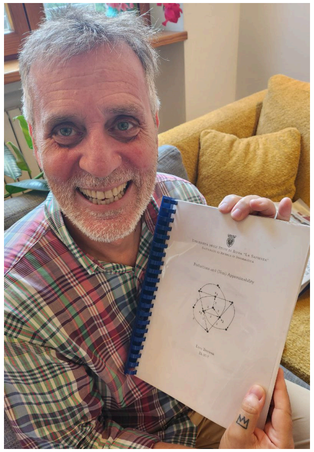
TREVISAN'S CONTRIBUTIONS TO DISTRIBUTED COMPUTING

Emanuele Natale

LucaFest
8 October 2024

LUCA AND ROMAN CS

- 1993. **First BSc degree** in CS at **Sapienza University**
- Most coauthored papers with Luca (Romans highlighted):
  1. Andrea Clementi (22)
  2. Francesco Pasquale (15)
  3. Salil P. Vadhan (15)
  4. Luca Becchetti (14)
  5. Madhu Sudan (10)
  6. Pierluigi Crescenzi (10)
  7. Shayan Oveis Gharan (8)
  8. Emanuele Natale (8)
  9. Riccardo Silvestri (8)
  10. ...



LUCA & ME

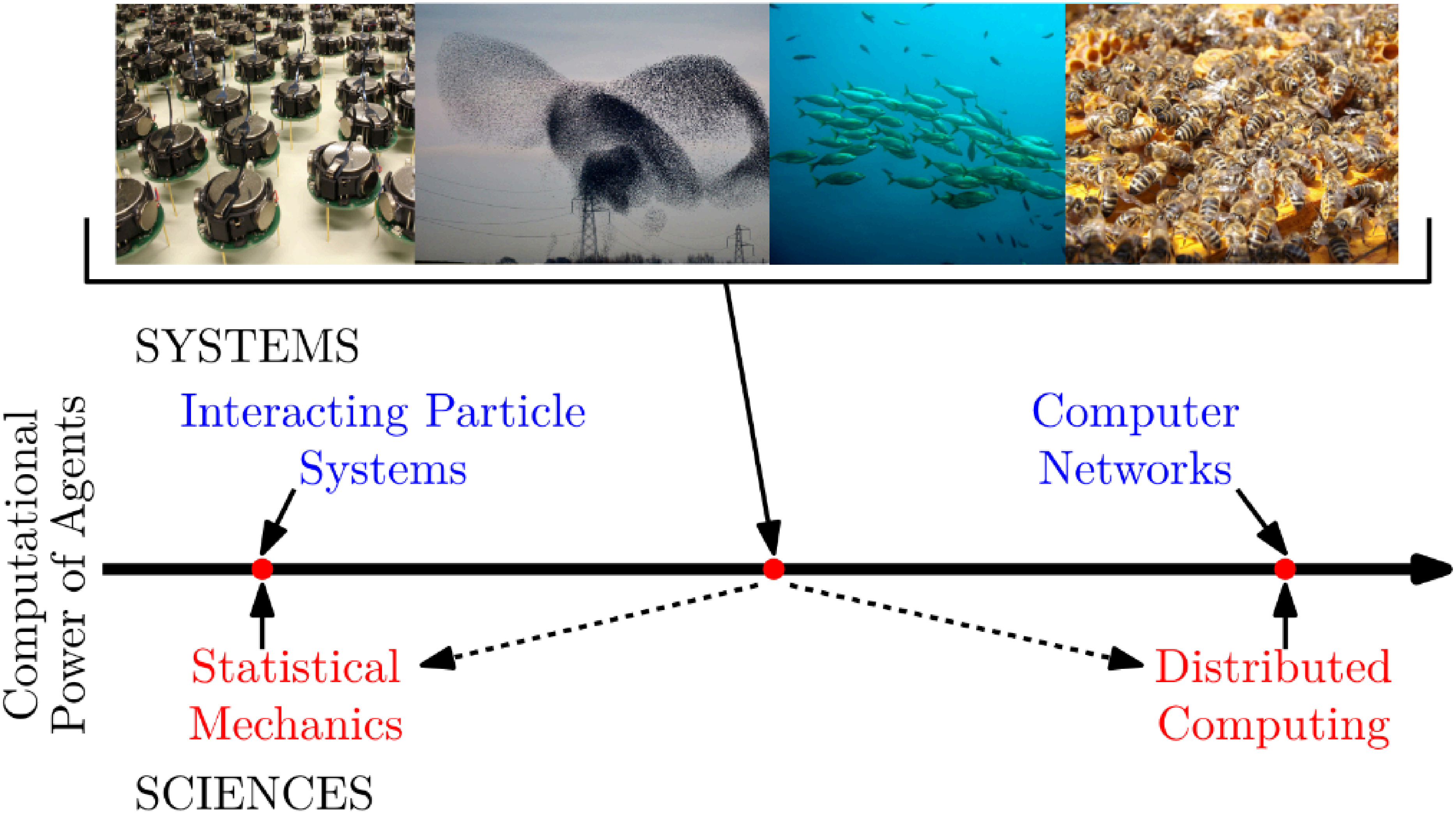
- Meeting in Rome since 2013
- 2016. **Simons'** *Counting Complexity and Phase Transitions* Program
- 2018. **Simons'** *The Brain and Computation* Program

ROMANS + LUCA T

- Simple dynamics for plurality consensus. SPAA 2014.
- Stabilizing Consensus with Many Opinions. SODA 2016.
- Find Your Place: Simple Distributed Algorithms for Community Detection. SODA 2017.
- Average Whenever You Meet: Opportunistic Protocols for Community Detection. ESA 2018.
- Finding a Bounded-Degree Expander Inside a Dense One. SODA 2020.
- Consensus vs Broadcast, with and Without Noise. ITCS 2020.
- Expansion and Flooding in Dynamic Random Networks with Node Churn. ICDCS 2021.
- Percolation and Epidemic Processes in One-Dimensional Small-World Networks. LATIN 2022.
- Bond Percolation in Small-World Graphs with Power-Law Distribution. SAND 2023.
- On the Role of Memory in Robust Opinion Dynamics. IJCAI 2023.
- The Minority Dynamics and the Power of Synchronicity. SODA 2024.

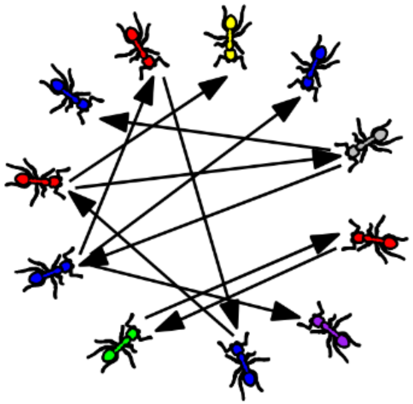
COMPUTATION IN SIMPLE SYSTEMS

A **computational lens** on how global behavior emerges from simple local interactions among individuals



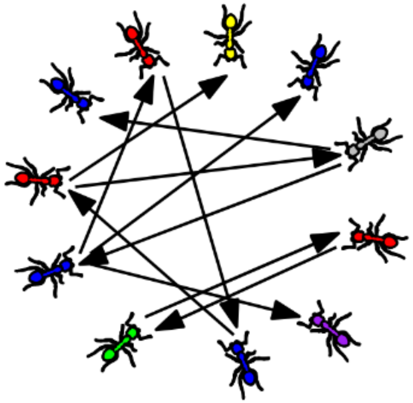
LUCA'S WORK ON SOME DYNAMICS

PULL Model. At each round each agent observes the state of h other randomly chosen agents



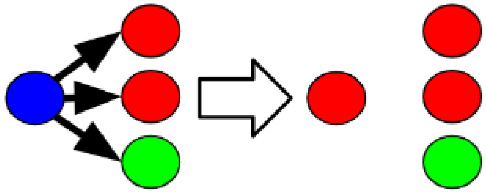
LUCA'S WORK ON SOME DYNAMICS

PULL Model. At each round each agent observes the state of h other randomly chosen agents

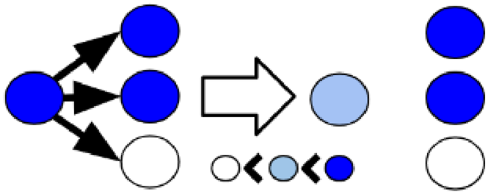


- Anonymous agents
- few possible states
- **simple** update function f of observed agents

Examples:



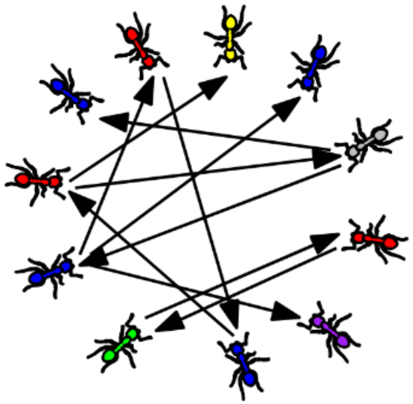
3-Majority



Averaging

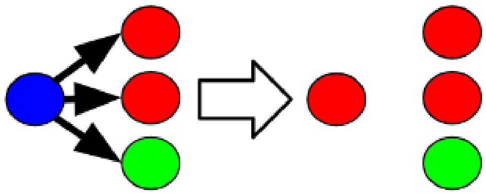
LUCA'S WORK ON SOME DYNAMICS

PULL Model. At each round each agent observes the state of h other randomly chosen agents



- Anonymous agents
- few possible states
- simple update function f of observed agents

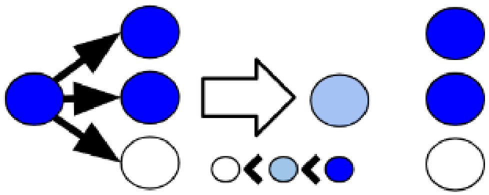
Examples:



3-Majority

More on Dynamics:

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



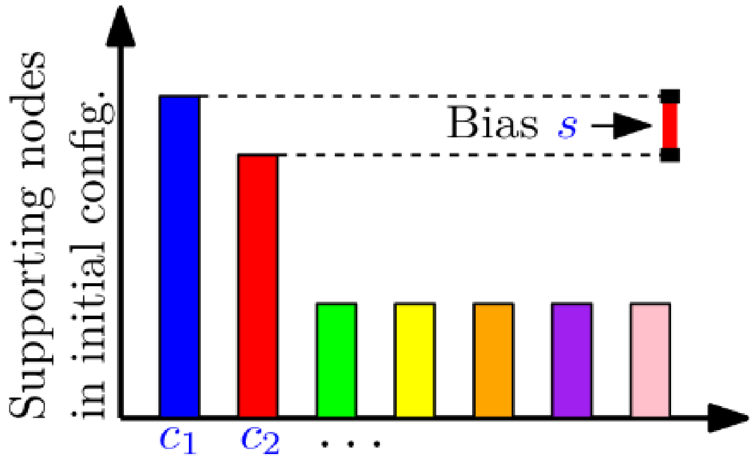
Averaging

MAJORITY DYNAMICS

What's the convergence time
with k colors?

MAJORITY DYNAMICS

What's the convergence time with k colors?

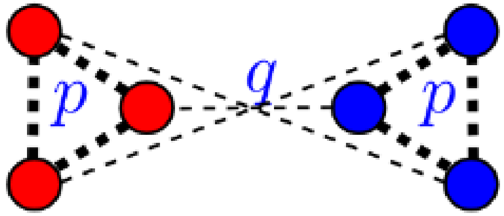


Theorem [SPAA'14, SODA '16]. n agents, k colors:

- From configuration with bias $\Omega(\sqrt{kn \log n})$, **3-Majority** converges to plurality in $O(k \log n)$ rounds w.h.p.
- h**-Majority** requires $\Omega(k/h^2)$ to converge
- **3-Majority** reaches almost-consensus even against $\tilde{O}(n^{\Theta(1)})$ adversary.

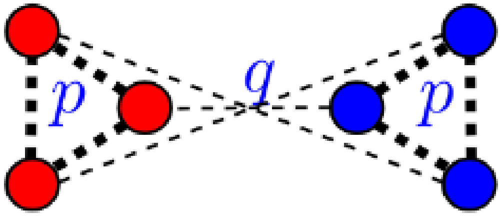
COMMUNITY DETECTION

Stochastic Block Model (SBM). Communities V_1 and V_2 of size $n/2$ such that:

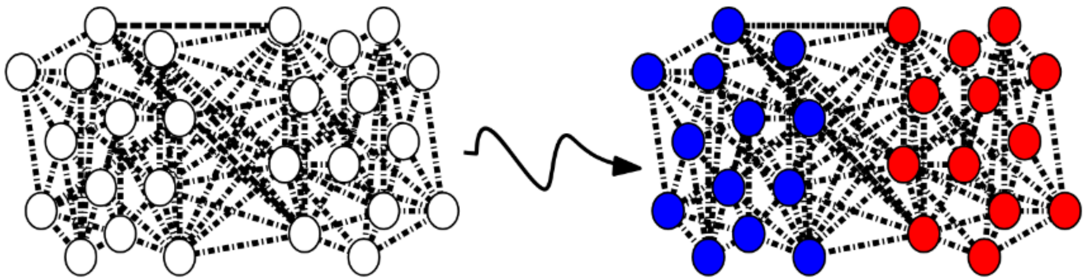


COMMUNITY DETECTION

Stochastic Block Model (SBM). Communities V_1 and V_2 of size $n/2$ such that:

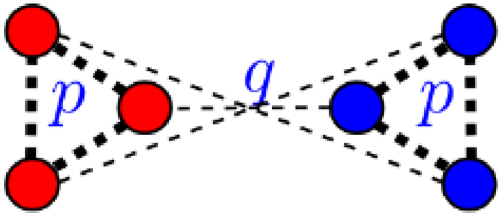


Reconstruction Problem. Given a graph generated by the SBM, reconstruct original partition.

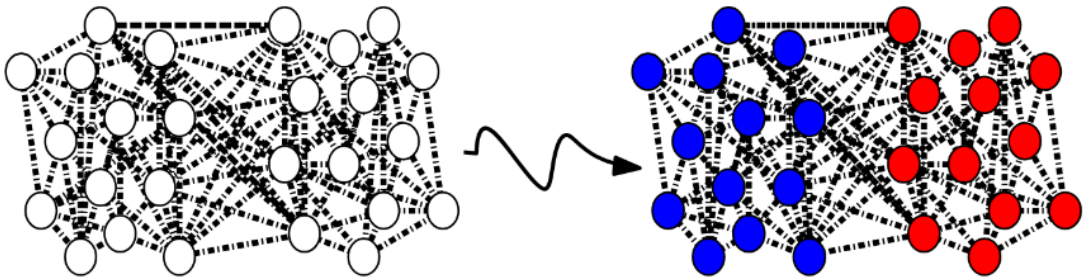


COMMUNITY DETECTION

Stochastic Block Model (SBM). Communities V_1 and V_2 of size $n/2$ such that:



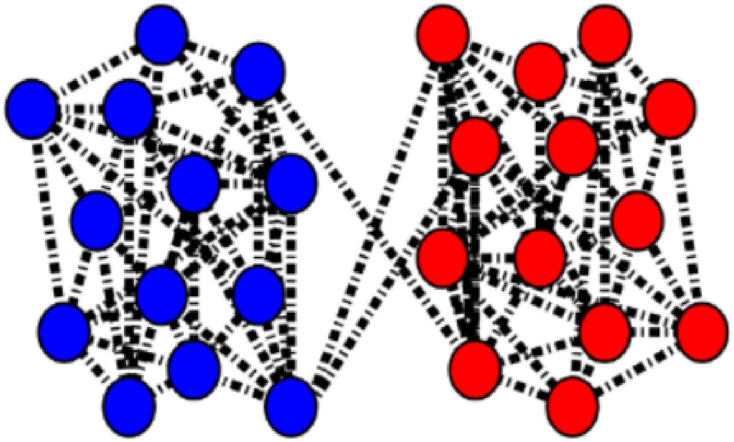
Reconstruction Problem. Given a graph generated by the SBM, reconstruct original partition.



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

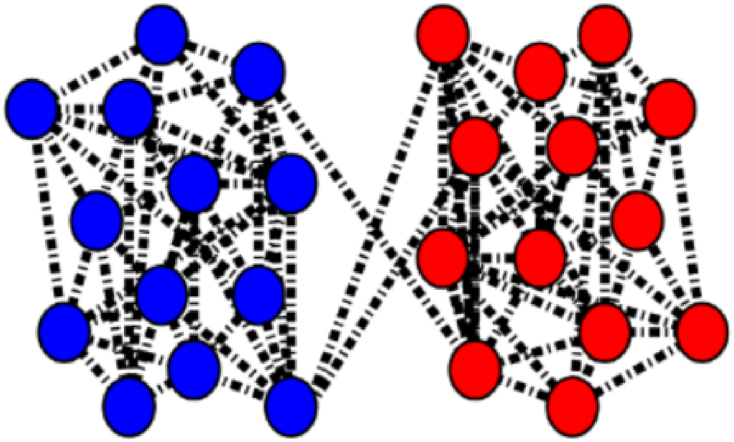
COMMUNITY DETECTION FASTER THAN MIXING TIME

- Community structure encoded in eigenvectors



COMMUNITY DETECTION FASTER THAN MIXING TIME

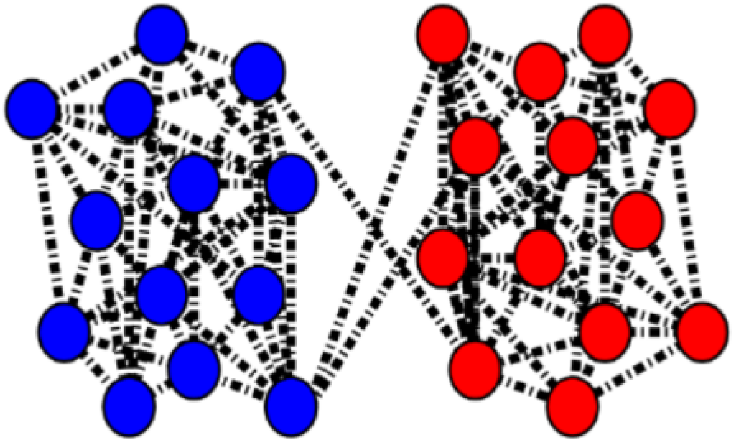
- Community structure encoded in eigenvectors
- Efficiently computing them requires **mixing time**★



★: time it takes for a random walk to converge to stationary distribution

COMMUNITY DETECTION FASTER THAN MIXING TIME

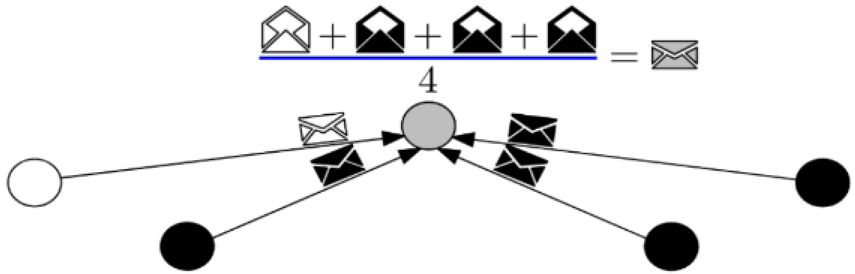
- Community structure encoded in eigenvectors
- Efficiently computing them requires **mixing time**★
- Reconstruction should be *easy* when mixing time *large*...



★: time it takes for a random walk to converge to stationary distribution

AVERAGING DYNAMICS

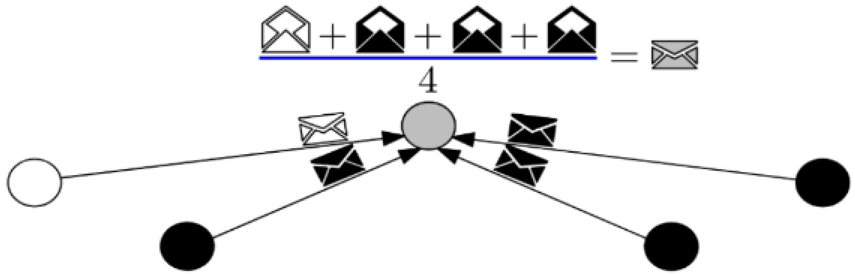
All nodes at each round update their value $x(t)$ to **average** of neighbors:



$$x^{(0)} = \begin{pmatrix} \circ \\ \bullet \\ \vdots \\ \bullet \\ \circ \end{pmatrix}, \quad x^{(t)} = \underset{\substack{\text{transition} \\ \text{matrix}}}{P} \cdot x^{(t-1)} = P^t \cdot x^{(0)}$$

AVERAGING DYNAMICS

All nodes at each round update their value $x(t)$ to **average** of neighbors:

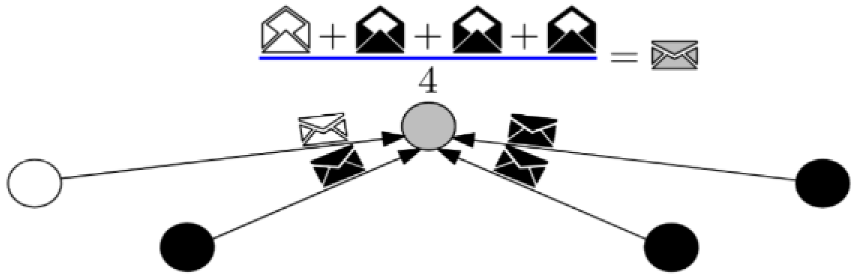


$$x^{(0)} = \begin{pmatrix} \circ \\ \bullet \\ \vdots \\ \bullet \\ \circ \end{pmatrix}, \quad x^{(t)} = \underbrace{P}_{\text{transition matrix}} \cdot x^{(t-1)} = P^t \cdot x^{(0)}$$

1st eigenvec is $(1, \ldots, 1)$ and 2nd eigenvec. for *nice* graphs is $\approx (1, \ldots, 1, -1, \ldots, -1)$

AVERAGING DYNAMICS

All nodes at each round update their value $x(t)$ to **average** of neighbors:

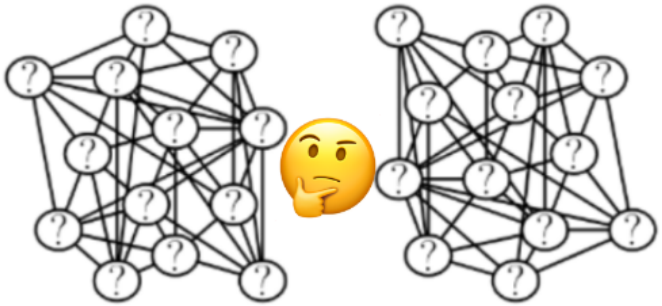


$$x^{(0)} = \begin{pmatrix} \circ \\ \bullet \\ \vdots \\ \bullet \\ \circ \end{pmatrix}, \quad x^{(t)} = \underbrace{P}_{\text{transition matrix}} \cdot x^{(t-1)} = P^t \cdot x^{(0)}$$

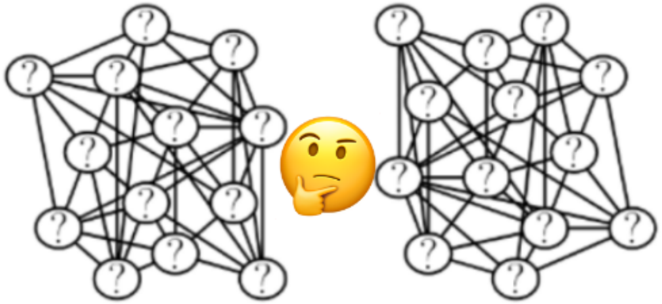
1st eigenvec is $(1, \ldots, 1)$ and 2nd eigenvec. for *nice* graphs is $\approx (1, \ldots, 1, -1, \ldots, -1)$

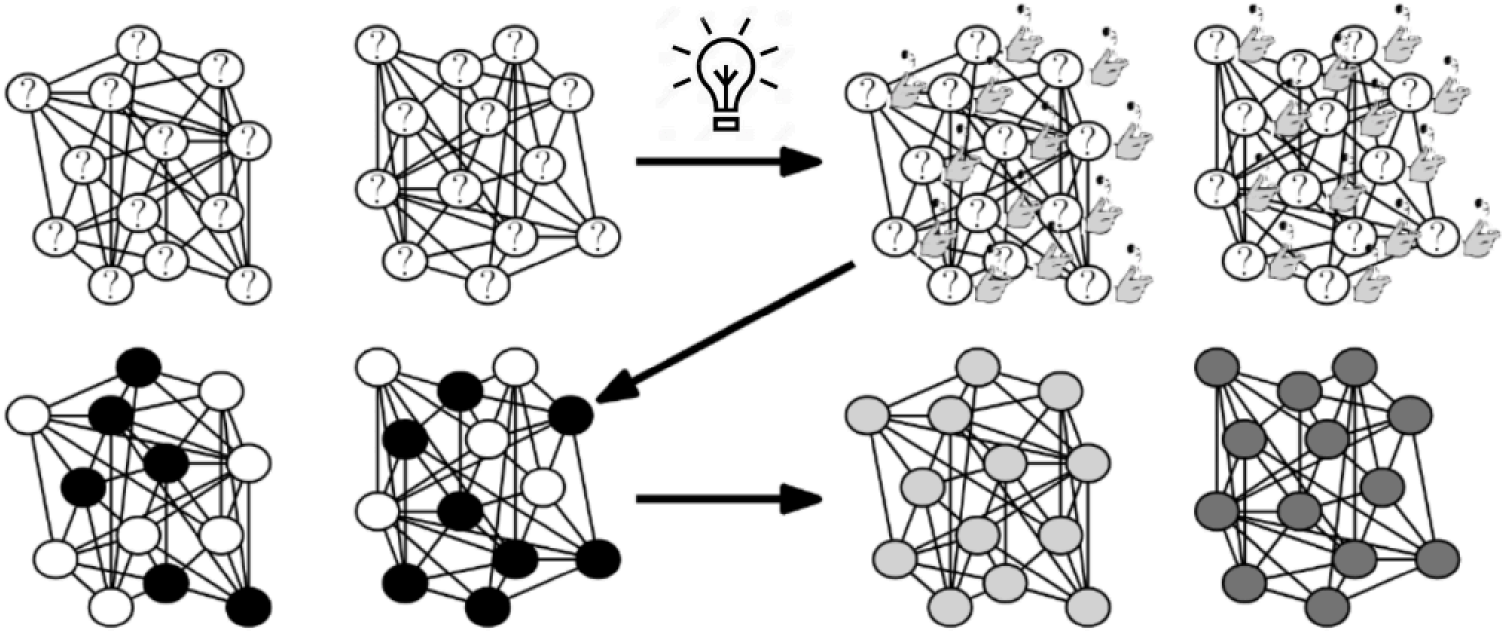
After **mixing time** averaging converges to weighted global average [Boyd et al. 2006].

BREAKING SYMMETRY AMONG COMMUNITIES

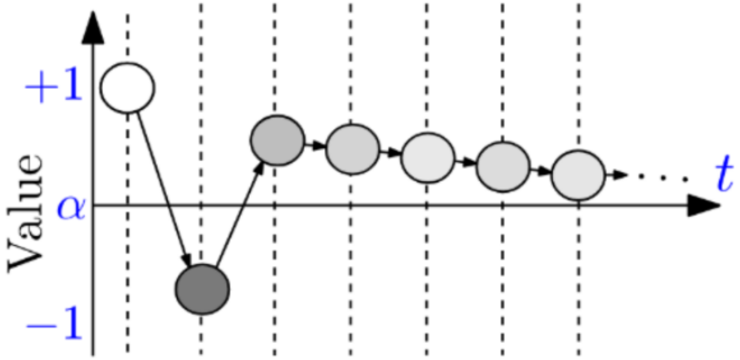


BREAKING SYMMETRY AMONG COMMUNITIES

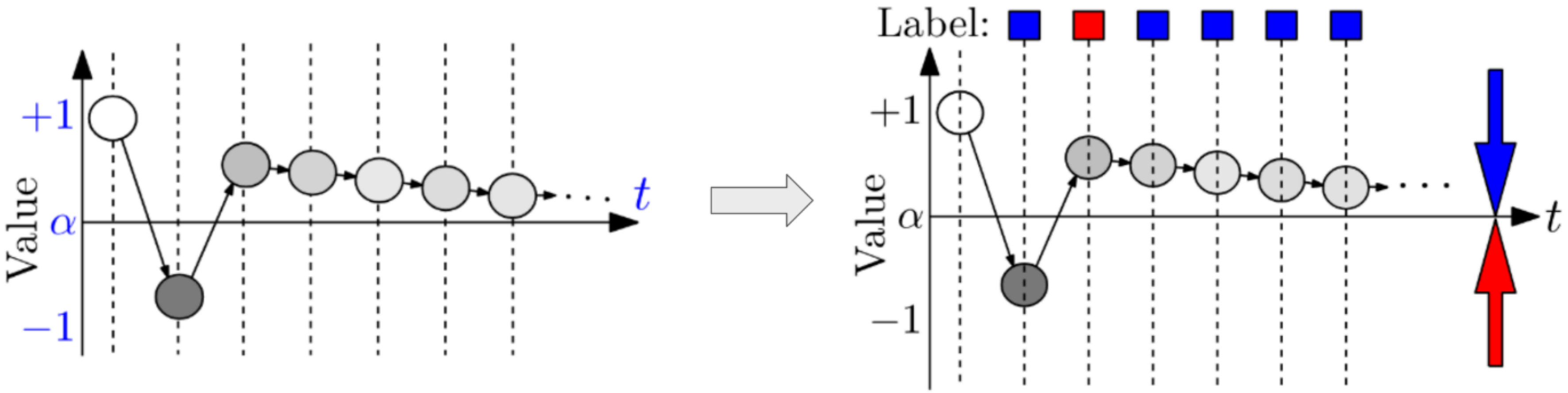




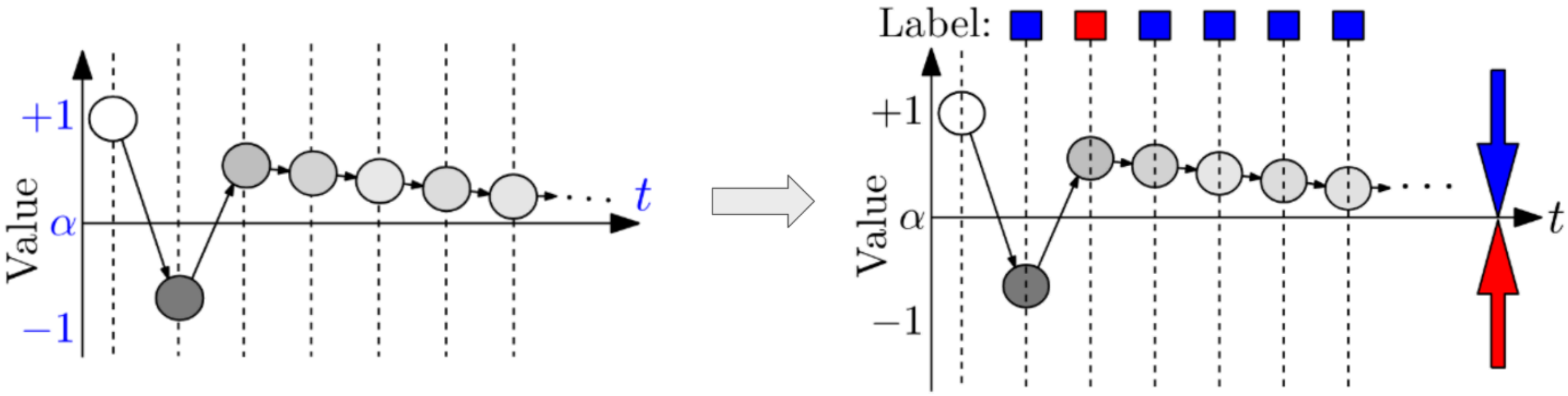
COMMUNITY DETECTION WITH AVERAGING DYNAMICS



COMMUNITY DETECTION WITH AVERAGING DYNAMICS



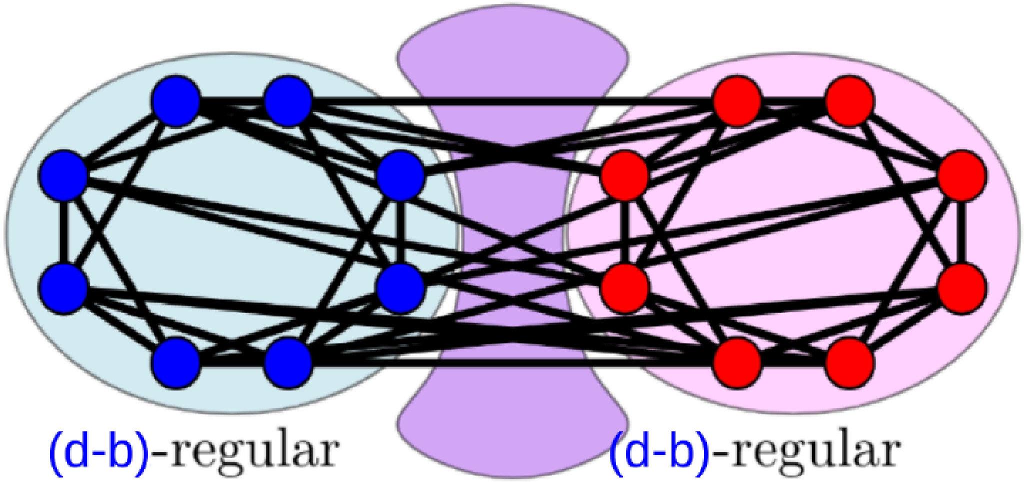
COMMUNITY DETECTION WITH AVERAGING DYNAMICS



At $t = 0$, randomly pick value $x(t) \in \{+1, -1\}$.
Then, at each round:

- Set value $x(t)$ to **average of neighbors**,
- At each step, set label to blue if $x(t) < x(t-1)$, red otherwise.

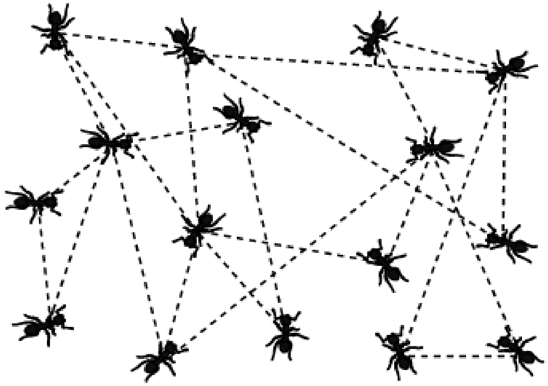
AVERAGING DYNAMICS ON THE SBM



Theorem [SIAM J. Comp. 2020]. Let G be a connected $(2n, d, b)$-**clustered regular** graph with 2nd eigenvalue $\lambda_2 > (1+\varepsilon)\max_{i \geq 3} |\lambda_i|$ for some $\varepsilon > 0$. Then Averaging yields **strong reconstruction** within $\mathcal{O}(\log n)$ rounds w.h.p.

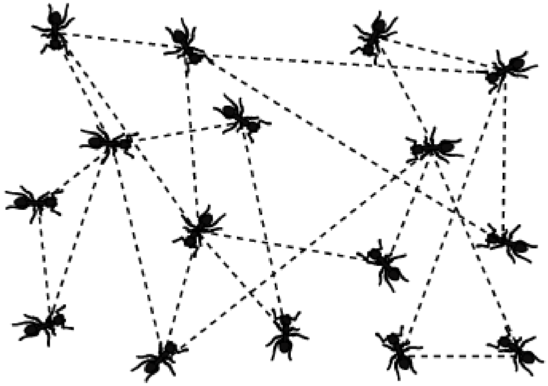
COMM. DET. IN POPULATION PROTOCOLS

At each round a **random edge** is chosen and the two corresponding agents interact.



COMM. DET. IN POPULATION PROTOCOLS

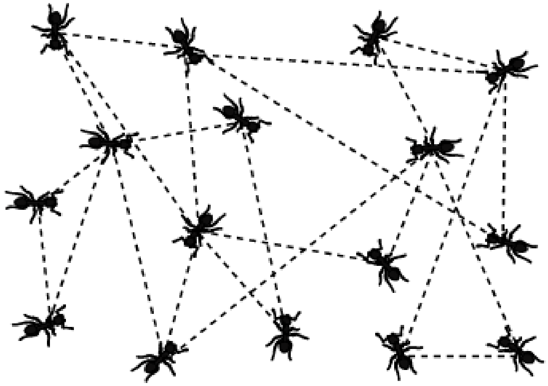
At each round a **random edge** is chosen and the two corresponding agents interact.



Can we leverage the Averaging Dynamics?

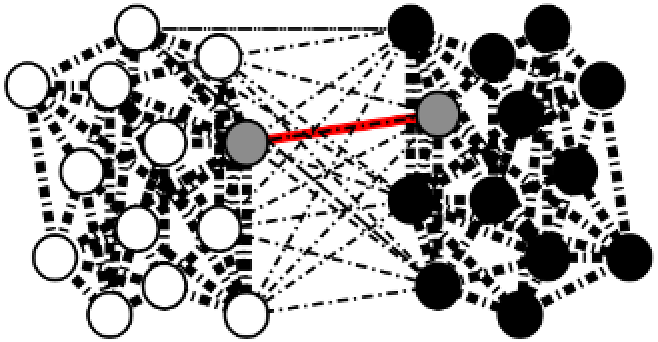
COMM. DET. IN POPULATION PROTOCOLS

At each round a **random edge** is chosen and the two corresponding agents interact.



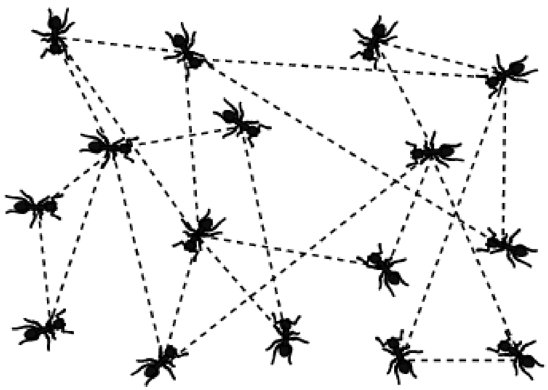
Can we leverage the Averaging Dynamics?

Asynchronous Averaging. If (u, v) activates at time t then $x_u(t) = x_v(t) = \frac{x_u(t-1)+x_v(t-1)}{2}$ [Boyd et al. 2006].



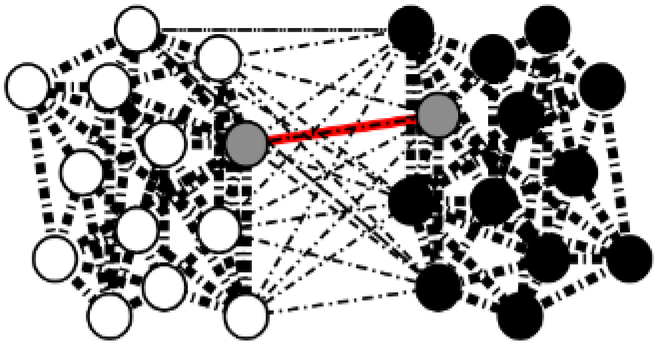
COMM. DET. IN POPULATION PROTOCOLS

At each round a **random edge** is chosen and the two corresponding agents interact.



Can we leverage the Averaging Dynamics?

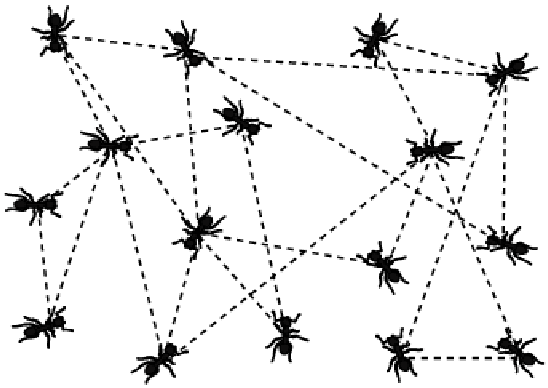
Asynchronous Averaging. If (u, v) activates at time t then $x_u(t) = x_v(t) = \frac{x_u(t-1)+x_v(t-1)}{2}$ [Boyd et al. 2006].



Process **variance** causes issues...

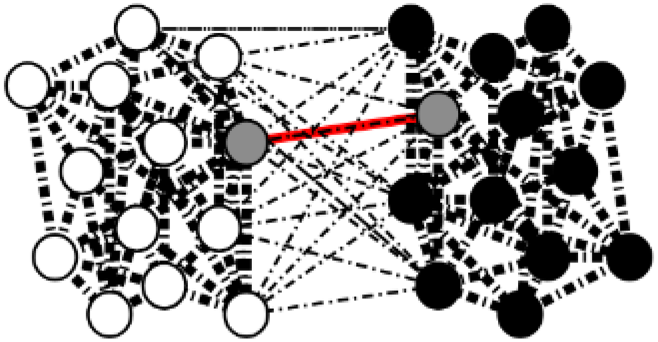
COMM. DET. IN POPULATION PROTOCOLS

At each round a **random edge** is chosen and the two corresponding agents interact.



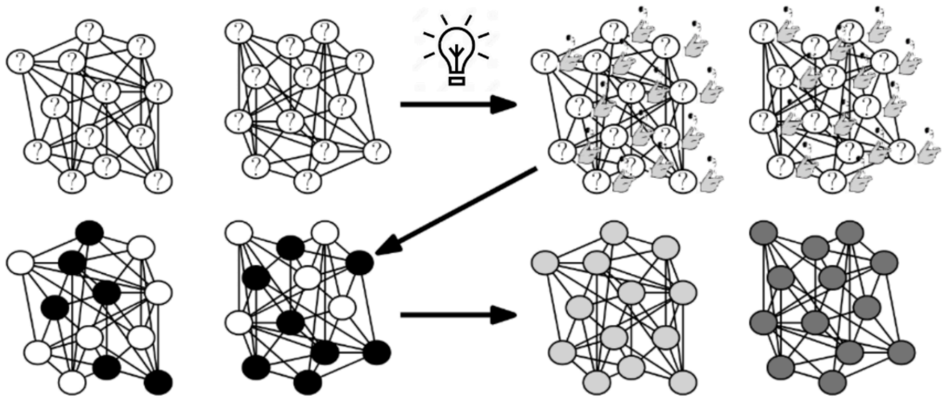
Can we leverage the Averaging Dynamics?

Asynchronous Averaging. If (u, v) activates at time t then $x_u(t) = x_v(t) = \frac{x_u(t-1)+x_v(t-1)}{2}$ [Boyd et al. 2006].

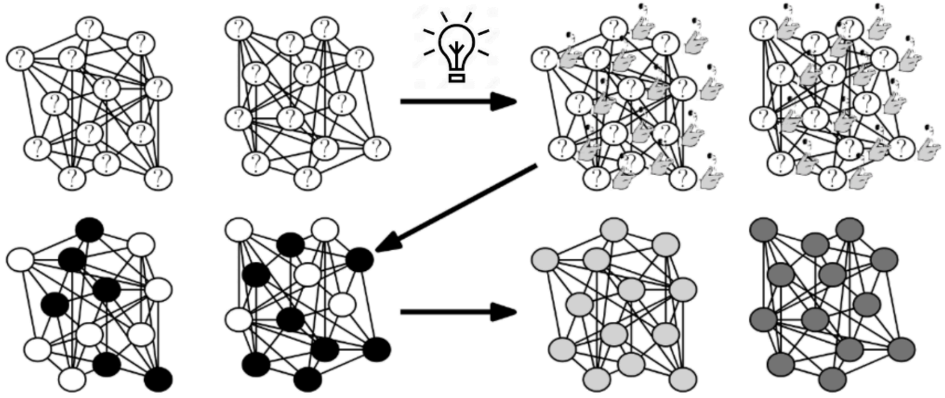


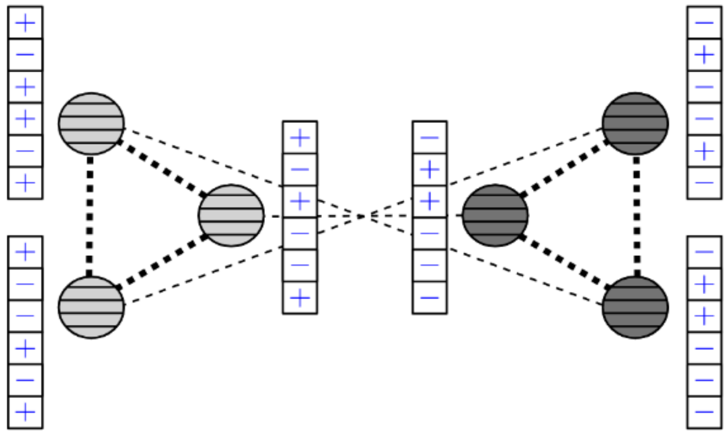
Process **variance** causes issues... (in 2018).

COMMUNITY SENSITIVE LABELING

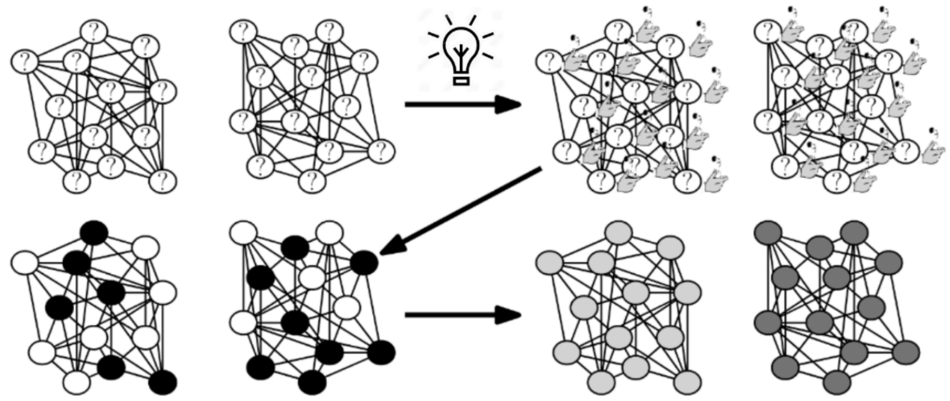


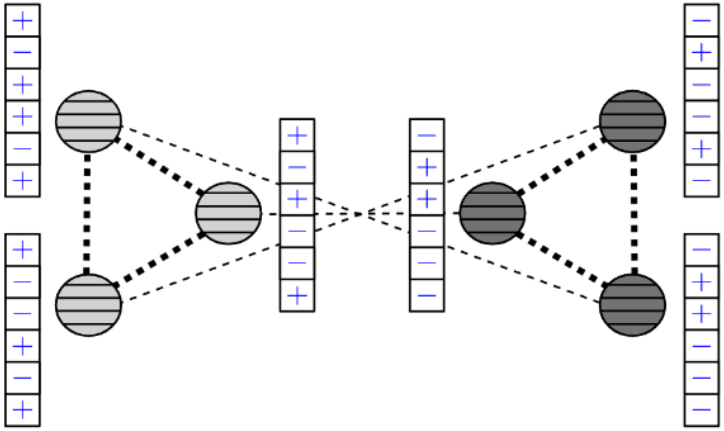
COMMUNITY SENSITIVE LABELING





COMMUNITY SENSITIVE LABELING

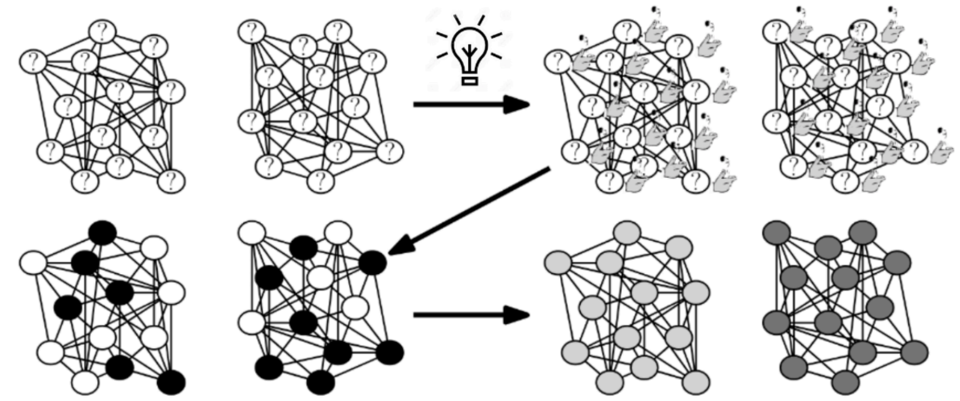


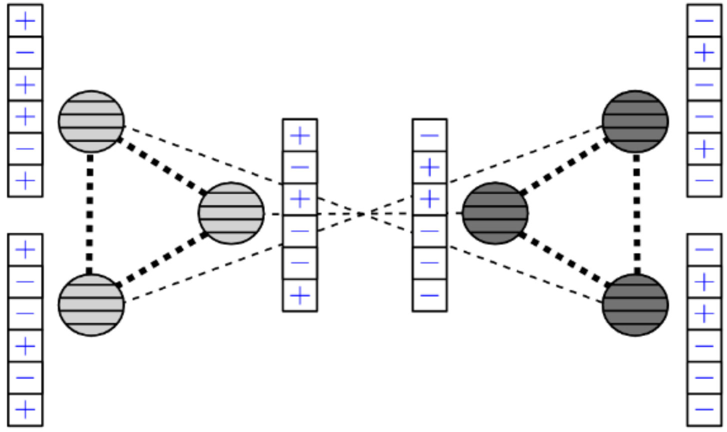


CSL. Run m **independent copies** of the Averaging. Each node u at time t has *signature*:

$$\begin{pmatrix} \text{sign}(x_u^{(1)}(t)) \\ \dots \\ \text{sign}(x_u^{(m)}(t)) \end{pmatrix}$$

COMMUNITY SENSITIVE LABELING





CSL. Run m **independent copies** of the Averaging. Each node u at time t has *signature*:

$$\begin{pmatrix} \operatorname{sign}(x_u^{(1)}(t)) \\ \dots \\ \operatorname{sign}(x_u^{(m)}(t)) \end{pmatrix}$$

Thm (Romans+P. Manurangsi+P. Raghavendra). G regular SBM s.t. $d\epsilon^4 \gg b \log n$. After $\Theta(\log n)$ rounds CSL with $m = \Theta(\epsilon^{-1} \log n)$ labels all nodes but $\leq \sqrt{\epsilon n}$ s.t. labels

- agree $> \frac{5}{6}$ in same community
- disagree $< \frac{5}{6}$ in different communities

THANK YOU

and thanks to Luca, from all his Roman colleagues