

Classificazione articoli ANSA via topic modelling

Alessandro Stefani¹ e Cristi Gutu²

¹ Corso di laurea in Statistica per le tecnologie e le scienze, nome d'arte Aleste 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 concentreremo sul confrontare le prestazioni del classificatore utilizzando diverse rappresentazioni dei documenti.

In particolare si confronteranno 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, oltre a questo esistono approcci basati su modelli probabilistici che permettono di ridurre notevolmente la dimensione dello spazio delle features come LDA.

Quindi in questa relazione si vogliono paragonare i risultati dei modelli di previsione usando entrambe le tecniche di rappresentazione via alberi decisionali.

Per quantificare le prestazioni del classificatore sono state prese in considerazione l'accuratezza³, per valutare la qualità delle previsioni, e i tempi di esecuzione.

Si è scelta l'accuratezza in quanto 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 5 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 tramite il motore di ricerca DuckDuckGo. Si è deciso di includere 6 macro categorie: (Economia, Politica, Cultura, Sport, Tecnologia, Cronaca), escludendo la categoria Mondo perché 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`

³ L'accuratezza è il valore definito dall'espressione
$$\left(\frac{\text{Previsioni_effettuate_correttamente}}{\text{Totale_tentativi_previsione}} \right)$$

- site:ansa.it/sito/notizie/cultura
- 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 e' composto da: (titolo, sottotitolo, testo, tags, categoria). Le 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 JSON⁴.

In totale si sono raccolti 2400 articoli⁵, che sono stati suddivisi casualmente in training set, validation set e test set.

3 Esperimenti e risultati

Prima di poter utilizzare modelli per l'analisi testuale è necessario preprocessare i dati, questo e' stato fatto costruendo la seguente pipeline di preprocessing: articoli |rimozione stopwords⁶ |stemming |rimozione tags html |rimozione punteggiatura |rimozione numeri |rimozione link.

3.1 Baseline

Come baseline si e' deciso di classificare utilizzando il modello *DummyClassifier* che classifica ogni articolo secondo la categoria piu' frequente del training set. L'accuratezza ottenuta e' del: 15%.

3.2 Analisi esplorative

Si e' notato che le categorie assegnate agli articoli erano in realta' micro-categorie, percio' un ultimo step di preprocessing e' stato riclassificare manualmente gli articoli, etichettandoli con le corrispondenti macro-categorie.

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

Successivamente si e' calcolata la term-document matrix⁷ la cui classe di supporto in Python da' la possibilita' di specificare 3 parametri: ngram_range, min_df, max_df; per scegliere il range di ngrammi da prendere in considerazione si e' fatto riferimento al libro Social Media e Sentiment Analysis[3] dove si suggerisce che generalmente n-grammi con piu' di 3 termini non aggiungono contenuto informativo, i parametri max_df, min_df sono invece stati scelti in base alla distribuzione delle "frequenze degli ngrammi nei documenti" del train set.

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

⁴ formato di serializzazione per dati

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

⁶ Utilizzando le parole dalla libreria TextWilla github.com/livioivil/TextWilla

⁷ e' una matrice che ha sulle colonne le singole parole e sulle righe 0 o 1 che indicano se la parola e' presente nel documento alla riga i oppure no e se' presente piu' volte nella cella e' memorizzato il numero di volte che compare in quel documento, calcolato attraverso `sklearn.feature_extraction.text.CountVectorizer`

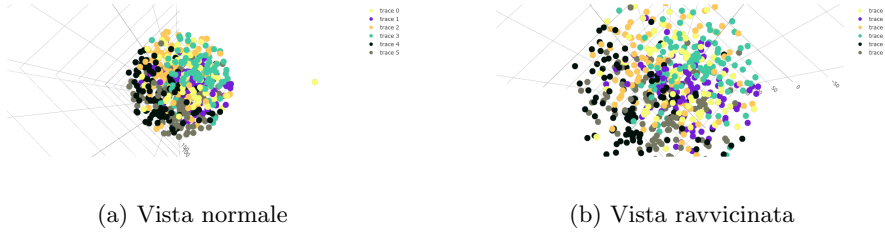


Figura 1: Riduzione della term document matrix da forma $(2400, 4640)$ a forma $(2400, 3)$ via tSNE.

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

Successivamente si e' provato attraverso la stessa pipeline, applicando in piu' Latent Dirichlet Allocation (LDA), impostando come parametri $n_components$ ⁸ a 6 e $learning_decay$ ⁹ al valore di default.

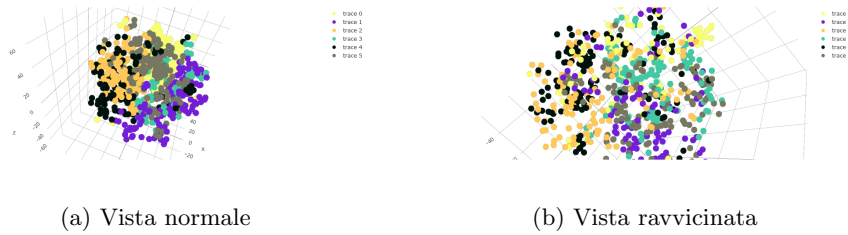


Figura 2: Riduzione della matrice da forma $(2400, 4640)$ a forma $(2400, 3)$.

A differenza della Figura 1 in questa figura (esplorazione interattiva¹⁰) si vede come i diversi temi sono raggruppati in piccoli cluster sparsi in maniera piu' 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 e' deciso di usare 3 configurazioni attraverso le quali effettuare le analisi, se ne possono vedere alcune caratteristiche nella Tabella 1.

Prima di iniziare le prove sono stati ottimizzati, per ogni configurazione, alcuni parametri dell'albero di decisione (profondita' massima, e numero minimo di osservazioni per foglia.) effettuando una ricerca a griglia, massimizzando l'accuratezza sull'insieme di validazione.

La prima prova effettuata riguarda la misurazione dell'accuratezza nel classificare al variare della dimensione dell'insieme di training.

⁸ numero di topic latenti che LDA dovrebbe individuare

⁹ parametro usato per regolare il learning rate; che si consiglia imporre nel range $(0.5, 1]$ per garantire la convergenza asintotica.

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

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.

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

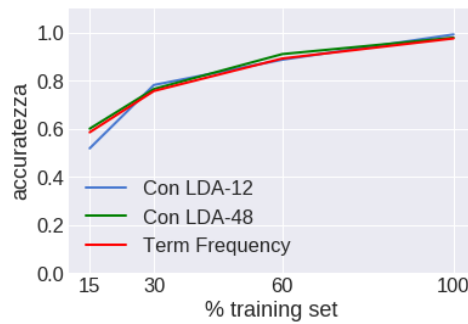


Figura 3: Performace modelli di classificazione sul test set in funzione della dimensione del training set.

Successivamente sono state misurate le performance delle 3 configurazioni usando l'intero insieme di training, sia dal punto di vista dell'accuratezza¹¹ 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.

Come riportato in Tabella 1, le rappresentazioni LDA-12 e Term Frequency hanno pari accuratezza, e sono i modelli migliori tra quelli che abbiamo stimato, classificando gli articoli dell'insieme di test.

Siccome l'accuratezza, che misura la qualita' complessiva del modello, risulta molto buona, ci si e' chiesto se il classificatore si comporta altrettanto bene anche per le singole classi, per verificarlo abbiamo calcolato ulteriori misure per il modello migliore.

¹¹ calcolata sull'insieme di test

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.

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 dalle tabelle che le metriche: precisione, richiamo, f1; sono molto alte per ogni classe, cosa che conferma l'accuratezza elevata ottenuta sul training set.

4 Conclusioni

Dai risultati delle analisi si vede che il modello basato sulla rappresentazione con LDA e' comparabile al modello basato sulla rappresentazione con Term Frequency dal punto di vista dell'accuratezza, inoltre se confrontiamo i tempi di esecuzione notiamo che il guadagno ottenuto riducendo la dimensione delle features e' annullato dal tempo che e' necessario per trasformare i dati, nell'ambiente python sfruttando le funzioni della libreria scikitlearn, concludiamo dicendo che riteniamo opportuni ulteriori utilizzando pacchetti che ottimizzino il processo di stima del modello LDA.

Riferimenti bibliografici

1. David m. Blei, Andrew Y, Ng, and Michael I. Jordan, Latent Dirichlet Allocation, 2003
2. Hastie T., 2016, Introduction to Statistical Learning , Decision Trees
3. Ceron, Curini, Iacus, 2014, Social Media e Sentiment Analysis.