Sentiment Classification: comparing Perceptron and Decision Tree

Gabriele Magrini

08 January 2021

1 Introduzione

Nel corso di questa relazione ci prefiggiamo una analisi comparativa tra due tipi di classificatori, il perceptron e il decision tree, nel caso in cui vengano adoperati per una Setiment Classification.
Per inciso, verranno esaminati i risultati di questi classificatori nell'identificare correttamente una recensione positiva o negativa, nel complesso campo cinematografico.
Sono presenti molteplici casi in cui è evidente come relative modifiche nel dataset training produca grosse differenze nel risultato finale.
L'obiettivo prefissato era tuttavia quello di riprodurre sperimentalmente i risultati esposti da Bo Pang e Lillian Lee in "Thumbs up? Sentiment Classification using Machine Learning Techniques".
Tuttavia, verranno esplorati altri casi particolari dove specifici accorgimenti producono interessanti conseguenze.

Andiamo adesso a discutere più nel dettaglio i suddetti risultati e la metodologia di lavoro.

2 Metodologia e strumenti di lavoro

Per la scrittura del codice è stato utilizzato Python,attraverso l'IDE PyCharm contenente a bordo Anaconda 3.6, mentre per l'implementazione del Perceptron e del Decision Tree è stata utilizzata la libreria Sklearn.
I dataset utilizzati sono principalmente 2: Uno fornito direttamente da Bo Pang e Lillian Lee, l'altro liberamente scaricabile da "kaggle.com/c/word2vec-nlp-tutorial/rules", che racchiude review e sentiment di una collection di 25000 recensioni di IMBD.
Inoltre, sono presenti alcune recensioni extra, sulle quali è stato testato il predittore.

Il codice è strutturato in 5 differenti file, 4 adibiti allo svolgimento di uno specifico test, più uno adibito alla traduzione in dataset del corpus di review disponibile.
Le 4 classi sono praticamente identiche tra loro, differenziate da alcune linee di codice che modificano il dataset: questo per comodità di testing e soprattutto per maggiore chiarezza nell'effetto del codice sul dataset.
In "SentimentAnalisys.py" viene eseguito il codice di base per il training e testing del classificatore, dove l'unica cosa notabile è la divisione del dataset in misure 70/30 tra training e testing.
Inoltre,è in questo file che viene testato l'uso di unigrammi e bigrammi.
In "SentimentAnalisysPOS.py", oltre al codice base vengono taggate tramite ntlk.pos_tag le singole parole di ogni recensione del dataset con il tipo di parola corrispondente(aggettivo, pronome, verbo, ecc), e poi unite al tag stesso con due semplici cicli.
In "SentimentAnalisysAdjectives.py", oltre al tagging viene eseguito un semplice filtraggio dove solo gli aggettivi vengono salvati.
Infine, in "SentimentAnalisysPosition.py", viene effettuato un "tagging" manuale delle parole, a seconda di dove esse si trovino (posizionalmente) nella recensione.
Esattamente come fatto da Pang e Lee, inoltre, il countVectorizer è stato impostato con una frequenza minima di "features" a 4, ovvero non sono presenti nel vocabolario elementi che appaiono meno di 4 volte.
Questa apparentemente insignificante differenza ha prodotto discreti risultati positivi nell'accuratezza del Perceptron(ad eccezione dei casi "bigrams" e "adjectives"), lascindo tuttavia inalterata o a volte peggiorando(di poco) quella del Decision Tree.

Qui di seguito ecco quindi i risultati ottenuti con i due classificatori.

3 Risultati ed esperimenti

Le due tabelle sotto espongono quindi alcune caratteristiche ricorrenti: principalmente, che il perceptron riesce a classificare sempre meglio, e spesso con notevole distacco, i sentimenti delle recensioni rispetto al decision tree; inoltre, risulta chiaro come(presumibilmente per lo stesso motivo), il perpceptron risulti enormemente più sensibile ai cambiamenti nella tecnica utilizzata.
Notiamo poi che, esattamente come appuntato da Pang e Lee, anche in questo esperimento risultino molto più accurati i classificatori nel caso si conti la pura presenza di parole, piuttosto che la loro frequenza.
Inoltre, piuttosto sorprendentemente, otteniamo il miglior risultato tra tutti i

	Features	# of features	Freq/Pres	P	DT
(1)	unigrams	12587	FREQ	81.5	63.3
(2)	unigrams+bigrams	42424	FREQ	82.6	62.3
(3)	bigrams	29837	FREQ	78.83	62
(4)	unigrams+POS	23814	FREQ	80.83	56.33
(5)	adjectives	1886	FREQ	72	58.5
(6)	unigrams+position	19128	FREQ	78.16	60

	Features	# of features	Freq/Pres	P	DT
(1)	unigrams	12587	PRES	85.66	63.66
(2)	unigrams+bigrams	42424	PRES	85.33	58.66
(3)	bigrams	29837	PRES	82	63
(4)	unigrams+POS	23814	PRES	81	60.66
(5)	adjectives	1886	PRES	72.16	61
(6)	unigrams+position	19128	PRES	80.66	61.66

Table 1: Tabella confronto classificatori FREQUENZA(sopra)
e PRESENZA(sotto)

casi usando semplicemente gli unigram, sia con il perceptron che con il decision tree, contrariamente a quanto visto negli esperimenti di Pang e Lee.
Piccola nota: il corpus di recensioni sopra usato è fatto da 2000 recensioni, ordinate in modo da formare un gruppo di sentimenti positivi e negativi perfettamente bilanciato, sia nel gruppo di training che di testing.
Tuttavia, è andando a usare l'altro dataset da 25.000 elementi che interessanti caratteristiche emergono:
vediamo adesso la tabella dei risultati ottenuti con il suddetto dataset.

	Features	# of features	Freq/Pres	P	DT
(1)	unigrams	26068	PRES	86.89(+1.22)	71(+7.34)
(2)	unigrams+bigrams	147806	PRES	88.36(+3.03)	71.4(+12.8)
(3)	bigrams	121738	PRES	85.86(+3.86)	68.8(+5.8)
(4)	unigrams+POS	69667	PRES	84.26(+3.26)	66.16(+5.50)
(5)	adjectives	4081	PRES	68.94(-3.22)	63.74(+2.74)
(6)	unigrams+position	44332	PRES	85.8(+5.14)	68.29(+6.63)

Table 2: Tabella confronto classificatoricon il secondo dataset(25.000 elementi

Ecco quindi che diventano espliciti dei risultati molto interessanti:
Innanzitutto, in tutti i casi eccetto quello degli aggettivi, c'è stato un sensibile incremento nell'accuratezza dei classificatori.
Più nello specifico, e come era prevedibile, i casi in cui il classificatore performava ottimamente già nel vecchio dataset subiscono un lievissimo incremento,

mentre eccezionali risultati emergono invece del Decision Tree, con un incremento massimo di 12(!) punti percentuali.
Tuttavia, dobbiamo tener conto della differenza di grandezza di dataset: mentre nel caso in cui fossimo costretti ad adoperare un albero di decisione come classificatore può anche aver senso usare questo dataset, è piuttosto sconveniente in termini di performance(=tempo di training e adattamento dataset) adoperarlo per il perceptron, data la piccola differenza di prestazioni.
In effetti, parliamo di un dataset contente più di 20 volte le recensioni di quello originale.
La differenza esigua di prestazioni dei classificatori si spiega però con un semplice dato: quello delle number(#) of features presenti.
Il vocabolario "finale" si discosta di pochissimo relativamente alla mole di recensioni aggiuntive.
Nel caso migliore per entrambi infatti, ovvero quello dell'unigram classico, il numero di features raddoppia solamente(ricordiamo che il corpus di review è adesso 20 volte maggiore!).
Un'altra osservazione, sulla quale tuttavia riservo ulteriori prove per assicurarmene la correttezza, è quella relativa alla differenza tangibile di prestazioni(per qualsiasi dataset e/o classificatore) ottenibile con l'uso della "Feature Presence" piuttosto che con l'uso della "Feature Frequency".
La mia personale ipotesi si focalizza infatti sulla natura del dataset, ovvero recensioni cinematografiche:
in tali recensioni, soprattutto in quelle professionistiche o semi-professionistiche, è intuibile come sia piuttosto raro l'uso dello stesso aggettivo più volte, e in generale degli stessi termini.
E' invece più probabile che in una sentiment analysis sia più importante la pura presenza dei termini positivi o negativi:
è chiaro che una recensione con un vocabolario(soprattutto di aggettivi) più ampio si polarizzi più facilmente verso uno dei due sentimenti.

4 Conclusioni

Abbiamo quindi osservato come la grandezza del dataset sia quasi logaritmica rispetto ai vantaggi ottenuti: solo all'inizio un aumento di grandezza del corpus equivale a un deciso miglioramento di prestazioni, mentre solo a enormi incrementi da un certo punto in poi equivalgono sensibili miglioramenti.
Come ben vediamo, tuttavia, non è dalla semplice presenza di determinati aggettivi che i classificatori danno il meglio: è anzi essenziale che essi siano allenati con una decisa varietà di vocabolo per risultare davvero efficaci.
Infine, notiamo chiaramente come il decision tree sia di fato un classificatore poco robusto per la sentiment analysis, probabilmente dato il suo principio di funzionamento, laddove il perceptron si adatta agilmente a ogni tipo di dataset risultando perciò sempre migliore nel riconoscimento di sentimento rispetto agli umani(68% di accuratezza).