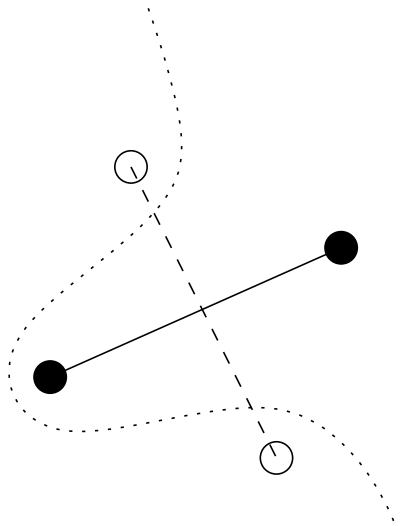
VC-dimension: Esempio

Quale è la VC-dimension di $\mathcal{H}_1$?

Quindi $VC(\mathcal{H}) \geq 3$. Cosa succede con 4 punti ? Non si riesce a frammentare 4 punti!!

Infatti esisteranno sempre due coppie di punti che se unite con un segmento provocano una intersezione fra i due segmenti e quindi, ponendo ogni coppia di punti in classi diverse, per separarli non basta una retta, ma occorre una curva. Quindi $VC(\mathcal{H}) = 3$



PAC learning

E' stato dimostrato che se si assume:

- $c \in \mathcal{H}$
- L consistente, cioè restituisce una ipotesi consistente con l'insieme di apprendimento (**Find-S** è un algoritmo consistente)

allora con probabilità almeno $(1 - \delta)$, l'algoritmo di apprendimento L restituisce un'ipotesi $h \in \mathcal{H}$ tale che $error_{\mathcal{D}}(h) \leq \epsilon$ a patto che il numero di esempi di apprendimento n soddisfi la seguente disuguaglianza:

$$n \geq \frac{1}{\epsilon}(4 \log_2(\frac{1}{\delta}) + 8VC(\mathcal{H}) \log_2(\frac{13}{\epsilon}))$$

Notare che vale $VC(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$

VC-dimension

Dimostriamo che $VC(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$

- Per ogni S tale che $\mathcal{H}$ frammenta S si ha $|\mathcal{H}| \geq 2^{|S|}$, infatti $\mathcal{H}$ può realizzare tutte le possibili dicotomie di S, che sono esattamente $2^{|S|}$.
- Scegliendo un S per cui vale $|S| = VC(\mathcal{H})$ si ottiene $|\mathcal{H}| \geq 2^{VC(\mathcal{H})}$

Quindi, applicando $\log_2$ ad entrambi i membri dell'ultima disuguaglianza, possiamo concludere che $\log_2(|\mathcal{H}|) \geq VC(\mathcal{H})$

Compiti di Classificazione più Complessi

In molte applicazioni del mondo reale non è sufficiente apprendere funzioni booleane con ingressi binari.

Gli Alberi di Decisione sono particolarmente adatti a trattare:

- istanze rappresentate da coppie attributo-valore;
- funzioni target con valori di output discreti (in generale più di 2 valori);
- esempi di apprendimento che possono contenere errori e/o avere valori mancanti.

Inoltre, algoritmi di apprendimento per Alberi di Decisione sono in genere molto veloci.

Per questi motivi gli Alberi di Decisione sono molto utilizzati in applicazioni pratiche.

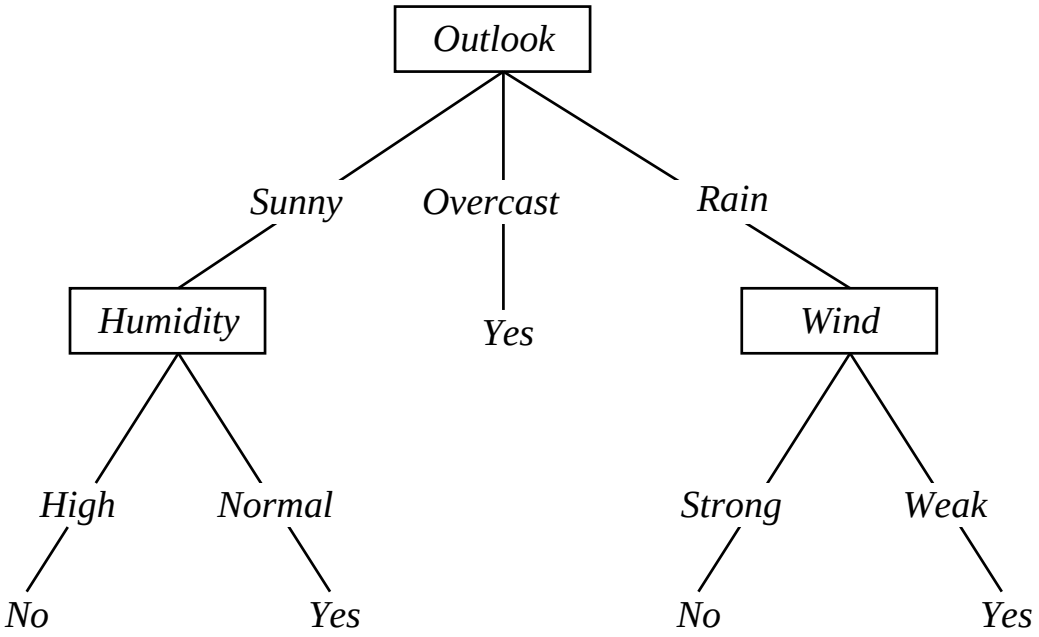
Giocare a Tennis!!

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

Alberi di Decisione

Decidere se è la giornata ideale per giocare a Tennis !!



Es. ingresso: (Outlook= *Sunny* , Temperature=*Hot*, Humidity=*High*, Wind=*Strong*).

attributo *valore attributo*

Alberi di Decisione (cont.)

In un Albero di decisione:

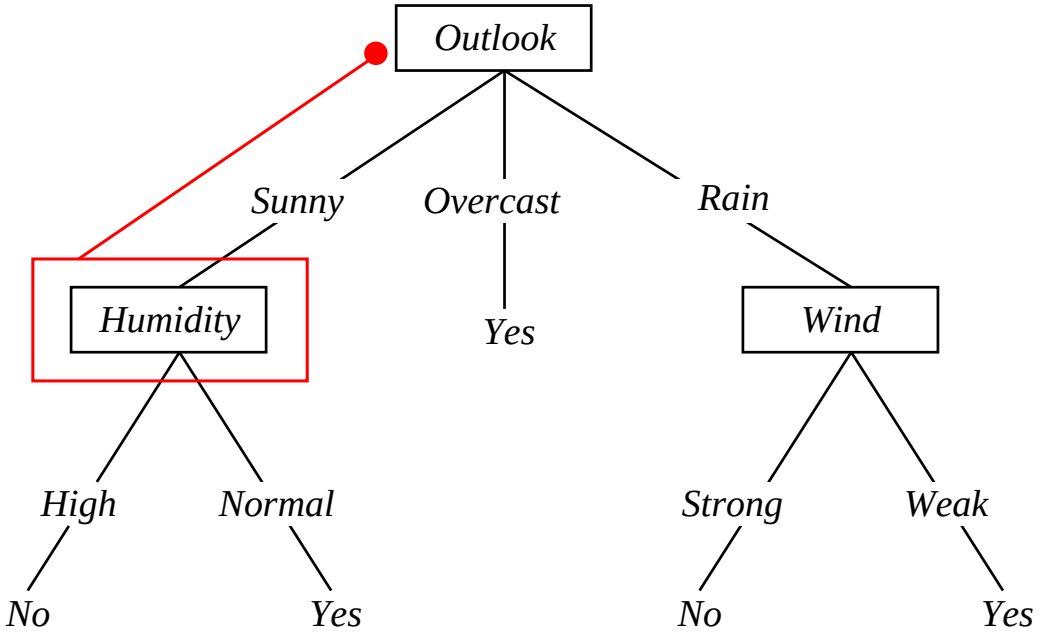
- Ogni nodo interno effettua un test su un attributo;
- Ogni ramo uscente da un nodo corrisponde ad uno dei possibili valori che l'attributo può assumere;
- Ogni foglia assegna una classificazione.

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 a partire dal nodo corrente ripetere dal passo 2.

Alberi di Decisione: Classificazione

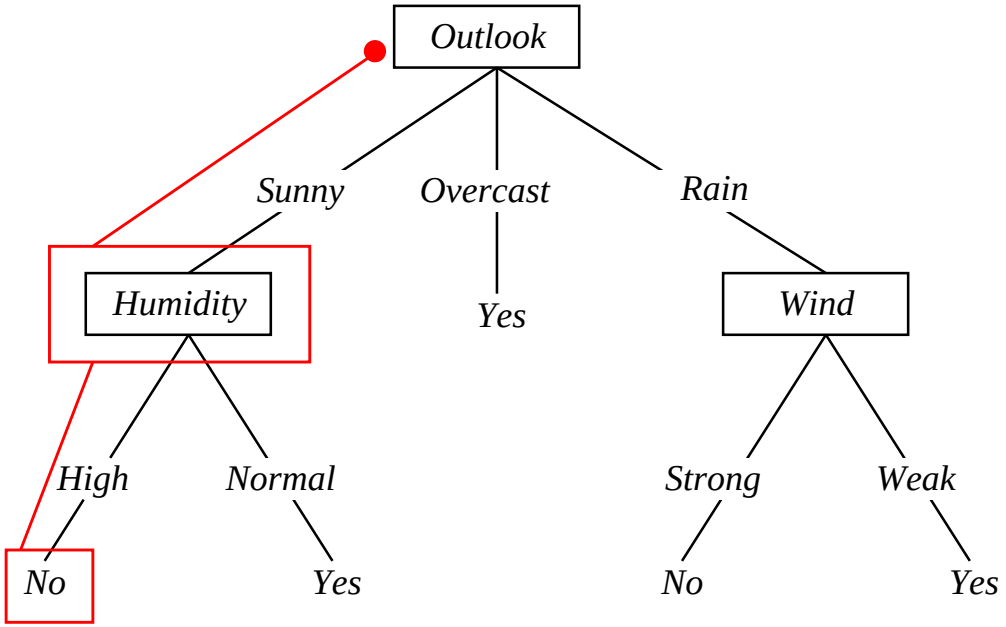
Alla radice è associato Outlook e quindi bisogna seguire il ramo *Sunny*



Es. ingresso: (Outlook=*Sunny*, Temperature=*Hot*, Humidity=*High*, Wind=*Strong*).

Alberi di Decisione: Classificazione

Al nodo raggiunto è associato Humidity e quindi bisogna seguire il ramo *High*



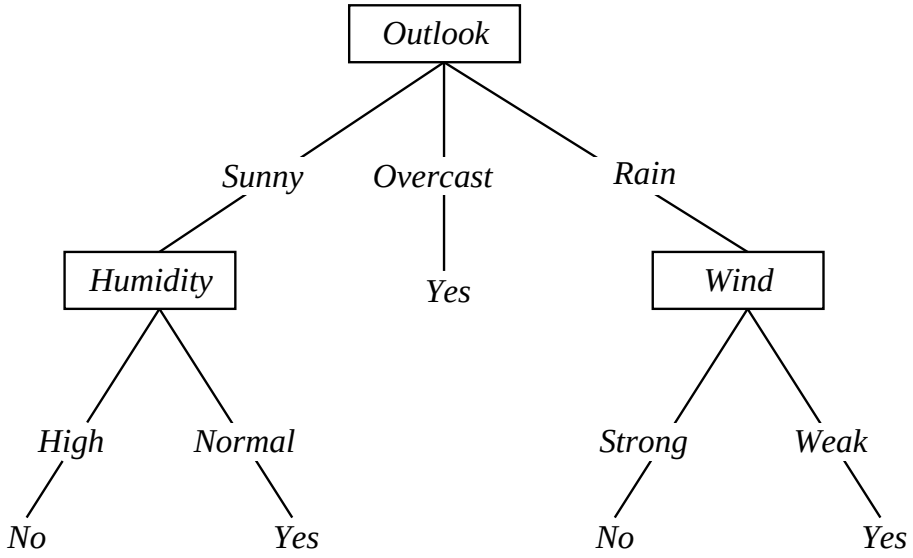
Es. ingresso: (Outlook=*Sunny*, Temperature=*Hot*, Humidity=*High*, Wind=*Strong*).

Si raggiunge una foglia: quindi l'istanza è classificata *No*

Alberi di Decisione e Funzioni Booleane

Con Alberi di Decisione ogni funzione booleana può essere rappresentata:

- 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=*Weak*)

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:

1. crea il nodo radice, $\hat{Tr} \leftarrow Tr$ e inserisci tutti gli attributi nell'insieme $\mathcal{A}$;
2. se gli esempi in $\hat{Tr}$ sono tutti della stessa classe (- o +), assegna al nodo l'etichetta della classe e fermati;
   altrimenti
   (a) se $\mathcal{A}$ è vuoto, assegna al nodo l'etichetta della classe più frequente e fermati;
   altrimenti assegna al nodo $A \leftarrow$ l'attributo "ottimo" in $\mathcal{A}$;
3. partiziona $\hat{Tr}$ secondo i possibili valori che A può assumere:
   $\hat{Tr}_{A=val_1}, \ldots, \hat{Tr}_{A=val_{m(A)}}$, dove $m(A) =$ numero valori distinti di A;
4. $\forall\, \hat{Tr}_{A=val_j} = \emptyset$ crea una foglia figlio con l'etichetta della classe più frequente in $\hat{Tr}$;
5. $\forall\, \hat{Tr}_{A=val_i} \neq \emptyset$ crea nodo figlio e vai a 2 con $\hat{Tr} \leftarrow \hat{Tr}_{A=val_i}$ e $\mathcal{A} \leftarrow \mathcal{A} \backslash A$.

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:

1. crea il nodo radice, $\hat{Tr} \leftarrow Tr$ e inserisci tutti gli attributi nell'insieme $\mathcal{A}$;
2. se gli esempi in $\hat{Tr}$ sono tutti della stessa classe (- o +), assegna al nodo l'etichetta della classe e fermati;
   altrimenti
   (a) se $\mathcal{A}$ è vuoto, assegna al nodo l'etichetta della classe più frequente e fermati;
   altrimenti assegna al nodo $A \leftarrow$ l'attributo "ottimo" in $\mathcal{A}$;
3. partiziona $\hat{Tr}$ secondo i possibili valori che A può assumere:
   $\hat{Tr}_{A=val_1}, \ldots, \hat{Tr}_{A=val_{m(A)}}$, dove $m(A) =$ numero valori distinti di A;
4. $\forall\, \hat{Tr}_{A=val_j} = \emptyset$ crea una foglia figlio con l'etichetta della classe più frequente in $\hat{Tr}$;
5. $\forall\, \hat{Tr}_{A=val_i} \neq \emptyset$ crea nodo figlio e vai a 2 con $\hat{Tr} \leftarrow \hat{Tr}_{A=val_i}$ e $\mathcal{A} \leftarrow \mathcal{A} \backslash A$.

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:

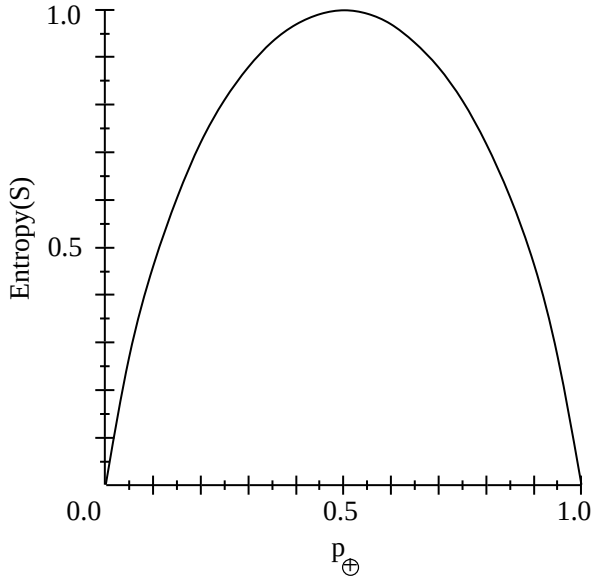
1. crea il nodo radice, $\hat{Tr} \leftarrow Tr$ e inserisci tutti gli attributi nell'insieme $\mathcal{A}$;
2. se gli esempi in $\hat{Tr}$ sono tutti della stessa classe (- o +), assegna al nodo l'etichetta della classe e fermati;
   altrimenti
   (a) se $\mathcal{A}$ è vuoto, assegna al nodo l'etichetta della classe più frequente e fermati;
   altrimenti assegna al nodo $A \leftarrow$ l'attributo "ottimo" in $\mathcal{A}$;
3. partiziona $\hat{Tr}$ secondo i possibili valori che A può assumere:
   $\hat{Tr}_{A=val_1}, \ldots, \hat{Tr}_{A=val_{m(A)}}$, dove $m(A) =$ numero valori distinti di A;
4. $\forall \hat{Tr}_{A=val_j} = \emptyset$ crea una foglia figlio con l'etichetta della classe più frequente in $\hat{Tr}$;
5. $\forall \hat{Tr}_{A=val_i} \neq \emptyset$ crea nodo figlio e vai a 2 con.. se necessario risalire ai nodi avi ⇑

ID3: Selezione Attributo Ottimo

Vari algoritmi di apprendimento si differenziano soprattutto (ma non solo) dal modo in cui si seleziona l'attributo ottimo: ID3: utilizza il concetto di *Entropia* e *Guadagno Entropico*

$$Entropia(S) = -p_- log_2(p_-) - p_+ log_2(p_+)$$

dove p_- (p_+) è la proporzione di esempi negativi (positivi) nell'insieme S



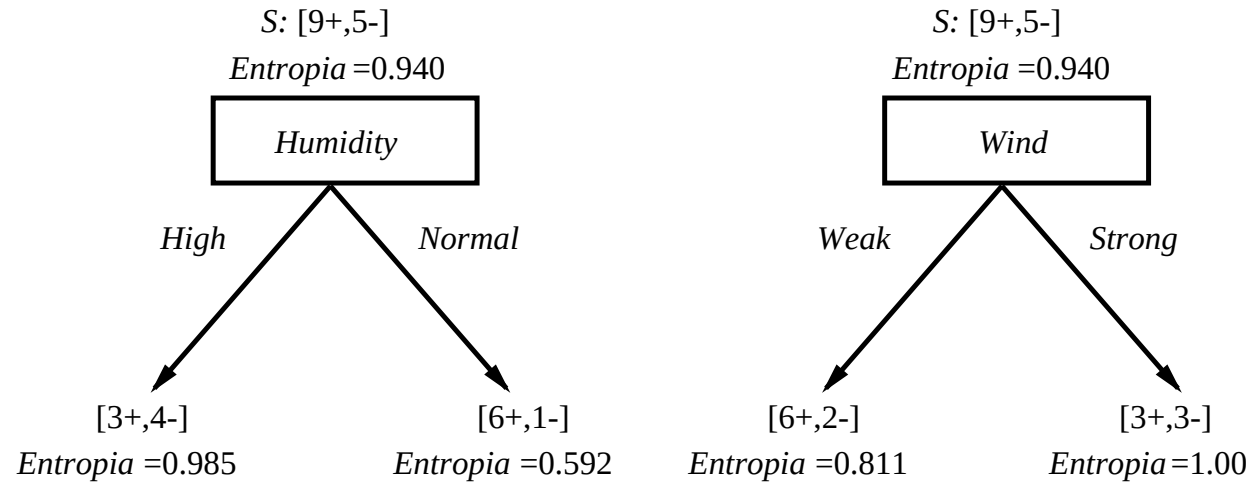
L' *Entropia* misura il "grado di purezza" dell'insieme degli esempi!

ID3: Selezione Attributo Ottimo

Si sceglie l'attributo A che massimizza il *Guadagno Entropico*:

$$Guadagno(S, A) = Entropia(S) - \sum_{v \in Valori(A)} \frac{|S_{A=v}|}{|S|} Entropia(S_{A=v})$$

Il *Guadagno* misura la riduzione aspettata della entropia nel partizionare i dati usando A



Guadagno (S, Humidity)
= .940 - (7/14).985 - (7/14).592
= .151

Guadagno (S, Wind)
= .940 - (8/14).811 - (6/14)1.0
= .048