Learning from distributed datasets

An introduction with two examples

Pedro Guerreiro and João Xavier
Institute for Systems and Robotics, Instituto Superior Técnico
University of Lisbon

Mathematics, Physics & Machine Learning seminar
June 18, 2020
Outline

- What is distributed learning?
- Example 1: distributed convex learning with a roaming token
- Example 2: distributed classification with random meetings
What is distributed learning?
The physical setup

- a team of robots, a wireless sensor network, a swarm of drones, ...

- agents are spatially distributed
- each agent measures data
Centralized learning

- each agent sends its data to a fusion center
- fusion center computes the solution
- drawbacks: single point of failure, traffic jams, lack of data privacy
Distributed learning

• agents are linked by channels
• a channel can appear and disappear (randomly)
• a channel might be directed
Distributed learning

- agents send messages through channels
- all agents obtain the solution
Example 1: distributed convex learning with a roaming token
to illustrate, let the learning problem be logistic regression

\[ D = \{(x_1, y_1), \ldots, (x_K, y_K)\} \] is dataset \((x_k=\text{features}, \ y_k = \pm 1)\)

goal is to learn the separating hyperplane
• goal is to learn
\[ \theta^* = \arg\min_{\theta} \log \left( 1 + e^{-y_1 \theta^T x_1} \right) + \cdots + \log \left( 1 + e^{-y_K \theta^T x_K} \right) + \theta^T P \theta \]
\[ f_D(\theta) \]

• ... but with dataset $D$ split across the agents:

• objective function has the form
\[ f_D = f_{D_1} + f_{D_2} + f_{D_3} + f_{D_4} + f_{D_5} + f_{D_6} + f_{D_7} \]
Learning with a random token

- goal is to learn $\theta^* = \arg\min_\theta f_{D_1}(\theta) + f_{D_2}(\theta) + \cdots + f_{D_N}(\theta)$

- the token carries a guess of $\theta^*$
- the token moves randomly across the network
- when the token visits an agent, it updates its guess of $\theta^*$
- token starts at agent 3 with some initialization $\theta(0)$
- token updates $\theta(1) = \theta(0) - \alpha \nabla f_{D_4}(\theta(0))$
• token updates $\theta(2) = \theta(1) - \alpha \nabla f_{D_3}(\theta(1))$
• token updates $\theta(3) = \theta(2) - \alpha \nabla f_{D_1}(\theta(2))$
Does this scheme work?

- ... no

- consider the network

- token chooses neighbor uniformly at random (current node included)
• scheme fails:

• it fails because it minimizes the wrong function:

\[ \pi_1 f_{D_1}(\theta) + \pi_2 f_{D_2}(\theta) + \cdots + \pi_N f_{D_N}(\theta) \]

• \((\pi_1, \pi_2, \ldots, \pi_N)\) is the stationary distribution of the random walk
scheme is minimizing

\[ \pi_1 f_{D_1} (\theta) + \pi_2 f_{D_2} (\theta) + \cdots + \pi_N f_{D_N} (\theta), \]

with \( \pi_1, \ldots, \pi_N \) depending on the network topology

what about unbiasing the scheme by redefining each local function as

\[ f_{D_n} \leftarrow \frac{1}{\pi_n} f_{D_n}? \]

this works, but each agent \( n \) would need to know \( \pi_n \)

what about agent \( n \) guessing \( \pi_n \) on the fly?
Distributed learning algorithm

1: $t = 0$
2: initialize $\theta(0)$
3: token starts at some node $n(0)$
4: **repeat**
5: $t \leftarrow t + 1$
6: token jumps to a random neighbor $n(t)$
7: token updates the parameter

$$
\theta(t) = \theta(t - 1) - \alpha \frac{1}{\pi(n(t), t)} \nabla f_{D_{n(t)}}(\theta(t - 1))
$$

where $\pi(n, t)$ is fraction of visits to node $n$, by time $t$
8: **until** some stopping criterion is met
• the algorithm succeeds:

• we can prove: with probability one, (and under suitable convexity),

\[
\lim_{t \to \infty} \inf f_D(\theta(t)) \leq f_D(\theta^*) + O(\alpha)
\]
Blueprint of the proof

• fix an agent \( n \)

• define the stopping times \( t_1, t_2, t_3, \ldots \), where \( t_k \) is (random) time of the \( k \)th visit to agent \( n \), along with filtration \( \mathcal{F}_{t_1}, \mathcal{F}_{t_2}, \mathcal{F}_{t_3}, \ldots \)

• use the ergodic theorem for markov chains to show

\[
E \left( \| \theta(t_{k+1}) - \theta^* \|^2 \mid \mathcal{F}_{t_k} \right) \\
\leq \| \theta(t_k) - \theta^* \|^2 - \alpha (f(\theta(t_k)) - f(\theta^*)) + O(\alpha^2)
\]

• use martingale arguments to get

\[
\lim \inf_{t \to \infty} f(\theta(t)) \leq f(\theta^*) + O(\alpha)
\]
Example 2: distributed classification with random meetings
• each agent $n$ observes a data stream $X_n(1), X_n(2), X_n(3), \ldots$

• streams are samples of unknown random source $\theta^* \in \{\theta_1, \ldots, \theta_P\}$

• channels change randomly over time

• agents want to learn which $\theta^*$ is generating the streams
• we are at time $t$

• each agent $n$ holds a belief $\mu_n(t) = (\mu_{n1}^\theta(t), \mu_{n2}^\theta(t), \mu_{n3}^\theta(t))$
• we move to time $t + 1$

• each agent $n$ meets with available neighbors and updates its belief
Distributed learning algorithm

1: each agent $n$ initialises $\mu_n(0) = (\mu_{n1}^1(0), \ldots, \mu_{nP}^P(0))$ and sets $t = 0$
2: repeat
3: local Bayes update: each agent $n$ observes $X_n(t)$ and updates

$$\mu_n^\theta(t+1/2) = \frac{P_n^\theta(X_n(t)) \mu_n^\theta(t)}{\sum_{\vartheta \in \Theta} P_n^\vartheta(X_n(t)) \mu_n^\vartheta(t)}, \quad \text{for } \theta \in \Theta$$

4: local fusion of beliefs: each agent $n$ receives $\mu_m^\theta(t+1/2)$ from its neighbors $m$ and updates

$$\mu_n^\theta(t+1) = \frac{\exp \left( \sum_{m=1}^N W_{nm}(t) \log \mu_m^\theta(t+1/2) \right)}{\sum_{\vartheta \in \Theta} \exp \left( \sum_{m=1}^N W_{nm}(t) \log \mu_m^\vartheta(t+1/2) \right)}, \quad \text{for } \theta \in \Theta$$

5: $t \leftarrow t + 1$
6: until some stopping criterion is met
• weight $W_{nm}(t) \neq 0$ only if agent $n$ is linked to agent $m$ at time $t$

• example:

\[
W(t) = \begin{bmatrix}
* & 0 & * & 0 & 0 & 0 & 0 \\
0 & * & * & 0 & 0 & 0 & 0 \\
* & * & * & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & * & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & * & * & 0 \\
0 & 0 & 0 & 0 & 0 & * & * \\
0 & 0 & 0 & 0 & 0 & 0 & * \\
\end{bmatrix}
\]
assumptions on the random weight matrices $W(t)$:

- $W_{nm}(t) \geq 0$
- each row of $W(t)$ sums to one
- the sequence $(W(t))_{t \geq 0}$ is i.i.d. and independent from $(X_n(t))_{t \geq 0}$
- $\mathbb{E}(W(t))$ is irreducible and aperiodic
• we can prove: with probability one,

$$\lim_{t \to +\infty} \frac{1}{t} \log \frac{\mu_n^{\theta^*}(t)}{\mu_n^\theta(t)} = \sum_{m=1}^{N} \mathbb{E}(\pi_m) D_{\text{KL}} \left( P_{m}^{\theta^*}, P_{m}^{\theta} \right), \quad \text{for } \theta \in \Theta$$

where \( \pi = (\pi_1, \ldots, \pi_N) > 0 \) comes from \( \prod_{t=1}^{\infty} W(t) = 1\pi^T \)

• interpretation: algorithm works

$$\mu_n^{\theta^*}(t) \approx e^{K(\theta^*, \theta)t} \mu_n^\theta(t) \text{ for large } t \quad \Rightarrow \quad \left\{ \begin{array}{l} \mu_n^{\theta^*}(t) \to 1 \\ \mu_n^\theta(t) \to 0, \text{ for } \theta \neq \theta^* \end{array} \right.$$
Blueprint of the proof

- fix $\theta \in \Theta$ and introduce the $N$ dimensional vectors

\[
u(t) := \begin{bmatrix}
\log \frac{\mu_{1}^{\theta}(t)}{\mu_{1}^{\theta}(t)} \\
\vdots \\
\log \frac{\mu_{N}^{\theta}(t)}{\mu_{N}^{\theta}(t)}
\end{bmatrix}
\text{ and } \quad \ell(t) := \begin{bmatrix}
\log \frac{P_{1}^{\theta}(X_{1}(t))}{P_{1}(X_{1}(t))} \\
\vdots \\
\log \frac{P_{N}^{\theta}(X_{N}(t))}{P_{N}(X_{N}(t))}
\end{bmatrix}
\]

- note that

\[
\frac{1}{t} u(t) = \frac{1}{t} \Phi(t, 1) u(0) + \frac{1}{t} \sum_{\tau = 0}^{t-1} \Phi(t, t - \tau) \ell(t - \tau),
\]

where

\[
\Phi(t, s) = W(t) \cdots W(s)
\]
• fix $\epsilon > 0$

• show that there exists a random time $T(t) \leq t$ such that

$$\frac{1}{t} \sum_{\tau=0}^{t-1} \Phi(t, t-\tau)l(t-\tau) = 1 \left( \frac{1}{t} \sum_{\tau=T(t)}^{t-1} v(t-\tau)^T l(t-\tau) \right) + O(\epsilon),$$

where $v(t-\tau)$ is random vector in $\prod_{s=t-\tau}^{\infty} W(s) = 1 v(t-\tau)^T$

• use the ergodic theorem to get

$$\frac{1}{t} \sum_{\tau=T(t)}^{t-1} v(t-\tau)^T l(t-\tau) \to K(\theta^*, \theta)$$
Numerical example

- $\Theta = \{\theta_1, \theta_2\}$ with $\theta_1$ being the active one
- Gaussian streams: $X_n(t) \sim \mathcal{N}\left(c_n^\theta, (\sigma_n^\theta)^2\right)$ for $\theta \in \Theta$
- Each channel is (independently) active with probability $p = 0.7$
Belief $\mu_1(t) = \left( \mu_1^{\theta_1}(t), \mu_1^{\theta_2}(t) \right)$ at agent 1
Belief $\mu_2(t) = \left( \mu_{5}^{\theta_1}(t), \mu_{5}^{\theta_2}(t) \right)$ at agent 5
Log beliefs ratio $\frac{1}{t} \log \frac{\mu_{1}^{\theta_1}(t)}{\mu_{1}^{\theta_2}(t)}$ at agent 1
Log beliefs ratio \( \frac{1}{t} \log \frac{\mu_{5}^{\theta_{1}}(t)}{\mu_{5}^{\theta_{2}}(t)} \) at agent 5


Thank you!