

Foundations of Machine Learning

Learning with Finite Hypothesis Sets

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

■ Some computational learning questions

- What can be learned efficiently?
- What is inherently hard to learn?
- A general model of learning?

■ Complexity

- **Computational complexity**: time and space.
- **Sample complexity**: amount of training data needed to learn successfully.
- **Mistake bounds**: number of mistakes before learning successfully.

This lecture

- PAC Model
- Sample complexity, finite H , consistent case
- Sample complexity, finite H , inconsistent case

Definitions and Notation

- X : set of all possible instances or examples, e.g., the set of all men and women characterized by their height and weight.
- $c: X \rightarrow \{0, 1\}$: the target concept to learn; can be identified with its support $\{x \in X: c(x) = 1\}$.
- C : concept class, a set of target concepts c .
- D : target distribution, a fixed probability distribution over X . Training and test examples are drawn according to D .

Definitions and Notation

- S : training sample.
- H : set of concept hypotheses, e.g., the set of all linear classifiers.
- The learning algorithm receives sample S and selects a hypothesis h_S from H approximating c .

Errors

- **True error or generalization error** of h with respect to the target concept c and distribution D :

$$R(h) = \Pr_{x \sim D} [h(x) \neq c(x)] = \mathbb{E}_{x \sim D} [1_{h(x) \neq c(x)}].$$

- **Empirical error**: average error of h on the training sample S drawn according to distribution D ,

$$\hat{R}_S(h) = \Pr_{x \sim \hat{D}} [h(x) \neq c(x)] = \mathbb{E}_{x \sim \hat{D}} [1_{h(x) \neq c(x)}] = \frac{1}{m} \sum_{i=1}^m 1_{h(x_i) \neq c(x_i)}.$$

- **Note:** $R(h) = \mathbb{E}_{S \sim D^m} [\hat{R}_S(h)]$.

PAC Model

(Valiant, 1984)

- **PAC learning:** Probably Approximately Correct learning.
- **Definition:** concept class C is **PAC-learnable** if there exists a learning algorithm L such that:

- for all $c \in C$, $\epsilon > 0$, $\delta > 0$, and all distributions D ,

$$\Pr_{S \sim D^m} [R(h_S) \leq \epsilon] \geq 1 - \delta,$$

- for samples S of size $m = \text{poly}(1/\epsilon, 1/\delta)$ for a fixed polynomial.

Remarks

- Concept class C is known to the algorithm.
- Distribution-free model: no assumption on D .
- Both training and test examples drawn $\sim D$.
- Probably: confidence $1 - \delta$.
- Approximately correct: accuracy $1 - \epsilon$.
- **Efficient PAC-learning:** L runs in time $\text{poly}(1/\epsilon, 1/\delta)$.
- What about the cost of the representation of $c \in C$?

PAC Model - New Definition

■ Computational representation:

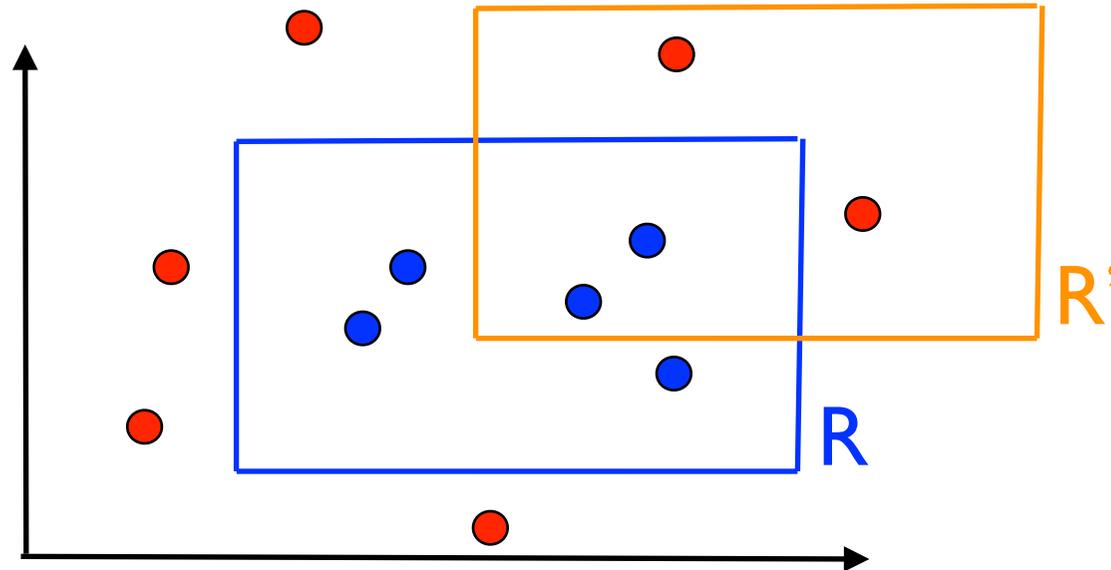
- cost for $x \in X$ in $O(n)$.
- cost for $c \in C$ in $O(\text{size}(c))$.

■ Extension: running time.

$$O(\text{poly}(1/\epsilon, 1/\delta)) \longrightarrow O(\text{poly}(1/\epsilon, 1/\delta, n, \text{size}(c))).$$

Example - Rectangle Learning

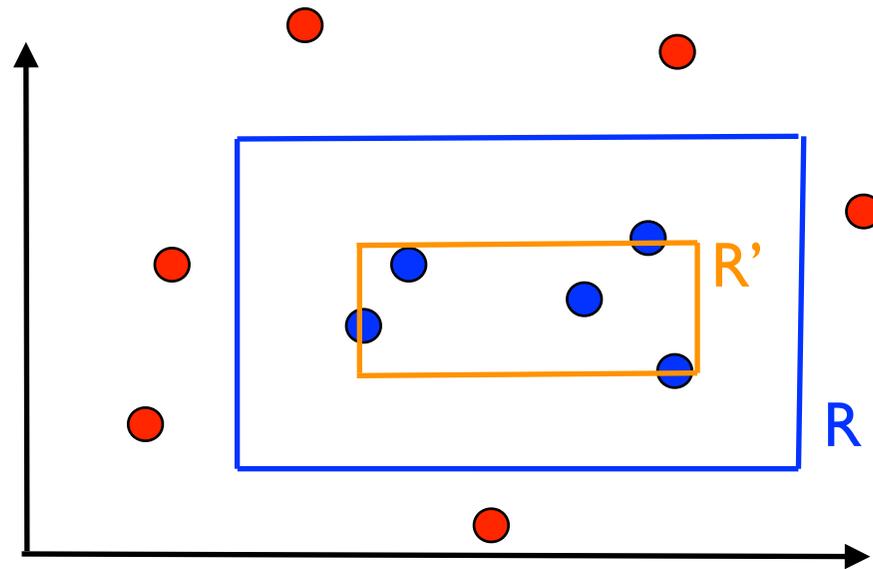
- **Problem:** learn unknown axis-aligned rectangle R using as small a labeled sample as possible.



- **Hypothesis:** rectangle R' . In general, there may be false positive and false negative points.

Example - Rectangle Learning

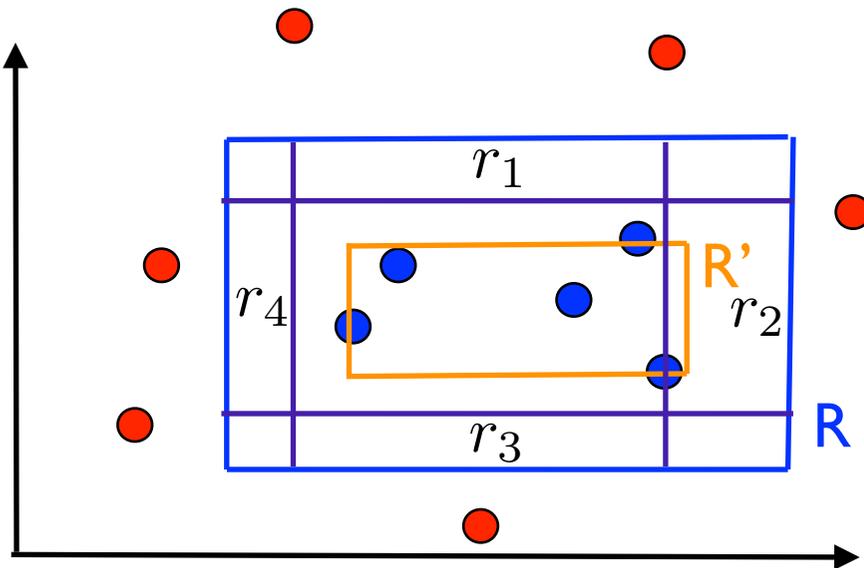
- **Simple method:** choose tightest consistent rectangle R' for a large enough sample. How large a sample? Is this class PAC-learnable?



- What is the probability that $R(R') > \epsilon$?

Example - Rectangle Learning

- Fix $\epsilon > 0$ and assume $\Pr_D[R] > \epsilon$ (otherwise the result is trivial).
- Let r_1, r_2, r_3, r_4 be four smallest rectangles along the sides of R such that $\Pr_D[r_i] \geq \frac{\epsilon}{4}$.



$$R = [l, r] \times [b, t]$$

$$r_4 = [l, s_4] \times [b, t]$$

$$s_4 = \inf\{s : \Pr [[l, s] \times [b, t]] \geq \frac{\epsilon}{4}\}$$

$$\Pr_D [[l, s_4] \times [b, t]] < \frac{\epsilon}{4}$$

Example - Rectangle Learning

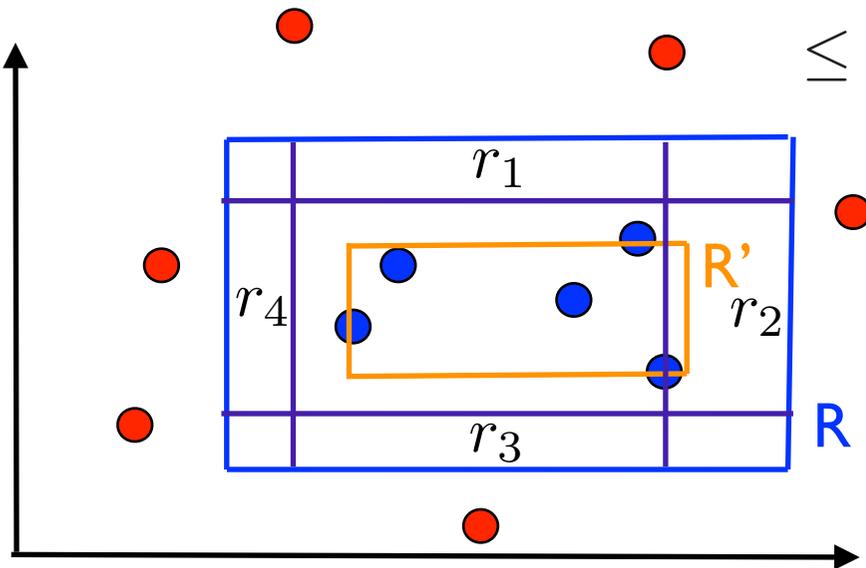
- Errors can only occur in $R - R'$. Thus (geometry),

$R(R') > \epsilon \Rightarrow R'$ misses at least one region r_i .

- Therefore, $\Pr[R(R') > \epsilon] \leq \Pr[\cup_{i=1}^4 \{R' \text{ misses } r_i\}]$

$$\leq \sum_{i=1}^4 \Pr[\{R' \text{ misses } r_i\}]$$

$$\leq 4(1 - \frac{\epsilon}{4})^m \leq 4e^{-\frac{m\epsilon}{4}}.$$



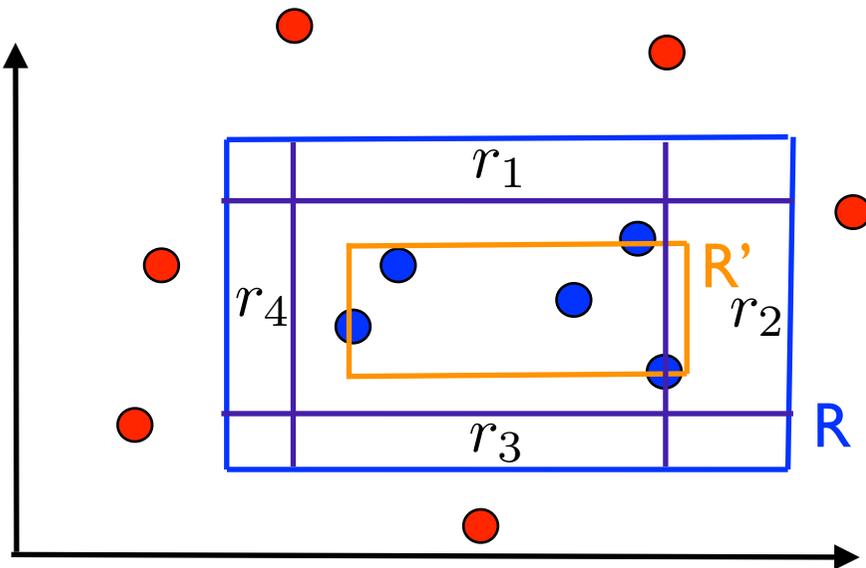
Example - Rectangle Learning

- Set $\delta > 0$ to match the upper bound:

$$4e^{-\frac{m\epsilon}{4}} \leq \delta \Leftrightarrow m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}.$$

- Then, for $m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}$, with probability at least $1 - \delta$,

$$R(R') \leq \epsilon.$$



Notes

- Infinite hypothesis set, but simple proof.
 - Does this proof readily apply to other similar concepts classes?
 - Geometric properties:
 - key in this proof.
 - in general non-trivial to extend to other classes, e.g., non-concentric circles (see HW2, 2006).
- Need for more general proof and results.

This lecture

- PAC Model
- Sample complexity, finite H , consistent case
- Sample complexity, finite H , inconsistent case

Learning Bound for Finite H - Consistent Case

- **Theorem:** let H be a finite set of functions from X to $\{0, 1\}$ and L an algorithm that for any target concept $c \in H$ and sample S returns a consistent hypothesis $h_S: \hat{R}_S(h_S) = 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,

$$R(h_S) \leq \frac{1}{m} (\log |H| + \log \frac{1}{\delta}).$$

Learning Bound for Finite H - Consistent Case

■ **Proof:** for any $\epsilon > 0$, define $H_\epsilon = \{h \in H : R(h) > \epsilon\}$. We want to prove that, with high probability, if h_S is consistent, then it has low error:

$$\begin{aligned}\mathbb{P}\left[\widehat{R}_S(h_S) = 0 \Rightarrow R(h_S) \leq \epsilon\right] &\geq 1 - \delta \Leftrightarrow \mathbb{P}\left[\widehat{R}_S(h_S) = 0 \wedge R(h_S) > \epsilon\right] \leq \delta \\ &\Leftrightarrow \mathbb{P}\left[\widehat{R}_S(h_S) = 0 \wedge h_S \in H_\epsilon\right] \leq \delta.\end{aligned}$$

$$\begin{aligned}&\mathbb{P}\left[\exists h \in H : \widehat{R}_S(h) = 0 \wedge h \in H_\epsilon\right] \\ &= \mathbb{P}\left[\exists h \in H_\epsilon : \widehat{R}_S(h) = 0\right] \\ &= \mathbb{P}\left[\widehat{R}_S(h_1) = 0 \vee \dots \vee \widehat{R}_S(h_{|H_\epsilon|}) = 0\right] \\ &\leq \sum_{h \in H_\epsilon} \mathbb{P}\left[\widehat{R}_S(h) = 0\right] && \text{(union bound)} \\ &\leq \sum_{h \in H_\epsilon} (1 - \epsilon)^m \leq |H_\epsilon|(1 - \epsilon)^m \leq |H|e^{-m\epsilon}.\end{aligned}$$

Remarks

- The algorithm can be ERM if problem realizable.
- Error bound linear in $\frac{1}{m}$ and only logarithmic in $\frac{1}{\delta}$.
- $\log_2 |H|$ is the number of bits used for the representation of H .
- Bound is loose for large $|H|$.
- Uninformative for infinite $|H|$.

Conjunctions of Boolean Literals

- Example for $n=6$.
- Algorithm: start with $x_1 \wedge \bar{x}_1 \wedge \dots \wedge x_n \wedge \bar{x}_n$ and rule out literals incompatible with positive examples.

0	1	1	0	1	1	+
0	1	1	1	1	1	+
0	0	1	1	0	1	-
0	1	1	1	1	1	+
1	0	0	1	1	0	-
0	1	0	0	1	1	+
0	1	?	?	1	1	

→ $\bar{x}_1 \wedge x_2 \wedge x_5 \wedge x_6.$

Conjunctions of Boolean Literals

- **Problem:** learning class C_n of conjunctions of boolean literals with at most n variables (e.g., for $n = 3$, $x_1 \wedge \overline{x_2} \wedge x_3$).
- **Algorithm:** choose h consistent with S .
 - Since $|H| = |C_n| = 3^n$, sample complexity:
$$m \geq \frac{1}{\epsilon} \left((\log 3) n + \log \frac{1}{\delta} \right).$$
$$\delta = .02, \epsilon = .1, n = 10, m \geq 149.$$
 - Computational complexity: polynomial, since algorithmic cost per training example is in $O(n)$.

This lecture

- PAC Model
- Sample complexity, finite H , consistent case
- Sample complexity, finite H , inconsistent case

Inconsistent Case

- No $h \in H$ is a consistent hypothesis.
- The typical case in practice: difficult problems, complex concept class.
- But, inconsistent hypotheses with a small number of errors on the training set can be useful.
- Need a more powerful tool: Hoeffding's inequality.

Hoeffding's Inequality

- **Corollary:** for any $\epsilon > 0$ and any hypothesis $h: X \rightarrow \{0, 1\}$ the following inequalities holds:

$$\Pr[R(h) - \hat{R}(h) \geq \epsilon] \leq e^{-2m\epsilon^2}$$

$$\Pr[\hat{R}(h) - R(h) \geq \epsilon] \leq e^{-2m\epsilon^2}.$$

- Combining these one-sided inequalities yields

$$\Pr[|R(h) - \hat{R}(h)| \geq \epsilon] \leq 2e^{-2m\epsilon^2}.$$

Application to Learning Algorithm?

- Can we apply that bound to the hypothesis h_S returned by our learning algorithm when training on sample S ?
- No, because h_S is not a fixed hypothesis, it depends on the training sample. Note also that $E[\widehat{R}(h_S)]$ is not a simple quantity such as $R(h_S)$.
- Instead, we need a bound that holds simultaneously for all hypotheses $h \in H$, a **uniform convergence bound**.

Generalization Bound - Finite H

- **Theorem:** let H be a finite hypothesis set, then, for any $\delta > 0$, with probability at least $1 - \delta$,

$$\forall h \in H, R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log |H| + \log \frac{2}{\delta}}{2m}}.$$

- **Proof:** By the union bound,

$$\begin{aligned} & \Pr \left[\max_{h \in H} |R(h) - \hat{R}_S(h)| > \epsilon \right] \\ &= \Pr \left[|R(h_1) - \hat{R}_S(h_1)| > \epsilon \vee \dots \vee |R(h_{|H|}) - \hat{R}_S(h_{|H|})| > \epsilon \right] \\ &\leq \sum_{h \in H} \Pr \left[|R(h) - \hat{R}_S(h)| > \epsilon \right] \\ &\leq 2|H| \exp(-2m\epsilon^2). \end{aligned}$$

Remarks

- Thus, for a finite hypothesis set, whp,

$$\forall h \in H, R(h) \leq \hat{R}_S(h) + O\left(\sqrt{\frac{\log |H|}{m}}\right).$$

- Error bound in $O\left(\frac{1}{\sqrt{m}}\right)$ (quadratically worse).
- $\log_2 |H|$ can be interpreted as the number of bits needed to encode H .
- Occam's Razor principle (theologian William of Occam): "plurality should not be posited without necessity".

Occam's Razor

- Principle formulated by controversial theologian William of Occam: “plurality should not be posited without necessity”, rephrased as “the simplest explanation is best”;
- invoked in a variety of contexts, e.g., syntax. Kolmogorov complexity can be viewed as the corresponding framework in information theory.
- here, to minimize true error, choose the most parsimonious explanation (smallest $|H|$).
- we will see later other applications of this principle.

Lecture Summary

- C is **PAC-learnable** if $\exists L, \forall c \in C, \forall \epsilon, \delta > 0, m = P\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right),$

$$\Pr_{S \sim D^m} [R(h_S) \leq \epsilon] \geq 1 - \delta.$$

- Learning bound, finite H consistent case:

$$R(h) \leq \frac{1}{m} (\log |H| + \log \frac{1}{\delta}).$$

- Learning bound, finite H inconsistent case:

$$R(h) \leq \hat{R}_S(h) + \sqrt{\frac{\log |H| + \log \frac{2}{\delta}}{2m}}.$$

- How do we deal with infinite hypothesis sets?

References

- Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, Volume 36, Issue 4, 1989.
- Michael Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*, MIT Press, 1994.
- Leslie G. Valiant. *A Theory of the Learnable*, *Communications of the ACM* 27(11):1134–1142 (1984).

Appendix

Universal Concept Class

■ **Problem:** each $x \in X$ defined by n boolean features.
Let C be the set of all subsets of X .

■ **Question:** is C PAC-learnable?

■ **Sample complexity:** H must contain C . Thus,

$$|H| \geq |C| = 2^{(2^n)}.$$

The bound gives $m = \frac{1}{\epsilon} ((\log 2) 2^n + \log \frac{1}{\delta})$.

■ It can be proved that C is **not PAC-learnable**, it requires an exponential sample size.

k -Term DNF Formulae

- **Definition:** expressions of the form $T_1 \vee \cdots \vee T_k$ with each term T_i conjunctions of boolean literals with at most n variables.
- **Problem:** learning k -term DNF formulae.
- **Sample complexity:** $|H| = |C| = 3^{nk}$. Thus, polynomial sample complexity $\frac{1}{\epsilon} ((\log 3) nk + \log \frac{1}{\delta})$.
- **Time complexity:** intractable if $RP \neq NP$: the class is then not efficiently PAC-learnable (proof by reduction from graph 3-coloring). But, a strictly larger class is!

k -CNF Expressions

- **Definition:** expressions $T_1 \wedge \cdots \wedge T_j$ of arbitrary length j with each term T_i a disjunction of at most k boolean attributes.
- **Algorithm:** reduce problem to that of learning conjunctions of boolean literals. $(2n)^k$ new variables:

$$(u_1, \dots, u_k) \rightarrow Y_{u_1, \dots, u_k}.$$

- the transformation is a bijection;
- effect of the transformation on the distribution is not an issue: PAC-learning allows any distribution D .

k -Term DNF Terms and k -CNF Expressions

- **Observation:** any k -term DNF formula can be written as a k -CNF expression. By associativity,

$$\bigvee_{i=1}^k u_{i,1} \wedge \cdots \wedge u_{i,n_i} = \bigwedge_{j_1 \in [1, n_1], \dots, j_k \in [1, n_k]} u_{1,j_1} \vee \cdots \vee u_{k,j_k}.$$

- **Example:** $(u_1 \wedge u_2 \wedge u_3) \vee (v_1 \wedge v_2 \wedge v_3) = \bigwedge_{i,j=1}^3 (u_i \vee v_j)$.
- But, in general converting a k -CNF (equiv. to a k -term DNF) to a k -term DNF is intractable.
- Key aspects of PAC-learning definition:
 - cost of representation of concept c .
 - choice of hypothesis set H .