

CS 561, Gradient Descent

Jared Saia

University of New Mexico

The Problem

Given:

- Convex space \mathcal{K}
- Convex function f

Goal: Find $x \in \mathcal{K}$ that minimizes $f(x)$

Variables

- $D = \max_{x,y \in \mathcal{K}} |x - y|$
- G is an upperbound on $|\nabla f(x)|$ for any $x \in \mathcal{K}$

Note: all norms are 2-norms. D is known as the *diameter* of \mathcal{K}

What is Gradient?

- The *gradient* of a function f (∇f) is just the derivatives of f written as a vector.
- Ex: The gradient of $f(x, y) = 2x + 3y$ is the vector $(2, 3)$
- Ex: The gradient of $f(x, y) = x^2 + y^2$ at the point $x = 2, y = 3$ is $(4, 6)$
- Ex: The gradient of $f(x, y) = xy$ at the point $x = 2, y = 3$ is $(3, 2)$

Gradient Descent Algorithm

$$\eta \leftarrow \frac{D}{G\sqrt{T}}$$

Repeat for $i = 0$ to T :

1. $y_{i+1} \leftarrow x_i - \eta \nabla f(x_i)$
2. $x_{i+1} \leftarrow$ Projection of y_{i+1} onto \mathcal{K}

Output $z = \frac{1}{T} \sum_i x_i$

Theorem 1

Theorem 1 *Let $x^* \in \mathcal{K}$ be the value that minimizes f . Then, for any $\epsilon > 0$, if we set $T = \frac{4D^2G^2}{\epsilon^2}$, we can ensure that:*

$$f(z) \leq f(x^*) + \epsilon$$

Convexity - Another View

A convex function that is differentiable satisfies the following (basically, this says that the function is above the tangent plane at any point). Recall that $\nabla f(x)$ is the vector whose i -th coordinate is $\partial f / \partial x_i$

$$f(x + z) \geq f(x) + \nabla f(x) \cdot z, \text{ for all } x, z$$

This is equivalent to:

$$f(x) - f(y) \leq \nabla f(x) \cdot (x - y) \text{ for all } x, y$$

Proof (I)

$$\begin{aligned} |x_{i+1} - x^*|^2 &\leq |y_{i+1} - x^*|^2 \\ &= |x_i - x^* - \eta \nabla f(x_i)|^2 \\ &= |x_i - x^*|^2 + \eta^2 |\nabla f(x_i)|^2 - 2\eta \nabla f(x_i) \cdot (x_i - x^*) \end{aligned}$$

First step holds since x_{i+1} projects y_{i+1} onto a space that contains x^* . Second step holds by definition of y_{i+1} . Last step holds by noting that $|v|^2 = v \cdot v$.

Proof (II)

From last slide, we have:

$$|x_{i+1} - x^*|^2 \leq |x_i - x^*|^2 + \eta^2 |\nabla f(x_i)|^2 - 2\eta \nabla f(x_i) \cdot (x_i - x^*)$$

Reorganizing, and using definition of G , we get:

$$\nabla f(x_i) \cdot (x_i - x^*) \leq \frac{1}{2\eta} (|x_i - x^*|^2 - |x_{i+1} - x^*|^2) + \frac{\eta}{2} G^2$$

Using Slide 6 (Convexity Fact), we then get:

$$f(x_i) - f(x^*) \leq \frac{1}{2\eta} (|x_i - x^*|^2 - |x_{i+1} - x^*|^2) + \frac{\eta}{2} G^2 \quad (1)$$

Proof (III)

Now sum last inequality for $i = 1$ to T . After cancellations, we get.

$$\sum_{i=1}^T (f(x_i) - f(x^*)) \leq \frac{1}{2\eta} (|x_1 - x^*|^2 - |x_{T+1} - x^*|^2) + \frac{T\eta}{2} G^2$$

Divide the above inequality by T . By convexity, $f(\frac{1}{T}(\sum_i x_i)) \leq \frac{1}{T} \sum_i f(x_i)$. Since $z = \frac{1}{T} \sum_i x_i$, we now get

$$f(z) - f(x^*) \leq \frac{D^2}{2\eta T} + \frac{\eta}{2} G^2.$$

Since $\eta = \frac{D}{G\sqrt{T}}$, the right hand side is at most $2\frac{DG}{\sqrt{T}}$. Then since $T = \frac{4D^2G^2}{\epsilon^2}$, we see that $f(z) \leq f(x^*) + \epsilon$

Online Gradient Descent

- Surprisingly, the gradient descent algorithm can work even when the function to minimize changes in every round!
- Even if these functions are chosen by an adversary! - So long as they are always convex.
- We just need to make a slight tweak in the algorithm (next slide - can you spot the differences?)

Online GD Algorithm

$$\eta \leftarrow \frac{D}{G\sqrt{T}}$$

Repeat for $i = 0$ to T :

1. $y_{i+1} \leftarrow x_i - \eta \nabla f_i(x_i)$
2. $x_{i+1} \leftarrow$ Projection of y_{i+1} onto \mathcal{K}

Online Gradient Theorem

Theorem 2 (Zinkevich's Theorem) *Let $x^* \in \mathcal{K}$ be the value that minimizes $\sum_i f_i(x^*)$. Then, for all $T > 0$,*

$$\frac{1}{T} \sum_i (f_i(x_i) - f_i(x^*)) \leq \frac{2DG}{\sqrt{T}}.$$

Notes: The left hand side of this inequality is called the *regret* per step.

Proof

- Equation 1 from Slide 9 bounds the regret for step i
- Sum these regrets over all i and divide by T to get the theorem!

Application: Portfolio Management

- Review Section 16.6 in Arora notes

Stochastic Gradient Descent

The final major trick of GD enables significant speed up. Assume we want to minimize over just one function, f , again.

- In each step, i , we estimate the gradient of f at x_i based on *one* random data item
- Call this random gradient g_i , where $E(g_i) = \nabla f(x_i)$
- Then, using the g_i 's we get essentially same results as if we had the true gradient

Stochastic GD Algorithm

$$\eta \leftarrow \frac{D}{G\sqrt{T}}$$

Repeat for $i = 0$ to T :

1. $g_i \leftarrow$ a random vector, such that $E(g_i) = \nabla f(x_i)$
2. $y_{i+1} \leftarrow x_i - \eta g_i$
3. $x_{i+1} \leftarrow$ Projection of y_{i+1} onto \mathcal{K}

$$\text{Output } z = \frac{1}{T} \sum_i x_i$$

Stochastic GD Theorem

Theorem 3 $E(f(z)) \leq f(x^*) + \frac{2DG}{\sqrt{T}}$.

Proof (1/2)

$$\begin{aligned} E(f(z)) &= E\left(f\left(\frac{1}{T}\sum_{i=1}^T x_i\right)\right) \\ &\leq E\left(\frac{1}{T}f\left(\sum_{i=1}^T x_i\right)\right) && \text{By convexity of } f \\ &\leq \frac{1}{T}E\left(f\left(\sum_{i=1}^T x_i\right)\right) && \text{Linearity of Exp.} \end{aligned}$$

Proof (2/2)

$$\begin{aligned} E(f(z) - f(x^*)) &\leq \frac{1}{T} E\left(\sum_i f(x_i) - f(x^*)\right) && \text{By previous slide} \\ &\leq \frac{1}{T} \sum_i E(\nabla f(x_i) \cdot (x_i - x^*)) && \text{Using Slide 6} \\ &= \frac{1}{T} \sum_i E(g_i \cdot (x_i - x^*)) && \text{Cuz } E(g_i \cdot x) = \nabla f(x_i) \cdot x \\ &= \frac{1}{T} \sum_i E(f_i(x_i) - f_i(x^*)) && \text{Letting } f_i(x) = g_i \cdot x \\ &= E\left(\frac{1}{T} \sum_i (f_i(x_i) - f_i(x^*))\right) && \text{Linearity of Exp.} \\ &\leq \frac{2DG}{\sqrt{T}} && \text{Regret bound using Zinkevich's Thm} \end{aligned}$$

Two Notes on Proof

- Requirement in Step 3: $E(g_i \cdot x) = \nabla f(x_i) \cdot x$, for all x
- Holds since dot product is linear, and $E(g_i) = \nabla f(x_i)$

- Requirement in Last Step: $f_i(x)$ is convex. Needed to use Zinkevich
- Holds since $f_i(x) = g_i \cdot x$ is *linear*

Take Away

Gradient Descent comes in 3 basic flavors:

- Standard Gradient Descent
- Online Gradient Descent
Works even when function is changing
- Stochastic Gradient Descent
Just need the correct gradient in **expectation**