15-780: Lecture 3

Aditi Raghunathan

Jan 22 2023

Recap

- We are building towards GPT4 and CLIP
- Supervised learning
  - Mapping input to targets
  - Notation: hypothesis function, hypothesis space, loss function
  - Minimizing training loss
  - ~~**Why does this work? Generalization (This lecture)**~~
  - How to minimize training loss? Optimization

h $\quad \mathcal{H}$

$h: \mathcal{X} \to \mathcal{Y} \quad \mathbb{R}^k$

single

$\boxed{\ell(h(x), y)}$

"cross entropy"

Minimize training loss

- Minimize **train loss**

$$\begin{array}{c} X_1, y_1 \\ X_2, y_2 \\ X_3, y_3 \\ \\ X_n, y_n \end{array} \longrightarrow \boxed{\textbf{Machine learning algorithm}} \longrightarrow f$$

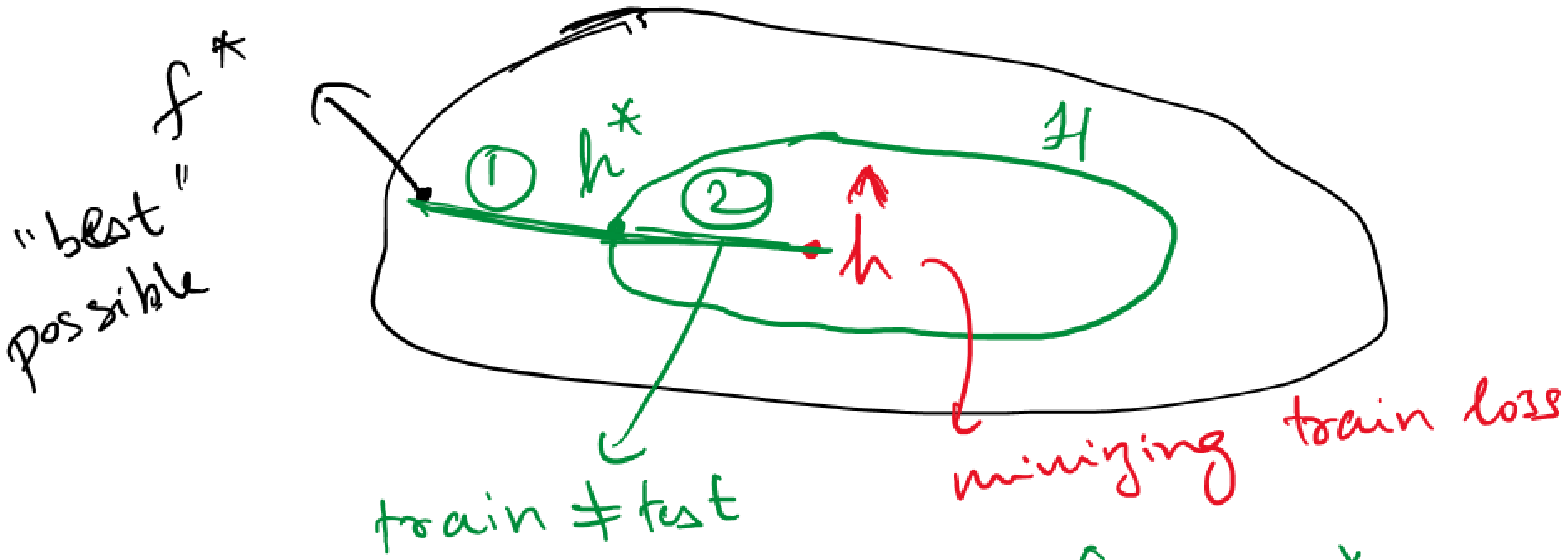
Training data

Test: new x, y

how well does f do on x, y

We care about good performance on unseen examples (test set)

Intuitive picture



① $h^* \neq f^*$

② $\hat{h} \neq h^*$

$\mathcal{H}$: set of hypothesis function h

$h: \mathcal{X} \to \mathbb{R}^k$ $\qquad \ell(h(x), y) \in \mathbb{R}$

p^*: underlying distribution over $\mathcal{X} \times \mathcal{Y}$

Train data: $x^{(i)}, y^{(i)} \overset{iid}{\sim} p^*$ (n of these)

Test data: $x, y \sim p^*$

Expected risk

$$L(h) = \mathbb{E}_{x,y \sim p^*}\left[\ell(h(x), y)\right] \Rightarrow \text{we care about this}$$

Expected risk minimizer

$$h^* \in \underset{\mathcal{H}}{\operatorname{argmin}} \; L(h)$$

"best possible hypothesis"

Empirical risk

$\Downarrow$ observed (training samples)

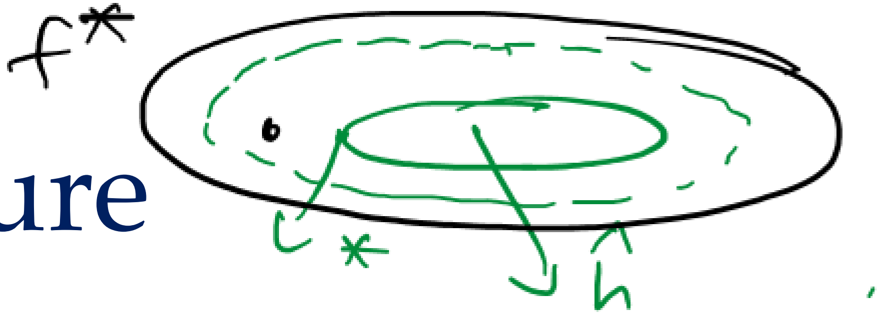
$$\hat{L}(h) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(h(x^{(i)}), y^{(i)}\right)$$

Empirical risk minimizer

$$\hat{h} \in \underset{\mathcal{H}}{\operatorname{argmin}} \ \hat{L}(h)$$

$$\boxed{\hat{h} \neq h^*}$$

Formalizing intuitive picture



- Approximation error

$$L(h^*) - L(f^*)$$

purely a fn of $\mathcal{H}$

- Estimation error

$$L(\hat{h}) - L(h^*)$$

"excess risk"

$$= L(\hat{h}) - L(f^*)$$

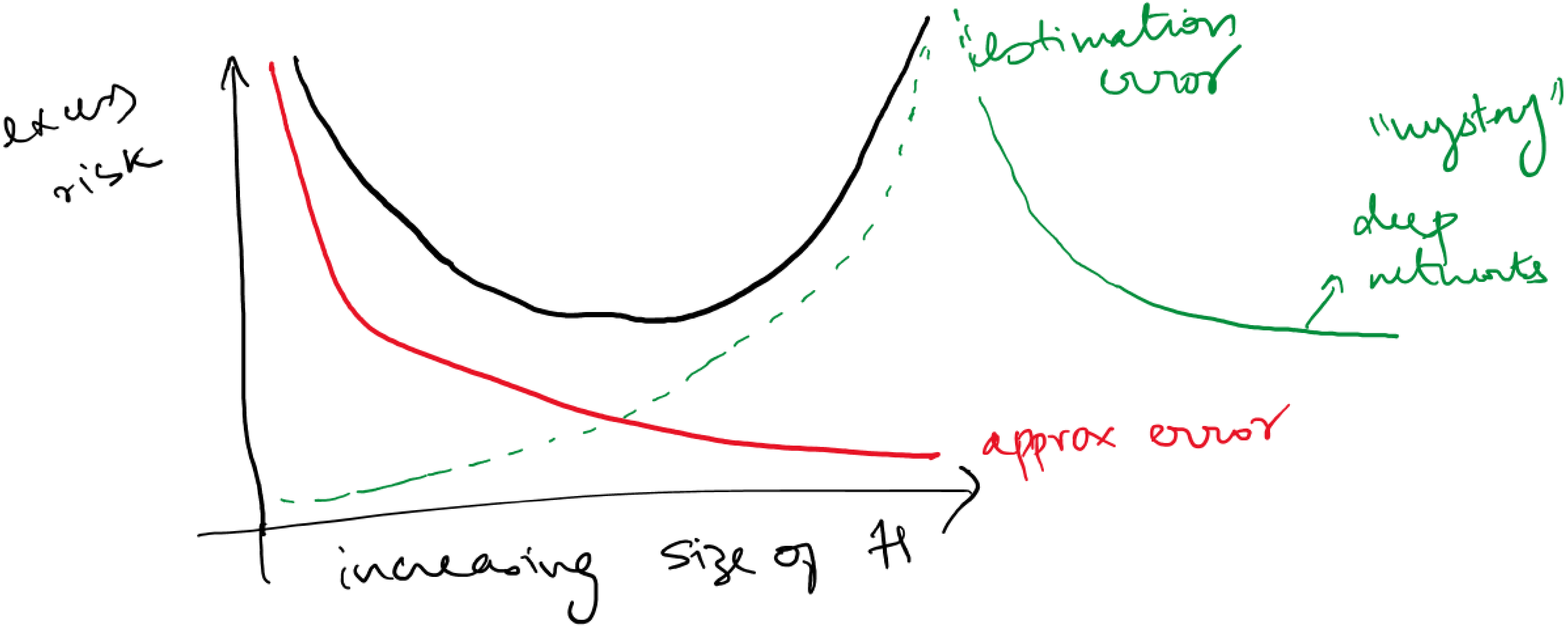
$$= \boxed{L(\hat{h}) - L(h^*)} + L(h^*) - L(f^*)$$

approx error

Piazza poll

$\mathcal{H} \subseteq \mathcal{H}'$

Effect of hypothesis class size



Understanding estimation error

estimation error : $L(\hat{h}) - L(h^*)$

$L(\hat{h})$: a random quantity

$$\mathbb{P}[\underbrace{L(\hat{h}) - L(h^*) > \epsilon}_{\text{estimation is high}}] < \delta$$

low probability

Simple realizable case

① $h: \mathcal{X} \to \{0,1\}$ "deterministic"

② loss function: zero-one error $\mathbb{1}[h(x) \neq y]$

③ $\mathcal{H}$ is finite

④ "realizable": $f^* = h^*$

$L(h^*) = 0$ $\quad \exists\ h^*$ s.t. $\forall x, y$

$h^*(x) = y$

$\Rightarrow \hat{L}(h^*) = 0 \Rightarrow \hat{L}(\hat{h}) = 0$

we want to bound $P[L(\hat{h}) - L(h^*) > \epsilon]$

$$L(\hat{h}) - \hat{L}(\hat{h}) + \hat{L}(\hat{h}) - L(h^*)$$

$\hat{h}$ is defined as minimizer of $\hat{L}$

$B : \{h \mid L(h) > \epsilon\}$ set of bad hypotheses

$$P[L(\hat{h}) - L(h^*) > \epsilon] = P[L(\hat{h}) > \epsilon]$$

0 (realizability)

$$= P[\hat{h} \in B]$$

$P[\hat{h} \in B]$ $\boxed{\hat{L}(\hat{h}) = 0}$

__Step one:__ If hypothesis which is bad $[L(h) > \epsilon]$ what is probability it has zero train loss

$n \to$ size of train set

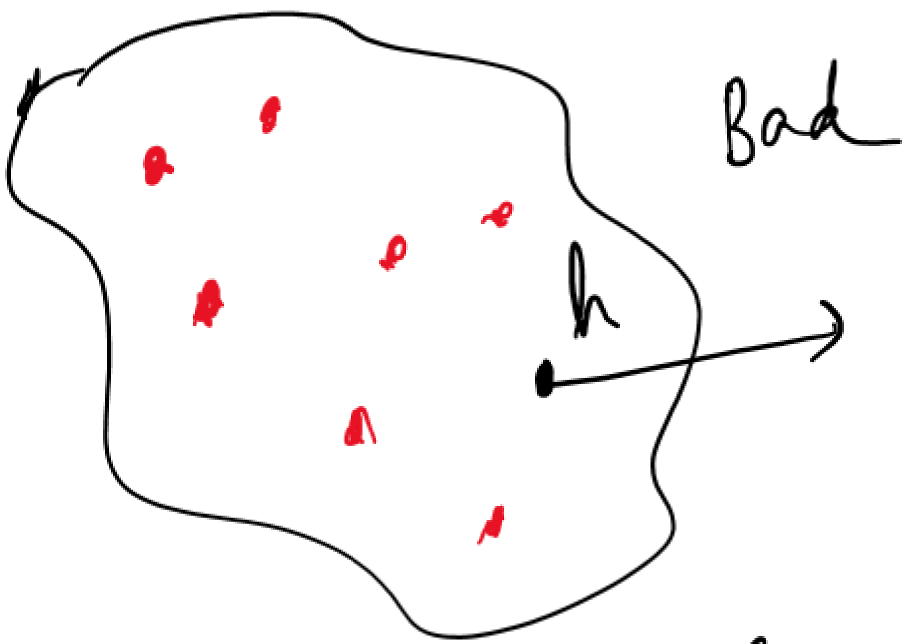
$$P[\hat{L}(h)=0] = (1-\epsilon)^{n} \le e^{-\epsilon n}$$

for $h \in B$

$1 - x \le e^{-x}$

ϵ mass where h is wrong

Is $P[\hat{h} \in B] \le (1-\epsilon)^{n}$?



"appears good" $(1-\epsilon)^n$

$\hat{h}$: any classifier that has zero train loss or appears good $P(\hat{h} \in B)$?

union bound : $P[\exists h \in B ; \hat{L}(h) = 0]$

$$\le \sum_{h \in B} P[\hat{L}(h) = 0] \le |B| e^{-\epsilon n} \le |H| e^{-\epsilon n}$$

$$P[\hat{h} \in B] \leq |H| e^{-\epsilon n} \quad \delta$$

estimation error $> \epsilon$

union Bound

appears good on n samples despite being bad

$$\text{w.p. } 1-\delta \quad L(\hat{h}) \leq \frac{\log(H) + \log(1/\delta)}{n}$$

excess risk = estimation error

as n increases, est error decreases

as (H) increases, est error increases

General recipe

test loss train loss

- **Convergence:** for fixed h, $L(h)$ is close to $\hat{L}(h)$ as $n \uparrow$, gap goes down
- **Uniform convergence:** convergence holds for all hypothesis simultaneously

makes it harder as $\mathcal{H}$ expands

- *Why uniform convergence?*

Takeaways

- Approximation error: decreases with increase in H
- Estimation error: more nuanced, depends on H
  - Very large H leads to high estimation error
- How to keep H small?

Regularization

- Linear classifiers: dimensionality, norm

$h(x) = \phi^T x \quad \phi \in \mathbb{R}^d \quad \|\phi\|$ small

- Regularized objective

objective: Train loss $+ \lambda \|\phi\|_2$

Any questions?