Confronto tra algoritmi per la creazione di alberi di decisione - ID3 vs C4.5

Federico Zanardo

25 Ottobre 2020

1 Introduzione

Un albero di decisione è un modello predittivo che permette di apprendere delle regole di decisione rappresentabili come alberi. La struttura di un albero di decisione è la seguente:

1. ogni *nodo interno* dell'albero rappresenta un **attributo**;
2. ogni *arco* corrisponde ad un possibile valore che può essere assunto dal nodo genitore, quindi, dall'attributo;
3. ogni nodo *foglia* assegna una **classificazione**.

Questo modello predittivo viene applicato in quei contesti in cui è necessario comprendere quali sono state le scelte che l'algoritmo ha fatto per restituire in output una determinata classificazione. Infatti, gli alberi di decisione possono essere tradotti in una serie di condizioni *if*. Così facendo, l'uomo è in grado di comprendere il percorso (nell'albero) che ha fatto l'algoritmo per giungere alla conclusione restituita.

2 Entropia

L'entropia è una misura del *grado di impurità* di un insieme di esempi S. Siano C il numero di classi, S_c il sottoinsieme di S di esempi di classi c e $p_c = \frac{|S_c|}{|S|}$.

I criteri principali per il calcolo dell'entropia sono:

1. **Cross-Entropy**: $-\sum_{c=1}^{m} p_c \cdot \log_2(p_c)$
2. **Gini Index**: $1 - \sum_{c=1}^{m} p_c^2$
3. **Misclassification**: $1 - max_c(p_c)$

3 Information Gain

Il *guadagno informativo* $G(S, a)$ rappresenta la riduzione *attesa* di entropia, ottenuta dal partizionamento dell'insieme S secondo i possibili valori che può assumere l'attributo a. L'attributo che massimizza il guadagno informativo è l'attributo che minimizza l'impurità (l'entropia). Pertanto, nella creazione dell'albero di decisione si cerca sempre l'attributo che riesce a massimizzare il guadagno informativo.

La formula per calcolare l'*Information Gain* è:

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

dove $a \in A$, E è l'*entropia* e $S_{a=v}$ è l'insieme degli esempi in S con attributo a e valore dell'attributo uguale a v.

4 ID3

Un algoritmo famoso per la costruzione di un albero di decisione è l'algoritmo **ID3**. L'albero viene costruito seguendo una procedura del tipo *divide-et-impera*, che costruisce l'albero *top-down* in modo ricorsivo.

Sia S l'insieme degli esempi che si hanno a disposizione e sia A l'insieme degli attributi. I passi eseguiti dall'algoritmo sono:

1. Crea il nodo radice dell'albero di decisione T;
2. Se gli esempi presenti in S appartengono tutti alla stessa classe c, allora l'algoritmo ritorna l'albero T etichettato con la classe c;
3. Se $A = \emptyset$, allora viene ritornato T etichettato con la classe di maggioranza presente in S;
4. Si scelta $a \in A$ tale che a sia un **attributo ottimo** in A (è possibile che vi siano più attributi ottimi; in questo si opterà nel sceglierne uno);
5. Si partizioni S secondo i possibili valori che può assumere l'attributo a, ovvero,
$$S_{a=v_1}, \ldots, S_{a=v_n}$$
dove n rappresenta il numero dei possibili valori distinti che l'attributo a può assumere.
6. Viene ritornato l'albero T che ha come sottoalberi gli alberi risultanti dalle chiamate ricorsive su ID3:
$$ID3(S_{a=v_i}, A - a)$$
per ogni i.

Gli svantaggi di questo algoritmo sono:

1. Si può verificare il fenomeno di *overfitting* nel caso in cui si disponga si un piccolo insieme di esempi;
2. Viene testato solo un attributo alla volta per prendere una decisione;
3. Non è in grado di gestire gli attributi numerici e i valori mancanti.

5 C4.5

Questo algoritmo è un'evoluzione dell'algoritmo ID3 che cerca di sopperire ai suoi principali svantaggi:

1. Gestisce gli attributi sia continui che discreti. Per gestire gli attributi continui, viene creata una *soglia*: l'elenco viene diviso tra i valori che sono inferiori o uguali alla soglia e quelli che invece sono superiori alla soglia;

2. Gestisce i dati incompleti;

3. Risolve il problema dell'overfitting applicando delle tecniche di *potatura* dell'albero. La potatura consiste nel sostiuire alcuni rami con dei nodi foglia.

Per quanto riguarda il calcolo degli attributi categoriali, C4.5 e ID3 utilizzano il medesimo processo. La principale differenza tra ID3 e C4.5 riguarda gli *attributi numerici*: C4.5 presenta due metodi per la gestione dei valori numerici di un attributo.

La principale differenza tra i due algoritmi consiste nei diversi metodi utilizzati per il calcolo del guadagno informativo. C4.5 utilizza due metodi:

1. Il primo metodo cerca di sopperire ad un problema legato a quei attributi che contengono delle ripetizioni di uno stesso valore (i.e. $a = \{5, 6, 3, 5, 7, 5, 9, 1\}$). Il problema è che l'attributo a ha così tanti valori possibili che vengono utilizzati per dividere gli esempi dell'insieme di train in sottoinsiemi *molto piccoli*, avendo così un guadagno informativo alto. Pertanto, il modello allenato non sarà un buon predittore per dati futuri. Una soluzione è l'utilizzo del **gain ratio**: con questo metodo penalizza gli attributi come a, comprendendo al suo interno un termine molto sensibile al modo in cui l'attributo a divide i dati in modo uniforme (lo **split info**).

$$GainRatio = \frac{InformationGain}{SpitInfo}$$

2. Il secondo metodo consiste nel considerare ogni valore di un attributo numerico come *candidato*: quindi per ciascun attributo numerico, si seleziona l'insieme di esempi minore o uguale al candidato e l'insieme di esempi maggiore di tale candidato. Inoltre, viene calcolata l'entropia per i due insiemi e il guadagno di informazioni per il candidato che si riferisce ai due insiemi di esempi.