CS412: Introduction to Data Mining	**Fall 2015**
Assignment 2	
Li Miao	*October 8, 2015*

Question 1 (16 points)

Assume that a base cuboid of 6 dimensions contains only 3 base cells:

$$(a_1, a_2, a_3, a_4, a_5, a_6),\ (b_1, b_2, a_3, a_4, a_5, a_6),\ \text{and}\ (c_1, c_2, a_3, a_4, a_5, a_6)$$

where $a_i \neq b_i$, $b_i \neq c_i$, and $a_i \neq c_i$, $\forall i = 1, 2$. There is no dimension with concept hierarchy. The measure of the cube is *count*. The *count* of each base cell is 1.

Requirements

- Include final results and explain how you calculate the cells in the Answer Document. Keep it brief and clear.

a. $2^6 = 64$.

b. $2^4 * (3 * 2^2 - 2) - 3 = 157$. For last four dimensions, we have only two choice (include $*$). Then for first two dimensions, we need to discuss 3 conditions. $a_1, *, a_2, *$,$b_1, *, b_2, *$,$c_1, *, c_2, *$. And we need to delete the 2 overlap of $*$. Finally, we need to delete the 3 base cells.

c. $2^4 = 16$. Only when first two dimensions are both stars.

d. 4. First two dimensions are stars, left 4 dimensions are not stars.

Question 2 (24 points)

We give you an artificially generated dataset `Data-Q2.txt` in the dataset file. It contains 100 business records. Each row is a business record, and the data fields in each row are separated by tabs. Each record contains the fields `Business_ID`, `City`, `State`, `Category`, `Price`, `Quarter`, `Year`. The four quarters in a year are denoted $Q1$, $Q2$, $Q3$ and $Q4$. We now want to construct a cube over the 4 dimensions `Location`, `Category`, `Price`, and `Time`, with *count* as the measure. The `Location` dimension has a `City-State` concept hierarchy and, similarly, the `Time` dimension has a `Quarter-Year` hierarchy.

a. (4', L1) T $= (2 + 1) * (1 + 1) * (1 + 1) * (2 + 1) = 36$

b. (4', L2) 56.

c. (4', L2) 34.

d. (4', L2) 33.

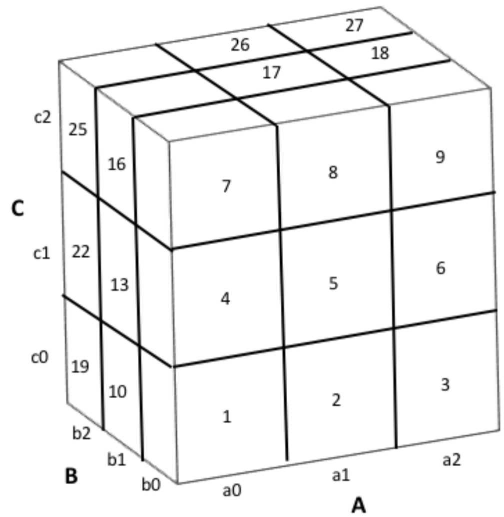


Figure 1: A 3-D array with dimensions A, B and C. This array is divided into 27 smaller chunks.

e. (4', L2) 10.

f. (4', L2) 4.

Question 3 (15 points)

We have a data array with 3 dimensions A, B and C. The 3-D array is divided into small chunks. Each dimension is divided into 3 equally sized partitions. See Figure 1. For example, dimension A is divided into a_0, a_1, and a_2, and dimension B is divided into b_0, b_1, and b_2. There are totally 27 chunks and each chunk is denoted by $a_i b_j c_k$. The sizes of the dimensions A, B, and C are 900, 300, and 600. Since we divide each dimension into 3 parts with equal size, the sizes of the chunks on dimensions A, B, and C are 300, 100, and 200 respectively. Now we want to use **Multiway Array Aggregation Computation** to materialize the 2-D cuboids AB, AC and BC.

a. (7', L2) For 2-D planes, we need to keep 1 BC, 3 AB and 9 AC chunks in memory per period.

   So the size is $BC + 3AB + 9AC = 100 * 200 + 3 * 300 * 100 + 9 * 300 * 200 = 650000$.

b. (8', L3) Yes. There exist other orders to scan the chunks so that the memory cost is less than that in sub-question (a). We hope to keep as few AC as possible because the size of AC chunk is larger than others, AB and BC. So we hope to scan AC only 1 time per run time. And we can scan the whole BC plane because the size of its chunk is the smallest in three 2-D planes.

   Therefore, we can do this following order: 1-10-19

4-13-22

7-16-25

2-11-28

5-14-23

8-17-26

3-12-21

6-15-24

9-18-27

Then we need to keep 1 AC, 3 AB, 9 BC chunks in memory per period. The size is $300 * 200 + 3 * 300 * 100 + 9 * 200 * 100 = 330000$.

Question 4 (15 points)

We have a 3-D data array with 3 dimensions A, B, C. The data contained in the array is as follows:

$$
\begin{array}{lll}
(a_0, b_0, c_0) : 1 & (a_0, b_0, c_1) : 1 & (a_0, b_0, c_2) : 1 \\
(a_0, b_1, c_0) : 1 & (a_0, b_1, c_1) : 1 & (a_0, b_1, c_2) : 1 \\
(a_0, b_2, c_0) : 1 & (a_0, b_2, c_1) : 1 & (a_0, b_2, c_2) : 1 \\
(a_0, b_3, c_0) : 1 & (a_0, b_3, c_1) : 1 & (a_0, b_3, c_2) : 1
\end{array}
$$

You will use the **Bottom-Up Computation (BUC)** algorithm to materialize the cube. Please answer the following questions.

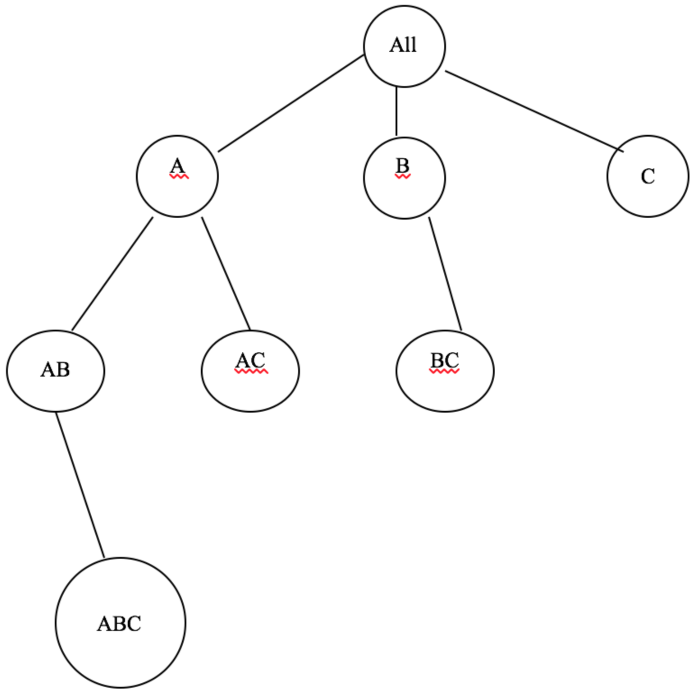


Figure 2: trace tree of expansion with the exploration order $A \to B \to C$

a.

b. (5', L3) 16 cells will be considered.

All($*, *, *$):12 - expansion

A(a_0,$*$,$*$):12 - expansion

AB($a_0, b_0, *$):3
AB($a_0, b_1, *$):3
AB($a_0, b_2, *$):3
AB($a_0, b_3, *$):3

AC($a_0, *, c_0$):4
AC($a_0, *, c_1$):4
AC($a_0, *, c_2$):4

B($*, b_0, *$):3
B($*, b_1, *$):3

B($*, b_2, *$):3
B($*, b_3, *$):3

C($*, *, c_0$):4
C($*, *, c_1$):4
C($*, *, c_2$):4

c. (5', L3) If we set $min\_support = 4$ with the exploration order $B \rightarrow A \rightarrow C$, 12 cells would be considered.
All($*, *, *$):12 - expansion

B($*, b_0, *$):3
B($*, b_1, *$):3
B($*, b_2, *$):3
B($*, b_3, *$):3

A(a_0,$*$,$*$):12 - expansion

AC($a_0, *, c_0$):4
AC($a_0, *, c_1$):4
AC($a_0, *, c_2$):4

C($*, *, c_0$):4
C($*, *, c_1$):4
C($*, *, c_2$):4

Question 5 (10 points)

a. **F.** Static Status: operational update of data does not occur in the data warehouse environment.

b. **F.** Cell B is a child of cell A.

c. **F.** In OLAP operations, we can see more general data information by rolling up.

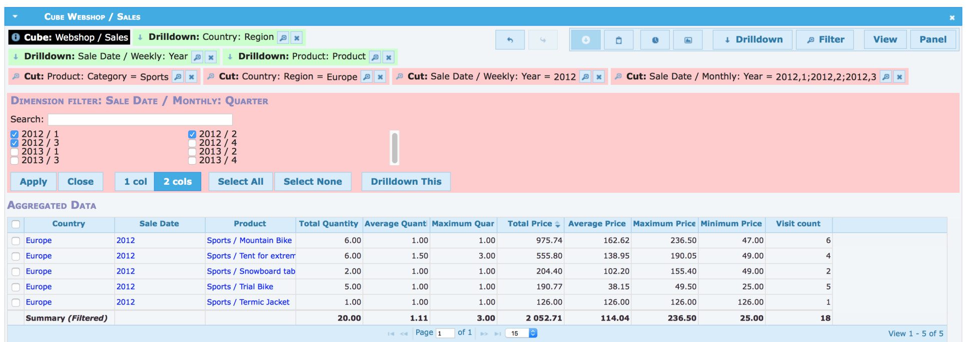
d. **T.** The Bottom-Up Computation (BUC) algorithm can be used to compute either the full cube or a partial cube. If min_sup = 1, then the full Cube. If partition does not satisfy min_sup, it decedents can be pruned.

e. **F.** The Multiway Array Aggregation Computation is most effective when the product of the cardinalities of dimensions is low.

Mini Machine Problem (20 points)

CubesViewer is a visual, web-based tool application for exploring and analyzing OLAP databases served by the Cubes OLAP Framework[1]. The CubesViewer Explorer demo can be found at http://crow.cs.illinois.edu:8080/cubesviewer/. You can login with user `cs412`, password `cs412f2015`

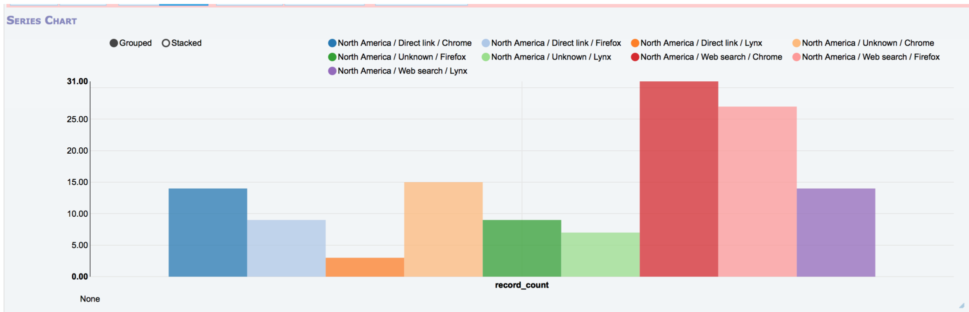
a. (5', L2) Mountain Bike has the highest revenue. Termic Jacket has the least revenue. Steps: Drill down:Country:Region → Drill down:Sale Date/Monthly:Year → Drill-down:Product:Category → Slice:Country:Region = Europe → Slice:Sale Date/Monthly:Year = 2012 → Slice:Product:Category = Sports → Drilldown:Product:Product → Dice:Sale Date/Monthly:Year = 2012,1;2012,2;2012,3



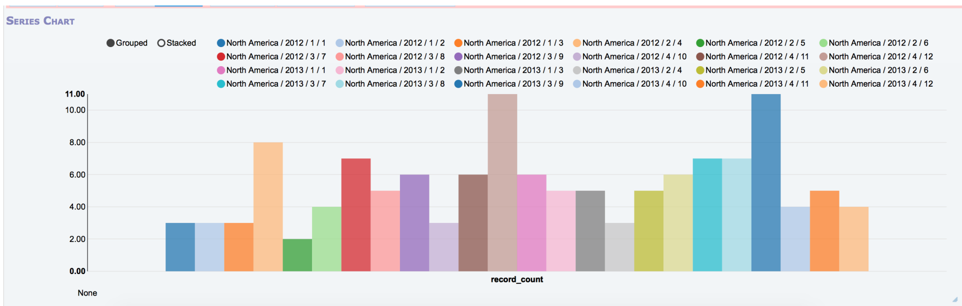
Country	Sale Date	Product	Total Quantity	Average Quant	Maximum Quar	Total Price	Average Price	Maximum Price	Minimum Price	Visit count
Europe	2012	Sports / Mountain Bike	6.00	1.00	1.00	975.74	162.62	236.50	47.00	6
Europe	2012	Sports / Tent for extrem	6.00	1.50	3.00	555.80	138.95	190.05	49.00	4
Europe	2012	Sports / Snowboard tab	2.00	1.00	1.00	204.40	102.20	155.40	49.00	2
Europe	2012	Sports / Trial Bike	5.00	1.00	1.00	190.77	38.15	49.50	25.00	5
Europe	2012	Sports / Termic Jacket	1.00	1.00	1.00	126.00	126.00	126.00	126.00	1
Summary *(Filtered)*			**20.00**	**1.11**	**3.00**	**2 052.71**	**114.04**	**236.50**	**25.00**	**18**

b. (4', L2) Use web search in chrome is the popular way in North America.
Steps: Drilldown: Country;Region → Drilldown: source→ Drilldown:Browser → Slice: Country:Region=North_America

[1] https://github.com/jjmontesl/cubesviewer

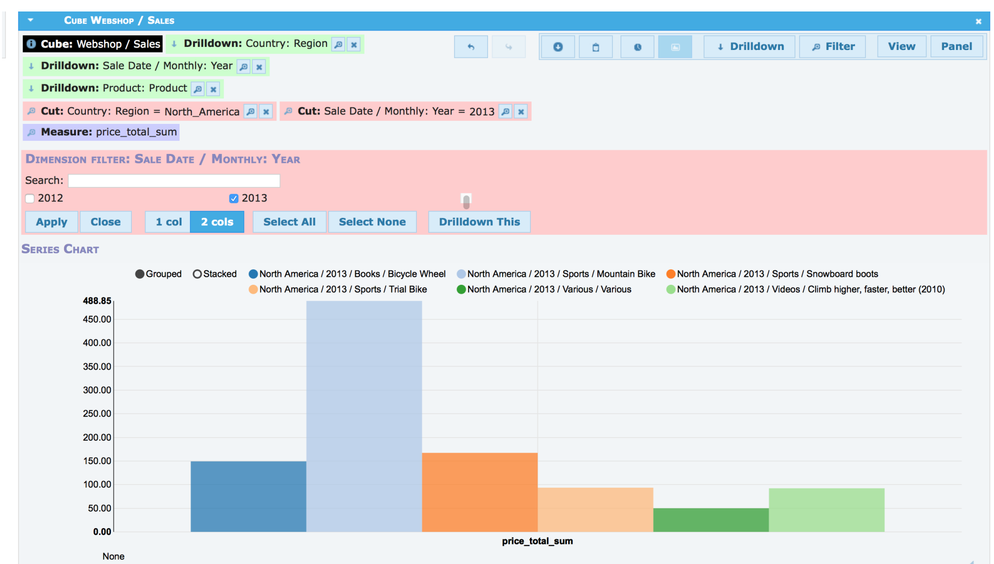


c. (3', L2)



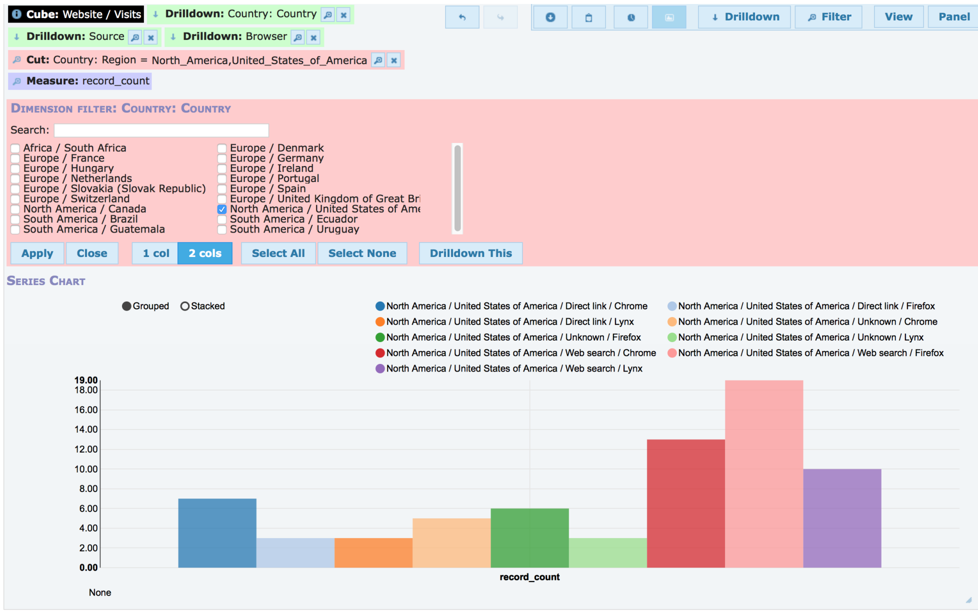
d. (8', L3)

- Webshop/Sales Steps: Drilldown:Country:Region→ Drilldown:Sale Date/Monthly:Year →Drilldown:Product:Product →Slice:Country:Region=North_America → Slice: Year = 2013



Therefore, in North America 2013, the Mountain Bike in sports has the highest avenue. So we should restock more Mountain Bike.

- Website/Visits Steps: DrilldownCountry:Country→ Drilldown:Source→ Drilldown:Browser→ Slice: Country:Region = North_America,United_States_of_America



Therefore, In United States of America, the most popular way for customers to visit online store is by web search on Firefox. So we should invest more ads on Firefox to attract more customers.