L'apprendimento degli alberi di decisione

Riccardo Malavolti - 5797787 22/1/2017

Introduzione

Gli alberi di decisione sono uno strumento molto utile per visualizzare processi di scelta e per stabilire delle strategie relative al dominio di applicazione.

I vantaggi non si fermano solo all'uso dell'utente finale: si prestano molto bene come strumenti attivi nei problemi di classificazione e regressione, dove si cerca di predire l'appartenenza alla classe o un valore target numerico dati in ingresso degli input con cui eseguire i test contenuti all'interno dell'albero.

Per ottenere un buon grado di affidabilità nella previsione è necessario applicare metodi di apprendimento basati su esempi, in modo tale da costruire tramite approccio top-down un albero che cresca "imparando" sulla base dei dati forniti.

Apprendere l'albero

L'algoritmo implementato (una variante di ID3) costruisce l'albero scegliendo volta per volta l'attributo più efficace a dividere gli esempi, usandolo come nodo da cui poi proseguire la costruzione ricorsivamente. Ma come si determina l'attributo più adatto? Una buona misura della bontà della scelta è rappresentata dal guadagno di informazione, basato sul concetto di entropia.

Così come nell'ambito fisico, l'entropia misura il "disordine" presente nella collezione di insiemi prodotta dalla scelta di un dato attributo: sarà massima se la proporzione tra esempi positivi e negativi sarà 50%-50% mentre sarà 0 se l'attributo realizza un insieme di insiemi tutti della stessa classe.

Il guadagno di informazione descrive semplicemente quanto è adatto l'attributo a suddividere gli esempi: è ovvio scegliere il test che realizza la divisione più netta possibile.

Con questa politica di scelta, l'algoritmo costruisce l'albero, diramandone i nodi fino a creare delle foglie (che rappresentano una risposta alla richiesta di classificazione). C'è il pericolo però che questa crescita sia esagerata e finisca con il produrre un albero particolarmente complesso e sovradattato agli esempi forniti, incapace quindi di fornire previsioni accurate e sensate.

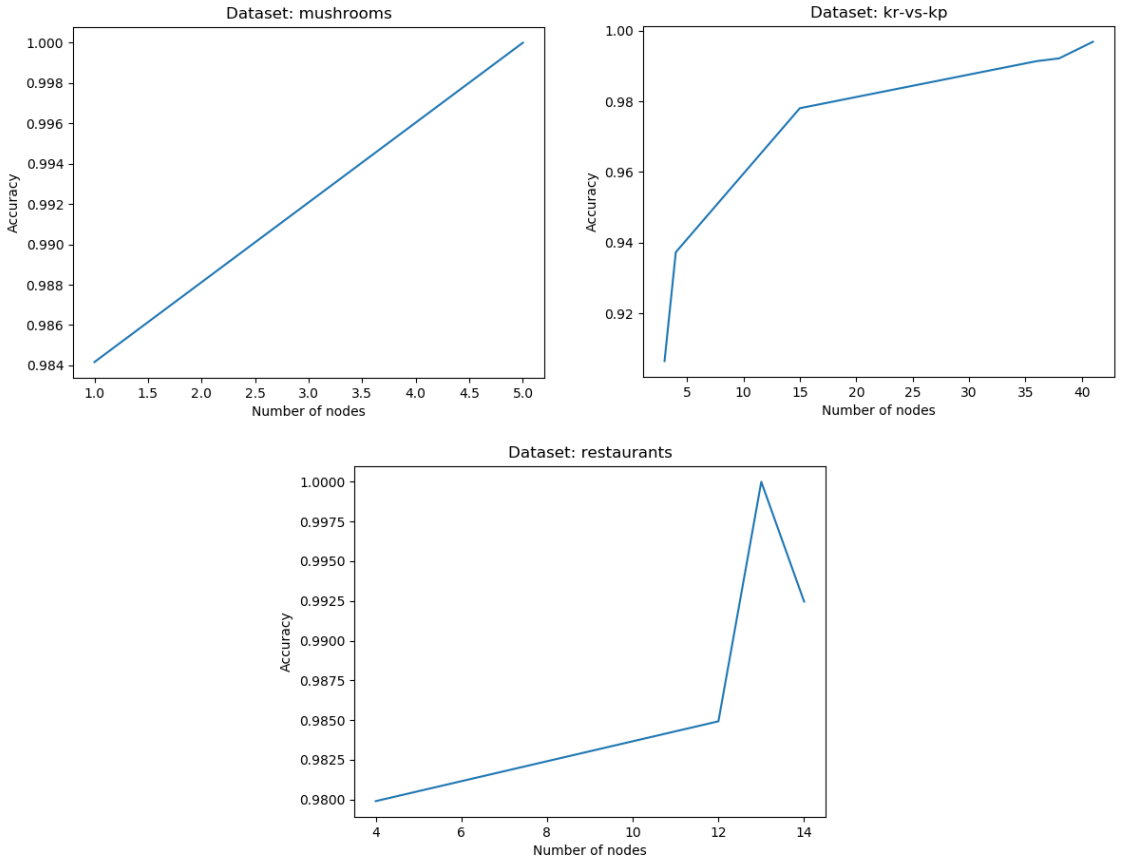
Nella variante implementata – come da richiesta – si è inserito un criterio di stop alla proliferazione dei nodi basato sulla percentuale di errori commessi da una foglia che

rimpiazza l'attributo in esame: se catalogare un pool di esempi (non completamente omogenei) sotto la stessa classe produce un errore accettabile (fornito come parametro) allora è conveniente posizionare la foglia invece di continuare a inserire test.

Risultati sperimentali

L'algoritmo così implementato è stato testato con alcuni dataset liberamente reperibili sul web.

- "kr-vs-kp": descrive le mosse di un fine partita tra re+torre bianchi e re+pedone neri associando la vittoria/non vittoria del bianco.
- "mushrooms": classifica varie specie fungine in due classi (edibile/velenoso) sulla base di attributi morfologici.
- "restaurants": una strategia di decisione sull'attendere o meno un tavolo al ristorante di un hotel basandosi sulle caratteristiche del ristorante stesso.



Osservando i grafici si nota che non sempre avere più test porta ad una maggiore precisione: nel dataset "restaurants" aggiungere troppi nodi porta a overfitting, assente nel caso di "kr-vs-kp". Il caso di "mushrooms" invece evidenzia un pattern molto semplice presente nei dati, con un solo test si raggiungono punteggi molto alti, prossimi al 100% se si utilizzano 5 nodi.

Fonti:

- Mitchell, T. M. (1997). "Machine Learning", Ch 3: "Decision Tree Learning" 52-76.
- Norvig, P. & Stuart, R. (2007). "Intelligenza Artificiale: un approccio moderno", Ch 18.3: "Apprendere alberi di decisione".
- "kr-vs-kp" : https://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+King-Pawn)
- "mushrooms" : https://archive.ics.uci.edu/ml/datasets/mushroom
- "restaurants": https://github.com/aimacode, dove è possibile generare un dataset sintetico di dimensione a piacere.