Network Science
IST-CMU PhD course
Fall 2016
Instructor: `jxavier@isr.ist.utl.pt`
TA: João Martins, `joaoa@andrew.cmu.edu`

Important: the homework is due the February 16. Send a scanned pdf file with your answers (or typed in LaTeX, if you prefer) to the TA's email.

Homework 5

1. *Consensus over digital channels: numerical simulation.* Consider a network of n agents in which each agent i encodes its opinion about some issue as a bit (binary digit) $\theta_i \in \{0, 1\}$. For example, θ_i can be the output of a decision algorithm that agent i ran to decide if an intruder is present ($\theta_i = 1$) or not ($\theta_i = 0$). The agents want to compute their average opinion

$$\overline{\theta} = \frac{1}{n} \sum_{i=1}^{n} \theta_i,$$

or, in a compact notation, $\overline{\theta} = \mathbf{1}^T \theta / n$, where $\theta = (\theta_1, \ldots, \theta_n)$.

Assume the agents are linked by a connected, undirected communication network. Any standard consensus algorithm can produce the desired $\overline{\theta}$ at all agents. For example, the algorithm given by $x(0) = \theta$ and

$$x(t+1) = x(t) - \alpha L x(t), \quad t \geq 0,. \tag{1}$$

can make $x(t) \to \overline{\theta}\mathbf{1}$ as $t \to \infty$. In (1), L is the $n \times n$ laplacian matrix of the network, and $\alpha > 0$ is an appropriately chosen stepsize.

Model (1) assumes that the state $x_i(t) \in \mathbf{R}$, sent by each agent i to its neighbors, passes without distortion through the channels linking them. However, in practice most communication channels are digital which, roughly speaking, means that they can pass only a finite number of bits (per channel use); they cannot directly pass real numbers such as $x_i(t) \in \mathbf{R}$ (per channel use). The state $x_i(t) \in \mathbf{R}$ has first to be quantized, that is, approximated by a finite number of bits. In this problem, we will consider highly limited digital channels: at each time slot $t = 1, 2, \ldots$, each channel can pass only one bit.

To handle such channels, we will consider the following distributed algorithm: $x(0) = \theta$ and

$$x(t+1) = x(t) - \alpha(t+1) L y(t+1), \quad t \geq 0, \tag{2}$$

where $y(t) = (y_1(t), \ldots, y_n(t))$ is a binary vector, that is, each $y_i(t)$ is a bit, $y_i(t) \in \{0, 1\}$. In (2), the bit $y_i(t+1) \in \{0, 1\}$ is the information that agent i sends to its neighbors at time t (compare with (1) where the state $x_i(t) \in \mathbf{R}$ is the information that agent i sends to its neighbors at time t). The bit $y_i(t+1) \in \{0, 1\}$ depends probabilistically on the state $x_i(t)$ as follows:

- if $0 \leq x_i(t) < d_{\max}\alpha(t+1)$, then $y_i(t+1) = 0$;

- if $d_{\max}\alpha(t+1) \leq x_i(t) \leq 1 - d_{\max}\alpha(t+1)$, then $y_i(t+1) = 1$ with probability $x_i(t)$ (and, of course, $y_i(t) = 0$ with probability $1 - x_i(t)$);
- if $1 - d_{\max}\alpha(t+1) < x_i(t) \leq 1$, then $y_i(t+1) = 1$.

Here, $d_{\max}$ is the maximum of all nodes' degrees, and $\alpha(t)$ is the stepsize sequence also appearing in (2) (to be discussed latter). Thus, if the state $x_i(t)$ is sufficiently close to 0, agent i sends the bit $y_i(t+1) = 0$ to its neighbors. If the state $x_i(t)$ is sufficiently close to 1, agent i sends the bit $y_i(t+1) = 1$ to its neighbors. Finally, if the state $x_i(t)$ is neither too close to 0 nor to 1, agent i creates a fresh random bit $y_i(t+1) \in \{0, 1\}$ with mean value $x_i(t)$—-specifically, $\mathbf{E}\left(y_i(t+1) \,|\, x_i(t)\right) = x_i(t)$—and sends it to its neighbors. The random bit is generated independently from all other previous ones.

Implement in Matlab algorithm (2) for the example in figure 1.

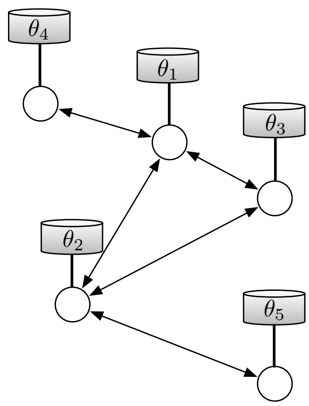

Figure 1: A set of five agents linked by undirected communication channels. Agent i holds θ_i. We consider $\theta_1 = 1$, $\theta_2 = 1$, $\theta_3 = 0$, $\theta_4 = 0$, $\theta_5 = 1$. So, $\overline{\theta} = 0.6$.

Use the stepsize sequence

$$\alpha(t) = \frac{1}{2d_{\max}t^{0.6}}, \quad t \geq 1,$$

and produce a plot of one run of the algorithm. You should obtain something similar to figure 2.

Note: in Matlab, you can generate a random bit $y \in \{0, 1\}$ with mean $x \in [0, 1]$ with the command `y = (rand <= x)`. Also, for the graph in figure 1, we have $d_{\max} = 3$.

2. *Consensus over digital channels: theoretical analysis (or, yes, still another nice application of the Robbins-Siegmund's supermartingale convergence lemma).* In this problem, you will prove that algorithm (2) works, that is, $x(t) \to \overline{\theta}\mathbf{1}$ (almost surely) as $t \to \infty$ for any undirected, connected network, provided the stepsize sequence satisfies $0 < \alpha(t) \leq 1/(2d_{\max})$ for all $t \geq 1$, $\sum_{t\geq 1} \alpha(t) = \infty$, and $\sum_{t\geq 1} \alpha(t)^2 < \infty$.

(a) Start by showing that the algorithm (2) is well-defined, that is, show that

$$0 \leq x_i(t) \leq 1$$

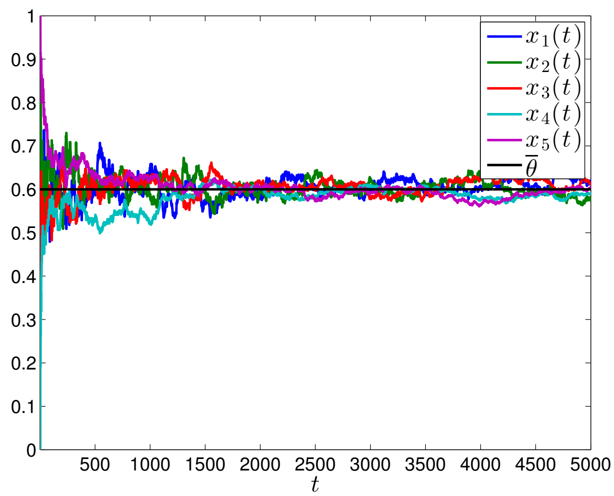

Figure 2: A run of the algorithm (2) for the network in figure 1.

for all $t \geq 0$ and $i = 1, \ldots, n$, that is, the agents' states are always confined to the interval $[0, 1]$ (if this was not the case, the algorithm (2) would be ill-defined: how could you generate a random bit $y \in \{0, 1\}$ with mean $x < 0$ or $x > 1$?).

(b) Do the usual orthogonal split of the network state: $x(t) = \overline{x}(t)\mathbf{1} + U\widehat{x}(t)$, where $\overline{x}(t) := \mathbf{1}^T x(t)/n \in \mathbf{R}$, $\widehat{x}(t) := U^T x(t) \in \mathbf{R}^{n-1}$, and

$$L = \begin{bmatrix} U & \frac{1}{\sqrt{n}}\mathbf{1} \end{bmatrix} \begin{bmatrix} \Lambda & \\ & 0 \end{bmatrix} \begin{bmatrix} U^T \\ \frac{1}{\sqrt{n}}\mathbf{1}^T \end{bmatrix}$$

is an eigenvalue decomposition of the laplacian matrix. Note that, because the graph is assumed connected, all diagonal entries of the diagonal matrix

$$\Lambda = \begin{bmatrix} \lambda_{n-1} & & & \\ & \lambda_{n-2} & & \\ & & \ddots & \\ & & & \lambda_1 \end{bmatrix}$$

are positive. Show that $\overline{x}(t) = \overline{\theta}$ for all $t \geq 0$.

(c) Since $\overline{x}(t) = \overline{\theta}$ for all $t \geq 0$, we need only to show that $\widehat{x}(t) \to 0$ (almost surely) as $t \to \infty$ to obtain our goal: $x(t) \to \overline{\theta}\mathbf{1}$.

Show that (2) implies

$$\widehat{x}(t+1) = \widehat{x}(t) - \alpha(t+1)\Lambda\widehat{y}(t+1) \tag{3}$$

where $\widehat{y}(t) := U^T y(t)$. Conclude that

$$\widehat{x}_i(t+1) = \widehat{x}_i(t) - \alpha(t+1)\lambda_i\widehat{y}_i(t+1), \tag{4}$$

for $i = 1, \ldots, n-1$, where $\widehat{x}(t) = (\widehat{x}_1(t), \ldots, \widehat{x}_{n-1}(t))$ and $\widehat{y}(t) = (\widehat{y}_1(t), \ldots, \widehat{y}_{n-1}(t))$.

(d) Let $\mathcal{F}(t) := \{x(0), y(1), x(1), y(2), \ldots, x(t-1), y(t), x(t)\}$ and note that $\mathcal{F}(t)$ does not contain $y(t+1)$. Recalling from problem 1 how $y_i(t+1)$ is generated from $x_i(t)$, we see that

$$\mathbf{E}\left(y_i(t+1) \,|\, \mathcal{F}(t)\right) = \begin{cases} 0 & \text{, if } 0 \le x_i(t) < d_{\max}\alpha(t+1) \\ x_i(t) & \text{, if } d_{\max}\alpha(t+1) \le x_i(t) \le 1 - d_{\max}\alpha(t+1) \\ 1 & \text{, if } 1 - d_{\max}\alpha(t+1) < x_i(t) \le 1. \end{cases} \tag{5}$$

We can express (5) more compactly as $\mathbf{E}\left(y_i(t+1) \,|\, \mathcal{F}(t)\right) = \phi_t\left(x_i(t)\right)$ where $\phi_t : [0,1] \to \mathbf{R}$ is defined as

$$\phi_t(x) = \begin{cases} 0 & \text{, if } 0 \le x < d_{\max}\alpha(t+1) \\ x & \text{, if } d_{\max}\alpha(t+1) \le x \le 1 - d_{\max}\alpha(t+1) \\ 1 & \text{, if } 1 - d_{\max}\alpha(t+1) < x \le 1. \end{cases}$$

Similarly, in vector form, we have $\mathbf{E}\left(y(t+1) \,|\, \mathcal{F}(t)\right) = \Phi_t\left(x(t)\right)$ where $\Phi_t : \mathbf{R}^n \to \mathbf{R}$ is defined as

$$\Phi_t(x_1, \ldots, x_n) = \begin{bmatrix} \phi_t(x_1) \\ \phi_t(x_2) \\ \vdots \\ \phi_t(x_n) \end{bmatrix}$$

Show that (4) implies

$$\mathbf{E}\left(\widehat{x}_i(t+1)^2 \,|\, \mathcal{F}(t)\right) = \widehat{x}_i(t)^2 - 2\alpha(t+1)\lambda_i \widehat{x}_i(t) u_i^T \Phi_t\left(x(t)\right) + \lambda_i^2 \alpha(t+1)^2 \mathbf{E}\left(\widehat{y}_i(t+1)^2 \,|\, \mathcal{F}(t)\right), \tag{6}$$

where u_i^T is the ith row of matrix U^T, i.e.,

$$U^T = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_{n-1}^T \end{bmatrix} \in \mathbf{R}^{(n-1)\times n}.$$

(e) Show that (6) implies

$$\mathbf{E}\left(\widehat{x}_i(t+1)^2 \,|\, \mathcal{F}(t)\right) \le \widehat{x}_i(t)^2 - 2\alpha(t+1)\lambda_i \widehat{x}_i(t) u_i^T \Phi_t\left(x(t)\right) + n\lambda_i^2 \alpha(t+1)^2. \tag{7}$$

(f) Using $\Phi_t(x(t)) = x(t) + (\Phi_t(x(t)) - x(t))$, show that (7) can be rewritten as

$$\mathbf{E}\left(\widehat{x}_i(t+1)^2 \,|\, \mathcal{F}(t)\right) \le \widehat{x}_i(t)^2 - 2\alpha(t+1)\lambda_i \widehat{x}_i(t)^2 - 2\alpha(t+1)\lambda_i u_i^T \left(\Phi_t(x(t)) - x(t)\right) + n\lambda_i^2 \alpha(t+1)^2. \tag{8}$$

(g) Show that $|\phi_t(x) - x| \le d_{\max}\alpha(t+1)$ for all $t \ge 0$ and $x \in [0,1]$, and conclude that

$$\|\Phi_t(x) - x\| \le \sqrt{n} d_{\max}\alpha(t+1)$$

for all $t \ge 0$ and $x \in [0,1]^n$.

(h) Show that

$$\mathbf{E}\left(\widehat{x}_i(t+1)^2 \,|\, \mathcal{F}(t)\right) \le \widehat{x}_i(t)^2 - 2\alpha(t+1)\lambda_i \widehat{x}_i(t)^2 + 2\alpha(t+1)^2 \lambda_i \sqrt{n} d_{\max} + n\lambda_i^2 \alpha(t+1)^2$$

and conclude that $\widehat{x}_i(t) \to 0$.