Mirror Descent and Online Learning*

Fred Zhang†

March 24, 2019

Abstract

We discuss the motivation and basic setup of *mirror descent*, a fundamental algorithm for convex optimization. We focus on the applications of mirror descent to online learning problems.

1 Introduction: Gradient Descent

The focus of this note is first-order method for optimizing convex functions $f : \mathbb{R}^n \to \mathbb{R}$. Throughout we assume that the objective function f is differentiable and L-Lipschitz, *i.e.*, $\|\nabla f(x)\| \le L$. *Gradient descent* is a natural algorithm for this task. Let $x^* = \arg\min f(x)$. Suppose we start with x_0 such that $\|x_0 - x^*\| \le R$. We choose the step size

$$\eta_t = \frac{R}{L\sqrt{t}}, \tag{step size}$$

and follow the simple update rule:

```
Gradient-Descent(x_0, f):
  For t = 1 to T :
      x_{t+1} ⟵ x_t − η_t · ∇f(x_t).
```

To obtain an ϵ-approximate optimal point, the `Gradient-Descent` requires $O\left(1/\epsilon^2\right)$ iterations. While the proof is not particularly insightful, we include here as it will be reused later when we come to its online version.

Theorem 1.1 (Convergence of `Gradient-Descent`)**.** Let x_0 such that $\|x_0 - x^*\| \le R$. The `Gradient-Descent` algorithm for T iterations, starting at x_0, satisfies

$$f\left(\frac{1}{T}\sum_{i=0}^{T-1} x_i\right) - f(x^*) \le \frac{RL}{\sqrt{T}}.$$

Proof. We first try to bound the distance of each point to the optimum.

$$\begin{aligned}
f(x_i) - f(x^*) &\le \nabla f(x_i)^\top (x_i - x^*) && \text{(by convexity)} \\
&= \frac{1}{\eta_i}(x_i - x_{i+1})^\top (x_i - x^*) && \text{(by definition of the algorithm)} \\
&= \frac{1}{2\eta_i}\left(\|x_i - x^*\|^2 + \|x_i - x_{i+1}\|^2 - \|x_{i+1} - x^*\|^2\right) && \left(\text{since } a^\top b = \frac{1}{2}\left(\|a\|^2 + \|b\|^2 - \|a-b\|^2\right)\right) \\
&= \frac{1}{2\eta_i}\left(\|x_i - x^*\|^2 - \|x_{i+1} - x^*\|^2\right) + \frac{\eta_i}{2}\|\nabla f(x_i)\|^2. && \text{(by definition of the algorithm)}
\end{aligned}$$

*This is a lecture note with certain technical details omitted. For a formal treatment of the materials here, see the monographs by Bubeck (2015) [Bub15] and Hazan (2016) [Haz16].

†Harvard University. Email: hzhang@g.harvard.edu.

Since $\|\nabla f(x_i)\|^2 \le L^2$ and $\|x_0 - x^*\| \le R$, summing the inequality above over all i gives

$$\sum_{i=0}^{T-1} \left(f(x_i) - f\left(x^*\right)\right) \le \frac{R^2}{2\eta_t} + \frac{\eta L^2 T}{2}. \tag{1}$$

Finally, plugging in the value of η_t (step size) and applying Jensen's inequality completes the proof. □

Remark 1.1. Suppose the problem is constrained with a convex feasible region $\mathscr{X}$. The theorem also holds for *projected gradient descent*, where after the gradient update we project the result back to $\mathscr{X}$ (by Euclidean distance). The analysis is almost identical.

So what is wrong with `Gradient-Descent`? If you revisit analysis, what is the gradient $\nabla f(x)$ to begin with? It is an object that tells you the directional derivative $D_u f$ of the function f, *i.e.*, the rate of change of f along direction u. In particular, the gradient is a *linear* map $\nabla f(x) : \mathbb{R}^n \to \mathbb{R}$ that takes in a vector u and spits out the rate of change of f if you move towards u (at point x). Now if we go back to the definition of `Gradient-Descent`, things look peculiar. The algorithm doesn't even compile! The first term x_t lives in $\mathbb{R}^n$, but the second term $\nabla f(x_t)$ is a linear map that lives in a different world. Now the question is why `Gradient-Descent` works at all and when it may not work so well. To answer that, we need to introduce some basic functional analysis.

2 Basics of Banach Space

Let's first examine the setup of our problem once more. Recall that we are interested in optimizing a differentiable, convex function $f : \mathbb{R}^n \to \mathbb{R}$. Moreover, we implicitly assumed that it is a normed space $X = (\mathbb{R}^n, \|\cdot\|)$, and that the norm is ℓ_2 norm, so we have the identity $a^\top b = \frac{1}{2}\left(\|a\|^2 + \|b\|^2 - \|a-b\|^2\right)$ and so on. The Lipschitz condition states that $\|\nabla f(x)\| \le L$. This seems very problematic, since we just said $\nabla f(x)$ should live in another space (the space of linear maps), so shouldn't it be equipped with another norm? The answer is two-fold.

(i) The gradients live in the *dual space* X^* of X, equipped with a *dual norm* $\|X\|_*$; but

(ii) the ℓ_2 space is self-dual, so `Gradient-Descent` compiles for ℓ_2 spaces.

What are these terms anyway? It turns out that to really generalize `Gradient-Descent` outside ℓ_2 space (*i.e.*, Hilbert space), we will work with Banach space, *i.e.*, complete[1] normed vector space. We will assume that the vector space is finite-dimensional and over the reals, and denote a Banach space by $X = (\mathbb{R}^n, \|\cdot\|)$.

Definition 2.1 (Dual space)**.** Let $X = (\mathbb{R}^n, \|\cdot\|)$ be a Banach space, where the underlying field is the reals. The (continuous) dual space X_* is the space of (continuous) linear maps from X into $\mathbb{R}$.

This is precisely the world where the gradients live! But what is the norm of the dual space?

Definition 2.2 (Dual norm)**.** Let $X = (\mathbb{R}^n, \|\cdot\|)$ be a Banach space. The dual space is equipped with the dual norm $\|\psi\|_* = \sup_{x \in X} \{\psi(x) : \|x\| \le 1\}$.

This is an important notion in optimization, as it gives rise to the ℓ_p-ℓ_q-norm duality.[2] Formally, one can show (by Hölder's inequality) that ℓ_p norm is the dual of ℓ_q norm, where $1/p + 1/q = 1$. But this means that the dual of ℓ_2 space is … ℓ_2 itself! Therefore, as long as we always measure things by ℓ_2 norm, `Gradient-Descent` is a well-defined algorithm, and Theorem 1.1 holds.

On the other hand, as you will see in Section 4, sometimes there are motivations to solve problems that are intrinsically *not* ℓ_2. For these problems, our algorithms need to delve into the dual space, and this leads to mirror descent.

[1] if you don't recall its definition, it's completely OK. It just means that the space is complete—there is no hole in it!

[2] it is also connected to the notions of Fenchel duality and convex conjugates.

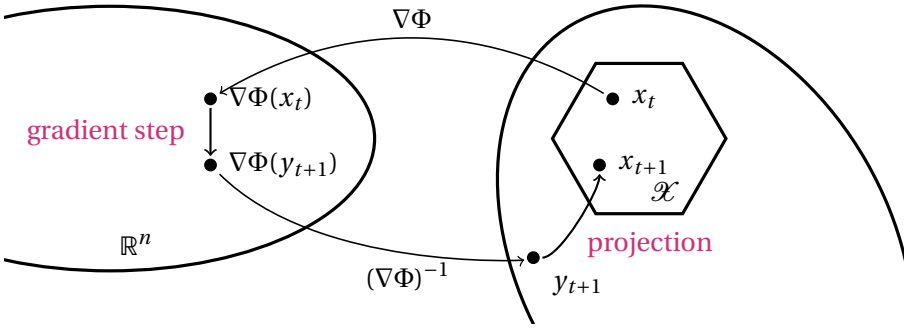

Figure 1: Illustration of mirror descent, in courtesy of [Bub15].

3 Mirror Descent

To appreciate mirror descent in its full generality, we will consider constrained convex optimization problem, and denote the convex feasible region by $\mathcal{X}$.

The idea of mirror descent is simple. See Figure 1 for an illustration. We will choose a strongly convex[3] function $\Phi : \mathbb{R}^n \to \mathbb{R}$, called *mirror map*, so that $\nabla\Phi$ maps from the primal space to the dual space. Then we map x_t to the dual space by $\nabla\Phi$ and perform gradient update there.

$$\nabla\Phi(y_{t+1}) = \nabla\Phi(x_t) - \eta\nabla f(x_t). \tag{gradient step}$$

Finally, we map the result back to the primal space using the inverse of Φ and project it into the feasible region.

$$x_{t+1} \in \Pi^{\Phi}_{\mathcal{X}}(y_{t+1}). \tag{projection}$$

The projection $\Pi^{\Phi}_{\mathcal{X}}$ is done with respect to the Bregman divergence associated with Φ:

$$\Pi^{\Phi}_{\mathcal{X}}(y) = \operatorname*{arg\,min}_{x\in\mathcal{X}} D_{\Phi}(x, y).$$

Here, the Bregman divergence associated with Φ is defined as

$$D_{\Phi}(x, y) = \Phi(x) - \Phi(y) - \nabla\Phi(y)^{\top}(x - y). \tag{Bregman divergence}$$

If you draw a picture of a convex function Φ, the Bregman divergence is the error of the linear approximation of x centered at y.[4] Therefore, one can think of Bregman divergence as a distance; as x, y become farther, the first-order approximation would be more crude.

Mirror descent achieves the same convergence rate as `Gradient-Descent` but works in much more general settings. The proof is not particularly interesting, so we omit it.

Theorem 3.1 (Convergence of mirror descent)**.** Let Φ be a mirror map α-strongly convex w.r.t. $\|\cdot\|$. Let $R^2 = \sup_{x\in\mathbb{R}^n} \Phi(x) - \Phi(x_0)$, and f be convex, differentiable and $\|\nabla f(x)\|_* \le L$. Then mirror descent with $\eta = \frac{R}{L}\sqrt{\frac{2\alpha}{t}}$ satisfies

$$f\left(\frac{1}{T}\sum_{i=0}^{T-1} x_i\right) - f(x^*) \le RL\sqrt{\frac{2}{\alpha T}}.$$

[3] We say that a function is α-strongly convex if $f(y) \ge f(x) + \nabla f(x)^{\top}(y - x) + \frac{\alpha}{2}\|y - x\|^2$.

[4] Check out http://mark.reid.name/blog/meet-the-bregman-divergences.html for an interactive figure.

Proximal view. In some sense, we have only presented the algorithm in a mathematical view. Algorithmically, one should understand the algorithm as the following. Let's rewrite it a little bit. In fact, this is often taken as the definition of mirror descent in the literature. Observe that

$$\begin{aligned}
x_{t+1} &= \arg\min_{x\in\mathscr{X}} D_\Phi(x, y_{t+1}) \\
&= \arg\min_{x\in\mathscr{X}} \Phi(x) - \nabla\Phi(y_{t+1})^\top x && \text{(by the definition of Bregman divergence)} \\
&= \arg\min_{x\in\mathscr{X}} \Phi(x) - \big(\nabla\Phi(x_t) - \eta\nabla f(x_t)\big)^\top x && \text{(by definition of the algorithm (gradient step))} \\
&= \arg\min_{x\in\mathscr{X}} \eta\nabla f(x_t)^\top x + D_\Phi(x, x_t) && \text{(by the definition of Bregman divergence) .}
\end{aligned}$$

This gives what's called the *proximal view* of mirror descent. The first term $\eta\nabla f(x_t)^\top x$ is a first-order approximation of the objective centered at x_t, and one should think of the Bregman divergence term as a regularizer that penalizes x_{t+1} being far from x_t. Thus, the method tries to minimize the local linearization of the objective, while not moving too far away from the previous point, with distances measured by the Bregman divergence of the mirror map.

Standard mirror maps. Note that the convergence of mirror descent (Theorem 3.1) depends on the choice of mirror map. We discuss two examples.

(i) Euclidean setup: $\Phi(x) = \frac{1}{2}\|x\|_2^2$. Clearly, Φ is 1-strongly convex w.r.t. ℓ_2 norm. It is not hard to show that $D_\Phi(x, y) = \|x - y\|_2^2$. Thus, under the proximal view, the mirror descent becomes

$$x_{t+1} = \arg\min_{x\in\mathscr{X}} \eta\nabla f(x_t)^\top x + \frac{1}{2}\|x - x_t\|_2^2. \tag{2}$$

Taking derivative and setting it to 0, we obtain exactly the gradient descent update rule $x_{t+1} = x_t - \eta\nabla f(x_t)$! So mirror descent generalizes gradient descent.

(ii) Simplex setup: $\Phi(x) = \sum_i x_i \log x_i$. It can be shown, by Pinsker's inequality, that the negative entropy function Φ is $\frac{1}{\ln 2}$-strongly convex w.r.t. the ℓ_1 norm. The Bregman divergence is given by

$$\begin{aligned}
D_\Phi(y, x) &= \sum_i y_i \log y_i - \sum_i x_i \log x_i - \sum_i (\log x_i + 1)(y_i - x_i) \\
&= \sum_i y_i \log \frac{y_i}{x_i} - \sum_i y_i + \sum_i x_i.
\end{aligned}$$

This is often called the *generalized KL-divergence*. This setup is often called the entropic regularization in online learning.

Moreover, if the feasible region is the probability simplex $\Delta = \{x \in \mathbb{R}^n : \sum_i x_i = 1, x_i \geq 0\}$, the mirror descent boils down to a multiplicative weights update scheme. To see that, first note that since $\nabla\Phi(x) = 1 + \log x$, where log is taken entrywise, the gradient step is

$$\log y_{t+1} = \log y_t - \eta\nabla f(x_t)$$

Thus, we take

$$y_{t+1}(i) \leftarrow y_t(i) e^{-\eta\nabla f(x_t)}. \tag{Multiplicative Weights}$$

What about the Bregman projection? Namely, how to find a point x_t in the simplex that minimizes the Bregman divergence to y_t? We claim that it is just a simple normalization w.r.t. ℓ_1 norm:

$$x_t \leftarrow \frac{y_t}{\|y_t\|_1}. \tag{ℓ_1 scaling}$$

Now we need to prove

$$\arg\min_{x\in\Delta} D_\Phi(x,y) = \frac{y_t}{\|y_t\|_1}. \tag{3}$$

Consider the Langrangian

$$\mathscr{L}(x,\lambda) = \sum_i x_i \log\frac{x_i}{y_i} + \sum_i (y_i - x_i) + \lambda\left(\sum_i x_i - 1\right). \tag{4}$$

To find the projection, for all i, we need

$$\frac{\partial}{\partial x_i}\mathscr{L} = \log\frac{x_i}{y_i} - 1 + \lambda = 0 \tag{5}$$

Hence, $x_i \propto y_i$, but $\sum_i x_i = 1$, so it must be the case that $x = y/\|y\|_1$.

4 Online Learning and Multiplicative Weights Update

So far we have seen what is wrong with `Gradient-Descent` only on a mathematical level. Technically, one can still implement it no matter what norm we have in mind. We are going to see soon that *algorithmically* `Gradient-Descent` can be broken as well, and this is precisely where one should use mirror descent instead. The example is *online linear optimization*.

We consider an online, iterative game, where the player gets to play a distribution $p_t \in \Delta_n$. Each time step t, the player receives a loss function $f_t : \Delta_n \to \mathbb{R}$ of the form $f_t(p) = \langle \ell_t, p\rangle$, where $\ell_t \in [-1,1]^n$. The goal is to achieve low *regret*; that is, the average loss is close to the loss incurred by the best strategy p^* in hindsight.

$$\text{regret}_T = \frac{1}{T}\sum_t f_t(p_t) - \min_{p\in\Delta_n}\sum_t f_t(p). \tag{6}$$

We analyze two algorithms, gradient descent and mirror descent.

(i) Projected gradient descent. In particular, we consider the online gradient descent algorithm, where the update rule is given by

$$x_{t+1} \leftarrow x_t - \eta_t \nabla f_t(x_t).$$

Then we project it back to the simplex; recall that we remarked that the projection does not hurt the convergence or require a different analysis. In particular, if one inspects the proof of Theorem 1.1, an analog of (1) still holds. By the same argument we can show

$$\sum_{t=1}^{T}\left(f_t(x_t) - f_t\left(x^*\right)\right) \le \frac{R^2}{2\eta_t} + \frac{\eta L^2 T}{2}, \tag{7}$$

so we have the same convergence, and the regret is given by

$$\text{regret}_T \le \frac{RL}{\sqrt{T}},$$

where f_t is L-Lipschitz and $\|x_t - x_0\| \le R$. However, note that since $\|\nabla f_t\|_\infty \le 1$, each loss function f_t is $\sqrt{n}$-Lipschitz w.r.t. the ℓ_2 norm. Thus, to obtain a ϵ-optimal point, we need $T = O(n/\epsilon^2)$ iterations. The dimension dependence looks really bad.

(ii) Mirror descent. Let's instantiate the algorithm with simplex setup. We apply Theorem 3.1. Note that $\|\nabla f_t\|_\infty \le 1$, so the Lipschitz constant is $L = 1$. Also, for the negative entropy Φ, we have $-\log n \le \Phi(x) \le 0$ for $x \in \Delta_n$, so $R^2 = \sup_{x\in\Delta_n}\Phi(x) - \Phi(x^*) = \log n$. Also, recall that Φ is $O(1)$-strongly convex w.r.t. ℓ_1 norm. By Theorem 3.1, we can obtain an ϵ-optimal solution only in $O(\log n/\epsilon^2)$ iterations, using mirror descent! Moreover, algorithmically, mirror descent with simplex setup under the simplex constraint is just multiplicative weights update, so it is fast and easy to implement.

References

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.