Distributed Optimization for Machine Learning

Lecture 11 - Average Consensus and Consensus Speed

Tianyi Chen

School of Electrical and Computer Engineering
Cornell Tech, Cornell University

October 1, 2025

Table of Contents

In-class average consensus game

- Each student = one **node** in a distributed network.
- Each node i holds a number $x_i(0)$ (your initial value).
- Goal: **All nodes converge to the same value - the average!**

Rules of this game:

1. You can talk to your assigned **neighbors**.
2. In each round, **simultaneously** update your number to the **average** of your number and your neighbors' numbers.

Goal of this game: Everyone updates together $\rightarrow$ Global agreement from local communication!

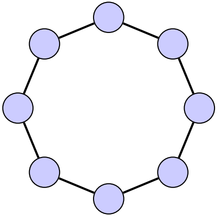
Step 1: Setup for synchronous consensus

Each student:

- Receives a random integer between 1-10.
- Writes it on a piece of paper (your $x_i(0)$).

Network topologies:

- **Line or Circle:** Talk to your left and right neighbor.
- (Optional) Instructor can assign a random neighbor graph.



Ring Graph

Goal: After several **synchronous rounds** of the consensus game, everyone's number should approach the same value.

Step 2: Run the synchronous consensus

Each round of the game

1. Share your current number with your neighbors.
2. Compute the average of your number and your neighbors' numbers.
3. Replace your number with this average.

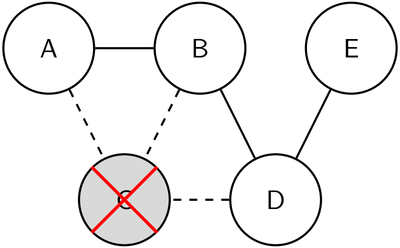
Repeat 4-5 rounds together.

- Observe numbers becoming closer.
- Notice that no one ever sees all other numbers!

Synchronous averaging $\rightarrow$ Fast and smooth convergence!

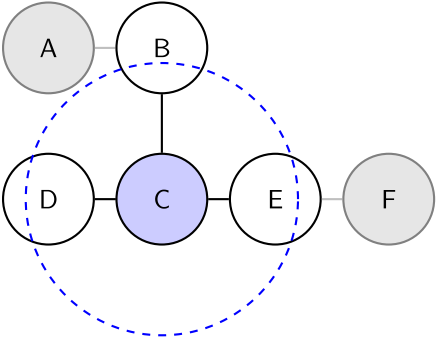
Average consensus - a decentralized algorithm

- **Robustness:** Inherently tolerant to node/link failures and churn. No single point of failure; relies only on local interactions.
- **Scalability:** Works well in massive, highly dynamic networks where the topology may be unknown or changing.



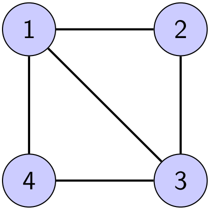
Consensus requirement

- **Consensus**: All nodes reach agreement on a certain quantity.
- Consensus is crucial for coordination, data fusion, and fairness.
- **Key challenge:** Each node has only an incomplete view of system.



Graph description of a network

- **Setup:** A network of N nodes connected by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.



- **Node set** $\mathcal{V}$:

$$\mathcal{V} = \{1, 2, 3, 4\}$$

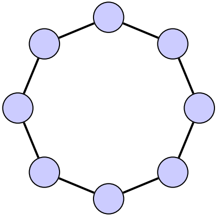
- **Edge set** $\mathcal{E}$:

$$\mathcal{E} = \{(1,2), (1,3), (1,4), (2,3), (3,4)\}$$

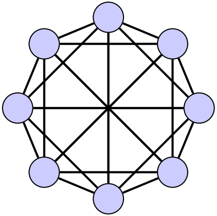
- **Adjacency Matrix A**:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

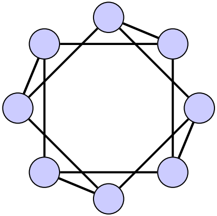
Various graph



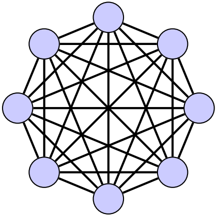
Ring Graph



Expander Graph



Torus Graph



Complete Graph

Average consensus

- **Setup:** Each node i holds a local state $x_i(k)$ at iteration k.
- **Input:** Each node i receives an input z_i and initializes $x_i(0) = z_i$.
- **Goal:** All nodes must agree on the average of the initial states:

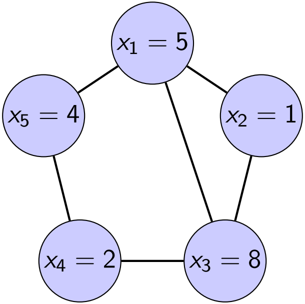
$$x^* = \frac{1}{N} \sum_{i=1}^{N} z_i$$

- **Constraint:** node i can only communicate with its neighbors $\mathcal{N}_i$.

Decentralized network setup

- We model the distributed system as a graph.

Example: 5-node Network



Consensus goal: All nodes must converge to $x^* = 4.0$

Average consensus protocol

Average consensus protocol

For $k = 1, 2, \cdots$, node i update x_i as a weighted average of its neighbors' states:

$$x_i(k+1) = \sum_{\{j:(i,j)\in\mathcal{E}\}} w_{ij} x_j(k)$$

For simplicity, we let $w_{ij} = 0$ if $(i, j) \notin \mathcal{E}$.

$$x_i(k+1) = \sum_{j=1}^{N} w_{ij} x_j(k)$$

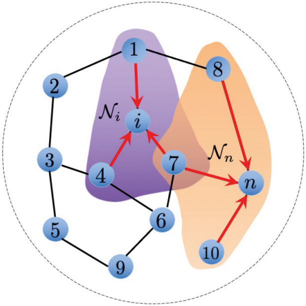
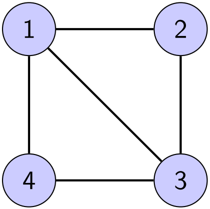


Figure: Synchronous consensus: all nodes update simultaneously using local information.

Introduce the weight matrix

- **Weight matrix W** is an $N \times N$ matrix whose (i, j) entry is w_{ij}.



$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & 0 \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & 0 & w_{43} & w_{44} \end{bmatrix}$$

- In vector form, the process is a simple linear iteration:

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k), \quad \text{where} \quad \mathbf{x}(0) = \mathbf{z} \in \mathbb{R}^N$$

where $\mathbf{x}(k) = [x_1(k), x_2(k), \cdots, x_N(k)]^\top$ and $\mathbf{z} = [z_1, z_2, \cdots, z_N]^\top$.

Properties of the weight matrix $\mathbf{W}$

- **Row stochastic (Row-Sum 1):**

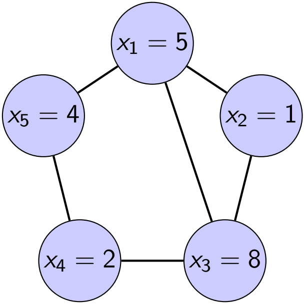
$$\sum_{j=1}^{N} W_{ij} = 1, \text{ for all } i.$$

- The weight matrix $\mathbf{W}$ is row stochastic since each node takes a weighted average of its neighbors' states.
- As a result, the weight matrix $\mathbf{W}$ has an eigenvalue $\lambda_1 = 1$, with corresponding eigenvector $\mathbf{1}$ (vector of ones).

$$\mathbf{W1} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & 0 \\ w_{31} & w_{32} & w_{33} & w_{34} \\ w_{41} & 0 & w_{43} & w_{44} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

How to choose the weights?

- Consider a simple choice: equal weights for neighbors

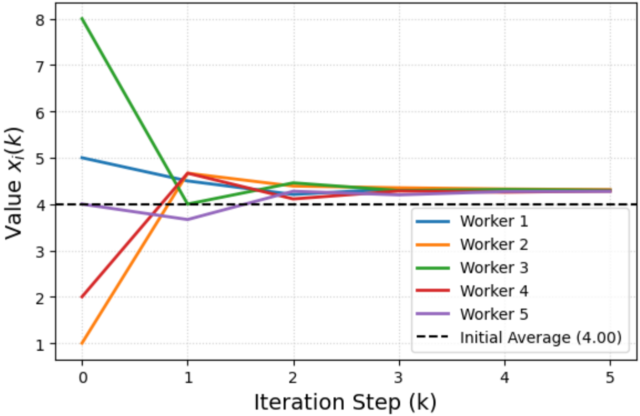


- **Weight matrix W**:

$$\mathbf{W} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 0 & 1/4 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$

- We expect that $x_i(k) \to \frac{1}{N}\sum_{i=1}^{N} z_i = 4$ as $k \to \infty$. **But...**

Failure to reach average consensus



- Reach consensus ≈ 4.3 but does not converge to the average! **Why?**

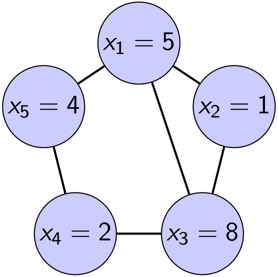
Failure to reach average consensus (cont.)

- Consider the average $\frac{1}{N}\sum_{i=1}^{N} x_i(k)$

$$\frac{1}{N}\sum_{i=1}^{N} x_i(k+1) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}x_j(k) = \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{i=1}^{N} w_{ij}\right)x_j(k)$$

- In the previous example, $\sum_{i=1}^{N} w_{ij} \neq 1$ for some j.

$$\mathbf{W} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 0 & 1/4 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$



Failure to reach average consensus (cont.)

- In the previous example, $\sum_{i=1}^{N} w_{ij} \neq 1$ for some j.

$$\mathbf{W} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 0 & 1/4 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 & 0 \\ 0 & 0 & 1/3 & 1/3 & 1/3 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$

- Consequently, the average **drifts**, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N} x_i(k+1) = \frac{1}{N}\sum_{j=1}^{N}\left(\sum_{i=1}^{N} w_{ij}\right)x_j(k) \neq \frac{1}{N}\sum_{i=1}^{N} x_i(k) \neq \cdots \neq \frac{1}{N}\sum_{i=1}^{N} z_i$$

Requirement of weight matrix

We require the following

$$\sum_{i=1}^{N} \mathbf{w}_{ij} = 1$$

to ensure the average is preserved.

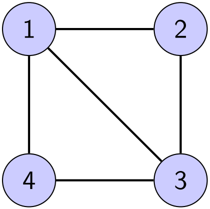
- **Column stochastic (Column-Sum 1)**: $\sum_{i=1}^{N} \mathbf{w}_{ij} = 1$.
- **Doubly stochastic: Row + Column stochastic.**

Assumption 1

Weight matrix $\mathbf{W}$ is doubly stochastic.

- **How to construct a doubly stochastic weight matrix?**

Pierre-Simon Laplace and Graph Laplacian

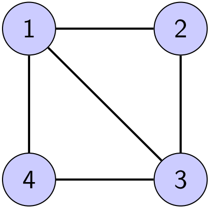


Pierre-Simon Laplace (1749 - 1827)

Laplacian matrix, also called the graph Laplacian, admittance matrix, or discrete Laplacian, is **a matrix representation of a graph**.

Graph Laplacian

Consider a 4-node undirected graph:



- **Adjacency matrix A**:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

- **Degree matrix** $\mathbf{D} = \mathrm{diag}(3, 2, 3, 2)$

Graph Laplacian L relates directly to the adjacency matrix **A** and degree matrix **D**: $\mathbf{L} = \mathbf{D} - \mathbf{A}$

$$\mathbf{L} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

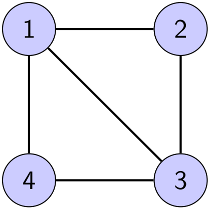
Construct a doubly stochastic weight matrix

- **Construction:** The weight matrix $\mathbf{W}$ is constructed as

$$\mathbf{W} = \mathbf{I} - c\mathbf{L},$$

  where $c > 0$ is a constant.

- **Example:** Choose $c = 1/(1 + \max\{\mathbf{D}\})$ with $\max\{\mathbf{D}\}$ denoting the maximum degree, e.g., $c = 1/4$ in the following example.



$$\mathbf{W} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/2 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 0 & 1/4 & 1/2 \end{bmatrix}$$

Simulation with doubly stochastic weights

- Reach average consensus if weight matrix is doubly stochastic

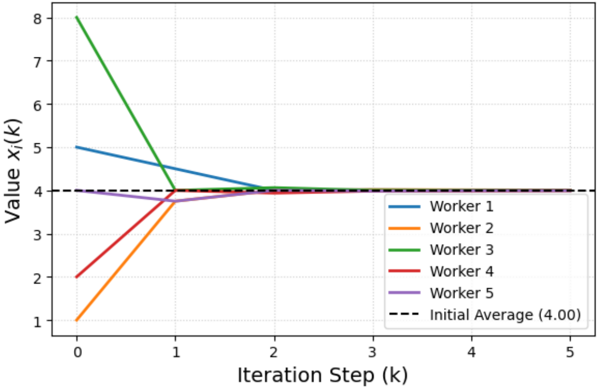


Table of Contents

How fast is consensus process?

- Now we investigate how fast the consensus algorithm converges.

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) \quad \text{where} \quad \mathbf{x}(0) = \mathbf{z} \in \mathbb{R}^N$$

- **Goal:** consensus on $x^* = \frac{1}{N}\sum_{i=1}^{N} x_i(0) = \frac{1}{N}\mathbf{1}^T\mathbf{z}$
- Consensus error of node i: $e_i(k) = x_i(k) - u^* = x_i(k) - \frac{1}{N}\mathbf{1}^T\mathbf{z}$
- We will stack $e_i(k)$ and use the following property:

$$\mathbf{1}(\frac{1}{N}\mathbf{1}^T\mathbf{z}) = \frac{1}{N}\mathbf{1}\mathbf{1}^T\mathbf{z} = \begin{bmatrix} \frac{1}{N}\mathbf{1}^T\mathbf{z} \\ \frac{1}{N}\mathbf{1}^T\mathbf{z} \\ \vdots \\ \frac{1}{N}\mathbf{1}^T\mathbf{z} \end{bmatrix} \quad \text{(Copying } x^* \text{ to all coordinates)}$$

Consensus error metric

- Recall the average consensus algorithm

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) = \mathbf{W}^k\mathbf{x}(0) \quad \text{where} \quad \mathbf{x}(0) = \mathbf{z} \in \mathbb{R}^N$$

- Stack the errors of all nodes into a vector:

$$\mathbf{e}(k) = \mathbf{x}(k) - \frac{1}{N}\mathbf{1}\mathbf{1}^T\mathbf{z} = \begin{bmatrix} x_1(k) - \frac{1}{N}\mathbf{1}^T\mathbf{z} \\ x_2(k) - \frac{1}{N}\mathbf{1}^T\mathbf{z} \\ \vdots \\ x_N(k) - \frac{1}{N}\mathbf{1}^T\mathbf{z} \end{bmatrix} \tag{1}$$

- **Convergence metric:** norm of $\mathbf{e}(k)$

$$\|\mathbf{e}(k)\| = \left\| \mathbf{x}(k) - \frac{\mathbf{1}\mathbf{1}^T\mathbf{z}}{N} \right\|$$

Convergence rate of consensus

Theorem 1 (Convergence rate of average consensus)

Under Assumption 1, it holds for the average consensus protocol that

$$\left\| \mathbf{x}(k) - \frac{\mathbf{1}\mathbf{1}^T\mathbf{z}}{N} \right\| \leq \rho^k \|\mathbf{z}\|,$$

where $0 \leq \rho < 1$ is a constant that depends on the connectivity of the underlying graph $\mathcal{G}$.

Q: How ρ depends on the connectivity of the underlying graph $\mathcal{G}$?

Using doubly stochastic property*

Starting from

$$\mathbf{x}(k) - \frac{\mathbf{1}\mathbf{1}^T\mathbf{z}}{N} = \left(\mathbf{W}^k - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)\mathbf{z},$$

we have

$$\mathbf{W}^k - \frac{\mathbf{1}\mathbf{1}^T}{N} = \left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)^k. \qquad (*)$$

That's crucial - it allows us to express the error term as repeated multiplication by the same "disagreement operator".

Why it holds? Because $\mathbf{W}$ and $\frac{\mathbf{1}\mathbf{1}^T}{N}$ commute:

$$\mathbf{W}\frac{\mathbf{1}\mathbf{1}^T}{N} = \frac{\mathbf{1}\mathbf{1}^T}{N}\mathbf{W} = \frac{\mathbf{1}\mathbf{1}^T}{N}.$$

We can then prove using mathematical induction (see next slide).

Using doubly stochastic property*

Mathematical induction: Assume the equality (*) holds for some $k \geq 1$:

$$
\begin{aligned}
\mathbf{W}^{k+1} - \frac{\mathbf{1}\mathbf{1}^T}{N} &= \mathbf{W}\mathbf{W}^k - \frac{\mathbf{1}\mathbf{1}^T}{N} \\
&= \mathbf{W}(\mathbf{W}^k - \frac{\mathbf{1}\mathbf{1}^T}{N}) + (\mathbf{W}\frac{\mathbf{1}\mathbf{1}^T}{N} - \frac{\mathbf{1}\mathbf{1}^T}{N}) \\
&= \mathbf{W}(\mathbf{W}^k - \frac{\mathbf{1}\mathbf{1}^T}{N}) \quad (\text{since } \mathbf{W}\frac{\mathbf{1}\mathbf{1}^T}{N} = \frac{\mathbf{1}\mathbf{1}^T}{N}) \\
&= \mathbf{W}\left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)^k \quad \text{(by induction hypothesis)} \\
&= \left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)^{k+1} \quad (\text{since } \mathbf{W} \text{ and } \frac{\mathbf{1}\mathbf{1}^T}{N} \text{ commute}).
\end{aligned}
$$

Proof of Theorem 1

- Partial average converges to the global average as follows

$$
\begin{aligned}
\left\| \mathbf{x}(k) - \frac{\mathbf{1}\mathbf{1}^T\mathbf{z}}{N} \right\| &= \left\| \mathbf{W}^k\mathbf{z} - \frac{\mathbf{1}\mathbf{1}^T}{N}\mathbf{z} \right\| \\
&= \left\| \left(\mathbf{W}^k - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) \mathbf{z} \right\| \\
&\leq \left\| \mathbf{W}^k - \frac{\mathbf{1}\mathbf{1}^T}{N} \right\| \|\mathbf{z}\| \\
&= \left\| \left(\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right)^k \right\| \|\mathbf{z}\| \quad \text{(Doubly stochastic)} \\
&\leq \left\| \mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right\|^k \|\mathbf{z}\| \qquad \text{(Sub-multiplicative)}
\end{aligned}
$$

- The convergence rate is determined by $\left\| \mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right\|$.

Proof of Theorem 1 (cont.)

Lemma 1

If $\mathbf{W}$ satisfies Assumption 1, there exists a constant $\rho \in [0, 1)$ such that

$$\left\| \mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right\| \leq \rho.$$

- The convergence rate of average consensus depends on

$$\left\| \mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right\|.$$

- since $\frac{\mathbf{1}\mathbf{1}^T}{N}$ projects any vector onto the “all-equal” subspace, $\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}$ thus captures *disagreement* after one iteration.

Key property of doubly stochastic matrices

A matrix $\mathbf{W}$ is **doubly stochastic** if

$$\mathbf{W}\mathbf{1} = \mathbf{1} \quad \text{and} \quad \mathbf{1}^T\mathbf{W} = \mathbf{1}^T.$$

Define the projection onto the consensus subspace:

$$\mathbf{P} := \frac{\mathbf{1}\mathbf{1}^T}{N}.$$

Then, because $\mathbf{W}$ is doubly stochastic,

$$\mathbf{W}\mathbf{P} = \mathbf{P} \quad \text{and} \quad \mathbf{P}\mathbf{W} = \mathbf{P}.$$

Interpretation: $\mathbf{P}$ projects any vector onto the “all-equal” (consensus) subspace, and $\mathbf{W}$ preserves this subspace (it does not change averages).

Connection to eigenvalues of $\mathbf{W}$

- When $\mathbf{W}$ is symmetric and doubly stochastic:

$$1 = \lambda_1(\mathbf{W}) > \lambda_2(\mathbf{W}) \geq \cdots \geq \lambda_N(\mathbf{W}) > -1.$$

- $\mathbf{1}$ is the eigenvector for $\lambda_1 = 1$ (the consensus direction).
  All other eigenvectors are orthogonal to $\mathbf{1}$ (disagreement directions).
- Since $\frac{\mathbf{1}\mathbf{1}^T}{N}$ has eigenvalues 1 (for $\mathbf{1}$) and 0 otherwise,

$$\mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N}$$

  has eigenvalues: 0 for $\mathbf{1}$, and $\lambda_i(\mathbf{W})$ for $i \geq 2$.

Spectral gap and consensus speed

- Therefore, the matrix norm

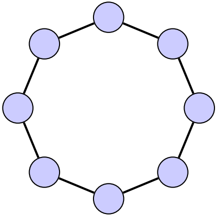
$$\left\| \mathbf{W} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right\| = \max_{i \geq 2} |\lambda_i(\mathbf{W})| \leq \rho.$$

- Assuming positive semidefinite of $\mathbf{W}$, define the **spectral gap:**

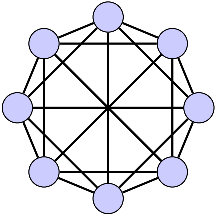
$$\text{gap}(\mathbf{W}) = 1 - \lambda_2(\mathbf{W}).$$

- Large gap $\Rightarrow$ well-connected graph $\Rightarrow$ fast consensus.
- Small gap $\Rightarrow$ weakly connected graph $\Rightarrow$ slow consensus.
- Examples:
  - Fully connected graph: $\lambda_2 = 0 \Rightarrow$ consensus in one step.
  - Ring graph: $\lambda_2 \approx 1 - O(1/N^2) \Rightarrow$ slow consensus.

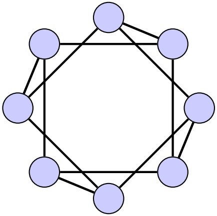
Various graph topologies



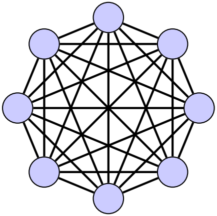
Ring Graph - $\lambda_2 \approx 1 - \mathcal{O}\left(\frac{1}{N^2}\right)$



Expander Graph - $\lambda_2 \approx \mathcal{O}(1)$



Torus Graph - $\lambda_2 \approx 1 - \mathcal{O}\left(\frac{1}{N}\right)$



Complete Graph - $\lambda_2 = 0$

Consensus time

Consensus time $T_{synch}(\epsilon)$: the number of iterations k needed to ensure ϵ consensus error:

$$\left\|\mathbf{x}(k) - \frac{\mathbf{1}\mathbf{1}^T\mathbf{z}}{N}\right\| \leq \epsilon$$

- From previous analysis:

$$\left\|\mathbf{x}(k) - \frac{\mathbf{1}\mathbf{1}^T\mathbf{z}}{N}\right\| \leq \rho^k\|\mathbf{z}\|$$

- For a large network $\rho \approx 1$, $\log(\rho) = \log(1-(1-\rho)) \approx -(1-\rho)$

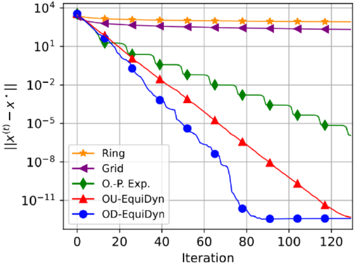
$$T_{synch}(\epsilon) \leq \frac{\log(\epsilon/\|\mathbf{z}\|)}{\log(\rho)} \approx \frac{\log(\|\mathbf{z}\|/\epsilon)}{1-\rho}$$

- **What is the order of $1/(1-\rho)$ for typical graphs?**

Consensus under various graph topologies

- The spectral gap of the graph strongly limits the consensus rate.

Network topology	Consensus rate ρ
Ring	$O(1 - \frac{1}{n^2})$
Grid	$O(1 - \frac{1}{n \ln(n)})$
Torus	$O(1 - \frac{1}{n})$
ExpoGraph	$O(1 - \frac{1}{\ln(n)})$
GeoMedian	$O(1 - \frac{\ln(n)}{n})$
Erdos-Renyi	$O(1)$
EquiGraph	$O(1)$



Simulation results are from [Song et. al., NeurIPS 2022]

- **Takeaway:** Well-connected graphs (like **Expander Graphs**) have large spectral gaps, enabling fast convergence.

Table of Contents

Why dynamic average consensus?

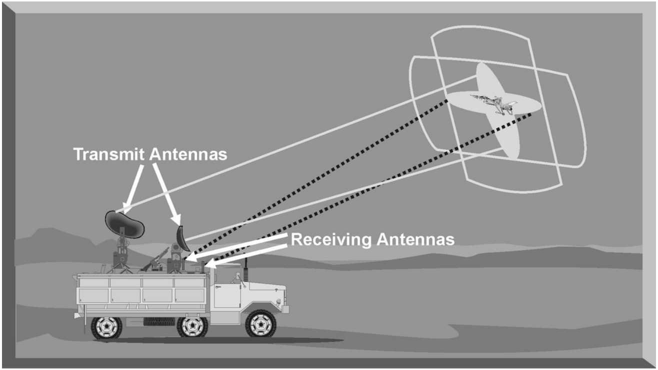


Figure: Sensing and moving target tracking.

Dynamic consensus protocol

- Recall the average consensus algorithm

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) \quad \text{where} \quad \mathbf{x}(0) = \mathbf{z} \in \mathbb{R}^N$$

- Average consensus tracks the global average of the static input $\mathbf{z}$
- If the input is changing with iteration, can we still be able to track the global average of the dynamic $\mathbf{z}(k)$?

Dynamic average consensus protocol

For $k = 1, 2, \cdots$, node i update x_i as a weighted average of its neighbors:

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) + \mathbf{z}(k+1) - \mathbf{z}(k) \quad \text{where} \quad \mathbf{x}(0) = \mathbf{z}(0)$$

Tracking property

- The dynamic average consensus recursion:

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) + \mathbf{z}(k+1) - \mathbf{z}(k) \quad \text{where} \quad \mathbf{x}(0) = \mathbf{z}(0)$$

- Left-multiplying $(1/N)\mathbf{1}^T$ to both sides of the above recursion

$$\begin{aligned}
&\left(\frac{1}{N}\sum_{j=1}^{N} x_j(k+1)\right)\mathbf{1} \\
&= \left(\frac{1}{N}\sum_{j=1}^{N} x_j(k)\right)\mathbf{1} + \left(\frac{1}{N}\sum_{j=1}^{N} z_j(k+1)\right)\mathbf{1} - \left(\frac{1}{N}\sum_{j=1}^{N} z_j(k)\right)\mathbf{1} \\
&= \left(\frac{1}{N}\sum_{j=1}^{N} z_j(k+1)\right)\mathbf{1} \quad \text{(tracks the global average of dynamic input)}
\end{aligned}$$

Analyze tracking errors

- Now we define

$$\bar{\mathbf{x}}(k) = \left(\frac{1}{n}\sum_{j=1}^{n} x_j(k)\right)\mathbf{1} \in \mathbb{R}^n \quad \text{and} \quad \bar{\mathbf{z}}(k) = \left(\frac{1}{n}\sum_{j=1}^{n} z_j(k)\right)\mathbf{1} \in \mathbb{R}^n$$

- With the recursion of $\bar{\mathbf{x}}(k)$ in the last page, we have

$$\begin{aligned}
&\|\mathbf{x}(k+1) - \bar{\mathbf{x}}(k+1)\|^2 \\
=&\|\mathbf{W}(\mathbf{x}(k) - \bar{\mathbf{x}}(k)) + \boldsymbol{\Delta}(k+1) - \overline{\boldsymbol{\Delta}}(k+1)\|^2 \\
=&\|(\mathbf{W} - \mathbf{1}\mathbf{1}^T/n)(\mathbf{x}(k) - \bar{\mathbf{x}}(k)) + \boldsymbol{\Delta}(k+1) - \overline{\boldsymbol{\Delta}}(k+1)\|^2 \\
\leq&\rho\|\mathbf{x}(k) - \bar{\mathbf{x}}(k)\|^2 + \frac{1}{1-\rho}\|\boldsymbol{\Delta}(k+1)\|^2
\end{aligned}$$

where $\boldsymbol{\Delta}(k) = \mathbf{z}(k) - \mathbf{z}(k-1)$ and $\overline{\boldsymbol{\Delta}}(k) = \bar{\mathbf{z}}(k) - \bar{\mathbf{z}}(k-1)$.

Asymptotic behavior

- If the dynamic input oscillates in a small range, i.e., $\|\mathbf{\Delta}(k)\| = \epsilon$,

$$\lim_{k\to\infty} \|\mathbf{x}(k) - \bar{\mathbf{x}}(k)\| = \frac{\epsilon}{1-\rho}$$

- Dynamic consensus converges to a small neighborhood around $\bar{\mathbf{x}}(k)$
- If the dynamic input converges to stationary points, i.e., $\|\mathbf{\Delta}(k)\| \to 0$, it holds that

$$\lim_{k\to\infty} \|\mathbf{x}(k) - \bar{\mathbf{x}}(k)\| = 0$$

- The convergence rate is determined by both ρ and the rate at which $\mathbf{\Delta}(k)$ approaches 0

Recap and fine-tuning

- What we have talked about today?

  ⇒ **Consensus** protocols enable robust, decentralized computation, ranging from static averaging to **dynamic signal tracking**.

  ⇒ **To ensure consensus**, we require $\mathbf{W}$ to be doubly stochastic.

  ⇒ **The consensus speed** largely depends on **graph topology** - Well-connected graphs (e.g., complete graphs) enable rapid consensus.

Welcome anonymous survey!