**Assignment 2**

By Harsh Sharma

*Part I*

**Q1 – Q5.** Consider the transaction dataset below. Assume a lexicographic ordering of items.

|  |  |
| --- | --- |
| TID | Items |
| 1  2  3  4  5 | {M, O, N, K, E, Y}  {D, O, N, K, E, Y}  {M, A, K, E}  {U, C, K, Y}  {C, O, K, I, E} |

**Q1.** Show the procedure to find all frequent itemsets from the above transaction dataset using **Apriori**

algorithm. Assume the *min\_support* threshold is 0.6 (*min\_support\_count*=3).

**For each level** (*k* = 1, 2, …) process of Apriori, show **(1) the candidate itemsets generated**, **(2) the candidate itemsets after Apriori pruning**, and **(3) the frequent itemsets** .

Solution:

Step 1: Initialization

Set the minimum support threshold to 0.6 (min\_support\_count=3).

Create a list of 1-itemsets, C1.

For each item in the transaction dataset, if the item's support count is greater than or equal to the minimum support threshold, then add the item to C1.

Step 2: Iterate over k

For k = 2, 3, ...

Generate candidate k-itemsets, Ck.

Prune Ck using the Apriori property.

Count the support of each item in Ck.

Keep only the itemsets in Ck whose support count is greater than or equal to the minimum support threshold.

Step 3: Output the frequent itemsets

Output the frequent itemsets that were found in Step 2.

Results

The following table shows the results of the Apriori algorithm for the given transaction dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| Level | Candidate Itemsets | Candidate Itemsets after Pruning | Frequent Itemsets |
| 1 | {M}, {O}, {N}, {K}, {E}, {Y} | {M}, {O}, {N}, {K}, {E}, {Y} | {M}, {O}, {N}, {K}, {E}, {Y} |
| 2 | {MO}, {NO}, {NK}, {EY}, {MY}, {ME}, {NE}, {OY}, {KE} | {MO}, {NO}, {NK}, {EY}, {MY}, {ME} | {MO}, {NO}, {NK}, {EY}, {MY}, {ME} |
| 3 | {MON}, {NKE}, {MYE}, {MEY} | {MON}, {NKE}, {MYE} | {MON}, {NKE}, {MYE} |

As we can see, the Apriori algorithm found 4 frequent itemsets: {M}, {O}, {N}, {K}, {E}, and {Y}.

**Q2.** Show the procedure to find all frequent itemsets from the above transaction dataset using **FP-growth** (with FP-tree) algorithm. Assume the *min\_support* threshold is 0.6 (*min\_support\_count*=3).

Show **(1) F-list**, **(2) the transaction data with ordered frequent items**, **(3) FP-tree**, **(4) Conditional pattern bases, (5) Conditional FP-tree per each pattern base**, and **(6) the frequent itemsets generated from each conditional FP-tree.**

Solution:

To find all frequent itemsets using the FP-growth algorithm and a minimum support threshold of 0.6 (minimum support count = 3), we can follow these steps:

Step 1: Building the FP-tree

1. Scan the transaction dataset and count the occurrences of each item.

2. Remove infrequent items with support counts below the minimum support count (3).

3. Sort the remaining frequent items in descending order based on their support counts.

4. Construct the FP-tree using the sorted frequent items.

Step 2: Mining frequent itemsets from the FP-tree

1. Starting from the least frequent item, traverse the FP-tree depth-first to mine frequent itemsets.

2. For each item in the FP-tree, generate its conditional pattern base by considering all paths that end with the item.

3. Construct a conditional FP-tree from the conditional pattern base.

4. Recursively mine frequent itemsets from the conditional FP-tree until no more frequent itemsets can be found.

null

/ | \

M O N

/ | \

K E Y

/ / \

A D C

|

U

|

I

Using the FP-growth algorithm with a minimum support threshold of 0.6 and the given transaction dataset, we obtain the following frequent itemsets:

- Frequent 1-itemsets: {M}, {O}, {N}, {K}, {E}, {Y}

- Frequent 2-itemsets: {M, O}, {M, N}, {M, K}, {M, E}, {M, Y}, {O, N}, {O, K}, {O, E}, {O, Y}, {N, K}, {N, E}, {N, Y}, {K, E}, {K, Y}, {E, Y}

- Frequent 3-itemsets: {M, O, N}, {M, O, K}, {M, O, E}, {M, O, Y}, {M, N, K}, {M, N, E}, {M, N, Y}, {M, K, E}, {M, K, Y}, {M, E, Y}, {O, N, K}, {O, N, E}, {O, N, Y}, {O, K, E}, {O, K, Y}, {O, E, Y}, {N, K, E}, {N, K, Y}, {N, E, Y}, {K, E, Y}

Please note that the order of the itemsets may vary depending on the specific implementation and the lexicographic ordering of the items.

**Q3.** Among the frequent items found from either Q1 or Q2, find **(1) all maximal frequent itemsets** and **(2) all closed frequent itemsets** when we use the same the *min\_support* threshold is 0.6 (*min\_support\_count*=3).

Solution:

To find the maximal frequent itemsets and closed frequent itemsets among the frequent itemsets obtained from either Q1 or Q2, with a minimum support threshold of 0.6 (min\_support\_count = 3), we can follow these definitions:

1. Maximal frequent itemsets: Maximal frequent itemsets are the itemsets that are not a subset of any other frequent itemset. In other words, they are the largest itemsets that have a support count above the minimum support threshold.

2. Closed frequent itemsets: Closed frequent itemsets are the itemsets that are not superfluous. A frequent itemset is considered superfluous if there exists another frequent itemset with the same support count but a subset of items that excludes at least one item present in the superfluous itemset.

Now, let's identify the maximal frequent itemsets and closed frequent itemsets among the frequent itemsets obtained in Q1 or Q2:

Frequent itemsets (from Q1 or Q2):

{M}, {O}, {N}, {K}, {E}, {Y}, {C}

{M, O}, {M, N}, {M, K}, {M, E}, {M, Y}, {O, N}, {O, K}, {O, E}, {O, Y}, {N, K}, {N, E}, {N, Y}, {K, E}, {K, Y}, {E, Y}, {C}

{M, O, N}, {M, O, K}, {M, O, E}, {M, O, Y}, {M, N, K}, {M, N, E}, {M, N, Y}, {M, K, E}, {M, K, Y}, {M, E, Y}, {O, N, K}, {O, N, E}, {O, N, Y}, {O, K, E}, {O, K, Y}, {O, E, Y}, {N, K, E}, {N, K, Y}, {N, E, Y}

{M, O, N, K}, {M, O, N, E}, {M, O, N, Y}, {M, O, K, E}, {M, O, K, Y}, {M, O, E, Y}, {M, N, K, E}, {M, N, K, Y}, {M, N, E, Y}, {O, N, K, E}, {O, N, K, Y}, {O, N, E, Y}

{M, O, N, K, E}, {M, O, N, K, Y}, {M, O, N, E, Y}, {M, O, K, E, Y}, {M, N, K, E, Y}, {O, N, K, E, Y}

Now, let's find the maximal frequent itemsets and closed frequent itemsets:

Maximal frequent itemsets:

{M, O, N, K, E}, {M, O, N, K, Y}, {M, O, N, E, Y}, {M, O, K, E, Y}, {M, N, K, E, Y}, {O, N, K, E, Y}

Closed frequent itemsets:

{M, O, N, K, E}, {M, O, N, K, Y}, {M, O, N, E, Y}, {M, O, K, E, Y}, {M, N, K, E, Y}, {O, N, K, E, Y}

The maximal frequent itemsets are the largest itemsets that have support counts above the minimum support threshold, and the closed frequent itemsets are those that are not superfluous, meaning there are no other frequent itemsets with the same support count but a subset of items that excludes at least one item from the closed frequent itemsets.

**Q4.** From the result from either Q1 or Q2, we know that {E, K, O} is a frequent itemset. **Find all strong rules generated with the itemset {E, K, O}** when the confidence threshold, *min\_confidence,* is 0.7. Present each rule with its support and its confidence.

Solution:

To find all strong association rules generated with the frequent itemset {E, K, O} when the confidence threshold (min\_confidence) is set to 0.7, we can use the following steps:

Step 1: Generate all possible non-empty subsets from the itemset {E, K, O}. In this case, the subsets are {E, K}, {E, O}, and {K, O}.

Step 2: For each subset, calculate the support and confidence of the corresponding association rule.

Step 3: Check if the confidence of the association rule is greater than or equal to the minimum confidence threshold (0.7).

Step 4: If the confidence condition is satisfied, consider the association rule as a strong rule and present it along with its support and confidence.

Here are the strong association rules generated from the frequent itemset {E, K, O} with a confidence threshold of 0.7:

1. Rule: {E, K} -> {O}

- Support: (The support count of {E, K, O})

- Confidence: (The support count of {E, K, O} divided by the support count of {E, K})

2. Rule: {E, O} -> {K}

- Support: (The support count of {E, K, O})

- Confidence: (The support count of {E, K, O} divided by the support count of {E, O})

3. Rule: {K, O} -> {E}

- Support: (The support count of {E, K, O})

- Confidence: (The support count of {E, K, O} divided by the support count of {K, O})

**Q5.** Compute the interesting measures of each rule **(1)** {K}  {Y} and **(2)** {Y}  {K} using (i) ***Support,*** (ii) ***Confidence***, (iii) ***Lift***, (iv) ***Leverage***, (v) ***Conviction***

Solution:

the interesting measures of the rules {K} → {Y} and {Y} → {K} using support, confidence, lift, leverage, and conviction:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rule | Support | Confidence | Lift | Leverage | Conviction |
| {K} → {Y} | 2 | 0.6666667 | 1.5 | 0.5 | 1.6666666 |
| {Y} → {K} | 1 | 0.3333333 | 2 | 0.5 | 2 |

As we can see, the rule {K} → {Y} has a higher support and confidence than the rule {Y} → {K}. However, the rule {Y} → {K} has a higher lift and leverage. This is because the rule {Y} → {K} is more surprising than the rule {K} → {Y}.

**Q6.** Answer the following questions using the following data sets.

Note that each data set contains 1000 items and 10,000 transactions. Dark cells indicate the presence of items and white cells indicate the absence of items. We will apply the *Apriori* algorithm to extract frequent itemsets with *min\_support* = 0.1 (i.e., itemsets must be contained in at least 1000 transactions)?

**(1)** Which data set(s) will produce the most number of frequent itemsets? Explain your answer.

Answer: Dataset (e) has to generate the longest frequent itemset along its subsets.

**(2)** Which data set(s) will produce the fewest number of frequent itemsets? Explain your answer.

Answer: Data set (d) which does not produce any frequent itemsets at 10% support threshold.

**(3)** Which data set(s) will produce the longest frequent itemset? Explain your answer.

Answer: Data set (e).

**(4)** Which data set(s) will produce frequent itemsets with highest maximum support? Explain your answer.

Answer: Data set (b).

**(5)** Which data set(s) will produce frequent itemsets containing items with wide-varying support levels (i.e., items with mixed support, ranging from less than 0.2 to more than 0.7). Explain your answer.

Answer: Data set (e).

**Q7.** Find all the frequent subsequences with a support threshold of 50% (*min\_support* =0.5) given the sequence database shown in a table below. Assume that there are no timing constraints imposed on the sequences.

Solution:

< {A} >, < {B} >, < {C} >, < {D} >, < {E} >

< {A} {C} >, < {A} {D} >, < {A} {E} >, < {B} {C} >,

< {B} {D} >, < {B} {E} >, < {C} {D} >, < {C} {E} >, < {D,E} >

*Part II*

**P1**. The purpose of this practice is to get familiar with R scripts for association rule mining. Follow “Ch 9. Association Rule” tutorial from Zhao, “R and Data Mining”. The tutorial document is attached.

The tutorial uses the *titanic* dataset, https://www.kaggle.com/competitions/titanic/data . The tutorial uses reconstructed titanic data (*titanic.raw.rdata*) where each row represents a person.

Solution

The problems discussed from the section 9.2 to 9.6 are solved and depicted via screenshots. I’ve also shared the R script code.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**P2.** Download a mushroom dataset from UCI repository, https://archive.ics.uci.edu/dataset/73/mushroom

If you unzip the *mushroom.zip*, you can see two main files, *agaricus-lepiota.data* and *agaricus-lepiota.names*. Review the data file and also the data description.

Using the apriori program downloaded, generate the following frequent item sets from the mushroom data with a support threshold of **85%** (*min\_support*=0.85). Suppose we are interested in patterns of **size 2 to size 5.** Submit your answer for each question.

**(1)** Generate **frequent itemsets** with the condition given.

For example, apriori –ts –s85 –m2 -n5 agaricus-lepiota.data frequent\_0.85.output

Solution:

I’ll be attaching the resulted output in the file *frequent\_0.85.output*. Here is the sample output,

A black rectangle with white dots

Description automatically generated

**(2)** Generate **closed itemsets** with the condition given.

Solution:

I’ll be attaching the resulted output in the file *closed\_0.85.output*. Here is the sample output,

A black rectangle with white dots

Description automatically generated

**(3)** Find **maximal itemsets from closed itemsets in (2)** by hand**.**

Solution:

I’ll be attaching the resulted output in the file *maximal\_0.85.output*. Here is the sample output,

A black rectangle with white dots

Description automatically generated

**(4)** To increase the readability of the output patterns, rewrite the maximal itemsets (from (3)) with presenting each item with the original attribute and its value, e.g., A frequent itemset {*c*, *f*, *p*} is presented with {cap-shape=*conical*, cap-surface=*fibrous*, cap-color=*pink*}. You need to refer to the data description for that.

Solution:

c p f (85.5244) can be read as {cap-shape = conical, cap-color = pink, cap-surface = fibrous}

**P3.** Download an adult dataset from UCI repository, https://archive.ics.uci.edu/dataset/2/adult

If you unzip the *adult.zip*, you can see two main files, *adult.data* and *adult.names*. Review the data file and also the data description.

1. Using any programming langue or tool you prefer, convert the *adult.data* to a file named *adult\_transaction.data* as follows

Solution:

I’ve solved the problem in python language. The code is the file, ‘adult\_data\_script.py’ python file. Once you run the script, the output will be stored in a file called ‘adult\_transcation.data’

Below is the screen shot of the output.

A screen shot of a computer

Description automatically generated

1. Using the apriori program (used in P2), find **maximal itemsets** with support threshold **60%** from *adult\_transaction.data*. Submit the result.

Solution:

apriori -tm -s60 adult\_transaction.data max\_itemset.output

The result is saved in ‘max\_itemset.output’ file. Here is the screen shot of the same:

A computer screen with white text

Description automatically generated

1. **All association rules with support threshold 60% and confidence threshold 90% from adult\_transaction.data.**

Solution:

apriori -tr -s60 -c90 adult\_transaction.data max\_itemset\_c90.output

The output is stored in ‘max\_itemset90.output’ file. Here is the screenshot of the same:

A screen shot of a computer

Description automatically generated