[Cerca i Anàlisi d'Informació Massiva]

Session 8: Locality sensitive hashing

Ana Mestre Borges

Carlos Roldán Montaner

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Introducción

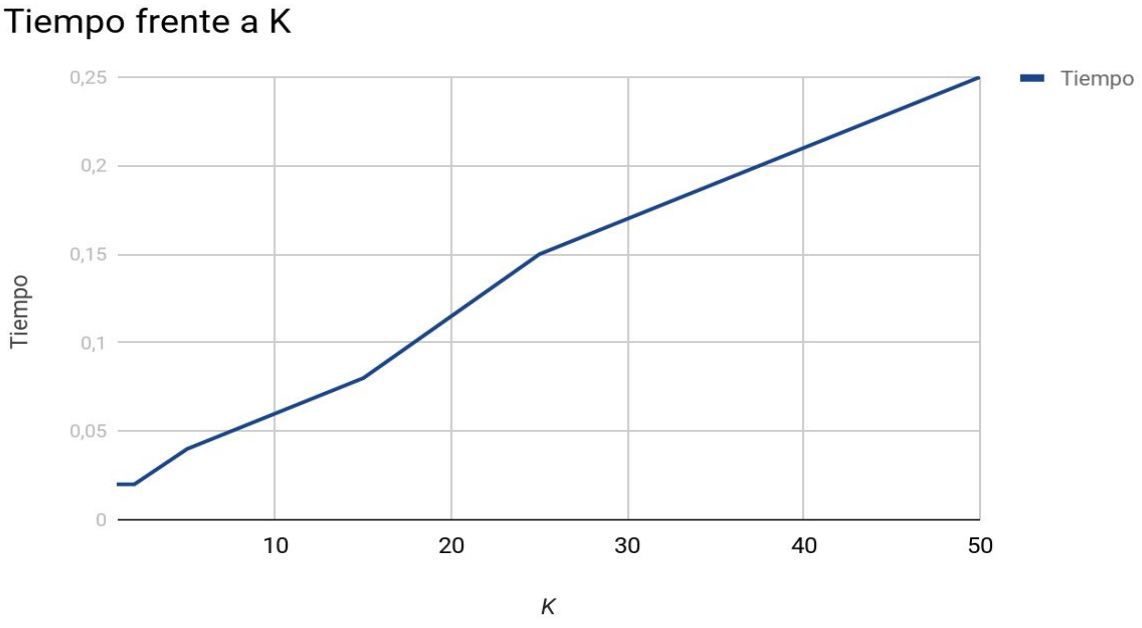
El objetivo de esta práctica es realizar Locality Sensitive Hashing sobre un set de imágenes y comparar cómo afectan los parámetros k y m tanto a los resultados como a los tiempos. Además, usaremos Locality Sensitive Hashing para calcular la imagen más cercana y su distancia y compararemos este método con el de fuerza bruta. Utilizaremos los resultados correspondientes a la ejecución para las 10 primeras imágenes de testing.

¿Cómo afectan los parámetros k y m a lsh?

Variación de k

Para nuestro primer experimento, fijamos m = 1, para que sólo haya una repetición y estudiamos cómo varía el tiempo de ejecución y el número de candidatos cuando modificamos k.

K	1	2	5	10	15	25	50
Segundos	0.02	0.02	0.04	0.06	0.08	0.15	0.25



Podemos observar que a medida que incrementamos el número de funciones de hash, el tiempo incrementa, de una forma más o menos lineal. En cuanto al número candidatos, este

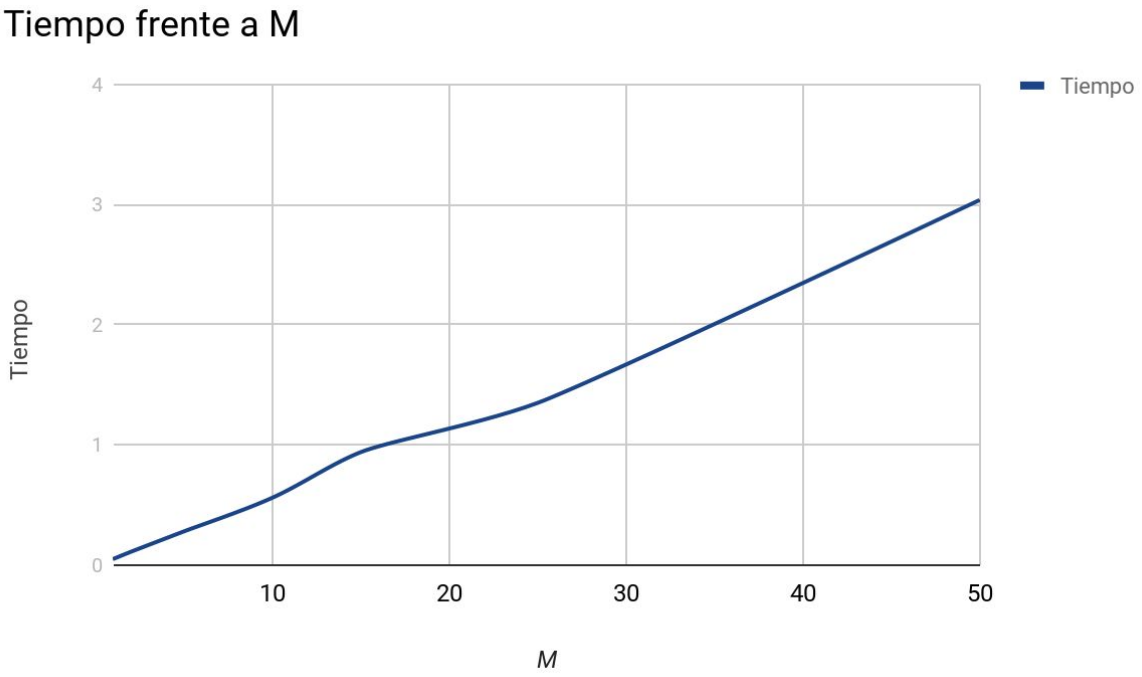
se reduce según k va incrementando, de manera que para ks elevadas, no tenemos prácticamente candidatos.

K	1	2	5	10	15	25	50
Avg.Cand	850.4	765.3	400.1	154.8	94.1	12.1	0.9

Variación de m

A continuación, dejamos fija una k = 10, que parece un número razonable de funciones de hash, e incrementamos m.

M	1	2	5	10	15	25	50
Segundos	0.05	0.11	0.28	0.56	0.94	1.35	3.04



Observamos que al igual que ocurría, al aumentar k, incrementar m conlleva un incremento del tiempo de ejecución, también de forma aproximadamente lineal. Sin embargo, como podemos ver en la siguiente tabla, el número de candidatos aumenta en lugar de disminuir.

M	1	2	5	10	15	25	50
Avg.Cand	154.8	549.5	808.4	996.6	1103.3	1217.3	1265.0

LSH vs método de fuerza bruta

Finalmente, comprobaremos cual es la mejora que aporta LSH frente a un método de fuerza bruta. Para ello probaremos distintos valores de **m** y **k** y compararemos los tiempos de ambos. Primero, haremos la prueba con un número pequeño de imágenes de testing, en este caso 10. El tiempo obtenido con Brute-Force Search es 0.27 segundos.

	K = 1 M = 1	K = 1 M = 5	K = 1 M = 10	K = 5 M = 1	K = 5 M = 5	K = 5 M = 10
LSH	0.18	0.35	0.39	0.11	0.43	0.65

	K = 10 M = 1	K = 10 M = 5	K = 10 M = 10	K = 15 M = 1	K = 15 M = 5	K = 15 M = 10
LSH	0.09	0.47	0.74	0.11	0.47	0.85

	K = 25 M = 1	K = 25 M = 5	K = 25 M = 10
LSH	0.15	0.70	1.33

Como podemos observar, excepto para los casos en los que sólo realizamos una repetición, utilizar el método de fuerza bruta produce tiempos de ejecución más pequeños. Esto es debido al tiempo gastado en realizar el hashing de todas las imágenes. Sin embargo, observemos que sucede cuando utilizamos un mayor número de imágenes de testing. El tiempo obtenido con Brute-Force Search es 2.66 segundos.

	K = 1 M = 1	K = 1 M = 5	K = 1 M = 10	K = 5 M = 1	K = 5 M = 5	K = 5 M = 10
LSH	1.47	2.68	2.84	0.76	2.66	3.02

	K = 10 M = 1	K = 10 M = 5	K = 10 M = 10	K = 15 M = 1	K = 15 M = 5	K = 15 M = 10
LSH	0.39	1.86	2.19	0.24	0.95	1.54

	K = 25 M = 1	K = 25 M = 5	K = 25 M = 10
LSH	0.15	0.74	1.41

Ahora, los tiempos de LSH superan en su mayoría a los de Brute-Force, debido que solo ha de comprobar en los candidatos. Además, vemos que al aumentar k, el tiempo se reduce. Esto es debido al hecho de que, como comentamos en el primer apartado de este informe, incrementar k reduce el número de candidatos, haciendo que las búsquedas sean más rápidas.

¿Cómo se ve afectado el resultado según el algoritmo?

Hemos representado cómo afectan los diferentes valores de **m** y **k** en el tiempo de ejecución pero, en cuanto al resultado, ¿obtenemos las mismas salidas siempre? No en todas las ejecuciones. En muchos casos, ambos consiguen llegar a la misma imagen como la más cercana y obviamente con la misma distancia pero hay algunos casos en que no es así. Veamos diferentes ejemplos de ello:

	K = 1, M = 1		K = 5, M = 1		K = 10, M = 1		K = 60, M = 10	
	LSH	**Brute-Force**	**LSH**	**Brute-Force**	**LSH**	**Brute-Force**	**LSH**	**Brute-Force**
Image	1502		1500		1509		1506	
Nearest image	840	1429	89	1416	1304	1458	Not found	1460
Num. Candidates	499	-	333	-	8	-	0	-
Distance	67	60	161	52	85	40	None	64

Después de varias ejecuciones con diferentes parámetros para K y M podemos llegar a las siguientes conclusiones:

- Los resultados suelen ser muy parecidos, la tabla de arriba es una representación de algunos casos en que no encuentra los mismos datos. Pero aún así, en general tanto LSH como Brute-Force encuentran las mismas imágenes cercanas con la misma distancia.

- En el caso de no encontrar la misma imagen, encuentran dos con una distancia muy similar.
- Está claro que Brute-Force va a encontrar sí o sí la óptima en cualquier caso, ya que hace una búsqueda exhaustiva entre todos los datos.
- Solemos dificultar el encuentro de la imagen más cercana para LSH en cuanto aumentamos la K, ya que le estamos disminuyendo el número de candidatos y en consecuencia, es más complicado hallar la imagen óptima entre ellos.
- Aumentar la M aumenta el número de candidatos, ergo aumenta también las posibilidades de acierto.