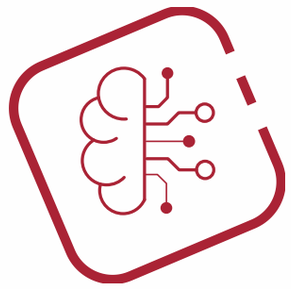
ITSAR ISTITUTO TECNICO SUPERIORE
ANGELO RIZZOLI

Algoritmi per il Machine Learning

Ing Andrea Colleoni



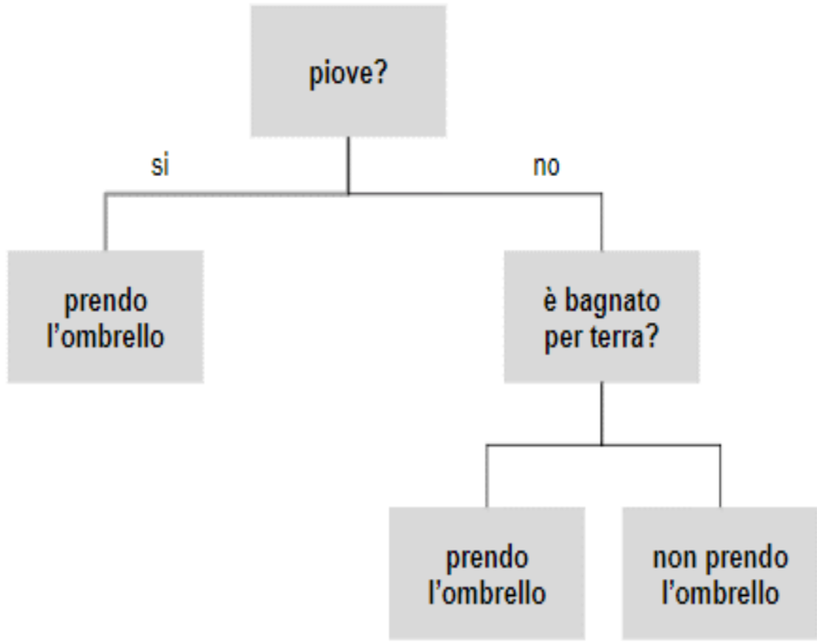
Alberi di decisione

- Un albero di decisione è un albero, tipicamente binario, nel quale ogni nodo interno rappresenta una possibile domanda a risposta (binaria o multipla), mentre le foglie rappresentano delle decisioni.
- Nel contesto del machine learning, un albero rappresenta una mappa $y \sim f(x)$

  a partire dalla radice, ciascun nodo interno e una domanda relativa al vettore di attributi x (tipicamente nella forma $\text{“}x_j \leq \theta?\text{”}$) e ogni foglia rappresenta una stima del valore di y (o una distribuzione di probabilità sui suoi possibili valori).

Esempio

- Comincia sempre dal nodo radice, il nodo genitore situato più in alto nella struttura, e procede verso il basso.
- La decisione finale si trova nei nodi foglia terminali, quelli più in basso.



Considerazioni

- Vantaggi
  - Gli alberi logici hanno l'indiscusso vantaggio della semplicità. Sono facili da capire e da eseguire.
  - Rispetto alle reti neurali l'albero decisionale è facilmente comprensibile dagli esseri umani.
  - L'uomo può verificare come la macchina giunge alla decisione. Eventualmente dissentire.
- Svantaggi
  - La rappresentazione ad albero decisionale è poco adatta per i problemi complessi, perché lo spazio delle ipotesi diventa troppo grande.
  - La complessità spaziale dell'algoritmo potrebbe diventare proibitiva.

Problemi da risolvere sull'imprevedibilità

- Quali sono gli attributi che hanno il maggiore contenuto informativo per poter prendere delle decisioni? Quelli che ci fanno decidere «meglio» e «prima»?
  => Quali attributi utilizzare per prendere delle decisioni?

Information Gain => Shannon Entropy

- Posizionare le decisioni in alto (verso la radice) o in basso (verso le foglie) può dare risultati diversi.
  => Come decidere a che livello prendere determinate decisioni?

Splitting measures => Gini Index

Addestramento e imprevedibilità

Consideriamo la variabile casuale Y da cui sono estratti i valori di uscita del dataset D. Alcune osservazioni di base sulla «prevedibilità» di Y sono le seguenti:

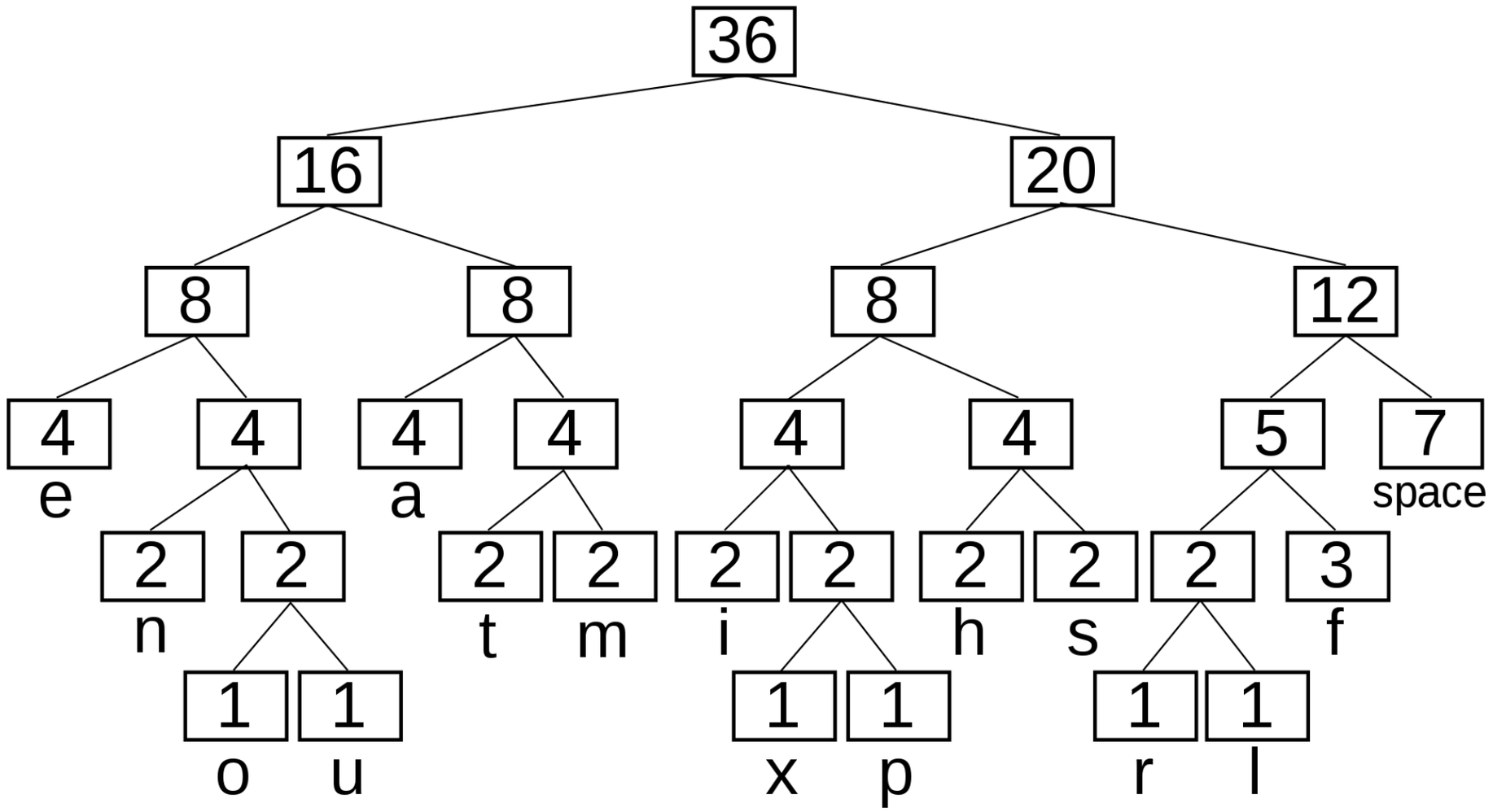
- se Y ha un solo valore, ovvero se ha più valori, ma uno solo di essi ha probabilità pari a 1, allora la variabile e prevedibile senza bisogno di ulteriori informazioni;
- più in generale, se un valore e molto più probabile degli altri (sole, pioggia o neve nel mezzo del Sahara oggi?), allora e possibile «azzardare» una previsione anche in assenza di informazioni aggiuntive (gli attributi x);
- più Y e vicina all'uniformità (tutti i suoi valori sono equiprobabili), più difficile e azzardare una previsione.

Codifica di Huffman (1)

- Nella teoria dell'informazione, per codifica di Huffman si intende un algoritmo di codifica dei simboli usato per la compressione di dati, basato sul principio di trovare il sistema ottimale per codificare stringhe basato sulla frequenza relativa di ciascun carattere.
- Questa tecnica funziona creando un albero binario di simboli:

1. Ordina i simboli, in una lista, in base al conteggio delle loro occorrenze.
2. Ripeti i seguenti passi finché la lista non contiene un unico simbolo:
    1. Prendi dalla lista i due simboli con la frequenza di conteggio minore. Crea un albero di Huffman che ha come "figli" questi due elementi, e crea un nodo «genitore»
    2. Assegna la somma del conteggio delle frequenze dei figli al genitore e ponilo nella lista in modo da mantenere l'ordine.
    3. Cancella i figli dalla lista.
3. Assegna una parola codice a ogni elemento basandosi sul path a partire dalla radice.

Codifica di Huffman (2)



Codifica di Huffman della frase "this is an example of a huffman tree" con rappresentazione binaria e indice di frequenza delle lettere.

Entropia di Shannon (1)

Supponiamo che Y abbia *l* valori diversi, di probabilità p1,...,p*l*; Alice e Bob conoscono la distribuzione di probabilità; Alice osserva una sequenza di eventi estratti da Y e vuole comunicare questa sequenza a Bob. Quanti bit deve usare Alice, come minimo? Consideriamo alcuni casi semplici:

- Se p1= 1 e p2=⋯=p*l*= 0, allora non c'è bisogno di inviare informazioni: Bob sa già che ogni evento risulterà nell'unico valore certo.
- Se *l* è una potenza di 2 (*l*= 2^r) e p1=⋯=p*l*=1/*l*, allora il meglio che Alice possa fare è codificare ogni valore di Y con una diversa combinazione di r bit.
- Supponiamo che *l*= 4 e che Y abbia la seguente distribuzione:

$$p_1 = \frac{1}{2}, \qquad p_2 = \frac{1}{4}, \qquad p_3 = p_4 = \frac{1}{8}.$$

Allora è possibile adottare una codifica di Huffman, in cui la lunghezza del codice dipende dalla probabilità del valore:

$$1 \mapsto 0, \qquad 2 \mapsto 10, \qquad 3 \mapsto 110, \qquad 4 \mapsto 111.$$

Entropia di Shannon (2)

Questa codifica permette di ridurre il numero atteso di bit da spedire al seguente valore:

$$\overbrace{\frac{1}{2}\cdot 1}^{1 \text{ bit se } Y=1} + \overbrace{\frac{1}{4}\cdot 2}^{2 \text{ bit se } Y=2} + \overbrace{\frac{1}{8}\cdot 3}^{3 \text{ bit se } Y=3} + \overbrace{\frac{1}{8}\cdot 3}^{3 \text{ bit se } Y=4} = \frac{7}{4} = 1.75,$$

il che costituisce un miglioramento rispetto all'uso della codifica uniforme a 2 bit.

Il numero atteso di bit da trasmettere rappresenta una misura dell'uniformità della variabile casuale. Si noti che, in tutti i casi elencati sopra, il numero di bit da trasmettere per comunicare l'esito *i* della variabile casuale è pari a:

$$b(i) = \log_2 \frac{1}{p_i} = -\log_2 p_i.$$

Entropia di Shannon (3)

Un risultato fondamentale della Teoria dell'Informazione di Shannon e precisamente che l'equazione precedente è vera in generale. Quindi, il numero atteso di bit necessari a trasmettere un evento estratto dalla variabile casuale Y:

$$\mathrm{H}(Y) = -\sum_{i=1}^{\ell} p_i \log_2 p_i .$$

La grandezza H(Y) è detta *entropia di Shannon* della variabile casuale Y. Se la distribuzione di probabilità di Y è concentrata in un singolo valore di probabilità 1, allora H(Y) = 0. Invece, l'entropia è massima quando Y è uniforme, e vale H(Y) = $\log_2 l$.

Impurità di Gini (1)

Supponiamo che Alice osservi un evento *i* estratto da Y . Bob cerca di indovinare *i* generando un valore

casuale *î* con la stessa distribuzione di probabilità, a lui nota. Qual e la probabilità che Bob sbagli?

- Se Y ha un solo valore di probabilità 1, allora Bob indovina di certo.
- Piu in generale, se Y ha un valore molto più probabile degli altri, e poco probabile che Bob sbagli.
- Intuitivamente, la probabilità di errore e massima quando la distribuzione di Y è uniforme.

Se Alice osserva il valore *i* Bob genererà il valore corretto con probabilità p*i*, quindi sbaglierà con probabilità (1 – p*i*). La probabilità di errore complessiva, mediata su tutti i possibili esiti, è:

$$\mathrm{GI}(Y) = \sum_{i=1}^{\ell} p_i(1-p_i) = 1 - \sum_{i=1}^{\ell} p_i^2.$$

Impurità di Gini (2)

La grandezza GI(Y) e detta impurità di Gini della variabile casuale Y .

- Se la distribuzione di probabilità di Y e concentrata in un singolo valore di probabilità 1, allora GI(Y) = 0.
- Invece, l'impurità di Gini e massima quando Y è uniforme, e vale:
  GI(Y) = 1 – 1//
  valore che tende asintoticamente a 1 al crescere del dominio di Y .

Esempio

	Gender	Height	Weight	Index
0	Male	174	96	4
1	Male	189	87	2
2	Female	185	110	4
3	Female	195	104	3
4	Male	149	61	3

As in all algorithms, the cost function is the basis of the algorithm. In the case of decision trees, there are two main cost functions: the Gini index and entropy.
Any of the cost functions we can use are based on measuring impurity. Impurity refers to the fact that, when we make a cut, how likely is it that the target variable will be classified incorrectly.

In the example above, impurity will include the percentage of people that weight >=100 kg that are not obese and the percentage of people with weight<100 kg that are obese. Every time we make a **split** and the classification is not perfect, the split is impure.