Andrés Calvente Rodríguez. 53855310M

Búsqueda y Minería de Información
Grado en Ingeniería Informática, 4º curso
Prueba intermedia – 21 de marzo 2019

3.5 pt 1. Una colección incluye, entre otros, los siguientes documentos, donde a, b, c, d, e, f representan términos (con ese orden alfanumérico):

d_0 $a\ b\ c\ a$
d_1 $f\ a\ d$
d_2 $a\ a\ c\ a\ b\ a\ c\ a\ a\ a\ b\ a$
d_3 $b\ e\ b\ b\ e\ b$
d_4 $a\ b\ e\ f\ b\ a$

a) Mostrar un índice posicional para estos documentos (suponiendo que no utilizamos una tabla hash).

b) Suponiendo que la colección contuviera en total 8 documentos, y que los términos de los cinco documentos de arriba no aparecen en ningún otro documento, calcular los pesos *tf-idf* de los términos en cada documento.

c) Calcular la puntuación de los documentos para la consulta $b\ d\ e\ f$ por el modelo vectorial, utilizando la implementación orientada a documentos, mostrando los heaps de implementación paso a paso, así como un heap de ránking para devolver los top 2 documentos.

1.5 pt 2. Dado el siguiente índice posicional:

a	1 (5 10) 2 (5 10) 3 (1 5 9 14 26)	c	3 (2 6 10 18 22) 4 (4 9) 5 (3 5)
b	1 (2 6) 3 (3 7 19) 4 (1 5)	d	2 (3 6) 3 (4 8 15 21) 5 (1 4)

Calcular la puntuación del documento 3 en una búsqueda proximal de la consulta $a\ b\ c\ d$. Mostrar el resultado final y los intervalos que dan lugar a las puntuaciones del documento.

1.5 pt 3. a) Comprimir la siguiente secuencia de valores enteros mediante el método de Huffman con códigos diferencia.

5 8 9 9 10 11 10 10 10 11 12 12 13 12 12 13 12 14 17

¿Qué longitud promedio de código resulta en la compresión (sin contar el primer valor, que no se comprime)?

b) Comprimir el siguiente conjunto de (tres) números enteros mediante códigos byte variable: 16386 0 161.

Indicación: $16386 = 2^{14} + 2$, $161 = 2^7 + 2^5 + 1$

¿Si la tasa de compresión resultante es del 50%, cuántos bytes se estaban utilizando para cada valor entero sin comprimir?

1 pt 4. a) Estimar el número de resultados para una consulta $q = t_1\ t_2 \ldots t_k$ en función del número n de documentos indexados por un motor de búsqueda (a cuyo índice tenemos acceso), y la longitud n_i de las listas de postings de los términos t_i, suponiendo que el buscador sólo devuelve documentos que contengan ***todas*** las palabras de la consulta. Fundamentar matemáticamente la respuesta.

b) Supongamos ahora que el buscador no es nuestro y no tenemos acceso al índice. Y supongamos que este buscador devuelve todos los documentos en los que aparezca ***alguna*** palabra de la consulta.

Para una cierta consulta formada por dos palabras, el buscador devuelve 4.9 millones de resultados, mientras que para las palabras por separado devuelve 3 millones y 2 millones de documentos respectivamente. Estimar el número de documentos indexados por el buscador.

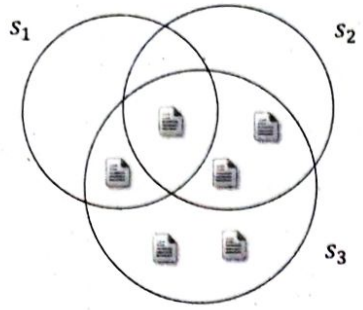
1.5 pt 5. a) Un ránking de búsqueda contiene documentos relevantes en las posiciones 2 y 5, con grado 1 y 2 respectivamente. Suponiendo que había un documento relevante más, con grado 1, no devuelto por el buscador, calcular MAP, R-precisión, MRR y nDCG.

b) Para una cierta consulta un buscador devuelve k documentos, consiguiendo precisión 0.5, y una media armónica coincidente con la media aritmética entre precisión y recall. ¿Cuántos documentos relevantes había en la colección para esa consulta?

1 pt 6. Un documento de texto contiene 1.000 palabras contando las repeticiones (tokens) y –para simplificar– tan sólo 4 palabras diferentes (token types). Si la distribución de palabras sigue la ley de Zipf (con exponente $\alpha = 1$), ¿cuál sería el número de repeticiones esperado de cada palabra? Mostrar cómo se llega a la respuesta.

Búsqueda y Minería de Información
Grado en Ingeniería Informática, 4º curso
Prueba intermedia – 25 de abril 2019

2 pt 1. Tenemos tres buscadores Web s_1, s_2, s_3. Ejecutamos una cierta consulta q en s_3, nos quedamos con los 6 primeros resultados, y comprobamos cuáles son devueltos también por los otros dos buscadores, obteniendo la situación que se muestra en la figura. Dados estos datos:



a) ¿Cuántos documentos Web podemos estimar que se indexan entre s_1 y s_2 (en función de N)?

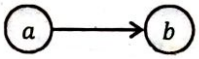
b) ¿Cuántos documentos podemos estimar que se indexan entre s_1, s_2 y s_3 juntos?

2 pt 2. Dados los siguientes documentos:

d_1 – a b c a b a a d
d_2 – b b c a a d a
d_3 – a c d c b b d c
d_4 – c a b d a

a) Estimar el valor de similitud de los documentos dos a dos para la detección de duplicados. Utilizar bigramas, valores hash ficticios, y mostrar paso a paso las operaciones para llegar al resultado.

b) Dados $n < m$ y suponiendo que dos documentos d_1, d_2 no contienen n-gramas ni m-gramas repetidos, demostrar que $\text{Jaccard}_n(d_1, d_2) \geq \text{Jaccard}_m(d_1, d_2)$, donde Jaccard_k representa la similitud con k-gramas de k palabras (es decir, que cuanto más largos se toman los k-gramas, más baja es la similitud).

2 pt 3. a) Calcular PageRank de a y b en el siguiente grafo con $r = 0.5$: 

b) Si en un grafo los únicos k sumideros que hay son nodos que tampoco tienen enlaces entrantes, ¿cuál es su valor de PageRank en función de r, k y el número de nodos N?

c) Demostrar que si en las ecuaciones de PageRank no dividimos el término de teleportación por N, los valores de PageRank resultantes (con $r > 0$) suman N. Indicación: la solución a las ecuaciones de PageRank es única para $r > 0$.

4 pt 4. Tres usuarios han expresado las siguientes preferencias por varias canciones:

	Suzanne	Hero	Get Lucky	Reality
María		4	3	
Carmen	5	2	5	
Raúl	4	3	2	

a) Predecir cuánto puede interesar "Suzane" a María mediante kNN colaborativo basado en usuario sin normalizar con similitud de Pearson.

b) Predecir cuánto puede interesar "Reality" a María mediante kNN basado en contenido con similitud Jaccard, utilizando los estilos de la música:

Suzanne	folk, songwriter
Hero	indie, folk, pop
Get Lucky	jazz, funk, electronic, dance
Reality	dance, electronic, house, pop

c) Supongamos que las celdas de fondo negro en la matriz de ratings se toman como test y el resto como entrenamiento, y un cierto recomendador revuelve estos ránkings:

María		Carmen		Raúl	
Reality	4	Get Lucky	5	Hero	4
Hero	1	Suzanne	2	Suzanne	3
Suzanne	0.5	Reality	1		

Calcular Recall@2 y RMSE de estas recomendaciones, considerando como relevantes los ratings a partir de 4.

Búsqueda y Minería de Información
Grado en Ingeniería Informática, 4º curso
Examen final – 20 de mayo 2019

Parte I

2.5 pt 1. Dado el siguiente índice posicional:

a 2 (1, 3, 9, 11) 4 (1, 3)
b 1 (0, 2, 4, 5) 2 (5, 6, 8, 12)
c 2 (2, 7, 13, 15) 3 (0,1)
d 2 (0, 4, 10, 14) 4 (0, 2)
e 1 (1, 3)

a) Reconstruir y mostrar el contenido de los documentos originales.

b) Calcular el ránking para la consulta $a\ b\ c\ d$ por coseno en el modelo vectorial (no es necesario mostrar los heaps paso a paso, es suficiente mostrar el cálculo).

c) Calcular la puntuación del documento 2 para la misma consulta por búsqueda proximal, mostrando los intervalos que determinan la puntuación.

0.5 pt 2. a) Comprimir la siguiente lista de posiciones utilizando el método de Huffman con códigos diferencia:

0, 2, 4, 5, 6, 8, 10, 11, 12, 14, 15, 18, 20, 24, 25, 26, 27, 30, 34, 37, 41, 46

b) ¿Cuál es la tasa de compresión frente a longs de 4 bytes? ¿Cuál es la longitud promedio de código?

1 pt 3. Para una cierta consulta sobre una colección de 100 documentos, un buscador devuelve dos documentos relevantes, en las posiciones 2 y 5, y ocho documentos no relevantes. Sabiendo que había 20 documentos relevantes en la colección, calcular MAP, Recall, MRR, nDCG@2 y fallout.

Indicación: fallout = FP / (FP + TN), donde FP son los falsos positivos y TN son los verdaderos negativos.

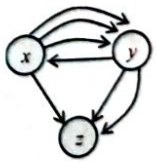
TNOR

Parte II

0.5 pt 4. Generando y ejecutando consultas aleatorias en un buscador X reunimos un total de medio millón de resultados, y comprobamos que de éstos la mitad están indexados en Google. Por otra parte, ejecutando las mismas consultas en Google reunimos un millón de resultados.

Sabiendo que X indexa dos mil millones de documentos, cuántos documentos podemos estimar que indexan entre X y Google?

1 pt 5. a) Calcular PageRank en el siguiente (multi)grafo de hiperenlaces web, suponiendo que el navegante aleatorio elige hiperenlaces salientes al azar sin tener en cuenta que estén repetidos.



b) Mostrar un ejemplo de grafo donde tenemos dos páginas x e y con $g_{in}(x) < g_{in}(y)$, pero $P(x) > P(y)$ (es decir x tiene menos enlaces entrantes que y pero más PageRank).

2 pt 6. Tres usuarios han expresado las siguientes preferencias por tres documentos de texto, cuyos vectores *tf-idf* se muestran:

	d_1	d_2	d_3
u_1	2	2	1
u_2	4	4	2
u_3		3	4

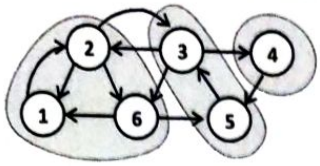
	d_1	d_2	d_3
a	4	4	1
b	2	0	2
c	0	0	2
d	4	0	1

a) Predecir cuánto puede interesar el documento d_1 al usuario u_3 mediante kNN colaborativo normalizado basado en ítem (sin restricción de vecindario).

b) Realizar la misma predicción mediante recomendación basada en contenido por centroides.

Parte III

2.5 pt 7. Dada la siguiente red:



a) Considerando la dirección de los arcos, realizar los siguientes cálculos:
   i. Betweenness (sin normalizar) del arco (2,1).
   ii. ¿Qué 2-cliques identificas en la red? ¿Son 2-clanes?
   iii. ¿Hay algún puente global?

b) Tomando la red como no dirigida (es decir ignorando la dirección de los arcos), determinar las siguientes cuestiones y métricas:
   i. El arraigo del arco (3,6).
   ii. El coeficiente de clustering de la red, por la fórmula global.
   iii. La modularidad de la red respecto a la partición que se muestra en la figura.
   iv. ¿Hay algún puente local?

Ahora, en una red cualquiera:

c) ¿Cuál es el clique más pequeño posible en una red fuertemente conexa con n nodos? Demostrar que no es posible un clique más pequeño.

d) Mostrar un ejemplo de red donde el radio es la mitad del diámetro. Mostrar otro ejemplo donde el radio es igual al diámetro.

   Indicación: el radio de una red es la excentricidad mínima entre todos sus nodos; la excentricidad de un nodo es su distancia al nodo más lejano.