Machine Learning

Learning Model

Fabio Vandin

October 9^{th}, 2023

Machine Learning

Learning Model

Fabio Vandin

October 9^{th}, 2023

Machine Learning

Learning Model

Fabio Vandin

October 9^{th}, 2023

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 $h : \mathcal{X} \rightarrow \mathcal{Y}$
   - 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: $\mathcal{X} \rightarrow \mathcal{Y}$ (**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**, $\ldots$
- often f is obvious, so omitted: $L_{\mathcal{D}}(h)$

Empirical Risk Minimization

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

from the training set

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?

It's a function of the hypotesis

m = # istances in the training set

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

# of instances $\in S$ for which h predicts the wrong label

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

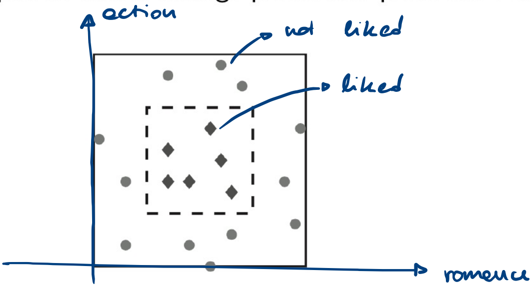
smallest training error

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

we assume there's a link between the training set and the "future data" (same probability distribution)

What can go wrong with ERM?

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



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

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

→ if x is in the training set

Is it a good predictor?

$L_S(h_S) = 0$ but $L_{\mathcal{D},f}(h_S) = 1/2$ whenever x is in the inner square (and was not in the training set)

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

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