CS412: Introduction to Data Mining **Fall 2015**

Assignment 2: Chapters 4, 5

Due: 10/08/2015 11:59pm

General Instruction

- Errata: After the assignment is released, any further corrections of errors or clarifications will be posted at the Errata page at Piazza. Please watch it.
- Feel free to talk to other members of the class while doing the homework. We are more concerned that you learn how to solve the problem than that you solve it entirely on your own. You should, however, write the solution yourself.
- Please use Piazza first if you have questions about the homework. Also feel free to send us e-mails and come to office hours.
- For each question, you should show the necessary calculation steps and reasoning—not only final results. Keep the solution brief and clear.
- For a good balance of cognitive activities, we label each question with an activity type:
  - **L1 (Knowledge)** Definitions, propositions, basic concepts.
  - **L2 (Practice)** Repeating and practicing algorithms/procedures.
  - **L3 (Application)** Critical thinking to apply, analyze, and assess.

Assignment Submission

- Please submit your work before the due time. **We do NOT accept late submission!**
- Please submit your answers electronically via Compass (`http://compass2g.illinois.edu`). Contact TAs if you have technical difficulties in submitting the assignment.
- Please **type** your answers in an **Answer Document**, and submit it in PDF. **Handwritten answers or hand-drawn pictures are not acceptable**. Your answers to all questions (including mini-MP) should be included in one Answer Document.
- Please **DO NOT** zip the Answer Document (PDF) so that the graders can read it directly on Compass. Compress other files into a single zip file. Overall, you need to submit one Answer Document (PDF file), named as `hw2_netid.pdf`, and one `zip` file, named as `hw2_netid.zip`.
- If scripts are used, you should submit the source code, and use file names to identify the questions or sub-questions being answered. E.g., `question1_netid.py` is the python code for Question 1; and `question1a_netid.py` that for sub-question 1(a). You can submit separate files for sub-questions or a single file for the entire question.

Dataset

- The data set file, `Assignment2-Data.zip`, can be found in the course website.

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.

Purpose

- Have a better understanding of cubes, multidimensional view of data, and cuboid structures.

Requirements

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

a. (4', L1) How many cuboids are there in the full data cube?

b. (4', L2) How many distinct aggregated (i.e., non-base) cells will the complete cube contain?

c. (4', L2) How many distinct aggregated cells will an iceberg cube contain, if the condition of the iceberg cube is $count \geq 3$?

d. (4', L2) How many non-star dimensions does the closed cell with $count = 3$ have?

Solution:

a. As there are 6 dimensions, and there is no dimension with concept hierarchy, so there are $(1+1)^6 = 64$ cuboids in the data cube.

b. 157. **Hint:** First, we consider those cuboids with at least one of the first two dimensions not aggregated(i.e. not $*$), for example cuboid $(d_1, *, ..., *)$. For each base cell, the number of such cuboids is $(2^2 - 1) \times 2^4 = 48$. Since we have 3 such base cells, thus the total number of cuboids in this case is $3 \times 48 = 144$. Second, we consider those cuboids with the first two dimensions aggregated, for example cuboid $(*, *, d_3, d_4, ..., d_6)$, the number of such cuboids is $2^4 = 16$. Since all 3 base cells are same if we only consider the last 4 dimensions, the total number of cuboids in this case is 16. Finally, the total number of non-based cells is $144 + 16 - 3 = 157$.

c. 16. **Hint:** Only those cells with the first two dimensions aggregated(i.e. *), for example the cell $(*, *, a_3, a_4, ..)$, have count 3. And the number of such cells is $2^4 = 16$.

d. 4. **Hint:** There is only one closed cell with count 3, i.e. $(*, *, a_3, a_4, a_5, a_6)$. So the number of non-star dimensions is $6 - 2 = 4$.

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.

Purpose

- Have a better understanding of measures and cuboid structures.

Requirements

- For sub-question (a), you should show the final result with a brief explanation or intermediate steps in the Answer Document.
- For sub-questions (b), (c), (d), (e), and (f), you should write scripts to manipulate data and show your answers in the Answer Document. There is no restrictions on the language you use and you are allowed to use any built-in functions. You are required to submit your source code too.

a. (4', L1) How many cuboids are there in the cube?

b. (4', L2) How many distinct cells are there in the cuboid (`Location`[`City`], `Category`, `Price`, `Time`[`Year`])?

c. (4', L2) If we roll up by climbing up in the `Location` hierarchy from `City` to `State`, how many distinct cells are there in the cuboid (`Location`[`State`], `Category`, `Price`, `Time`[`Year`])?

d. (4', L2) How many distinct cells are there in the cuboid (`*`, `Category`, `Price`, `Time`[`Quarter`])?

e. (4', L2) What is the count for the cell (`Location`[`State`] = *Illinois*, `Category` = *Food*, `*`, `Time`[`Quarter`] = $Q1$)?

f. (4', L2) What is the count for the cell (`Location`[`City`] = *Chicago*, `*`, `Price` = *cheap*, `Time`[`Year`]= 2013)?

Solution:

a. 36. **Hint:** Suppose the i_{th} dimension has L_i levels, then the number of cuboids is $\prod_i(Li + 1)$. So here the number is $(2 + 1) \times (1 + 1) \times (1 + 1) \times (2 + 1) = 36$.

b. 56. **Hint:** As long as you understand the concept of cuboid and cell, you can find the answer with minimum effort for question [b] - [f].

c. 34.

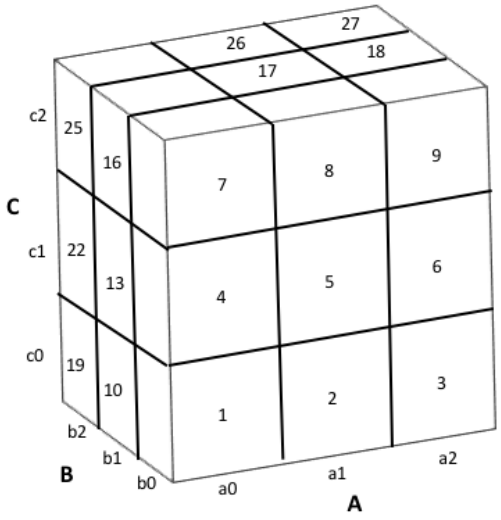


Figure 1: A 3-D array with dimensions A, B and C. This array is divided into 27 smaller chunks.

d. 33.

e. 10.

f. 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.

Purpose

- Have a better understanding of Multiway Array Aggregation Computation.

Requirements

- Show your results in the Answer Document. You should also provide some important intermediate steps in calculation. Only providing a result will not get credits.

a. (7', L2) If we scan the chunks in the order $1, 2, 3...., 27$ when materializing the 2-D cuboids AB, AC and BC, to avoid reading 3-D chunks into memory repeatedly, what is the minimum memory for holding all the related 2-D planes?

b. (8', L3) Do you think there exist other orders to scan the chunks so that the memory cost is less than that in sub-question (a)? If yes, show that order using chunk numbers (e.g. $1, 2, 3..., 27$) and the minimum memory required. Otherwise, explain why.

Solution:

a. If we scan the cube base on the order of $1, 2, 3...27$, we have: 900×600 (for the whole AC plane) + 900×100 (for one row of the AB plane) $+100\times200$ (for one BC plane chunk) $= 650,000$ memory units. (Do not consider the 3-D chunk)
Or we have : $300\times100\times200$ (for one 3-D chunk)$+900\times600$ (for the whole AC plane) + 900×100 (for one row of the AB plane) $+100\times200$ (for one BC plane chunk) $= 6650,000$ memory units. (Consider the 3-D chunk)

b. If we scan the cube base on the order of $1, 10, 19, 4, 13, 22, 7, 16, 25, 2, 11, 20...27$, we have: 300×200 (for one AC chunk) $+ 300 \times 300$ (for one column of the AB plane) + 300×600 (for the BC plane) $= 330,000$ memory units. (Do not consider the 3-D chunk)
Or we have : $300 \times 100 \times 200$ (for one 3-D chunk) $+ 300 \times 200$ (for one AC chunk) + 300×300 (for one column of the AB plane) $+300\times600$ (for the BC plane) $= 6330,000$ memory units. (Consider the 3-D chunk)

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:

$(a_0, b_0, c_0) : 1$	$(a_0, b_0, c_1) : 1$	$(a_0, b_0, c_2) : 1$
$(a_0, b_1, c_0) : 1$	$(a_0, b_1, c_1) : 1$	$(a_0, b_1, c_2) : 1$
$(a_0, b_2, c_0) : 1$	$(a_0, b_2, c_1) : 1$	$(a_0, b_2, c_2) : 1$
$(a_0, b_3, c_0) : 1$	$(a_0, b_3, c_1) : 1$	$(a_0, b_3, c_2) : 1$

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

Purpose

- Have a better understanding of the BUC algorithm.

Requirements

- For sub-question (a), you are allowed to use any software to draw the tree; paste your plot to Answer Document.
- For sub-questions (b) and (c), write your answers in the Answer Document.

a. (5', L2) Draw the trace tree of expansion with the exploration order $A \rightarrow B \rightarrow C$.

b. (5', L3) If we set $min\_support = 4$ with the exploration order $A \rightarrow B \rightarrow C$, how many cells will be considered/computed? For these cells, please list each of them with its count, and report whether it is expansible in the **BUC** process. (*Hint: For you to better understand the question and know how to answer the question, please refer to* `SampleQuestionBUC.pdf` *in Chapter 5 at our course website*).

c. (5', L3) If we set $min\_support = 4$ with the exploration order $B \rightarrow A \rightarrow C$, how many cells would be considered/computed? For these cells, please also list each of them with its count and report whether it is expansible in the **BUC** process.

Solution:

a. See Figure 2.

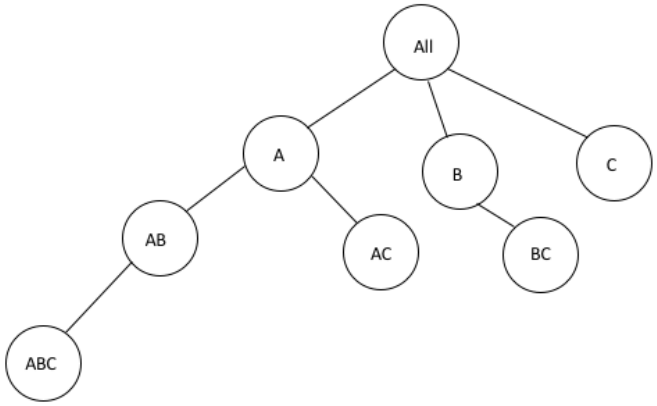


Figure 2: Answer for the Trace Tree.

b. If we follow the order of $A \rightarrow B \rightarrow C$, the cells with their information (count, expansibility) that need to be computed are listed as follows:

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 $(*, *, c_2) : 4$

Thus, there are totally 16 cells which would have to be computed.

c. If we follow the order of $B \to A \to C$, the cells with their information (count, expansibility) that need to be computed are listed as follows:

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 $(*, *, c_2) : 4$

Thus, there are totally 12 cells which would have to be computed.

Question 5 (10 points)

For each question below, indicate if the statement is **true** or **false**.

Purpose

- Have a better understanding of some basic concepts.

Requirements

- For each sub-question, select **true (T)** or **false (F)** and provide a brief explanation for your selection in the Answer Document. You will NOT get credit without an explanation.

a. (2', L1) **T/F.** Operational update is a very important issue for data warehousing.

b. (2', L1) **T/F.** Suppose we pick two cells A and B from a data cube; A is $(a_0, b_0, *, d_0)$ and B is (a_0, b_0, c_0, d_0). Then, cell A is a child of cell B.

c. (2', L1) **T/F.** In OLAP operations, we can see more detailed data information by rolling up.

d. (2', L3) **T/F.** The Bottom-Up Computation (BUC) algorithm can be used to compute either the full cube or a partial cube.

e. (2', L3) **T/F.** The Multiway Array Aggregation Computation is most effective when the product of the cardinalities of dimensions is very high.

Solution:

a. **F**. there is no operational updates in data warehouse.

b. **F**. Cell B is a child of cell A since cell A aggregate 1 dimension base on cell B.

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

d. **T** The Bottom-Up Computation (BUC) algorithm can be used to compute either the full cube or the partial cube. If $min\_support = 1$, it computes the full cube. If $min\_support > 1$, it computes the partial cube.

f. **F** The Multiway Array Aggregation Computation is **not** effective when the product of the cardinalities of dimensions is very high since the memory cost would very high.

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`

Purpose

- Have a better understanding of Data Cubes and OLAP operations.
- Get some hands-on experience with OLAP.

Requirements

- List the OLAP operations necessary to reach a particular cube, and include the screenshots of the final results. For each operation, you need to specify the operation type (`roll-up, drill-down, slice, dice`) and the related dimension (`product, geo, browser`, etc.).
- Play around with CubesViewer to find interesting insights, such as "the sale of sports goods during quarter x in 2012 is much better than that of other quarters of the same year".

a. (5', L2) For the dataset `Webshop/Sales` in CubesViewer, which product in category `Sports` has the highest revenue in Europe during the first three quarters of the year 2012? And which has the least? List the OLAP operations necessary to reach the cube that can answer the questions above. Show the screenshot of the chart generated for the cube by CubesViewer (you must choose the appropriate measure in the `View` menu in order to generate the chart).

b. (4', L2) For dataset `Website/Visits`, what is the most popular way, specified by (`source, browser`), used by customers from North America to visit the online store? The popularity here is measured by the visit count. List the OLAP operations necessary to reach the cube that can answer the questions above. Show the screenshot of the chart generated for the resulting cube by CubesViewer.

c. (3', L2) For `Website/Visits`, show the screenshot of the chart that describes the changes of the visit counts from North America over time. Draw the chart with one of the granularities: week, month, quarter or year.

d. (8', L3) For each of the datasets `Webshop/Sales` and `Website/Visits`, come up with an interesting cube that might help the shop owner make decisions. This is an open question. You will receive full marks by listing the OLAP operations to reach the cubes and what kinds of decisions can be made from those cubes.

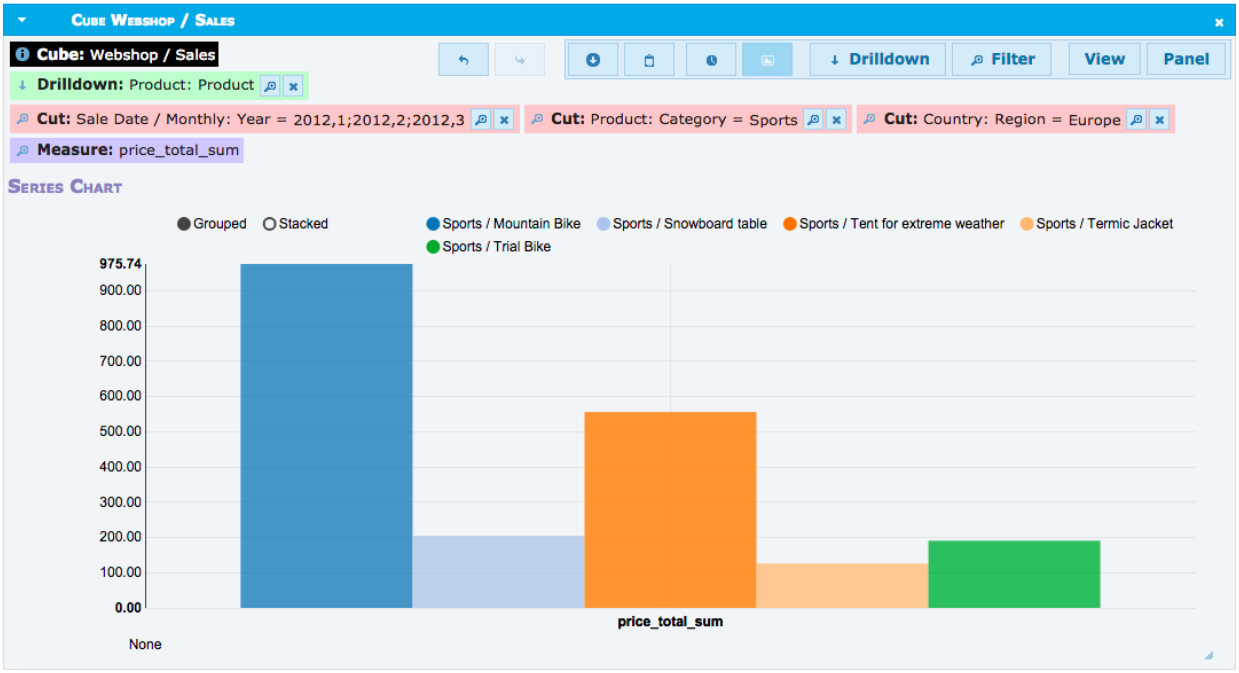
Solution:

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

a. OLAP operations:

   – Drill-down on `Product`
   – Dice on `Quarter`=2012-1,2,3 and `Region`=Europe and `Category`=Sports

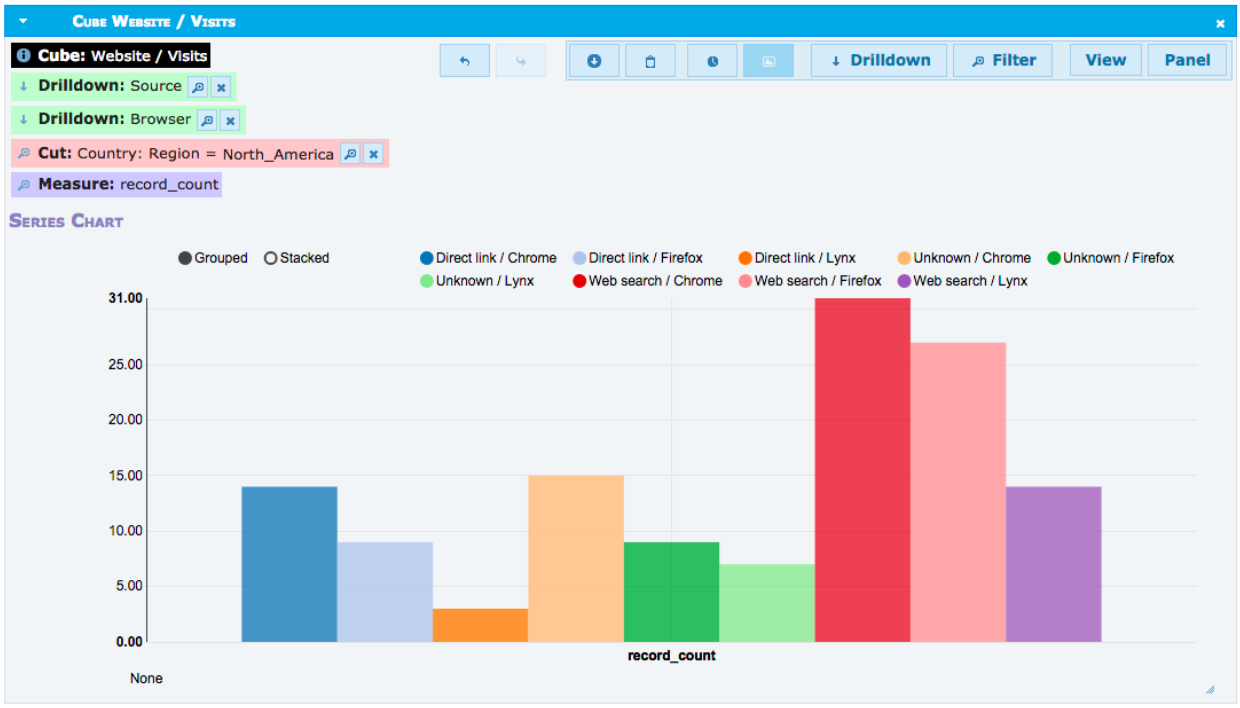
   The highest is Mountain Bike, the lowest is Termic Jacket.



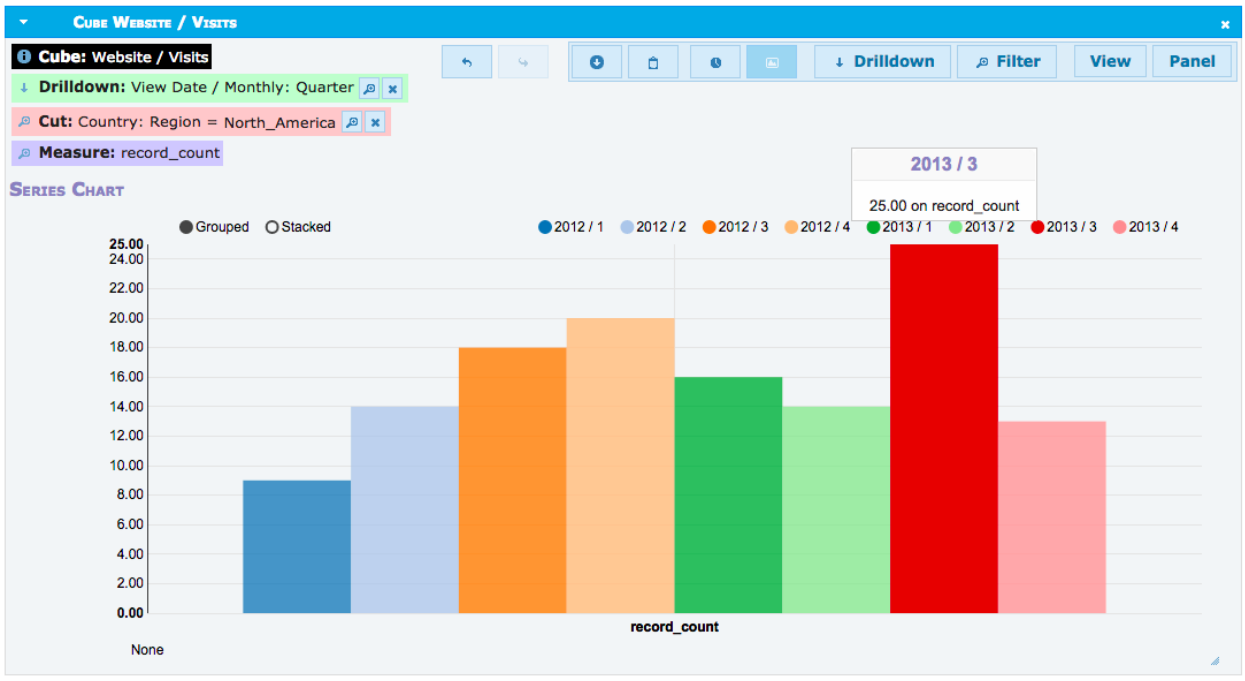
b. OLAP operations:

   – Drill-down on `Source` and `Browser`
   – Slice on `Region`=North_America

   The highest is Web Search / Chrome, the lowest is Direct Link / Lynx.



c. You can choose any granularity. If choosing quarters, the figure is as below:



d. This is an open question. You will receive full marks by listing the OLAP operations to reach the cubes and what kinds of decisions can be made from those cubes. For each dataset, -1 if missing discussion about the decisions; -1 if missing the list of OLAP operations; -4 if missing one of the two datasets in the subquestion; -8 if missing the entire subquestion.