On the Computational Power of Simple Dynamics

Emanuele Natale

SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome
December 21, 2016

Communication in Biological Systems

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

Communication in Biological Systems

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

Schools of fish
[Sumpter et al. '08]

Communication in Biological Systems

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

Schools of fish
[Sumpter et al. '08]

Insects colonies
[Franks et al. '02]

Communication Model

Animal communication:

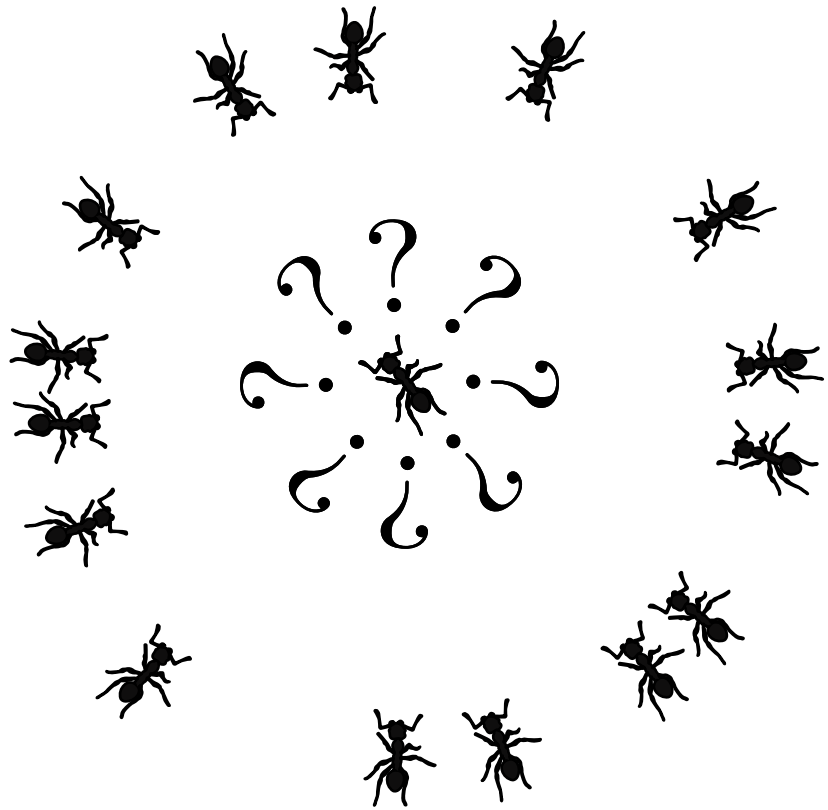
- Chaotic
- Anonymous
- Parsimonious
- Uni-directional (Passive/Active)
- Noisy

Communication Model

Animal communication:

- Chaotic
- Anonymous
- Parsimonious
- Uni-directional (Passive/Active)
- Noisy

$\mathcal{PUSH}(h, \ell)$ **model** [**Demers '88**]: at each round each agent can *send a ℓ-bit message* to h other agents chosen independently and uniformly at random.

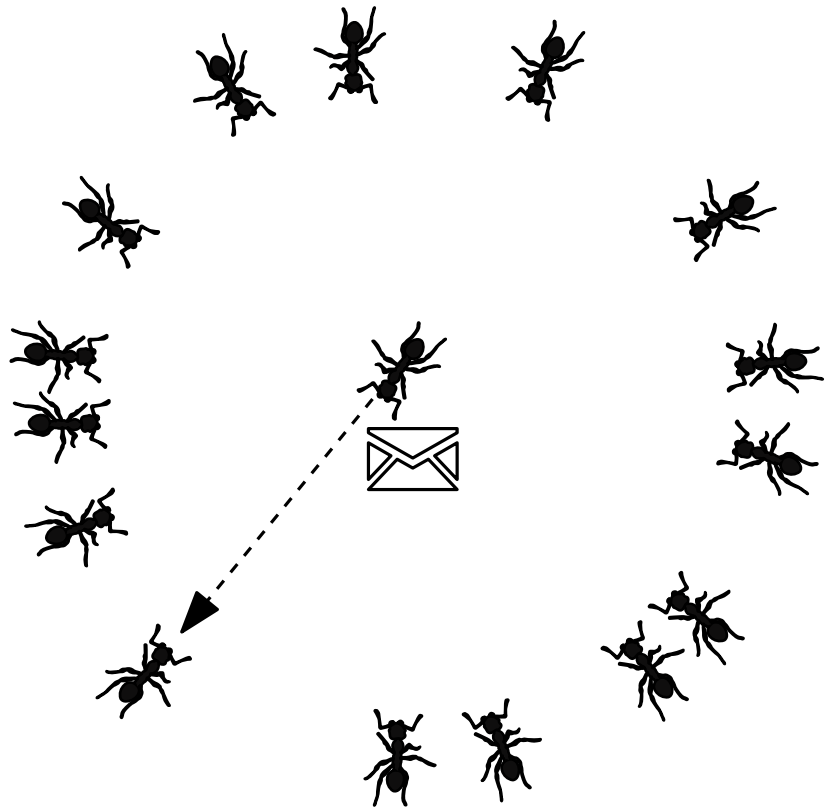


Communication Model

Animal communication:

- Chaotic
- Anonymous
- Parsimonious
- Uni-directional (~~Passive~~/Active)
- Noisy

$\mathcal{PUSH}(h, \ell)$ **model** [**Demers '88**]: at each round each agent can *send a ℓ-bit message* to h other agents chosen independently and uniformly at random.

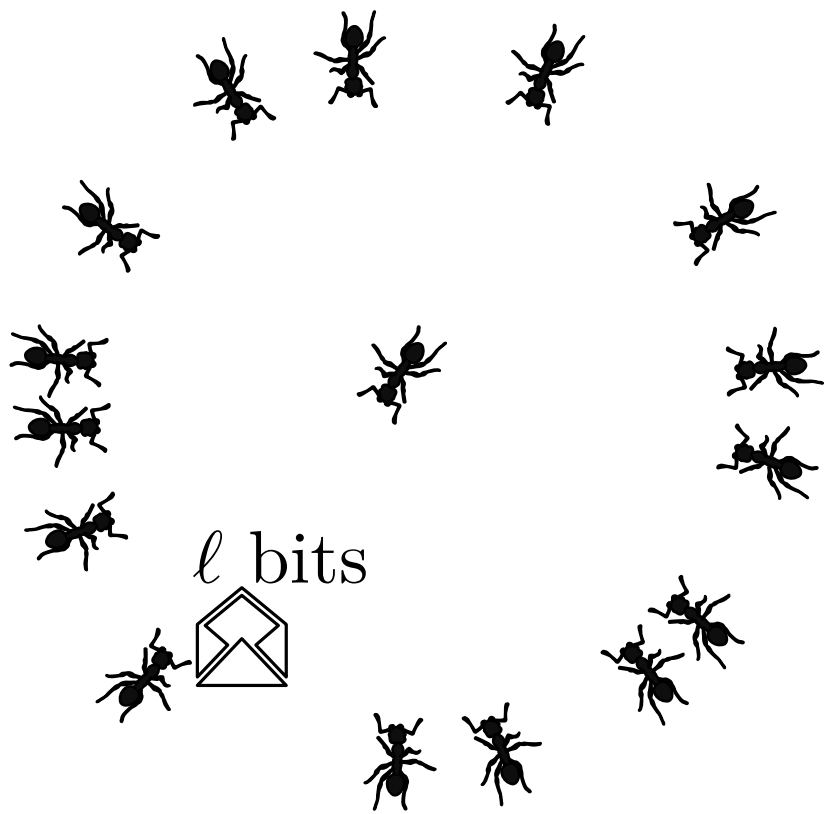


Communication Model

Animal communication:

- ✓ Chaotic
- ✓ Anonymous
- ✓ Parsimonious
- Uni-directional (~~Passive~~/Active ✓)
- Noisy

$\mathcal{PUSH}(h, \ell)$ **model** [**Demers '88**]: at each round each agent can *send a ℓ-bit message* to h other agents chosen independently and uniformly at random.

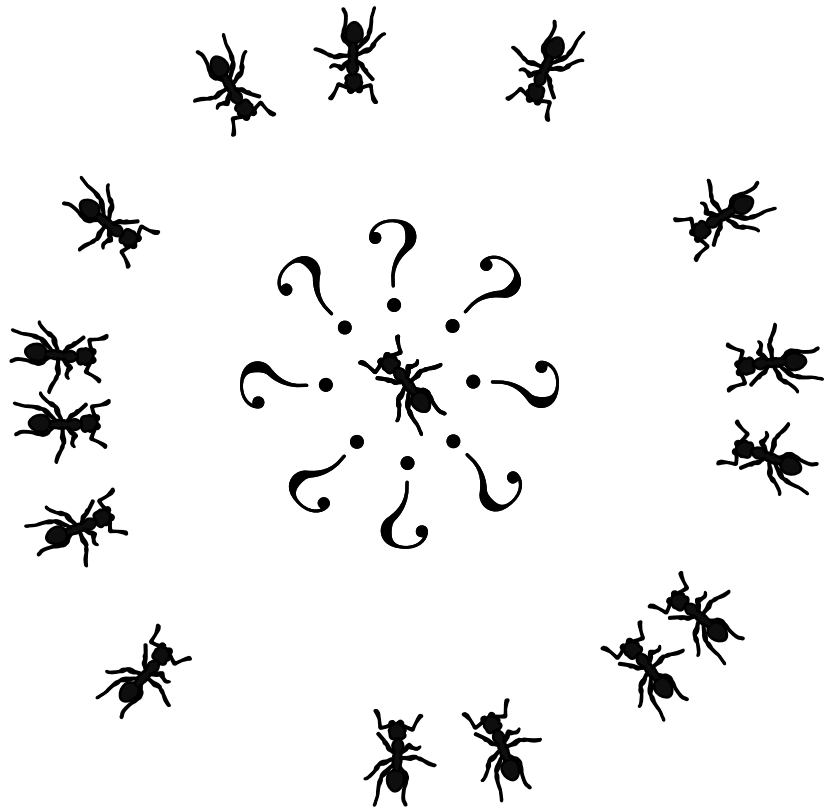


Communication Model

Animal communication:

- ✓ Chaotic
- ✓ Anonymous
- ✓ Parsimonious
- Uni-directional (Passive/Active) ✓ ✗
- Noisy

$\mathcal{PULL}(h, \ell)$ **model** [**Demers '88**]: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.



Communication Model

Animal communication:

- ✓ Chaotic
- ✓ Anonymous
- ✓ Parsimonious
- Uni-directional (Passive/~~Active~~) ✓
- Noisy

$\mathcal{PULL}(h, \ell)$ **model** [**Demers '88**]: at each round each agent can *observe* h other agents chosen independently and uniformly at random, and *shows* ℓ *bits* to her observers.

Dynamics

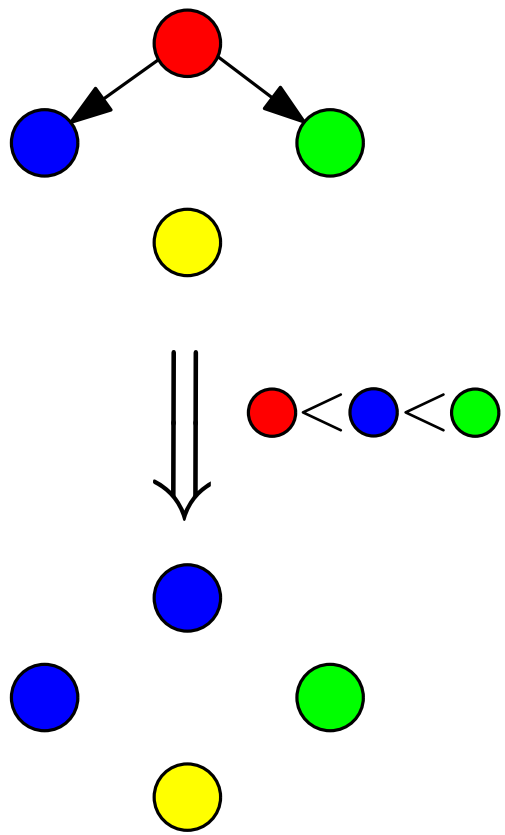
(informal) *Very simple* distributed algorithms: For every graph, agent and round, states are updated according to fixed rule of current state and symmetric function of states of neighbors.

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

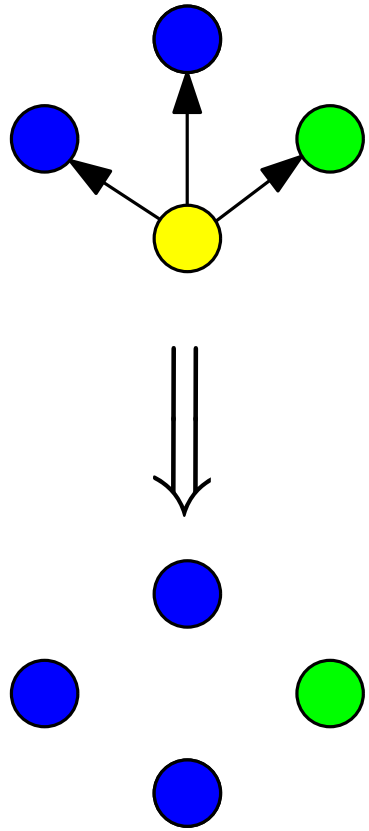


Dynamics

(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

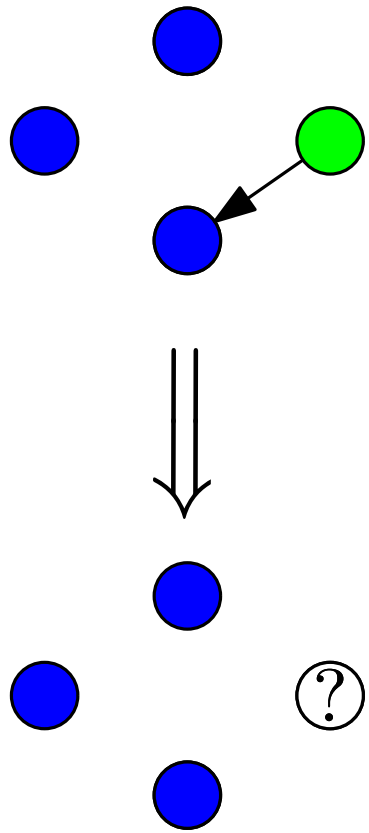


Dynamics

(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

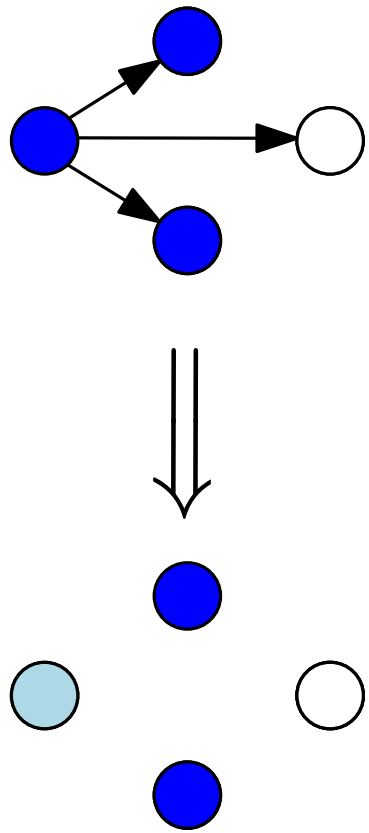


Dynamics

(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).

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).

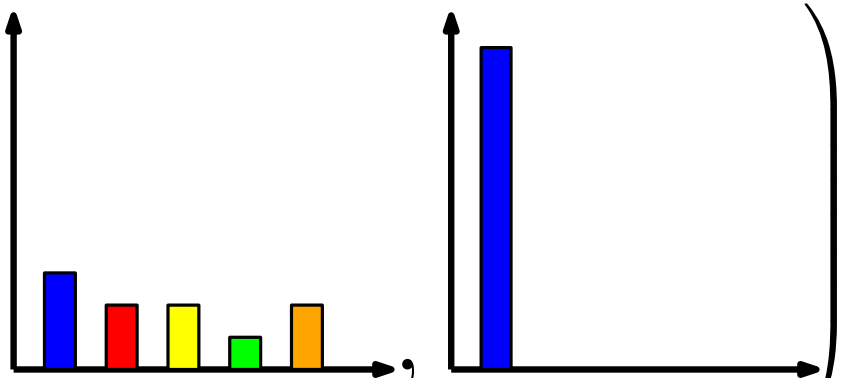
Computing the Majority

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 $\text{poly}(k)$. 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 $\tilde{\Theta}(\text{md}(\mathbf{c}))$ rounds w.h.p.

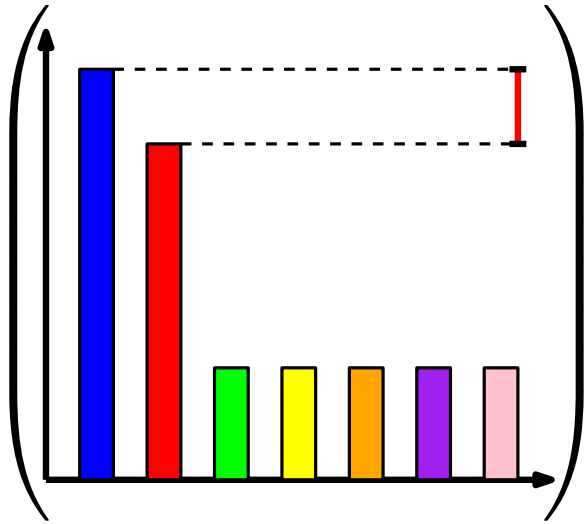
A Global Measure of Bias

$$\mathrm{md}(\mathbf{c^{(0)}}) := \sum_{i=1}^{k} \left(\frac{c_i^{(0)}}{c_{maj}^{(0)}} \right)^2 = 1 + \mathcal{D}\left(\right.$$

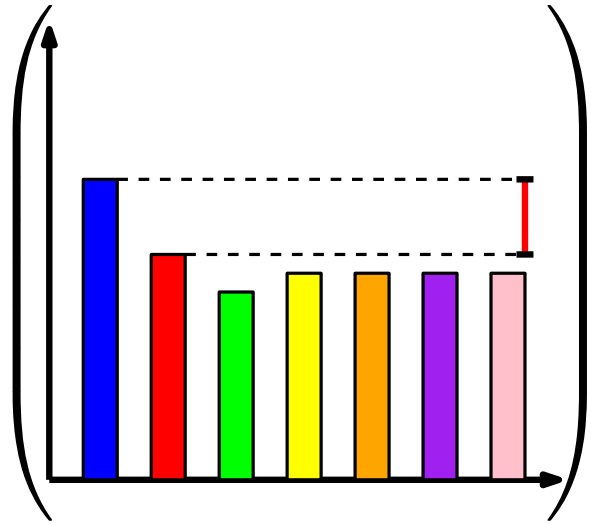


$$\left. \right)$$

$$1 \leq \mathrm{md}\left(\right.$$



$$\left. \right) \ll \mathrm{md}\left(\right.$$

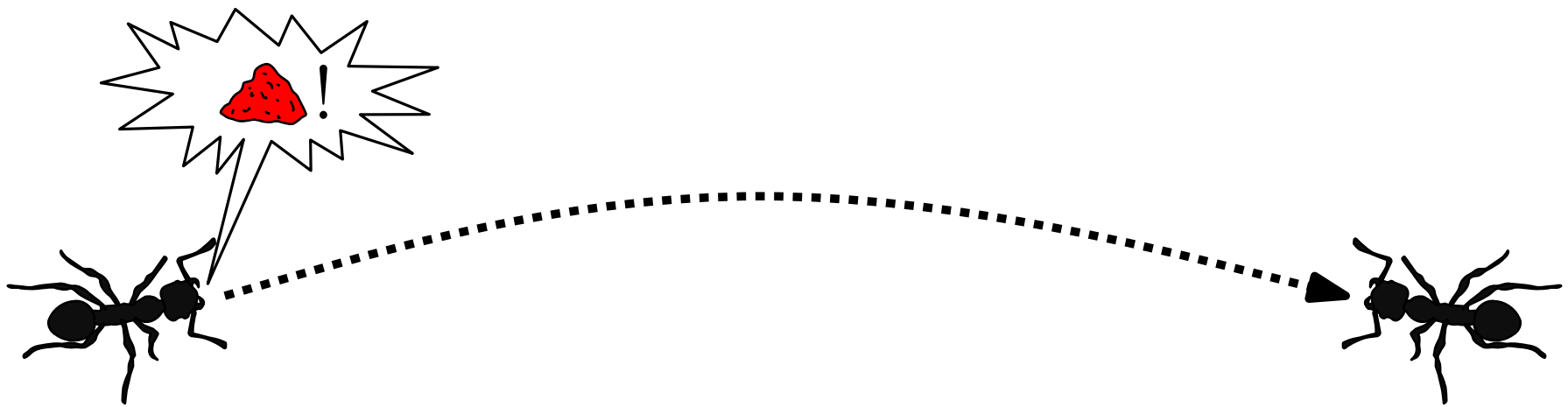


$$\left. \right) \leq k$$

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 $\tilde{\Theta}(\mathrm{md}(\mathbf{c}))$ rounds w.h.p.

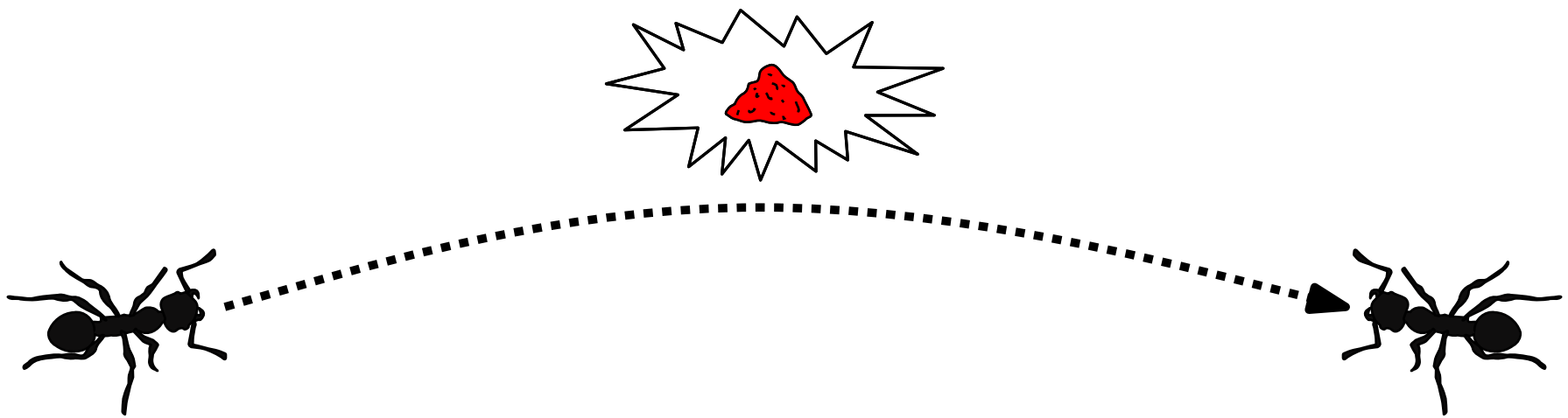
Bit Dissemination with Noisy Communication

Noisy $\mathcal{PULL}$ model: messages are randomly corrupted



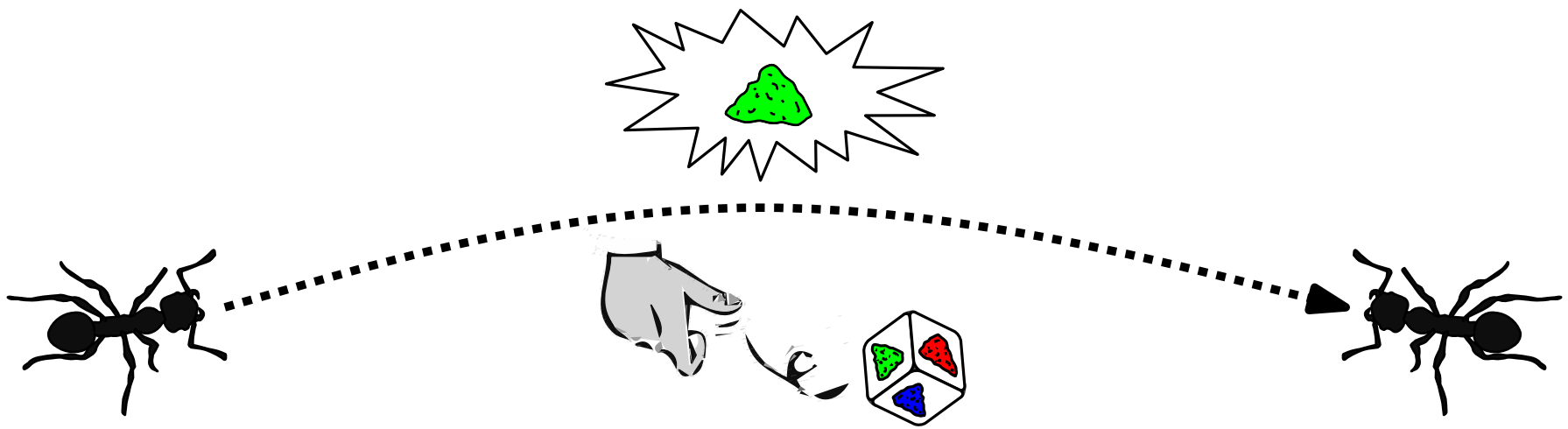
Bit Dissemination with Noisy Communication

Noisy $\mathcal{PULL}$ model: messages are randomly corrupted



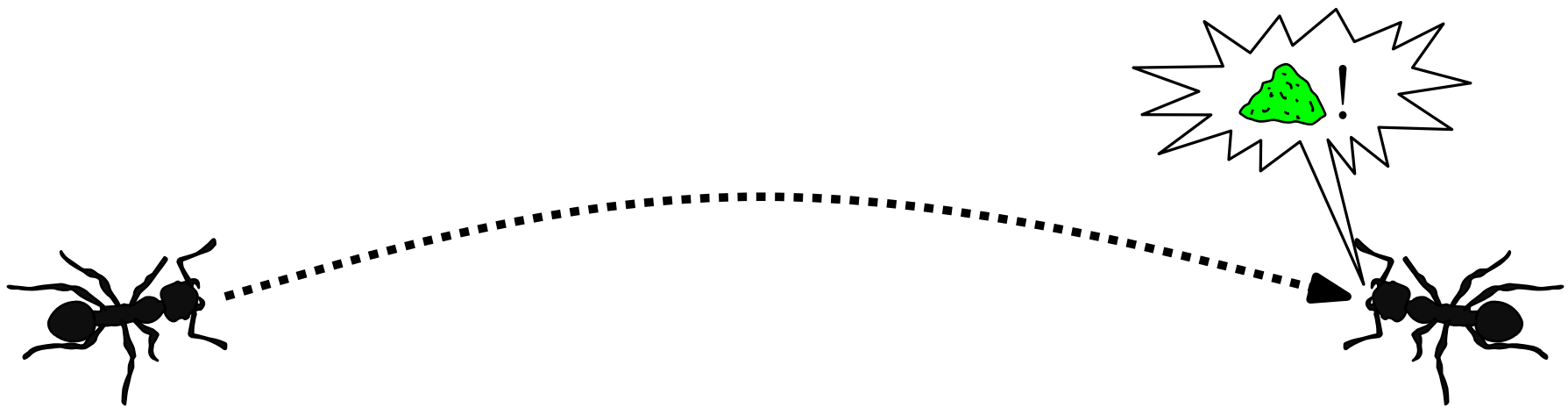
Bit Dissemination with Noisy Communication

Noisy $\mathcal{PULL}$ model: messages are randomly corrupted



Bit Dissemination with Noisy Communication

Noisy $\mathcal{PULL}$ model: messages are randomly corrupted



Bit Dissemination with Noisy Communication

Noisy $\mathcal{PULL}$ model: messages are randomly corrupted

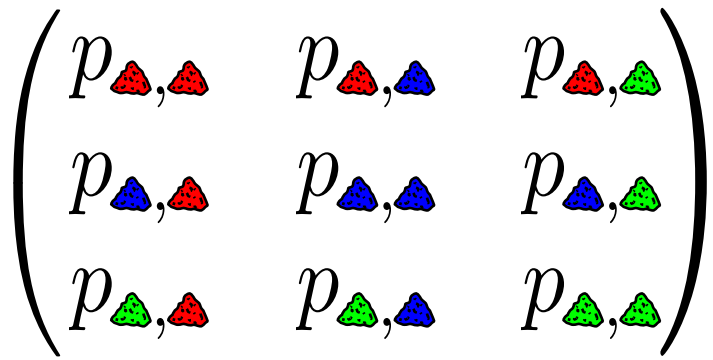
Noise Matrix:

$$\sim P := \begin{pmatrix} p_{\blacktriangle,\blacktriangle} & p_{\blacktriangle,\blacktriangle} & p_{\blacktriangle,\blacktriangle} \\ p_{\blacktriangle,\blacktriangle} & p_{\blacktriangle,\blacktriangle} & p_{\blacktriangle,\blacktriangle} \\ p_{\blacktriangle,\blacktriangle} & p_{\blacktriangle,\blacktriangle} & p_{\blacktriangle,\blacktriangle} \end{pmatrix}$$

Bit Dissemination with Noisy Communication

Noisy $\mathcal{PULL}$ model: messages are randomly corrupted

Noise Matrix:

$\sim P :=$ 

Noise affects animal communication, but animals cannot use *coding theory*...

FHK '14: Natural rules efficiently solve rumor spreading and majority consensus despite noise when

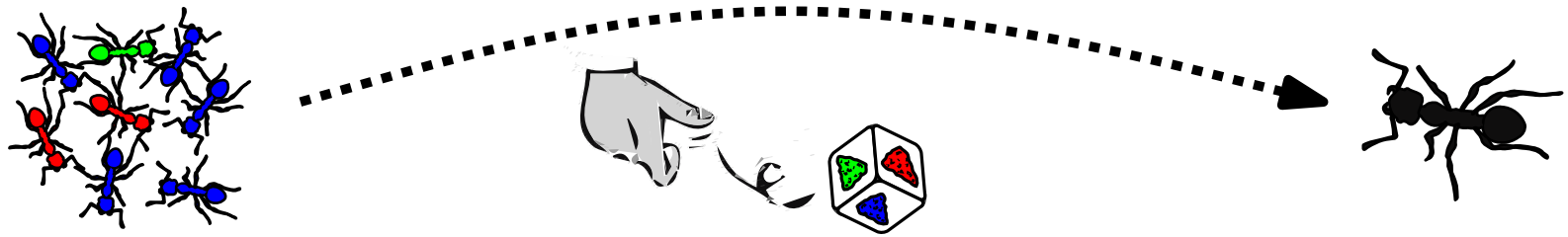
$$P = \begin{pmatrix} 1/2+\varepsilon & 1/2-\varepsilon \\ 1/2-\varepsilon & 1/2+\varepsilon \end{pmatrix}$$

Majority-Preserving Matrix

Random sender in conf. $\mathbf{c}$

Noise acting according to matrix P

Message distributed as $c \cdot P$



Majority-Preserving Matrix

Configuration $\mathbf{c} := (\#\,/n, \#\,/n, \#\,/n)$

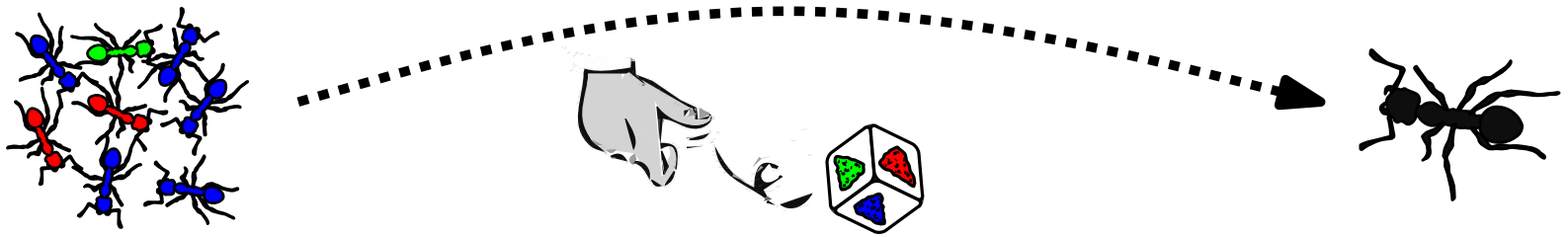
δ-majority-biased configuration w.r.t. :

$$\#\,/n - \#\,/n > \delta$$
$$\#\,/n - \#\,/n > \delta$$

Random sender in conf. $\mathbf{c}$

Noise acting according to matrix P

Message distributed as $c \cdot P$



Majority-Preserving Matrix

Configuration $\mathbf{c} := (\#\,/n, \#\,/n, \#\,/n)$

δ-majority-biased configuration w.r.t. :

$$\#\,/n - \#\,/n > \delta$$
$$\#\,/n - \#\,/n > \delta$$

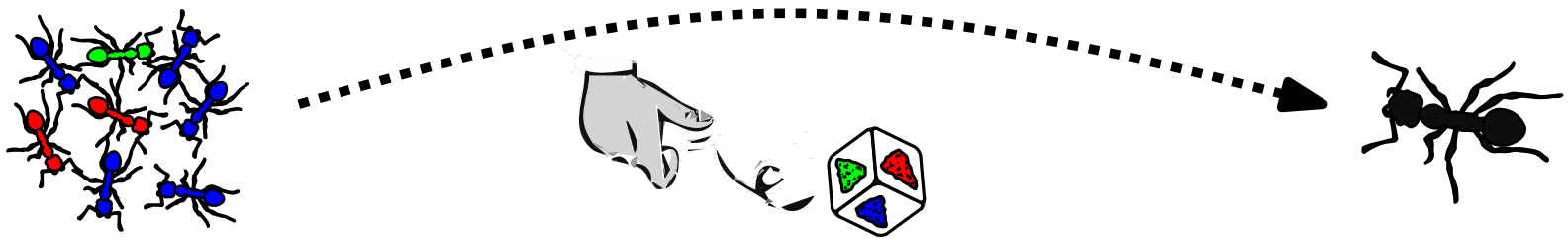
(ε, δ)-majority-preserving noise matrix:

$$(\mathbf{c}P)_{\blacktriangle} - (\mathbf{c}P)_{\blacktriangle} > \varepsilon\delta, \qquad (\mathbf{c}P)_{\blacktriangle} - (\mathbf{c}P)_{\blacktriangle} > \varepsilon\delta$$

Random sender in conf. $\mathbf{c}$

Noise acting according to matrix P

Message distributed as $c \cdot P$



Majority-Preserving Matrix

Configuration $\mathbf{c} := (\#\,/n, \#\,/n, \#\,/n)$

δ-majority-biased configuration w.r.t. :

$$\#\,/n - \#\,/n > \delta$$
$$\#\,/n - \#\,/n > \delta$$

(ε, δ)-majority-preserving noise matrix:

$$(\mathbf{c}P)_{\blacktriangle} - (\mathbf{c}P)_{\blacktriangle} > \varepsilon\delta, \quad (\mathbf{c}P)_{\blacktriangle} - (\mathbf{c}P)_{\blacktriangle} > \varepsilon\delta$$

[PODC '16]: Let S initial set of agents with k opinions. S is $\delta = \Omega(\sqrt{\log n/|S|})$-majority-biased. $|S| = \Omega(\frac{\log n}{\epsilon^2})$. P is (ϵ, δ)-majority-preserving. Plurality consensus can be solved in $O(\frac{\log n}{\epsilon^2})$ rounds w.h.p., with $O(\log\log n + \log\frac{1}{\epsilon})$ memory per node.

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.

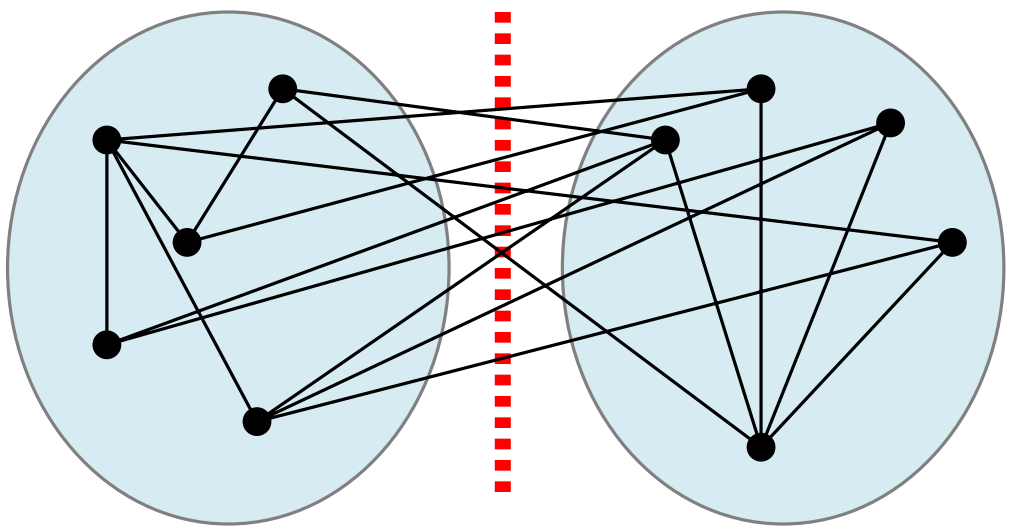
The Median, the Mode and... the Mean

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

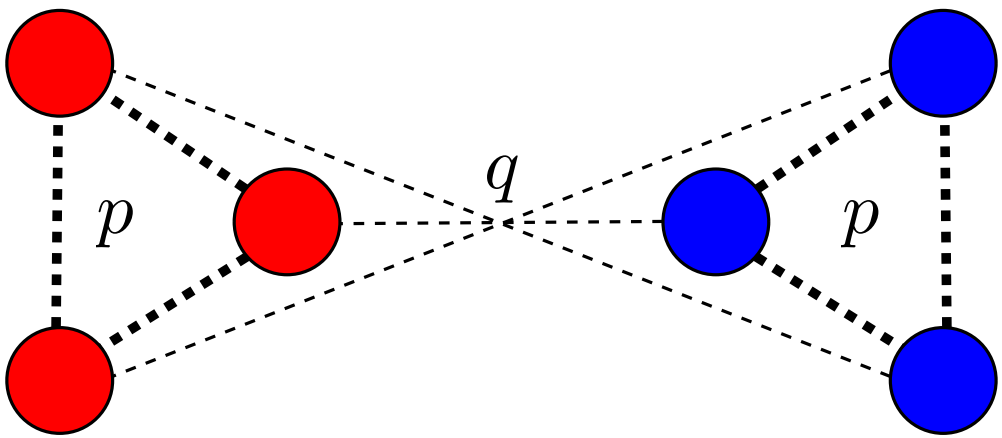
Minimum Bisection Problem.
Input: a graph G with $2n$ nodes.
Output: $S = \arg \min_{\substack{S \subset V \\ |S|=n}} E(S, V - S)$.



[Garey, Johnson, Stockmeyer '76]:
Min-Bisection is *NP-Complete*.

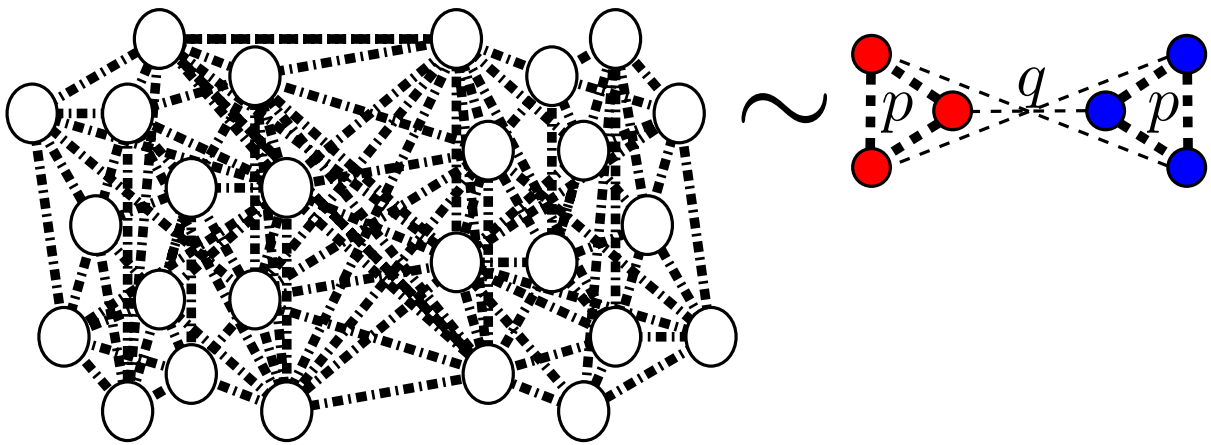
The Stochastic Block Model

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



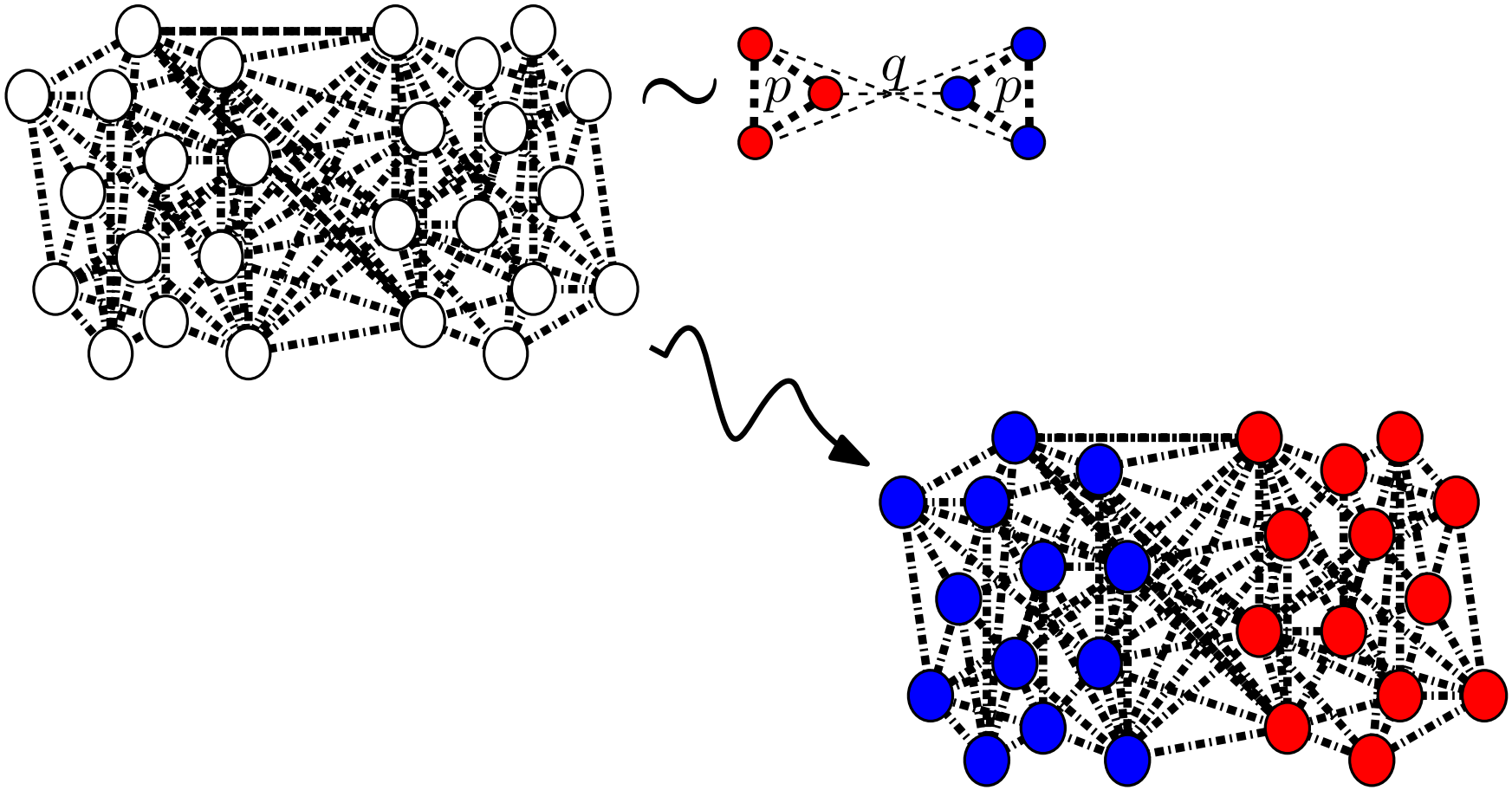
The Stochastic Block Model

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



The Stochastic Block Model

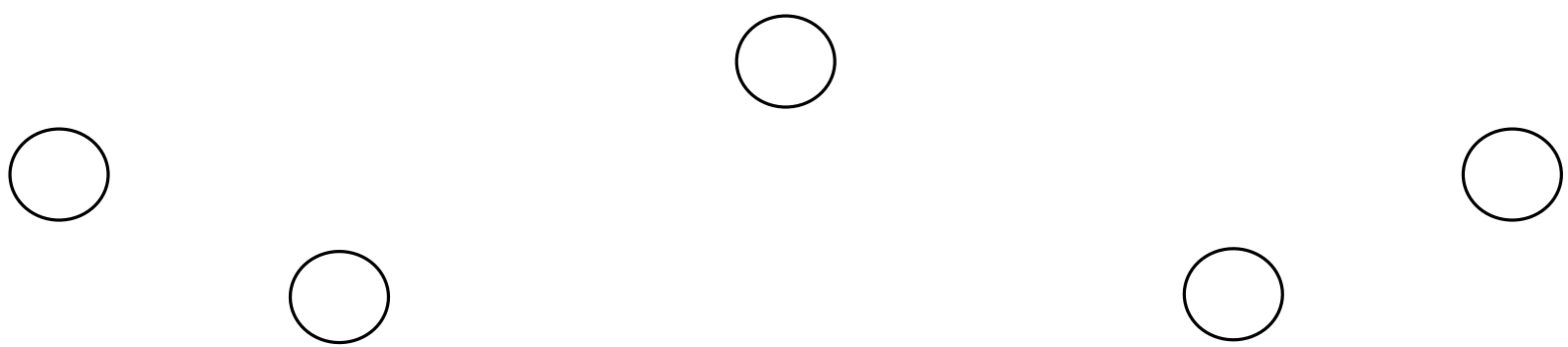
Reconstruction problem. Given graph generated by SBM, find original partition.



The Averaging Dynamics

Al nodes at the same time:

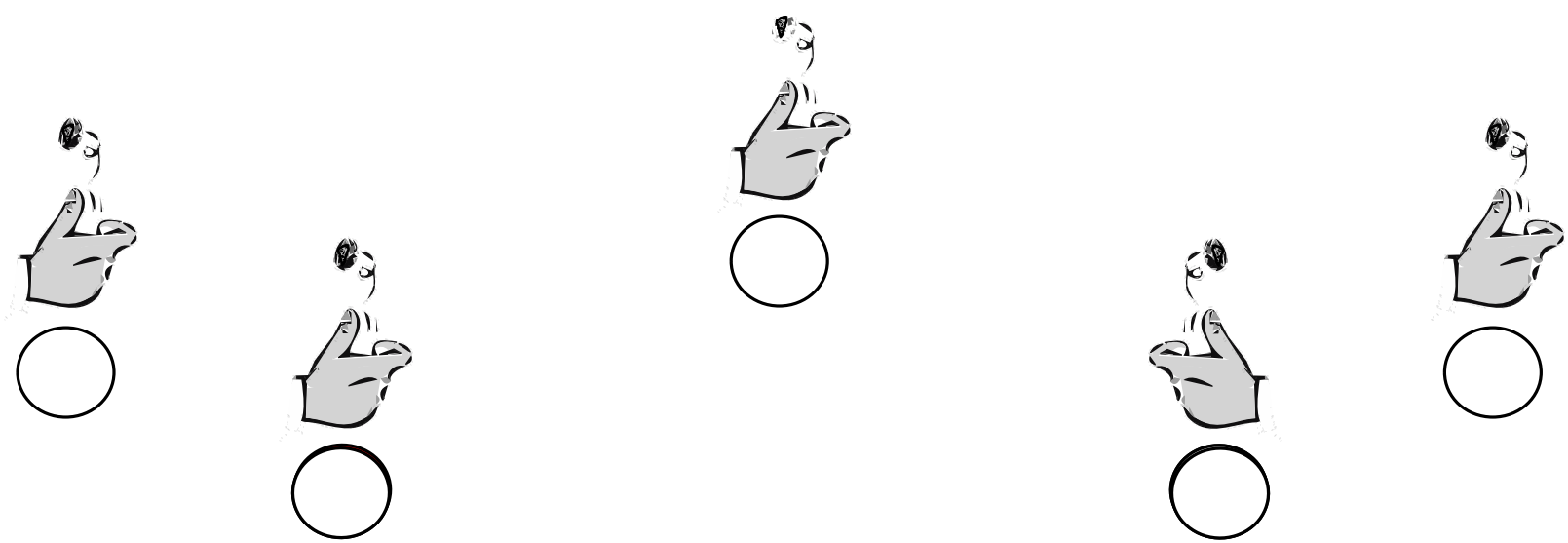
- At $t = 0$, randomly pick value $x^{(t)} \in \{+\mathbf{1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics

Al nodes at the same time:

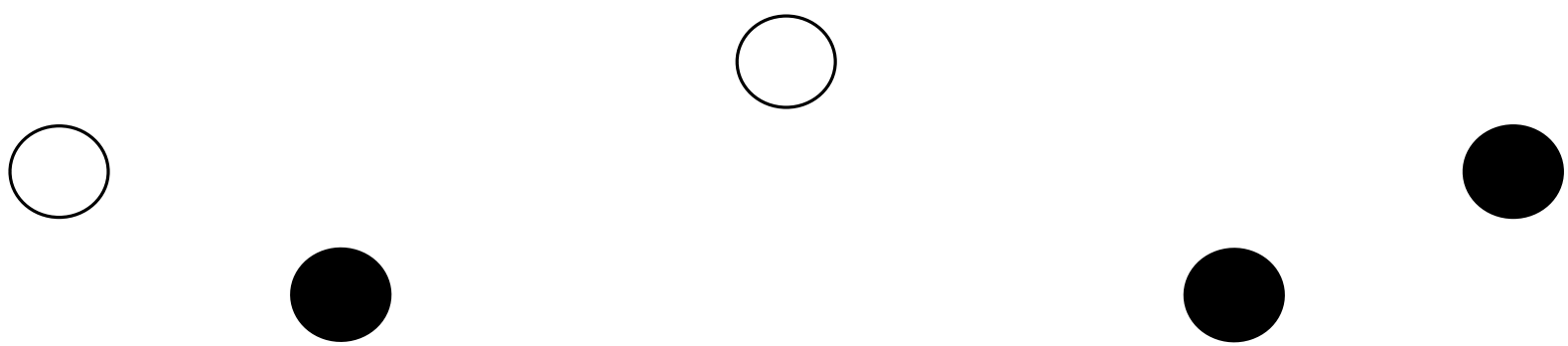
- At $t = 0$, randomly pick value $x^{(t)} \in \{+\mathbf{1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics

Al nodes at the same time:

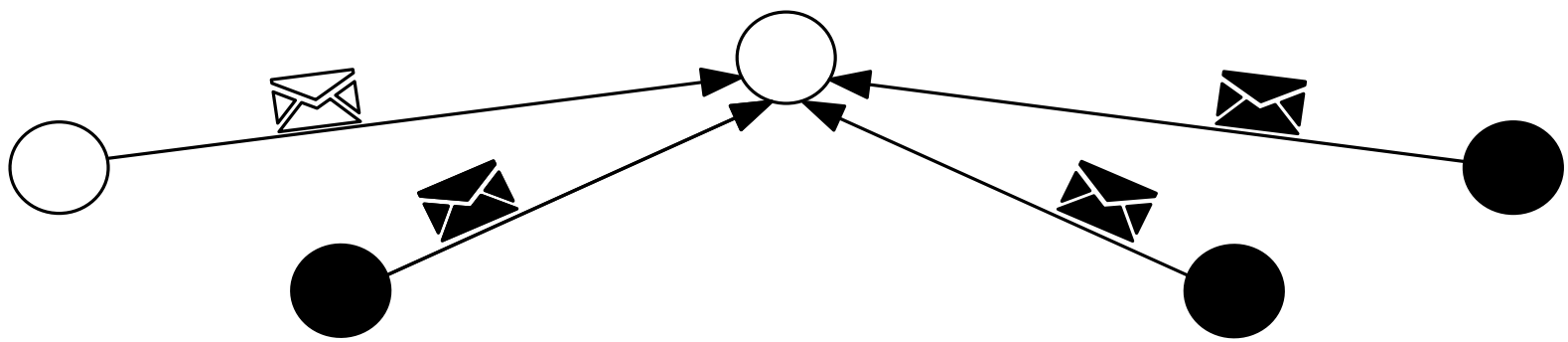
- At $t = 0$, randomly pick value $x^{(t)} \in \{+\mathbf{1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics

Al nodes at the same time:

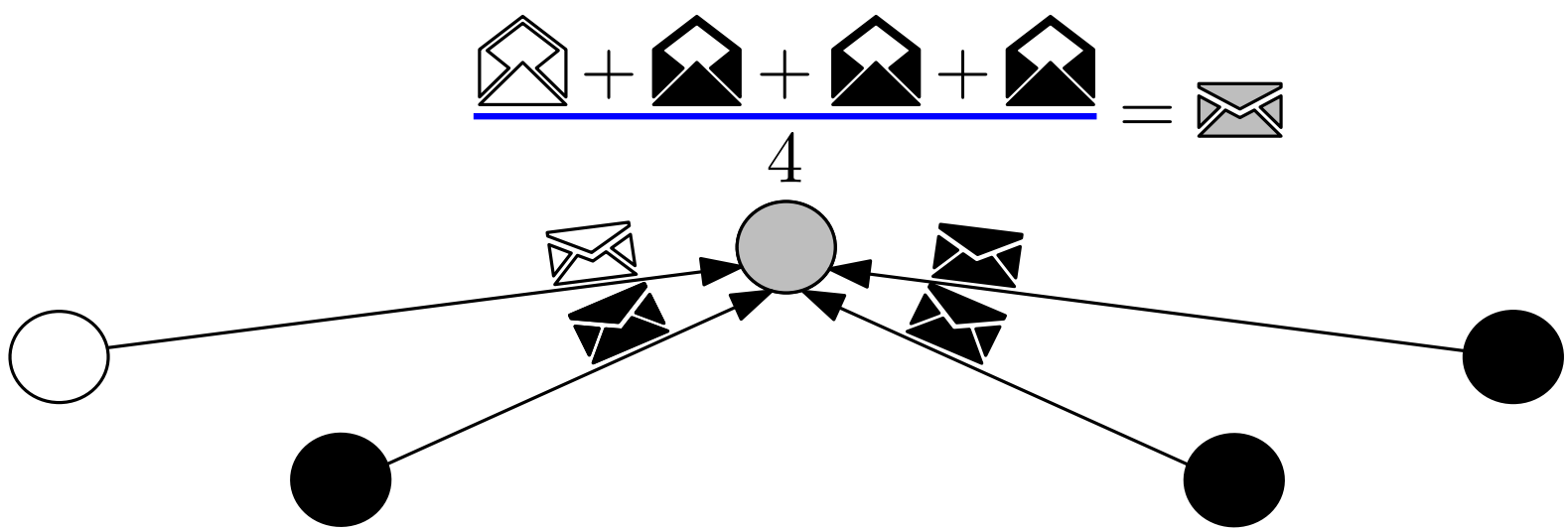
- At $t = 0$, randomly pick value $x^{(t)} \in \{+\mathbf{1}, \mathbf{-1}\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.



The Averaging Dynamics

Al nodes at the same time:

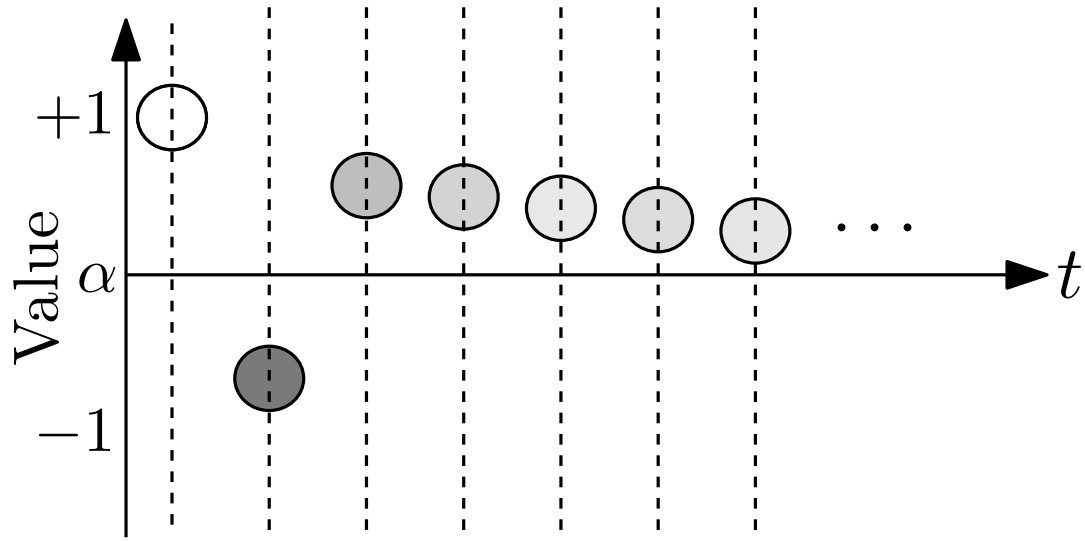
- At $t = 0$, randomly pick value $x^{(t)} \in \{+1, -1\}$.
- Then, at each round
  1. Set value $x^{(t)}$ to average of neighbors,
  2. Set label to **blue** if $x^{(t)} < x^{(t-1)}$, **red** otherwise.

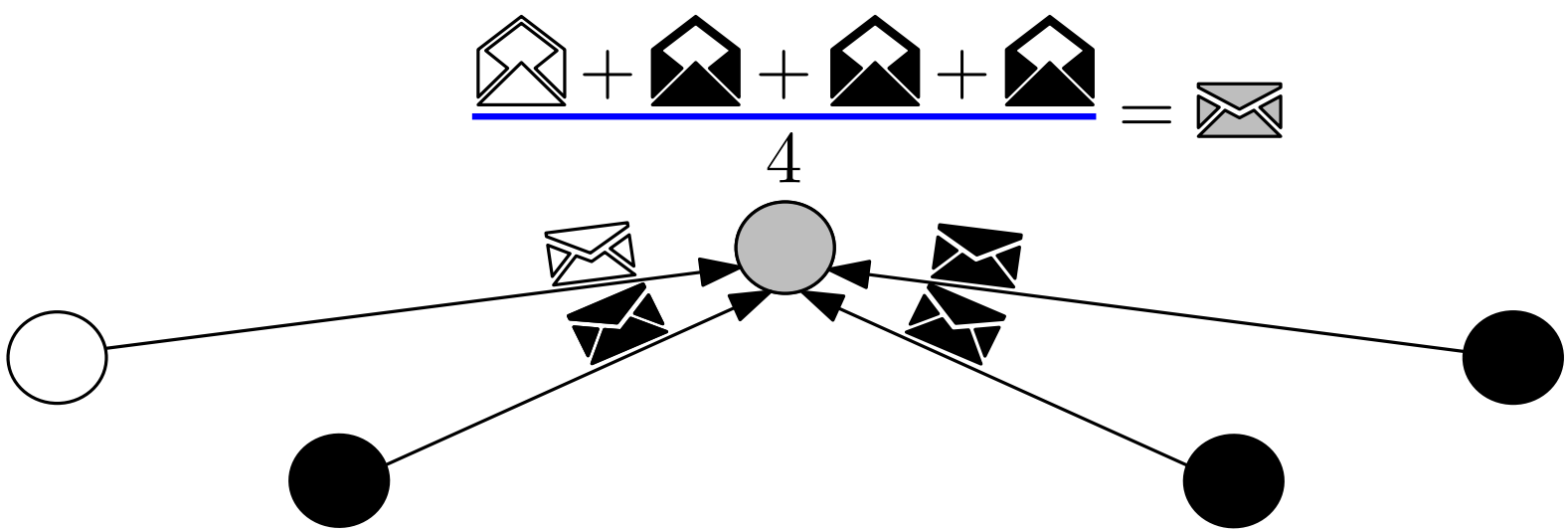


The Averaging Dynamics

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.

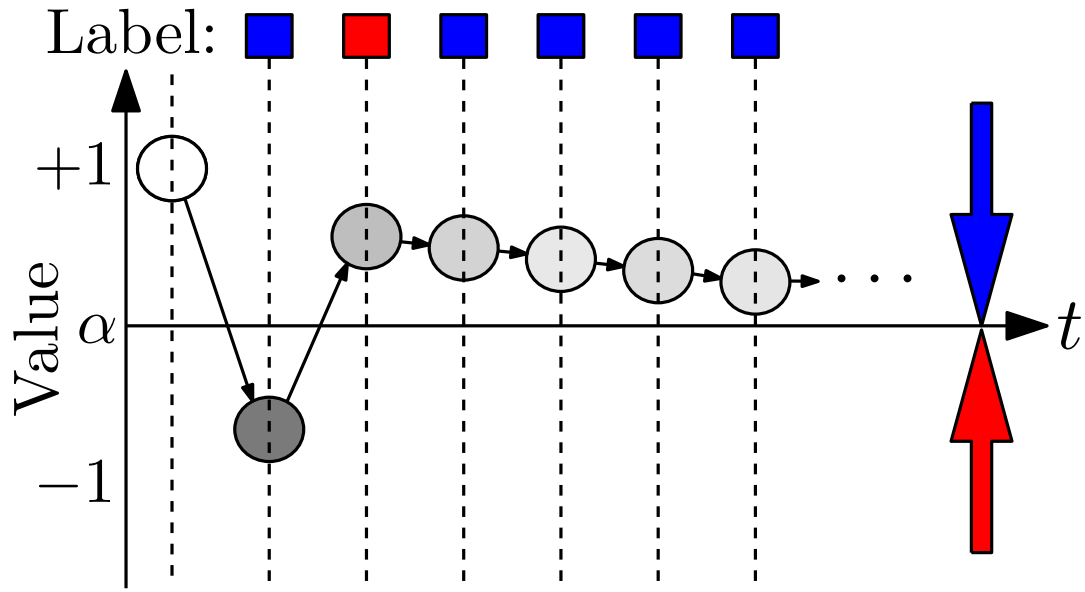


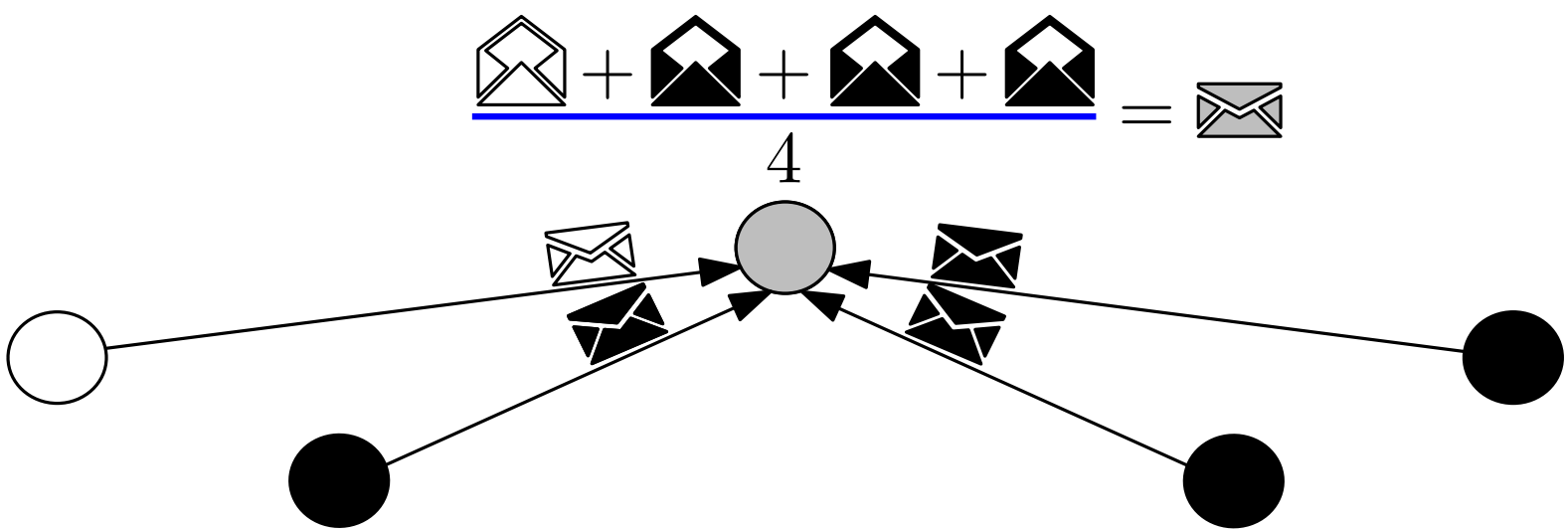


The Averaging Dynamics

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.





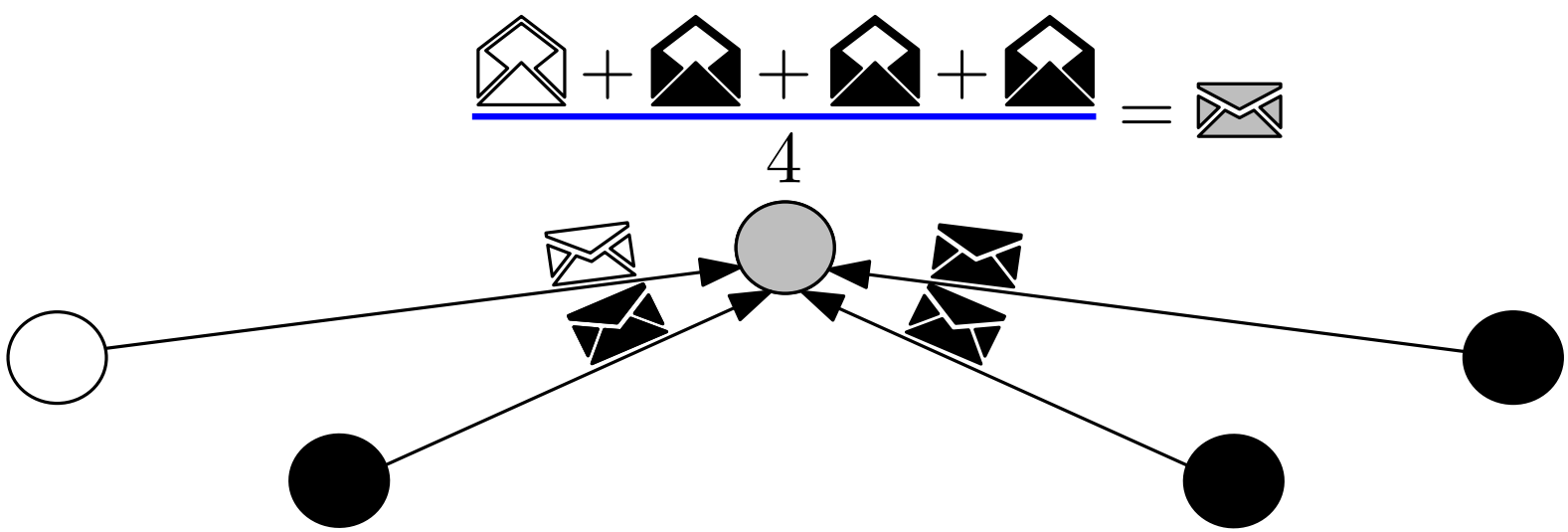
The Averaging Dynamics

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.

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

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.

Well studied process [Shah '09]:

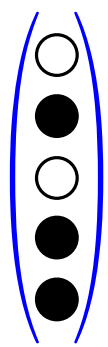
- Converges to (weighted) global average of initial values,
- Convergence time = mixing time of G,
- Important applications in fault-tolerant self-stabilizing consensus.

$A = (\mathbb{1}_{((u,v)\in E)})_{u,v\in V}$ adjacency matrix of G

D diagonal matrix of node degrees in G

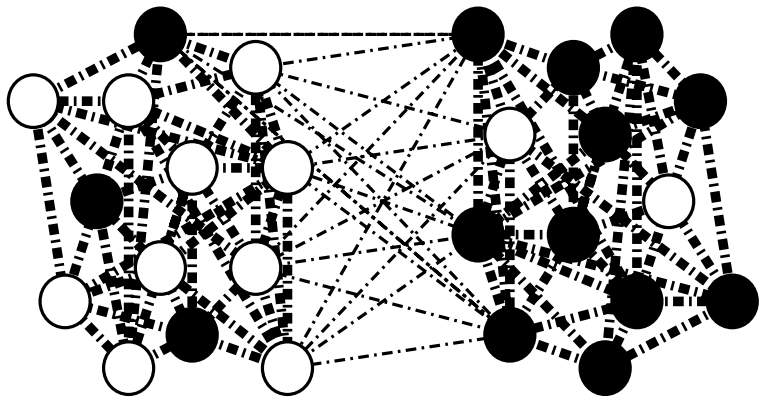
$P = D^{-1}A$ transition matrix of random walk

Averaging is a **linear** dynamics

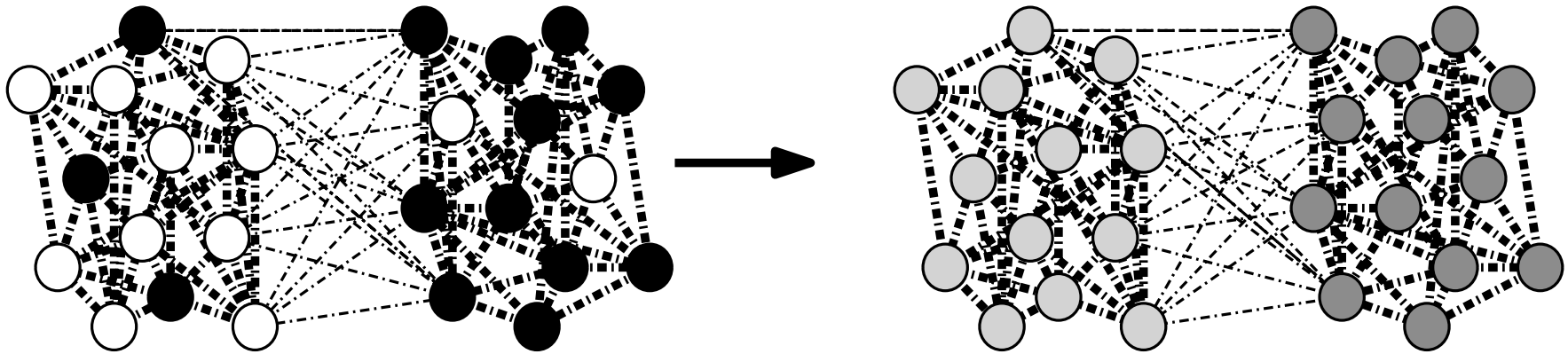
$\mathbf{x}^{(t)} =$ 

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

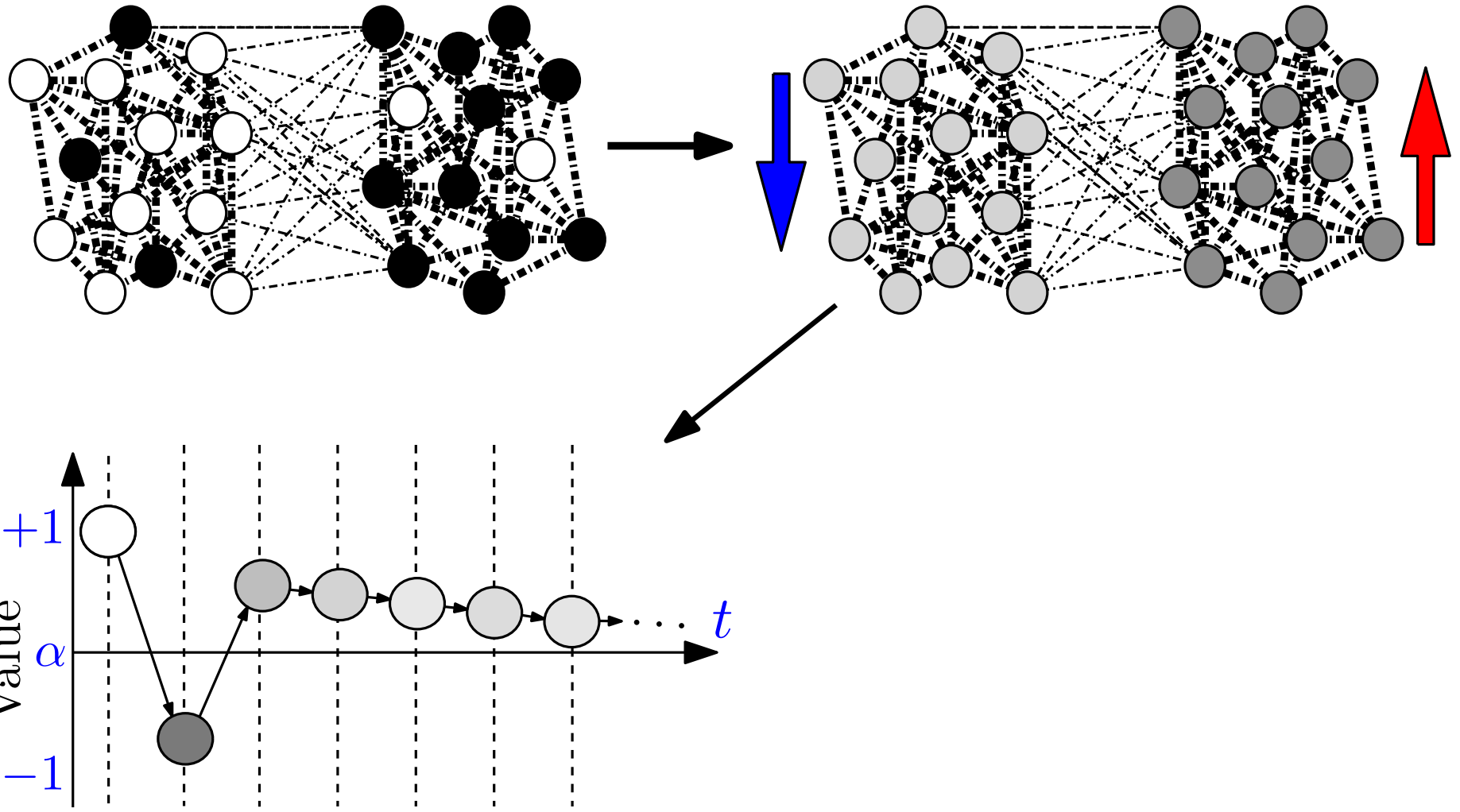
Community Detection via Averaging Dynamics



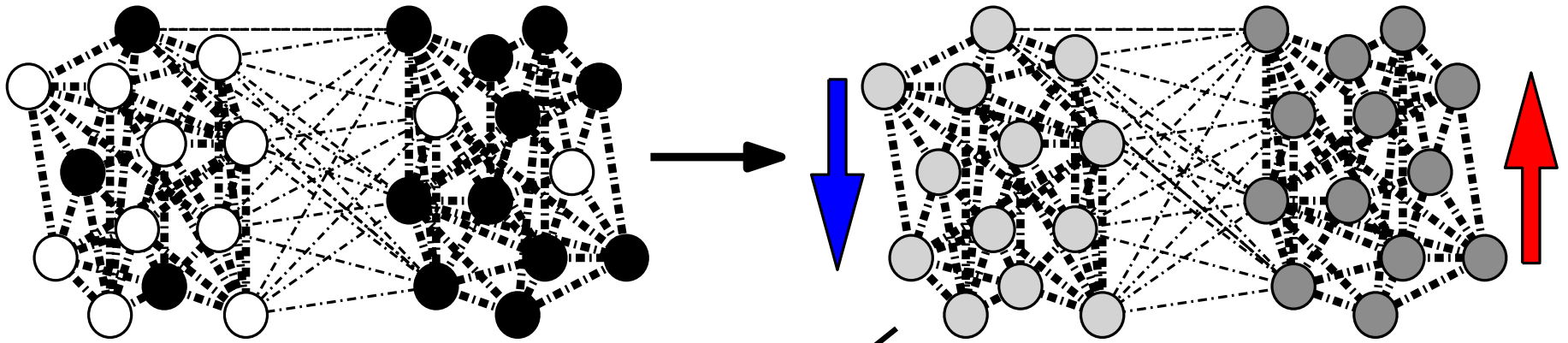
Community Detection via Averaging Dynamics

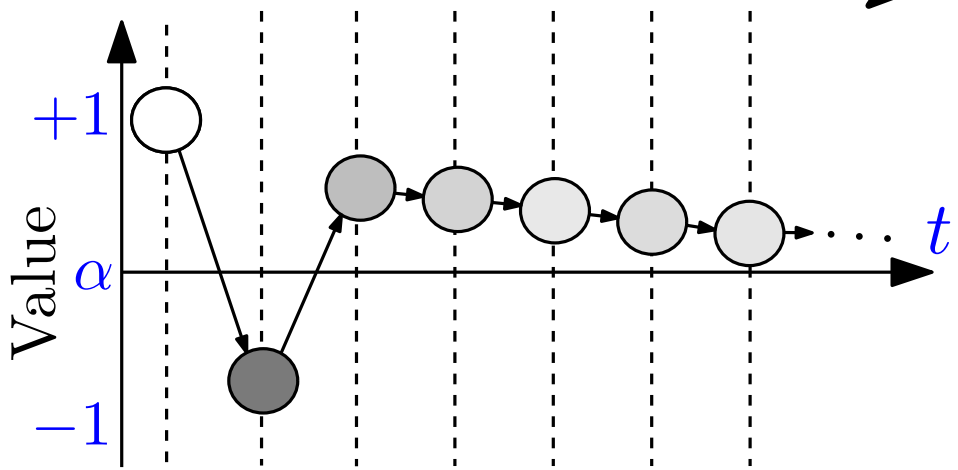


Community Detection via Averaging Dynamics



Community Detection via Averaging Dynamics





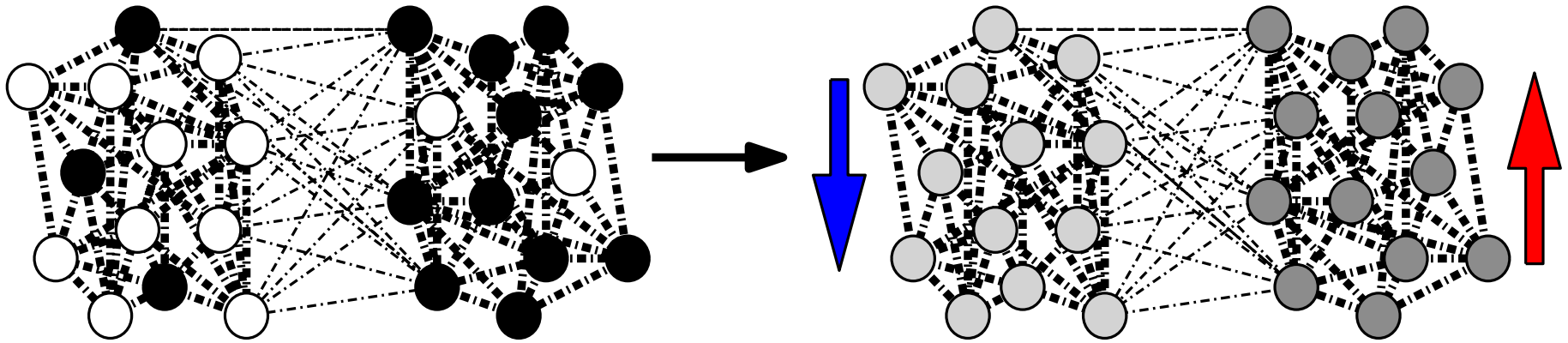
$v_1, ..., v_n$ eigenvectors of random walk matrix P:

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

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

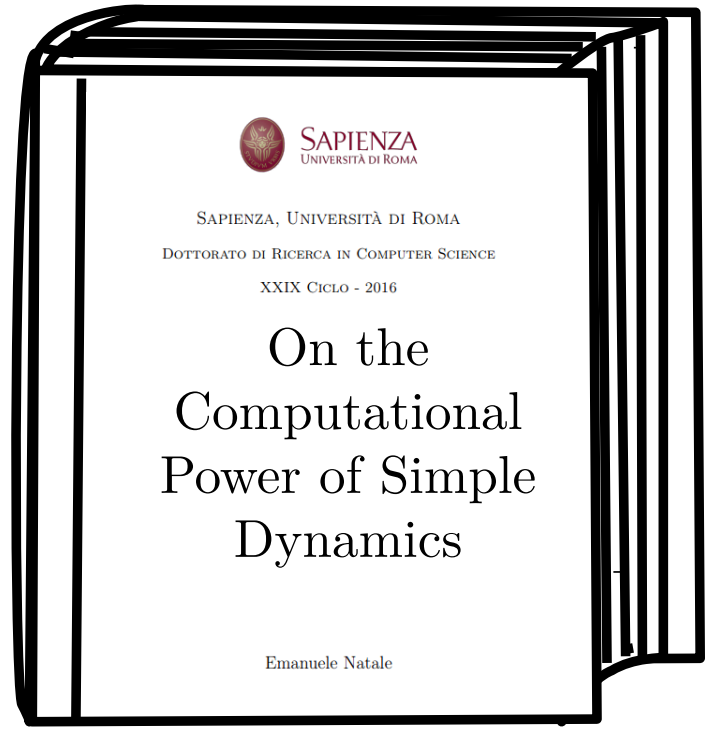
"nice" graph

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).

Check Out my Thesis!



https://goo.gl/Q0LC6x

Thank you!