DBDM Assignment 2

Paul Couzy (s1174347), Irene Haasnoot (s1258745),
Eva van Weenen (s1376969), Alex van Vorstenbosch (s1526146)

October 2018

Question 1

The Census-Income (KDD) Data Set

(a) Explain how many 1-d aggregate cuboids there are in total, and calculate how many cells each of them consists of.
The amount of 1-d aggregate cuboids is the same as the amount of columns. The number of cells in each 1-d aggregate cuboid is the number of unique elements in each column.
(Attached: python code `assignment2.py` and `1D_length.csv`)

(b) Using sum() as aggregation function, calculate the Apex (0-d aggregate cuboid), the smallest and the largest 1-d aggregate cuboid (excluding "target" as dimensions), and for each 1-d aggregate cuboid the respective 2-d aggregate cuboid by adding "target" as second dimension. *Store each X-d aggregate cuboid as table with X+1 columns (X dimensions plus one aggregated measurement) in a separate CSV file. The records of the 1-d aggregate cuboids should be sorted on ascending dimension values. The records of the 2-d aggregate cuboids should be clustered on "target" and for each "target" value sorted on ascending dimension values.*
(Attached: python code `assignment2.py` and `0D.csv`, `1D_largest.csv`, `1D_smallest.csv` and `2D_xxx.csv` where `xxx` is the column name.)

Question 2

Suppose that a base cuboid has three dimensions, A, B, C, with the following number of cells: $|A|$ = 1,000,000, $|B|$ = 100, and $|C|$ = 1000. Suppose that each dimension is evenly partitioned into 10 portions for chunking.

(a) Assuming each dimension has only one level, draw the complete lattice of the cube.
See Figure 1

(b) If each cube cell stores one measure with four bytes, what is the total size of the computed cube if the cube is dense?
Starting with the base cuboid the size is AxBxC times 4 bytes = $4 \cdot 10^{11}$ bytes
For the two dimensional cuboids the sizes are (AxB + AxC + BxC) times 4 bytes = 4,400,400,000 bytes
The one dimensional cuboids (A + B + C) have a size of 4,004,400 bytes
Then for the apex, containing 1 cell, we have an extra 4 bytes.

This totals to 404,404,404,404 bytes $\approx$ 404.4 GB

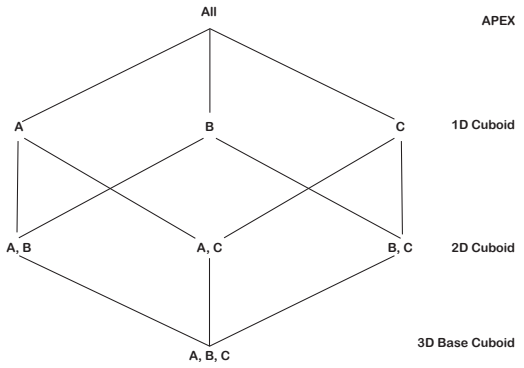

Figure 1

(c) State the order for computing the chunks in the cube that requires the least amount of space and compute the total amount of main memory space required for computing the 2-dimensional planes.

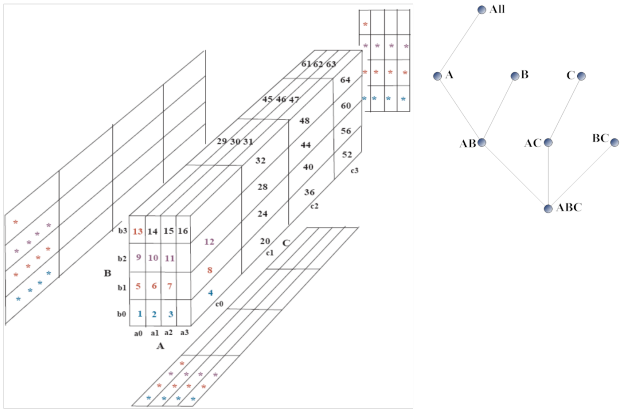

Figure 2

Looking at the example figure 2, we need to store the AB plane, a column of the AC plane and one chunk of the BC plane.

To keep the memory needed to a minimum, we should aggregate in the order B-C-A, so that the smallest plane BxC is stored completely and for the biggest plane (AxC) only the chunks need to be stored. This totals to:

$(10^2 \cdot 10^3) + (10^2 \cdot \frac{10^6}{10}) + (\frac{10^3}{10} \cdot \frac{10^6}{10}) = 2.01 \cdot 10^7$ memory units.

With 4 bytes per unit the required memory results in 80.4 MB

Question 3

Assume that a 9-dimensional cuboid contains only three base cells: (1) (a1, d2, d3, d4, ..., d9), (2) (d1, b2, d3, d4, ..., d9), and (3) (d1, d2, c3, d4, ..., d9), where a1 $\neq$ d1, b2 $\neq$ d2,

and c3 $\neq$ d3. The measure for the cube is count().

(a) How many non-empty cuboids will a full data cube contain?
There are nine dimensions, for each dimension there is the option that it is in the combination or not. This gives us 2^9 non-empty cuboids.

(b) How many non-empty aggregate (i.e., non-base) cells will a full cube contain?
It will contain 3 times all the combinations minus the base cell. Therefore there are $3 \cdot (2^9 - 1)$ non-empty aggregate cells in a full cube. While doing this some combinations are counted several times. Therefore we have to correct for this. The combinations that we counted several times are:

$(*, *, *, d4, d5, d6, d7, d8, d9)$	counted three times therefore subtract $2 \cdot 2^6$
$(\boldsymbol{d1}, *, *, d4, d5, d6, d7, d8, d9)$	counted twice therefore subtract 2^6
$(*, \boldsymbol{d2}, *, d4, d5, d6, d7, d8, d9)$	counted twice therefore subtract 2^6
$(*, *, \boldsymbol{d3}, d4, d5, d6, d7, d8, d9)$	counted twice therefore subtract 2^6

where normal text (e.g. $d4$) indicates a non-fixed position so with two options for the position, $*$ indicates a fixed empty position and bold text (e.g. $\boldsymbol{d1}$) indicates a fixed filled position. (We only use this notation for question 3b and 3c.) So the final number of non-empty aggregate cells in a full cube is $3 \cdot (2^9 - 1) - 5 \cdot 2^6 = 1213$

(c) How many non-empty aggregate cells will an iceberg cube contain if the condition of the iceberg cube is "count >= 2"?

Count = 3 :$(*, *, *, d4, d5, d6, d7, d8, d9)$	2^6 combinations
Count = 2 :$(\boldsymbol{d1}, *, *, d4, d5, d6, d7, d8, d9)$	2^6 combinations
$(*, \boldsymbol{d2}, *, d4, d5, d6, d7, d8, d9)$	2^6 combinations
$(*, *, \boldsymbol{d3}, d4, d5, d6, d7, d8, d9)$	2^6 combinations

So the total number of non-empty aggregate cells in an icecube if the count > 2 is $4 \cdot 2^6 = 256$

(d) A cell c is a closed cell if there exists no cell d such that d is a specialization of cell c (i.e., d is obtained by replacing a * in c by a non-* value) and d has the same measure value as c. A closed cube is a data cube consisting of only closed cells. How many closed cells are in the full cube?
We have the 4 trivial closed cells, the 3 base cells given and their intersection of all the 3 base cells:

$$(a1, d2, d3, d4, d5, d6, d7, d8, d9) : 1$$
$$(d1, b2, d3, d4, d5, d6, d7, d8, d9) : 1$$
$$(d1, d2, c3, d4, d5, d6, d7, d8, d9) : 1$$
$$(*, *, *, d4, d5, d6, d7, d8, d9) : 3$$

Now we need to find the 3 intersection between two of the base cells, this gives us the last closed cells with measure 2.

$$(d1, *, *, d4, d5, d6, d7, d8, d9) : 2$$
$$(*, d2, *, d4, d5, d6, d7, d8, d9) : 2$$
$$(*, *, d3, d4, d5, d6, d7, d8, d9) : 2$$

This gives us finally 7 closed cells:

$$(a1, d2, d3, d4, d5, d6, d7, d8, d9) : 1$$
$$(d1, b2, d3, d4, d5, d6, d7, d8, d9) : 1$$
$$(d1, d2, c3, d4, d5, d6, d7, d8, d9) : 1$$
$$(d1, *, *, d4, d5, d6, d7, d8, d9) : 2$$
$$(*, d2, *, d4, d5, d6, d7, d8, d9) : 2$$
$$(*, *, d3, d4, d5, d6, d7, d8, d9) : 2$$
$$(*, *, *, d4, d5, d6, d7, d8, d9) : 3$$

Question 4

Consider a data cube C with D dimensions where each dimension has exactly V distinct values in the base cuboid. Assume there are no concept hierarchies associated with the dimensions.

(a) What is the minimum number of (non-empty) cells possible in the base cuboid?
We have V distinct values for each dimension. The minimal number of cells would be V distinct tuples, where each tuple is distinct with the other tuples in all the dimensions. These V tuples, become the V cells, which are the minimal amount needed for the base cuboid.
The minimum number of cells can be illustrated with the following number of tuples:

$$\left\{ \begin{array}{ll} (a_1, a_2, & .., a_D) \\ (b_1, b_2, & .., b_D) \\ (c_1, c_2, & .., c_D) \\ \vdots & \vdots \end{array} \right\} V$$

(b) What is the maximum number of (non-empty) cells possible in the base cuboid?
The maximum number of cells is where each combination of distinct values in each dimension exists. So for each dimension $d \in D$ we have V distinct values: $\left\{ \begin{array}{l} a_d \\ b_d \\ c_d \\ \vdots \end{array} \right\} V$

Therefore there can be V^D different cells.

(c) What is the minimum number of (non-empty) cells possible in the entire data cube C (including both base cells and aggregate cells)?
We have 2^D cuboids, this follows from question 3a, this includes the apex. For each cuboid, the minimal number of cells is V, as there are V distinct values for each dimension. This would give us $2^D \cdot V$ cells, but the apex cuboid is different, it does not contain V cells, but only 1 by construction. This gives us the minimal number of cells for the full cube is $(2^D - 1) \cdot V + 1$.

(d) What is the maximum number of (non-empty) cells possible in the entire data cube C (including both base cells and aggregate cells)? We need the maximum of cells for the entire data cube and it is similar to question 4b. In addition to the V values for the dimension we get one additional value namely $*$, this comes from the aggregation of that cell on that dimension. This would mean that the entire amount of cells is $(V + 1)^D$.