Large Scale Optimization for Machine Learning: Lecture 23

Lecturer: Meisam Razaviyayn Scribe: Max Gaungyu Li

Nov 16, 2017

1 Online Convex Optimization

In this lecture we consider optimization as a *process*. In many practical applications the environment is so complex that it is infeasible to lay out a comprehensive theoretical model and use classical algorithmic theory and mathematical optimization. It is necessary as well as beneficial to take a robust approach, by applying an optimization method that learns as one goes along, learning from experience as more aspects of the problem are observed.

The online convex optimization (OCO) framework can be seen as a structured repeated game. The protocol of this learning framework is as follows: At iteration t, the online player chooses $\boldsymbol{w}_t \in \mathcal{S}$. After the player has committed to this choice, a convex cost function $f_t : \mathcal{S} \to \mathbb{R}$ is revealed. Here $f_t \in F$ is the bounded family of cost functions available to the adversary. The cost incurred by the online player is $f_t(\boldsymbol{w}_t)$, the value of the cost function for the choice $\boldsymbol{w}_t$. Let T denote the total number of game iterations.

Online Convex Optimization Framework

1: Input: A convex set $\mathcal{S}$
2: **for** $t = 1, 2, 3, \ldots$ **do**
3: predict a vector $\boldsymbol{w}_t \in \mathcal{S}$
4: receive a convex loss function $f_t : \mathcal{S} \to \mathbb{R}$
5: suffer loss $f_t(\boldsymbol{w}_t)$

What would make an algorithm a good OCO algorithm? As the framework is game-theoretic and adversarial in nature, the appropriate performance metric also comes from game theory: define the regret of the decision maker to be the difference between the total cost she has incurred and that of the best fixed decision in hindsight. In OCO we are usually interested in an upper bound on the worst case regret of an algorithm. Let $\mathcal{A}$ be an algorithm for OCO, which maps a certain game history to a decision in the decision set.We formally define the **regret** of $\mathcal{A}$ after T iterations w.r.t a fixed point $\boldsymbol{u} \in \mathcal{S}$ as:

$$\mathrm{Reg}_T(\boldsymbol{u}) = \sum_{t=1}^{T} f_t(\boldsymbol{w}_t) - \sum_{t=1}^{T} f_t(\boldsymbol{u})$$

Intuitively, an algorithm performs well if its regret w.r.t the best fixed strategy $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u} \in \mathcal{S}} \sum_{t=1}^{T} f_t(\boldsymbol{u})$ is sublinear as a function of T, i.e. $\mathrm{Reg}_T(\boldsymbol{u}^*) = o(T)$, since this implies that on the average the algorithm performs as well as the best fixed strategy in hindsight.

1.1 Follow-The-Leader

The most natural learning rule is to use (at any online round) any vector which has minimal loss on all past rounds. In the context of online convex optimization it is usually referred to as Follow-The-Leader.

Algorithm 1 Follow-The-Leader

1: **for** $t = 1, 2, 3, \dots$ **do**
2: $\quad \boldsymbol{w}_t = \arg\min_{\boldsymbol{w} \in \mathcal{S}} \sum_{i=1}^{t-1} f_i(\boldsymbol{w})$

To analyze FTL, we first show that the regret of FTL is upper bounded by the cumulative difference between the loss of $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t+1}$.

Lemma 1.1 *Let $\boldsymbol{w}_1, \boldsymbol{w}_2, \dots$ be the sequence of vectors produced by FTL. Then, for all $\boldsymbol{u} \in S$ we have*

$$Reg_T(\boldsymbol{u}) = \sum_{t=1}^{T} \Big(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{u}) \Big) \leq \sum_{t=1}^{T} \Big(f_t(\boldsymbol{w}_t) - f_t(\boldsymbol{w}_{t+1}) \Big)$$

To prove this lemma, we subtracting $\sum_t f_t(\boldsymbol{w}_t)$ from both sides fo the inequality and rearranging. The desired inequality can be rewritten as

$$\sum_{t=1}^{T} f_t(\boldsymbol{w}_{t+1}) \le \sum_{t=1}^{T} f_t(\boldsymbol{u})$$

We procve this inequality by induction. The base case of $T = 1$ follows directly from the definition of $\boldsymbol{w}_{t+1}$. Assume the inequality holds for $T-1$, then for all $\boldsymbol{u} \in \mathcal{S}$ we have

$$\sum_{t=1}^{T-1} f_t(\boldsymbol{w}_{t+1}) \le \sum_{t=1}^{T-1} f_t(\boldsymbol{u})$$

Adding $f_T(\boldsymbol{w}_{T+1})$ to both sides we get

$$\sum_{t=1}^{T} f_t(\boldsymbol{w}_{t+1}) \le f_T(\boldsymbol{w}_{T+1}) + \sum_{t=1}^{T-1} f_t(\boldsymbol{u})$$

The above holds or all $\boldsymbol{u}$ and in particular for $\boldsymbol{u} = \boldsymbol{w}_{T+1}$. Thus,

$$\sum_{t=1}^{T} f_t(\boldsymbol{w}_{t+1}) \le \sum_{t=1}^{T} f_T(\boldsymbol{w}_{T+1}) = \min_{\boldsymbol{u}\in\mathcal{S}} \sum_{t=1}^{T} f_t(\boldsymbol{u})$$

where the last equation follows from the definition of $\boldsymbol{w}_{T+1}$. This concludes our inductive argument.

In next lecture we will see predictions provided by FTL may not be stable $\boldsymbol{w}_t$ shifts drastically from round to round where we only added a single loss function to the objective of the optimization problem in the definition of $\boldsymbol{w}_t$. One way to stabilize FTL is by adding regularization.

References

[1] S. Shalev-Shwartz, Online Learning and Online Convex Optimization, Foundations and Trends in Machine Learning 2012.

[2] E. Hazan, Introduction to Online Convex Optimization 2016.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, The Nonstochastic Multiarmed Bandit Problem, SIAM 2003.

[4] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014