15-850: Advanced Algorithms **CMU, Fall 2018**
Lecture #19: Gradient descent & Mirror descent October 10, 2018
Lecturer: Anupam Gupta Scribe: Eliot Robson and Taisuke Yasuda

1 Introduction

Let us start by recalling the online gradient descent for optimizing convex functions. Remember the set up: given a fixed $\epsilon > 0$, we present at each time step t a vector x_t in a closed convex set $K \subseteq \mathbb{R}^n$, the adversary will then choose a function $f_t : K \to \mathbb{R}$ which is convex and smooth. We also assume f_t is G-Lipschitz with respect to $\|\cdot\|_2$, which means

$$\frac{\|f_t(x) - f_t(y)\|_2}{\|x-y\|_2} \leq G \text{ for all distinct } x, y \in K, \text{ or equivalently } \|\nabla f_t(x)\|_2 \leq G \text{ for all } x \in K.$$

We showed that for any $x^* \in K$, a slightly modified variant of the gradient descent algorithm, starting from a point $x_0 \in K$ with $\|x_0 - x^*\|_2 \leq D$, produces $x_1, \dots, x_T$ after T steps such that $x_i \in K$ for $i = 1, \dots, T$, and

$$\sum_{t=1}^{T} f_t(x_t) \leq \sum_{t=1}^{T} f_t(x^*) + \frac{\eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|_2^2}{2} + \frac{\|x^* - x_0\|^2}{2\eta}. \tag{19.1}$$

We may set $\eta = \frac{D}{G\sqrt{T}}$ to get

$$\sum_{t=1}^{T} f_t(x_t) \leq \sum_{t=1}^{T} f_t(x^*) + \frac{GD}{\sqrt{T}}. \tag{19.2}$$

Then, we can set $T = (\frac{GD}{\epsilon})^2$ and $\hat{x} = \frac{1}{T}\sum_{i=1}^{T} x_i$ to get

$$\begin{aligned} \sum_{t=1}^{T} f_t(\hat{x}) &\leq \sum_{t=1}^{T} f_t(x_t) && \text{(By convexity of } f_t) \\ &\leq \sum_{t=1}^{T} f_t(x^*) + \underbrace{\epsilon}_{\text{regret}} && \text{(By 19.2)} \end{aligned}$$

Notice that this gradient descent algorithm works for all convex functions over convex bodies, while the Multiplicative Weight (MW) algorithm only works for linear functions and over $\Delta_n = \{x \in \mathbb{R}^n_+ : \sum_{i=1}^{n} x_i = 1\}$, i.e. the simplex in $\mathbb{R}^n$. However, in the following example, we see that the gradient descent algorithm does significantly worse than the specialized Hedge algorithm.

Example 19.1. Suppose $f_t : \Delta_n \to \mathbb{R}$ and $f_t(x) = \langle \ell_t, x \rangle$, where $\ell_t \in [-1,1]^n$ for $t = 1, \dots, T$. Notice that for all $t = 1, \dots, T$, function f_t is $(\sqrt{n})$-Lipschitz, and for any $x_0 \in \Delta_n$ we have $\|x_0 - x^*\|_2 \leq \sqrt{2}$ for all $x^* \in \Delta_n$. Hence, applying the online gradient descent method for $T = (\frac{\sqrt{2}\sqrt{n}}{\epsilon})^2 = \frac{2n}{\epsilon^2}$ outputs a solution $\hat{x}$ with regret at most ϵ.

On the other hand, this problem is an MW problem. Hence, we can apply Hedge algorithm for $T = \frac{\ln n}{\epsilon}$ steps to get a regret of at most ϵ.

Therefore, gradient descent needs significantly more steps to be able to guarantee an ϵ regret compared to Hedge algorithm. Thus, we want a more generalized descent method which can adapt to the specific geometry of a problem. We first introduce a few definitions.

2 Norms and their Duals

In the previous section we described a gradient descent method which relied on the Euclidean norm $\|\cdot\|_2$. However, we can use different norm functions to adapt to the geometry of different problems. First we need to formally define a norm and its dual.

Definition 19.2. A function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is a *norm* if

1. If $\|x\| = 0$ for $x \in \mathbb{R}^n$, then $x = 0$;
2. for $\alpha \in \mathbb{R}$ and $x \in \mathbb{R}^n$ we have $\|\alpha x\| = |\alpha|\|x\|$; and
3. for $x, y \in \mathbb{R}^n$ we have $\|x + y\| \le \|x\| + \|y\|$.

Example 19.3. ℓ_p-norm for $p \in \mathbb{Z}_+$ is defined as $\|x\|_p = (\sum_{i=1}^n x_i^p)^{\frac{1}{p}}$ for $x \in \mathbb{R}^n$. Also ℓ_∞-norm is defined as $\|x\|_\infty = \max_{i=1,\ldots,n} x_i$ for $x \in \mathbb{R}^n$. See Figure 19.1 for further illustration.

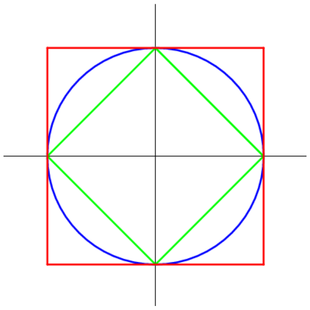

Figure 19.1: The unit ball in ℓ_1-norm (Green), ℓ_2-norm (Blue), and ℓ_∞-norm (Red).

Definition 19.4. Let $\|\cdot\|$ be a norm. Then the dual norm of $\|\cdot\|$ is a function $\|\cdot\|_*$ defined as

$$\|y\|_* = \sup\{\langle x, y\rangle \ : \ \|x\| \le 1\}.$$

Corollary 19.5. *For $x, y \in \mathbb{R}^n$, we have $\langle x, y\rangle \le \|x\|\|y\|_*$.*

Proof. Assume $\|x\| \neq 0$, otherwise both sides are 0. Since $\left\|\frac{x}{\|x\|}\right\| = 1$, we have $\left\langle \frac{x}{\|x\|}, y\right\rangle \le \|y\|_*$. □

Example 19.6. The dual norm of ℓ_2-norm is ℓ_2-norm. The dual norm of ℓ_1-norm is the ℓ_∞-norm.

Theorem 19.7. *The dual norm of ℓ_p-norm $\|\cdot\|_p$ is ℓ_q-norm $\|\cdot\|_q$, where $\frac{1}{p} + \frac{1}{q} = 1$.*

Theorem 19.8. *We have $(\|\cdot\|_*)_* = \|\cdot\|$, for $\|\cdot\|$ defined on a finite dimensional space.*

Using the notion of dual norms, we introduce the following definition of Lipschitz continuity for a norm $\|\cdot\|$.

Definition 19.9. Let f be a differentiable function. Then f is G-Lipschitz with respect to $\|\cdot\|$ if

$$\|\nabla f(x)\|_* \le G \text{ for all } x \in \mathbb{R}^n.$$

3 Online Mirror Descent

We now introduce the mirror descent algorithm introduced by Nemirovski and Yudin [NY78]. As discussed, in gradient descent, at each step we set $x_{t+1} = x_t - \eta \nabla f_t(x_t)$. However, recall from multivariable calculus that the gradient is actually defined as a linear map from $\mathbb{R}^n$ to $\mathbb{R}$ and hence most naturally belongs to the dual space of $\mathbb{R}^n$. In the gradient descent method, we work in $\mathbb{R}^n$ endowed with ℓ_2-norm, and this normed space is in fact self-dual as in Example 19.6, so mirror descent on this space is just gradient descent. However, Example 19.1 suggests that ℓ_2-norm might not always be the "right" norm for the primal space.

Since ∇f_t is a function in the dual space, $-\eta \nabla f_t(x_t)$ is a step in the dual space. Hence, we need to map our current point x_t to a point in the dual space, namely θ_t. After taking the gradient step, $\theta_{t+1} = \theta_t - \eta \nabla f_t(x_t)$ we still have to map θ_{t+1} back to a point in the primal space x'_{t+1}. Similar to gradient descent, x'_{t+1} might not be in the closed convex feasible region K, so we need to project x'_{t+1} back to a "close" x_{t+1} in K. This is akin to tuning our descent method to the geometry of the problem (See Figure 19.2).

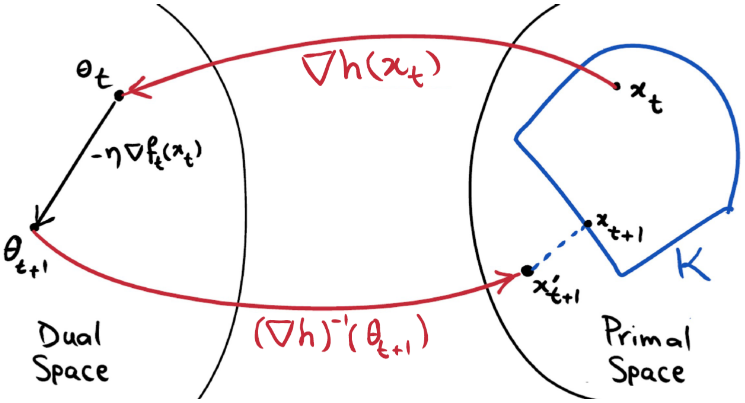

Figure 19.2: The four basic steps in each iteration of the mirror descent algorithm

To justify the appellation of the algorithm, notice that the dual space acts as a mirror to the primal space. That is why we call the functions that map x_t to θ_t and θ_{t+1} to x'_{t+1} the *mirror maps*. To find a suitable mirror map, we need to define α-strongly convex function with respect to a norm $\|\cdot\|$.

3.1 Convex Analysis Preliminaries

Definition 19.10. Convex and differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is α-strongly convex with respect to $\|\cdot\|$ if

$$h(y) \geq h(x) + \langle \nabla h(x), y - x \rangle + \frac{\alpha}{2}\|y - x\|^2.$$

Example 19.11. Function $h_1 : \mathbb{R}^n \to \mathbb{R}$ defined as $h_1(x) = \frac{1}{2}\|x\|_2^2$ is 1-strongly convex with respect to $\|\cdot\|_2$.

Example 19.12. Function $h_2 : \mathbb{R}^n \to \mathbb{R}$ defined as $h_2(x) = \sum_{i=1}^n x_i \log x_i$ is $\frac{1}{\ln 2}$-strongly convex with respect to $\|\cdot\|_1$. Function h_2 is the negative entropy function.

Let $h : \mathbb{R}^n \to \mathbb{R}$ be an α-strongly-convex function with respect to $\|\cdot\|$. Then, we will use $\nabla(h) : \mathbb{R}^n \to \mathbb{R}^n$ as our mirror map. Thus, we will set $\theta_t = \nabla h(x_t)$, and $x'_{t+1} = (\nabla h)^{-1}(\theta_{t+1})$. A function h used in this way is called a "distance generating" function. See Figure 19.2.

Example 19.13. Recall function h_1 is Example 19.11. We have $\nabla h_1(x) = x$, and $(\nabla h_1)^{-1}(\theta) = \theta$.

Example 19.13 gives a nice intuition why the gradient descent algorithm works within the primal and dual space unnoticed.

Example 19.14. Consider function h_2 in Example 19.12. We have $\nabla h_2(x)_i = (\ln x_i + 1)_i$, and $(\nabla h_2)^{-1}(\theta)_i = (e^{\theta_i - 1})_i$.

3.2 The Algorithm

Using these notions, we now present an outline of the algorithm, which works very similar to gradient descent. We let $K \subseteq \mathbb{R}^n$ be the feasible region of the primal space and $\nabla h : \mathbb{R}^n \to \mathbb{R}^n$ be a suitable mirror map. At each step of the algorithm, we do the following:

(i) Map x_t from the primal space to θ_t in the dual space, $\nabla h(x_t) = \theta_t$.

(ii) Take a gradient step in the dual space, $\theta_{t+1} = \theta_t - \eta_t \cdot \nabla f_t(x_t)$.

(iii) Map θ_{t+1} from the dual space back to x_{t+1} in the primal space, $x_{t+1} = (\nabla h)^{-1}(\theta_{t+1})$.

(iv) Since it could be that $x_{t+1} \notin K$, project x_{t+1} back into the feasible region.

To compute the last step, we must apply Bregman projection, which we will describe shortly. For now, we see how mirror descent agrees with previous algorithms we have used. As mentioned before, the mirror descent algorithm becomes gradient descent when we are working in $\mathbb{R}^n$ normed with $\|\cdot\|_2$, treating ∇h_1 as the mirror map.

In $\mathbb{R}^n$ normed with $\|\cdot\|_1$ and mirror map ∇h_2, the algorithm behaves similar to Hedge. For simplicity, we refer to $x_t, x'_{t+1}, x_{t+1}, \theta_t$, and θ_{t+1} by x, x', x^+, θ, and θ^+, respectively. In this case, mirror descent becomes:

(i) Start with x and compute $\theta = (\ln x_i + 1)_i$, i.e. map x to θ using the mirror map ∇h_2 to the dual space.

(ii) Set $\theta^+ = (\theta - \eta \nabla f_t(x)) = (\ln x_i + 1 - \eta \nabla f_t(x)_i)_i$, i.e. take the gradient step in the dual space.

(iii) Find $x' = (e^{\ln \theta_i^+ - 1})_i = (e^{\ln x_i - \eta(\nabla f_t(x))_i})_i = (x_i \cdot e^{-\eta(\nabla f_t(x))_i})_i$, i.e. map θ^+ back to the primal space.

Remember Example 19.1 where $f_t(x) = \langle \ell_t, x \rangle$, in this case $\nabla f_t = \ell_t$, so the mirror descent algorithm finds $x' = (x_i e^{-\eta \ell_i})_i$, which is similar to Hedge algorithm.

There is still one missing step in the algorithm:

(iv) Project x' back to point x^+ in the feasible region K.

In order to do this, we need to define Bregman distance.

Definition 19.15. The *Bregman distance* of x and y with respect to function h is defined as

$$D_h(y \| x) := h(y) - h(x) - \langle \nabla h(x), y - x \rangle.$$

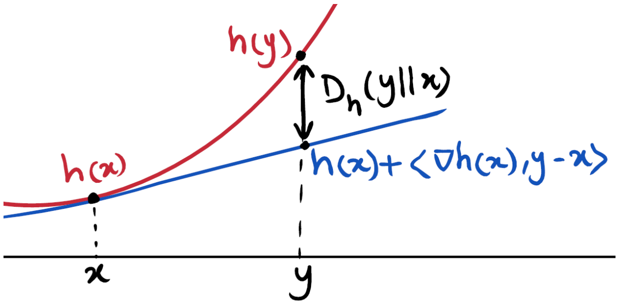

Figure 19.3: $D_h(y\|x)$ for function $h : \mathbb{R} \to \mathbb{R}$.

Figure 19.3 describes the Bregman distance geometrically for $h : \mathbb{R} \to \mathbb{R}$.

As promised, we can now define the notation of Bregman projection.

Definition 19.16. The Bregman projection of point x' onto convex set K is

$$x^+ = \arg\min_{x \in K} D_h(x\|x').$$

Example 19.17. Consider function $h_1(x) = \frac{1}{2}\|x\|_2^2$ from Example 19.11. Then

$$\begin{aligned} D_{h_1}(y\|x) &= \frac{1}{2}\|y\|_2^2 - \frac{1}{2}\|x\|_2^2 - \langle x, y - x\rangle \\ &= \frac{1}{2}\|y\|_2^2 + \frac{1}{2}\|x\|_2^2 - \langle x, y\rangle \\ &= \frac{1}{2}\|y - x\|_2^2. \end{aligned}$$

Therefore, when we apply the mirror descent algorithm with ℓ_2-norm and mirror function h_1, the projection step is the same as the projection step in gradient descent. This is because for h_1, Bregman distance is basically the Euclidean distance.

Example 19.18. For function $h_2(x) = \sum_{i=1}^n x_i \ln x_i$ from Example 19.12, we have

$$\begin{aligned} D_{h_2}(y\|x) &= \sum_{i=1}^n y_i \ln y_i - \sum_{i=1}^n x_i \ln x_i - \sum_{i=1}^n (\ln x_i + 1)(y_i - x_i) \\ &= -\sum_{i=1}^n y_i + \sum_{i=1}^n x_i + \underbrace{\sum_{i=1}^n y_i \ln \frac{y_i}{x_i}}_{KL(y\|x)}, \end{aligned}$$

$KL(y\|x)$ is known as the Kullback-Leibler divergence. For a related aside, see Lemma 19.19. Now in the case of ℓ_1-norm with mirror map h_2, step (iv) is

(iv) $x^+ = \left(\frac{x_i' e^{\eta \ell_i}}{\sum_{j=1}^n x_j' e^{-\eta \ell_j}}\right)_i$, i.e. take Bregman projection of x' onto the feasible region (the unit simplex Δ_n) with respect to Bregman distance D_{h_2}.

We observe that Bregman projection in this case is simply a rescaling, which again agrees with the Hedge algorithm.

Below, we see one final lemma about the Bregman distance.

Lemma 19.19. *For $h_2(x) = \sum_{i=1}^n x_i \ln x_i$, we have that*

$$\arg\min_{b \in \Delta_n} D_{h_2}(b \| a) = \frac{a}{\|a\|_1}.$$

Proof.

$$\arg\min_{b \in \Delta_n} D_{h_2}(b \| a) = \arg\min_{b \in \Delta_n} \left\{ \sum_i b_i \ln \frac{b_i}{a_i} - \sum_i b_i + \sum_i a_i \right\}.$$

Using Lagrange multiplier λ, we have that

$$F(b, \lambda) = \sum_i b_i \ln \frac{b_i}{a_i} - \sum_i b_i + \sum_i a_i + \lambda \left(\sum_i b_i - 1 \right).$$

Then, if we set $\frac{\partial F}{\partial b_i} = 0$, then we obtain

$$\begin{aligned} \ln \frac{b_i}{a_i} + 1 - 1 + \lambda = 0 \\ \implies b_i = a_i \cdot e^{-\lambda} \\ \implies 1 = \sum_i b_i = \sum_i a_i \cdot e^{-\lambda} \\ \implies \lambda = \ln \sum_i a_i \\ \implies b_i = a_i \cdot e^{-\ln \sum_i a_i} = \frac{a_i}{\|a_i\|_1}. \end{aligned}$$

□

4 Analysis

We prove the following theorem.

Theorem 19.20. *Let $\|\cdot\|$ be a norm function, $f_1, \ldots, f_T$ be convex and differentiable functions such that $\|\nabla f_t\|_* \leq G$, and h be an α-strongly convex function with respect to $\|\cdot\|$, then the mirror descent algorithm starting with x_0 and taking constant step size η in every iteration, produces $x_1, \ldots, x_T$ such that*

$$\sum_{t=1}^{T} f_t(x_t) \leq \sum_{t=1}^{n} f_t(x^*) + \frac{D_h(x^* \| x_0)}{\eta} + \frac{\eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|_*^2}{2\alpha} \quad \text{, for all } x^* \tag{19.3}$$

Before proving Theorem 19.20, let us take a look at Inequality 19.3 in the two cases we discussed at length in the previous section.

If $\|\cdot\|$ is ℓ_2-norm and h is function h_1 from Example 19.11, then Inequality 19.3 becomes

$$\sum_{t=1}^{T} f_t(x_t) \leq \sum_{t=1}^{n} f_t(x^*) + \frac{\|x^* - x_0\|_2^2}{2\eta} + \frac{\eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|_2^2}{2} \quad \text{, for all } x^*,$$

which is Inequality 19.1.

If $\|\cdot\|$ is ℓ_1-norm and h is function h_2 from Example 19.12, then Inequality 19.3 becomes

$$\sum_{t=1}^{T}\langle \ell_t, x_t\rangle \leq \sum_{t=1}^{T}\langle \ell_t, x^*\rangle + \frac{\sum_{i=1}^{n} x_i^* \ln \frac{x_i^*}{(x_0)_i}}{\eta} + \frac{\eta\sum_{t=1}^{T}\|\ell_t\|_\infty^2}{2/\ln 2} \quad \text{, for all } x^* \in \Delta_n.$$

Since $\|\ell_t\|_\infty \leq 1$, we have

$$\sum_{t=1}^{T}\langle \ell_t, x_t\rangle \leq \sum_{t=1}^{T}\langle \ell_t, x^*\rangle + \frac{KL(x^*\|x_0)}{\eta} + \frac{\eta T}{2/\ln 2} \quad \text{, for all } x^* \in \Delta_n.$$

Proof of Theorem 19.20. Define potential $\Phi_t = \frac{D_h(x^*\|x_t)}{\eta}$. The amortized cost at time t is

$$f_t(x_t) - f_t(x^*) + (\Phi_{t+1} - \Phi_t). \tag{19.4}$$

Now

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= \frac{1}{\eta}\big(D_h(x^*\|x_{t+1}) - D_h(x^*\|x_t)\big)\\
&= \frac{1}{\eta}\big(h(x^*) - h(x_{t+1}) - \langle \underbrace{\nabla h(x_{t+1})}_{\theta_{t+1}}, x^* - x_{t+1}\rangle - h(x^*) + h(x_t) + \langle \underbrace{\nabla h(x_t)}_{\theta_t}, x^* - x_t\rangle\big)\\
&= \frac{1}{\eta}\big(h(x_t) - h(x_{t+1}) - \langle \theta_t - \eta \underbrace{\nabla f_t(x_t)}_{\nabla_t}, x^* - x_{t+1}\rangle + \langle \theta_t, x^* - x_t\rangle\big)\\
&= \frac{1}{\eta}\big(h(x_t) - h(x_{t+1}) - \langle \theta_t, x_t - x_{t+1}\rangle + \eta\langle \nabla f_t(x_t), x^* - x_{t+1}\rangle\big)\\
&\leq \frac{1}{\eta}\left(\frac{\alpha}{2}\|x_{t+1} - x_t\|^2 + \eta\langle \nabla f_t(x_t), x^* - x_{t+1}\rangle\right) \qquad \text{(By } \alpha\text{-strong convexity of } h \text{ wrt to } \|\cdot\|)
\end{aligned}$$

Plug this back to 19.4

$$\begin{aligned}
f_t(x_t) - f_t(x^*) + (\Phi_{t+1} - \Phi_t) &\leq f_t(x_t) - f_t(x^*) + \frac{\alpha}{2\eta}\|x_{t+1} - x_t\|^2 + \langle \nabla f_t(x_t), x^* - x_{t+1}\rangle\\
&\leq \underbrace{f_t(x_t) - f_t(x^*) + \big\langle \nabla f_t(x_t), x^* - x_t\big\rangle}_{\leq 0 \text{ by convexity of } f_t} + \frac{\alpha}{2\eta}\|x_{t+1} - x_t\|^2 + \langle \nabla f_t(x_t), x_t - x_{t+1}\rangle\\
&\leq \frac{\alpha}{2\eta}\|x_{t+1} - x_t\|^2 + \|\nabla f_t(x_t)\|_*\|x_t - x_{t+1}\| \qquad \text{(By Corollary 19.5)}\\
&\leq \frac{\alpha}{2\eta}\|x_{t+1} - x_t\|^2 + \frac{1}{2}\left(\frac{\eta}{\alpha}\|\nabla f_t(x_t)\|_*^2 + \frac{\alpha}{\eta}\|x_t - x_{t+1}\|^2\right) \qquad \text{(By AM-GM)}\\
&\leq \frac{\eta}{2\alpha}\|\nabla f_t(x_t)\|_*^2.
\end{aligned}$$

Thus,

$$\begin{aligned}
\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) &\leq \Phi_0 - \Phi_{T+1} + \sum_{t=1}^{T}\frac{\eta}{2\alpha}\|\nabla f_t(x_t)\|_*^2\\
&\leq \Phi_0 + \sum_{t=1}^{T}\frac{\eta}{2\alpha}\|\nabla f_t(x_t)\|_*^2\\
&\leq \frac{D_h(x^*\|x_0)}{\eta} + \frac{\eta\sum_{t=1}^{T}\|\nabla f_t(x_t)\|_*^2}{2\alpha}.
\end{aligned}$$

□

There are two things to notice about this proof. The first is that it is very similar to the original proof of gradient descent, just using more general language. The second is that, in the constrained case, we require that $D_h(x^* \| x_{t+1}) \leq D_h(x^* \| x'_{t+1})$ for $x^* \in K$, $x'_{t+1} \notin K$ ("Extended Pythagoras Theorem" for Bregman distance).

5 Alternative Views of Mirror Descent

In this lecture, we reviewed mirror descent algorithm as a gradient descent scheme where we do the gradient step in the dual space. We now provide some alternative views of mirror descent.

5.1 As Preconditioned Gradient Descent

For any given space which we use a descent method on, we can linearly transform the space with some map Q to make the geometry more regular. This technique is known as preconditioning, and improves the speed of the descent. Using the linear transformation Q, our descent rule becomes

$$x_{t+1} = x_t - \eta \left[\nabla^2 f(x_t)\right]^{-1} \nabla f(x_t).$$

More on this view later.

5.2 As Proximal Gradient Descent

A shorter (but less intuitive) description of mirror descent in the following.

Algorithm 1 Proximal Gradient Descent Algorithm

```
for $t \leftarrow 0$ to $T - 1$ do
    $x_{t+1} \leftarrow \arg\min_{x \in K} \{\eta \langle \nabla f_t(x_t), x \rangle + D_h(x \| x_t)\}$
end for
```

Essentially, the D_h term serves as a regularization term in the descent from the inner product, so if we move too far this term rises sharply. Again, if we let $D_h(x \| x_t) = \frac{1}{2}\|x - x_t\|_2^2$, we can minimize the term in the algorithm by setting the gradient to 0. Thus, we obtain

$$\begin{aligned} \eta \cdot \nabla f_t(x_t) + (x - x_t) &= 0 \\ \implies x = x_t - \eta \cdot \nabla f_t(x_t)&, \end{aligned}$$

again matching the normal gradient descent algorithm.

Acknowledgments

These lecture notes were scribed by Eliot Robson and Taisuke Yasuda, based on previous scribe notes of Arash Haddadan.

References

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, November 2015.

[NY78] Arkadi Nemirovski and D. Yudin. On cesaros convergence of the gradient descent method for finding saddle points of convex-concave functions. *Daklady Akademii Nauk SSSR*, 239(4):291–307, 1978. 3