Machine Learning

Learning Model

Fabio Vandin October 16th, 2023

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

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!

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.

Before: $\mathcal{D}$ such that $\forall \vec{x} \in \mathcal{X}$, there was a label y : $\Pr[y \mid \vec{x}] = 1$

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 → Now $y \neq$ from "$f(x)$" for some $f()$

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:

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

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:

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

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

Empirical risk: as before, that is

$$L_{\mathcal{S}}(h) \overset{def}{=} \frac{|\{i, 0 < i \leq m : h(x_i) \neq y_i\}|}{m}$$

Note: $L_{\mathcal{S}}(h) =$ probability that for a pair (x_i, y_i) taken uniformly at random from S the event "$h(x_i) \neq y_i$" holds.

↳ from this: $\mathbb{E}[L_S(h)] = L_{\mathcal{D}}(h)$

An Optimal Predictor

Learner's goal: find $h : \mathcal{X} \to \mathcal{Y}$ 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 $g : \mathcal{X} \to \{0, 1\}$, it holds $L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$.

PROOF: Exercize

Can we use such predictor? No, because we don't know $Pr[y=1|x]$ (we don't know $\mathcal{D}$)

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 $\geq 1 - \delta$ (over the choice of the m training examples):

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

this was 0 in simplified PAC

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 $m_{\mathcal{H}}\colon (0,1)^2 \to \mathbb{N}$ 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 $\geq 1 - \delta$ (over the choice of the m training examples):

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

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

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

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

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

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

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

(Generalized) Loss Functions

Definition

Given any hypotheses set $\mathcal{H}$ and some domain $\underset{\mathcal{X} \times \mathcal{Y}}{\underset{\shortparallel}{Z}}$, a *loss function* is any function $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$

(Generalized) Loss Functions

Definition

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

Generalization error

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

$$L_{\mathcal{D}}(h) \overset{def}{=} \mathbb{E}_{\underset{\underset{(x,y)}{\downarrow}}{z}\sim\mathcal{D}}[\ell(h,z)]$$

(Generalized) Loss Functions

Definition

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

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

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

Empirical risk = expected loss over a given sample $S = (z_1, \ldots, z_m) \in Z^m$:

(x_1, y_1) (x_m, y_m)

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

comparing $h(x_i)$ with y_i, how much do I lose? $\ell(h, z_i)$

$z_i = (x_i, y_i)$

Some Common Loss Functions

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

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

Some Common Loss Functions

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

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

Commonly used in binary or multiclass classification.

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

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

$$\ell(h, (x,y)) = |h(x) - y|$$

Some Common Loss Functions

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

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

Commonly used in binary or multiclass classification.

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

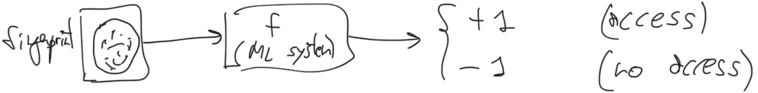
$$\ell_{sq}(h,(x,y)) \overset{def}{=} (h(x) - y)^2$$

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?

Example classification of fingerprints



Two types of errors: false accepts and false rejects

How do we choose the loss?

0-1 loss

		"true" value	
		+1	-1
predicted value	+1	0 no error	1 false accept
	-1	1 false reject	0 no error

What if this is access to nuclear weapons?

0	100
1	0

What about discount of a supermarket?

0	1
10	0

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 $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$ if there exist a function $m_{\mathcal{H}}$: $(0,1)^2 \to \mathbb{N}$ 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 $L_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$

Leslie Valiant, Turing award 2010

For transformative contributions to the theory of computation, including the theory of probably approximately correct (PAC) learning, the complexity of enumeration and of algebraic computation, and the theory of parallel and distributed computing.

Bibliography

Up to now:
[UML] Chapter 2 and Chapter 3