Classificazione articoli ANSA via topic modelling

Alessandro Stefani[1] e Cristi Gutu[2]

[1] Corso di laurea in Statistica per le tecnologie e le scienze, matricola 1148387
alessandro.stefani.6@studenti.unipd.it
[2] Corso di laurea in Statistica per le tecnologie e le scienze, matricola 1147351
gheorghecristi.gutu@studenti.unipd.it

Sommario In questo progetto si è affrontato il problema della classificazione in macro categorie di articoli provenienti dall'agenzia ANSA.
Ci si è concentrati sul confrontare le prestazioni del classificatore utilizzando diverse rappresentazioni dei documenti.
In particolare si sono confrontati risultati utilizzando la rappresentazione con term-document matrix e la rappresentazione ottenibile con la tecnica di topic modeling chiamata Latent dirichlet allocation(LDA)[1].

Keywords: *Classification · Text mining · Text classification · n-gram · LDA · Python · sklearn*

1 Introduzione

L'efficienza, la scalabilità e la qualità degli algoritmi di classificazione per documenti testuali dipende largamente dalla loro rappresentazione; uno tra i metodi più comuni per rappresentare i documenti testuali è la term-document matrix che però normalmente genera spazi vettoriali di grande dimensione e ciò può portare a difficoltà nell'analisi dei dati; per attenuare questo problema esistono approcci basati su modelli probabilistici, che permetto di ridurre notevolmente la dimensione dello spazio delle features come LDA.

Quindi in questa relazione si vogliono paragonare i risultati della classificazione usando le due tecniche di rappresentazione citate sopra, coadiuvate da alberi decisionali[2], per la classificazione effettiva.

Per quantificare le prestazioni del classificatore sono state prese in considerazione l'accuratezza[3], per valutare la qualità delle previsioni, e i tempi di esecuzione.

Si è scelta l'accuratezza inquanto misura facilmente interpretabile, inoltre avendo classi con un numero di osservazioni bilanciato e tutte di equa importanza, non si rischia di incorrere in problemi come il "paradosso" dell'accuratezza.

Nella sezione 2 si è descritto come è stato ottenuto e come è stato strutturato il dataset, nella sezione 3 si è trattata la presentazione delle analisi svolte e dei risultati, infine nella sezione 4 sono state riportate delle conclusioni sui risultati ottenuti.

2 Dataset

I dati sono stati reperiti dall'agenzia ANSA, nota in Italia. Il campionamento degli articoli è stato fatto ottenendo i link traminte il motore di ricerca DuckDuckGo. Si è deciso di includere 6 macro categorie: (Economia, Politica, Cultura, Sport, Tecnologia, Cronaca), escludendo la categoria Mondo perchè si confonde con categorie come Economia e Politica, per ogni categoria sono stati reperiti 400 articoli sfruttando la ricerca mirata solo al sito ansa.it via le espressioni:

- site:ansa.it/sito/notizie/economia
- site:ansa.it/sito/notizie/politica
- site:ansa.it/sito/notizie/cultura

[3] L'accuratezza è il valore valore definito dall'espressione $(\frac{Previsioni\_effettuate\_correttamente}{Totale\_tentativi\_previsione})$

- site:ansa.it/sito/notizie/sport
- site:ansa.it/sito/notizie/tecnologia
- site:ansa.it/sito/notizie/cronaca

Cercando con queste espressioni si reperiscono risultati che appartengono soltanto ai temi citati sopra.

Ogni articolo è composto da: (titolo, sottotitolo, testo, tags, categoria). I tags sono parole chiavi che dovrebbero aiutare il lettore a contestualizzare il contenuto dell'articolo e di conseguenza categorizzarlo in qualche maniera. Si sono estratte da ogni articolo i campi citati sopra i cui valori sono stati salvati in formato JSON[4].

In totale si sono raccolti 2400 articoli[5], che sono stati suddivisi casualmente in training set, validation set e test set, con rispettivamente 50%, 25% e 25% dei documenti totali.

3 Esperimenti e risultati

Prima di poter utilizzare modelli per l'analisi testuale è necessario preprocessare i dati, questo è stato fatto costruendo la seguente pipeline di preprocessamento: articoli |rimozione stopwords[6] |stemming |rimozione tags html |rimozione punteggiatura |rimozione numeri |rimozione link.

3.1 Baseline

La basline per la classificazione con questo dataset e insieme di trainig si può considerare 17.8% di accuratezza, risultato ottenuto classificando utilizzando un *DummyClassifier* che classifica ogni articolo secondo la categoria più frequente del training set.

3.2 Analisi esplortive

Dopo aver ottenuto i dati si è notato che le categorie assegnate agli articoli erano in realtà micro-cateogire, perciò un ultimo step di preprocessamento è stato riclassificare manualmente gli articoli, etichettandoli con le corrispondenti macro-categorie.

Ad esempio: (Libri, Cinema, Film) → Cultura; gli articoli con micro-categoria Libri o Cinema o Film, vengono rietichettati con la macro-categoria Cultura.

Successivamente si è calcolata la term-document matrix[7] la cui classe di supporto in Python dà la possibilità di specificare 3 parametri: ngram_range, min_df e max_df; per scegliere il range di ngrammi da prendere in considerazione si è fatto riferimento al libro Social Media e Sentiment Analysis[3] dove si suggerisce che generalmente n-grammi con più di 3 termini non aggiungono contenuto informativo, i parametri max_df, min_df sono invece stati scelti in base alla distribuzione delle "frequenze degli n-grammi nei documenti" del train set.

Per continuare l'esplorazione, si è cercato di visualizzare come i dati sono raggrupati in categorie riducendo lo spazio delle features con tSNE, l'approccio iniziale è consistito nell'utilizzare la term document matrix come insieme di variabili esplicative, seguendo la pipeline: articoli preprocessati |Term Document Matrix |tSNE a 3 componenti.

Si vede in Figura 1 come articoli dello stesso tema sono abbastanza distaziati tra loro e sparsi nell'agglomerato di articoli.

Successivamente si è riprovato attraverso la stessa pipeline con in più la Latent Dirichlet (LDA), impostando come parametri n_components [8] a 6 e learning_decay[9] al valore di default.

[4] formato di serializzazione per dati

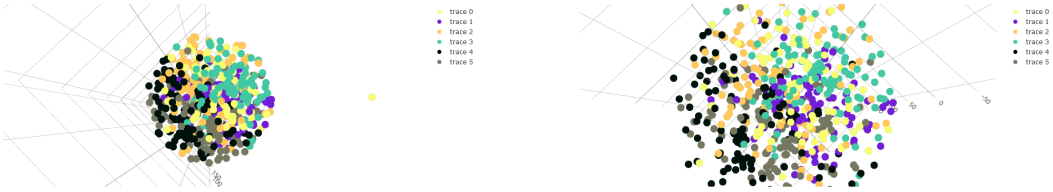
[5] Dataset scaricabili in formato json all'indirizzo: *https* : *//github.com/mastershef/big_data*

[6] Utilizzando le stopword dalla libreria TextWiller github.com/livioivil/TextWiller

[7] è una matrice che ha sulle colonne le singole parole (i) e sulle righe i dicumenti (j), le singole celle sono non negative e contano quante volte la parola i è presente nel documento j, calcolata attraverso *sklearn.feature_extraction.text.CountVectorizer*

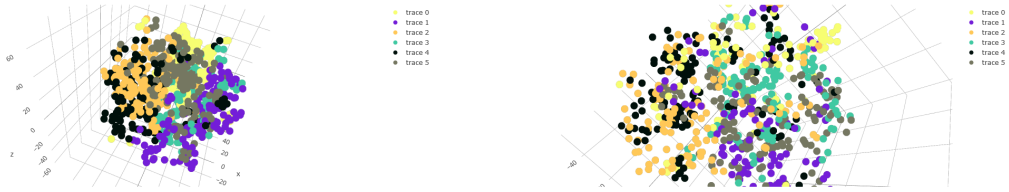
[8] numero di topic latenti che LDA dovrebbe individuare

[9] parametro usato per regolare il "learning rate", che si consiglia imposare nel range (0.5, 1] per garantire la convergenza asintotica.



(a) Vista normale (b) Vista ravvicinata

Figura 1: Riduzione della term document matrix da forma (2400, 4640) a forma (2400, 3) via tSNE.



(a) Vista normale (b) Vista ravvicinata

Figura 2: Riduzione della matrice da forma (2400, 4640) a forma (2400, 3).

A differenza della Figura 1 in questa figura (esplorazione interattiva [10]) si vede come i diversi temi sono raggruppati in piccoli cluster sparsi in maniera più o meno uniforme lungo i tre assi, inoltre articoli dello stesso tema nella Figura 2 sembrano essere meno distanti tra loro. Sempre dalla Figura 2 si evince come un albero potrebbe essere una soluzione accettabile per il problema di classificazione.

3.3 Prove

Si è deciso di usare 3 configurazioni attraverso le quali efettuare le analisi, se ne possono vedere alcune caratteristiche nella Tabella 1.

Pipeline \Trasformation	LDA-12	LDA-48	Term Frequency
T.D Matrix	✓	✓	✓
LDA	✓	✓	x
Classifier	✓	✓	✓

Tabella 1: Configurazioni LDA-12 e LDA-48, indicano i modelli fittati con 12 e 48 componenti.

Prima di iniziare le prove sono stati ottimizzati, per ogni configurazione, alcuni parametri dell'albero di decisione (profondità massima e numero minimo di osservazioni per foglia.) effettuando una ricerca a griglia e massimizzando l'accuratezza sull'inisieme di validazione. La prima prova effettuata riguarda la misurazione dell'accuratezza nel classificare al variare della dimensione dell'inisieme di training.

[10] https://plot.ly/create/?fid=cristi.gutzu:5&fid=cristi.gutzu:6

Dalla figura 3, si nota che tutte e tre le configurazioni hanno un andamento piuttosto simile se non quando l'insieme di training è abbastanza ristretto, come si vede a livello 15% in questi casi le configurazioni con più features sembrano funzionare meglio.

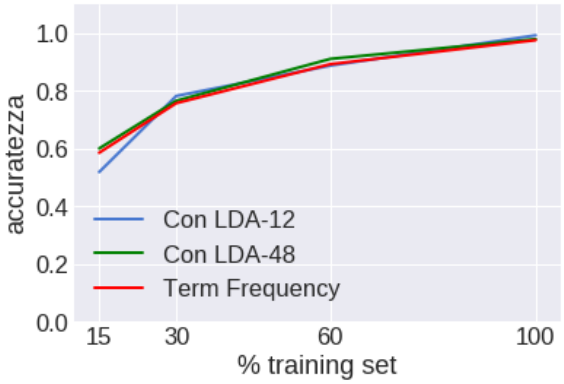


Figura 3: Performace modelli di classificazione sul test set in funzione della dimensione del training set.

Successivamente sono state misurate le performance delle 3 configurazioni usando l'intero insieme di training, sia dal punto di vista dell'accuratezza[11] che dal punto di vista dei tempi di esecuzione. I risultati sono riassunti nelle tabelle 2 e 3.

Rappresentazione	Accuratezza
Dummy	17.8 %
LDA-12	**99.2 %**
LDA-48	97.8 %
Term Frequency	98.5 %

Tabella 2: Risultati utilizzando i diversi modelli di rappresentazione con l'intero training set.

Rappresentazione	Tempi trasf.	Tempi class.
LDA-12	99.55 s	**0.02 s**
LDA-48	141.54 s	0.07 s
Term Frequency	**5.29 s**	1.93 s

Tabella 3: Tempi misurati sulla trasformazione delle variabili e sulla classificazione.

Come si nota in Tabella 2, tutte le rappresentazioni superano abbondantemente la soglia definita dalla baseline, in particolare, la LDA-12 risulta performare meglio dal punto di vista dell'accuratezza classificando gli articoli dell'insieme di test.

[11] calcolata sull'insieme di test

Siccome l'accuratezza, che misura la qualità complessiva del modello, risulta molto buona, ci si è chiesti se il classificatore, con la configurazione migliore, si comporta altrettanto bene anche per le singole classi, per verificarlo abbiamo calcolato ulteriori misure, riportate in Tabella 4.

Categoria	precision	recall	f1	support
Cronaca	1.00	1.00	1.00	107
Cultura	0.98	0.98	0.98	88
Economia	1.00	0.98	0.99	99
Politica	1.00	0.98	0.99	107
Sport	0.98	1.00	0.99	96
Tech	0.99	1.00	1.00	103
micro avg	0.99	0.99	0.99	600
macro avg	0.99	0.99	0.99	600
weighted avg	0.99	0.99	0.99	600

Tabella 4: Report metriche sul classificatore stimato con LDA-12.

Si osserva dalla tabella che le metriche: precisione, richiamo, f1; sono molto alte per tutte le classi e quindi in accordo con l'accuratezza confermano la validità del modello utilizzato.

4 Conclusioni

Dai risultati delle analisi si vede che il modello basato sulla rappresentazione con LDA è comparabile al modello basato sulla rappresentazione con Term Frequency dal punto di vista dell'accuratezza, inoltre se confrontiamo i tempi di esecuzione, presenti nella Tabella 3, notiamo che il guadagno otteuto riducendo la dimensione delle features è annullato dal tempo che è necessario per trasformare i dati; ciò è valido perlomeno nell'ambiente python con le funzioni e classi della libreria scikitlearn che sono state utilizzate, concludiamo dicendo che riteniamo sarebbero opportune ulteriori analisi utilizzando pacchetti che ottimizzino il processo di stima del modello LDA.

Nota: il dataset è composto soprattutto da articoli caricati intorno ai mesi di maggio e giugno 2019 e perci, vista la variabilità dei termini, dovuta al tempo, il modello adattato con questo insieme di training potrebbe non dare risultati altrettanto buoni se utilizzato per classificare articoli troppo distanti nel tempo.

Riferimenti bibliografici

1. David m. Blei, Andrew Y, Ng, and Michael I. Jordan, 2013, Latent Dirichlet Allocation
2. Hastie T., 2016, Introduction to Statistical Learning, Decision Trees
3. Ceron, Curini, Iacus, 2014, Social Media e Sentiment Analysis.