CS 412 HW2

Kaiqi Zheng

Due: Oct 16 2018

Problem 1

a. The total number of cuboids is 2^{10}

b. There are 3 closed cells,including$(a1, *, a3, *, ..., a9, *)$ and two base cells.

c. Each of the base cell generates 2^{10}-1 aggregated cells. Among them there are 2^3 cells overlapped:
$(a1, *, a3, ...a9, *), (a1, *, a3, ...*, *), (a1, *, *, ..., a9, *), (*, *, a3, ..., a9, *)$
$(a1, *, *, ...*, *), (*, *, a3, ...*, *), (*, *, *, ...a9, *), (*, *, *, ...*, *)$
So the result is $2 * (2^{10} - 1) - 2^3 = 2038$

d. There is only 1 nonempty aggregate closed cell:
$(a1, *, a3, *, ..., a9, *)$

e. There are 8 aggregate cells with count no less than 2.
$(a1, *, a3, ...a9, *), (a1, *, a3, ...*, *), (a1, *, *, ..., a9, *), (*, *, a3, ..., a9, *)$
$(a1, *, *, ...*, *), (*, *, a3, ...*, *), (*, *, *, ...a9, *), (*, *, *, ...*, *)$

Problem 2

a. Standard deviation is algebraic. because:
$STD(X) = \sqrt{\frac{sum(x^2)}{count()} - (\frac{sum(x)}{count()})^2}$
$sum(x^2), sum(x), count()$ are all distributive measures.

b. It is algebraic because it equals to $\frac{min()+max()}{2}$

c. It is algebraic.
We can first take the maximum 50 elements from each block, and compute the overall maximum 50 elements from the dataset. Then, we compute the sum of these elements.

d. It is not algebraic.
There is no constant bound of memory cache needed to compute the sum since the elements we have to store increase linearly with the magnitude of n. in this case, it is not an algebraic measure.

e. It is algebraic.
We can compute sum() and count() first, both are distributive. Then we compare the result.
If $sum() > count()$, then return 1.
If $sum() == count()$, then return 0, 1.
If $sum() < count()$, then return 0.

Problem 3

$min_sup = 0.6, min_conf = 0.7$

a. 1-frequent itemsets are:
(A): 3 (B): 4 (C): 4 (D): 4
2-frequent itemsets are:
(A, D): 3 (B, C): 4 (B, D): 3 (C, D): 3
3-frequent itemset is:
(B,C,D): 3
the largest k is 3.

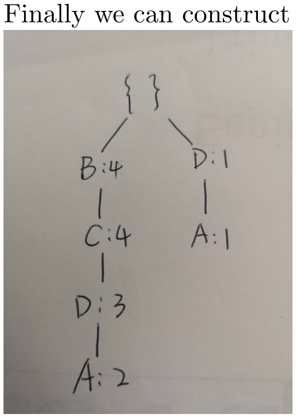
b. S=(A,B):2 satisfies the condition. The nonempty subsets (A) : 3, (B) : 4 are frequent, but S itself is not frequent.

c. Closed patterns are: (A,D):3, (B,C):4, (D):4, (B,C,D):3

d. The max pattern are: (A,D):3, (B,C,D):3

e. we have:
$buys(x, B) \bigwedge buys(x, C) \Rightarrow buys(x, D)[.6, .75]$
$buys(x, B) \bigwedge buys(x, D) \Rightarrow buys(x, C)[.6, 1.]$
$buys(x, C) \bigwedge buys(x, D) \Rightarrow buys(x, B)[.6, 1.]$

f. First we have: B:4 C:4 D:4 A:3
Then we have ordered frequent itemlist:
001 B,C,D,A
002 D,A
003 B,C,D
004 B,C,D,A
005 B,C
Finally we can construct the FP tree:



g. A's conditional database is:
A BCD:2 D:1

Problem 4

a. The frequent itemset satisfying $sum(S.price) >= 45$ are:
(B,C):4, (B,D):3, (C,D):3, (B,C,D):3

b. $sum(S.price) >= 45$ is monotonic.
if any itemset satisfies the condition, any of its superset will satisfy the condition since prices are nonnegative.

$sum(S.price) <= 45$ is anti-monotonic.
if any itemsets has sum of prices greater than 45 then any of its superset will also violate the condition.

In Apriori approach, we still start finding all 1-frequent itemsets first, and then find 2-frequent itemsets by constructing all 2 itemsets, and so on. if the current itemset violates the constraint $sum(S.price) <= 45$, then we can terminate this candidate-constructing branch, that is, prune it and start a new branch from a new itemset.

c. Both $avg(S.price) >= 30$ and $avg(S.price) <= 30$ are convertible.
$avg(S.price) >= 30$ cannot be converted into anti-monotonic cases. It can only be converted into monotonic cases.
$avg(S.price) <= 30$ can be converted into anti-monotonic cases by rearranging the table items into ascending order. A:10 B:20 D:30 G:30 C:40 H:50 E:90 F:90