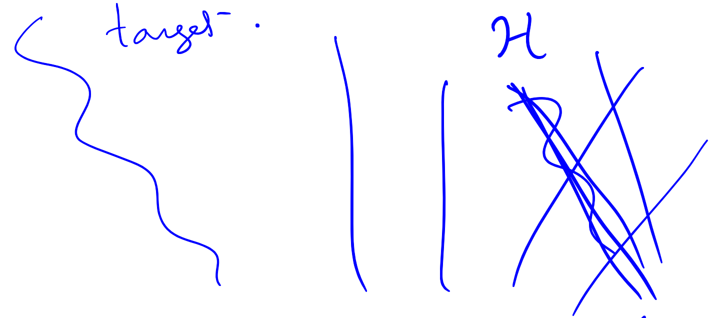
THEORY OF MACHINE LEARNING

LECTURE 9

INTRODUCTION TO OPTIMIZATION

SO FAR IN THE COURSE

$\{h : X \to \{0,1\}\}$.



$h : X \to \{0,1\}$.

- Reasoning about learning
  - Learning and generalization (distribution D)
  - Inductive bias and necessity of choosing a hypothesis class (H)
  - PAC learnability of a class (given examples, find hypothesis that is as good as the best in class); agnostic, improper

    improper → output of ALG is outside of $\mathcal{H}$.

    agnostic → target hypothesis does not belong to our class.

- Growth function of a class $\tau_H(m)$; small growth function => random sample is representative => class is learnable

  x_1, y_1 (x_2, y_2)

- Fundamental theorem: VC dimension "captures" learnability

  x

- Homework 1

  (next ~~Monday~~ Wednesday).

$X \subseteq \mathbb{R}^d$.

$x = (x_1, x_2, \ldots, x_d)$

$(x_1^2, x_2^2, \ldots,)$

FUNDAMENTAL THEOREM OF (STAT) LEARNING THEORY

- Theorem: The following statements are *equivalent:*
  - Class H is PAC learnable
  - Class H is *agnostically* PAC learnable
  - Class H has finite VC dimension

$\frac{d \log(d/\delta)}{\varepsilon^2}$ samples.

RECALL: IMPLICATIONS

- If H has infinite VC dimension, it is **<u>not</u>** PAC learnable! (same proof as no-free-lunch theorem – homework)
- Agnostic case usually as hard as *realizable case*
- **Caveat.** Generalization guarantees only apply to ERM, not (say) to an improper learner

assumes that you're doing proper learning.

- ERM is all you need, assuming you have enough samples
  - Doing ERM efficiently is a challenge
  - Loss + optimization is the standard approach

OPTIMIZATION

SOLVING ERM EFFICIENTLY

RECAP: ERM IS OFTEN HARD WITH FINITE ~~CLASSES~~

classification.

$X \subseteq \mathbb{R}^d$. $w \cdot x$ $\vec{w}$

- Finding best linear classifier (fewest mistakes) is NP hard even to approximate!

$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots$ $\mathcal{H}: \{ h = \text{sign}(\langle w, x \rangle + b) : w \in \mathbb{R}^d \}$

↳ Solving the ERM problem is NP-hard.

- Common remedy: loss functions

- Discrete optimization (mistake bounds) to continuous

[objective becomes continuous.].

$w = (w_1, w_2, \ldots, w_d)$

$x = (x_1, x_2, \ldots, x_d)$

$\langle w, x \rangle = w_1 x_1 + \ldots + w_d x_d$.

proxy for solving ERM

ERM minimizes # of mistakes → minimize total loss

LOSS MINIMIZATION

$\in \mathbb{R}^m$ $\{\pm 1\}$.

- Given training examples $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, find hypothesis h to minimize "loss"

ERM

$$\text{minimize } \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\left[h(x_i) \neq y_i\right]$$

Loss minimization.

$$\frac{1}{m}\sum_{i=1}^{m} \ell\left(h(x_i), y_i\right)$$

square loss $(y_i - h(x_i))^2$

ℓ_1-loss $|y_i - h(x_i)|$

$\log\left(\frac{1}{1+e^{-y_i h(x_i)}}\right)$

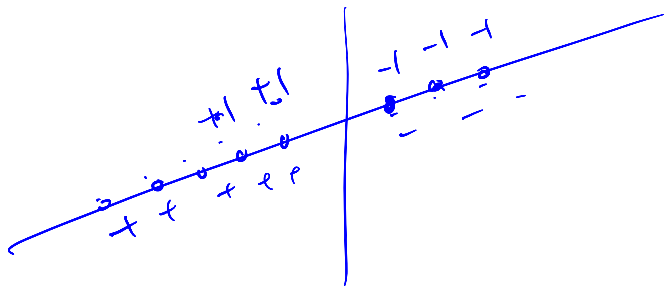
can be solved efficiently.

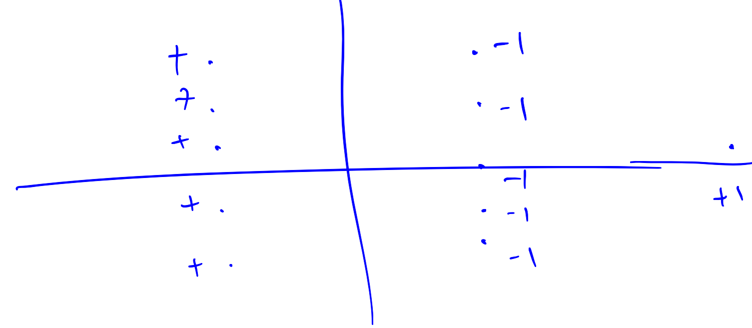
- Examples: square loss, l1 loss, logistic loss, …
- Parametrize h
- Not all loss functions can be minimized efficiently!

for linear classifiers: $h(x) = \langle w, x\rangle + b$

Square loss minimization: $\frac{1}{m}\sum_i (y_i - \langle w, x\rangle - b)^2$

find w, b.





Exercise: think about bad examples for different losses.

loss formulation does not faithfully capture ERM.

Turns out: Loss minimization is essentially way we can generally deal with ERM.

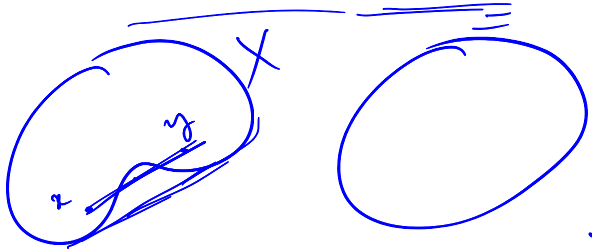
Optimization: general umbrella term for loss minimization.

CONVEX OPTIMIZATION

[Known to be "easy"].

- **Problem.** Given a convex function defined over a convex domain, find the minimizer (or min value). →

Convex domain $\mathcal{D} \subseteq \mathbb{R}^d$

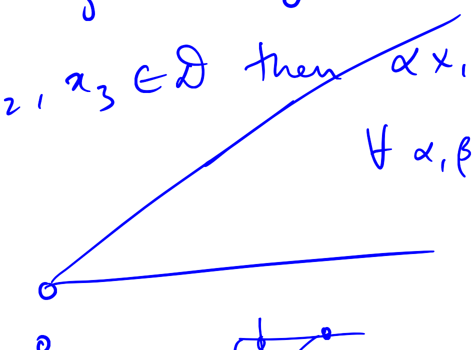


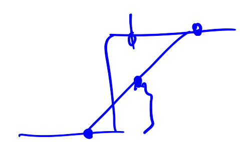
→ Formally, $\forall\, x, y$ in $\mathcal{D}$, the entire line segment $\overline{xy}$ belongs to $\mathcal{D}$.

if $x_1, x_2, x_3 \in \mathcal{D}$ then $\alpha x_1 + \beta x_2 + \gamma x_3 \in \mathcal{D}$

$\forall\, \alpha, \beta, \gamma \geq 0$, s.t. $\alpha + \beta + \gamma = 1$

- All of $\mathbb{R}^d$ is convex.
- Convex sets can be unbounded



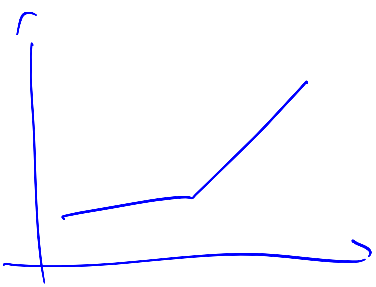


Convex function f :

$f:\mathbb{R}^d \to \mathbb{R}$.

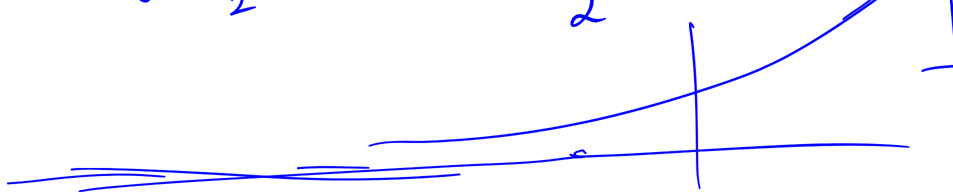
One-dim: $f:\mathbb{R}\to\mathbb{R}$, then $f''(x) \geq 0 \quad \forall x$ → criteria for Convexity.

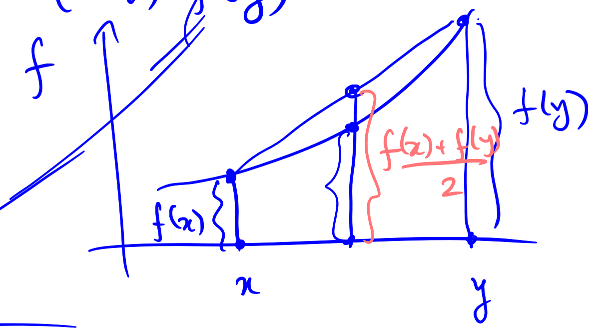
- Formal definition: f is convex if $\forall\, x, y \in \text{Domain}(f)$,



$$\rightarrow f\big(t x + (1-t)\, y\big) \leq t \cdot f(x) + (1-t)\, f(y) \quad \forall t \in [0,1].$$

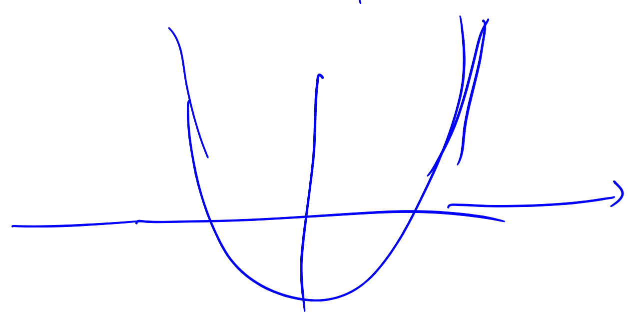
$$f\left(\frac{x+y}{2}\right) \leq \frac{f(x)+f(y)}{2}.$$





$f(x) = e^x$

$f(x) = x^2 - 1$



- Minimizing a convex fn f over convex domain $\mathcal{D}$ is "easy"

- Maximizing is not; $-f$ is not convex!

  max of a convex fn is always attained at a boundary.

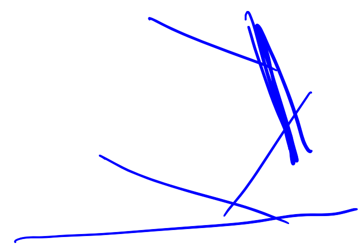
- For convex minimization, any local opt is a global opt.

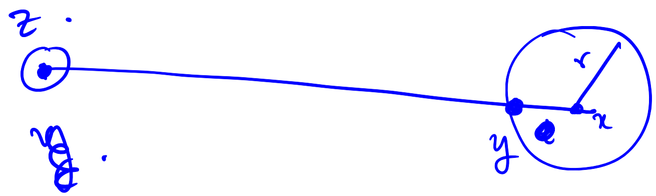
  - say $x \in \mathcal{D}$
  - know that $\exists$ a ball, i.e., radius val. $r > 0$

    s.t. $\forall\, y \in \mathrm{Ball}(x, r)$, $f(x) \le f(y)$

  local min.

  Then $\displaystyle\min_{z \in \mathcal{D}} f(z) = f(x)$.





Suppose $f(z) < f(x)$

$\exists$ $t > 0$ such that $y = tx + (1-t)z$.

$$f(y) \le t f(x) + (1-t) \underbrace{f(z)}_{< f(x)} < f(x)$$

This gives hope for "local search" methods.

- start at some x
- iteratively move to a "neighboring point" with smaller f value.

[if you're unable to find such a point to move to, then you have found local opt $\rightsquigarrow$ global opt.].

TAYLOR APPROXIMATION

- Functions over R^d, gradients, Hessian
- First order approximation

GRADIENT DESCENT ALGORITHM

- Very general approach

NATURAL ISSUES

- "Learning rate"
- Staying in the domain

GRADIENT DESCENT – VANILLA ANALYSIS

- Consider ‘f’ that is L-Lipschitz, fixed learning rate, domain all of R^d
- Suppose OPT was distance B away from initial point