Notes on online convex optimization*

Karl Stratos

Online convex optimization (OCO) is a principled framework for online learning:

OnlineConvexOptimization
Input: convex set S, number of steps T

- For $t = 1, 2, \ldots, T$:
  - Select $w_t \in S$.
  - Receive a convex loss $f_t : S \to \mathbb{R}$ chosen adversarially.
  - Suffer loss $f_t(w_t)$.

Each hypothesis is a vector in some convex set S. The loss function $f_t : S \to \mathbb{R}$ is convex and defined for each time step t individually. Our goal is to have small "regret" with respect to a hypothesis space U, namely $\mathrm{Regret}_T(U) := \max_{u \in U} \mathrm{Regret}_T(u)$ where

$$\mathrm{Regret}_T(u) := \sum_{t=1}^{T} f_t(w_t) - f_t(u)$$

1 Unregularized aggregate loss minimization

At time t, we have observed losses $f_1 \ldots f_{t-1}$, so a natural choice of w_t is one that minimizes the sum of all past losses. This is known as **Follow-the-Leader (FTL)**:

$$w_t = \arg\min_{w \in S} \sum_{i=1}^{t-1} f_i(w) \tag{1}$$

Lemma 1.1. *If we use Eq. (1) in OCO, we have*

$$\mathit{Regret}_T(S) \leq \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1})$$

2 Regularized aggregate loss minimization

Lemma 1.1 suggests a need for containing $f_t(w_t) - f_t(w_{t+1})$. If we assume f_t is L_t-Lipschitz with respect to S and some norm $||\cdot||$, we have

$$f_t(w_t) - f_t(w_{t+1}) \leq L_t \, ||w_t - w_{t+1}||$$

*This is a bird's eye view of the incredible tutorial by Shai Shalev-Shwartz (2011). For full details, see the original tutorial.

which in turn suggests a need for containing $||w_t - w_{t+1}||$. If the objective in Eq. (1)

$$F_t(w) := \sum_{i=1}^{t-1} f_i(w)$$

happens to be σ*-strongly-convex*, $||w_t - w_{t+1}||$ cannot be arbitrarily large: by the definition of w_t and w_{t+1} and strong convexity,

$$F_t(w_{t+1}) - F_t(w_t) \geq \frac{\sigma}{2} ||w_t - w_{t+1}||^2$$
$$F_{t+1}(w_t) - F_{t+1}(w_{t+1}) \geq \frac{\sigma}{2} ||w_t - w_{t+1}||^2$$

Adding these two inequalities, we get:

$$||w_t - w_{t+1}|| \leq \frac{f_t(w_t) - f_t(w_{t+1})}{\sigma} \leq \frac{L_t}{\sigma}$$

We can always endow σ-strong-convexity on F_t by adding a σ-strongly-convex regularizer $R : S \to \mathbb{R}$. This is known as **Follow-the-Regularized-Leader (FoReL)**:

$$w_t = \arg\min_{w \in S} R(w) + \sum_{i=1}^{t-1} f_i(w) \tag{2}$$

By treating R as the (convex) "loss at time $t = 0$", we get the following corollary from Lemma 1.1.

Corollary 2.1. *If we use Eq. (2) in OCO, for all $u \in S$ we have*

$$\textit{Regret}_T(u) \leq R(u) - \min_{v \in S} R(v) + \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1})$$

Theorem 2.2. *Let $f_t : S \to \mathbb{R}$ be convex loss functions that are L_t-Lipschitz over convex S with respect to $||\cdot||$. Let $L \in \mathbb{R}$ be a constant such that $L^2 \geq (1/T) \sum_{t=1}^{T} L_t^2$, and let $R : S \to \mathbb{R}$ be a σ-strongly-convex regularizer. Then the regret of FoReL with respect to $u \in S$ is bounded above as:*

$$\textit{Regret}_T(u) \leq R(u) - \min_{v \in S} R(v) + \frac{TL^2}{\sigma}$$

3 Linearization of convex losses

Theorem 2.2 assumes an oracle that solves Eq. (2), so it's not very useful for deriving concrete algorithms. But a technique known as "linearization" of convex losses greatly simplifies this task. Since S is a convex set and f_t is convex, at each round of OCO we can select $z_t \in \partial f_t(w_t)$ so that

$$f_t(w_t) - f_t(w_{t+1}) \leq \langle z_t, w_t \rangle - \langle z_t, w_{t+1} \rangle \tag{3}$$

Thus given a general convex loss f_t, we can pretend that it's a *linear* loss $g_t(u) := \langle z_t, u \rangle$ where z_t is a sub-gradient of f_t at w_t. In light of Corollary 2.1 and Eq. (3), running FoReL on these linearized losses:

$$w_t = \arg\min_{w \in S} R(w) + \sum_{i=1}^{t-1} \langle w, z_i \rangle \tag{4}$$

enjoys the same regret bound in Theorem 2.2.

3.1 Online mirror descent

Eq. (4) can be additionally analyzed in a dual framework known as online mirror descent (OMD). OMD frames Eq. (4) as two separate steps: starting with $\theta_1 := 0$,

$$\begin{aligned} w_t &= g(\theta_t) \\ \theta_t &= \theta_{t-1} - z_{t-1} \end{aligned}$$

where $g(\theta) := \arg\max_{w \in S} \langle w, \theta \rangle - R(w)$ is known as the link function. The particular form of the link function comes from the convex conjugate of R (R is assumed to be closed and convex):

$$R^\star(\theta) := \max_{w \in S} \langle w, \theta \rangle - R(w)$$

A property of $R^\star$ is that if $z \in \partial R^\star(\theta)$, then $R^\star(\theta) = \langle z, \theta \rangle - R(z)$. Thus $g(\theta_t) = z_t \in \partial R^\star(\theta_t)$. This framework can be used to show that OMD achieves

$$\text{Regret}_T(u) \leq R(u) + \min_{v \in S} R(v) + \sum_{t=1}^{T} D_{R^\star}\left(-\sum_{i=1}^{t} z_i \Bigg|\Bigg| - \sum_{i=1}^{t-1} z_i\right) \tag{5}$$

where $D_{R^\star}(u||v)$ is the Bregman divergence between u and v under $R^\star$. If R is $(1/\eta)$-strongly-convex with respect to $||\cdot||$, then $R^\star$ is η-strongly-smooth with respect to the dual norm $||\cdot||_\star$: in this case,

$$\text{Regret}_T(u) \leq R(u) + \min_{v \in S} R(v) + \frac{\eta}{2} \sum_{t=1}^{T} ||z_t||_\star^2 \tag{6}$$

3.2 Example algorithms

We can now crank out algorithms under the OMD framework. All these algorithms enjoy the bound in Theorem 2.2 (or Eq. (6)).

Online gradient descent (OGD): Assumes an unconstrained domain $S = \mathbb{R}^d$ and an l_2 regularizer $R(w) = \frac{1}{2\eta} ||w||_2^2$. We have $g(\theta) = \eta\theta$ and

$$w_t = w_{t-1} - \eta z_{t-1} \tag{7}$$

Online gradient descent with lazy projections (OGDLP): Assumes a general convex set S and an l_2 regularizer $R(w) = \frac{1}{2\eta} ||w||_2^2$. Note that

$$w_t = \arg\min_{w \in S} \frac{1}{2\eta} ||w||_2^2 - \langle w, \theta_t \rangle = \arg\min_{w \in S} ||w - \eta\theta_t||_2^2 \tag{8}$$

Thus the link function $g(\theta)$ projects $\eta\theta$ onto S.

Unnormalized exponentiated gradient descent (UEG): Assumes an unconstrained domain $S = \mathbb{R}^d$ and a shifted entropy regularizer $R(w) = \frac{1}{\eta} \sum_i w_i(\log w_i - 1 - \log \lambda)$ where $\lambda > 0$. We have $g_i(\theta) = \lambda \exp(\eta\theta_i)$, thus $w_1 = (\lambda \ldots \lambda)$ and for $i > 1$:

$$[w_t]_i = [w_{t-1}]_i \exp(-\eta[z_{t-1}]_i) \tag{9}$$

Normalized exponentiated gradient descent (NEG): Assumes a probability simplex $S = \{w \in \mathbb{R}^d : w \geq 0, \sum_i w_i = 1\}$ and an entropy regularizer $R(w) = \frac{1}{\eta} \sum_i w_i \log w_i$. We have $g_i(\theta) = \frac{\exp(\eta\theta_i)}{\sum_j \exp(\eta\theta_j)}$, thus $w_1 = (1/d \dots 1/d)$ and for $i > 1$:

$$[w_t]_i = \frac{[w_{t-1}]_i \exp(-\eta[z_{t-1}]_i)}{\sum_j [w_{t-1}]_j \exp(-\eta[z_{t-1}]_j)} \tag{10}$$

4 Applications to classification problems

The central step in applying OCO to a classification problem is finding the right "convex surrogate" of the problem.

4.1 Perceptron

At each round, we're given a point $x_t \in \mathbb{R}^d$. We predict $p_t \in \{-1, +1\}$ and receive the true class $y_t \in \{-1, +1\}$. The (non-convex) loss is given by

$$l(p_t, y_t) := \begin{cases} 1 & \text{if } p_t \neq y_t \\ 0 & \text{if } p_t = y_t \end{cases}$$

Note that the cumulative loss $M := \sum_t l(p_t, y_t)$ is the number of mistakes.

Convex surrogate: We maintain a vector $w_t \in \mathbb{R}^d$ that defines $p_t := \text{sign}\langle w_t, x_t \rangle$. We use a "hinge" loss

$$f_t(w_t) := \max(0, 1 - y_t \langle w_t, x_t \rangle)$$

which by the particular construction is convex and upperbounds the original loss $l(p_t, y_t)$. Using a sub-gradient $z_t \in \partial f_t(w_t)$ where $z_t = -y_t x_t$ if $y_t \langle w_t, x_t \rangle \leq 1$ and $z_t = 0$ otherwise, we can now run OGD using some $\eta > 0$: $w_1 := 0$ and

$$w_{t+1} := \begin{cases} w_t + \eta y_t x_t & \text{if } y_t \langle w_t, x_t \rangle \leq 1 \\ w_t & \text{if } y_t \langle w_t, x_t \rangle > 1 \end{cases}$$

Let $L := \max_t ||z_t||$. It's possible to apply Eq. (6) and show that for any $u \in \mathbb{R}^d$

$$M \leq \sum_t f_t(u) + ||u||_2 \, L \sqrt{\sum_t f_t(u)} + L^2 \, ||u||_2^2$$

In particular, if there exists $u \in \mathbb{R}^d$ such that $\sum_t f_t(u) = 0$, we have $M \leq L^2 \, ||u||_2^2$.

4.2 Weighted majority

At each round, we're given a point $x_t \in \mathcal{X}$ and d hypotheses $\mathcal{H} = \{h_1, \dots, h_d\}$ where $h_i : \mathcal{X} \to \{0, 1\}$. We make a choice $p_t \in [d]$ and receive the true class $y_t \in \{0, 1\}$. The (non-convex) loss is given by

$$l(p_t, y_t) := \begin{cases} 1 & \text{if } h_{p_t}(x_t) \neq y_t \\ 0 & \text{if } h_{p_t}(x_t) = y_t \end{cases}$$

Convex surrogate: We maintain a vector $w_t \in \{w \in \mathbb{R}^d : w \geq 0, \sum_i w_i = 1\}$. This vector defines "weighted majority": $p_t = 1$ if $\sum_{i=1}^d [w_t]_i h_i(x_t) \geq 1/2$ and $p_t = 0$ otherwise. We use the convex loss function:

$$f_t(w_t) := \sum_{i=1}^{d} [w_t]_i \, |h_i(x_t) - y_t| = \langle w_t, z_t \rangle$$

where $[z_t]_i := |h_i(x_t) - y_t|$ (thus z_t is also the gradient of f_t). Hence we have an online linear problem suitable for NEG. It's possible to show that if there exists some $h \in \mathcal{H}$ such that $\sum_{t=1}^T |h(x_t) - y_t| = 0$, then NEG achieves $\sum_t f_t(w_t) \leq 4 \log d$.

4.2.1 Multi-armed bandit

A problem closely related to weighted majority is the so-called multi-armed bandit problem. At each round, there d slot machines ("one-armed bandits") to choose from. We make a choice $p_t \in [d]$ and receive the cost of playing that machine: $[y_t]_{p_t} \in [0, 1]$. A crucial aspect of the problem is the existence of unobserved costs $[y_t]_i \in [0, 1]$ for $i \neq p_t$, because if we observe all $y_t \in [0, 1]^d$ we can just formulate it as an online linear problem by minimizing the expected loss

$$f_t(w_t) := \langle w_t, y_t \rangle$$

where $w_t \in \{w \in \mathbb{R}^d : w \geq 0, \sum_i w_i = 1\}$ again defines "weighted majority" over d machines. Since y_t is the gradient of f_t, another way of stating the difficulty is that gradients are not fully observed.

A solution is to use a p_t-dependent estimator $z_t^{(p_t)}$ of the gradient y_t as follows:

$$[z_t^{(p_t)}]_i \begin{cases} [y_t]_i / [w_t]_i & \text{if } i = p_t \\ 0 & \text{if } i \neq p_t \end{cases}$$

This is indeed an unbiased estimator of y_t over the randomness of p_t since

$$\mathbf{E}[z_t^{(p_t)}]_i := \sum_{p_t=1}^{d} p(p_t)[z_t^{(p_t)}]_i = w_i \frac{[y_t]_i}{[w_t]_i} + \sum_{p_t \neq i} 0 = [y_t]_i$$

Thus we can run NEG by substituting the unobserved gradient y_t with $z_t^{(p_t)}$. Note that the algorithm will be slightly different from the weighted majority algorithm since we need to actually make the prediction $p_t \sim w_t$ which is required for computing $z_t^{(p_t)}$. It's possible to derive regret bounds where the regret is defined as the difference between the algorithm's expected cumulative cost (over the randomness of p_t) and the cumulative cost of the best machine:

$$\mathbf{E}\left[\sum_{t=1}^{T} [y_t]_{p_t}\right] - \min_{i \in [d]} \sum_{t=1}^{T} [y_t]_i$$

Reference

Shalev-Shwartz, S. (2011). Online Learning and Online Convex Optimization.