Dynamics and Community Structure in Networks

Emanuele Natale

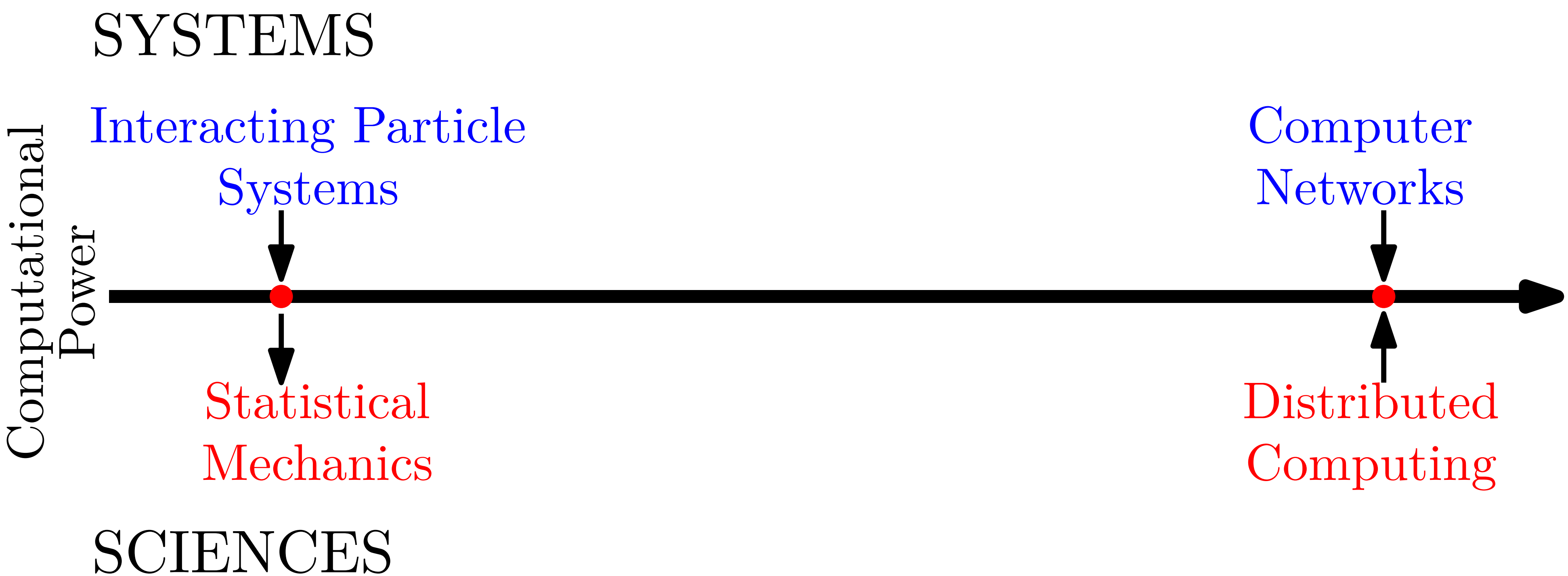
Computational Aspects of Complex Networks

Rome, December 6, 2024

Roadmap

- $\boxed{\text{Intro to Computational Dynamics}}$
- Community Detection via Synchronous Averaging
- Community Detection via Asynchronous Averaging
- 2-Choices on Clustered Graphs & Evolution

Communication in *Simple* Systems

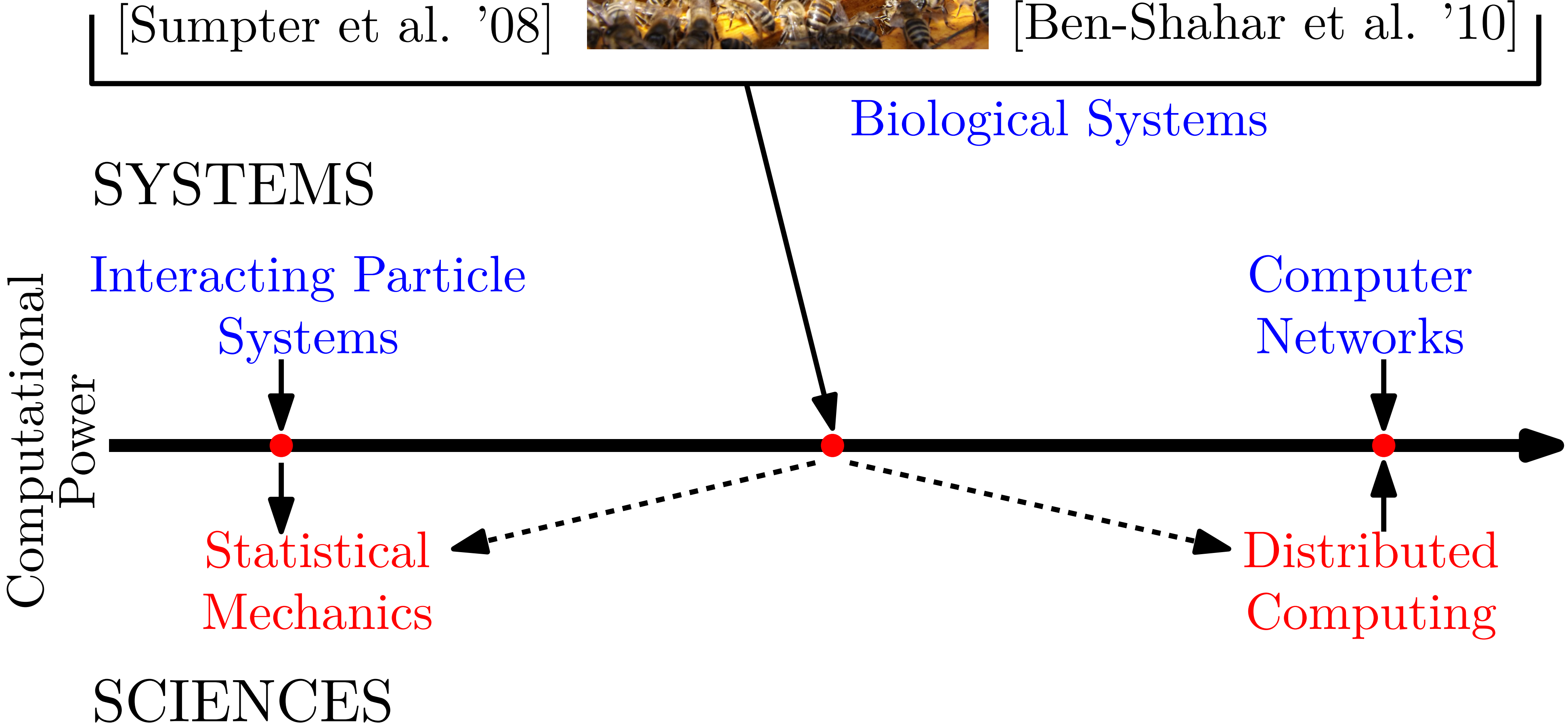


Communication in *Simple* Systems

Schools of fish
[Sumpter et al. '08]

Insects colonies
[Franks et al. '02]

Flocks of birds
[Ben-Shahar et al. '10]



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.

To go beyond this talk:

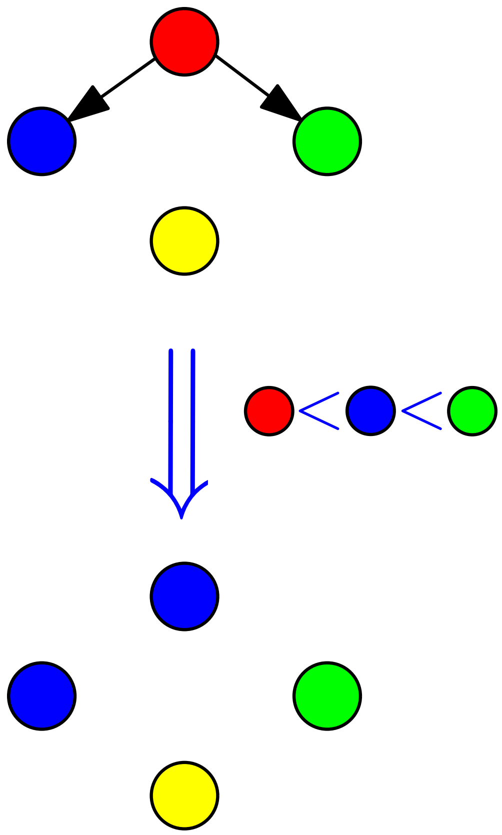
- Becchetti et al. *Consensus Dynamics: An Overview.* 2020.
- Mossel & Tamuz. *Opinion exchange dynamics.* 2017.
- Shah. *Gossip Algorithms.* 2007.

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

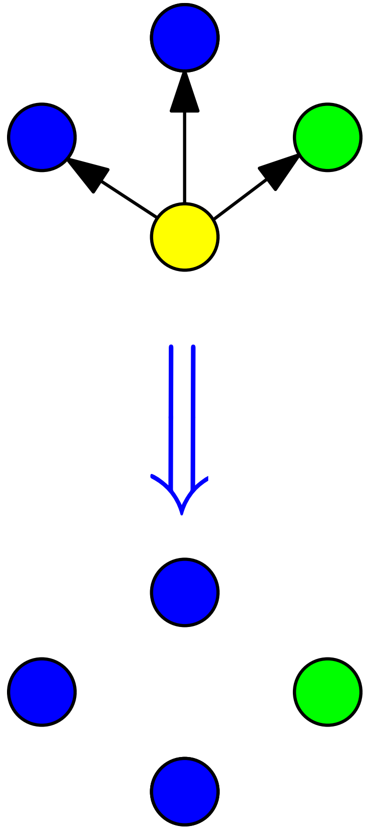


Dynamics

(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

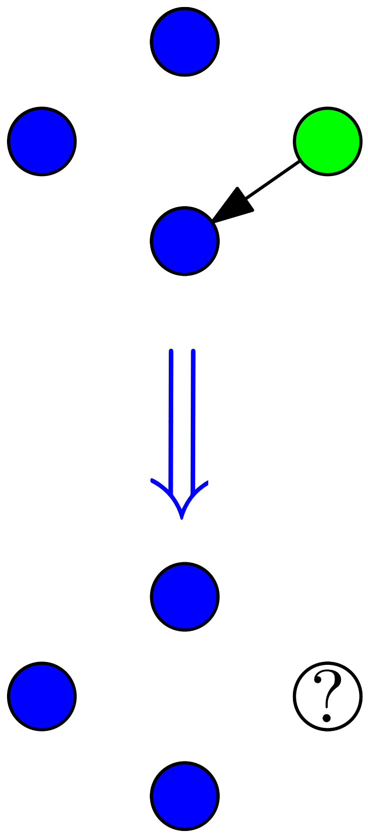


Dynamics

(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

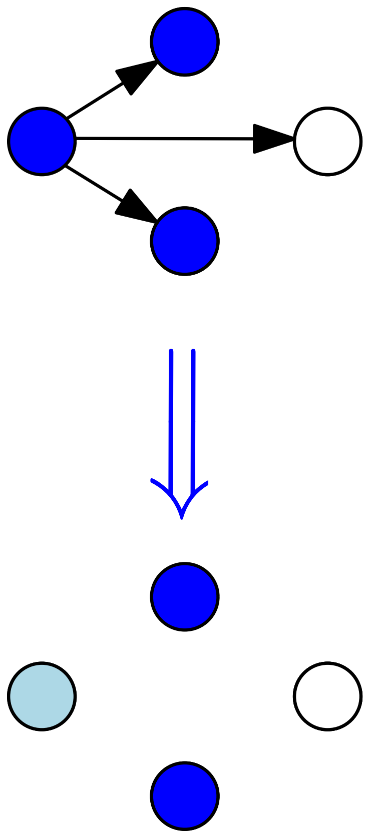


Dynamics

(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.

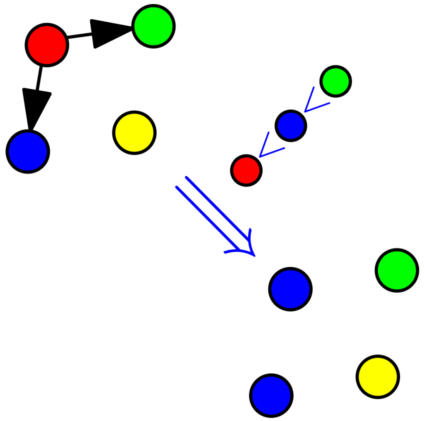
Examples of Dynamics

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



The Power of Dynamics: Plurality Consensus

Computing the Median

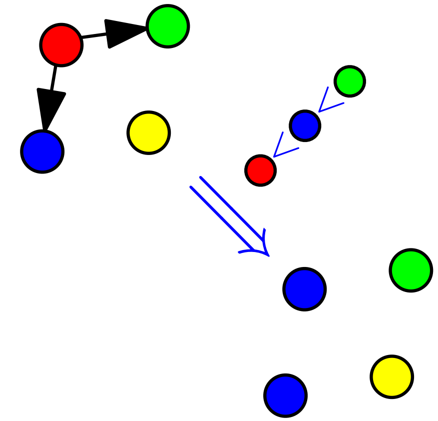


- 3-Median dynamics [Doerr et al. '11]. Converge to $\mathcal{O}(\sqrt{n \log n})$ approximation of median of system in $\mathcal{O}(\log n)$ rounds w.h.p., even if $\mathcal{O}(\sqrt{n})$ states are arbitrarily changed at each round ($\mathcal{O}(\sqrt{n})$-bounded adversary).

The Power of Dynamics: Plurality Consensus

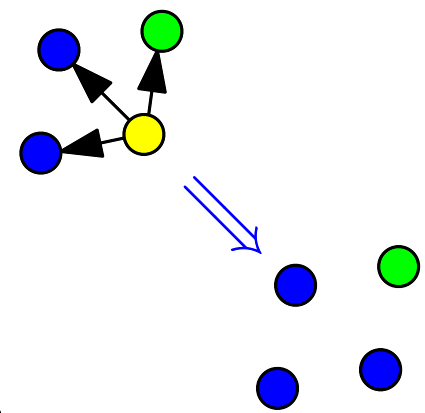
Computing the Median

- 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).



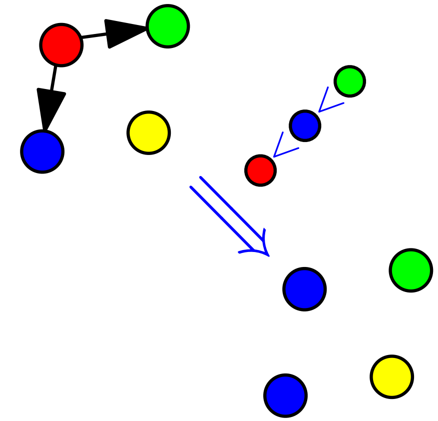
Computing the Majority

- 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 $\text{poly}(k)$. h-majority converges in $\Omega(k/h^2)$.



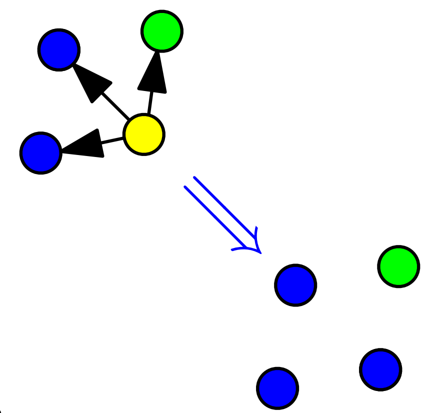
The Power of Dynamics: Plurality Consensus

Computing the Median

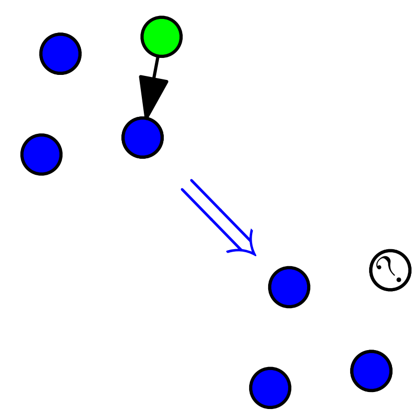


- 3-Median dynamics [Doerr et al. '11]. Converge to $\mathcal{O}(\sqrt{n \log n})$ approximation of median of system in $\mathcal{O}(\log n)$ rounds w.h.p., even if $\mathcal{O}(\sqrt{n})$ states are arbitrarily changed at each round ($\mathcal{O}(\sqrt{n})$-bounded adversary).

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?

Roadmap

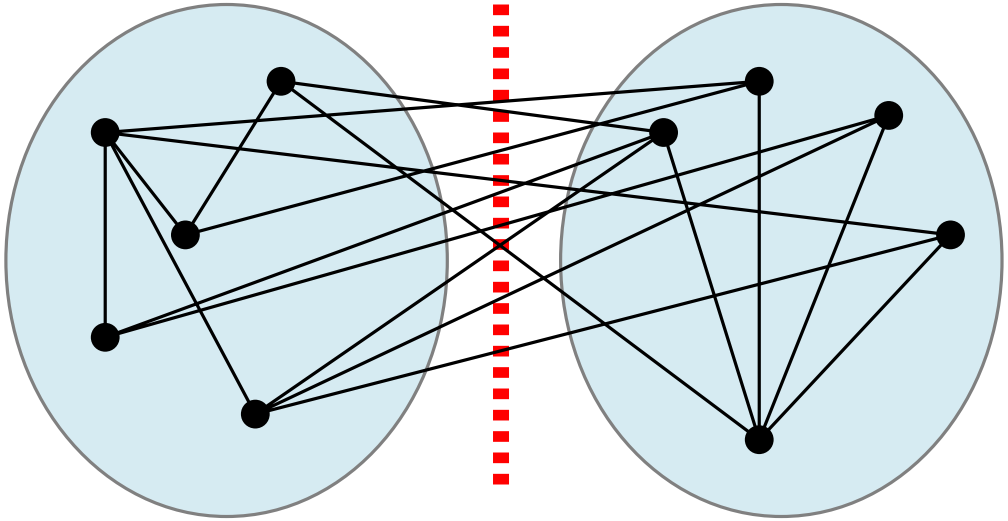
- Intro to Computational Dynamics
- Community Detection via Synchronous Averaging
- Community Detection via Asynchronous Averaging
- 2-Choices on Clustered Graphs & Evolution

Community Detection as Minimum Bisection

Minimum Bisection Problem.

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

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

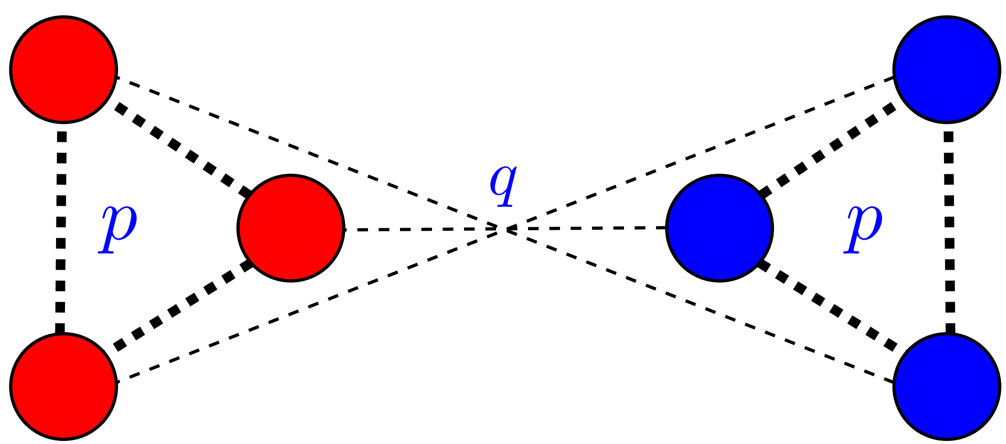


[Garey, Johnson, Stockmeyer '76]:

Min-Bisection is *NP-Complete*.

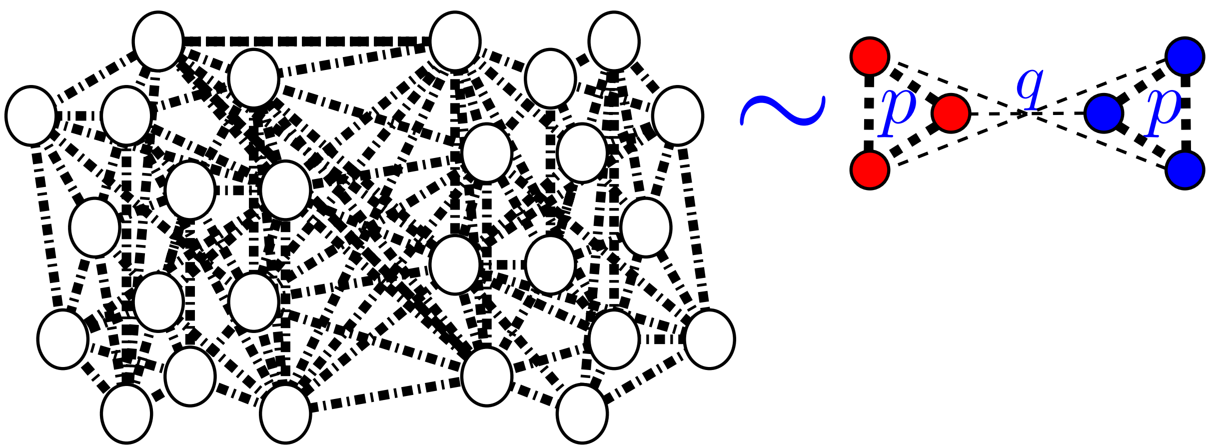
The Stochastic Block Model

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 $q = \frac{b}{n} < p$.



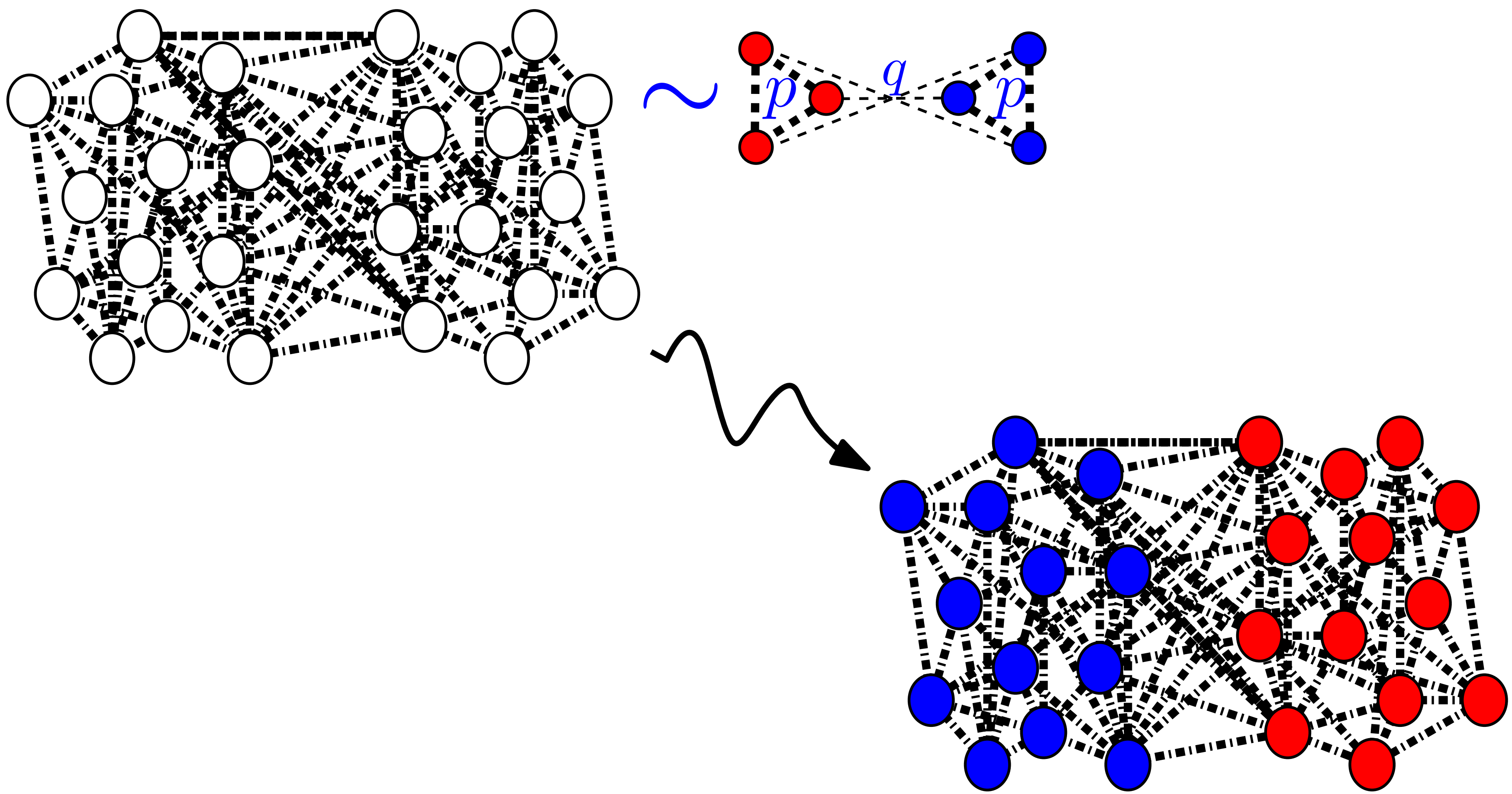
The Stochastic Block Model

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



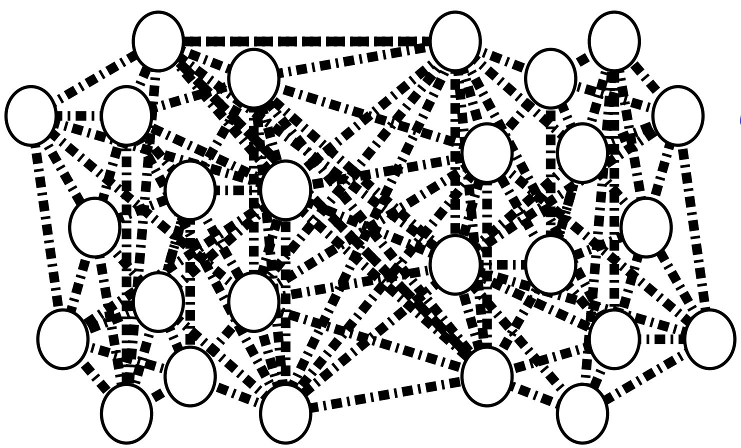
The Stochastic Block Model

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

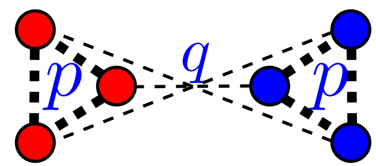


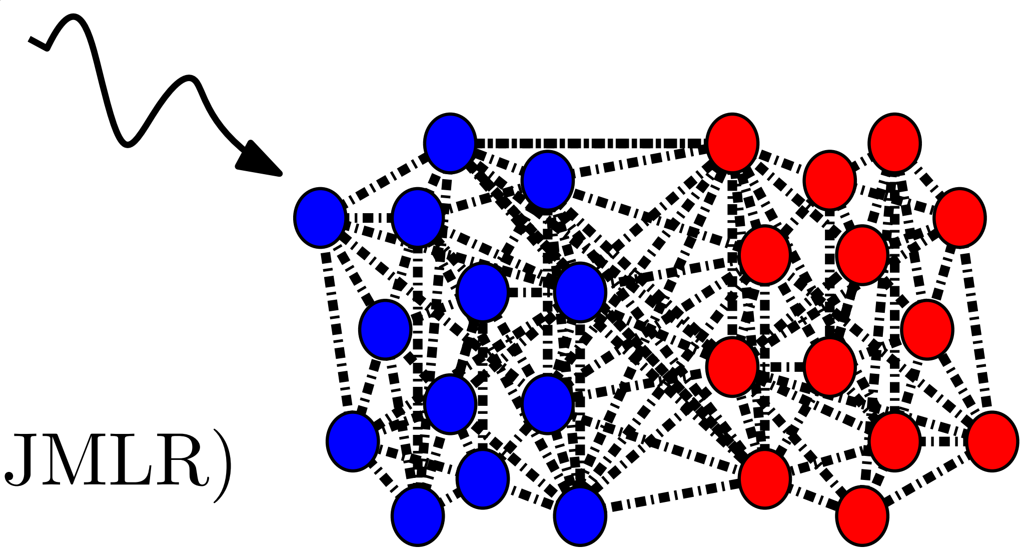
The Stochastic Block Model

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



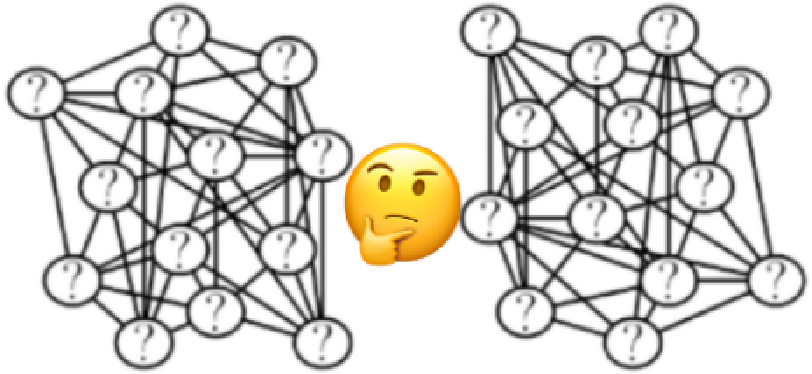
$\sim$



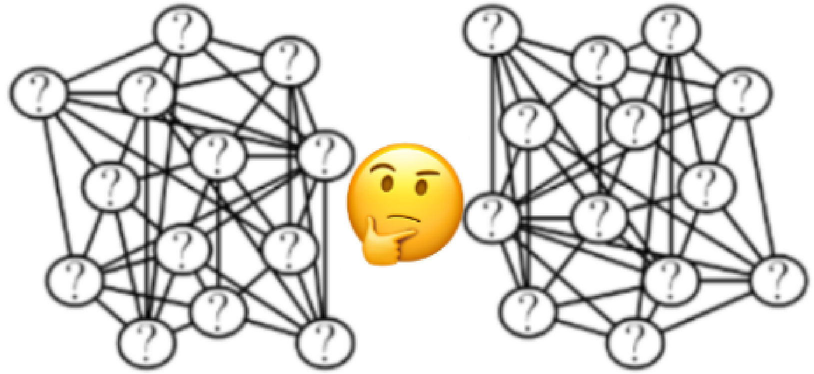


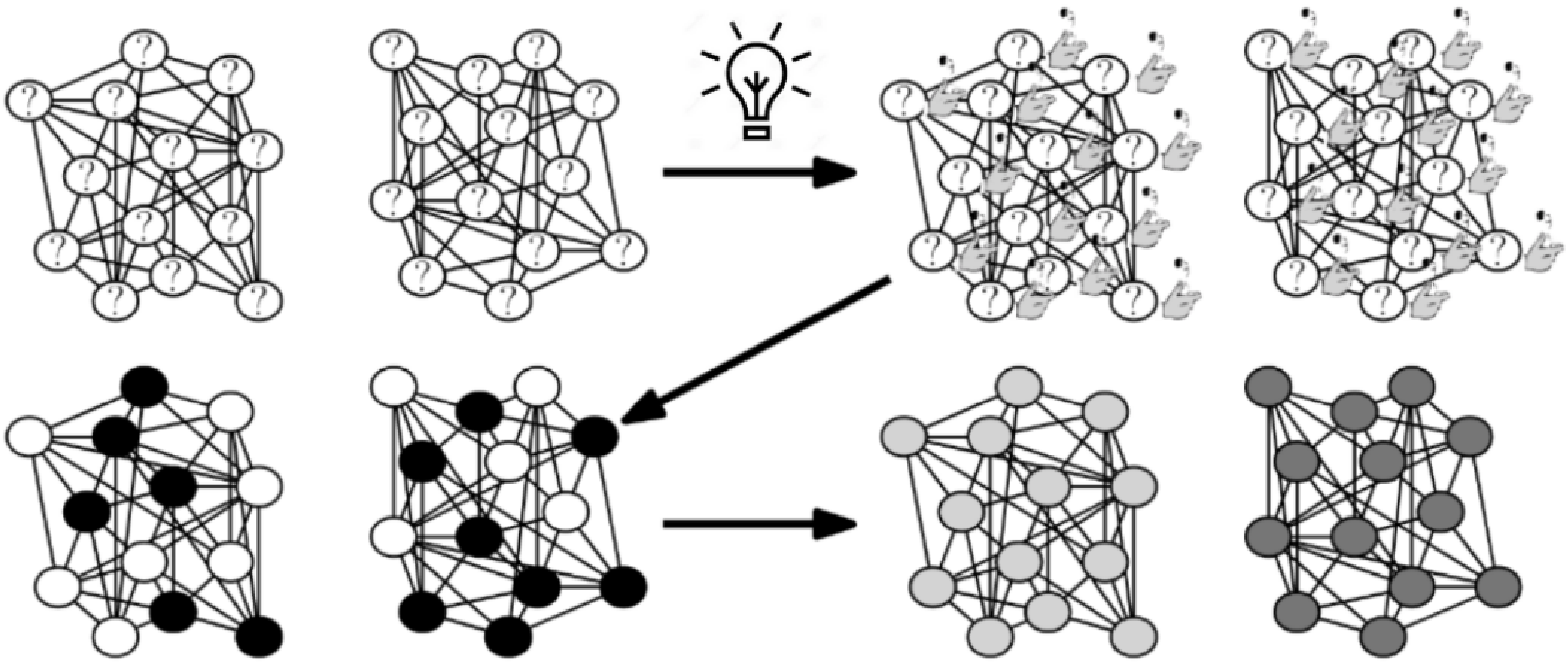
Exact reconstruction possible if
$\sqrt{p} - \sqrt{q} \geq \sqrt{2 \log n / n}$
(cfr. survey Abbe 2017 JMLR)

Community Detection via Averaging Dynamics

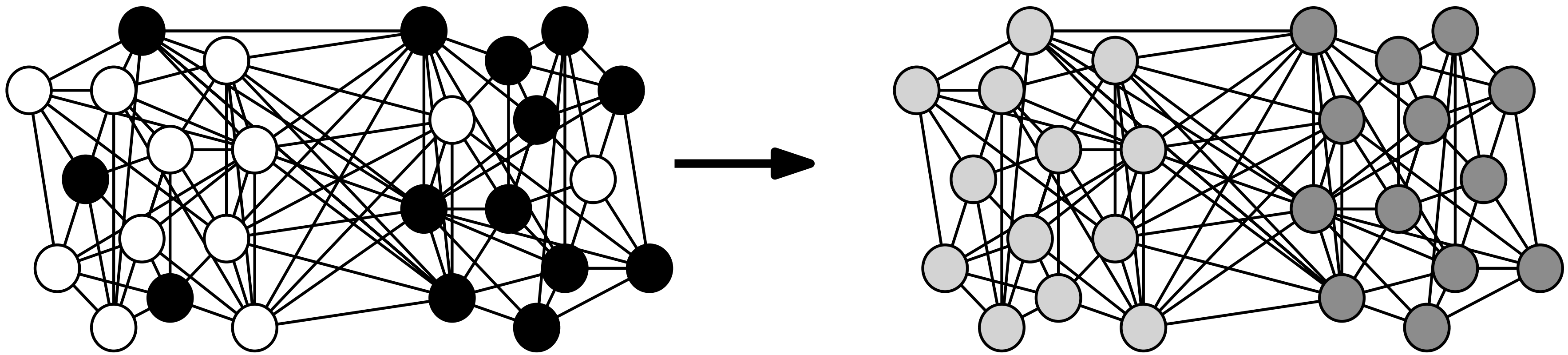


Community Detection via Averaging Dynamics

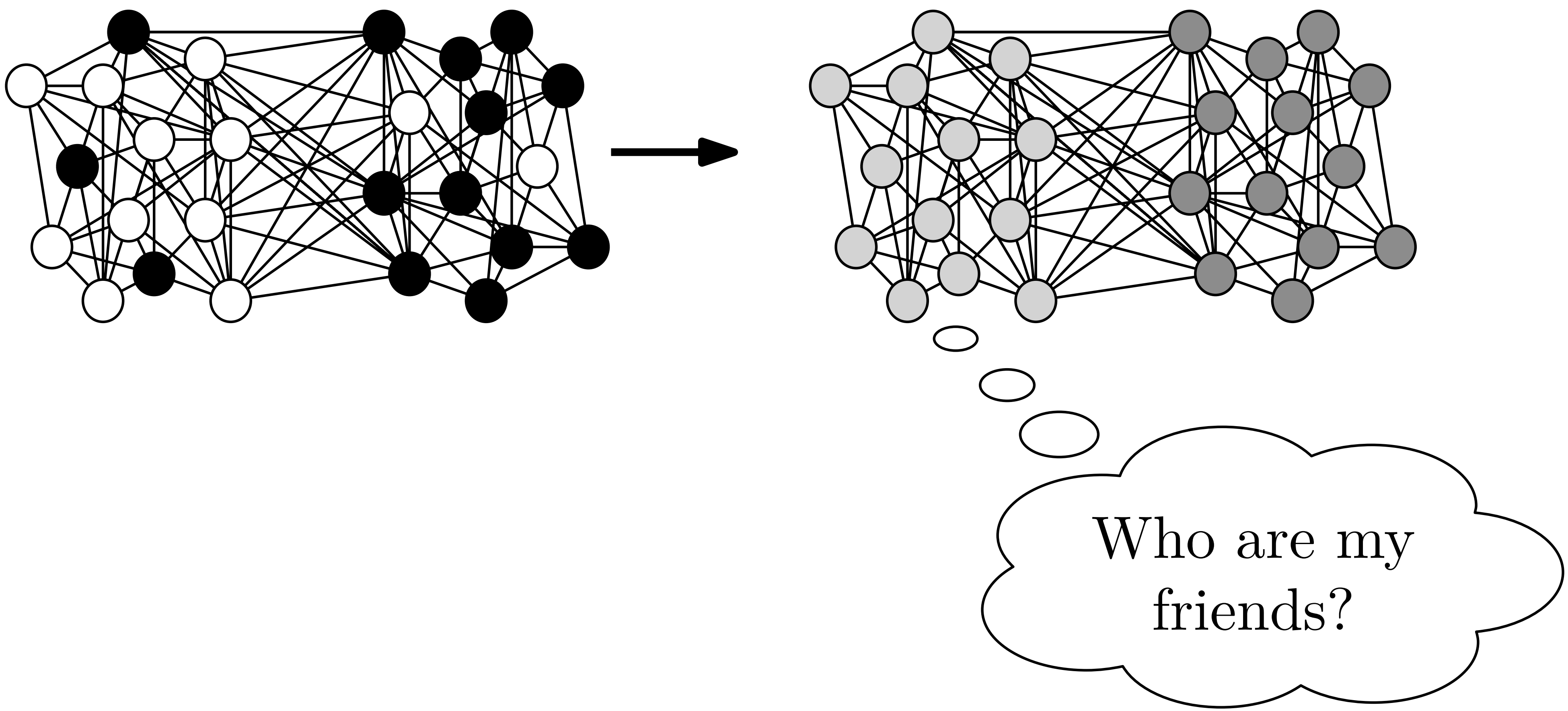




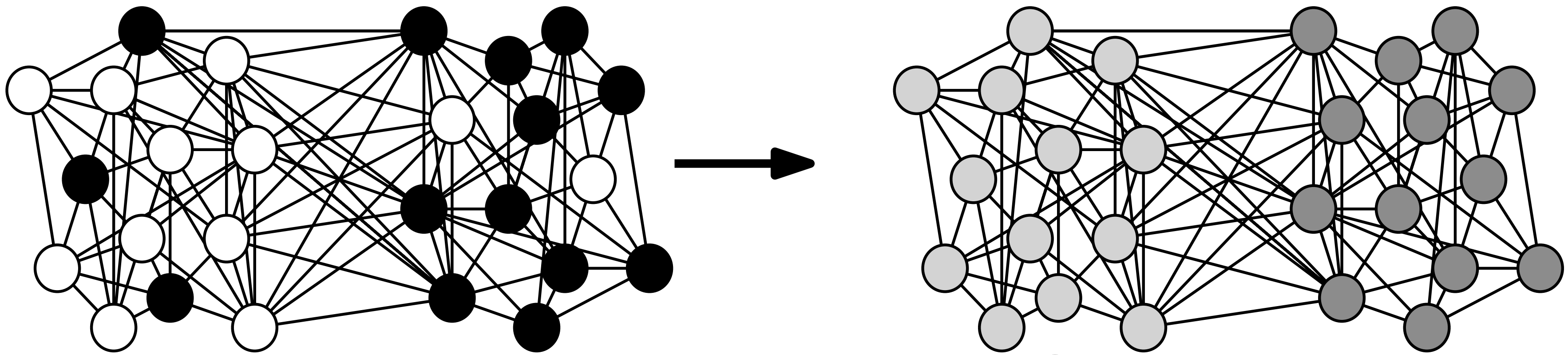
Community Detection via Averaging Dynamics



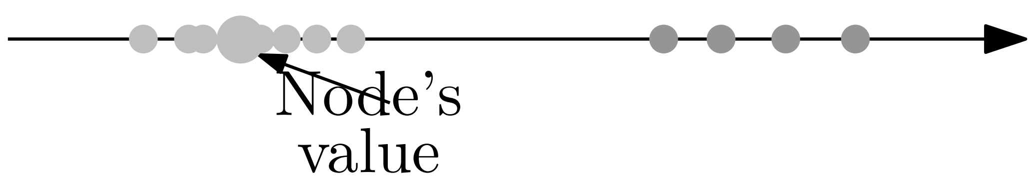
Community Detection via Averaging Dynamics



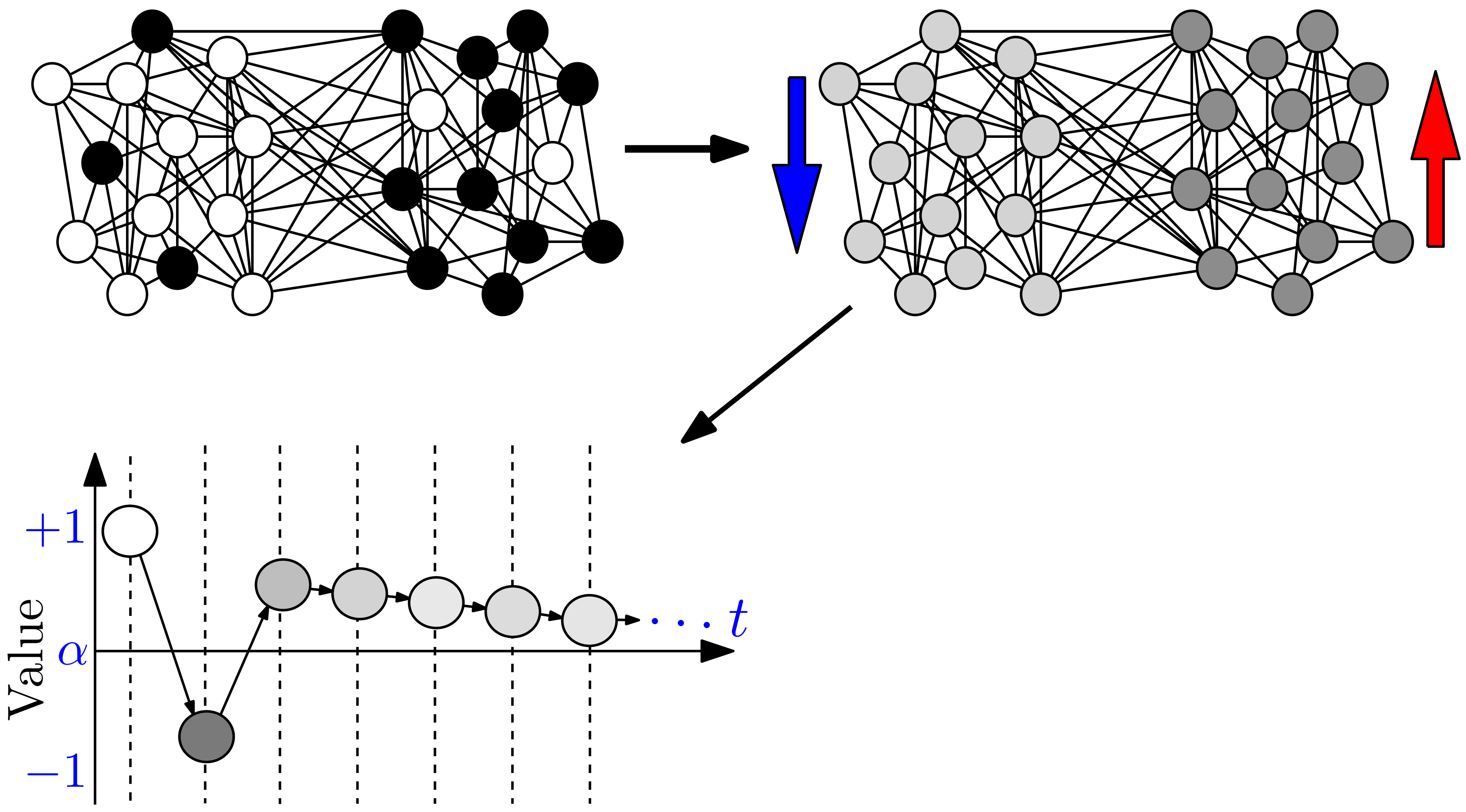
Community Detection via Averaging Dynamics



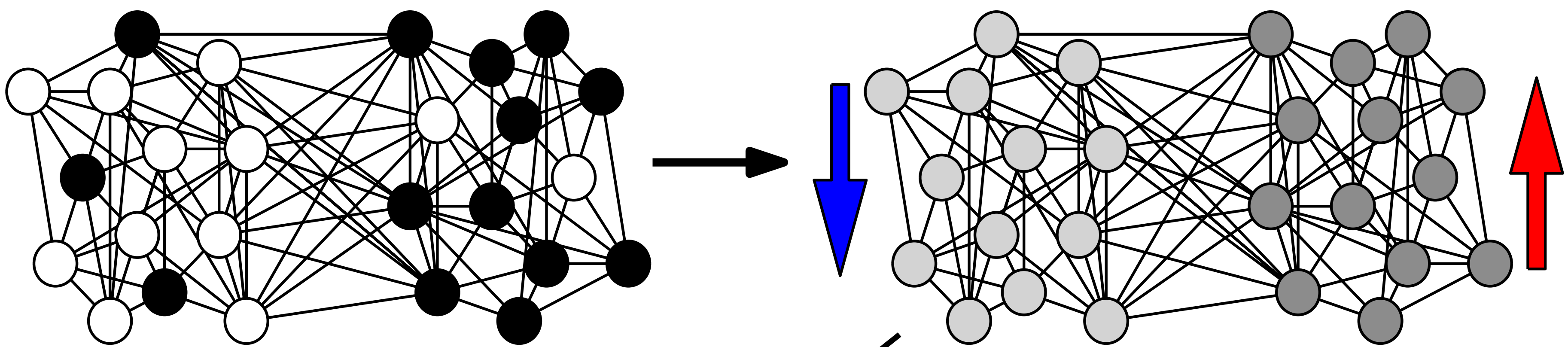
Local view of a node:

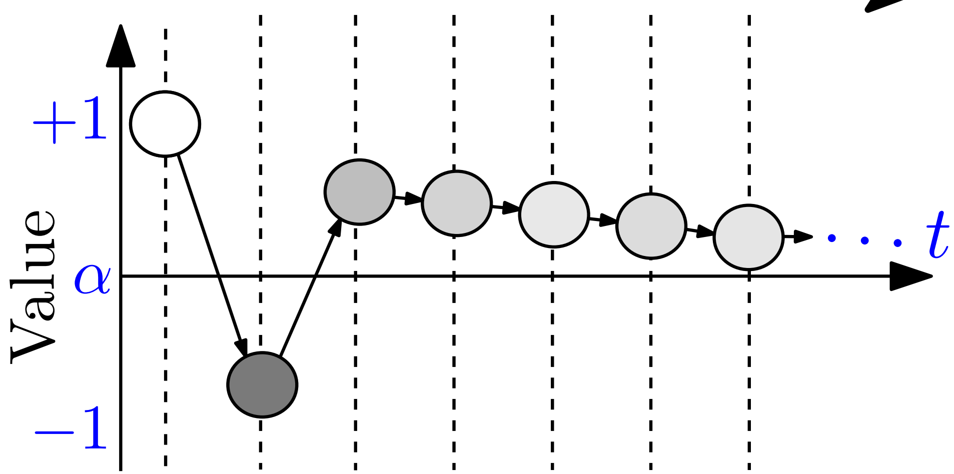


Community Detection via Averaging Dynamics



Community Detection via Averaging Dynamics





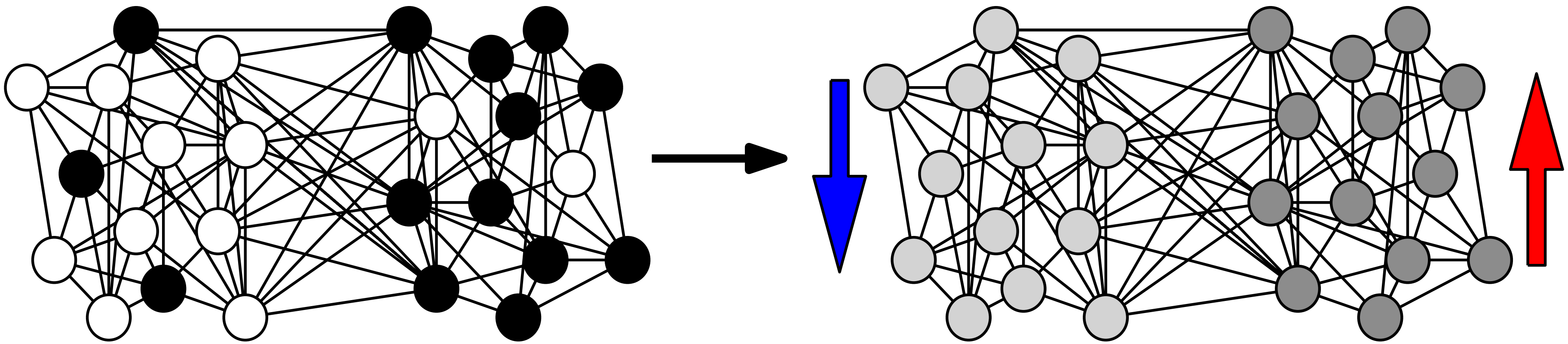
$v_1, \ldots, v_n$ eigenvectors of random walk matrix P:

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

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

"nice" graph

Community Detection via Averaging Dynamics

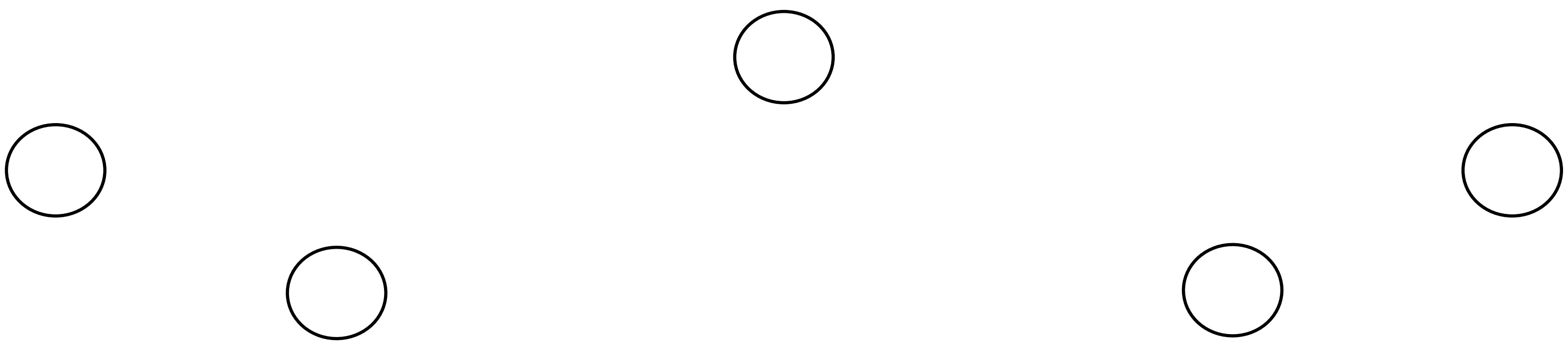


[SODA '17](Informal). $G = (V_1 \dot{\bigcup} V_2, E)$ s.t.
i) $\chi = \mathbf{1}_{V_1} - \mathbf{1}_{V_2}$ close to right-eigenvector of eigenvalue λ_2 of transition matrix of G, and
ii) gap between λ_2 and $\lambda = \max\{\lambda_3, |\lambda_n|\}$ large enough,
then Averaging (approximately) identifies (V_1, V_2) in $\mathcal{O}(\log n)$ rounds
(even when mixing time is polynomial!)

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

Al nodes at the same time:

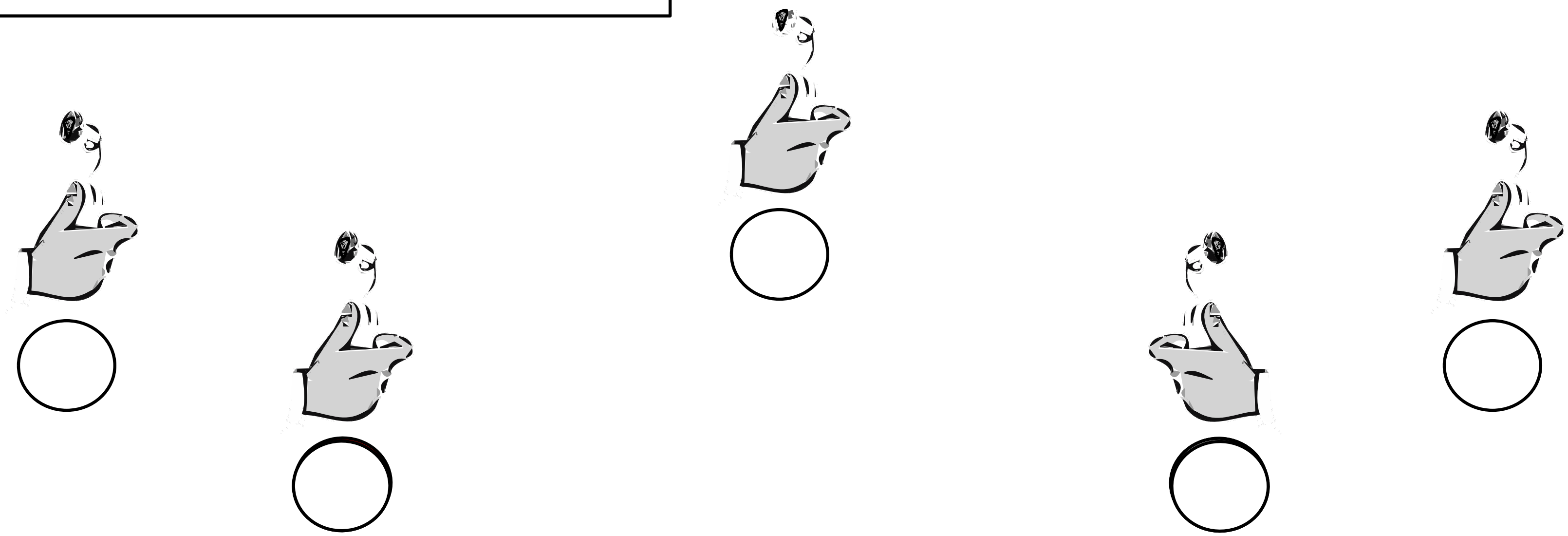
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  - Set value $x^{(t)}$ to average of neighbors,
  - Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



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

Al nodes at the same time:

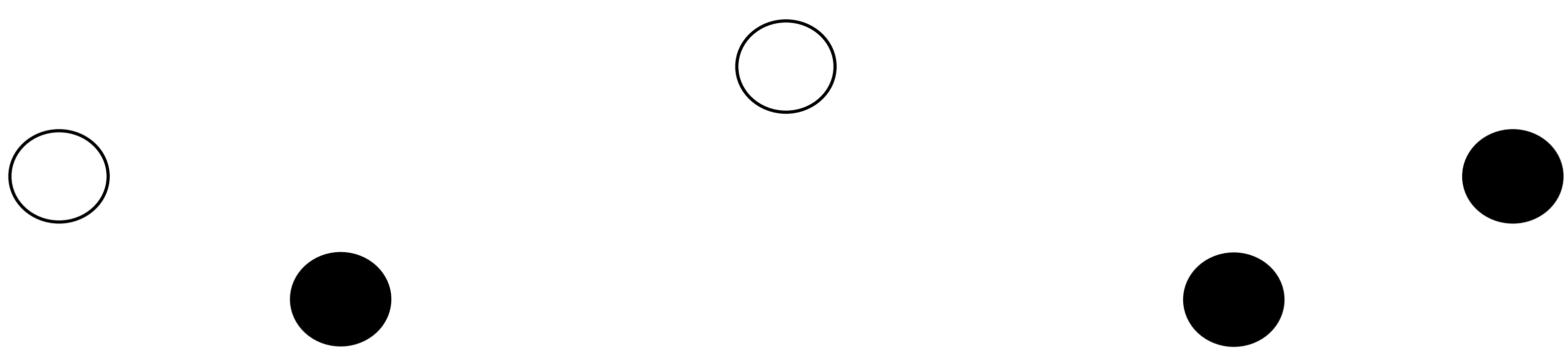
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  - Set value $x^{(t)}$ to average of neighbors,
  - Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



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

Al nodes at the same time:

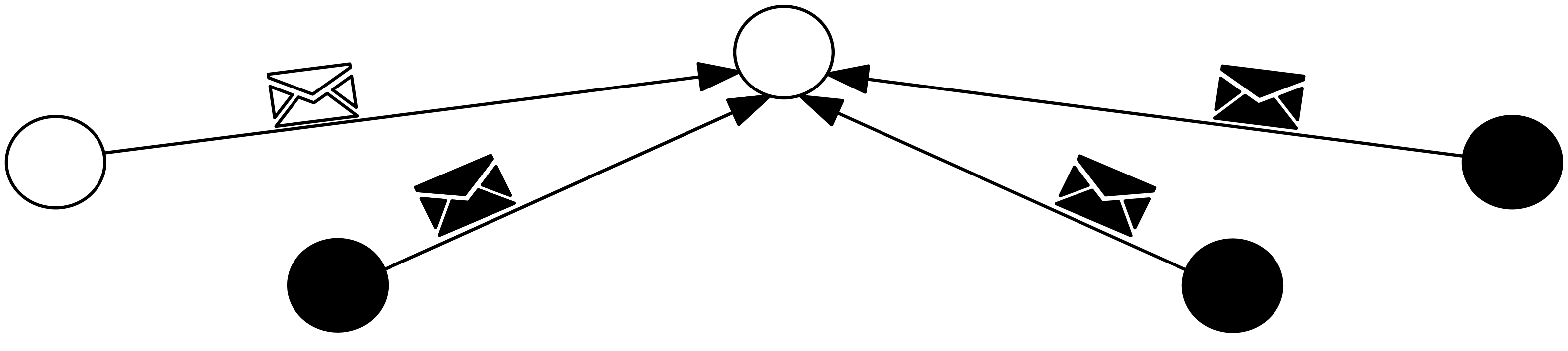
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{\text{-}1}\}$.
- Then, at each round
  - Set value $x^{(t)}$ to average of neighbors,
  - Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



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

Al nodes at the same time:

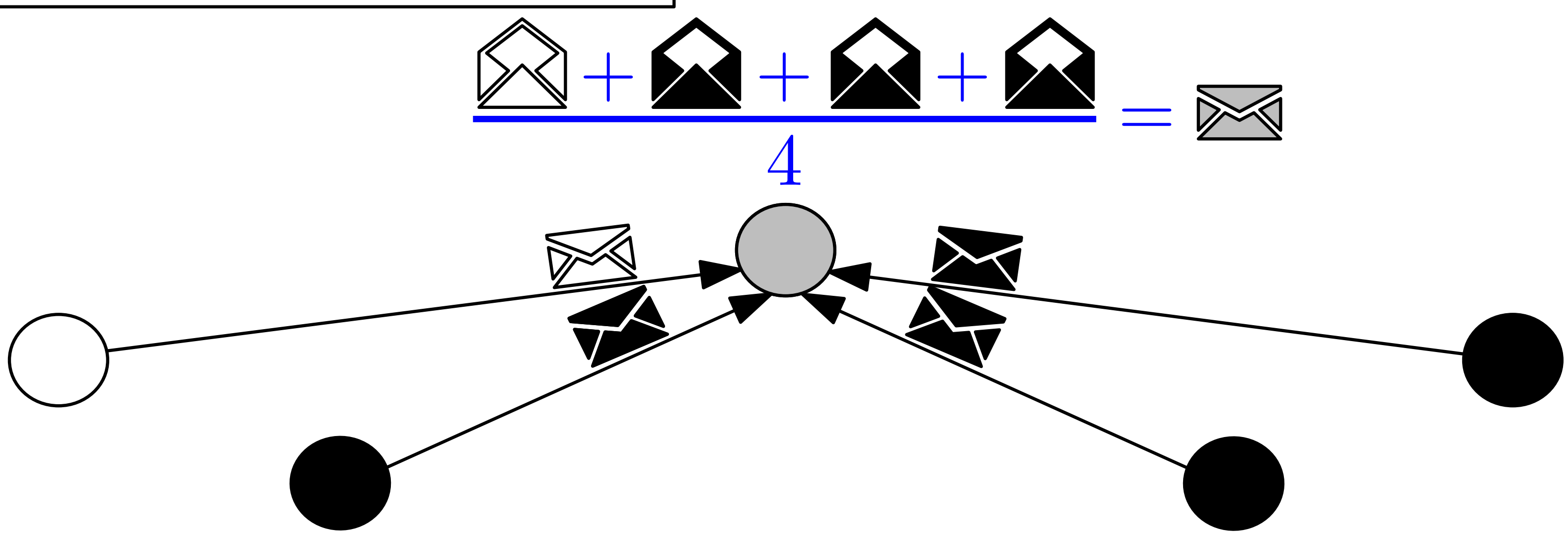
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  - Set value $x^{(t)}$ to average of neighbors,
  - Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



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

Al nodes at the same time:

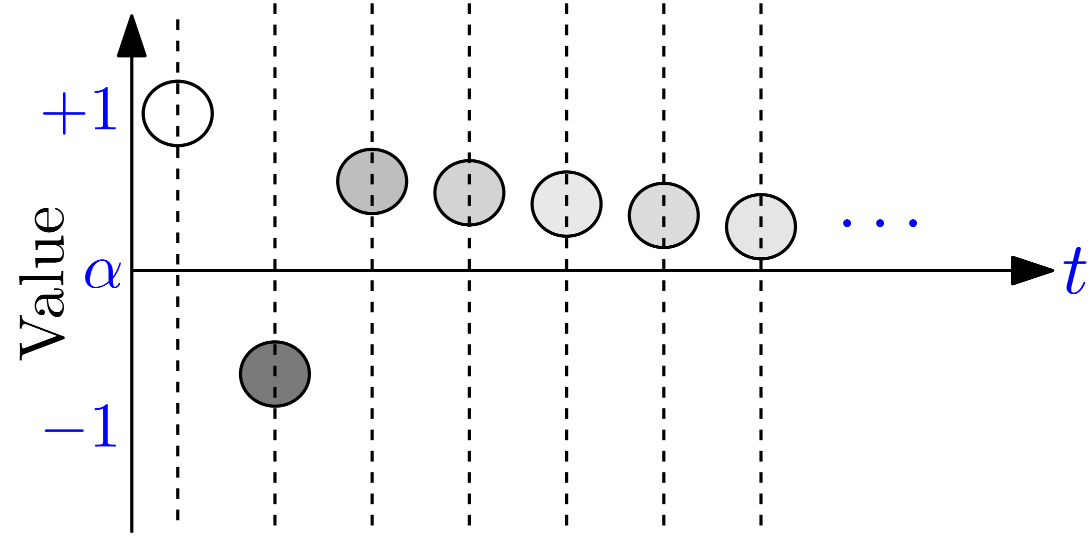
- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  - Set value $x^{(t)}$ to average of neighbors,
  - Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

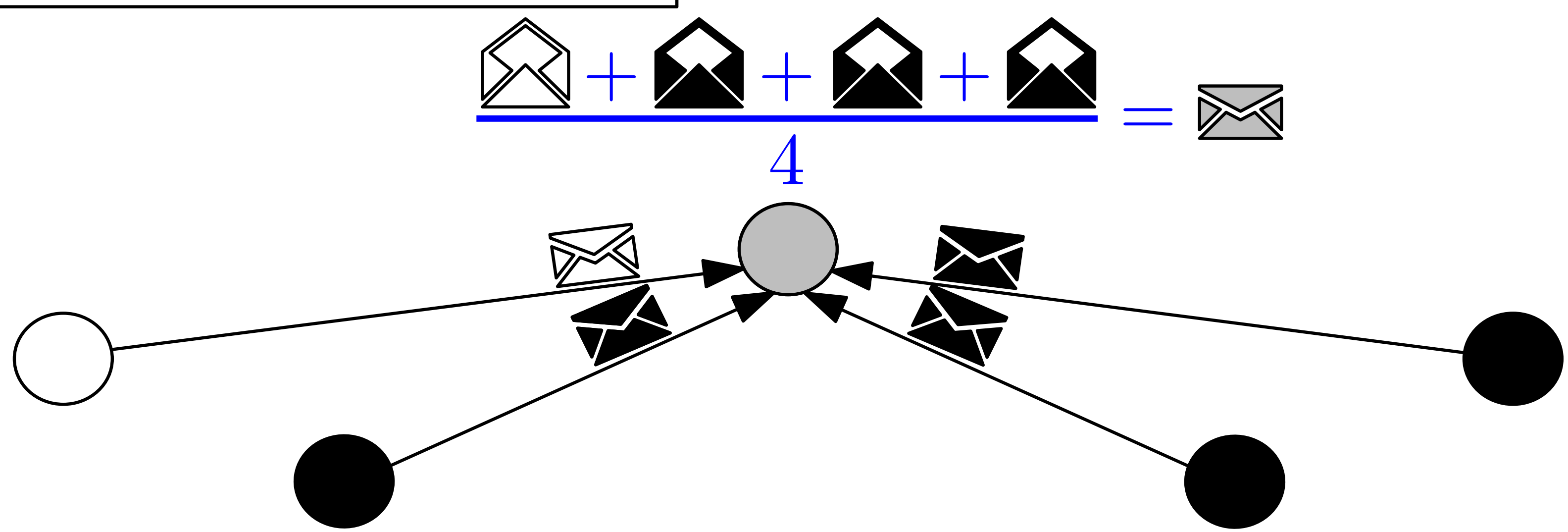


The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

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

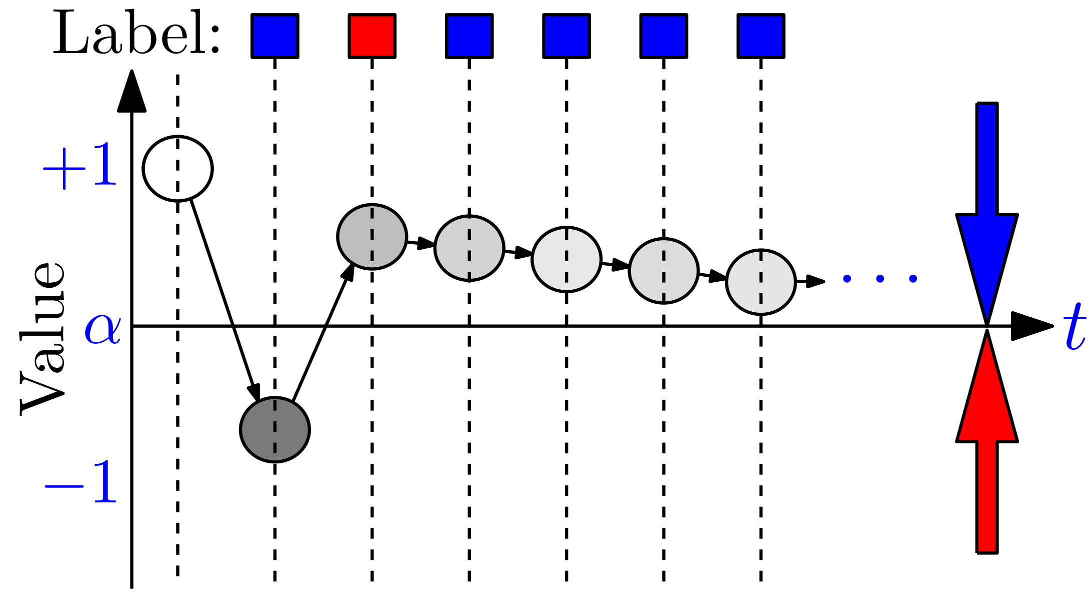


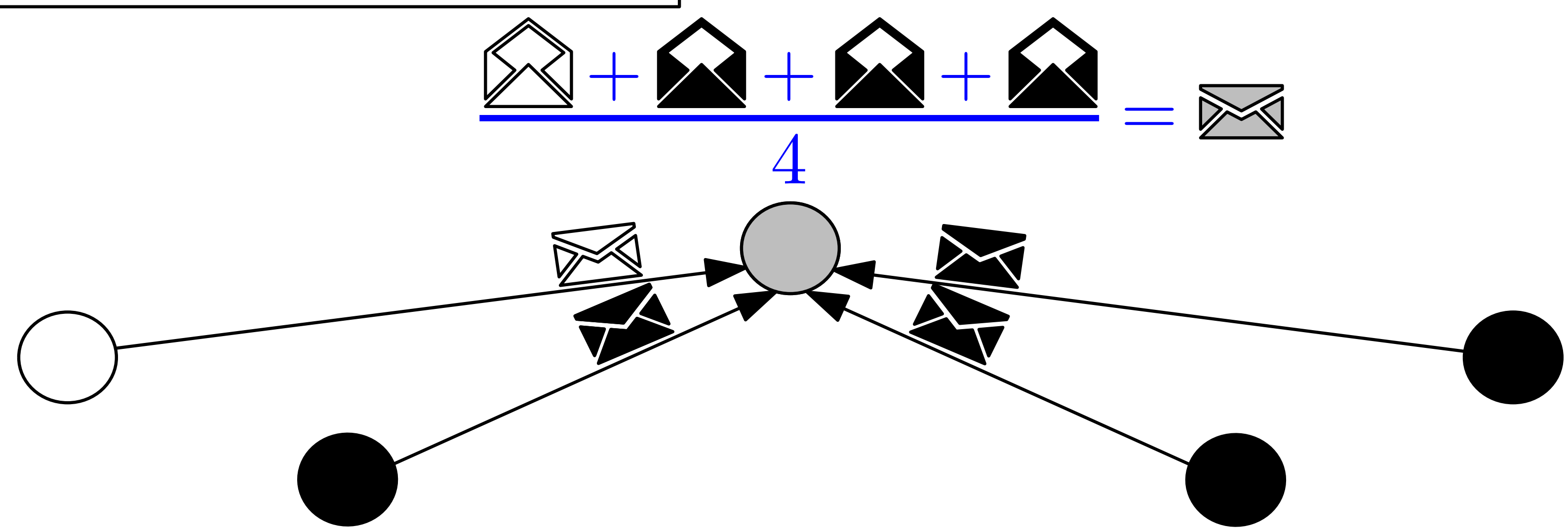


The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

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





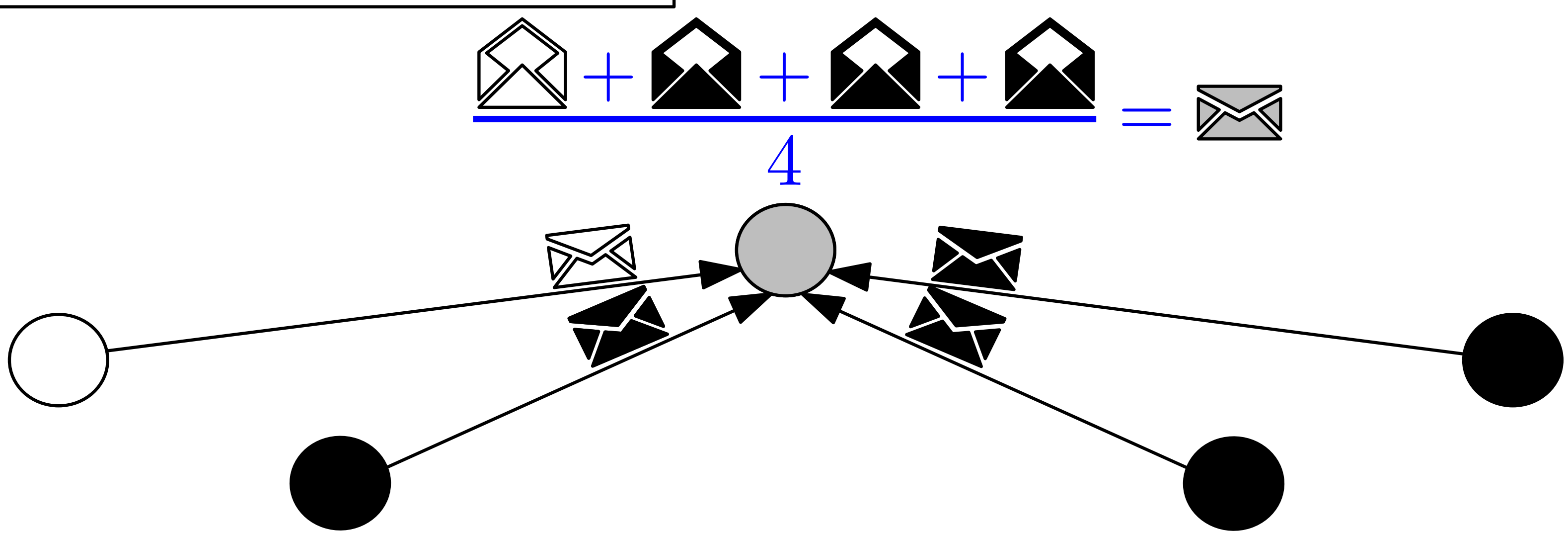
The Averaging Dynamics in the $\mathcal{LOCAL}$ Model

Al nodes at the same time:

- At $t = 0$, randomly pick value $x^{(t)} \in \{\mathbf{+1}, \mathbf{-1}\}$.
- Then, at each round
  - Set value $x^{(t)}$ to average of neighbors,
  - 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 $x^{(t)} \in \{+\mathbf{1}, \textbf{-1}\}$.
- Then, at each round
  - Set value $x^{(t)}$ to average of neighbors,
  - 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

$$\mathbf{x}^{(t)} = \begin{pmatrix} \circ \\ \bullet \\ \circ \\ \bullet \\ \bullet \end{pmatrix}$$

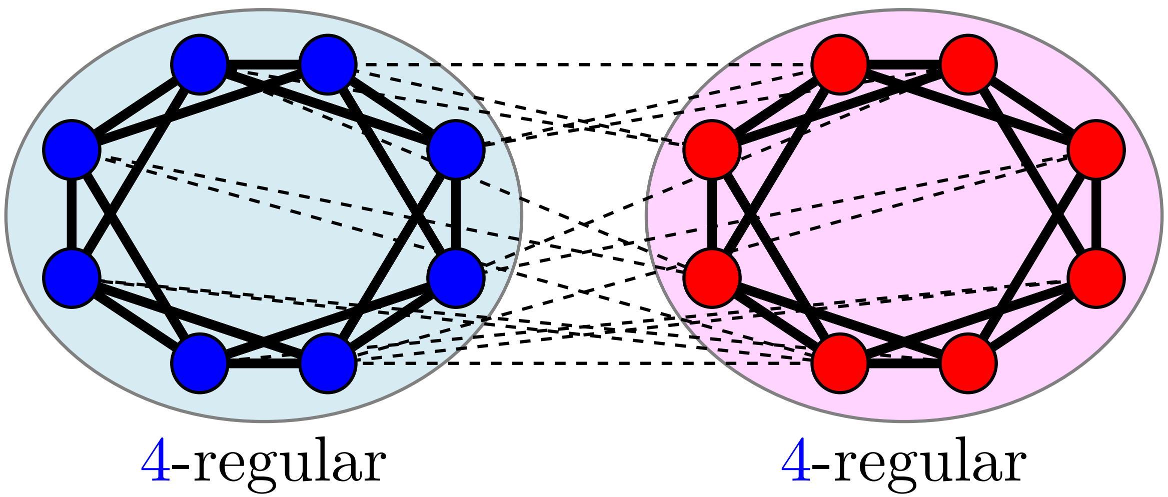
$$\boxed{\mathbf{x}^{(t)} = P \cdot \mathbf{x}^{(t-1)} = P^t \cdot \mathbf{x}^{(0)}}$$

P transition matrix of random walk

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.

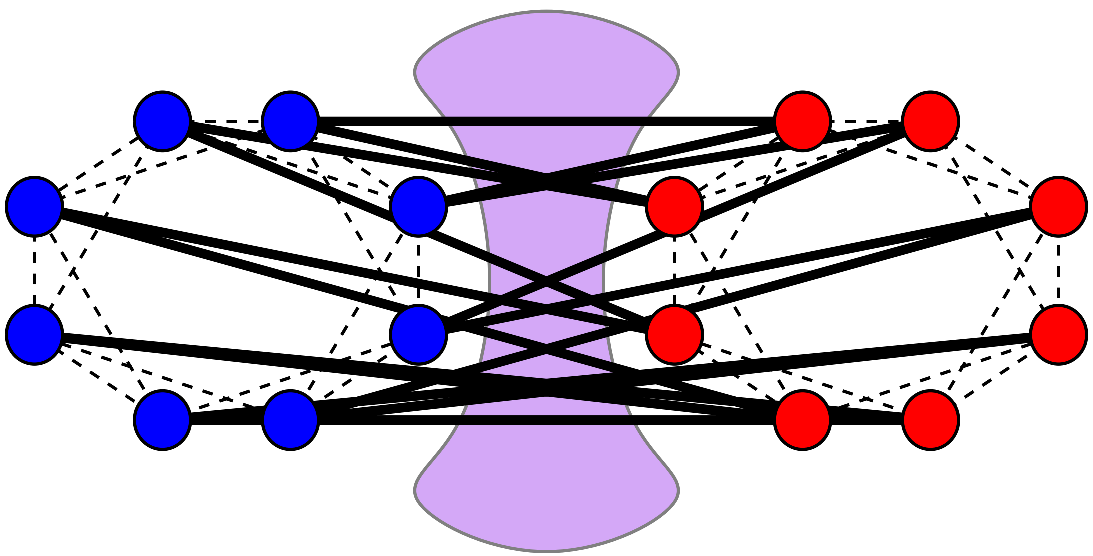
- $|V_1| = |V_2|$,
- $G\big|_{V_1}, G\big|_{V_2}$ ~ random a-regular graphs
- $G\big|_{E(V_1,V_2)}$ ~ random b-regular bipartite graph.



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.

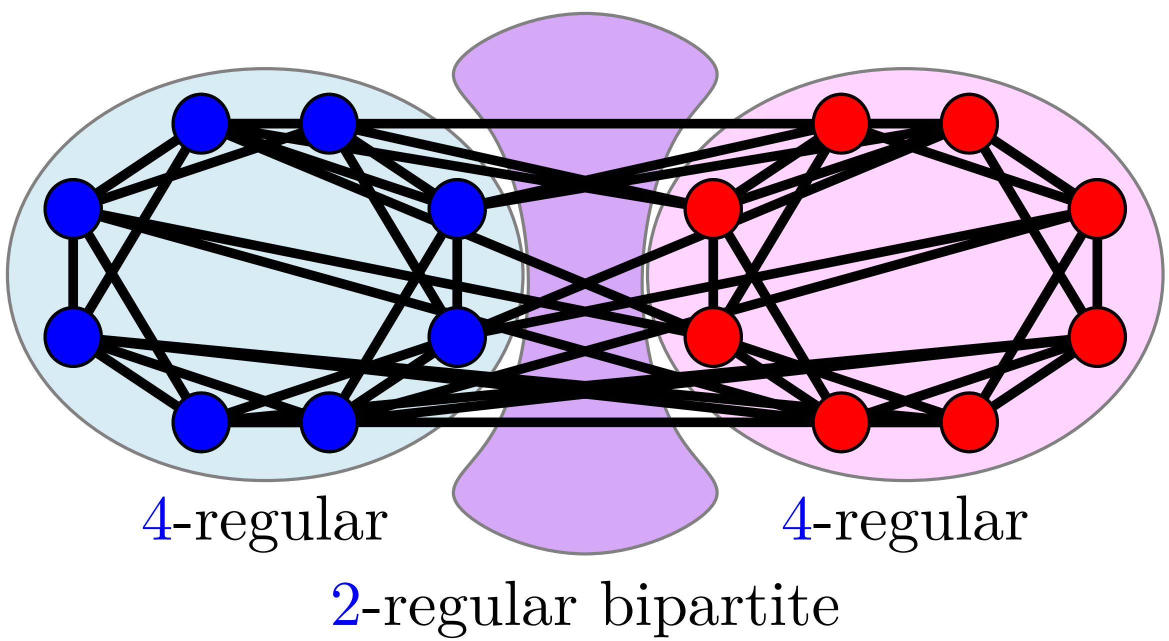


2-regular bipartite

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

$P \longrightarrow$ 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

$P \longrightarrow$ 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 $\mathbf{v}_1, ..., \mathbf{v}_n$ 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

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

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

Analysis on Regular SBM

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

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

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

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

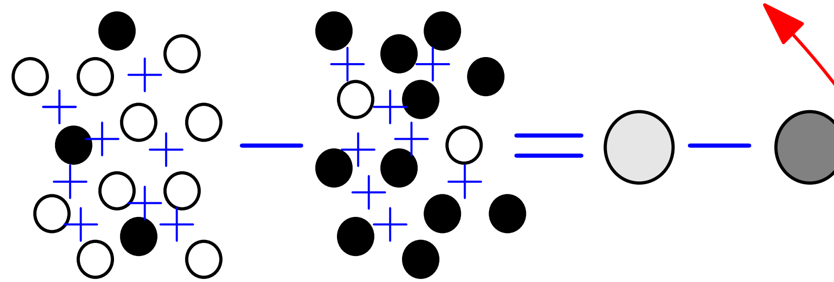
W.h.p. $\max\{\lambda_3, |\lambda_n|\}(1+\delta) < \frac{a-b}{a+b} = \lambda_2$, then

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

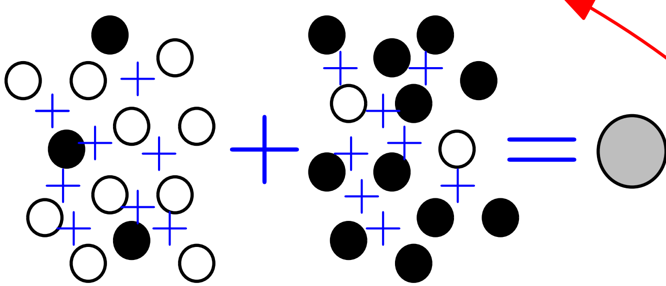
with $\|\mathbf{e}^{(t)}\| \leq (\max\{\lambda_3, |\lambda_n|\})^t \sqrt{n}$

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

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

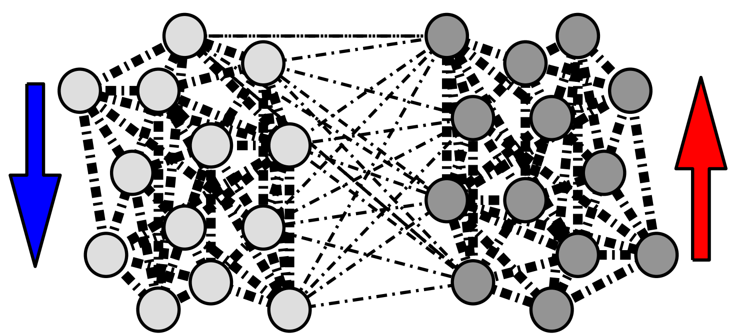
$$\mathbf{x}^{(t)} = \frac{1}{n}(\mathbf{1}^\intercal \mathbf{x}^{(0)})\mathbf{1} + \underbrace{\left(\frac{a-b}{a+b}\right)}_{=\lambda_2}{}^{t}\,\frac{1}{n}(\chi^\intercal \mathbf{x}^{(0)})\chi + \mathbf{e}^{(t)}$$

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

Analysis on Regular SBM

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

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



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

Roadmap

- Intro to Computational Dynamics
- Community Detection via Synchronous Averaging
- Community Detection via Asynchronous Averaging
- 2-Choices on Clustered Graphs & Evolution

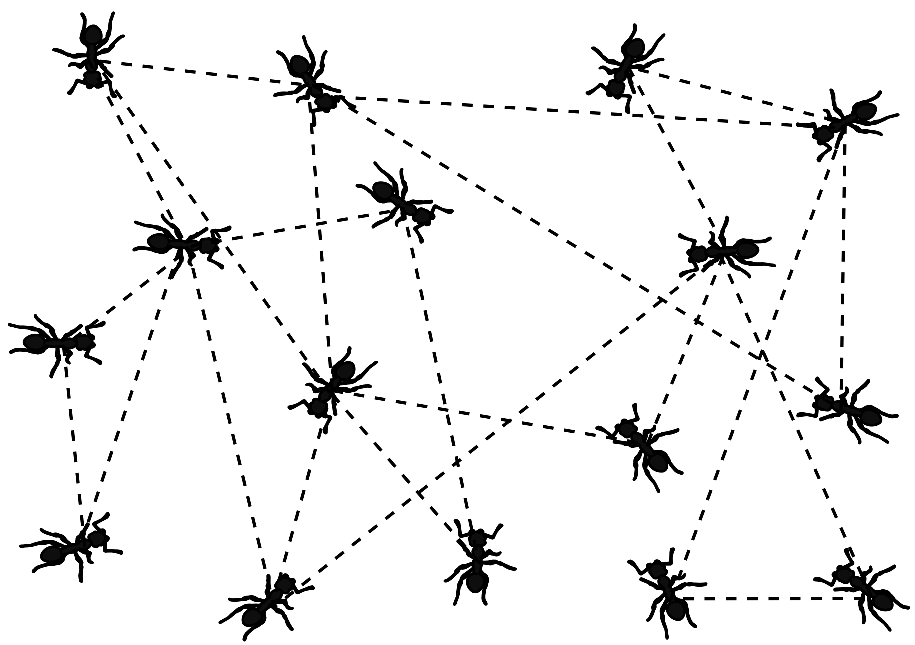
Communication Model: Population Protocol

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

Communication Model: Population Protocol

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

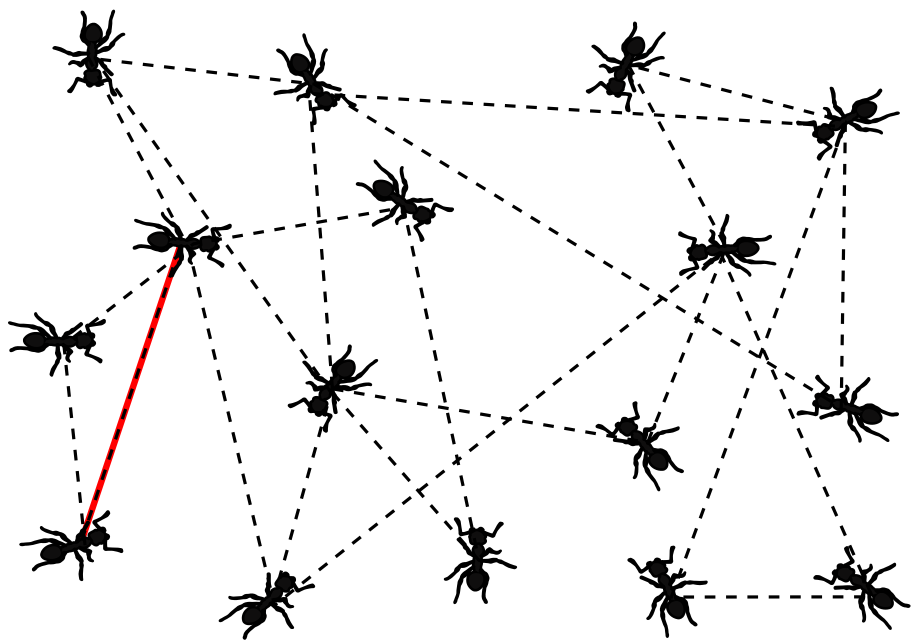
Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



Communication Model: Population Protocol

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

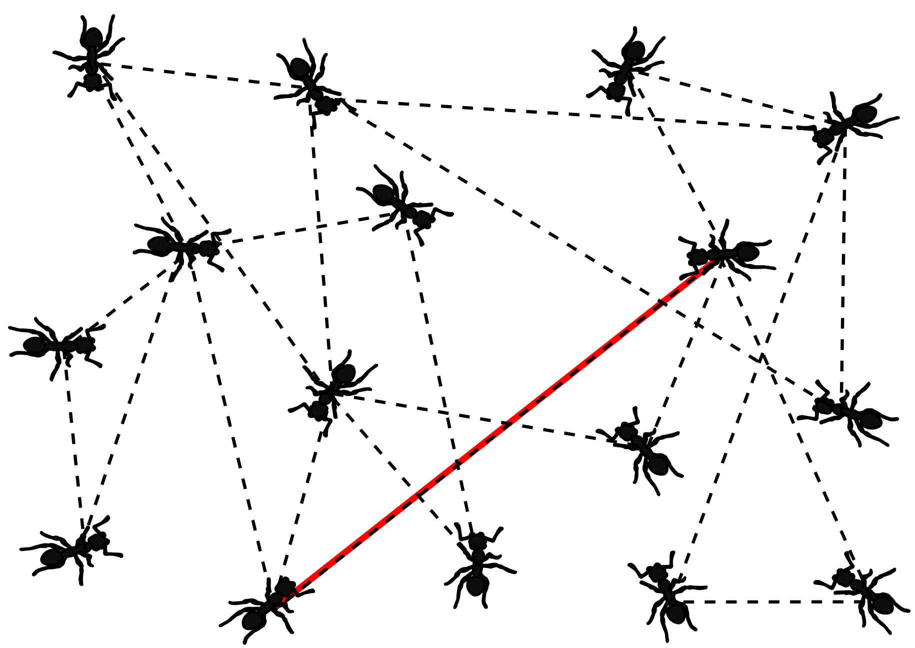
Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



Communication Model: Population Protocol

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

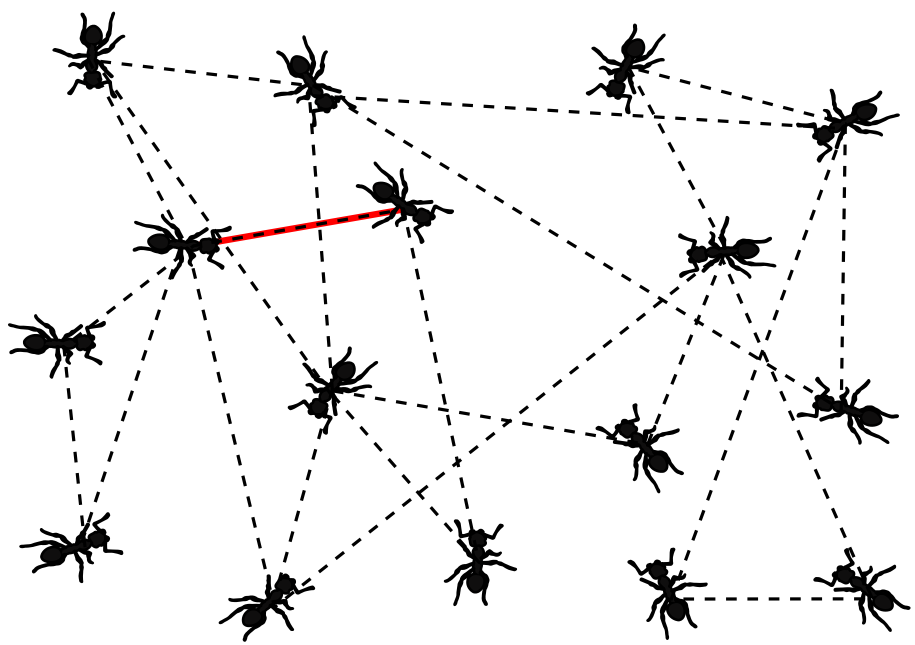
Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



Communication Model: Population Protocol

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

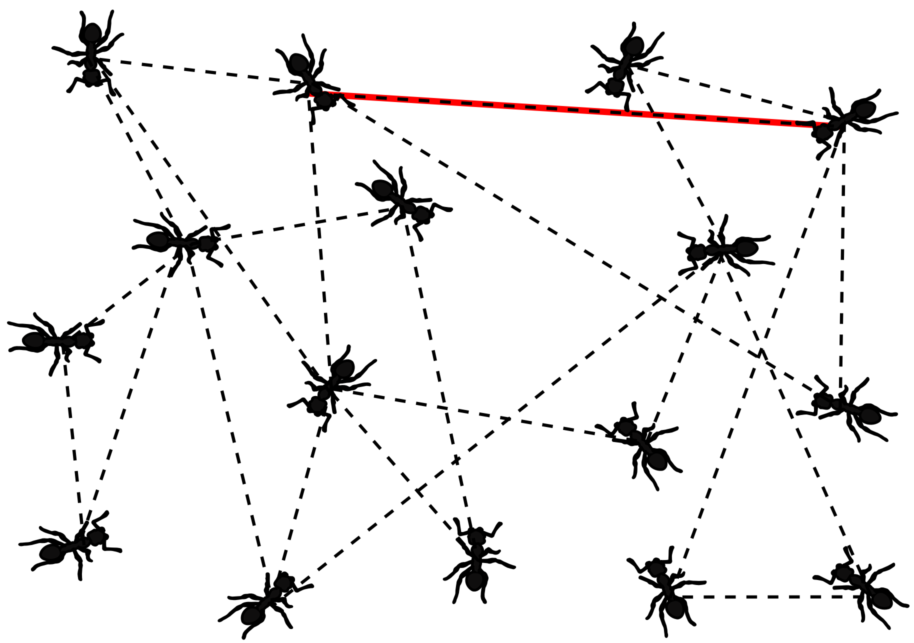
Population protocol: at each round a random edge is chosen and the two corresponding agent interact.



Communication Model: Population Protocol

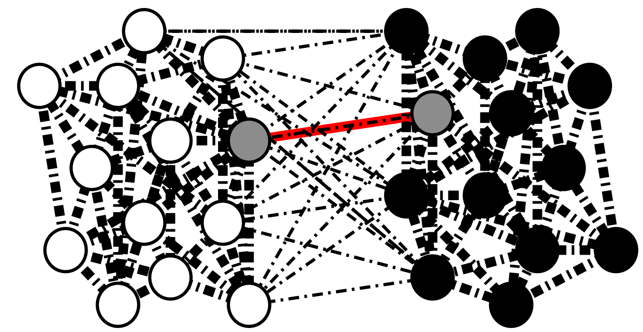
Averaging Dynamics in $\mathcal{LOCAL}$ Model:
$\mathcal{O}(d)$ messages per round :-(

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



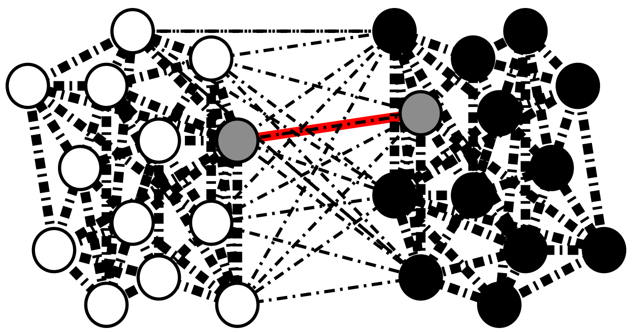
Sparsification of the Averaging Dynamics

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



Sparsification of the Averaging Dynamics

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

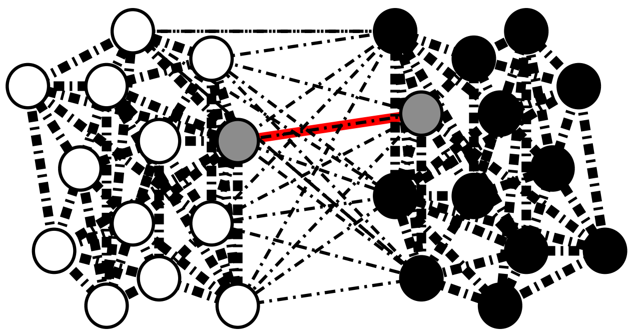


Can we *sparsify* the process?

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

Sparsification of the Averaging Dynamics

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



Can we *sparsify* the process?

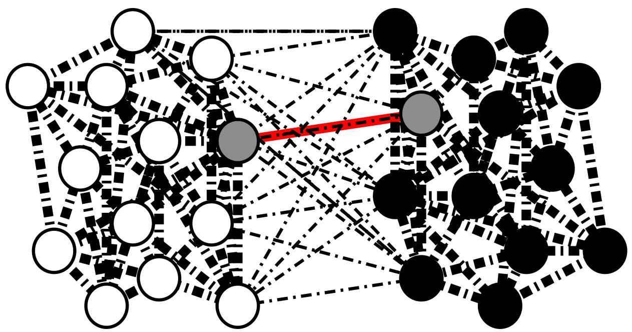
$\Longrightarrow$ Do averaging only over some *random* edges.

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

Random matrices!

Sparsification of the Averaging Dynamics

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



Can we *sparsify* the process?

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

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

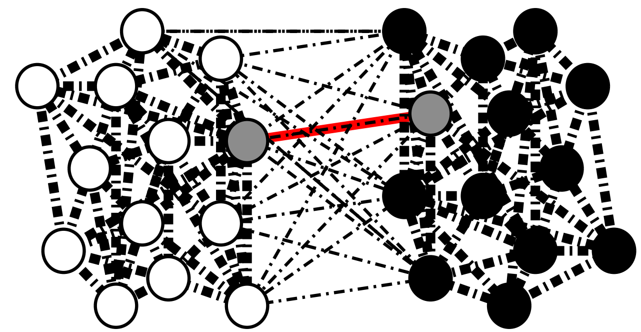
Random matrices!

Expected behavior:

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

Sparsification of the Averaging Dynamics

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



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:

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

Problem: can't use concentration tools for matrix *products*
(cfr. use of Matrix Freedman ineq. by Kathuria et al. 2020)

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

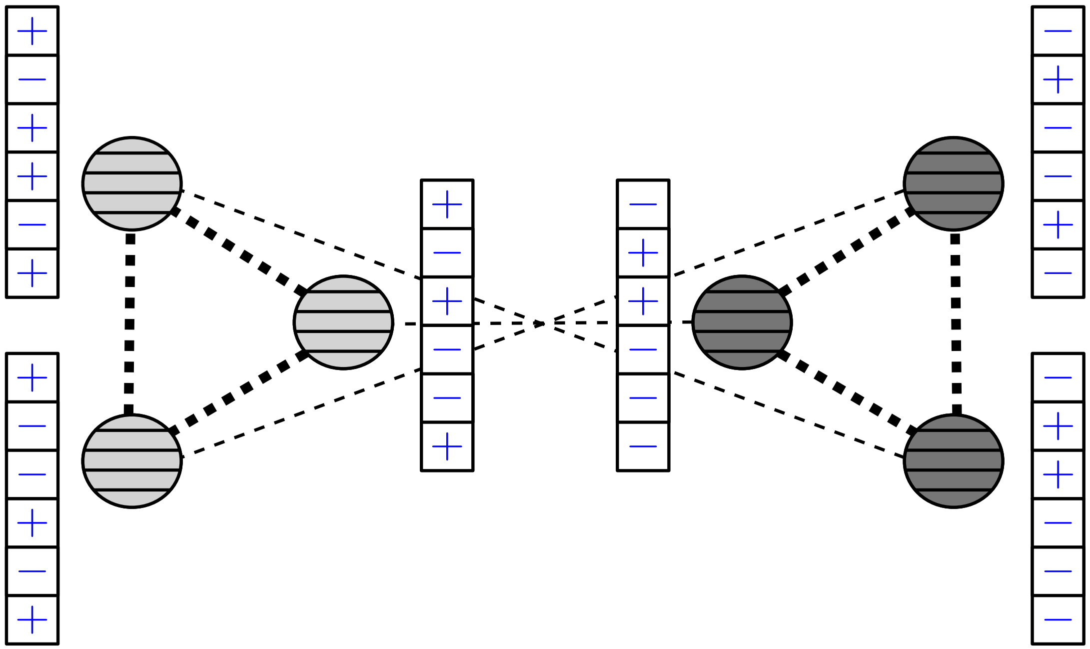
- the labels of nodes in the same community agree on at least $5/6$ entries, and
- the labels of nodes in different communities differ in more than $1/6$ entries.

Community Sensitive Labeling

Example:

> 2 different labels
$\Longrightarrow$ foes!

≤ 2 different labels
$\Longrightarrow$ friends!



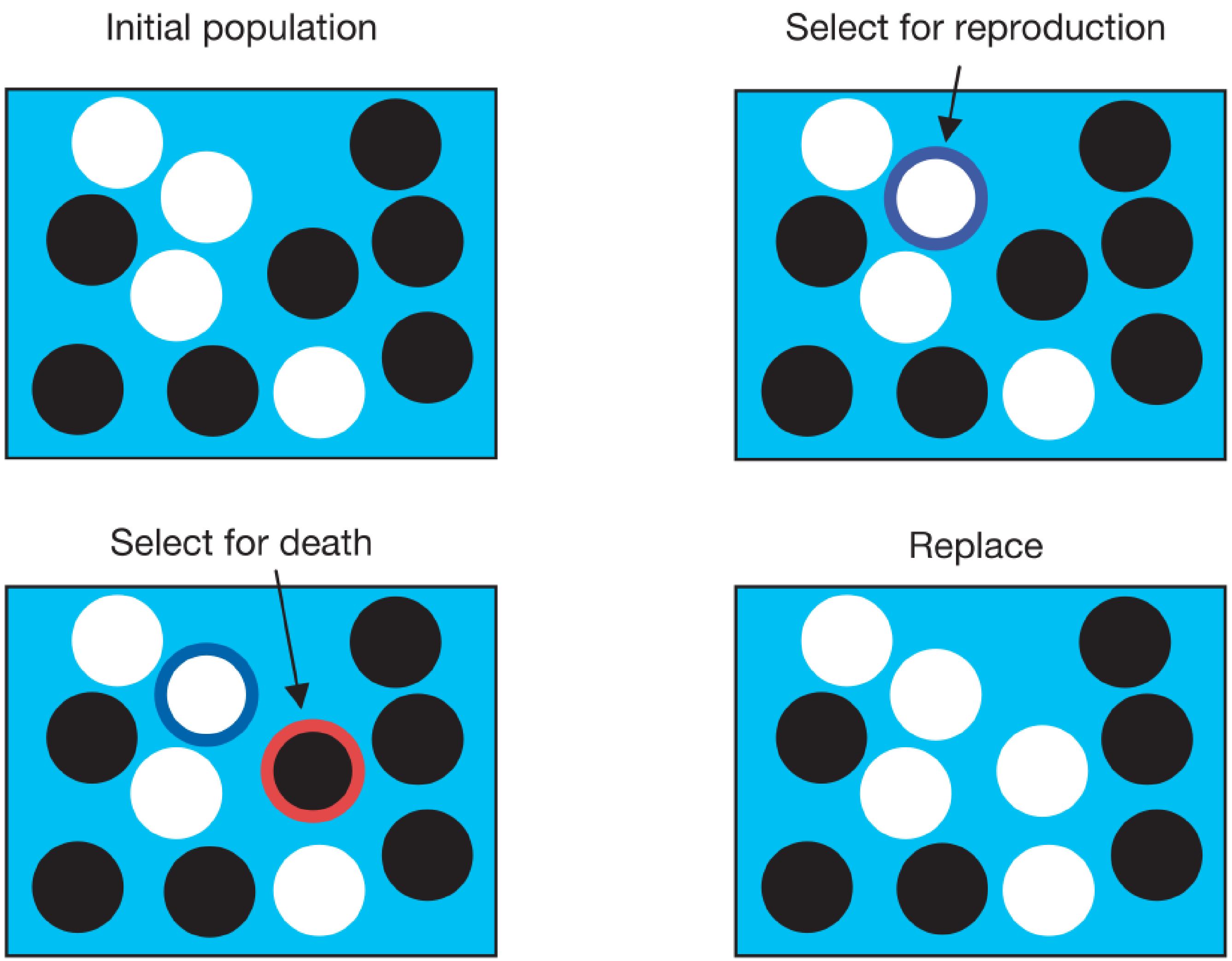
Warning: not a dynamics!

Roadmap

- Intro to Computational Dynamics
- Community Detection via Synchronous Averaging
- Community Detection via Asynchronous Averaging
- 2-Choices on Clustered Graphs & Evolution

Evolutionary Dynamics on Graphs

[Lieberman, Hauert & Nowak, Nature '05]:

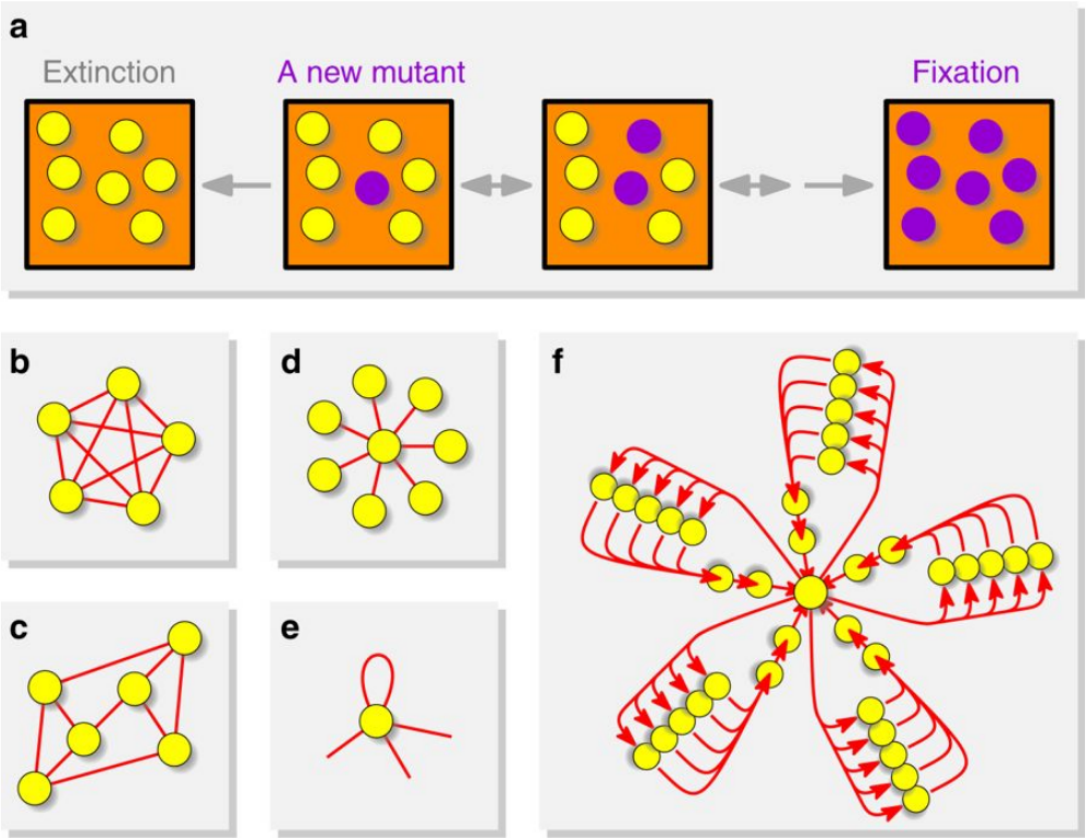


A node is selected randomly according to its fitness and it replaces a random neighbor

Picture from Lieberman et al.

The Moran Process and Fixation Probability

[Giakkoupis '16, Galanis et al. J. ACM '17, Goldberg et al.x2 '18, Pavlogiannis et al. Comm. Bio. '18]: Probability that a mutant with fitness r conquers a population with fitness 1 on a family of graphs $\{G_n\}_n$. Are there families G_n with probability $1 - o_n(1)$?



Picture from Pavlogiannis et al.

The Speed of Speciation

The Moran process doesn't provide an explanation for *speciation*

"What is needed now is a shift in focus to identifying more general rules and patterns in the dynamics of speciation. The crucial step in achieving this goal is the development of simple and general dynamical models that can be studied not only numerically but analytically as well. [...] Speciation is expected to be triggered by changes in the environment. Once genetic changes underlying speciation start, they go to completion very rapidly."

[Gavrilets, Evolution '03]

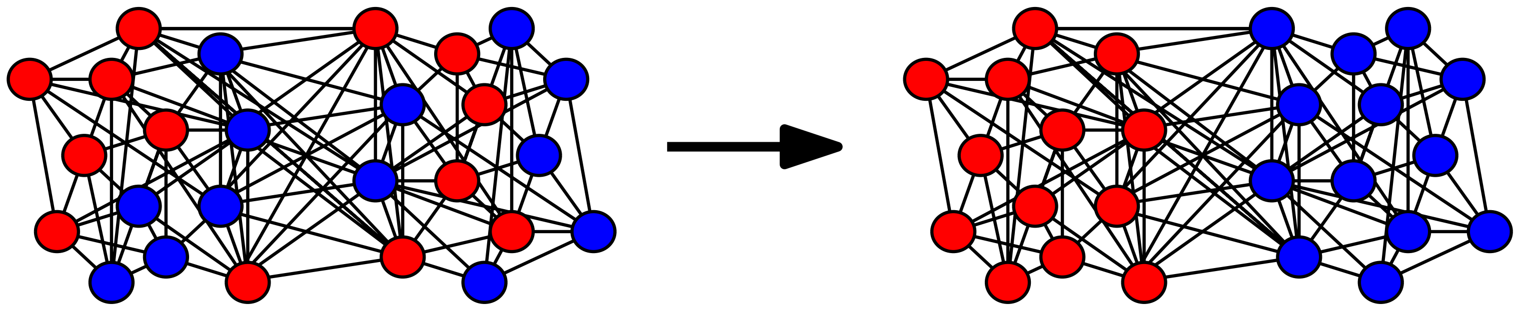
The Speed of Speciation

The Moran process doesn't provide an explanation for *speciation*

"What is needed now is a shift in focus to identifying more general rules and patterns in the dynamics of speciation. The crucial step in achieving this goal is the development of simple and general dynamical models that can be studied not only numerically but analytically as well. [...] Speciation is expected to be triggered by changes in the environment. Once genetic changes underlying speciation start, they go to completion very rapidly."

[Gavrilets, Evolution '03]

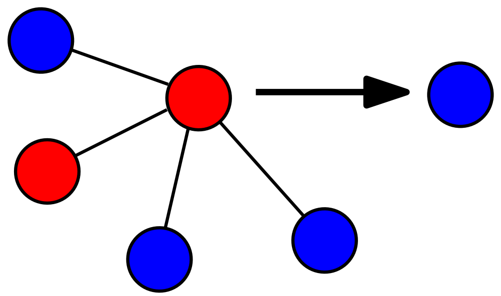
Problem: A simple evolutionary-graph-theoretic proof of principle for speciation.



y-Degree Majority Dynamics

Node gets color x with probability

$$\left(\frac{\#\text{neighbors with col. } x}{\text{degree}}\right)^y$$



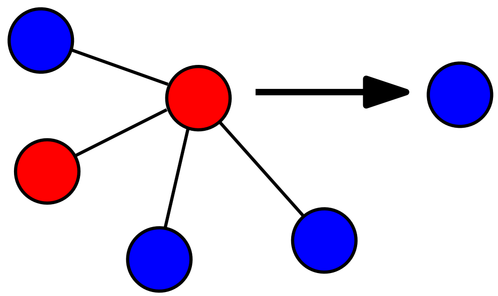
$y = 1 \implies$ Voter Dynamics (Moran Process)

$y = 2 \implies$ 2-Choices Dynamics

y-Degree Majority Dynamics

Node gets color x with probability

$$\left(\frac{\#\text{neighbors with col. } x}{\text{degree}}\right)^y$$



$y = 1 \implies$ Voter Dynamics (Moran Process)
$y = 2 \implies$ 2-Choices Dynamics

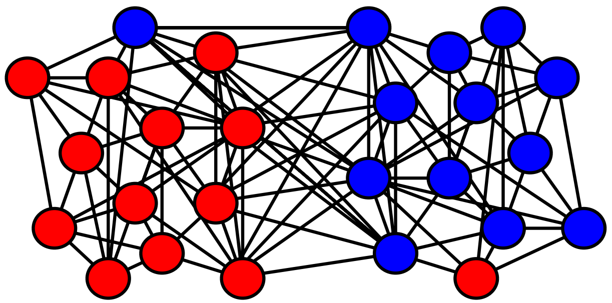
[Cooper et al.x3, ICALP'14, DISC'15, DISC'17]: 2-Choice Dynamics can be related to the *spectral structure* of the graph!

$$\sum_{x \in V} \left(\frac{B(x)}{d}\right)^2 = \|P\mathbf{1}_B\|_2^2 \leq \frac{B^2}{n} + \lambda^2 B.$$

$B(x)$ blue neighbors of x, P trans. matrix of graph, $\mathbf{1}_B$ indicator vector of blue-col. nodes, B overall number of blue-col. nodes, λ second-largest eigenvalue of P

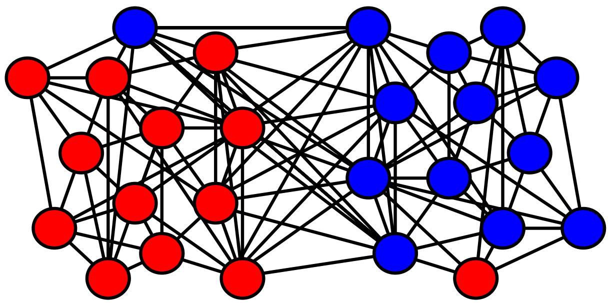
Metastability of 2-Choices Dynamics

Theorem [Cruciani, N., Scornavacca, AAAI'19].
G d-regular graph divided in 2 *clusters*, where cut is a b-regular bipartite graph. Each node initially blue or red u.a.r. If $b/d = \mathcal{O}(1/\sqrt{n})$ and spectral radius of clusters is $\mathcal{O}(n^{-\frac{1}{4}})$, then with prob. $\Omega(1)$, after $\mathcal{O}(\log n)$ *time*, clusters are *almost-monochromatic*, with *different colors*, and remains so for $n^{\Omega(1)}$ *time* w.h.p.



Metastability of 2-Choices Dynamics

Theorem [Cruciani, N., Scornavacca, AAAI'19].
G d-regular graph divided in 2 *clusters*, where cut is a b-regular bipartite graph. Each node initially blue or red u.a.r. If $b/d = \mathcal{O}(1/\sqrt{n})$ and spectral radius of clusters is $\mathcal{O}(n^{-\frac{1}{4}})$, then with prob. $\Omega(1)$, after $\mathcal{O}(\log n)$ *time*, clusters are *almost-monochromatic*, with *different colors*, and remains so for $n^{\Omega(1)}$ *time* w.h.p.

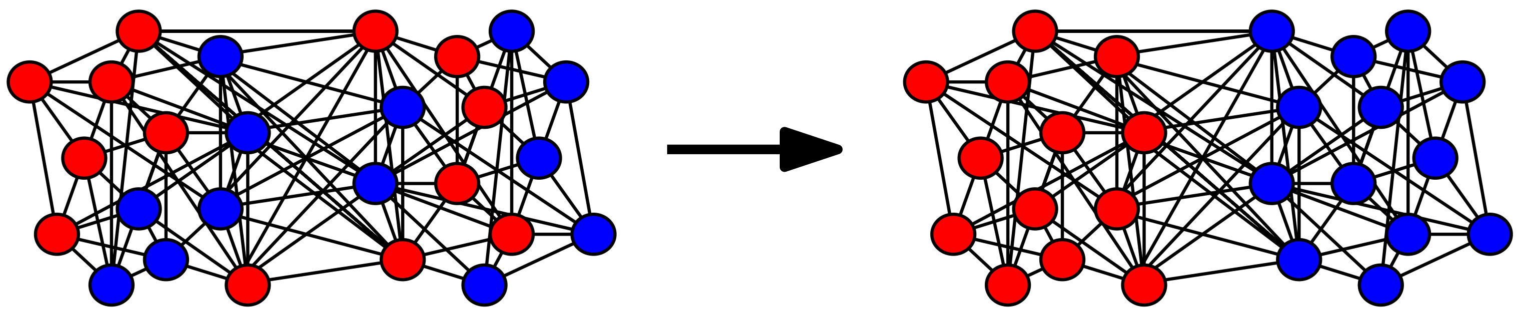


Corollary: LPA. First analytical result on a sparse Label Propagation Algorithm (class of clustering heuristics).

Conclusions

Computational dynamics have a rich interaction with the underlying graph topology:

- synchronous averaging dyn. on SBM
- averaging pop. protocol on SBM
- 2-Choices dynamics on SBM



Open problems. New techniques for

- Analyze majority on non-expander graphs
- Tighter analysis of 2-Choices on RSBM
- ....

Thank You!