Norwegian University of Science and Technology

TDT4300 — **Solution to Assignment 2**

ASSOCIATION ANALYSIS

tdt4300-undass@idi.ntnu.no

Spring 2023

1 Apriori Algorithm

Given the data in Table 1 (market basket transaction), your task is to describe the purchasing behavior of customers in the form of association rules. First, you need to generate the frequent itemsets and second, you need to generate the association rules. Apply the Apriori algorithm for following tasks and **describe thoroughly the process and the outcome of each step.**

(a) Generate frequent 2-, 3- and 4-itemsets using the $F_{k-1} \times F_{k-1}$ method. Consider the support threshold $minsup = 0.5$ and use the data presented in Table 1.

(b) Using the frequent 4-itemsets from the previous task, generate association rules. Consider the confidence threshold $minconf = 0.8$.

TID	**Items**
110	A, C, F, G, H
111	B, C, D, E, G
112	B, C, E, F, H
113	A, B, C, G
114	C, D, E, H
115	A, B, C, G, H
116	A, B, C, D, G, H
117	B, C, E, G
118	A, B, C, F, G, H
119	A, B, C, D, E, G, H

Table 1: Market basket transactions.

1.1 Solution

(a) Frequent Itemset Generation

First step is to extract all 1-itemsets (items) occurring in the transactions (Table 1) and calculate for each itemset the support metric. The support metric is defined as

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}, \tag{1}$$

which is equivalent to

$$s(Z) = \frac{\sigma(Z)}{N}, \tag{2}$$

where $\sigma(Z)$ is the support count, the number of transactions having the itemset Z as a subset, and N is the number of all transactions.

1-itemset	**Support**
A	0.6
B	0.8
C	1.0
D	0.4
E	0.5
F	0.3
G	0.8
H	0.7

Table 2: 1-itemsets.

The extracted 1-itemsets are the candidate itemsets (Table 2) before the pruning is applied. Performing the pruning, the items D and F (marked red) are discarded, because their support is less than the given support threshold $minsup = 0.5$. We get the frequent itemsets A, B, C, E, G, and H. In the following tables we will use the same notation to mark the pruned itemsets.

The 2-itemset candidates (Table 3) are generated as combination of all frequent 1-itemsets. Pruning is performed to get the frequent 2-itemsets.

2-itemset	**Support**
AB	0.5
AC	0.6
AE	0.1
AG	0.6
AH	0.5
BC	0.8
BE	0.4
BG	0.7
BH	0.5
CE	0.5
CG	0.8
CH	0.7
EG	0.3
EH	0.3
GH	0.5

Table 3: 2-itemsets.

For generating the candidate 3-itemsets we can finally apply the $F_{k-1} \times F_{k-1}$ method and perform the pruning afterwards. The idea of the $F_{k-1} \times F_{k-1}$ method is to merge a pair of frequent k-itemsets only if the first (k-1) items are identical. In this way we get the candidate (k+1)-itemsets. For example, in the Table 3 we have frequent itemsets BC and BG which we can merge to a new candidate itemset BCG, because they have the same prefix B.

ABC	0.5
ABG	0.5
ABH	0.4
ACG	0.6
ACH	0.5
AGH	0.5
BCG	0.7
BCH	0.5
BGH	0.4
CEG	0.3
CEH	0.3
CGH	0.5

Table 4: 3-itemsets.

We generate the 4-itemset again by applying the $F_{k-1} \times F_{k-1}$ method and perform the pruning.

ABCG	0.5
ACGH	0.5

Table 5: 4-itemsets.

And because we only have one frequent 4-itemset, the frequent itemset generation stops here.

(b) Rule Generation

As requested in the assignment, the association rules should be generated only for the frequent 4-itemsets $ABCG$ and $ACGH$. We apply the level-wise approach for generating association rules, the confidence-based pruning and the $F_{k-1} \times F_{k-1}$ method for generating the consequents.

The first step is to generate all candidate rules with one item (1-itemset) in the consequent and calculate for each rule the confidence metric necessary for the pruning. The confidence metric is defined as

$$c(X \to Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}. \tag{3}$$

Rule	Confidence
$ABC \rightarrow G$	1.0
$ABG \rightarrow C$	1.0
$ACG \rightarrow B$	0.83
$BCG \rightarrow A$	0.71
$ACG \rightarrow H$	0.83
$ACH \rightarrow G$	1.0
$AGH \rightarrow C$	1.0
$CGH \rightarrow A$	1.0

Table 6: Association rules with 1-itemset consequent.

One rule is pruned (Table 6) because the given confidence threshold $minconf = 0.8$ is higher than the confidence value of the rule $BCG \rightarrow A$. The pruned rules are marked red in the following tables.

The rules with 2-itemset consequent are generated as a combination of 1-itemset consequences of the high-confidence rules. The confidence is calculated and pruning performed (see Table 7).

Rule	Confidence
$AB \rightarrow CG$	1.0
$AC \rightarrow BG$	0.83
$AG \rightarrow BC$	0.83
$AC \rightarrow GH$	0.83
$AG \rightarrow CH$	0.83
$CG \rightarrow AH$	0.63
$AH \rightarrow CG$	1.0
$CH \rightarrow AG$	0.71
$GH \rightarrow AC$	1.0

Table 7: Association rules with 2-itemset consequent.

Now we can apply the $F_{k-1} \times F_{k-1}$ method to generate the 3-itemset consequents of the rule candidates. Table 8 presents the generated and pruned rules.

Rule	**Confidence**
$A \rightarrow BCG$	0.83
$A \rightarrow CGH$	0.83

Table 8: Association rules with 3-itemset consequent.

In total, we have generated 16 high-confidence rules.

2 FP-Growth Algorithm

2.1 Assignment

Use the Frequent Pattern Growth algorithm to discover the frequent itemsets in the given transaction dataset (Table 1). Consider the support threshold $minsup = 0.5$. Construct an FP-tree and mine the frequent itemsets by creating conditional (sub-)pattern bases. Use the table notation with columns: item, conditional pattern base, conditional FP-tree, frequent patterns generated. The recursive steps of the FP-Growth algorithm must be clearly captured using the aforementioned table notation. Sort items alphabetically in case of ties in the item support. **Describe thoroughly the process and the outcome of each step.**

2.2 Solution

For simplification, we are going to use the minimum support count which is 5 and equivalent to $minsup = 0.5$. First, we calculate the support count for each item (see Table 9).

Item	Support Count
A	6
B	8
C	10
D	4
E	5
F	3
G	8
H	7

Table 9: Item support counts.

Second, we eliminate infrequent items, that is items with support count less than 5 (marked red in Table 9), and sort frequent items based on their support (Table 10). Items with equal support are sorted alphabetically as requested.

TID	Items
110	C, G, H, A
111	C, B, G, E
112	C, B, H, E
113	C, B, G, A
114	C, H, E
115	C, B, G, H, A
116	C, B, G, H, A
117	C, B, G, E
118	C, B, G, H, A
119	C, B, G, H, A, E

Table 10: Items sorted by support and with discarded infrequent items D and F.

We proceed with building an FP-tree data structure (Figure 1). The FP-tree nodes correspond to items and have a counter. A counter is incremented when transactions share items, respectively, have the same prefix. It is important to maintain the item ordering from the Table 10. Also pointers are maintained between same items (dashed arrows).

Figure 1: FP-tree.

The next step is the frequent itemsets mining from the FP-tree. It is a bottom-up recursive approach repeating three steps: extracting prefix path sub-trees (conditional pattern bases), building conditional FP-trees and finding frequent itemsets. **See page 369 of the book Introduction to Data Mining (Tan et al.), and/or the pseudo-algorithm in the book Data Mining (Han et al.) at the page 246.** The outcome of each step is thoroughly depicted in the Table 11.

Suffix	**Conditional Pattern Base**	**Conditional FP-tree**	**Frequent Patterns Generated**
null	—	—	{E: 5}
{E}	{{C, H: 1}, {C, B, H: 1}, {C, B, G: 2}, {C, B, G, H, A: 1}}	⟨C: 5⟩	{C, E: 5}
null	—	—	{A: 6}
{A}	{{C, G, H: 1}, {C, B, G: 1}, {C, B, G, H: 4}}	⟨C: 6, G: 6, H: 5⟩, ⟨C: 6, B: 5, G: 5⟩	{C, A: 6}, {G, A: 6}, {B, A: 5}, {H, A: 5}
{C, A}	—	—	—
{G, A}	{{C: 1}}, {{C, B: 5}}	⟨C: 6, B: 5⟩	{C, G, A: 6}, {B, G, A: 5}, {C, B, G, A: 5}
{B, A}	{{C: 5}}	⟨C: 5⟩	{C, B, A: 5}
{H, A}	{{C, G: 1}}, {{C, B, G: 4}}	⟨C: 5, G: 5⟩	{C, H, A: 5}, {G, H, A: 5}, {C, G, H, A: 5}
null	—	—	{H: 7}
{H}	{{C, G: 1} {{C, B: 1} {C, B, G: 4}, {C: 1}}	⟨C: 7, B: 5, G: 5⟩	{C, H: 7}, {B, H: 5}, {G, H: 5}, {C, B, H: 5}, {C, G, H: 5}, {B, G, H: 5}
null	—	—	{G: 8}
{G}	{{C, B: 7}}, {C: 1}	⟨C: 8, B: 7⟩	{C, G: 8}, {B, G: 7}, {C, B, G: 7}
null	—	—	{B: 8}
{B}	{{C: 8}}	⟨C: 8⟩	{C, B: 8}
null	—	—	{C: 10}
{C}	—	—	—

Table 11: Mining FP-tree.

Column *Frequent Patterns Generated* in Table 11 shows the frequent itemsets found by applying the FP-Growth algorithm.

3 KNIME

For this task you will need to install and use the KNIME[1] data analytics platform. You are given a file *market_basket_transactions.arff* which contains the very same transaction as in Table 1. Your task is to implement two simple workflows for mining association rules, one implementing Apriori algorithm and second implementing FP-Growth algorithm. Use the WEKA nodes both for Apriori and FP-Growth. Use the same parameters as in the previous tasks, e.i. $minsup = 0.5$ and $minconf = 0.8$. **Present pictures of your workflows, and the outputs from both Apriori and FP-Growth nodes. Deliver also the exported KNIME workflows.**

3.1 Solution

Figure 2 show the very simple KNIME workflow for mining association rules with Apriori and FP-Growth algorithm.

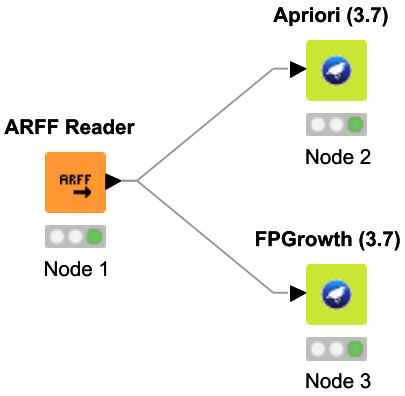

Figure 2: KNIME workflow.

Given the *market_basket_transactions.arff* file (and parameters $minsup = 0.5$ and $minconf = 0.8$), the outputs are as follows.

Apriori node output:

```
Apriori
=======

Minimum support: 0.5 (5 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 10

Generated sets of large itemsets:
```

[1] http://www.knime.org/downloads/overview

```
Size of set of large itemsets L(1): 6

Size of set of large itemsets L(2): 11

Size of set of large itemsets L(3): 8

Size of set of large itemsets L(4): 2

Best rules found:

 1. B=t 8 ==> C=t 8    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 2. G=t 8 ==> C=t 8    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 3. H=t 7 ==> C=t 7    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 4. B=t G=t 7 ==> C=t 7    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 5. A=t 6 ==> C=t 6    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 6. A=t 6 ==> G=t 6    <conf:(1)> lift:(1.25) lev:(0.12) [1] conv:(1.2)
 7. A=t G=t 6 ==> C=t 6    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
 8. A=t C=t 6 ==> G=t 6    <conf:(1)> lift:(1.25) lev:(0.12) [1] conv:(1.2)
 9. A=t 6 ==> C=t G=t 6    <conf:(1)> lift:(1.25) lev:(0.12) [1] conv:(1.2)
10. E=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
11. A=t B=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
12. A=t B=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
13. A=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
14. G=t H=t 5 ==> A=t 5    <conf:(1)> lift:(1.67) lev:(0.2) [2] conv:(2)
15. A=t H=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
16. B=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
17. G=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
18. A=t B=t G=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
19. A=t B=t C=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
20. A=t B=t 5 ==> C=t G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
21. C=t G=t H=t 5 ==> A=t 5    <conf:(1)> lift:(1.67) lev:(0.2) [2] conv:(2)
22. A=t G=t H=t 5 ==> C=t 5    <conf:(1)> lift:(1) lev:(0) [0] conv:(0)
23. A=t C=t H=t 5 ==> G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
24. G=t H=t 5 ==> A=t C=t 5    <conf:(1)> lift:(1.67) lev:(0.2) [2] conv:(2)
25. A=t H=t 5 ==> C=t G=t 5    <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
26. G=t 8 ==> B=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
27. B=t 8 ==> G=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
28. C=t G=t 8 ==> B=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
29. B=t C=t 8 ==> G=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
30. G=t 8 ==> B=t C=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
31. B=t 8 ==> C=t G=t 7    <conf:(0.88)> lift:(1.09) lev:(0.06) [0] conv:(0.8)
32. A=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
33. A=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
34. A=t C=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
35. A=t 6 ==> B=t C=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
36. A=t G=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
37. A=t 6 ==> B=t G=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
38. A=t C=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
39. A=t 6 ==> C=t H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
40. A=t G=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
41. A=t 6 ==> G=t H=t 5    <conf:(0.83)> lift:(1.67) lev:(0.2) [2] conv:(1.5)
42. A=t C=t G=t 6 ==> B=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
43. A=t G=t 6 ==> B=t C=t 5    <conf:(0.83)> lift:(1.04) lev:(0.02) [0] conv:(0.6)
```

```
 44. A=t C=t 6 ==> B=t G=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 45. A=t 6 ==> B=t C=t G=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 46. A=t C=t G=t 6 ==> H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 47. A=t G=t 6 ==> C=t H=t 5    <conf:(0.83)> lift:(1.19) lev:(0.08) [0] conv:(0.9)
 48. A=t C=t 6 ==> G=t H=t 5    <conf:(0.83)> lift:(1.67) lev:(0.2) [2] conv:(1.5)
 49. A=t 6 ==> C=t G=t H=t 5    <conf:(0.83)> lift:(1.67) lev:(0.2) [2] conv:(1.5)
 50. C=t 10 ==> B=t 8    <conf:(0.8)> lift:(1) lev:(0) [0] conv:(0.67)
 51. C=t 10 ==> G=t 8    <conf:(0.8)> lift:(1) lev:(0) [0] conv:(0.67)
```

FP-Growth node output:

```
FPGrowth found 49 rules (displaying top 49)

 1. [G=t]: 8 ==> [C=t]: 8   <conf:(1)> lift:(1) lev:(0) conv:(0)
 2. [B=t]: 8 ==> [C=t]: 8   <conf:(1)> lift:(1) lev:(0) conv:(0)
 3. [H=t]: 7 ==> [C=t]: 7   <conf:(1)> lift:(1) lev:(0) conv:(0)
 4. [A=t]: 6 ==> [C=t]: 6   <conf:(1)> lift:(1) lev:(0) conv:(0)
 5. [E=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
 6. [A=t]: 6 ==> [G=t]: 6   <conf:(1)> lift:(1.25) lev:(0.12) conv:(1.2)
 7. [G=t, B=t]: 7 ==> [C=t]: 7   <conf:(1)> lift:(1) lev:(0) conv:(0)
 8. [G=t, H=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
 9. [A=t]: 6 ==> [C=t, G=t]: 6   <conf:(1)> lift:(1.25) lev:(0.12) conv:(1.2)
10. [C=t, A=t]: 6 ==> [G=t]: 6   <conf:(1)> lift:(1.25) lev:(0.12) conv:(1.2)
11. [G=t, A=t]: 6 ==> [C=t]: 6   <conf:(1)> lift:(1) lev:(0) conv:(0)
12. [B=t, H=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
13. [B=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
14. [H=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
15. [B=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
16. [G=t, H=t]: 5 ==> [A=t]: 5   <conf:(1)> lift:(1.67) lev:(0.2) conv:(2)
17. [H=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
18. [B=t, A=t]: 5 ==> [C=t, G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
19. [C=t, B=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
20. [G=t, B=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
21. [G=t, H=t]: 5 ==> [C=t, A=t]: 5   <conf:(1)> lift:(1.67) lev:(0.2) conv:(2)
22. [C=t, G=t, H=t]: 5 ==> [A=t]: 5   <conf:(1)> lift:(1.67) lev:(0.2) conv:(2)
23. [H=t, A=t]: 5 ==> [C=t, G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
24. [C=t, H=t, A=t]: 5 ==> [G=t]: 5   <conf:(1)> lift:(1.25) lev:(0.1) conv:(1)
25. [G=t, H=t, A=t]: 5 ==> [C=t]: 5   <conf:(1)> lift:(1) lev:(0) conv:(0)
26. [G=t]: 8 ==> [B=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
27. [B=t]: 8 ==> [G=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
28. [G=t]: 8 ==> [C=t, B=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
29. [C=t, G=t]: 8 ==> [B=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
30. [B=t]: 8 ==> [C=t, G=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
31. [C=t, B=t]: 8 ==> [G=t]: 7   <conf:(0.88)> lift:(1.09) lev:(0.06) conv:(0.8)
32. [A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
33. [A=t]: 6 ==> [H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
34. [A=t]: 6 ==> [C=t, B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
35. [C=t, A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
36. [A=t]: 6 ==> [C=t, H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
37. [C=t, A=t]: 6 ==> [H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
38. [A=t]: 6 ==> [G=t, B=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
39. [G=t, A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
```

```
40. [A=t]: 6 ==> [G=t, H=t]: 5   <conf:(0.83)> lift:(1.67) lev:(0.2) conv:(1.5)
41. [G=t, A=t]: 6 ==> [H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
42. [A=t]: 6 ==> [C=t, G=t, B=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
43. [C=t, A=t]: 6 ==> [G=t, B=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
44. [G=t, A=t]: 6 ==> [C=t, B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
45. [C=t, G=t, A=t]: 6 ==> [B=t]: 5   <conf:(0.83)> lift:(1.04) lev:(0.02) conv:(0.6)
46. [A=t]: 6 ==> [C=t, G=t, H=t]: 5   <conf:(0.83)> lift:(1.67) lev:(0.2) conv:(1.5)
47. [C=t, A=t]: 6 ==> [G=t, H=t]: 5   <conf:(0.83)> lift:(1.67) lev:(0.2) conv:(1.5)
48. [G=t, A=t]: 6 ==> [C=t, H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
49. [C=t, G=t, A=t]: 6 ==> [H=t]: 5   <conf:(0.83)> lift:(1.19) lev:(0.08) conv:(0.9)
```

4 Compact Representation of Frequent Itemsets

Given the compact representation of frequent itemsets in Table 12, use the appropriate algorithm to generate all frequent itemsets including the support counts. **Describe thoroughly each step of the algorithm and present the final result.**

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 12: Closed frequent itemsets.

4.1 Solution

The frequent itemsets (including their support counts) were generated following the Algorithm 6.4 in the book Introduction to Data Mining (Tan et al.) at page 357. The calculations bellow depict the states of each iteration of the algorithm. The results are presented in the Table 13.

$F = \{\{a, c, d, e\}, \{a, b, e\}, \{a, c, d\}, \{a, c, e\}, \{a, d, e\}, \{b, d, e\}, \{c, d, e\}, \{a, b\},$
$\{a, c\}, \{a, d\}, \{a, e\}, \{b, d\}, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$

$k_{max} = 4$

$F_{k_{max}} = \{\{a, c, d, e\}\}$

$k = 3$

$F_k = \{\{a, b, e\}, \{a, c, d\}, \{a, c, e\}, \{a, d, e\}, \{b, d, e\}, \{c, d, e\}\}$

$sup(\{a, c, e\}) = max\{sup(\{a, c, d, e\})\} = 5$

$sup(\{a, d, e\}) = max\{sup(\{a, c, d, e\})\} = 5$

$sup(\{c, d, e\}) = max\{sup(\{a, c, d, e\})\} = 5$

$k = 2$

$F_k = \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b,d\}, \{b,e\}, \{c, d\}, \{c, e\}, \{d, e\}\}$

$sup(\{a, b\}) = max\{sup(\{a, b, e\})\} = 7$

$sup(\{a, c\}) = max\{sup(\{a, c, d\}),sup(\{a, c, e\})\} = 6$

$sup(\{a, e\}) = max\{sup(\{a, b, e\}),sup(\{a, c, e\}),sup(\{a, d, e\})\} = 7$

$sup(\{c, d\}) = max\{sup(\{a, c, d\}),sup(\{c, d, e\})\} = 6$

$sup(\{c, e\}) = max\{sup(\{a, c, e\}),sup(\{c, d, e\})\} = 5$

$k = 1$

$F_k = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$

$sup(\{a\}) = max\{sup(\{a, c\}),sup(\{a, d\}),sup(\{a, e\})\} = 11$

$sup(\{c\}) = max\{sup(\{a, c\}),sup(\{c, d\}),sup(\{c, e\})\} = 6$

$sup(\{e\}) = max\{sup(\{a, e\}),sup(\{b, e\}),sup(\{c, e\}),sup(\{d, e\}))\} = 8$

Figure 3: Illustration of closed frequent itemset

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

Table 13: All frequent itemsets (including closed frequent itemsets).