INTELLIGENZA ARTIFICIALE

ANNO ACCADEMICO
2008-2009

Machine Learning:Version Space

Sommario

- Ripasso di alcune definizioni sul ML
- Lo spazio delle ipotesi H
- Apprendimento di una funzione booleana: il concept learning
- Version Space

Apprendimento induttivo (Inductive Learning method)

- Un algoritmo per l'apprendimento (deterministico) supervisionato riceve in ingresso il valore corretto in alcuni punti di una funzione sconosciuta.
- In base a tali valori deve cercare di ricostruire la funzione nel modo più fedele possibile.
- Definiamo esempio una coppia $(x, f(x))$, in cui x è l'input e $f(x)$ è l'output della funzione.

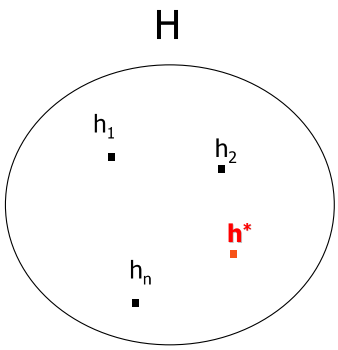
Apprendimento induttivo (Inductive Learning method)

- Il compito dell'inferenza induttiva pura (o induzione) è il seguente:

  Data una collezione di esempi di f, restituire una funzione h che approssima f.

- La funzione h prende il nome di ipotesi.

Spazio delle Ipotesi H



- Spazio delle ipotesi H: insieme di tutte le possibili funzioni candidate ad essere la funzione incognita da apprendere
- Determinazione dell'ipotesi *h* che più si avvicina ad una funzione target h^*
- $H=\{h_1, h2, \ldots, h_i, \ldots, h_n\}$

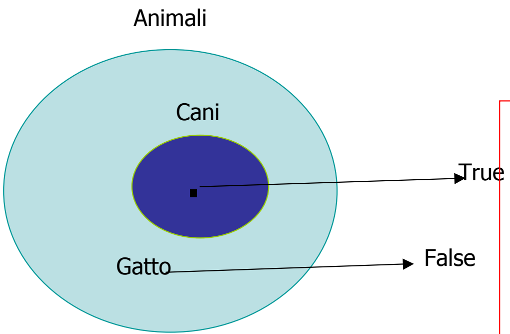
Apprendimento come ricerca di una particolare ipotesi h^* in uno spazio delle ipotesi H

Apprendimento di una funzione booleana:

Il Concept Learning

Concept Learning

Individuare automaticamente una funzione a valori booleani a partire dall'insieme degli esempi <x,(f(x)>



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

f: Animali->[true,false]

f(a)=true se a ∈ Cani

f(a)=false se a = altro animale

Concept Learning

Esempio del ristorante già visto per l'induzione con alberi di decisione

Istanze x

Esempio	Attributi										Obiettivo
	Alt	Bar	Ven	Fame	Affoll	Prezzo	Piogg	Prenot	Tipo	Attesa	Aspettiamo
X_1	Sì	No	No	Sì	Qualcuno	$$$	No	Sì	Francese	0–10	Sì
X_2	Sì	No	No	Sì	Pieno	$	No	No	Thai	30–60	No
X_3	No	Sì	No	No	Qualcuno	$	No	No	Fast-food	0–10	Sì
X_4	Sì	No	Sì	Sì	Pieno	$	Sì	No	Thai	10–30	Sì
X_5	Sì	No	Sì	No	Pieno	$$$	No	Sì	Francese	>60	No
X_6	No	Sì	No	Sì	Qualcuno	$$	Sì	Sì	Italiano	0–10	Sì
X_7	No	Sì	No	No	Nessuno	$	Sì	No	Fast-food	0–10	No
X_8	No	No	No	Sì	Qualcuno	$$	Sì	Sì	Thai	0–10	Sì
X_9	No	Sì	Sì	No	Pieno	$	Sì	No	Fast-food	>60	No
X_{10}	Sì	Sì	Sì	Sì	Pieno	$$$	No	Sì	Italiano	10–30	No
X_{11}	No	No	No	No	Nessuno	$	No	No	Thai	0–10	No
X_{12}	Sì	Sì	Sì	Sì	Pieno	$	No	No	Fast-food	30–60	Sì

- Output
- Input

Apprendere a predire il valore della funzione *Aspettiamo* sulla base degli attributi del giorno stesso, avendo a disposizione l'insieme degli esempi <x,f(x)>

Concept Learning

Funzione booleana da apprendere: "Giorni in cui il mio amico Aldo pratica il suo sport d'acqua preferito"

Istanze x

Esempio	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
x_1	Sunny	Warm	Normal	Strong	Warm	Same	**Yes**
x_2	Sunny	Warm	High	Strong	Warm	Same	**Yes**
x_3	Rainy	Cold	High	Strong	Warm	Change	**No**
x_4	Sunny	Warm	High	Strong	Cool	Change	**Yes**

Training set

- Output
- Input

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

(Mitchell, 1997)

Studiamo questa seconda funzione =>

Cosa fare

Dobbiamo mettere in piedi un procedimento per l'individuazione di una ipotesi *h* che si avvicini il più possibile alla funzione target h^* *nello spazio delle ipotesi H*

Punti da approfondire

1. Come rappresentiamo la funzione ipotesi da apprendere *h*
2. Come rappresentiamo lo spazio delle ipotesi H
3. Come cerchiamo la funzione h^* nello spazio delle ipotesi H
4. Come rappresentiamo l'insieme degli esempi

Costruzione della funzione ipotesi h

La cosa più semplice e naturale è quella di costruire l'ipotesi *h* come unione degli attributi della singola istanza *x:*

$h \equiv Sky \cup AirTemp \cup Humidity \cup Wind \cup Water \cup Forecast$

Rappresentazione della funzione ipotesi h

Sulla base di quanto detto precedentemente, rappresentiamo l'ipotesi *h* con un vettore di 6 elementi:

h=<Sky, AirTemp, Humidity, Wind, Water, Forecast>

Vediamo ora quali sono i valori ammissibili di ciascun attributo per costruire la generica ipotesi *h*

Valori degli attributi

Definiamo l'insieme dei valori ammissibili per ciascun attributo che compare nella ipotesi *h* nel modo seguente:

- **?**=> ogni valore dell'attributo è ammissibile
- Il valore tra quelli consentiti
- **0** => nessun valore ammissibile per l'attributo

In tal modo si riescono a rappresentare TUTTE le ipotesi dello spazio H

Esempio di ipotesi h

h=<?, Cold, High, ?,?,?>

Aldo pratica lo sport solo nei giorni freddi con umidità alta

Valori degli attributi

- Sotto le precedenti ipotesi possiamo individuare soprattutto due Ipotesi h importanti:

- Ipotesi più generale
  - h = <?,?,?,?,?,?> =>ogni giorno è buono per fare sport d'acqua
- Ipotesi h più specifica
  - h = <0,0,0,0,0,0>=>nessun giorno è buono per fare sport d'acqua

Definizioni

Diamo alcune definizioni importanti per il prosieguo del discorso

- **Insieme delle istanze X:** **l'insieme di valori sui quali il concetto è definito. (Insieme di tutti i possibili giorni)**
- **Target Concept c:** **la funzione che deve essere appresa.**
- **Training Examples D:** **formato da n istanze x∈X insieme a n valori corrispondenti c(x): <x, c(x)>**

Come già illustrato, se c(x)=1 si parla di *esempi positivi* mentre per c(x)=0 si parla di *esempi negativi*

Il task

Riassumendo, il task che andiamo a costruire ha le seguenti caratteristiche:

- **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,...,6 + <?> e <0>
- **Funzione target c**:EnjoySport: X->{0,1}
- **Esempi di training** D:esempi positivi e negativi della funzione target

- **Obiettivi:**
  - Determinare una ipotesi $h \in H$ tale che $\mathbf{h(x)=c(x)}\ \forall\ \mathbf{x} \in \mathbf{X}$

Ipotesi di apprendimento induttivo

Riprendiamo quanto visto nella precedente lezione sull'apprendimento induttivo e applichiamolo al nostro esempio

- Obiettivo: determinare una ipotesi h identica al concetto target c sull'intero set di istanze X
- Informazione disponibile: coppie <x,c(x)>

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

Ma quante sono le ipotesi h contenute in H?

Calcolo della cardinalità dell'insieme H

Vediamo come calcolare il numero di ipotesi *h* presenti in H
per la funzione *EnjoySport*

- **Numero di istanze in X**:
  1. Sky (Sunny, Cloudy e Rainy) (3)
  2. AirTemp (Warm, Cold) (2)
  3. Humidity (Normal, High) (2)
  4. Wind (Strong, Weak) (2)
  5. Water (Warm, Cool) (2)
  6. Forecast (Same, Change) (2)

  Totale= 3x2x2x2x2x2=96 istanze possibili in X.

- **Numero di ipotesi h sintatticamente diverse** (dobbiamo aggiungere il valore <0> e <?> a ciascuno degli attributi):
  - **Totale=5x4x4x4x4x4 = 5120 ipotesi h diverse tra loro.** (Es. h=<Sunny, Warm,Normal, ?, Cool,Change>)
- **Numero di ipotesi h semanticamente diverse**
  - **Totale=1+4x3x3x3x3x3 = 973 ipotesi h diverse tra loro**. Infatti bisogna togliere tutte le ipotesi che hanno uno <0> in almeno un campo in quanto classificano tutte le istanze come negative c(x)=0 + 1 che le rappresenta tutte

Apprendimento come ricerca nello spazio delle ipotesi H

- Il concept learning può essere visto quindi come il compito di trovare la funzione *h* attraverso una ricerca nello spazio delle ipotesi H
- Obiettivo della ricerca in H: trovare la funzione *h* che meglio approssimi la funzione *c* sui training examples (best fit h)

Studio di algoritmi e tecniche di ricerca in H

Ordinamento dell'insieme H

Poiché la ricerca è più semplice in un insieme ordinato, introduciamo nell'insieme H un ordinamento che faciliti la ricerca della funzione h. Altrimenti dovremmo cercare in modo esaustivo (H può essere infinito)

Ordinamento General-to-Specific di H (G-t-S) (Mitchell,1997)

- h_1 = <Sunny,?,?,Strong,?,?>
- h_2 = <Sunny,?,?,?,?,?>
- Consideriamo l'insieme delle istanze considerate positiva da h_1e 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 => **h_2 più generale di h_1**

Formalizzazione dell'ordinamento G-t-S

Definizione: $\forall\ x \in X$, $\forall\ h \in H$, x 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 ($h_j \geq_g h_k$) se e solo se

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

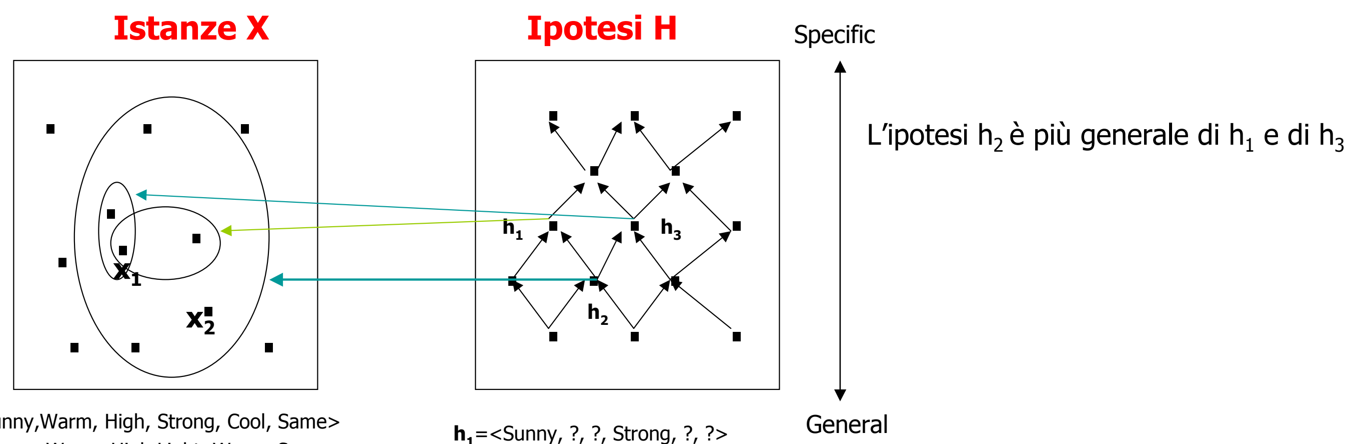
Ordinamento G-t-S più stretto

- 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 ($h_j >_g h_k$)
Se e solo se
$$(h_j(x) \geq_g h_k(x)) \wedge (h_k(x) \not\geq_g h_j(x))$$

h_k si dice more_specific_than h_j

Esempio



x_1=<Sunny,Warm, High, Strong, Cool, Same>
x_2=<Sunny, Warm, High,Light, Warm, Same>

h_1=<Sunny, ?, ?, Strong, ?, ?>
h_2=<Sunny, ?, ?, ?, ?, ?>
h_3=<Sunny,?, ?,?, Cool, ?>

- 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 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
  - Non c'è bisogno di effettuare ricerche su tutte le funzioni h di H

Esempio di algoritmo di ricerca: l'algoritmo Find-s

- 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)

- Algoritmo Find –S
  - Inizializza h alla ipotesi più specifica in H
  - 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← h_0=<0,0,0,0,0,0>
•x_1 h←<Sunny, Warm, Normal, Strong, Warm, Same>
•X_2 h←<Sunny, Warm, ?, Strong, Warm, Same>
•X_3 negativo. Passa oltre
•X_4 h←<Sunny, Warm, ?, Strong, ?, ?>
•Risultato: h←<Sunny, Warm, ?, Strong, ?, ?>

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
- Perchè prediligere l'ipotesi h 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?
- Per rispondere alle suddette domande si ha bisogno di strutturare lo spazio H in modo più articolato.

Definizioni

- Una ipotesi h è consistente con un insieme di training examples D se e solo se h(x)=c(x) per ogni esempio <x, c(x)> 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

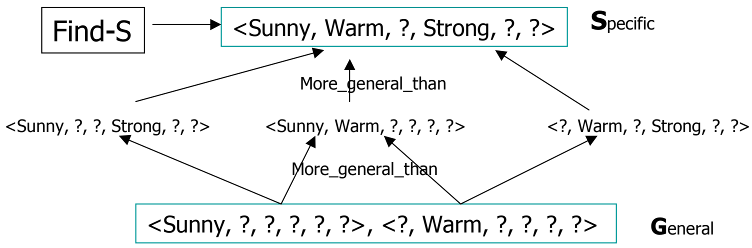
$$\boxed{VS_{H,D} \equiv \{\forall\ h \in H \mid Consistent(h,D)\}}$$

Algoritmo di ricerca sul Version Space
List-Then-Eliminate

- Version Space $\leftarrow$ una lista di tutte le ipotesi in H
- Per ogni training example
  - Rimuovi dal version space ogni ipotesi h tale che $h(x) \neq c(x)$ (inconsistenza)
- 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, G ed S che delimitano il version space dentro lo spazio parzialmente ordinato delle ipotesi, come nell’esempio della funzione EnjoySport:



Version space per la funzione EnjoySport

Definizioni

- Insieme G:rispetto allo spazio delle ipotesi H ed al training set D, è l’insieme dei membri più generali (maximally general) h di H e consistenti con D:

$$\mathbf{G \equiv \{\forall g \in H \mid Consistent(g,D) \wedge (\neg \exists g' \in H) \mid [(g' >_g g) \wedge Consistent(g',D)]\}}$$

- Insieme S:rispetto allo spazio delle ipotesi H ed al training set D, è l’insieme dei membri meno generali (maximally specific) h di H e consistenti con D:

$$\mathbf{S \equiv \{\forall s \in H \mid Consistent(s,D) \wedge (\neg \exists s' \in H) \mid [(s >_g s') \wedge Consistent(s',D)]\}}$$

Teorema di rappresentazione del Version Space

- Sia X un arbitrario insieme di istanze e sia H un insieme di ipotesi h a valori booleani definite su X. Sia c : X -> {0,1} un arbitrario concetto target definito su X e sia D un insieme arbitrario di esempi di training {<x,c(x)>}. Per tutti gli X, H, c e D tali che siano definiti gli insiemi G ed S si ha:

$$\mathbf{VS_{H,D} = \{\forall\ h \in H \mid (\exists\ s \in S,\ \exists\ g \in G) \Rightarrow (g \geq_g h \geq_g s)\}}$$

  - Il teorema afferma in sostanza che il Version Space si può rappresentare in termini dei suoi membri più generali e più specifici e quindi ogni ipotesi h si trova compresa tra S e G.