CS 412: Homework #2

Due on Tuesday Oct. 18th

Kurt Rudolph
rudolph9

Problem 1

We model the users in a social network as a data cube. Suppose each user has 10 dimensions of information, such as age, gender, city and income. Assume a base cuboid of 10 dimensions contains three base cells: (1) $(b1, b2, a3, a4, a5, \dots, a9, a10)$: $count = 10$, (2)$(b1, a2, b3, a4, a5, \dots, a9, a10)$: $count = 20$, and (3)$(a1, b2, b3, a4, a5, \dots, a9, a10)$:count=50, where $a_i! = b_i, a_i! = a_j$, etc. The count measure of the cube means the number of users who satisfy such information.

(1) How many nonempty cuboids will a full data cube contain?

Solution

$$2^{10} - 1 = 1023$$

(2) How many nonempty aggregate (i.e., non-base) cells will a full cube contain?

Solution

$$totalCells - (overlappingCells * overlappingTimes)$$
$$= totalCells - (overlapTwiceCells * 2 + overlapOnceCells * 1 + baseCells)$$
$$= 3 * 2^{10} - (2^7 * 2 + 2^7 * 3 * 1 + 3) = 2429$$

(3) How many nonempty aggregate cells will an iceberg cube contain if the condition of the iceberg cube is "$count \geq 70$"?

Solution

Since one base cell has $count = 20$ and one base cell has $count = 70$ we simply need to compute the number of instances where subsets of these two base cells overlap, using our calculations from the previous problem we find:

$$27 * 2 + 27 * 3 * 1 = 135$$

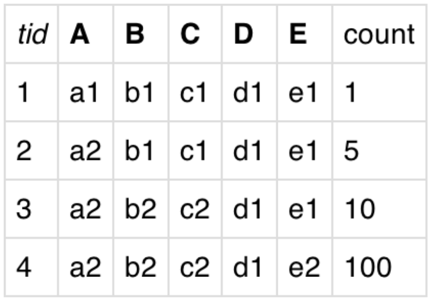
(4) How many closed cells are in the full cube?

Solution

There are 4, the three base cell and $(*, *, *, a4, a5, \dots, a9, a10)$.

Problem 2

Given the following base cuboid with count as the measure.



tid	**A**	**B**	**C**	**D**	**E**	count
1	a1	b1	c1	d1	e1	1
2	a2	b1	c1	d1	e1	5
3	a2	b2	c2	d1	e1	10
4	a2	b2	c2	d1	e2	100

(1) Briefly outline the major steps to compute Shell-Fragment cube (refer to VLDB04 paper High-Dimensional OLAP: A Minimal Cubing Approach), suppose we divide the 5 dimensions into 2 shell fragments: AB and CDE.

Solution

The major steps to compute Shell-Fragment cube start with vertically partitioning the high dimensional dataset in a set of disjoint low dimensional datasets (i.e. our two fragments AB and CDE). For each of the fragments we compute its local data cube. Next we register the set of tuple-ids that contribute to the non-empty cells in the fragment data cube. These are used to bridge the gap between various garments and re-construct the corresponding cuboids upon request. Essentially the data cubes of the original high diminutional space are dynamically assembled together view these fragments.

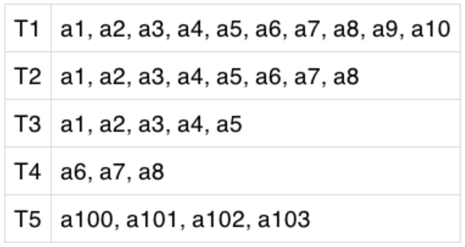
(2) Briefly describe how to compute subcube query (a2,b2,*,*,? : count())

Solution

First seek a set of cuboid cells in the original data space, where one of the relevant dimensions in the query is inquired. Next seek the next relevant component of the query and ignore all ambiguous entries.

Problem 3

Given a database of five transactions ($min\_support = 2$):



T1	a1, a2, a3, a4, a5, a6, a7, a8, a9, a10
T2	a1, a2, a3, a4, a5, a6, a7, a8
T3	a1, a2, a3, a4, a5
T4	a6, a7, a8
T5	a100, a101, a102, a103

(1) How many frequent patterns?

Solution

2: a1, a2, a3, a4, a5 and a6, a7, a8

(2) What is the set of frequent closed patterns (list both pattern and support)?

Solution

a1, a2, a3, a4, a5 support = 2/3% and a6, a7, a8 support = 1%

(3) What is the set of frequent max-patterns (list both pattern and support)?

Solution

a1, a2, a3, a4, a5 support = 2/3% and a6, a7, a8 support = 1%

(4) Show an example association rule that matches $(a1, a2, a3, a4, itemX) \rightarrow (itemY)[min\_support = 2, min\_confidence = 70\%]$

Solution

(5) For association rule $a1 \rightarrow a6$, compute the following measures: confidence, lift, kulc.

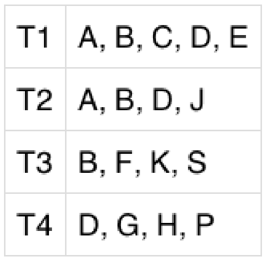
Solution

(6) Among the above three measures, which ones are null-invariant?

Solution

Problem 4

Given a database of four transactions (*min_support* = 2):



T1	A, B, C, D, E
T2	A, B, D, J
T3	B, F, K, S
T4	D, G, H, P

(1) Show the major steps to find the frequent patterns using Apriori.

Solution

(1) The join step - first all transaction are scanned in order to count the number of occurrences of each item. Next the support is computed equal to 50%. To discover the set of frequent 2-items L_2, the algorithm now uses the join.

(2) The prune step - After completing the join step transactions are scanned and accumulated. The set C_2 is now computed having minimum support.

(2) Show the major steps to find the frequent patterns using FP-Growth (no need to draw the trees).

Solution

(1) The FP-tree is constructed - First scan the transaction database. Collect frequency of items. Next create the root of the FP-tree and label it as ”null”.

(2) Mine the FP-tree by calling FP-Growth(FP-tree, null)

(3) Compare the three algorithms: Apriori, FP-growth and ECLAT, by concisely discussing the major differences.

Solution

Both algorithms start by initially scanning all items in the database, however the algorithms largely differ when the analytical steps take place. The major difference is the use joins in the Apriori algorithm whereas in the FP-Growth a tree structure is built.