Lecture #10: Convex Optimization, Gradient Descent

Instructor: Aditya Bhaskara *Scribe: Manila Devaraja*

CS 5966/6966: Theory of Machine Learning

February 10th, 2022

Abstract

In this lecture, we will understand the convex optimization, the standard gradient descent theorem and vanilla analysis of gradient descent.

1 Introduction

So far, we have seen Empirical Risk Minimization(ERM) generalizes well. Generalization bounds tell us that if your hypothesis class is nice, ERM is all you need. We need to find a hypothesis $h \in H$ that minimizes the empirical loss (risk):

$$argmin_{h \in H} L_S(h) = \sum_{i=1}^{N} \mathbb{1}[h(x_i) \neq y_i]$$

Even for a simple class like a best linear classifier, to reduce empirical risk i.e to reduce the number of mistakes it is NP hard to approximate. The problem is that the loss function $\mathbb{1}[h(x_i) \neq y_i]$ is too discrete. This motivates us to consider different loss functions that can be minimized easily, and can act as a proxy for the number of mistakes.

Loss functions are proxy for Error Risk Minimization(ERM).

The loss function can be written as,

$$L_S(h) = \sum_{i=1}^{N} loss(h(x_i), y_i)$$

The squared loss is given by,

$$L_S(h) = \sum_{i=1}^{N} (h(x_i) - y_i)^2$$

Moving to loss minimization converts ERM into more tractable optimization problem.

2 Convex functions and minimization

Formally, we can define a convex function as follows:

2.1 definition (convex function) : A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be a convex function over a convex set D if

(1) $f(\alpha u + (1-\alpha)v) \leq \alpha f(u) + (1-\alpha) f(v), \forall \alpha \in (0,1)$

In words, f is convex if for any u, v, the graph of f between u and v lies below the line segment joining $f(u)$ and $f(v)$. An illustration of a convex function, $f : \mathbb{R} \to \mathbb{R}$, is depicted in the following image.

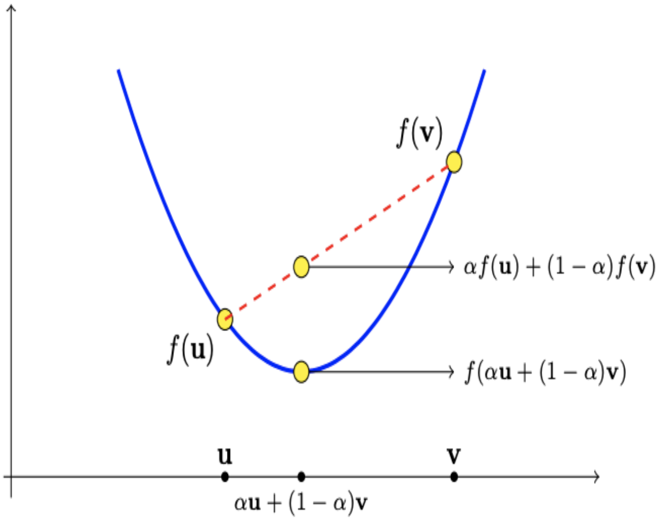
$f(\mathbf{v})$
$\alpha f(\mathbf{u}) + (1-\alpha) f(\mathbf{v})$
$f(\mathbf{u})$
$f(\alpha\mathbf{u} + (1-\alpha)\mathbf{v})$
$\mathbf{u}$
$\mathbf{v}$
$\alpha\mathbf{u} + (1-\alpha)\mathbf{v}$

Figure 1: Convex function

One of important properties of convex functions, as we discussed in last lecture, is that every local minimum of the function is also a global minimum.

Let us try to understand the concept of gradient descent. For a given point x in convex domain, we need to find a point such that $f(x+\delta) < f(x)$. According to Taylor's expansion,

$$f(x+\delta) = f(x) + \delta f'(x) + \frac{\delta^2}{2!} f''(x) + ... + \frac{\delta^(k)}{k!} f^k(x)$$

The first order "Taylor approximation" for f can be written as:

(2) $f(x+\delta) \approx f(x) + \langle \delta, \nabla f(x) \rangle \quad \forall x, \delta \in \mathbb{R}^d$

Where

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial x1} \\ \frac{\partial f}{\partial x2} \\ . \\ . \\ . \\ \frac{\partial f}{\partial x_d} \end{pmatrix}$$

Another important property of convex functions is that for every w we can construct a tangent to f at w that lies below f everywhere. If f is differentiable, this tangent is the linear function

$$l(\mathbf{u}) = \mathbf{f}(\mathbf{w}) + \langle \nabla \mathbf{f}(\mathbf{w}), \mathbf{u} - \mathbf{w} \rangle$$

where $\nabla f(\mathbf{w})$ is the gradient of f at $\mathbf{w}$. That is, for convex differentiable functions,

$$(3) \quad \forall \mathbf{u}, \mathbf{f}(\mathbf{u}) \geq \mathbf{f}(\mathbf{w}) + \langle \nabla \mathbf{f}(\mathbf{w}), \mathbf{u} - \mathbf{w} \rangle$$

Intuitively, $\nabla f(\mathbf{w})$ is the the direction in which f increases. Hence, we move opposite to it to reach the true minimum of $f(x)$.

Formally, gradient descent is an iterative optimization procedure in which at each step we improve the solution by taking a step along the negative of the gradient of the function to be minimized at the current point.

In order to get $f(x + \delta) < f(x)$, we can make $\delta = -C\nabla f(x)$ for some x. Which means, we get

$$\langle \delta, \nabla f(x) \rangle = -C||\nabla f(x)||^2$$

As you progress and reach a point where $\nabla f(x) = 0$, we have reached a local optima which in turn is global optima for convex functions.

3 Gradient Descent Algorithm

For higher dimensions, let us consider continuous differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ at w. Gradient descent is an iterative algorithm. We start with an initial value of w (say, $w^{(1)} = 0$). Then, at each iteration, we take a step in the direction of the negative of the gradient at the current point from $t = (0...T - 1)$. The update of step is given by,

$$w^{(t+1)} = w^{(t)} - \eta \nabla f(w(t)$$

Where η is some parameter we will later discuss.
By considering Taylor's approximation (2), Therefore, for w close to w(t) we have that

η is called step size in this context. It is a parameter that should be chosen appropriately, depending on f.

$$f(w) \approx f(w^{(t)}) + \langle w - w^{(t)}, \nabla f(w^{(t)}) \rangle$$

Hence we can minimize the approximation of f(w) by choosing T such that $||\nabla f(w^{(t)})||$ is "small enough".
This algorithm can also be applied to non-convex functions except the local optima found need not be global optima.

But does the algorithm always guarantee loss minimization? What does it depend on? How do we choose η and T ? What if $w^{(t+1)}$, updated step is not in Domain?

- For some naive reasons, $\eta = \frac{1}{\sqrt{t}}$ works well and usually started with.
- The right way to ensure the updated step is in domain is through pro-

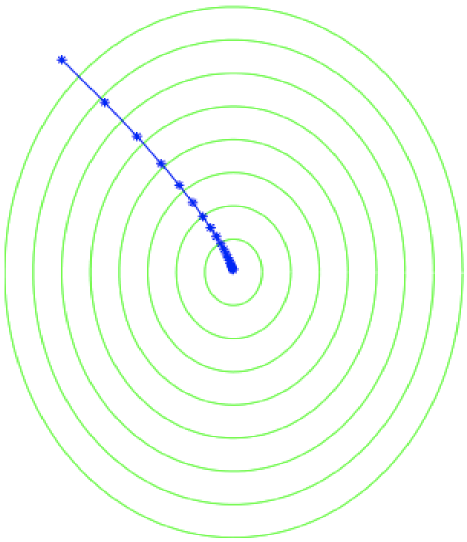

Figure 2: An illustration of the gradient descent algorithm.

jection i.e project w^{t+1} to D.

A projection of a point x onto a convex set D is defined to be the closest point to x in D.

- If D is complex, finding projection itself might become an optimization problem.

- Fixing η is generally a problematic thing. There is a chance that descent continues to oscillate within the function. It is better for now to see $\eta = \eta_t$ which means it changes at each step. In further lectures, we will discuss how to choose η.

4 Vanilla Analysis of Gradient Descent

4.1 definition (Lipschitz property). A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be L-Lipschitz if $|f(x) - f(y)| \leq L||x - y|| \quad \forall x, y$.
Corollary means, $||\nabla f(x)|| < L$
Intuitively, a Lipschitz function cannot change too fast.

The Vanilla Analysis is studying gradient descent algorithm under the two assumptions:

- f is L-Lipschitz, and domain $D = \mathbb{R}^d$
- The starting point $x(0)$ is at a distance $\leq B$ from the optimum w

Theorem: Consider running T steps of gradient descent with a fixed learning rate η, then we have

$$\frac{1}{T}\sum_{t=1}^{T} f(w_t) - f(w) \leq \frac{B^2}{2\eta T} + \frac{L^2\eta}{2} \tag{4}$$

where w is the true minimizer of f i.e, $w = argmin_{x \in \mathbb{R}^2} f(x)$

According to the basic inequality of convexity, the function lies "above" the tangent plan at *any* point.

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

The drop in potential is "related" to change in function value

As we move from any point on a convex function and move towards the minimizer, by inequality we can write the following:

$$f(w) - f(w_t) \geq \langle \nabla f(w_t), w - w_t \rangle$$

In the next lecture, we prove the above theorem using the inequality given above. For now, we will understand what it intuitively means.

This bound can be used to determine the setting of η and T. The LHS is the average difference between the function values of the current iterate and the optimum. Intuitively, this means there is region around optimum which can be reached over many iterations starting at any point on the convex function.