GENERAZIONE DELLE CATEGORIE DI WIKIPEDIA ATTRAVERSO IL CLUSTERING

CAZZARO DALLA CIA LOVISOTTO VIANELLO

INTRODUZIONE

- descrizione del problema in generale e del constesto
  - delineazione degli obiettivi:
  - gli articoli di wikipedia sono clusterizzabili?
  - rapporto tra cluster e le categorie?
  - cosa succede con varie tecniche di clustering?

DATASET E ANALISI PRELIMINARE

Il dataset che abbiamo utilizzato per effettuare questa analisi non è altro che un *dump* di wikipedia. Tale *dump* è composto da un JSON contenente centomila articoli di Wikipedia in lingua inglese. Per ogni articolo di Wikipedia abbiamo a disposizione il titolo, il testo, l'id e le categorie assegnate all'articolo dai suoi autori.

La nostra analisi mira a valutare la relazione semantica tra gli articoli e le loro categorie, quindi prima di effettuare l'analisi si è deciso di eliminare tutti gli articoli che non risultano essere associati a nessuna categoria. Queste pagine sono dette *di disambiguazione* e quindi non sono utili per i nostri scopi.

//todo analisi delle categorie (sort e distribuzione delle categorie)

RAPPRESENTAZIONE DEL DATASET

Prima di procedere con qualsiasi operazione sull'intero corpus si è deciso di preprocessare il data set eliminando le cosiddette *stop words* presenti nel testo e lemmatizzando le parole rimaste.

Per l'eliminazione delle *stop words* ci siamo basati su una lista di parole fornita dal sito http://www.ranks.nl/stopwords. Questi termini vengono filtrati dal corpus in quanto portano un contenuto informativo sull'argomento dell'articolo pressoché nullo.

La lemmatizzazione invece é stata eseguita utilizzando il `Lemmatizer` di spark al fine di raggruppare assieme le variazioni semantiche delle parole.

VETTORIALIZZIONE DEGLI ARTICOLI

Per interagire con i più comuni algoritmi di clustering, ad esempio K-means, si è reso necessario trasformare gli articoli di Wikipedia in vettori. Per fare ciò abbiamo adottato una tecnica nota in letteratura con il nome Word2Vec. Tale algoritmo ideato da Tomas Mikolov non è altro che una rete neurale a due strati il cui scopo è quello di trasformare parole del linguaggio naturale in vettori. Nello spazio vettoriale generato le parole semanticamente più simili saranno più vicine, viceversa parole semanticamente diverse risulteranno distanti.

Tale funzionalità è già implementata nella suite software di Spark, per sfruttarla è necessario inizializzare alcuni parametri di settaggio. Tra questi uno dei più importanti è sicuramente la dimensione del vettore in uscita. In letteratura si è valutato che una dimensionalità nel ordine dei 100/300 [1] è sufficiente a rappresenta un buon compromesso in termini di performance. Uno volta settati i parametri l'algoritmo di Word2Vec necessita di essere allenato. Tale allenamento è stato fatto su tutto il corpus.

Per trasformare un articolo è stato sufficiente effettuare la media vettoriale di tutte le parole presenti nel testo di un'articolo. Tale operazione è stata eseguita attraverso il BLAS (Basic Linear Algebra Subprograms) di Spark per eseguire tali conti nella maniera più efficiente possibile.

BAG OF WORDS

L'algoritmo Bag of Words[3] effettua la conversione degli articoli in vettori considerando le occorrenze dei termini in essi. In particolare viene considerata come metrica la *Term frequency-inverse document frequency* (Tf-Idf).

Si definisce la Tf-Idf relativa all'i-esimo termine nel j-esimo articolo come:

$$w_{i,j} = tf_{i,j} \times \log(\frac{N}{df_i})$$

dove $tf_{i,j}$ rappresenta il numero di occorrenze di i in j, df_i è il numero di documenti contenenti il termine i, ed N è il numero totale di documenti.

Abbiamo scelto di limitare l'analisi alle 3000 parole più frequenti nel dataset (escludendo le stopwords).

Pertanto il modello prodotto associa ad ogni documento un vettore di dimensione 3000, contenente gli indici TfIdf associati ad ogni articolo per tutte le parole selezionate.

CLUSTERING

Tecniche di Clustering

Hopkins Statistic

Riportiamo lo score che ci dice che il nostro dataset è ben clusterizzabile

Kmeans

Kmeans e il suo score con un bel grafico

Altri metodi

Abbiamo provato anche il clustering gerarchico e il Gaussian Mixture Model ma non abbiamo abbastanza potenza di calcolo

Latent Dirichlet Allocation

Vediamo cosa viene fuori e un bel grafico

Valutazioni del Clustering

Simple Silhouette

Utilizzo della versione semplificata di Silhouette con i centroidi Rimozione dei cluster con un solo articolo dal punteggio Magari buttiamoci un peso a sta metrica

Normalized Mutual Information

L'informazione mutua è una quantità che misura la mutua dipendenza di due variabili aleatorie, ovvero quanta informazione porta sull'altra la conoscenza del valore di una delle due.

Questa misura può essere impiegata per valutare quanto due partizioni, o *clustering*, concordano nel suddividere un set di punti [2].

Per fare questo, ad ogni cluster è stata associata una variabile indicatrice ω, che assume valore 1 se il punto considerato appartiene al cluster e 0 altrimenti. Ogni clustering viene perciò individuato dall'insieme Ω di queste variabili aleatorie mutualmente esclusive e a somma unitaria.

Con questa descrizione del problema è possibile calcolare l'informazione mutua tra due distinti clustering Ω e Φ. Questa matrica è stata normalizzata in $(0, 1)$ per garantire un confronto alla pari tra clustering di dimensione diversa.

NMI viene quindi definita come

$$\mathrm{NMI}(\Omega, \Phi) = \frac{\mathrm{I}(\Omega, \Phi)}{[\mathrm{H}(\Omega) + \mathrm{H}(\Phi)]/2}$$

dove

$$\begin{aligned} \mathrm{I}(\Omega, \Phi) &= \sum_{\omega \in \Omega} \sum_{\phi \in \Phi} \mathrm{P}(\omega \cap \phi) \log \frac{\mathrm{P}(\omega \cap \phi)}{\mathrm{P}(\omega)\mathrm{P}(\phi)} \\ \mathrm{H}(\Omega) &= -\sum_{\omega \in \Omega} \mathrm{P}(\omega) \log \mathrm{P}(\omega) \\ \mathrm{H}(\Phi) &= -\sum_{\phi \in \Phi} \mathrm{P}(\phi) \log \mathrm{P}(\phi) \end{aligned} \tag{1}$$

All'atto pratico, come valore delle probabilità sono stati impegate stime a massima verosimiglianza, per esempio

$$\mathrm{P}(\omega) = \frac{\text{numero di punti in } \omega}{\text{numero di punti totali}}$$

.

Purtroppo questa definizione non è direttamente applicabile al confronto tra cluster e categorie perché, mentre i clustering ottenuti con K-means e LDA sono delle effettive partizioni del dataset, non si può dire lo stesso delle categorie, dato che un articolo può possederne più di una.

Il primo approccio per scogliere questo nodo è stato quello di eseguire un *ranking* con *Inverse Document Frequency* tra le categorie di ciascun articolo per eleggere la più rappresentativa. Grazie a questo passaggio NMI viene calcolata come dalla sua definizione (equazione 1).

Il secondo approccio tenta invece di estendere NMI al caso di cluster che si sovrappongano l'uno all'altro, considerando quindi ogni classe c con la sua complementare $\bar{c}$ una partizione dell'insieme dei punti. L'informazione mutua viene calcolata quindi per ogni clustering $C = \{c, \bar{c}\}$ e si valuta la loro somma.

A causa di questa passaggio si perde la normalizzazione tra 0 e 1: i risultati sono comunque confrontabili, perché risultano tutti multipli di un fattore che è funzione della distribuzione delle categoerie, quindi indipendente dalle tecniche di clustering.

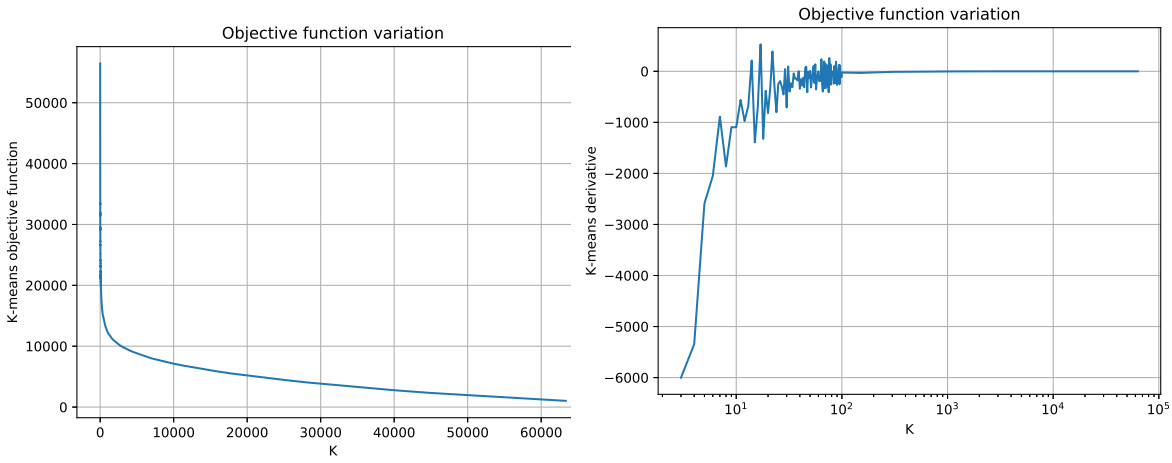
RISULTATI

Numero di cluster

Il clustering K-means risulta il più semplice e rapido da ottenere e per questo motivo abbiamo deciso di ispezionare un ampio range di valori per K, numero di cluster.

La funzione obiettivo cala continuamente al variare del numero di cluster, come da figura 1: questa osservazione è confermata dall'andamento della derivata della funzione obiettivo rispetto a K. Essa raggiunge un tasso di incremento trascurabile in prossimità del valore K=100.

Perciò con la tecnica di clustering LDA ci siamo concentrati su valori di K compresi tra 50 e 150.



(a) Funzione obiettivo

(b) Derivata della funzione obiettivo

Figure 1: La funzione obiettivo smette di calare in modo significativo tra 50 e 150.

Figure 2: Silhouette

Validazione con Simple Silhouette

Confronto tra i Cluster ottenuti con Normalize Mutual Information

Figure 3: Informazione mutua tra K-means e il clustering indotto dalle categorie.

CONCLUSIONI

Le conclusioni generali dalle analisi effettuate Proposte di punti da approfondire in studi futuri

REFERENCES

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. ICLR Workshop, 2013.

[2] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008

[3] Harris, Zellig S. "Distributional structure." Word 10.2-3 (1954): 146-162.

[4] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research 3.Jan (2003): 993-1022.