Alexandru-Ilie Popa
Pol Casacuberta Gil

CAIM Lab, Session 6: MapReduce and Document clustering

Implementació del map-reduce per a k-means:
Per aconseguir-ho primer hem hagut d'implementar la nostra versió de la similitud de Jaccard entre el que anomenem “prototip” i un document. Hem seguit la fórmula del document per a implementar-ho a l'hora de fer el còmput del document hem assumit que les probabilitats són 1, ja que si es troba al document la probabilitat que la paraula estigui és d'1, ja que sabem segur que hi és. La llista prototype no cal tractar-la de forma especial.

Per a calcular l'agregació del prototip hem escollit fer servir una estructura que ens permeti fer els càlculs eficientment, per a les freqüències un diccionari. Per als documents una simple llista ens serveix. És important l’elecció d’estructures i l’eficiència, ja que aquests càlculs s’hauran de fer moltes vegades i un cost massa alt podria ser prohibitiu.

En la funció mapper fem servir la funció de jaccard per a calcular el prototip més proper simplement calculem la distància del document a tots els prototips i escollim el millor.

Hem tingut problemes en implementar la distància de Jaccard, però al final ho hem entès en mirar la pàgina de Wikipedia. No vam entendre gaire bé com havíem de calcular la distància entre el prototype i el document. Ens hem adonat que la distància és diguem-ne un tipus de similitud i que cal agafar les probabilitats de cada paraula del prototip i comptar el nombre de paraules de cada document, ja que cada paraula del document diguem que té probabilitat 1.

Creiem que hem comès un error al codi o bé en fer els experiments, ja que ens hem trobat amb resultats que no esperàvem que fossin exactament com són. El principal error creiem que està al Kmeans, o en els experiments relacionats amb ell, ja que sempre genera 2 o 1 clústers.

Experiments

Anàlisi del nombre de clústers, nombre de paraules i freqüència

Hem fet els experiments sobre els documents del fitxer arxiv_abs.
Hem analitzat quins clústers resultants obtenim en canviar el nombre de paraules i la freqüència mínima i maxima.
Per fer els experiments hem creat diversos scripts.
Per poder veure millor que paràmetres posar a cada execució hem fet servir els següents:

- Clusters.sh: Hem fixat la freqüència entre [0.1,0.3] i el nombre de paraules a 200, Per veure com afectava el nombre de clústers a l'execució.
- Freq.sh: Provem amb el nombre de paraules 200, clusters 8, freqüència minima 0.1 i maxima de 0.3 a 1.0 en increments de 0.2.
- Freq2.sh: Els paràmetres són els mateixos que a l'script freq.sh, ara variem tant la freqüència minima com la maxima
- Size.sh:Ara com hem vist amb els scripts anteriors els paràmetres són els següents: freqüència [0.1-0.3], clusters 8. Variem el nombre de paraules.

Les conclusions que traiem després de veure els resultats dels scripts són les següents:

Numwords només limita el nombre de tokens seleccionats pel rang de freqüència, de manera que no afecta gaire als clústers resultants.
Esperavem que cada vegada que escollim freqüències més grans, tenim menys clústers. Ja que quan escollim tokens freqüents per representar els documents, tots els documents són molt semblants, per tant, la distància entre ells és molt petita.
El mètode convergeix ràpidament. Perquè el centroide del clúster es calcula a partir dels documents assignats, de manera que en la següent iteració tindrem el centroide molt proper als documents assignats a la iteració anterior.
Per contra, també esperavem que es compleixi el que s'explica a l'enunciat sobre les freqüències que estan en un rang baix. És a dir quan la freqüència és baixa tenim un petit factor aleatori, ja que els centroides dels clústers són escollits aleatòriament entre els documents. Però pel fet que la similitud de documents no és tan alta, els documents se solen classificar en clústers diferents, de manera que el nombre de clústers final s'acosta al límit que hi posem, 8 clústers. Hem triat aquest nombre, ja que a la carpeta arxiv_abs hi ha 8 subdirectoris hem decidit triar 8 clústers.

Canviant la mida del vocabulari:
num_words=(10 50 100 200 1000 10000)
Minfreq = 0.1
Maxfreq =0.3
n_clusters=8

Canviant la freqüència màxima:
maxfreq=(0.3 0.5 0.7 0.9 1)
Minfreq = 0.1
num_words= 200
N_clusters = 8

Canviant la posició dels rangs de freqüència:
Frequency range= (0.1,0.3 0.3,0.5 0.5,0.7 0.7,0.9)
n_clusters= 8
Num_words =200

Canviant el nombre de clusters:
n_clusters=(2 4 8 16 32 64 128)
Minfreq = 0.1
Maxfreq = 0.3
Num_words = 200

Per alguna raó que no acabem de veure obtenim resultats molt semblants pel que fa als clústers resultants, sempre o quasi sempre obtenim dos clústers finals, per molt que canviem el nombre de clústers, les freqüències, o bé el nombre de paraules.
On sí que podem veure canvis significatius és en les paraules resultants canviant les freqüències utilitzades.

Exemple de resultat:
CLASS4
[(1.0, 'within'), (0.8854567307692308, 'we'), (0.6835336538461538, 'from'), (0.6290865384615385, 'use'), (0.6230769230769231, 'which')]
CLASS0
[(0.8897554806070826, 'we'), (0.6488827993254638, 'from'), (0.6369940978077572, 'which'), (0.6357925801011804, 'use'), (0.5504637436762226, 'can')]