Università degli studi di Trento

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea triennale in INFORMATICA

Tesi Finale

Clustrify: clustering non supervisionato di tweets

Relatore
Prof. Alberto Montresor

Laureando
Mattia Larentis

Anno accademico 2013-2014

Dedicata a Michela

Contenuti

Prefazione

> "È evidente che l'uomo sia un essere sociale più di ogni ape e più di ogni animale da gregge. Infatti, la natura non fa nulla, come diciamo, senza uno scopo: l'uomo è l'unico degli esseri viventi a possedere la parola."
>
> — Aristotele, Politica

L'uomo, in quanto essere sociale, per natura è portato al bisogno di comunicare. Dall'invenzione del telegrafo all'utilizzo di Internet è cambiato solamente il mezzo di divulgazione, ma non l'atto di voler condividere pensieri con il resto dell'umanità.

Questi pensieri vengono talvolta espressi sottoforma di testi, di dimensione variabile, che un individuo scrive nella speranza che da altri vengano letti. La grande mole di messaggi può essere elaborata per ricavarne informazioni utili, potenzialmente per la risoluzione di alcuni problemi [Fede - Perche‘ 'la grande mole'? Perche‘ 'potenzialmente'?].

Clusterify è un'applicazione web che risolve [mira alla possibilità di risolvere] più problemi contemporaneamente: dividere un ammasso di tweets in insiemi distinti, [Fede - questo mi sembra po triviale no? E‘ la definizione di clustering] [dove ognuno è caraterizzato da un forte legame che le proprie componenti hanno tra di loro, e da un legame debole nei confronti delle altre]. Questo processo può aprire le porte a molteplici risultati, come la possibilità di filtrare testi per una macro-area intessata, piuttosto che caraterizzare persone interessate ad un'attività per capire come meglio investire nel prossimo semestre.

Il mio progetto ha l'intento di mostrare la potenzialità del clustering non supervisionato per la suddivisione di testi. Nella fattispecie si ha intenzione di applicare degli algoritmi di clustering per definire questi sottogruppi ed, una volta composti, utilizzarli come strumento per filtrare gli stessi messaggi. Supponendo che l'utente medio di Twitter non parli solamente dell'argomento per cui è stato seguito, è possibile che questo scriva di materie al nostro utente interessanti o, al contrario, ininfluenti.

Clusterify, così facendo, permette di leggere solamente tweets per lui interessanti. [Ottimo, e un use-case molto utile per altro. Se quello e‘ un tuo obbiettivo pero‘, non conviene dare all'utente la possibilita di mettere un 'threshold' cosi‘ decide lui quanto tollera‘ falsi positivi versus falsi negativi?]

Chapter 1

Clusters e clustering

Il *clustering* è una tecnica di raggruppamento di oggetti in insiemi, secondo determinate regole, con l'intento che gli appartenenti allo stesso *cluster* siano più simili di quelli contentuti negli altri gruppi.

Queste regole tendono a minimizzare la *distanza* interna a ciascun gruppo ed a massimizzare quella tra i gruppi stessi: questa distanza viene quantificata per mezzo di misure di similarità.

La distanza è una qualsiasi funzione $d : X \times X \to \mathbb{R}$ che soddisfa:

$$
\begin{aligned}
&d(x, y) \geq 0 \\
&d(x, y) = 0 \iff x = y \\
&d(x, y) = d(y, x) \\
&d(x, y) \leq d(x, z) + d(z, y)
\end{aligned}
$$

Non esiste un solo algoritmo per raggruppare in modo corretto, bensì un solo compito da portare a termine. Questo obiettivo può essere raggiunto in più modi, basandosi su qualsivoglia tipo di nozione col solo fine di raggruppare in modo efficiente e sensato gli oggetti dati.

Esistono molteplici nozioni a cui ci si può ispirare per l'atto del raggruppamento; la creazione di gruppi caratterizzati da una piccola distanza tra i propri membri, piuttosto che l'utilizzo di particolari distribuzioni statistiche.

Seppur non esista non solo modo per fare clustering, molti di questi algoritmi godono di alcuni tratti in comune che definiscono i cosidetti *modelli*.

I modelli di clustering tipici sono:

- Clustering partizionale
- Clustering gerarchico
- Clustering density-based

1.1 Clustering partizionale

Gli algoritmi di clustering di questa famiglia creano una partizione delle osservazioni minimizzando la seguente funzione di costo:

$$\sum_{j=1}^{k} E(C_j)$$

ove k è il numero dei cluster richiesti in output, C_j è il j-esimo cluster e $E : C \to R^+$ è la funzione di costo associata al singolo cluster.

Questa tipologia di algoritmi solitamente richiede all'utente di specificare k, il numero di cluster distinti che si vogliono raggiundere a processo terminato, e mira ad identificare i gruppi naturali presenti nel dataset, generando una partizione composta da cluster disgiunti la cui unione ritorna il dataset originale.

1.1.1 Algoritmi conosciuti

Gli algoritmi più famosi appartenenti questa categoria sono:

- k-means
- k-medoids
- CLARANS

1.2 Clustering gerarchico

Gli algoritmi di clustering gerarchico, invece, creano una rappresentazione gerarchia ad albero dei cluster. Le strategie per il clustering gerarchico sono tipicamente di due tipi:

- Agglomerativo
- Divisivo

1.2.1 Metodo agglomerativo

Il metodo agglomerativo segue un approccio *bottom up* al problema dove, inzialmente, si ha un cluster per ogni oggetto e, successivamente, si procede all'unione di questi cluster, basando la selezione dei cluster da unire ad una *funzione di similarità*.

1.2.2 Metodo divisivo

Il metodo divisivo, invece, segue un approccio *top down* al problema dove, inizialemtente, si ha un unico cluster contente tutti gli oggetti e, via via, viene suddiviso in più sotto-cluster, basando la selezione del cluster da dividere ad una *funzione di similarità*. Solitamente si impone un numero minimo di elementi che ogni cluster deve contenere alla fine del processo.

1.2.3 Dissimilarità tra cluster

Nella maggior parte dei metodi di clustering gerarchico si fa uso di metriche specifiche che quantificano la distanza tra coppie di elementi e di un criterio di collegamento che specifica la dissimilarità di due insiemi di elementi (cluster) come funzione della distanza a coppie tra elementi nei due insiemi.

1.2.4 Metriche

La scelta della metrica influenza la forma dei cluster, poiché alcuni elementi possono essere più vicini utilizzando una data distanza e più lontani utilizzandone un'altra.

Le metriche comuni sono le seguenti:

- Distanza euclidea
- Distanza di Manhattan

1.2.5 Criteri di collegamento

Il criterio di collegamento specifica la distanza tra insiemi di elementi come funzione di distanze tra gli elementi negli insiemi. I criteri di collegamento comuni sono i seguenti:

- Complete linkage: calcola la distanza tra i due cluster come la distanza massima tra elementi appartenenti ai due clusters
- Minimum o single-linkage: calcola la distanza tra i due cluster come la distanza minima tra elementi appartenenti a cluster diversi
- Average linkage: calcola la distanza tra i due cluster come la media delle distanze tra i singoli elementi

1.2.6 Algoritmi conosciuti

Gli algoritmi più famosi appartenenti questa categoria sono:

- SLINK *(single-linkage)*
- CLINK *(complete-linkage)*

1.3 Clustering density-based

Negli algoritmi di clustering density-based il raggruppamento avviene analizzando l'intorno di ogni punto dello spazio, connettendo regioni di punti con densità sufficientemente alta.

1.3.1 Algoritmi conosciuti

Gli algoritmi più famosi appartenenti questa categoria sono:

- DBscan

1.4 Clustering distribution-based

Density-based clustering

Chapter 2

Twitter

twitter è un social network blablabla

2.1 Twitter API

twitter ha delle API blablabla

Chapter 3

dataTXT

dataTXT blabla

3.1 Funzionamento

funziona così...

3.2 dataTXT API

dataTXT ha delle API

3.3 dataTXT NEX

DataTXT NEX blabla

3.4 dataTXT REL

DataTXT REL blabla

Chapter 4

Clusterify

Clusterify blabla

4.1 Backend

funziona così...

4.2 Frontend

funziona così...

Chapter 5

Conclusioni e sviluppi futuri

5.1 Conclusioni

Conclusioni...

5.2 Sviluppi futuri

Sviluppi futuri

Ringraziamenti

Anche se appare solamente il mio nome sulla copertina di questa tesi, questa non sarebbe mai esistita senza l'aiuto di molte persone: è un piacere e un dovere render loro grazie.

Ringrazio il mio supervisore Alberto Montresor per il suo incoraggiamento e il supporto fornitomi durante la stesura di questa composizione.

Un ringraziamento speciale va riservato a Stefano Parmesan ed a Federico Vaggi, i quali hanno prestato molta pazienza e dedizione nel seguirmi ed impegno nell'aiutarmi.

Ho acquisito molte conoscenze circa lo sviluppo web durante le mie attività lavorative e alla passione che i miei colleghi sono stati in grado di trasmettermi; un grazie al team di SpazioDati e ai trenta3dev per il loro appoggio e per i bei momenti vissuti con loro.

Ringrazio mamma e papà in quanto mamma e papà.

Un grazie è tutto per Michela e per Carlotta che mi hanno spinto a dare del mio meglio anche quest'anno.

Infine vorrei ringraziare Michele Pittoni e tutti i miei compagni di corso che mi hanno supportato in questi tre anni di studio.

Bibliography

[1] Eric W. Weisstein. Triangle inequality.
http://mathworld.wolfram.com/TriangleInequality.html.