Introduction to Machine Learning

Machine Learning: Jordan Boyd-Graber
University of Maryland

FEATURE ENGINEERING

Content Questions

Quiz!

Admin Questions

- Writeup must fit in one page
- Unit tests are **not** comprehensive
- Don't break autograder
- HW3 due next week

PAC Learnability: Rectangles

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

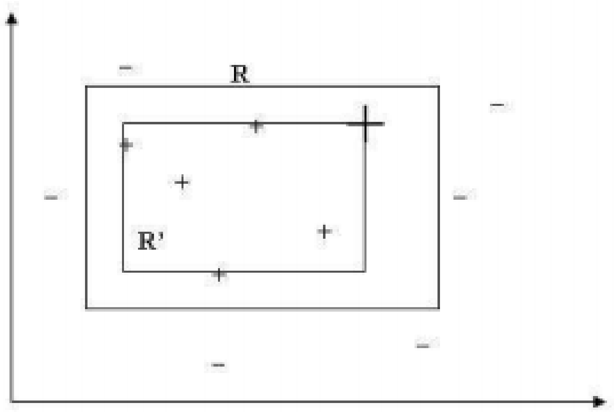
PAC Learnability: Rectangles

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

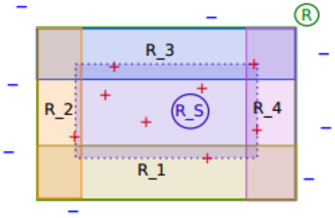
Let $c \equiv [b,t] \times [l,r]$. By construction, $h_S \in c$, so it can only give false negatives. The region of error is precisely $c \setminus h_S$.

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 $P(R) \geq \epsilon$.

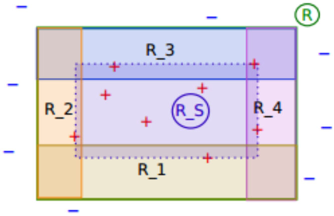
Proof

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 $R_1 \ldots R_4$:



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

Bounds

$$
\begin{aligned}
\Pr[error] &= \Pr[\Cup_{i=1}^{4} x \notin R_i] && (1)\\
&\leq \sum_{i=1}^{4} \Pr[x \notin R_i] && (2)\\
&= \sum_{i=1}^{4} (1 - P(R_i))^m && (3)
\end{aligned}
$$

Bounds

$$
\begin{aligned}
\Pr[error] &= \Pr[\Cup_{i=1}^{4} x \notin R_i] && (1)\\
&\leq \sum_{i=1}^{4} \Pr[x \notin R_i] && (2)\\
&= \sum_{i=1}^{4} (1 - P(R_i))^m && (3)
\end{aligned}
$$

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

$$\Pr[error] = \Pr[\Cup_{i=1}^{4} x \notin R_i] \tag{1}$$
$$\leq \sum_{i=1}^{4} \Pr[x \notin R_i] \tag{2}$$
$$= \sum_{i=1}^{4} (1 - P(R_i))^m \tag{3}$$

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

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

Solving for m gives

$$m \geq \frac{4 \ln 4/\delta}{\epsilon} \tag{5}$$

Concept Learning

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

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

(say no to everything)

Algorithm

Start with

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

(say no to everything) For every positive example you see, remove the negation of all dimensions present in that example.

Algorithm

Start with

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

(say no to everything) For every positive example you see, remove the negation of all dimensions present in that example. Example: 10001, 11001, 10000, 11000

Algorithm

Start with

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

(say no to everything) For every positive example you see, remove the negation of all dimensions present in that example. Example: 10001, 11001, 10000, 11000

- After first example, $x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 x_5$

Algorithm

Start with

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

(say no to everything) For every positive example you see, remove the negation of all dimensions present in that example. Example: 10001, 11001, 10000, 11000

- After first example, $x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 x_5$
- 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 too specific. It might not say yes when it should.

Observations

- Having seen no data, h says no to everything
- Our algorithm can be too 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: $x_1, \bar{x_1}$)

Solving for number of examples

Solving for number of examples

General learning bounds for consistent hypotheses

$$m \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\frac{1}{\delta}\right) \quad (7)$$

Solving for number of examples

General learning bounds for consistent hypotheses

$$m \geq \frac{1}{\epsilon}\left(\ln|H| + \ln\frac{1}{\delta}\right) \tag{7}$$

$$m \geq \frac{1}{\epsilon}\left(n \cdot \ln 3 + \ln\frac{1}{\delta}\right) \tag{8}$$

3-DNF

3-DNF

Not efficiently learnable unless P = NP.