Elijah Andrushenko

CPTS 315

Data Mining

Jana Doppa

Homework #1 (Analytical Part)

1.

1A) **4**

1B) 4/6 = **66.66%**

1C) Conf(A, B) = Support(A, B) / (Support(A) = 4/6 = **66.66%**

2.

2A) (7, 8) would be located at the position **(0, 6)** in the triangular matrix

2B) The Tabular Method would be better than the Triangular Matrix for the given situation. This can be shown mathematically. Since there are 20 entries, the Triangular Matrix will hold n^2 items or 20^2 = 400 items. Since each item is 4 bytes, 4\*400 = 1600 bytes for the Triangular Matrix. The Tabular Method will hold (n^2 - n)/2 items. (20^2 – 20)/2 = 190. Since we know that only 10% of the items will have a nonzero count, 190 \* 0.1 = 19. 12 Bytes are used for each row/item in the table. So 19\*12 = 228 bytes. **Since 228 < 1600, the Tabular Method is better for this situation, since it takes less bytes/memory to store the relevant values.**

3.

|  |  |
| --- | --- |
| Item Pairs | Support |
| (1, 2) | 2 |
| (1, 3) | 3 |
| (2, 3) | 3 |
| (2, 4) | 4 |
| (3, 4) | 4 |
| (3, 5) | 4 |
| (4, 5) | 3 |
| (4, 6) | 3 |
| (5, 6) | 2 |
| (1, 5) | 1 |
| (2, 6) | 1 |
| (1, 4) | 2 |
| (2, 5) | 2 |
| (3, 6) | 2 |

3A)

|  |  |
| --- | --- |
| Item | Support |
| 1 | 4 |
| 2 | 6 |
| 3 | 8 |
| 4 | 8 |
| 5 | 6 |
| 6 | 4 |

|  |  |
| --- | --- |
| Item Pairs | Bucket |
| (1, 2) | 2 |
| (1, 3) | 3 |
| (2, 3) | 6 |
| (2, 4) | 8 |
| (3, 4) | 1 |
| (3, 5) | 4 |
| (4, 5) | 9 |
| (4, 6) | 2 |
| (5, 6) | 8 |
| (1, 5) | 5 |
| (2, 6) | 1 |
| (1, 4) | 4 |
| (2, 5) | 10 |
| (3, 6) | 7 |

3B)

3C) Buckets **1**, **2**, **4**, and **8** are frequent

3D) **(2, 4)**, **(3, 4)**, and **(3, 5)**

4.

**Winnowing: Local Algorithms for Document Fingerprinting**

Winnowing is an efﬁcient local ﬁngerprinting algorithm that is within 33% of the lower bound. Detecting perfect copies is simple, but detecting partial copies can be more challenging. Partial copy detection has gained a lot of attention due to its many potential uses. When it came to large data sets like Web Data, Recomputing a 64-bit hash from scratch reduces the throughput of the ﬁngerprinting algorithm by more than a factor of four. Some software programs already have winnowing implemented as part of their code. One example of that would be MOSS, which is a plagiarism detection application used for primarily code or other types of texts. Due to this, MOSS is able to detect when things are supposed to be copied, like in statistics or other use of sources, rather than incorrectly labeling them as plagiarism. It does this with through fingerprinting and attaching things to identify those fingerprints as non-plagiarized sources. MOSS is great at what it does since it does not get false positives, and false negatives are always traced back to the source. In conclusion, there is now a faster and better local document fingerprinting algorithm called, winnowing.