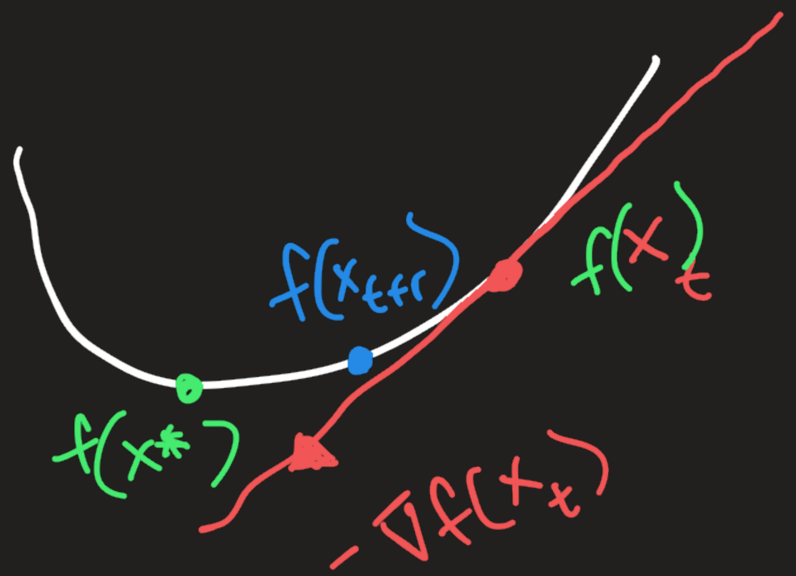
ML and Opt Lec 11

- Nonsmooth Cvx Optimization:
    - subdifferentials & subgradients
    - optimality conditions
- Gradient Descent & Projected Gradient Descent

Gradient Descent

f – cvx smooth objective
and we are doing unconstrained optimization



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

$$f(x_{t+1}) \approx f(x_t) + \nabla f(x_t)^T (x_{t+1} - x_t)$$

if x_{t+1} is close enough to x_t,

so if we want to decrease $f(x_{t+1})$, take

$x_{t+1} = x_t + \Delta$ and note

$$f(x_{t+1}) \approx f(x_t) + \nabla f(x_t)^T \Delta$$

Take $\Delta = -\alpha \nabla f(x_t)$, we have

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

Caveats

1) f needs to be "nice"
2) α needs to be small enough that the linear approx is accurate

Gradient Descent Alg

For solving $(U)\ \min_{x \in \mathbb{R}^d} f(x)$ 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 $t = 0$ to $T-1$

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

For now we want to show GD converges under some reasonable assumptions on f; that tells us:

- how to choose our stepsizes
- how many steps T to take to get approximate convergence.

ε - approximate convergence

$$f(x_T) \leq f(x^*) + \varepsilon$$

A reasonable assumption is that our function's gradient is at least Lipschitz-continuous:

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

Called β-smoothness.

Ex of a function that is β-smooth.

$$f(x) = \|x\|_2^2 \implies \nabla f(x) = 2x$$

$$\implies \|\nabla f(x) - \nabla f(y)\|_2 = 2\|x - y\|_2$$

so f is 2-smooth

Ex of a function that is not β-smooth

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

is there a β s.t. $\forall x, y$

$|\nabla f(x) - \nabla f(y)| = |e^x - e^y| \leq \beta |x-y|$?

Prf no such β exists

$e^x - e^y = e^z (x-y)$ for some $z \in [x, y]$

so for any β I can find x & $y = x+1$ so that

$\beta < e^x \leq e^z$ so

$|e^x - e^y| = |e^z||x-y| = |e^z| > \beta$

$\blacksquare$

Consequences of β-smoothness

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

(take $y = x^*$ in the def and note $\nabla f(y) = \nabla f(x^*) = 0$)

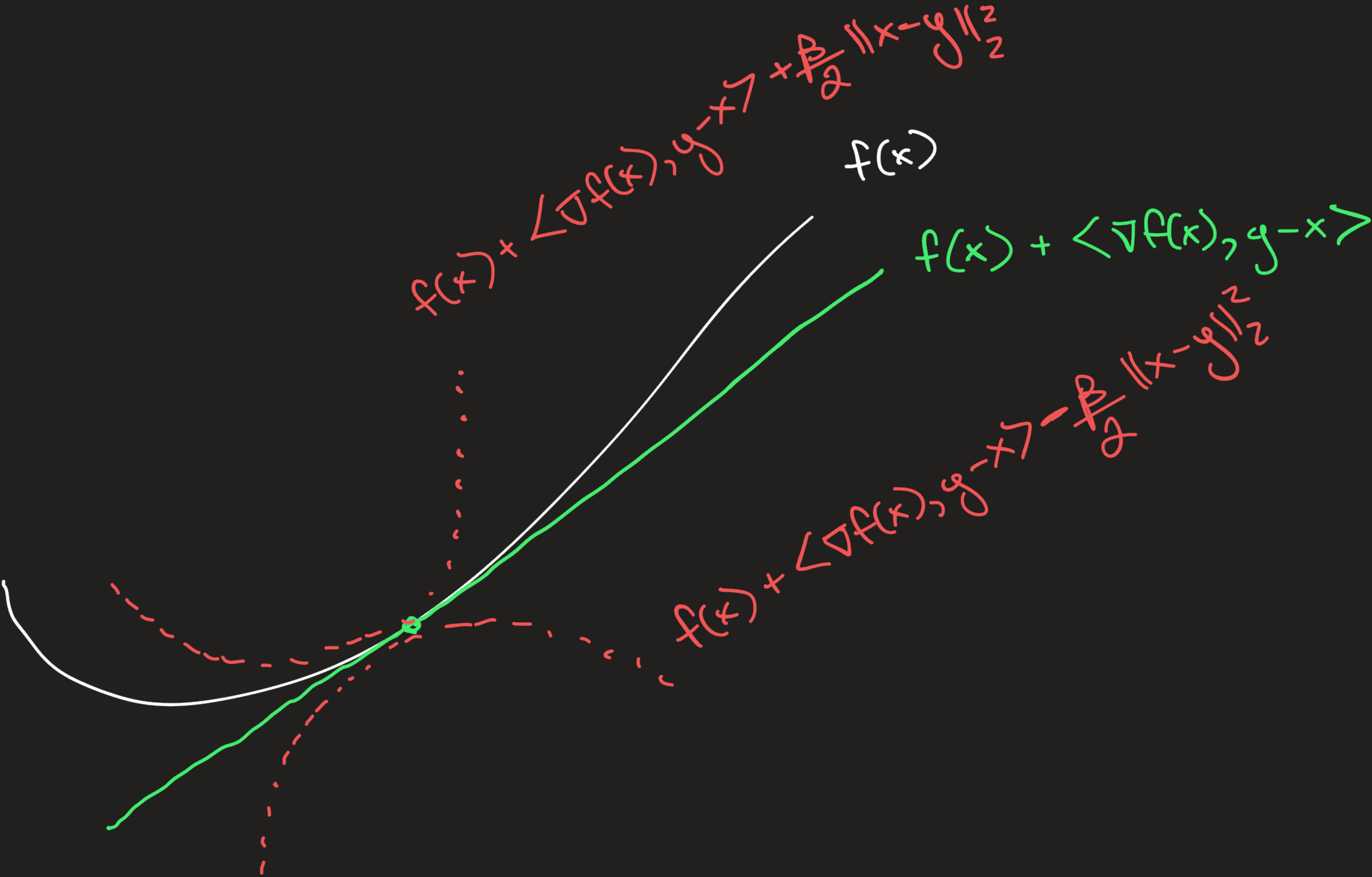
2) This implies a concrete bound on how good the approximation

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) \text{ is.}$$

Precisely,

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

Chapter 3 of Bubeck's Convex Optimization



$f(x) + \langle \nabla f(x), y-x \rangle + \frac{\beta}{2}\|x-y\|_2^2$
$f(x)$
$f(x) + \langle \nabla f(x), y-x \rangle$
$f(x) + \langle \nabla f(x), y-x \rangle - \frac{\beta}{2}\|x-y\|_2^2$

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

Then we see that

$$f\left(x_t - \frac{1}{\beta}\nabla f(x_t)\right) - \left[f(x_t) + \left\langle \nabla f(x_t), -\frac{1}{\beta}\nabla f(x_t)\right\rangle\right]$$

$$\leq \frac{\beta}{2}\left\|-\frac{1}{\beta}\nabla f(x_t)\right\|_2^2 = \frac{1}{2\beta}\|\nabla f(x_t)\|_2^2$$

and $\left\langle \nabla f(x_t), -\frac{1}{\beta}\nabla f(x_t)\right\rangle = -\frac{1}{\beta}\|\nabla f(x_t)\|_2^2$, so

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

so

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

gradient descent w/ constant step-size <u>always</u> decreases f.

Note: we did not use the fact that the function is convex

See Bubeck sec 3.2 for a full proof of this next result.

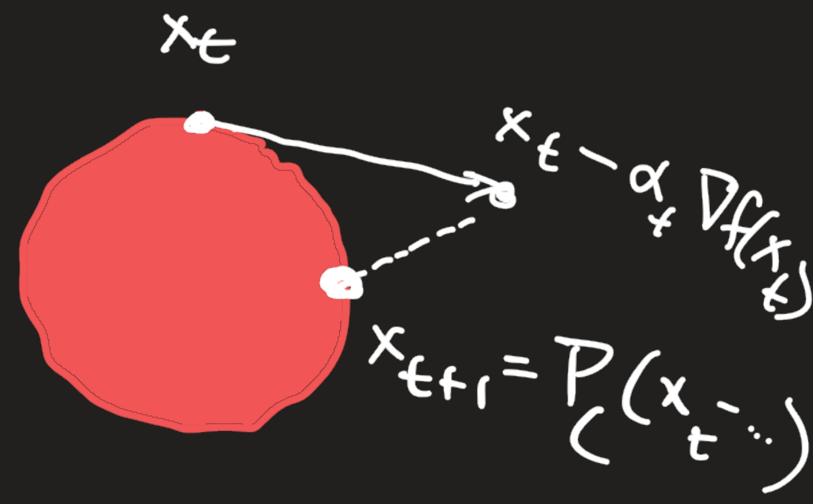
<u>Fact</u> If f is convex and β-smooth, then gradient descent with $\alpha_t = \frac{1}{\beta}$ guarantees that

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

We say GD takes $T = O\left(\frac{\beta}{\varepsilon}\right)$ iterations to get to an ε-approximate minimizer (<u>iteration complexity</u>)

Alternatively GD has convergence rate $O\left(\frac{\beta}{T}\right)$

What about constrained optimization?



Projected Gradient Descent

$$(\cup) \quad \min_{x \in C} f(x)$$

Inputs:

- ∇f
- T
- x_0
- $\alpha_1, \dots, \alpha_T$
- P_C : projection operator onto C

Outputs

$x_0, x_1, \dots, x_T$

Alg

for t in 0 to $T-1$:

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