



# Machine Learning - 2

---

Corso di Intelligenza Artificiale – II Modulo



## Sommario

---

- Concept Learning: introduzione
- Concept Learning: definizioni
- Un esempio di concept learning
  - Notazioni
  - L'ipotesi di apprendimento induttivo
- Concept Learning come ricerca in  $H$ 
  - Ordinamento general-to-specific nell'insieme  $H$
- Algoritmo Find-S
- Version Space
  - Rappresentazione
  - Rappresentazione compatta di  $H$
  - Algoritmo Candidate-elimination
  - Esempio
- Riflessioni



## Riferimenti

---

- Mitchell, T.M. (1997). *Machine Learning*. McGraw Hill Int. Ed.. Capp. I-II.(via Pincherle)
- Russel, S., Norvig, P. (1995). *Artificial Intelligence A Modern Approach*. Prentice Hall Int. Ed.. Sezz.18.5-18.6.
- Lucidi (disponibili sul sito)



# Introduzione

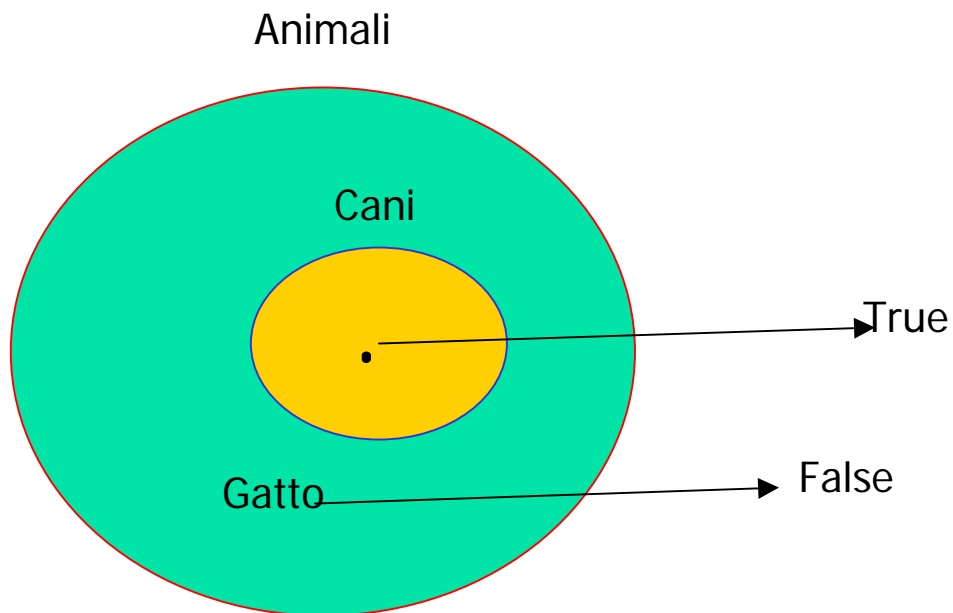
---

- **Learning**: indurre funzioni generali da esempi particolari di training
- **Concept learning**: acquisire la definizione di una categoria generale dato un insieme di esempi positivi e negativi della categoria stessa
- **Best Fit hypothesis**: determinare l'ipotesi  $h$  che meglio si adatta agli esempi di training
- **General-to-Specific**: introduzione nello spazio delle ipotesi  $H$  di un ordinamento parziale
- **Find-s/Version Space**: studio di algoritmi che convergono in  $H$  all'ipotesi  $h$  corretta.



# Definizioni

**Concept Learning:** Inferenziare automaticamente una funzione booleana a partire dall'insieme dei training examples  
<input, output>



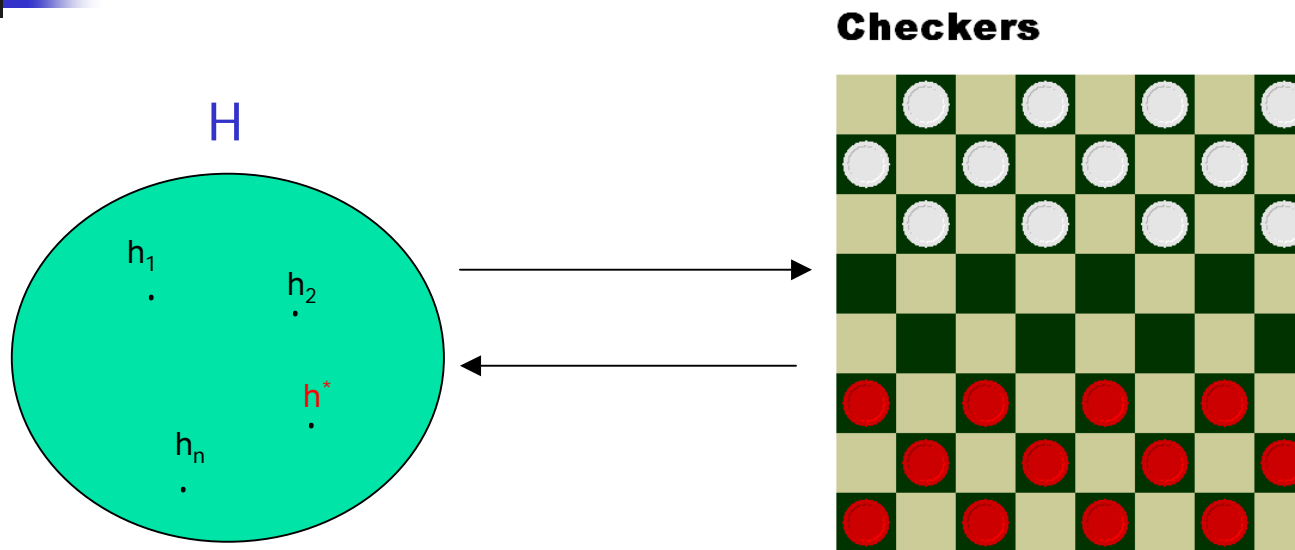
Esempio: una funzione  $f$  definita sull'insieme degli animali

$f: A \rightarrow [\text{true}, \text{false}]$

$f(a) = \text{true}$  se  $a = \text{cane}$

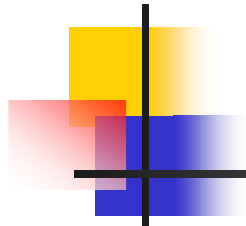
$f(a) = \text{false}$  se  $a = \text{altro animale}$

# Spazio delle Ipotesi H



- Spazio delle ipotesi  $H$
- Determinazione dell'ipotesi  $h$  che più si avvicina alla funzione target  $h^*$

**Apprendimento come ricerca di una ipotesi  $h^*$  in uno spazio delle ipotesi  $H$**



## Esempio (Mitchell, 1997)

- **Target Concept:** "Giorni in cui il mio amico Aldo pratica il suo sport d'acqua preferito"

Istanze x

Esempio	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Training set

- Output
- Input

Apprendere a predire il valore di EnjoySport per un giorno arbitrario, sulla base degli attributi del giorno stesso



# Rappresentazione della funzione ipotesi $h$

- **Ipotesi semplice**

- Ogni ipotesi  $h$  consiste nell'unione dei vincoli espressi negli attributi ammessi nella singola istanza

- **Rappresentazione di  $h$**

- Un vettore di 6 elementi/vincoli che specificano i valori dei 6 attributi *Sky*, *AirTemp*, *Humidity*, *Wind*, *Water* e *Forecast*
- Per ogni attributo, l'ipotesi potrà contenere i seguenti valori:
  - $? \Rightarrow$  ogni valore dell'attributo è accettabile
  - Il valore tra quelli consentiti
  - $0 \Rightarrow$  nessun valore per quell'attributo
- Se una istanza  $x$  soddisfa tutti i vincoli dell'ipotesi  $h$  allora  $h$  classifica  $x$  come esempio positivo:  $h(x)=1$
- Es. di ipotesi  $h$ :  $\langle ?, \text{Cold}, \text{High}, ?, ?, ? \rangle$ . Aldo pratica lo sport solo nei giorni freddi con umidità alta





# Concept Learning Task

---

- **Ipotesi h più generale**
  - $\langle ?, ?, ?, ?, ?, ? \rangle \Rightarrow h$ : ogni giorno è buono per fare sport d'acqua
- **Ipotesi h più specifica**
  - $\langle 0, 0, 0, 0, 0, 0 \rangle \Rightarrow h$ : nessun giorno è buono per fare sport d'acqua

- Insieme di istanze  $x$  sulle quali è definita la funzione target
- La funzione target
- L'insieme delle ipotesi  $h$  considerate dal sistema che apprende
- Insieme di training



# Definizioni

---

- **Insieme delle istanze  $X$ :** l'insieme di items sui quali il concetto è definito. (Insieme di tutti i possibili giorni)
- **Target Concept  $c$ :** la funzione/concept che deve essere appresa. In generale  $c$  può essere qualunque funzione booleana definita su  $X$ :  $c: X \rightarrow [0,1]$
- **Training Examples  $D$ :** formato da  $n$  istanze  $x \in X$  insieme a  $n$  valori corrispondenti  $c(x)$ :  $\langle x, c(x) \rangle$ . Se  $c(x)=1$  si parla di *esempi positivi* mentre per  $c(x)=0$  si parla di *esempi negativi*
- **Insieme  $H$ :** insieme di tutte le possibili ipotesi  $h$  candidate alla determinazione della funzione target
- **Obiettivo:** determinare  $h \in H$  tale che  $h(x)=c(x)$  per tutti gli  $x \in X$



# Concept Learning Task

---

## ■ Dati:

- **Istanze X**: giorni possibili, ciascuno dei quali descritto dagli attributi:
  1. Sky (Sunny, Cloudy e Rainy)
  2. AirTemp (Warm, Cold)
  3. Humidity (Normal, High)
  4. Wind (Strong, Weak)
  5. Water (Warm, Cool)
  6. Forecast (Same, Change)
- **Ipotesi H**: ciascuna ipotesi  $h$  è descritta dall'unione di vincoli sugli attributi  $1, \dots, 6$  +  $\langle ? \rangle$  e  $\langle 0 \rangle$
- **Concetto Target C**: EnjoySport:  $X \rightarrow \{0, 1\}$
- **Esempi di training D**: esempi positivi e negativi della funzione target

## ■ Determinare:

- Una ipotesi  $h \in H$  tale che  $\mathbf{h(x)=c(x) \ \forall \ x \in X}$



## Ipotesi di apprendimento induttivo

---

- **Obiettivo:** determinare una ipotesi  $h$  identica al concetto target  $c$  sull'intero set di istanze  $X$
- **Informazione disponibile:** coppie  $\langle x, c(x) \rangle$

### Ipotesi di apprendimento induttivo

Ogni ipotesi  $h$  che approssima la bene funzione target su un numero sufficientemente grande di esempi di training

Approssimerà bene la funzione target su un insieme di nuovi esempi



## Concept Learning as Search

---

- Il **concept learning** può essere visto come il compito di trovare la funzione  $h$  attraverso una ricerca nello spazio  $H$
- **Obiettivo della ricerca in  $H$** : trovare la funzione  $h$  che meglio approssimi  $c$  sui training examples (**best fit  $h$** )
- **Esempio EnjoySport**: **96** istanze possibili in  $X$ . **5120** possibili ipotesi  $h$  in  $H$  **sintatticamente** distinte  $\Rightarrow$  **973** ipotesi  $h$  **semanticamente** distinte:  **$1 + (4 \times 3 \times 3 \times 3 \times 3 \times 3)$**

**Studio di algoritmi e tecniche di ricerca in  $H$**



## Ordinamento di H: General-to-Specific

---

- **Ragionamento**: si introduce nell'insieme H un ordinamento che faciliti la ricerca della funzione h. Altrimenti dovrei cercare in modo esaustivo (H può essere infinito)

### Esempio di ordinamento di H

- $h_1 = \langle \text{Sunny}, ?, ?, \text{Strong}, ?, ? \rangle$
- $h_2 = \langle \text{Sunny}, ?, ?, ?, ?, ? \rangle$
- Consideriamo l'insieme delle istanze considerate positiva da  $h_1$  e da  $h_2$
- $h_2$  classifica più istanze come positive rispetto ad  $h_1$
- Ogni istanza classificata come positiva da  $h_1$  è classificata come positiva anche da  $h_2 \Rightarrow$   **$h_2$  più generale di  $h_1$**



## Definizione dell'ordinamento G-t-S

- Definizione:  $\forall x \in X, \forall h \in H, x \text{ soddisfa } h \Leftrightarrow h(x)=1$
- Date le ipotesi  $h_j$  e  $h_k$ ,  $h_j$  è more general than or equal to  $h_k$  se e solo se ogni istanza che soddisfa  $h_k$  soddisfa anche  $h_j$

Siano  $h_j$  ed  $h_k$  due funzioni booleane definite su  $X$ .

Allora

$h_j$  è more\_general\_than\_or\_equal\_to  $h_k$  ( $\geq_g$ ) se e solo se

$$\forall x \in X \mid h_k(x)=1 \Rightarrow h_j(x)=1$$



## Ordinamento G-t-S

---

- Introduciamo un ordinamento più stretto sull'insieme H

Siano  $h_j$  ed  $h_k$  due funzioni booleane definite su X.

Allora

$h_j$  è **more\_general\_than**  $h_k$  ( $>_g$ )

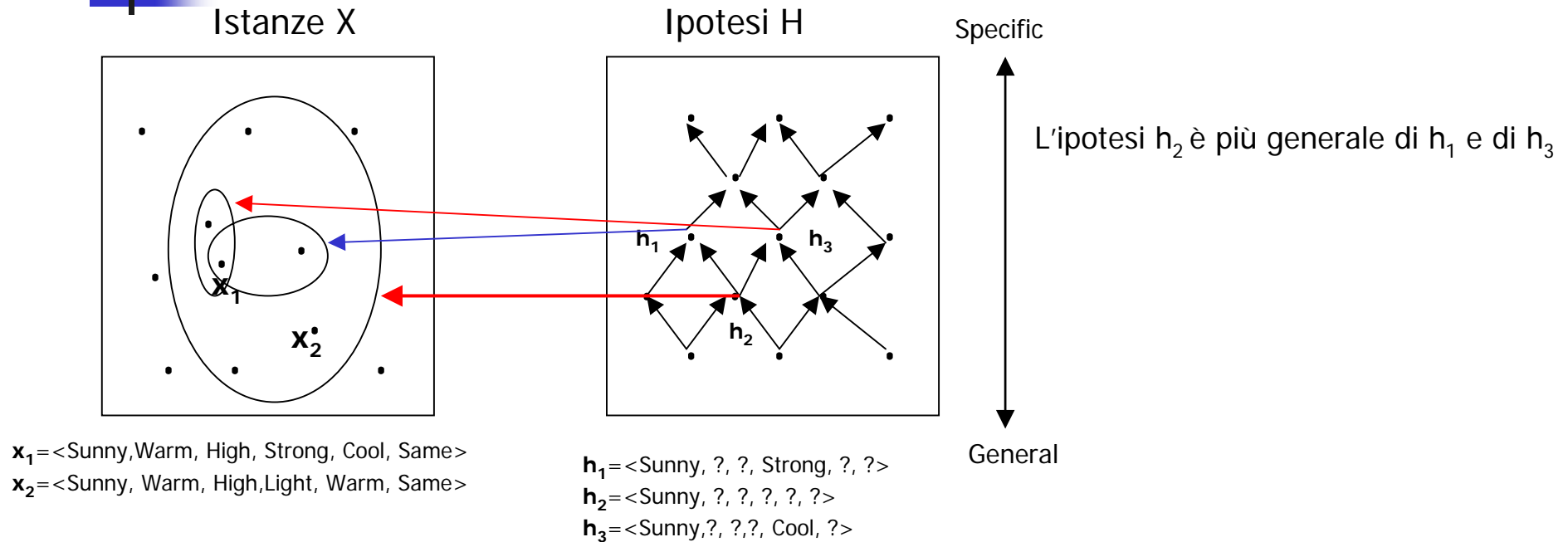
Se e solo se

$$(h_j(x) \geq_g h_k(x)) \wedge (h_k(x) \not\geq_g h_j(x))$$

**$h_k$  è more\_specific\_than  $h_j$**



# Esempio



■ Si è introdotto nello spazio H un **ordinamento parziale**  $\geq_g$  indipendente dal concept che gode delle seguenti proprietà:

- Riflessività
- Antisimmetria
- Transitività



## Riflessioni

---

- La struttura di ordinamento parziale introdotta G-t-S è molto importante perchè:
  - Fornisce una struttura utile su  $H$  per **qualunque** problema di concept learning
  - Si possono utilizzare algoritmi semplici per la ricerca sullo spazio  $H$  di  $h$
  - Non c'è bisogno di effettuare ricerche su tutte le funzioni  $h$  di  $H$



## Algoritmo Find-S: ricerca dell'ipotesi più specifica

- Come utilizzare l'ordinamento `more_general_than` per organizzare la ricerca di una ipotesi  $h$  consistente con gli esempi di training?
  - Iniziamo con l'ipotesi più specifica esistente in  $H$
  - Generalizziamo  $h$  ogniqualvolta non soddisfi un training example  $x$  (ossia  $h(x)=1$ )
- Find -S
  1. Inizializza  $h$  alla ipotesi più specifica in  $H$
  2. Per ogni istanza positiva di training  $x$ 
    - Per ogni attributo con vincolo  $a_i$  in  $h$ 
      - Se il vincolo  $a_i$  non è soddisfatto da  $x$
      - Allora sostituisci  $a_i$  in  $h$  con il vincolo più generale soddisfatto da  $x$
- Output ipotesi  $h$



## Esempio: EnjoySport

- Ipotesi più specifica  $h$  in  $H$ :  $h \leftarrow \langle 0, 0, 0, 0, 0, 0 \rangle$ 
  - Es. 1  $h \leftarrow \langle \text{Sunny}, \text{Warm}, \text{Normal}, \text{Strong}, \text{Warm}, \text{Same} \rangle$
  - ES. 2:  $h \leftarrow \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, \text{Warm}, \text{Same} \rangle$
  - Es. 3: negativo. Passa oltre ( $H$  contiene true concept)
  - Es. 4:  $h \leftarrow \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ? \rangle$
- Risultato:  $h \leftarrow \langle \text{Sunny}, \text{Warm}, ?, \text{Strong}, ?, ? \rangle$

Esempio	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Ipotesi più specifica consistente con il training set



# Riflessioni

---

- $h$  di Find-S è il target concept? Find-S non ci garantisce che non esistano altre ipotesi  $h$  in  $H$  consistenti con i dati. Ci vorrebbe un algoritmo che desse maggiori informazioni sullo spazio  $H$  e  $h$
- Perché predilire l'ipotesi  $h_v$  più specifica?
- Quanto sono consistenti gli esempi di training? Find-S può essere guidato male dal rumore (ignorando esempi negativi). Vorremmo un algoritmo che tenesse conto del rumore nei dati
- Se esistono più ipotesi  $h$  consistenti con i dati?

=>

Version Space ed altri algoritmi



## Definizioni

---

- Una ipotesi  $h$  è **consistente** con un insieme di training examples  $D$  se e solo se  $h(x)=c(x)$  per ogni esempio  $\langle x, c(x) \rangle$  in  $D$ 
  - **Consistent( $h, D$ )**  $\Leftrightarrow (\forall \langle x, c(x) \rangle \in D \Rightarrow h(x)=c(x))$
- La consistenza dipende dal target concept
- **Version Space:**  $VS_{H,D}$  rispetto allo spazio delle ipotesi  $H$  ed all'insieme  $D$  degli esempi di training è il sottoinsieme di  $H$  formato da tutte le ipotesi  $h$  consistenti in  $D$

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



## Algoritmo List-Then-Eliminate

---

1. Version Space  $\leftarrow$  una lista di tutte le ipotesi in  $H$
  2. Per ogni training example
    - Rimuovi dal version space ogni ipotesi  $h$  tale che  $h(x) \neq c(x)$  (inconsistenza)
  3. Output la lista delle ipotesi rimanenti in  $H$
- Commenti:
- L'algoritmo può fornire un set di ipotesi  $h$
  - Richiede la enumerazione esaustiva di tutte le ipotesi di  $H$
  - Semplicità



## Rappresentazione compatta del version space

- Version Space: lo rappresentiamo dai suoi membri più generali e meno generali
- Essi formano due insiemi di confine che delimitano il version space dentro lo spazio parzialmente ordinato delle ipotesi

