Categorizzazione di documenti testuali

Francesca Del Lungo

Aprile 2018

1 Introduzione

Nel seguente elaborato si andranno ad analizzare gli aspetti relativi alla classificazione di documenti testuali. In particolare si utilizzano due algoritmi: Bernoulli Naive Bayes e Perceptron su due distinti datasets: 20NewsGroups e Reuters-21578.

2 Datasets

I datasets di documenti testuali utilizzati sono: 20 NewsGroups e Reuters-21578.

Per il primo dataset si è utilizzata la versione originale, i dati sono distribuiti in 20 file, ognuno corrispondente ad un diverso argomento. Ciascun gruppo è composto da 1000 documenti, per gli esperimenti sono stati utilizzati tutti. Vengono, però, rimosse le parti del testo relative a: headers, footers e quotes, per prevenire il fenomeno dell' overfitting.

I dati relativi al secondo dataset sono distribuiti in 22 file, nei primi 21 sono contenuti 1000 documenti, mentre nell'ultimo 578. Il dataset è formato da 90 categorie distinte, ma, come richiesto, sono stati usati solo documenti delle 10 categorie più frequenti.

Molti documenti hanno diversi topic, per semplificare l'esperimento si è deciso di assegnare a ciascun documento una sola fra le sue categorie, scegliendo quella a cui appartenevano meno documenti all'interno del dataset. Inoltre i documenti relativi alle due classi più frequenti (earn e acq) sono stati ridotti da 2800/1500 a 700 per rendere il dataset più equilibrato.

3 Implementazione

Si implementa quanto descritto con il linguaggio di programmazione Python. Gli algoritmi utilizzati fanno parte della libreria *scikit-learn*. I programma è formato dai seguenti file:

- **main.py**: in questo file vengono richiamate le funzioni definite negli altri file. In sequenza esegue: Naive Bayes e perceptron sul dataset 20NewsGroups e poi ancora Naive Bayes e perceptron sul dataset Reuters-21578.

- **News20.py**: in questo file vengono acquisiti i dati del dataset (attraverso la specifica funzione fetch_20newsgroups), vengono poi preparati per il classificatore (i documenti vengono trasformati in matrici) e infine vengono calcolate le learning curves relative. Viene inoltre calcolata l'accuratezza della predizione sui documenti del test.
- **reuters.py**: qui vengono acquisiti i documenti e i relativi target del secondo dataset. Attraverso la funzione get_10most_common_categories si vanno a selezionare le 10 categorie con più documenti all'interno del dataset. Grazie alla funzione create_new_dataset vengono considerati nel dataset solo i documenti appartenenti alle 10 categorie selezionate. Poi troviamo, anche qui, preparazione del dataset per il classificatore e le funzioni per il calcolo delle learning curves. Viene inoltre calcolata l'accuratezza della predizione sui documenti del test.
- **curve plot.py**: in questo file viene implementata la funzione per plottare il grafico con le learning curves.

4 Risultati sperimentali

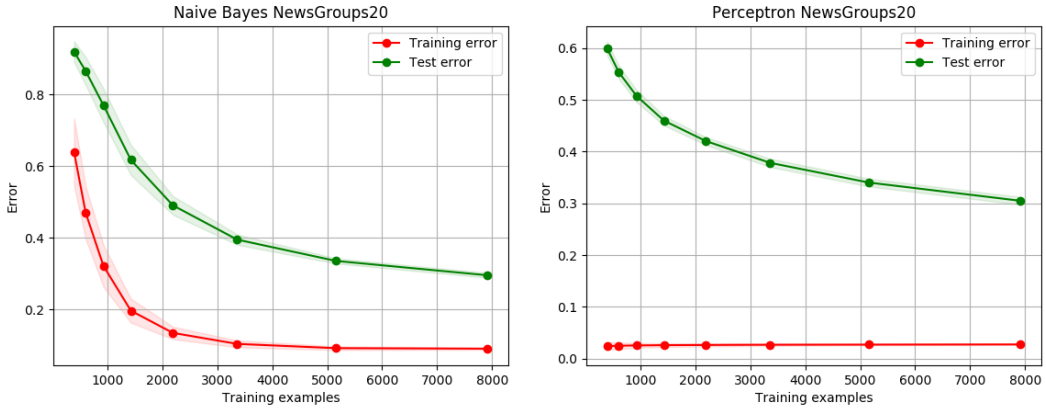

Figure 1: Learning curves relative al dataset 20NewsGroups. A sinistra quella relativa all'algoritmo Naive Bayes, a destra il Perceptron.

5 Riproduzione risultati

Per eseguire l'esperimento è necessario installare le librerie sklearn e nltk e reperire i due dataset: 20NewsGroups reperibile alla pagina `www.qwone.com/~jason/20Newsgroups/` e Reuters-21578 reperibile alla pagina `www.daviddlewis.com/resources/testcollections/reuters21578/` . Per eseguire il programma

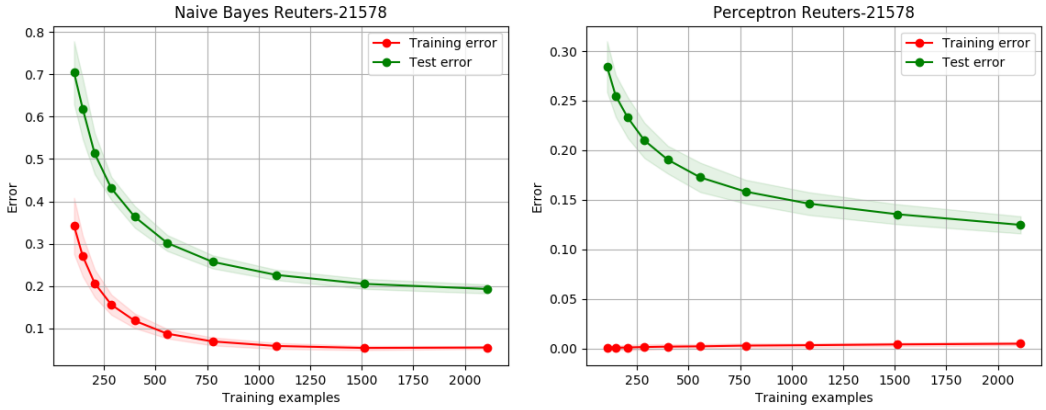

Figure 2: Learning curves relative al dataset Reuters-21578. A sinistra quella relativa all'algoritmo Naive Bayes, a destra il Perceptron.

è sufficiente eseguire il file **main.py** che produrrà di default le learning curves relative a tutti e quattro gli esperimenti trattati (quattro esperimenti dati da due dataset e due classificatori), per eseguire solo parte dei test sarà sufficiente andare ad agire su questo file.

6 Osservazioni

Possiamo notare come per il dataset 20NewsGroups il valore dell'errore del training rimanga in proporzione sempre maggiore rispetto ai valori dell'errore del secondo dataset. Ciò è sicuramente dovuto al fatto che questo è molto più grande dell'altro dataset e soprattutto al fatto che esso contiene alcuni duplicati.

Reuters-21578 con le modifiche apportate ha delle learning curves che presentano un valore dell'errore molto basso all'aumentare dei documenti (per entrambi i classificatori), questo si può riscontrare nell'accuratezza che è di circa 85% per entrambi i classificatori.

L' andamento delle learning curves relative al classificatore Bernoulli Naive Bayes è giustificato dal fatto che esso lavora con le probabilità quindi l'errore va a diminuire e a convergere lentamente via via che procede l'apprendimento con l'aumento dei documenti. Le learning curves relative al Perceptron rispecchiano le sue proprietà, esso, infatti, utilizza il vettore dei pesi per modificare il suo output e quindi per produrre una classificazione corretta. Tale classificatore è utilizzabile solo per dataset linearmente separabili.

Come riportato in: *"McCallum & Nigam 1998 - A Comparison of Event Models for Naive Bayes Text Classification"* si nota come per la classificazione di documenti testuali, l'algoritmo Multinomial Naive Bayes abbia prestazioni migliori della versione Bernoulli, quest' ultima (utilizzata nel test) non considera il numero di volte che ogni parola appare in un documento, ma solo se essa

appare o no; in particolare include esplicitamente la probabilità che le parole non appaiano nel documento.

7 Conclusioni

Entrambi i classificatori hanno prodotto buoni risultati in poco tempo, si può notare come un dataset più vasto e variegato produca una minore accuratezza. Piccoli accorgimenti verso i documenti del dataset, così come lievi variazioni ai parametri delle funzioni possono diminuire molto l'errore visibile dalle learning curves.