Homework Assignment 3

Name: **Asrar Syed**

Due by: **October 6, 2024**

Course Section: **CSC 4740-002**

Q1: The Apriori algorithm makes use of *prior knowledge* of subset support properties. (10 points )

1. Explain that the (relative) support of any nonempty subset *s′* of frequent itemset *s* must be at least as great as the (relative) support of *s*.

Details:

* *s* is a frequent itemset
* *s’* is a subset of *s*

Apriori algorithm makes use of prior knowledge of subset support properties to help select pruning candidate itemsets. The key principle to the Apriori algorithm states that the support of any non-empty subset of a frequent itemset must be at least as great as the support of the itemset itself.

We can figure out if an itemset is frequent if it satisfies a given a minimum support threshold. A support of an itemset *s* can be understood as the fraction of transactions in the dataset that contain the itemset *s*. When you are calculating the support of an itemset *s*, you are checking for how many transactions contain all the items in *s*. However, if you are only checking a subset of that itemset (like *s’*), it is more likely that a larger number of transactions will contain at least the items in *s’*. This is because the condition for matching those transactions is less strict.

Essentially the support of the subset *s’* is expected to be equal to or greater than the support of the larger itemset *s*, because for every transaction that contains *s* also contains *s’*, but some transactions might contain *s’* without containing *s*.

1. Given frequent itemset L and subset s of L, prove that the confidence of the rule “*s′* ⇒ (L − *s′*)” cannot be more than the confidence of “*s* ⇒ (L − *s*)”, where *s′* is a subset of *s*.

Details:

* L is a frequent itemset
* *s* is a subset of L, and
* *s’* is a subset of *s*

Prove the **confidence** of the association rule *s′* ⇒ (L − *s′*) (rule 1) is **less than or equal to** the confidence of the rule *s* ⇒ (L − *s*) (rule 2).

Since we know that the , therefore...

This means the confidence of *s′* ⇒ (L − *s′*) cannot be more than the confidence of *s* ⇒ (L − *s*).

Q2: A database has 5 transactions. Let min sup = 60% and min conf = 80%.

|  |  |
| --- | --- |
| TID | Items Bought |
| T100 | {M, O, N, K, E, Y} |
| T200 | {D, O, N, K, E, Y } |
| T300 | {M, A, K, E} |
| T400 | {M, U, C, K, Y} |
| T500 | {C, O, K, I, E} |

**Note: min sup = 0.60 x 5 = 3, this means an itemset must appear in at least 3 transactions to be considered frequent.**

1. Find all frequent itemsets using Apriori. Please do **NOT** call the functions from existing package or library. Do it step by step by hands or by your own code. Grades are given based on intermediate results from each step. (20 points)

**1–Itemset (Single Items) Candidates**

|  |  |
| --- | --- |
| **1–Item** | **Support Count** |
| M | 3 |
| O | 3 |
| N | 2 |
| K | 5 |
| E | 4 |
| Y | 3 |
| D | 1 |
| A | 1 |
| U | 1 |
| C | 2 |
| I | 1 |

Pruning based on minimum support (remove anything with less than 3) is...

**2–Itemset Candidates**

|  |  |
| --- | --- |
| **2–Itemset** | **Support Count** |
| {M, O} | 1 |
| {M, K} | 3 |
| {M, E} | 2 |
| {M, Y} | 2 |
| {O, K} | 3 |
| {O, E} | 3 |
| {O, Y} | 2 |
| {K, E} | 4 |
| {K, Y} | 3 |
| {E, Y} | 2 |

We keep only the 2–Itemset with a support count of 3 or more

**3–Itemset Candidates**

|  |  |
| --- | --- |
| **3–Itemset** | **Support Count** |
| {M, K, E} | 2 |
| {O, K, E} | 3 |
| {O, K, Y} | 2 |
| {K, E, Y} | 2 |

We keep only the 3–Itemset with a support count of 3 or more

Since there is only one frequent 3–Itemset ({O, K, E}), there is no further combinations that can be made if we decide to attempt count for a 4–Itemset so .

Final Frequent Itemsets:

1. Find all frequent itemsets using FP-growth. Please do **NOT** call the functions from existing package or library. Do it step by step by hands or by your own code. Grades are given based on intermediate results from each step. (35 points)

**Counting the Frequency of Each Item**

|  |  |
| --- | --- |
| **Item** | **Frequency** |
| M | 3 |
| O | 3 |
| N | 2 |
| K | 5 |
| E | 4 |
| Y | 3 |
| D | 1 |
| A | 1 |
| U | 1 |
| C | 2 |
| I | 1 |

Pruning based on minimum support (remove anything with less than 3) is...

We then reorder the transactions so that the frequent items appear in descending order of support count. The support counts for the frequent items are:

* K = 5
* E = 4
* M = 3
* O = 3
* Y = 3

**Reordering each transaction**

|  |  |
| --- | --- |
| **TID** | **Filtered and Ordered Items** |
| T100 | {K, E, M, O, Y} |
| T200 | {K, E, O, Y} |
| T300 | {K, E, M} |
| T400 | {K, M, Y} |
| T500 | {K, E, O} |

**Constructing the FP-Tree**

(Null)

|

(K:5)

/ \

(E:4) (M:1)

/ \ \

(M:2) (O:2) (Y:1)

| |

(O:1) (Y:1)

|

(Y:1)

Constructing the Conditional FP-Tree, starting from the least frequent. For each frequent item, we create its conditional pattern base, which is the set of prefix paths that lead to that item.

|  |  |  |
| --- | --- | --- |
| **Items** | **Conditional Pattern Base** | **Conditional FP-Tree** |
| Y | {{K,E,M,O : 1}, {K,E,O : 1}, {K,E,O : 1}} | {K : 3} |
| O | {{K,E,M : 2}, {K,E : 1}} | {K,E : 3} |
| M | {{K,E : 2}, {K : 1}} | {K : 3} |
| E | {K : 4} | {K : 4} |
| K |  |  |

|  |  |
| --- | --- |
| **Items** | Frequent Pattern Generated |
| Y | {K,Y : 3} |
| O | {K,O : 3, E,O : 3, E,K,O : 3} |
| M | {K,M : 3} |
| E | {E,K : 3} |
| K |  |

\*Finding the frequent itemsets is more “efficient” using FP-Growth because it is able to mine conditional pattern bases which may reduce the size of the data sets that needs to be searched.

1. List all of the *strong* association rules (with support s and confidence c *satisfying* the *min requirement* ) matching the following rule, where *X* is a variable representing a customer, and item\_i denotes variables representing items (e.g., *“A”*, *“B”*, *“M”* i.e., if any customer buys A and B, then the customer will buy M)(5 points)

List of Strong Association Rules:

Q3: Give a short example to show that items in a strong association rule (Ex: min\_support=10%, min\_confidence=50%) may actually be *negatively correlated.* (10 points)

|  |  |  |  |
| --- | --- | --- | --- |
|  | *A* |  |  |
| *B* | 70 | 30 | 100 |
|  | 50 | 20 | 70 |
|  | 120 | 50 | 170 |

If we let the minimum support be 10% and the minimum confidence be 50%.

Then A⇒B is a strong rule because it satisfies minimum support and minimum confidence with a support of 70/170 = 41.17% and a confidence of 70/120 = 58.33%.

The correlation between A and B is , which is less than 1, meaning that the occurrence of A is negatively correlated with the occurrence of B.

Q4: The following contingency table summarizes supermarket transaction data, where *hot dogs* refers to the transactions containing hot dogs, refers to the transactions that do NOT contain hot dogs, *hamburgers* refers to the transactions containing hamburgers, and refers to the transactions that do NOT contain hamburgers.

|  |  |  |  |
| --- | --- | --- | --- |
|  | *hot dogs* |  |  |
| *hamburgers* | 2000 | 500 | 2500 |
|  | 1000 | 1500 | 2500 |
|  | 3000 | 2000 | 5000 |

(a) Suppose that the association rule “hot dogs -> hamburgers” is mined. Given a minimum support threshold of 25% and a minimum confidence threshold of 50%, is this association rule strong? (5 points)

For the rule the support is = 2000/5000 = 40% and the confidence is = 2000/3000 = 66.7%.

This means the association rule is strong.

1. Based on the given data, is the purchase of hot dogs independent of the purchase of hamburgers? If not, what kind of correlation relationship (positively or negatively) exists between the two? (5 points)

This means that the purchase of hotdogs is not independent of the purchase of hamburgers and there is a positive correlation between the two purchases.

1. Compute and compare the **all confidence, max confidence, Kulczynski, cosine measures, and lift** on the given data (definition of each metric are in the lecture slides) (10 points)

**All Confidence**

Support for hot dogs P(hotdogs) = 3000/5000 = 0.6

Support for hamburgers P(hamburgers)=2500/5000 = 0.5

**Max Confidence**

Confidence of A→B (hotdogs → hamburgers)

Confidence of B→A (hamburgers → hotdogs)

**Kulczynski**

**Cosine Measures**

**Lift**