Gems of TCS

PAC Learning

Sasha Golovnev

April 29, 2021

Classes of Learning Problems

- Classification

Binary classifier: $X \to \{0,1\}$

.jpeg "animal"

Labeled data: (x_1, y_1)

(x_2, y_2)

.jpeg "food"

$\ldots$

(x_m, y_m)

.jpeg "food"

Classes of Learning Problems

- Classification
- Ranking

web search query

labeled data (x_1, y_1)

webpage 0.5

(x_2, y_2)

webpage 0.1

train . - - - -

$\Rightarrow$ classifier: $X \rightarrow Y$

Classes of Learning Problems

- Classification
- Ranking
- Regression

classifier: $X \to \mathbb{R}$

stock values

labeled data: (x_1, y_1)

$\vdots$

(x_m, y_m)

$\xrightarrow{\text{train}}$ classifier: $X \to \mathbb{R}$

Classes of Learning Problems

- Classification
- Ranking
- Regression
- Clustering

labeled data

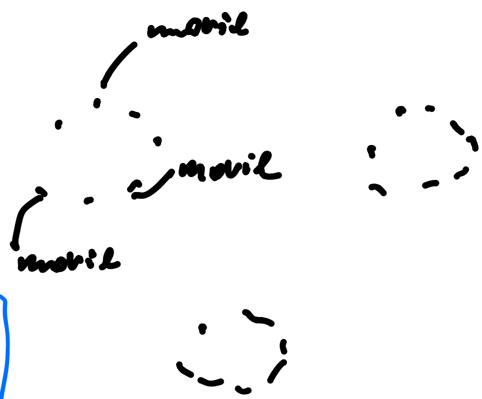
supervised learning

expert. $x_1 \to y_1$

$x_2 \to y_2$

unlabeled data

unsupervised learning



- semi-supervised
- online
- active
- reinforcement

Classes of Learning Problems

- Classification
- Ranking
- Regression
- Clustering
- ...

Example

Spam filter = binary classifier

- Given a set of labeled emails

$(x_1, y_1 \in \{0,1\})$

email

spam/non-spam

$(x_2, y_2 \in \{0,1\})$

Example

- Given a set of labeled emails
- Build a classifier that predicts spam/non-spam labels for incoming emails

Setup

- Partition labeled data into three sets:
  - training sample
  - validation sample
  - test sample

SETUP

- Partition labeled data into three sets:
  - training sample
  - validation sample
  - test sample
- Identify relevant features

- sender
- loans, bank account, prince, business
- nickname, real name
- links to shady webpages
- ....

$\Rightarrow v \in \mathbb{R}^n$

good features $\xRightarrow{\text{usually}}$ good classifier

Setup

- Partition labeled data into three sets:
  - training sample
  - validation sample
  - test sample
- Identify relevant features
- Train on training sample

Ex. of our classifier:

given features $v \in \mathbb{R}^n$

classifier: $\boxed{v_1 \cdot 2 - v_2 \cdot 0.1 + v_3 \cdot 5 + v_4 \cdot 4 + \dots} \geq \boxed{\theta}$ spam

$< \theta$ non-spam

If θ = large $\Rightarrow$ conservative classifier

If θ = small $\Rightarrow$ aggressive classifier

Setup

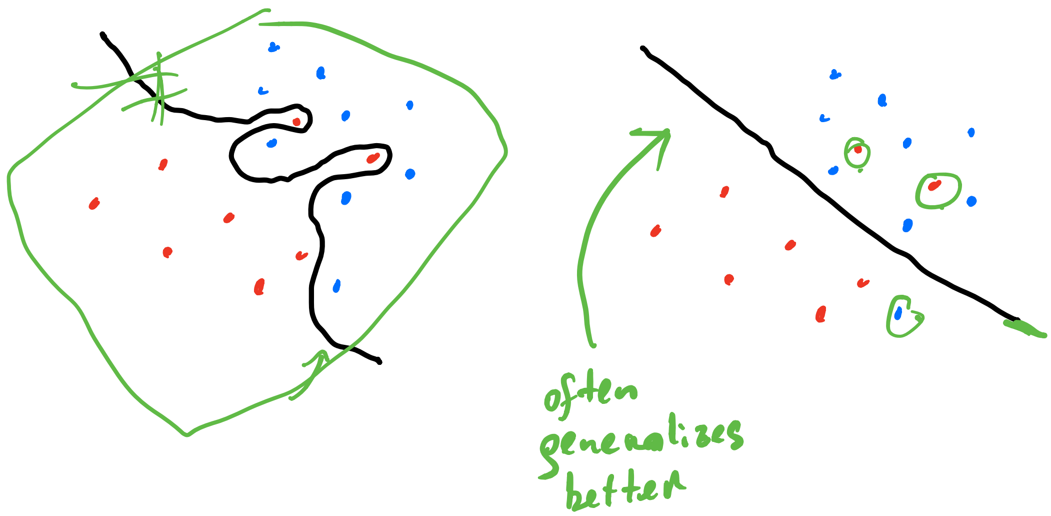
- Partition labeled data into three sets:
  - training sample
  - validation sample
  - test sample
- Identify relevant features
- Train on training sample
- Tune parameters using validation sample

  For example, choose θ

Setup

- Partition labeled data into three sets:
  - training sample
  - validation sample
  - test sample
- Identify relevant features
- Train on training sample
- Tune parameters using validation sample
- Evaluate using test sample

$2 \cdot v_1 - 0.1 v_2 + 5 v_3 + 4 v_4 \geq 5.5$


often
generalizes
better

What can be learned?

- What can be learned?
- What cannot be learned?
- How many samples do we need to learn?

What can be learned?

- What can be learned?
- What cannot be learned?
- How many samples do we need to learn?
- Framework of PAC learning (L. Valiant, 1984)

Definitions

- X—set of all possible instances/examples

X = set of emails
= set of feature vectors

Definitions

- X—set of all possible instances/examples
- $\mathcal{D}$—target distribution over X

Definitions

- X—set of all possible instances/examples
- $\mathcal{D}$—target distribution over X
- c—target concept

"correct classifier"

c : email $\rightarrow \{0, 1\}$

spam non-spam

Definitions

- X—set of all possible instances/examples
- $\mathcal{D}$—target distribution over X
- c—target concept
- C—concept class

C = class of all concepts

ex.: C = all functions from feature vectors $\to \{0,1\}$

$c \in C$

labeled data

$x_1 \in X \quad y_1 = c(x_1)$

$x_2 \in X \quad y_2 = c(x_2)$

$x_m \in X \quad y_m = c(x_m)$

Definitions

- X—set of all possible instances/examples
- $\mathcal{D}$—target distribution over X
- c—target concept
- C—concept class
- H—set of concept hypotheses

e.g., H = linear functions from features $\rightarrow \{0,1\}$

$v_1 \cdot 3 + v_2 \cdot 0.1 + v_3 \cdot 5 + \ldots \geq 100$

Definitions

- X—set of all possible instances/examples
- $\mathcal{D}$—target distribution over X
- c—target concept
- C—concept class
- H—set of concept hypotheses
- Goal: given training set, select $h \in H$ that approximates c well

For a random email ($x \in X$ sampled from D)

$h(x) = c(x)$ with high probability

Generalization Error

For hypothesis h, target concept c, and target distribution D:

$$R(h) = \Pr_{x \sim D}[h(x) \neq \boxed{c(x)}].$$

We want to minimize $R(h)$

Errors

Generalization Error

For hypothesis h, target concept c, and target distribution D:

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

don't know how to compute

Empirical Error

x_1 $y_1 = c(x_1)$

$\vdots$

x_m $y_m = c(x_m)$

training

For hypothesis h, target concept c, and sample $S = \underline{(x_1, \ldots, x_m)}$:

$$\boxed{\widehat{R}(h)} = \frac{|\{x_i : \boxed{h(x_i)} \neq \boxed{c(x_i)}\}|}{m}.$$

know to compute

Average Error

Empirical error

Generalization error

$$\mathbb{E}_{S \sim D^m}[\widehat{R}(h)] = \underline{R(h)}\,.$$

PAC (Probably Approximately Correct)

Concept class C is PAC-learnable if there exists learning algorithm s.t.

- for all $c \in C$, $\boxed{\varepsilon} > 0$, $\delta > 0$, all distributions D:

sample S

learning

alg $\Rightarrow h_S$

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

$\boxed{\varepsilon = 0.01}$ $\boxed{\delta = 0.01}$

PAC Learning

PAC (Probably Approximately Correct)

Concept class C is PAC-learnable if there exists learning algorithm s.t.

- for all $c \in C, \varepsilon > 0, \delta > 0$, all distributions D:

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

approximately

probably

- for random samples of size

$$\boxed{m} \leq \mathbf{poly}(\boxed{1/\varepsilon}, \boxed{1/\delta}, \boxed{n}) .$$

- Probably: confidence $1 - \delta$
- Approximately correct: accuracy $1 - \varepsilon$

PAC

EXAMPLE

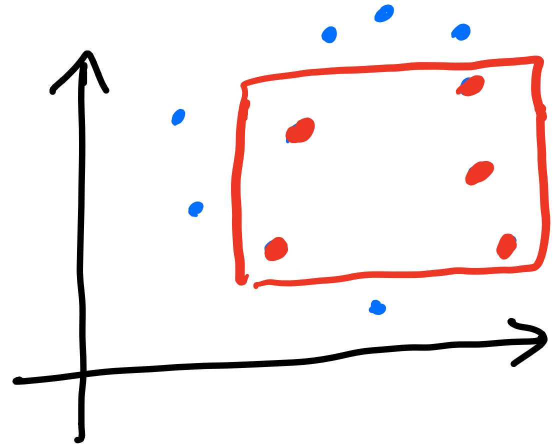
(two features representing email)

$x \in \mathbb{R}^2$

C = set of axis-aligned rectangles

Is C PAC-learnable?

We'll prove it is



Learning alg:

takes $(x_1 \in \mathbb{R}^2,\ y_1 = c(x_1))$

$(x_2 \in \mathbb{R}^2,\ y_2 = c(x_2))$

$m = \mathrm{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}\right)$

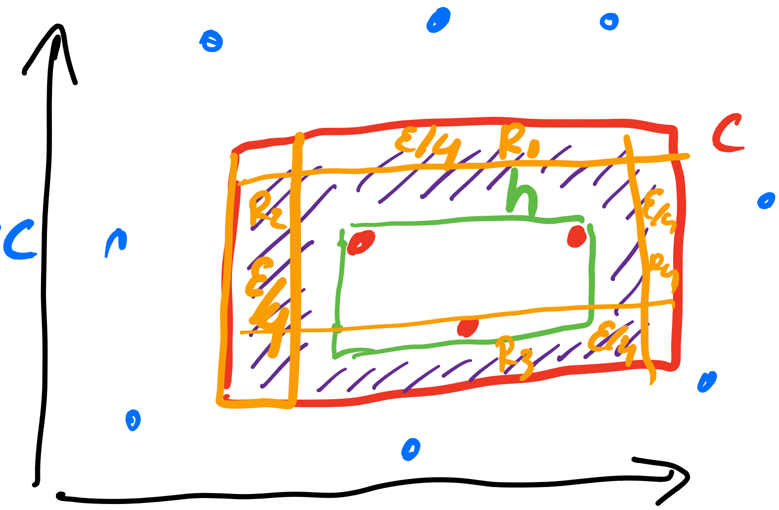
$(x_m \in \mathbb{R}^2,\ y_m \in c(x_m))$

returns Rectangle h s.t. w.p. $1-\delta$

h & c are ε-close

m labeled points

we don't know c



Alg smallest axis-aligned rectangle that contains all red points.


h

WTS: Alg PAC learns C

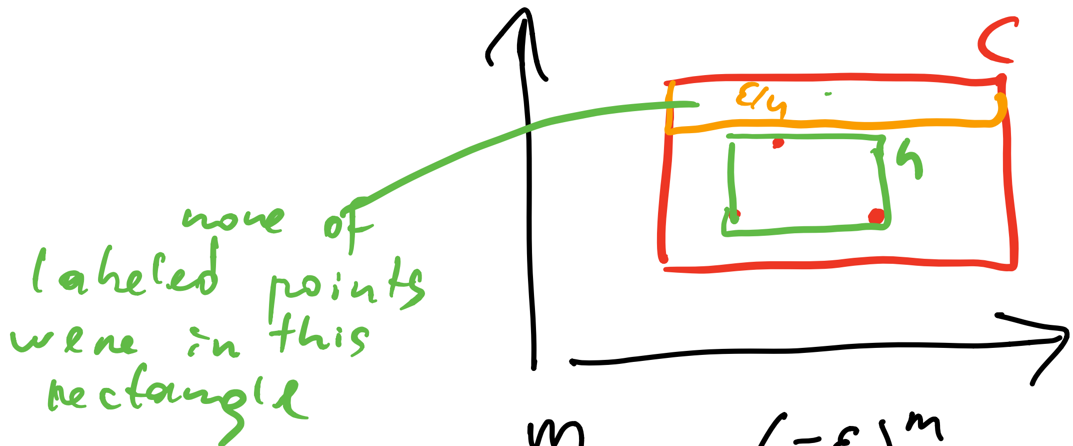
— we'll make errors $h(x) \neq c(x)$

If h is not ε-close c $\Rightarrow$ area$[c \setminus h] > \varepsilon \Rightarrow$
h misses R_1 OR R_2 OR R_3 OR R_4
(if all of them touch h $\Rightarrow$ h is ε-close to c)

$Pr[h \text{ is not } \varepsilon\text{-close to } c] \leq$

$Pr[h \text{ misses } R_1] +$
$Pr[h \text{ misses } R_2] +$
$Pr[h \text{ misses } R_3] +$
$Pr[h \text{ misses } R_4]$

$= 4 \cdot Pr[h \text{ misses rectange of area } \varepsilon/4]$



$$= \underline{4 \cdot (1 - \varepsilon/4)^m} \leq 4 \cdot \left(e^{-\frac{\varepsilon}{4}}\right)^m = \underline{4 e^{-m\varepsilon/4}}$$

Taylor series $\forall x: \underline{1+x} \leq e^x$

$$\boxed{\delta = 4e^{-m\epsilon/4}}$$

$$m = \frac{4}{\epsilon}\log\left(\frac{4}{\delta}\right) \Rightarrow$$

h will be ϵ-close to c w.p. $1-\delta$

PAC-learn C

$$m = \underline{\frac{4}{\epsilon}\log\left(\frac{4}{\delta}\right)} = \underline{\mathrm{poly}\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)}$$

$\epsilon = 0.01 \quad \delta = 0.02 \Rightarrow$

some concrete m of labeled points suffice to learn rectangle