CS412: Introduction to Data Mining **Summer 2016**

Assignment 2

Due: 06/28/2016 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 in 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 give out a final result. Necessary calculation steps and reasoning are required.
- 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 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 part 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); replace `netid` with your netid. You can submit separate files for sub-questions or a single file for the entire question.

Question 1 (10 points)

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

Purpose

- Have a better general understanding of basic concepts covered in class.

a. (2, L1) **(T/F)** The snowflake schema model saves storage space compared to the star schema model.

b. (2, L1) **(T/F)** OLAP systems usually adopts an entity-relationship data model.

c. (2, L1) **(T/F)** Standard deviation can be derived by applying the function to n aggregate values in a data cube.

d. (2, L1) **(T/F)** One benefit of the Multiway Array Aggregation cube is the equal efficiency regardless of scan order.

e. (2, L1) **(T/F)** Unlike the Multiway Array Aggregation method, BUC cannot take advantage of pruning.

Question 2 (25 points)

Assume a base cuboid of 5 dimensions contains three base cells.

$$(b_1, a_2, a_3, a_4, a_5),\ (a_1, b_2, a_3, a_4, a_5),\ (a_1, a_2, b_3, a_4, a_5)$$

where $a_i \neq b_i,\ \forall i = 1, 2, 3, 4, 5$. There is no dimension with a concept hierarchy and the count of each base cell is 1.

Purpose

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

Requirements

- Provide all the intermediate steps and explain how you calculated the final values. Please keep it brief and clear.

a. (5, L2) How many cuboids are there in the full data cube?

b. (7, L2) How many **nonempty aggregate** cells are there in the full cube?

c. (7, L2) How many **nonempty aggregate closed** cells are there in the full cube?

d. (6, L2) How many **nonempty aggregate** cells will an iceberg cube will contain if the condition of the iceberg cube is *count* ≥ 3?

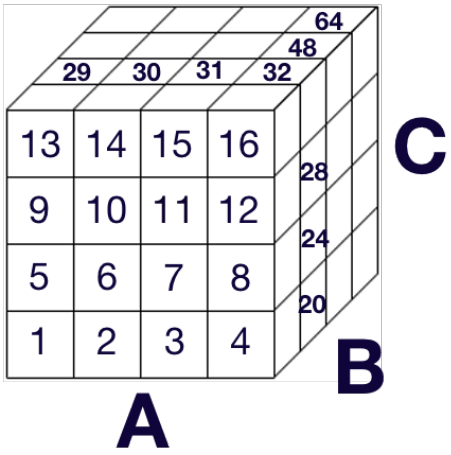

Figure 1: A 3-D array with dimension A, B, and C. This array is divided into 64 smaller chunks.

Question 3 (15 points)

We have a data array with 3 dimensions A, B and C. The size of the dimensions A, B, and C are 100, 200, and 200, respectively. The 3-D array is divided into small chunks. Each dimension is divided into 4 equally sized partitions. Since we divide each dimension into 4 parts with equal size, the sizes of the chunks on dimensions A, B, and C are 25, 50, and 50, respectively. Now we want to use **Multiway Array Aggregation Computation** to material the 2-D cuboids AB, AC, and BC.

Purpose

- Have a better understanding of Multiway Array Aggregation Computation

Requirements

- Provide all the intermediate steps and explain how you calculated the final values. Please keep it brief and clear.
- See Figure 1 for the chunk numbers.

a. (7, L2) If we can the chunks in the order 1, 2, 3, ..., 64 when materializing the 2-D cuboids AB, AC, and BC, to avoid reading 3-D chunks into memory repeatedly, what is the minimum memory needed for holding all the related 2-D planes?

b. (8, L3) Do you think there exists other orders to scan the chunks so that the memory cost is less or equal to the order presented in sub-question (a)? If yes, show the order using chunk numbers (e.g. 1, 2, 3,..., 64) and compute the minimum memory required. Otherwise, explain why.

Question 4 (20 points)

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

$(a_0, b_0, c_0) : 1$	$(a_0, b_1, c_0) : 2$	$(a_0, b_2, c_0) : 3$
$(a_0, b_0, c_1) : 1$	$(a_0, b_1, c_1) : 2$	$(a_0, b_2, c_1) : 3$
$(a_0, b_0, c_2) : 1$	$(a_0, b_1, c_2) : 2$	$(a_0, b_2, c_2) : 3$

Purpose

- Have a better understanding of the BUC algorithm.

Requirements

- Provide all the intermediate steps and explain how you calculated the final values. Please keep it brief and clear.
- For (a), you are allowed to use any software to draw the tree; paste your plot to the final pdf file.

a. (6, L1) Draw the trace tree of expansion with the exploration order $A \to B \to C$ and $B \to C \to A$.

b. (7, L3) If we set $min\_support = 8$ with the exploration order $A \to B \to C$, 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.

c. (7, L3) Propose another exploration order which will result in a lower number of cells being computed compared to (b) with $min\_support = 8$. Please list each computed cell with its count, and report whether it is expansible in the **BUC** process.

Machine Problems (30 points)

Requirements

- You can access the MPs for Assignment 2 using this link for Machine Problems.
- MP5 is **required** to be solved by all students. The deadline for the required MPs is same as the Assignment.
- You should attach the output figures and codes (if asked to attach) for MP5 in the same **PDF** file as your Assignment 1.
- MP6 and MP7 are *optional* and you will get extra credits for solving them. We have separate submission for optional MPs on Compass2G.
- The deadline for optional MPs (MP6 and MP7) is two days after the Midterm Exam (07/13/2016).
- Each MP webpage consists of instructions, input data and the expected output data.