Introduction to Machine Learning (67577)
Lecture 3

Shai Shalev-Shwartz

School of CS and Engineering,
The Hebrew University of Jerusalem

General Learning Model and Bias-Complexity tradeoff

Outline

Relaxing the realizability assumption – Agnostic PAC learning

- So far we assumed that labels are generated by some $f \in \mathcal{H}$
- This assumption may be too strong
- Relax the realizability assumption by replacing the "target labeling function" with a more flexible notion, a data-labels generating distribution

Relaxing the realizability assumption – Agnostic PAC learning

- Recall: in PAC model, $\mathcal{D}$ is a distribution over $\mathcal{X}$

Relaxing the realizability assumption – Agnostic PAC learning

- Recall: in PAC model, $\mathcal{D}$ is a distribution over $\mathcal{X}$
- From now on, let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$

Relaxing the realizability assumption – Agnostic PAC learning

- Recall: in PAC model, $\mathcal{D}$ is a distribution over $\mathcal{X}$
- From now on, let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$
- We redefine the risk as:

$$L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \stackrel{\text{def}}{=} \mathcal{D}(\{(x,y) : h(x) \neq y\})$$

Relaxing the realizability assumption – Agnostic PAC learning

- Recall: in PAC model, $\mathcal{D}$ is a distribution over $\mathcal{X}$
- From now on, let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$
- We redefine the risk as:

$$L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \stackrel{\text{def}}{=} \mathcal{D}(\{(x,y) : h(x) \neq y\})$$

- We redefine the "approximately correct" notion to

$$L_{\mathcal{D}}(A(S)) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$$

PAC vs. Agnostic PAC learning

	PAC	Agnostic PAC
Distribution	$\mathcal{D}$ over $\mathcal{X}$	$\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$
Truth	$f \in \mathcal{H}$	not in class or doesn't exist
Risk	$L_{\mathcal{D},f}(h) = \mathcal{D}(\{x : h(x) \neq f(x)\})$	$L_{\mathcal{D}}(h) = \mathcal{D}(\{(x,y) : h(x) \neq y\})$
Training set	$(x_1,\ldots,x_m) \sim \mathcal{D}^m$ $\forall i,\ y_i = f(x_i)$	$((x_1,y_1),\ldots,(x_m,y_m)) \sim \mathcal{D}^m$
Goal	$L_{\mathcal{D},f}(A(S)) \leq \epsilon$	$L_{\mathcal{D}}(A(S)) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$

Beyond Binary Classification

Scope of learning problems:

- Multiclass categorization: $\mathcal{Y}$ is a finite set representing $|\mathcal{Y}|$ different classes. E.g. $\mathcal{X}$ is documents and $\mathcal{Y} = \{\text{News}, \text{Sports}, \text{Biology}, \text{Medicine}\}$
- Regression: $\mathcal{Y} = \mathbb{R}$. E.g. one wishes to predict a baby's birth weight based on ultrasound measures of his head circumference, abdominal circumference, and femur length.

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example, $(\mathbf{x}, y) \in Z$, how good is h on $(\mathbf{x}, y)$?

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example, $(\mathbf{x}, y) \in Z$, how good is h on $(\mathbf{x}, y)$?
- Loss function: $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example, $(\mathbf{x}, y) \in Z$, how good is h on $(\mathbf{x}, y)$?
- Loss function: $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$
- Examples:

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example, $(\mathbf{x}, y) \in Z$, how good is h on $(\mathbf{x}, y)$?
- Loss function: $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$
- Examples:
  - 0-1 loss: $\ell(h, (x, y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example, $(\mathbf{x}, y) \in Z$, how good is h on $(\mathbf{x}, y)$?
- Loss function: $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$
- Examples:
  - 0-1 loss: $\ell(h, (x, y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$
  - Squared loss: $\ell(h, (x, y)) = (h(x) - y)^2$

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example, $(\mathbf{x}, y) \in Z$, how good is h on $(\mathbf{x}, y)$?
- Loss function: $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$
- Examples:
  - 0-1 loss: $\ell(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$
  - Squared loss: $\ell(h,(x,y)) = (h(x) - y)^2$
  - Absolute-value loss: $\ell(h,(x,y)) = |h(x) - y|$

Loss Functions

- Let $Z = \mathcal{X} \times \mathcal{Y}$
- Given hypothesis $h \in \mathcal{H}$, and an example, $(\mathbf{x}, y) \in Z$, how good is h on $(\mathbf{x}, y)$?
- Loss function: $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$
- Examples:
  - 0-1 loss: $\ell(h, (x, y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$
  - Squared loss: $\ell(h, (x, y)) = (h(x) - y)^2$
  - Absolute-value loss: $\ell(h, (x, y)) = |h(x) - y|$
  - Cost-sensitive loss: $\ell(h, (x, y)) = C_{h(x),y}$ where C is some $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix

The General PAC Learning Problem

We wish to Probably Approximately Solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(h, z)] \; .$$

The General PAC Learning Problem

We wish to Probably Approximately Solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(h, z)] .$$

- Learner knows $\mathcal{H}$, Z, and ℓ

The General PAC Learning Problem

We wish to Probably Approximately Solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(h, z)] \ .$$

- Learner knows $\mathcal{H}$, Z, and ℓ
- Learner receives accuracy parameter ϵ and confidence parameter δ

The General PAC Learning Problem

We wish to Probably Approximately Solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(h, z)] \ .$$

- Learner knows $\mathcal{H}$, Z, and ℓ
- Learner receives accuracy parameter ϵ and confidence parameter δ
- Learner can decide on training set size m based on ϵ, δ

The General PAC Learning Problem

We wish to Probably Approximately Solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(h,z)] \ .$$

- Learner knows $\mathcal{H}$, Z, and ℓ
- Learner receives accuracy parameter ϵ and confidence parameter δ
- Learner can decide on training set size m based on ϵ, δ
- Learner doesn't know $\mathcal{D}$ but can sample $S \sim \mathcal{D}^m$

The General PAC Learning Problem

We wish to Probably Approximately Solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(h, z)] \; .$$

- Learner knows $\mathcal{H}$, Z, and ℓ
- Learner receives accuracy parameter ϵ and confidence parameter δ
- Learner can decide on training set size m based on ϵ, δ
- Learner doesn't know $\mathcal{D}$ but can sample $S \sim \mathcal{D}^m$
- Using S the learner outputs some hypothesis $A(S)$

The General PAC Learning Problem

We wish to Probably Approximately Solve:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \quad \text{where} \quad L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}} [\ell(h, z)] \ .$$

- Learner knows $\mathcal{H}$, Z, and ℓ
- Learner receives accuracy parameter ϵ and confidence parameter δ
- Learner can decide on training set size m based on ϵ, δ
- Learner doesn't know $\mathcal{D}$ but can sample $S \sim \mathcal{D}^m$
- Using S the learner outputs some hypothesis $A(S)$
- We want that with probability of at least $1 - \delta$ over the choice of S, the following would hold: $L_{\mathcal{D}}(A(S)) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$

Formal definition

A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$, if there exists a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm, A, with the following property: for every $\epsilon, \delta \in (0,1)$, $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, and distribution $\mathcal{D}$ over Z,

$$\mathcal{D}^m \left(\left\{ S \in Z^m \,:\, L_{\mathcal{D}}(A(S)) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \right\} \right) \;\geq\; 1 - \delta$$

Outline

Representative Sample

Definition (ϵ-representative sample)

A training set S is called ϵ-representative if

$$\forall h \in \mathcal{H}, \quad |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon \ .$$

Representative Sample

Lemma

Assume that a training set S is $\frac{\epsilon}{2}$-representative. Then, any output of $\mathrm{ERM}_{\mathcal{H}}(S)$, namely any $h_S \in \mathrm{argmin}_{h\in\mathcal{H}} L_S(h)$, satisfies

$$L_{\mathcal{D}}(h_S) \ \leq \ \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \ .$$

Representative Sample

Lemma

Assume that a training set S is $\frac{\epsilon}{2}$-representative. Then, any output of $\mathrm{ERM}_{\mathcal{H}}(S)$*, namely any* $h_S \in \mathrm{argmin}_{h \in \mathcal{H}} L_S(h)$*, satisfies*

$$L_{\mathcal{D}}(h_S) \;\leq\; \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \;.$$

Proof: For every $h \in \mathcal{H}$,

$$L_{\mathcal{D}}(h_S) \leq L_S(h_S) + \tfrac{\epsilon}{2} \leq L_S(h) + \tfrac{\epsilon}{2} \leq L_{\mathcal{D}}(h) + \tfrac{\epsilon}{2} + \tfrac{\epsilon}{2} = L_{\mathcal{D}}(h) + \epsilon$$

Uniform Convergence is Sufficient for Learnability

Definition (uniform convergence)

$\mathcal{H}$ has the *uniform convergence property* if there exists a function $m_{\mathcal{H}}^{\mathsf{UC}} : (0,1)^2 \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$, and every distribution $\mathcal{D}$,

$$\mathcal{D}^m \left(\{ S \in Z^m : S \text{ is } \epsilon \text{ -representative} \} \right) \geq 1 - \delta$$

Uniform Convergence is Sufficient for Learnability

Definition (uniform convergence)

$\mathcal{H}$ has the *uniform convergence property* if there exists a function $m_{\mathcal{H}}^{\mathsf{UC}} : (0,1)^2 \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$, and every distribution $\mathcal{D}$,

$$\mathcal{D}^m \left(\{ S \in Z^m : S \text{ is } \epsilon \text{ -representative} \} \right) \geq 1 - \delta$$

Corollary

- *If $\mathcal{H}$ has the uniform convergence property with a function $m_{\mathcal{H}}^{\mathsf{UC}}$ then $\mathcal{H}$ is agnostically PAC learnable with the sample complexity $m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{\mathsf{UC}}(\epsilon/2, \delta)$.*
- *Furthermore, in that case, the $\mathrm{ERM}_{\mathcal{H}}$ paradigm is a successful agnostic PAC learner for $\mathcal{H}$.*

Finite Classes are Agnostic PAC Learnable

We will prove the following:

Theorem

Assume $\mathcal{H}$ is finite and the range of the loss function is $[0,1]$. Then, $\mathcal{H}$ is agnostically PAC learnable using the $\mathrm{ERM}_{\mathcal{H}}$ algorithm with sample complexity

$$m_{\mathcal{H}}(\epsilon,\delta) \le \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil .$$

Finite Classes are Agnostic PAC Learnable

We will prove the following:

Theorem

Assume $\mathcal{H}$ is finite and the range of the loss function is $[0,1]$. Then, $\mathcal{H}$ is agnostically PAC learnable using the $\mathrm{ERM}_{\mathcal{H}}$ algorithm with sample complexity

$$m_{\mathcal{H}}(\epsilon,\delta) \le \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil .$$

Proof: It suffices to show that $\mathcal{H}$ has the uniform convergence property with

$$m_{\mathcal{H}}^{\mathsf{UC}}(\epsilon,\delta) \le \left\lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2} \right\rceil .$$

Proof (cont.)

- To show uniform convergence, we need:

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) < \delta \ .$$

Proof (cont.)

- To show uniform convergence, we need:

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) < \delta \ .$$

- Using the union bound:

$$\begin{aligned} &\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \ = \\ &\mathcal{D}^m(\cup_{h\in\mathcal{H}}\{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \ \leq \\ &\sum_{h\in\mathcal{H}} \mathcal{D}^m(\{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \ . \end{aligned}$$

Proof (cont.)

- Recall: $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^{m}\ell(h,z_i)$.

Proof (cont.)

- Recall: $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^{m} \ell(h,z_i)$.
- Denote $\theta_i = \ell(h, z_i)$.

Proof (cont.)

- Recall: $L_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$ and $L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i)$.
- Denote $\theta_i = \ell(h, z_i)$.
- Then, for all i, $\mathbb{E}[\theta_i] = L_{\mathcal{D}}(h)$

Proof (cont.)

- Recall: $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^m \ell(h,z_i)$.
- Denote $\theta_i = \ell(h,z_i)$.
- Then, for all i, $\mathbb{E}[\theta_i] = L_{\mathcal{D}}(h)$

Lemma (Hoeffding's inequality)

Let $\theta_1,\ldots,\theta_m$ be a sequence of i.i.d. random variables and assume that for all i, $\mathbb{E}[\theta_i] = \mu$ and $\mathbb{P}[a \le \theta_i \le b] = 1$. Then, for any $\epsilon > 0$

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^m \theta_i - \mu\right| > \epsilon\right] \;\le\; 2\exp\left(-2\,m\,\epsilon^2/(b-a)^2\right) \;.$$

Proof (cont.)

- Recall: $L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$ and $L_S(h) = \frac{1}{m}\sum_{i=1}^m \ell(h,z_i)$.
- Denote $\theta_i = \ell(h,z_i)$.
- Then, for all i, $\mathbb{E}[\theta_i] = L_{\mathcal{D}}(h)$

Lemma (Hoeffding's inequality)

Let $\theta_1,\ldots,\theta_m$ be a sequence of i.i.d. random variables and assume that for all i, $\mathbb{E}[\theta_i] = \mu$ and $\mathbb{P}[a \le \theta_i \le b] = 1$. Then, for any $\epsilon > 0$

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^m \theta_i - \mu\right| > \epsilon\right] \;\le\; 2\,\exp\left(-2\,m\,\epsilon^2/(b-a)^2\right) \;.$$

This implies:

$$\mathcal{D}^m(\{S : |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \;\le\; 2\,\exp\left(-2\,m\,\epsilon^2\right) \;.$$

Proof (cont.)

We have shown:

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) \;\leq\; 2\,|\mathcal{H}|\,\exp\left(-2\,m\,\epsilon^2\right)$$

So, if $m \geq \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2}$ then the right-hand side is at most δ as required.

□

The Discretization Trick

- Suppose $\mathcal{H}$ is parameterized by d numbers
- Suppose we are happy with a representation of each number using b bits (say, $b = 32$)
- Then $|\mathcal{H}| \leq 2^{db}$, and so

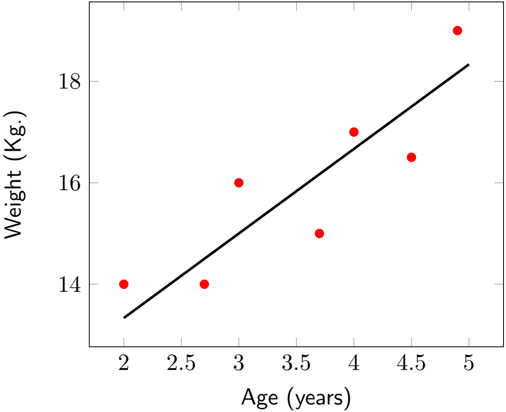
$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{2db + 2\log(2/\delta)}{\epsilon^2} \right\rceil .$$

- While not very elegant, it's a great tool for upper bounding sample complexity

Outline

Linear Regression

- $\mathcal{X} \subset \mathbb{R}^d$, $\mathcal{Y} \subset \mathbb{R}$, $\mathcal{H} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \mathbf{w} \in \mathbb{R}^d\}$
- Example: $d = 1$, predict weight of a child based on his age.



The Squared Loss

- Zero-one loss doesn't make sense in regression
- Squared loss: $\ell(h,(\mathbf{x},y)) = (h(\mathbf{x}) - y)^2$
- The ERM problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m}(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i)^2$$

- Equivalently, suppose X is a matrix whose ith column is $\mathbf{x}_i$, and $\mathbf{y}$ is a vector with y_i on its ith entry, then

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|X^\top \mathbf{w} - \mathbf{y}\|^2$$

Background: Gradient and Optimization

- Given a function $f : \mathbb{R} \to \mathbb{R}$, its derivative is

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x + \Delta) - f(x)}{\Delta}$$

Background: Gradient and Optimization

- Given a function $f : \mathbb{R} \to \mathbb{R}$, its derivative is

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}$$

- If x minimizes $f(x)$ then $f'(x) = 0$

Background: Gradient and Optimization

- Given a function $f : \mathbb{R} \to \mathbb{R}$, its derivative is

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x + \Delta) - f(x)}{\Delta}$$

- If x minimizes $f(x)$ then $f'(x) = 0$
- Now take $f : \mathbb{R}^d \to \mathbb{R}$

Background: Gradient and Optimization

- Given a function $f : \mathbb{R} \to \mathbb{R}$, its derivative is

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}$$

- If x minimizes $f(x)$ then $f'(x) = 0$
- Now take $f : \mathbb{R}^d \to \mathbb{R}$
- Its gradient is a d-dimensional vector, $\nabla f(\mathbf{x})$, where the ith coordinate of $\nabla f(\mathbf{x})$ is the derivative of the scalar function $g(a) = f((x_1, \ldots, x_{i-1}, x_i + a, x_{i+1}, \ldots, x_d))$.

Background: Gradient and Optimization

- Given a function $f : \mathbb{R} \to \mathbb{R}$, its derivative is

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}$$

- If x minimizes $f(x)$ then $f'(x) = 0$
- Now take $f : \mathbb{R}^d \to \mathbb{R}$
- Its gradient is a d-dimensional vector, $\nabla f(\mathbf{x})$, where the ith coordinate of $\nabla f(\mathbf{x})$ is the derivative of the scalar function $g(a) = f((x_1, \ldots, x_{i-1}, x_i + a, x_{i+1}, \ldots, x_d))$.
- The derivative of g is called the partial derivative of f

Background: Gradient and Optimization

- Given a function $f : \mathbb{R} \to \mathbb{R}$, its derivative is

$$f'(x) = \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}$$

- If x minimizes $f(x)$ then $f'(x) = 0$
- Now take $f : \mathbb{R}^d \to \mathbb{R}$
- Its gradient is a d-dimensional vector, $\nabla f(\mathbf{x})$, where the ith coordinate of $\nabla f(\mathbf{x})$ is the derivative of the scalar function $g(a) = f((x_1, \ldots, x_{i-1}, x_i + a, x_{i+1}, \ldots, x_d))$.
- The derivative of g is called the partial derivative of f
- If $\mathbf{x}$ minimizes $f(\mathbf{x})$ then $\nabla f(\mathbf{x}) = (0, \ldots, 0)$

Background: Jacobian and the chain rule

- The Jacobian of $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ at $\mathbf{x} \in \mathbb{R}^n$, denoted $J_{\mathbf{x}}(\mathbf{f})$, is the $m \times n$ matrix whose i, j element is the partial derivative of $f_i : \mathbb{R}^n \to \mathbb{R}$ w.r.t. its j'th variable at $\mathbf{x}$

Background: Jacobian and the chain rule

- The Jacobian of $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ at $\mathbf{x} \in \mathbb{R}^n$, denoted $J_{\mathbf{x}}(\mathbf{f})$, is the $m \times n$ matrix whose i, j element is the partial derivative of $f_i : \mathbb{R}^n \to \mathbb{R}$ w.r.t. its j'th variable at $\mathbf{x}$
- Note: if $m = 1$ then $J_{\mathbf{x}}(f) = \nabla f(\mathbf{x})$ (as a row vector)

Background: Jacobian and the chain rule

- The Jacobian of $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ at $\mathbf{x} \in \mathbb{R}^n$, denoted $J_{\mathbf{x}}(\mathbf{f})$, is the $m \times n$ matrix whose i, j element is the partial derivative of $f_i : \mathbb{R}^n \to \mathbb{R}$ w.r.t. its j'th variable at $\mathbf{x}$
- Note: if $m = 1$ then $J_{\mathbf{x}}(f) = \nabla f(\mathbf{x})$ (as a row vector)
- Example: If $\mathbf{f}(\mathbf{w}) = A\mathbf{w}$ for $A \in \mathbb{R}^{m,n}$ then $J_{\mathbf{w}}(\mathbf{f}) = A$

Background: Jacobian and the chain rule

- The Jacobian of $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ at $\mathbf{x} \in \mathbb{R}^n$, denoted $J_{\mathbf{x}}(\mathbf{f})$, is the $m \times n$ matrix whose i, j element is the partial derivative of $f_i : \mathbb{R}^n \to \mathbb{R}$ w.r.t. its j'th variable at $\mathbf{x}$
- Note: if $m = 1$ then $J_{\mathbf{x}}(f) = \nabla f(\mathbf{x})$ (as a row vector)
- Example: If $\mathbf{f}(\mathbf{w}) = A\mathbf{w}$ for $A \in \mathbb{R}^{m,n}$ then $J_{\mathbf{w}}(\mathbf{f}) = A$
- Chain rule: Given $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ and $\mathbf{g} : \mathbb{R}^k \to \mathbb{R}^n$, the Jacobian of the composition function, $(\mathbf{f} \circ \mathbf{g}) : \mathbb{R}^k \to \mathbb{R}^m$, at $\mathbf{x}$, is

$$J_{\mathbf{x}}(\mathbf{f} \circ \mathbf{g}) = J_{g(\mathbf{x})}(\mathbf{f}) J_{\mathbf{x}}(\mathbf{g}) \ .$$

Least Squares

- Recall that we'd like to solve the ERM problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|X^\top \mathbf{w} - \mathbf{y}\|^2$$

Least Squares

- Recall that we'd like to solve the ERM problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|X^\top \mathbf{w} - \mathbf{y}\|^2$$

- Let $\mathbf{g}(\mathbf{w}) = X^\top \mathbf{w} - \mathbf{y}$ and $\mathbf{f}(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}\|^2 = \sum_{i=1}^m v_i^2$

Least Squares

- Recall that we'd like to solve the ERM problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \|X^\top \mathbf{w} - \mathbf{y}\|^2$$

- Let $\mathbf{g}(\mathbf{w}) = X^\top \mathbf{w} - \mathbf{y}$ and $\mathbf{f}(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}\|^2 = \sum_{i=1}^m v_i^2$
- Then, we need to solve $\min_{\mathbf{w}} \mathbf{f}(\mathbf{g}(\mathbf{w}))$

Least Squares

- Recall that we'd like to solve the ERM problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|X^\top \mathbf{w} - \mathbf{y}\|^2$$

- Let $\mathbf{g}(\mathbf{w}) = X^\top \mathbf{w} - \mathbf{y}$ and $\mathbf{f}(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}\|^2 = \sum_{i=1}^m v_i^2$
- Then, we need to solve $\min_{\mathbf{w}} \mathbf{f}(\mathbf{g}(\mathbf{w}))$
- Note that $J_{\mathbf{w}}(\mathbf{g}) = X^\top$ and $J_{\mathbf{v}}(\mathbf{f}) = (v_1, \ldots, v_m)$

Least Squares

- Recall that we'd like to solve the ERM problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|X^\top\mathbf{w}-\mathbf{y}\|^2$$

- Let $\mathbf{g}(\mathbf{w}) = X^\top\mathbf{w}-\mathbf{y}$ and $\mathbf{f}(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}\|^2 = \sum_{i=1}^m v_i^2$
- Then, we need to solve $\min_{\mathbf{w}} \mathbf{f}(\mathbf{g}(\mathbf{w}))$
- Note that $J_{\mathbf{w}}(\mathbf{g}) = X^\top$ and $J_{\mathbf{v}}(\mathbf{f}) = (v_1,\ldots,v_m)$
- Using the chain rule:

$$J_{\mathbf{w}}(\mathbf{f}\circ\mathbf{g}) = J_{g(\mathbf{w})}(\mathbf{f})J_{\mathbf{w}}(\mathbf{g}) = \mathbf{g}(\mathbf{w})^\top X^\top = (X^\top\mathbf{w}-\mathbf{y})^\top X^\top$$

Least Squares

- Recall that we'd like to solve the ERM problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|X^\top \mathbf{w} - \mathbf{y}\|^2$$

- Let $\mathbf{g}(\mathbf{w}) = X^\top \mathbf{w} - \mathbf{y}$ and $\mathbf{f}(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}\|^2 = \sum_{i=1}^m v_i^2$
- Then, we need to solve $\min_{\mathbf{w}} \mathbf{f}(\mathbf{g}(\mathbf{w}))$
- Note that $J_{\mathbf{w}}(\mathbf{g}) = X^\top$ and $J_{\mathbf{v}}(\mathbf{f}) = (v_1, \ldots, v_m)$
- Using the chain rule:

$$J_{\mathbf{w}}(\mathbf{f} \circ \mathbf{g}) = J_{g(\mathbf{w})}(\mathbf{f}) J_{\mathbf{w}}(\mathbf{g}) = \mathbf{g}(\mathbf{w})^\top X^\top = (X^\top \mathbf{w} - \mathbf{y})^\top X^\top$$

- Requiring that $J_{\mathbf{w}}(\mathbf{f} \circ \mathbf{g}) = (0, \ldots, 0)$ yields

$$(X^\top \mathbf{w} - \mathbf{y})^\top X^\top = \mathbf{0}^\top \quad \Rightarrow \quad X X^\top \mathbf{w} = X\mathbf{y} \ .$$

Least Squares

- Recall that we'd like to solve the ERM problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\|X^\top \mathbf{w} - \mathbf{y}\|^2$$

- Let $\mathbf{g}(\mathbf{w}) = X^\top \mathbf{w} - \mathbf{y}$ and $\mathbf{f}(\mathbf{v}) = \frac{1}{2}\|\mathbf{v}\|^2 = \sum_{i=1}^m v_i^2$
- Then, we need to solve $\min_{\mathbf{w}} \mathbf{f}(\mathbf{g}(\mathbf{w}))$
- Note that $J_{\mathbf{w}}(\mathbf{g}) = X^\top$ and $J_{\mathbf{v}}(\mathbf{f}) = (v_1, \ldots, v_m)$
- Using the chain rule:

$$J_{\mathbf{w}}(\mathbf{f}\circ\mathbf{g}) = J_{g(\mathbf{w})}(\mathbf{f}) J_{\mathbf{w}}(\mathbf{g}) = \mathbf{g}(\mathbf{w})^\top X^\top = (X^\top \mathbf{w} - \mathbf{y})^\top X^\top$$

- Requiring that $J_{\mathbf{w}}(\mathbf{f}\circ\mathbf{g}) = (0, \ldots, 0)$ yields

$$(X^\top \mathbf{w} - \mathbf{y})^\top X^\top = \mathbf{0}^\top \quad \Rightarrow \quad XX^\top \mathbf{w} = X\mathbf{y} \ .$$

- This is a linear set of equations. If $XX^\top$ is invertible, the solution is

$$\mathbf{w} = (XX^\top)^{-1} X\mathbf{y} \ .$$

Least Squares

- What if $XX^\top$ is not invertible ?
- In the exercise you'll see that there's always a solution to the set of linear equations using pseudo-inverse

Least Squares

- What if $XX^\top$ is not invertible ?
- In the exercise you'll see that there's always a solution to the set of linear equations using pseudo-inverse

Non-rigorous trick to help remembering the formula:

- We want $X^\top \mathbf{w} \approx \mathbf{y}$
- Multiply both sides by X to obtain $XX^\top \mathbf{w} \approx X\mathbf{y}$
- Multiply both sides by $(XX^\top)^{-1}$ to obtain the formula:

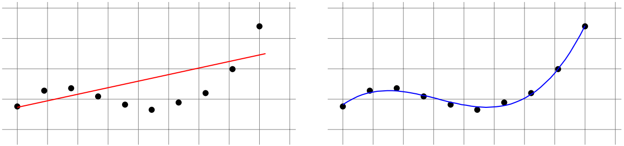
$$\mathbf{w} = (XX^\top)^{-1} X\mathbf{y}$$

Least Squares — Interpretation as projection

- Recall, we try to minimize $\|X^\top \mathbf{w} - \mathbf{y}\|$
- The set $C = \{X^\top \mathbf{w} : \mathbf{w} \in \mathbb{R}^d\} \subset \mathbb{R}^m$ is a linear subspace, forming the range of $X^\top$
- Therefore, if $\mathbf{w}$ is the least squares solution, then the vector $\hat{\mathbf{y}} = X^\top \mathbf{w}$ is the vector in C which is closest to $\mathbf{y}$.
- This is called the projection of $\mathbf{y}$ onto C
- We can find $\hat{\mathbf{y}}$ by taking V to be an $m \times d$ matrix whose columns are orthonormal basis of the range of $X^\top$, and then setting $\hat{\mathbf{y}} = VV^\top \mathbf{y}$

Polynomial Fitting

- Sometimes, linear predictors are not expressive enough for our data
- We will show how to fit a polynomial to the data using linear regression



Polynomial Fitting

- A one-dimensional polynomial function of degree n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

Polynomial Fitting

- A one-dimensional polynomial function of degree n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

- Goal: given data $S = ((x_1, y_1), \ldots, (x_m, y_m))$ find ERM with respect to the class of polynomials of degree n

Polynomial Fitting

- A one-dimensional polynomial function of degree n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

- Goal: given data $S = ((x_1, y_1), \ldots, (x_m, y_m))$ find ERM with respect to the class of polynomials of degree n
- Reduction to linear regression:

Polynomial Fitting

- A one-dimensional polynomial function of degree n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

- Goal: given data $S = ((x_1, y_1), \ldots, (x_m, y_m))$ find ERM with respect to the class of polynomials of degree n
- Reduction to linear regression:
- Define $\psi : \mathbb{R} \to \mathbb{R}^{n+1}$ by $\psi(x) = (1, x, x^2, \ldots, x^n)$

Polynomial Fitting

- A one-dimensional polynomial function of degree n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

- Goal: given data $S = ((x_1, y_1), \ldots, (x_m, y_m))$ find ERM with respect to the class of polynomials of degree n
- Reduction to linear regression:
- Define $\psi : \mathbb{R} \to \mathbb{R}^{n+1}$ by $\psi(x) = (1, x, x^2, \ldots, x^n)$
- Define $\mathbf{a} = (a_0, a_1, \ldots, a_n)$ and observe:

$$p(x) = \sum_{i=0}^{n} a_i x^i = \langle \mathbf{a}, \psi(x) \rangle$$

Polynomial Fitting

- A one-dimensional polynomial function of degree n:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

- Goal: given data $S = ((x_1, y_1), \ldots, (x_m, y_m))$ find ERM with respect to the class of polynomials of degree n
- Reduction to linear regression:
- Define $\psi : \mathbb{R} \to \mathbb{R}^{n+1}$ by $\psi(x) = (1, x, x^2, \ldots, x^n)$
- Define $\mathbf{a} = (a_0, a_1, \ldots, a_n)$ and observe:

$$p(x) = \sum_{i=0}^{n} a_i x^i = \langle \mathbf{a}, \psi(x) \rangle$$

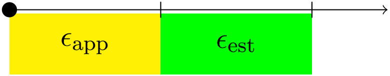
- To find $\mathbf{a}$, we can solve Least Squares w.r.t. $((\psi(x_1), y_1), \ldots, (\psi(x_m), y_m))$

Outline

Error Decomposition

- Let $h_S = \mathrm{ERM}_{\mathcal{H}}(S)$. We can decompose the risk of h_S as:

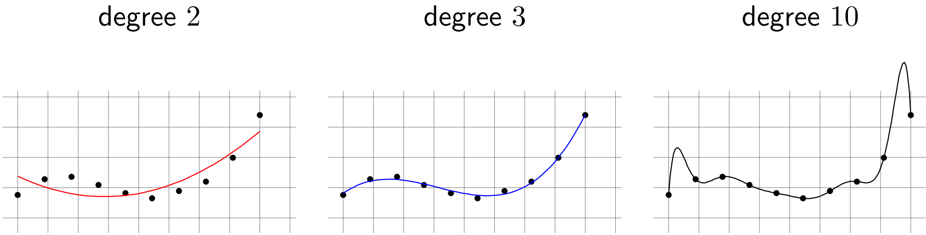
$$L_{\mathcal{D}}(h_S) = \epsilon_{\mathrm{app}} + \epsilon_{\mathrm{est}}$$



- **The approximation error,** $\epsilon_{\mathrm{app}} = \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$:
  - How much risk do we have due to restricting to $\mathcal{H}$
  - Doesn't depend on S
  - Decreases with the complexity (size, or VC dimension) of $\mathcal{H}$
- **The estimation error,** $\epsilon_{\mathrm{est}} = L_{\mathcal{D}}(h_S) - \epsilon_{\mathrm{app}}$:
  - Result of L_S being only an estimate of $L_{\mathcal{D}}$
  - Decreases with the size of S
  - Increases with the complexity of $\mathcal{H}$

Bias-Complexity Tradeoff

- How to choose $\mathcal{H}$?



Outline

Validation

- We have already learned some hypothesis h

Validation

- We have already learned some hypothesis h
- Now we want to estimate how good is h

Validation

- We have already learned some hypothesis h
- Now we want to estimate how good is h
- Simple solution: Take "fresh" i.i.d. sample $V = (x_1, y_1), \ldots, (x_{m_v}, y_{m_v})$

Validation

- We have already learned some hypothesis h
- Now we want to estimate how good is h
- Simple solution: Take "fresh" i.i.d. sample $V = (x_1, y_1), \ldots, (x_{m_v}, y_{m_v})$
- Output $L_V(h)$ as an estimator of $L_{\mathcal{D}}(h)$

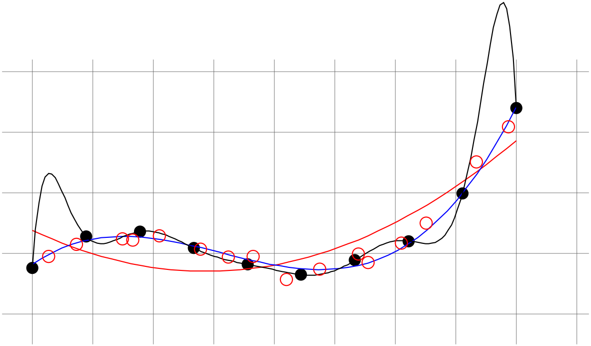
Validation

- We have already learned some hypothesis h
- Now we want to estimate how good is h
- Simple solution: Take "fresh" i.i.d. sample $V = (x_1, y_1), \ldots, (x_{m_v}, y_{m_v})$
- Output $L_V(h)$ as an estimator of $L_\mathcal{D}(h)$
- Using Hoeffding's inequality, if the range of ℓ is $[0, 1]$ we have

$$|L_V(h) - L_\mathcal{D}(h)| \;\leq\; \sqrt{\frac{\log(2/\delta)}{2\,m_v}}\;.$$

Validation for Model Selection

- Fitting polynomials of degrees 2,3, and 10 based on the black points
- The red points are validation examples
- Choose the degree 3 polynomial as it has minimal validation error

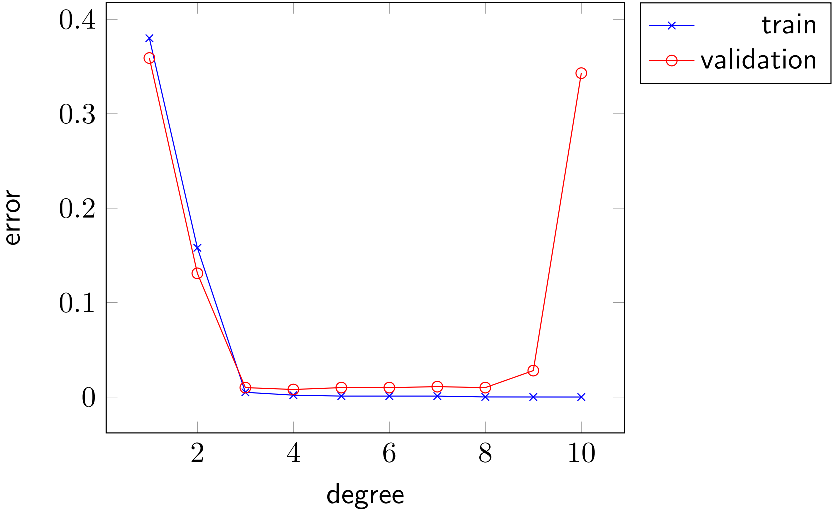


Validation for Model Selection — Analysis

- Let $\mathcal{H} = \{h_1, \ldots, h_r\}$ be the output predictors of applying ERM w.r.t. the different classes on S
- Let V be a fresh validation set
- Choose $h^* \in \mathrm{ERM}_{\mathcal{H}}(V)$
- By our analysis of finite classes,

$$L_{\mathcal{D}}(h^*) \le \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \sqrt{\frac{2 \log(2|\mathcal{H}|/\delta)}{|V|}}$$

The model-selection curve



Train-Validation-Test split

- In practice, we usually have one pool of examples and we split them into three sets:
  - Training set: apply the learning algorithm with different parameters on the training set to produce $\mathcal{H} = \{h_1, \ldots, h_r\}$
  - Validation set: Choose h^* from $\mathcal{H}$ based on the validation set
  - Test set: Estimate the error of h^* using the test set

k-fold cross validation

- The train-validation-test split is the best approach when data is plentiful. If data is scarce:

k-Fold Cross Validation for Model Selection

input:
training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
learning algorithm A and a set of parameter values Θ
partition S into $S_1, S_2, \ldots, S_k$
foreach $\theta \in \Theta$
for $i = 1 \ldots k$
$h_{i,\theta} = A(S \setminus S_i; \theta)$
$\text{error}(\theta) = \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$
output
$\theta^\star = \text{argmin}_\theta \left[\text{error}(\theta)\right], \quad h_{\theta^\star} = A(S; \theta^\star)$

Summary

- The general PAC model
  - Agnostic
  - General loss functions
- Uniform convergence is sufficient for learnability
- Uniform convergence holds for finite classes and bounded loss
- Least squares
  - Linear regression
  - Polynomial fitting
- The bias-complexity tradeoff
  - Approximation error vs. Estimation error
- Validation
- Model selection