CS-E4740 - Federated Learning

FL Networks

Assoc. Prof. Alexander Jung

Spring 2025

Playlist

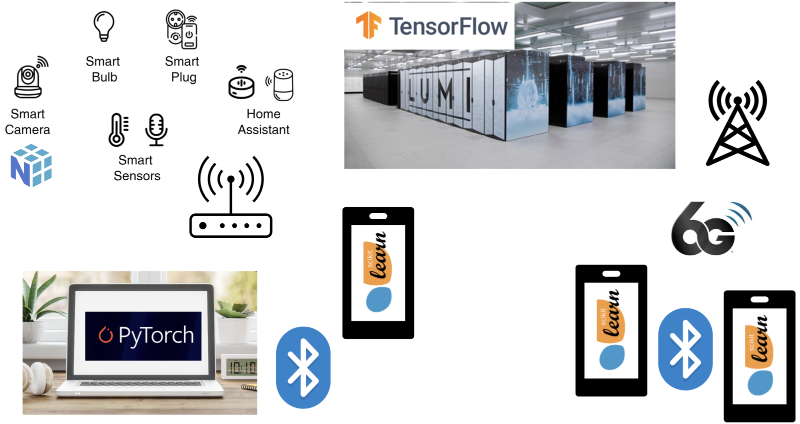
Glossary

Course Site

Table of Contents

Table of Contents

A ("Real-World") FL System

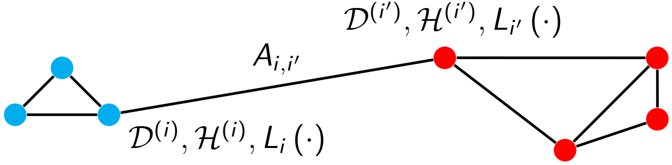


Abstracting Away Details

To analyze an FL system, we (need to) ignore many details:

- physical properties of communication links
- low-level communication protocols
- hardware configuration of devices
- operating systems of devices
- scientific computing software (Python packages)

An FL Network



- FL network consists of devices, denoted $i = 1, \ldots, n$.
- Some i, i' connected by edge with the weight $A_{i,i'} > 0$.
- Device i **generates data** $\mathcal{D}^{(i)}$ and **trains model** $\mathcal{H}^{(i)}$.
- Data $\mathcal{D}^{(i)}$ used to construct loss func. $L_i(\cdot)$.

FL Network is an Approximation

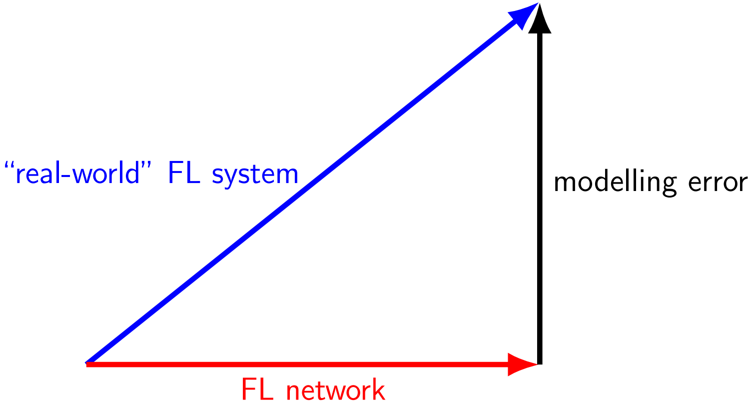


Table of Contents

A Precise Definition

An FL network consists of:

- a finite set of **nodes**, denoted as $\mathcal{V} := \{1, \ldots, n\}$
- a **local model** $\mathcal{H}^{(i)}$ at each node $i \in \mathcal{V}$
- a **local loss function** $L_i(\cdot)$ at each node $i \in \mathcal{V}$
- a set of undirected **edges**, denoted as $\mathcal{E}$
- a positive **edge weight** $A_{i,i'} > 0$ for each edge $\{i, i'\} \in \mathcal{E}$

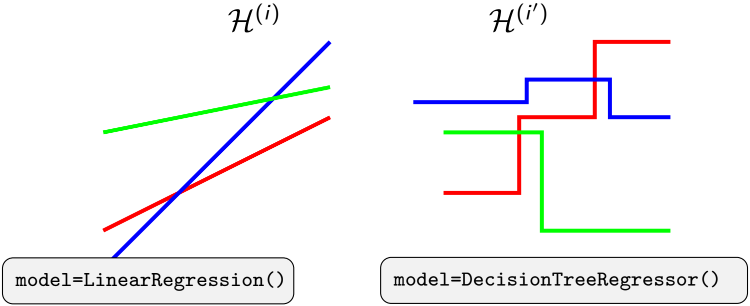
We represent the nodes $\mathcal{V}$, edges $\mathcal{E}$, and edge weights $A_{i,i'}$ of the FL network as an **undirected weighted graph** $\mathcal{G}$.

Nodes of an FL Network

- Consider an FL system with a finite number of devices n.
- We index devices as $i = 1, \ldots, n$.
- These indices form the set of nodes $\mathcal{V}$ in an FL network.
- Each node $i \in \mathcal{V}$ **represents** a physical device.
- We use "device i" and "node i" interchangeably.

Local Models of an FL Network

- Consider an FL system with devices $i = 1, \ldots, n$.
- Each device trains local (personal) model $\mathcal{H}^{(i)}$.
- The devices might use (very) different local models.
- We use local model parameters $\mathbf{w}^{(i)}$ for parametric $\mathcal{H}^{(i)}$.



Local Loss Functions of an FL Network

- Consider device i, training its local model $\mathcal{H}^{(i)}$.
- *To train a model* is to learn a useful hypothesis $h^{(i)} \in \mathcal{H}^{(i)}$.
- We measure usefulness of $h^{(i)}$ by a local loss function

$$L_i(\cdot) : \mathcal{H}^{(i)} \rightarrow \mathbb{R} : h^{(i)} \mapsto L_i\left(h^{(i)}\right)$$

- Different devices might use different loss functions.

Local Loss Functions of an FL Network - ctd.

- FL methods use different constructions of loss funcs.
- for param. models $\mathcal{H}^{(i)}$, with parameters $\mathbf{w}^{(i)} \in \mathbb{R}^d$, use

$$L_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R} : \mathbf{w}^{(i)} \mapsto L_i\left(\mathbf{w}^{(i)}\right)$$

- can use average loss on local dataset

$$L_i\left(\mathbf{w}^{(i)}\right) := \frac{1}{m_i} \sum_{r=1}^{m_i} \left(y^{(i,r)} - \left(\mathbf{w}^{(i)}\right)^T \mathbf{x}^{(i,r)}\right)^2$$

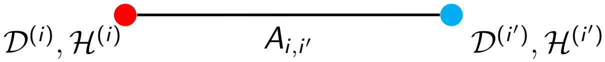
- use reward signals to estimate loss (federated reinf. learning)

Edges in an FL Network

- FL network consists of **undirected weighted** edges $\mathcal{E}$.
- $\{i, i'\} \in \mathcal{E}$ signifies a **similarity** between devices i and i'.
- We **quantify similarity using edge weight** $A_{i,i'} > 0$.
- FL applications employ various notions of similarity.
- We will primarily treat edges as a **design choice**.

Effect of Placing an Edge

We will design FL algorithms that are based on an FL network.



Placing an edge $\{i, i'\} \in \mathcal{E}$ between devices i, i' has two consequences on FL algorithms:

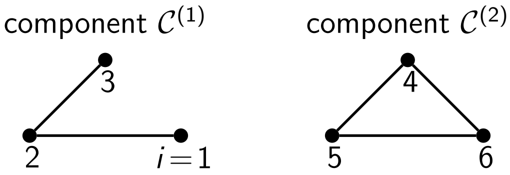
- We must communicate results of computations between devices i, i' ($A_{i,i'} \approx$ channel capacity).
- The local models at i, i' are forced to be similar.

Connectivity of an FL Network

Consider an FL network with graph $\mathcal{G}$. We define:

- $\mathcal{G}$ is **connected** if there is a path between any $i, i' \in \mathcal{V}$.
- A **component** $\mathcal{C} \subseteq \mathcal{V}$ is a connected subgraph with no edges between $\mathcal{C}$ and $\mathcal{V} \setminus \mathcal{C}$.
- The **neighborhood** of $i \in \mathcal{V}$ is $\mathcal{N}^{(i)} := \{i' \in \mathcal{V} : \{i, i'\} \in \mathcal{E}\}$.
- The **weighted node degree** of i is $d^{(i)} := \sum_{i' \in \mathcal{N}^{(i)}} A_{i,i'}$.
- The **maximum node degree** is $d_{\rm max} := \max_{i \in \mathcal{V}} d^{(i)}$.

Connectivity of an FL Network - Example



- FL network with graph $\mathcal{G}$ containing $n=6$ nodes.
- Uniform edge-weights, $A_{i,i'} = 1$ for all $\{i, i'\} \in \mathcal{E}$.
- Two components $\mathcal{C}^{(1)} = \{1, 2, 3\}, \mathcal{C}^{(2)} = \{4, 5, 6\}$.
- $d^{(1)} = 1$, $\mathcal{N}^{(2)} = \{1, 3\}$, $d_{\max} = 2$.

Design Choices

- Each FL network involves key design choices for
  - **Nodes.** Which devices should be included?
  - **Local models and loss functions.** What type of models should devices use, and how should we evaluate them?
  - **Edges.** Which devices should be connected, and how should similarity be defined?
- These choices determine the **computational and statistical properties** of FL algorithms.
- Trade-offs between **comp. complexity, accuracy, robustness, explainability, and privacy-prot.**

Design Space and Objectives

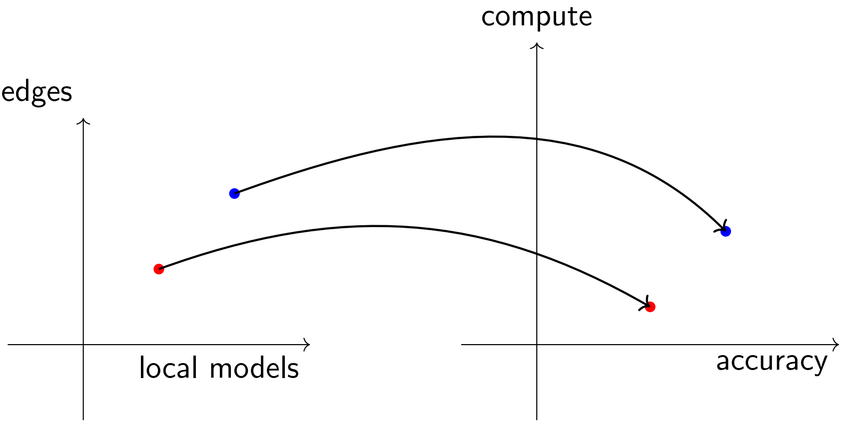


Table of Contents

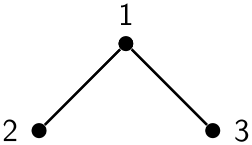
Laplacian Matrix

- Consider FL network with a weighted, undirected graph $\mathcal{G}$.
- The **Laplacian matrix** $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{n \times n}$ is defined element-wise as:

$$L_{i,i'}^{(\mathcal{G})} := \begin{cases} -A_{i,i'} & \text{for } i \neq i', \{i, i'\} \in \mathcal{E} \\ \sum_{i'' \neq i} A_{i,i''} & \text{for } i = i' \\ 0 & \text{else.} \end{cases}$$

Laplacian Matrix - Example

Here is a graph $\mathcal{G}$ with uniform edge weights $A_{i,i'} = 1$.



$$\mathbf{L}^{(\mathcal{G})} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Properties of the Laplacian Matrix

The Laplacian matrix $\mathbf{L}^{(\mathcal{G})}$ of an FL network is

- symmetric $\mathbf{L}^{(\mathcal{G})} = \left(\mathbf{L}^{(\mathcal{G})}\right)^T$ (since edges are undirected)
- and positive semi-definite (psd),

$$\mathbf{w}^T \mathbf{L}^{(\mathcal{G})} \mathbf{w} \geq 0 \text{ for every } \mathbf{w} \in \mathbb{R}^n. \tag{1}$$

The psd property (1) follows from the identity

$$\mathbf{w}^T \mathbf{L}^{(\mathcal{G})} \mathbf{w} = \underbrace{\sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} \left(w^{(i)} - w^{(i')}\right)^2}_{\text{total variation}}$$

which holds for every $\mathbf{w} = \left(w^{(1)}, \ldots, w^{(n)}\right)^T \in \mathbb{R}^n$.

The Spectrum of the Laplacian Matrix

- We can decompose any Laplacian matrix $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{n \times n}$ as

$$\mathbf{L}^{(\mathcal{G})} = \sum_{j=1}^{n} \lambda_j \mathbf{u}^{(j)} \left(\mathbf{u}^{(j)}\right)^T,$$

- with orthonormal eigenvecs. $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(n)} \in \mathbb{R}^n$, i.e.,

$$\left(\mathbf{u}^{(j)}\right)^T \mathbf{u}^{(j')} = \begin{cases} 1 & \text{for } j = j' \\ 0 & \text{otherwise,} \end{cases}$$

- and non-neg. eigvals

$$0 = \lambda_1 \leq \ldots \leq \lambda_n \leq 2d_{\max}.$$

The **spectrum** of $\mathbf{L}^{(\mathcal{G})}$ is the set of distinct eigenvalues.

Spectral Characterization of FL Networks

FL network $\mathcal{G}$ with k connected components $\mathcal{C}^{(1)}, \ldots, \mathcal{C}^{(k)}$.

Then, the Laplacian matrix $\mathbf{L}^{(\mathcal{G})} = \sum_{j=1}^{n} \lambda_j \mathbf{u}^{(j)} \big(\mathbf{u}^{(j)}\big)^T$

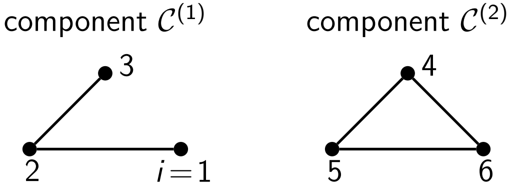
- has eigvals. $\lambda_c = 0$ for $c = 1, \ldots, k$, with
- corresponding eigvecs. $\mathbf{u}^{(c)}$, given entry-wise as

$$u_i^{(c)} = \begin{cases} \frac{1}{\sqrt{\left|\mathcal{C}^{(c)}\right|}} & \text{for } i \in \mathcal{C}^{(c)} \\ 0 & \text{otherwise.} \end{cases}$$

$\boxed{\mathcal{G} \text{ is connected } (k = 1) \text{ if and only if } \lambda_2 > 0.}$

Spectral Clustering - Toy Example

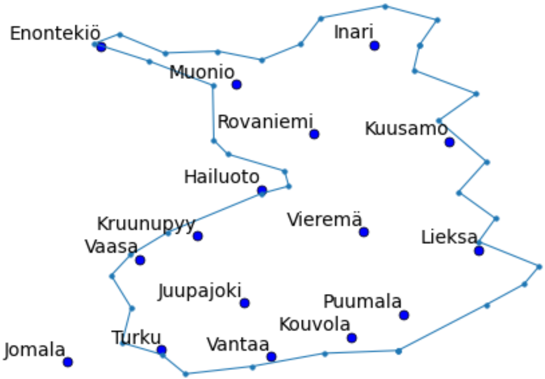
Consider a FL network $\mathcal{G}$ with two components:



- The Laplacian matrix has two zero eigvals. $\lambda_1 = \lambda_2 = 0$.
- What are corresp. eigvecs. $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}$? Are they unique?

Table of Contents

Weather Stations across Finland



Each weather station i collects data (observations) $\mathcal{D}^{(i)}$ that can be used to train a local model $\mathcal{H}^{(i)}$

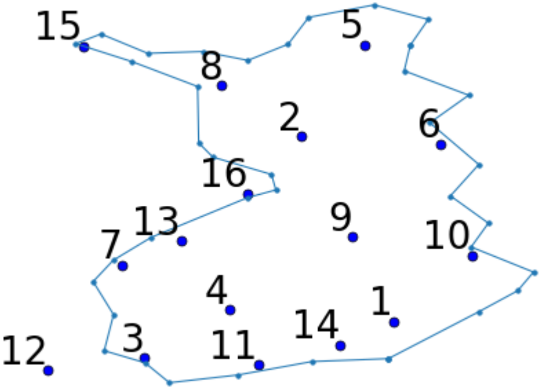
Python script for reproducing the Fig.:

Local Dataset of a FMI Station

Each FMI station i generates a local dataset $\mathcal{D}^{(i)}$ of the form

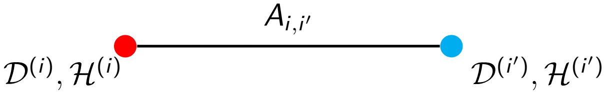
Time	**Air Temperature**
2025-01-13 16:08:00	-1.5
2025-01-13 16:09:00	-1.5
2025-01-13 16:10:00	-1.4
2025-01-13 16:11:00	-1.5
2025-01-13 16:12:00	-1.5

FL Network for FMI



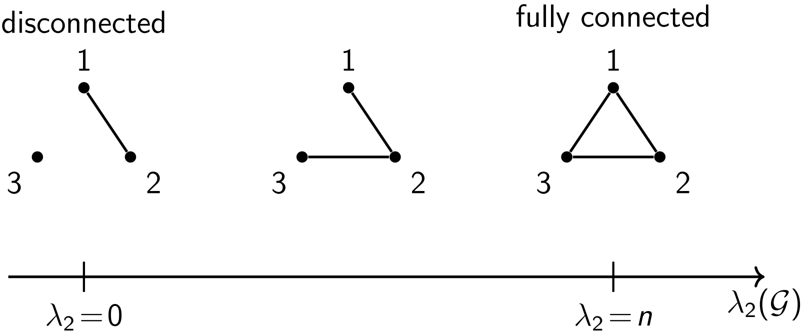
Which nodes (FMI stations) should be connected by edges ?

The Effect of Adding an Edge



- **Communication requirement.** Adding an edge means model parameters (updates) must be exchanged between i and i', requiring a communication link.
- **Coupling effect.** The local model parameters $\mathbf{w}^{(i)}$ and $\mathbf{w}^{(i')}$ become coupled, with interaction strength determined by $A_{i,i'}$.

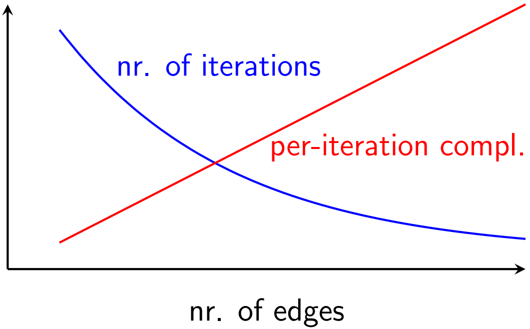
Connectivity measured by λ_2



- FL algorithms are faster for $\mathcal{G}$ with large $\lambda_2(\mathcal{G})$.
- Place (given number of) edges to maximize $\lambda_2(\mathcal{G})$.

Computational Aspects

- FL algorithms operate by iterative message passing.
- Each edge adds compute/comm. per-iteration.
- More edges speed up alg. $\Rightarrow$ needs fewer iterations.



Statistical Aspects

Consider an FL network with nodes $i = 1, \ldots, n$ that generate local data $\mathcal{D}^{(i)}$ and train local model $\mathcal{H}^{(i)}$.

Having an edge $\{i, i'\} \in \mathcal{E}$

- enforces similarity between local models at i, i', which
- can be detrimental if i, i' have different data distributions.

Place edges only between *statistically similar* nodes i, i'!

How to measure the stat. similarity between nodes i, i'?

Measuring Statistical Similarity

- Consider the local (weather) dataset $\mathcal{D}^{(i)}$

Time	**Air Temperature**
2025-01-13 16:08:00	-1.5
2025-01-13 16:09:00	-1.5
2025-01-13 16:12:00	-1.5

- Let's interpret the data as (the realization of) a random process with parametrized prob. distr. $p(\mathcal{D}^{(i)}; \boldsymbol{\theta})$.

- We estimate $\boldsymbol{\theta}$ by a function $\widehat{\boldsymbol{\theta}}^{(i)}$ of $\mathcal{D}^{(i)}$.

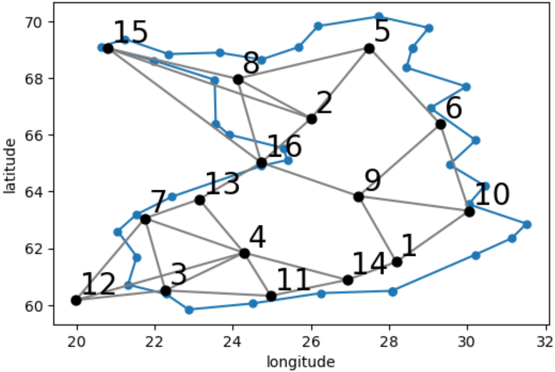
- Measure similarity between i, i' by $\left\| \widehat{\boldsymbol{\theta}}^{(i)} - \widehat{\boldsymbol{\theta}}^{(i')} \right\|$.

Measuring Statistical Similarity (ctd.)

- Est. $\widehat{\boldsymbol{\theta}}^{(i)}$ is one example of vector repr. $\mathbf{z}^{(i)} \in \mathbb{R}^k$ of $\mathcal{D}^{(i)}$.
- Place edges between nearest neighb. using $\left\| \mathbf{z}^{(i)} - \mathbf{z}^{(i')} \right\|$.
- We can also use other constructions for $\mathbf{z}^{(i)}$, e.g.,
  - for FMI stations, can use $\mathbf{z}^{(i)} := \big(\text{latitude}, \text{longitude}\big)^T$,
  - use gradient $\mathbf{z}^{(i)} := \nabla L_i(\mathbf{w})$ of local loss func.,
  - construct $\mathbf{z}^{(i)}$ by auto-encoder (learnt embedding).

Example: FMI Weather Stations

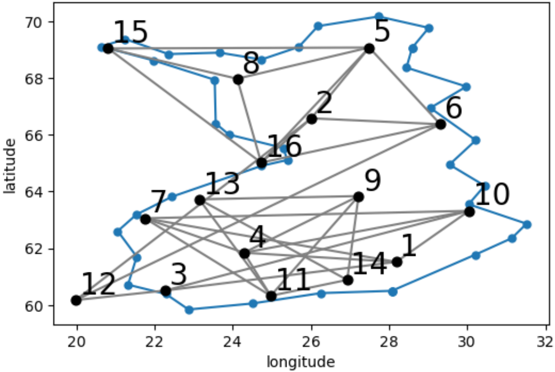
Connect FMI station i to nearest neighb. using vector $\mathbf{z}^{(i)} := \big(\text{latitude}, \text{longitude}\big)^T$.



Python script for reproducing the Fig.:

Example: FMI Weather Stations (ctd.)

Connect each FMI station to nearest neighbours using $\mathbf{z}^{(i)} :=$ avg. temp at station i during 2024-05-15.



Python script for reproducing the Fig.:

What's Next?

The next module formulates FL as an optimization problem defined over an FL network.

Later modules use FL networks for the design and analysis of FL systems.