IEEE MAG 3: Directed Graphs

Sunday, April 7, 2024 5:18 PM

- When the underlying communication graph is directed, the construction of **doubly stochastic weight matrix A** is quite computationally **expensive**.
- We will consider a consensus algorithm that uses a **column-stochastic weight matrix A** and combine it with a gradient method.

- CONSENSUS ALGORITHM:

$$A \in \mathbb{R}^{m \times m} \qquad G = ([m], \mathcal{E})$$

Weight matrix A is column-stochastic and compatible with graph G.

$$a_{jj} > 0 \text{ and } a_{ij} > 0 \text{ with:}$$

$$\sum_{i \in \mathcal{N}_j^{out}} a_{ij} = 1 \quad \text{and,}$$

$$a_{ij} = 0 \quad \text{when } i \notin \mathcal{N}_j^{out} \cup \{j\}$$

$$\begin{array}{cc} & j=1 \\ i=1 \rightarrow & A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} \end{array}$$

a_{ij}

$\mathcal{N}_j^{out}$: neighbors of agent j directed outwards

$$\mathcal{N}_j^{out} = \{ i \mid (j,i) \in \mathcal{E} \}$$

Common Choice: $a_{ij},\ i \in \mathcal{N}_j^{out} \cup \{j\} \rightarrow$ all same values equal to cardinality of $\mathcal{N}_j^{out} \cup \{j\}$

Each agent i maintains variables xi^(k) and yi^(k) at time k. At time (k+1), each agent j sends out aij*xj^(k) and aij*yj^(k) to all of its out-neighbors.
Then, every agent i updates by simply summing the x and y variables its received from its neighbors.

$$\left.\begin{aligned} x_i^{(k+1)} &= \sum_{j \in \mathcal{N}_i^{in} \cup \{i\}} a_{ij} x_j^{(k)} \\ y_i^{(k+1)} &= \sum_{j \in \mathcal{N}_i^{in} \cup \{i\}} a_{ij} y_j^{(k)} \end{aligned}\right\} \text{PUSH-SUM METHOD}$$

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

$$y_i^{(k)} \in \mathbb{R}, \quad y_i^{(0)} = 1$$

Leads to consensus as long as G is strongly connected

$$\hookrightarrow z_i^{(k)} = \frac{x_i^{(k)}}{y_i^{(k)}} \quad \forall i \in [m]$$

$\hookrightarrow$ "Ratio Consensus"

- DISTRIBUTED GRADIENT METHOD FOR ML:

$x_i^{(0)}, i \in [m] \ \forall i$

$y_i^{(0)}, i \in [m] \ \forall i$

Each agent sends $a_{ji} x_j^{(k)}$ and $a_{ji} y_j^{(k)}$ to out-neighbors.
Then each agent updates their x and y:

$$v_i^{(k+1)} = \sum_{j \in N_i^{in} \cup \{i\}} a_{ij} x_j^{(k)}$$

$$y_i^{(k+1)} = \sum_{j \in N_i^{in} \cup \{i\}} a_{ij} y_j^{(k)}$$

$$z_i^{(k+1)} = \frac{v_i^{(k+1)}}{y_i^{(k+1)}}$$

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

"GRADIENT PUSH METHOD"

- Analysis: $$\frac{x_i^{(k+1)}}{y_i^{(k+1)}} = z_i^{(k+1)} - \underbrace{\gamma_i^{(k)}}_{} \nabla f(z_i^{(k+1)})$$

Converges to $\tilde{x}$
consensus point

$$\gamma_i^{(k)} = \frac{\alpha_k}{y_i^{(k+1)}} > 0$$

Convexity of fi and appropriate alpha are important to ensure that x~ is in fact a solution to the aggregate agent learning problem of minimizing the average sum:

$$\frac{1}{m} \sum_{i=1}^{m} f_i(x)$$

- If f_i is convex, convergence is $O\left(\frac{\log(k)}{\sqrt{k}}\right)$
- If f_i is strongly convex, then $O\left(\frac{\log(k)}{k}\right)$