3) LSH, algo de teoría, algo de práctica.

1. “Algo de Teoría” (20 puntos)

Tenemos una colección de millones de archivos binarios y queremos encontrar rápidamente archivos parecidos con el propósito de detectar malware. A fin de resolver este problema se nos plantea la idea de usar LSH para estos archivos. A modo de guía le pedimos que piense y resuelva los siguientes problemas:

a.1) ¿Cuál sería la función de minhash a utilizar? compruebe que cumple las propiedades necesarias para ser un minhash.

a.2) En base a la función de minhash planteada indique de qué forma se realizaría la amplificación de la misma usando b=2 y r=16

a.3) En base a los dos puntos anteriores indique qué pre-procesamiento debería realizar sobre los archivos para poder encontrar los similares a un archivo dado.

a.4) Considerando el punto anterior una vez realizado el pre-procesamiento indique de qué forma encontraría los archivos candidatos a ser similares a un nuevo archivo.

1. “Algo de Práctica” (10 puntos)

Los siguientes vectores representan calificaciones de canciones del 1 al 5, donde 0 significa que ese usuario no escuchó la canción:

 v1 = [1 5 3 2]      v2 = [0 0 1 2]      v3 = [4 4 5 0]      v4 = [5 1 0 1]

Utilizando la técnica de los hiperplanos, con b = 1, r = 4, hallar hiperplanos correspondientes para que el más similar a v1 sea v3 (y ningún otro).

RESOLUCIÓN

a.1) Una buena función de minhash a utilizar puede ser la de Jaccard debido a que se puede obtener shingles de longitud fija para cada uno de los archivos binarios. A estos shingles se les aplica una función de minhash del estilo:

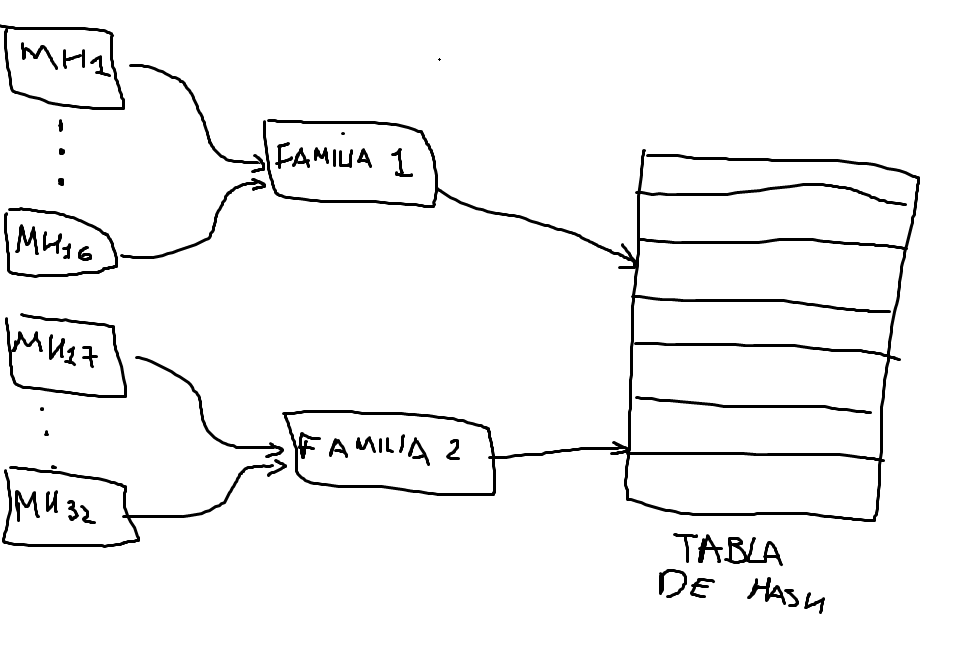
(C0.S0+…+Cn.Sn) % p % m = MH(S)

Y nos quedamos con el mínimo de los minhash de todos los shingles correspondientes al archivo para obtener un minhash.

Esta función cumple con las propiedades para ser minhash debido a que es rápida de calcular, y la probabilidad de que dos elementos colisionen y resulten en un mismo resultado del minhash dependerá directamente de su distribución binaria y de nada más (Si dos archivos son iguales, entonces el minhash obtenido es el mismo).

a)2) Utilizando b=2 y r=16 significaría que habrá 16 minhashes por cada grupo, y habrá 2 grupos en los cuales se aplicará una familia que relacione los 16 minhashes de la misma. De esta forma, se obtienen 2 posiciones resultantes en las cuales se almacenará el archivo en la tabla de hashing.

El esquema será como el siguiente:



Donde la familia relaciona a los minhashes de la siguiente manera:

(MHn . K1 + … + MHn+16 . K16) % p % m = F(Mhn,…,Mh16+n)

Siendo Kn constantes, p un numero primo mayor a m, y m el numero de buckets de la table de hash.

A su vez, cada minhash posee la misma función que en el ítem a)1), pero diversificando sus constantes C0,…,Cn para que sean minhashes distintos.

a)3) El pre-procesamiento requerido sería pasar a todos los elementos por el hashing de LSH, para que de esta manera queden almacenados todos en su posición respectiva de la tabla.

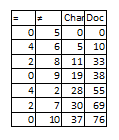
Una vez que estos archivos se encuentran almacenados, entonces se puede proceder a resolver una consulta con un Query, buscando que colisione con sus elementos semejantes al establecerse en una posición determinada de la tabla pasado por Hashing.

Pasar un elemento por hashing significa aplicarle los r minhashes de cada grupo, y relacionarlos en cada una de las b familias que corresponden, para que el archivo se almacene en su posición correspondiente.

a)4) Para encontrar a los archivos similares al candidato se pasa al candidato por hashing; se le aplica los r\*b minhashes, y se los relaciona con las b familias de hashing. Esto nos otorgará 1 o 2 posiciones en las cuales el archivo query se almacenaría; entonces tomamos los elementos de las posiciones que le correspondan al query y esos serán los candidatos a ser similares al archivo de la consulta.

Si se busca más precisión, ahora se pueden comparar 1 a 1 los archivos obtenidos en hashing, ya que solo estaremos comparando con elementos que son altamente similares al archivo query, y nos ahorramos el tener que comparar con todos los archivos que poseemos.

4) A partir de la siguiente estructura de un índice Invertido:



madreigueragistradomaleficiotanantialmanifiesto

1010010011

01101010010010101011111

01011

11101010110101010

01001001000100

01110100101011010100100

(las posiciones en los documentos están numeradas desde la posición 1, y todo está codificado en gamma)

Armado con la finalidad de poder resolver consultas por proximidad, se pide extraer la información de los documentos y resolver la consulta “**Madriguera Maleficio** ” utilizando **TF-IDF**. Detallar cada paso realizado (\*\*\*\*) (20pts)

Términos

Madre, Madriguera, Magistrado, Maleficio, Maleta, Manantial, Manifiesto

**RESOLUCIÓN**

Para resolver el ejercicio debo resolver la consulta madriguera y la consulta maleficio. De esta manera primero comienzo con la consulta de Madriguera:

*Consulta de madriguera*

* Hay 7 elementos, por lo que comienzo con búsqueda binaria en el índice y se accede a la posición 3 del índice (1 Acceso a índice)
* Esta posición contiene 9 caracteres distintos del anterior, y se debe leer del char 19 al 28(Lectura del próximo elemento del índice). El término obtenido es maleficio. (1 acceso a disco)
* Madriguera es menor que maleficio por lo que se hace búsqueda binaria entre los elementos 0,1, y 2. Se accede al elemento 1. (1 Acceso a índice)
* Se lee del char 5 al 11, y comparte 4 caracteres con el anterior, por lo que habrá dos accesos a disco, y 2 lecturas mas de índice (2 indice, 2 disco).
* El elemento obtenido es Madriguera, por lo que ahora debo ver que documentos contiene, leyendo desde el 10 hasta el 33.
* El resultado obtenido es:

011 010 1 00100 1 010 1 011 1 1 1

Que traducido significa:

D3:2(1, 5), D4:2(1,4), D5:1(1)

*Consulta de maleficio*

* Hay 7 elementos, por lo que comienzo con búsqueda binaria en el índice y se accede a la posición 3 del índice (1 Acceso a índice)
* Esta posición contiene 9 caracteres distintos del anterior, y se debe leer del char 19 al 28(Lectura del próximo elemento del índice). El término obtenido es maleficio. (Ya en memoria, 1 acceso a memoria).
* Como maleficio es el término que buscaba accedo a los documentos del mismo, con el puntero a doc que es el 38, y accedo a la posición de documento del siguiente que es la 55.

Los documentos obtenidos son:

1 1 1 010 1 011 010 1 010

Que traducido significa:

D1:1(1), D3:1(3), D5:1(2)

Entonces para finalizar con el query debo ver el rankeo con TF-IDF:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | TF | | | | | IDF |
| D1 | D2 | D3 | D4 | D5 | Log((N+1)/tfi) |
| Madriguera | - | - | 2 | 2 | 1 | 0,3010 |
| Maleficio | 1 | - | 1 | - | 1 | 0,3010 |

* D1 = 0,3010
* D2 = 0
* D3 = 0,9030
* D4 = 0,6020
* D5 = 0,6020

|  |
| --- |
| **RANKING FINAL** |
| D3 = 0,9030 |
| D5 = 0,6020 // D4 = 0,6020 |
| D1 = 0,3010 |