hw01_written

March 15, 2021

1 HW 1 – Frequent Pattern Mining

Name: Robb Alexander Class: CSCI349
Semester: 2021SP
Instructor: Brian King

1.1 Exercise 1 – The apriori algorithm

a) Find all frequent itemsets using the Apriori algorithm. Show your work.

```
min_sup = 60%
min_conf = 80%

1-itemsets:
{M} 0.6
{O} 0.6
{N} 0.4 min_sup pruned
{K} 1.0
{E} 0.8
{Y} 0.6
{D} 0.2 min_sup pruned
{A} 0.2 min_sup pruned
{U} 0.2 min_sup pruned
{C} 0.4 min_sup pruned
{I} 0.2 min_sup pruned

2-itemsets:
{M, O} 0.2 min_sup pruned
{M, N} --- apriori pruned
{M, K} 0.6
{M, E} 0.4 min_sup pruned
{M, Y} 0.2 min_sup pruned
{M, A} --- apriori pruned
{M, U} --- apriori pruned
{M, C} --- apriori pruned
{O, N} --- apriori pruned
{O, K} 0.6
{O, E} 0.6
{O, Y} 0.4 min_sup pruned
```

```
{O, D} --- apriori pruned
{O, C} --- apriori pruned
{O, I} --- apriori pruned
{N, *} --- apriori pruned
{K, E} 0.8
{K, Y} 0.6
{K, A} --- apriori pruned
{K, C} --- apriori pruned
{K, I} --- apriori pruned
{E, Y} 0.4 min_sup pruned
{E, D} --- apriori pruned
{E, A} --- apriori pruned
{E, I} --- apriori pruned
{E, C} --- apriori pruned
{D, *} --- apriori pruned
{A, *} --- apriori pruned
{U, *} --- apriori pruned
{C, *} --- apriori pruned
{I, *} --- apriori pruned

3-itemsets:
{M, K, O} 0.2 min_sup pruned
{M, K, E} 0.4 min_sup pruned
{M, K, Y} 0.4 min_sup pruned
{O, K, E} 0.6
{O, K, Y} 0.4 min_sup pruned

itemsets:
{M} 0.6
{O} 0.6
{K} 1.0
{E} 0.8
{Y} 0.6
{M, K} 0.6
{O, K} 0.6
{O, E} 0.6
{K, E} 0.8
{K, Y} 0.6
{O, K, E} 0.6
```

b) What is a closed frequent itemset? List Closed frequent itemset: An itemset that is frequent but does not have the same support value as its superset.

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

c) What is a max frequent itemset? List Closed frequent itemset: An itemset that is frequent but does not have any supersets above that is considered frequent.

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

d) Generate all strong association rules

```
{M} -> {K}  confidence: 1.0  lift: 1.0
{K} -> {M}  confidence: 0.6  lift: 1.0
{K} -> {E}  confidence: 0.8  lift: 1.0
{E} -> {K}  confidence: 1.0  lift: 1.0
{O} -> {E}  confidence: 1.0  lift: 1.25
{E} -> {O}  confidence: 0.75  lift: 1.25
{O} -> {K}  confidence: 1.0  lift: 1.0
{K} -> {O}  confidence: 0.6  lift: 1.0
{K} -> {Y}  confidence: 0.6  lift: 1.0
{Y} -> {K}  confidence: 1.0  lift: 1.0
{O, K} -> {E}  confidence: 1.0  lift: 1.25
{K, E} -> {O}  confidence: 0.75  lift: 1.25
{O, E} -> {K}  confidence: 1.0  lift: 1.0
{K} -> {O, E}  confidence: 0.6  lift: 1.0
{O} -> {K, E}  confidence: 1.0  lift: 1.25
{E} -> {K, O}  confidence: 0.75  lift: 1.25
```

e) What is the strongest rule output

```
{O, K} -> {E}  confidence: 1.0  lift: 1.25
{O} -> {K, E}  confidence: 1.0  lift: 1.25
```

These are the strongest rules because they have the highest confidence and also the most dependency due to the highest lift.

1.2 Exercise 2 – The FP-growth algorithm

a) Create the ordered initial F-list

```
F-list:
{K} 1.0
{E} 0.8
{M} 0.6
{O} 0.6
{Y} 0.6
{N} 0.4
{C} 0.4
{D} 0.2
{A} 0.2
{U} 0.2
{I} 0.2
```

{K, E, M, O, Y N, C, D, A, U, I}

b) Create the initial FP-tree

[3]:
```
from IPython import display
# display.Image("/Users/rale/Documents/Programming/csci349_2021sp/hw/img.png")
```

c) Execute the FP_growth algorithm

[4]:
```
# display.Image("/Users/rale/Documents/Programming/csci349_2021sp/hw/img2.png")
```

d) Compare and contrast the computational requirements Apriori feels faster to do on paper, but the tree implimentation makes the space used much less than every single one of the aprioris which needs to be pruned. In the end the fpgrowth is a more modern approach with less speed and space saved. fpgrowth also only needs two scans versus the multiscan of the apriori. All of this was done to prevent the need to do subset itemset generation.

1.3 Exercise 3 – The Eclat algorithm

a) Convert the dataset in Exercise 1 to a vertical data format

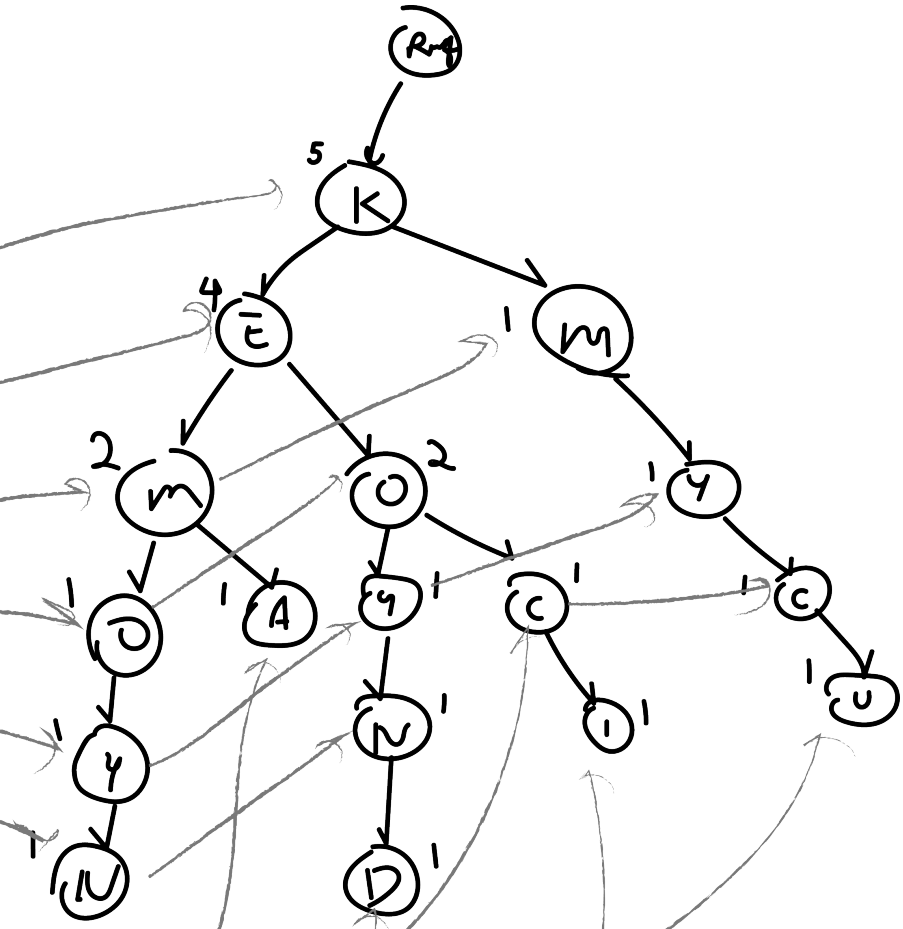
```
{K} : [T100, T200, T300, T400, T500]
{E} : [T100, T200, T300, T500]
{O} : [T100, T200, T500]
{M} : [T100, T300, T400]
{Y} : [T100, T200, T400]
{N} : [T100, T200]
{C} : [T400, T500]
{D} : [T200]
{A} : [T300]
{U} : [T400]
{I} : [T500]
```

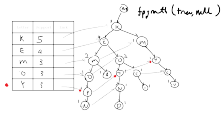
b) Find the frequent itemsets using the Eclat algorithm.

```
min_sup = 3
k = 1
{K}: [T100, T200, T300, T400, T500]
{E}: [T100, T200, T300, T500]
{O}: [T100, T200, T500]
{M}: [T100, T300, T400]
{Y}: [T100, T200, T400]

k = 2
{K, E}: [T100, T200, T300, T500]
{K, O}: [T100, T200, T500]
{K, M}: [T100, T300, T400]
{K, Y}: [T100, T200, T400]
```

Letter	Support	Link
K	5	
E	4	
M	3	
O	3	
Y	3	
N	2	
C	2	
D	1	
A	1	
U	1	
I	1	





CPB

KEMO : 1

OEK : 1

MK : 1

K : 3

~~E : 4~~

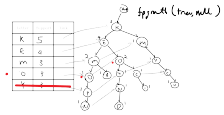
~~M : 2~~

~~O : 1~~

fpgrowth (tree, Y)

{Y, K : 3}

{Y : 3}



CPB

MEK : 1

EK : 2

K : 3

~~E : 3~~

fpgrowth (tree, O)

{O : 3}

CPB EO

K : 3

fpgrowth (tree, OK)

{K, OE : 3}

{O, E : 3}

CPB

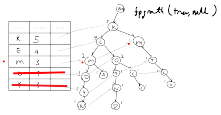
MEK : 1

EK : 2

~~K : 3~~

~~E : 3~~

{O, K : 3}



CPB :

EK : 2

K : 1

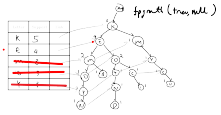
K : 3

~~E : 4~~

fpgrowth (tree, M)

{M, K : 3}

{M : 3}



CPB

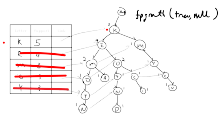
K : 4

K : 4

fpgrowth (tree, E)

{KE : 4}

{E : 4}



CPB

none

{K : 5}

```
{E, O}: [T100, T200, T500]
{E, M}: [T100, T300]
{E, Y}: [T100, T200]
{O, M}: [T100]
{O, Y}: [T100, T200]
{M, Y}: [T100, T400]

k = 3
{K, E, O}: [T100, T200, T500]
{K, E, M}: [T100, T300]
{K, E, Y}: [T100, T200]
{K, O, M}: [T100]
{K, O, Y}: [T100, T200]
{K, M, Y}: [T100, T400]
{E, O, M}: [T100]
{E, O, Y}: [T100, T200]
{E, M, Y}: [T100]
{O, M, Y}: [T100]

k = 4
{K, E, O, M}: [T100]
{K, E, O, Y}: [T100, T200]
{K, E, M, Y}: [T100]
{K, O, M, Y}: [T100]
{E, O, M, Y}: [T100]

{K}: [T100, T200, T300, T400, T500]
{E}: [T100, T200, T300, T500]
{O}: [T100, T200, T500]
{M}: [T100, T300, T400]
{Y}: [T100, T200, T400]
{K, E}: [T100, T200, T300, T500]
{K, O}: [T100, T200, T500]
{K, M}: [T100, T300, T400]
{K, Y}: [T100, T200, T400]
{E, O}: [T100, T200, T500]
{K, E, O}: [T100, T200, T500]
```

1.4 Exercise 4 – Correlation

.	A	~A	total
B	65	40	105
~B	35	10	45
total	100	50	150

a) Compute support and confidence for the rule A->B. Is this a strong rule?

```
min_sup = 0.4
min_conf = 0.6

support_AB = 65/150 = 43.3%
support_A = 100/150 = 66.6%
confidence = 0.433/0.666 = 65%
```

This is a strong rule, the confidence and support is high enough over the threshold, but not extremely strong since it is close to the threshold.

b) What does the lift measure tell us? Compute lift(A,B). What does this suggest about the occurrence of A and B? What does it suggest about the rule? Lift tells us about the dependency between the two variables

```
support_B = 105/150 = 70%
lift = 0.433/(0.666*0.7) = 0.9287859288
```

This means that there is a significant correlation between A and B, but negatively, meaning that A prevents B instead of encouraging.

.	A	~A
B	70	35
~B	30	15

c) Compute the expected values for each observed value above, showing your results in a table.

d) Compute the X2 correlation coefficient using the table above and your expected values you computed in the previous question. Does the value imply dependency among A and B?

```
5*5/70 = 0.3571428571
5*5/35 = 0.7142857143
5*5/30 = 0.8333333333
5*5/15 = 1.6666666667
```

$$\chi^2 = 3.57$$

With 1 degree of freedom, 3.57 lies between 0.10 and 0.5, which implies there is dependence but very slight.

e) Consider the rule A -> NOT B. What is the support, confidence and lift for this rule?

```
support_A~B = 35/150 = 23.3%
support_A = 100/150 = 66.6%
confidence = 0.233/0.666 = 35%
```

```
support_~B = 45/150 = 30%
lift = 0.233/(0.666*0.3) = 1.166
```

f) What is the confidence and lift of the rule NOT B -> A ? You should notice there is an imbalance between your answer here and the previous question. Which rule is stronger? Why?

```
support_~BA = 35/150 = 23.3%
support_~B = 45/150 = 30%
confidence = 0.233/0.30 = 77%

support_A = 100/150 = 66.6%
lift = 0.233/(0.666*0.3) = 1.166
```

This rule is stronger, since the confidence is more than double. This is because the inverse is not the same, and this shows that ~B implies A moreso and not the other way around. #### g) Compute the Kulczynski measure for the items A and NOT B.

$$K = \frac{1}{2}\big(P(A|\neg B) + P(\neg B|A)\big)$$

```
A given ~B = 35/100 = 0.35
~B given A = 35/45 = 0.78

K = 0.564
```

h) Compute the imbalance ratio (IR) on A and NOT B. What do these results say? Does the result confirm your observations on questions e) and f) above?

$$IR = \frac{\big|support(A) - support(\neg B)\big|}{support(A) + support(\neg B) - support(A \cup \neg B)}$$

```
support_A = 100/150 = 66.6%
support_~B = 45/150 = 30%
support_A~B = 35/150 = 23.3%

IR = (0.66 - 0.3) / (0.66 + 0.3 - 0.233)
   = 0.4951856946
```

These results show that there indeed in a coorlation between these two observations. The K is close to 0.5 which means there is low dependency. The IR is not close to 0, which then we can take as being significant and shows a form of skewness.

1.5 Exercise 5 – Distributed mining

Suppose that a large store has a transaction database that is distributed among four locations. Transactions in each component database have the same format, namely T_j: {i_1...i_m} where Tj is a transaction identifier, and i_k (1<=k<=m) is the identifier of an item purchased in the transaction. Propose an efficient algorithm to mine global association rules (without considering multilevel associations). Partition:

Scan DB Only Twice, we can use the apriori principles to help in pruning candidates across the four dbs without the multilevel associations.

Source: Section V