

Lezione 3

Introduzione alla Modellazione Predittiva Dal ML al DM: Concept Learning

Mercoledì, 18 gennaio 2007

Giuseppe Manco

Riferimenti:

Chapter 2, Mitchell

Chapter 10 Hand, Mannila, Smith

Chapter 7 Han, Kamber

Lecture Outline

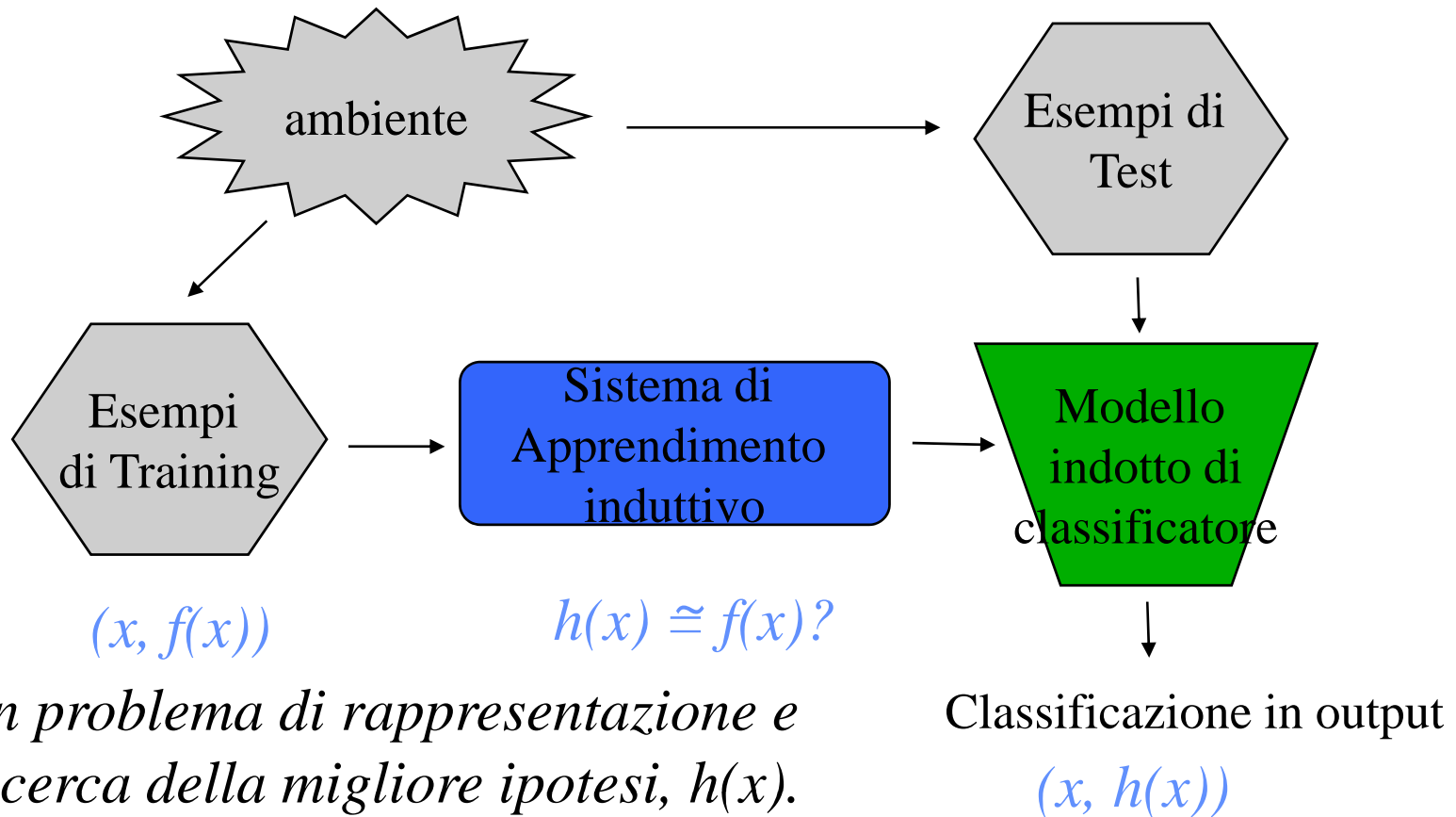
- **Apprendimento dagli esempi**
 - **Nozioni (teoriche) di base**
 - **Approccio naïve: non assume rumore sui dati e illustra i concetti fondamentali**
- **Ordinamento General-to-Specific**
 - **Version space: ordinamento parziale dell'insieme delle ipotesi**
 - **L'algoritmo Candidate elimination**
 - **Apprendimento induttivo**
- **L'influenza induttiva**

(Supervised) Concept Learning

- **Dati:** delle istanze di Allenamento (training) $\langle x, f(x) \rangle$ di una funzione f sconosciuta
- **Trovare:** una buona approssimazione di f
- **Esempi**
 - **Diagnosi medica**
 - x = caratteristiche del paziente (sua storia medica, sintomi, esami di laboratorio)
 - f = malattia (o terapia consigliata)
 - **Risk assessment**
 - x = caratteristiche del consumatore/assicurato (dati demografici, incidenti passati)
 - f = livello di rischio
 - **Guida automatica**
 - x = immagine bitmap della superficie della strada di fronte al veicolo
 - f = gradi di sterzata del volante
 - **Part-of-speech tagging**
 - **Fraud/intrusion detection**
 - **Web log analysis**

Sistemi di classificazione e apprendimento induttivo

Struttura di base



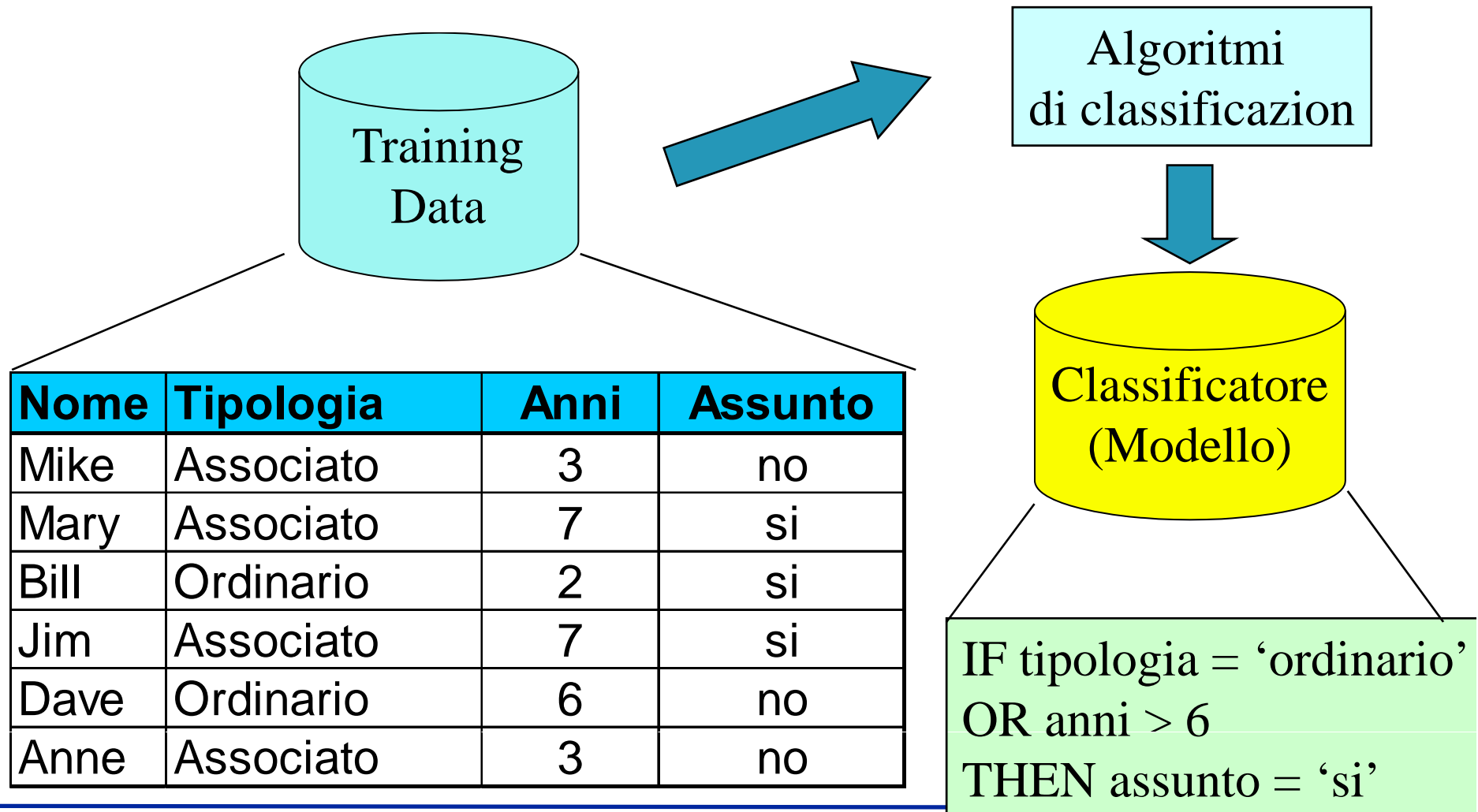
Un processo a due fasi

- **Costruzione del modello: descrizione di un insieme di classi predefinite**
 - Ogni esempio/istanza appartiene ad una classe predefinita, come descritto dall'**attributo di classe**
 - L'insieme di tuple utilizzate per la costruzione del modello: **training set**
 - Il modello è rappresentato utilizzando un linguaggio specifico
- **Utilizzo del modello: per la classificazione di nuovi oggetti (sconosciuti)**
 - **Stima dell'accuratezza del modello**
 - Su un insieme di Test (le cui etichette di classe sono note)
 - Si confronta l'etichetta del test con il risultato della classificazione
 - Il tasso di accuratezza è la percentuale delle istanze del Test set classificata correttamente dal modello
 - Il test set è indipendente dal training set

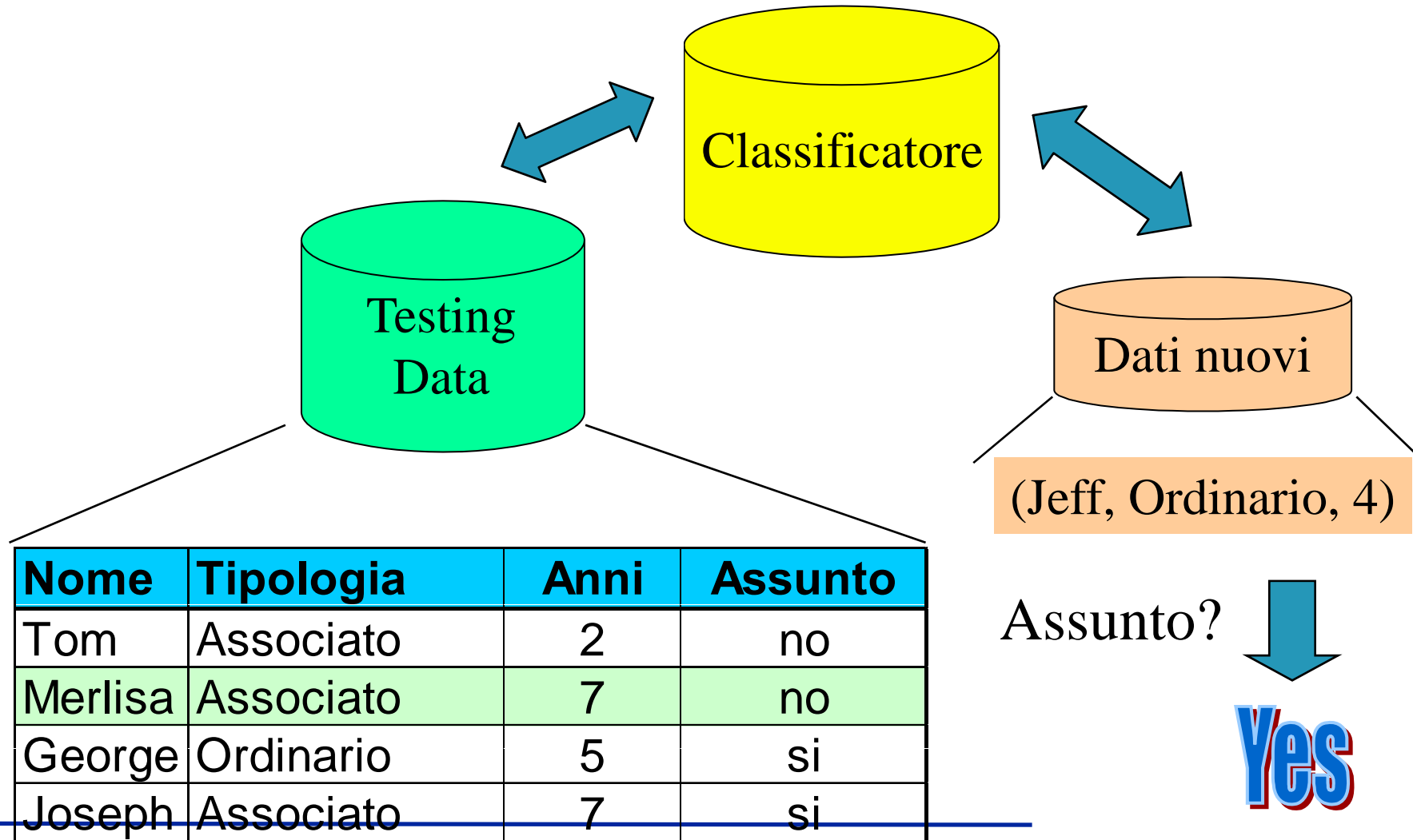
Train & Test

- **Il dataset è partizionato in training set e test set**
- **La classificazione è effettuata in due stadi:**
 - **Training – costruzione del modello dal training set**
 - **Test – controllo dell'accuratezza del modello utilizzando il test set**

Il processo di classificazione: (1): Costruzione del modello



Il processo di Classificazione (2): Utilizzo del modello in Predizione



Supervised vs. Unsupervised Learning

- **Supervised learning (classificazione)**
 - **Supervisione: il training set contiene l'etichetta che indica la classe da apprendere**
 - **I nuovi dati sono classificati sulla base di quello che si apprende dal training set**
- **Unsupervised learning (clustering)**
 - **L'etichetta di classe è sconosciuta**
 - **Le istanze sono fornite con l'obiettivo di stabilire se vi sono raggruppamenti (classi) tra i dati**

Classificazione, regressione, Predizione

- **Classificazione:**
 - Predice attributi categorici
- **Regressione**
 - Predice funzioni numerici
- **Predizione:**
 - Predice valori mancanti

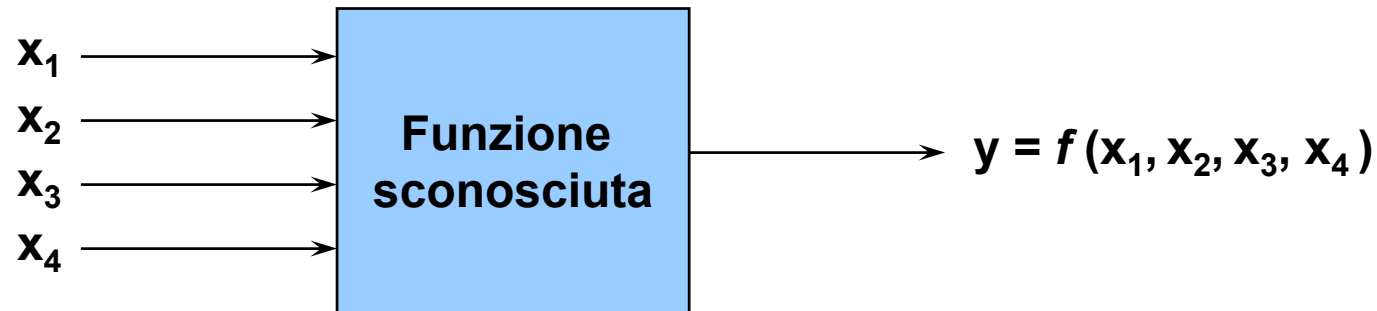
Problematiche correlate (1): Data Preparation

- **Data cleaning**
 - I dati vanno preprocessati per ridurre rumore e trattare i valori mancanti
- **Relevance analysis (feature selection)**
 - Vengono rimossi gli attributi ridondanti o irrilevanti
- **Data transformation**
 - I dati vengono generalizzati e/o trasformati

Problematiche correlate (2): metodi di valutazione

- **Accuratezza predittiva**
- **Velocità**
 - Tempo necessario per costruire il modello
 - Tempo necessario per applicare il modello
- **Robustezza**
 - Capacità di gestire dati sporchi
- **Scalabilità**
 - Efficienza con dati residenti su disco
- **Interpretabilità**
 - Comprensibilità del modello appreso
- **Bontà del modello**
 - Dimensione e compattezza del modello

Un problema di apprendimento



Esempio	x_1	x_2	x_3	x_4	y
0	0	1	1	0	0
1	0	0	0	0	0
2	0	0	1	1	1
3	1	0	0	1	1
4	0	1	1	0	0
5	1	1	0	0	0
6	0	1	0	1	0

- $x_i: t_i, y: t, f: (t_1 \times t_2 \times t_3 \times t_4) \rightarrow t$
- La nostra funzione di learning: $(t_1 \times t_2 \times t_3 \times t_4 \times t) \rightarrow (t_1 \times t_2 \times t_3 \times t_4) \rightarrow t$

Lo spazio delle ipotesi (senza restrizioni)

- $|A \rightarrow B| = |B|^{|A|}$
- $|H^4 \rightarrow H| = |\{0,1\} \times \{0,1\} \times \{0,1\} \times \{0,1\} \rightarrow \{0,1\}| = 2^{2^4} = 65536$ funzioni possibili
- **Ignoranza completa: è possibile apprendere?**
 - Si deve considerare ogni possibile combinazione input/output
 - Dopo 7 esempi, ci sono ancora $2^9 = 512$ possibilità per f

Example	x_1	x_2	x_3	x_4	y
0	0	0	0	0	?
1	0	0	0	1	?
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	?
8	1	0	0	0	?
9	1	0	0	1	1
10	1	0	1	0	?
11	1	0	1	1	?
12	1	1	0	0	0
13	1	1	0	1	?
14	1	1	1	0	?
15	1	1	1	1	?

Il Training set per *EnjoySport*

- **Specifica per esempi**
 - Simile alla definizione di tipo di dato
 - 6 attributi: Sky, Temp, Humidity, Wind, Water, Forecast
 - Nominal-valued (symbolic) attributes - enumerative data type
- **Concetto binario**
- **Supervised Learning Problem: descrivere il concetto generale**

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
0	Sunny	Warm	Normal	Strong	Warm	Same	Yes
1	Sunny	Warm	High	Strong	Warm	Same	Yes
2	Rainy	Cold	High	Strong	Warm	Change	No
3	Sunny	Warm	High	Strong	Cool	Change	Yes

La rappresentazione delle ipotesi

- Molte alternative
- Ipotesi h : congiunzione di vincoli sugli attributi
- Tipi di vincoli
 - Valori specifici (esempio: *Water = Warm*)
 - Qualunque valore (“*Water = ?*”)
 - Nessun valore (“*Water = ∅*”)
- Esempio di ipotesi
 - | Sky | AirTemp | Humidity | Wind | Water | Forecast |
|----------------|----------|----------|---------------|----------|---------------|
| < <i>Sunny</i> | <i>?</i> | <i>?</i> | <i>Strong</i> | <i>?</i> | <i>Same</i> > |
 - È consistente con gli esempi di training?
 - Esempi di ipotesi consistenti con gli esempi

Concept Learning

- **Dati**

- Un insieme X di istanze
- Esempio: I possibili giorni, descritti dagli attributi *Sky, AirTemp, Humidity, Wind, Water, Forecast*
- La funzione target $c \equiv \text{EnjoySport}: X \rightarrow \mathbf{H} \equiv \{\{\text{Rainy, Sunny}\} \times \{\text{Warm, Cold}\} \times \{\text{Normal, High}\} \times \{\text{None, Mild, Strong}\} \times \{\text{Cool, Warm}\} \times \{\text{Same, Change}\}\} \rightarrow \{0, 1\}$
- Il linguaggio delle ipotesi H : congiunzione di letterali (ad esempio, $\langle ?, \text{Cold, High, }, ?, ?, ? \rangle$)
- Un insieme di allenamento D : esempi positivi e negativi della funzione target

$$\langle \mathbf{x}_1, \mathbf{c}(\mathbf{x}_1) \rangle, \dots, \langle \mathbf{x}_m, \mathbf{c}(\mathbf{x}_m) \rangle$$

- **Determinare**

- L'ipotesi $h \in H$ tale che $h(x) = c(x)$ per ogni $x \in D$
- Che sia consistente con il training set

- **Nota**

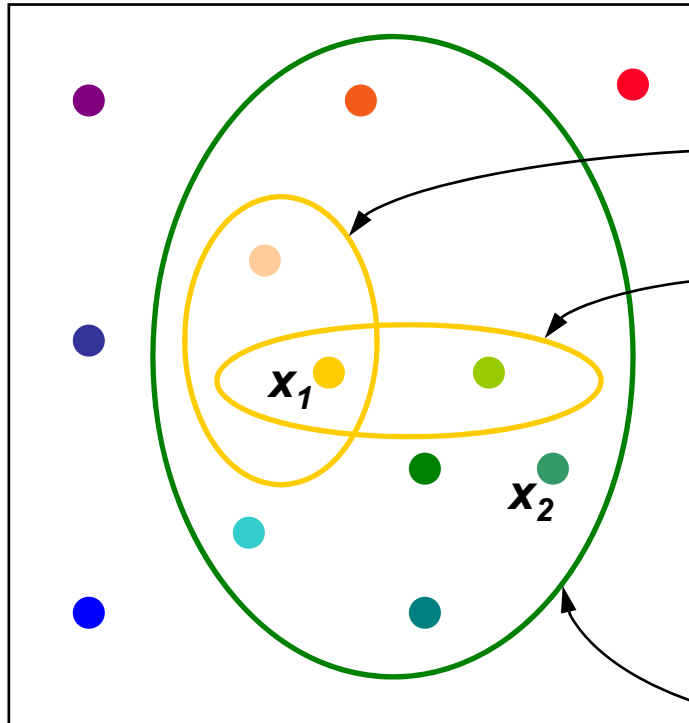
- Assunzione: nessun valore mancante
- Nessun rumore su c (etichette non contraddittorie)

L'ipotesi di apprendimento induttivo

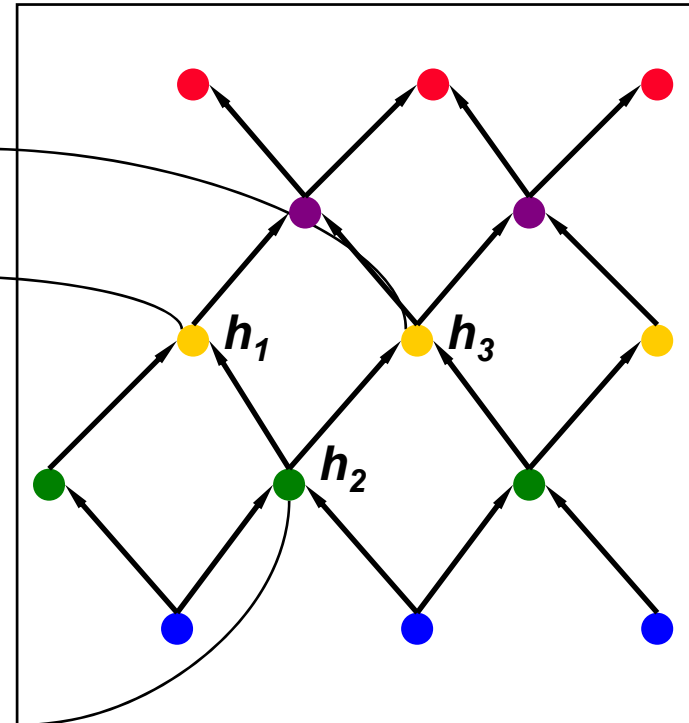
- **Assunzione fondamentale**
- **Informalmente**
 - Ogni ipotesi che approssima bene la funzione target su un insieme sufficientemente grande di dati di training, approssima la stessa anche sugli esempi non osservati
 - Definizioni tralasciate (per il momento) : *sufficientemente grande, approssima bene, non osservati*
- **Formalmente, giustificazioni**
 - statistica
 - probabilistica
 - computazionale
- **Come trovare tale ipotesi?**

Istanze, ipotesi e ordinamento parziale

istanze X



Ipotesi H



Specifico

Generale

$x_1 = \langle \text{Sunny, Warm, High, Strong, Cool, Same} \rangle$
 $x_2 = \langle \text{Sunny, Warm, High, Light, Warm, Same} \rangle$

$h_1 = \langle \text{Sunny, ?, ?, Strong, ?, ?} \rangle$
 $h_2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle$
 $h_3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle$

$\leq_p \equiv \text{meno-specifico-di} \equiv \text{più-generale-di}$

$h_2 \leq_p h_1$
 $h_2 \leq_p h_3$

Definizione

- Siano h_i e h_j due ipotesi nello spazio H
- h_j è più generale di h_i ($h_i \leq h_j$) se e solo se
 - Per ogni esempio x $h_i(x) = h_j(x)$
- La relazione \leq impone un ordinamento parziale su (H) (riflessivo, antisimmetrico e transitivo)

L'algoritmo Find-S

1. Inizializza h all'ipotesi più specifica in H

H : lo spazio delle ipotesi (parzialmente ordinato, con la relazione *Meno-specifico-di*)

2. Per ogni esempio positivo x

Per ogni vincolo di attributo a_i in h

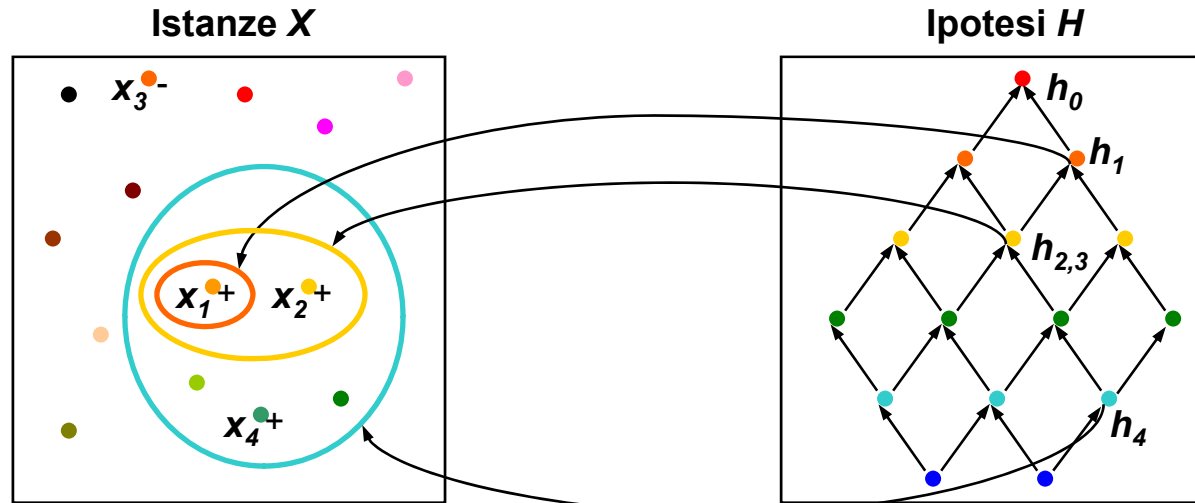
SE il vincolo a_i in h è soddisfatto da x

ALLORA non fare niente

ALTRIMENTI rimpiazza a_i in h con il più vicino vincolo più generale soddisfatto da x

3. Restituisci l'ipotesi h

Esempio



$x_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle, +$
 $x_2 = \langle \text{Sunny, Warm, High, Strong, Warm, Same} \rangle, +$
 $x_3 = \langle \text{Rainy, Cold, High, Strong, Warm, Change} \rangle, -$
 $x_4 = \langle \text{Sunny, Warm, High, Strong, Cool, Change} \rangle, +$

$h_0 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$
 $h_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$
 $h_2 = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$
 $h_3 = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$
 $h_4 = \langle \text{Sunny, Warm, ?, Strong, ?, ?} \rangle$

- **Quali sono i problemi di *Find-S*?**
 - Non si può dire se il concetto sia stato appreso
 - Non si può verificare se l'insieme di training è inconsistente
 - Viene trovata sempre l'ipotesi h più specifica (perché?)
 - Possono esserci più ipotesi che soddisfano l'insieme

Version Spaces

- **Definizione: ipotesi consistenti**
 - Un'ipotesi è consistente con un training set D e un concetto target c se e solo se $h(x) = c(x)$ per ogni esempio $\langle x, c(x) \rangle$ in D .
 - $\text{Consistent}(h, D) \equiv \forall \langle x, c(x) \rangle \in D . h(x) = c(x)$
- **Definizione: Version Space**
 - Il version space $VS_{H,D}$, relativo allo spazio H e al training set D , è il sottoinsieme delle ipotesi di H consistenti con D .
 - $VS_{H,D} \equiv \{ h \in H \mid \text{Consistent}(h, D) \}$

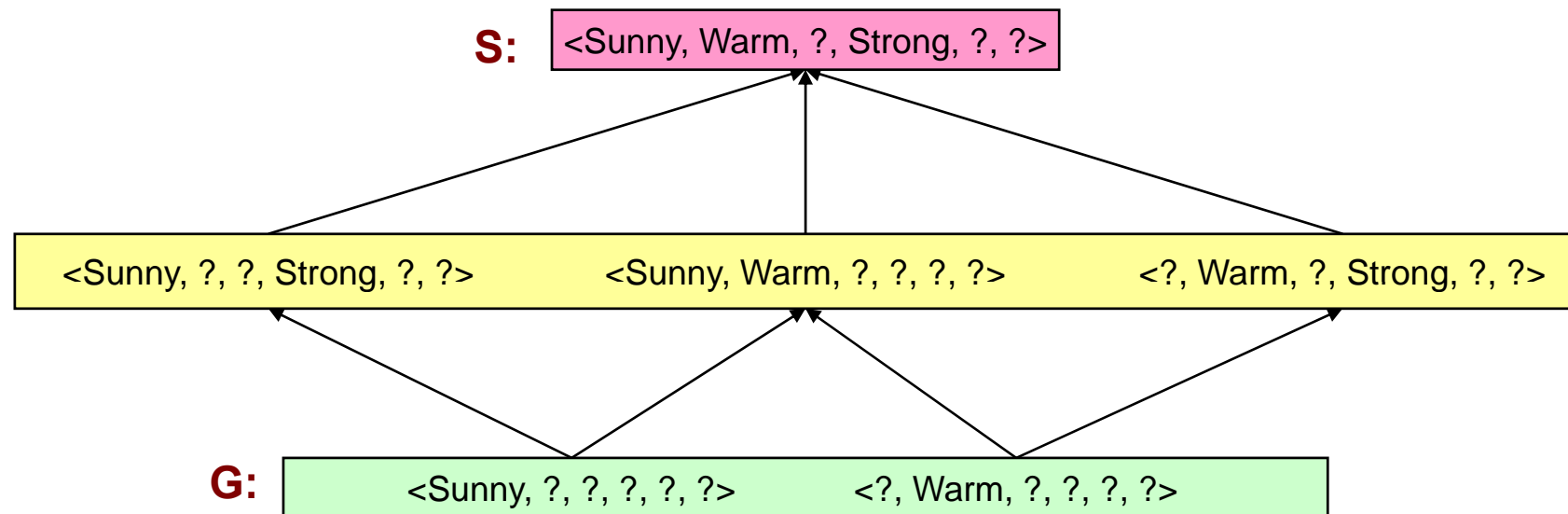
L'algoritmo *List-Then-Eliminate*

1. Inizializzazione: *VersionSpace* \leftarrow una lista contenente tutte le ipotesi di *H*


2. Per ogni esempio $\langle x, c(x) \rangle$

Rimuovi dal *VersionSpace* tutte le ipotesi per le quali $h(x) \neq c(x)$

3. Restituisci la lista delle ipotesi di *VersionSpace*



Esempio di Version Space



Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
0	Sunny	Warm	Normal	Strong	Warm	Same	Yes
1	Sunny	Warm	High	Strong	Warm	Same	Yes
2	Rainy	Cold	High	Strong	Warm	Change	No
3	Sunny	Warm	High	Strong	Cool	Change	Yes

Rappresentare Version Spaces

- **Lo spazio delle ipotesi**
 - Un semireticolo finito con minimo (ordinamento parziale: Meno-Specifico-di; $\perp \equiv$ tutti ?)
 - Ogni coppia di ipotesi ha un *greatest lower bound* (GLB)
 - $VS_{H,D} \equiv$ L'insieme dei consistenti (un sottoinsieme di H)
- **Definizione: Bordo generale**
 - Il bordo generale G di un version space $VS_{H,D}$ è l'insieme dei suoi membri più generali
 - I più generali \equiv *gli elementi minimali di* $VS_{H,D} \equiv$ “l'insieme delle condizioni necessarie”
- **Definizione: bordo specifico**
 - Bordo specifico S di un version space $VS_{H,D}$: l'insieme dei suoi membri più specifici
 - Più specifici \equiv elementi massimali *di* $VS_{H,D} \equiv$ “insieme delle condizioni sufficienti”
- **Version Space**
 - Ogni elemento di un version space si trova tra S and G
 - $VS_{H,D} \equiv \{ h \in H \mid \exists s \in S . \exists g \in G . g \leq_p h \leq_p s \}$ dove $\leq_p \equiv$ Meno-Specifico-Di

L'algoritmo Candidate Elimination [1]

1. Inizializzazione

$G \leftarrow$ ipotesi più generale in H , $\{<?, \dots, ?>\}$

$S \leftarrow$ ipotesi più specifica in H , $\{<\emptyset, \dots, \emptyset>\}$

2. Per ogni istanza d

Se d è positivo (*Update-S*)

Rimuovi da G tutte le ipotesi inconsistenti con d

Per ogni ipotesi s in S inconsistente con d

Rimuovi s da S

Aggiungi a S tutte le generalizzazioni minimali h di s tali che

1. h sia consistente con d
2. esista un elemento di G più generale di h

Rimuovi da S tutte le ipotesi più generali di una qualsiasi altra ipotesi in S

L'algoritmo Candidate Elimination [2]

(continua)

Se d è negativo (*Update-G*)

Rimuovi da S ogni ipotesi inconsistente con d

Per ogni ipotesi g in G inconsistente con d

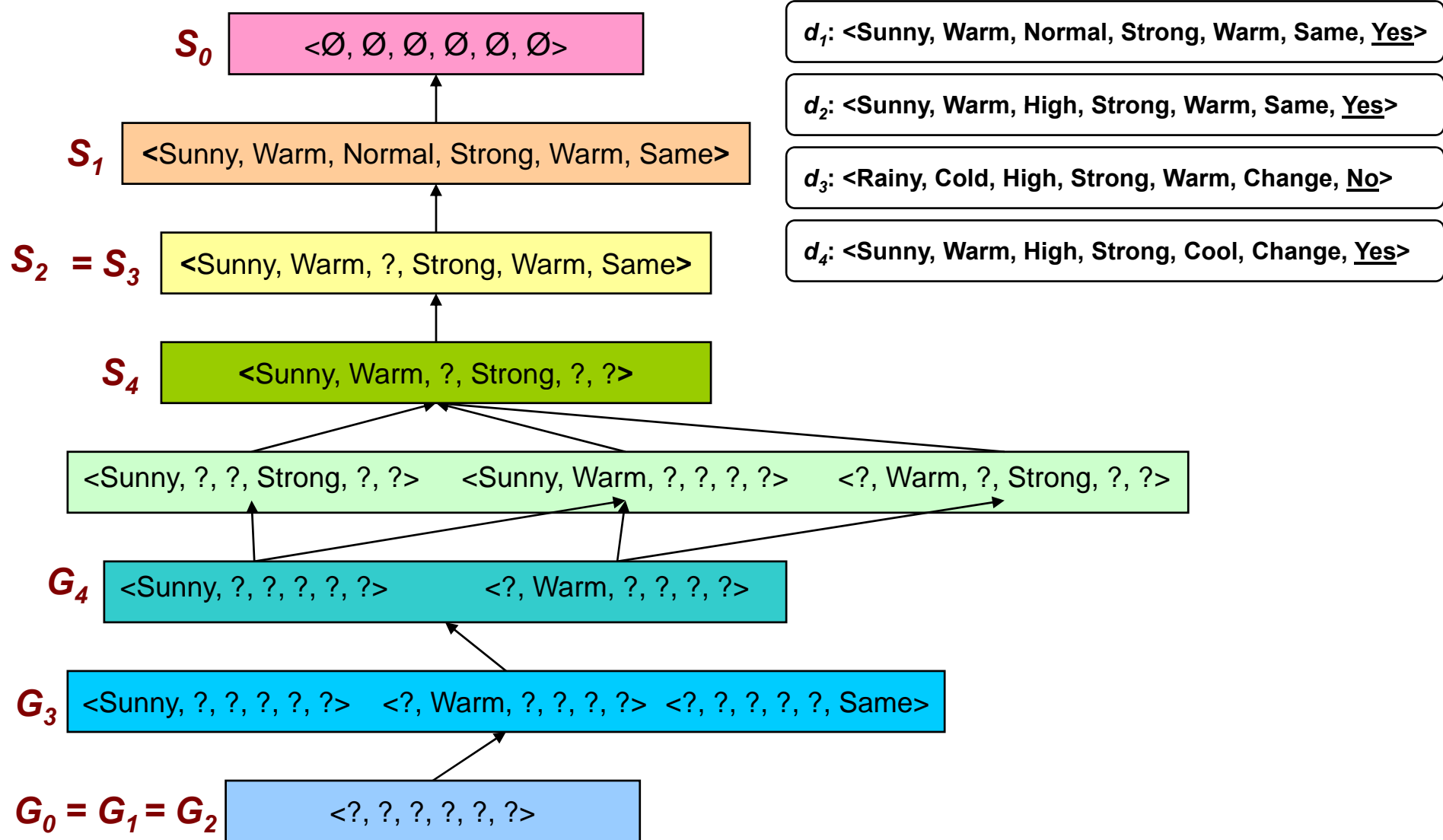
Rimuovi g da G

Aggiungi a G tutte le specializzazioni minimali h di g tali che

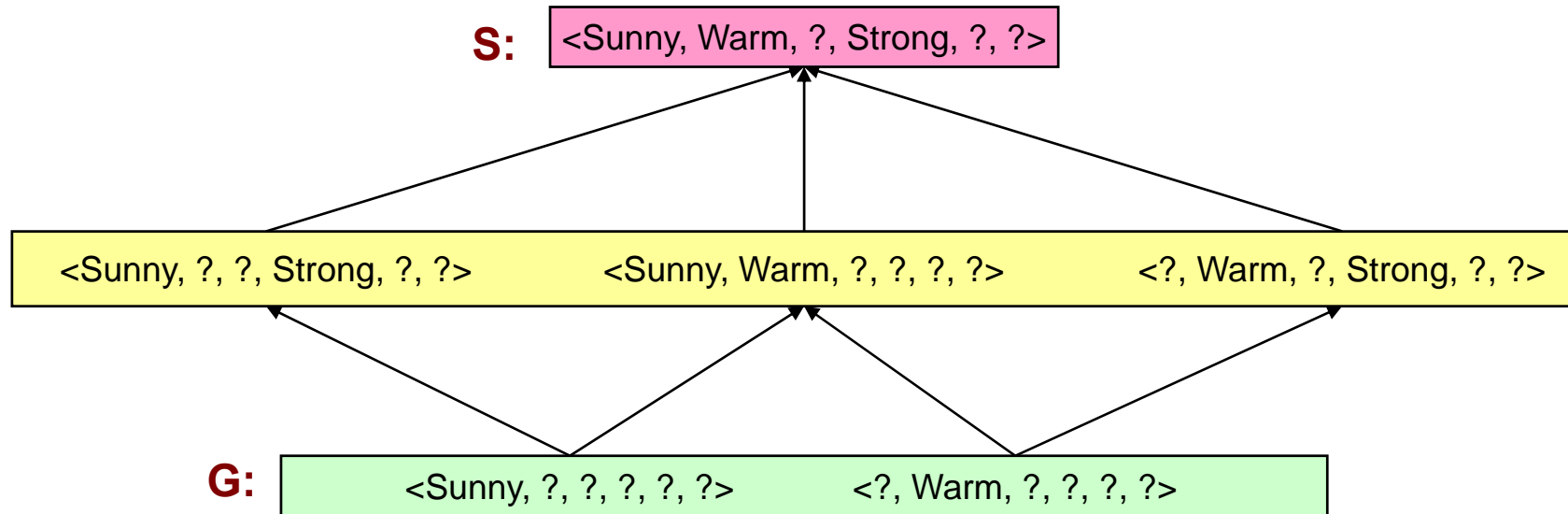
1. h sia consistente con d
2. esista un elemento di S più specifico di h

Rimuovi da G ogni ipotesi meno generale di una qualsiasi altra ipotesi in G

Esempio



Come proseguirebbe la costruzione?



- **Cosa chiederebbe l'algoritmo?**
- **Come devono essere classificati questi esempi?**
 - <Sunny, Warm, Normal, Strong, Cool, Change>
 - <Rainy, Cold, Normal, Light, Warm, Same>
 - <Sunny, Warm, Normal, Light, Warm, Same>
 - <Sunny, Cold, Normal, Strong, Warm, Same>

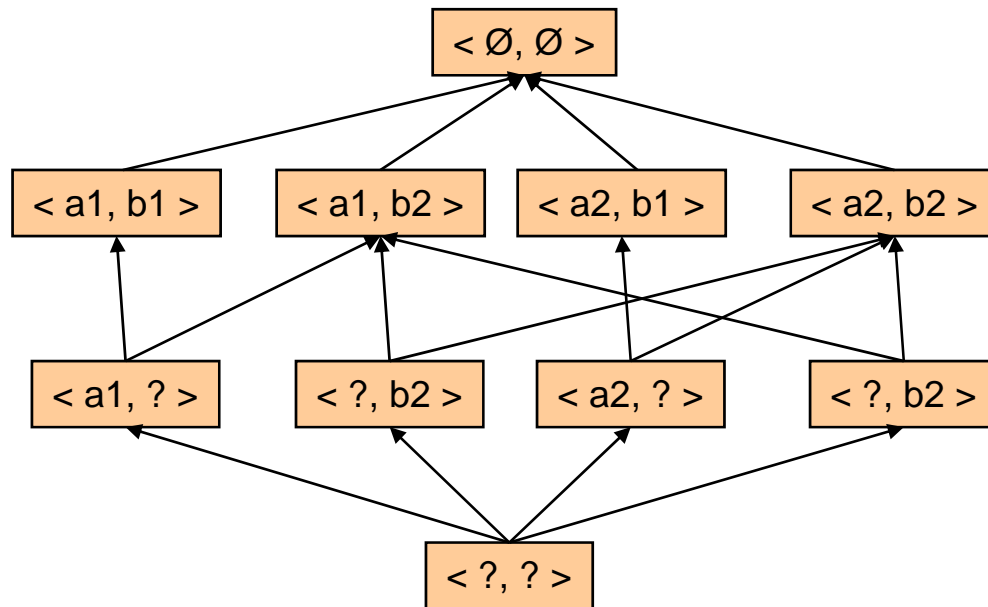
Sommario: Terminologia adottata

- **Supervised Learning**
 - Concetto – funzione che associa una categoria ad una istanza (finora, solo categorie booleane: +/-)
 - (funzione) Target – la funzione vera f
 - Ipotesi- la funzione h che si assume simile a f
 - Spazio delle ipotesi- Insieme di tutte le ipotesi che possono essere generate dall'algoritmo di apprendimento
 - Istanza – tupla della forma $\langle x, f(x) \rangle$
 - Spazio delle istanza – spazio di tutte le possibili istanze
- **Version Space**
 - Algoritmi: *Find-S*, *List-Then-Eliminate*, candidate elimination
 - Ipotesi consistente- predice correttamente l'istanza osservata
 - Version space – spazio di tutte le ipotesi consistenti
- **Apprendimento induttivo**
 - generalizzazione – processo di generazione delle ipotesi capaci di descrivere i casi non ancora osservati
 - L'ipotesi di apprendimento induttivo

Qual'è la giustificazione per l'induzione?

- **Esempio: generalizzazione induttiva**
 - Esempio positivo: <Sunny, Warm, Normal, Strong, Cool, Change, Yes>
 - Esempio positivo: <Sunny, Warm, Normal, Light, Warm, Same, Yes>
 - Concetto indotto: <Sunny, Warm, Normal, ?, ?, ?>
- **Chi ci dice che possiamo classificare i casi non visti?**
 - Ad esempio., <Sunny, Warm, Normal, Strong, Warm, Same>
 - C'è abbastanza informazione (in questo nuovo esempio) per fare una predizione?

Un altro esempio...



d_1 : <a1,b1, Yes>

d_2 : <a1,b2,, Yes>

d_3 : <a2,b1, No>

d_4 : <a2,b2, Yes>

•Qual è il risultato?

Un classificatore Unbiased

- **Lo spazio delle ipotesi di Candidate-Elimination H è biased**
 - *Concetti congiuntivi con il “?”*
 - *Quali concetti non possono essere espressi in H ? (Suggerimento: Quali sono le sue limitazioni sintattiche?)*
- **Idea**
 - *Scegliamo H' affinché possa esprimere ogni ipotesi possibile*
 - *ovvero, H' rappresenta ogni combinazione in X*
 - *$|A \rightarrow B| = |B|^{|A|}$ ($A = X$; $B = \{\text{labels}\}$; $H' = A \rightarrow B$)*
 - *$\{\{\text{Rainy, Sunny}\} \times \{\text{Warm, Cold}\} \times \{\text{Normal, High}\} \times \{\text{None, Mild, Strong}\} \times \{\text{Cool, Warm}\} \times \{\text{Same, Change}\}\} \rightarrow \{0, 1\}$*
- **Un linguaggio completo**
 - *H' = disgiunzioni (\vee), congiunzioni (\wedge), negazioni (\neg) sui vincoli di H*
 - *$|H'| = 2^{(2 \cdot 2 \cdot 2 \cdot 3 \cdot 2 \cdot 2)} = 2^{96}$; $|H| = 1 + (3 \cdot 3 \cdot 3 \cdot 4 \cdot 3 \cdot 3) = 973$*
- **Cosa sono S , G per H' ?**
 - *$S \leftarrow$ disgiunzione di tutti gli esempi positivi*
 - *$G \leftarrow$ congiunzione di tutti i negati degli esempi negativi*

Cosa apprende un classificatore Unbiased?

- **Il training set soltanto!**
 - Sia x un'istanza non appartenente a D
 - Siano S e G i bordi calcolati
 - Per ogni h t.c. $S \leq h \leq G$,
 - $h' = h \wedge \neg x$ e $h'' = h \wedge x$ sono ipotesi consistenti con D
 - Quindi, vale $S \leq h' \leq G$, $S \leq h'' \leq G$
 - **Conseguenza: non sappiamo come classificare x !**

Bias Induttivo

- **Componenti della definizione**

- Algoritmo di apprendimento L
- Istanze X , target c
- Training set $D_c = \{ \langle x, c(x) \rangle \}$
- $L(x_j, D_c)$ = classificazione assegnata all'istanza x_j da L dopo la fase di training su D_c

- **Definition**

- Il bias induttivo di L è l'insieme minimo di asserzioni B che garantisce

$$\forall x_j \in X . [(B \wedge D_c \wedge x_j) \vdash L(x_j, D_c)]$$

- informalmente: preferenza per certe ipotesi tramite vincoli strutturali (sintattici)

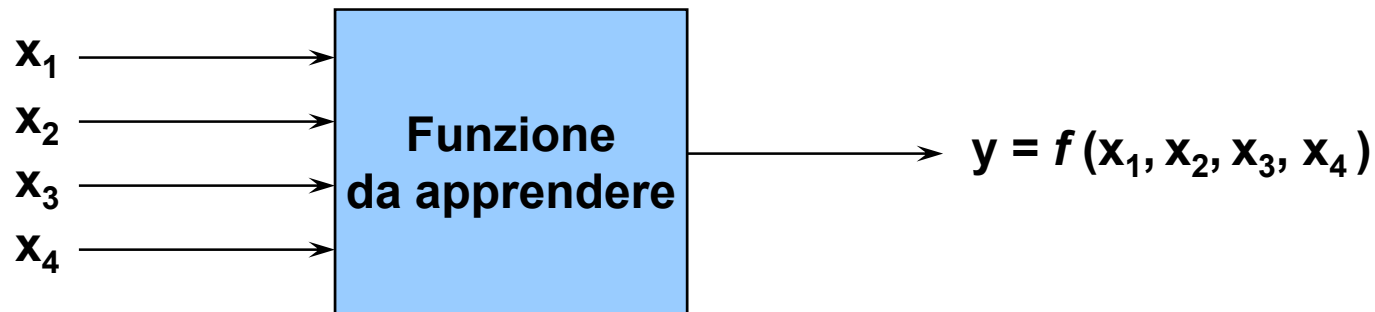
- **Razionale**

- Assunzione a priori sul concetto target

Tre algoritmi con different Bias

- **Rote Learner**
 - Bias: Tutto quello che vedo prima
 - Memorizzo gli esempi
 - Classifico x se e solo se corrisponde ad un esempio già osservato
- **Candidate Elimination**
 - Bias più forte: I concetti sono congiunzioni
 - Memorizzo le generalizzazioni e specializzazioni estreme
 - Classifico x se e solo se si trova tra S e G
- **Find-S**
 - Bias ancora più forte: le ipotesi più specifiche
 - Assunzione: ogni istanza non positiva è negativa
 - Classifico x sulla base dell'insieme S

Lo spazio delle ipotesi con restrizioni sintattiche



Istanza	x_1	x_2	x_3	x_4	y
0	0	1	1	0	0
1	0	0	0	0	0
2	0	0	1	1	1
3	1	0	0	1	1
4	0	1	1	0	0
5	1	1	0	0	0
6	0	1	0	1	0

• **Bias: Regole congiuntive semplici**

- Solo 16 regole congiuntive della forma $y = x_i \wedge x_j \wedge x_k$
- $y = \emptyset, x_1, \dots, x_4, x_1 \wedge x_2, \dots, x_3 \wedge x_4, x_1 \wedge x_2 \wedge x_3, \dots, x_2 \wedge x_3 \wedge x_4, x_1 \wedge x_2 \wedge x_3 \wedge x_4$
- Nell'esempio, nessuna regola può esprimere l'intero insieme (perché?)
- Clausole semplici (congiunzioni e disgiunzioni) si comportano analogamente

Lo spazio delle ipotesi: regole m-di-n

- Regole *m-di-n*
 - 32 possibili regole, del tipo:
“ $y = 1$ se e solo se almeno m of delle n variabili valgono 1”

Istanza	x_1	x_2	x_3	x_4	y
0	0	1	1	0	0
1	0	0	0	0	0
2	0	0	1	1	1
3	1	0	0	1	1
4	0	1	1	0	0
5	1	1	0	0	0
6	0	1	0	1	0

variabili	controesempio				Variabili	Controesempio			
	1-di	2-di	3-di	4-di		1-di	2-di	3-di	4-di
$\{x_1\}$	2	–	–	–	$\{x_2, x_4\}$	0	2	–	–
$\{x_2\}$	0	–	–	–	$\{x_3, x_4\}$	0	3	–	–
$\{x_3\}$	0	–	–	–	$\{x_1, x_2, x_3\}$	0	2	2	–
$\{x_4\}$	6	–	–	–	$\{x_1, x_2, x_4\}$	0	2	2	–
$\{x_1, x_2\}$	0	2	–	–	$\{x_1, x_3, x_4\}$	0	•	2	–
$\{x_1, x_3\}$	0	2	–	–	$\{x_2, x_3, x_4\}$	0	4	2	–
$\{x_1, x_4\}$	5	2	–	–	$\{x_1, x_2, x_3, x_4\}$	0	4	2	2
$\{x_2, x_3\}$	0	2	–	–					

La consistenza rivisitata

- **Definizione: consistenza di ipotesi**
 - La consistenza di un'ipotesi h con un training set D e un concetto target c è il numero di esempi $\langle x, c(x) \rangle$ in D tali che $h(x) = c(x)$.
 - $Consistent(h, D) \equiv |\{ \langle x, c(x) \rangle \in D . h(x) = c(x) \}|$
- **(Ri)Definizione: Version Space**
 - Il version space $VS_{H,D}$, relativo allo spazio H e al training set D , è il sottoinsieme delle ipotesi di H con consistenza massimale.
 - $VS_{H,D} \equiv \{ h \in H \mid \forall h' \in H Consistent(h, D) \geq Consistent(h', D) \}$

Due strategie per l'apprendimento

- **Esprimere conoscenza di dominio (Prior Knowledge)**
 - Il ruolo della conoscenza di dominio: guidare la ricerca di ipotesi
- **Sviluppare spazi di ipotesi flessibili**
 - Collezioni strutturate di ipotesi
 - Agglomerazioni: collezioni annidate (gerarchie)
 - partizionamenti: alberi di decisione
 - Reti neurali; cases, etc.
 - Spazi di dimensione adattativa
- **In entrambi i casi, gli algoritmi trovano ipotesi che “fittano” i dati**
 - Che generalizzano bene