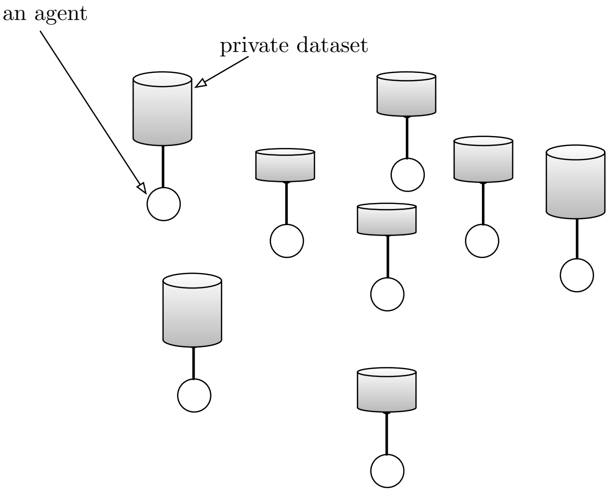
Network Science
Models and Distributed Algorithms

IST-CMU Phd course
João Xavier
TA: João Martins

October 3, 2016

The big picture: multi-agent system

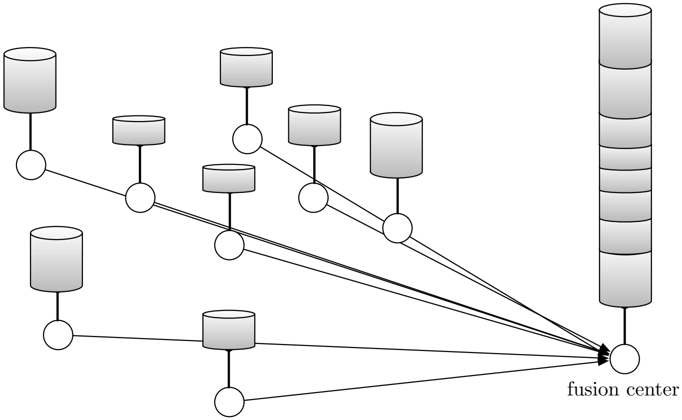


- each agent has a private dataset
- agents want to compute some quantity from the **total** dataset

Multi-agent systems model:

- robotic teams
- wireless sensor networks
- smart grids
- vehicular networks
- computer networks
- wireless camera networks
- cognitive radio
- distributed learning for Big Data

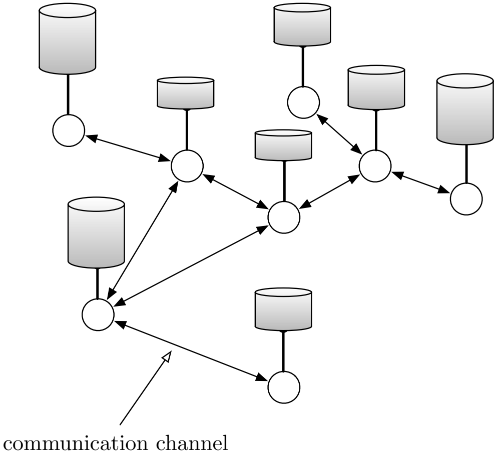
Centralized approach



- agents send data to a fusion center that computes the quantity
- this approach is not robust and doesn't scale

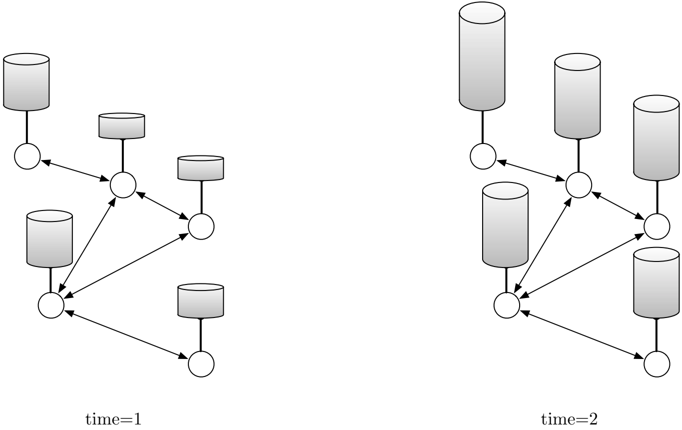
This course focuses on **distributed** approaches

Distributed approach

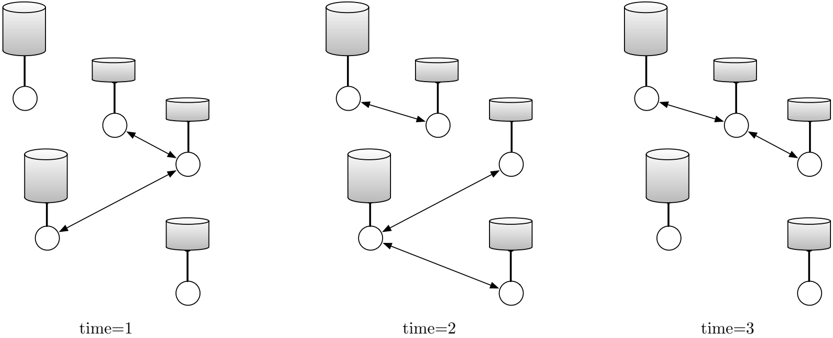


- a communication network links the agents
- agents collaborate through the network to compute the quantity

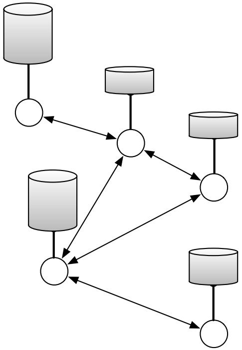
Many challenges



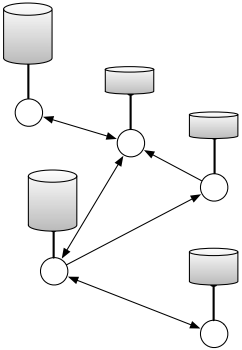
- datasets change or grow
- happens in online and tracking applications



- communication network may change randomly
- happens with wireless channels or randomized protocols (ex: gossip)



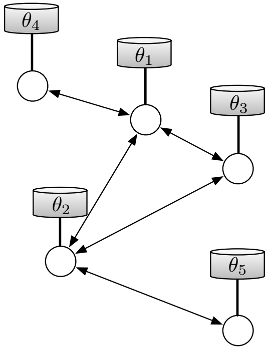
undirected network



directed network

- communication network may be directed
- happens when channels do not have feedback

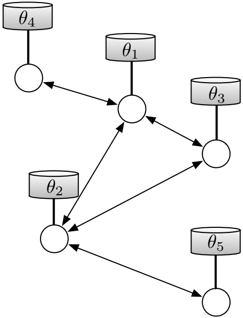
Example: consensus



- agent i holds $\theta_i \in \mathbf{R}$
- goal: compute the average

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

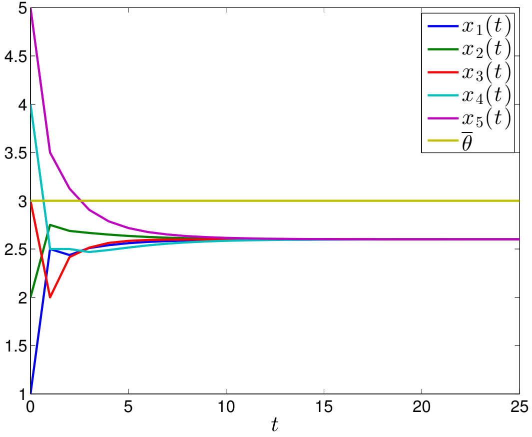
- applications: flocking of robots, clock synchronization, data fusion



- naive scheme: agents repeatedly compute local averages
- $x_i(0) := \theta_i$ and

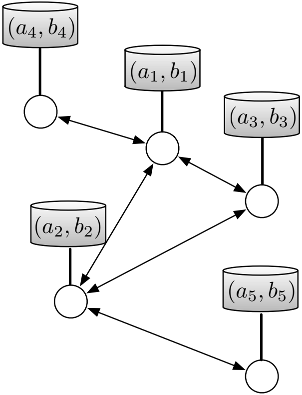
$$\begin{cases} x_1(t+1) &= \frac{x_1(t)+x_2(t)+x_3(t)+x_4(t)}{4} \\ x_2(t+1) &= \frac{x_1(t)+x_2(t)+x_3(t)+x_5(t)}{4} \\ x_3(t+1) &= \frac{x_1(t)+x_2(t)+x_3(t)}{3} \\ x_4(t+1) &= \frac{x_1(t)+x_4(t)}{2} \\ x_5(t+1) &= \frac{x_2(t)+x_5(t)}{2} \end{cases}$$

- naive scheme doesn't work:



- how can we fix this?

Example: distributed logistic regression

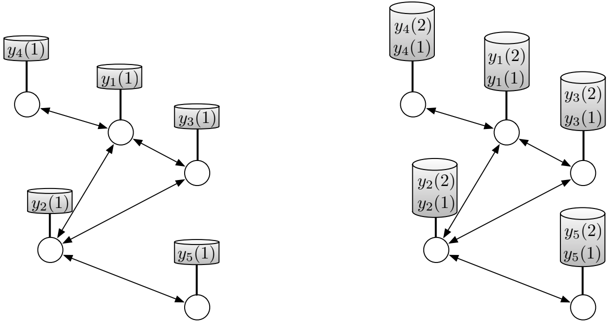


- agent i holds $(a_i, b_i) \in \mathbf{R}^d \times \{\pm 1\}$
- goal: find optimum classifier x by solving

$$\underset{x \in \mathbf{R}^d}{\text{minimize}} \quad \sum_{i=1}^{n} \underbrace{-b_i a_i^T x + \log\left(1 + e^{a_i^T x}\right)}_{f_i(x)}$$

- how can we solve it without sharing the datasets?

Example: distributed detection



time=1

time=2

- nature is in state $\mathcal{H}_0$ or $\mathcal{H}_1$
- agent i observes data stream $y_i(1), y_i(2), y_i(3), \ldots$
- distribution of $y_i(t)$ depends on active hypothesis:

$$y_i(t) \sim \begin{cases} q_i, & \text{under } \mathcal{H}_0 \\ p_i, & \text{under } \mathcal{H}_1 \end{cases}$$

- goal: decide which hypothesis $\mathcal{H}_0$ or $\mathcal{H}_1$ is active
- at time t, centralized detector would decide as:

$$\frac{1}{t}\sum_{s=1}^{t}\frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{p_i\left(y_i(s)\right)}{q_i\left(y_i(s)\right)}\right)\underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} 0$$

- centralized detector would know all observations, at all times
- how can we decide in a distributed manner?

Course outline

Part 1: static networks

- background: graphs and Perron-Frobenius theory
- consensus with undirected and directed communications
- distributed optimization
- distributed detection and estimation

Part 2: dynamic networks

- background: random matrix theory and martingales
- consensus with undirected and directed communications
- distributed optimization
- distributed detection and estimation

- only a slice of current research!

- no textbook: we'll use book chapters but mostly research papers

- students should be familiar with:
  - matrix analysis (ex: Jordan forms, EVD, SVD)
  - probability (ex: expectation operator, covariance matrix)
  - optimization (ex: gradient, Hessian)

- grade = 60% (homeworks) + 40% (24h take-home exam)

- homeworks explore variations of the lectures' topics