Pràctica 8: Locality Sensitive Hashing

Ricard Meyerhofer
Raúl Gómez

Q1-2016/17

1. Understanding the code and basic functioning of lsh

Amb l'objectiu d'avaluar l'evolució del temps d'execució en funció de la variació de la variable k, s'han realitzat modificacions al codi proporcionat que automatitzen un increment de la variable i executen l'algorisme iterativament amb cada increment. Els valors de k utilitzats han estat:

$$k = [20, 50, 100, 500, 750, 1000, 1250, 1500, 1750, 2000]$$

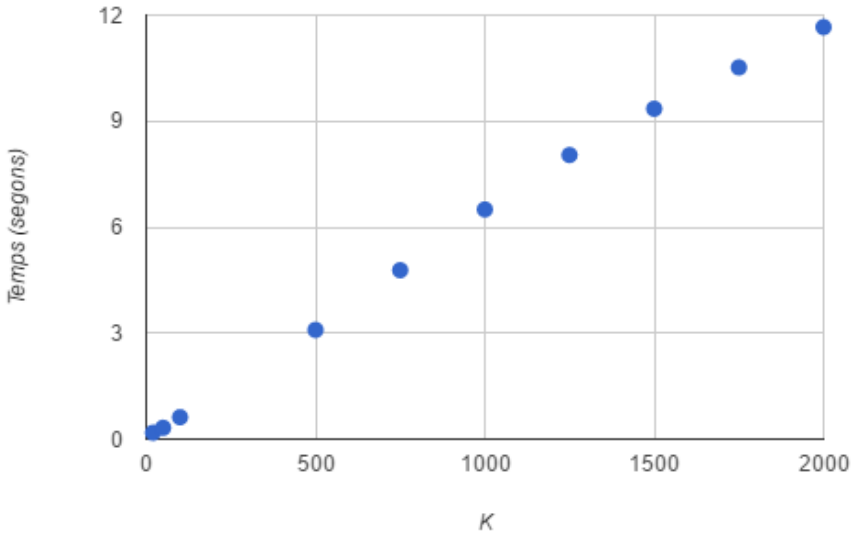

Figura 1: Variació del temps respecte la variable k

L'increment del temps corresponent a l'augment del valor de k és causat a què s'augmenten el nombre de bits de la hash, factor que implica que el temps d'execució s'incrementi.

Per avaluar la variable m s'ha repetit el procediment anterior, els valors utilitzats han estat:

$$m = [5, 10, 15, 25, 50, 75, 100, 125, 150, 200]$$

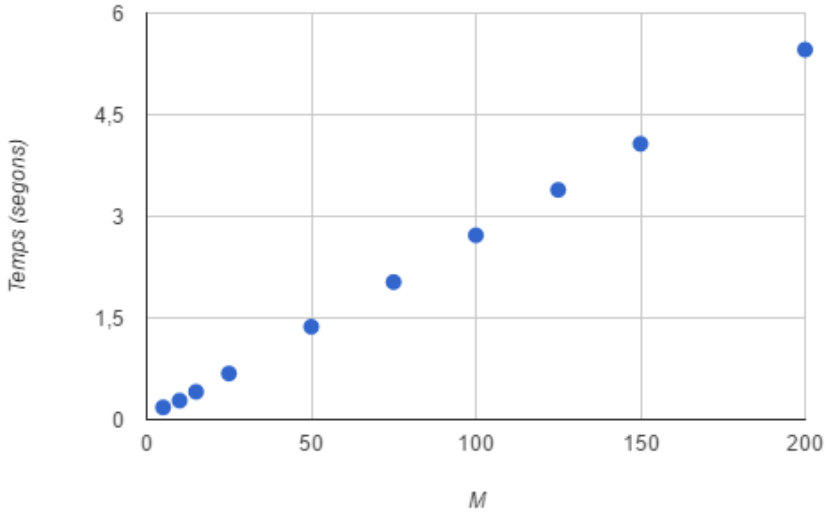

Figura 2: Variació del temps respecte la variable m

La variable m determina el nombre d'iteracions i, com a conseqüència, el temps d'execució augmenta en funció d'aquesta variable. Per aquest motiu, es pot detectar que l'augment del temps és lineal, doncs és directament proporcional al nombre d'iteracions.

Com a comprovació, s'ha volgut realitzar una tercera prova incrementant simultàniament les variables k i m amb els valors anteriors. El resultat ha sigut el següent:

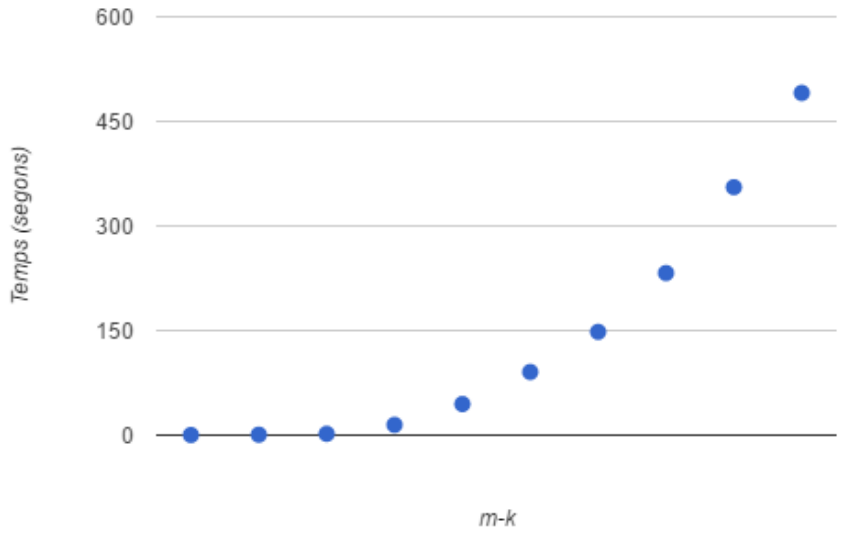

Figura 3: Variació del temps respecte les variables k i m

2. Does lsh work?

Un cop entès tot el codi proporcionat i realitzades les proves prèvies, s'ha procedit amb les implementacions demanades:

- ***Distance function:*** Tal com es demanava, la funció computa la *distance* entre les dues imatges d'entrada.
- ***Brute-force search:*** En aquest mètode per tal de computar les primeres 1500 imatges del *dataset* i trobar la mínima, en primera instància s'ha realitzat el càlcul amb la primera imatge amb l'objectiu d'inicialitzar la variable *lowdist* de la mínima distància trobada amb un valor real. El bucle itera les següents 1499 i reescriu a la variable *lowdist* si alguna imatge és menor a la trobada fins al moment.
- ***Nearest neighbor:*** Per implementar aquest mètode s'ha realitzat un canvi a la implementació demanada, on si no existeixen candidats, es demana que la funció retorni *None*. A la implementació realitzada assignem en primera instància el valor -1 i en el cas que no existeixin candidats aquest és el valor resultant. El canvi realitzat no modifica els resultats esperats de la pràctica.

2.1. Execucions

k=20 i m=5

```
Running lsh.py with parameters k = 20 and m = 5
There are 130 candidates for image 1500
Difference between nearest neighbour VS brute-force search for 1500 is: 0.0
Brute-force search for 1500 is 131
Nearest neighbour for 1500 is 200

There are 69 candidates for image 1501
Difference between nearest neighbour VS brute-force search for 1501 is: 0.0
Brute-force search for 1501 is 169
Nearest neighbour for 1501 is 1073

There are 130 candidates for image 1502
Difference between nearest neighbour VS brute-force search for 1502 is: 0.0
Brute-force search for 1502 is 177
Nearest neighbour for 1502 is 1148

There are 188 candidates for image 1503
Difference between nearest neighbour VS brute-force search for 1503 is: 0.0
Brute-force search for 1503 is 169
Nearest neighbour for 1503 is 712

There are 200 candidates for image 1504
Difference between nearest neighbour VS brute-force search for 1504 is: 1.0
Brute-force search for 1504 is 73
Nearest neighbour for 1504 is 1027

There are 91 candidates for image 1505
Difference between nearest neighbour VS brute-force search for 1505 is: 0.0
Brute-force search for 1505 is 171
Nearest neighbour for 1505 is 247

There are 278 candidates for image 1506
Difference between nearest neighbour VS brute-force search for 1506 is: 0.0
Brute-force search for 1506 is 28
Nearest neighbour for 1506 is 311

There are 133 candidates for image 1507
Difference between nearest neighbour VS brute-force search for 1507 is: 0.0
Brute-force search for 1507 is 198
Nearest neighbour for 1507 is 1149

There are 27 candidates for image 1508
Difference between nearest neighbour VS brute-force search for 1508 is: 1.0
```

```
Brute-force search for 1508 is 297
Nearest neighbour for 1508 is 1462

There are 72 candidates for image 1509
Difference between nearest neighbour VS brute-force search for 1509 is: 0.0
Brute-force search for 1509 is 7
Nearest neighbour for 1509 is 348

0.79 sec
'main' ((), {}) 0.79 sec
```

Tal com es pot comprovar, lsh funciona més ràpidament que $brute - force$ amb k i m reduïts. Encara això, els resultats d'ambdós no coincideix amb el resultat de la imatge més propera. Avaluant la diferència entre lsh i $brute - force$ es detecta que en alguns casos no coincideixen (diferència = 1).

Per evitar el problema s'ha incrementat la variable m i, encara que el temps d'execució ha augmentat, no provoca errors entre els dos algorismes.

k=20 i m=20

```
Running lsh.py with parameters k = 20 and m = 20
There are 312 candidates for image 1500
Difference between nearest neighbour VS brute-force search for 1500 is: 0.0
Brute-force search for 1500 is 131
Nearest neighbour for 1500 is 200

There are 257 candidates for image 1501
Difference between nearest neighbour VS brute-force search for 1501 is: 0.0
Brute-force search for 1501 is 169
Nearest neighbour for 1501 is 1073

There are 193 candidates for image 1502
Difference between nearest neighbour VS brute-force search for 1502 is: 0.0
Brute-force search for 1502 is 177
Nearest neighbour for 1502 is 1148

There are 249 candidates for image 1503
Difference between nearest neighbour VS brute-force search for 1503 is: 0.0
Brute-force search for 1503 is 169
Nearest neighbour for 1503 is 712

There are 361 candidates for image 1504
Difference between nearest neighbour VS brute-force search for 1504 is: 0.0
Brute-force search for 1504 is 73
Nearest neighbour for 1504 is 365

There are 174 candidates for image 1505
Difference between nearest neighbour VS brute-force search for 1505 is: 0.0
Brute-force search for 1505 is 171
Nearest neighbour for 1505 is 534

There are 470 candidates for image 1506
Difference between nearest neighbour VS brute-force search for 1506 is: 0.0
Brute-force search for 1506 is 28
Nearest neighbour for 1506 is 311

There are 347 candidates for image 1507
Difference between nearest neighbour VS brute-force search for 1507 is: 0.0
Brute-force search for 1507 is 198
Nearest neighbour for 1507 is 285

There are 160 candidates for image 1508
Difference between nearest neighbour VS brute-force search for 1508 is: 0.0
```

```
Brute-force search for 1508 is 297
Nearest neighbour for 1508 is 1152

There are 181 candidates for image 1509
Difference between nearest neighbour VS brute-force search for 1509 is: 0.0
Brute-force search for 1509 is 7
Nearest neighbour for 1509 is 348

1.41 sec
'main' ((), {}) 1.41 sec
```