Miquel Gotanegra Estañol
Oscar Daniel Sans Pallarés

INTRODUCCIÓ

En aquesta pràctica se'ns demana implementar un algorisme per a classificar documents en categories a través del K-Means, mitjançant una funció de similitud que en aquest cas es tracta del Jaccard index.

Hem tingut problemes per executar els scripts desde casa,j a que el MRKmeansStep no ens detecta el paràmetre --prot.
ValueError: unrecognized arguments: --prot /home/user/Desktop/CAIM/session6mapreduce/prototypes0.txt

EXPERIMENTACIÓ

Per a fer l'experimentació que ens proposa la pràctica hem agafat 6 intervals dels valors minfreq i maxfreq, els quals són els que s'indiquen a la seguent taula. Durant els testos hem mantngut el nombre d'iteracions per defecte (5) i el nombre de paraules dels paramatres ExtractData.py a 200 paraules. El nombre de clústers el situarem a 10

	Test 1		Test 2		Test 3		Test 4		Test 5		Test 6	
	0.01	0.025	0.05	0.1	0.1	0.3	0.5	0.7	0. 3	0.6	0.2	0.7
Iteració 1	2.41s		3.6s		5.30s		1.96s		3.04s		5.45s	
Iteració 2	10.55s		15.74s		13.73s		1.98s		4.99s		9.32s	
Iteració 3	6.15s		5.61s		4.70s		1.98s		3.69s		4.04s	

Iteració 4	7.98s			1.98s		
Iteració 5	8.25s					
Clústers	xxxxxx	xxxxxx	xxxxx	xxxxxx	xxxxxxx	xxxxxxx

A continuació, hem experimentat amb el temps d'execució en base al nombre de cores, amb el flag ncores. Hem utilitzat dos mostres, una amb 100 paraules i una altra amb 250. Hem utilitzat l'interval [0.1,0.3].

AMB 100 PARAULES

	1 Core	2 Core	3 Core	4 Core
Iteració 1	3.89	2.98	3.41	3.37
Iteració 2	6.32	4.79	5.22	5.43
Iteració 3	5.01	4.15	4.42	5.02
Iteració 4				
Iteració 5				

AMB 250 PARAULES

	1 Core	2 Core	3 Core	4 Core
Iteració 1	4.77s	3.67s	3.88s	4.30s
Iteració 2	11.14s	8.26s	8.65s	9.29s
Iteració 3	7.68s	5.86s	6.06s	6.44s
Iteració 4				

Iteració 5				

La conclusió que traiem després d'experimentar amb el nclust, és que hi ha una millora considerable en el temps d'execució en utilitzar 1 core a 2 cores, tot i que aquesta millora es perd al utilitzar 3 o 4 cores. Això pot ser degut a que la nostra imatge virtual está limitada a 2 cores. El motiu pel qual les primeres iteracions són més ràpides pot ser degut a que són les generades per GenereatePrototypes.py que genera els prototips per a la primera iteració del kmeans, el qual té menys càrrega que la reata d'iteracions.