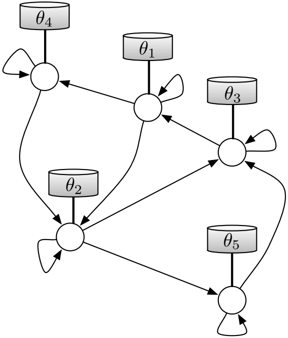
Network Science
Models and Distributed Algorithms

IST-CMU Phd course
João Xavier
TA: João Martins

October 3, 2016

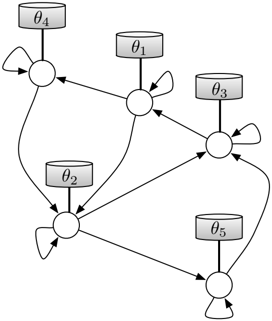
Consensus in static directed networks



- n agents; agent i holds $\theta_i \in \mathbf{R}$
- communication network is static and directed
- communication happens in discrete time $t = 0, 1, 2, 3, \ldots$
- goal: compute the average

$$\overline{\theta} = \frac{\theta_1 + \cdots + \theta_n}{n}$$

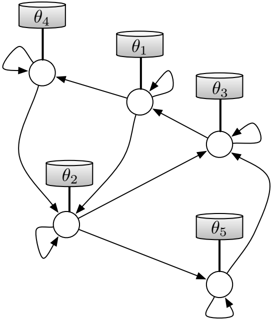
- we model the network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:
  - $\mathcal{V} = \{1, 2, \ldots, n\}$ is set of agents
  - $\mathcal{E}$ is set of communication channels
- agent i can send messages to agent j if and only if $(i, j) \in \mathcal{E}$
- example:



$$\begin{aligned}
\mathcal{V} &= \{1, 2, 3, 4, 5\} \\
\mathcal{E} &= \{(1,1), (1,2), (1,4), (2,2), (2,3), (2,5), (3,1), (3,3) \\
&\quad (4,2), (4,4), (5,3), (5,5)\}
\end{aligned}$$

- we also use the notation: $i \to j \quad \Leftrightarrow \quad (i, j) \in \mathcal{E}$

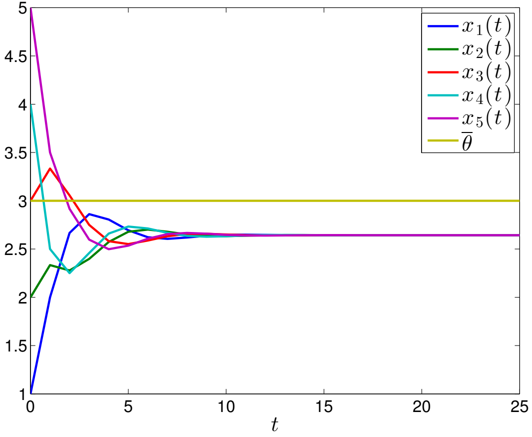
Naive scheme with row stochastic matrix



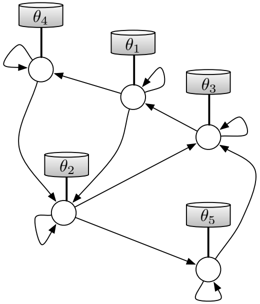
- agents repeatedly compute averages of themselves and neighbors
- $x_i(0) := \theta_i$ and

$$\begin{cases} x_1(t+1) &= \frac{x_1(t)+x_3(t)}{2} \\ x_2(t+1) &= \frac{x_1(t)+x_2(t)+x_4(t)}{3} \\ x_3(t+1) &= \frac{x_2(t)+x_3(t)+x_5(t)}{3} \\ x_4(t+1) &= \frac{x_1(t)+x_4(t)}{2} \\ x_5(t+1) &= \frac{x_2(t)+x_5(t)}{2} \end{cases}$$

- naive scheme doesn't work:



- how can we fix this?



- naive scheme in matrix form: $x(0) = \underbrace{(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)}_{\theta}$

$$\underbrace{\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \\ x_5(t+1) \end{bmatrix}}_{x(t+1)} = \underbrace{\begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}}_{x(t)}$$

- important: in distributed algorithms, $W_{ij} = 0$ whenever $j \not\to i$

- matrix W is row-stochastic:

$$W\mathbf{1} = \mathbf{1}$$

- unrolling the recursion $x(t+1) = Wx(t)$ gives

$$x(t) = W^t \theta$$

  for all $t \geq 0$

- analysis boils down to analyzing the powers of W

- W is a primitive matrix with $\rho(W) = 1$

- from Perron-Frobenius theorem:

$$W = \underbrace{\begin{bmatrix} \mathbf{1} & s_2 & \cdots & s_n \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} 1 & & & \\ & J_{\lambda_2} & & \\ & & \ddots & \\ & & & J_{\lambda_p} \end{bmatrix}}_{J} \underbrace{\begin{bmatrix} w^T \\ \widetilde{s}_2^T \\ \vdots \\ \widetilde{s}_n^T \end{bmatrix}}_{S^{-1}}$$

with $w > 0$, $w^T\mathbf{1} = 1$, and $|\lambda_i| < 1$ for $i = 2, \ldots, p$

- it follows that

$$W^t = \underbrace{\begin{bmatrix} \mathbf{1} & s_2 & \cdots & s_n \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} 1 & & & \\ & J_{\lambda_2}^t & & \\ & & \ddots & \\ & & & J_{\lambda_p}^t \end{bmatrix}}_{J^t} \underbrace{\begin{bmatrix} w^T \\ \widetilde{s}_2^T \\ \vdots \\ \widetilde{s}_n^T \end{bmatrix}}_{S^{-1}}$$

- we reduced the analysis from powers of matrices to powers of Jordan blocks

- how does the sequence of powers of a Jordan block behave?

- fact[1]: if

$$J_\lambda = \begin{bmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \\ & & & & \lambda \end{bmatrix} \in \mathbf{C}^{n\times n},$$

  then

$$J_\lambda^t \underset{t\to\infty}{\to} \begin{cases} 0 & \text{, if } |\lambda| < 1 \\ \infty & \text{, if } |\lambda| > 1 \\ \infty & \text{, if } |\lambda| = 1 \text{ and } n > 1 \\ \text{cycle} & \text{, if } |\lambda| = 1, \lambda \neq 1, \text{ and } n = 1 \\ 1 & \text{, if } \lambda = 1, \text{ and } n = 1 \end{cases}$$

---

[1] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, p. 630;R. Horn, C. R. Johnson, *Matrix Analysis*.Theorem 5.6.12, p. 298.

- we conclude:

$$W^t \underset{t\to\infty}{\to} \mathbf{1}w^T$$

  and

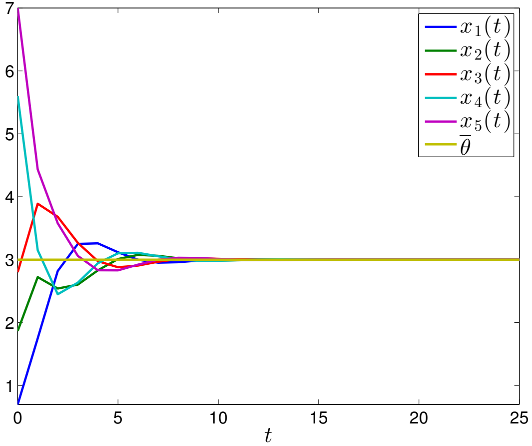
$$x(t) \underset{t\to\infty}{\to} (w^T\theta)\mathbf{1}$$

- interpretation: agents converge to a convex combination of the initial data θ

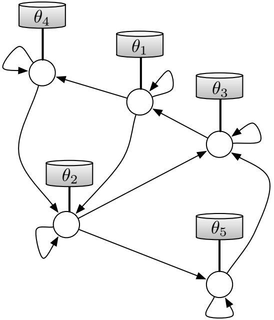
- a fix: if n and w_i are known at each agent i, initialize

$$x_i(0) = \frac{\theta_i}{n w_i}$$

- back to example in page 5:



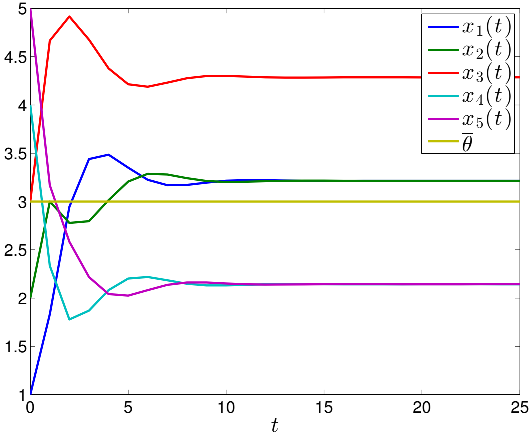
Naive scheme with column stochastic matrix



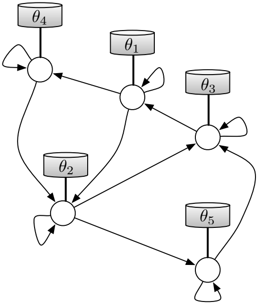
- agents repeatedly send fractions of their state to neighbors
- $x_i(0) := \theta_i$ and

$$\begin{cases} x_1(t+1) &= \frac{1}{3}x_1(t) + \frac{1}{2}x_3(t) \\ x_2(t+1) &= \frac{1}{3}x_1(t) + \frac{1}{3}x_2(t) + \frac{1}{2}x_4(t) \\ x_3(t+1) &= \frac{1}{3}x_2(t) + \frac{1}{2}x_3 + \frac{1}{2}x_5(t) \\ x_4(t+1) &= \frac{1}{3}x_1(t) + \frac{1}{2}x_4(t) \\ x_5(t+1) &= \frac{1}{3}x_2(t) + \frac{1}{2}x_5(t) \end{cases}$$

- naive scheme doesn't work:



- how can we fix this?



- naive scheme in matrix form: $x(0) = \underbrace{(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)}_{\theta}$

$$\underbrace{\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ x_3(t+1) \\ x_4(t+1) \\ x_5(t+1) \end{bmatrix}}_{x(t+1)} = \underbrace{\begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \end{bmatrix}}_{x(t)}$$

- important: in distributed algorithms, $W_{ij} = 0$ whenever $j \not\to i$

- matrix W is column-stochastic:

$$\mathbf{1}^T W = \mathbf{1}^T$$

- unrolling the recursion $x(t+1) = Wx(t)$ gives

$$x(t) = W^t \theta$$

  for all $t \geq 0$

- analysis boils down to analyzing the powers of W

- W is a primitive matrix with $\rho(W) = 1$

- from Perron-Frobenius theorem:

$$W = \underbrace{\begin{bmatrix} v & s_2 & \cdots & s_n \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} 1 & & & \\ & J_{\lambda_2} & & \\ & & \ddots & \\ & & & J_{\lambda_p} \end{bmatrix}}_{J} \underbrace{\begin{bmatrix} \mathbf{1}^T \\ \widetilde{s}_2^T \\ \vdots \\ \widetilde{s}_n^T \end{bmatrix}}_{S^{-1}}$$

with $v > 0$, $\mathbf{1}^T v = 1$ and $|\lambda_i| < 1$ for $i = 2, \ldots, p$

- we conclude:

$$W^t \underset{t\to\infty}{\to} v\mathbf{1}^T$$

and

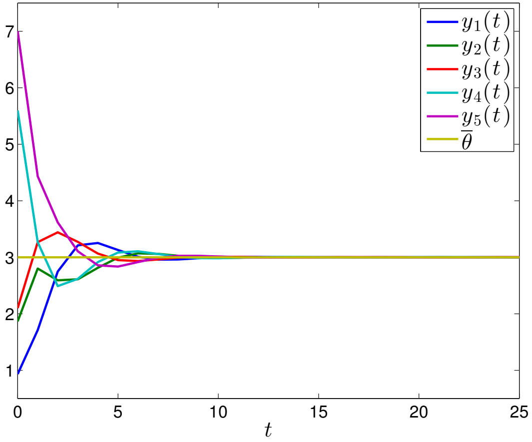
$$x(t) \underset{t\to\infty}{\to} (\mathbf{1}^T\theta)v$$

- interpretation: agents converge to a fraction of the sum of initial data θ

- a fix: if n and v_i are known at each agent i, he computes on the side

$$y_i(t) = \frac{x_i(t)}{nv_i}$$

- back to example in page 14:



- another fix: the push-sum algorithm

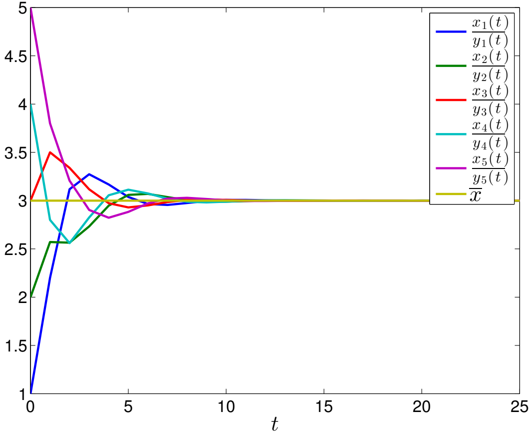
- initialize $x(0) = x$, $y(0) = \mathbf{1}$, and iterate

$$\begin{aligned} x(t+1) &= Wx(t) \\ y(t+1) &= Wy(t) \end{aligned}$$

- at each agent i:

$$\frac{x_i(t)}{y_i(t)} \underset{t\to\infty}{\rightarrow} \frac{v_i \mathbf{1}^T x}{v_i \mathbf{1}^T \mathbf{1}} = \frac{\mathbf{1}^T x}{n} = \overline{x}$$

- back to example in page 14:



To know more

- Push-sum algorithm
  - D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," *IEEE Symp. on Found. of Comp. Science*, 2003.
- Matrix analysis (Jordan forms, EVD, SVD)
  - C. Meyer, *Matrix Analysis and Applied Linear Algebra*.
  - R. Horn, C. R. Johnson, *Matrix Analysis*.