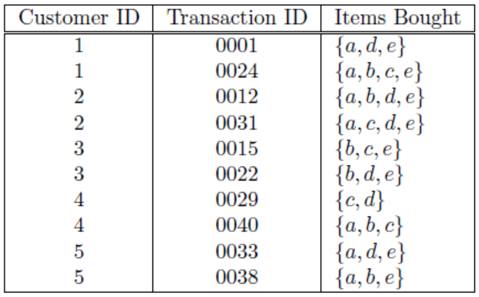
CS-422 Assignment 4

Association Rule Mining, MapReduce

Explain every answer briefly in YOUR OWN WORDS, just a short answer without explanation will be zero points.

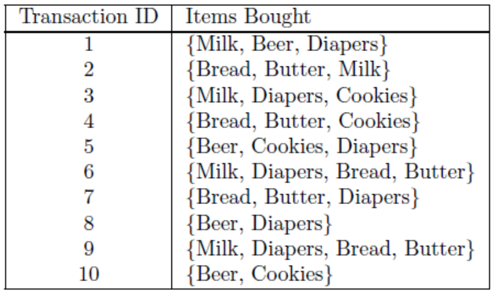
Total 100 points

1. 25 points
   a) Consider the market basket transactions shown in table to the right. Compute the support for itemsets {*e*}, {*b,d*}, and {*b,d,e*} by treating each transaction ID as a market basket.
   b) Compute the confidence for the association rules {*b,d*} –→ {*e*} and {*e*} –→ {*b,d*}. Explain each step in your calculation and show all calculations.
   c) Is confidence a symmetric measure?



Customer ID	Transaction ID	Items Bought
1	0001	$\{a, d, e\}$
1	0024	$\{a, b, c, e\}$
2	0012	$\{a, b, d, e\}$
2	0031	$\{a, c, d, e\}$
3	0015	$\{b, c, e\}$
3	0022	$\{b, d, e\}$
4	0029	$\{c, d\}$
4	0040	$\{a, b, c\}$
5	0033	$\{a, d, e\}$
5	0038	$\{a, b, e\}$

2. 15 points
   a) Consider the market basket transactions shown in table to the right. What 2-itemset has the largest support? Show how you calculate the value of the support for that itemset.
   b) For the itemset from (a), it is a 2-itemset and contains 2 items, *a* and *b.* Calculate the confidence for the rules {*a*} –→ {*b*} and {*b*} –→ {*a*}. What can you say about the confidence for this rule? How do you explain this results from the data and also from your conclusion in exercise 1?



Transaction ID	Items Bought
1	{Milk, Beer, Diapers}
2	{Bread, Butter, Milk}
3	{Milk, Diapers, Cookies}
4	{Bread, Butter, Cookies}
5	{Beer, Cookies, Diapers}
6	{Milk, Diapers, Bread, Butter}
7	{Bread, Butter, Diapers}
8	{Beer, Diapers}
9	{Milk, Diapers, Bread, Butter}
10	{Beer, Cookies}

3. 35 points

   For the questions below: For each step of your tree build process show what condition you used. The answers should be concise. Only show the details to prove that you build the tree yourself. For part (b) show the details of placing itemsets down each level of the tree.

   Consider the following set of candidate 3-itemsets:

{1,2,3},{1,2,6},{1,3,4},{2,3,4},{2,4,5},{3,4,6},{4,5,6}

(a) Construct a hash tree for the above candidate 3-itemsets. Assume the tree uses a hash function where all odd-numbered items are hashed to the left child of a node, while the even-numbered items are hashed to the right child. A candidate *k*-itemset is inserted into the tree by hashing on each successive item in the candidate and then following the appropriate branch of the tree according to the hash value. Once a leaf node is reached, the candidate is inserted based on one of the following conditions:

Condition 1: If the depth of the leaf node is equal to *k* (the root is assumed to be at depth 0), then the candidate is inserted regardless of the number of itemsets already stored at the node.

Condition 2: If the depth of the leaf node is less than k, then the candidate can be inserted as long as the number of itemsets stored at the node is less than *maxsize*. Assume *maxsize* = 2 for this question.

Condition 3: If the depth of the leaf node is less than k and the number of itemsets stored at the node is equal to *maxsize*, then the leaf node is converted into an internal node. New leaf nodes are created as children of the old leaf node. Candidate itemsets previously stored in the old leaf node are distributed to the children based on their hash values. The new candidate is also hashed to its appropriate leaf node.

(b) Consider a transaction that contains the following items: {1,2,3,5,6}. What are the candidate 3-itemsets contained in the transaction?

(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 matched against the transaction?

4. 25 points

How will you implement the Grep tool using MapReduce? The tool has to extract matching strings from text files and counts how many time they occurred. How many MapReduce job are required? Define the Input and Output key-value pairs for each MapReduce job. Write the pseudocode for the implementation of the Mapper and Reducer for each MapReduce job.