ML & Optimization Lecture 11

(see Chapter 3 of Convex Optimization by Sébastien Bubeck)

- Kernel Learning, via Fernet's condition

- Gradient Descent for β-smooth cvx optim. problems

$$\omega^* = \operatorname*{argmin}_{\omega} \|X\omega - y\|_2^2 \qquad X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

$$\iff \nabla_\omega \|X\omega^* - y\|_2^2 = 0$$

$$\iff X^T(X\omega^* - y) = 0$$

$$\iff \omega^* = (X^TX)^{-1}X^Ty$$

Let $X = U\Sigma V^T$ where the rhs is the SVD of X

Then V is an onb for the row span of X

Notice

$$X^TX = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T$$
$$(X^TX)^{-1} = V\Sigma^{-2}V^T$$

so

$$\omega^* = (X^TX)^{-1}X^Ty \text{ is in the span of the data points}$$
$$= \sum_{i=1}^{n} \alpha_i^* x_i$$

Similarly, consider an l_2 regularized ERM

$$w^* = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(w^T x_i, y_i) + \lambda \|w\|_2^2$$

$$\Leftrightarrow \quad 0 \in \frac{1}{n} \sum_{i=1}^{n} x_i \underbrace{\partial \ell(w^{*T} x_i, y_i)}_{\text{denotes the subdifferential of } \ell \text{ w.r.t. the first argument}} + 2\lambda w^*$$

$$\Leftrightarrow \quad \exists z_i \in \partial \ell(w^{*T} x_i, y_i):$$

$$0 = \frac{1}{n} \sum_{i=1}^{n} z_i x_i + 2\lambda w^*$$

$$\Leftrightarrow \quad w^* = \sum_{i=1}^{n} \alpha_i^* x_i$$

Optimal parameter is again in the span of the data points!

Now we see that solving the original ERM is equivalent to solving

$$\alpha^* = \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \; \frac{1}{n} \sum_{i=1}^{n} \ell\left(\sum_{j=1}^{n} \alpha_j \langle x_j, x_i \rangle, y_i \right) + \lambda \left\| \sum_{j=1}^{n} \alpha_j x_j \right\|_2^2$$

Notice

$$\left\| \sum_{j=1}^{n} \alpha_j x_j \right\|_2^2 = \sum_{j=1}^{n} \sum_{i=1}^{n} \alpha_j \alpha_i \underbrace{\langle x_j, x_i \rangle}_{:= K_{ij}}$$

where $K = \left[\langle x_i, x_j \rangle \right]_{i,j}$, $K \in \mathbb{R}^{n \times n}$ is the linear kernel matrix

$$= \alpha^T K \alpha$$

and

$$\sum_{j=1}^{n} \alpha_j \langle x_j, x_i \rangle = \sum_{j=1}^{n} K_{ij} \alpha_j = (K\alpha)_i$$

so the original ERM becomes a <u>kernel machine</u>

$$\alpha^* = \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \; \frac{1}{n} \cdot \sum_{i=1}^{n} \ell\left((K\alpha)_i, y_i\right) + \alpha^T K \alpha$$

$\hookrightarrow = e_i^T K \alpha$, so is an affine function of α

where $K = XX^T$ is the linear kernel matrix

Notice that K is PSD, because for any vector z,

$$z^T K z = z^T X X^T z = \|X^T z\|_2^2 \geq 0$$

so this is a convex optimization problem

Why is this useful? We converted a d-dim optimization problem to an n-dim optim problem.

Consider learning using nonlinear combinations of the features x, via a <u>feature map</u>

$$\phi : \mathbb{R}^d \to \mathbb{R}^D \quad \text{where } D \gg d$$

<u>Example</u>

ϕ : $x \mapsto$ all monomials of total degree up to p in the features of x

e.g. $\phi(x) = [1, x_1, \dots, x_d, x_1^2, \dots, x_d^2, x_1x_2, \dots, x_{d-1}x_d]$

when $p=2$,

$$D = 1 + d + \binom{d}{2} + d = \Theta(d^2)$$

in general

$$D = \Theta(d^p)$$

Now our ERM to learn with these features is

$$\omega_* = \underset{\omega \in \mathbb{R}^D}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(\omega^T \phi(x_i), y_i\right) + \lambda \|\omega\|_2^2$$

is a D-dim optim problem. When $D > n$, it's more efficient to work with the kernel problem

$$\alpha_* = \underset{\alpha \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell\left((K\alpha)_i, y_i\right) + \alpha^T K \alpha$$

We assume ϕ is such that the <u>kernel function</u>

$K(x, z) = \langle \phi(x), \phi(z) \rangle$ can be computed efficiently (in $O(d)$ time) so K can be computed in $O(n^2 d)$, whose

$$K = \left[K(x_i, x_j)\right]_{i,j=1,\dots,n} = \Phi \Phi^T \in \mathbb{R}^n, \quad \Phi = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \end{bmatrix} \in \mathbb{R}^{n \times D}$$

To predict, notice

$$(\omega^*)^T \phi(x) = \left\langle \sum_{i=1}^{n} \alpha_i^* \phi(x_i), \phi(x) \right\rangle$$

$$= \sum_{i=1}^{n} \alpha_i^* \langle \phi(x_i), \phi(x) \rangle$$

$$= \sum_{i=1}^{n} \alpha_i^* K(x_i, x)$$

In fact, we can take D to be infinite, e.g.

$$K(x, y) = \exp\left(\frac{-\|x - y\|_2^2}{2\sigma^2} \right)$$ RBF or Gaussian kernel

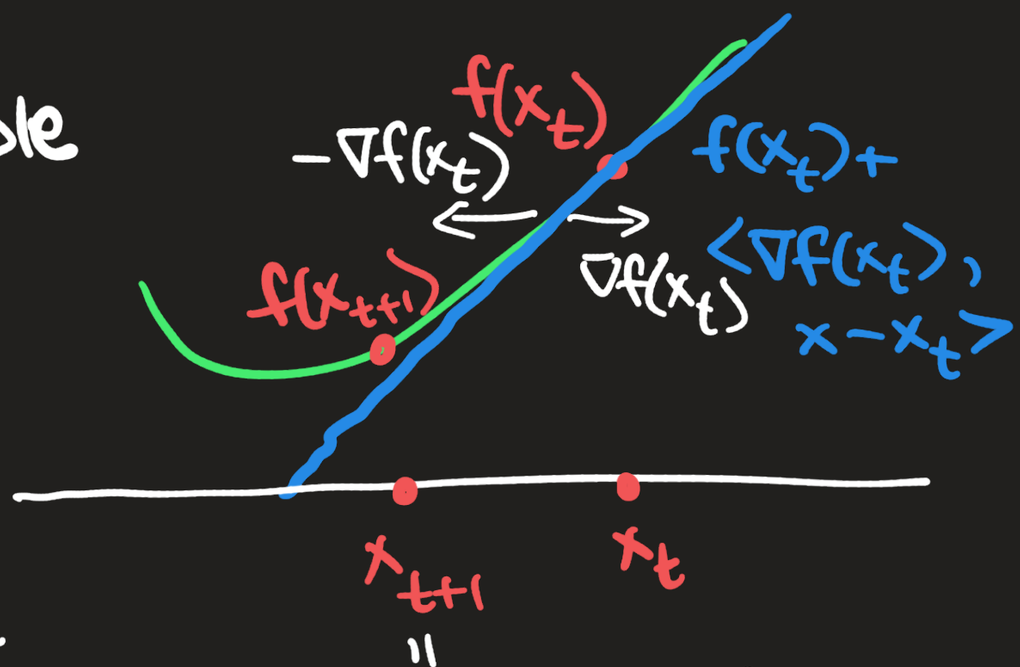
$$K(x, y) = \exp\left(\frac{-\|x - y\|_1}{\beta} \right)$$ Laplacian kernel

Gradient Descent

f – cvx objective, differentiable

want to solve

$$x_* \in \underset{x \in \mathbb{R}^d}{\operatorname{argmin}} f(x)$$



The basic idea of GD is to note that by T.S. expansion argument:

$$f(x_{t+1}) \approx f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle$$

$$= f(x_t) + \langle \nabla f(x_t), -\eta_t \nabla f(x_t) \rangle$$

$$= f(x_t) - \eta_t \|\nabla f(x_t)\|_2^2$$

$$< f(x_t)$$

accurate when η_t is small

GD Alg

For solving

$$(U) \quad \min_{x \in \mathbb{R}^d} f(x)$$ where f is smooth & convex

Input

∇f - gradient oracle

T - iterations

x_0 - starting guess

$\alpha_1, \dots, \alpha_T$ - step sizes for each iteration

Output

$x_1, \dots, x_T \in \mathbb{R}^d$ estimates of x^*

Alg

For $t = 1$ to T

$$x_t = x_{t-1} - \alpha_t \nabla f(x_{t-1})$$

For now we want to know GD converges under some reasonable conditions on f, that tell us:

- how to choose our stepsizes,
- how many steps T to take to reach approximate convergence

Three different notions of convergence

ϵ-suboptimality ↪ our focus for cvx optim

1) $f(x_t) < f(x^*) + \epsilon$ approx conv in function value

2) $\|x_t - x^*\|_2^2 < \epsilon$ " " " parameter

3) $\|\nabla f(x_t)\|_2^2 < \epsilon$ " " of gradient to zero

↑ ϵ-stationarity — usual convergence criterion for noncvx optim

All equivalent if f is strongly convex

A reasonable assumption for GD to work is that f is β-smooth:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq \beta \|x - y\|_2$$

i.e. ∇f is β-Lipschitz continuous

Ex of a function that is β-smooth

$$f(x) = \|Ax - b\|_2^2 \qquad \nabla f(x) = A^T(Ax - b)$$

$$\|\nabla f(x) - \nabla f(y)\|_2 = \|A^TA(x-y)\|_2 \leq \|A^TA\|_2 \|x-y\|_2$$

$$= \|A\|_2^2 \|x-y\|_2$$

So f is β-smooth where $\beta = \|A\|_2^2$

Ex of a function that is not β-smooth for any β

$$f(x) = e^x \Rightarrow f'(x) = e^x$$

is there a β s.t.

$$|f'(x) - f'(y)| \leq \beta |x-y| \quad \text{for all } x, y$$

$$\Leftrightarrow$$

$$|e^x - e^y| \leq \beta |x-y|$$

No, b/c by the mean value theorem

$$\frac{e^x - e^y}{x-y} = e^z \quad \text{for some } z \in [x, y]$$

notice $e^z \geq e^x$

$$\Rightarrow |e^x - e^y| \geq e^x |x-y|$$

so given a $\beta > 0$, take $x > \log \beta$, we have $|e^x - e^y| > \beta |x-y|$

Consequences of β-smoothness

1) $\|\nabla f(x)\|_2 \leq \beta \|x - x_*\|_2$

because

$$\|\nabla f(x) - \nabla f(x^*)\|_2 \leq \beta \|x - x_*\|_2$$

and $\nabla f(x^*) = 0$

2) This implies a concrete bound on the accuracy of the first order T.S. expansion around a point x:

$$f(y) \approx f(x) + \nabla f(x)^T (y - x)$$

β-smoothness implies

$$\left| f(y) - f(x) - \nabla f(x)^T (y - x) \right| \leq \frac{\beta}{2} \|x - y\|_2^2$$

Now assume f is β-smooth. Take $\alpha_t = \frac{1}{\beta}$

Descent Lemma with this choice for α_t, GD ensures that

$$f(x_{t+1}) - f(x_t) \leq -\frac{1}{2\beta} \|\nabla f(x_t)\|_2^2$$

Pf

Because f is β-smooth,

$$f(x_{t+1}) - f(x_t) - \langle \nabla f(x_t), x_{t+1} - x_t \rangle \leq \frac{\beta}{2} \|x_{t+1} - x_t\|_2^2.$$

Using $x_{t+1} = x_t - \frac{1}{\beta} \nabla f(x_t)$, this becomes

$$f(x_{t+1}) - f(x_t) + \frac{1}{\beta} \|\nabla f(x_t)\|_2^2 \leq \frac{\beta}{2} \| -\frac{1}{\beta} \nabla f(x_t) \|_2^2$$

or

$$f(x_{t+1}) - f(x_t) \leq -\frac{1}{2\beta} \|\nabla f(x_t)\|_2^2$$

This says that if f is β-smooth and $\alpha_t = \frac{1}{\beta}$ for each step, then the value of $f(x_t)$ will be monotonically decreasing. In fact, the following is true (see Bubeck Ch 3 for proof).

<u>Fact</u> If f is convex and β-smooth, then GD w/ constant stepsize $\alpha = \frac{1}{\beta}$ guarantees that

$$f(x_T) - f(x^*) \leq \frac{2\beta \|x^* - x_0\|_2^2}{T}$$

We say GD has <u>iteration complexity</u> $T = O\left(\frac{\beta}{\epsilon}\right)$ to get to a ϵ-suboptimal minimizer

Alternatively, GD has <u>convergence rate</u> $O\left(\frac{\beta}{T}\right)$

Iteration Complexity : how many iterations are needed to achieve $f(x_T) - f(x^*) \leq \epsilon$?

We want

$$f(x_T) - f(x^*) \leq \epsilon .$$

It suffices to take T to ensure

$$f(x_T) - f(x^*) \leq \frac{2\beta \|x_0 - x^*\|_2^2}{T} \leq \epsilon ,$$

i.e.

$$T = \frac{2\beta \|x_0 - x^*\|_2^2}{T} , \text{ so iteration complexity is } O\left(\frac{\beta}{T}\right)$$

Convergence rate : after T iterations, how small is

$$f(x_T) - f(x^*)$$

Since $f(x_T) - f(x^*) \leq \frac{2\beta \|x_0 - x^*\|_2^2}{T}$,

convergence rate is $O\left(\frac{\beta}{T}\right)$