Zach Quinn

**Mining of Massive Data Sets Problem Sets**

**3.1.1**

Jackard Similarity

AB

|S n T |/|S u T |

A = {1, 2, 3, 4}

B = {2, 3, 4, 5}

C = {2, 4, 6}

n - Intersection = In both A and B.

u - Union = In both A or B.

A u B = 1, 2, 3, 4, 5

A n B = 2, 3, 4

A u B/A n B = 2/6

Divide by 2 to simplify fraction:

= 1/3

Convert to decimal:

0.33.

AC

A u C = 1, 2, 3, 4, 6

A n C = 2, 4

A u C/A n C = 2/5.

Divide fraction to find decimal value:

0.4.

BC

B u C = 2, 3, 4, 5, 6

B n C = 2, 4

B u C/B n C = 1/6.

Divide fraction to find decimal value:

0.17

**3.2.1**

String to evaluate:

“The most effective way to represent documents as sets, for the purpose of identifying lexically similar documents is to construct from the document the set of short strings that appear within it.”

We are seeking these shorter strings by using the shingle methodology.

Final shingle count in string:

'In', 'n t', 'th', 'thi', 'his', 'is', 's s', 'se', 'sec', 'ect'

**3.3.3**

In order to properly accomplish the goal of minhashing: To create a unique signature for each hypothetical document, I created a matrix of sets and organized them into a table of values.

As I understand it, a minhash of a given set is the number of the row or, in this case, element, with the first non-zero value in the permuted order of pi.

Therefore, I created the following tables by randomly permuting rows. I picked P = P random row permutations. In the below table, element represents an index value of 0 – 5. S denotes signature. I.e. signature 1, signature 2, signature 3, etc. I then take the h(x) = the modulo values, 2x + mod6, 3x + mod6 and 5x+2mod6.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Element | S1 | S2 | S3 | S4 | 2x + mod 6 | 3x + mod 6 | 5x + 2 mod 6 |
| 0 | 0 | 1 | 0 | 1 | 1 | 2 | 2 |
| 1 | 0 | 1 | 0 | 0 | 3 | 5 | 1 |
| 2 | 1 | 0 | 0 | 1 | 5 | 2 | 0 |
| 3 | 0 | 0 | 1 | 0 | 1 | 5 | 5 |
| 4 | 0 | 0 | 1 | 1 | 3 | 2 | 4 |
| 5 | 1 | 0 | 0 | 0 | 5 | 5 | 3 |

To create a minhash function, I take the h(x) to create different permutations, i.e. h1, h2, h3, etc. Below, I take the h(x) x = the elements in the above table and determine if they are present in the minhash signatures. I replace the element values for x in h(x).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | S1 | S2 | S3 | S4 |
| h1(0) | / | 1 | / | 1 |
| h2(0) | / | 2 | / | 2 |
| h3(0) | / | 2 | / | 2 |
| h1(1) | / | 1 | / | 1 |
| h2(1) | / | 2 | / | 2 |
| h3(1) | / | 1 | / | 2 |
| h1(2) | 5 | 1 | / | 1 |
| h2(2) | 2 | 2 | / | 2 |
| h3(2) | 0 | 1 | / | 0 |
| h1(3) | 5 | 1 | 1 | 1 |
| h2(3) | 2 | 2 | 5 | 2 |
| h3(3) | 0 | 1 | 5 | 0 |
| h1(4) | 5 | 1 | 1 | 1 |
| h2(4) | 2 | 2 | 2 | 2 |
| h3(4) | 0 | 1 | 4 | 0 |
| h1(5) | 5 | 1 | 1 | 1 |
| h2(5) | 2 | 2 | 2 | 2 |
| h3(5) | 0 | 1 | 4 | 0 |

This yields a final table of just the values inherent in each of the given minhash signatures (S1 – S4).

|  |  |  |  |
| --- | --- | --- | --- |
| S1 | S2 | S3 | S4 |
| 5 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 |
| 0 | 1 | 4 | 0 |

B) In the above table, only function h3 counts as a true permutation because the linear order of the values was rearranged throughout the iterations of h(x).

C) The estimated Jaccard similarities are not close to the actual computed values as demonstrated by the table below. I calculated the column-to-column and signature-to-signature values to derive the following table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Similarities | 1-2 | 1-3 | 1-4 | 2-3 | 2-4 | 3-4 |
| Col/col | 0 | 0 | 0.25 | 0 | 0.25 | 0.25 |
| Sig/sig | 0.33 | 0.33 | 0.67 | 0.67 | 0.67 | 0.67 |

**3.4.1**

Letting s = 0.1 - 0.9 and b = 10, r = 3, I computed the following using a calculator:

|  |  |
| --- | --- |
| s | 1 – (1 - sr)b |
| 0.1 | 0.0100 |
| 0.2 | 0.0772 |
| 0.3 | 0.2394 |
| 0.4 | 0.4839 |
| 0.5 | 0.7369 |
| 0.6 | 0.9123 |
| 0.7 | 0.9850 |
| 0.8 | 0.9992 |
| 0.9 | 1.0000 |

Letting s = 0.1 – 0.9 and b = 20, r = 6, I computed the following using a calculator:

|  |  |
| --- | --- |
| s | 1 – (1 - sr)b |
| 0.1 | 0.0000 |
| 0.2 | 0.0013 |
| 0.3 | 0.0145 |
| 0.4 | 0.0788 |
| 0.5 | 0.2702 |
| 0.6 | 0.6154 |
| 0.7 | 0.9182 |
| 0.8 | 0.9977 |
| 0.9 | 1.0000 |

Letting s = 0.1 – 0.9 and b = 50, r = 5, I computed the following using a calculator:

|  |  |
| --- | --- |
| s | 1 – (1 - sr)b |
| 0.1 | 0.0005 |
| 0.2 | 0.0159 |
| 0.3 | 0.1145 |
| 0.4 | 0.4023 |
| 0.5 | 0.7956 |
| 0.6 | 0.9825 |
| 0.7 | 0.9999 |
| 0.8 | 1.0000 |
| 0.9 | 1.0000 |