Laboratori de CAIM

Pràctica 5

Walter J. Troiani

Prof: Ignasi Gómez Sebastià

26/11/2023 - 2023/24 Q1

1 Implementació distribuïda de K-Means

En aquesta pràctica implementarem el famós algorisme de les K-Mitjanes emprant el mètode Expectation-Maximization, donat que és un problema NP-hard. L'aplicarem a un conjunt de dades no estructurades, com són els documents d'Arxiv i serà implementat en el paradigma distribuït de MapReduce d'Hadoop.

La implementació, de manera abstracta (el codi canvia lleugerament) és la següent: Cada funció map assignarà a cadascun dels documents un centroide, el més proper a ell emprant la similitud de Jaccard.

Algorithm 1 k-means in Hadoop MapReduce

```
1: function MAP(document_i, centroids)
2:     closestCentroid = 0
3:     for centroid in centroids do
4:         closestCentroid ← FindTheClosest(centroid, closestCentroid, document_i)
5:     end for
6:     return (closestCentroid, document_i)
7: end function
8:
9: function REDUCE(centroid, documentList)
10:    return (centroid, computeMean(documentList))
11: end function
```

En la implementació final es pot apreciar també que en el valor es retorna la funció de distorsió dels clústers mitjana, per a totes les iteracions d'una execució de K-means, més endavant s'explicarà que és i perquè ha estat emprada.

2 Experimentació

En aquest apartat ens centrarem a experimentar amb els possibles paràmetres rellevants un cop implementat l'algorisme de manera quantitativa, i a posteriori farem un anàlisi qualitativa dels resultats de l'algorisme, ja que a pesar de ser conegut, res ens assegura que emprant la similaritat de jaccard, a l'hora de comparar documents els resultats tinguin força sentit, a més a més els paràmetres poden arribar a influir en la qualitat dels clústers.

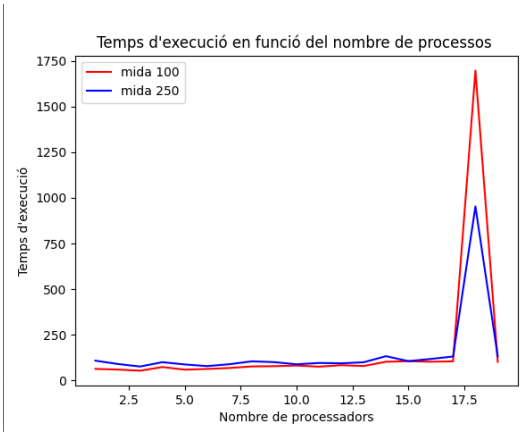
- N_c, nombre de prototips
- N_p, nombre de processos en paral·lel
- **m**, la freqüència mínima que ha de tenir una paraula per a ser considerada en la lectura de documents.
- **M**, la freqüència màxima que ha de tenir una paraula per a ser considerada en la lectura de documents.
- N_{iter}, nombre màxim d'iteracions si no hi ha convergència.

També podem afegir, com a variables objectiu, el temps d'execució, el nombre d'iteracions i la mida del vocabulari, que s'obtenen amb l'algorisme amb els fitxers d'entrada i els paràmetres. També en l'anàlisi del nombre de prototips introduirem mètodes i mesures per determinar un nombre de prototips adient.

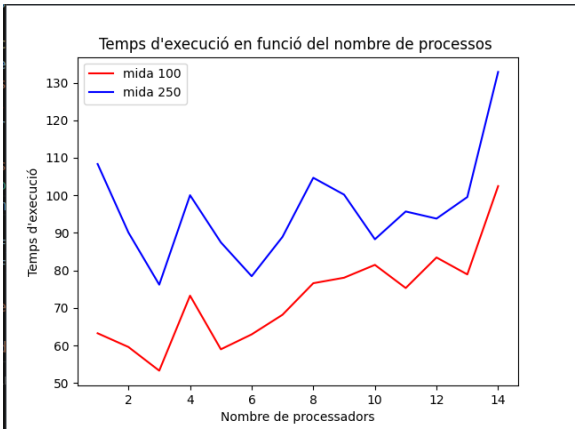
Finalment, cal recalcar que aquesta experimentació es farà fixant els paràmetres i fixant els documents d'entrada al conjunt de dades d'Arxiv. Els paràmetres fixats per defecte han sigut fruit de l'experimentació i exploració manual de diverses execucions. La màquina on ha estat executat és un processador Intel i7-1265 amb 10 nuclis i 32 GB de RAM.

2.1 Anàlisi de $\mathbf{N}_p$

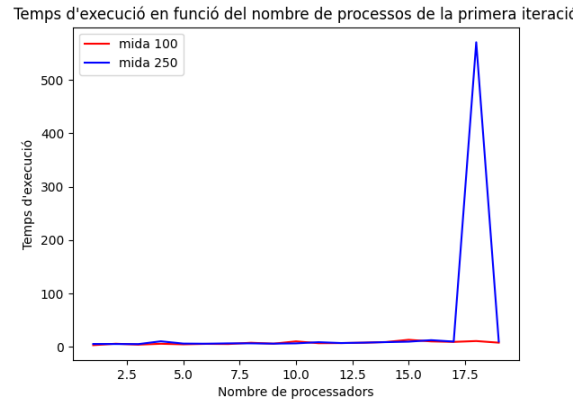
Primerament, analitzarem l'efecte dels processos en el temps d'execució, fixant els paràmetres $m = 0.05, M = 0.1, N_c = 10, N_{iter} = 10$ i analitzat per a dues mides diferents de vocabulari, hem analitzat el temps d'execució en funció del nombre de processos. S'ha tingut en compte un estudi separat pel temps d'execució en total del programa i també únicament el de la primera iteració que misteriosament sempre té temps menor a les següents iteracions. També s'han gràficat 2 cops, per extreure els "outliers" que provoquen molt soroll al gràfic i no deixen llegir-lo bé:



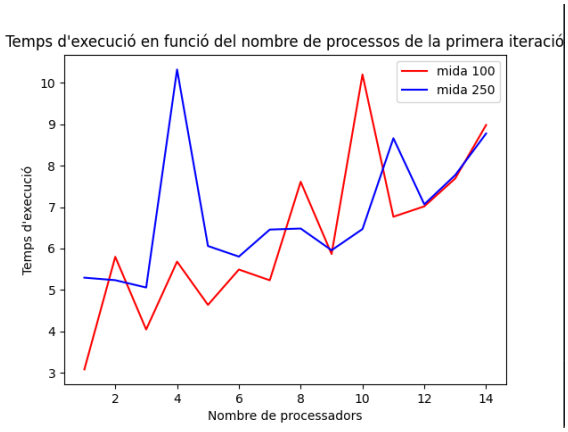
(a) Temps d'execució en funció del nombre de processos



(b) gràfic sense outliers



(a) Temps d'execució de la primera iteració en funció del nombre de processos



(b) gràfic sense outliers

La primera conclusió que podem treure, és que les primeres iteracions són més ràpides que la resta, això deu ser probablement al fet que inicialment, el nombre de paraules que conté un centroide, es molt menor i això provoca temps d'execució molt menors. Contra més documents s'associen a aquell clúster, més es trigarà a computar la tasca. En aquest cas la mida del vocabulari no sembla tenir una comparació clara.

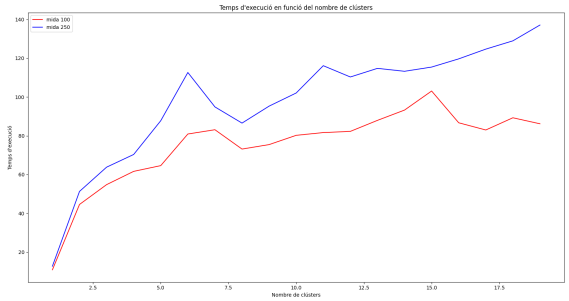
Altrament, podem veure que per a una execució sencera de l'algorisme (i també per la primera iteració) el mínim global s'arriba quan $N_p = 3$ per algun estrany motiu. Augmentar més enllà de 12 té efectes contraproduents, a causa dels grans

temps d'espera de sincronització i canvis de context dels threads segurament. Tampoc sabem com simula MRjob el Hadoop, aleshores potser hi ha altres factors d'arquitectura del processador que afecten. En ambdós gràfics s'ha produït una execució llarguíssima quan $N_p = 18$, la qual no hem pogut identificar el motiu, ja que els resultats dels prototips són completament normals.

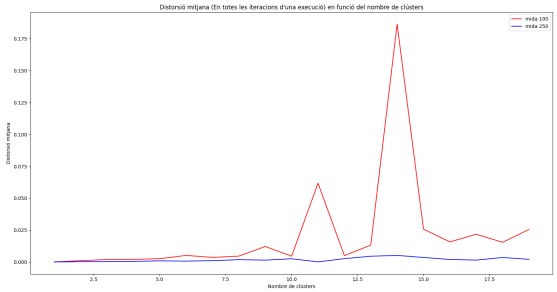
2.2 Anàlisi de $\mathbf{N}_c$

Per als clústers, hem estudiat 2 possibles factors: els valors de la funció de distorsió i el temps d'execució, els dos en funció del nombre de clústers, amb paràmetres fixats $m = 0.05, M = 0.1, N_p = 8, N_{iter} = 10$. Aquesta funció de distorsió és una mesura útil per estimar com són de llunyans els documents als centroides que li han estat assignats. Això ens serà útil per aplicar el conegut mètode del colze per a poder estimar el nombre correcte de clústers.

$$\sum_{i=1}^{n} Jaccard(C_k, \text{Document}_i)^2/\text{n}$$



(a) Temps d'execució en funció del nombre de centroides

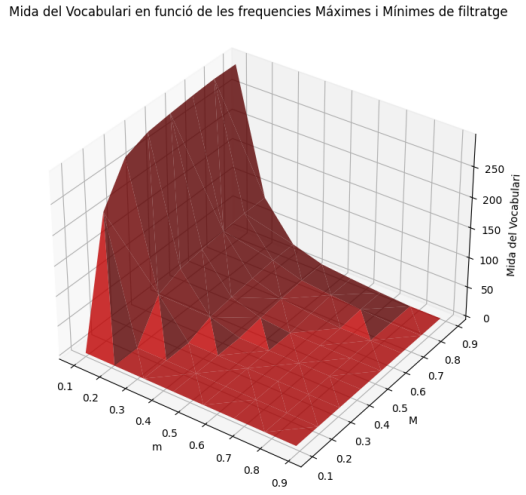


(b) Funció de distorsió promig en funció del nombre de centroides

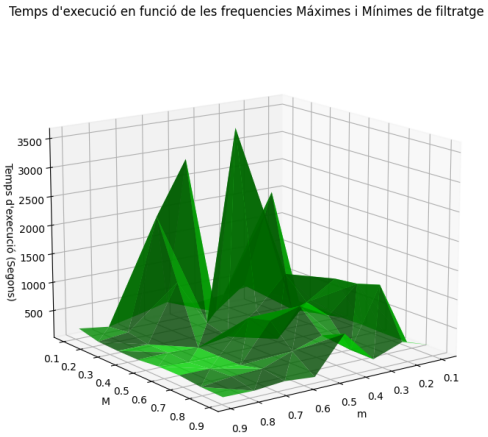
Com bé podem apreciar, el temps d'execució creix linealment amb el nombre de clústers, com era d'esperar, ja que el nombre d'operacions a realitzar es força major. L'altra observació clau, és veure que la funció de distorsió té un comportament estrany, impropi d'aquesta mena de funcions. Concloem que la implementació no deu ser del tot correcta o bé no s'ha executat bé i hi ha algún error en el codi, ja que aquest comportament estàtic i creixent a vegades no té sentit (Amb un clúster, els resultats haurien de ser força diferents que amb 2, amb pràcticament qualsevol conjunt de dades reals.

2.3 Anàlisi de m, M i mida del vocabulari

Un altre punt interessant és el següent, la mida del vocabulari i el temps d'execució d'aquest en funció dels paràmetres de les freqüències. De manera multivariable, tenim:



(a) Mida del vocabulari en funció de (m,M)



(b) Temps d'execució en funció de (m,M)

Primerament, cal tenir en compte que el valor més petit de m es 0.1, no hem provat d'experimentar amb valors més petits, ja que trigaven força (Això és a causa del fet que hi ha moltes paraules amb freqüència baixes), llavors es podria apreciar millor l'efecte exponencial que m té en la mida del vocabulari, en comparació amb M que té un impacte més lineal, més petit.

Pel que fa al temps d'execució, és difícil d'estimar la relació, ja que el gràfic té molt soroll per les execucions més altes, però podem veure que per valors petits de m, lògicament el temps d'execució augmenta.

2.4 Anàlisi qualitativa dels clústers

Finalment, un estudi dels valors dels clústers serà donat, per establir si l'algorisme és efectiu reunint documents semblants o per al contrari, no acaba de funcionar acceptablement. Mirarem els resultats de dues execucions diferents , per a 10 clústers i també per 6:

Cluster	Top Words
0	support (0.412), sinc, machin, address, neural, art, domain, cost, practic, finit
1	code (0.867), input, art, scheme, neural, impact, key, correct, open, tool
2	open (0.248), us, among, therefor, toward, precis, after, particl, light, presenc
3	shown (0.339), variabl, common, exhibit, correct, context, variat, neural, finit, behavior
4	sim (0.213), account, assum, best, optic, emiss, less, total, matter, separ
5	spatial (0.352), galaxi, stellar, univers, veloc, reveal, becaus, resolut, cluster, probabl

Table 1: Paraules més rellevants per a 6 centroides

Cluster	Top Words
0	probabl (0.688), either, much, code, assum, constant, graph, log, whether, finit
1	after (0.264), behavior, becaus, distanc, contrast, larger, impact, enhanc, wave, occur
2	emiss (0.263), optic, sim, fit, total, best, veloc, spectra, ray, gas
3	valid (0.661), help, literatur, precis, global, neural, machin, art, full, tool
4	exhibit (0.691), shown, common, behavior, among, presenc, pattern, critic, contrast, variat
5	variabl (0.930), sinc, represent, finit, period, formul, input, neural, machin, support
6	galaxi (0.460), survey, stellar, massiv, univers, cluster, popul, spatial, sim, gas
7	magnet (0.666), flow, electron, spin, layer, domain, rotat, across, conduct, particl
8	scheme (0.517), correct, code, adapt, often, converg, group, art, context, cost
9	strategi (0.462), establish, exploit, continu, employ, art, neural, machin, address, cost

Table 2: Paruales més rellevants per a 10 centroides

Els resultats són millors dels que esperàvem i veiem que per a 10 clústers els resultats milloren força i tot. Com podem veure hi ha algunes paraules que semblen més irrellevants com "either", però a pesar d'això els clústers que ha creat tenen força sentit (0 va sobre matemàtiques diverses, 2 sobre física clàssica, 3 sembla relacionat amb aprenentatge automàtic, el 6 es tracta de cosmologia, 7 de física quàntica/partícules...) Encara que bé hi ha grups mixtos com el 5 que sembla un mix entre computació i aprenentatge automàtic, potser perquè el nombre de clústers ideals era menor que el que s'ha emprat. També ens adonem que el clúster 1 no acaba de tenir massa sentit. Podem veure que els clústers més significatius, solen ser aquells que no tenen pesos petits (El clúster 2 a penes té témática o jo no li trobo, per exemple)

3 Conclusions i dificultats

La principal dificultat d'aquesta pràctica ha estat com sempre, el temps limitat que he pogut invertir en ella i a tot això sumant-li que és més feina degut a falta de personal... Però a part d'això "debuggar" aquest codi ha estat complicat, ja que l'API de MRjob no permet la posada en pantalla d'errors i sobretot entendre l'estructura dels nombrosos fitxers que conté la pràctica.

Pel que fa a l'experimentació, ha sigut difícil d'executar-la a causa dels enormes temps d'execucions d'algunes parts del projecte (també s'ha de tenir en compte que la quantitat de fitxers és força grossa en aquest índex) que han trigat hores. Entre això i la manca de temps, no he pogut descobrir que estava ocasionant que els valors de la funció de distorsió fossin tan estranys. Com que he hagut d'executar això mentre estava al treball, fent tasques pesades com executar Spark en local i entrenar models d'aprenentatge automàtic, que han influït sense cap mena de dubte a la presa de temps d'execució i poden haver estat els causants del soroll estadístic observat a l'experimentació, sobretot tenint en compte els grans outliers. Llavors podem concloure que aquesta pràctica no és molt eficient tant en energia elèctrica com en temps, no la tornaré a repetir pel bé del planeta.

Per a conclusions detallades dels paràmetres consultar l'apartat anterior, però resumidament, podem dir que l'algorisme classifica els documents en grups satisfactòriament, a pesar que hi ha paraules que no aporten tanta informació que apareixen en el top10, l'algorisme que hem fet encara té força potencial de millora.

Concloem també que MRjob no és molt bo simulant un sistema distribuït en un sol processador, com a mínim en el meu portàtil, ja que l'escalabilitat horitzontal és pèssima. Segurament un procés de MapReduce d'Hadoop executat en diversos computadors seria infinitament millor, a pesar de les sincronitzacions de xarxa o altres probablement escalaria millor.