10-725: Convex Optimization **Spring 2023**

Lecture 9: February 14th

Lecturer: Siva Balakrishnan

Happy Valentines Day! Today we'll finally leave the realm of Euclidean geometry, and talk about the mirror descent algorithm which aims to generalize the GD algorithm, by trying to better exploit properties of the function we are trying to optimize and the constraint set over which we're optimizing.

We will follow quite closely Chapter 4 of Bubeck's book, which is a very clear presentation of the idea of mirror descent.

9.1 The Hidden Dimension Dependence of the Subgradient Method/GD

We've repeatedly made a point to emphasize the dimension-free nature of the guarantees of GD/the subgradient method. For instance, if we're optimizing a function which is G-Lipschitz and we initialize in some radius R from an optimal point, we know that after k iterations,

$$f(x^{\text{best}}) - f(x^*) \leq \frac{GR}{\sqrt{k}}.$$

Suppose for example that we're trying to optimize the function,

$$f(x) = \|x\|_1,$$

over the set C which is the unit (Euclidean) ball $B_2(0, 2)$. The set has diameter 2, so $R \leq 2$. However, the subgradient/gradient of f has a norm which is $\sqrt{d}$. Consequently, the best guarantee we can get from our analysis so far is that,

$$f(x^{\text{best}}) - f(x^*) \leq \frac{2\sqrt{d}}{\sqrt{k}},$$

which can be disappointing if d is large. However, notice that subgradients of f do have their ℓ_∞ norm bounded (by 1), so if we could derive a similar guarantee replacing G by an upper bound on the ℓ_∞ norm then we'd be much happier.

Notice, that this isn't a deficiency of our analysis, if we're optimizing $\|x\|_1$, and are at the point $(1, \frac{1}{d-1}, \ldots, \frac{1}{d-1})$ then we'll have to choose tiny step-sizes in the subgradient method (roughly like $1/d$) to avoid bouncing around.

9.1.1 Things you can ignore

The problem above is a manifestation of a general phenomenon that we won't belabour too much. The gradient of a function, defined over a space $\mathcal{D}$, typically doesn't live in the same space, rather it's an element of the dual space. The example to keep in mind is the $f(x) = \|x\|_1$ function. Suppose we moved to an infinite dimensional space, the function would still be well-defined (finite) for any x for which $\|x\|_1 < \infty$, however the gradient/subgradient will never have finite ℓ_1 (or ℓ_2) norm. Rather the gradient/subgradient will have finite ℓ_∞ norm, and this is the space in which it lives (the space of vectors with bounded ℓ_∞ norm), which is dual to the space where the function is well-defined.

This in turn suggests that GD updates, $x^{t+1} = x^t - \eta \nabla f(x^t)$ don't really make sense in many infinite dimensional settings (since the gradient and iterate are points that live in very different spaces – and their "lengths" are on very different scales). We might not care so much about infinite dimensional optimization, but we'd still like to measure things like Lipschitzness and diameters in the correct way and design our algorithms accordingly (or we might pay a hefty dimension-dependent price).

9.1.2 Punchline

To keep in mind a relatively concrete example. Suppose we're interested in optimizing a convex function f, with ℓ_∞ bounded subgradients, over the simplex $\Delta^d = \{x \geq 0 : \sum_{i=1}^d x_i = 1\}$. We've discussed above that the subgradient method can guarantee that,

$$f(x^{\text{best}}) - f(x^*) \leq \frac{\sqrt{d}}{\sqrt{k}}.$$

On the other hand, using the idea of mirror descent, we'll derive a different algorithm (roughly, an exponentiated subgradient method) which will yield the guarantee that,

$$f(x^{\text{best}}) - f(x^*) \lesssim \sqrt{\frac{\log d}{k}},$$

which is an exponential improvement in terms of the dimension dependence.

It is also worth noting that unlike in previous lectures where our focus was on improving the dependence on k under different hypotheses on the function we were optimizing, in this lecture our focus is on improving other parts of the result.

9.2 Mirror Maps

Throughout we'll assume we're optimizing a convex function f over a convex set C. Throughout the rest of this lecture note the norm $\|\cdot\|$ will be some (not necessarily Euclidean) norm

(we get to pick it).

The first thing we'll need to describe the MD algorithm, is the so-called mirror map Φ. The mirror map Φ will be a differentiable function defined on a convex, open domain D whose closure $\bar{D} \supseteq C$.

The mirror map Φ will be an α-strongly convex function on C with respect to the norm $\|\cdot\|$, i.e. for any $x, y \in C$, we have that,

$$\Phi(y) \geq \Phi(x) + \nabla\Phi(x)^T(y-x) + \frac{\alpha}{2}\|x-y\|^2.$$

You can ignore these conditions on first reading:

1. We will assume that $\nabla\Phi$ takes all possible values in $\mathbb{R}^d$, i.e. $\nabla\Phi(D) = \mathbb{R}^d$.
2. We'll also assume that $\|\nabla\Phi(x)\| \to \infty$ as $x \to \text{boundary}(D)$, i.e. the gradient blows up at the boundary of D.

We'll briefly revisit why they're useful properties when we describe the mirror descent algorithm.

9.3 Bregman Divergences

Associated with this convex function, is a Bregman divergence, i.e. given $x, y \in D$:

$$D_\Phi(x,y) = \Phi(x) - \Phi(y) - \nabla\Phi(y)^T(x-y).$$

Given this Bregman divergence and any point y (potentially outside C but inside D) we can define the Bregman projection,

$$\Pi_C(y) = \arg\min_{x \in C} D_\Phi(x,y).$$

There are some main examples to keep in mind for this lecture:

1. **Usual Gradient Descent:** Suppose we take $\Phi(x) = \frac{1}{2}\|x\|_2^2$ (this is a 1-strongly convex function with respect to the Euclidean norm). Then we get,

   $$D_\Phi(x,y) = \frac{1}{2}\|x\|_2^2 - \frac{1}{2}\|y\|_2^2 - y^T(x-y) = \frac{1}{2}\|x-y\|_2^2.$$

   As we will see in a little while our mirror descent updates in this case are identical to our (projected) GD updates from before.

2. **Exp Gradient Descent:** Suppose we take $\Phi(x) = \sum_{i=1}^d x_i \log x_i$, which is defined over the (strictly) positive reals. We get,

$$\begin{aligned} D_\Phi(x,y) &= \sum_{i=1}^d x_i \log x_i - \sum_{i=1}^d y_i \log y_i - \sum_{i=1}^d (1+\log y_i)(x_i - y_i) \\ &= \sum_{i=1}^d x_i \log(x_i/y_i) - \sum_{i=1}^d (x_i - y_i). \end{aligned}$$

It turns out that $\Phi(x)$ is strictly convex over the simplex *with respect to the ℓ_1-norm.* To see this, we recall Pinsker's inequality (you might have seen this in a Stats class like 36-705/36-709) which tells us that for two distributions p, q (vectors on the d-dimensional simplex):

$$\ell_1(p,q) \leq \sqrt{2\mathrm{KL}(p,q)}.$$

Thus, over the simplex we see that,

$$D_\Phi(x,y) = \mathrm{KL}(x,y) \geq \frac{1}{2}\|x-y\|_1^2,$$

i.e. equivalently Φ is 1-strongly convex with respect to the ℓ_1-norm on the simplex.

9.3.1 Properties of Bregman Divergences

There are a few properties of Bregman divergences that will be useful in our proof of the rate of convergence of mirror descent.

Lemma 9.1 (Three-point Property) *For $x, y, z \in D$,*

$$D_\Phi(x,y) + D_\Phi(z,x) - D_\Phi(z,y) = (\nabla\Phi(x) - \nabla\Phi(y))^T(x-z).$$

Proof: We simply use the definition of the Bregman divergence. ∎

Lemma 9.2 (Pythagoras Theorem) *Suppose that C is a convex set, $x \in C$ and $y \in \mathbb{R}^d$. Then,*

$$D_\Phi(x, \Pi_C(y)) + D_\Phi(\Pi_C(y), y) \leq D_\Phi(x,y).$$

Proof: We simply use the first-order optimality conditions for the Bregman projection, i.e. we know that,

$$\Pi_C(y) = \arg\min_{x \in C} D_\Phi(x,y),$$

so this means that,

$$(\nabla\Phi(\Pi_C(y)) - \nabla\Phi(y))^T(\Pi_C(y) - x) \leq 0,$$

for any $x \in C$. This is the claimed result. ∎

9.4 Local Approximation Description of Mirror Descent

A natural way to generalize the gradient descent algorithm is simply to use a general Bregman divergence to measure proximity in the local linear approximation of gradient descent. Despite being a seemingly minor modification to the update step, it's worth noting that changing this term essentially re-shapes the space we're optimizing over in a non-trivial way – changing how we measure distances is similar to stretching and shrinking the space around the current iterate, and is also at the heart of things like Newton's method.

Concretely, given our current iterate x^t we compute the next iterate by solving the program:

$$x^{t+1} = \arg\min_{x \in C} f(x^t) + \nabla f(x^t)^T(x - x^t) + \frac{1}{\eta} D_\Phi(x, x^t).$$

We cannot always solve for this iteration in closed-form (similar to how we can't always solve a prox. computation in closed form). However, it will turn out that for nice mirror maps this iteration has a simple description. We'll develop this more in the next section.

9.5 Mirror Description of Mirror Descent

This presentation is now closer to the classical presentation of Nemirovski and Yudin. We could equivalently describe the iteration from the previous section as:

$$x^{t+1} = \arg\min_{x \in C} \Phi(x) - (\nabla\Phi(x^t) - \eta\nabla f(x^t))^T x.$$

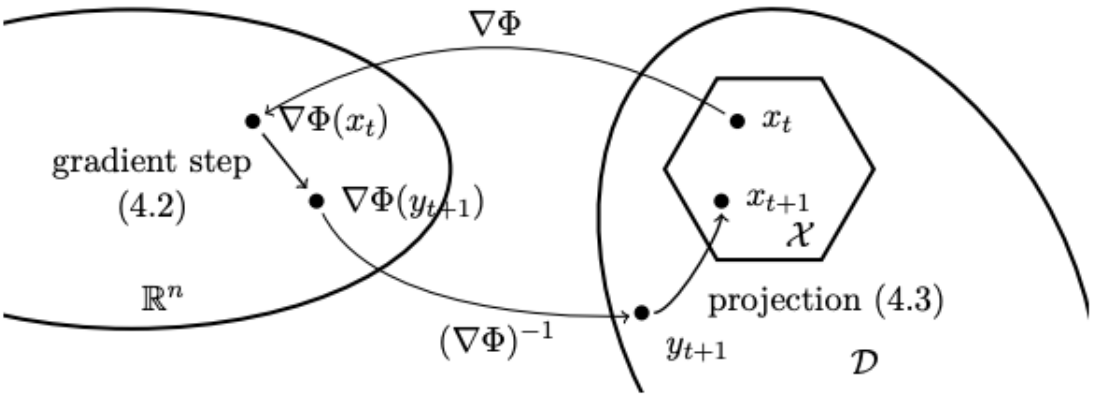
Suppose we could find a point y^{t+1} such that,

$$\nabla\Phi(y^{t+1}) = \nabla\Phi(x^t) - \eta\nabla f(x^t),$$

then we could re-write our iteration as:

$$x^{t+1} = \arg\min_{x \in C} \Phi(x) - \nabla\Phi(y^{t+1})^T x = \arg\min_{x \in C} D_\Phi(x, y^{t+1}) = \Pi_C(y^{t+1}).$$

This iteration is best described/understood with a figure (from Bubeck's book).



The technical assumptions we have made (that $\nabla\Phi$ takes all possible values, and that Φ is strongly convex) ensure that we can always find this point y^{t+1} (and that it is unique). The assumption that Φ blows up at the boundary of D ensures that the Bregman projection is well-defined.

9.6 Mirror Descent on the Simplex

As we described earlier, our goal is to minimize a function f, over the simplex. We use the entropy as the mirror map, and then we see that the mirror descent updates are given by,

$$y^{t+1} = x^t \exp(-\eta \nabla f(x^t)),$$

where the operations are interpreted element-wise.

Then the Bregman projection onto the simplex is simply,

$$x^{t+1} = y^{t+1} / \|y^{t+1}\|_1.$$

(This is a good exercise to check. The easiest way involves Lagrange multipliers for the simplex constraints – an idea that we'll learn about shortly.)

This algorithm is sometimes called (projected) exponentiated gradient descent.

9.7 Lipschitzness

We'll also need to define a notion of Lipschitzness that uses a general norm. The standard way is to say a function f is G-Lipschitz with respect to a norm $\|\cdot\|$ if, the dual norm of

the gradient/subgradient of f is bounded, i.e. for any $x \in \text{dom}(f)$,

$$\|g_x\|_* \leq G.$$

Recall, that for any norm, the *dual* norm is defined by:

$$\|v\|_* = \sup_{u:\|u\|\leq 1} u^T v.$$

Some canonical examples include:

1. The dual norm for the ℓ_2 norm, is the ℓ_2 norm.
2. The dual norm for the ℓ_1 norm, is the ℓ_∞ norm.

It's easy to see that for a pair of vectors with $v \neq 0$ we always have a Cauchy-Schwarz inequality,

$$u^T v = \|v\| u^T \frac{v}{\|v\|} \leq \|v\| \|u\|_*.$$

Now, we can go back to the connection to Lipschitzness, for a convex function f we know that if the subgradients have bounded dual norm then,

$$|f(x) - f(y)| \leq |g_x^T (y - x)| \leq \|g_x\|_* \|x - y\| \leq G\|x - y\|.$$

This is our usual definition of Lipschitzness (just generalized to work with general norms). We have verified one direction, i.e. if the function has gradients bounded in the dual norm then it is Lipschitz with respect to the norm. The other direction also turns out to be easy to check, i.e. if the function is Lipschitz with respect to some norm, then its gradients are bounded in the dual norm.

9.8 Mirror Descent Rates of Convergence

The main theorem for mirror descent is the following:

Theorem 9.3 *Suppose that f is G-Lipschitz with respect to the norm $\|\cdot\|$, and that Φ is an α-strongly convex mirror map with respect to the same norm. Suppose we denote an upper bound on $D_\Phi(x^*, x_0)$ by R^2, then the iterates of mirror descent with step-size $\eta = \frac{R}{G}\sqrt{\frac{2\alpha}{k}}$ have the property that,*

$$f\left(\frac{1}{k}\sum_{i=0}^{k-1} x^i\right) - f(x^*) \leq RG\sqrt{\frac{2}{\alpha k}}.$$

1. This result generalizes our previous result for the subgradient method (in which case we use $\Phi(x) = \|x\|_2^2/2$ and we use the usual Euclidean norm).
2. In the case of the simplex, using $\Phi(x) = \sum_{i=1}^d x_i \log x_i$, we see that, $R^2 \leq \log d$ if we initialize at $x^0 = (1/d, \ldots, 1/d)$ (i.e. the uniform distribution), since,

$$R^2 := \sup_{x \in \Delta^d} D_\Phi(x, x^0) = \sup_{x \in \Delta^{d-1}} \sum_{i=1}^d [x_i \log x_i + x_i \log d] \leq \log d.$$

   If our function has subgradients bounded in ℓ_∞ norm (by 1 say), then we see that,

$$f\left(\frac{1}{k}\sum_{i=0}^{k-1} x^i\right) - f(x^*) \leq \sqrt{\frac{2\log d}{k}},$$

   which can be much faster than the rate that (vanilla) gradient descent achieves in this setting. This observation is at the heart of the multiplicative weights algorithm (sometimes also called 'Hedge'), and of much of online learning.

Proof: We begin by observing that (by convexity),

$$\begin{aligned}
f(x^t) - f(x^*) &\leq g_{x^t}^T (x^t - x^*) \\
&= \frac{1}{\eta}(\nabla\Phi(x^t) - \nabla\Phi(y^{t+1}))^T (x^t - x^*) \\
&= \frac{1}{\eta}(D_\Phi(x^*, x^t) + D_\Phi(x^t, y^{t+1}) - D_\Phi(x^*, y^{t+1})) \\
&\leq \frac{1}{\eta}(D_\Phi(x^*, x^t) + D_\Phi(x^t, y^{t+1}) - D_\Phi(x^*, x^{t+1}) - D_\Phi(x^{t+1}, y^{t+1})).
\end{aligned}$$

Now we additionally observe that,

$$\begin{aligned}
D_\Phi(x^t, y^{t+1}) - D_\Phi(x^{t+1}, y^{t+1}) &= \Phi(x^t) - \Phi(x^{t+1}) - \nabla\Phi(y^{t+1})^T (x^t - x^{t+1}) \\
&\leq (\nabla\Phi(x^t) - \nabla\Phi(y^{t+1}))^T (x^t - x^{t+1}) - \frac{\alpha}{2}\|x^t - x^{t+1}\|^2 \\
&= \eta g_{x^t}^T (x^t - x^{t+1}) - \frac{\alpha}{2}\|x^t - x^{t+1}\|^2 \\
&\leq \eta G \|x^t - x^{t+1}\| - \frac{\alpha}{2}\|x^t - x^{t+1}\|^2 \\
&\leq \frac{(\eta G)^2}{2\alpha},
\end{aligned}$$

where the last inequality simply follows by maximizing the RHS.

From this we obtain that,

$$f(x^t) - f(x^*) \leq \frac{1}{\eta}(D_\Phi(x^*, x^t) - D_\Phi(x^*, x^{t+1})) + \frac{\eta G^2}{2\alpha},$$

telescoping we obtain that,

$$\frac{1}{k}\sum_{t=0}^{k-1} f(x^k) - f(x^*) \leq \frac{1}{k\eta} D_\Phi(x^*, x^0) + \frac{\eta G^2}{2\alpha},$$

and this by our choice of step-size and the observation that,

$$f\left(\frac{1}{k}\sum_{i=0}^{k-1} x^i\right) \leq \frac{1}{k}\sum_{t=0}^{k-1} f(x^k)$$

concludes the proof. ■