Nate Jensvold

CPTS 315

9/21/20

HW 1

Q1. (25 points) Consider the following market-basket data, where each row is a basket and shows the list of items that are part of that basket.

1. {A, B, C}
2. {A, C, D, E}
3. {A, B, F, G, H}
4. {A, B, X, Y, Z}
5. {A, C, D, P, Q, R, S}
6. {A, B, L, M, N}

a) What is the absolute support of item set {A, B} ?

4

b) What is the relative support of item set {A, B} ?

4/6 = 67.7%

c) What is the confidence of association rule A ⇒ B ?

a = 6

b = 4

4/6 = 66.7%

Q2. (15 points) Answer the below questions about storing frequent pairs using triangular matrix and tabular method.

a) Suppose we use a triangular matrix to count pairs and the number of items n = 20. If we store this triangular matrix as a ragged one-dimensional array Count, what is the index where count of pair (7, 8) is stored?

Find pair {i, j}, where i<j, at the position

$(i - 1)(n - i/2) + j - i$

(7-1)(20-7/2) + 8 - 7= **position 100**

b) Suppose you are provided with the prior knowledge that only ten percent of the total pairs will have a non-zero count. In this case, which method among triangular matrix and tabular method should be preferred and why?

In this example the tabular method would be faster because less than 1/3 of all pairs have a nonzero count.

The comparison between the two approaches states that the tabular approaches beats out the triangular matrix when at most 1/3 of the pairs have a nonzero count.

Q3. (35 points) This question is about the PCY algorithm for counting frequent pairs of items. Suppose we have six items numbered 1, 2, 3, 4, 5, 6. Consider the following twelve baskets.

1. {1, 2, 3}
2. {2, 3, 4}
3. {3, 4, 5}
4. {4, 5, 6}
5. {1, 3, 5}
6. {2, 4, 6}
7. {1, 3, 4}
8. {2, 4, 5}
9. {3, 5, 6}
10. {1, 2, 4}
11. {2, 3, 5}
12. {3, 4, 6}

Suppose the support threshold is 4. On the first pass of the PCY algorithm, we use a hash table with 11 buckets, and the set {i, j} is hashed to i × j mod 11.

a) By any method, compute the support for each item and each pair of items.

1: 4
2: 6
3: 8
4: 7
5: 6
6: 4

Relative Support:
1 = 4/12 = 33.3%
2 = 6/12 = 50%
3 = 8/12 = 66.7%

4 = 8/12 = 66.7%
5 = 6/12 = 50%
6 = 4/12 = 33.3%

1,2: 2/12 = 16.7%
1,3: 3/12 = 25%
1,4: 2/12 = 16.7%
1,5: 1/12 = 8.3%
1,6: 0/12 = 0
2,3: 3/12 = 25%
2,4: 4/12 = 33.3%
2,5: 2/12 = 16.7%
2,6: 1/12 = 8.3%
3,4: 4/12 = 33.3%
3,5: 4/12 = 33.3%
3,6: 2/12 = 16.7%
4,5: 3/12 = 25%
4,6: 3/12 = 25%
5,6: 2/12 = 16.7%

b) Which pairs hash to which buckets?

1,2 = 2 mod 11 = 2

1,3 = 3 mod 11 = 3

1,4 = 4 mod 11 = 4

1,5 = 5 mod 11 = 5

1,6 = 6 mod 11 = 6

2,3 = 6 mod 11 = 6

2,4 = 8 mod 11 = 8

2,5 = 10 mod 11 = 10

2,6 = 12 mod 11 = 1

3,4 = 12 mod 11 = 1

3,5 = 15 mod 11 = 4

3,6 = 18 mod 11 = 7

4,5 = 20 mod 11 = 9

4,6 = 24 mod 11 = 2

5,6 = 30 mod 11 = 8

c) Which buckets are frequent?

0: 0
1: 5
2: 5
3: 3
4: 6
5: 1
6: 3
7: 2
8: 6
9: 3
10: 2

1,2,4, and 8 are frequent

d) Which pairs are counted on the second pass of the PCY algorithm?

2,6

3,4

1,2

4,6

1,4

3,5

2,4

5,6

Q4. (25 points) Please read the following paper and write a brief summary of the main points in at most ONE page. You can skip the theoretical parts.

When people reuse material, text or code in products it is very easy to detect exact matches between documents but quite difficult to detect partial matches. By dividing a document into several subsections and then hashing each subsection it suddenly becomes possible to detect if there are sections of similarity between the two objects. The downside of this method is that it can result in failure if the program somehow hashes the similar sections to different numbers resulting in a miss instead of a collision. These misses stem from the fact that a subsection can be partially like something from the other object without being an exact match. All these techniques used together are called fingerprinting and despite the downside of this algorithm is can be improved by using the following method.

The k-gram method creates subsections for every single possible section in a document and then only compares hashes from the two comparable documents that both hash to the same 0 mod p value, where p is some value chosen by the algorithm.

A even more advanced technique to determine similarities is called winnowing. Winnowing is where a k-gram method is used on every single section in the document. However, in addition to a section hashing itself it also looks at the hashes surrounding it to create a fingerprint of hashes. Now to determine similarities to other documents all one has to do is compare the array of hashes to other documents that have been winnowed, this makes it so the more similar an array of hashes is the more similar that section is to somewhere in another document.