MIT 6.7220/15.084 — Nonlinear Optimization Thu, May 2nd 2024

Lecture 18
Online mirror descent

Instructor: Prof. Gabriele Farina (✉ gfarina@mit.edu)*

We saw in Lecture 9 that the mirror descent algorithm is a generalization of projected gradient descent that works for generalized notions of distance. In this lecture, we will see how to apply mirror descent to online learning problems. The resulting algorithm is called *online mirror descent*, often abbreviated with the acronym OMD.

1 From mirror descent to online mirror descent

Recall that the mirror descent algorithm works with generalized notions of distances called *Bregman divergences*. Bregman divergences are "generated" by a 1-strongly convex function $\varphi : \Omega \to \mathbb{R}$ called a *distance-generating function*. In turn, the Bregman divergence defines the *proximal operator*:

$$\begin{aligned}\mathrm{Prox}_{\varphi}(\eta \nabla f(x_t), x_t) \coloneqq \arg\min_{x} \eta \langle \nabla f(x_t), x \rangle + \mathrm{D}_{\varphi}(x \,\|\, x_t)\\ \text{s.t. } x \in \Omega.\end{aligned}$$

When φ is the squared Euclidean norm $\varphi(x) = \frac{1}{2}\|x\|_2^2$, the proximal operator reduces to the usual Euclidean projection operator $\Pi_{\Omega}(x_t - \eta \nabla f(x_t))$, so that mirror descent reduces to regular (Euclidean) projected gradient descent.

With this notation under our belt, the regular (offline) version of mirror descent picks iterates according to

$$x_{t+1} \coloneqq \mathrm{Prox}_{\varphi}(\eta \nabla f(x_t), x_t).$$

The *online* version of mirror descent is equally straightforward: simply replace the function f with the time-varying objective f_t:

$$\boxed{x_{t+1} \coloneqq \mathrm{Prox}_{\varphi}(\eta \nabla f_t(x_t), x_t)}\,.$$

So, when φ is the squared Euclidean norm, the online mirror descent algorithm reduces to an *online* version of projected gradient descent, called *online projected gradient descent*. As we remarked in Lecture 9, other choices of distance-generating functions φ are possible depending on the domain Ω of interest. In Section 2, we will for example see how the negative entropy distance-generating function $\varphi(x) = x_1 \log x_1 + \cdots + x_n \log x_n$ leads to an especially appealing algorithm in the context of games, where Ω is the set of randomized strategies of the players, that is, the set of probability distributions over the actions available to each player.

*These notes are class material that has not undergone formal peer review. The TAs and I are grateful for any reports of typos.

1.1 Regret analysis

Theorem 1.1. Let $\|\cdot\|$ be the norm with respect to which the DGF φ is 1-strongly convex, and $\|\cdot\|_*$ be the dual norm. If all functions $f_t : \Omega \to \mathbb{R}$ are convex, the regret of online mirror descent is bounded by

$$R_T := \sum_{t=1}^{T} f_t(x_t) - \min_{x\in\Omega} \sum_{t=1}^{T} f_t(x) \leq \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_1) + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f_t(x_t)\|_*^2.$$

In particular, assuming that all dual gradient norms are upper bounded by G, and setting

$$\eta := \sqrt{\frac{2\mathrm{D}_\varphi(x \,\|\, x_1)}{G^2 T}}$$

we find

$$R_T \leq G\sqrt{2\mathrm{D}_\varphi(x \,\|\, x_1) \cdot T}.$$

Proof. Since f_t is convex, we can use the mirror descent lemma (see Lecture 9), which states that,

$$f_t(x_t) \leq f_t(x) + \langle \nabla f_t(x_t), x_t - x_{t+1}\rangle - \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_{t+1}) + \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_t) - \frac{1}{\eta}\mathrm{D}_\varphi(x_{t+1} \,\|\, x_t) \qquad \forall x \in \Omega.$$

Using the Cauchy-Schwarz inequality, we can bound the right-hand side by

$$f_t(x_t) \leq f_t(x) + \|\nabla f_t(x_t)\|_* \cdot \|x_t - x_{t+1}\| - \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_{t+1}) + \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_t) - \frac{1}{\eta}\mathrm{D}_\varphi(x_{t+1} \,\|\, x_t).$$

Using Young's inequality, as well as the 1-strong convexity of φ, which implies $\frac{1}{\eta}\mathrm{D}_\varphi(x_{t+1} \,\|\, x_t) \geq \frac{1}{2\eta}\|x_{t+1} - x_t\|^2$, we therefore have

$$\begin{aligned} f_t(x_t) &\leq f_t(x) + \frac{\eta}{2}\|\nabla f_t(x_t)\|_*^2 + \frac{1}{2\eta}\|x_t - x_{t+1}\|^2 - \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_{t+1}) + \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_t) - \frac{1}{2\eta}\|x_{t+1} - x_t\|^2 \\ &\leq f_t(x) + \frac{\eta}{2}\|\nabla f_t(x_t)\|_*^2 - \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_{t+1}) + \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_t). \end{aligned}$$

Summing over $t = 1, ..., T$, we find

$$\sum_{t=1}^{T} f_t(x_t) \leq \sum_{t=1}^{T} f_t(x) + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f_t(x_t)\|_*^2 - \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_T) + \frac{1}{\eta}\mathrm{D}_\varphi(x \,\|\, x_1).$$

Ignoring the negative term yields the first part of the statement. □

1.2 Coping with an unknown time horizon

One might object that in an online learning setting, the time horizon T is not known in advance. Luckily, it turns out that the bound we just established can be adapted to this case. We start with a black-box reduction, called the *doubling trick*.

Remark 1.1. If T is not known in advance, we can use the *doubling trick* to obtain a bound of $O\Big(G\sqrt{\mathrm{D}_\varphi(x \,\|\, x_1)\log T}\Big)$. The idea is to change the value of η every time we reach an iteration count that is a power of 2. So, we set $\eta = \sqrt{2\mathrm{D}_\varphi(x \,\|\, x_1)/(G^2 2^k)}$ for $2^k \leq T < 2^{k+1}$.

Proof. The regret incurred by the algorithm is upper bound by the regret incurred in each of the intervals $2^k \leq T < 2^{k+1}$. Suppose the algorithm has been run until time $2^K \leq T < 2^{K+1}$. Hence, the regret is upper bounded by

$$\begin{aligned}\mathrm{Reg}_T &\leq \left(G\sqrt{\frac{\mathrm{D}_\varphi(x \,\|\, x_1)}{2}} \sum_{k=0}^{K} \left(\sqrt{2}\right)^k\right) + \left(\sum_{k=0}^{K-1} \sqrt{\frac{\mathrm{D}_\varphi(x \,\|\, x_1)}{2G^2 2^k}} G^2 2^k\right) + \left(\sqrt{\frac{2\mathrm{D}_\varphi(x \,\|\, x_1)}{2G^2 2^K}}\right) G^2 (T - 2^K) \\ &= G\sqrt{\frac{\mathrm{D}_\varphi(x \,\|\, x_1)}{2}} \left(\sum_{k=0}^{K} \left(\sqrt{2}\right)^k + \sum_{k=0}^{K-1} \left(\sqrt{2}\right)^k\right) + \left(\sqrt{\frac{\mathrm{D}_\varphi(x \,\|\, x_1)}{2G^2 2^K}}\right) G^2 (T - 2^K)\end{aligned}$$

In particular, since $T < 2^{K+1}$,

$$\begin{aligned}\mathrm{Reg}_T &\leq G\sqrt{\frac{\mathrm{D}_\varphi(x \,\|\, x_1)}{2}} \left(\sum_{k=0}^{K} \left(\sqrt{2}\right)^k + \sum_{k=0}^{K-1} \left(\sqrt{2}\right)^k\right) + \left(\sqrt{\frac{\mathrm{D}_\varphi(x \,\|\, x_1)}{2G^2 2^K}}\right) G^2 2^K \\ &= G\sqrt{\frac{\mathrm{D}_\varphi(x \,\|\, x_1)}{2}} \left(\sum_{k=0}^{K} \left(\sqrt{2}\right)^k + \sum_{k=0}^{K} \left(\sqrt{2}\right)^k\right) \\ &= G\sqrt{2\mathrm{D}_\varphi(x \,\|\, x_1)} \sum_{k=0}^{K} \left(\sqrt{2}\right)^k \\ &= G\sqrt{2\mathrm{D}_\varphi(x \,\|\, x_1)} \frac{\left(\sqrt{2}\right)^{K+1} - 1}{\sqrt{2} - 1} \leq G\sqrt{2\mathrm{D}_\varphi(x \,\|\, x_1) \cdot T} \frac{\sqrt{2}}{\sqrt{2} - 1}.\end{aligned}$$

Hence, the doubling trick guarantees a degradation in the regret bound by a factor at most $\sqrt{2}/\left(\sqrt{2} - 1\right) \approx 3.41$. □

Remark 1.2. An alternative approach to not knowing the time horizon is to *dynamically* change the stepsize η at all times t, by simply setting it to

$$\eta_t := \sqrt{\frac{2\mathrm{D}_\varphi(x \,\|\, x_1)}{G^2 t}}$$

(that is, replace T with t in the formula for η). This approach is sound, but working out the details requires being more careful when using the telescoping argument enabled by the mirror descent lemma used in the proof of Theorem 1.1.

2 Two notable examples

Let's investigate applying the online mirror descent algorithm with two standard choices of distance-generating functions: the squared Euclidean norm and negative entropy.

2.1 Online gradient descent

When the distance-generating function is the squared Euclidean norm, proximal steps are Euclidean projection. In this case, with are left with the algorithm

$$\boxed{x_{t+1} := \Pi_\Omega(x_t - \eta \nabla f_t(x_t))},$$

which is known with the name *online gradient descent (OGD)*.

2.2 Multiplicative weights update

When $\Omega = \Delta^n$ is a probability simplex (this is the case when playing a normal-form games), the negative entropy is a natural choice for the distance-generating function (see also Lecture 9). In this case, we have the algorithm

$$\boxed{x_{t+1} \propto x_t \odot \exp\{-\eta \nabla f_t(x_t)\}},$$

where $\odot$ denotes componentwise multiplication, and the exponential is meant again componentwise. Starting from the uniform distribution $x_1 = \left(\frac{1}{n}, ..., \frac{1}{n}\right)$ Applying the general bound derived in Theorem 1.1, we find that the regret of this algorithm is bounded by

$$R_T \leq G\sqrt{2\mathrm{D}_\varphi(x \,\|\, x_1) \cdot T} \leq G\sqrt{2\log(n) \cdot T},$$

where again $G \coloneqq \max \|\nabla f_t(x_t)\|_\infty$.

Remark 2.1. In games, G is bounded by the maximum absolute value of any payoff in the game. Hence, the regret of the multiplicative weights update algorithm scales only with the *logarithm* of the number of actions of the player!