# Classification: The PAC Learning Framework

Machine Learning: Jordan Boyd-Graber
University of Colorado Boulder

LECTURE 5B

# Content Questions

# Content Questions

## Content Questions

# Content Questions

# Content Questions

# Content Questions

## Content Questions

# Content Questions

# Content Questions

# Content Questions

# Quiz!

## Admin Questions

# Admin Questions

# Admin Questions

## PAC Learnability: Rectangles

Is the hypothesis class of axis-aligned rectangles PAC learnable?

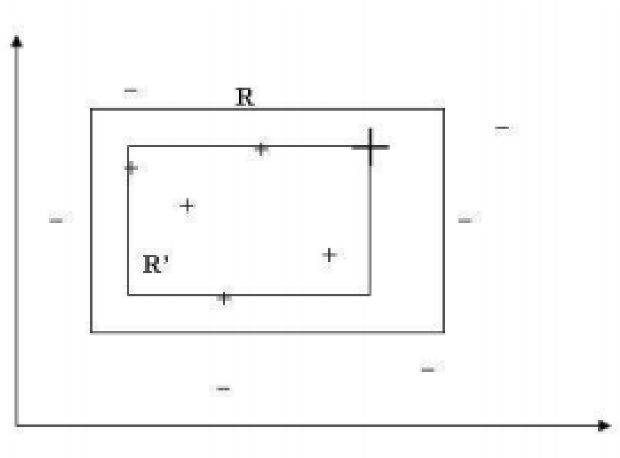
## PAC Learnability: Rectangles

Is the hypothesis class of axis-aligned rectangles PAC learnable?

A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. Journal of the ACM (JACM), 36(4):929?965, 1989

# What's the learning algorithm

## What's the learning algorithm




Call this $h_S$, which we learned from data. $h_s \in c$

# Proof

## Proof

Let $c \equiv [b,t] \times [l,r]$.

## Proof

Let $c \equiv [b,t] \times [l,r]$. By construction, $h_S \in c$, so it can only give false negatives.

## Proof

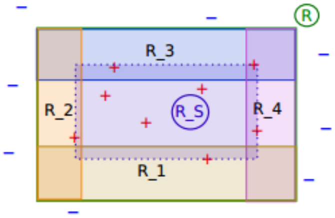
Let $c \equiv [b,t] \times [l,r]$. By construction, $h_S \in c$, so it can only give false negatives. The region of error is precisely $c \setminus h_S$.

## Proof

Let $c \equiv [b, t] \times [l, r]$. By construction, $h_S \in c$, so it can only give false negatives. The region of error is precisely $c \setminus h_S$. WLOG, assume $P(R) \geq \epsilon$.

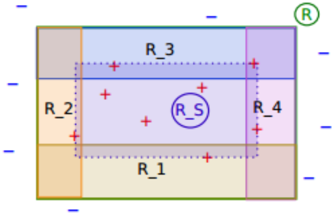
## Proof

Let $c \equiv [b, t] \times [l, r]$. By construction, $h_S \in c$, so it can only give false negatives. The region of error is precisely $c \setminus h_S$. WLOG, assume $P(R) \geq \epsilon$. Consider rectangles $R_1 \dots R_4$:



## Proof

Let $c \equiv [b,t] \times [l,r]$. By construction, $h_S \in c$, so it can only give false negatives. The region of error is precisely $c \setminus h_S$. WLOG, assume $P(R) \geq \epsilon$. Consider rectangles $R_1 \dots R_4$:




We get a bad $h_S$ only if we have an observation fall in this region. So let's bound this probability.

## Bounds

$$\Pr[error] = \Pr[\Cup_{i=1}^{4} x \notin R_i] \quad (1)$$

$$\leq \sum_{i=1}^{4} \Pr[x \notin R_i] \quad (2)$$

$$= \sum_{i=1}^{4} (1 - P(R_i))^m \quad (3)$$

## Bounds

$$\Pr[error] = \Pr[\Cup_{i=1}^{4} x \notin R_i] \tag{1}$$

$$\leq \sum_{i=1}^{4} \Pr[x \notin R_i] \tag{2}$$

$$= \sum_{i=1}^{4} (1 - P(R_i))^m \tag{3}$$

If we assume that $P(R_i) \geq \frac{\epsilon}{4}$, then

$$\Pr[error] \leq 4\left(1 - \frac{\epsilon}{4}\right)^m \leq 4 \cdot \exp\left\{-\frac{m\epsilon}{4}\right\} \tag{4}$$

## Bounds

$$\Pr[error] = \Pr[\Cup_{i=1}^{4} x \notin R_i] \quad (1)$$

$$\leq \sum_{i=1}^{4} \Pr[x \notin R_i] \quad (2)$$

$$= \sum_{i=1}^{4} (1 - P(R_i))^m \quad (3)$$

If we assume that $P(R_i) \geq \frac{\epsilon}{4}$, then

$$\Pr[error] \leq 4\left(1 - \frac{\epsilon}{4}\right)^m \leq 4 \cdot \exp\left\{-\frac{m\epsilon}{4}\right\} \quad (4)$$

Solving for $m$ gives

$$m \geq \frac{4 \ln 4/\delta}{\epsilon} \quad (5)$$

Are Boolean conjunctions PAC learnable? Think of every feature as a Boolean variable; in a given example the variable is given the value 1 if its corresponding feature appears in the examples and 0 otherwise. In this way, if the number of measured features is $n$ the concept is represented as a Boolean function $c : \{0,1\} \longmapsto \{0,1\}$. For example we could define a chair as something that has four legs **and** you can sit on **and** is made of wood. Can you learn such a conjunction concept over $n$ variables?

# Algorithm

## Algorithm

Start with

$$h = \bar{x}_1 x_1 \bar{x}_2 x_2 \ldots \bar{x}_n x_n \tag{6}$$

## Algorithm

Start with

$$h = \bar{x}_1 x_1 \bar{x}_2 x_2 \ldots \bar{x}_n x_n \quad (6)$$

For every positive example you see, remove the negation of all dimensions present in that example.

## Algorithm

Start with

$$h = \bar{x}_1 x_1 \bar{x}_2 x_2 \dots \bar{x}_n x_n \quad (6)$$

For every positive example you see, remove the negation of all dimensions present in that example. Example: 10001, 11001, 10000, 11000

## Algorithm

Start with

$$h = \bar{x}_1 x_1 \bar{x}_2 x_2 \ldots \bar{x}_n x_n \tag{6}$$

For every positive example you see, remove the negation of all dimensions present in that example. Example: 10001, 11001, 10000, 11000

- After first example, $x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$

## Algorithm

Start with

$$h = \bar{x}_1 x_1 \bar{x}_2 x_2 \ldots \bar{x}_n x_n \quad (6)$$

For every positive example you see, remove the negation of all dimensions present in that example. Example: 10001, 11001, 10000, 11000

- After first example, $x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \bar{x}_5$
- After last example, $x_1 \bar{x}_3 \bar{x}_4$

# Observations

## Observations

- Having seen no data, *h* says no to everything
- Our algorithm can be two specific. It might not say yes when it should.

## Observations

- Having seen no data, $h$ says no to everything
- Our algorithm can be two specific. It might not say yes when it should.
- We make an error on a literal if we've never seen it before (there are $2n$ literals: $x_1, \bar{x}_1$)

# Bounds

## Bounds

Let $p(z)$ be the probability that our concept returns a positive example in which literal $z$ is false.

$$R(h) \leq \sum_{z} p(z) \tag{7}$$

A literal $z$ is bad if $p(z) \geq \frac{\epsilon}{2n}$.

## Bounds

Let $p(z)$ be the probability that our concept returns a positive example in which literal $z$ is false.

$$R(h) \leq \sum_z p(z) \tag{7}$$

A literal $z$ is bad if $p(z) \geq \frac{\epsilon}{2n}$.
If $h$ has no bad literals, then $h$ will have error less than $\epsilon$.

# Solving for number of examples

## Solving for number of examples

$$m \geq \frac{2n}{\epsilon}\left(\ln 2n + \ln \frac{1}{\delta}\right) \quad (8)$$

# 3-DNF

# 3-DNF

Not efficiently learnable unless P = NP.