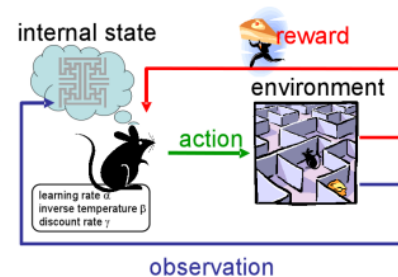


# Lezione 2 06/10/2023

## Key elements to be defined

- Problem (Task) definition
  - Instances + corresponding output (“label” or “target”)
  - or: Instances      esempi senza output      si può usare il clustering
  - Interactive environment (action => reward/penalty)
- Correspondingly:
  - Supervised learning
  - Unsupervised learning      mancanza di risposte esatte
  - Reinforcement learning

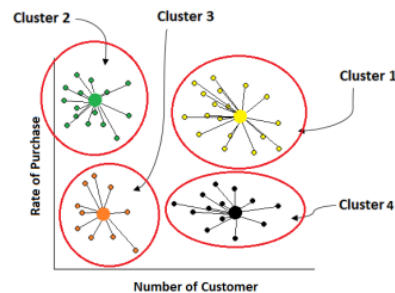


## Prior information about the problem, or other help

- Do we know anything about a possible correct  $f(I)=O$  ?
- Shall we choose our answer constrained by some set of hypothesis?
- May we ask about specific chosen input examples?  
In such a case we have *\*active\** learning, VS passive

## How a solution to the problem is represented

- Boolean function? From a specific set?       $I \rightarrow \{0,1\}$
- A (multi)classification function?       $I \rightarrow \{A, B, C, \dots\}$
- A function computing the output       $O = f(I)$  real valued or else
- A clustering (groups) of instances



## Come è valutata una soluzione

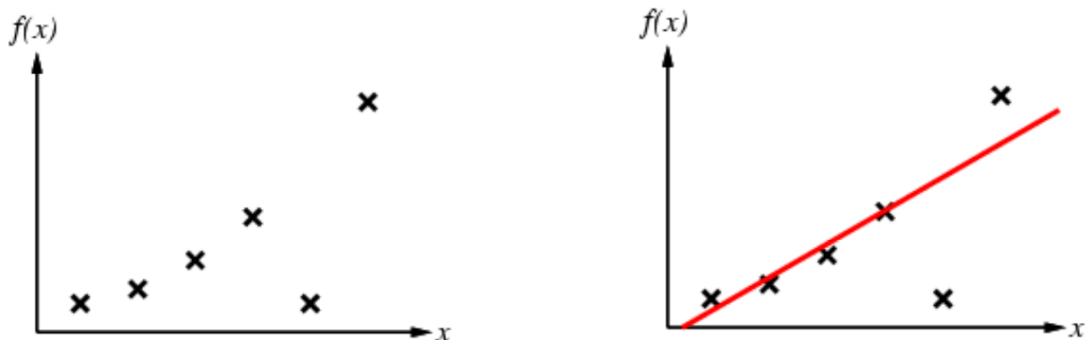
Dipende dalle situazioni, può essere sempre giusta o sempre sbagliata, è un metodo estremo di valutazione e quindi non molto realistico. Nel caso booleano si potrebbero avere falsi positivi o falsi negativi, questi possono essere misurati. Con la regressione si può partire dai risultati per capire dove sono stati gli errori.

## Inductive learning method

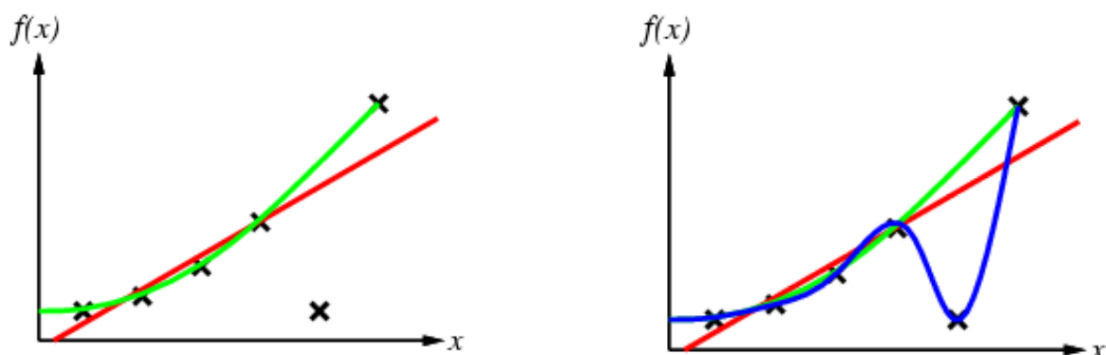
Abbiamo una funzione booleana  $f$ , gli esempi sono coppie  $(x, f(x))$ .

Il problema è trovare un'ipotesi  $h$  di modo che  $h(x)$  sia simile a  $f(x)$  dato che  $f$  è sconosciuta, ho solo esempi di  $f(x)$ .

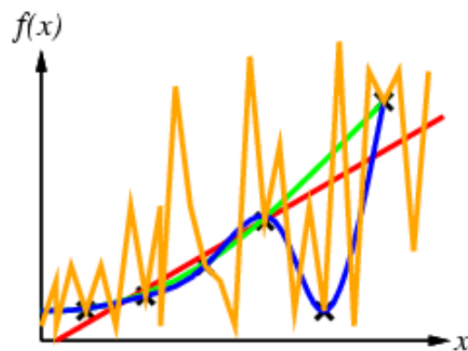
Partiamo da un approccio grafico. Si traccia una linea che diventerà quindi  $h(x)$ , ed è la mia ipotesi di funzione.



Il sistema può provare con una riga dritta ma non è abbastanza precisa, quindi si passa ad una soluzione più elaborata.  $h$  è consistente solo se è d'accordo con tutti gli esempi di  $f(x)$ .

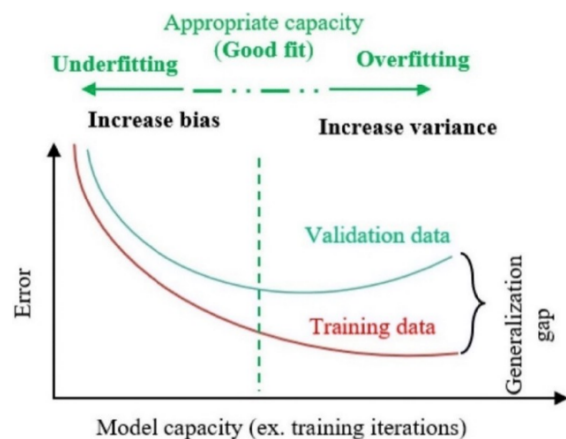


Il sistema però può anche esagerare, anche se il **fitting** è rispettato (la curva rispetta comunque gli esempi). La risposta è però troppo complicata, questo non va bene.



Vale il principio di Occam, è **preferibile l'ipotesi più semplice e consistente con i dati**.

Infatti vogliamo che il sistema possa anche predire il futuro, non solo che sia consistente con  $f(x)$ . Quindi nonostante anche l'ultima sia consistente, difficilmente sarà in grado di **predire** altri valori, quella blu sarà probabilmente meglio. L'esempio giallo è chiamato "**overfitting**"



Non sapendo qual è la soluzione giusta, e vedendo che il sistema risponde bene agli esempi, bisognerà ogni volta capire se si è in una situazione di underfitting o overfitting (più comune).

Training loss	Validation loss	Situation	Solution
High	High	Underfitting	Increase capacity
Low	High	Overfitting	Decrease capacity (shrink or regularization techniques)
Low	Low	Good fit	Run test
High	Low	Unlikely	Debug

Raccogliamo le coppie  $\langle x, c(x) \rangle$  degli esempi in un insieme  $D$ ,  $c$  è la risposta giusta mentre  $h$  è l'ipotesi.

Il sistema che produrrà l'ipotesi è consistente se è chiara la coerenza dell'ipotesi rispetto agli esempi presenti.

$$\text{Consistent}(h, D) := \forall \langle x, c(x) \rangle \in D \quad h(x) = c(x)$$

Il **version space** è l'insieme di tutte le ipotesi coerenti, dato  $H$  che è l'insieme delle ipotesi possibili e il training set  $D$ .

$$VS_{H,D} = \{h \in H \mid \text{Consistent}(h, D)\}$$

## Concept learning

inferring a boolean-valued function from training examples of its input and output

Il risultato è quindi booleano, ovvero un problema di classificazione a due valori.

Vediamo un esempio:

**Concept:** "days on which my friend Aldo enjoys his favourite water sports"

**Task:** predict the value of "Enjoy Sport" for an arbitrary day based on the values of the other attributes

attributes

Sky	Temp	Humid	Wind	Water	Fore-cast	Enjoy Sport
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

L'ingresso è un vettore di valori, ciascuno dei quali ha due scelte ma potrebbero anche averne di più. Nel caso del concept learning deve essere a due valori (penso, forse intendeva l'output).

Ci sono quindi gli attributi (etichette) e le istanze (dati).

Come rappresento le ipotesi? Ogni ipotesi ha 6 elementi, nell'ipotesi potrei avere dei punti di domanda, che significa che ci sono più possibilità. Ci potrebbero anche essere colonne dove non ci sono valori accettabili (null hypothesis).

Sky	Temp	Humid	Wind	Water	Forecast
< Sunny	?	?	Strong	?	Same >

Il set di ipotesi  $H$  sarà quindi formato in questo modo. Determineremo quali ipotesi sono consistenti con i dati.

Un'ipotesi che approssima la funzione target negli esempi, sarà anche in grado di approssimare la funzione target in esempi non osservati.

Consideriamo le due ipotesi:

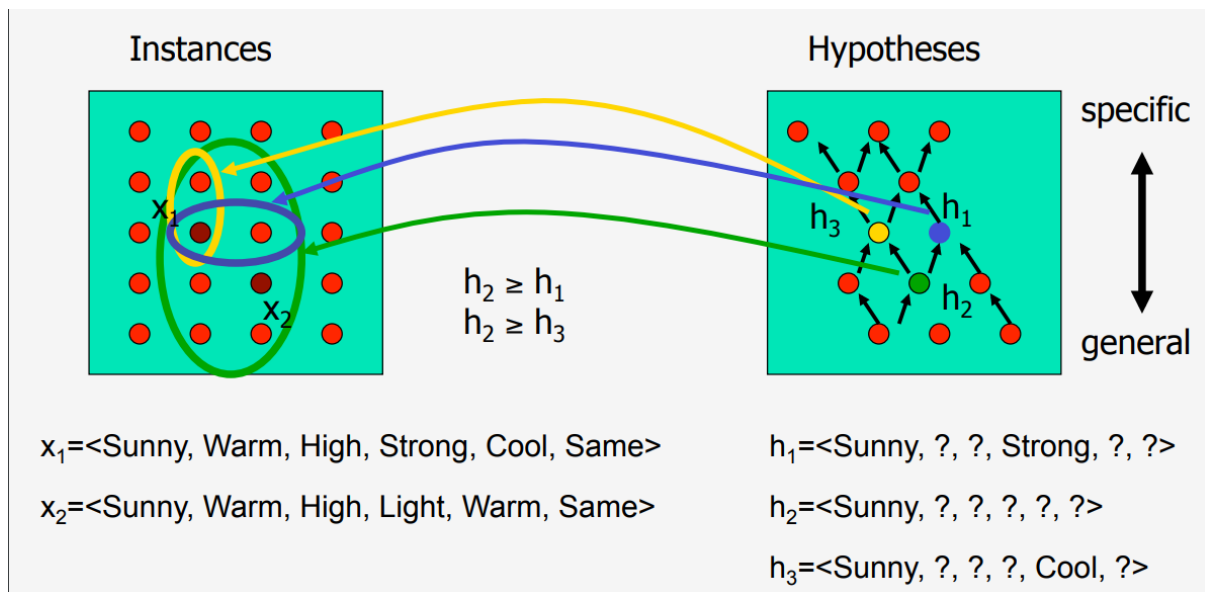
$$h_1 = \langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$$
$$h_2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$$

la prima è più vincolata, la seconda è meno vincolata e quindi classificherà più istanze di  $x$  come positive, es  $h(x)=1$

**Def.** Let  $h_j$  and  $h_k$  be boolean-valued functions defined over  $X$ . Then  $h_j$  is more general than or equal to  $h_k$  (written  $h_j \geq h_k$ ) if and only if

$$\forall x \in X : [(h_k(x) = 1) \rightarrow (h_j(x) = 1)]$$

$h_2$  è quindi più generale.



$x_1$  e  $x_2$  sono le istanze di esempio,  $h_1$ ,  $h_2$  e  $h_3$  sono le ipotesi, alcune più stringenti.  $h_1$  è più specifica,  $h_2$  è più generale ( $h_2 \geq h_1$ ) e poi  $h_3$  è più specifica di  $h_2$  ( $h_2 \geq h_3$ ).  $h_1$  e  $h_3$  invece non sono confrontabili perché divergono in alcuni punti. Le frecce a destra serviranno a scegliere l'ipotesi.

## Find-S algorithm

```

Initialize  $h$  to the most specific hypothesis in  $H$ 
For each positive training instance  $x$ 
  For each attribute constraint  $a_i$  in  $h$ 
    If the constraint  $a_i$  in  $h$  is satisfied by  $x$ 
      then do nothing
    else replace  $a_i$  in  $h$  by the next more
      general constraint that is satisfied by  $x$ 
Output hypothesis  $h$ 

```

L'algoritmo parte dall'ipotesi più specifica e pian piano va ad accettare ipotesi più generali guardando gli esempi.

Scandisce gli attributi dell'ipotesi e se quell'esempio soddisfa l'ipotesi allora va bene, altrimenti bisogna allentare i constraints.

$x_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle +$	$h_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \rangle$
$x_2 = \langle \text{Sunny, Warm, High, Strong, Warm, Same} \rangle +$	$h_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$
$x_3 = \langle \text{Rainy, Cold, High, Strong, Warm, Change} \rangle -$	$h_{2,3} = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$
$x_4 = \langle \text{Sunny, Warm, High, Strong, Cool, Change} \rangle +$	$h_4 = \langle \text{Sunny, Warm, ?, Strong, ?, ?} \rangle$

Dall'alto al basso, man mano vengono allargate le ipotesi, quando l'esempio non è rispettato.

Lo spazio delle ipotesi è fatto da congiunzioni di attributi.

Find-S darà in output l'ipotesi più specifica che è consistente con gli esempi, anche quelli negativi.

Find-S ha dei limiti, la conoscenza è limitata, ma magari ci sono altri esempi diversi. Quindi è compatibile, ma non per forza ho raggiunto quella segreta.

Potrebbe anche essere che il dataset sia già inconsistente.

Our hypothesis space is unable to represent a simple disjunctive target concept like:  
 $(\text{Sky} = \text{Sunny}) \vee (\text{Sky} = \text{Cloudy})$

Non per forza l'ipotesi è una congiunzione, ma potrebbe anche essere questo or.

■ Let:

$x_1 = \langle \text{Sunny Warm Normal Strong Cool Change} \rangle +$

$x_2 = \langle \text{Cloudy Warm Normal Strong Cool Change} \rangle +$

■ Then:

$S : \{ \langle ? \text{ Warm Normal Strong Cool Change} \rangle \}$

■ Given

$x_3 = \langle \text{Rainy Warm Normal Strong Cool Change} \rangle -$

we will have  $S = \{ \}$

Con  $x_1$  e  $x_2$  troverei quindi l'ipotesi valida  $S$ , ma se introducessi un terzo esempio  $x_3$  che ha un terzo caso, allora l'ipotesi non sarebbe più corretta.

- Idea: Choose  $H$  that expresses every teachable concept, that means  $H$  is the set of all possible subsets of  $X$ 
  - $|X| = 96$
  - $|P(X)| = 2^{96} \sim 10^{28}$  distinct concepts
- $H$  = disjunctions, conjunctions, negations  
e.g. <Sunny Warm Normal ? ? ?>  $\vee$  <? ? ? ? ? Change>
- $H$  surely contains the target concept

Se volessi avere un sistema di apprendimento che non fa presupposizioni sull'ipotesi, allora rimuoviamo ??

- What are  $S$  and  $G$  in this case?
  - Assume positive examples  $(x_1, x_2, x_3)$  and negative examples  $(x_4, x_5)$
  - Then:
$$S : \{(x_1 \vee x_2 \vee x_3)\} \quad \text{and} \quad G : \{\neg (x_4 \vee x_5)\}$$
- The only examples that are classified are the training examples themselves. In other words in order to learn the target concept one would have to present every single instance in  $X$  as a training example

Questa è una forma di overfitting, è coerente agli esempi ma non faccio nessuna predizione su quelli non conosciuti. Quindi non c'è una scorciatoia, facendo bias ogni tanto sbaglierò, se invece voglio che sia perfettamente coerente con gli esempi allora farò overfitting.