Alberi di Decisione

decision trees

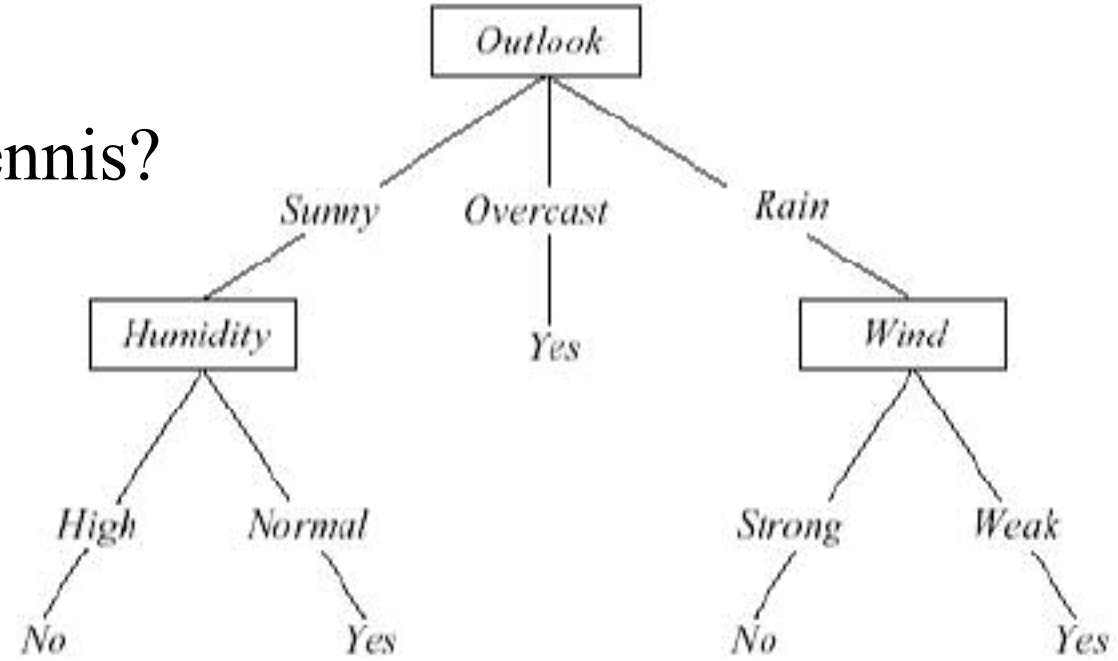
Alberi di Decisione

- Tipo di apprendimento: **supervisionato**, **da esempi**
- Tipo di funzione appresa: come per Find-S e VS, una **funzione simbolica**
- La funzione di classificazione appresa è un **albero di decisone**, alternativamente esprimibile mediante una **espressione logica disgiuntiva**
- Vantaggi: Maggior **potere espressivo** della funzione obiettivo, **tolleranza al rumore e ai dati incompleti**

Alberi di Decisione

- Un albero di decisione prende in ingresso un'istanza $x \in X$ descritta mediante **un vettore di coppie (attributo, valore)** ed emette in uscita una "**decisione**", ad es. binaria (sì o no)

PlayTennis?



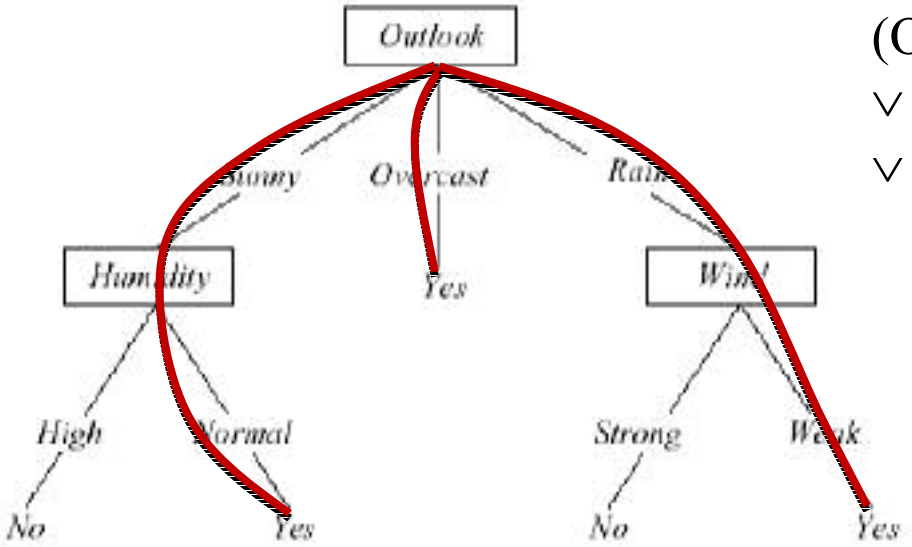
(Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong) → no

(Outlook = Rain, Temperature = Hot, Humidity = High, Wind = Weak) → sì

Obiettivo

- Lo scopo è, come in tutti i modelli di apprendimento induttivo da esempi, **imparare la definizione di una funzione obiettivo** espressa in termini di albero di decisioni.
- Un albero di decisione può essere espresso come una disgiunzione (OR) di congiunzioni di vincoli sui valori degli attributi delle istanze

PlayTennis?



(Outlook = Sunny ∧ Humidity = Normal)
∨ (Outlook = Overcast)
∨ (Outlook = Rain ∧ Wind = Weak)

Obiettivo

- Lo scopo è, come in tutti i modelli di apprendimento induttivo da esempi, **imparare la definizione di una funzione obiettivo** espressa in termini di albero di decisioni.
- Un albero di decisione può essere espresso come una disgiunzione (OR) di congiunzioni di vincoli sui valori degli attributi delle istanze
    - Gli alberi di decisione hanno il potere espressivo dei linguaggi proposizionali, ovvero qualsiasi funzione booleana può essere scritta come un albero di decisione e viceversa

Rappresentazione degli Esempi

- L'albero di decisione viene appreso a partire dagli esempi nell'insieme di addestramento $D \subseteq X \times O$, dove X è l'insieme delle possibili istanze e O l'output dell'albero (es. se l'albero emette risposta booleana $O = \{ 0, 1 \}$)
- Per descrivere le istanze di X si scelgono n **attributi** a_1, a_2, …, a_n
  - Gli attributi sono proprietà che descrivono gli esempi del dominio (es. Outlook = { Sunny, Overcast, Rain })
- Un esempio $x \in X$ è rappresentato da un vettore che specifica i valori degli n attributi:
  - $x = (a_1=val_i, a_2=val_j, \ldots, a_n=val_m)$

Quando è appropriato usare gli Alberi di Decisione

- Gli esempi (istanze) sono rappresentabili in termini di coppie attributo-valore
- La funzione obiettivo assume valori nel discreto. Un albero di decisioni assegna classificazioni booleane ma può essere esteso al caso di funzioni a più valori. Non è comune, ma possibile, l'utilizzo di questa tecnica per apprendere funzioni nel continuo (discretizzando i valori di f(x)).
- E'appropriato rappresentare il concetto da apprendere mediante una forma normale disgiuntiva
- I dati di apprendimento possono contenere errori, oppure attributi di cui il valore è mancante

Come utilizzare gli esempi D?

- **L'insieme di addestramento** D è l'insieme completo degli esempi sottoposti al sistema di apprendimento

Day	Outlook	Temperature	Humidity	Wind	PlayTenni
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Come utilizzare gli esempi D?

- **L'insieme di addestramento** D è l'insieme completo degli esempi sottoposti al sistema di apprendimento

Day	Outlook	Temperature	Humidity	Wind	PlayTenni
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- Una soluzione semplice sarebbe creare una espressione congiuntiva per ogni esempio e costruire una disgiunzione: ma il sistema non avrebbe alcun potere predittivo su esempi non visti! Il problema è estrarre uno **schema** dagli esempi, che sia in grado di **estrapolare** al caso di esempi non visti
- L'obiettivo è (come sempre) di estrarre uno schema conciso.

Il principio del Rasoio di Occam

- Scegli l'ipotesi più semplice che sia consistente con tutte le osservazioni

"L'ipotesi più semplice è che si sia tagliata con il rasoio di Occam"

Algoritmo di apprendimento di alberi di decisione

- Il problema di identificare l'albero *più piccolo* è intrattabile. Tuttavia esistono euristiche che consentono di trovare alberi "abbastanza" piccoli.
- L'idea consiste nell'analizzare **dapprima gli attributi più importanti**, ovvero quelli che discriminano di più.
- Più avanti vedremo come la teoria dell'informazione può aiutare nella scelta dell'attributo migliore.
- Supponendo per ora di poter fare questa scelta ad ogni passo i, l'algoritmo di creazione di un albero delle decisioni da un training set D è il seguente:

L'algoritmo ID3

function ID3(D, A) **returns** un albero di decisione (meglio, la sua radice) che classifica correttamente gli esempi in D

- D è l'insieme di addestramento
- A è la lista di altri attributi che devono ancora essere testati dall'albero

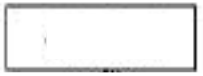
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k
- **if** A = ∅, **then return** *la radice con etichetta* VALORE-MAGGIORANZA(D)
- a ← *l'attributo di A che* ***classifica meglio*** *gli esempi* D
- *L'attributo di decisione per il nodo radice è dunque a*
- **for each** *valore* v_i *dell'attributo a,*
  - *Aggiungi un nuovo ramo sotto la radice, corrispondente al test* $a = v_i$
  - *Sia* D_{vi} *il sottoinsieme di esempi in D che assumono valore* v_i *per l'attributo a*
  - **if** $D_{vi} = \emptyset$ **then** sotto questo nuovo ramo, aggiungi una foglia con etichetta VALORE-MAGGIORANZA(D)
  - **else** *sotto il nuovo ramo, aggiungi il sottoalbero dato da ID3(*D_{vi}, A - { a })
- **return** *il nodo radice*

Esempio

ID3(D, {Outlook, Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



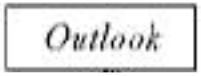
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k
- **if** A = ∅, **then return** *la radice con etichetta* VALORE-MAGGIORANZA(D)

Esempio

ID3(D, {Outlook, Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



per ora ci fidiamo che l'attributo che classifica meglio gli esempi di D è Outlook

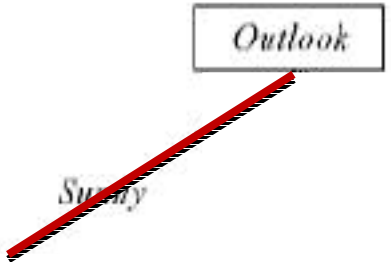
- a ← *l'attributo di A che* ***classifica meglio*** *gli esempi* D
- *L'attributo di decisione per il nodo radice è dunque a*

Esempio

ID3(D, {Outlook, Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



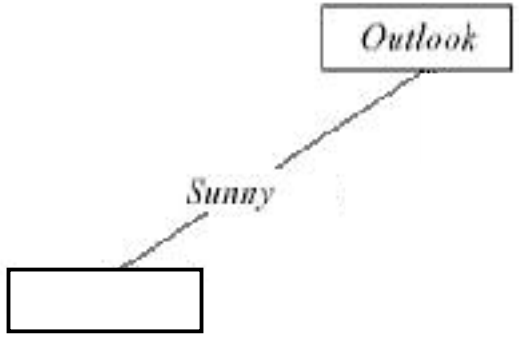
- **for each** *valore v_i dell'attributo a,*
  - *Aggiungi un nuovo ramo sotto la radice, corrispondente al test $a = v_i$*
  - *Sia D_{vi} il sottoinsieme di esempi in D che assumono valore v_i per l'attributo a*
  - **if** $D_{vi} = \varnothing$ **then** sotto questo nuovo ramo, aggiungi una foglia con etichetta VALORE-MAGGIORANZA(D)
  - **else** *sotto il nuovo ramo, aggiungi il sottoalbero dato da ID3(D_{vi},* A - { a })

Esempio

ID3($D_{Outlook=Sunny}$, {Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



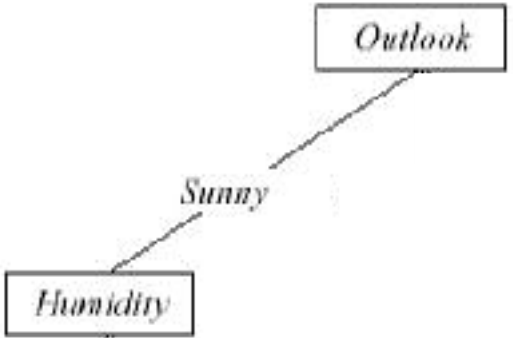
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k
- **if** A = ∅, **then return** *la radice con etichetta* VALORE-MAGGIORANZA(D)
- a ← *l'attributo di A che* ***classifica meglio*** *gli esempi* D
- *L'attributo di decisione per il nodo radice è dunque a*

Esempio

ID3($D_{Outlook=Sunny}$, {Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



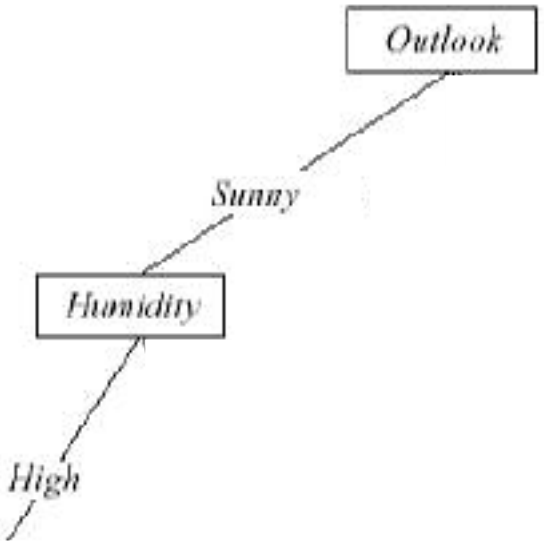
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k
- **if** A = ∅, **then return** *la radice con etichetta* VALORE-MAGGIORANZA(D)
- a ← *l'attributo di A che* ***classifica meglio*** *gli esempi* D
- *L'attributo di decisione per il nodo radice è dunque a*

Esempio

ID3($D_{Outlook=Sunny}$, {Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



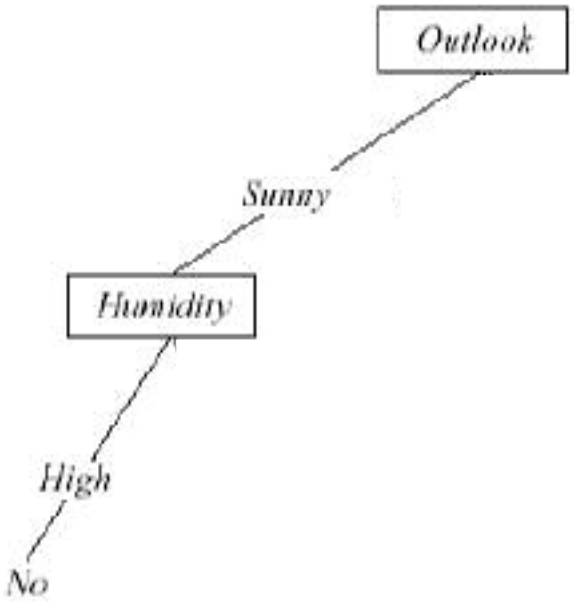
- **for each** *valore v_i dell'attributo a,*
  - *Aggiungi un nuovo ramo sotto la radice, corrispondente al test $a = v_i$*
  - *Sia D_{vi} il sottoinsieme di esempi in D che assumono valore v_i per l'attributo a*
  - **if** $D_{vi} = \varnothing$ **then** sotto questo nuovo ramo, aggiungi una foglia con etichetta VALORE-MAGGIORANZA(D)
  - **else** *sotto il nuovo ramo, aggiungi il sottoalbero dato da ID3(D_{vi},* A - { a })

Esempio

ID3($D_{Humidity=High}$, {Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



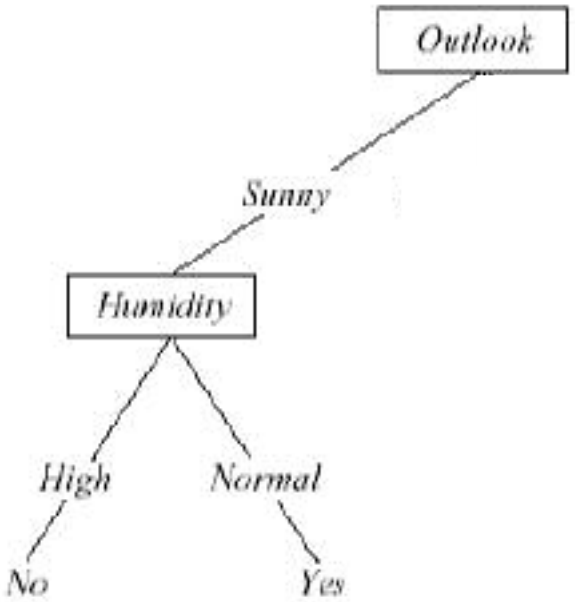
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k

Esempio

ID3($D_{Humidity=Normal}$, {Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



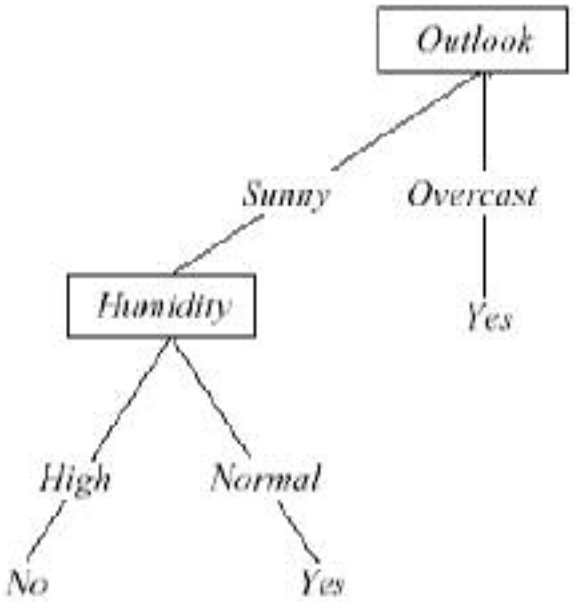
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k

Esempio

ID3($D_{Outlook=Overcast}$, {Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



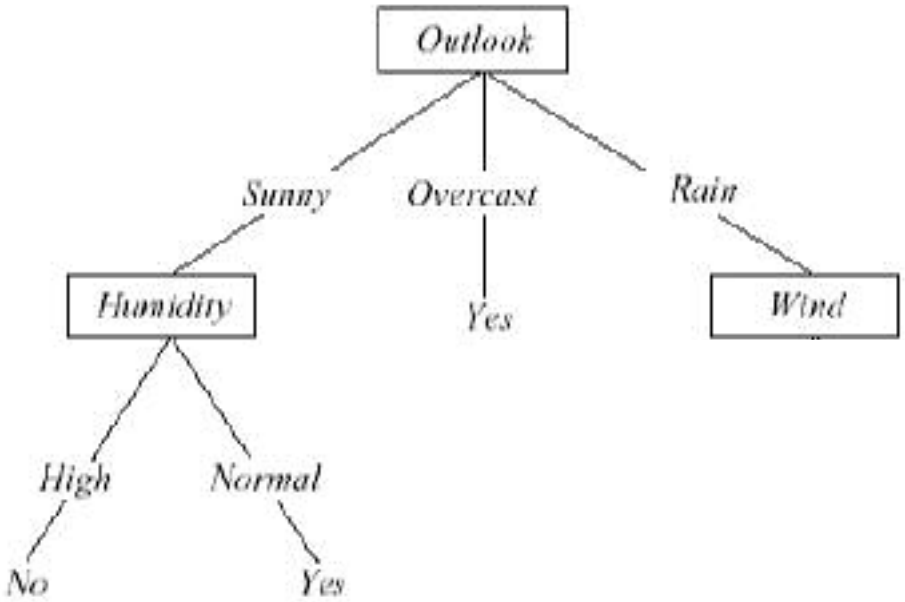
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k

Esempio

ID3($D_{Outlook=Rain}$, {Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



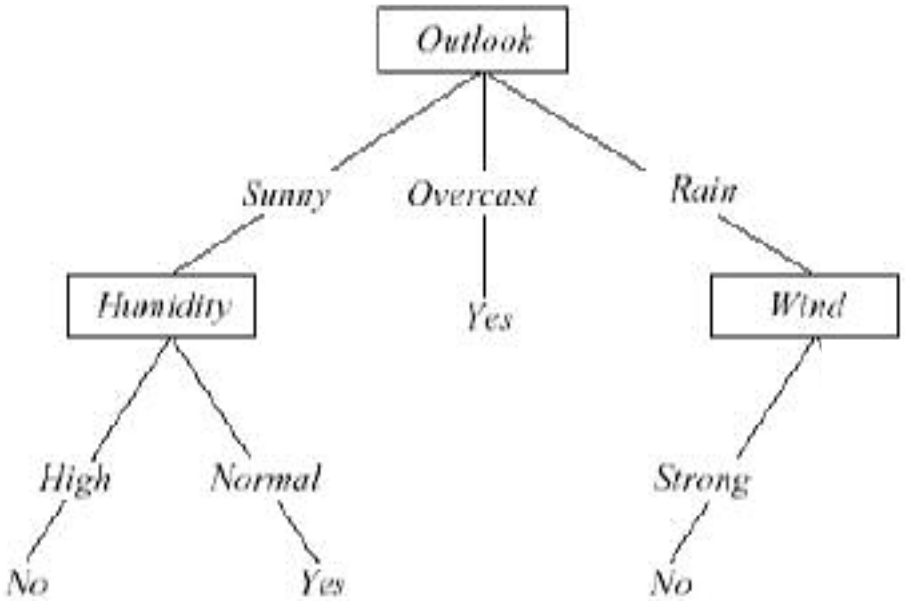
- **crea** un nodo radice per l'albero
- **if** D *contiene solo esempi di classe* c_k, **then return** *la radice con etichetta* c_k
- **if** A = ∅, **then return** *la radice con etichetta* VALORE-MAGGIORANZA(D)
- a ← *l'attributo di A che* ***classifica meglio*** *gli esempi* D
- *L'attributo di decisione per il nodo radice è dunque a*

Esempio

ID3($D_{Outlook=Rain}$, {Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



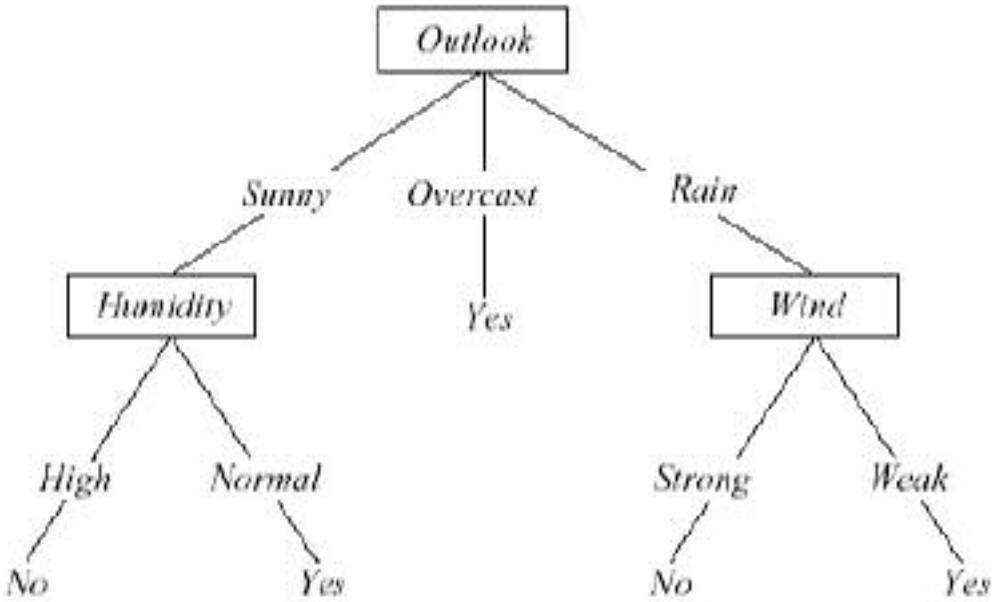
- **for each** *valore v_i dell'attributo a,*
  - *Aggiungi un nuovo ramo sotto la radice, corrispondente al test $a = v_i$*
  - *Sia D_{vi} il sottoinsieme di esempi in D che assumono valore v_i per l'attributo a*
  - **if** $D_{vi} = \varnothing$ **then** sotto questo nuovo ramo, aggiungi una foglia con etichetta VALORE-MAGGIORANZA(D)
  - **else** *sotto il nuovo ramo, aggiungi il sottoalbero dato da ID3(D_{vi},* A - { a })

Esempio

ID3($D_{Outlook=Rain}$, {Humidity, Wind})

D =

Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D3	Overcast	High	Weak	Yes
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D7	Overcast	Normal	Strong	Yes
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D10	Rain	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes
D14	Rain	High	Strong	No



- **for each** *valore v_i dell'attributo a,*
  - *Aggiungi un nuovo ramo sotto la radice, corrispondente al test $a = v_i$*
  - *Sia D_{vi} il sottoinsieme di esempi in D che assumono valore v_i per l'attributo a*
  - **if** $D_{vi} = \varnothing$ **then** sotto questo nuovo ramo, aggiungi una foglia con etichetta VALORE-MAGGIORANZA(D)
  - **else** *sotto il nuovo ramo, aggiungi il sottoalbero dato da ID3(D_{vi},* A - { a })

Algoritmo ID3

- ID3 è un algoritmo **greedy** che accresce l'albero secondo un'ordine **top-down**, selezionando ad ogni nodo l'attributo che classifica meglio gli esempi correntemente disponibili
- L'algoritmo procede finché tutti gli esempi sono classificati perfettamente, o sono stati esaminati tutti gli attributi
- Il passo "cruciale" è la **scelta dell'attributo migliore**

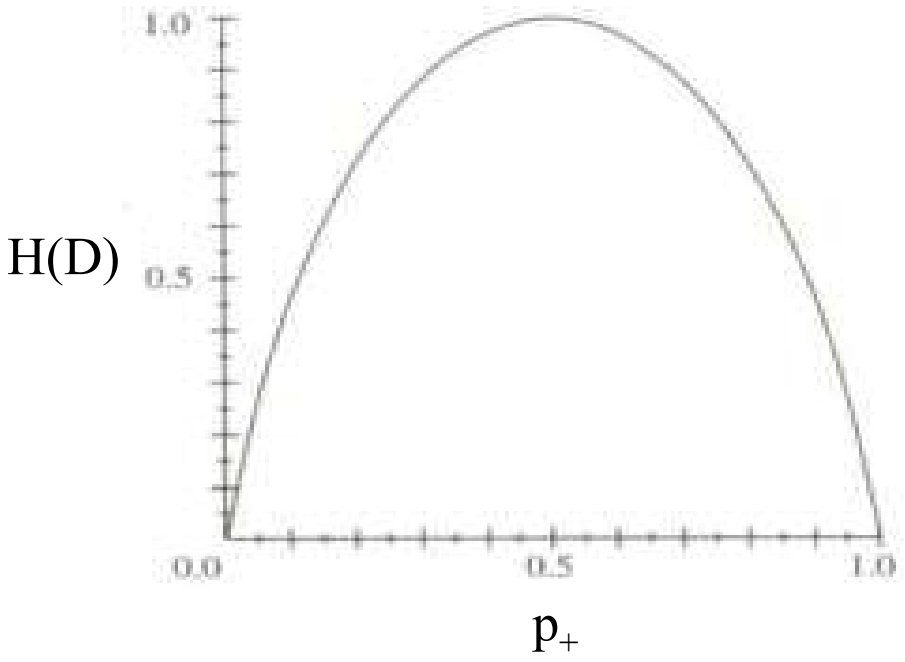
Entropia

- Interpretazione “fisica”: misura del *disordine*
- In **Teoria dell’Informazione** è una misura dell’impurità di una collezione arbitraria di oggetti (esempi nel nostro caso)
- Data una collezione D, contenente esempi positivi e negativi (ovvero gli esempi di D sono classificati in modo booleano):

$$H(D) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

  - Dove p_+ è la frazione di esempi positivi e p_- la frazione di esempi negativi in D

Entropia per classificazioni booleane



- H(D)
- Notare che: $p_+ + p_- = 1$,

  ovvero $p_- = 1 - p_+$

$$0 \le H(D) \le 1$$

- Se $p_+ = 0$ e $p_- = 1$:

$$H(D) = -0\log_2 0 - 1\log_2 1 = 0$$

- Se $p_+ = p_- = ½$:

$$H(D) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} =$$

$$-2\frac{1}{2}\log_2\frac{1}{2} = -\log_2\frac{1}{2} = \log_2 2 = 1$$

Esempio
per classificazioni booleane

- $D = D^+ \cup D^-$, dove:

  $D^+ = \{ x_1, x_2, x_4, x_5 \}$

  $D^- = \{ x_3 \}$

$$H(D) = -\frac{4}{5}\log_2 \frac{4}{5} - \frac{1}{5}\log_2 \frac{1}{5} =$$

$$-0{,}8\log_2 0{,}8 - 0{,}2\log_2 0{,}2 = 0{,}72$$

Entropia per classificazioni con n classi

- Avendo n classi O = { c_1, c_2, …, c_n }, definiamo pi come la frazione di elementi nell'insieme D classificati con la classe c_i
- L'entropia di D è:

$$H(D) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

- H(D) viene definito come il **bisogno informativo**, o numero di bit necessari per codificare la classificazione di un arbitrario elemento x di X (ecco perché $\log_2$)

Stima dell'entropia di una classificazione

- D è l'insieme di esempi di addestramento
- Nota: H(D) è una stima dell'entropia della classificazione "reale" C che vogliamo apprendere
- Posso stimare la probabilità di una classe c_i su D (p "cappuccio" è la stima di p):

$$\hat{p}_i = \frac{|D_{c_i}|}{|D|}$$

- La stima di H(C) è data da:

$$\hat{H}(C) = H(D) = -\sum_{i=1}^{n} \frac{|D_{c_i}|}{|D|} \log_2 \frac{|D_{c_i}|}{|D|}$$

Esempio
per classificazioni con n classi

- Classi O = { c_1, c_2, c_3, c_4 }
- D = $D_1 \cup D_2 \cup D_3 \cup D_4$, dove:

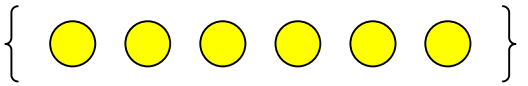
  D_1 = { x_1, x_2, x_4, x_5 }, D_2 = { x_3 },

  D_3 = { x_6 }, D_4 = { x_7, x_8 }

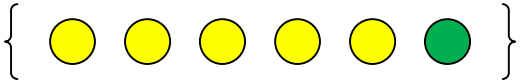
$$H(D) = -\frac{4}{8}\log_2\frac{4}{8} - \frac{1}{8}\log_2\frac{1}{8}$$

$$-\frac{1}{8}\log_2\frac{1}{8} - \frac{2}{8}\log_2\frac{2}{8} = 1{,}75$$

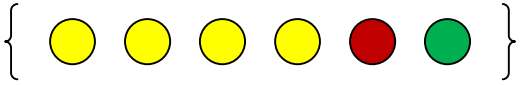
Esempi
per classificazioni con n classi



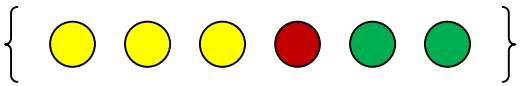
$$H(D) = -\frac{6}{6}\log_2\frac{6}{6} - 0\log_2 0 - 0\log_2 0 = 0$$



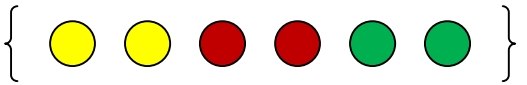
$$H(D) = -\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6} - 0\log_2 0 = 0{,}65$$



$$H(D) = -\frac{4}{6}\log_2\frac{4}{6} - \frac{1}{6}\log_2\frac{1}{6} - \frac{1}{6}\log_2\frac{1}{6} = 1{,}25$$



$$H(D) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{1}{6}\log_2\frac{1}{6} - \frac{2}{6}\log_2\frac{2}{6} = 1{,}459$$



$$H(D) = -\frac{2}{6}\log_2\frac{2}{6} - \frac{2}{6}\log_2\frac{2}{6} - \frac{2}{6}\log_2\frac{2}{6} = 1{,}58$$

Misure di impurità di una classificazione

- L'entropia non è l'unica misura di impurità di una classificazione
- Nel caso della classificazione booleana, una misura $\varphi(p, 1\text{-}p)$ di impurità deve avere le seguenti proprietà:
  - $\varphi\left(\frac{1}{2},\frac{1}{2}\right) \geq \varphi(p,1-p)$
  - $\varphi(0,1) = \varphi(1,0) = 0$
  - $\varphi(p,1-p)$ è crescente in p su [0, ½] e decrescente su [½, 1]

Misure di impurità di una classificazione

- Entropia

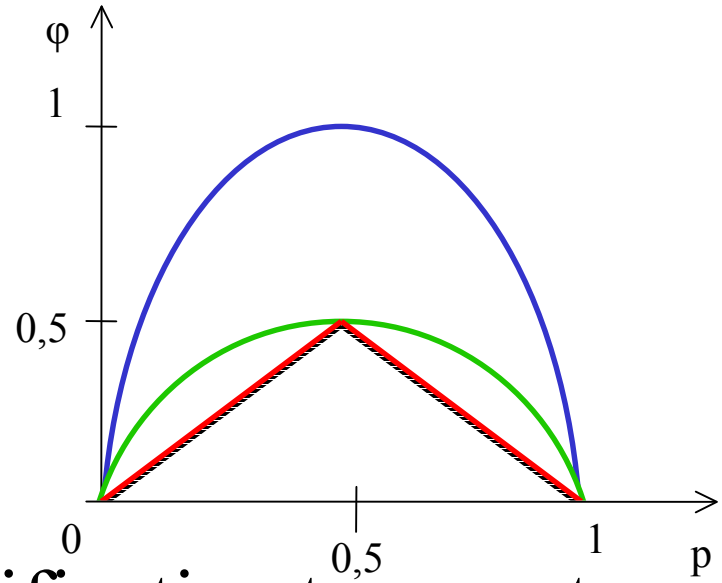
$$\varphi(p,1-p) = -p\log_2 p - (1-p)\log_2(1-p)$$

- Gini index

$$\varphi(p,1-p) = 2p(1-p)$$

- Misclassification error

$$\varphi(p,1-p) = 1 - \max(p,1-p)$$



- Non ci sono differenze significative tra queste misure nel determinare l'impurità di una classificazione

Scelta dell'attributo "migliore"

- Il **guadagno informativo** Gain(D, a) misura la **riduzione di entropia** ottenuta ripartendo gli esempi D secondo i valori dell'attributo a, cioè la riduzione del "bisogno informativo" che si otterrebbe conoscendo i valori di a:

$$Gain(D,a) = H(D) - \sum_{v \in Val(a)} \frac{|D_v|}{|D|} H(D_v)$$

- **L'attributo migliore** *a*, dato un insieme D di esempi classificati e una lista A di attributi, è quello che massimizza il guadagno informativo

Esempio

X è l'insieme degli studenti rappresentati mediante gli attributi: (media, età, studia, sesso). Dato x, c(x) = promosso?

D^+ = (1=(A,D,**si**,**F**), 2=(B,D,**si**,**M**), 3=(A,E,**no**,F), 4=(C,E,**si**,**M**))

D^- = (5=(C,E,**no**,**M**), 6=(C,E,**no**,**F**))

$H(D)=-4/6\log(4/6)-2/6\log(2/6)=0,92$

$D_{sesso=F}$={ 1+,3+,6- } $D_{sesso=M}$={ 2+,4+,5- }

$H(D_{sesso=F})=-2/3\log 2/3-1/3\log 1/3=0,92$

$H(D_{sesso=M})=-2/3\log 2/3-1/3\log 1/3=0,92$

$p_{sesso=F}=0,5$, $p_{sesso=M}=0,5$

$Gain(sesso)=0,92-0,5\times0,92-0,5\times0,92=0$!!!

$D_{studia=si}$={ 1+,2+,4+ }, $D_{studia=no}$={ 3+,5-,6- }

$H(D_{studia=si})=-3/3\log 3/3=0$

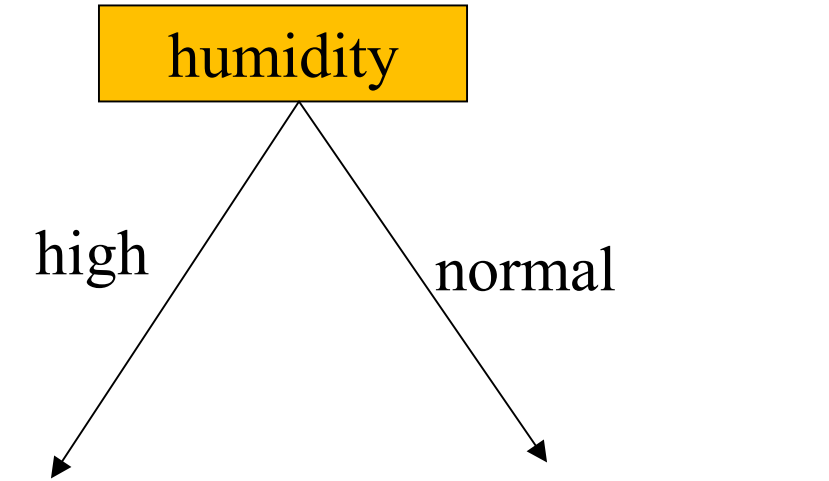
$H(D_{studia=no})=-1/3\log 1/3-2/3\log 2/3=0,92$

$p_{studia=si}=3/6$, $p_{studia=no}=3/6$

$Gain(studia)=0,92-0,5\times0-0,5\times0,92=0,46$

Esempio 2

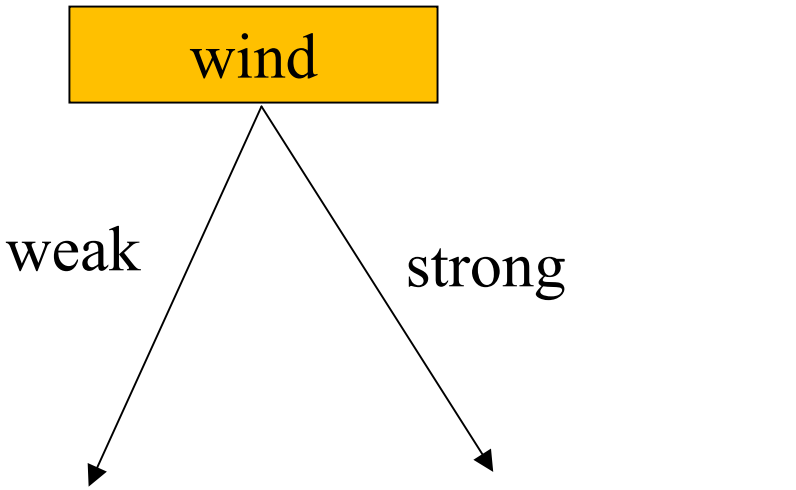
- D contiene 14 esempi così ripartiti: [9+,5-] => H(D)=0,940
- Due attributi: *humidity* = {high,normal}, *wind* = {weak,strong}
- Quale preferire?



D_{high} : [3+,4-] D_{norm} : [6+,1-]

H_{high}=0,985 H_{norm}=0,592

Gain(humidity)=0,940-(7/14)0,958 -(7/14)0,592=**0,151**



D_{weak} : [6+,2-] D_{strog} : [3+,3-]

H_{high}=0,811 H_{norm}=1,00

Gain(humidity)=0,940-(8/14)0,811 -(6/14)1,0=**0,048**

Misure alternative per selezionare l'attributo "migliore"

- **Problema:** Il guadagno informativo **predilige attributi con molti valori**
- Se aggiungessimo un attributo Data, che ha un numero elevatissimo di valori possibili (es. 11 ottobre 2007), predirebbe perfettamente gli esempi in D
  - Albero a profondità 1, ma non generalizza!
- **Soluzione:** penalizzare tali attributi mediante l'informazione di split

Split Information e Gain Ratio

- Misura sensibile a quanto ampiamente e uniformemente l'attributo separa (split) i dati

$$SplitInformation(D,a) = -\sum_{v \in Val(a)} \frac{|D_v|}{|D|} \log_2 \frac{|D_v|}{|D|}$$

  - Non è altro che l'entropia di D rispetto ai valori dell'attributo *a*
- Misura di scelta dell'attributo "migliore":

$$GainRatio(D,a) = \frac{Gain(D,a)}{SplitInformation(D,a)}$$

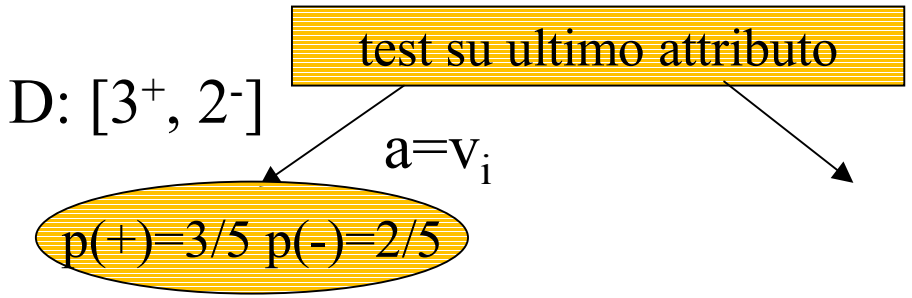
Problemi nell'apprendimento da esempi

- Dati rumorosi
- Sovradattamento
- Gestione dei valori di attributi mancanti

Come si comporta ID3???

Problema del rumore negli AD

- **Problema**:
  - se i dati sono rumorosi, posso esaurire tutti gli attributi senza ottenere delle ripartizioni perfette dei D_i in D^+ (SI) o D^- (NO). Quindi non posso emettere delle decisioni "perfette"
- **Soluzioni**:
  - associare a ciascuna foglia la classificazione della maggioranza degli esempi (vedi condizione dell'algoritmo ID3: if A=∅ then associa classificazione di maggioranza in D)
  - associare a ciascuna foglia la probabilità stimata di correttezza, in base alle frequenze relative (agente probabilistico basato sulla teoria delle decisioni)



Sovradattamento

- Ricordate il **problema**: che succede se l'algoritmo viene "sovra-addestrato"?
- Per aderire al meglio agli esempi, tende a generare un apprendista con ridotte capacità di generalizzazione, ovvero, un algoritmo che si comporta bene sugli esempi di D, ma peggio su esempi non visti durante l'apprendimento
- Come si misura il "comportamento" di un apprendista rispetto a questo problema?

Misura della prestazione di un apprendista

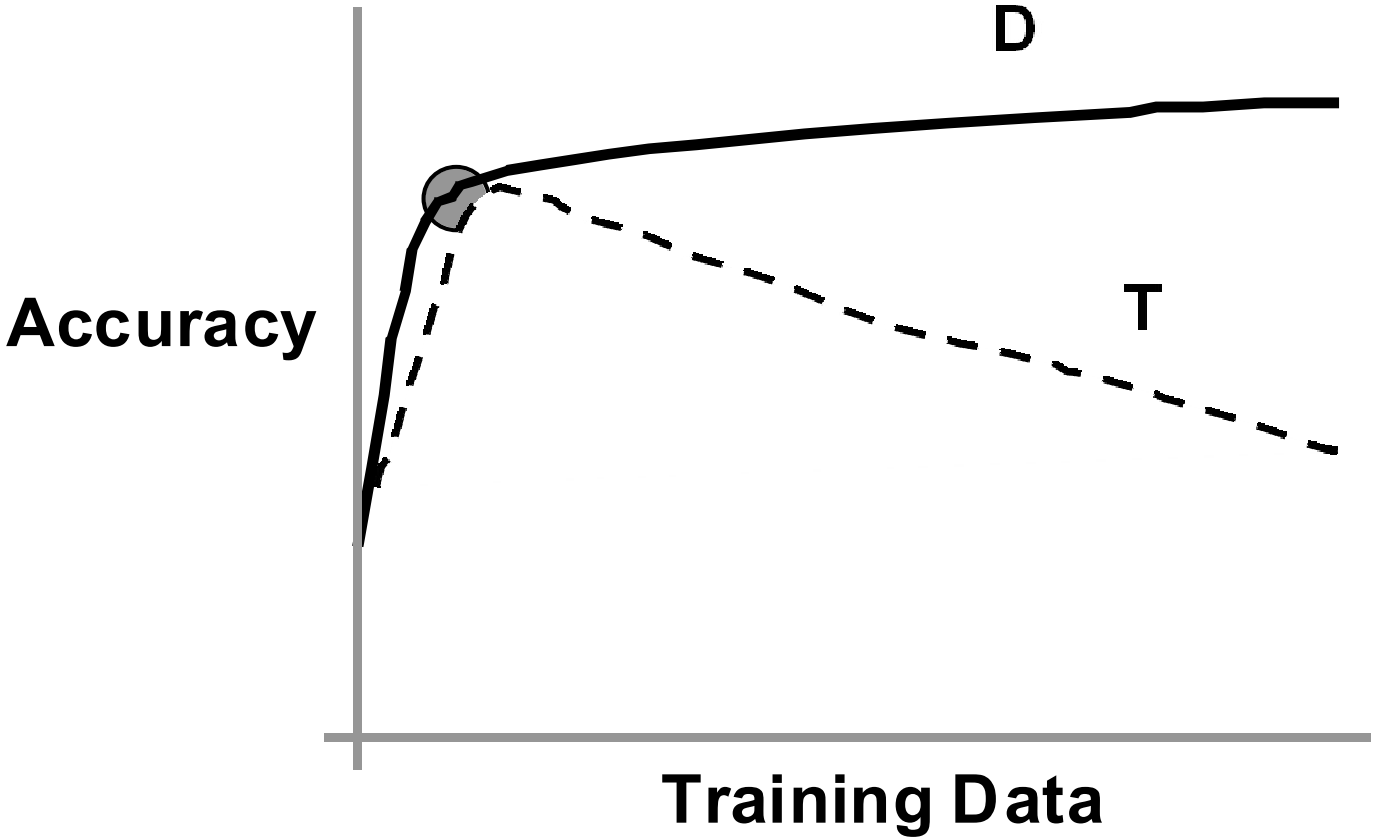
- Sia T un insieme di N esempi di cui è nota la classificazione (**test set**)
- Sia L un **apprendista** o **learner** (ad es., un albero di decisione)
- Sia n^{tp} il numero di esempi **positivi** che L classifica come **positivi**, n^{tn} il numero di esempi **negativi** che L classifica come **negativi**, n^{fp} il numero di esempi **negativi** che L classifica come **positivi**, n^{fn} il numero di esempi **positivi** che L classifica come **negativi**

$$accuracy(L) = \frac{n^{tp} + n^{tn}}{n^{tp} + n^{tn} + n^{fp} + n^{fn}} = \frac{n^{tp} + n^{tn}}{N}$$

- Nota che **in generale, T ≠ D** (altrimenti si ha una sovrastima!!!)

Curve di apprendimento

T test set
D training set



Metodi per ridurre il sovradattamento (1)

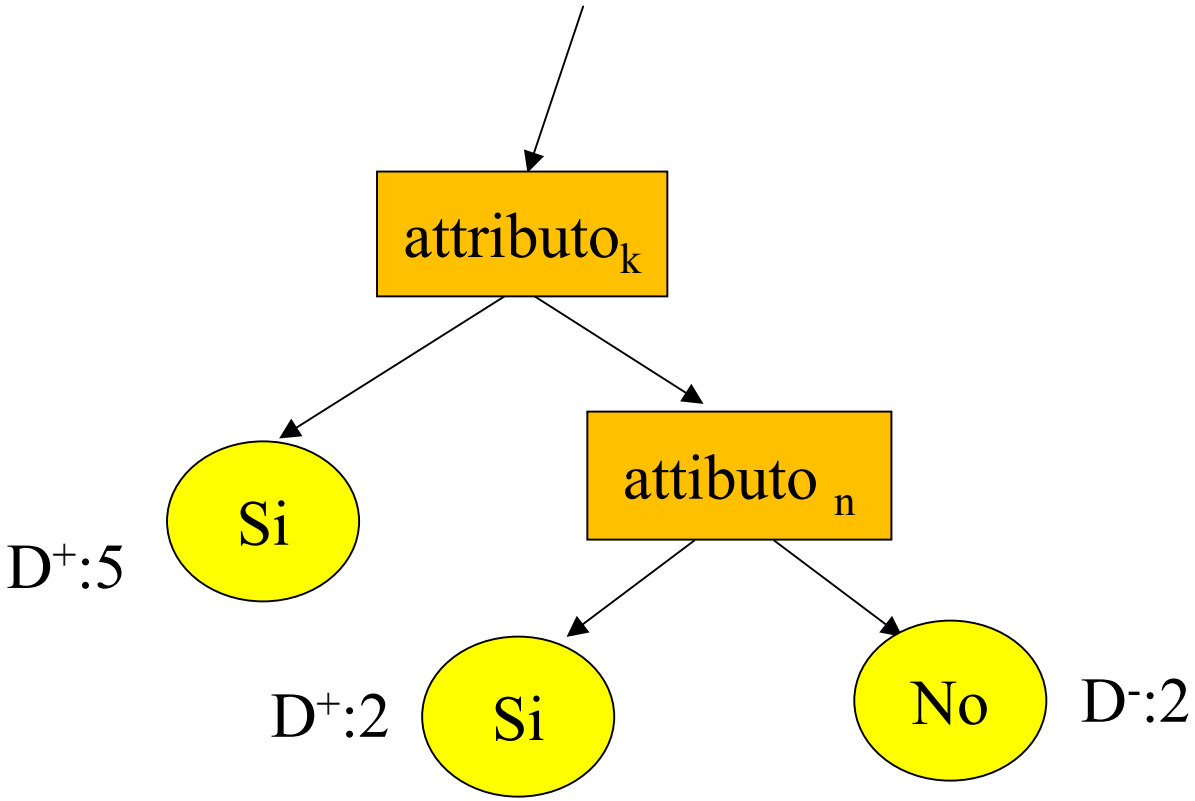
1. Reduced error pruning

- Si considera ogni nodo n_i di un albero di decisione
- Si rimuove il sottoalbero avente per radice il nodo n_i, rendendolo in tal modo una "foglia" dell'albero più generale
- Si assegna ad n_i **la classificazione più probabile del sottoinsieme di esempi affiliati al nodo**
- Si misura l'accuratezza su T dell'albero non potato e dell'albero potato
- Si effettua la potatura solo se la potatura sotto n_i non produce un peggioramento
- Si procede iterativamente considerando tutti i nodi finché non si misurano ulteriori miglioramenti.

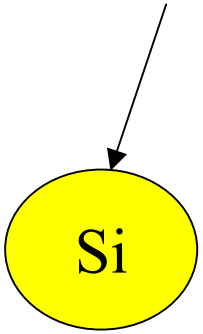
Reduced error pruning

- Questa potatura ha l'effetto di ridurre il problema delle "coincidenze" visto che difficilmente le coincidenze si verificano anche sul set T
- Questo procedimento è applicabile quando i dati a disposizione **sono molti**. Sarà dunque possibile considerarne una parte per generare l'albero, ed una parte per potarlo.

Esempio



Esempio



Confidenza: 7/9

Metodi per ridurre il sovradattamento (2)

2. Rule post-pruning

- Deriva un albero di decisione dai dati D, eventualmente consentendo un sovradattamento
- Converti l'albero in un insieme di regole. Ogni regola rappresenta un percorso dalla radice ad una foglia.
- Generalizza ogni regola, provando a rimuovere incrementalmente ogni precondizione della regola che generi un miglioramento dell'accuratezza
- Ordina le regole così ottenute per accuratezza, e utilizzale in questa sequenza quando si classificano istanze nuove.

Es.: IF (tempo=assolato)&(umidità=alta) THEN playtennis=no
Prova a rimuovere (tempo=assolato) e poi (umidità=alta)

Valori di Attributo Mancanti (1)

- Supponiamo di trovarci sul nodo n e consideriamo l'esempio:
  - D_{15} = (Sunny, ?, High, Weak, Yes)
- Come calcolare il Gain(D_n, Temperature)?
- Strategia 1: assegnare come valore per Temperature nell'esempio D_{15}
  - il valore di maggioranza per Temperature su tutto D_n
  - il valore di maggioranza per Temperature sul sottoinsieme di esempi in D_n classificati come D_{15}, ovvero D_{yes}

Valori di Attributo Mancanti (2)

- Supponiamo di trovarci sul nodo n e consideriamo l'esempio:
  - D_{15} = (Sunny, ?, High, Weak, Yes)
- Come calcolare il Gain(D_n, Temperature)?
- Strategia 2: assegnare una probabilità a ogni valore dell'attributo Temperature
  - Si stima la probabilità sulle frequenze osservate in D_n dei vari valori di Temperature
  - Utilizziamo queste probabilità per frazionare il contributo di D_{15} sui vari valori di Temperature nel calcolare il Gain

Applicazioni di alberi di decisione

- Progetto di sistemi di **separazione del petrolio dal gas**: il sistema di separazione ha una struttura che dipende da numerosi attributi quali: proporzione fra gas, petrolio e acqua, intensità del flusso, viscosità, …
  - La GASOIL ha costruito un sistema esperto con 2500 regole, generate da un albero di decisione
- **Addestratore di volo**
  - Esempi generati monitorando piloti esperti e generando esempi ogni volta che un pilota fissava una variabile di controllo (es manetta o flap).
  - 90.000 esempi estratti da 30x3 piani di volo eseguiti da 3 piloti esperti. 20 variabili di stato.
  - Utilizza il programma C4.5 (Quinlan)
- **Fraud Detection**
  - Sulla base di un campione di verifiche tributarie ciascuna registrata con un esito (positivo, negativo, ammontare dell'imposta se incassata) costruisce un albero di decisioni per decidere, sulla base della denuncia dei redditi, se effettuare o meno un controllo (KDD group all'Università di Pisa).
- Consultate SW DataMining basato su Decision Tree:
  - http://www.kdnuggets.com/software/classification.html#Decision

Per esercitarsi

- C4.5 programma freeware per apprendere alberi di decisioni (Quinlan)
- Scaricare C4.5 da:
  - http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz
- spiegazioni ulteriori su C4.5 le trovate sul tutorial
  - www.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html
- Anche sul sito WEKA (J4.8)
  - www.cs.waikato.ac.nz/ml/**weka**
- Insiemi di dati (**datasets**) scaricabili dal sito:
  - http://www.ics.uci.edu/~mlearn/MLSummary.html
- Varie decine di applicazioni, medicina, economia, classificazione di specie, architettura, scacchi, ecc…