DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

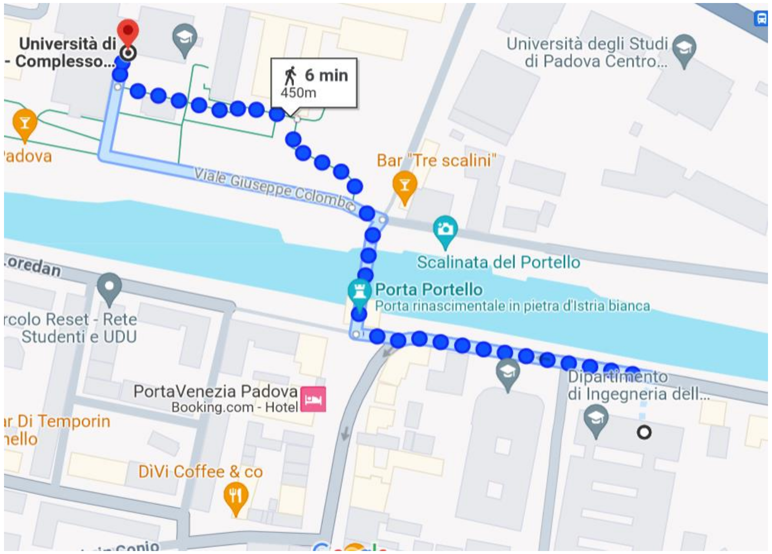
UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Machine Learning Model

Machine Learning 2023-24
UML Book Chapter 2
Slides P. Zanuttigh (some material F. Vandin)

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

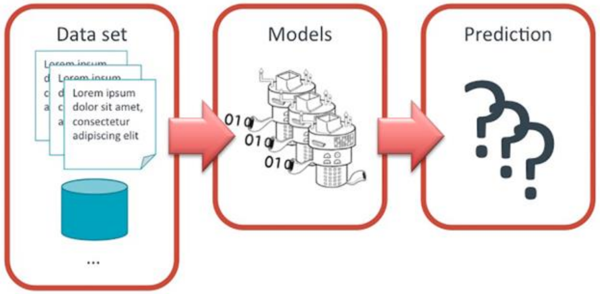
Friday: Lecture in Rn



Lectures:

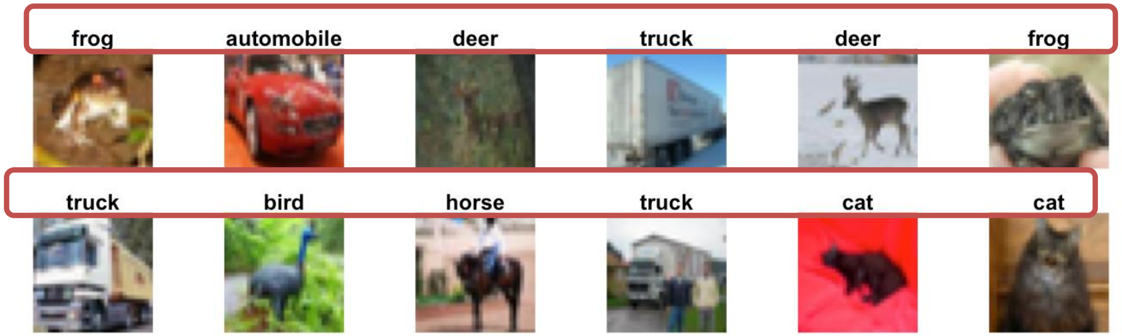
- Tue 16:30 -18:00 Room Ae
- Fri 12:30 - 14:00 lecture or lab
- **Next Friday the lecture will be in Room Rn (Vallisneri building)**

Machine Learning



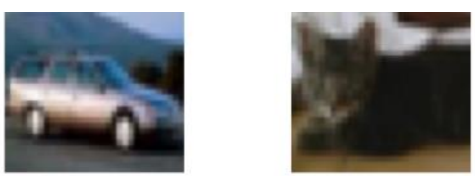
- Machine learning (ML) is a set of methods that give computer systems the ability to "*learn*" from (*training*) data to make predictions about novel data samples, without being explicitly programmed for the considered task
- ML techniques: data driven methods
- Training data can be provided with or without corresponding correct predictions (labels)
  - *Unsupervised learning*: no labels are provided for training data
  - *Supervised learning*: training data with labels

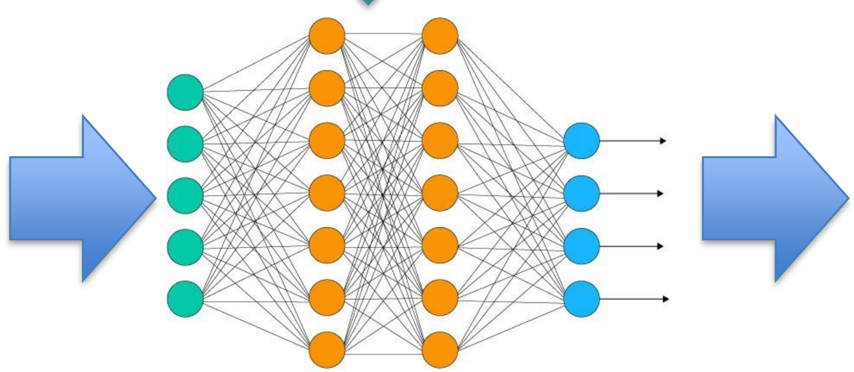
Supervised Learning



Training data
with labels

Training procedure



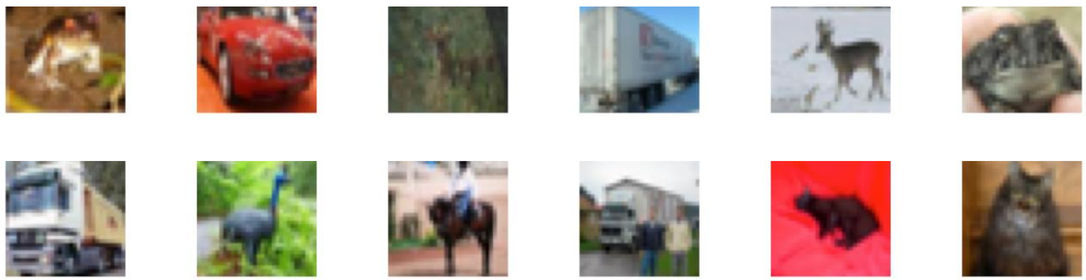


Data to be analyzed

ML model
(training: estimate parameters)

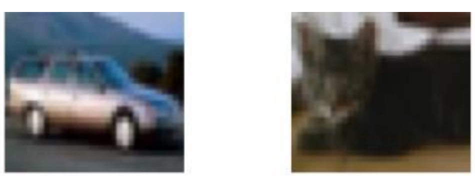
In most of the course we will focus on supervised learning

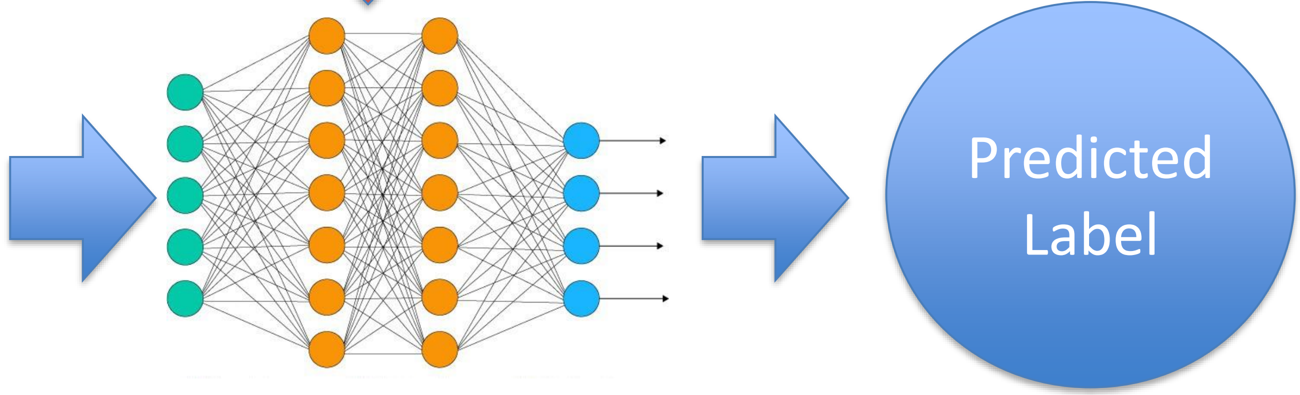
Unsupervised Learning



Training data (unlabeled)

Training procedure





Data to be analyzed

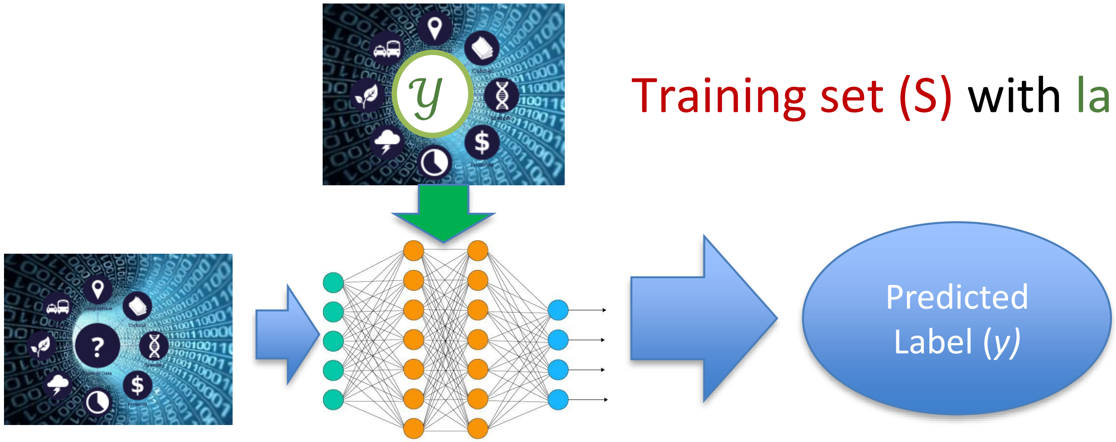
ML model
(training: estimate parameters)

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Supervised Learning: Data Representation

Training set (S) with labels (y)

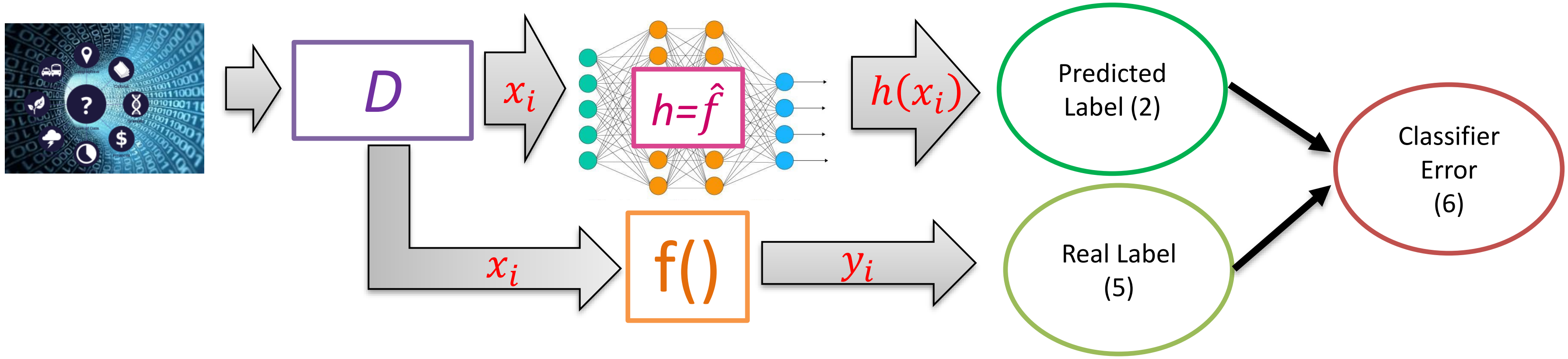
Domain set ($\mathcal{X}$)



The machine learning algorithm has access to:

1. Domain set (or *instance space*) $\mathcal{X}$: set of all possible objects to make predictions about
   - $x \in \mathcal{X}$ is a domain point or instance
   - It is typically (but not always) represented by a vector of numbers (*features*)
2. Label set $\mathcal{Y}$: set of possible labels
   - E.g., simplest case: binary classification $\mathcal{Y} = \{0,1\}$
3. Training set $S = \left((x_1, y_1), \dots, (x_m, y_m)\right)$: finite sequence of *labeled* (*→supervised learning*) domain points (in $\mathcal{X} x \mathcal{Y}$)
   - It is the input of the ML algorithm !

Make Predictions on Data

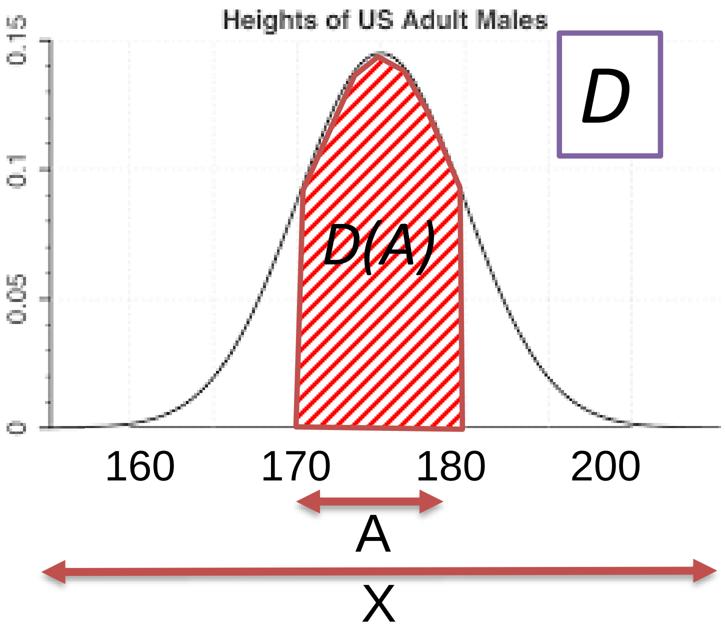


4. Prediction rule h: $\mathcal{X} \to \mathcal{Y}$ (sometimes called also $\hat{f}$)
   - The learner's output, called also predictor, hypothesis or classifier
   - $A(S)$: prediction rule produced by ML alg. A when training set S is given to it

5. Data-generation model: instances are
   - Generated by a probability distribution $\mathcal{D}$ over $\mathcal{X}$ (*NOT KNOWN BY THE ML ALGORITHM*)
   - Labeled according to a function f (NOT KNOWN BY THE ML ALGORITHM)
   - Training set: $\forall\ x_i \in S$, sample x_i according to $\mathcal{D}$ then label it as $y_i = f(x_i)$

6. Measure of success = error of the classifier = probability it does not predict the correct label on a random data point generated by $\mathcal{D}$

Notes: Data Generating Distribution



$$x \in \mathcal{X} = R^+$$
$$A: 170 < x < 180$$
$$D(A) = D(\{x: 170 < x < 180\}) = 0.3$$

$$\pi(x) = \begin{cases} 1: & 170 < x < 180 \\ 0: & otherwise \end{cases}$$

- Samples $x \in X$ are produced by a probability distribution D: $x \sim D$
- Consider a domain subset $A \subset X$:
  - *A: event,* expressed by $\pi: X \rightarrow \{0,1\}$,i.e., $A = \{x \in X: \pi(x) = 1\}$
  - $D(A)$: probability of observing a point $\mathrm{x} \in A$ (it is a number in the 0-1 range)
  - We get that $P_{x\sim D}[\pi(x) = 1] = D(A)$

Measure of Success: Loss Function

Recall:

- Assume a domain subset $A \subset X$
- *A: event,* expressed by $\pi: X \to \{0,1\}$, i.e., $A = \{x \in X: \pi(x) = 1\}$
- $D(A)$: probability of observing a point $X \in A$
- We get that $P_{x \sim D}[\pi(x)] = D(A)$

Error of prediction rule in classification problems $h: X \to Y$

$$\boxed{L_{D,f}(h) \stackrel{\text{def}}{=} P_{x \sim D}[h(x) \neq f(x)] = D(x\colon\ h(x) \neq f(x))}$$

Predicted label

correct label

Notes:

- $L_{D,f}(h)$: loss depends on distribution D and labelling function f
- $L_{D,f}(h)$ has many different names: generalization error, true error, true risk, loss
- Often f is omitted: $L_D(h)$

Empirical Risk Minimization

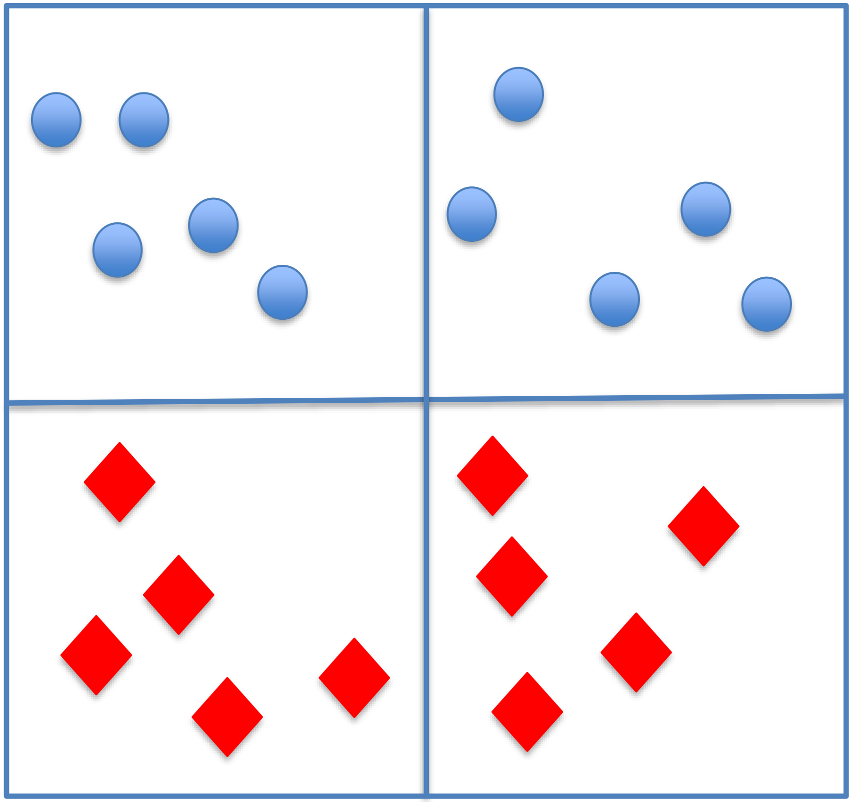
- Learner outputs $h_S : \mathcal{X} \rightarrow \mathcal{Y}$ (note the dependency on S!)
- *Goal*: find h_S which minimizes the generalization error $L_{D,f}(h)$
  - But $L_{D,f}(h)$ is unknown !

- What about considering the error on the training data ?

- Training error: $L_S(h) \triangleq \frac{|\{i: h(x_i) \neq y_i \ ,1 \leq i \leq m\}|}{m} = \frac{\text{\# wrong predictions}}{\text{\# training samples}}$
  - Assuming a classification problem and 0-1 loss, otherwise different definition
  - also called empirical error or empirical risk

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

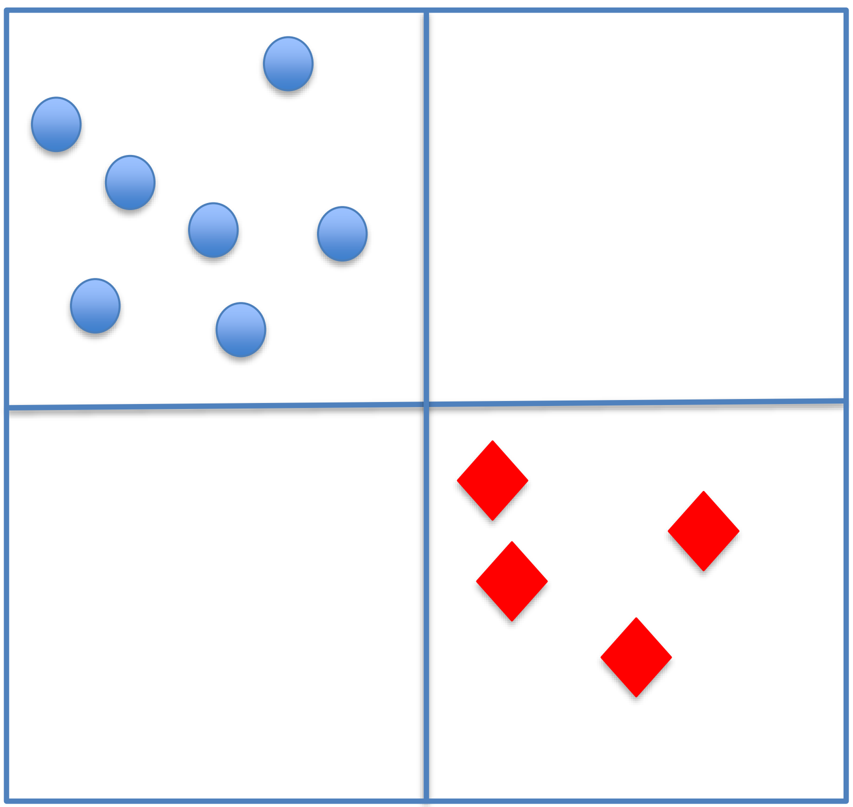
Is training error a good measure of true error ?



Assume following *D*:

- Instance *x* is taken uniformly at random in the square
- *f* : label is 0 if *x* in upper side, 1 if lower side (red vs blue)
- Area of the two sides is the same

Is training error a good measure of true error ?



Training set: samples in the figure

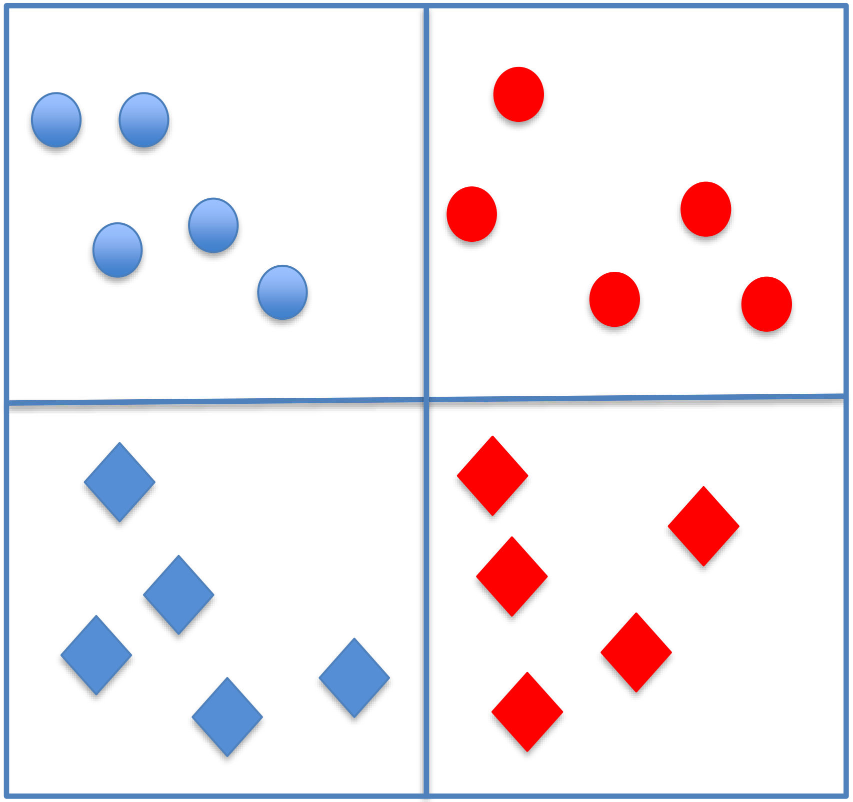
Consider this predictor:

$$h_S(x) = \begin{cases} 0 & \text{if } x \text{ in left side} \\ 1 & \text{if } x \text{ in right side} \end{cases}$$

- $L_S(h_S) = 0$
- Minimizes training loss (i.e., empirical risk) !
- Is it a good predictor ?

Is training error a good measure ?

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE



- $L_{D,f}(h_S) = \frac{1}{2}$
- Same loss as random guess
- Poor performances: *overfitting* on training data!
- In this case very good performances on training set an poor performances in general
- When does ERM lead to good performances w.r.t. generalization error?

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Hypothesis Class

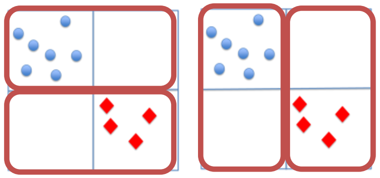
- Apply ERM over a restricted set of possible hypotheses
  - $\mathcal{H}$ = hypothesis class
  - Each $h \in \mathcal{H}$ is a function $h: \mathcal{X} \rightarrow \mathcal{Y}$
  - Restricting to a set of hypothesis→making assumptions (*priors*) on the problem at hand

- $ERM_{\mathcal{H}}$ learner:

∈ : there can be multiple optimal solutions

$$ERM_{\mathcal{H}} \in \underset{h \in \mathcal{H}}{\operatorname{argmin}} L_S(h)$$

Recall previous example!



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

Assumptions

1. Assume $\mathcal{H}$ is a finite hypothesis class, i.e., $\mathcal{H} < \infty$
2. Let h_s be the output of $ERM_{\mathcal{H}}$(S), i.e., $h_s \in \underset{h \in \mathcal{H}}{\operatorname{argmin}} L_s(h)$

Two further assumptions:

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

→*Note: these assumptions are very difficult to be satisfied in practice*

- ❑ Realizability assumption implies that $L_S(h^*) = 0$
- ❑ Can we learn h^* ?

PAC Learning

Probably Approximately Correct (PAC) learning

Since the training data comes from 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 for which $L_{D,f}(h_S) \leq \epsilon$
- confidence parameter δ: want h_S to be a good hypothesis with probability $\geq 1 - \delta$

Theorem

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

$$m \geq \frac{\log\left(\frac{|\mathcal{H}|}{\delta}\right)}{\epsilon}$$

Notice: m grows with $|\mathcal{H}|$ and is inversely proportional to δ and ϵ

Then, for any f and any D for which the realizability assumption holds, with probability $\boxed{\geq 1-\delta}$ we have that for every ERM hypothesis h_S, computed on a training set S of size m sampled i.i.d. from D, it holds that

$$L_{D,f}(h_S) \leq \boxed{\epsilon}$$

probably approximately correct

m: size of the training set (i.e., S contains *m* I.I.D. samples)

Idea of the Demonstration

- The critical issue are the training sets leading to a "misleading" predictor h with $L_s(h) = 0$ but $L_{D,f}(h) > \epsilon$
- Place an upper bound to the probability of sampling *m* instances leading to a *misleading training set,* i.e., producing a "misleading" predictor
- Using the union bound after various mathematical computations the bound of the theorem can be obtained
- *Message of the theorem*: if $\mathcal{H}$ is a finite class then ERM will not overfit, provided it is computed on a sufficiently big training set
- *Demonstration not part of the course, but you can find it on the book if you are interested*

Theorem: Graphical Illustration

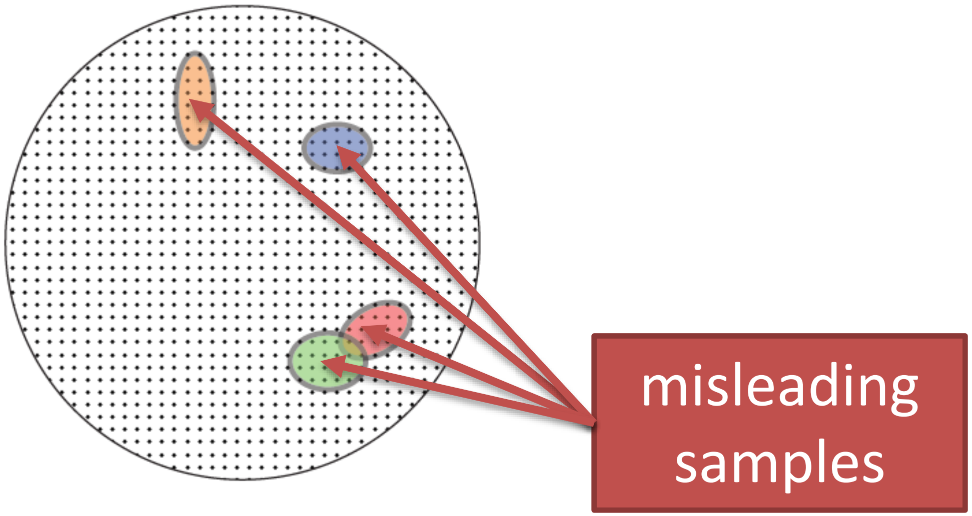


Figure 2.1 Each point in the large circle represents a possible m-tuple of instances. Each colored oval represents the set of "misleading" m-tuple of instances for some "bad" predictor $h \in \mathcal{H}_B$. The ERM can potentially overfit whenever it gets a misleading training set S. That is, for some $h \in \mathcal{H}_B$ we have $L_S(h) = 0$. Equation (2.9) guarantees that for each individual bad hypothesis, $h \in \mathcal{H}_B$, at most $(1-\epsilon)^m$-fraction of the training sets would be misleading. In particular, the larger m is, the smaller each of these colored ovals becomes. The union bound formalizes the fact that the area representing the training sets that are misleading with respect to some $h \in \mathcal{H}_B$ (that is, the training sets in M) is at most the sum of the areas of the colored ovals. Therefore, it is bounded by $|\mathcal{H}_B|$ times the maximum size of a colored oval. Any sample S outside the colored ovals cannot cause the ERM rule to overfit.

Demonstration: some notes (1)

$$\underbrace{D(\{x_i : h(x_i) = y_i\}) = 1 - L_{D,f}(h)}_{\boxed{1}} \underbrace{\leq 1 - \epsilon}_{\boxed{2}}$$

- *In this step we are considering a single sample x_i*

1. First step: $D(\{x_i : h(x_i) = y_i\})$ *is the probability of a correct prediction (i.e., 1 - probability of error)*
2. Second step: $h \in \mathcal{H}_B$ *(set of bad hypotheses)* → *probability of error for h is bigger than ϵ, i.e., $L_{D,f}(h) > \epsilon$*

Demonstration not part of the course

Here are just some notes for critical steps, refer to the book and lecture notes for the complete demonstration

Demonstration: some notes (2)

$$D^m\left(\left\{S\Big|_x : L_{D,f}(h_S) > \epsilon\right\}\right) \le \sum_{h\in\mathcal{H}_B} D^m\left(\left\{S\Big|_x : L_S(h) = 0\right\}\right)$$

$$D^m\left(\left\{S\Big|_x : L_S(h) = 0\right\}\right) \le e^{-\epsilon m}$$

- *First equation: from union bound*
- *Second equation: consequence of previous slide result*
- *By combining the 2 equations (substituting the red part)*

$$D^m\left(\left\{S\Big|_x : L_{D,f}(h_S) > \epsilon\right\}\right) \le \sum_{h\in\mathcal{H}_B} e^{-\epsilon m} = |\mathcal{H}_B| e^{-\epsilon m} \le |\mathcal{H}| e^{-\epsilon m}$$

- $\mathcal{H}_B$ *is a subset of* $\mathcal{H}$ $\rightarrow$ *last inequality*
- Notice the difference between *true error* $L_{D,f}(h_S)$ *and empirical errror* $L_S(h)$

Demonstration not part of the course

Demonstration: some notes (3)

- *Thesis of the theroem*: the probability of having a small error is $\geq 1\text{-}\delta$
  - corresponds to probability of large error is $\leq \delta$
  - i.e. ,we need to demonstrate that: $D^m(\{S|_x: L_{D,f}(h_S) > \epsilon \}) \leq \delta$
- We have obtained:

$$D^m\left(\left\{S\Big|_x : L_{D,f}(h_S) > \epsilon \right\}\right) \leq |\mathcal{H}|e^{-\epsilon m}$$

- *Finally*: purple part is smaller than red, to satisify the theorem we need to find *m* for which red is smaller than δ :
  - Set $m \geq \log(\frac{|\mathcal{H}|}{\delta})/\epsilon$

Demonstration not part of the course