

# 7506-2021 Parcialito de LSH

Román Vázquez Lareu

TOTAL POINTS

**90 / 100**

QUESTION 1

**1 LSH 90 / 100**

1) Métrica y minhash

✓ - **10 pts** No detalla cómo es la función de minhashing ni sus parámetros, o define mal.

**1** Sería "cantidad de grupos de notas", para hacer referencia al concepto de shingle

**2** Bien

**3** No me termina de quedar en claro por qué la representación en 4 bits. El alfabeto tiene 13 valores posibles, entonces es  $13^n$ . De la misma manera que en los tweets teníamos  $26^n$  por 26 caracteres distintos

**4** Bien

**5** Bien

**6** Esto no está bien. Estarías introduciendo muchísimos falsos positivos. No hay razón por la cual usar una tabla de solo 100.000 posiciones. Por ejemplo, una tabla de 70M de posiciones con entradas de 4 bytes solo ocupa 0.28GB, lo cual no es nada.

**7** Bien

**8** Bien

**9** Bien

**10** Bien

**11** Bien

**12** Bien

**13** Bien

**14**

Bien. Faltaría decir que es con la semejanza de Jaccard y a través de los shingles.

1) Elijo jaccard. Esto porque los números se almacenan como una combinación de 4 bits y puedo definir la semejanza como la cantidad de bites en común. Además la longitud es variable, por lo que pinta cerca y euclides habría que agregar un padding.

Puede tomar 2 valores: cero o 1

$\Rightarrow$  para los shingles tomo  $2^m$ ;  $m = 27$

$$2^{27} \approx 130M > 70M$$

3

usando los shingles, voy hashearlos y conservo el mínimo.

Al quedarme con el mínimo de esos hash  $\rightarrow$  Minhash.

Luego tendré que hashear con una función de familia universal.

$m$  depende del valor del tamaño de tabla

Los parámetros que multiplican dependen de las restricciones presentes ( $a, b, c, d$ ).  $p$  primo  $\geq m$

Ej con largo max 4:

5

$$h_t(\text{tira-bit}) = (a * \text{tira-bit}[0] + b * \text{tira-bit}[1] + c * \text{tira-bit}[2] + d * \text{tira-bit}[3]) \bmod p \bmod m$$

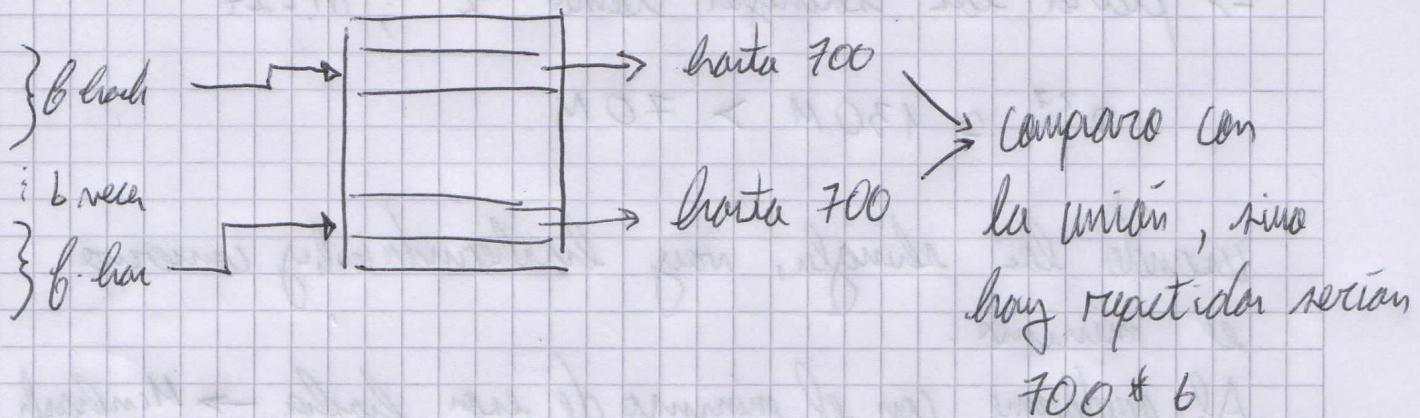
~~Algoritmo~~

Haciendo 70 millones de canciones.

si tomo  $m$ , por ejemplo, 100 000 podría tener hasta 700 canciones por posición.

6

En una cantidad considerable, pero hay que tener en cuenta el  $b$ , porque compararía con 700 candidatos  $b$  veces



2)

- ) 0,8 semejanza  $\Rightarrow$  si semejanza entre canciones  $> 0,8 \Rightarrow$  parecidas

0,8: límite mínimo de semejanza



$$1 - 0,8 = d_1$$

$$\boxed{1 - d_1 = 0,8}$$

- ) 0,2 semejanza  $\Rightarrow$  si semejanza entre canciones  $< 0,2 \Rightarrow$  no parecidas

0,2: límite máximo de semejanza

$$1 - 0,2 = d_2$$

$$\boxed{1 - d_2 = 0,2}$$

- ) de 1000, devuelven tope 160 no relevantes

$$\Rightarrow P(\text{no relevantes}) = \frac{160}{1000} = \frac{160}{1000} = \boxed{0,16 = p_2}$$

$$P_{\text{rel}}(\text{semejantes}) = \boxed{0,88 = p_1}$$

8

$$) \rho_1 = 1 - (1 - (1-d_1)^r)^b$$

$$\rho_1 = 1 - (1 - 0,8^r)^b$$

$$r=2, b=2$$

$$\rho_1 = 1 - (1 - 0,8^2)^2 = 0,87 < 0,88$$

$\Rightarrow \uparrow b$  : debba ser mer permisivo

$$\rho_1 = 1 - (1 - 0,8^2)^3 = 0,95 > 0,88$$

$$b=3, r=2 \text{ cumple } \rho_1$$

9

$$) \rho_2 = 1 - (1 - (1-d_2)^r)^b$$

$$\rho_2 = 1 - (1 - 0,2^r)^b$$

$$b=3, r=2 \rightarrow \rho_2 = 0,11 < 0,16$$

$\Rightarrow$   $b=3$  y  $r=2$  merciforcan

10

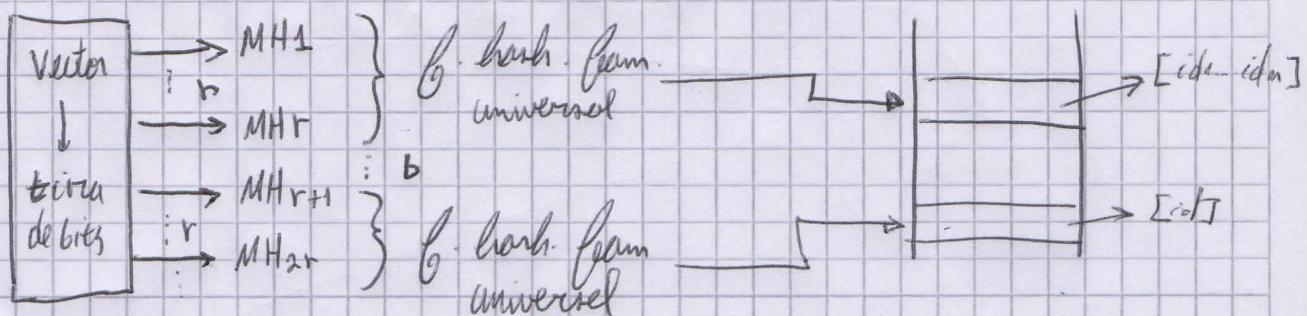
$\Rightarrow$  permiso  $r * b = 3 * 2 = 6$  ministrashei.

3) Toma los vectores y crea la tira de bits.

Los vectores son hasheados por función de hash  $b$  y luego se conservan los mínimos.

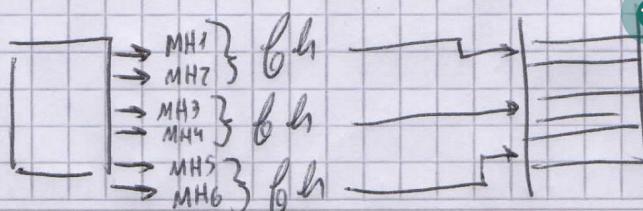
De ahí que son minihash. Una vez hecho esto, una función de hash de familia universal de long. fija toma cada valor de cada minihash y devuelve una posición (o múltiple dependiendo del  $b$ ) y, en el caso de ser un procesamiento, almacenar allí el identificador de la canción/ secuencia de bits.

11



$$r=2 \quad ; \quad b=3$$

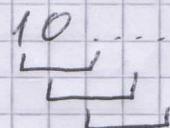
12



9) Se reúne el vector y se almacena la tira de bits. Se van armando los shingle y posheando, conservando el mínimo.

Luego una función de hash recibe esos valores y devuelve una posición en la tabla de hash final.

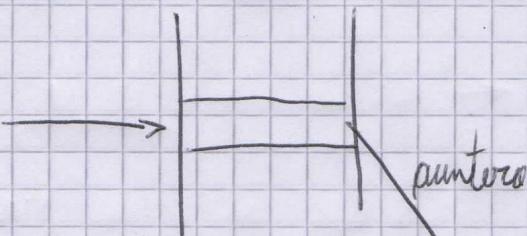
Por ej. supongamos:



$MH_1 = 10$   
 $MH_2 = 200$

TABLA DE HASH

hash ( $MH_1, MH_2, \dots$ )



13  
lista de id  
de canciones ]

Si el  $b > 1$ , devolvería varias posiciones y habría que comparar con todos los candidatos de todas las posiciones. Aparte puede tener un diccionario donde con un id da la canción.

Una vez obtenido esto ahora si comparo mi canción query con los candidatos y devuelvo la más parecida.

Entonces previo al query debería tener estas dos estructuras

## 1 LSH 90 / 100

1) Métrica y minhash

✓ - **10 pts** No detalla cómo es la función de minhashing ni sus parámetros, o define mal.

**1** Sería "cantidad de grupos de notas", para hacer referencia al concepto de shingle

**2** Bien

**3** No me termina de quedar en claro por qué la representación en 4 bits. El alfabeto tiene 13 valores posibles, entonces es  $13^n$ . De la misma manera que en los tweets teníamos  $26^n$  por 26 caracteres distintos

**4** Bien

**5** Bien

**6** Esto no está bien. Estarías introduciendo muchísimos falsos positivos. No hay razón por la cual usar una tabla de solo 100.000 posiciones. Por ejemplo, una tabla de 70M de posiciones con entradas de 4 bytes solo ocupa 0.28GB, lo cual no es nada.

**7** Bien

**8** Bien

**9** Bien

**10** Bien

**11** Bien

**12** Bien

**13** Bien

**14** Bien. Faltaría decir que es con la semejanza de Jaccard y a través de los shingles.