Cpts 315 HW 1

Zach Fechko

25 September, 2022

Problem 1

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. $\{A, B, C, D, E\}$
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 the itemset $\{A, B\}$?
b. What is the *relative* support of the itemset $\{A, B\}$?
c. What is the confidence of association rule $A \implies B$?

Problem 1 Solution

a. The absolute support of $\{A, B\}$ is 4. Because the itemset $\{A, B\}$ appears in 4 baskets
b. The relative support of $\{A, B\}$ is $\frac{\text{number of baskets containing itemset}}{\text{total number of baskets}} = \frac{4}{6} \approx 0.6667$.
c. The confidence of the association rule $A \implies B$ is $\frac{\text{number of baskets containing itemset}}{\text{number of baskets containing item A}} = \frac{4}{6} \approx 0.6667$.

Problem 2

Answer the below questions about storing frequent pairs using triangluar matrices and the 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-dimmenional array *Count*, what is the index where count of pair $(7, 8)$ is stored? b. Suppose you are provided with prior knowledge that only 10 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?

Problem 2 Solution

a. Position in a ragged array $= (i-1) \times (n - \frac{i}{2}) + (j - i)$, plugging in the numbers for position $(7, 8)$, and $n = 20$ we get $(7-1) \times (20 - \frac{7}{2}) + (8 - 7) = 6 \times 16.5 + 1 = 100$
b. Because we are given that only 10 percent of the total pairs will have a non-zero count, we should use the tabular method. This is because the tabular method will only store the non-zero counts, and will not store the zero counts. This will save space, and will be more efficient.

Problem 3

This question is about the PCY algorithm for counting frequent pairs of items. Suppose we have six items numbered $1, 2, 3, 4, 5, 6$ and the following 12 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 $i \times j \mod 11$.
a. By any method, compute the support for each item and each pair of items
b. Which pairs hash to which buckets?
c. Which buckets are frequent?
d. Which pairs are counted on the second pass of the PCY algorithm?

Problem 3 Solution

a. The support for each item is:

- $\{1\}$: 4
- $\{2\}$: 6
- $\{3\}$: 8
- $\{4\}$: 8
- $\{5\}$: 6
- $\{6\}$: 4
- $\{1, 2\}$: 2
- $\{1, 3\}$: 3
- $\{1, 4\}$: 2
- $\{1, 5\}$: 1
- $\{1, 6\}$: 0
- $\{2, 3\}$: 3
- $\{2, 4\}$: 3
- $\{2, 5\}$: 2
- $\{2, 6\}$: 1
- $\{3, 4\}$: 4
- $\{3, 5\}$: 4
- $\{3, 6\}$: 2
- $\{4, 5\}$: 3
- $\{4, 6\}$: 3
- $\{5, 6\}$: 2

b. Using the hash function $i \times j \mod 11$ we get

- $\{1, 2\}$: 2
- $\{1, 3\}$: 3
- $\{1, 4\}$: 4
- $\{1, 5\}$: 5

- $\{1,6\}$: 6
- $\{2,3\}$: 6
- $\{2,4\}$: 8
- $\{2,5\}$: 10
- $\{2,6\}$: 1
- $\{3,4\}$: 1
- $\{3,5\}$: 4
- $\{3,6\}$: 7
- $\{4,5\}$: 9
- $\{4,6\}$: 2
- $\{5,6\}$: 8

c. The frequent buckets are 1, 2, 4, 8

d. The pairs that are counted on the second pass of the PCY algorithm are $\{2,6\}$, $\{3,4\}$, $\{1,2\}$, $\{4,6\}$, $\{1,4\}$, $\{3,5\}$, $\{2,4\}$ $\{5,6\}$

Problem 4

Read the following paper and write a brief summary of the main points in at most ONE page. You can skip the theoretical parts - Local Algorithms for Document Fingerprinting

Problem 4 Solution

The paper is about a local algorithm for document fingerprinting, which is a method of accurately identifying copied information within large sets of documents. The researchers made use of an idea called k-grams, which are substrings of length k. By dividing a document into k-grams, and then hashing each k-gram, the researchers were able to create a fingerprint for each document. Which contains positional information describing the document, and where in the document that k-gram was found. One disadvantage of this method is that there are no guarantees that matches are deteced. A k-gram shared between documents can only be found if its hash is $0 \mod p$ for some fixed p. This leads into analyzing the winnowing algorithm, which is a local algorithm for selecting fingerprints from the hashes of k-grams. The winnowing algorithm works by selecting the minimum hash value from a window of k-grams, if there is more than one hash with the minimum value, the rightmost value is selected, and then all selected hashes are saved as a document's fingerprints. The reason that the minimum hash value is used is because the minimum hash in one window is very likely to be the minimum hash in the next window, and so on. One issue with winnowing is that in low-entropy strings (like "aabbaa" where there's a lot of the same information, or raw sensor data), there are lots of identical hash values, which means that there are a lot of ties, which leaves a lot of gaps in the document's fingerprint data. Thus the winnowing algorithm had to be tweaked to be more robust, which broke ties by preferring a hash that has already been chosen by a previous algorithm. However, it's no longer a local algorithm, because it requires the entire document to be hashed before the algorithm can begin. In an experiment to test the winnowing algorithm, the researchers used 500,000 HTML documents from the web, and compared the winnowing algorithm to finding hashes with fingerprints equal to $0 \mod p$. They found that the winnowing algorithm was more accurate, with the measured density being closer the theoretical