Appunti di Machine Learning

4 ottobre 2022

Rosso Carlo

Contents

1 Decision Trees

L'apprendimento mediante alberi di decisione è uno dei metodi più utilizzati e pratici per l'apprendimento induttivo. L'apprendimento mediante alberi di decisione cerca in uno spazio di ipotesi completamente espressivo. Il loro bias è la preferenza per alberi piccoli rispetto ad alberi grandi.

Gli alberi di decisione approssima una funzione discreta, dove la funzione appresa è un albero di decisione; possono anche essere rappresentati come una serie di *if-then rules*, per migliorare la comprensione. Ciascun nodo dell'albero coincide con il testi di un attributo, e ciascun arco, che connette i figli, indica il valore dell'attributo. I nodi foglia indicano la classe dell'istanza. In effetti si tratta di una tecnica di classificazione. Fondamentalmente, ciascun nodo dell'albero rappresenta una condizione ("if"), e quindi una congiunzione ("and") di condizioni; mentre ciascun arco rappresenta un il valore delle condizioni possibili ("is a"), e quindi una disgiunzione ("or"). Per questo motivo, si capisce bene che questo approccio è possibile solamente se le variabili sono discrete.

1.1 Algoritmo ID3

L'algoritmo ID3 è l'algoritmo di base, da cui partiamo. In questo caso l'albero è costruito a partire dalla root e quindi ha una costruizione bottom-up. Funziona testando l'attributo che contiene la maggior informazione. C'è quindi bisogno di chiarire come selezionare l'attributo più utile per classificare gli esempi. L'*information gain* è una proprietà statistica che misora quanto un attributo separa gli esempi di allenamento in base alla loro classe di appartenenza.

Def. 1.1 *Sia S una collezione, contenente esempi positivi e negativi, l'entropia di S relativa alla rappresentazione booleana è così definita:*

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i \tag{1}$$

dove p_i è la proporzione di esempi in S che appartengono alla classe i. L'entropia vale 0 se tutti gli esempi sono dello stesso tipo, e 1 se gli esempi sono divisi equamente tra le classi. L'entropia è il numero di bit necessari per rappresentare un esempio di S, per questo motivo è utilizzato il logaritmo in base 2.

Def. 1.2 *Sia l'entropia una misura di impurità in una collezione di esempi. Possiamo ora misurare quanta informazione contiene ciascun attributo per classificare gli esempi. L'information gain, banalmente, è la partizione di esempi secondo l'attributo. Più precisamente, l'information gain, $G(S, A)$ di un attributo A, relativo ad una collezione di esempi S, è così definita:*

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2}$$

dove $Values(A)$ sono tutti i possibili valori che può assumere un attributo A, e S_v è il suttoinsieme di S per cui l'attributo A assume valore v.

- L'information gain è proprio la misura usata dall'algoritmo ID3 per effettuare la scelta greedy e decidere quale attributo controllare per primo;
- L'albero è costruito in modo tale che lo stesso attributo compare solo una volta in ciascun percorso dell'albero decisionale; Ciascun processo continua per ciascun nuovo nodo foglia fino a che o tutti gli attributi sono stati inclusi nel percorso, oppure tutti gli esempi associati alla foglia hanno medesima classificazione;
- L'algoritmo ID3 è uno spazio completo di funzioni finite a valori discreti;
- L'ID3 non permette il backtracking, le decisioni di costruzione dell'albero iniziale permangono valide anche se gli esempi di training cambiano. L'albero cresce per ottenere una soluzione ottima locale, ma non globale. La soluzione in merito a questo è utilizzare l'algoritmo di *post-pruning*, per cui l'albero viene come ribilanciato.

Una buona approssimazione dell'ID3 è la seguente: alberi più piccoli sono preferiti ad alberi più grandi. Sono preferiti gli alberi che pongono gli attributi con maggior information gain vicino alla radice.

Per cui il bias è dato dalla sua strategia di ricerca. In genere questo tipo di bias è chiamato preference bias.

2 lezioni

- **hold-out**: sono trattenuti v esempi dal training set. Questi sono utilizzati come validation set, ovvero per misurare l'accuratezza del modello allenato;
- **k-fold cross validation**: si suddivide il training set in k partizioni, di cui una viene utilizzata come validation set, mentre le altre k-1 vengono utilizzate per l'allenamento. Questo processo viene ripetuto k volte, in modo da utilizzare ogni volta una partizione diversa come validation set. L'accuratezza del modello viene calcolata mediando le k misurazioni ottenute;

3 metriche

- **accuracy**: è la percentuale di esempi correttamente classificati:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{P + N} \tag{3}$$

  dove T sta per true, F per false, P per positive, N per negative;

- **precision**: è la percentuale di esempi positivi classificati rispetto ai TP e ai FP:

$$precision = \frac{TP}{TP + FP} \tag{4}$$

- **recall**: è la percentuale di esempi positivi classificati rispetto ai TP e ai FN:

$$recall = \frac{TP}{TP + FN} \tag{5}$$