Lecture 8

Consensus: design of graph weights, applications and generalizations

Textbook §5.4, §1.5.1, §11.1, §11.2, §11.4

Giancarlo Ferrari Trecate[1]

[1]Dependable Control and Decision Group
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
giancarlo.ferraritrecate@epfl.ch

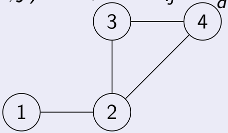
Design of graph weights. . . to guarantee consensus

The equal-neighbour model

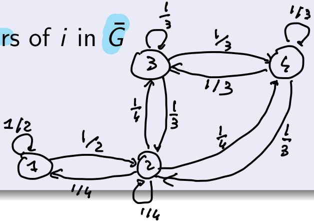
1) Given $\bar{G} = (V, \bar{E})$ undirected and connected, build the corresponding digraph $G = (V, E)$ with all self-loops

2) if $(i, j) \in E$, set $w_{ij} = \frac{1}{\bar{d}(i)+1}$ $\bar{d}(i) = \#$ neighbours of i in $\bar{G}$

$\bar{G}$



Compute $G \Rightarrow \ldots$



Definition. The degree matrix of $\bar{G}$ is

$$\bar{D} = \begin{bmatrix} \bar{d}(1) & & \\ & \ddots & \\ & & \bar{d}(n) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & & & \\ & \frac{1}{4} & & \\ & & \frac{1}{3} & \\ & & & \frac{1}{3} \end{bmatrix}$$

Proposition

Let $\bar{A}$ be binary adjacency matrix of $\bar{G}$ and A be the weighted adjacency matrix of G. Then

$$A = \left(\bar{D} + I\right)^{-1} \left(\bar{A} + I\right) \tag{1}$$

For the example

$$\bar{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \rightarrow A = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

Proposition

The matrix A in (1) is

- row stochastic
- primitive $\Leftrightarrow$ $\bar{G}$ is connected
- doubly stochastic if $\bar{G}$ is balanced (that is if the degree of each node is identical)

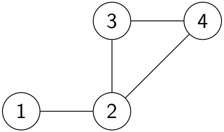
1hL11

The Metropolis-Hasting model

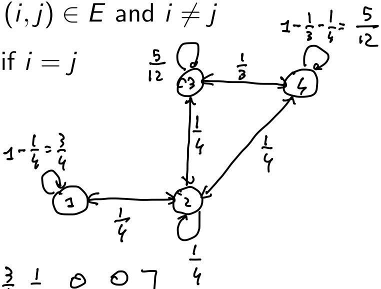
1. Given $\bar{G} = (V, \bar{E})$ undirected, build the corresponding digraph $G = (V, E)$ with all self-loops
2. If $(i, j) \in E$ set

$$w_{ij} = \begin{cases} \frac{1}{1+\max\left(\bar{d}(i), \bar{d}(j)\right)} & \text{if } (i,j) \in \bar{E} \text{ and } i \neq j \\ 1 - \sum_{(i,h)\in\bar{E}} w_{ih} & \text{if } i = j \end{cases}$$

$\bar{G}$



Compute $G \Rightarrow \ldots$



$$\bar{D} = \begin{bmatrix} 1 & & & \\ & 3 & & \\ & & 2 & \\ & & & 2 \end{bmatrix}$$

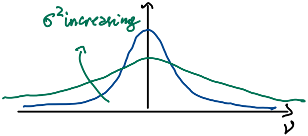
$$A = \begin{bmatrix} \frac{3}{4} & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{5}{12} & \frac{1}{3} \\ 0 & \frac{1}{4} & \frac{1}{3} & \frac{5}{12} \end{bmatrix}$$

Proposition

Let A be the weighted adjacency matrix of G. Then,

- A is symmetric and doubly stochastic
- A is primitive $\Leftrightarrow$ $\bar{G}$ is connected

Applications of consensus algorithms



1. Parameter estimation: sensor networks with noisy measurements

Sensor i measures $y_i = \vartheta + v_i \quad v_i \sim \mathcal{N}(0, \sigma_i^2) \quad i = 1, \ldots, n$ where

- $\mathcal{N}(m, \sigma^2)$ is the Gaussian density with mean m and variance σ^2
- $\vartheta \in \mathbb{R}$
- v_i are independent
  - $\frac{1}{\sigma_i} \simeq$ sensor accuracy

Best estimate: BLUE (Best Linear Unbiased Estimator)

$$\hat{\vartheta} = \sum_{i=1}^{n} \alpha_i y_i \quad \alpha_i = \frac{\sigma_i^{-2}}{\sum_{i=1}^{n} \sigma_i^{-2}}$$

"Unbiased" means $\mathbb{E}[\hat{\vartheta}] = \vartheta$. Check: $\mathbb{E}\left[\sum_{i=1}^{n} \alpha_i y_i\right] = \sum_{i=1}^{n} \alpha_i \underbrace{\mathbb{E}[y_i]}_{\vartheta} = \vartheta \underbrace{\sum_{i=1}^{n} \alpha_i}_{=1} = \vartheta$

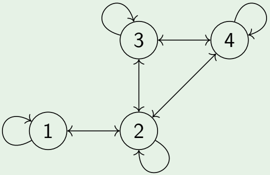
How to compute BLUE in a distributed way?

- Write $\hat{\vartheta}$ as the ratio of two averages

$$\hat{\vartheta} = \frac{\frac{1}{n}\sum_{i=1}^{n} \sigma_i^{-2} y_i}{\frac{1}{n}\sum_{i=1}^{n} \sigma_i^{-2}}$$

- Run two average consensus algorithms (one for the numerator, one for the denominator) at each node in a graph G modeling the network connecting sensors

Example



- Assume Metropolis-Hasting weights
- $x_i(0) = \sigma_i^{-2} y_i$ $\quad \hat{x}_i(0) = \sigma_i^{-2}$
- Iterations $x^+ = Ax$

  $\hat{x}^+ = A\hat{x}$
- At each node i compute

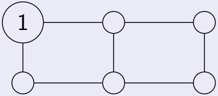
$$\hat{\vartheta}_i = \frac{x_i(\infty)}{\hat{x}_i(\infty)}$$

Applications of consensus algorithms

2. Node counting in a network

Problem

Make each node aware of the number of nodes in a network through distributed computations



n nodes

Consensus algorithm

- Set $x_1(0) = 1$ and $x_i(0) = 0 \quad i > 1$
- Run average consensus
- $x_i(\infty) = \frac{1}{n} \quad \forall i \in V$ ($G = (V, E)$ is the network)
- retrive, at each node, $n = \frac{1}{x_i(\infty)}$

Applications of consensus algorithms

3. Consensus on vector-valued and matrix-valued quantities

Consider the system

This is $(Ax)_i$

$$x_i^+ = \underbrace{\sum_{j \in \mathcal{N}^{\text{out}}(i)} A_{ij} x_j}_{} \quad i = 1, \ldots, n \qquad (\blacksquare)$$

where $x_i \in \mathbb{R}^m$ or $x_i \in \mathbb{R}^{m \times p}$ and $A_{ij} \geq 0$ are scalars.

Ex: $x_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix}$

$\nearrow$ $(\blacksquare)$ gives

$$\begin{bmatrix} x_{i,1}^+ \\ x_{i,2}^+ \end{bmatrix} = \begin{bmatrix} \sum_{j \in \mathcal{N}^{out}(i)} A_{ij} x_{j,1} \\ \sum_{j \in \mathcal{N}^{out}(i)} A_{ij} x_{j,2} \end{bmatrix}$$

each row encodes a consensus algorithm over scalar variables

The same results on consensus, seen for $x_i \in \mathbb{R}$, hold!
For instance, if A is primitive and stochastic,

$$\exists \bar{x}: \; x_i \to \bar{x} \text{ as } k \to +\infty$$

If, moreover, A is doubly stochastic

$$x_i \to \text{ average } (x_i(0), i = 1, \ldots, n) \text{ as } k \to \infty$$

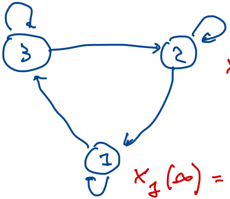
Remark

Consensus for $x_i \in \mathbb{R}^2$ means that $\exists \bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$ such that

$\begin{bmatrix} x_{i,1}(k) \\ x_{i,2}(k) \end{bmatrix} \rightarrow \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$ as $k \rightarrow +\infty$

But $\bar{x}_1$ can be different from $\bar{x}_2$ (not to be confused with $x \rightarrow \alpha \mathbb{1}_n$ as $k \rightarrow +\infty$)

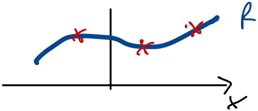
$x_3(\infty) = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$



$x_2(\infty) = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$

$x_1(\infty) = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$

Applications of consensus algorithms



4. Distributed computation of Least-Squares (LS) regression

LS problem: estimate a function $y = f(x)$ $f : \mathbb{R}^{n_x} \to \mathbb{R}$ from a dataset $D = \{(x_j, y_j) \quad j = 1, \ldots, n\}$

Review: LS estimation

- Parametrize the estimate as $f_\vartheta(x) = \sum_{i=1}^{M} \vartheta_i g_i(x)$
- $g_i(\cdot)$: given basis functions (e.g. polynomials)
- $\vartheta_i \in \mathbb{R}$: parameters

  $\hookrightarrow \vartheta = [\vartheta_1, \ldots, \vartheta_M]^T$
- Define $\hat{\vartheta}_{\text{LS}} = \underset{\vartheta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - f_\vartheta(x_i))^2$

Explicit formula for $\hat{\vartheta}_{LS}$

Set

$$g_j = \begin{bmatrix} g_1(x_j) \\ \vdots \\ g_M(x_j) \end{bmatrix} \in \mathbb{R}^M \quad j = 1, \dots, n$$

→ datapoints

$$G = \begin{bmatrix} g_1^T \\ \vdots \\ g_n^T \end{bmatrix} \in \mathbb{R}^{n \times M} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Assume $\left(G^T G\right)^{-1}$ exists, then

$$\hat{\vartheta}_{LS} = \left(G^T G\right)^{-1} G^T y = \left(\sum_{j=1}^{n} g_j g_j^T\right)^{-1} \left(\sum_{i=1}^{n} g_j y_j\right) \qquad (*)$$

$$G^T G = \begin{bmatrix} g_1 & g_2 & \cdots & g_n \end{bmatrix} \begin{bmatrix} g_1^T \\ \vdots \\ g_n^T \end{bmatrix} =$$

Goal

each node j of a sensor network measures a single data point (x_j, y_j) and knows the functions $g_1(\cdot), \dots, g_M(\cdot)$. Make each node compute $\hat{\vartheta}_{LS}$ with a distributed algorithm

Solution

write $(*)$ as

$$\hat{\vartheta}_{LS} = \left(\frac{1}{n} \sum_{j=1}^{n} g_j g_j^T \right)^{-1} \left(\frac{1}{n} \sum_{j=1}^{n} g_j y_j \right) \qquad (**)$$

- $(**)$ involves two averages! Run two average-consensus algorithms with initial conditions

$$x_i^{gg}(0) = g_i g_i^T \in \mathbb{R}^{M \times M}$$
$$x_i^{gy}(0) = g_i y_i \in \mathbb{R}^{M}$$

- Retreive $\hat{\vartheta}_{LS}$ as $\left(x_i^{gg}(\infty)\right)^{-1} x_i^{gy}(\infty)$ in each node

Time-varying consensus algorithms

Theory and experimental evidence suggest that consensus algorithms are robust to

a - changes in the graph topology
b - lossy communication
c - transmission delays
d - quantization

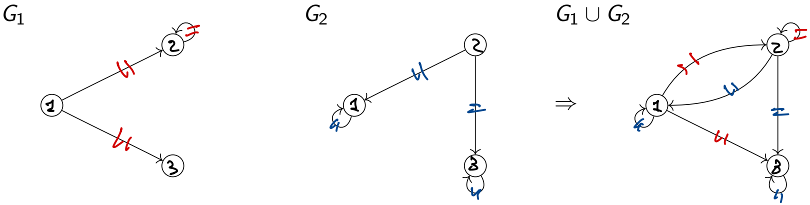
Let's discuss (a) and (b) $\rightarrow$ both induce a time-varying communication graph

Time-varying consensus algorithms

Definition

The union of digraphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ is the digraph $G = G_1 \cup G_2 = (V, E)$ where $E = E_1 \cup E_2$

Remark: even if G_1 and G_2 are weighted, the definition only specifies the edges of of $G_1 \cup G_2$, not the weights.

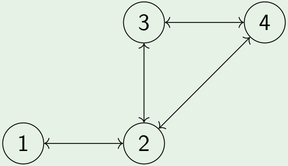


Definition

$\mathcal{G}_{SL}$ is the set of digraphs including all self-loops

- $G \in \mathcal{G}_{SL}$ is a mild assumption in consensus (each node usually knows its own internal state)

Example (Round Robin communication protocol)



Reference network

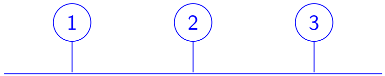
- in each communication slot, just one node v transmits to $\mathcal{N}^{out}(v)$
  $\hookrightarrow$ each $j \in \mathcal{N}^{out}(v)$ computes

$$x_j^+ = \frac{1}{2}(x_v + x_j) \qquad (\square)$$

  $\hookrightarrow$ all other nodes compute

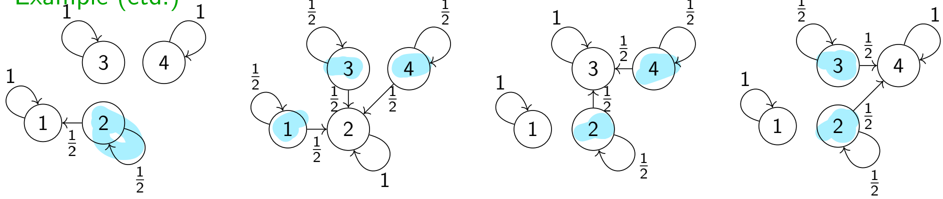
$$x_j^+ = x_j$$

- If the communication network is a bus, the communication protocol guarantees absence of collisions



- slots are associated to nodes in a round-robin fashion: node v transmits only at times $v, n+v, 2n+v, \ldots$ $n = \#$ of nodes

Example (ctd.)



v $v+n$ $v+2n$

Times $1, 5, 9, \ldots$ Times $2, 6, 10, \ldots$ Times $3, 7, 11, \ldots$ Times $4, 8, 12, \ldots$

Let A_i be the adjacency matrix at time i. We have

$$x(n) = A_{n-1} \cdot A_{n-2} \cdot \cdots \cdot A_0 x(0)$$

$x(i+1) = A_i \cdot x(i)$

$x(1) = A_0 x(0)$

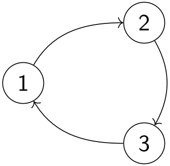
$x(2) = A_1 A_0 x(0)$

$\vdots$

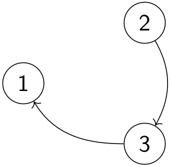
Questions: consensus? Average consensus?

Example of changes in the graph topology

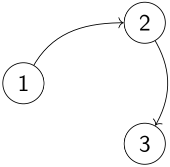
Sporadic packet drops



$k = 0$ (no drops)



$k = 1$ drop of the packet $2 \to 1$



$k = 2$ drop of the packet $1 \to 3$

Questions: consensus? Average consensus?

Theorem (time-varying consensus)

Let $A(k)$, $k \in \mathbb{N}$ be row-stochastic matrices with associated digraphs $G(k)$. Assume that

$(i,j) \in E(k)$

a) $G(k) \in \mathcal{G}_{SL}$
b) There is $\epsilon > 0$ such that, if in $G(k)$ the weight $a_{ij}(k)$ is strictly positive, then $a_{ij}(k) \geq \epsilon$ ($a_{ii}(k)$ is included) ϵ is the same at all times!
c) There is a duration $\delta \in \mathbb{N}$ such that, $\forall k \in \mathbb{N}$, the digraphs

$$\mathcal{G}(k) = G(k) \cup \cdots \cup G(k + \delta - 1)$$

contains a globally reachable node.

Then

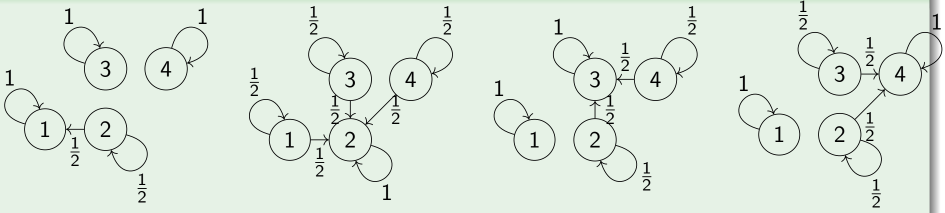
$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

i) $\exists w \succeq 0, w^T \mathbb{1}_n = 1$ such that $\lim_{k \to +\infty}(A(k)A(k-1)\dots A(0)) = \mathbb{1}_n w^T$
ii) the solution to $x(k+1) = A(k)x(k)$ converges exponentially fast to $\left(w^T x(0)\right) \mathbb{1}_n$
iii) if, in addition, each $A(k)$ is doubly stochastic, then $w = \frac{1}{n}\mathbb{1}_n$ and average consensus is achieved

Remarks

- Assumption (b) prevents weights from converging to zero
- Assumption (c) guarantees that δ does <u>not</u> depend on $k \rightarrow$ uniform connectivity requirement
- In *(iii)*, if $A(k), \dots, A(k+\delta-1)$ are doubly stochastic, then their product is double stochastic (prove it at home!)
- In *(iii)* we are not assuming primitivity of some matrix A
- The convergence can be slow for large δ

Example - Round Robin



Times $1, 5, 9, \ldots$ Times $2, 6, 10, \ldots$ Times $3, 7, 11, \ldots$ Times $4, 8, 12, \ldots$

$\rightarrow \varepsilon = \frac{1}{3}$

- Assumptions (a) and (b) are verified
- Assumption (c) is verified by construction $\rightarrow \delta = 4$

However, matrices $A(k)$ are not double stochastic $\rightarrow$ no guarantee of average consensus

Example - packet drops

Point $(c) \Rightarrow$ no node is isolated forever.