CLUSTERING DI WIKIPEDIA E RELAZIONE CON LE CATEGORIE

CAZZARO DALLA CIA LOVISOTTO VIANELLO

1 INTRODUZIONE

Per il progetto del corso di Data Mining abbiamo deciso di svolgere la traccia proposta B, che proponeva di indagare fino a che punto un clustering sulle pagine di Wikipedia è consistente con le categorie associate alle pagine stesse. Per fare ciò avevamo a disposizione due dataset, di differenti grandezze, formati da articoli di Wikipedia in lingua inglese.

A partire da tali premesse ci siamo prefissati i seguenti quesiti:

- quanto sono clusterizzabili gli articoli del dataset?
- le categorie associate a questi articoli sono sensate?
- quali sono dei metodi validi per rappresentare opportunamente i nostri dati in input?
- come variano i cluster utilizzando tecniche diverse?
- c'è un numero ottimale di cluster per il dataset? Quanto vale?
- qual è il rapporto tra il cluster ottenuto e le categorie?

Tali quesiti troveranno risposta nel seguito della trattazione, che è strutturata nel modo che segue. La sezione 2 illustra l'analisi preliminare effettuata sugli articoli e sulle categorie. Successivamente il capitolo 3 descrive le tecniche utilizzate per rappresentare il dataset. Il capitolo 4 tratta poi degli algoritmi di clustering applicati al dataset e delle tecniche di valutazione impiegate. Di seguito la sezione 5 riporta e discute i risultati ottenuti nel capitolo precedente, offrendo anche un confronto tra i metodi utilizzati. Infine la sezione 6 riassume il lavoro svolto, ciò che abbiamo ottenuto e propone alcuni spunti di ricerca per futuri sviluppi.

2 DATASET E ANALISI PRELIMINARE

Ci sono stati messi a disposizione due dataset, di differenti grandezze, per effettuare questa analisi. Nello specifico abbiamo utilizzato quello minore, il quale è un *dump* composto da un JSON contenente centomila articoli di Wikipedia in versione inglese. Per ogni articolo abbiamo a disposizione il titolo, il testo, l'id e le categorie dell'articolo. In particolare le categorie sono assegnate a discrezione degli autori e dei successivi revisori di ogni voce, in quanto Wikipedia non prevede una struttura o dei vincoli particolari per l'assegnazione di queste categorie.

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.

Living people	14994	Association football midfielders	611
Year of birth missing (living people)	1172	Association football defenders	502
Place of birth missing (living people)	751	Association football forwards	409
American films	741	Year of birth unknown	404
English-language films	691	English Football League players	398

Table 1: Le categorie con maggior numero di articoli associati

Per quanto concerne le categorie, una prima ispezione manuale rivela che esse sono alquanto arbitrarie e spesso così specifiche da essere associate ad un articolo soltanto. Esempio di ciò sono le categorie "Roads on the National Register of Historic Places in Illinois", "United Nations Security Council resolutions concerning Sudan" e "Singaporean people of Yemeni descent". C'è inoltre da sottolineare come ad ogni articolo siano spesso associate più di una categoria, con una media di 1.99 categorie per ogni voce del dataset.

Un'analisi più approfondita sulla distribuzione delle categorie è riportata in figura 1.

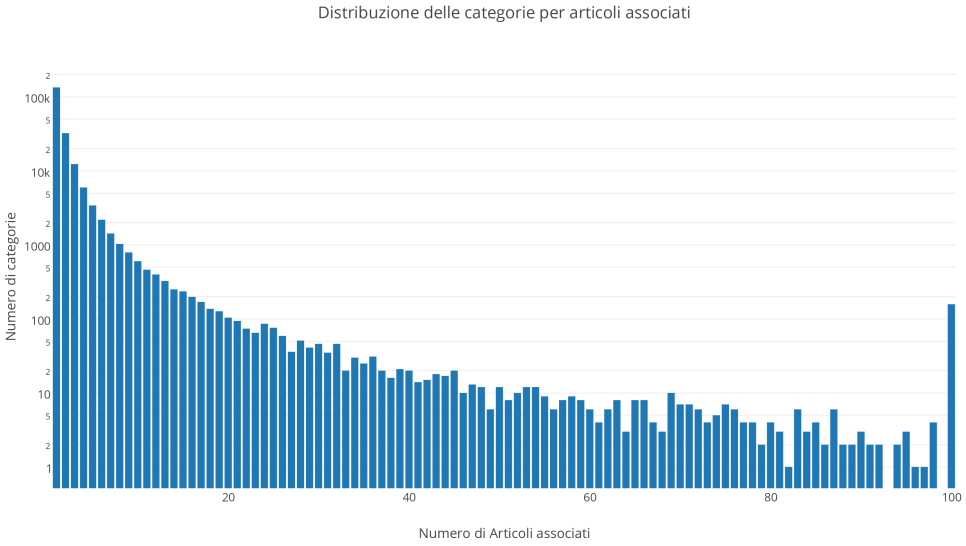

Figure 1: Distribuzione delle categorie. Nota: per rendere il grafico leggibile il picco finale raggruppa il numero di categorie con cento o più articoli.

L'analisi effettuata rivela infatti che, sul totale di 198609 distinte categorie presenti nel dataset, circa due terzi (precisamente 134524) di queste sono uniche, ovvero sono associate ad un solo articolo. Dato che queste non portano alcun contenuto informativo utile alla clusterizzazione, abbiamo provveduto ad escludere tali categorie dalle successive analisi.

Nondimeno è anche presente un numero ristretto di 158 categorie associate a 100 o più articoli, le quali sono composte per lo più da categorie del genere "1900 deaths" e "1900 births" che si ripetono variando solamente l'anno. In tabella 1 sono riportate a titolo di esempio le categorie con più articoli associati.

3 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 è stata eseguita utilizzando il Lemmatizer di Spark al fine di raggruppare assieme le variazioni semantiche delle parole.

3.1 Vettorializzazione 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 strumento, 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, mentre parole che esprimono concetti differenti risulteranno distanti.

Tale funzionalità è già implementata nella suite software di Spark, ma per sfruttarla al meglio è necessario settare alcuni parametri. Tra questi uno dei più importanti è la dimensione del vettore in uscita. In letteratura si è valutato che una dimensionalità nel ordine dei 100/300[1] risulta un buon compromesso tra prestazioni e capacità di descrivere il dataset.

Dopo aver allenato il modello *Word2Vec* sul nostro corpus di testi, trasformando così le singole parole in vettori, ad ogni articolo è stata associata la media vettoriale di tutte le parole presenti nel suo testo. Il *Basic Linear Algebra Subprograms* di Spark ci ha permesso di eseguire questa operazione molto rapidamente.

3.2 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:

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

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.

4 CLUSTERING

4.1 Tecniche di Clustering

4.1.1 *Hopkins Statistic*

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

4.1.2 *K-means*

Dato un insieme di punti $P \subset M$, dove M è uno spazio metrico indotto da una funzione distanza d, l'algoritmo K-means punta a costruire una partizione $C = \{C_1, ..., C_K\}$ con centri dei cluster $\{c_1, ..., c_K\}$ che minimizzi la seguente funzione obiettivo.

$$\Phi_{K-means}(C) = \sum_{i=1}^{K} \sum_{p \in C_i \cap P} d(p, c_i) \tag{1}$$

Per il nostro progetto abbiamo utilizzato l'implementazione Java *KMeans* di Spark *mllib*, reperibile online.

4.1.3 *Latent Dirichlet Allocation*

La Latent Dirichlet Allocation (LDA)[4] é un modello generativo statistico che considera ogni documento come il risultato della sovrapposizione di più topic, ognuno dei quali é caratterizzato da un insieme di parole. Tale processo é stato applicato alla rappresentazione per bag of words degli articoli.

Analogamente a k-means puó essere utilizzata come algoritmo di clustering unsupervised associando gli articoli al topic avente influenza più alta nel loro corpus.

Tra i vantaggi principali dell'utilizzo di questa tecnica applicata all'analisi testuale vi é sicuramente la possibilitá di caratterizzare il tema dei topic delineati esaminando le parole con incidenza più alta in ogni gruppo.

Nel nostro lavoro abbiamo scelto di usare 100 iterazioni per l'algoritmo in quanto oltre tale numero i miglioramenti nella qualitá del clustering non sono rilevanti [7].

Come metrica che definisce la bontá della categorizzazione abbiamo scelto la log-likelihood [6].

Data la complessitá computazionale di questo metodo ci siamo limitati ad effettuare gli esperimenti con un numero di topic $k = 80$, seguendo l'indicazione offerta dalle tecniche presentate in precedenza.

4.1.4 *Altri metodi*

La libreria *mllib* contiene altre tecniche, tra cui *Gaussian mixture* e *Bisecting k-means*, ma esse si sono rivelate inapplicabili, la prima per l'eccessivo tempo di calcolo e la seconda perché affine al già citato *K-means*, ottimizzando infatti la stessa funzione obiettivo.

4.2 Valutazioni del Clustering

4.2.1 *Simplified Silhouette*

Silhouette è un metodo unsupervised per interpretare e validare un dato cluster. In particolare questa tecnica calcola per ogni punto la distanza media dal proprio cluster e la minima distanza media dagli altri cluster, valutando così sia la coesione che la separazione dei cluster, e restituisce un valore compreso tra -1 e 1.

Tale approccio però richiede tempo $\mathcal{O}(n^2)$, dove n è la taglia dell'input, e il calcolo diretto risulta impraticabile a causa della taglia del dataset e le risorse di calcolo a nostra disposizione.

Pertanto abbiamo provveduto a implementare e applicare una versione modificata che approssima questo l'algoritmo, denominata *Simplified Silhouette*[8].

Indicando con a la distanza tra il punto considerato e il centro del proprio cluster e con b la minima distanza tra il punto e gli altri centri del cluster, il Simplified Silhouette Coefficient è così definito:

$$SSC = \frac{b - a}{max(a, b)} = \frac{b - a}{b} \tag{2}$$

Si nota che il denominatore viene semplificato nel secondo passaggio. Ciò è possibile perché, per definizione dell'algoritmo *K-means*, la distanza euclidea tra il punto e il suo centro sarà sempre minore della distanza tra il punto e un qualsiasi altro centroide.

Questa approssimazione permette quindi di ottenere una buona stima di *Silhouette*[9] riducendo la complessità dell'algoritmo a $\mathcal{O}(Kn)$, dove K è il numero di cluster e n il numero di punti in input.

Tale fattore di miglioramento delle prestazioni si indebolisce però all'aumentare di K, ritornando per $K \to n$ a una complessità quadratica. Nel caso generale tuttavia si cerca un numero di cluster minore di n e *Simplified Silhouette* porta ad un apprezzabile miglioramento di performance.

Considerando la media dei coefficienti SSC sui vari punti del dataset si ottiene infine un'indicazione globale sulla bontà del clustering.

Un'ulteriore osservazione è il fatto che, con l'aumentare di K, il punteggio calcolato sia con *Silhouette* che con *Simplified Silhouette* tende a 1 (il miglior punteggio possibile) e raggiunge effettivamente tale valore con $K = n$. In questo caso estremo ogni punto costituisce un cluster a sè stante e perciò a si annulla portando a 1 il valore di SSC.

Indicando con d la distanza media tra il punto e il suo cluster e con $min(d)$ la minima distanza media tra il punto e gli altri cluster, si ha infatti che

$$\text{Silhouette} = \frac{min(d) - d}{max(d,\, min(d))} \quad \xrightarrow[K \to n]{} \quad 1 \tag{3}$$

Questo rende difficoltoso usare *Silhouette* come strumento per individuare un K ottimale, perché tende a provilegiare K elevati. Tale problematica può essere affrontata introducendo un peso al coefficiente restituito da Silhouette in maniera da evitare questa convergenza a 1. Questa indagine esula tuttavia dagli scopi di questo lavoro non abbiamo trattato oltre questo punto che tuttavia può essere oggetto di studi futuri.

4.2.2 *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. Essa 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

$$NMI(\Omega, \Phi) = \frac{I(\Omega, \Phi)}{[H(\Omega) + H(\Phi)]\,/2} \tag{4}$$

dove

$$I(\Omega, \Phi) = \sum_{\omega \in \Omega} \sum_{\phi \in \Phi} P(\omega \cap \phi) \log \frac{P(\omega \cap \phi)}{P(\omega) P(\phi)}$$

$$H(\Omega) = -\sum_{\omega \in \Omega} P(\omega) \log P(\omega), \quad H(\Phi) = -\sum_{\phi \in \Phi} P(\phi) \log P(\phi)$$

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

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

.

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* (vedi sezione 3.2) tra le categorie di ciascun articolo per eleggere la più rappresentativa. Grazie a questo passaggio NMI può essere calcolata direttamente dalla sua definizione (equazione 4).

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.

I risultati sono presentati nella sezione seguente.

5 RISULTATI

5.1 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 2, ma il calo diventa sempre più trascurabile al crescere di K. Questa osservazione è confermata dall'andamento della derivata della funzione obiettivo: essa raggiunge un tasso di incremento sostanzialmente nullo in prossimità del valore K=100.

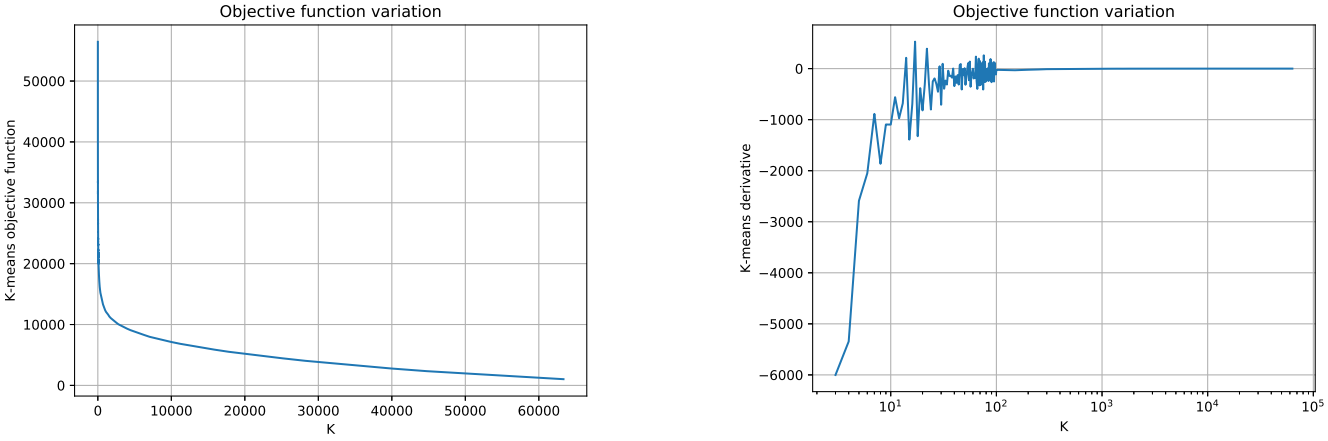

Figure 2: La funzione obiettivo smette di calare in modo significativo tra 50 e 150.

Per questo motivo nelle analisi successive ci siamo concentrati su valori di K compresi tra 50 e 150.

5.2 Confronto tra i Cluster ottenuti con Normalize Mutual Information

Come si puo notare in figura 3, l'andamento dell'informazione mutua del clustering rispetto alle categorie risulta analogo per entrambi gli approcci impiegati, descritti nella sezione 4.2.2.

L'aumento di informazione al crescre di K ci suggerisce che aumentando il numero di cluster si ottiene un clustering più vicino alla suddivisione indotta dalle categorie, ma che il miglioramento è tanto meno apprezzabile quanto più si sale con il valore di K.

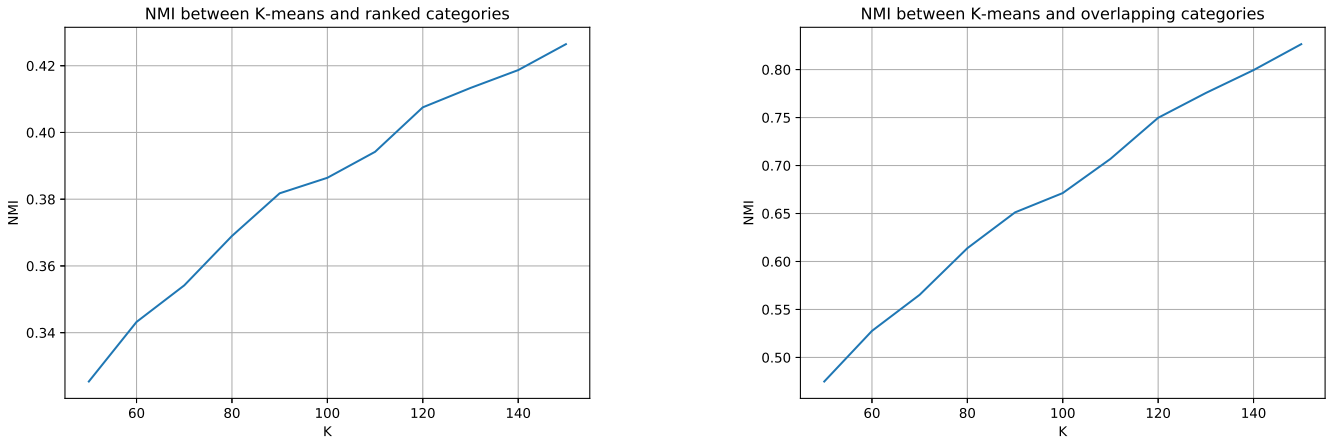

Figure 3: NMI tra K-means e il clustering indotto dalle categorie

5.3 Validazione con Simplified Silhouette

Figure 4: Silhouette

6 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.

[5] Chang, Jonathan, et al. "Reading tea leaves: How humans interpret topic models." Nips. Vol. 31. 2009.

[6] Griffiths, Thomas L., and Mark Steyvers. "Finding scientific topics." Proceedings of the National academy of Sciences 101.suppl 1 (2004): 5228-5235.

[7] Wei, Xing, and W. Bruce Croft. "LDA-based document models for ad-hoc retrieval." Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2006.

[8] Garcia, Kemilly Dearo, and Murilo Coelho Naldi. "Multiple parallel mapreduce k-means clustering with validation and selection." Intelligent Systems (BRACIS), 2014 Brazilian Conference on. IEEE, 2014.

[9] Eler, Danilo Medeiros, et al. "Simplified Stress and Simplified Silhouette Coefficient to a Faster Quality Evaluation of Multidimensional Projection Techniques and Feature Spaces." Information Visualisation (iV), 2015 19th International Conference on. IEEE, 2015.