Machine Learning

Learning Model

Fabio Vandin

October 11^{th}, 2022

A Formal Model (Statistical Learning)

We have a *learner* (us, or the machine) has access to:

1. **Domain set** $\mathcal{X}$: set of all possible objects to make predictions about
   - domain point $x \in \mathcal{X}$ = *instance*, usually represented by a vector of *features*
   - $\mathcal{X}$ is the *instance space*
2. **Label set** $\mathcal{Y}$: set of possible labels.
   - often two labels, e.g $\{-1, +1\}$ or $\{0, 1\}$
3. **Training data** $S = ((x_1, y_1), \ldots, (x_m, y_m))$: finite sequence of labeled domain points, i.e. pairs in $\mathcal{X} \times \mathcal{Y}$
   - this is the learner's **input**
   - S: *training example* or *training set*

4. **Learner's output** h: prediction rule $h : \mathcal{X} \to \mathcal{Y}$
   - also called *predictor*, *hypothesis*, or *classifier*
   - $A(S)$: prediction rule produced by learning algorithm A when training set S is given to it
   - sometimes $\hat{f}$ used instead of h
5. **Data-generation model**: instances are generated by some probability distribution and labeled according to a function
   - $\mathcal{D}$: probability distribution over $\mathcal{X}$ (**NOT KNOWN TO THE LEARNER!**)
   - labeling function f: $\mathcal{X} \to \mathcal{Y}$ (**NOT KNOWN TO THE LEARNER!**)
   - label y_i of instance x_i: $y_i = f(x_i)$, for all $i = 1, \ldots, m$
   - each point in training set S: first sample x_i according to $\mathcal{D}$, then label it as $y_i = f(x_i)$
6. **Measures of success**: *error of a classifier* = probability it does not predict the correct label on a random data point generate by distribution $\mathcal{D}$

Loss

Given domain subset $A \subset \mathcal{X}$, $\mathcal{D}(A)$ = probability of observing a point $x \in A$.

Let A be defined by a function $\pi : \mathcal{X} \to \{0, 1\}$:

$$A = \{x \in \mathcal{X} : \pi(x) = 1\}$$

In this case we have $\mathbb{P}_{x \sim \mathcal{D}}[\pi(x)] = \mathcal{D}(A)$

Error of prediction rule $h : \mathcal{X} \to \mathcal{Y}$ is

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

Notes:

- $L_{\mathcal{D},f}(h)$ has many different names: **generalization error**, *true error*, *risk*, **loss**, ...
- often f is obvious, so omitted: $L_{\mathcal{D}}(h)$

Learner outputs $h_S : \mathcal{X} \to \mathcal{Y}$.

↳ training set

Learner outputs $h_S : \mathcal{X} \to \mathcal{Y}$.

Goal: find h_S which minimizes the generalization error $L_{\mathcal{D},f}(h)$

Learner outputs $h_S : \mathcal{X} \to \mathcal{Y}$.

Goal: find h_S which minimizes the generalization error $L_{\mathcal{D},f}(h)$

$L_{\mathcal{D},f}(h)$ is unknown!

What about considering the error on the training data, that is, reporting in output h_S that minimizes the error on training data?

Training error: $L_S(h) \stackrel{def}{=} \frac{|\{i : h(x_i) \neq y_i, 1 \leq i \leq m\}|}{m}$

# of instance in S for which h predicts the wrong label

of hypothesis h

training set $S = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_m, y_m)\}$

Empirical Risk Minimization

Learner outputs $h_S : \mathcal{X} \to \mathcal{Y}$.

Goal: find h_S which minimizes the generalization error $L_{\mathcal{D},f}(h)$

$L_{\mathcal{D},f}(h)$ is unknown!

What about considering the error on the training data, that is, reporting in output h_S that minimizes the error on training data?

Training error: $L_S(h) \stackrel{def}{=} \frac{|\{i:h(x_i)\neq y_i, 1\leq i\leq m\}|}{m}$

Note: the *training error* is also called *empirical error* or *empirical risk*

Empirical Risk Minimization

Learner outputs $h_S : \mathcal{X} \to \mathcal{Y}$.

Goal: find h_S which minimizes the generalization error $L_{\mathcal{D},f}(h)$

$L_{\mathcal{D},f}(h)$ is unknown!

What about considering the error on the training data, that is, reporting in output h_S that minimizes the error on training data?

Training error: $L_S(h) \stackrel{def}{=} \frac{|\{i : h(x_i) \neq y_i, 1 \leq i \leq m\}|}{m}$

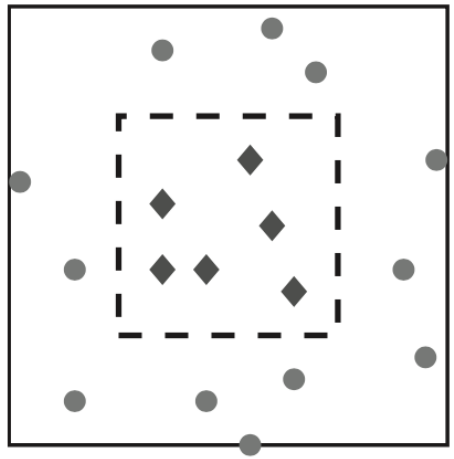
Note: the *training error* is also called *empirical error* or *empirical risk*

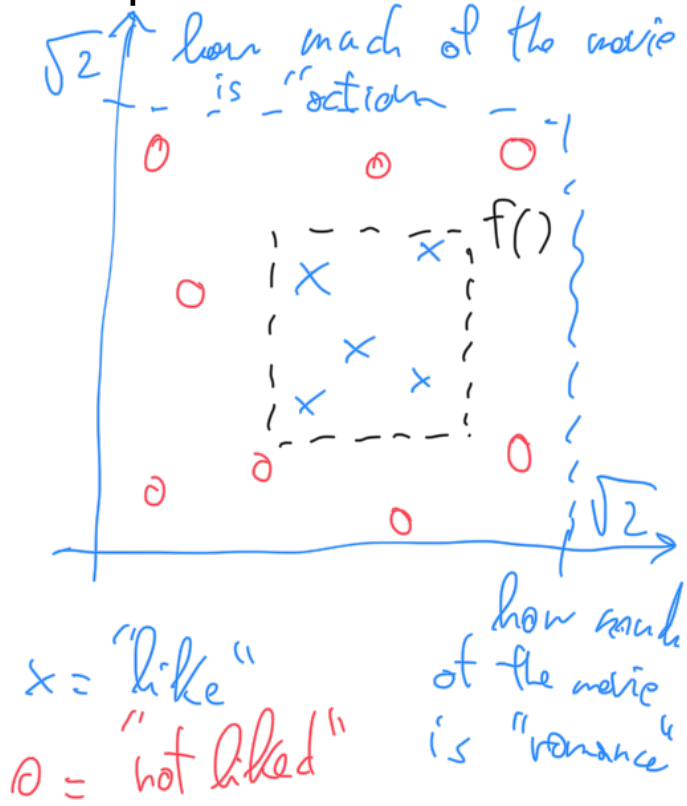
Empirical Risk Minimization (ERM): produce in output h minimizing $L_S(h)$

What can go wrong with ERM?

Consider our simplified movie ratings prediction problem. Assume data is given by:

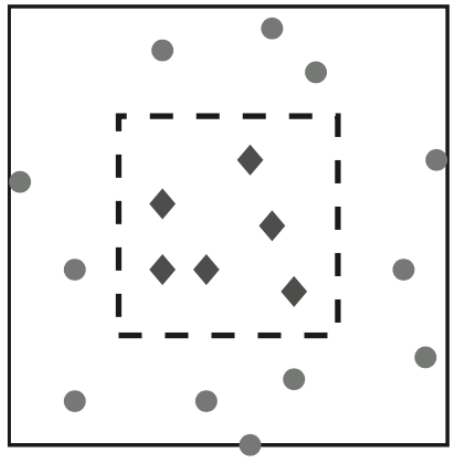
$\mathcal{X} = \mathbb{R}^2$





What can go wrong with ERM?

Consider our simplified movie ratings prediction problem. Assume data is given by:



$$L_{D,f}(h_S) = \Pr_{\vec{x}\sim D}\left[h_S(\vec{x}) \neq f(\vec{x})\right]$$
$$= \Pr_{\vec{x}\sim D}\left[\vec{x} \text{ is in the inner square}\right]$$
$$= \frac{1}{2}$$

Assume $\mathcal{D}$ and f are such that:

- instance x is taken uniformly at random in the square ($\mathcal{D}$)
- label is 1 if x inside the inner square, 0 otherwise (f)
- area inner square $= 1$, area larger square $= 2$

Consider classifier given by

$$h_S(x) = \begin{cases} y_i & \text{if } \exists i \in \{1,\ldots,m\} : x_i = x \\ 0 & \text{otherwise} \end{cases}$$

$S = \{(\vec{x}_i, y_i) : 1 \le i \le m\}$

Training error $L_S(h_S) = 0$

Is it a good predictor?

$L_S(h_S) = 0$ but $L_{\mathcal{D},f}(h_S) = 1/2$

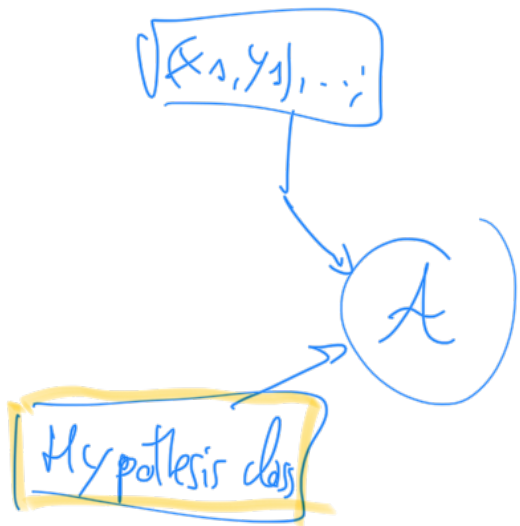
Good results on training data but poor generalization error
$\Rightarrow$ **overfitting**

When does ERM lead to good performances in terms of generalization error?

Hypothesis Class and ERM

Apply ERM over a **restricted set** of hypotheses $\mathcal{H}$ = *hypothesis class*

- each $h \in \mathcal{H}$ is a function $h : \mathcal{X} \to \mathcal{Y}$



Hypothesis Class and ERM

Apply ERM over a **restricted set** of hypotheses $\mathcal{H}$ = *hypothesis class*

- each $h \in \mathcal{H}$ is a function $h : \mathcal{X} \to \mathcal{Y}$

$\text{ERM}_{\mathcal{H}}$ learner:

$$\text{ERM}_{\mathcal{H}} \in \arg\min_{h \in \mathcal{H}} L_S(h)$$

Hypothesis Class and ERM

Apply ERM over a **restricted set** of hypotheses $\mathcal{H}$ = *hypothesis class*

- each $h \in \mathcal{H}$ is a function $h : \mathcal{X} \to \mathcal{Y}$

$\text{ERM}_{\mathcal{H}}$ learner:

$$\text{ERM}_{\mathcal{H}} \in \arg\min_{h \in \mathcal{H}} L_S(h)$$

Which hypothesis classes $\mathcal{H}$ do not lead to overfitting?

Finite Hypothesis Classes

Assume $\mathcal{H}$ is a finite class: $|\mathcal{H}| < \infty$

movie example: $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^2$ $\quad Y = \{-1, 1\}$

$$\mathcal{H} = \left\{ h(\vec{x}) : h(\vec{x}) = \text{sign}(a\, x_1 + b\, x_2),\ a, b \in \mathbb{R} \right\}$$

$$|\mathcal{H}| = +\infty$$

Finite Hypothesis Classes

Assume $\mathcal{H}$ is a finite class: $|\mathcal{H}| < \infty$

Let h_S be the output of $\text{ERM}_{\mathcal{H}}(S)$, i.e. $h_S \in \arg\min_{h \in \mathcal{H}} L_S(h)$

Finite Hypothesis Classes

Assume $\mathcal{H}$ is a finite class: $|\mathcal{H}| < \infty$

Let h_S be the output of $\text{ERM}_{\mathcal{H}}(S)$, i.e. $h_S \in \arg\min_{h \in \mathcal{H}} L_S(h)$

Assumptions

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

Finite Hypothesis Classes

Assume $\mathcal{H}$ is a finite class: $|\mathcal{H}| < \infty$

Let h_S be the output of $\text{ERM}_{\mathcal{H}}(S)$, i.e. $h_S \in \arg\min_{h \in \mathcal{H}} L_S(h)$

Assumptions

- **Realizability:** there exists $h^* \in \mathcal{H}$ such that $L_{\mathcal{D},f}(h^*) = 0$
- **i.i.d.:** examples in the training set are independently and identically distributed (i.i.d) according to $\mathcal{D}$, that is $S \sim \mathcal{D}^m$

$L_S(h^*) = 0$

Finite Hypothesis Classes

Assume $\mathcal{H}$ is a finite class: $|\mathcal{H}| < \infty$

Let h_S be the output of $\text{ERM}_{\mathcal{H}}(S)$, i.e. $h_S \in \arg\min_{h \in \mathcal{H}} L_S(h)$

Assumptions

- **Realizability:** there exists $h^* \in \mathcal{H}$ such that $L_{\mathcal{D},f}(h^*) = 0$
- **i.i.d.:** examples in the training set are independently and identically distributed (i.i.d) according to $\mathcal{D}$, that is $S \sim \mathcal{D}^m$

Observation: realizability assumption implies that $L_S(h^*) = 0$

Can we *learn* (i.e., find using ERM) h^*?

(Simplified) PAC learning

Probably Approximately Correct (PAC) learning

Since the training data comes from $\mathcal{D}$:

- we can only be **approximately** correct
- we can only be **probably** correct

Parameters:

- *accuracy parameter* ε: we are satisfied with a *good* h_S: $L_{\mathcal{D},f}(h_S) \leq \varepsilon$

(Simplified) PAC learning

Probably Approximately Correct (PAC) learning

Since the training data comes from $\mathcal{D}$:

- we can only be **approximately** correct
- we can only be **probably** correct

Parameters:

- *accuracy parameter* ε: we are satisfied with a *good* h_S: $L_{\mathcal{D},f}(h_S) \leq \varepsilon$ (ε small)
- *confidence parameter* δ: want h_S to be a *good* hypothesis with probability $\geq 1 - \delta$ (δ small)

Theorem

Let $\mathcal{H}$ be a finite hypothesis class. Let $\delta \in (0,1)$, $\varepsilon \in (0,1)$, and $m \in \mathbb{N}$ such that

$$m \geq \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon}.$$

m = # of training samples $= |S|$

Then for any f and any $\mathcal{D}$ for which the realizability assumption holds, with probability $\geq 1-\delta$ we have that for every ERM hypothesis h_S it holds that

$$L_{\mathcal{D},f}(h_S) \leq \varepsilon.$$

$\mathcal{D}$: data distribution (unknown)

f: true labeling function (unknown)

Note: log = natural logarithm

With finite hypothesis classes, I can "always" find a good hypothesis $\rightarrow L_{\mathcal{D},f}(h_S) \leq \varepsilon$ with prob. $\geq 1-\delta$ if I have enough data.

for any $\mathcal{H}$
for any $\mathcal{D}$
for any f

if $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon}$

Proof (see book as well, Corollary 2.3)

Let $S|_x = \{x_1, x_2, \ldots, x_m\}$ be the instances in the training set S.

We want to bound (i.e., an upper bound) to: $\mathcal{D}^m(\{S|_x : L_{\mathcal{D},f}(h_S) > \varepsilon\})$.

Let $\mathcal{H}_B = \{h \in \mathcal{H} : L_{\mathcal{D},f}(h) > \varepsilon\}$ (BAD HYPOTHESES)

and $M = \{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$ (MISLEADING SAMPLES)

Since we have the realizability assumption: $L_S(h_S) = 0$

$\Rightarrow L_{\mathcal{D},f}(h_S) > \varepsilon$ only if some $h \in \mathcal{H}_B$ has $L_S(h) = 0$.

That is, our training data must be in the set M (for this to happen): $\{S|_x : L_{\mathcal{D},f}(h_S) > \varepsilon\} \subseteq M$.

Note that: $M = \bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}$ because of

Therefore $\mathcal{D}^m(\{S|_x : L_{\mathcal{D},f}(h_S) > \varepsilon\}) \leq \mathcal{D}^m(M) = \mathcal{D}^m\left(\bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}\right)$

$\left(\text{union bound}\right) \leq \sum_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\})$ (*)

Now let's fix $h \in \mathcal{H}_B : L_S(h) = 0 \iff \forall i = 1, \dots, m : h(x_i) = f(x_i)$

Therefore: $D^m(\{S|_x : L_S(h) = 0\}) = D^m(\{S|_x : \forall i = 1, \dots, m, h(x_i) = f(x_i)\})$

(because $x_1, \dots, x_m$ are i.i.d. from D) $\rightarrow = \prod_{i=1}^{m} D(\{x_i : h(x_i) = f(x_i)\})$

Consider some i, $1 \le i \le m$: $D(\{x_i : h(x_i) = f(x_i)\}) = 1 - D(\{x_i : h(x_i) \neq f(x_i)\})$

$L_{D,f}(h) = \Pr_{x \sim D}[h(x) \neq f(x)]$

(since $h \in \mathcal{H}_B$) $= 1 - L_{D,f}(h)$

$\le 1 - \varepsilon$

$\le e^{-\varepsilon}$

(because ... Taylor's expansion)