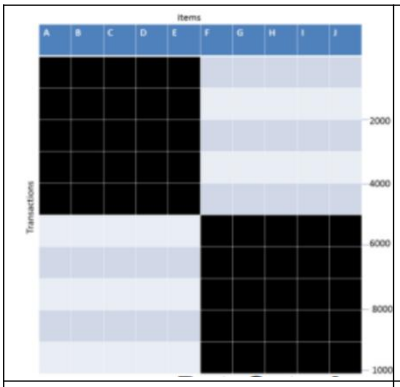
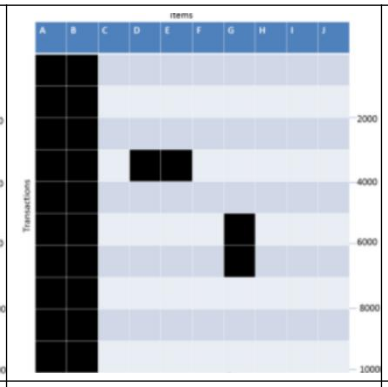
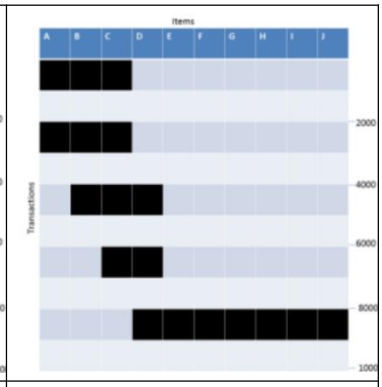
Data Mining Homework 3.0

Problem 1:

		
Data Set:A	Data Set:B	Data Set:C

a. **What is the number of frequent item sets for each data set? Which data set will produce the most number of frequent item sets?**

The minimum number of occurrences of frequent itemsets is:

$$10000 \times 20\% = 2000$$

For data set A:

Given that the total number of transactions is 10,000, items A-E appear in the first 5,000 transactions, and items F-J appear in the last 5,000 transactions. The single - item frequent item sets are A, B, C, D, E, F, G, H, I, J , totaling 10.

We then consider Multi - item frequent item sets:

For the 5 elements of (A - E), the number of two - element combinations is $C_5^2 = 10$; the number of three - element combinations is $C_5^3 = 10$, the number of four - element combinations is $C_5^4 = 5$; the number of five - element combinations is $C_5^5 = 1$, So the number of non-empty subsets composed of A-E is: 10+10+5+1=26

Similarly, the number of non - empty subsets composed of F-J is also 26.

Therefore, the total number of frequent item sets in Data set A is:

$$10 + 26 + 26 = 62$$

For data set B:

Single-item frequent item sets:{A},{B},{G}which respectively appears in 10000, 10000,2000 transactions,totaling 3.

Multi-item frequent item sets:{A,B} (10000 times),{B,G}(2000 times),{A,G}(2000 times) ,{A,B,G}(2000 times),totaling 4

Therefore, the total number of frequent item sets in Data set B is:

$$3+4=7$$

For data set C:

Single-item frequent item sets:{A}(2000times),{B}(3000 times),{C}(4000 times), {D}(3000 times),totaling 4;

Multi-item frequent item sets:{A,B}(2000 times),{B,C}(3000 times), {C,D}(2000 times),{A,C}(2000 times),{A,B,C}(2000 times),totaling 5

Therefore, the total number of frequent item sets in Data set C is:

$$4+5=9$$

So the number of frequent item sets for data set A,B,C is 62,7,9 and data set A will produce the most number of frequent item sets.

b. **Which dataset will produce the longest frequent itemset?**

For dataset A:

The longest frequent itemset is {A,B,C,D,E}or{F,G,H,I,J},containing 5 elements.

For dataset B:

The longest frequent itemset is {A,B,G},containing 3 elements.

For dataset C:

The longest frequent itemset is {A,B,C},containing 3 elements.

Therefore ,data set A will produce the longest frequent item set,

c.Which dataset will produce frequent itemsets with highest maximum support?

For dataset A:

The frequent item set with highest maximum support is {A,B,C,D,E}or {F,G,H,I,J} or their respective subsets ,which all support 50%.

For dataset B:

The frequent itemset with highest maximum support is{A,B}or{A}{B},the support is 100%.

For dataset C:

The frequent item set with highest maximum support is {C} ,and the support is $\frac{4000}{10000}=0.4$ (40%)

Therefore, data set B will produce frequent itemsets with highest maximum support(100%)

d.Which dataset will produce frequent itemsets containing items with widely varying support levels (i.e., itemsets containing items with mixed support, ranging from 20% to more than 70%)?

For dataset A:

All the frequent itemsets support 50%

For dataset B:

The lowest support of frequent itemsets is {A,B,G} or its subsets,support 20%,the highest is{A,B}or{A}or{B},support 100%

For dataset C:

The lowest support of frequent itemsets is {A}or{C,D}or{A,B}or{A,C},Supports 20%, the highest is{C},the support is 40%

Therefore , dataset B will produce frequent itemsets containing items with widely varying support levels(20% - 100%)

e.What is the number of maximal frequent itemsets for each dataset? Which dataset will produce the most number of maximal frequent itemsets?

For data set A:

The maximal frequent item sets are {A,B,C,D,E} {D,E,F,G,H}

For data set B:

The maximal frequent item set is{A,B,G}

For data set C:

The maximal frequent item sets are{A,B,C},{C,D}

Therefore ,the number of each data set is 2,1,2,and data set A and C will produce the most number of maximal frequent item sets.

f.What is the number of closed frequent itemsets for each dataset? Which dataset will produce the most number of closed frequent itemsets?

For data set A:

The closed frequent itemsets are {A,B,C,D,E}(50%),{D,E,F,G,H}(50%)

For data set B:

The closed frequent itemsets are {A,B}(100%),{A,B,G}(20%)

For data set C:

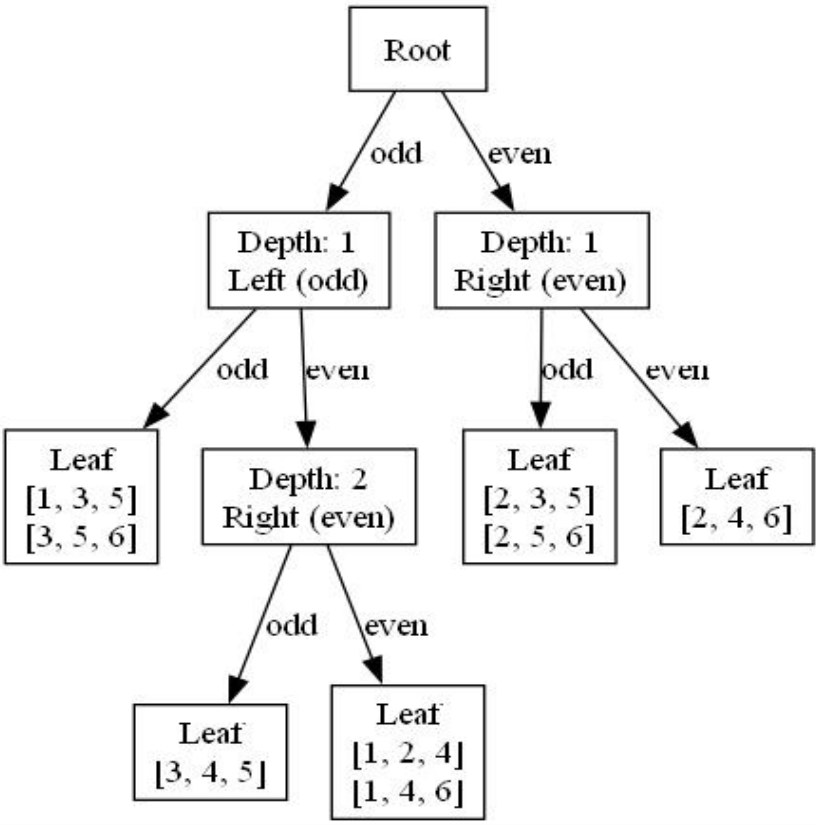
The closed frequent itemsets are{C}(40%),{D}(30%),{B,C}(30%),{C,D}(20%), {A,B,C}(20%)

Therefore ,the number of closed frequent itemsets for each dataset is 2,2,5 and data set C will produce the most number of closed frequent itemsets.

Problem 2:

a. Construct a hash tree for the given candidate 3 - itemsets. The hash function used in the tree maps all odd - numbered items to the left child of a node and all even - numbered items to the right child. To insert a candidate k - itemset into the tree, hash each successive item in the candidate and then follow the corresponding branch of the tree according to the hash value. Once a leaf node is reached, insert the candidate based on one of the following conditions.(.....)

Given the conditions of the problem ,we can get the hash tree:



The construction of the hash tree:

The construction of the hash tree commences from the root node at depth 0. For the given candidate 3 - itemsets, an insertion operation is executed based on a specific hash function rule where odd - numbered items are hashed to the left child node and even - numbered items to the right child node. For instance, for the itemset {1, 2, 4}, hashing 1 (an odd number) leads to the left branch to reach a depth - 1 "Left (odd)" node; then hashing 2 (an even number) takes us to the right branch to a depth - 2 node, and hashing 4 (also even) keeps us on the right branch. As the depth of the reached leaf node equals the itemset's element count (3), {1, 2, 4} is inserted regardless of the node's existing itemset count. Similarly, for {1, 3, 5}, hashing 1, 3, and 5 all leads to the left branch, and it is inserted after reaching the depth - 3 leaf node. When inserting a new itemset, if the reached leaf node has a depth less than 3 and the number of stored itemsets is below the maximum capacity of 2, the new itemset can be directly inserted. But if the leaf node's depth is less than 3 and the maximum capacity of 2 is

reached, the leaf node is converted into an internal node, new leaf nodes are created as its children, and the original and new itemsets are re - distributed according to their hash values. By following these rules, insertion operations are carried out on all candidate 3 - itemsets {1, 4, 6}, {2, 3, 5}, {2, 5, 6}, {3, 4, 5}, {3, 5, 6}, and {2, 4, 6} in sequence to construct a complete hash tree.

b. How many leaf nodes are there in the candidate hash tree? How many internal nodes are there?

According to the hash tree ,There are 5 leaf nodes and 4 internal nodes in hash tree!

c. Consider a transaction that contains the following items: {1, 2, 3, 5, 6}. Using the hash tree constructed in part (a), which leaf nodes will be checked against the transaction? What are the candidate 3-itemsets contained in the transaction?

For the transaction T = {1, 2, 3, 5, 6}, in order to find the candidate 3 - itemsets within it, we need to traverse the hash tree. First, we list all its 3 - item subsets, such as {1, 2, 3}, {1, 2, 5}, etc. Taking {1, 2, 3} as an example, we determine the path according to the parity of the items. Starting from the root node, we pass through "Depth: 1 Left (odd)" and "Depth: 2 Right (even)", and finally reach the leaf node storing {3, 4, 5}. After performing such operations on all subsets, we find that multiple leaf nodes will be accessed. However, the itemsets in the leaf nodes storing {3, 4, 5}, {1, 2, 4}, and {1, 4, 6} are not completely contained in the transaction T. Therefore, the actual leaf nodes that need to be checked are the leaf node storing {1, 3, 5} and {3, 5, 6}, and the leaf node storing {2, 3, 5} and {2, 5, 6}. The candidate 3 - itemsets {1, 3, 5}, {2, 3, 5}, {2, 5, 6}, and {3, 5, 6} in these nodes are the candidate 3 - itemsets contained in the transaction T.

In conclusion ,we need check 4 leaf nodes: {1, 3, 5}, {2, 3, 5}, {2, 5, 6}, {3, 5, 6}

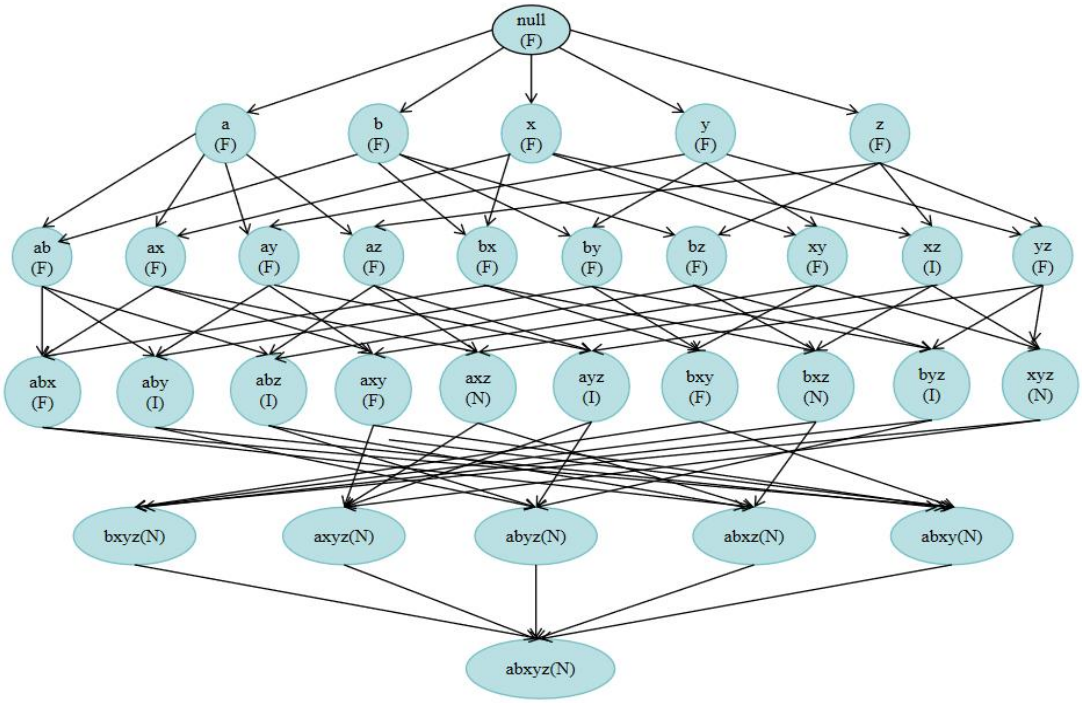
Problem 3:

The Apriori algorithm employs a generate - and - count approach to obtain frequent itemsets. In the candidate generation step, candidate itemsets of size k + 1 are formed by combining a pair of frequent itemsets of size k. Then, in the candidate pruning step, if any subset of a candidate itemset is discovered to be infrequent (in this case, with a minimum support threshold of 30%, meaning an itemset that appears in fewer than 3 transactions is considered infrequent), the candidate itemset is discarded. This algorithm is applied to the data set presented in Table 2.0.

Table: 2.0

Transaction ID	**Items** Bought
T1	a, b, x, y
T2	b, x, y
T3	a, y, z
T4	a, b, x, z
T5	x, y
T6	b, z
T7	a, x, y, z
T8	a, b
T9	b, y, z
T10	a, b, x, y

a. **Draw an item set lattice representing the data set given in Table 2.0 . Label each node in the lattice with the following letter(s):**



①Single-item sets:

Frequent item sets(F):{a}(60%) ,{b}(70%),{x}(60%),{y}(70%),{z}(50%)

There are no infrequent items

②Two-item sets:

Frequent item sets(F):{a,b}(40%),{a,x}(40%),{a,y}(40%),{a,z}(30%),{b,x}(40%)
{b,y}(40%),{b,z}(30%),{x,y}(50%),{y,z}(30%)

Candidate Infrequent item sets(I):{x,z}(20%)

③Three-item sets:

We can join frequent 2-itemsets to form candidates, ensuring all subsets are frequent.and we can calculate each item set:

Frequent item sets(F):{a,b,x}(30%),{a,x,y}(30%),{b,x,y}(30%)
Candidate Infrequent item sets(I): {a,b,y}(20%),{a,y,z}(20%),{b,y,z}(10%)
{a,b,z}(10%)
Pruned (subset {x, z} is infrequent)→(N):{a, x, z}, {b, x, z}, {x, y, z}

④Four-item sets:

We can also join frequent 3-itemsets to form candidates:
{a,b,x,y}(N) : its subset{a,b,y}is infrequent
{a, b, x, z}, {a, b, y, z}, {a, x, y, z}, {b, x, y, z}(N):they are not generated

⑤Five-item sets:

{a,b,x,y,z}(N):Not generated(subset{a,b,x,y} is pruned)

⑥Root(F)： The root node corresponding to the empty set is a subset of all frequent sets.

b.What is the percentage of frequent item sets (with respect to all item sets in the lattice)?

Total item sets:

$$1+C_5^1+C_5^2+C_5^3+C_5^4+C_5^5=32$$

Frequent item sets:

We have previously got the result of frequent item sets:
Root:1 1-itemsets:5 2-itemsets:9 3-itemsets:3
Total: 1+5+9+3=18

Therefore, the percentage is: $\frac{18}{32}\times 100\%=56.25\%$

c.What is the pruning ratio of the Apriori algorithm on this data set? The pruning ratio is defined as the percentage of itemsets not considered to be a candidate because (1) they are not generated during candidate generation or (2) they are pruned during the candidate pruning step.

There are 9 item sets marked as 'N',and the number of all nodes of item sets is 32

So we can get Pruning ratio: $\frac{9}{32}\times 100\%=28.125\%$

d. What is the false alarm rate (i.e., percentage of candidate item sets that are found to be infrequent after performing support counting)?

The number of candidate item sets is : 5+10+7+1=23
There are 5 item sets marked as 'I';

We can calculate the false alarm rate is: $\frac{5}{32}\times 100\%=15.6\%$