Networked Control Systems (ME-427)- Exercise session 13

Prof. G. Ferrari Trecate

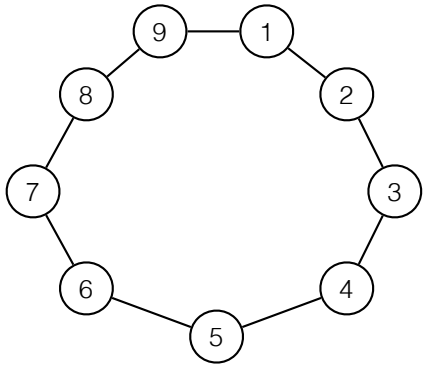
1. **Problem 1** (Distributed least-squares in a sensor network)

   We want to estimate a function $y = f(x)$, $f(x) : \mathbb{R} \to \mathbb{R}$ from the 9 data points (x_i, y_i), $i = 1, \dots, 9$ defined by

   $$x = \begin{bmatrix} 1.4 & 1.6 & 0.05 & 1.7 & 1 & 0.05 & 0.3 & -0.8 & 1.7 \end{bmatrix}^T,$$
   $$y = \begin{bmatrix} -1.9 & -2.3 & 1 & -2.7 & -1.2 & 0.8 & 0.6 & 2.7 & -2.4 \end{bmatrix}^T.$$

   Assume the estimate is given by $f'_\theta(x) = \theta_1 x + \theta_2$, $\theta = [\theta_1, \theta_2]^T$.

   Design a consensus algorithm for computing the least-square estimate using the ring network in figure, where nodes are sensors (each storing a single datapoint) and communication links are bidirectional. Assume weights are assigned according to the Metropolis-Hastings model.

   Code the algorithm in MatLab and show that it is correct by comparing the obtained parameter with those produced by the (centralized) formula

   $$\theta_{LS} = (G^T G)^{-1} G^T y, \quad y = [y_1, \dots, y_9]^T, \quad G = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_9 & 1 \end{bmatrix}$$

   Hint: Download `Exercise13problem1.m` from moodle. It contains relevant data structures for performing consensus iterations. In particular

   - x_{gg} is a cell array and $x_{gg}\{i, k\}$ is a storeplace for the matrices $x_i^{gg}(k)$ seen in the lectures.
   - Similarly, $x_{gy}\{i, k\}$ is a storeplace for $x_i^{gy}(k)$

   Solution: See the MatLab file available on moodle. In particular, the adjacency matrix resulting from the Metropolis-Hasting model is

   $$A = \frac{1}{3} \begin{bmatrix} 1 & 1 & 0 & 0 & \dots & 0 & 1 \\ 1 & 1 & 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & 1 & 1 & \dots & 0 & 0 \\ & & & & & & \\ 0 & 0 & 0 & 0 & \dots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & \dots & 1 & 1 \end{bmatrix}$$

2. **Problem 2** (Consensus on time-varying graph)

   Consider a team of 9 agents in the set $V = \{0, \dots, 8\}$. Each $v \in V$ stores a state $x_v(k) \in \mathbb{R}^2$, $k = 0, 1, 2, \dots$. The nodes update their state according to the following rules.

   - At each time k just one node v awakens. Node v and $\tilde{v} = [v+1]_{\text{mod}9}$ connect and both update their state to the value

     $$x_v(k+1) = x_{\tilde{v}}(k+1) = \frac{1}{2}\left(x_v(k) + x_{\tilde{v}}(k)\right)$$

     The notation $[x]_{\text{mod}9}$ is used the integer x modulo 9 (i.e. $[x]_{\text{mod}9} = n$ where $n \in \mathbb{N}$ verifies $0 \leq n \leq 8$ and $x = n + t9$ for some $t \in \mathbb{N}$. In MatLab: `n = mod(x,9)`).

     All other nodes update their state as $x_i(k+1) = x_i(k)$, $i \neq v$, $i \neq \tilde{v}$.

   - Node v awakens at times $v, v+9, v+18, v+27, \dots$

   Does the algorithm converge to consensus? Does it converge to average consensus? Justify your claims using the results presented in the lectures. Then, validate your conclusions by running simulations in MatLab starting from initial states $x_i(0) = [x_{i1}(0), x_{i2}(0)]$ where $x_{ij}(0)$ are randomly generated in the interval $[0, 10]$.

   Hint: For the simulations, adapt the code developed for Problem 1. Values $x_{ij}(0)$ can be generated in MatLab using `10*rand(1)`.

3. **Problem 3** (The equal-neighbor row-stochastic matrix for weighted directed graphs)

   Consider the same setup of Problem 2, but assume that each agent does not know the number of nodes present in the network (that is $n = 9$) and it has to compute $\frac{1}{n}\bar{x}$, where $\bar{x} \in \mathbb{R}^2$ is the consensus state reached in Problem 2. Adapt the MatLab code of Problem 2 to achieve the goal.