In this lecture, we will study the gradient descent algorithm and analyze it in the context of convex optimization.

1 Preliminaries

First, recall the following definitions:

Definition 18.1 (Convex Set)**.** A set $K \subseteq \mathbb{R}^n$ is called *convex* if for all $x, y \in K$,

$$\lambda x + (1-\lambda)y \in K,$$

for all values of $\lambda \in [0,1]$. Geometrically, this means that for any two points in K, the line connecting them is contained in K.

Definition 18.2 (Convex Function)**.** A function $f : K \to \mathbb{R}$ defined on a convex set K is called *convex* if for all $x, y \in K$,

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y),$$

for all values of $\lambda \in [0,1]$.

In the context of this lecture, we will always assume that the function f is differentiable.

Fact 18.3 (First-order condition)**.** *A function $f : K \to \mathbb{R}$ is convex if and only if*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle,$$

for all $x, y \in K$.

Geometrically, Fact 18.3 states that the function always lies above its tangent plane at all points in K (see Fig 18.1).

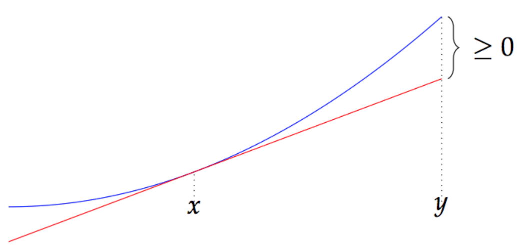

Figure 18.1: The blue line denotes the function and the red line is the tangent line at x. [VT]

If the function f is twice differentiable, then we denote by $\mathcal{H}f$ its *Hessian matrix*, i.e. its matrix of second derivatives.

$$(\mathcal{H}f)_{i,j} := \frac{\partial^2 f}{\partial x_i \partial x_j}.$$

Fact 18.4 (Second-order condition)**.** *A twice-differentiable function f is convex if and only if $\mathcal{H}f$ is pointwise positive semidefinite.*

Definition 18.5 (Lipschitz)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *G-Lipschitz* with respect to the norm $\|\cdot\|$ if

$$|f(x) - f(y)| \leq G \|x - y\|,$$

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

For today, we will only work with the ℓ_2-norm $\|\cdot\|_2$. We will consider generalizations to other norms in the next lecture when we talk about Mirror Descent.

Fact 18.6. *A differentiable function f is G-Lipschitz with respect to $\|\cdot\|_2$ if and only if*

$$\|\nabla f(x)\|_2 \leq G,$$

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

2 Convex Minimization and Gradient Descent

There are two kinds of problems that we will study.

1. Unconstrained Convex Minimization (UCM): Given a convex function f, find

$$\min_{x \in \mathbb{R}^n} f(x).$$

2. Constrained Convex Minimization (CCM): Given a convex function f and convex set K, find

$$\min_{x \in K} f(x).$$

   This is a more general problem, since setting $K = \mathbb{R}^n$ gives us the unconstrained case.

2.1 Unconstrained Convex Minimization

One useful property of convex functions is that that all local minima are also global minima. Hence, solving

$$\nabla f(x) = 0$$

would enable us to compute the global minima exactly. Quite often however, it is not possible to solve $\nabla f = 0$. For instance, the function f may not be given explicitly, but we may be given an oracle to compute gradients at any point. Even when we can write down and solve $\nabla f = 0$, it may be too expensive, and gradient descent may be a faster way to get better solutions. One example is in solving linear systems: when $f(x) = \frac{1}{2} x^\intercal A x - bx$, we have that $\nabla f(x) = 0 \iff Ax = b \iff x = A^{-1}b$, which can be solved in $O(n^\omega)$ (i.e., matrix-multiplication) time—but for "nice" matrices A we may be able to approximate a solution much faster.

Gradient descent seeks to iteratively approximate the optimal solution x^*. The main idea is simple: the gradient tells us the direction of steepest increase, so to decrease the fastest we'd like to move opposite to the direction of the gradient. Selecting an initial position x_0 and a step size η, we obtain the classical gradient descent algorithm.

$x_0 \leftarrow$ starting point;
for $t \leftarrow 1$ **to** $T-1$ **do**
  $x_t \leftarrow x_{t-1} - \eta \cdot \nabla f(x_{t-1})$;
end
return $\hat{x} := \frac{1}{T}\sum_{t=0}^{T-1} x_i$

Algorithm 1: Gradient Descent

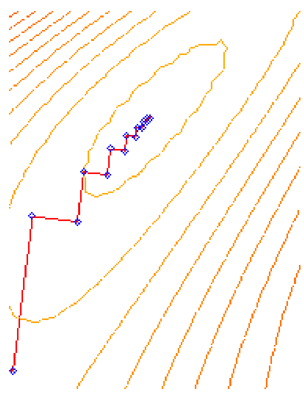

Figure 18.2: The yellow lines denote the level sets of the function f and the red walk denotes the steps of gradient descent. [Com06]

Some comments: in 2-dimensions, this is easy to visualize, since we can draw the level sets of the function f, and the gradient at a point is the normal to the tangent line at that point. The algorithm's path may be a zig-zagging walk towards the optimum goal (see Fig 18.2).

Proposition 18.7. *Let x be any point in $\mathbb{R}^d$. Let $T = \frac{1}{\varepsilon^2}G^2\|x_0 - x\|^2$ and $\eta = \frac{\|x_0-x\|}{G\sqrt{T}}$. Then the solution $\hat{x}$ returned by gradient descent satisfies*

$$f(\hat{x}) \leq f(x) + \varepsilon.$$

In particular, this holds when x is the minimizer of f.

The core of this proposition lies in the following theorem

Theorem 18.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, differentiable and G-Lipschitz. Then the gradient descent algorithm ensures that*

$$\sum_{t=0}^{T-1} f(x_t) \leq \sum_{t=0}^{T-1} f(x^*) + \frac{1}{2}\eta T G^2 + \frac{1}{2\eta}\|x_0 - x^*\|^2$$

Like in the proof of the multiplicative weights algorithm, we will use a potential function. We define

$$\Phi_t := \frac{\|x_t - x^*\|^2}{2\eta}.$$

Before we can prove the Theorem 18.8, we prove a lemma describing how the potential changes over time.

Lemma 18.9. $f(x_t) + (\Phi_{t+1} - \Phi_t) \le f(x^*) + \frac{1}{2}\eta G^2$.

Proof. By the definition of Φ_t, we see that

$$\begin{aligned} f(x_t) + (\Phi_{t+1} - \Phi_t) &= f(x_t) + \frac{1}{2\eta}\left(\|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2\right) \qquad (18.1) \\ &= f(x_t) + \frac{1}{2\eta}\left(\|(x_{t+1} - x_t) + (x_t - x^*)\|^2 - \|x_t - x^*\|^2\right) \end{aligned}$$

Now, we apply the identity that $\|a+b\|^2 = \|a\|^2 + 2\langle a, b\rangle + \|b\|^2$

$$\begin{aligned} f(x_t) + (\Phi_{t+1} - \Phi_t) &= f(x_t) + \frac{1}{2\eta}\Big(\|(x_{t+1} - x_t)\|^2 + 2\langle x_{t+1} - x_t, x_t - x^*\rangle \\ &\qquad + \|(x_t - x^*)\|^2 - \|x_t - x^*\|^2\Big) \\ &= f(x_t) + \frac{1}{2\eta}\left(\|(x_{t+1} - x_t)\|^2 + 2\langle x_{t+1} - x_t, x_t - x^*\rangle\right) \end{aligned}$$

Referring back to the gradient descent algorithm, we note that $x_{t+1} - x_t = -\eta\nabla f(x_t)$. Since f is G-Lipschitz, $\|\nabla f(x)\| \le G$ for all x. Thus,

$$\begin{aligned} f(x_t) + (\Phi_{t+1} - \Phi_t) &= f(x_t) + \frac{1}{2\eta}\|(x_{t+1} - x_t)\|^2 + \frac{1}{\eta}\langle x_{t+1} - x_t, x_t - x^*\rangle) \\ &= f(x_t) + \frac{1}{2\eta}\|\eta\nabla f(x_t)\|^2 + \frac{1}{\eta}\langle -\eta\nabla f(x_t), x_t - x^*\rangle \\ &\le f(x_t) + \frac{1}{2}\eta G^2 - \langle \nabla f(x_t), x_t - x^*\rangle \\ &\le f(x_t) + \langle \nabla f(x_t), x^* - x_t\rangle + \frac{1}{2}\eta G^2 \end{aligned}$$

Since f is convex, we know that $f(x_t) + \langle\nabla f(x_t), x^* - x_t\rangle \le f(x^*)$. Thus, we conclude that

$$f(x_t) + (\Phi_{t+1} - \Phi_t) \le f(x^*) + \frac{1}{2}\eta G^2$$

□

Now that we understand how our potential changes over time, proving the theorem is straightforward.

Proof of Theorem 18.8. We start with the inequality

$$f(x_t) + (\Phi_{t+1} - \Phi_t) \le f(x^*) + \frac{1}{2}\eta G^2$$

Summing over $t = 0, \ldots, T-1$, we see that

$$\sum_{t=0}^{T-1} f(x_t) + \sum_{t=0}^{T-1}(\Phi_{t+1} - \Phi_t) \le \sum_{t=0}^{T-1} f(x^*) + \frac{1}{2}\eta G^2 T$$

Note that the sum of potentials on the left is a telescoping sum, so we find that

$$\sum_{t=0}^{T-1} f(x_t) + \Phi_T - \Phi_0 \le \sum_{t=0}^{T-1} f(x^*) + \frac{1}{2}\eta G^2 T$$

Since the potentials are nonnegative, we can drop the Φ_T term. Thus, we see that

$$\sum_{t=0}^{T-1} f(x_t) - \Phi_0 \leq \sum_{t=0}^{T-1} f(x^*) + \frac{1}{2}\eta G^2 T$$

Substituting in the definition of Φ_0 and moving it over to the right hand side completes the proof. $\square$

Proof of Proposition 18.7. By definition, of $\hat{x}$ and by the convexity of f,

$$f(\hat{x}) = f\left(\frac{1}{T}\sum_{t=0}^{T-1} x_t\right) \leq \frac{1}{T}\sum_{t=0}^{T-1} f(x_t).$$

By Theorem 18.8, we know that

$$\frac{1}{T}\sum_{t=0}^{T-1} f(x_t) \leq f(x^*) + \frac{1}{2}\eta G^2 + \frac{1}{2\eta T}\|x_0 - x^*\|^2.$$

Substituting in $\eta = \frac{\|x_0 - x^*\|}{G\sqrt{T}}$, we see that

$$\begin{aligned} f(\hat{x}) &\leq f(x^*) + \frac{1}{2\sqrt{T}}\|x_0 - x^*\|G + \frac{1}{2\sqrt{T}}\|x_0 - x^*\|G, \\ &= f(x^*) + \frac{\|x_0 - x^*\|G}{\sqrt{T}}. \end{aligned}$$

Finally, we set $T = \frac{1}{\varepsilon^2}G^2\|x_0 - x^*\|^2$ to obtain

$$f(\hat{x}) \leq f(x^*) + \varepsilon.$$

$\square$

This analysis, and in particular the $1/\varepsilon^2$ dependence on ε is tight if we just assume f is G-Lipschitz. Moreover, we did not (and cannot) show that $\hat{x}$ is close in distance to x^*; we just show that $f(\hat{x}) \approx f(x^*)$. Indeed, if the function is very flat close to the origin, we cannot hope to be close in distance. (To improve on the $1/\varepsilon^2$ dependence, or to show physical closeness of x^* and $\hat{x}$, we need further assumptions; see Section 4.)

2.2 Constrained Convex Minimization

Unlike the unconstrained case, now the derivative may not be 0 at the optimum. Nonetheless, the main idea of gradient descent still yields a good algorithm. Here is some intuition why. When f is a convex function defined on $\mathbb{R}^n$, the following conditions are equivalent

1. x^* is a local minimum,
2. $\nabla f(x^*) = 0$,
3. For all $y \in \mathbb{R}^n$, $\langle \nabla f(x^*), y - x^* \rangle \geq 0$.

Property 3 is essentially that $\langle a, b\rangle = 0$ for all b if and only if $a = 0$.

When we constrain our domain to convex set K, the minimum may not have gradient zero. However, if the minimum doesn't have gradient zero, it must necessarily be on the boundary of K. Either way, we can show that x^* is a local minimum if and only if

$$\langle \nabla f(x^*), y - x^*\rangle \geq 0 \qquad \text{for all } y \in K.$$

When x^* is in the interior of K, this is equivalent to $\nabla f(x^*) = 0$, but this is not so when x^* is on the boundary of K. Here's another interpretation of the statement: starting from x^*, if we walk a little bit in some direction but stay in K, then f should increase. This means stepping in the reverse direction of the gradient is still a good idea!

2.2.1 Projected Gradient Descent

We need change our algorithm to ensure that the new point x_{t+1} lies within K. To ensure this, we simply project each step back onto K. Let $\mathrm{Proj}_K : \mathbb{R}^n \to K$ be defined as

$$\mathrm{Proj}_K(y) = \arg\min_{x \in K} \|x - y\|_2 .$$

The modified algorithm is given below in Algorithm 2, with the changes highlighted in blue.

$x_0 \leftarrow$ starting point;
for $t \leftarrow 1$ **to** $T - 1$ **do**
  $x'_t \leftarrow x_{t-1} - \eta_t \cdot \nabla f(x_{t-1})$;
  $x_t \leftarrow \mathrm{Proj}_K(x'_t)$;
end
return $\hat{x} = \frac{1}{T}\sum_{t=0}^{T-1} x_i$

Algorithm 2: Gradient Descent For CCM

We will show below that a theorem (and analysis) similar to that of Theorem 18.8 holds.

Theorem 18.10. *Let $f : K \to \mathbb{R}$ be a G-Lipschitz convex function defined on a convex set K with diameter D. Then provided that $x_0 \in K$, $T = \left(\frac{GD}{\varepsilon}\right)^2$, and $\eta = \frac{\varepsilon}{G^2}$, the solution $\hat{x}$ produced by the projected gradient descent algorithm satisfies*

$$f(\hat{x}) - f(x^*) \leq \varepsilon.$$

Proof. The argument is essentially the same as that for Theorem 18.8. The only hiccup is that now $-\eta\nabla f(x_t) = x'_{t+1} - x^*$, not $x_{t+1} - x^*$. But this is okay: if we could replace x_{t+1} with x'_{t+1} in (18.1), we would be all set. This boils down to showing

$$\left\|x'_{t+1} - x^*\right\| \geq \|x_{t+1} - x^*\| .$$

But this is easy, because $x_{t+1} = \mathrm{Proj}_K(x'_{t+1})$ and $x^* \in K$. Because K is convex, projecting to it gets us closer to *every point* in K, in particular to x^*. This is because the angle $x^* \to x_{t+1} \to x'_{t+1}$ cannot be acute: if it were acute, we could show that K wasn't actually convex. See Figure 18.3. □

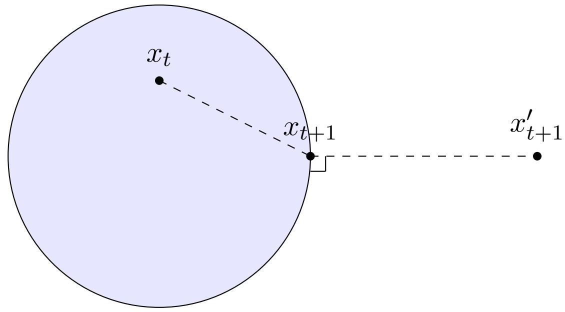

Figure 18.3: Projection onto a convex body

3 Online Gradient Descent, and Relationship with MW

We considered Gradient Descent for the *offline* convex minimization problem, but one can use it even when the function changes over time. Indeed, consider the *online convex optimization (OCO)* problem: at each time step, you propose an $x_t \in K$ and an adversary exhibits a function $f_t : K \to \mathbb{R}$ with $\|\nabla f_t\| \leq G$. The cost of each time step is $f_t(x_t)$ and your objective is to minimize

$$\text{regret } = \sum_t f_t(x_t) - \min_{x^* \in K} \sum_t f_t(x^*).$$

To solve this problem, we can use the same algorithm, with one natural modification: the update rule is now taken with respect to gradient of the *current* function f_t.

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

Looking back at the proof in Section 2, Lemma (18.9) immediately extends to give us

$$f_t(x_t) + \Phi_{t+1} - \Phi_t \leq f_t(x^*) + \frac{1}{2}\eta G^2.$$

Now summing this over all times t gives

$$\begin{aligned}
\sum_{t=0}^{T-1} \big(f_t(x_t) - f_t(x^*)\big) &\leq \sum_{t=0}^{T-1} \Phi_t - \Phi_{t+1} + \frac{1}{2}\eta G^2 \\
&= \Phi_0 - \Phi_T + \frac{1}{2}\eta T G^2 \\
&\leq \Phi_0 + \frac{1}{2}\eta T G^2
\end{aligned}$$

and using $T \geq \frac{1}{\varepsilon^2}\|x_0 - x^*\|^2 G^2$ and $\eta = \frac{\|x_0 - x^*\|}{G\sqrt{T}}$ as above, this implies

$$\frac{1}{T}\sum_{t=0}^{T} \big(f_t(x_t) - f_t(x^*)\big) \leq \frac{\|x_0 - x^*\| G}{\sqrt{T}} \leq \varepsilon.$$

One advantage of this algorithm (and analysis) is that it holds for all convex bodies K and all convex functions, as opposed to the MW algorithm which, as stated, works just for the unit simplex and linear losses. Of course it now depends on $\|x_0 - x^*\|$ (which, in the worst case is the diameter of

K), and G (which is related to the class of functions). If we consider these quantities as constants, the $(\frac{1}{\varepsilon^2})$ dependence is the same.

In many cases we do care about the fine-grained dependence on K and functions, so let's compare the two for the unit simplex and linear lossses (i.e., functions $f_t(x) = \langle \ell_t, x\rangle$ with $\|\ell\|_\infty = 1$). The regret bound above give us $T = \frac{2N}{\varepsilon^2}$ because $\|x_0 - x^*\| \leq \mathrm{diam}(K) = \sqrt{2}$ and $\|\nabla l_i\|_2 \leq \sqrt{N}$. This is much worse compared to $T = \frac{\log N}{\varepsilon^2}$, which is the guarantee that multiplicative weights provides.

The problem, at a high level, is that we are "choosing the wrong norm": we are working in ℓ_2 instead of ℓ_1. In the next lecture we will see what this means, and how this dependence on N be improved via the Mirror Descent framework.

3.1 Subgradients

What if the convex function f is not differentiable? Staring at the proofs above, all we need is the following:

Definition 18.11 (Subgradient)**.** A vector z_x is called a *subgradient* at point x if

$$f(y) \geq f(x) + \langle z_x, y - x\rangle \qquad \text{for all } y \in \mathbb{R}^n.$$

Now we can use subgradients at the point x wherever we used $\nabla f(x)$, and the entire proof goes through. In some cases, an approximate subgradient may also suffice.

4 Stronger Assumptions

If the function f is better-behaved, then we can improve the guarantees for gradient descent in two ways: we can reduce the dependence on ε, and we can weaken (or remove) the dependence on parameters G, D. There are two standard assumptions one can make on the convex function: that it is "not too flat" (captured by the idea of *strong convexity*), and it is not "not too curved" (i.e., it is *smooth*). We now use these assumptions to improve guarantees.

4.1 α-strongly convex functions

Definition 18.12 (Strong Convexity)**.** A function is *α-strongly convex* if

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\alpha}{2}\|x - y\|^2 \text{ for all } x, y \in K. \tag{18.2}$$

Fact 18.13. *A twice-differentiable convex function f is α-strongly convex if and only if all eigenvalues of $\mathcal{H}f$ are at least α at every point.*

In this case, the gradient descent algorithm with step size $\eta_t = O\big(\frac{1}{\alpha t}\big)$ converges to a solution with error ε in $T = O(\frac{G^2}{\alpha\varepsilon})$.

4.2 β-smooth function

Definition 18.14. A function f is a β-smooth convex function if

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\beta}{2}\|x - y\|^2 \text{ for all } x, y \in K. \tag{18.3}$$

Fact 18.15. *A twice-differentiable convex function f is β-smooth if and only if all eigenvalues of $\mathcal{H}f$ are at most β at every point.*

In this case, the gradient descent algorithm with step size $\eta_t = O\big(\frac{1}{\beta}\big)$ yields an $\hat{x}$ which satisfies $f(\hat{x}) - f(x^*) \le \varepsilon$ when $T = O\left(\frac{\|x_0 - x^*\|\beta}{\varepsilon}\right)$.

4.3 Well-conditioned Functions

Functions that are both β-smooth and α-strongly convex are known as "well-conditioned" functions. From the facts above, the eigenvalues of the Hessian $\mathcal{H}f$ must lie in the interval $[\alpha, \beta]$ at all points $x \in K$. In this case, we get a much stronger convergence—ε-closeness in time $T = O(\log \frac{1}{\varepsilon})$.

Theorem 18.16. *For a function f which is β-smooth and α-strongly convex, let x^* be the solution to the unconstrained convex minimization problem* $\arg\min_{x\in\mathbb{R}^n} f(x)$*. Then running gradient descent with $\eta_t = 1/\beta$ gives*

$$f(x_t) - f(x^*) \le \frac{\beta}{2} \exp\left(\frac{-t}{\kappa}\right) \|x_0 - x^*\|^2 .$$

Proof. For β-smooth f, we can use Definition 18.14 to get

$$f(x_{t+1}) \le f(x_t) - \eta\|\nabla f(x_t)\|^2 + \eta^2\frac{\beta}{2}\|\nabla f(x_t)\|^2.$$

The right hand side is minimized by setting $\eta = \frac{1}{\beta}$, when we get

$$f(x_{t+1}) - f(x_t) \le -\frac{1}{2\beta}\|\nabla f(x_t)\|^2. \tag{18.4}$$

For α-strongly-convex f, we can use Definition 18.12 to get:

$$\begin{aligned} f(x_t) - f(x^*) &\le \langle \nabla f(x_t), x_t - x^*\rangle - \frac{\alpha}{2}\|x_t - x^*\|^2 ,\\ &\le \|\nabla f(x_t)\| \; \|x_t - x^*\| - \frac{\alpha}{2}\|x_t - x^*\|^2 ,\\ &\le \frac{1}{2\alpha}\|\nabla f(x_t)\|^2 , \end{aligned}$$

where we use that the right hand side is maximized when $\|x_t - x^*\| = \|\nabla f(x_t)\|/\alpha$. Now combining with (18.4) we have that

$$f(x_{t+1}) - f(x_t) \le -\frac{\alpha}{\beta}\Big(f(x_t) - f(x^*)\Big),$$

or setting $\Delta_t = f(x_t) - f(x^*)$ and rearranging, we get

$$\Delta_{t+1} \le \left(1 - \frac{\alpha}{\beta}\right)\Delta_t \le \left(1 - \frac{1}{\kappa}\right)^t \Delta_0 \le \exp\left(-\frac{t}{\kappa}\right)\cdot\Delta_0.$$

We can control the value of Δ_0 by using (18.3) in $x = x^*, y = x_0$; since $\nabla f(x^*) = 0$, get $\Delta_0 = f(x_0) - f(x^*) \le \frac{\beta}{2}\|x_0 - x^*\|^2$. ∎

Strongly-convex (and hence well-conditioned) functions have the nice property that if $f(x)$ is close to $f(x^*)$ then x is close to x^*: intuitively, since the function is curving at least quadratically, the function values at points far from the minimizer must be significant. Formally, use (18.2) with $x = x^*, y = x_t$ and the fact that $\nabla f(x^*) = 0$ to get

$$\|x_t - x^*\|^2 \le \frac{2}{\alpha}(f(x_t) - f(x^*)).$$

Acknowledgments

These lecture notes were scribed by Mark Gillespie and Daniel Anderson, based on previous scribe notes of Guru Guruganesh and Ziv Scully.

References

[Com06] Wikimedia Commons. Gradient ascent(countour), 2006. Available at http://en.wikipedia.org/wiki/Gradientdescent#/media/File:Gradientascent(contour).png. 18.2

[VT] Nisheeth Vishnoi and Jakub Tarnawski. Fundamentals of convex optimization. lecture 1 basics, gradient descent and its variants. Available at http://tcs.epfl.ch/files/content/sites/tcs/files/Lec1-Fall14-Ver1.pdf. 18.1