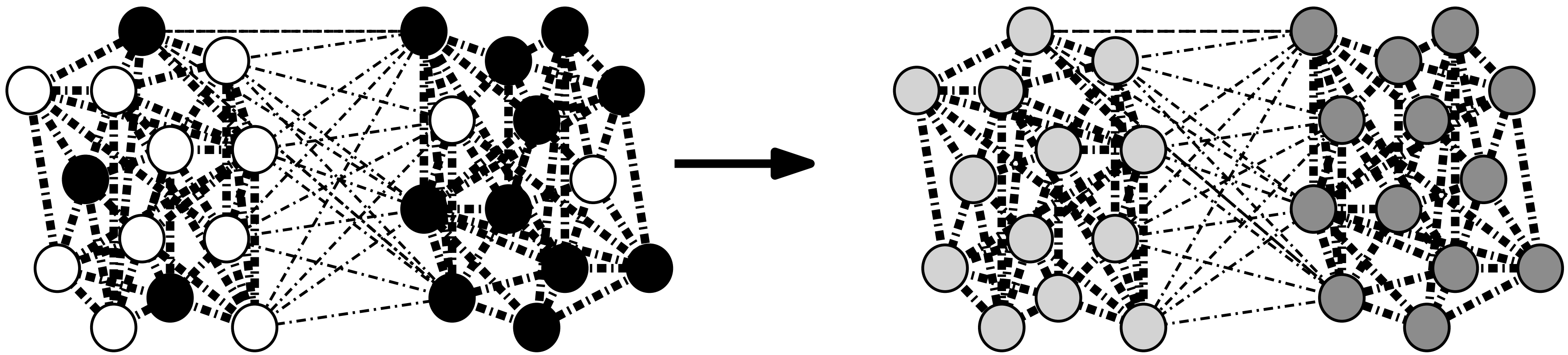




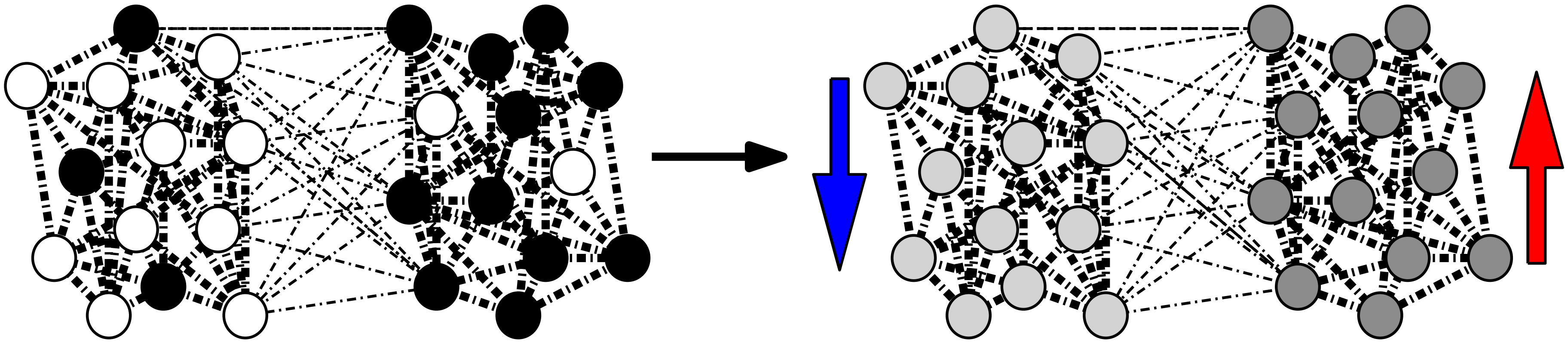
Community Detection via (Parallel) Averaging

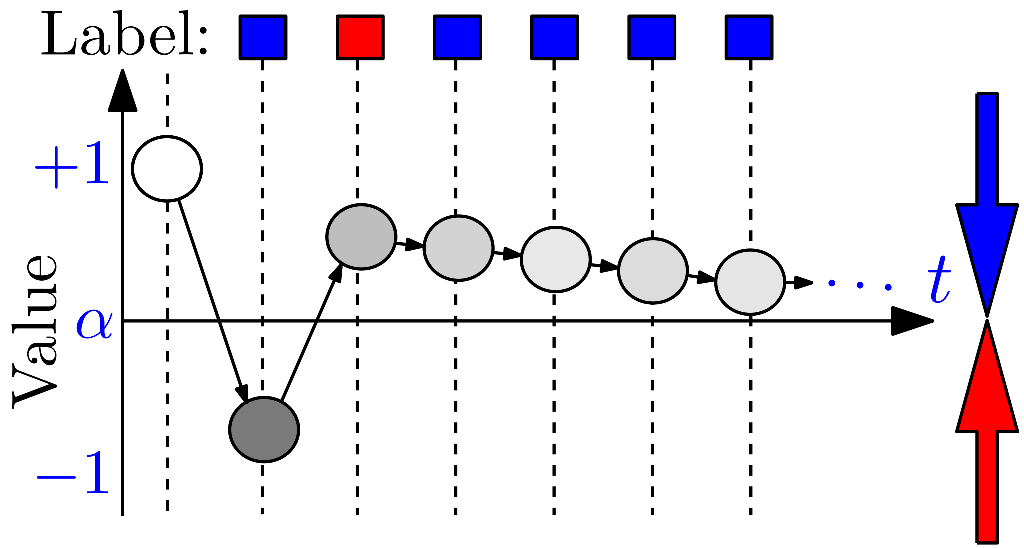


Community Detection via (Parallel) Averaging



Community Detection via (Parallel) Averaging





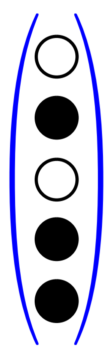
Community Detection via (Parallel) Averaging

[BCNPT '17] **(Informal).** $G = (V_1 \dot{\bigcup} V_2, E)$ s.t.
i) $\chi = \mathbf{1}_{V_1} - \mathbf{1}_{V_2}$ close to right-eigenvector of eigenvalue λ_2 of transition matrix of G, and
ii) gap between λ_2 and λ_3 sufficiently large, then Averaging (approximately) identifies (V_1, V_2).

Community Detection via (Parallel) Averaging

[BCNPT '17] **(Informal).** $G = (V_1 \dot{\bigcup} V_2, E)$ s.t.
i) $\chi = \mathbf{1}_{V_1} - \mathbf{1}_{V_2}$ close to right-eigenvector of eigenvalue λ_2 of transition matrix of G, and
ii) gap between λ_2 and λ_3 sufficiently large, then Averaging (approximately) identifies (V_1, V_2).

Averaging is a **linear** dynamics

$\mathbf{x}^{(t)} =$ 

$$\mathbf{x}^{(t)} = P \cdot \mathbf{x}^{(t-1)} = P^t \cdot \mathbf{x}^{(0)}$$

P transition matrix of lazy random walk

Analysis on Regular SBM

$a = p\frac{n-1}{2}, b = qn \qquad \chi = (1, ..., 1, -1, ..., -1)$

P symmetric $\implies$ orthonormal eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_n$ and real eigenvalues $\lambda_1, ..., \lambda_n$.

Analysis on Regular SBM

$a = p\frac{n-1}{2}, b = qn \qquad \chi = (1, ..., 1, -1, ..., -1)$

P symmetric $\implies$ orthonormal eigenvectors $\mathbf{v}_1, ..., \mathbf{v}_n$ and real eigenvalues $\lambda_1, ..., \lambda_n$.

$$\mathbf{x}^{(t)} = P^t \cdot \mathbf{x}^{(0)} = \sum_i \lambda_i^t (\mathbf{v}_i^\intercal \mathbf{x}^{(0)}) \mathbf{v}_i$$

Analysis on Regular SBM

$a = p\frac{n-1}{2}, b = qn \qquad \chi = (1, \ldots, 1, -1, \ldots, -1)$

P symmetric $\implies$ orthonormal eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and real eigenvalues $\lambda_1, \ldots, \lambda_n$.

$$\mathbf{x}^{(t)} = P^t \cdot \mathbf{x}^{(0)} = \sum_i \lambda_i^t (\mathbf{v}_i^\intercal \mathbf{x}^{(0)}) \mathbf{v}_i$$

$\mathbf{v}_1 = \frac{1}{\sqrt{n}} \mathbb{1} = \frac{1}{\sqrt{n}}(1, \ldots, 1)$ with (largest) eigenvalue 1

Analysis on Regular SBM

$a = p\frac{n-1}{2}, b = qn \qquad \chi = (1, \dots, 1, -1, \dots, -1)$

P symmetric $\implies$ orthonormal eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ and real eigenvalues $\lambda_1, \dots, \lambda_n$.

$$\mathbf{x}^{(t)} = P^t \cdot \mathbf{x}^{(0)} = \sum_i \lambda_i^t (\mathbf{v}_i^\intercal \mathbf{x}^{(0)}) \mathbf{v}_i$$

$\mathbf{v}_1 = \frac{1}{\sqrt{n}} \mathbb{1} = \frac{1}{\sqrt{n}}(1, \dots, 1)$ with (largest) eigenvalue 1

Regular SBM $\implies P \frac{1}{\sqrt{n}} \chi = (\frac{a-b}{a+b}) \cdot \frac{1}{\sqrt{n}} \chi$

$$\frac{1}{a+b} \left(\begin{array}{c|c} \cdots\cdots\cdots & \cdots\cdots\cdots \\ \cdots a \text{ ``1''s} \cdots & \cdots b \text{ ``1''s} \cdots \\ \cdots\cdots\cdots & \cdots\cdots\cdots \\ \hline \cdots\cdots\cdots & \cdots\cdots\cdots \\ \cdots b \text{ ``1''s} \cdots & \cdots a \text{ ``1''s} \cdots \\ \cdots\cdots\cdots & \cdots\cdots\cdots \end{array} \right) \cdot \begin{pmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{pmatrix} = \frac{a-b}{a+b} \begin{pmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{pmatrix}$$

Analysis on Regular SBM

$a = p\frac{n-1}{2}, b = qn \qquad \chi = (1, \ldots, 1, -1, \ldots, -1)$

P symmetric $\implies$ orthonormal eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ and real eigenvalues $\lambda_1, \ldots, \lambda_n$.

$\mathbf{x}^{(t)} = P^t \cdot \mathbf{x}^{(0)} = \sum_i \lambda_i^t (\mathbf{v}_i^\intercal \mathbf{x}^{(0)}) \mathbf{v}_i$

$\mathbf{v}_1 = \frac{1}{\sqrt{n}} \mathbb{1} = \frac{1}{\sqrt{n}}(1, \ldots, 1)$ with (largest) eigenvalue 1

Regular SBM $\implies P \frac{1}{\sqrt{n}} \chi = (\frac{a-b}{a+b}) \cdot \frac{1}{\sqrt{n}} \chi$

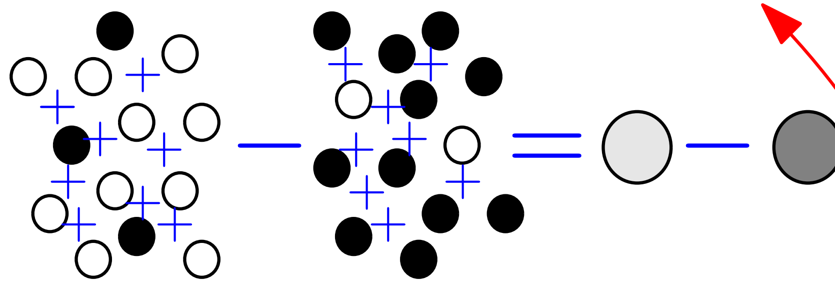
W.h.p. $\lambda_3(1+\delta) < \frac{a-b}{a+b} = \lambda_2$, then

$$\mathbf{x}^{(t)} = \frac{1}{n}(\mathbf{1}^\intercal \mathbf{x}^{(0)})\mathbf{1} + \left(\frac{a-b}{a+b}\right)^t \frac{1}{n}(\chi^\intercal \mathbf{x}^{(0)})\chi + \mathbf{e}^{(t)}$$

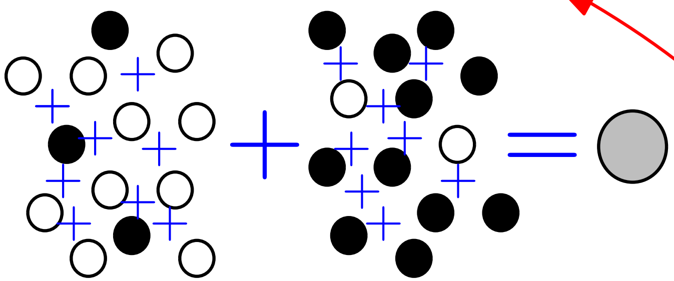
with $\|\mathbf{e}^{(t)}\| \leq \lambda_3^t \sqrt{n}$

Analysis on Regular SBM

$\frac{1}{n}\sum_{u\in V_1} \mathbf{x}^{(0)}(u) - \frac{1}{n}\sum_{u\in V_2} \mathbf{x}^{(0)}(u)$



$\frac{1}{n}\sum_{u\in V} \mathbf{x}^{(0)}(u)$



W.h.p. $\lambda_3(1+\delta) < \frac{a-b}{a+b} = \lambda_2$, then

$$\mathbf{x}^{(t)} = \frac{1}{n}(\mathbf{1}^\intercal \mathbf{x}^{(0)})\mathbf{1} + \left(\frac{a-b}{a+b}\right)^t \frac{1}{n}(\chi^\intercal \mathbf{x}^{(0)})\chi + \mathbf{e}^{(t)}$$

with $\|\mathbf{e}^{(t)}\| \leq \lambda_3^t \sqrt{n}$

Analysis on Regular SBM

$$\mathbf{x}^{(t)} = \frac{1}{n}(\mathbf{1}^\intercal \mathbf{x}^{(0)})\mathbf{1} + \underbrace{\left(\frac{a-b}{a+b}\right)}_{=\lambda_2}{}^{t} \frac{1}{n}(\chi^\intercal \mathbf{x}^{(0)})\chi + \mathbf{e}^{(t)}$$

Analysis on Regular SBM

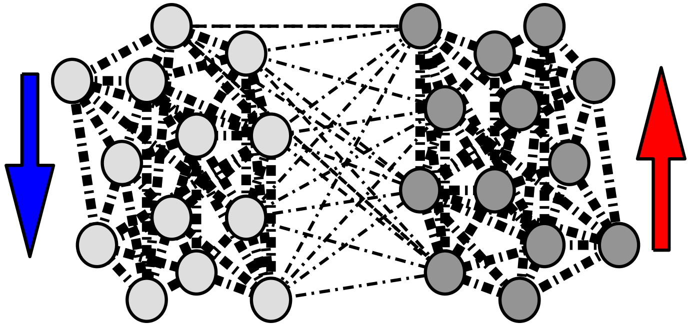
$$\mathbf{x}^{(t)} = \frac{1}{n}(\mathbf{1}^\intercal \mathbf{x}^{(0)})\mathbf{1} + \underbrace{\left(\frac{a-b}{a+b}\right)}_{=\lambda_2}{}^{t} \frac{1}{n}(\chi^\intercal \mathbf{x}^{(0)})\chi + \mathbf{e}^{(t)}$$

$$\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)} = (\chi^\intercal \mathbf{x}^{(0)})\lambda_2^{t-1}(\lambda_2 - 1)\chi + \underbrace{\mathbf{e}^{(t)} - \mathbf{e}^{(t-1)}}_{o(\lambda_2^t) \text{ if } t = \Omega(\log n)}$$

Analysis on Regular SBM

$$\mathbf{x}^{(t)} = \frac{1}{n}(\mathbf{1}^\intercal \mathbf{x}^{(0)})\mathbf{1} + \underbrace{\left(\frac{a-b}{a+b}\right)}_{=\lambda_2}{}^{t}\frac{1}{n}(\chi^\intercal \mathbf{x}^{(0)})\chi + \mathbf{e}^{(t)}$$

$$\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)} = (\chi^\intercal \mathbf{x}^{(0)})\lambda_2^{t-1}(\lambda_2 - 1)\chi + \underbrace{\mathbf{e}^{(t)} - \mathbf{e}^{(t-1)}}_{o(\lambda_2^t) \text{ if } t=\Omega(\log n)}$$

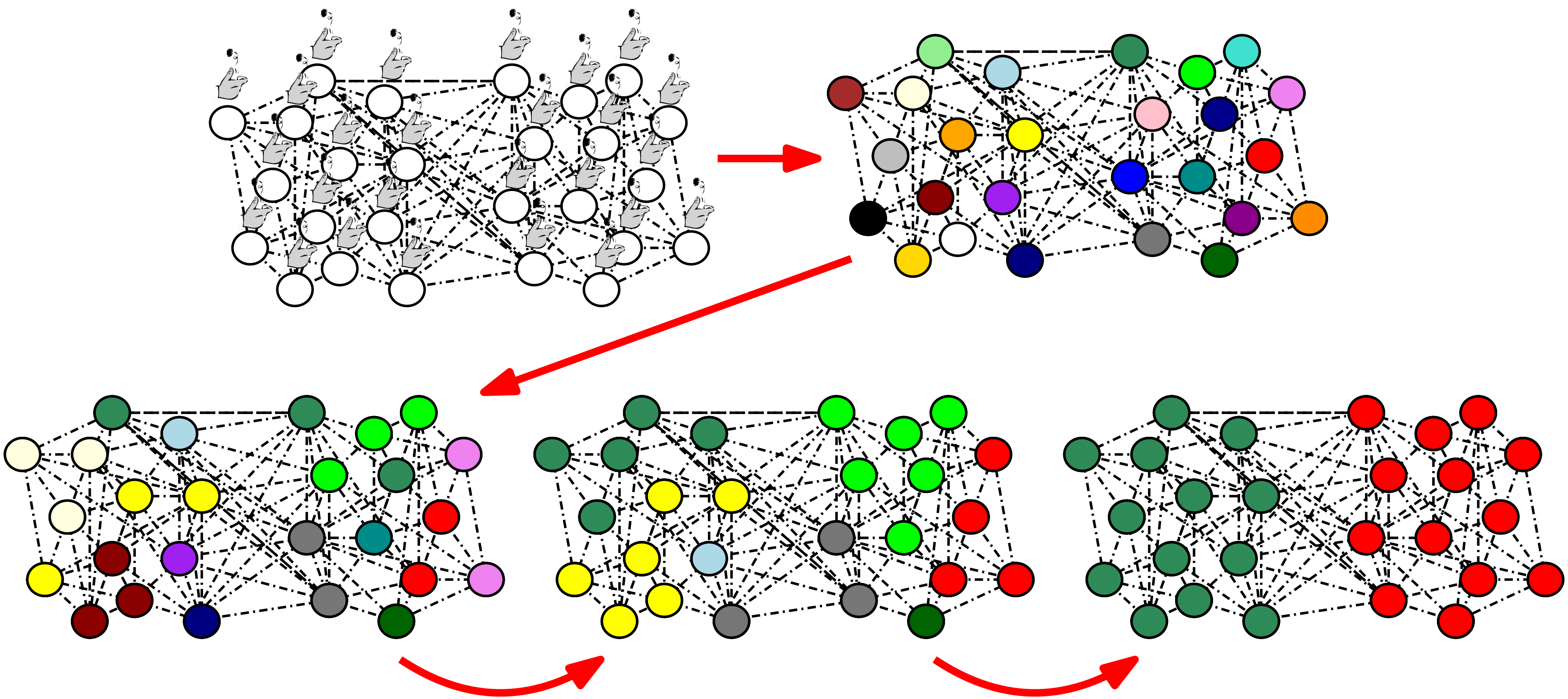


$$\text{sign}(\mathbf{x}^{(t)}(u) - \mathbf{x}^{(t-1)}(u)) \propto \text{sign}(\chi(u))$$

Open Problem: Analyzing LPAs

Averagins is a "linearization" of Label Propagation Algorithms:

- Each node initially sample a random color, then
- at each round, each node switch to the majority label of a sample of neighbors.



Thank you!