Friend or Foe? Population Protocols can perform Community Detection

Emanuele Natale[◇]

joint work with

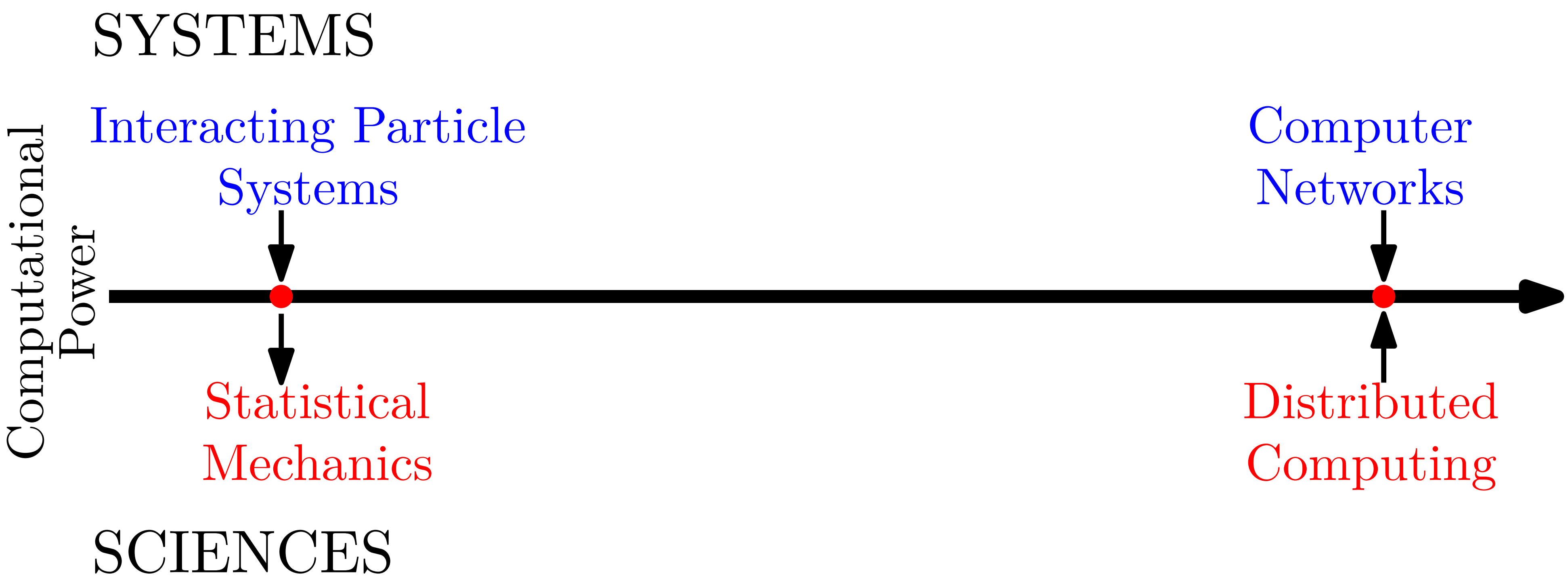
Luca Becchetti[†], Andrea Clementi[⋆], Francesco Pasquale[⋆], Prasad Raghavendra[∗] and Luca Trevisan[∗]

[†] SAPIENZA UNIVERSITÀ DI ROMA [⋆] UNIVERSITA' DEGLI STUDI DI ROMA TorVergata [∗] Berkeley UNIVERSITY OF CALIFORNIA [◇] max planck institut informatik

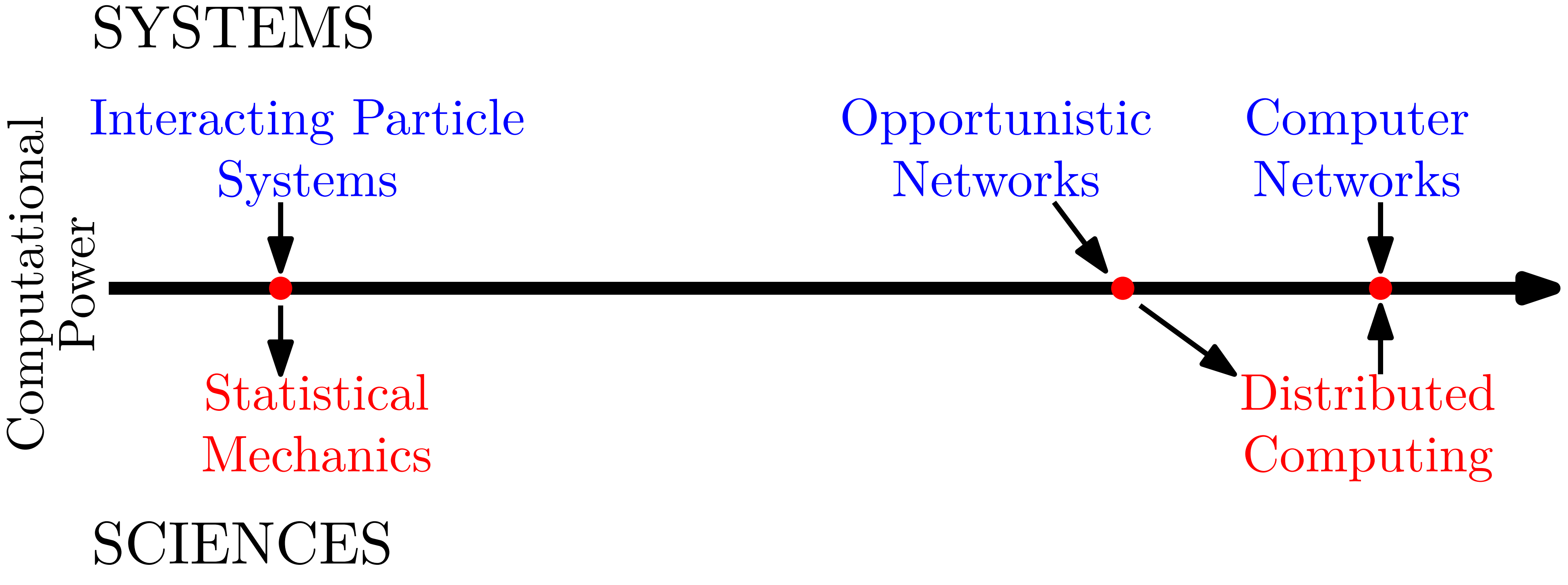
INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

IRIF Algorithms and Complexity seminar
21 March 2017, Paris

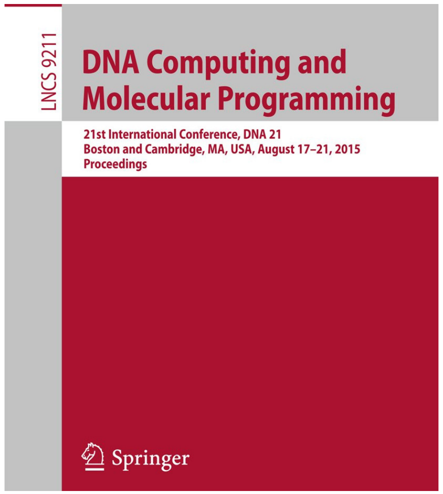
Communication in *Simple* Systems

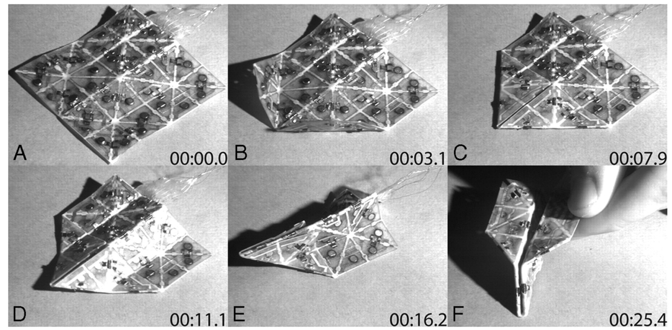


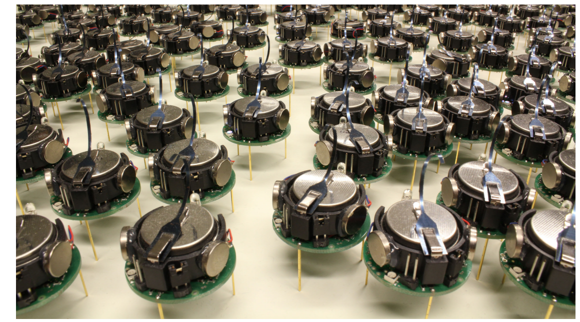
Communication in *Simple* Systems



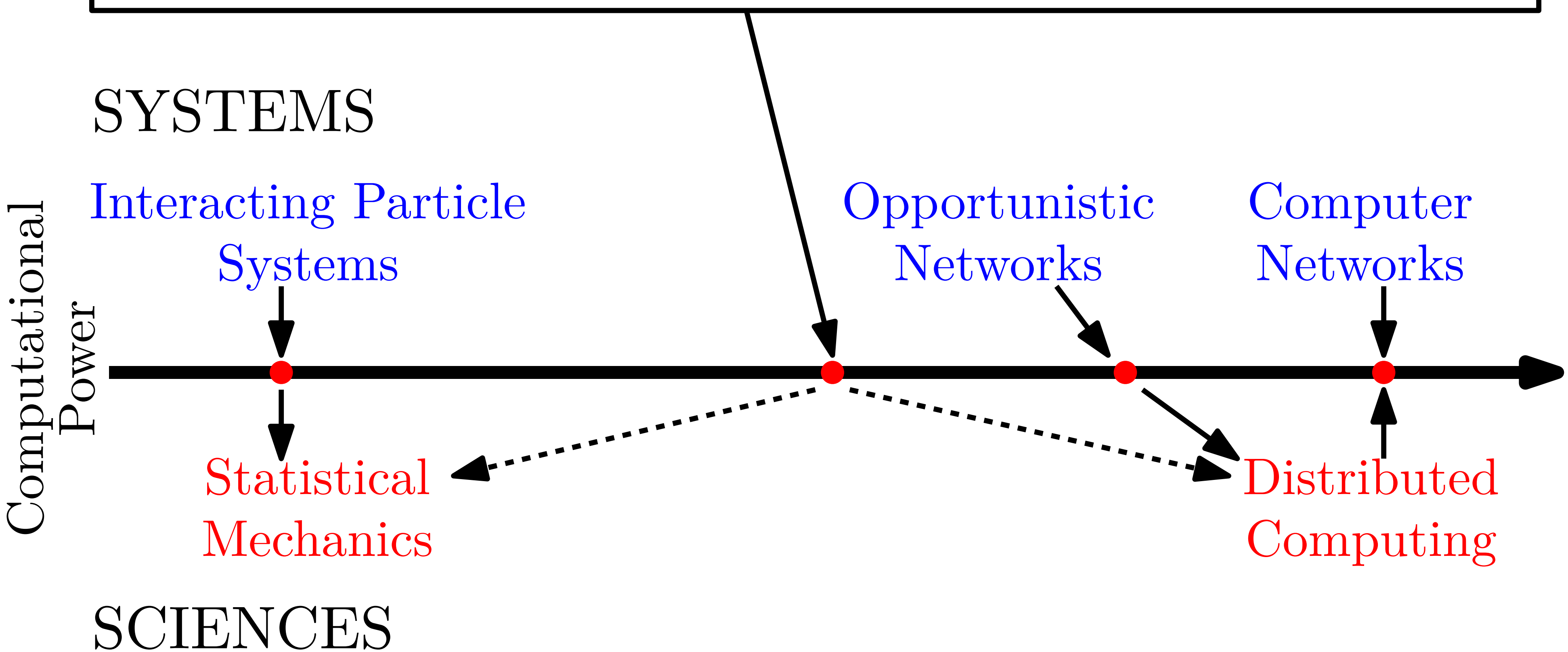
Communication in *Simple* Systems



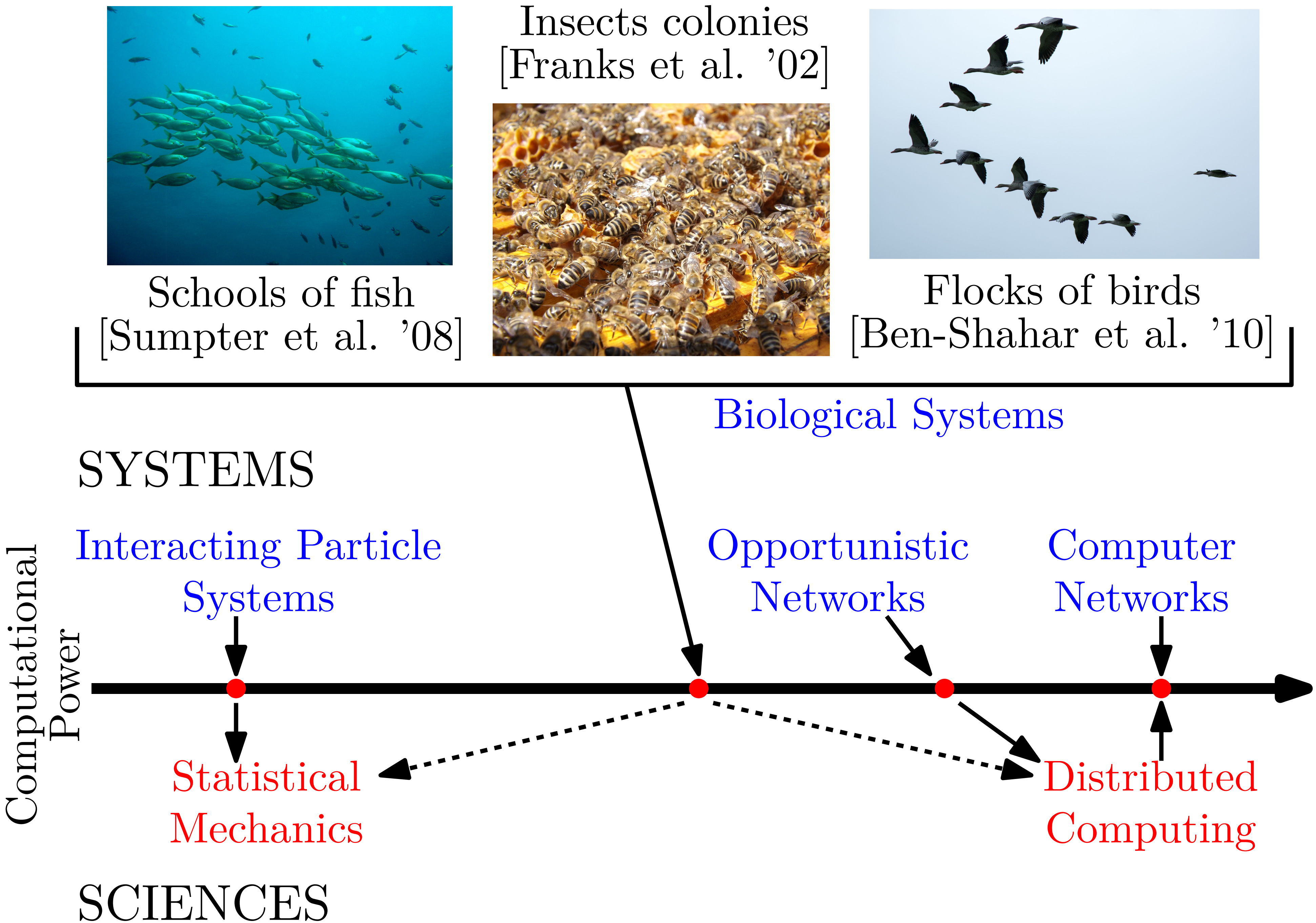




DNA/Molecular Computing, Programmable Matter, Swarms of Simple Robots



Communication in *Simple* Systems



Dynamics

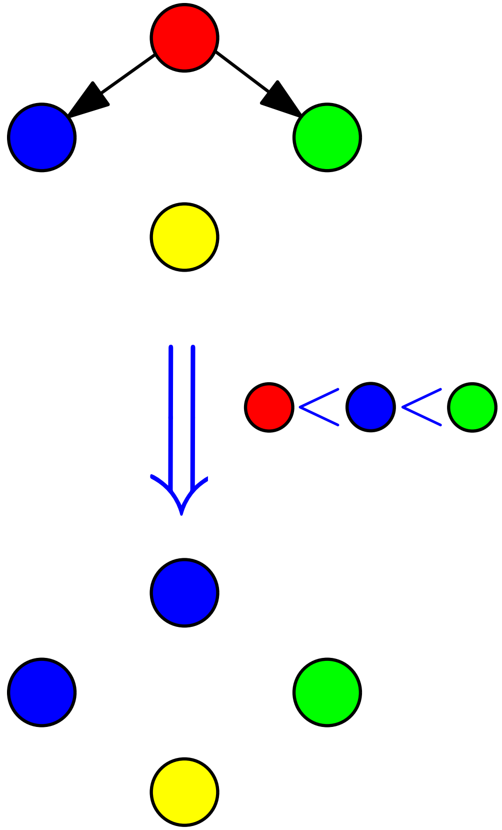
(informal) *Very simple* distributed algorithms: For every graph, agent and round, states are updated according to fixed rule of current state and symmetric function of states of neighbors.

Dynamics

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

Examples of Dynamics

- 3-Median dynamics

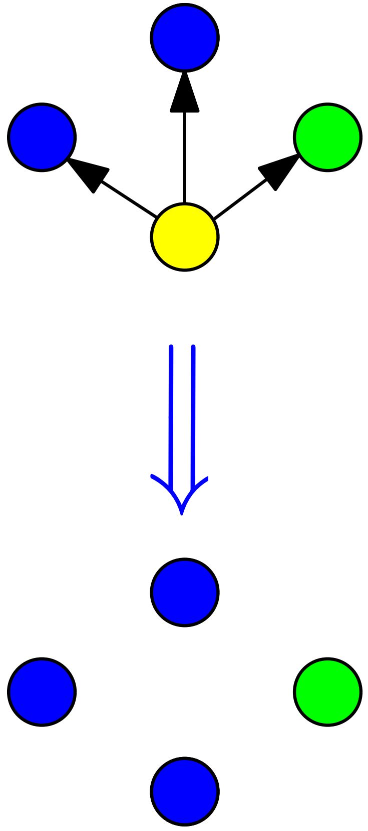


Dynamics

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

Examples of Dynamics

- 3-Median dynamics
- 3-Majority dynamics

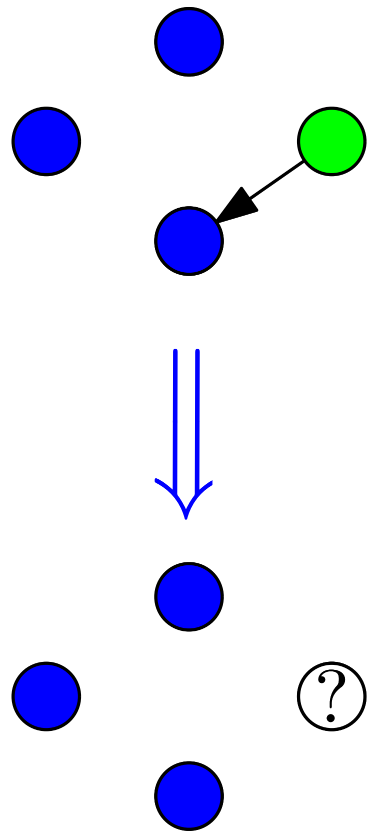


Dynamics

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

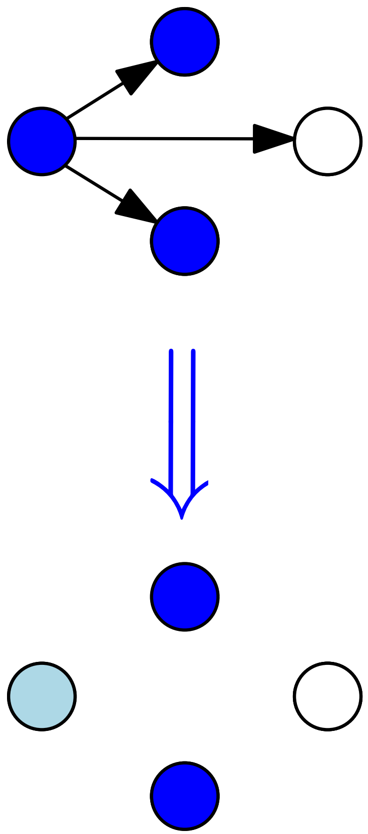
Examples of Dynamics

- 3-Median 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 rule of current state and symmetric function of states of neighbors.

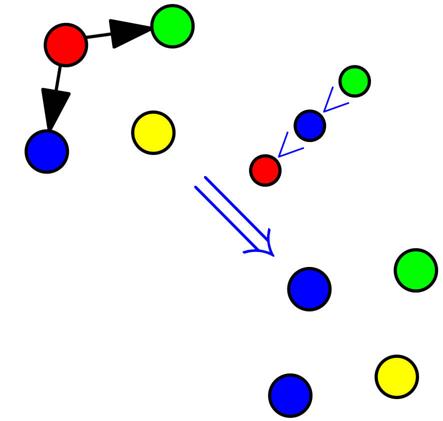


Examples of Dynamics

- 3-Median dynamics
- 3-Majority dynamics
- Undecided-state dynamics
- Averaging dynamics

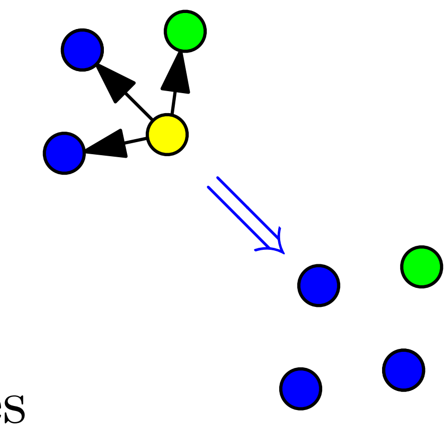
The Power of Dynamics: Plurality Consensus

Computing the Median

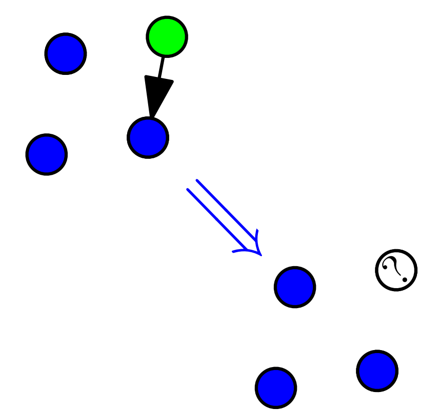


- 3-Median dynamics [Doerr et al. '11]. Converge to $\mathcal{O}(\sqrt{n \log n})$ approximation of median of system in $\mathcal{O}(\log n)$ rounds w.h.p., even if $\mathcal{O}(\sqrt{n})$ states are arbitrarily changed at each round ($\mathcal{O}(\sqrt{n})$-bounded adversary).

Computing the Majority



- 3-Majority dynamics [SPAA '14, SODA '16]. 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 poly(k). h-majority converges in $\Omega(k/h^2)$.



- Undecided-State dynamics [SODA '15]. If majority/second-majority ($c_{maj}/c_{2^{nd}maj}$) is at least $1+\epsilon$, system converges to plurality within $\tilde{\Theta}(\sum_{i=1}^{k} \left(c_i^{(0)}/c_{maj}^{(0)}\right)^2)$ rounds w.h.p.

The Median, the Mode and... the Mean

Dynamics can solve Consensus, Median, Majority, in robust and fault tolerant ways, but this is trivial in centralized setting.

The Median, the Mode and... the Mean

Dynamics can solve Consensus, Median, Majority, in robust and fault tolerant ways, but this is trivial in centralized setting.

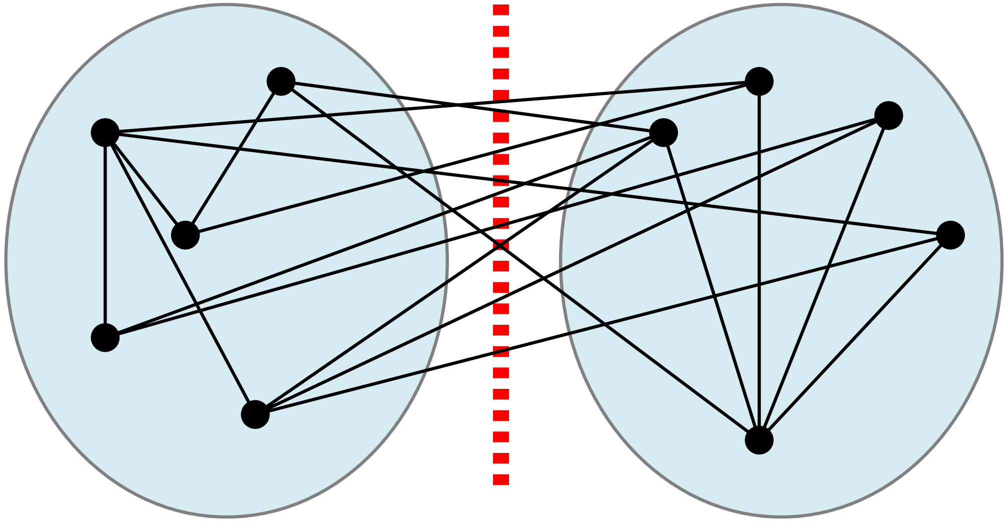
Can dynamics solve a problem non-trivial in centralized setting?

Community Detection as Minimum Bisection

Minimum Bisection Problem.

Input: a graph G with $2n$ nodes.

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

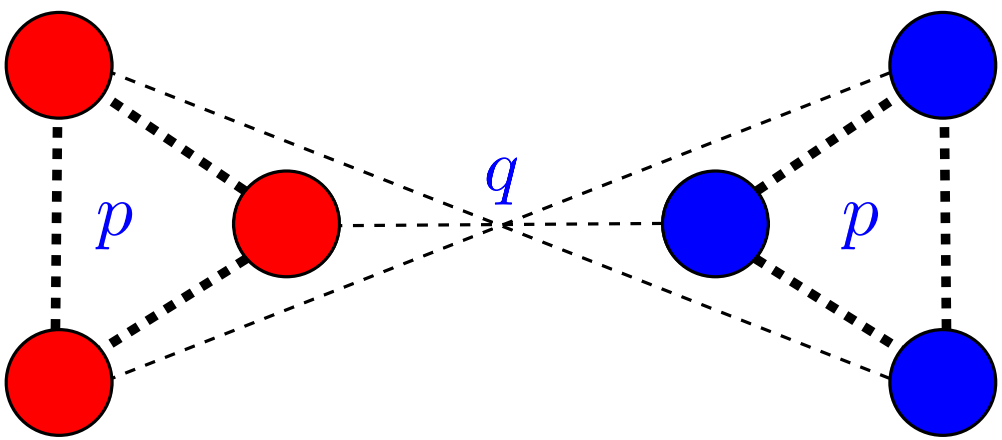


[Garey, Johnson, Stockmeyer '76]:

Min-Bisection is *NP-Complete*.

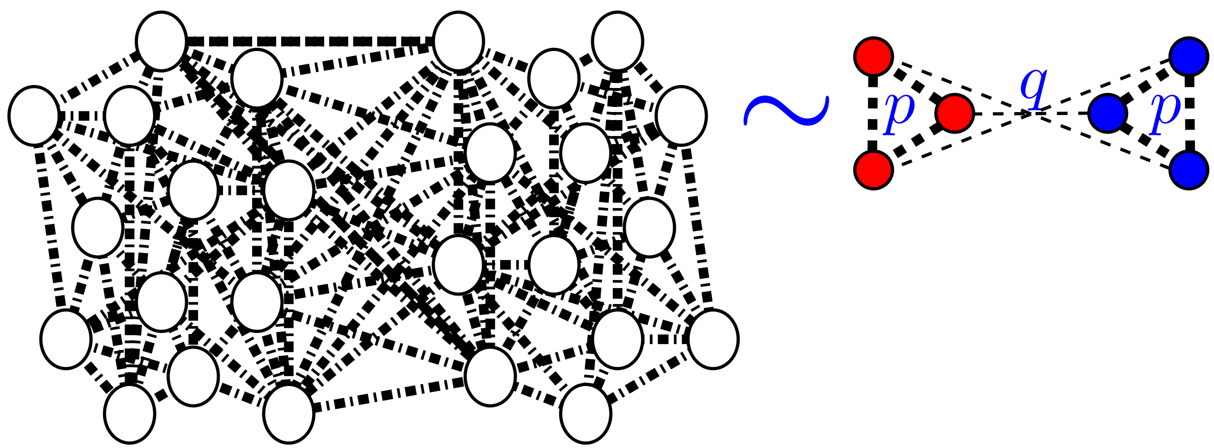
The Stochastic Block Model

Stochastic Block Model (SBM). Two "communities" of equal size V_1 and V_2, each edge inside a community included with probability $p = \frac{a}{n}$, each edge across communities included with probability $q = \frac{b}{n} < p$.



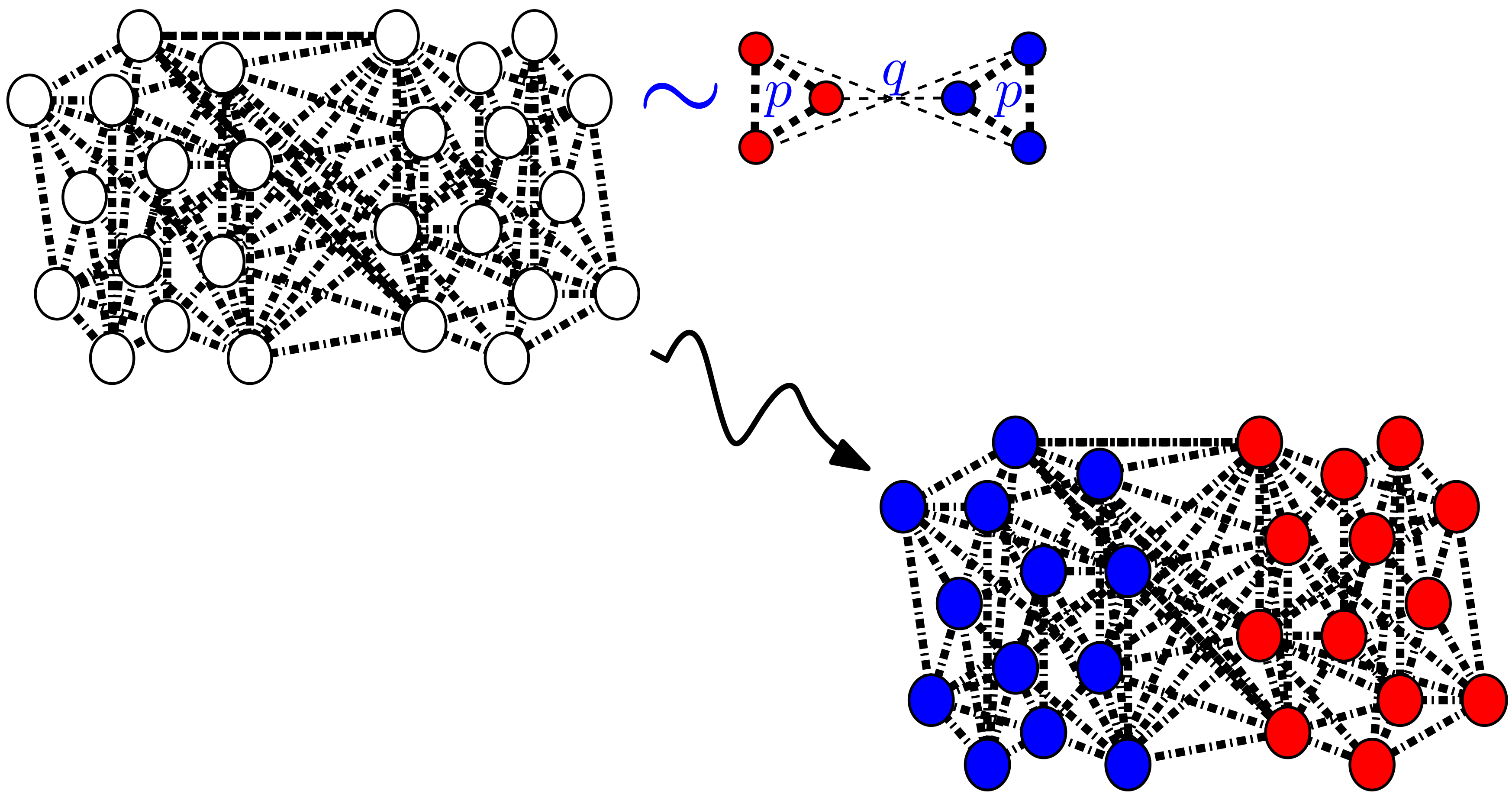
The Stochastic Block Model

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



The Stochastic Block Model

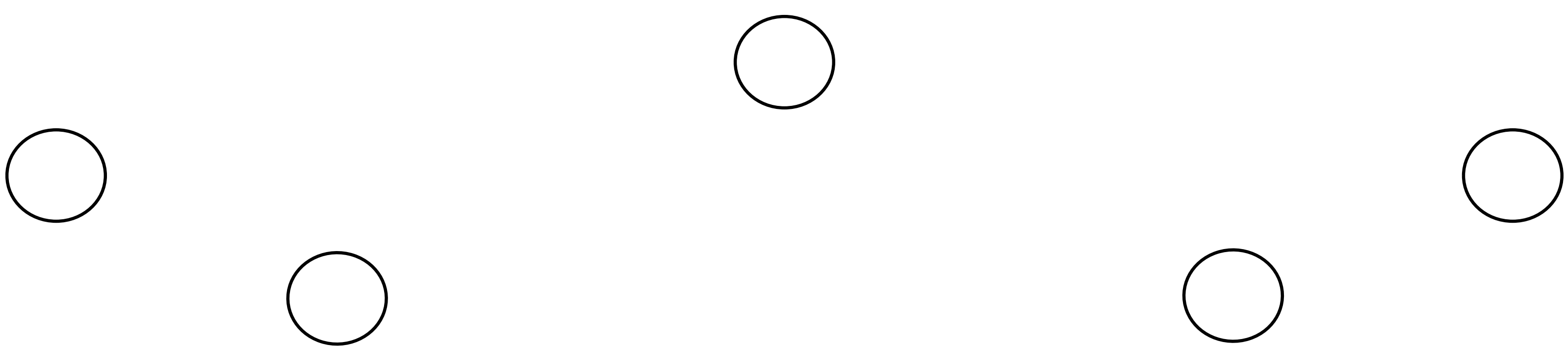
Reconstruction problem. Given graph generated by SBM, find original partition.



The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

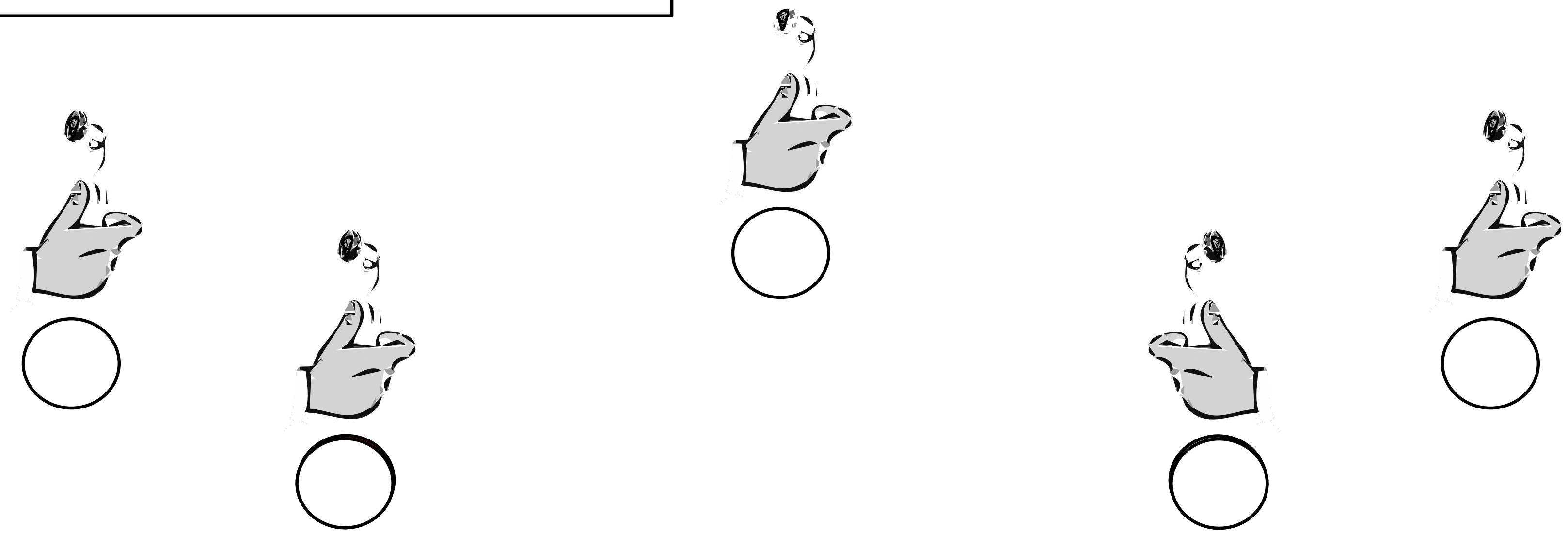
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

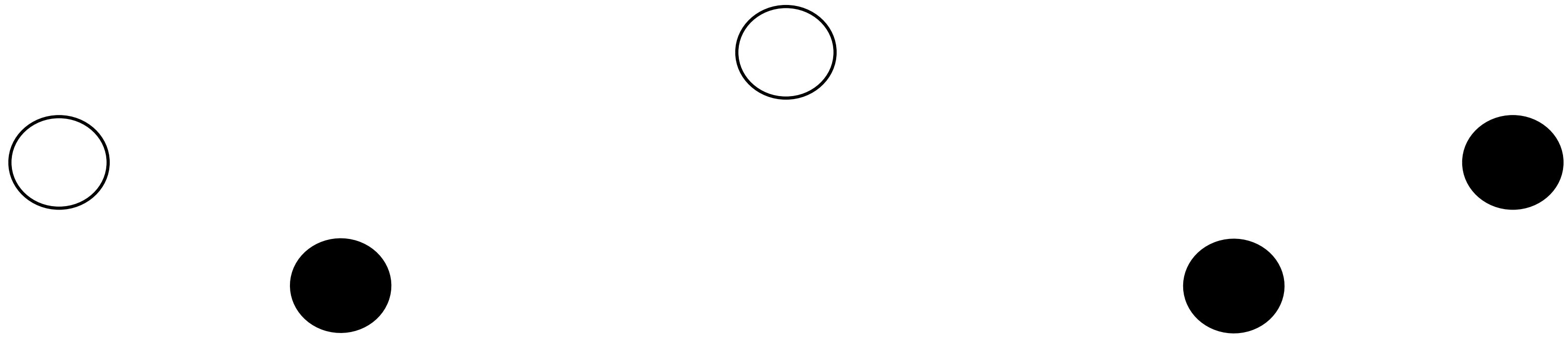
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

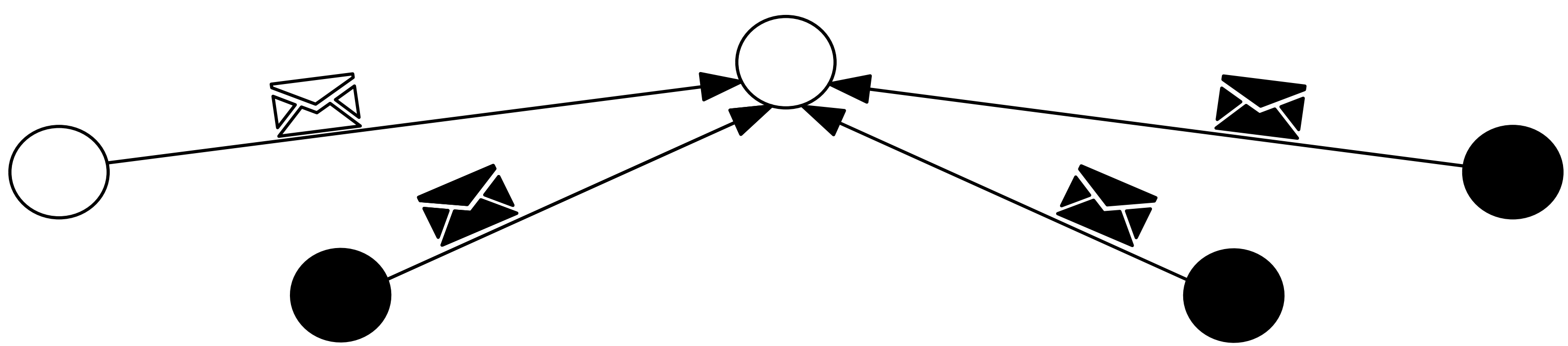
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

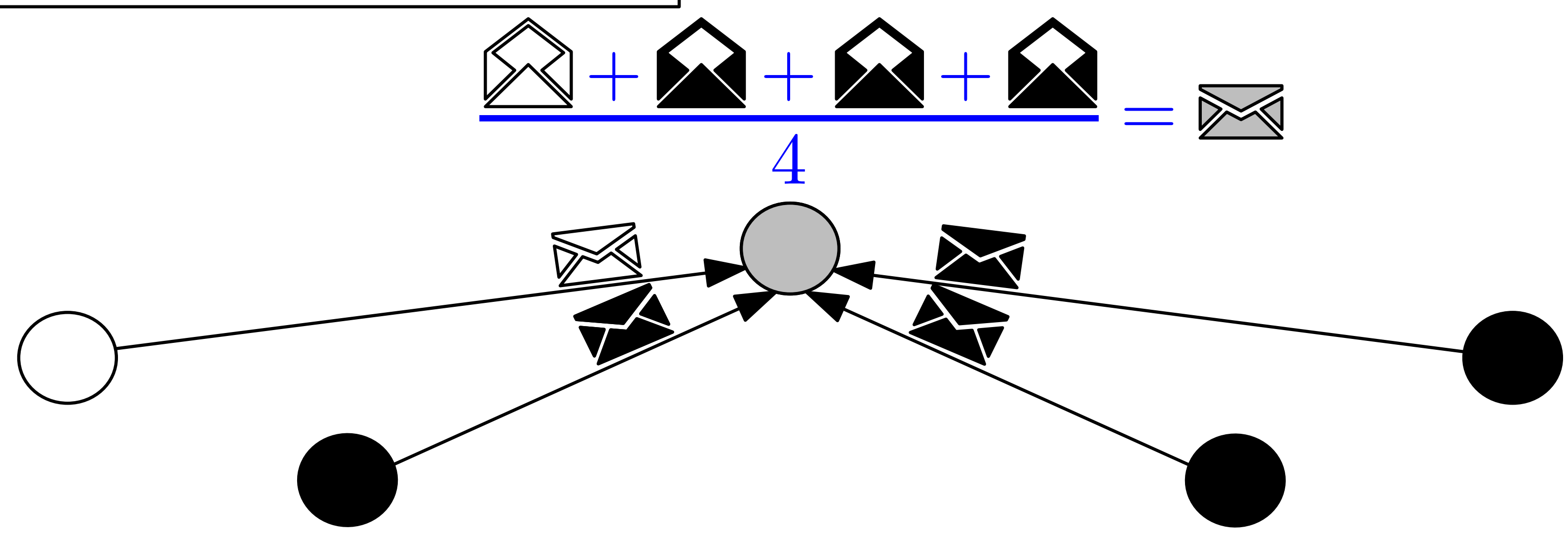
- At $t = 0$, randomly pick value $x^{(t)} \in \{+\mathbf{1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

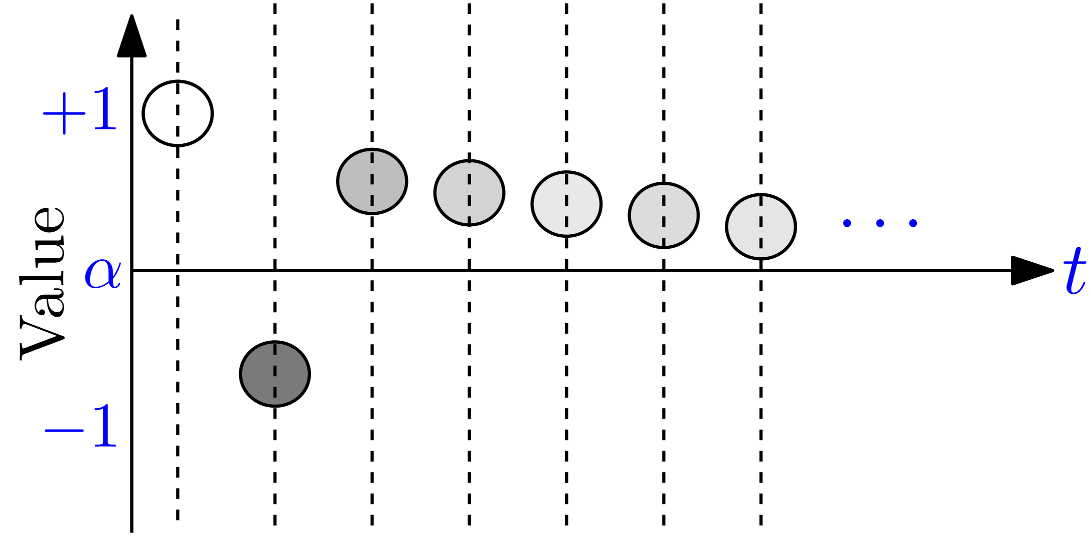
- At $t = 0$, randomly pick value $x^{(t)} \in \{+1, -1\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

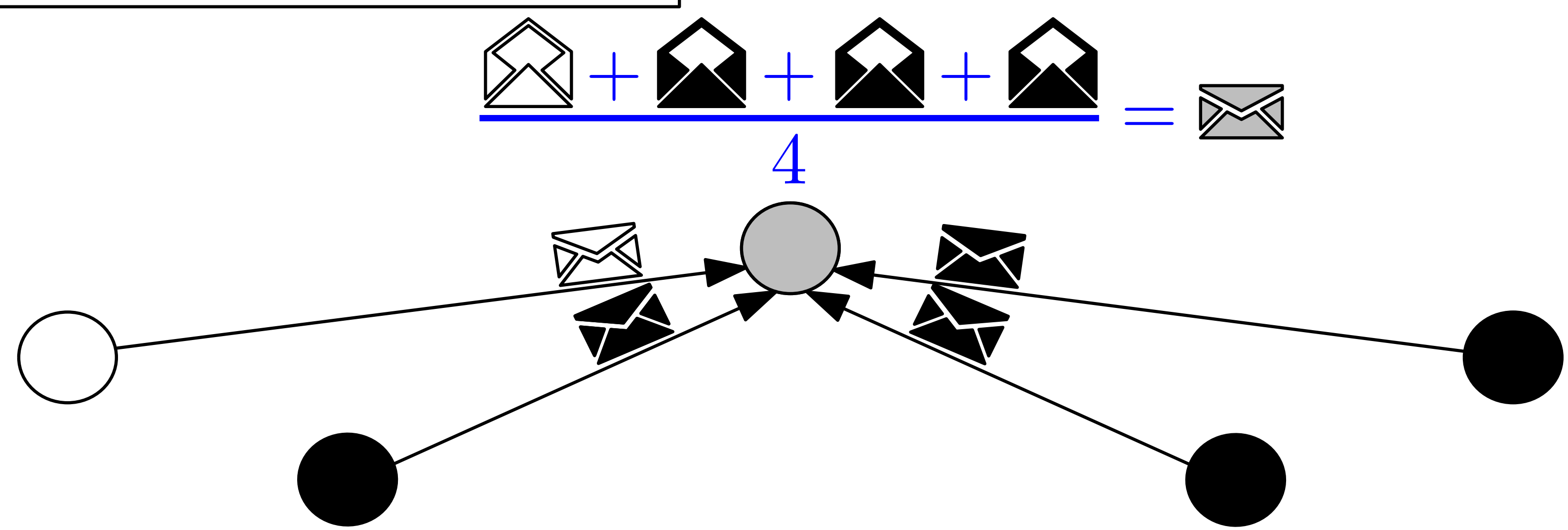


The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

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 average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

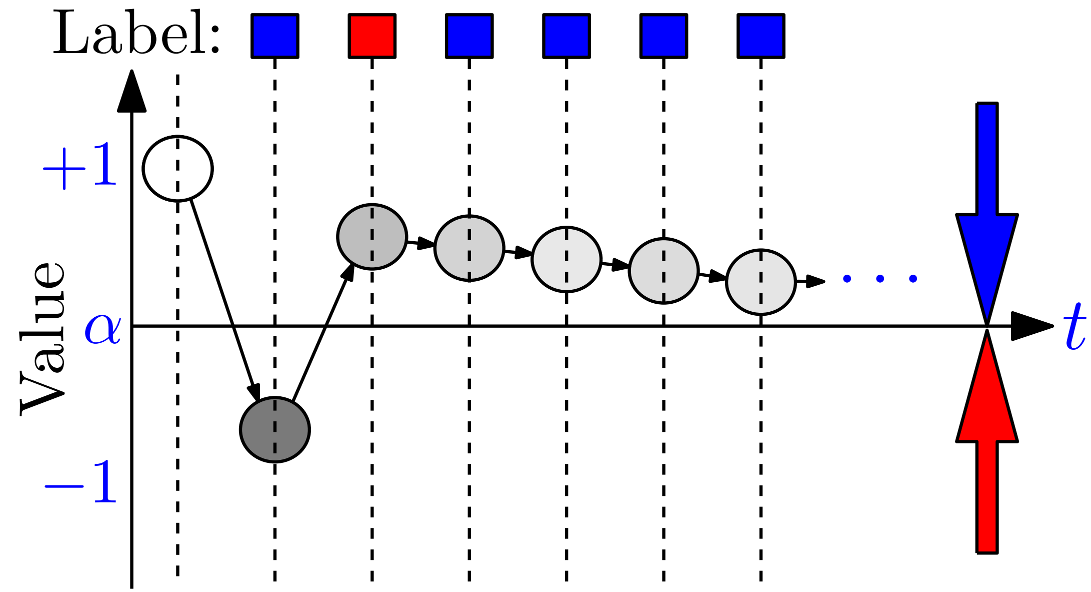


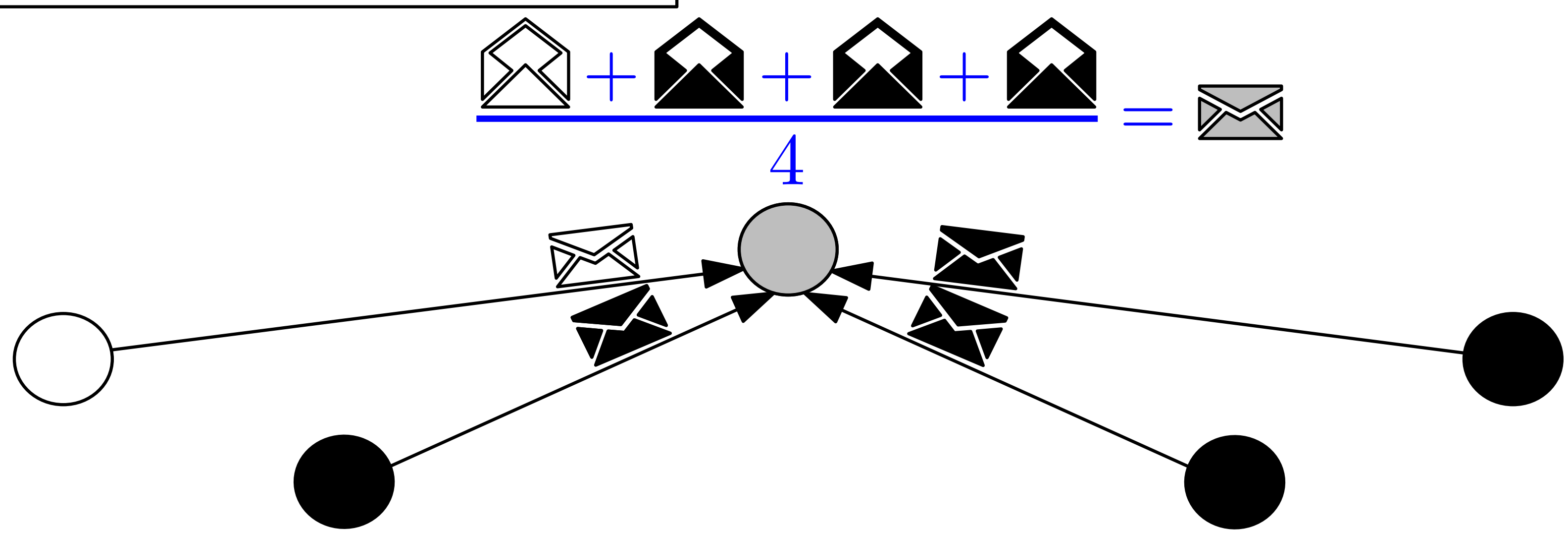


The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

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 average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.





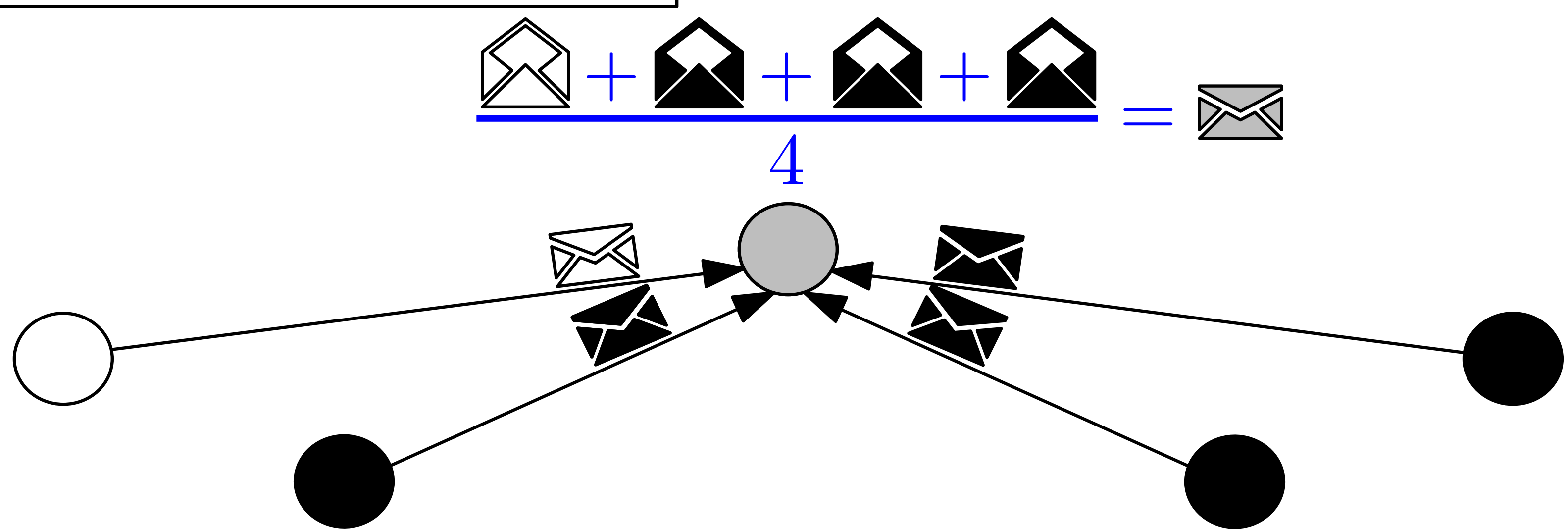
The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

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 average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

Well studied process [Shah '09]:

- Converges to (weighted) global average of initial values,
- Convergence time $=$ mixing time of G,
- Important applications in fault-tolerant self-stabilizing consensus.



The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

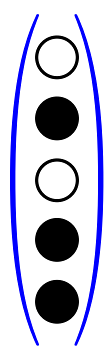
Al nodes at the same time:

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

Well studied process [Shah '09]:

- Converges to (weighted) global average of initial values,
- Convergence time = mixing time of G,
- Important applications in fault-tolerant self-stabilizing consensus.

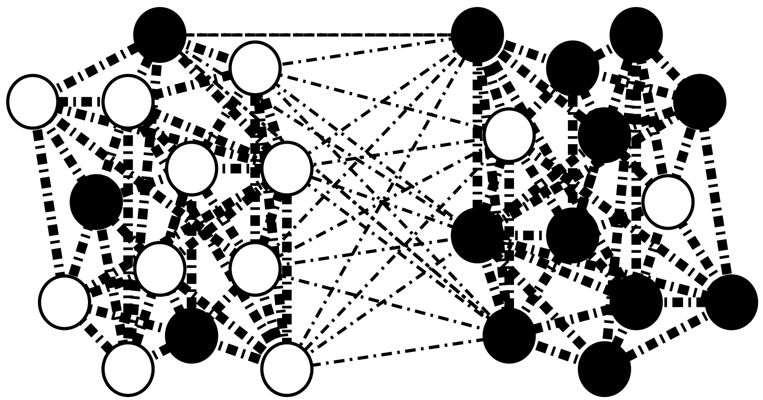
Averaging is a **linear** dynamics

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

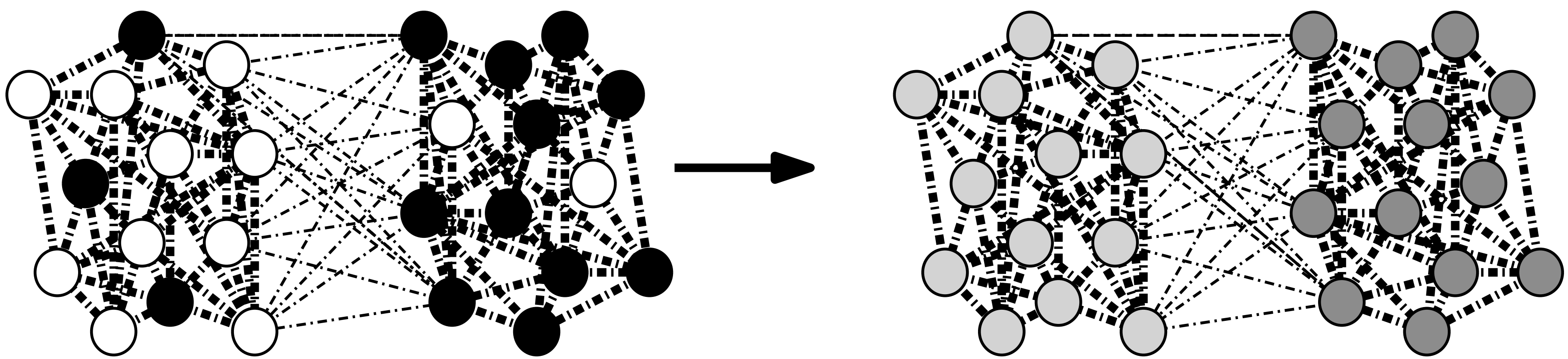
$$\mathbf{x}^{(t)} = P \cdot \mathbf{x}^{(t-1)} = P^t \cdot \mathbf{x}^{(0)}$$

P transition matrix of random walk

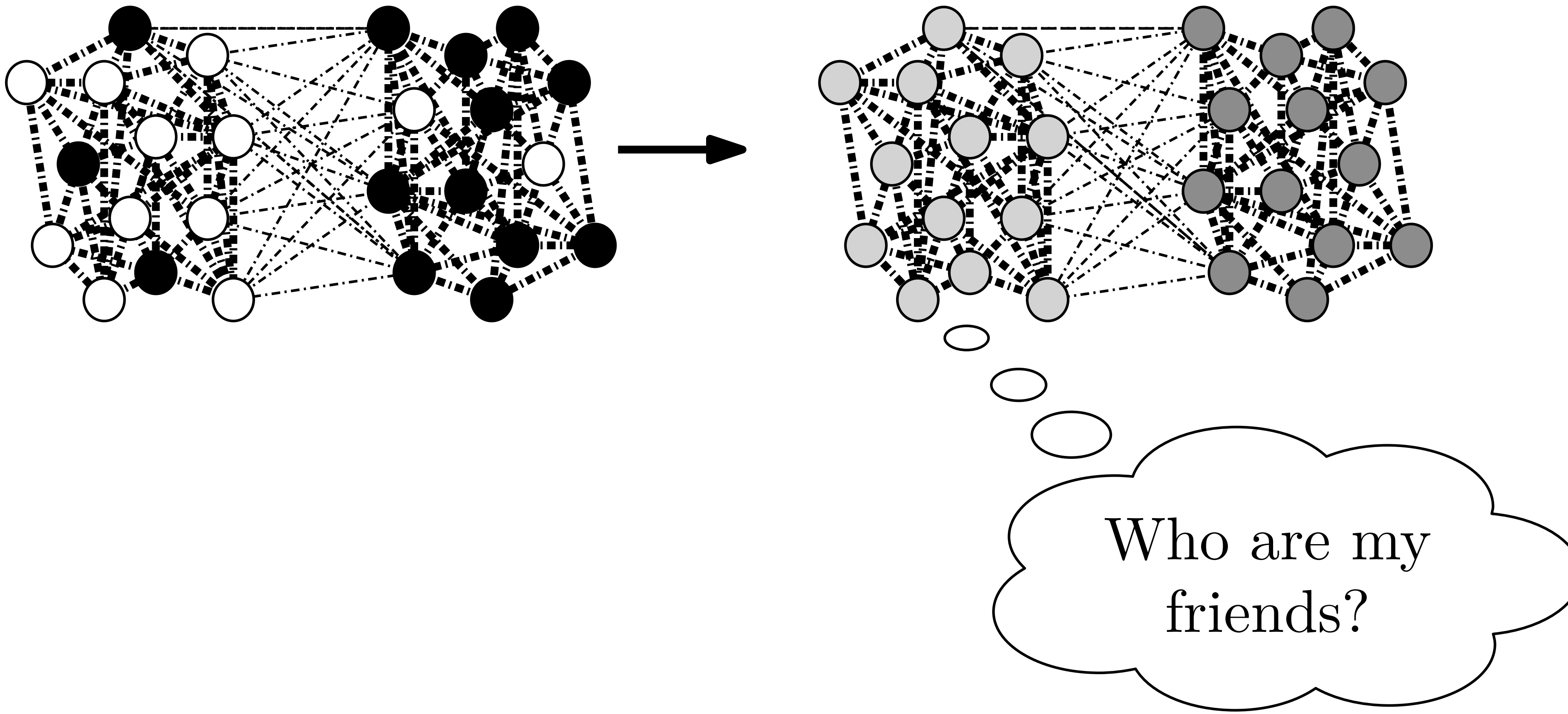
Community Detection via Averaging Dynamics



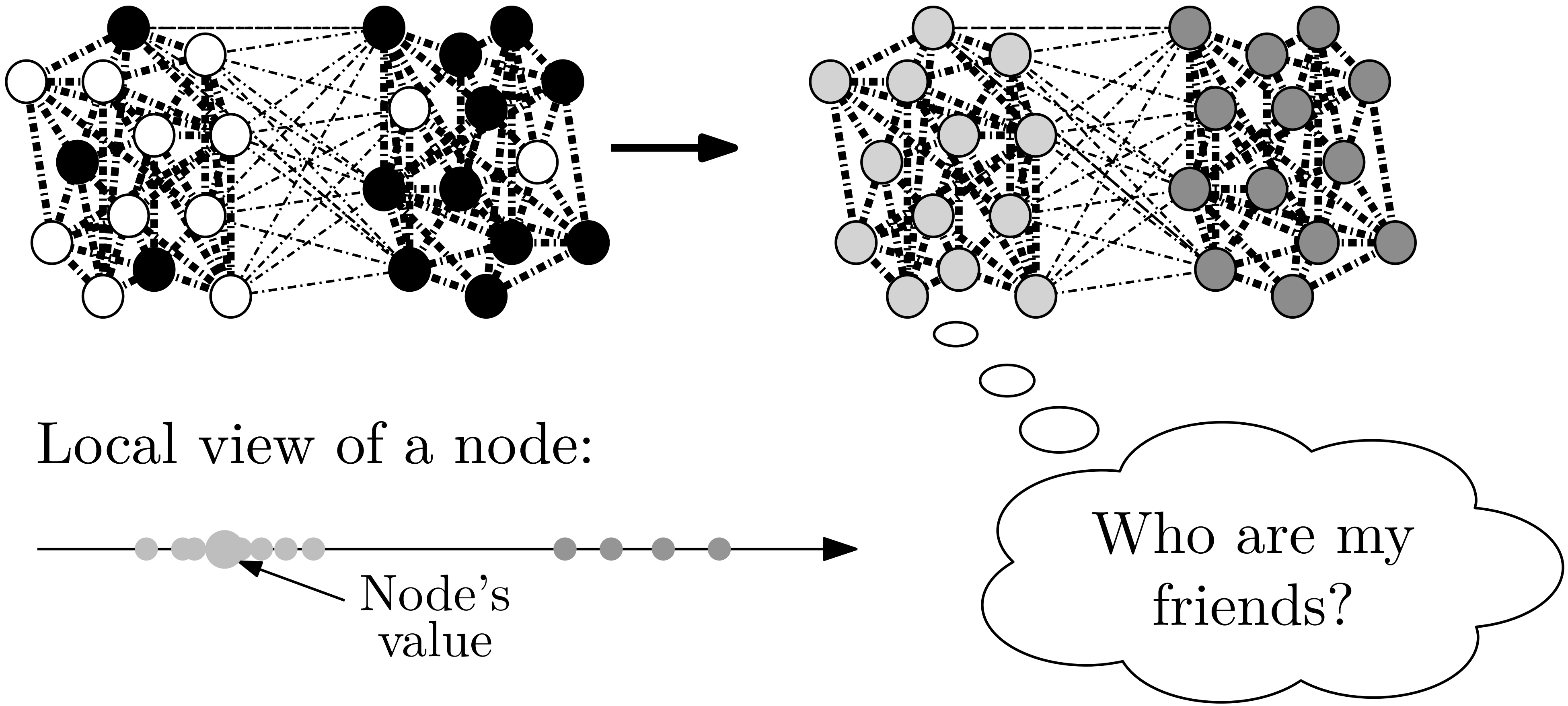
Community Detection via Averaging Dynamics



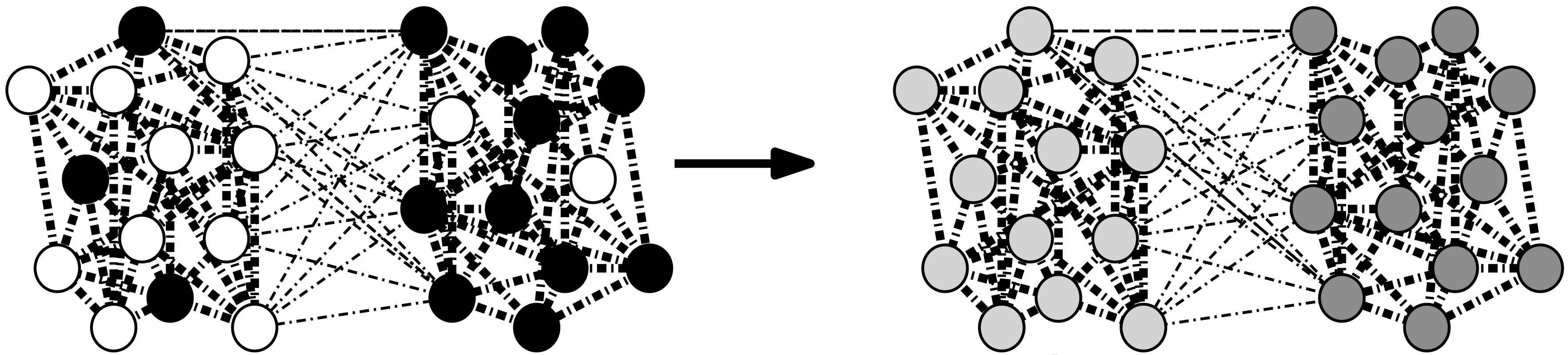
Community Detection via Averaging Dynamics



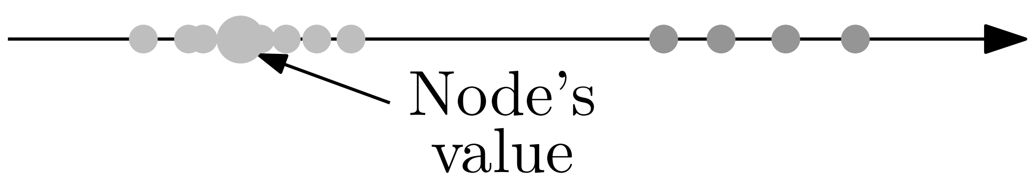
Community Detection via Averaging Dynamics



Community Detection via Averaging Dynamics



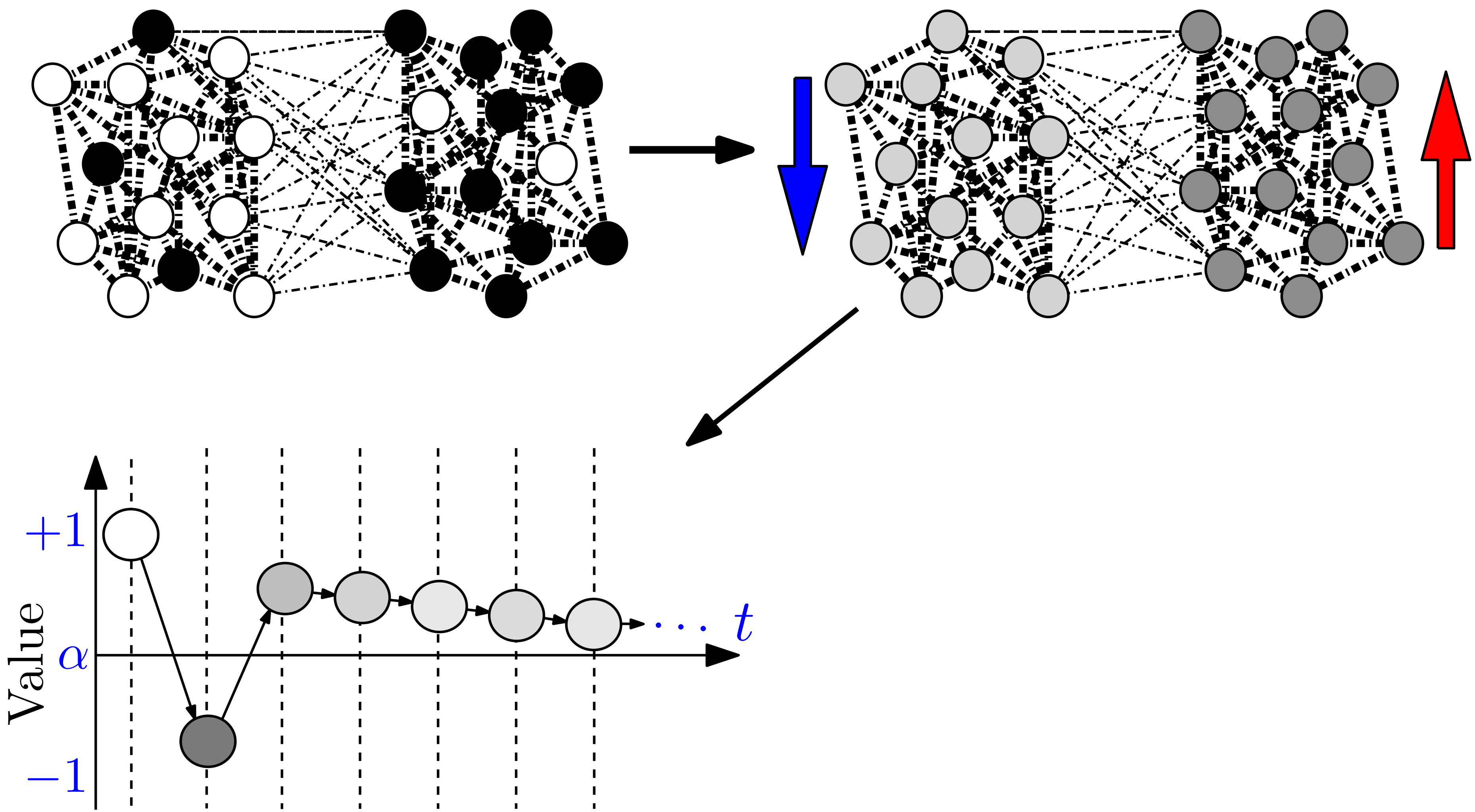
Local view of a node:



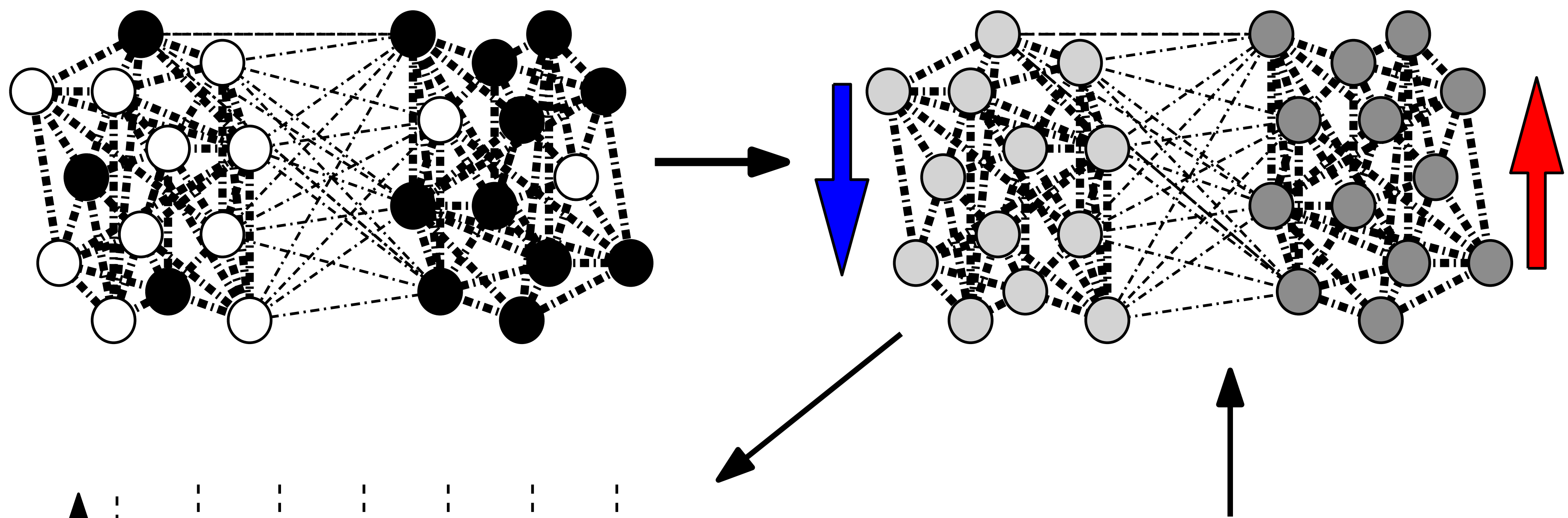
Irregular case:

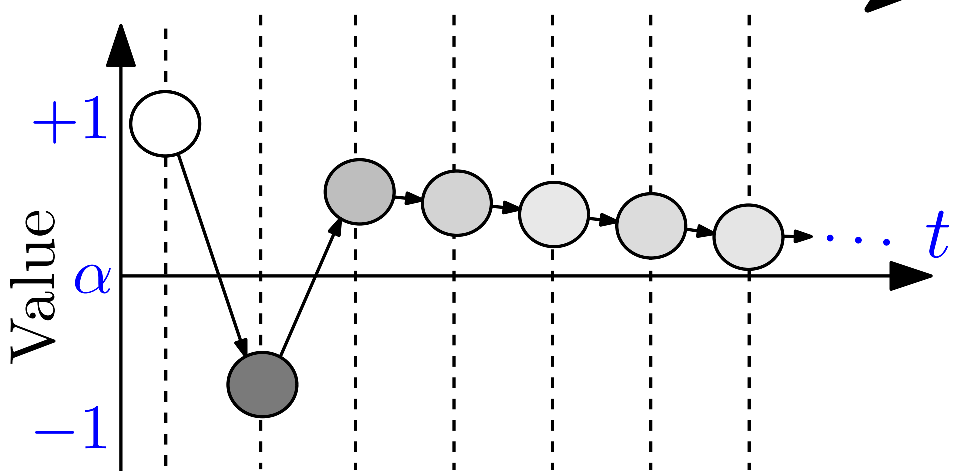
- *outliers*?
- *no neighbors* in the other community?

Community Detection via Averaging Dynamics



Community Detection via Averaging Dynamics





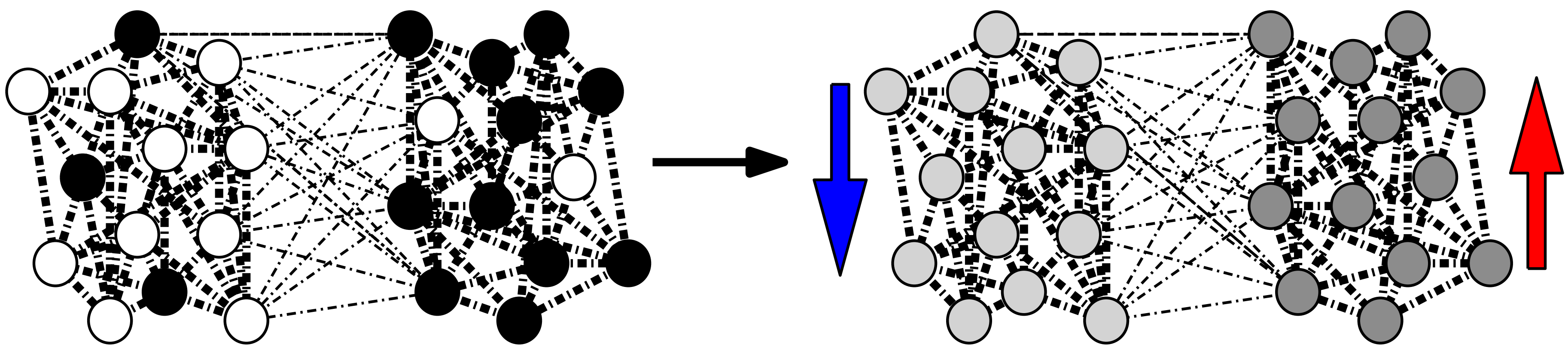
$v_1, \ldots, v_n$ eigenvectors of random walk matrix P:

$$v_1 = \mathbb{1} = (1, \ldots, 1)$$

$$v_2 \approx \chi = (1, \ldots, 1, -1, \ldots, -1)$$

"nice" graph

Community Detection via Averaging Dynamics

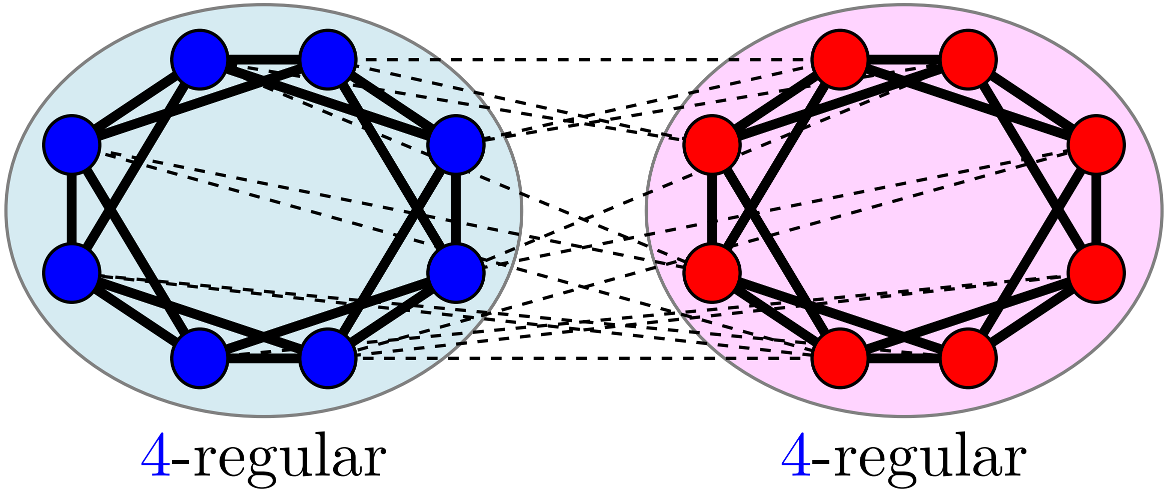


[SODA '17](Informal). $G = (V_1 \dot{\bigcup} V_2, E)$ s.t.
i) $\chi = \mathbf{1}_{V_1} - \mathbf{1}_{V_2}$ close to right-eigenvector of eigenvalue λ_2 of transition matrix of G, and
ii) gap between λ_2 and $\lambda = \max\{\lambda_3, |\lambda_n|\}$ sufficiently large,
then Averaging (approximately) identifies (V_1, V_2).

Toy Case: Regular Stochastic Block Model

Regular SBM (RSBM) [Brito et al. SODA'16]. A graph $G = (V_1 \dot{\bigcup} V_2, E)$ s.t.

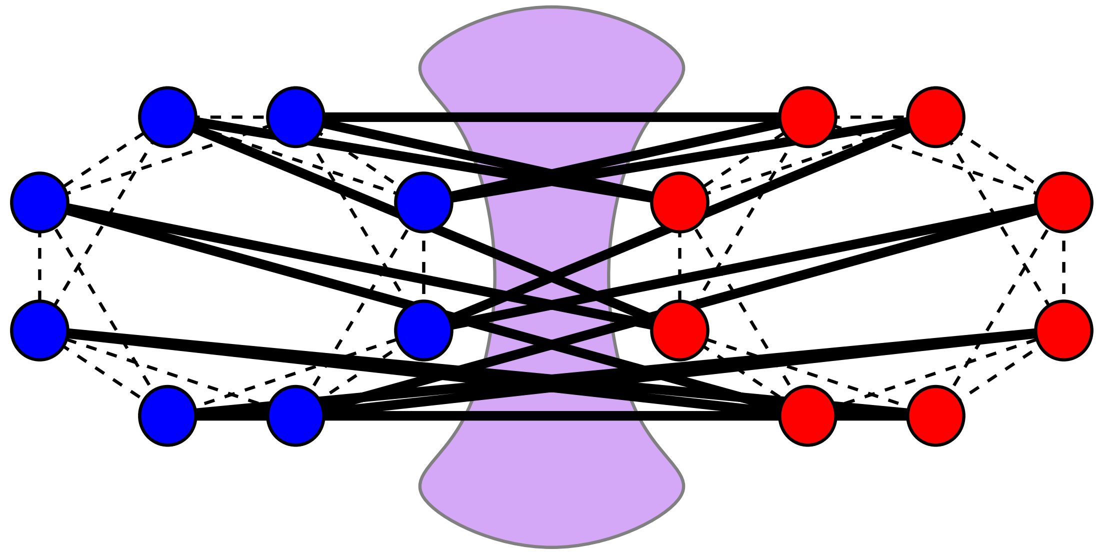
- $|V_1| = |V_2|$,
- $G\big|_{V_1}, G\big|_{V_2} \sim$ random a-regular graphs
- $G\big|_{E(V_1,V_2)} \sim$ random b-regular bipartite graph.



Toy Case: Regular Stochastic Block Model

Regular SBM (RSBM) [Brito et al. SODA'16]. A graph $G = (V_1 \dot{\bigcup} V_2, E)$ s.t.

- $|V_1| = |V_2|$,
- $G\big|_{V_1}, G\big|_{V_2}$ ~ random a-regular graphs
- $G\big|_{E(V_1,V_2)}$ ~ random b-regular bipartite graph.

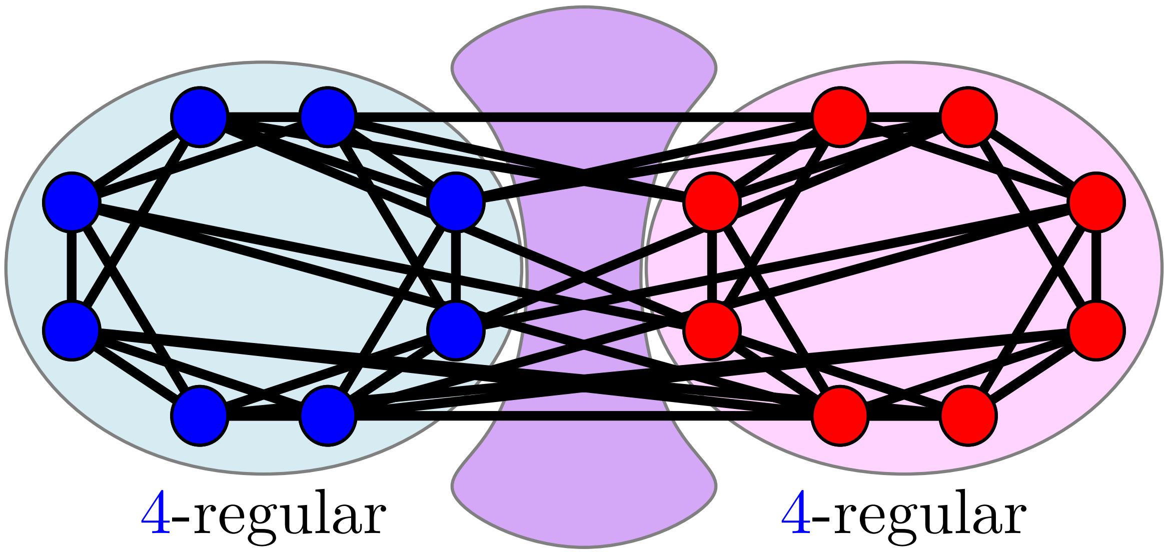


2-regular bipartite

Toy Case: Regular Stochastic Block Model

Regular SBM (RSBM) [Brito et al. SODA'16]. A graph $G = (V_1 \dot{\bigcup} V_2, E)$ s.t.

- $|V_1| = |V_2|$,
- $G\big|_{V_1}, G\big|_{V_2} \sim$ random a-regular graphs
- $G\big|_{E(V_1,V_2)} \sim$ random b-regular bipartite graph.



Analysis on Regular SBM

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

Analysis on Regular SBM

P → symmetric $\Longrightarrow$ 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

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

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

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

Analysis on Regular SBM

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

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

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

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

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

Analysis on Regular SBM

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

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

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

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

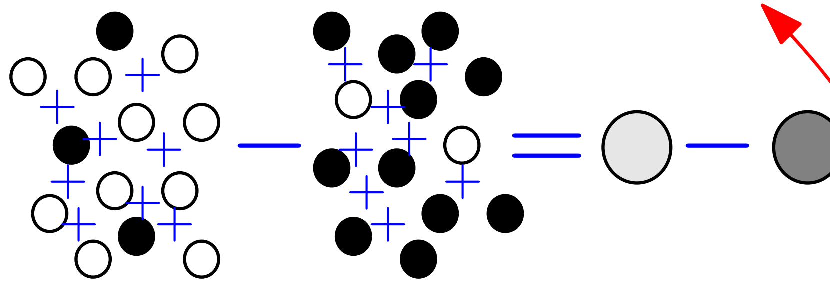
W.h.p. $\max\{\lambda_3, |\lambda_n|\}(1+\delta) < \frac{a-b}{a+b} = \lambda_2$, then

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

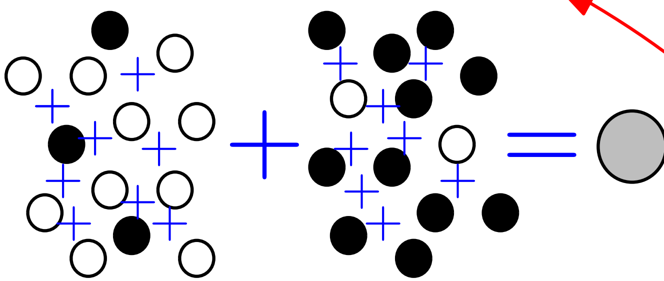
with $\|\mathbf{e}^{(t)}\| \leq (\max\{\lambda_3, |\lambda_n|\})^t \sqrt{n}$

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. $\max\{\lambda_3, |\lambda_n|\}(1+\delta) < \frac{a-b}{a+b} = \lambda_2$, 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 $\|\mathbf{e}^{(t)}\| \le (\max\{\lambda_3, |\lambda_n|\})^t\sqrt{n}$

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

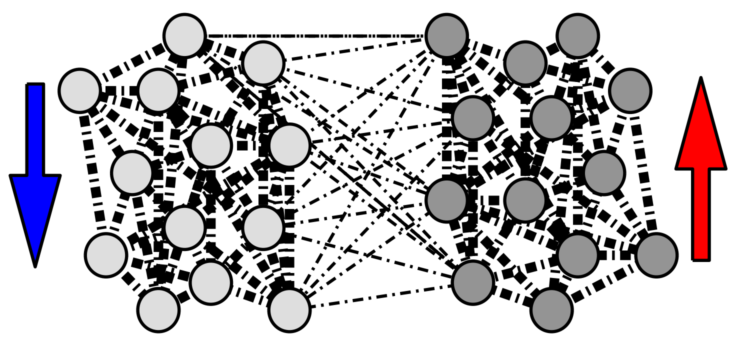
$$\mathbf{x}^{(t)} = \frac{1}{n}(\mathbf{1}^\intercal \mathbf{x}^{(0)})\mathbf{1} + \underbrace{\left(\frac{a-b}{a+b}\right)}_{=\lambda_2}^{t} \frac{1}{n}(\chi^\intercal \mathbf{x}^{(0)})\chi + \mathbf{e}^{(t)}$$

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

Analysis on Regular SBM

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

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



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

Sparsification of the Averaging Dynamics

Averaging Dynamics in $\mathcal{LOCAL}$ Model:
$\mathcal{O}(d)$ messages per round :-(

Sparsification of the Averaging Dynamics

Averaging Dynamics in $\mathcal{LOCAL}$ Model:
$\mathcal{O}(d)$ messages per round :-(

Can we *sparsify* the process?
$\Longrightarrow$ Do averaging only over some *random* edges.

Sparsification of the Averaging Dynamics

Averaging Dynamics in $\mathcal{LOCAL}$ Model:
$\mathcal{O}(d)$ messages per round :-(

Can we *sparsify* the process?
$\Longrightarrow$ Do averaging only over some *random* edges.

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

Random matrices!

Sparsification of the Averaging Dynamics

Averaging Dynamics in $\mathcal{LOCAL}$ Model:

$\mathcal{O}(d)$ messages per round :-(

Can we *sparsify* the process?

$\Longrightarrow$ Do averaging only over some *random* edges.

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

Random matrices!

Expected behavior:

$$\mathbf{E}\left[\mathbf{x}^{(t)} \,|\, \mathbf{x}^{(0)}\right] = \mathbf{E}\left[P\right] \cdot \mathbf{E}\left[\mathbf{x}^{(t-1)} \,|\, \mathbf{x}^{(0)}\right] = \left(\mathbf{E}\left[P\right]\right)^{t} \cdot \mathbf{x}^{(0)}$$

Sparsification of the Averaging Dynamics

Averaging Dynamics in $\mathcal{LOCAL}$ Model:
$\mathcal{O}(d)$ messages per round :-(

Can we *sparsify* the process?
$\Longrightarrow$ Do averaging only over some *random* edges.

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

Random matrices!

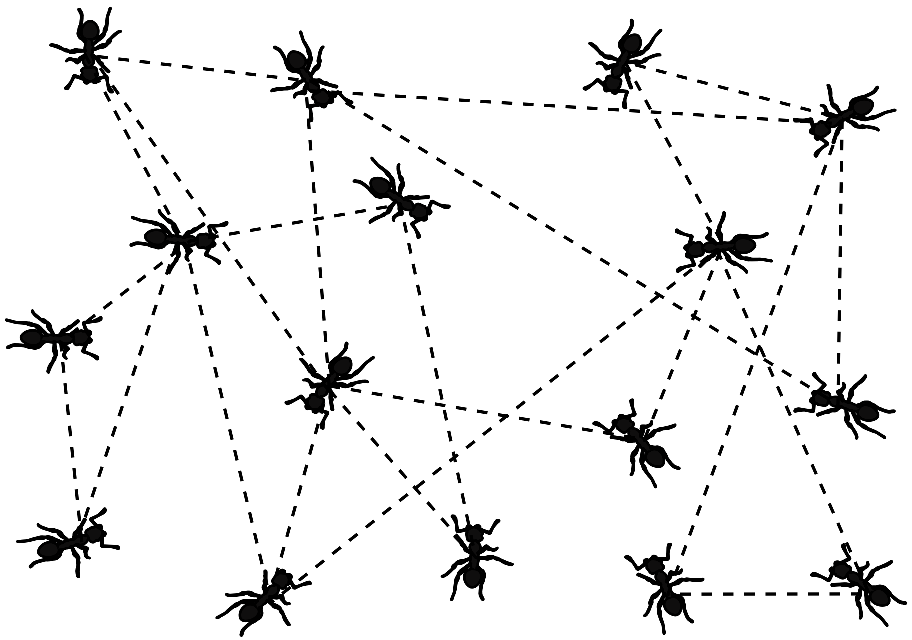
Expected behavior:

$$\mathbf{E}\left[\mathbf{x}^{(t)} \,|\, \mathbf{x}^{(0)}\right] = \mathbf{E}\left[P\right] \cdot \mathbf{E}\left[\mathbf{x}^{(t-1)} \,|\, \mathbf{x}^{(0)}\right] = \left(\mathbf{E}\left[P\right]\right)^t \cdot \mathbf{x}^{(0)}$$

Problem: no concentration tools for matrix *products*
(e.g. no logarithm for noncommutative matrices)

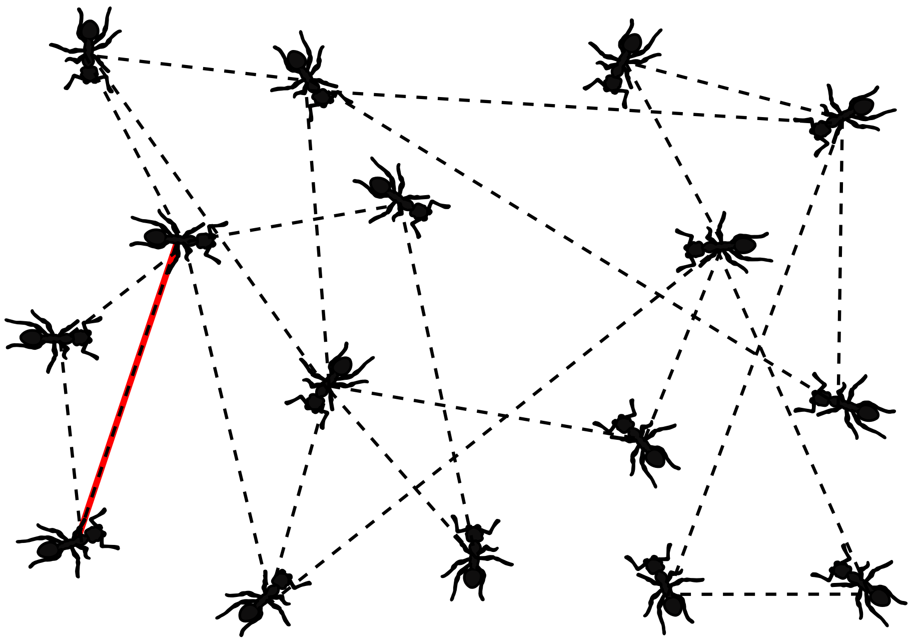
Communication Model: Population Protocol

Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



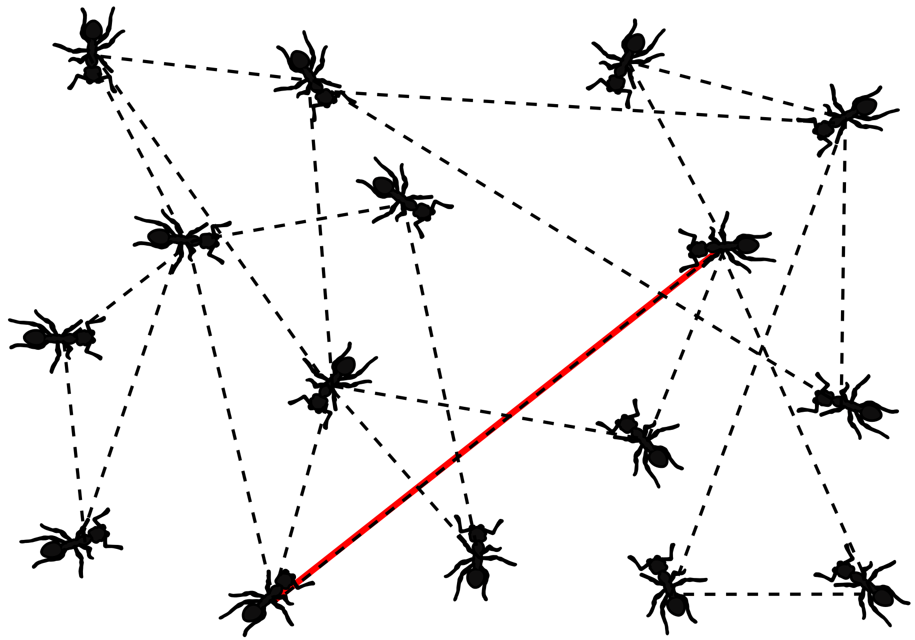
Communication Model: Population Protocol

Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



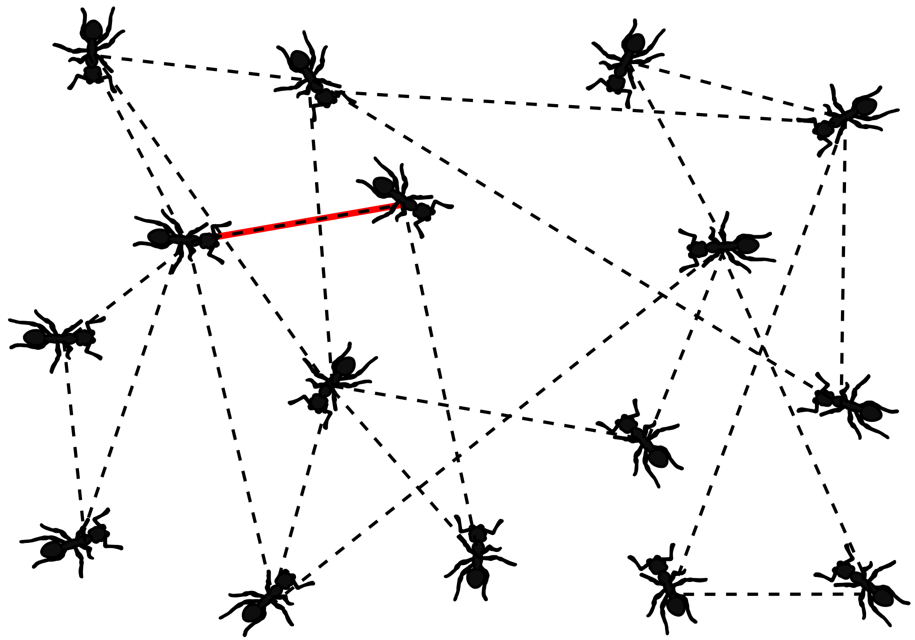
Communication Model: Population Protocol

Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



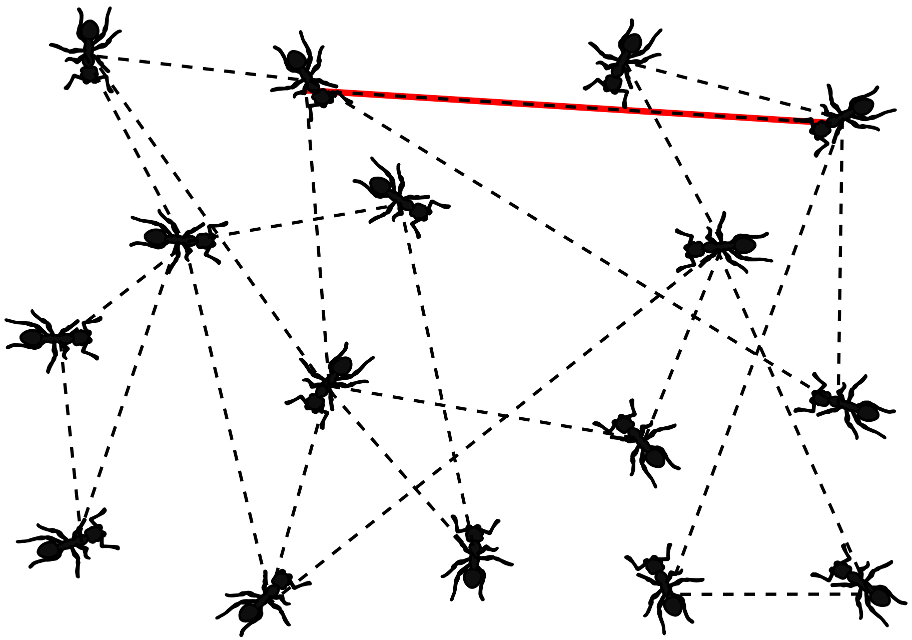
Communication Model: Population Protocol

Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



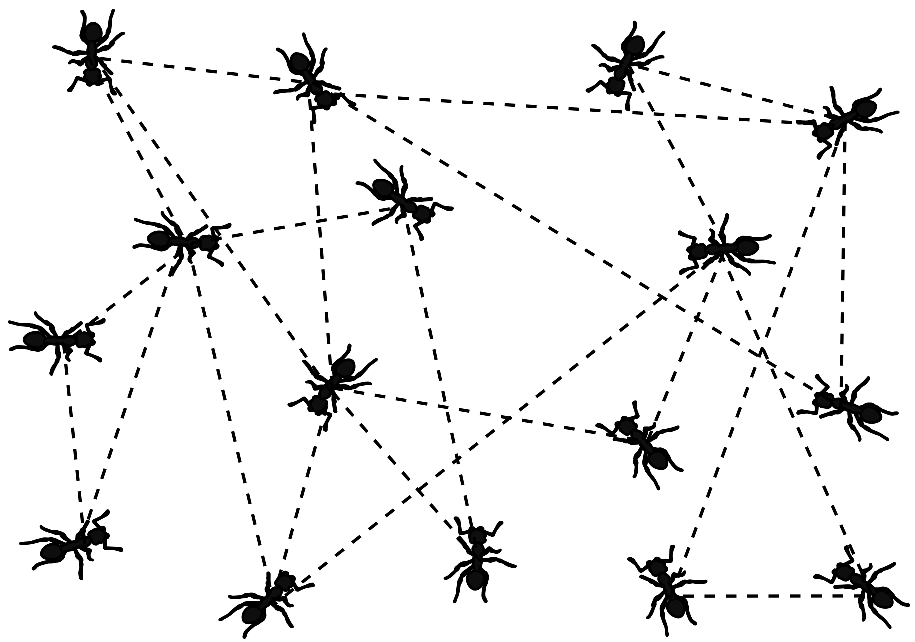
Communication Model: Population Protocol

Population protocol: at each round a random edge is chosen and the two corresponding agent interact.

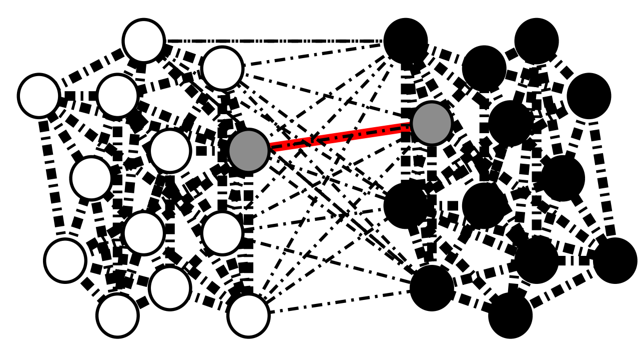


Communication Model: Population Protocol

Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



!!!: The *variance* of picking a random edge breaks the monotonicity and seems to prevent concentration.



Community Sensitive Labeling

CSL(m,T):

- At the outset
$$\mathbf{x}_u^{(0)} \sim \mathrm{Unif}(\{-1,+1\}^m).$$
- In each round, the endpoints of the random edge choose a random index $j \in [m]$ and set
$$\mathbf{x}_u(j) = \mathbf{x}_v(j) = \frac{\mathbf{x}_u(j) + \mathbf{x}_v(j)}{2};$$ (cfr [Boyd et al. '06]).
- At the T-th update of j-th component, u sets $\mathbf{h}_u(j) = \mathbf{sgn}(\mathbf{x}_u(j))$.

Community Sensitive Labeling

CSL(m,T):

- At the outset $$\mathbf{x}_u^{(0)} \sim \mathrm{Unif}(\{-1,+1\}^m).$$
- In each round, the endpoints of the random edge choose a random index $j \in [m]$ and set $$\mathbf{x}_u(j) = \mathbf{x}_v(j) = \frac{\mathbf{x}_u(j) + \mathbf{x}_v(j)}{2};$$ (cfr [Boyd et al. '06]).
- At the T-th update of j-th component, u sets $\mathbf{h}_u(j) = \mathbf{sgn}(\mathbf{x}_u(j))$.

Thm. $G = (V_1 \dot{\bigcup} V_2, E)$ regular SBM s.t. $d\epsilon^4 \gg b \log^2 n$, then CSL($m$,$T$) with $m = \Theta(\epsilon^{-1} \log n)$ and $T = \Theta(\log n)$ labels all nodes but a set U with size $|U| \leq \sqrt{\epsilon} n$, in such a way that

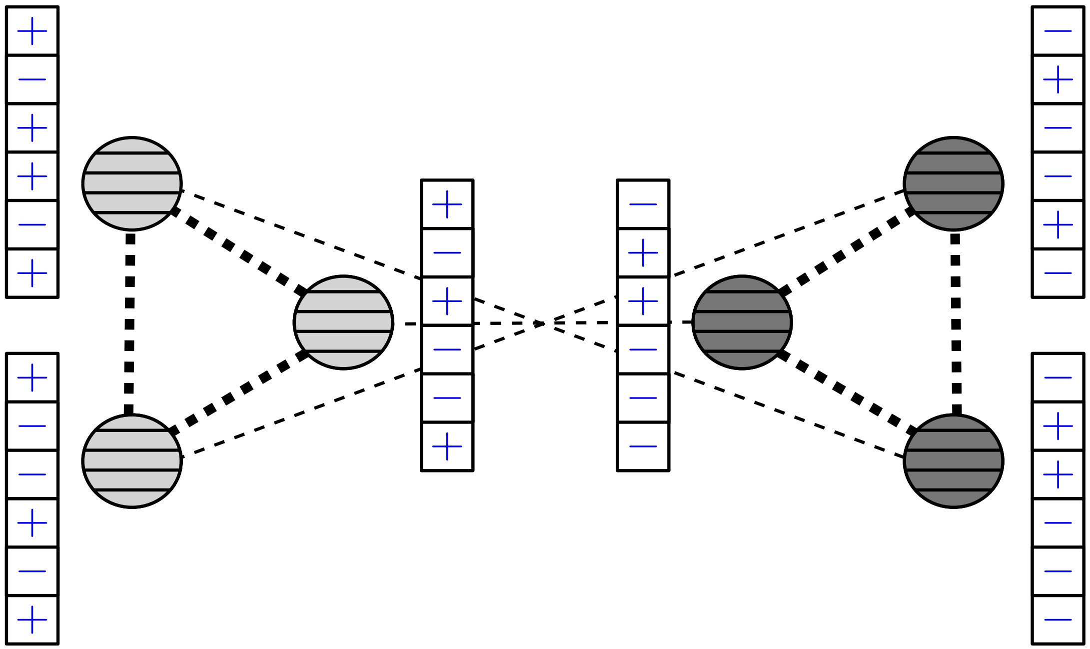
- the labels of nodes in the same community agree on at least $5/6$ entries, and
- the labels of nodes in different communities differ in more than $1/6$ entries.

Community Sensitive Labeling

Example:

> 2 different labels
$\Longrightarrow$ foes!

≤ 2 different labels
$\Longrightarrow$ friends!



Warning: not a dynamics!

Analysis 1/4

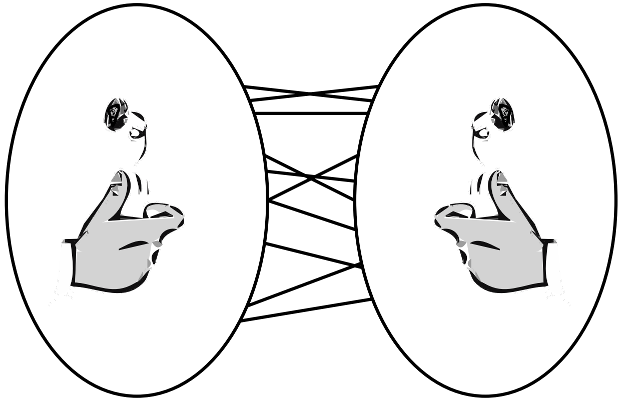
Proof Ingredient 1. We are done if, for any fixed component j, all *lucky* nodes $u \notin U$ are such that

$$\Pr\left(h_u = \mathbf{sgn}\Big(\sum_{v \in V(u)} \mathbf{x}_v\Big)\right) \geq \frac{99}{100}.$$

Analysis 1/4

Proof Ingredient 1. We are done if, for any fixed component j, all *lucky* nodes $u \notin U$ are such that

$$\Pr\left(h_u = \mathbf{sgn}\Big(\underbrace{\sum_{v \in V(u)} \mathbf{x}_v}\Big)\right) \geq \frac{99}{100}.$$



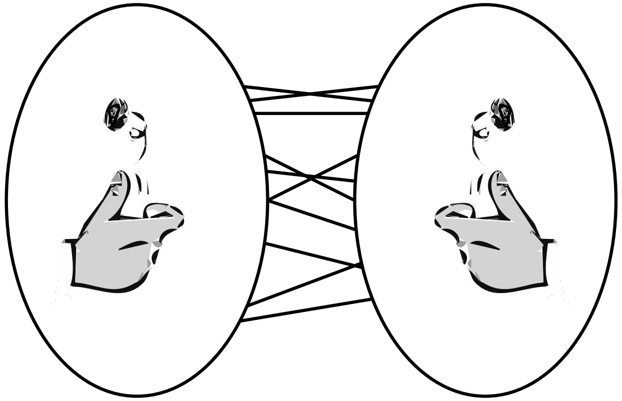
$\mathbf{x}_u^{(0)} \sim \mathrm{Unif}(\{-1, +1\})$.

$\Pr(\sum_{v \in V_1} \mathbf{x}_v^{(0)} > 0 > \sum_{v \in V_2} \mathbf{x}_v^{(0)}) \approx \frac{1}{2}$

Analysis 1/4

Proof Ingredient 1. We are done if, for any fixed component j, all *lucky* nodes $u \notin U$ are such that

$$\Pr\left(h_u = \mathbf{sgn}\left(\underbrace{\sum_{v \in V(u)} \mathbf{x}_v}\right)\right) \geq \frac{99}{100}.$$



$\mathbf{x}_u^{(0)} \sim \mathrm{Unif}(\{-1, +1\})$.

$\Pr(\sum_{v \in V_1} \mathbf{x}_v^{(0)} > 0 > \sum_{v \in V_2} \mathbf{x}_v^{(0)}) \approx \frac{1}{2}$

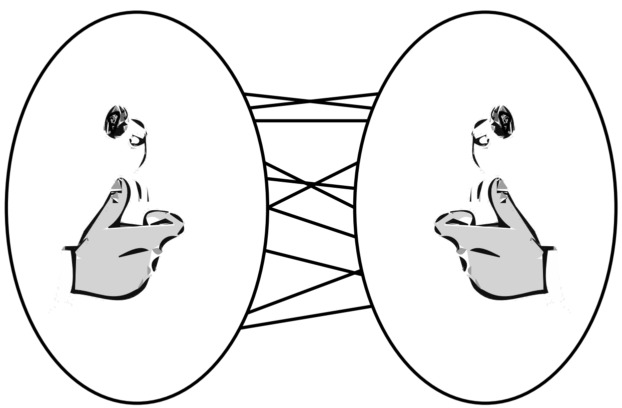
(Obs. $\Pr(|\sum_{v \in V_i} \mathbf{x}_v^{(0)}| < n^{\epsilon}) \ll \frac{n^{\epsilon}}{\sqrt{n}}$)

Analysis 1/4

Proof Ingredient 1. We are done if, for any fixed component j, all *lucky* nodes $u \notin U$ are such that

sign of $\mathbf{x}_u$ at (local) time T

$$\Pr\left(h_u = \mathbf{sgn}\Big(\underbrace{\sum_{v \in V(u)} \mathbf{x}_v}\Big)\right) \geq \frac{99}{100}.$$



$\mathbf{x}_u^{(0)} \sim \mathrm{Unif}(\{-1, +1\})$.

$\Pr(\sum_{v \in V_1} \mathbf{x}_v^{(0)} > 0 > \sum_{v \in V_2} \mathbf{x}_v^{(0)}) \approx \frac{1}{2}$

(Obs. $\Pr(|\sum_{v \in V_i} \mathbf{x}_v^{(0)}| < n^{\epsilon}) \ll \frac{n^{\epsilon}}{\sqrt{n}}$)

Problem: bound $|U| = \#$*unlucky* nodes (i.e. $\mathbf{sgn}(\mathbf{x}_u^{(T)})$ is wrong with prob. $> 1/100$).

Analysis 2/4

Proof Ingredient 2. h_u is a random variable!

$\implies$ Synchronicity issue, cannot union bound...

Analysis 2/4

Proof Ingredient 2. h_u is a random variable!

$\implies$ Synchronicity issue, cannot union bound...

W.h.p. T happens in (global) time $\Theta(n \log n)$.

Analysis 2/4

Proof Ingredient 2. h_u is a random variable!
$\implies$ Synchronicity issue, cannot union bound...

W.h.p. T happens in (global) time $\Theta(n \log n)$.

$\implies$ if for any $t = \Theta(n \log n)$ we prove
$\approx \epsilon^2 n$ nodes u are *bad*, namely

$$(\mathbf{x}_u^{(t)} - \sum_{v \in V(u)} \mathbf{x}_v^{(0)})^2 > \frac{\epsilon^2}{n}$$

then we can bound the *unlucky nodes* by bounding a *spreading process*:

- At time $10n \log n$, $\approx \epsilon^2 n$ nodes are *bad/unlucky*, and
- at each following round, a good node become bad **iff** we pick a *cross edge* or an *edge touching a bad node*.

Analysis 2/4

Proof Ingredient 2. h_u is a random variable!

$\implies$ Synchronicity issue, cannot union bound...

W.h.p. T happens in (global) time $\Theta(n \log n)$.

Next idea

$\implies$ if for any $t = \Theta(n \log n)$ we prove $\approx \epsilon^2 n$ nodes u are *bad*, namely

$$(\mathbf{x}_u^{(t)} - \sum_{v \in V(u)} \mathbf{x}_v^{(0)})^2 > \frac{\epsilon^2}{n}$$

then we can bound the *unlucky nodes* by bounding a *spreading process*:

- At time $10n \log n$, $\approx \epsilon^2 n$ nodes are *bad/unlucky*, and
- at each following round, a good node become bad **iff** we pick a *cross edge* or an *edge touching a bad node*.

Analysis (3-)4/4: Second Moment Analysis

Proof Ingredient 3. If $\sum_u (\mathbf{x}_u^{(10n \log n)} - \sum_{v \in V(u)} \mathbf{x}_v^{(0)})^2$ is small *(Ingredient 4)*, it remains small for $\mathcal{O}(n \log n)$ rounds.

Analysis (3-)4/4: Second Moment Analysis

Proof Ingredient 3. If $\sum_u (\mathbf{x}_u^{(10n\log n)} - \sum_{v \in V(u)} \mathbf{x}_v^{(0)})^2$ is small *(Ingredient 4)*, it remains small for $\mathcal{O}(n \log n)$ rounds.

(Essentially triangle and Markov ineq.s on Ingredient 4.)

Analysis (3-)4/4: Second Moment Analysis

Proof Ingredient 3. If $\sum_u (\mathbf{x}_u^{(10n\log n)} - \sum_{v\in V(u)} \mathbf{x}_v^{(0)})^2$ is small *(Ingredient 4)*, it remains small for $\mathcal{O}(n\log n)$ rounds.

(Essentially triangle and Markov ineq.s on Ingredient 4.)

Proof Ingredient 4. Use Markov ineq. on

$$\mathbf{E}\Big[\sum_u (\mathbf{x}_u^{(t)} - \sum_{v\in V(u)} \mathbf{x}_v^{(0)})^2\Big]$$

$$= \mathbf{E}\left[\|\mathbf{x}^{(t)} - \pi_{\mathbf{v}_{1,2}}(\mathbf{x}^{(0)})\|^2\right]$$

$$= \mathbf{E}\left[\|\pi_{\mathbf{v}_{\geq 2}}(\mathbf{x}_u^{(t)}) - \pi_{\mathbf{v}_2}(\mathbf{x}_u^{(0)})\|^2\right]$$

$$\leq \mathbf{E}\left[\|\prod P^{(i)}\pi_{\mathbf{v}_2}(\mathbf{x}_u^{(t)}) - \pi_{\mathbf{v}_2}(\mathbf{x}_u^{(0)})\|^2\right]$$

$$+ \mathbf{E}\left[\|\prod P^{(i)}\pi_{\mathbf{v}_{\geq 3}}(\mathbf{x}_u^{(0)})\|^2\right].$$

$\pi_{\mathbf{v}_i}(\mathbf{x})$ projection on i-th eigenspace

$P^{(i)}$ matrix of averaging at time i

Analysis (3-)4/4: Second Moment Analysis

Proof Ingredient 3. If $\sum_u (\mathbf{x}_u^{(10n\log n)} - \sum_{v\in V(u)} \mathbf{x}_v^{(0)})^2$ is small *(Ingredient 4)*, it remains small for $\mathcal{O}(n\log n)$ rounds.

(Essentially triangle and Markov ineq.s on Ingredient 4.)

Proof Ingredient 4. Use Markov ineq. on

$$\mathbf{E}\Big[\sum_u (\mathbf{x}_u^{(t)} - \sum_{v\in V(u)} \mathbf{x}_v^{(0)})^2\Big]$$

$$= \mathbf{E}\left[\|\mathbf{x}^{(t)} - \pi_{\mathbf{v}_{1,2}}(\mathbf{x}^{(0)})\|^2\right]$$

$$= \mathbf{E}\left[\|\pi_{\mathbf{v}_{\geq 2}}(\mathbf{x}_u^{(t)}) - \pi_{\mathbf{v}_2}(\mathbf{x}_u^{(0)})\|^2\right]$$

$$\leq \mathbf{E}\left[\|\prod P^{(i)}\pi_{\mathbf{v}_2}(\mathbf{x}_u^{(t)}) - \pi_{\mathbf{v}_2}(\mathbf{x}_u^{(0)})\|^2\right] \longleftarrow \text{Not hard to bound}$$

$$+\,\mathbf{E}\left[\|\prod P^{(i)}\pi_{\mathbf{v}_{\geq 3}}(\mathbf{x}_u^{(0)})\|^2\right]. \longleftarrow \text{Need double recurrence}$$

$\pi_{\mathbf{v}_i}(\mathbf{x})$ projection on i-th eigenspace

$P^{(i)}$ matrix of averaging at time i

Thank you!