PROBLEMA DELL' OVERFITTING

Ovvero quando l'albero costruito risulta essere troppo fitto e con troppi branch, e con alcuni di essi che portano a delle anomalie e missclassification. Da tutto ciò l'accuratezza scende.

Come capire se siamo in overfitting?

Andando a confrontare l'accuratezza del modello sui dati di training e poi su quelli di test.
Se abbiamo che:

$$A_{Test} \ll A_{Training} \Rightarrow \text{OVERFITTING}$$

Soluzione: PRUNING.

TREE PRUNING

Iniziamo parlando di stima dell'errore per il training (e_{TR}) e per l'errore nel test set (e_{TS}).
Approcciando il calcolo in modo pessimistico, possiamo dire che per ogni foglia:

$$e_{TS} = e_{TR} + 0.5 .$$

Di conseguenza l'errore totale:

$$e_{TS} = e_{TR} + N \cdot 0,5$$

Con N = numero di foglie.

Esempio considerando un albero con 30 foglie e un set di training grande 1000 ma con un errore di 10 su 1000 esempi:

$$e_{TR} = 10$$
$$e_{TS} = 10 + 30 \cdot 0,5 = 25$$

Che se lo rapportiamo in percentuale su tutto il dataset abbiamo 1% (10/1000) di errore nel training e il 2.5% (25/1000) di errore nel test.

Per stimare e_{TS}, detto anche GENERALIZATION ERROR, dobbiamo introdurre un nuovo set, detto PRUNING SET che è difatti un ulteriore split del training set.

Per potare un DT possiamo usare due approcci:

1) PRE-PRUNING
2) POST-PRUNING

PRE-PRUNING

L'idea è di fermare la costruzione del DT prima del previsto. Per farlo introduciamo nuove stopping condition

1) Fermati se le istanze della classe di uno split sono sotto una di predeterminata threshold.
2) Fermati se la classe etichetta è indipendente rispetto agli attributi rimasti (Chi-square ...)
3) Fermati se il nodo corrente non migliora l'impurità dello split.

POST-PRUNING

Dopo aver costruito l'intero DT, potarlo, rimuovendo branches, o interi sotto-alberi. Una volta rimosso un ramo o un sotto-albero esso va rimpiazzato con una foglia. La foglia conterrà la classe di maggioranza del sotto-albero.
Per decidere come potare l'albero, bisogna prendere la decisione nell'ottica di ridurre l'errore di pruning.
Il guadagno dato dal tagliare il sotto-albero T con radice in un nodo, supponiamo il nodo v, è:

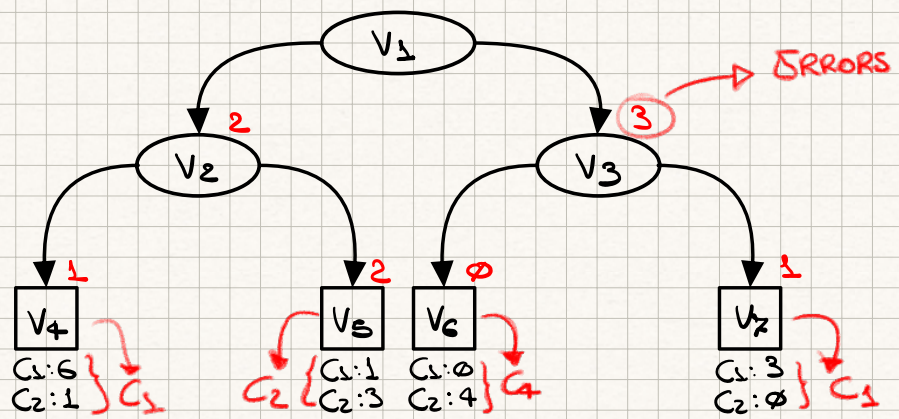
$$GAIN = \#misclassification(T) - \#misclassification(v).$$

Potare ai nodi con GAIN maggiore, finché non rimangono solo nodi con GAIN negativo.

Restrizione: possiamo potare T solo se T stesso NON contiene sotto-alberi con errore minore di T.

L'intuizione della formula del GAIN è che, se c'è più errore mantenendo l'albero così come è (# misclassification(T)) rispetto all'errore che si avrebbe con v come foglia con la classe maggioritaria di T (# misclassification(v)), allora si procede col pruning.

Esempio



Costruendo un DT dal training set, esce il DT di sopra. Supponiamo di analizzarlo sul seguente PRUNING SET:

V4	$C_1:2$	$C_2:1$
V5	$C_1:2$	$C_2:1$
V6	$C_1:\emptyset$	$C_2:2$
V7	$C_1:3$	$C_2:1$

PRUNING SET

Conviene tagliare il sotto-albero in V_2?

Lasciando l'albero così com'è abbiamo che

$$\#\text{misclassification}(T) = 3$$

Che è dato dalla somma degli errori in V_4 e V_5.
In V_4 abbiamo errore = 1 perché classifica secondo la sua classe di maggioranza: C_1. Ma nel pruning V_4 ha una tupla che in realtà appartiene alla classe C_2, da qui l'errore.
Se noi decidiamo di tagliare in V_2, dobbiamo mettere in V_2 una foglia con la classe di maggioranza del sotto-albero, che in questo caso è C_1 (perché avremmo C_1: 7 e C_2: 4):

$$\#\text{misclassification}(V_2) = 2.$$

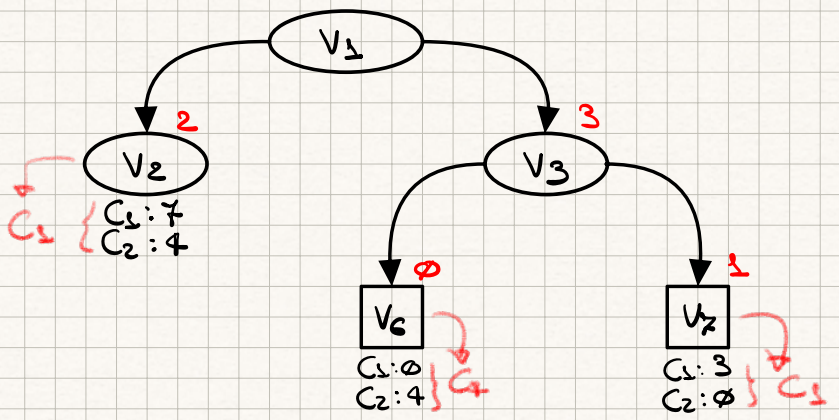
Due perché ora dobbiamo considerare il pruning set dato dalla somma dei pruning set per V_4 e V_5.
Quindi:

$$GAIN(V_2) = 3 - 2 = 1$$

Seguendo lo stesso approccio testiamo anche V_3:

$$GAIN(V_3) = 1 - 3 = -2$$

Conviene fare il pruning su V_2!



PROBLEMI

Un DT potato può soffrire ancora di

1) Ripetizione : stesso attributo testato più volte nel ramo
2) Replicazione : più sotto-alberi uguali