Alberi di Decisione

Corso di AA, anno 2018/19, Padova

Fabio Aiolli

31 Ottobre 2018

Alberi di decisione (Decision Trees)

- In molte applicazioni del mondo reale non è sufficiente apprendere funzioni booleane che sono semplici congiunzioni di letterali esenti da rumore come nel caso dell'apprendimento di concetti.
- Gli alberi di decisione sono una ottima alternativa, permettono di apprendere funzioni/regole di decisione rappresentabili con alberi.
- Gli alberi di decisione possono essere facilmente tradotti in una serie di regole del tipo `if-then` rendendo questa rappresentazione delle ipotesi facilmente comprensibile per l'umano.
- Questa ultima caratteristica (non presente in molte delle tecniche che vedremo in seguito) li rende particolarmente interessanti per applicazioni di tipo medico, biologico, finanziario ecc. dove risulti necessario per l'utente esperto poter interpretare il risultato di un algoritmo di apprendimento.

Alberi di decisione: definizione

In un albero di decisione:

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

La classificazione di una istanza avviene nel modo seguente:

1. Partiamo dalla radice;
2. Selezioniamo l'attributo associato al nodo corrente;
3. Seguiamo il ramo associato al valore di quell'attributo nella istanza;
4. Se abbiamo raggiunto una foglia restituiamo l'etichetta associata alla foglia, altrimenti ripetiamo dal punto 2 partendo dal nodo corrente.

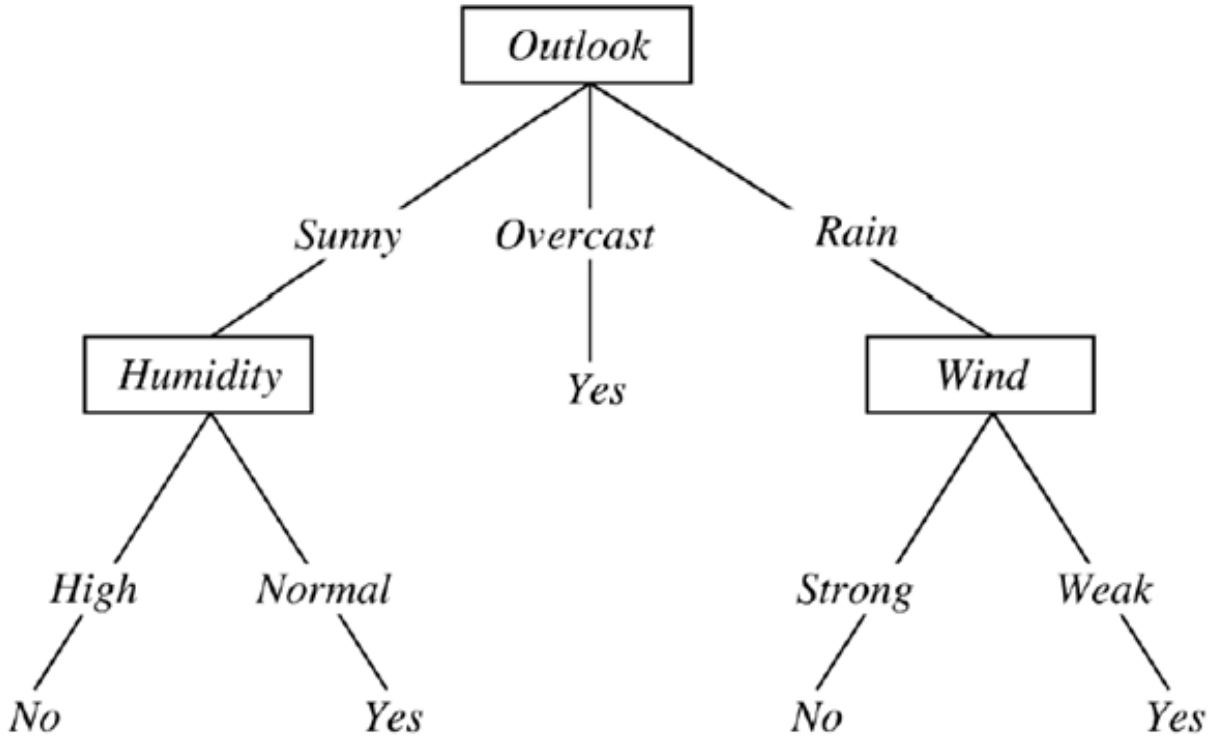
Giochiamo a tennis?

È una giornata adatta per una partita di 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

Giochiamo a tennis?

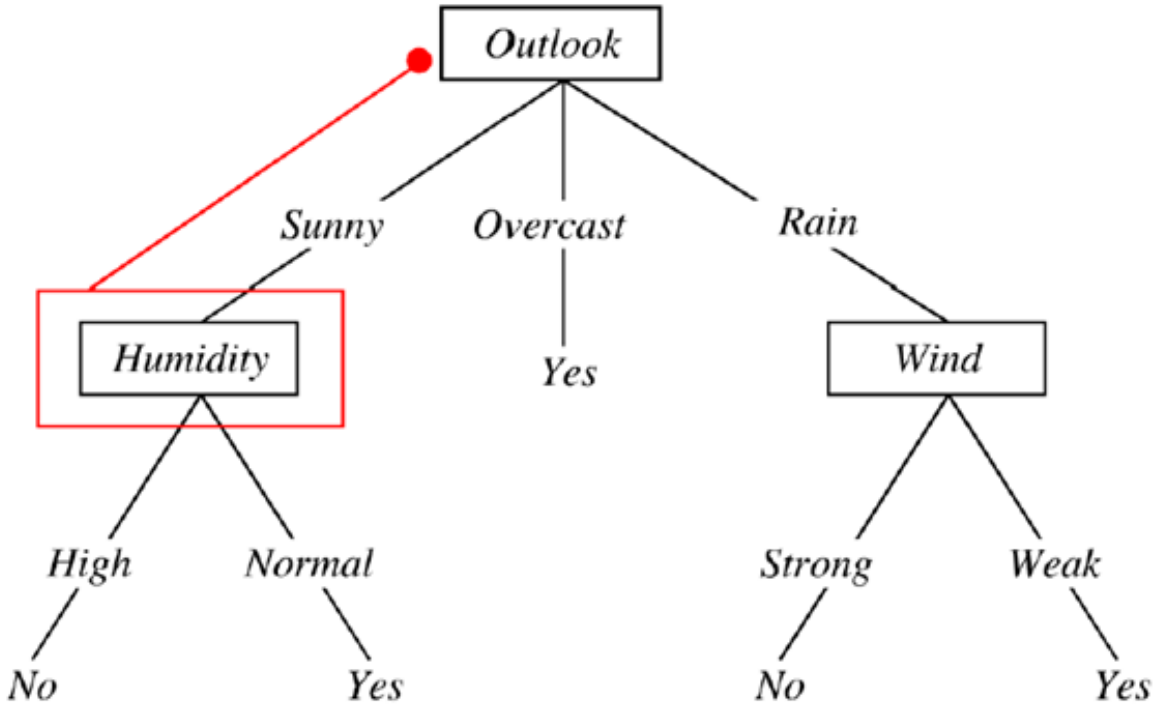
Come possiamo decidere se è una giornata adatta per giocare a tennis:



Esempio: `[O=Sunny, T=Hot, H=High, W=Strong]` ?

Giochiamo a tennis?

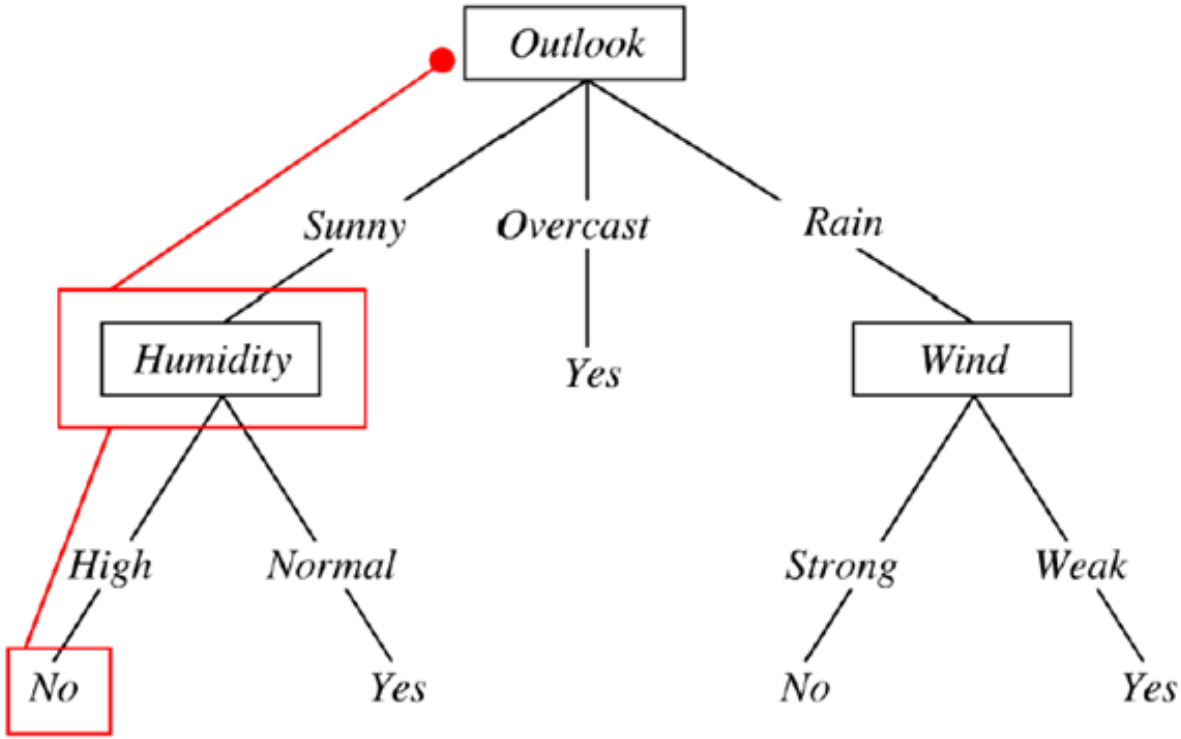
Alla radice è associato l'attibuto `Outlook (O)` quindi, essendo `Outlook = Sunny` nell'esempio, si segue il ramo `Sunny`:



Esempio: `[O=Sunny, T=Hot, H=High, W=Strong]` ?

Giochiamo a tennis?

Al nodo raggiunto è associato l'attibuto `Humidity (H)` e quindi, essendo `Humidity = High` nell'esempio, si segue il ramo `High` ottenendo così la classificazione `NO`:



Esempio: `[O=Sunny, T=Hot, H=High, W=Strong]` ? NO

Alberi di decisione: quando usarli

Gli alberi di decisione sono particolarmente adatti a trattare:

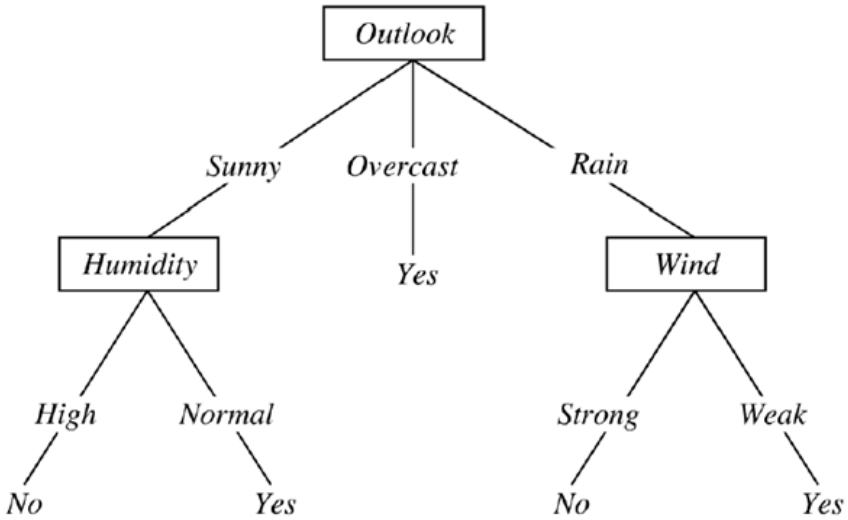
- Istanze rappresentate da coppie attributo-valore:
  - Insieme fissato di attributi e valori
  - Pochi valori possibili per gli attributi
  - Attributi a valori discreti (ma anche reali)
- Funzioni target con valori di output discreti (anche più di 2 valori);
- Concetti target descritti da disgiunzioni di funzioni booleane;
- Esempi di apprendimento che possono contenere errori e/o valori mancanti.

Gli algoritmi di apprendimento per alberi di decisione sono in genere molto efficienti. Per questi motivi gli alberi di decisione sono (ancora) molto utilizzati in applicazioni pratiche.

Alberi di decisione e funzioni booleane

Un albero di decisione può essere rappresentato da una funzione booleana:

- 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;
- Le due regole sopra definiscono una serie di DNF (disjunctive normal form), una per classe.



DNF corrispondente a YES

(O=Sunny **and** H=Normal)
or
(O=Overcast)
or
(O=Rain **and** W=Week)

ID3: algoritmo di apprendimento alberi di decisione

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

ID3(S: insieme di esempi, A: insieme di attributi)

- Crea il nodo radice T
- Se gli esempi in S sono tutti della stessa classe c, ritorna T etichettato con la classe c;
- Se A è vuoto, ritorna T con etichetta la classe di maggioranza in S;
- Scegli $a \in A$, l'attributo "ottimo" in A;
- Partiziona S secondo i possibili valori che a può assumere: $S_{a=v_1}, \ldots, S_{a=v_m}$ dove m è il numero di valori distinti possibili dell'attributo a;
- Ritorna l'albero T avente come sottoalberi gli alberi ottenuti richiamando ricorsivamente **ID3**($S_{a=v_j}$,$A - a$), per ogni j.

ID3: scelta dell'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 di informazione.

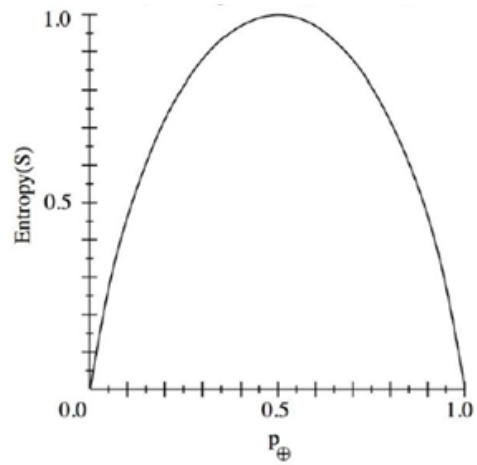
L'entropia è una misura del "grado di impurità" di un insieme di esempi S. Sia C il numero di classi e S_c il sottoinsieme di S di esempi di classe c, l'entropia è calcolata con la formula:

$$E(S) = -\sum_{c=1}^{m} p_c \log(p_c), \text{ dove } p_c = \frac{|S_c|}{|S|}$$

che nel caso di classificazione binaria diventa:

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

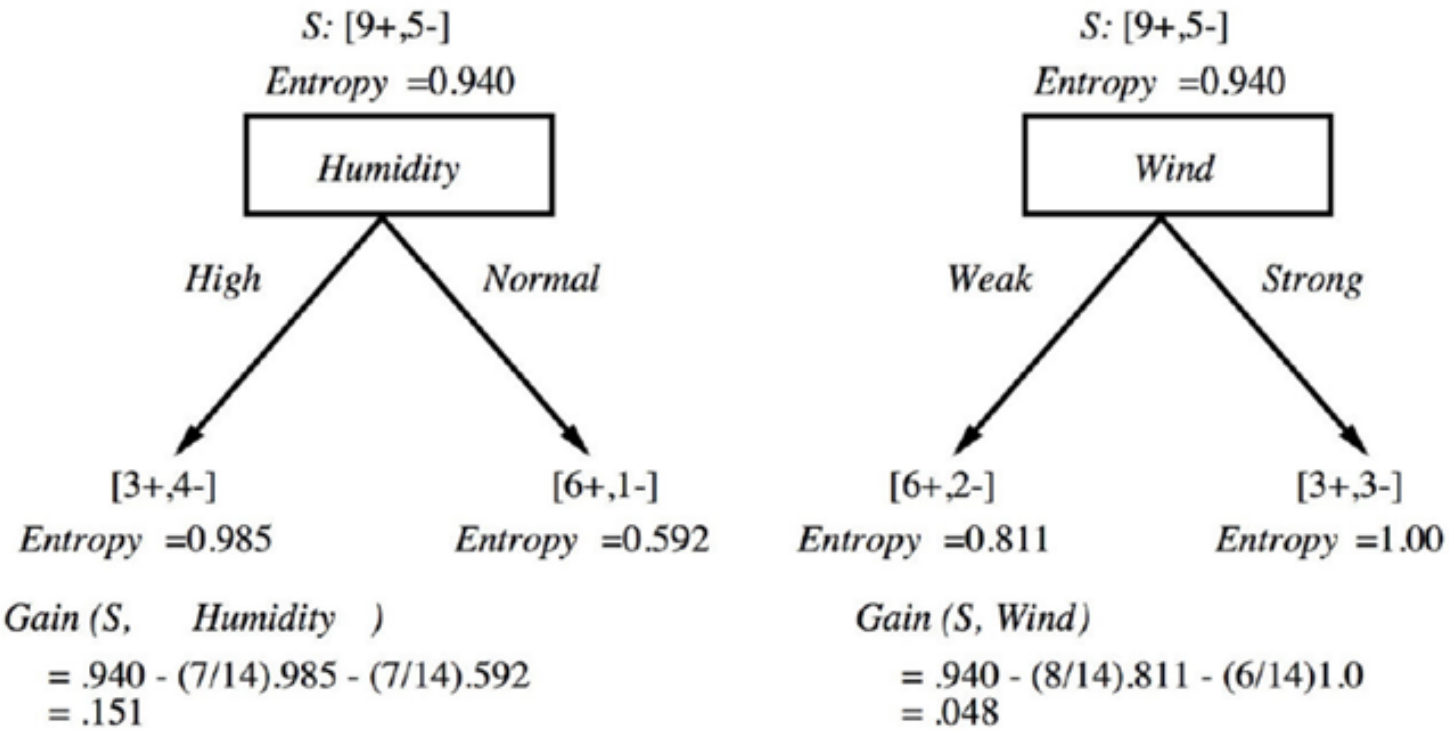
con p_-, p_+ la proporzione di esempi negativi (risp. positivi) in S.



ID3: scelta dell'attributo ottimo

Si seleziona l'attributo che massimizza l'Information Gain $G(S, a)$, ovvero la riduzione *attesa* di entropia che si ottiene partizionando gli esempi in S usando l'attributo a:

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



Criteri alternativi per la scelta dell'attributo ottimo

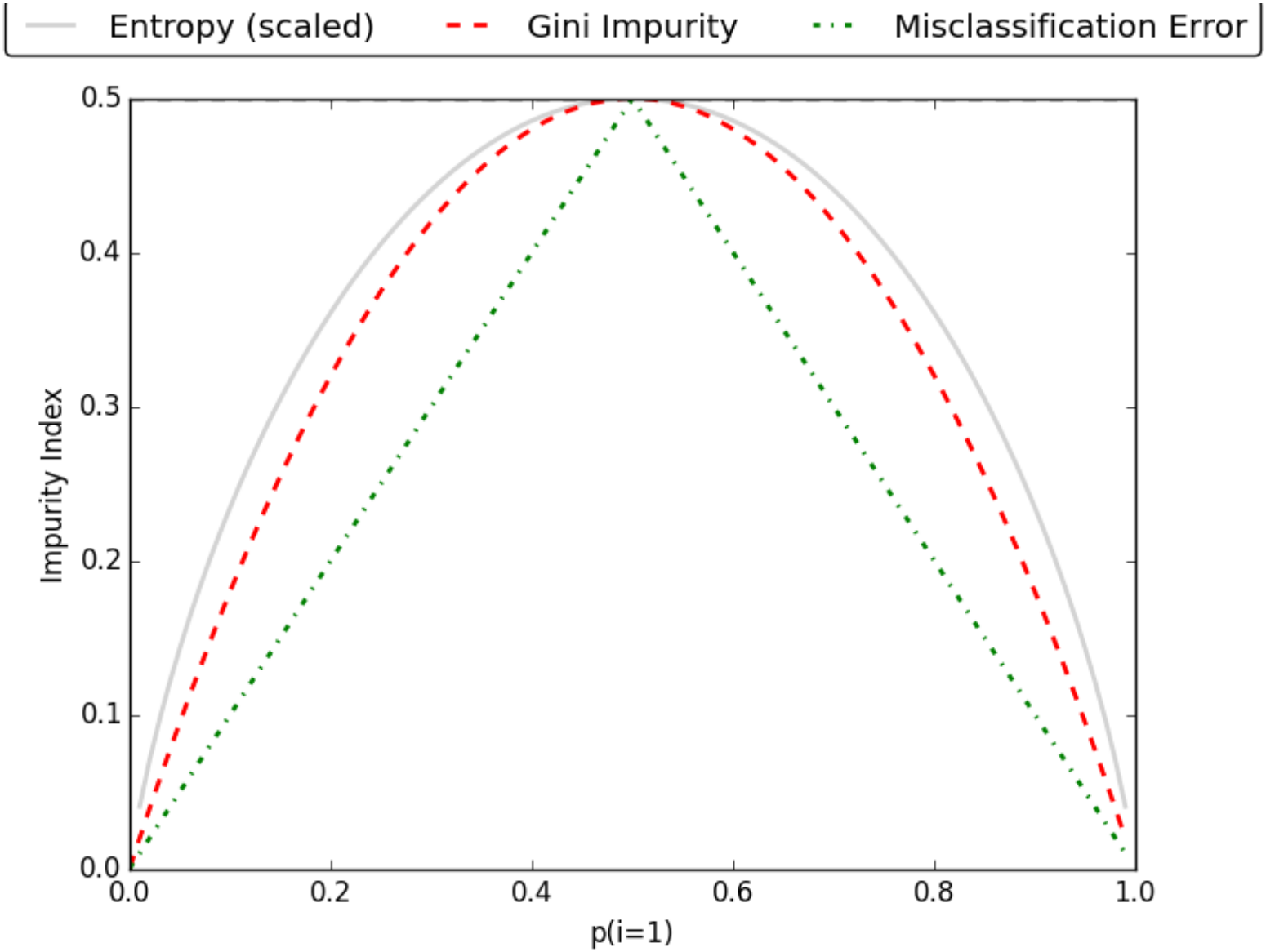
Possiamo definire la nozione di Information Gain sulla base di altre misure della impurità:

- Cross-Entropy: $I_H = -\sum_{c=1}^{m} p_c \log(p_c)$
- Gini Index: $I_G = 1 - \sum_{c=1}^{m} p_c^2$
- Misclassification: $I_E = 1 - \max_c(p_c)$

Sia I uno qualsiasi di questi criteri di impurità, analogamente a quanto fatto in precedenza, possiamo definire il guadagno di informazione come:

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

Criteri alternativi per la scelta dell'attributo ottimo



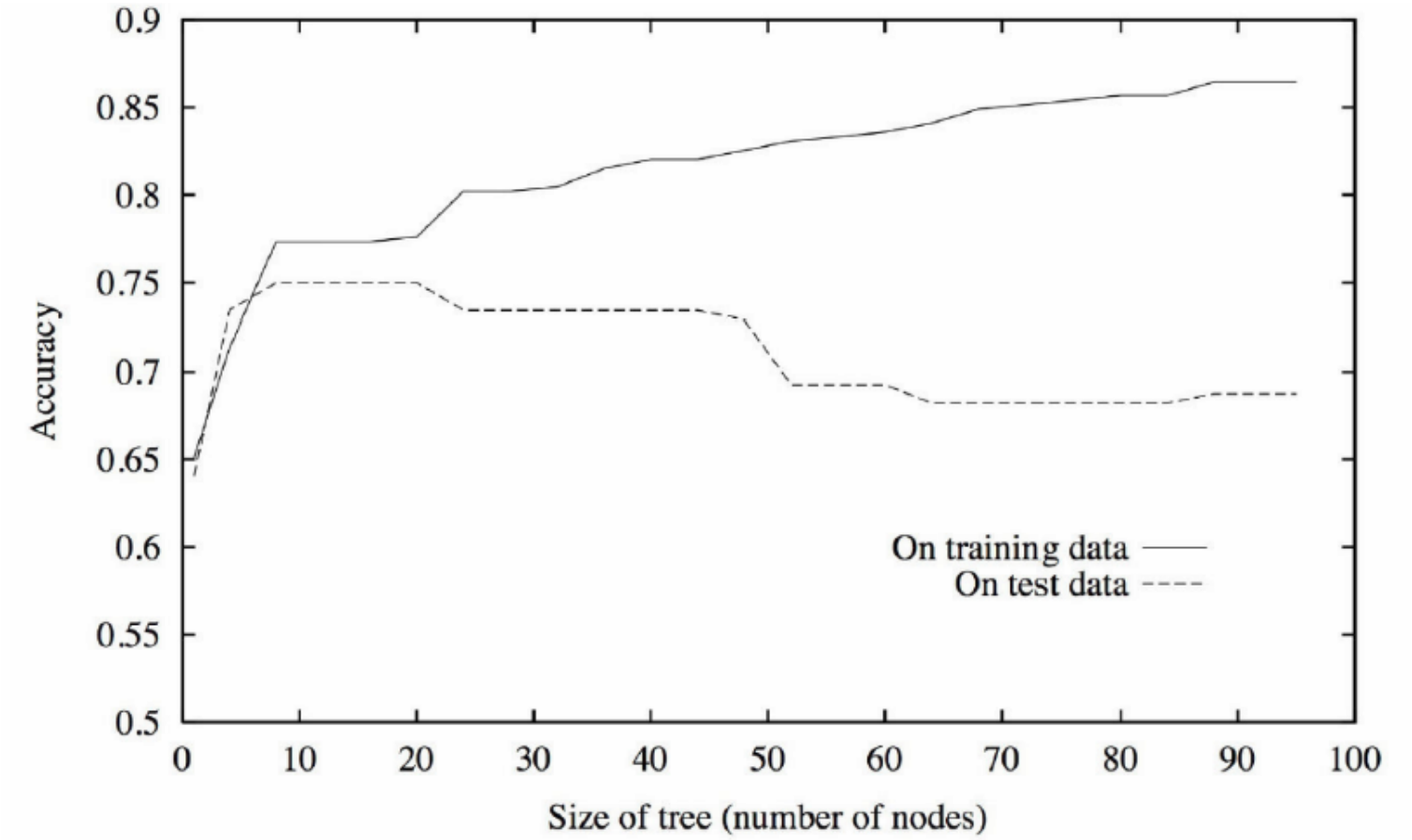
Information gain, controindicazioni

Problema. L'information gain favorisce troppo gli 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 impurità nulla), anche se in realtà quell'attributo non è significativo.

Apprendimento di alberi di decisione: overfitting

Problema: Overfitting



Parziale soluzione (sklearn): Selezionare il numero minimo di esempi da utilizzare in nodi foglia o limitare la massima profondità dell'albero

Recap

Nozioni

- Alberi di decisione: quando usarli?
- Spazio delle ipotesi di alberi di decisione (DNF)
- Algoritmo ID3
- Scelta dell'attributo ottimo (Info Gain, misure di impurità)

Esercizi

- Usando l'algoritmo ID3, calcolare (a mano) l'albero di decisione per formule booleane semplici (AND, OR, XOR, ...)