Andamento dell'errore di generalizzazione con il numero di esempi per Naive Bayes Multinomial classifier

Fabio Tognaccini

January 26, 2020

Abstract

In questo progetto si cercato di replicare i risultati di [1] per quanto riguarda I dataset OHSUMED[2], 20 newsgroups[3], and Rcv1[4]. Il codice con file readme usato disponibile nella repository github[1].

1 Procedura

Per diverse disponibilit ogni dataset stato elaborato in modo diverso, in tutti i casi si cercato di mantenersi pi fedeli possibile a [1]. Per ogni dataset la funzione di valutazione ha eseguito 30 tests per ogni dimensione del training set ed stata calcolata media sia della accuratezza che della AUC come da [1]. Quando elaborazione del testo sia stana necessaria si seguito procedure come da[4]

1.1 OHSUMED

Per il dataset OHSUMED stata usata la versione preprocessata a [5], la stessa usata da [1], bastato ricreare la matrice di token counts dal file e il relativo array di categorie per fornirlo alla funzione di valutazione.

1.2 20 Newsgroup

20 Newsgroup non era disponibile in formato preprocessato, stato quindi elaborato tramite un tentativo di rimuovere quanto pi metadata possibile[2], per poi eseguire trasformazione in minuscolo, rimozione cartteri non alfanumerici e parole di solo cifre, rimozione di accenti, tokenizzazione e stemming in modo simile a come decritto in[5]. Dopo di questo stata eseguita la stessa procedura di OSHUMED. A differenza del dataset OSHUMED era disponibile una pre divisione in set di addestramento e test ed stata usata nei test.

[1] `https://github.com/Cacho-tognax/Naive-bayes-text-categorization`

[2] documenti diversi hanno formati diversi quindi richiedono procedure diverse, nel particolare stato rimosso qualunque testo prima di 'Lines:', 'Writes:' e 'Version:' se presenti

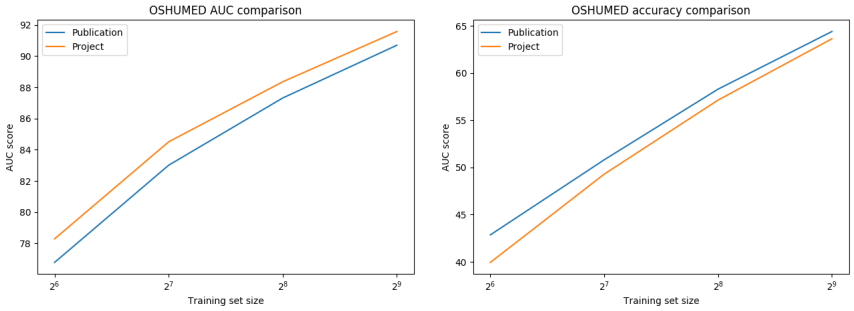
1.3 Rcv1-V2

Questo dataset stato diviso in 4 sottodataset corrispondenti alle categorie figlie di 'Root', e come categorie sono state usate le categorie discendenti dirette delle 4 macrocategorie. Le informazioni sono state estratte dalla versione preprocessata disponibile in[4]. Da qui bastato eseguire un conteggio dei tokens per fornire i dati necessari alla funzione di valutazione. Anche in questo caso era disponibile una pre divisione in training e test set ed stata usata.

2 Risultati

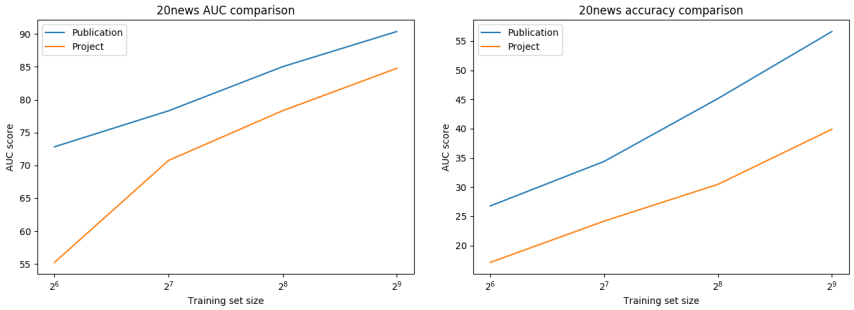
Qui verrano riportati i risultati dei miei test e confrontati ai risultati di[1], questa sezione divisa per dataset. Dato che lo studio riguardava l'andamento della generalizzazione dell'errore in funzione dalla dimensione del dataset, e la riproduzione dei risultati di un articolo, grafici che mostrano l'andamento della performance del classificatore in funzione della dimensione del training set(etichetta = Project) confrontata con i risultati dell'articolo (etichetta = Publication) sembrata la scelta migliore. Quando possibile sia il punteggio AUC sia l'accuratezza sono mostrate. I valori sull'asse x sono scalati logaritmicamente, i valori sull'asse y sono espressi in percentuale.

2.1 OHSUMED



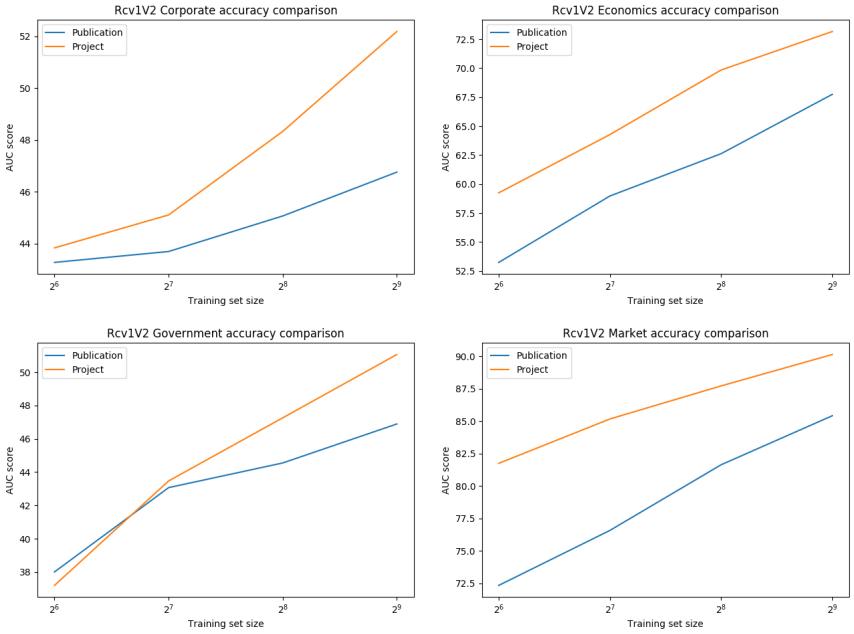
Possiamo vedere tendenze simili, ma un comportamento migliore per il punteggio AUC del progetto.

2.2 20 Newsgroup



Possiamo notare un comportamento decisamente peggiore con piccolo training set, anche se pu essere attribuito a varianza dovuta al gran numero di classi in confronto alla dimensione del training set (20 vs 64), ma possiamo notare una performance molto peggiore, possibilmente causata da un peggiore lavoro di preprocessing.

2.3 Rcv1-V2



Per motivi sconosciuti non stato possibile calcolare la performance AUC per questo specifico dataset. Si pu notare simili andamenti ma il progetto ha performance nettamente superiore, forse causate dallo skew del dataset e non indicative di qualit migliore del processo di classificazione.

3 Conclusioni

Ci sono svariati passaggi che possono essere implementati in modi diversi, quindi le ragioni delle differenze tra i risultati del progetto e della pubblicazione possono essere molteplici. Questa relazione pu solo confermare che un training set di maggiori dimensioni aumenta drasticamente le performance del Multinomial Naive Bayes classifier.

A References

[1] Jiang Su, Jelber Sayyad-Shirabad, Stan Matwin Large Scale Text Classification using Semi-supervised Multinomial Naive Bayes `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.231.7128&rep=rep1&type=pdf`

[2] available at `https://trec.nist.gov/data/t9_filtering.html`

[3] available at `http://qwone.com/~jason/20Newsgroups/`

[4] Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397, 2004. `http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf` Data available at `https://perun.pmf.uns.ac.rs/radovanovic/dmsem/cd/datasets/text/Forman/`

[5] `https://perun.pmf.uns.ac.rs/radovanovic/dmsem/cd/datasets/text/Forman/`, most likely originally from

E.H. Han and G. Karypis.Centroid-based document classification algorithms: Ana-lysis & experimental results.Technical Report TR-00-017, Department of ComputerScience, University of Minnesota, Minneapolis, 2000. Also possibly from

Eui-Hong Sam Han and George Karypis. Centroid-Based Document Classification: Analysis & Experimental Results. In Proceedings of the Fourth European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD), pages 424-431, Lyon, France, 2000.

[6] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. OReilly Media Inc