Intro to online learning: Learning from expert advice.

University of California, Santa Barbara

May 29, 2019

Overview

1. Review of optimizatio
2. Online learning
3. Online learning algorithms

Review of optimization

$$\min_{\theta \in \mathcal{C}} f(\theta) \tag{1}$$

Here $\mathcal{C}$ is a convex set and $f(\cdot)$ is a convex function
We care about the complexity

$$f(\theta^k) - f(\theta^*) \leq \epsilon \tag{2}$$

If $k = \log \frac{1}{\epsilon}$, it is linear convergence

Complexity Table

	convex	+ smooth	+ strong convex
gradient	$\frac{1}{\epsilon^2}$	$\frac{1}{\epsilon} \to \frac{1}{\sqrt{\epsilon}}$	$\frac{L}{m} \log \frac{1}{\epsilon} \to \sqrt{\frac{L}{m}} \log \frac{1}{\epsilon}$
SGD	$\frac{1}{\epsilon^2}$	$\frac{1}{\epsilon^2}$	$\frac{1}{m\epsilon}$
finite sum + SGD	$\frac{1}{\epsilon^2}$	$n + \frac{1}{\epsilon} \to \frac{1}{\sqrt{\epsilon}}$	$(n + \frac{L}{m}) \log \frac{1}{\epsilon} \to (n + \sqrt{\frac{L}{m}}) \log \frac{1}{\epsilon}$

Table: Complexity of first order methods

Not covered

- Second order method: LBFGS, quasi-newton
- Non-convex optimization: have to convexify or adding noise to escape from local solutions
- How about DNN: too many local/global solutions!

Online learning

- Problem statement for statistical learning: Given any dataset $(x_1, y_1), \ldots, (x_n, y_n)$ iid from $\mathcal{D}$. To find $\mathcal{H} : X \rightarrow Y$
- Reliable setting: $\exists h^* \in \mathcal{H}$ s.t. $\mathbf{P}(h^*(x) = y) = 1$
- Error $error(f) = \mathbb{E}\mathbf{1}(h(x) \neq y) \approx \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i)$
- Online learning:
    1. set x_1, choose $h_1 \in \mathcal{H}$, such as $\hat{y}_1 = h_1(x_1)$,
    2. set x_2, choose $h_2 \in \mathcal{H}$, such as $\hat{y}_2 = h_2(x_2)$, ...

  The cumulative loss $\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(\hat{y}_i \neq y_i)$
- Design algorithm such that $M(A) \leq \mathcal{O}(n)$
- Example: $X \in \{0, 1\}^d$, $Y \in \{0, 1\}$, $h = x(1)$ or $x(4)$ or $x(16)$

Algorithm 1: Online ERM/FTL

- $V_1 = H$
- for $t = 1, 2, \ldots, n$:
  1. receive x_t, pick any $h \in V_t$
  2. predict $\hat{y}_t = h(x_t)$
  3. Receive y_t
  4. loss $\mathbf{1}(\hat{y}_t \neq y_t)$ Update $V_{t+1} = \{h \in V_t, h(x_t) = y_t\}$
- Convergence speed: $1 \leq |V_t| \leq |H| - M_t$, so $M_t \leq |H| - 1$

Algorithm 2: Majority voting (Halving)

- $V_1 = H$
- for $t = 1, \ldots, n$
  1. Receive x_t
  2. Majority voting: for any $h \in V_t$, $\hat{y}_t = \arg\max \sum_h \mathbf{1}(h(x_t) = y_t)$
  3. Receive y_t
  4. Update $V_{t+1} = \{h \in V_t, h(x_t) = y_t\}$
- $1 \leq |V_t| \leq |H| \frac{1}{2}^{m_t}$, so $m_t \leq \log_2(|H|)$

Agnostic online learning

If there does not exist a h such that $h(x_i) = y_i, \forall i = 1, \dots, n$

$$regret(h) = \sum_{i=1}^{n} \mathbf{1}(y_t = h_t(x_t)) - \min_{h \in \mathcal{H}} \sum_{i=1}^{n} \mathbf{1}(y_t \neq h(x_t)) \tag{3}$$

Example: stock prediction, Google

	Dearaj	Omid	Yuqing	Paul the Octopus	Truth
Day 1	Down	Up	Up	Down	Down
Day 2	Up	Up	Down	Down	Down
Day 3	Up	Down	Up	Up	Up
Day 4					

Table: Choices of expert

	Dearaj	Omid	Yuqing	Paul the Octopus
Day 1	1	1	1	1
Day 2	1	$\frac{1}{2}$	$\frac{1}{2}$	1
Day 3	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{2}$	1
Day 4	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{1}{2}$	1

Table: Weights of expert

Algorithm 3: Weighted majority

- M: Number of mistakes
- m: Number of mistakes of the best expert
- n: Number of expert
- $W_t = \sum_{i=1}^{n} w_{it}$, $W_1 = n$, and $W_{t+1} \leq W_t(1 - \frac{1}{4})$

$$(\frac{1}{2})^m \leq W \leq n(\frac{3}{4})^m \quad (4)$$

- $M \leq -\frac{\log 1/2}{\log 3/4} m + \frac{\log n}{\log 3/4}$

ϵ-weighted majority

$$W_{i_{t+1}} = W_{it}(1-\epsilon) \text{ if } \hat{y}_{it} \neq y_t \tag{5}$$

Then

$$(1-\epsilon)^m \leq W \leq m(1-\frac{1}{2}(1-\epsilon))^m \tag{6}$$

$$m\log(1-\epsilon) \leq \log m + m\log(\frac{1}{2}+\frac{1}{2}\epsilon) \tag{7}$$

$$M \leq \frac{-\log(1-\epsilon)}{-log(\frac{1}{2}+\frac{1}{2}\epsilon)}m + \frac{\log n}{-\log(\frac{1}{2}+\frac{1}{2}\epsilon)} \leq 2(1+\epsilon)m + \mathcal{O}(\log m) \tag{8}$$

Algorithm 4: randomized weighted majority (RWM)

- Set $W_1^{(i)} = 1$ for all i
- for $t = 1, \ldots, T$,

$$Output = \begin{cases} Up & \text{with probability } \frac{\sum_i W^i \mathbf{1}(y_t^i = up)}{W} \\ Down & \text{Otherwise} \end{cases}$$

Analysis

- $F_t = \frac{\sum_{i=1}^{n} W_t^i \mathbf{1}(\hat{y}_t^i \neq y^t)}{W_t}$, $W_t = n(1-\epsilon F_1)\ldots(1-\epsilon F_T)$
- $m\log(1-\epsilon) \leq \log(W_{t+1}) \leq \log n + \sum_{i=1}^{n} \log(1-\epsilon F_t)$
  $\leq \log n - \epsilon \sum_{t=1}^{T} F_t = \log n - \mathbb{E}(M)$
- $\mathbb{E}(M) \leq \frac{\log n}{\epsilon} + \frac{-\log(1-\epsilon)}{\epsilon} m \approx (1+\frac{\epsilon}{2})m + \frac{\log n}{\epsilon} \leq m + \sqrt{\frac{m\log n}{2}}$
- The last equality holds when $\epsilon = \sqrt{\frac{2\log n}{m}}$

Relationship with convex optimization

- Learning with expert: $\min \sum_i f_i(\theta_i)$
- $f_i(\theta_i) = \langle \theta_i, \ell \rangle = \mathbb{E}_{\theta_i}[l_i]$
- $C_i \sum \theta_i = 1, \theta_i \geq 0$