[Cerca i Anàlisi d'Informació Massiva]

Session 6: MapReduce and Document clustering

Ana Mestre Borges

Carlos Roldán Montaner

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Introducción e implementación

En cuanto a la problemática y dificultad de la práctica, nos las hemos encontrado al comprender como funcionaba todo el sistema del MapReduce. Después de darle muchas vueltas y discutir bien como funcionaba, pudimos implementarla.

En cuanto al código se refiere, las estructuras que hemos usado han sido: para la función **`assign_prototype`** del mapper devolvemos el prototipo asignado y un pair (id documento, lista de palabras), para **`aggregate_prototype`** del reducer se devuelve como key el prototipo asignado y como valor un pair (lista de los documentos asignados a ese cluster ordenada alfabéticamente, lista de pairs ordenada alfabéticamente según el primer elemento formada por (palabra, la frecuencia de esa palabra dividida por el nombre de documentos asignados al cluster)).

Experimentación

Experimentación con los valores para la frecuencia mínima y máxima

Para poder hacer una mejor comparativa, subimos el número de clases a 10, mantenemos el número de iteraciones en 5 y usamos siempre 200 palabras. Se han obtenido los siguientes resultados para pruebas con frecuencia mínima 0.01, 0.025, 0.1, 0.3, 0.2 y frecuencia máxima 0.02, 0.05, 0.3, 0.6 y 0.7 respectivamente.

	Experimentos									
	1		**2**		**3**		**4**		**5**	
	F min	**F max**	**F min**	**F max**	**F min**	**F max**	**F min**	**F max**	**F min**	**F max**
Iteración	0.01	0.02	0.025	0.05	0.1	0.3	0.3	0.6	0.2	0.7
1	9.068150 s		10.090137 s		7.343174 s		7.400383 s		9.256073 s	
2	5.024831 s		10.479413 s		14.684536 s		9.649929 s		9.166459 s	
3	7.169896 s		8.749996 s		15.553978 s		15.835083 s		9.065594 s	
4	6.868361 s		10.559290 s		19.255669 s		14.146619 s		16.889445 s	
5	7.649593 s		11.262156 s		15.037523 s		-		12.887940 s	
Clases	class6, class4, class9		class6, class4, class8		class8		class7		class4	

Después de realizar varios experimentos podemos observar como al mantener unas frecuencias más bajas se consigue expandir más el número de clases pero juega un pequeño factor aleatorio, debido a que las palabras son demasiado específicas. En cambio cuando ampliamos este número, los resultados empiezan a ser demasiado similares, muy comunes entre ellos. Por ese motivo hemos decidido quedarnos con las frecuencias mínimas 0.1 y 0.3 porque sigue siendo baja pero se encuentra en un punto intermedio.

Experimentación con los valores para --mappers y --reducers

Para experimentar con los valores que podemos asignarle a los mappers y los reducers, primero crearemos dos datasets a partir de los cuales podremos comparar. Ambos los crearemos con las frecuencia mínima y máxima que se ha experimentado en el apartado anterior. El primer dataset estará formado por 100 palabras y el segundo por 250. Para esta comparación hemos trabajado con k = 20 (número de clases) y con 5 iteraciones. Después de ejecutar se obtienen los siguientes resultados:

Experimentos con 100 palabras												
	Map	Red	Map	Red	Map	Red	Map	Red	Map	Red	Map	Red
	2	4	4	2	2	8	8	2	4	4	4	8
1	7.717594 s		7.689389 s		7.229982 s		7.893904 s		7.635010 s		7.746722 s	
2	14.343315 s		15.059917 s		15.705862 s		15.740178 s		15.033008 s		15.517237 s	
3	9.950949 s		10.085616 s		10.245059 s		9.423628 s		9.474312 s		10.035048 s	
4	18.478085 s		18.353634 s		17.489847 s		18.353551 s		18.487457 s		17.805996 s	
5	14.166233 s		14.321498 s		13.784410 s		14.474452 s		15.042882 s		15.185292 s	
Clases	class15		class15		class15		class15		class15		class15	

Experimentos con 250 palabras												
Iteraciones	Map	Red	Map	Red	Map	Red	Map	Red	Map	Red	Map	Red
	2	4	4	2	2	8	8	2	4	4	4	8
1	9.791126 s		9.257121 s		10.147840 s		9.880983 s		9.623485 s		10.015347 s	
2	29.347339 s		29.107394 s		32.595971 s		33.126136 s		31.324007 s		31.433637 s	
3	18.824487 s		17.858325 s		17.948360 s		17.327249 s		15.775855 s		18.135333 s	
4	27.364803 s		26.881103 s		27.675807 s		27.384681 s		23.861785 s		24.508580 s	
5	15.578470 s		16.808895 s		16.720390 s		15.707455 s		14.305951 s		15.724765 s	
Clases	class2		class2		class2		class2		class2		class2	

Experimentando tanto con el número de reducers como de mappers se puede comprobar que los tiempos son muy similares para cada iteración.

Adicionalmente podemos comprobar que cuanto mayor es el número de palabras, como era de esperar, más tardan las ejecuciones.

También es obvio ver en cada experimento que la primera iteración es la más rápida, esto es debido a que en la primera iteración un prototipo es un documento y por lo tanto, las palabras son las correspondientes a ese documento. En el resto de iteraciones, contiene las palabras de todos los que forman parte del cluster y las operaciones que tiene que realizar son más lentas.

Palabras más frecuentes

Para finalizar el análisis de los resultados producidos, estudiaremos las palabras más frecuentes obtenidas a partir del último prototipo asignado. Para realizar esto, ejecutaremos Kmeans con 20 iteraciones.

Utilizamos frecuencia mínima 0.01 y frecuencia máxima 0.02, número de palabras 250 y 20 clústers. Al acabar la ejecución, observamos que los documentos se han agrupado en 3 clases. Tiene sentido que no sean muchas ya que muchos de los temas están bastante relacionados y pueden agruparse (política, religión y ateísmo, deportes, tecnología y ciencia). Aún así esperábamos algun clúster más. Los términos agrupados en cada una de las

tres clases también son bastante coherentes. En la primera (Clase 15), algunos de los más importantes son ‘cop’, ‘firearm’, ‘traffic’ (temática de policía), ‘medicin’, diseas’, ‘abort’, ‘laboratori’ (medicina y ciencia), etc. La segunda clase (clase17), parece más orientada a la política, terrorismo, guerra y religión, con términos como ‘territori’, ‘troop’, ‘border’, ‘agreement’, ‘soldier’, ‘terrorist’, ‘bomb’, ‘declar’, ‘occupi’... Finalmente, la última clase (clase 19), es más difícil de clasificar, y parecer más miscelánea, con términos aparentemente más inconexos (‘setup’, ‘pittsburgh’, ‘misc’, ‘editor’).