Alberi di Decisione

Fabio Aiolli
www.math.unipd.it/~aiolli

Sito web del corso
www.math.unipd.it/~aiolli/corsi/1617/aa/aa.html

Alberi di Decisione

- In molte applicazioni del mondo reale non è sufficiente apprendere funzioni booleane con ingressi binari (vedi concept learning)
- Gli alberi di decisione sono particolarmente adatti a trattare:
  - Istanze rappresentate da coppie attributo-valore;
  - Funzioni target con valori di output discreti (anche più di 2 valori);
  - Concetti descritti da disgiunzioni di funzioni booleane;
  - Esempi di apprendimento che possono contenere errori e/o valori mancanti.
- Inoltre, algoritmi di apprendimento per alberi di decisione sono in genere molto efficienti.
- Per questi motivi gli alberi di decisione sono molto utilizzati in applicazioni pratiche.

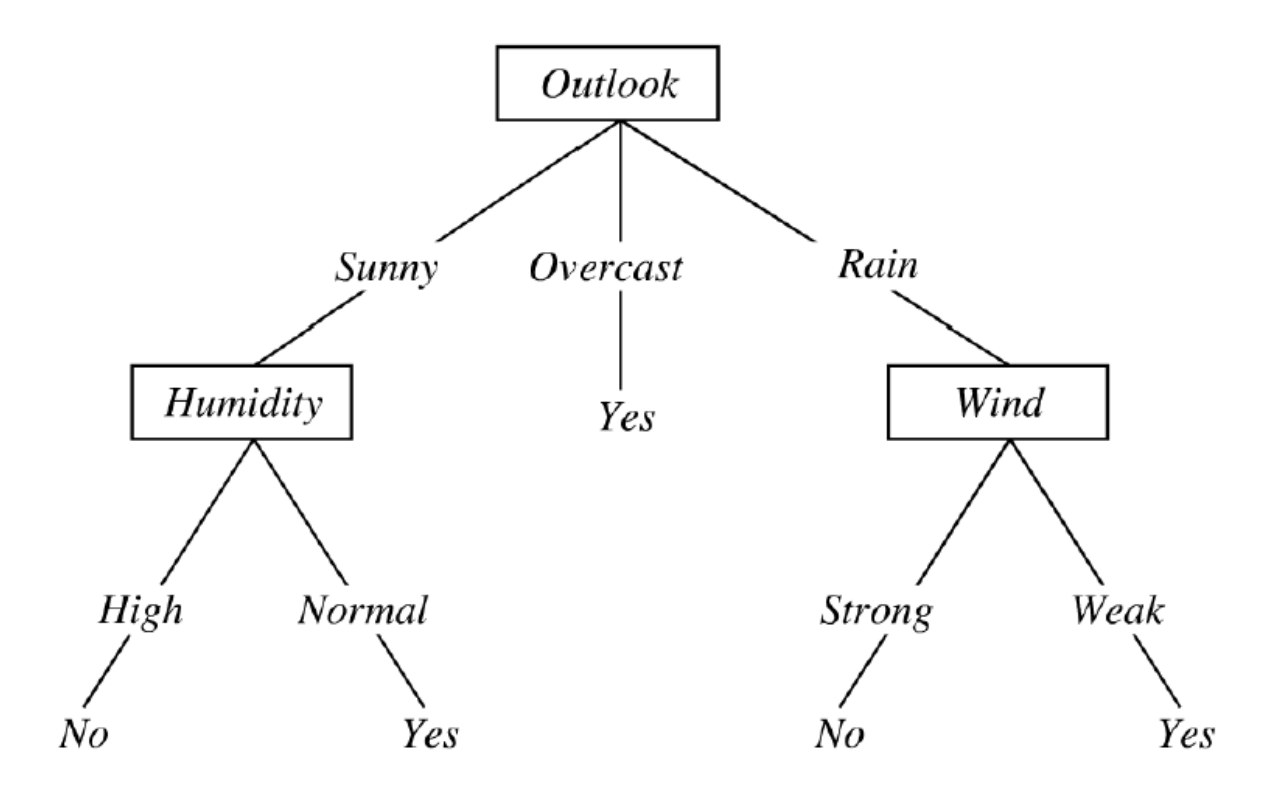
Play tennis?

- E’ la giornata ideale per giocare a Tennis?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

Play tennis?

- Come decidere se giocare a Tennis



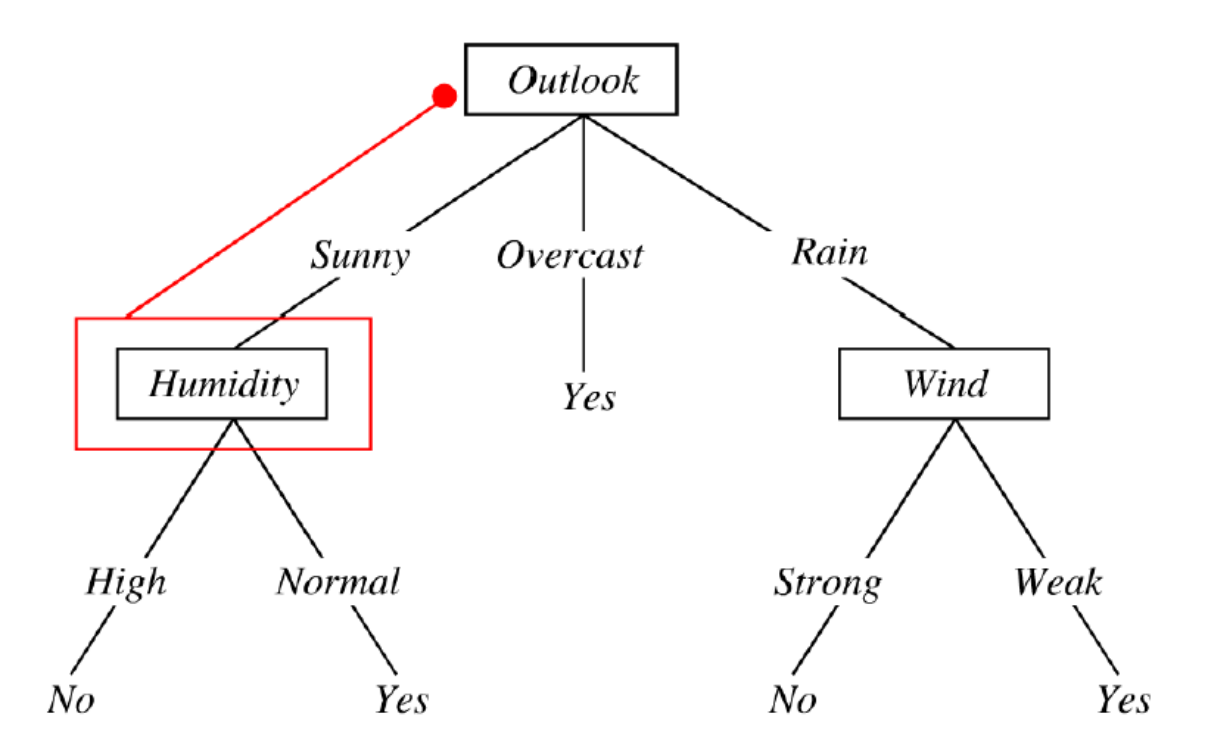
Esempio: (O=Sunny,T=Hot,H=high,W=Strong) ?

Alberi di Decisione (cont.)

- In un albero di decisione:
  - Ogni nodo interno effettua un test su un attributo;
  - Ogni ramo uscente da un nodo corrisponde ad uno dei possibili valori che l'attributo può assumere;
  - Ogni foglia assegna una classificazione.
- Per classificare una istanza con un albero di decisione bisogna:
  1. Partire dalla radice;
  2. Selezionare l'attributo della istanza associato al nodo corrente;
  3. Seguire il ramo associato al valore assegnato a tale attributo nella istanza;
  4. Se si raggiunge una foglia restituire l'etichetta associata alla foglia, altrimenti partire dal nodo corrente e ripetere dal passo 2.

Play tennis?

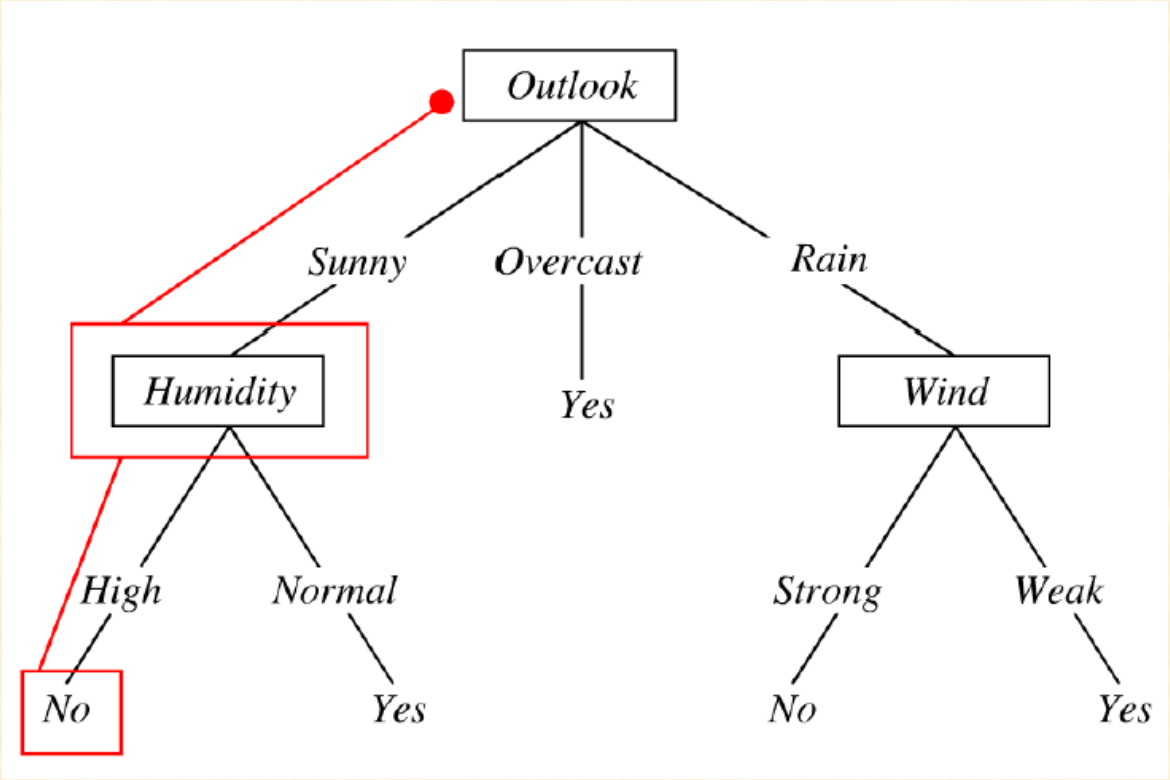
- Alla radice è associato l'attributo Outlook e quindi bisogna seguire il ramo Sunny



Esempio: (O=Sunny,T=Hot,H=high,W=Strong) ?

Play tennis?

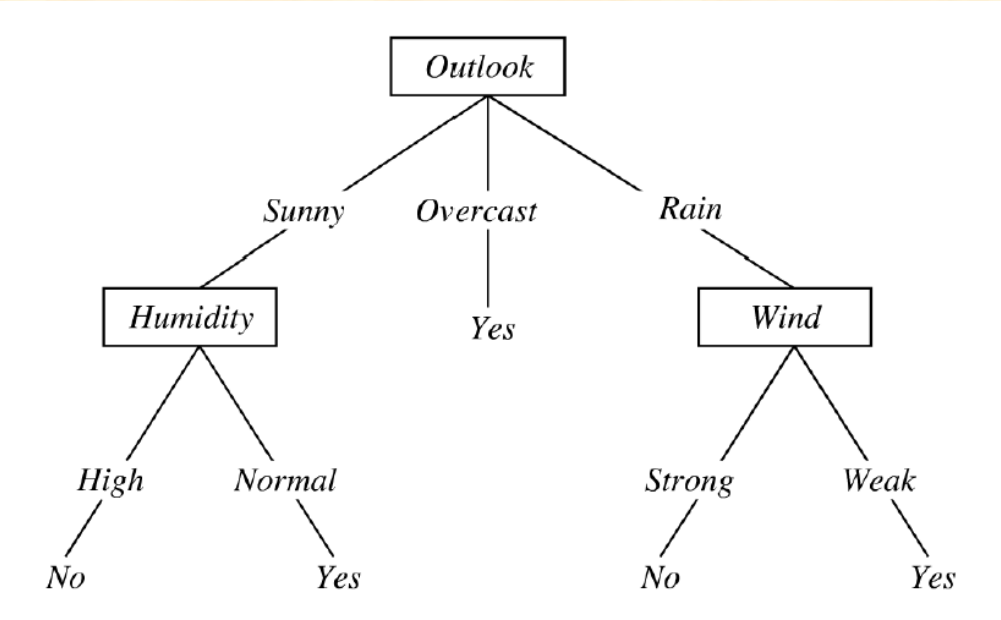
- Al nodo raggiunto è associato l'attributo Humidity e quindi bisogna seguire il ramo High



Esempio: (O=Sunny,T=Hot,H=high,W=Strong) ? NO

Alberi di decisione e funzioni booleane

- Ogni funzione booleana può essere rappresentata con alberi di decisione:
  - Ogni cammino dalla radice ad una foglia codifica una congiunzione di test su attributi;
  - Più cammini che conducono allo stesso tipo di classificazione codificano una disgiunzione di congiunzioni;



(Outlook=Sunny **and** Humidity=Normal)
or
Outlook=Overcast
or
(Outlook=Rain **and** Wind=Week)

Esempio di algoritmo di apprendimento ID3

L'apprendimento di alberi di decisione tipicamente procede attraverso una procedura di tipo divide et impera che costruisce l'albero top-down:

ID3(Tr,A)

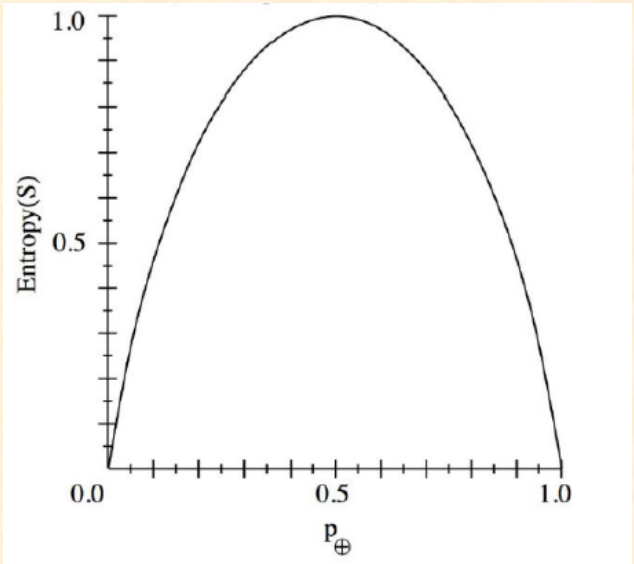
- Crea il nodo radice, T←Tr e inserisci tutti gli attributi nell'insieme A;
- Se gli esempi in T sono tutti della stessa classe (+o-), ritorna l'albero con un solo nodo e etichetta la classe;
- Se A è vuoto, ritorna l'albero con un solo nodo e etichetta la classe di maggioranza in T
- Altrimenti:
  - Scegli a∈A, l'attributo *migliore* in A;
  - Partiziona T secondo i possibili valori che a può assumere: $T_{a=val1}$,…, $T_{a=valm}$ dove valm = numero valori distinti possibili di a;
  - Per ogni $T_{a=valj}$, *se* è vuoto crea una foglia figlio con l'etichetta della classe più frequente in T, *altrimenti* crea sottoalbero richiamando ID3($T_{a=valj}$,A-{a})
  - Ritorna T

ID3: selezione attributo ottimo

I vari algoritmi di apprendimento di alberi di decisione si differenziano soprattutto (ma non solo) dal modo in cui si seleziona l'attributo ottimo: ID3 utilizza I concetti di Entropia e Guadagno Entropico

$$E(S) = -p_- \log_2(p_-) - p_+ \log_2(p_+)$$

dove p_+ e p_- sono la proporzione di esempi positivi e negativi, rispettivamente, nell'insieme S



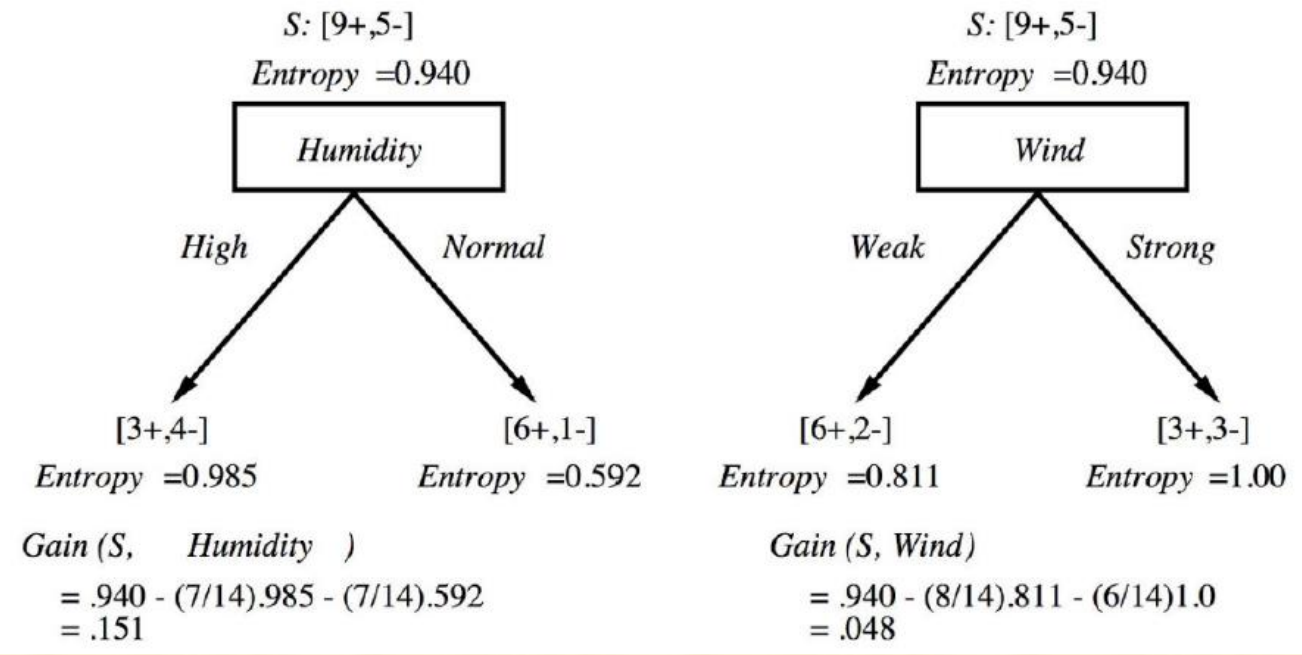
L'entropia misura il "grado di impurità" dell'insieme degli esempi

ID3: selezione attributo ottimo

Si seleziona l'attributo che massimizza il Guadagno Entropico:

$$\mathrm{G}(S,a) = E(S) - \sum_{v \in V(a)} \frac{|S_{a=v}|}{|S|} E(S_{a=v})$$

Il guadagno misura la riduzione attesa di entropia nel partizionare i dati usando l'attributo a.



Selezione attributo ottimo, problema.

Problema. Il Guadagno Entropico favorisce troppo attributi che possono assumere tanti valori diversi.

Esempio. Se al problema di decidere quando giocare a tennis si aggiunge un attributo che consiste nella data del giorno considerato (es. Data="11 Novembre"), allora l'attributo Data è quello con guadagno massimo (ogni sottoinsieme costituirà un sottoinsieme diverso e puro, quindi con entropia nulla), anche se in realtà non è significativo.

Selezione attributo ottimo: Gain Ratio

Per rimediare a questo problema si definisce il Gain Ratio:

$$GR(S,a) = \frac{G(S,a)}{SI(S,a)}$$

dove $SI(S,a) = -\sum_{v \in V(a)} \frac{|S_{a=v}|}{|S|} \log_2 \left(\frac{|S_{a=v}|}{|S|} \right)$

La *SplitInformation* (SI) misura quanti, e quanto uniformi sono, i sottoinsiemi generati dall'attributo a, a partire dall'insieme S.

SplitInformation corrisponde alla entropia di S dati i valori di a.

Si osservi che il termine SI sfavorisce attributi che suddividono S in molti sottoinsiemi tutti della stessa cardinalità.

GainRatio NON risolve tutti i problemi!

Criteri alternativi all'entropia

Variance Impurity:

$$p_- \cdot p_+$$

(Weighted) Gini Impurity (generalizzazione di Variance Impurity a più classi):

$$\sum_{c,c' \in C, c \neq c'} \lambda_{c,c'} p_c \cdot p_{c'}$$

Misclassification Impurity:

$$1 - max_{c \in C} p_c$$