IEEE MAG 2: Undirected Graphs

Thursday, April 4, 2024 6:09 PM

- Data points pertinent to the same task are collected at "m" distinct centers that need not be collated.
- We call these centers "agents", labeled by 1, 2,..., m.

Agent i has the set S_i of data points. $i = 1, \cdots, m$

P is the total number of data points (samples)

$$P = |S_1| + |S_2| + \ldots + |S_m|$$

$|S_i|$ = "cardinality" or number of elements within set S.

- Original Problem: $\min_{x \in \mathbb{R}^n} \frac{1}{P} \sum_{i=1}^{P} f_i(x)$ P data points

  $f_i(x) = c\rho(x) + \ell(x; z_i, y_i)$

- Reformulated: $\min_{x \in \mathbb{R}^n} \frac{1}{m} \sum_{i=1}^{m} f_i(x)$ m agents (4)

  $f_i(x) = \frac{m}{P} \sum_{s \in S_i} \left(c\rho(x) + \ell(x; z_s, y_s) \right)$

$f: \mathbb{R}^n \to \mathbb{R}$ private loss function known only to agent i.

Assume that the m agents are communicating over an undirected network represented by a graph G.

graph $G = ([m], \mathcal{E})$

$[m] = \{1, 2, \cdots, m\}$: set of agents (nodes)

$\mathcal{E}$: set of undirected edges

$\{i, j\}$: an edge connecting agents i and j

The agents want to solve the problem collaboratively.
They can share some estimates with their immediate neighbors but are not allowed to share their data.
This means they cannot reveal their loss functions fi.

- Given graph $G = ([m], \mathcal{E})$, reformulate the problem as:

$$\min_{x \in \mathbb{R}^n, i \in [m]} \left(\frac{1}{m} \sum_{i=1}^{m} f_i(x_i) \right) \quad (6)$$

subject to: $x_i = x_j \quad \forall \{i, j\} \in \mathcal{E}$

Each agent is assigned a copy x_i of the decision variable x.

$x_i = x$ for all agents $i \in [m]$

↳ $x_i = x_j$ for all $i, j \in [m]$

↳ $x_i = x_i$ for all $\{i, j\} \in \mathcal{E}$

↳ $x_i = x_j$ for all $i,j \in [m]$

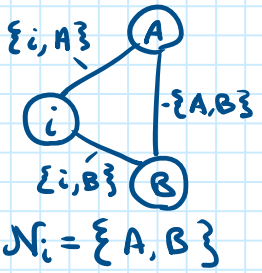
↳ $x_i = x_j$ for all $\{i,j\} \in \mathcal{E}$

- When graph G is connected, then problems (4-6) are equivalent.
- The objective function of (6) is decoupled, as each fi depends on its own variable xi.
- However, these variables are coupled through edge-based constraints.
- Strategy:
  - Distribute the problem among the agents.
  - Each agent knows its neighbors in the graph.
    - Each agent i is aware of agents j such that:

$\{i,j\} \in \mathcal{E}$, which constitutes the set:

$\mathcal{N}_i$: the neighbors of i in the graph.

$\mathcal{N}_i = \{j \mid \{i,j\} \in \mathcal{E}\}$

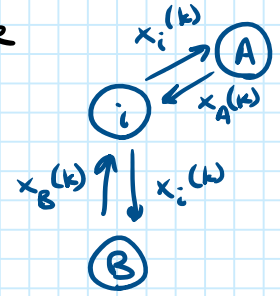


$\mathcal{N}_i = \{A, B\}$

Using the local agent knowledge of the graph and functions, every agent i can solve its own local part of the overall problem.
However, for the agents to collectively solve the overall problem, each agent needs to align its variables with the variables:

x_j corresponding to its neighbors $j \in \mathcal{N}_i$

- CONSENSUS ALGORITHM: a distributed method that the agents can use to asymptotically agree on a decision vector.
  - Each agent starts with an arbitrary vector $x_i^{(0)}$
  - At every iteration k, every agent sends its current iterate $x_i^{(k)}$ to its neighbors $j \in \mathcal{N}_i$ and receives $x_j^{(k)}$ from its neighbors $j \in \mathcal{N}_i$
  - Then, every agent i executes the consensus update step:



$$x_i^{(k+1)} = a_{ii}\, x_i^{(k)} + \sum_{j \in \mathcal{N}_i} a_{ij}\, x_j^{(k)}$$

$a_{ii} > 0$ and $a_{ij} > 0$ such that:

$a_{ii} + \sum_{j \in \mathcal{N}_i} a_{ij} = 1$

The positive scalars a_{ij}, $j \in \mathcal{N}_i \cup \{i\}$: "convex weights".

The vector $x_i^{(k+1)}$: "convex combination" or "weighted average" of the points x_j, $j \in \mathcal{N}_i \cup \{i\}$

(neighbors of i) ∪ (agent i itself)

- A MORE COMPACT REPRESENTATION:

$A \in \mathbb{R}^{m \times m}$

$a_{ii} > 0$ and $a_{ij} > 0$ with $a_{ii} + \sum_{j \in \mathcal{N}_i} a_{ij} = 1$

$A \in \mathbb{R}^{m \times m}$

$a_{ii} > 0$ and $a_{ij} > 0$ with $a_{ii} + \sum_{j \in N_i} a_{ij} = 1$

and:

$a_{ij} = 0$ when $j \notin N_i \cup \{i\}$

Matrix A is "compatible" with G when A has a positive entry in the i-jth position only when {i, j} is a link in graph G.

Consensus algorithm for every $i \in [m]$

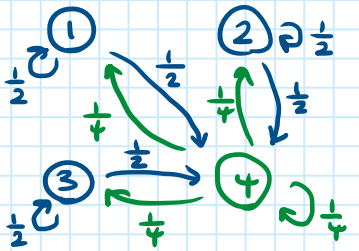
$$x_i^{(k+1)} = \sum_{j=1}^{m} a_{ij} x_j^{(k)}$$

The sum of elements in each raw of A is equal to 1.

↳ such a non-negative matrix is "raw-stochastic".

- Example:

$$A = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \in \mathbb{R}^{4 \times 4}$$



Matrix A is constructed using the **equal-weight rule**: Each agent i gives the same weight to itself and all of its neighbors.

- Consensus Point

When matrix A is raw-stochastic and compatible with G, the iterate sequences $\{x_i^{(k)}, i \in [m]\}$ generated by the consensus algorithm converge to the same limit point $\tilde{x}$.

$\tilde{x}$: "consensus" or "agreement point".

$\tilde{x}$ is given as a convex combination of the initial values $\{x_i^{(0)}, i \in [m]\}$:

$$\tilde{x} = \sum_{i=1}^{m} \pi_i x_i^{(0)}$$

$$\pi = [\pi_1, \pi_2, \cdots, \pi_m]$$

↳ left eigenvector of matrix A corresponding to $\lambda = 1$ $(\pi A = \pi)$

The convergence result is obtained by viewing **matrix A as a one-step transition matrix** of a Markov chain and employing the **ergodicity theory** for **Markov chains**.

If matrix A is doubly stochastic, then $\pi_i = \frac{1}{m}$ for all i.

↳ Consensus point is the average of the initial values:

$$\tilde{x} = \frac{1}{m} \sum_{i=1}^{m} x_i^{(0)}$$

→ Consensus point is the average of the initial values:

$$\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i^{(0)}$$

- **Distributed Optimization: 2 steps:**

At the beginning of every iteration k, every agent sends its own current iterate xi^(k) to its neighbors j in Ni and receives xj^(k) from its neighbors j in Ni. Then, every agent i executes the following 2 steps:

1) Mixing (consensus): $v_i^{(k)} = \sum_{j=1}^{m} a_{ij} x_j^{(k)}$

2) Gradient based: $x_i^{(k+1)} = v_i^{(k)} - \alpha_k \nabla f_i(v_i^{(k)})$ } (10)

$v_i^{(k)}$: a convex combination of the points x_j, $j \in N_i \cup \{i\}$

- This algorithm is distributed because every agent updates by using a gradient of its own private function and is local in the sense that it relies on local information exchange.
- This is also called **"consensus-based gradient method"** due to its mixing step that resembles the distributed consensus process.
- Can be viewed as an **extension** of the **gradient method** in which the mixing step is introduced to align the agents' iterates. This step acts like a **"virtual coordinator"** of the agents iterates in a system where there is no central coordinator or master node.

- **Example:**
  - Take $\{x_i^{(k)}, i \in [m]\}$ the iterate sequences produced by the method at all agents in the system.
  - Take the average of these iterates across all agents at any given instance: α_k: positive step size

$$\frac{1}{m}\sum_{i=1}^{m} x_i^{(k+1)} = \frac{1}{m}\sum_{i=1}^{m} v_i^{(k)} - \frac{\alpha_k}{m}\sum_{i=1}^{m} \nabla f_i(v_i^{(k)})$$

$$= \frac{1}{m}\sum_{j=1}^{m}\left(\sum_{i=1}^{m} a_{ij}\right) x_j^{(k)} - \frac{\alpha_k}{m}\sum_{i=1}^{m} \nabla f_i(v_i^{(k)})$$

  - When A is doubly stochastic, $\sum_{i=1}^{m} a_{ij} = 1$ for all j. Thus,

$$\frac{1}{m}\sum_{i=1}^{m} x_i^{(k+1)} = \frac{1}{m}\sum_{j=1}^{m} x_j^{(k)} - \frac{\alpha_k}{m}\sum_{i=1}^{m} \nabla f_i(v_i^{(k)})$$

  - Let $x^{(-k)}$ denote iterate average across agents at time k:

$$x^{(-k)} = \frac{1}{m}\sum_{i=1}^{m} x_i^{(k)} \qquad x_i^{(k)}: x \text{ at agent } i \text{ at iteration } k.$$

$$x^{(-k+1)} = x^{(-k)} - \frac{\alpha_k}{m}\sum_{i=1}^{m} \nabla f_i(v_i^{(k)})$$

This very closely resembles the centralized gradient descent update, but now the gradient of fi is computed at point vi^(k) instead of x^(-k).

- Adding and subtracting the correct gradients:

$$x^{(-k+1)} = x^{(-k)} - \frac{\alpha_k}{m}\sum_{i=1}^{m}\nabla f_i(x^{(-k)}) + \epsilon^{(k)}$$

$$\epsilon^{(k)} = \frac{\alpha_k}{m}\sum_{i=1}^{m}\left(\nabla f_i(x^{(-k)}) - \nabla f_i(v_i^{(k)})\right)$$

$x^{(-k)}$: average across agents; $v_i^{(k)}$: conv. comb. of x

ϵ: error is based on difference of gradients

- Mixing matrix A is crucial because it ensures the following:
  1) The averaged iterate sequence $\{x^{(-k)}\}$ converges to a solution x^*
  2) The disagreement sequence $\{\|x_i^{(k)} - x_i^{(-k)}\|\}$ converges to zero for every agent i.

A consensus-based process affected by two forces:
1) Consensus by the mixing step (influenced by matrix A)
2) Agent-based gradient descent of the objective functions fi

The mixing step is also referred to as "diffusion" since it allows for the local agent information to diffuse over the entire network after enough iterations.

The mixing and gradient update steps can be changed to produce an alternative variant of the distributed method:

Adapt-then-combine

$$v_i^{(k)} = x_i^{(k)} - \alpha_k \nabla f_i(x_i^{(k)})$$

$$x_i^{(k+1)} = \sum_{j=1}^{m} a_{ij} v_i^{(k)}$$

Combine-then-adapt

Original:

$$v_i^{(k)} = \sum_{j=1}^{m} a_{ij} x_j^{(k)}$$

$$x_i^{(k+1)} = v_i^{(k)} - \alpha_k \nabla f_i(v_i^{(k)})$$

- CONVERGENCE RATE:

Depends critically on architecture of G and spectral properties of A.

(10): $O\left(\frac{\log(k)}{\sqrt{k}}\right)$ k: iterations

Assuming: $\sum_{k=0}^{\infty}\alpha_k = \infty$, $\sum_{k=0}^{\infty}\alpha_k^2 < \infty$

(centralized GD): $O\left(\frac{1}{\sqrt{k}}\right)$

↳ Centralized is faster than distributed!