APPRENDIMENTO AUTOMATICO AA

- Definizione 1:
  - Learning is constructing or modifying representations of what is being experienced [Michalski 1986], pag. 10
- Definizione 2:
  - Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time [Simon 1984], pag. 28

Impieghi dell'AA

- A) estrazione di conoscenza
  - da utilizzare per il funzionamento di sistemi basati su conoscenza (ad esempio, sistemi per la classificazione)
  - a fini scientifici, ovvero scoperta di nuovi fatti e teorie attraverso l'osservazione e la sperimentazione
- B) miglioramento delle performance di una macchina
  - ad esempio, miglioramento delle capacita' motrici e cognitive di un robot

Tecniche di AA

- Tecniche simboliche (impieghi A e B)
    - diversi tipi di rappresentazione
        - rappresentazione attributo valore
        - rappresentazione del primo ordine
- Tecniche statistiche (impiego B)
- Reti neurali (impiego B)

Apprendimento induttivo

- Apprendimento induttivo: il sistema parte dai fatti e dalle osservazioni derivanti da un insegnante o dall'ambiente circostante e le generalizza, ottenendo conoscenza che, auspicabilmente, sia valida anche per casi non ancora osservati (**induzione**). Due tipi di apprendimento induttivo:
  - apprendimento da esempi: conoscenza acquisita a partire da un insieme di esempi positivi che sono istanze del concetto da imparare e di esempi negativi che sono non-istanze del concetto
  - apprendimento di regolarità: non c'e' un concetto da imparare, l'obiettivo e' quello di trovare regolarità (ovvero caratteristiche comuni) nelle istanze date

Apprendimento induttivo di concetti da esempi (1)

- Universo U: insieme U di tutti gli oggetti del dominio
- Concetto C: sottoinsieme C degli oggetti del dominio $C \subseteq U$
- Un linguaggio di descrizione degli oggetti Lo
- Un linguaggio di descrizione dei concetti Lc
- Una procedura che interpreti i linguaggi e verifichi se la descrizione Dc del concetto C e' soddisfatta dalla descrizione Dx di un oggetto x (si dice che Dc copre Dx).

Apprendimento induttivo di concetti da esempi (2)

- Informalmente:
  - apprendere un concetto C significa trovare una descrizione di C che consenta di dire se un oggetto $x \in U$ e' una istanza di C, ovvero se $x \in C$.

Apprendimento induttivo di concetti da esempi (3)

- Fatto: descrizione di un oggetto
- Esempio per un concetto C: fatto etichettato, con etichetta + se l'oggetto e' un'istanza di C, etichetta - se l'oggetto non e' una istanza di C.
- Training set: insieme E degli esempi (fatti etichettati). E e' composto dall'insieme degli esempi positivi E+ e dall'insieme degli esempi negativi E-

Apprendimento induttivo di concetti da esempi (4)

- Ipotesi: descrizione del concetto da imparare
- Se un fatto soddisfa una ipotesi si dice che l'ipotesi **copre** il fatto.
- Funzione per il test di copertura:

  copre(H,e)

  ritorna vero se e e' coperto da H e falso altrimenti
- Estensione ad insiemi di esempi:

  $$copre(H,E)=\{e \in E | copre(H,e)= true\}$$

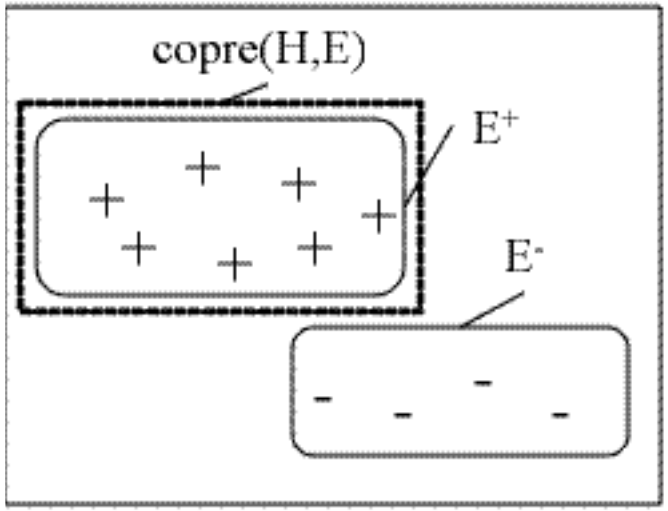
Apprendimento induttivo di concetti da esempi (5)

- Dato un insieme E di esempi positivi e negativi di un concetto C, espressi in un linguaggio di descrizione degli oggetti Lo,
- trovare un'ipotesi H, espressa in un dato linguaggio di descrizione dei concetti Lc, tale che:
  - ogni esempio positivo e+ $\in$ E+ sia coperto da H
  - nessun esempio negativo e- $\in$ E- sia coperto da H

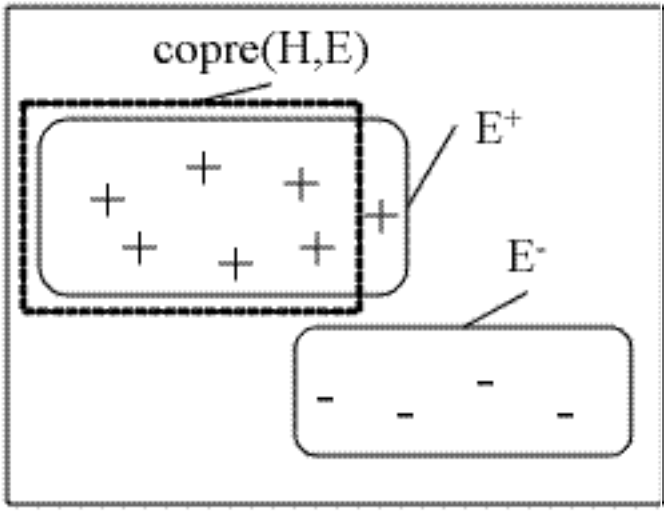
Completezza e consistenza

- Un'ipotesi H si dice completa se copre tutti gli esempi positivi, cioe' se copre(H,E+)=E+
- Un'ipotesi H si dice consistente se non copre nessun esempio negativo, cioe' se copre(H,E-)=∅

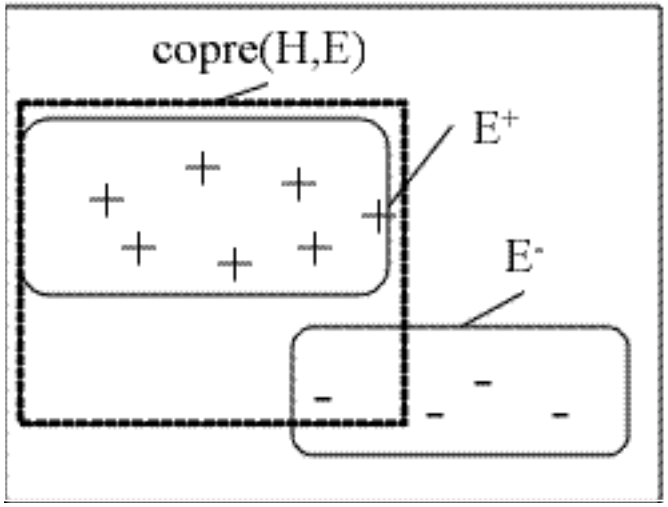
H: completa, consistente



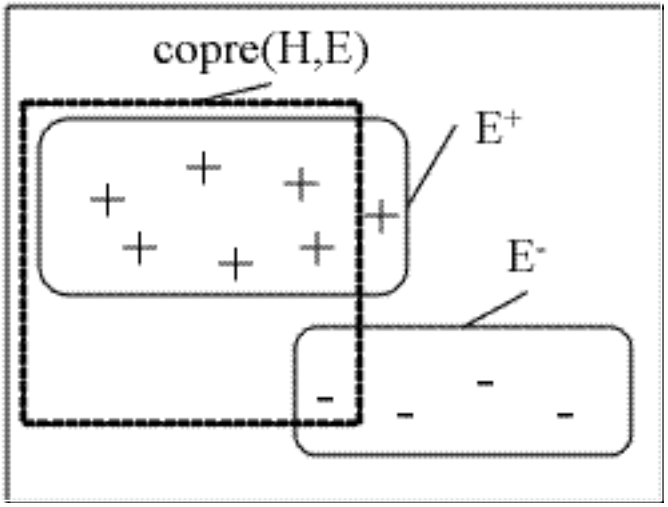
H: incompleta, consistente



H: completa, inconsistente



H: incompleta, inconsistente



Rappresentazione della conoscenza

- Linguaggi utilizzati per la rappresentazione di esempi e concetti:
  - Linguaggi attributo-valore
  - Linguaggi relazionali
  - Logica del primo ordine

Linguaggi attributo-valore

- Ciascuna istanza e' descritta dai valori assunti da un insieme di attributi (fisso per tutte le istanze).
- Gli attributi possono essere:
  - booleani o binari
  - nominali
  - ordinali
  - numerici
- Se k sono gli attributi, ciascuna istanza puo' essere rappresentata come un punto in uno spazio k-dimensionale

Esempio

- Universo: insieme degli atleti
- Istanze descritte dagli attributi
  - altezza, peso, corporatura

- Esempio di istanza

  altezza=1.85m, peso=110kg, corporatura=robusta

Regole di produzione

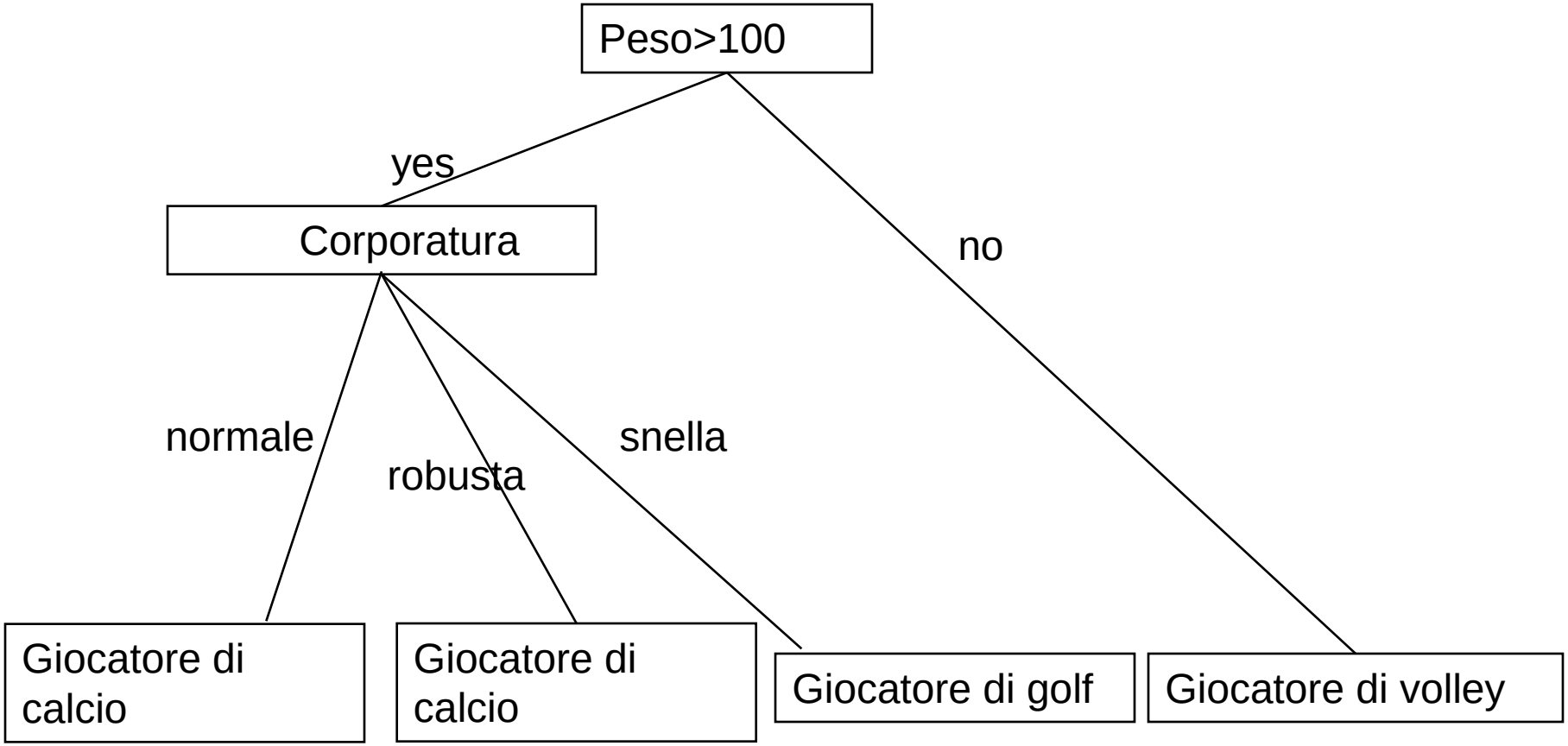
- Linguaggio di descrizione dei concetti:
  - regole di produzione
    - contengono nell'antecedente congiunzioni e disgiunzioni di uguaglianze o disuguaglianze tra un attributo e un valore e nel conseguente la classe dell'istanza
- Esempio:
- Concetto: giocatore di calcio
- Esempio di descrizione del concetto

peso>100 and (corporatura = normale or corporatura = robusta) $\rightarrow$ giocatore_di_calcio

Alberi di decisione

- Linguaggio di descrizione dei concetti:
  - alberi di decisione
    - ogni nodo corrisponde ad un test su un attributo (uguaglianza, disuguaglianza) e ciascun ramo che parte dal nodo e' etichettato con il risultato del test
    - ad ogni foglia corrisponde una classe

Alberi di decisione (esempio)



Linguaggi attributo-valore

- Linguaggio equivalente alla logica proposizionale:
  - ogni uguaglianza o disuguaglianza puo' essere vista come una proposizione (predicato con arita' 0)
  - non ci sono variabili, quantificatori e predicati con arita' > 1

Linguaggi relazionali

- I linguaggi attributo valore non sono adatti a rappresentare istanze costituite da sottoparti.
- Esempio: famiglia jones, componenenti:
  - name: dave, son: mike, father: ron
  - name: mike, son: junior, father: dave
  - name: junior, father: mike
- La descrizione di famiglia potrebbe essere trasformata in una lista di coppie attributo valore prevedendo un numero di attributi pari al prodotto del numero massimo di componenti per il numero massimo di attributi per componente: spreco di memoria

Linguaggi relazionali

- I linguaggi attributo valore sono quindi inefficienti per rappresentare questi universi. Vengono perciò utilizzati linguaggi relazionali (vedi esempio precedente).

- Tali linguaggi consentono descrizioni dei concetti che possono contenere variabili e quantificatori (non proposizionali).

- Descrizione del concetto di famiglia con un nonno

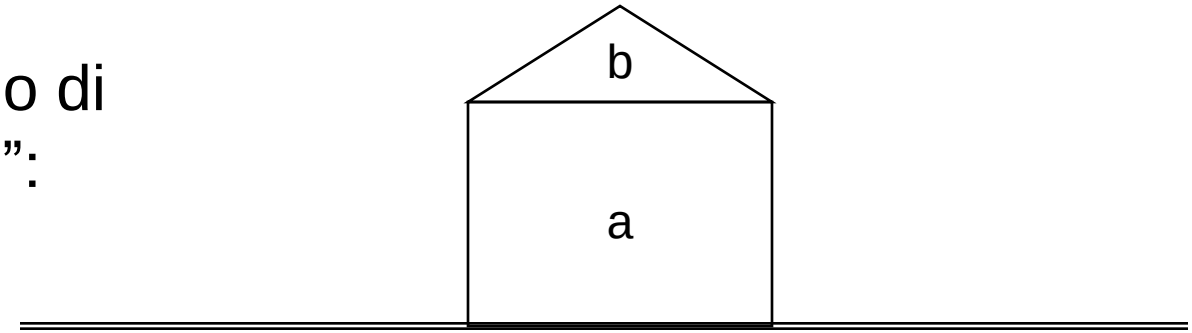
$$\forall x,y,z\ (son(x)=y \text{ and } son(y)=z)$$

Linguaggi del primo ordine

- Una ulteriore estensione del linguaggio consiste nell'utilizzare linguaggi logici del primo ordine.
- Essi forniscono la possibilità di rappresentare, oltre alle parti e agli attributi delle parti, anche relazioni tra le parti di una istanza.
- Forniscono inoltre una maniera semplice per rappresentare le parti di un oggetto utilizzando un predicato del tipo "oggetto" (nome-oggetto, nome-parte)
- Esempi non più descritti da componenti ma da letterali ground. Ogni attributo di un componente diventa un predicato.

Esempio: mondo a blocchi

•Esempio di "oggetto":



- Istanza e, forma attributo valore: componenti:
  - name: a, shape=square, size=large, on-table=yes
  - name: b, shape=triangle, size=small, on-table=no
- Istanza e, logica primo ordine:
- oggetto(e,a), oggetto(e,b), square(a), triangle(b), large(a), small(b), on-table(a) , **on(b,a)**
- "l'oggetto e contiene le parti a e b, esse hanno le proprieta'….."

Logica del primo ordine

- Rispetto ai linguaggi relazionali, essi consentono inoltre:
  - di definire i concetti in maniera ricorsiva

  - **`ancestor(X,Y):-father(X,Z),ancestor(Z,Y)`**

- rappresentano un linguaggio più studiato e quindi maggiormente formalizzato. Impiego della programmazione logica per la rappresentazione di oggetti e concetti

Tecniche di apprendimento

- Apprendimento attributo valore:
  - alberi di decisione
  - regole di produzione

- Apprendimento del primo ordine:
  - Programmazione Logica Induttiva

Apprendimento di alberi di decisione

- Sistemi che apprendono alberi di decisione: CLS, IDR, C4, ASSISTANT, ID5, C4.5 etc.
- Problemi appropriati:
  - le istanze sono rappresentate da coppie attributo valore
  - la funzione target ha valori discreti
  - descrizioni disgiuntive di concetti possono essere richieste
  - l'insieme dei dati di training può contenere errori
  - l'insieme dei dati di training può contenere dati mancanti

Alberi di decisione: c4.5

- c4.5 [Qui93b,Qui96] Evoluzione di ID3, altro sistema del medesimo autore, J.R. Quinlan
- Ispirato ad uno dei primi sistemi di questo genere, CLS (Concept Learning Systems) di E.B. Hunt
- Continuo studio ed aggiornamento (release 8): rimane uno dei punti di riferimento nella propria classe di algoritmi.
- Algoritmo scritto in C per Unix: disponibile da
- http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz

Algoritmo di determinazione dell'albero

- T: insieme degli esempi;
- $\{C_1, C_2, \ldots, C_k\}$: insieme delle classi;
- Considera l'insieme T:
  - T contiene uno o più esempi, tutti appartenenti alla medesima classe => singola foglia con etichetta la classe
  - T non contiene nessun esempio (insieme vuoto)=> singola foglia con etichetta la classe piu' frequente nell'insieme padre
  - T contiene casi che appartengono a più classi => partizionamento di T in piu' sottoinsiemi secondo un test su un attributo=> nodo associato al test, con un sottoalbero per ogni possibile risultato del test stesso. Richiama l'algoritmo su ogni ramo/sottoinsieme

Condizione di terminazione

- c4.5 non si ferma solo quando trova insiemi uniformi o vuoti, si ferma anche se
  - Nessun test e' tale che almeno due sottoinsiemi contengano un numero minimo di casi
  - Il numero minimo di casi e' un'opzione di c4.5, di default vale 2

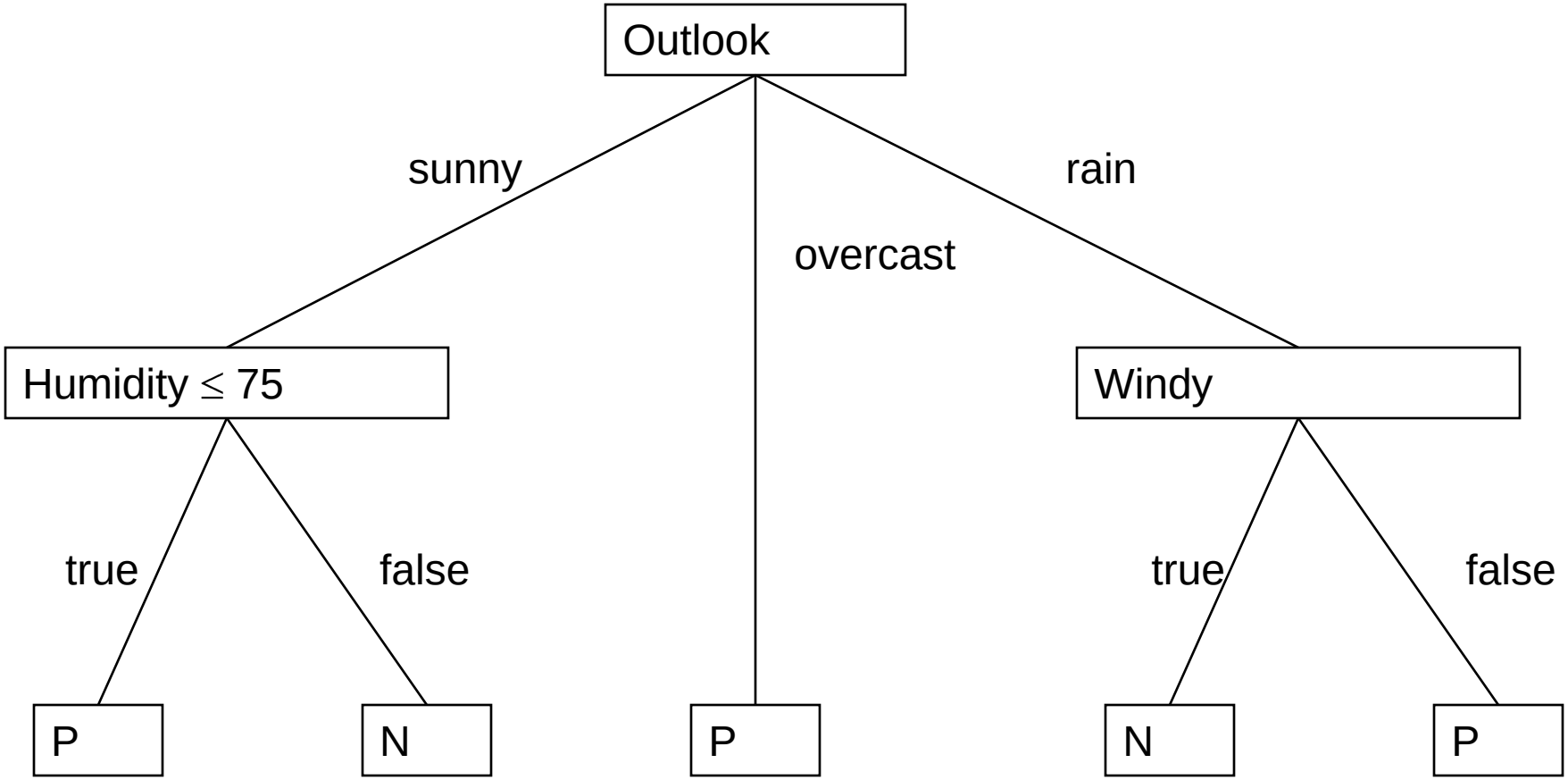
Esempio

- Istanze: sabati mattina
- Concetti: sabato adatto a giocare a tennis e
  sabato non adatto a giocare a tennis
- Attributi:
  - outlook, con valori {sunny,overcast,rain}
  - temperature, con valori numerici
  - humidity, con valori numerici
  - windy, con valori {true, false}

Training set

No	Outlook	Temp (°F)	Humid (%)	Windy	Class
D1	sunny	75	70	T	P
D2	sunny	80	90	T	N
D3	sunny	85	85	F	N
D4	sunny	72	95	F	N
D5	sunny	69	70	F	P
D6	overcast	72	90	T	P
D7	overcast	83	78	F	P
D8	overcast	64	65	T	P
D9	overcast	81	75	F	P
D10	rain	71	80	T	N
D11	rain	65	70	T	N
D12	rain	75	80	F	P
D13	rain	68	80	F	P
D14	rain	70	96	F	P

Albero di decisione



Albero di decisione

Outlook=sunny

| humidity ≤ 75: P

| humidity > 75: N

Outlook=overcast: P

Outlook=rain

| windy=True: N

| windy=False: P

Determinazione dell'albero

- Come scegliere il test (attributo) ad ogni passo?
- Generazione di tutti i possibili alberi: non praticabile dal punto di vista computazionale.
- Viene quindi scelto un test sulla base di una euristica con scelta irrevocabile (*nonbacktracking*)
- Varie euristiche basate su:
  - sull'informazione
  - sull'errore
  - sulla significativita'
- Vedremo solo quella basata sull'informazione

Entropia di un insieme di esempi

- Entropia di un insieme di esempi: informazione media necessaria ad identificare la classe di un esempio in T. Dato l'insieme di esempi S e la classe C_j

$$info(S) = -\sum_{j=1}^{k} \frac{freq(C_j, S)}{|S|} \times \log_2 \left(\frac{freq(C_j, S)}{|S|} \right)$$

- Entropia del training set

info(T)

- Può anche essere vista come una misura della disuniformita' di una arbitraria collezione di esempi: piu' e' alta piu' la collezione e' disuniforme

Esempio (1)

- Caso di due sole classi, siano p^+ e p^- la proporzione di esempi positivi e negativi nel training set ($p^-=1-p^+$).

- L'entropia e' data da:

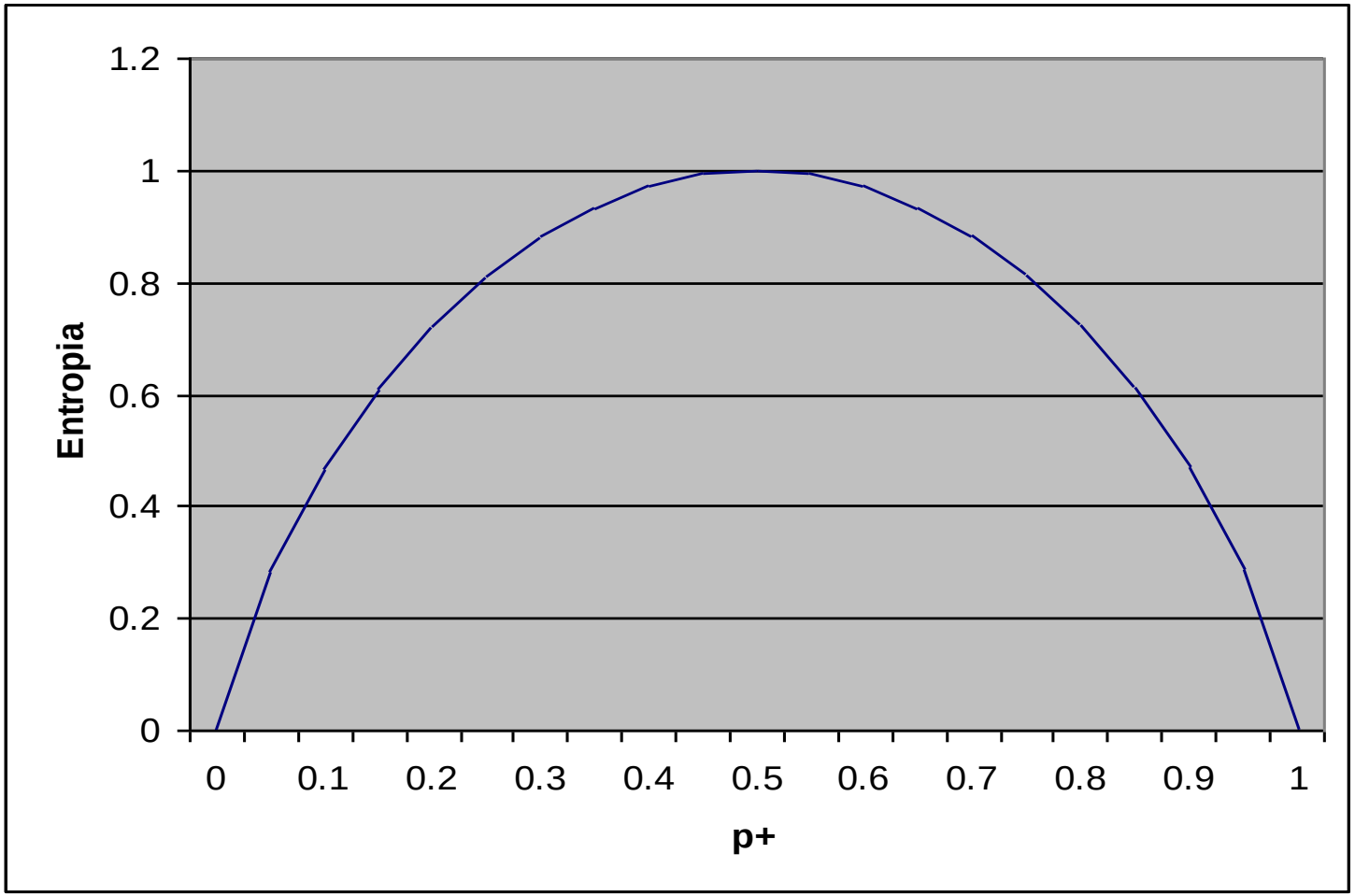
$$info(T) = -p^+ \times \log_2 p^+ - p^- \times \log_2 p^-$$

Esempio (2)

- L'entropia e' minima quando tutti gli esempi in T appartengono alla stessa classe:
  - $p^+=0$ (oppure $p^+=1$)=> info(T)=0
    - definendo $0*\log_2 0=0$
  - Il ricevente sa che l'esempio estratto sarà sempre negativo (o positivo) e nessun messaggio deve essere mandato

- L'entropia e' massima quando meta' degli esempi appartiene ad una classe e l'altra meta' all'altra classe: $p^+=0.5$ => info(T)=1

Entropia



Training set

No	Outlook	Temp (°F)	Humid (%)	Windy	Class
D1	sunny	75	70	T	P
D2	sunny	80	90	T	N
D3	sunny	85	85	F	N
D4	sunny	72	95	F	N
D5	sunny	69	70	F	P
D6	overcast	72	90	T	P
D7	overcast	83	78	F	P
D8	overcast	64	65	T	P
D9	overcast	81	75	F	P
D10	rain	71	80	T	N
D11	rain	65	70	T	N
D12	rain	75	80	F	P
D13	rain	68	80	F	P
D14	rain	70	96	F	P

Esempio

- Consideriamo il problema della decisione di giocare o meno a tennis
- 14 esempi, 9 positivi, 5 negativi
- Entropia di T

$$info(T) = -p^{+} \times \log_2 p^{+} - p^{-} \times \log_2 p^{-}$$

- $info(T) = -(9/14)log_2(9/14) - (5/14)log_2(5/14) = 0.940$

Entropia di un insieme di esempi

- Entropia dopo il partizionamento (secondo il test X): media pesata delle entropie dei singoli sotto-insiemi

$$info_X(T) = -\sum_{j=1}^{n} \frac{|T_i|}{|T|} \times info(T_i)$$

- Guadagno di informazione

$$gain(X) = info(T) - info_X(T)$$

- L'entropia diminuisce man mano che ciascun sottoinsieme contiene sempre più esempi da una classe e sempre meno dalle altre ovvero man mano che i sottoinsiemi diventano più uniformi.

Training set

No	Outlook	Temp (°F)	Humid (%)	Windy	Class
D1	sunny	75	70	T	P
D2	sunny	80	90	T	N
D3	sunny	85	85	F	N
D4	sunny	72	95	F	N
D5	sunny	69	70	F	P
D6	overcast	72	90	T	P
D7	overcast	83	78	F	P
D8	overcast	64	65	T	P
D9	overcast	81	75	F	P
D10	rain	71	80	T	N
D11	rain	65	70	T	N
D12	rain	75	80	F	P
D13	rain	68	80	F	P
D14	rain	70	96	F	P

Esempio

- Test sull'attributo wind:
  - wind=F => T_F contiene 6 positivi e 2 negativi
  - wind=T => T_T contiene 3 positivi e 3 negativi
- T=[9,5]
- T_F=[6,2] $info(T_F)=-6/8*\log_2(6/8)-2/8*\log_2(2/8)=0.811$
- T_T=[3,3] $info(T_T)=-3/6*\log_2(3/6)-3/6*\log_2(3/6)=1$

$gain(wind)=info(T)-(8/14*info(T_F)+6/14*info(T_T))=$
$0.940-(8/14)*0.811-(6/14)*1=0.048$

Esempio

- Test sull'attributo outlook:
  - outlook=rain => T_{rain} contiene 3 pos e 2 neg
  - outlook=sunny => T_{sunny} contiene 2 pos e 3 neg
  - outlook=overcast => $T_{overcast}$ contiene 4 pos e 0 neg

- T=[9,5], T_{rain}=[3,2], T_{sunny}=[2,3], $T_{overcast}$=[4,0]

gain(outlook)=info(T)-((5/14)*info(T_{rain})+(5/14)*info(T_{sunny})+ (4/14)*info($T_{overcast}$))=

0.940-0.357*0.971-0.357*0.971-0.286*0=0.246

Generazione dei test

- **Problema:** generazione dell'insieme di test possibili.
- Definizione a priori delle forme ammissibili
- Tre possibili tipi di test
  - Attributo discreto, un risultato per ogni valore.
  - Attributo discreto, un risultato per ogni gruppo di valori.
  - Attributo continuo, due possibili risultati (test binario).
- Ulteriore vincolo: almeno due tra $T_1, T_2, \ldots, T_n$ devono contenere un numero minimo di esempi.

Test su attributi discreti

- Un risultato per ogni valore: completamente definito.

- Un risultato per ogni gruppo di valori: bisogna stabilire quanti e quali gruppi considerare.
  - Conoscenza sul domino: creazione a priori di raggruppamenti significativi (nuovi attributi).
  - Test di tutti i possibili partizionamenti dell'insieme dei valori.

Test su attributi continui

- L'attributo continuo A assume m valori in T
- Siano essi $\{V_1, V_2, \ldots, V_m\}$ in ordine dal più piccolo al più grande
- Valore di soglia $Z=V_i$, divide gli m valori in due gruppi:
  - $A \leq Z$: $\{V_1, \ldots, V_i\}$
  - $A > Z$: $\{V_{i+1}, \ldots, V_m\}$
- m-1 candidati: V_1, ..., V_{m-1}
- Valutazione di ciascuno degli m-1 candidati per il valore di soglia.

Attributi con valore non specificato

- Valutazione dei test
  - informazione nell'attributo non specificato: nulla
- Guadagno: posto $F \subseteq T$ insieme in T per cui A e' specificato e sia X un test su A

$$info(T)=info(F)$$

$$info_X(T)=info_X(F)$$

- perché il test su A per gli esempi in T\F non da' alcuna informazione riguardante la classe.
- Gain(X)=(Probabilita' che A sia noto)x(info(T)-$info_X$(T))+(Probabilita' che A sia incognito)x0=

$$= \frac{|F|}{|T|} \times (info(F) - info_X(F))$$

Esempio

- Si supponga che nel database di esempio il caso:
- Outlook=overcast, Temperature=72, Humidity=90, Windy=T
- sia sostituito con
- Outlook=?, Temperature=72, Humidity=90, Windy=T
- Considerando i 13 casi per cui Outlook e' noto si hanno le seguenti frequenze

Outlook	Play	Don't Play	Totale
Sunny	2	3	5
Overcast	3	0	3
Rain	3	2	5
Totale	8	5	13

Esempio

- Si usino le frequenze relative per calcolare il gain di un test su Outlook
- info(T)=-8/13*$\log_2$(8/13)-5/13*$\log_2$(5/13)=0.961
- $\text{info}_{\text{Outlook}}$(T)=5/13*(-2/5*$\log_2$(2/5)-3/5*$\log_2$(3/5))
  +3/13*(-3/3*$\log_2$(3/3)-0/3*$\log_2$(0/3))
  +5/13*(-3/5*log2(3/5)-2/5*log2(2/5))
  =0.747
- gain(Outlook)=13/14*(0.961-0.747)=0.199 (un po' inferiore al precedente)

Partizionamento

- Generalizzato in modo probabilistico: si associa ad ogni esempio e in T un peso w, inizialmente pari ad 1
- Si consideri un test su A
  - se un esempio e, con peso w in T, ha $A=V_i$, si pone e in T_i con peso w e in T_j con $j \neq i$ con peso 0
  - se un esempio e, con peso w in T, ha A sconosciuto, si pone e in ciascun T_j con peso pari a $w \times P_{Vj}$, dove P_{Vj} e' la probabilita' del valore V_j
  - P_{Vj} puo' venire stimato come la somma dei pesi dei casi in T che hanno $A=V_j$ diviso la somma di tutti i pesi dei casi in T hanno A noto

Esempio

- Partizionamento secondo l'attributo Outlook
- I 13 casi per cui Outlook e' noto non presentano problemi
- Il caso rimanente e' assegnato a tutti i blocchi, corrispondenti ai valori sunny, overcast e rain, con pesi 5/13, 3/13 e 5/13 rispettivamente

Esempio

- Consideriamo il subset corrispondente a outlook=sunny

No	Outlook	Temp	Humidity	Windy	Classe	Weight
D1	sunny	75	70	T	P	1
D2	sunny	80	90	T	N	1
D3	sunny	85	85	F	N	1
D4	sunny	72	95	F	N	1
D5	sunny	69	70	F	P	1
D6	?	72	90	T	P	5/13

- Se questo set e' partizionato dallo stesso test su humidity, la distribuzione delle classi nei sottoinsiemi e'
- Humidity ≤ 75 2 classe P, 0 classe N
- Humidity > 75 5/13 classe P, 3 classe N

Esempio

- Il secondo sottoinsieme contiene ancora esempi da due classi ma nessun test produce due sottoinsiemi con almeno due esempi ciascuno quindi=> stop

Outlook=sunny

| humidity $\leq$ 75: P (2.0)

| humidity > 75: N (3.4/0.4)

Outlook=overcast: P (3.2)

Outlook=rain

| windy=True: N (2.4/0.4)

| windy=False: P (3.0)

Interpretazione dell'output

Numeri (A,B) associati a ciascuna foglia

- A=numero totale di esempi del training set associati alla foglia
- B=numero di esempi del training set associati alla foglia classificati erroneamente
- Ad esempio

N (3.4/0.4)

- Significa che 3.4 casi appartengono alla foglia di cui 0.4 non appartengono alla classe N

Classificazione di nuovi casi

- Generalizzata in modo probabilistico. Classificazione di un esempio e avente A sconosciuto:
  - si assegna inizialmente ad e il peso 1
  - se si incontra un nodo con un test su A, si esplorano tutti i possibili sottorami. Si esplora il sottoramo per V_j assegnando ad e il peso P_{Vj}
  - alla fine si raggiungeranno più foglie: si ottiene una distribuzione di classi invece di una singola classe. La probabilità di ciascuna classe sarà data dal peso dell'esempio che ha raggiunto la foglia
  - se due foglie sono associate alla stessa classe C, la probabilità di C e' data dalla somma delle probabilità dei pesi dell'esempio nelle due foglie

Esempio

- Si vuole classficare l'esempio
  - Outlook=sunny, Temperature=70, Humidity=?, Windy=F
  - Outlook=sunny => primo sottoalbero
  - Humidity=? Non ci permette di determinare se humidity ≤ 75
- Si scende lungo i due rami assegnando all'esempio due pesi frazionari:
  - ramo humidity ≤ 75 con peso 2.0/5.4=0.370
  - ramo humidity > 75 con peso 3.4/5.4=0.630

Esempio

- La frazione di esempio scesa lungo il ramo humidity ≤ 75 viene classificata come P con probabilita' 100%
- La frazione di esempio scesa lungo il ramo humidity > 75 viene classificata come
  - N con probabilita' 3/3.4=88%
  - P con probabilita' 0.4/3.4=12%
- Complessivamente, la distribuzione di classi relativa all'esempio e'
  - P: 0.370*100%+0.630*12%=44%
  - N: 0.630*88%=56%

Osservazioni sulla ricerca di c4.5 (1)

- c4.5 compie una ricerca hill-climbing attraverso l'insieme di tutti i possibili alberi di decisione partendo dalla ipotesi piu' semplice e andando via via verso ipotesi piu' complesse
- Osservazioni:
  - Lo spazio dei possibili alberi di decisione equivale al powerset di tutti i possibili esempi quindi c4.5 non rischia che lo spazio delle ipotesi possa non contenere il concetto target
  - c4.5 mantiene solo una singola ipotesi durante la ricerca.

Osservazioni sulla ricerca di c4.5 (2)

- c4.5 non compie backtracking nella ricerca. Corre quindi il rischio di incorrere in una soluzione ottimale solo localmente
- c4.5 utilizza tutti gli esempi di training ad ogni passo per decidere come raffinare l'albero differentemente da altri metodi che prendono le decisioni in maniera incrementale basandosi su esempi individuali. Il risultato e' che la ricerca di c4.5 e' meno sensibile agli errori nei singoli esempi.

Link utili

- C4.5

  http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz
- C5.0, versione commerciale di c4.5, versione demo (al massimo 400 esempi) scaricabile da

  http://www.rulequest.com/download.html

Bibliografia

[Mit97] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997

[Qui93b] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann Publishers, San Mateo, California, 1993

[Qui96] J. R. Quinlan, *Improved Use of Continuous Attributes in C4.5*, Journal of Artificial Intelligence Research, 4, pag. 77--90, 1996. ftp://ftp.cs.cmu.edu/project/jair/volume4/quinlan96a.ps

[Wit99] I.H. Witten, E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999.