Università degli Studi di Firenze
Scuola di Ingegneria
Corso di Laurea Triennale in Ingegneria Informatica
Intelligenza Artificiale
a.a. 2019–2020

Potatura di Regole negli Alberi di Decisione

Francesco Ballerini

Sommario

Si illustrano di seguito i risultati di un'implementazione Python 3.7 per l'apprendimento di alberi di decisione—con e senza potatura delle regole—su quattro data set distinti—uno generato artificialmente e tre preesistenti. I test eseguiti consistono nel tracciamento della curva di apprendimento e in una *k-fold cross-validation*.

1 Data set

I data set soggetti a sperimentazione sono i seguenti:

- RESTAURANT, generato dal codice in `restaurant_dataset.py` sulla base dell'albero di decisione in [1, Figura 18.2]. Contiene un numero di esempi a scelta dell'utente con 11 attributi; ciascun attributo ha un dominio discreto e finito.
- PLANTS, BOOKS e BUSINESS, contenuti nei file `plants.csv`, `books.csv` e `business.csv`, rispettivamente, e scaricabili da [2]—per il link specifico si veda il file `testing_functions.py`:
  - PLANTS contiene 2691 esempi con 7 attributi;
  - BOOKS contiene 2687 esempi con 7 attributi;
  - BUSINESS contiene 1257 esempi con 6 attributi.

  In tutti e tre i data set gli attributi hanno un dominio discreto e finito.

2 Test

I file `learning_curve_test.py` e `cross_validation_test.py` contengono test che chiedono all'utente di scegliere il data set su cui operare—tra quelli descritti nella sezione 1—e quanti esempi del data set scelto utilizzare. Gli esempi, una volta che se ne è scelta la quantità, vengono divisi in *training set* e *test set* secondo le modalità specifiche del test; l'algoritmo di apprendimento senza potatura delle regole sfrutta l'intero training set per la costruzione dell'albero di decisione, mentre l'algoritmo con potatura ne utilizza due terzi per la costruzione dell'albero e un terzo come *validation set* per realizzare la potatura stessa.

I tempi di esecuzione riportati nelle sezioni 2.1 e 2.2 si riferiscono ai risultati ottenuti su un portatile con processore Intel Core i7 quad-core a 2 GHz, 8 GB di RAM e sistema operativo Ubuntu 16.04.

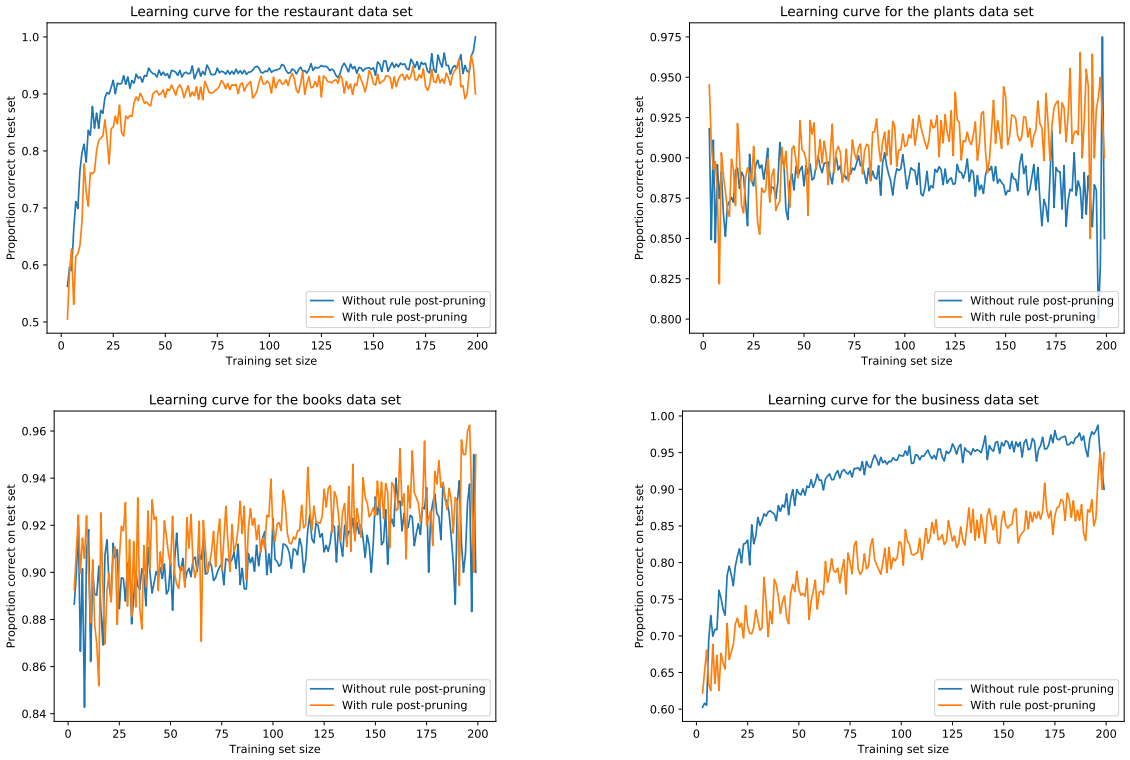

Figura 1: Curve di apprendimento con 200 esempi estratti dai data set RESTAURANT (4 minuti), PLANTS (5 minuti), BOOKS (6 minuti) e BUSINESS (7 minuti)

2.1 Curva di apprendimento

Il file `learning_curve_test.py` traccia le curve apprendimento dei due algoritmi con albero di decisione—uno con e uno senza potatura delle regole. I grafici riportano l'accuratezza—misurata rispetto al test set—delle previsioni dei due algoritmi al crescere del training set—e, conseguentemente, al decrescere del test set. Ogni punto in ciascun grafico è il risultato della media di 20 valori.

Un esempio dei risultati ottenibili è mostrato in figura 1; sono riportati anche i corrispondenti tempi di esecuzione.

2.2 K-fold cross-validation

Il file `cross_validation_test.py` calcola l'accuratezza dei due algoritmi di apprendimento rispetto al test set applicando una *k-fold cross-validation*, con $k = 10$: si eseguono 10 round di apprendimento, in ciascuno dei quali 1/10 degli esempi è utilizzato come test set e il resto come training set, e si calcola la media dei 10 valori di accuratezza prodotti.

Un esempio dei risultati ottenibili è mostrato in tabella 1.

Data Set	Accuratezza		Tempo (secondi)	
	senza potatura	con potatura	senza potatura	con potatura
RESTAURANT	0.997	0.961	0.104	7.767
PLANTS	0.934	0.945	0.396	446.565 (≈ 7 min)
BOOKS	0.941	0.942	0.337	841.848 (≈ 14 min)
BUSINESS	0.999	0.974	0.158	52.778

Tabella 1: Risultati di una 10-fold cross-validation su 1000 esempi di RESTAURANT, 2691 di PLANTS, 2687 di BOOKS e 1257 di BUSINESS

Riferimenti bibliografici

[1] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Third Edition. Pearson, 2010.

[2] MLData repository. URL: `http://mldata.org`