# 18
# Mirror Descent

The gradient descent algorithm of the previous chapter is general and powerful: it allows us to (approximately) minimize convex functions over convex bodies. Moreover, it also works in the model of online convex optimization, where the convex function can vary over time, and we want to find a low-regret strategy—one which performs well against every fixed point $x^*$.

This power and broad applicability means the algorithm is not always the best for specific classes of functions and bodies: for instance, for minimizing linear functions over the probability simplex $\Delta_n$, we saw in §17.4.1 that the generic gradient descent algorithm does significantly worse than the specialized Hedge algorithm. This suggests asking: *can we somehow change gradient descent to adapt to the "geometry" of the problem?*

The *mirror descent* framework of this section allows us to do precisely this. There are many different (and essentially equivalent) ways to explain this framework, each with its positives. We present two of them here: the *proximal point* view, and the *mirror map* view, and only mention the others (the *preconditioned or quasi-Newton gradient flow* view, and the *follow the regularized leader* view) in passing.

## 18.1 Mirror Descent: the Proximal Point View

Here is a different way to arrive at the gradient descent algorithm from the last lecture: Indeed, we can get an expression for $x_{t+1}$ by

**Algorithm 16:** Proximal Gradient Descent Algorithm

```
x_1 ← starting point
for t ← 1 to T do
    x_{t+1} ← arg min_x { η⟨∇f_t(x_t), x⟩ + (1/2)‖x − x_t‖² }
```

setting the gradient of the function to zero; this gives us the expression

$$\eta \cdot \nabla f_t(x_t) + (x_{t+1} - x_t) = 0 \quad \Longrightarrow \quad x_{t+1} = x_t - \eta \cdot \nabla f_t(x_t),$$

which matches the normal gradient descent algorithm. Moreover, the intuition for this algorithm also makes sense: if we want to minimize the function $f_t(x)$, we could try to minimize its linear approximation $f_t(x_t) + \langle \nabla f_t(x_t), x - x_t \rangle$ instead. But we should be careful not to "over-fit": this linear approximation is good only close to the point $x_t$, so we could add in a penalty function (a "regularizer") to prevent us from straying too far from the point $x_t$. This means we should minimize

$$x_{t+1} \leftarrow \arg\min_x \{ f_t(x_t) + \langle \nabla f_t(x_t), x - x_t \rangle \;+\; \frac{1}{2}\|x - x_t\|^2 \}$$

or dropping the terms that don't depend on $x$,

$$x_{t+1} \leftarrow \arg\min_x \{ \langle \nabla f_t(x_t), x \rangle \;+\; \frac{1}{2}\|x - x_t\|^2 \} \tag{18.1}$$

If we have a constrained problem, we can change the update step to:

$$x_{t+1} \leftarrow \arg\min_{x \in K} \{ \eta \langle \nabla f_t(x_t), x \rangle \;+\; \frac{1}{2}\|x - x_t\|^2 \} \tag{18.2}$$

The optimality conditions are a bit more complicated now, but they again can show this algorithm is equivalent to projected gradient descent from the previous chapter.

Given this perspective, we can now replace the squared Euclidean norm by other distances to get different algorithms. A particularly useful class of distance functions are *Bregman divergences*, which we now define and use.

### *18.1.1 Bregman Divergences*

Given a *strictly* convex function $h$, we can define a distance based on how the function differs from its linear approximation:

**Definition 18.1.** The *Bregman divergence* from $x$ to $y$ with respect to function $h$ is

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

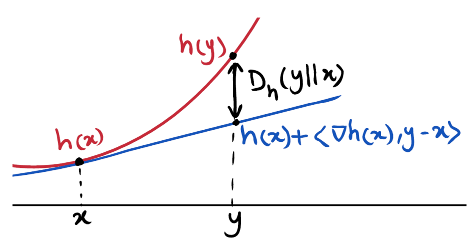


Figure 18.1: $D_h(y\|x)$ for function $h : \mathbb{R} \to \mathbb{R}$.

The figure on the right illustrates this definition geometrically for a univariate function $h : \mathbb{R} \to \mathbb{R}$. Here are a few examples:

1. For the function $h(x) = \frac{1}{2}\|x\|^2$ from $\mathbb{R}^n$ to $\mathbb{R}$, the associated Bregman divergence is

   $$D_h(y\|x) = \tfrac{1}{2}\|y - x\|^2,$$

   the squared Euclidean distance.

2. For the (un-normalized) negative entropy function $h(x) = \sum_{i=1}^{n}(x_i \ln x_i - x_i)$,
$$D_h(y\|x) = \textstyle\sum_i \left(y_i \ln \frac{y_i}{x_i} - y_i + x_i\right).$$
Using that $\sum_i y_i = \sum_i x_i = 1$ for $y, x \in \Delta_n$ gives us $D_h(y\|x) = \sum_i y_i \ln \frac{y_i}{x_i}$ for $x, y \in \Delta_n$: this is the *Kullback-Leibler (KL) divergence* between probability distributions.

Many other interesting Bregman divergences can be defined.

### 18.1.2 *Changing the Distance Function*

Since the distance function $\frac{1}{2}\|x - y\|^2$ in (18.1) is a Bregman divergence, what if we replace it by a generic Bregman divergence: what algorithm do we get in that case? Again, let us first consider the unconstrained problem, with the update:

$$x_{t+1} \leftarrow \arg\min_x \{\eta \langle \nabla f_t(x_t), x\rangle \; + \; D_h(x\|x_t)\}.$$

Again, setting the gradient at $x_{t+1}$ to zero (i.e., the optimality condition for $x_{t+1}$) now gives:

$$\eta \nabla f_t(x_t) + \nabla h(x_{t+1}) - \nabla h(x_t) = 0,$$

or, rephrasing

$$\nabla h(x_{t+1}) = \nabla h(x_t) - \eta \nabla f_t(x_t) \tag{18.3}$$
$$\implies x_{t+1} = \nabla h^{-1}\big(\nabla h(x_t) - \eta \nabla f_t(x_t)\big) \tag{18.4}$$

Let's consider this for our two running examples:

1. When $h(x) = \frac{1}{2}\|x\|^2$, the gradient $\nabla h(x) = x$. So we get
$$x_{t+1} = x_t - \eta \nabla f_t(x_t),$$
the standard gradient descent update.

2. When $h(x) = \sum_i (x_i \ln x_i - x_i)$, then $\nabla h(x) = (\ln x_1, \dots, \ln x_n)$, so
$$(x_{t+1})_i = e^{\ln(x_t)_i - \eta \nabla f_t(x_t)} = (x_t)_i \, e^{-\eta \nabla f_t(x_t)}.$$
Now if $f_t(x) = \langle \ell_t, x\rangle$, its gradient is just the vector $\ell_t$, and we get back precisely the *weights* maintained by the Hedge algorithm!

The same ideas also hold for constrained convex minimization: we now have to search for the minimizer within the set $K$. In this case the algorithm using negative entropy results in the same Hedge-like update, followed by scaling the point down to get a probability vector, thereby giving the *probability values* in Hedge.

To summarize: this algorithm that tries to minimize the linear ap-

What is the "right" choice of $h$ to minimize the function $f$? A little thought shows that $h$ should equal $f$, because adding $D_f(x\|x_t)$ to the linear approximation of $f$ at $x_t$ gives us back exactly $f$. Of course, the update now requires us to minimize $f(x)$, which is the original problem. So we should choose an $h$ that is somehow "similar" to $f$, and yet such that the update step is tractable.

**Algorithm 17:** Proximal Gradient Descent Algorithm

$x_1 \leftarrow$ starting point
**for** $t \leftarrow 1$ ***to*** $T$ **do**
  $x_{t+1} \leftarrow \arg\min_{x \in K}\{\eta\langle\nabla f_t(x_t), x\rangle \ + \ D_h(x\|x_t)\}$

proximation of the function, regularized by a Bregman distance $D_h$, gives us vanilla gradient descent for one choice of $h$ (which is good for quadratic-like functions over Euclidean space), and Hedge for another choice of $h$ (which is good for linear functions over the space of probability distributions). Indeed, depending on how we choose the function, we can get different properties from this algorithm—this is the *mirror descent framework*.

## 18.2 *Mirror Descent: The Mirror Map View*

A different view of the mirror descent framework is the one originally presented by Nemirovski and Yudin. They observe that in gradient descent, at each step we set $x_{t+1} = x_t - \eta\nabla f_t(x_t)$. However, the gradient was actually defined as a linear functional on $\mathbb{R}^n$ and hence naturally belongs to the dual space of $\mathbb{R}^n$. The fact that we represent this functional (i.e., this *covector*) as a *vector* is a matter of convenience, and we should exercise care.

A *linear functional* on vector space $X$ is a linear map from $X$ into its underlying field $\mathbb{F}$.

In the vanilla gradient descent method, we were working in $\mathbb{R}^n$ endowed with $\ell_2$-norm, and this normed space is self-dual, so it is perhaps reasonable to combine points in the primal space (the iterates $x_t$ of our algorithm) with objects in the dual space (the gradients). But when working with other normed spaces, adding a covector $\nabla f_t(x_t)$ to a vector $x_t$ might not be the right thing to do. Instead, Nemirovski and Yudin propose the following:

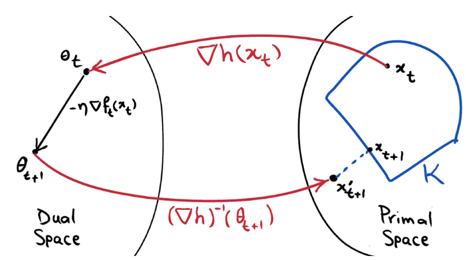


Figure 18.2: The four basic steps in each iteration of the mirror descent algorithm

1. we map our current point $x_t$ to a point $\theta_t$ in the dual space using a *mirror map*.

2. Next, we take the gradient step
$$\theta_{t+1} \leftarrow \theta_t - \eta\nabla f_t(x_t). \tag{18.5}$$

3. We map $\theta_{t+1}$ back to a point in the primal space $x'_{t+1}$ using the inverse of the mirror map from Step 1.

4. If we are in the constrained case, this point $x'_{t+1}$ might not be in the convex feasible region $K$, so we to project $x'_{t+1}$ back to a "close-by" $x_{t+1}$ in $K$.

The name of the process comes from thinking of the *dual space as being a mirror image of the primal space*. But how do we choose these mirror maps? Again, this comes down to understanding the geometry of the problem, the kinds of functions and the set $K$ we care about, and the kinds of guarantees we want. In order to discuss these, let us discuss the notion of norms in some more detail.

### *18.2.1 Norms and their Duals*

**Definition 18.2** (Norm)**.** A function $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ is a *norm* if

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

The well-known $\ell_p$-norms for $p \geq 1$ are defined by

$$\|x\|_p := (\sum_{i=1}^{n} |x_i|^p)^{1/p}$$

for $x \in \mathbb{R}^n$. The $\ell_\infty$-norm is given by

$$\|x\|_\infty := \max_{i=1}^{n} |x_i|$$

for $x \in \mathbb{R}^n$.

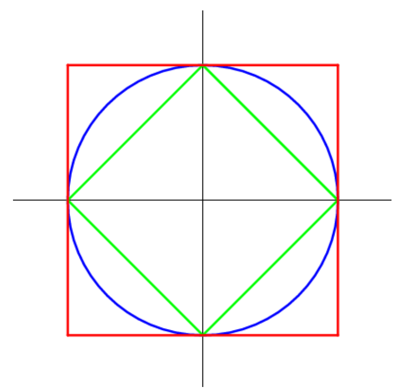

Figure 18.3: The unit ball in $\ell_1$-norm (Green), $\ell_2$-norm (Blue), and $\ell_\infty$-norm (Red).

**Definition 18.3** (Dual Norm)**.** Let $\|\cdot\|$ be a norm. The dual norm of $\|\cdot\|$ is a function $\|\cdot\|_*$ defined as

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

The dual norm of the $\ell_2$-norm is again the $\ell_2$-norm; the Euclidean norm is self-dual. The dual for the $\ell_p$-norm is the $\ell_q$-norm, where $1/p + 1/q = 1$.

**Corollary 18.4** (Cauchy-Schwarz for General Norms)**.** *For $x, y \in \mathbb{R}^n$, we have $\langle x, y\rangle \leq \|x\| \, \|y\|_*$.*

*Proof.* Assume $\|x\| \neq 0$, otherwise both sides are 0. Since $\|x/\|x\|\| = 1$, we have $\langle x/\|x\|, y\rangle \leq \|y\|_*$. □

**Theorem 18.5.** *For a finite-dimensional space with norm $\|\cdot\|$, we have $(\|\cdot\|_*)_* = \|\cdot\|$.*

Using the notion of dual norms, we can give an alternative characterization of Lipschitz continuity for a norm $\|\cdot\|$, much like Fact 17.6 for Euclidean norms:

*Fact* 18.6. For $f$ be a differentiable function. Then $f$ is $G$-Lipschitz with respect to norm $\|\cdot\|$ if and only if for all $x \in \mathbb{R}$,

$$\|\nabla f(x)\|_* \leq G.$$

### 18.2.2 Defining the Mirror Maps

To define a mirror map, we first fix a norm $\|\cdot\|$, and then choose a differentiable convex function $h : \mathbb{R}^n \to \mathbb{R}$ that is $\alpha$-strongly-convex with respect to this norm. Recall from §17.5.1 that such a function must satisfy

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

We use two familiar examples:

1. $h(x) = \frac{1}{2}\|x\|_2^2$ is 1-strongly convex with respect to $\|\cdot\|_2$, and
2. $h(x) := \sum_{i=1}^n x_i(\log x_i - 1)$ is 1-strongly convex with respect to $\|\cdot\|_1$; the proof of this is called *Pinsker's inequality*.

Check out the two proofs pointed to by Aryeh Kontorovich, or this proof (part 1, part 2) by Madhur Tulsiani.

Having fixed $\|\cdot\|$ and $h$, the *mirror map* is

$$\nabla(h) : \mathbb{R}^n \to \mathbb{R}^n.$$

Since $h$ is differentiable and strongly-convex, we can define the inverse map as well. This defines the mappings that we use in the Nemirovski-Yudin process: we set

The function $h$ used in this way is often called a *distance-generating* function.

$$\theta_t = \nabla h(x_t) \qquad \text{and} \qquad x'_{t+1} = (\nabla h)^{-1}(\theta_{t+1}).$$

For our first running example of $h(x) = \frac{1}{2}\|x\|^2$, the gradient (and hence its inverse) is the identity map. For the (un-normalized) negative entropy example, $(\nabla h(x))_i = \ln x_i$, and hence $(\nabla h)^{-1}(\theta)_i = e^{\theta_i}$.

### 18.2.3 The Algorithm (Again)

Let us formally state the algorithm again, before we state and prove a theorem about it. Suppose we want to minimize a convex function $f$ over a convex body $K \subseteq \mathbb{R}^n$. We first fix a norm $\|\cdot\|$ on $\mathbb{R}^n$ and choose a distance-generating function $h : \mathbb{R}^n \to \mathbb{R}$, which gives the mirror map $\nabla h : \mathbb{R}^n \to \mathbb{R}^n$. In each iteration of the algorithm, we do the following:

(i) Map to the dual space $\theta_t \leftarrow \nabla h(x_t)$.

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

(iii) Map $\theta_{t+1}$ back to the primal space $x'_{t+1} \leftarrow (\nabla h)^{-1}(\theta_{t+1})$.

(iv) Project $x'_{t+1}$ back into the feasible region $K$ by using the Bregman divergence: $x_{t+1} \leftarrow \min_{x \in K} D_h(x \| x'_{t+1})$. In case $x'_{t+1} \in K$, e.g., in the unconstrained case, we get $x_{t+1} = x'_{t+1}$.

Note that the choice of $h$ affects almost every step of this algorithm.

## 18.3 The Analysis

We prove the following guarantee for mirror descent, which captures the guarantees for both Hedge and gradient descent, and for other variants that you may use.

The theorem is stated for the unconstrained version, but extending it to the constrained version is an easy exercise.

**Theorem 18.7** (Mirror Descent Regret Bound). *Let $\|\cdot\|$ be a norm on $\mathbb{R}^n$, and $h$ be an $\alpha$-strongly convex function with respect to $\|\cdot\|$. Given $f_1, \dots, f_T$ be convex, differentiable functions such that $\|\nabla f_t\|_* \leq G$, the mirror descent algorithm starting with $x_0$ and taking constant step size $\eta$ in every iteration produces $x_1, \dots, x_T$ such that for any $x^* \in \mathbb{R}^n$,*

$$\sum_{t=1}^{T} f_t(x_t) \leq \sum_{t=1}^{T} f_t(x^*) + \underbrace{\frac{D_h(x^* \| x_1)}{\eta} + \frac{\eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|_*^2}{2\alpha}}_{regret}. \tag{18.6}$$

Before proving Theorem 18.7, observe that when $\|\cdot\|$ is the $\ell_2$-norm and $h = \frac{1}{2}\|\cdot\|^2$, the regret term is

$$\frac{\|x^* - x_1\|_2^2}{2\eta} + \frac{\eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|_2^2}{2},$$

which is what Theorem 17.8 guarantees. Similarly, if $\|\cdot\|$ is the $\ell_1$-norm and $h$ is the negative entropy, the regret versus any point $x^* \in \Delta_n$ is

$$\frac{1}{\eta} \sum_{i=1}^{n} x_i^* \ln \frac{x_i^*}{(x_1)_i} + \frac{\eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|_\infty^2}{2/\ln 2}.$$

For linear functions $f_t(x) = \langle \ell_t, x \rangle$ with $\ell_t \in [-1,1]^n$, and $x_1 = \frac{1}{n} \cdot \mathbb{1}$, the regret is

$$\frac{KL(x^* \| x_1)}{\eta} + \frac{\eta T}{2/\ln 2} \quad \leq \quad \frac{\ln n}{\eta} + \eta T.$$

The last inequality uses that the KL divergence to the uniform distribution on $n$ items is at most $\ln n$. (Exercise!) In fact, if we start with a distribution $x_1$ that is closer to $x^*$, the first term of the regret gets smaller.

### 18.3.1 The Proof of Theorem 18.7

The proof here is very similar in spirit to that of Theorem 17.8: we give a potential function

$$\Phi_t = \frac{D_h(x^* \| x_t)}{\eta}$$

and bound the amortized cost at time $t$ as follows:

$$f_t(x_t) - f_t(x^*) + (\Phi_{t+1} - \Phi_t) \leq f_t(x^*) + \text{blah}_t. \tag{18.7}$$

Summing over all times,

$$\sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) \le \Phi_1 - \Phi_{T+1} + \sum_{t=1}^{T} \text{blah}_t$$
$$\le \Phi_1 + \sum_{t=1}^{T} \text{blah}_t = \frac{D_h(x^* \| x_1)}{\eta} + \sum_{t=1}^{T} \text{blah}_t.$$

The last inequality above uses that the Bregman divergence is always non-negative for convex functions. To complete the proof, it remains to show that $\text{blah}_t$ in inequality (18.7) can be made $\frac{\eta}{2\alpha}\|\nabla f_t(x_t)\|_*^2$. Let us focus on the unconstrained case where $x_{t+1} = x'_{t+1}$, and prove an analog of Lemma 17.9 for our generalized setting:

**Lemma 18.8** (Potential Change)**.**

$$\Phi_{t+1} - \Phi_t \le \langle \nabla f_t(x_t), x^* - x_t \rangle + \frac{\eta}{2\alpha}\|\nabla f_t(x_t)\|_*^2.$$

Note that we use the dual norm $\|\cdot\|_*$ for the gradient. Moreover, restricting Lemma 18.8 to the case of $h(x) = \|x\|^2$ and using the fact that the Euclidean norm is self-dual gives us back Lemma 17.9 bounding the potential change for standard gradient descent. The calculations in the proof below are fairly routine, and can be skipped at the first reading

*Proof of Lemma 18.8.* The change in potential is

$$= \frac{1}{\eta}\big(\underbrace{D_h(x^* \| x_{t+1}) - D_h(x^* \| x_t)}_{(\star)}\big);$$

now using the definition of the divergence,

$$(\star) = h(x^*) - h(x_{t+1}) - \langle \underbrace{\nabla h(x_{t+1})}_{\theta_{t+1}}, x^* - x_{t+1}\rangle - h(x^*) + h(x_t) + \langle \underbrace{\nabla h(x_t)}_{\theta_t}, x^* - x_t\rangle$$
$$= h(x_t) - h(x_{t+1}) - \langle \theta_{t+1}, x^* - x_{t+1}\rangle + \langle \theta_t, (x^* - x_{t+1}) + (x_{t+1} - x_t)\rangle. \tag{18.8}$$

Now we can use the $\alpha$-strong convexity of $h$ wrt to $\|\cdot\|$ to claim

$$h(x_{t+1}) \ge h(x_t) + \langle \theta_t, x_{t+1} - x_t \rangle + \frac{\alpha}{2}\|x_{t+1} - x_t\|^2.$$

Substituting into (18.8),

$$(\star) \le -\frac{\alpha}{2}\|x_{t+1} - x_t\|^2 + \langle \theta_t - \theta_{t+1}, (x_t - x_{t+1}) + (x^* - x_t)\rangle$$
$$\le \underbrace{-\frac{\alpha}{2}\|x_{t+1} - x_t\|^2 + \|\eta \nabla f_t(x_t)\|_* \|x_t - x_{t+1}\|}_{(\dagger)} + \eta \left\langle \nabla f_t(x_t), x^* - x_t \right\rangle,$$

where the latter inequality used the update rule (18.5) for mirror descent, and the Cauchy-Schwarz inequality Corollary 18.4 for general norms. Now using the AM-GM inequality shows that

$$(\dagger) \leq \frac{1}{2\alpha} \|\eta \nabla f_t(x_t)\|_*^2.$$

Finally, remembering that the change in potential is given by $\frac{1}{\eta}(\star)$ finishes the proof of Lemma 18.8. □

The rest of the proof of Theorem 18.7 follows now-familiar lines. Using Lemma 18.8, and then the convexity of $f$ on the first two terms:

$$\begin{aligned} f_t(x_t) + (\Phi_{t+1} - \Phi_t) &\leq f_t(x_t) + \langle \nabla f_t(x_t), x^* - x_t \rangle + \frac{\eta}{2\alpha} \|\nabla f_t(x_t)\|_*^2 \\ &\leq f_t(x^*) + \frac{\eta}{2\alpha} \|\nabla f_t(x_t)\|_*^2. \end{aligned}$$

Hence $\text{blah}_t$ in (18.7) is at most $\frac{\eta}{2\alpha} \|\nabla f_t(x_t)\|_*^2$, as claimed, completing the proof of Theorem 18.7.

In order to extend this to the constrained case, we need to show that if $x'_{t+1} \notin K$, and $x_{t+1} = \arg\min_{x \in K} D_h(x \| x'_{t+1})$, then

$$D_h(x^* \| x_{t+1}) \leq D_h(x^* \| x'_{t+1})$$

for any $x^* \in K$. This is a *Generalized Pythagorean Theorem* for Bregman divergences, and is left as an exercise.

## *18.4 Alternative Views of Mirror Descent*

To complete and flesh out. 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.

### *18.4.1 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

$$x_{t+1} = x_t - \eta \, H_h(x_t)^{-1} \, \nabla f(x_t).$$

Some of you may have seen Newton's method for minimizing convex functions, which has the following update rule:

$$x_{t+1} = x_t - \eta \, H_f(x_t)^{-1} \, \nabla f(x_t).$$

This means mirror descent replaces the Hessian of the function itself by the Hessian of a strongly convex function $h$. Newton's method has very strong convergence properties (it gets error $\varepsilon$ in $O(\log \log 1/\varepsilon)$ iterations!) but is not "robust"—it is only guaranteed to converge when the starting point is "close" to the minimizer. We can view mirror descent as trading off the convergence time for robustness. Fill in more on this view.

### *18.4.2 As Follow the Regularized Leader*