TDT4300 - Data Warehousing and Data Mining

Assignment 2

Martin Johannes Nilsen

17th February 2022

1 Apriori Algorithm

1.1 Generation of frequent itemsets

First of all, we start with generating the 1-itemset. This is done by counting the total amount of transactions each item is present in, which will be the support count. For the table given in the assignment, the 1-itemset will look like this:

Item	Support count σ
H	4
I	4
C	3
B	2
K	2
U	1

Table 1: 1-itemsets of transactions 1

It is given in the assignment that we are to find all frequent itemsets with minimum support 33.33%, being a support count of 2. Thus, we can remove U as a more specialized itemset is not likely to be more frequent than the general item. We continue on and generate the 2-itemset.

Item		Support count σ
H	I	2
H	C	2
H	B	2
H	K	1
I	C	3
I	B	0
I	K	1
C	B	0
C	K	0
B	K	1

Table 2: 2-itemsets of transactions 1

Again we remove the itemsets with support count less than 2, and continue on generating the 3-itemset.

Item			Support count σ
H	I	C	2
H	I	B	0

Table 3: 3-itemsets of transactions 1

Where we see that the only itemset of 3 items is {H,I,C}.

1.2 Find all association rules based on {H,I,C}

Based on the table of the last task, and the information given in the assignment, we are now tasked to find all the association rules based on the set {H,I,C}. The first step in the process is finding all the combinations of this set, before we calculate the confidence of each association rule.

- $\{H, I\} \rightarrow \{C\}$
- $\{H, C\} \rightarrow \{I\}$
- $\{I, C\} \rightarrow \{H\}$
- $\{I\} \rightarrow \{H, C\}$
- $\{C\} \rightarrow \{H, I\}$
- $\{H\} \rightarrow \{C, I\}$

For calculating the confidence we use the following equation

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

giving us the following table below. Keep in mind that $\sigma(X \cup Y)$ is 2 for each of them as shown in a former task, in the form of $\sigma(\{H, I, C\}) = 2$.

Rule $X \rightarrow Y$	$\sigma(X)$	Confidence	Accepted
$\{H, I\} \rightarrow \{C\}$	2	$\frac{2}{2} = 1$	Yes
$\{H, C\} \rightarrow \{I\}$	2	$\frac{2}{2} = 1$	Yes
$\{I, C\} \rightarrow \{H\}$	3	$\frac{2}{3} = 0.6\bar{6}$	Yes
$\{I\} \rightarrow \{H, C\}$	4	$\frac{2}{4} = 0.5$	No
$\{C\} \rightarrow \{H, I\}$	3	$\frac{2}{3} = 0.6\bar{6}$	Yes
$\{H\} \rightarrow \{C, I\}$	4	$\frac{2}{4} = 0.5$	No

Table 4: Confidence and acceptance of each association rule

In conclusion, these 4 association rules will be generated:

- $\{H, I\} \rightarrow \{C\}$
- $\{H, C\} \rightarrow \{I\}$
- $\{I, C\} \rightarrow \{H\}$
- $\{C\} \rightarrow \{H, I\}$

2 FP-Growth Algorithm

We are tasked to use the Frequent Pattern Growth algorithm to discover the frequent itemsets in the given dataset (transactions 2). The first thing we need to do is to create the table of 1itemsets with decreasingly sorted support counts.

Item	Support count σ
b	7
e	7
d	6
f	3
a	1
c	1
g	1
h	1
i	1
j	1

Table 5: 1-itemsets of transaction 2

Further on, we are to sort the transactions into the order of the table above.

TID	Items
T1	beg
T2	bdi
T3	bedf
T4	eda
T5	ed
T6	bdj
T7	bedfc
T8	bedf
T9	beh

Table 6: Sorted transactions

The next thing up is to generate the tree. Keep in mind that when we come to an item less frequent than the minsup count (2), we prune it. For our case, we prune the items {a,c,g,h,i,j}.

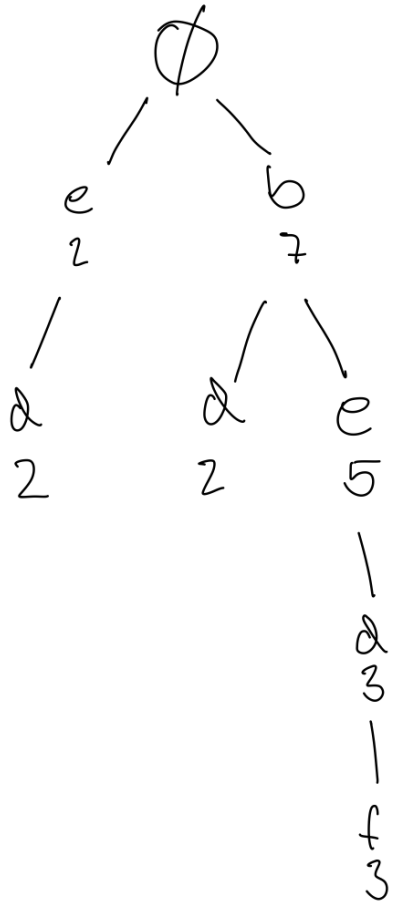


Figure 1: FP Growth tree

The final thing left to do then is to create a table for the frequent patterns.

Items	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns
b	{Ø:7}	<Ø:7>	{b:7}
e	{Ø:2, b:5}	<Ø:2, b:5>	{e:2, be:5}
d	{e:2, b:2, be:3}	<b:5, e:5>	{bd:5, ed:5, bed:3}
f	{bed:3}	<b:3, e:3, d:3>	{bedf:3}

Explanation of the columns above:

Conditional Pattern Base is the base path of each of the occurences in the tree. That is, the path up until the correct item, with the corresponding support count for the item in the end.

Conditional FP-tree is as I have understood it the summation of the support counts for each item in the conditional pattern base.

Frequent Patterns is the frequent patterns we find along the tree, where you can add the item to the conditional pattern base.

3 KNIME

For solving this task, I started out by downloading KNIME and the WEKA 3.7 plugin. First I had to create a new wokflow, and searched the node repository for an arff-reader, and the two algorithms we are supposed to run, being apriori and fpgrowth. After dragging these into the workflow, the next thing was to connect both algorithms to the arff reader, setup the path to the file containing the data, and configure the apriori properties given in the assignment. I sat the *lowerBoundMinSupport* to 0.5, and *minMetric* to 0.8, and was ready to run both algorithms. The workflow used, and output in both views are presented below.

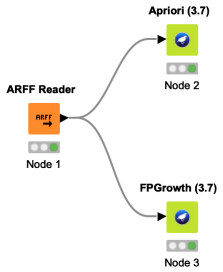


Figure 2: Workflow

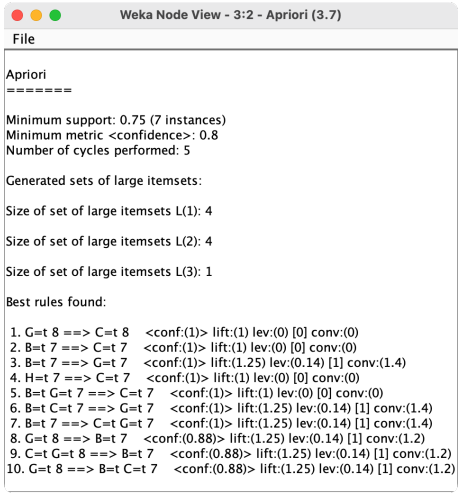


Figure 3: Apriori

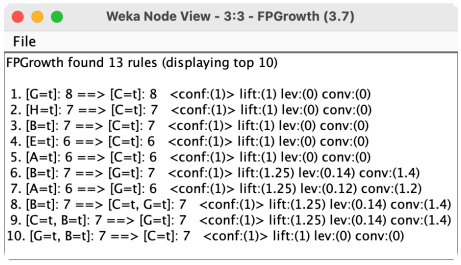


Figure 4: FP Growth

4 Compact Representation of Frequent Itemsets

In the assignment we are given the compact representation of a frequent itemset in the table below.

Closed Frequent Itemsets	Support count
{b}	10
{d}	13
{a, d}	11
{b, d}	7
{b, e}	8
{d, e}	6
{a, b, e}	7
{a, c, d}	6
{b, d, e}	4
{a, c, d, e}	5

Table 7: Closed frequent itemsets with IDs

We are tasked to generate all frequent itemsets including the support counts. For solving this I am going to take use of algorithm 6.4 from page 356 in Tan et al. The algorithm is given as:

Algorithm 6.4 Support counting using closed frequent itemsets.

1: Let C denote the set of closed frequent itemsets
2: Let $k_{\max}$ denote the maximum size of closed frequent itemsets
3: $F_{k_{\max}} = \{f | f \in C,\ |f| = k_{\max}\}$ {Find all frequent itemsets of size $k_{\max}$.}
4: **for** $k = k_{\max} - 1$ downto 1 **do**
5: $F_k = \{f | f \subset F_{k+1},\ |f| = k\}$ {Find all frequent itemsets of size k.}
6: **for** each $f \in F_k$ **do**
7: **if** $f \notin C$ **then**
8: $f.support = \max\{f'.support | f' \in F_{k+1},\ f \subset f'\}$
9: **end if**
10: **end for**
11: **end for**

We start off by finding $k_{max} = 4$ as the maximum itemset is $\{a, c, d, e\}$. The set of frequent itemsets of size k_{max} is given as $F_{k_{max}} = \{\{a, c, d, e\}\}$.

Then we go through a loop from $k = (k_{max} - 1)$ down to 1, where we find all frequent itemsets of size k and take the maximum support count for the more specific itemsets which yields the maximum support count it can have. The algorithm will be performed on the next side.

$k = 3$:

We find all[1] frequent itemsets of size 3 connected to the former closed frequent itemset: $\{a, c, e\}, \{a, d, e\}, \{c, d, e\}$

Calculate the support:

ace.support = max{acde.support} = max{5} = 5
ade.support = max{acde.support} = max{5} = 5
cde.support = max{acde.support} = max{5} = 5

$k = 2$:

We find all[2] frequent itemsets of size 2 connected to the former frequent itemsets: $\{a, b\}, \{a, c\}, \{a, e\}, \{c, d\}, \{c, e\}$

Calculate the support:

ab.support = max{abe.support} = max{7} = 7
ac.support = max{acd.support, ace.support} = max{6, 5} = 6
ae.support = max{abe.support, ace.support, ade.support} = max{7, 5, 5} = 7
cd.support = max{acd.support, cde.support} = max{6, 5} = 6
ce.support = max{ace.support, cde.support} = max{5, 5} = 5

$k = 1$:

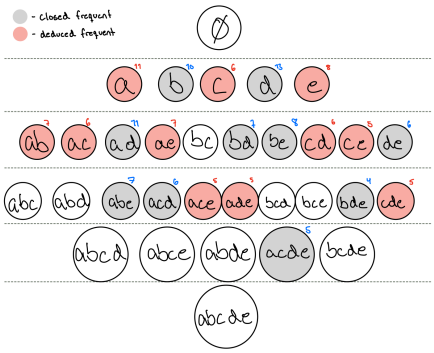
We find all[3] frequent itemsets of size 2 connected to the former frequent itemsets: $\{a\}, \{c\}, \{e\}$

Calculate the support:

a.support = max{ab.sup, ac.sup, ad.sup, ae.sup} = max{7, 6, 11, 7} = 11
c.support = max{ac.support, cd.support, ce.support} = max{6, 6, 5} = 6
e.support = max{ae.sup, be.sup, ce.sup, de.sup} = max{7, 8, 5, 6} = 8

This gives us all the frequent itemsets:

a:11, b:10, c:6, d:13, e:8
ab:7, ac:6, ad:11, ae:7, bd:7, be:8, cd:6, ce:5, de:6
abe:7, acd:6, ace:5, ade: 5, bde:4, cde:5
acde: 5



[1]Have been given the support for $\{a, b, e\}, \{a, c, d\}, \{b, d, e\}$ in the closed frequent itemsets
[2]Have been given the support for $\{a, d\}, \{b, d\}, \{b, e\}, \{d, e\}$ in the closed frequent itemsets
[3]Have been given the support for $\{b\}, \{d\}$ in the closed frequent itemsets