

A More General Learning Model: Remove Realizability Assumption (Agnostic PAC Learning)

Realizability Assumption: there exists $h^* \in \mathcal{H}$ such that $L_{\mathcal{D},f}(h^*) = 0$

Informally: the label is fully determined by the instance x

\Rightarrow Too strong in many applications!

Relaxation: \mathcal{D} is a probability distribution over $\mathcal{X} \times \mathcal{Y}$
 \Rightarrow \mathcal{D} is the *joint distribution* over domain points and labels.

For example, two components of \mathcal{D} :

- \mathcal{D}_x : (marginal) distribution over domain points
- $\mathcal{D}((x, y)|x)$: conditional distribution over labels for each domain point

Given x , label y is obtained according to a conditional probability $\mathbb{P}[y|x]$.

The Empirical and True Error

With \mathcal{D} that is a probability distribution over $\mathcal{X} \times \mathcal{Y}$ the *true error* (or risk) is:

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

As before \mathcal{D} is not known to the learner; the learner only knows the training data S

Empirical risk: as before, that is

$$\underline{L_S(h) \stackrel{\text{def}}{=} \frac{|\{i, 0 \leq i \leq m : h(x_i) \neq y_i\}|}{m}}$$

Note: $L_S(h)$ = probability that for a pair (x_i, y_i) taken uniformly at random from S the event “ $h(x_i) \neq y_i$ ” holds.

An Optimal Predictor

Learner's goal: find $h: \mathcal{X} \rightarrow \mathcal{Y}$ minimizing $L_{\mathcal{D}}(h)$

Is there a *best predictor*?

Given a probability distribution \mathcal{D} over $\mathcal{X} \times \{0, 1\}$, the best predictor is the **Bayes Optimal Predictor**

$$f_{\mathcal{D}}(x) = \begin{cases} 1 & \text{if } \mathbb{P}[y = 1|x] \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Proposition

For any classifier $g: \mathcal{X} \rightarrow \{0, 1\}$, it holds $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$.

PROOF: Exercise

Can we use such predictor? \rightarrow We don't know $\mathbb{P}[y=1|x]$, so we can't

Agnostic PAC Learnability

Consider only predictors from a hypothesis class \mathcal{H} .

We are going to be ok with not finding the best predictor, but not being too far off.

Definition

A hypothesis class \mathcal{H} is agnostic PAC learnable if there exist a function $m_{\mathcal{H}}: (0, 1)^2 \rightarrow \mathbb{N}$ and a learning algorithm such that for every $\delta, \varepsilon \in (0, 1)$, for every distribution \mathcal{D} over $\mathcal{X} \times \mathcal{Y}$, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. examples generated by \mathcal{D} the algorithm returns a hypothesis h such that, with probability $> 1 - \delta$ (over the choice of the m training examples):

$$\underline{L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \varepsilon.}$$

Note: this is a generalization of the previous learning model.

A More General Learning Model: Beyond Binary Classification

Binary classification: $\mathcal{Y} = \{0, 1\}$

Other learning problems:

- multiclass classification: classification with > 2 labels
- regression: $\mathcal{Y} = \mathbb{R}$

Multiclass classification: same as before!

Regression

◦ Vector with p components

Domain set: \mathcal{X} is usually \mathbb{R}^p for some p .

Target set: \mathcal{Y} is \mathbb{R}

Training data: (as before) $S = ((x_1, y_1), \dots, (x_m, y_m))$

Learner's output: (as before) $h : \mathcal{X} \rightarrow \mathcal{Y}$

Loss: the previous one does not make much sense...

(Generalized) Loss Functions

Definition

Given any hypotheses set \mathcal{H} and some domain Z , a loss function is any function $\ell : \mathcal{H} \times Z \rightarrow \mathbb{R}_+$

Risk function = expected loss of a hypothesis $h \in \mathcal{H}$ with respect to \mathcal{D} over Z :

$$\underline{L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]}$$

Empirical risk = expected loss over a given sample

$S = (z_1, \dots, z_m) \in Z^m$:

$$\underline{L_S(h) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^m \ell(h, z_i)}$$

Some Common Loss Functions

0-1 loss: $Z = \mathcal{X} \times \mathcal{Y}$

$$\ell_{0-1}(h, (x, y)) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } h(x) = y \\ 1 & \text{if } h(x) \neq y \end{cases}$$

Commonly used in binary or multiclass classification.

Squared loss: $Z = \mathcal{X} \times \mathcal{Y}$

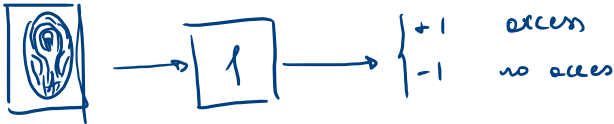
$$\underline{\ell_{sq}(h, (x, y)) \stackrel{\text{def}}{=} (h(x) - y)^2}$$

Commonly used in regression.

Note: in general, the loss function may depend on the application!
But computational considerations play a role...

How to Choose the Loss Function?

Ex of classification of fingerprints



Two types of errors: false accept and false reject

		"True" value		
		+1	-1	
Predicted Value	+1	0 (no error)	1 (false accept)	→ false accept could be 100 if it's worse than a false reject
	-1	1 (false reject)	0	

↳ Depends on the situation₂₄

Agnostic PAC Learnability for General Loss Functions

Definition

A hypothesis class \mathcal{H} is agnostic PAC learnable with respect to a set Z and a loss function $\ell : \mathcal{H} \times Z \rightarrow \mathbb{R}_+$ if there exist a function $m_{\mathcal{H}} : (0, 1)^2 \rightarrow \mathbb{N}$ and a learning algorithm such that for every $\delta, \varepsilon \in (0, 1)$, for every distribution \mathcal{D} over Z , when running the learning algorithm on $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. examples generated by \mathcal{D} the algorithm returns a hypothesis h such that, with probability $\geq 1 - \delta$ (over the choice of the m training examples):

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \varepsilon$$

where $L_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$

Machine Learning

Linear Models

Fabio Vandin

October 16th, 2023

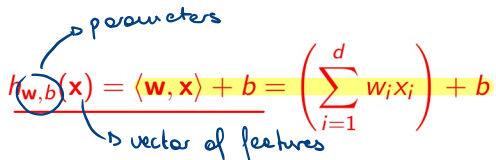
Linear Predictors and Affine Functions

Consider $\mathcal{X} = \mathbb{R}^d$

“Linear” (affine) functions:

$$\underline{L_d = \{h_{\mathbf{w},b} : \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}\}}$$

where



The diagram shows the equation $\underline{h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left(\sum_{i=1}^d w_i x_i \right) + b}$. A blue circle is drawn around the $h_{\mathbf{w},b}$ term, with a blue arrow pointing from the word "parameters" to it. Another blue arrow points from the text "vector of features" to the \mathbf{x} term in the inner product.

$$\underline{h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \left(\sum_{i=1}^d w_i x_i \right) + b}$$

Note:

- each member of L_d is a function $\mathbf{x} \rightarrow \langle \mathbf{w}, \mathbf{x} \rangle + b$
- b : *bias*

Linear Models

Hypothesis class \mathcal{H} : $\phi \circ L_d$, where $\phi : \mathbb{R} \rightarrow \mathcal{Y}$

• $h \in \mathcal{H}$ is $h : \mathbb{R}^d \rightarrow \mathcal{Y}$

Turns the output of L_d to the desired output

ϕ depends on the learning problem

Example

- binary classification, $\mathcal{Y} = \{-1, 1\} \Rightarrow \phi(z) = \text{sign}(z)$
- regression, $\mathcal{Y} = \mathbb{R} \Rightarrow \phi(z) = z$

Equivalent Notation

Given $\mathbf{x} \in \mathcal{X}$, $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$, define:

- $\mathbf{w}' = (b, w_1, w_2, \dots, w_d) \in \mathbb{R}^{d+1} \rightarrow$ all parameters
- $\mathbf{x}' = (1, x_1, x_2, \dots, x_d) \in \mathbb{R}^{d+1}$

Then:

$$\underline{h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \langle \mathbf{w}', \mathbf{x}' \rangle} \quad (1)$$

\Rightarrow we will consider bias term as part of \mathbf{w} and assume

$\mathbf{x} = (1, x_1, x_2, \dots, x_d)$ when needed, with $h_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$

Linear Classification

$\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$, 0-1 loss

Hypothesis class = *halfspaces*

$$HS_d = \text{sign} \circ L_d = \{\mathbf{x} \rightarrow \text{sign}(h_{\mathbf{w},b}(\mathbf{x})) : h_{\mathbf{w},b} \in L_d\}$$

Example: $\mathcal{X} = \mathbb{R}^2$

