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

Assignment 2

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 demonstrate that you solved it entirely on your own. You should, however, write down the solution yourself.
- Try to keep the solution brief and clear.
- 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 will **NOT** get full credit if you only show the final result. Necessary calculation steps and reasoning are required.

Assignment Submission

- Please submit your work before the due time. **We do NOT accept late homework!**
- We will be using Compass for collecting the homework assignments. Please submit your answers via Compass (`http://compass2g.illinois.edu`). Please do NOT hand in a hard copy of your write-up. Contact the TAs if you are having technical difficulties in submitting the assignment.
- The homework **MUST** be submitted in **pdf** format. **Scanned handwritten and hand-drawn pictures** inside your documents **are not acceptable**. Answers to the written questions and mini-MP should be included in one `.pdf` file.
- Please **DO NOT** zip the PDF file so that graders can access your PDF directly on Compass. You can compress other files into a single zip file. In summary, you need to submit one PDF file, named as `hw2_netid.pdf`, and one `.zip` file, named as `hw2_netid.zip`.
- If scripts are used to solve problems, you are required to submit the source code, and use the file names to identify the corresponding questions or sub-questions. For instance, `question1_netid.py` refers to the python source code for Question 1; and `question1a_netid.py` refers to the python source code 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, `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. Keep it brief and clear.

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

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

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

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

Question 2 (24 points)

We give you an artificially generated dataset `Data_Q2.txt` in `data.zip`. It contains 100 business records. Each row is a business record and data fields in each row are separated by tabs. For each business, it has (`Business_ID`, `City`, `State`, `Category`, `Price`, `Quarter-of-Year`, `Year`). The four quarters in a year are represented as $Q1$, $Q2$, $Q3$ and $Q4$. We now want to construct a cube over four dimensions (`Location`, `Category`, `Price`, `Time`) with *count* as the measure. For the `Location` dimension and the `Time` dimension, there is a concept hierarchy, i.e. `City-State` and `Quarter-Year`. You are required to answer following questions.

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 PDF file you will submit.
- For sub-questions (b), (c), (d), (e), (f), you should write scripts to manipulate data and show your answers in the PDF file you will submit. 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.

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

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

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

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

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

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

Question 3 (15 points)

We have a data array containing 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 represented by a sub-cube $a_i b_j c_k$. The cardinality (size) of the dimensions A, B, and C is 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 cubes. The base cuboid ABC is computed as a 3-D array. We want to materialize the 2-D cuboids AB, AC and BC. Please answer the following questions.

Purpose

- Have a better understanding of Multiway Array Aggregation Computation.

Requirements

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

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

b. (8') Do you think there exist other orders for chunk scanning so that the memory cost is less than that in sub-question (a)? If yes, show that order for chunk scanning using chunk numbers (e.g. $1, 2, 3..., 27$) and the minimum memory requirement with your calculating process. Otherwise, provide your reason.

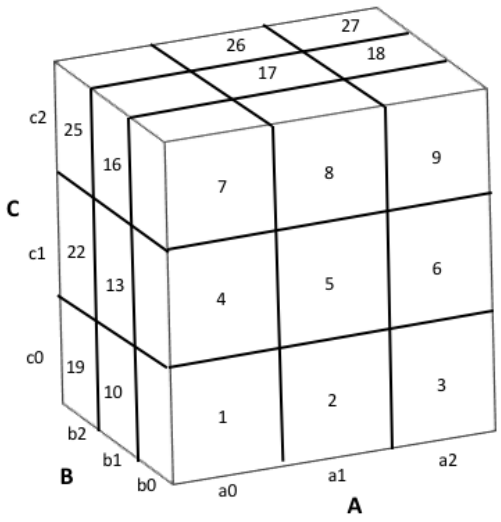

Figure 1: A 3-D array with dimensions A, B and C. This array is divided into 27 smaller chunks.

Question 4 (15 points)

We have a 3-D data array containing three 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 are required to use the **Bottom-Up Computation (BUC)** method 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 painting software to draw the tree and paste your plot to the PDF file you will submit.
- For sub-questions (b) and (c), you should write your answers on the PDF file you will submit.

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

b. (5') If we set $min\_support = 4$ with the exploration order of A $\rightarrow$ B $\rightarrow$ C, how many cells would be considered/computed? You should report the number of cells which would be considered/computed in the PDF file you will submit. For these cells, you should also list each cell with its count and report whether the cell is expansible in the **BUC** process. (*Hint: To make you better understand the question and know how to answer the question, please refer to the SampleQuestionBUC.pdf file in Chapter 5 on the course website.*)

c. (5') If we set $min\_support = 4$ with the exploration order of B $\rightarrow$ A $\rightarrow$ C, how many cells would be considered/computed? You should report the number of cells which would be considered/computed in the PDF file you will submit. For these cells, you should also list each cell with its count and report whether the cell is expansible in the **BUC** process.

Question 5 (10 points)

This is a set of **true** or **false** questions. Please answer the following questions.

Purpose

- Have a better understanding of some basic concepts in Chapter 4 and Chapter 5.

Requirements

- For each sub-question, select **true (T)** or **false (F)** and provide a brief explanation for your selection. You will not get credit without explanation.Write your answer in the PDF file you will submit.

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

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

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

e. (2') **T/F.** The Multiway Array Aggregation Computation is most effective when the product of the cardinalities of dimensions is 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 OLAP operations necessary to reach a particular cube, and include the screenshots of 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 other quarters of the same year".

a. (5') Regarding dataset "`Webshop/Sales`" on CubesViewer, what is the product under category `Sports` with the most revenue in Europe during the first three quarters of year 2012? And what is the least? List OLAP operations necessary to reach the cube that can answer the questions above. Show the screenshot of the chart that is generated from the resulting cube by CubesViewer (you must choose the appropriate measure in the View menu in order to generate the chart).

b. (4') Regarding dataset "`Website/Visits`" on CubesViewer, what is the popular way, specified by a particular source and a particular browser, customers from North America used to visit the online store? List OLAP operations necessary to reach the cube that can answer the questions above. Show the screenshot of the chart that is generated from the resulting cube by CubesViewer (you must choose the appropriate measure in the View menu in order to generate the chart).

c. (3') Regarding dataset "`Website/Visits`" on CubesViewer, show the screenshot of the chart that describes the changes of the visit counts from North America along time.

d. (8') For each dataset ("`Webshop/Sales`" and "`Website/Visits`"), come up with an interesting cube that might help the shop owner make some decisions. This is an open question. You will have the full mark if you can list the OLAP operations to reach the cubes, and what kinds of decisions we can make after looking at the cubes.

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