COMP9318 (17S1) ASSIGNMENT 1

DUE ON 23:59 28 MAY, 2017 (SUN)

Q1. (*40 marks*)

Consider the following base cuboid *Sales* with *four* tuples and the aggregate function SUM:

Location	*Time*	*Item*	*Quantity*
Sydney	2005	PS2	1400
Sydney	2006	PS2	1500
Sydney	2006	Wii	500
Melbourne	2005	XBox 360	1700

Location, *Time*, and *Item* are dimensions and *Quantity* is the measure. Suppose the system has built-in support for the value **ALL**.

(1) List the tuples in the complete data cube of R in a tabular form with 4 attributes, i.e., $Location, Time, Item, \text{SUM}(Quantity)$?

(2) Write down an equivalent SQL statement that computes the same result (i.e., the cube). You can *only* use standard SQL constructs, i.e., no **CUBE BY** clause.

(3) Consider the following *ice-berg cube* query:

```
SELECT  Location, Time, Item, SUM(Quantity)
FROM    Sales
CUBE BY Location, Time, Item
HAVING  COUNT(*) > 1
```

Draw the result of the query in a tabular form.

(4) Assume that we adopt a MOLAP architecture to store the full data cube of R, with the following mapping functions:

$$f_{Location}(x) = \begin{cases} 1 & \text{if } x = \text{‘Sydney’}, \\ 2 & \text{if } x = \text{‘Melbourne’}, \\ 0 & \text{if } x = \textbf{ALL}. \end{cases}$$

$$f_{Time}(x) = \begin{cases} 1 & \text{if } x = 2005, \\ 2 & \text{if } x = 2006, \\ 0 & \text{if } x = \textbf{ALL}. \end{cases}$$

$$f_{Item}(x) = \begin{cases} 1 & \text{if } x = \text{‘PS2’}, \\ 2 & \text{if } x = \text{‘XBox 360’}, \\ 3 & \text{if } x = \text{‘Wii’}, \\ 0 & \text{if } x = \textbf{ALL}. \end{cases}$$

Draw the MOLAP cube (i.e., sparse multi-dimensional array) in a tabular form of $(ArrayIndex, Value)$. You also need to write down the function you chose to map a multi-dimensional point to a one-dimensioinal point.

Q2. (*30 marks*)

Consider binary classification where the class attribute y takes two values: 0 or 1. Let the feature vector for a test instance be a d-dimension column vector $\boldsymbol{x}$. A linear classifier with the model parameter $\boldsymbol{w}$ (which is a d-dimension column vector) is the following function:

$$y = \begin{cases} 1 & \text{, if } \boldsymbol{w}^\top \boldsymbol{x} > 0 \\ 0 & \text{, otherwise.} \end{cases}$$

We make additional simplifying assumptions: $\boldsymbol{x}$ is a binary vector (i.e., each dimension of $\boldsymbol{x}$ take only two values: 0 or 1).

- Prove that if the feature vectors are d-dimension, then a Naïve Bayes classifier is a linear classifier in a $d+1$-dimension space. You need to explicitly write out the vector $\boldsymbol{w}$ that the Naïve Bayes classifier learns.
- It is obvious that the Logistic Regression classifier learned on the same training dataset as the Naïve Bayes is also a linear classifier in the same $d+1$-dimension space. Let the parameter $\boldsymbol{w}$ learned by the two classifiers be $\boldsymbol{w}_{\text{LR}}$ and $\boldsymbol{w}_{\text{NB}}$, respectively. Briefly explain why learning $\boldsymbol{w}_{\text{NB}}$ is much easier than learning $\boldsymbol{w}_{\text{LR}}$.

Hint 1. $\log\left(\prod_i x_i\right) = \sum_i \log\left(x_i\right)$

Q2. (*30 marks*)

Consider the (slightly incomplete) k-means clustering algorithm as depicted in Algorithm 1.

(1) Assume that the stopping criterion is till the algorithm converges to the final k clusters. Can you insert several lines of pseudo-code to the algorithm to implement this logic? You are **not** allowed to change the first 7 lines though.

(2) The cost of k clusters is just the total cost of each group g_i, or formally

$$cost(g_1, g_2, \ldots, g_k) = \sum_{i=1}^{k} cost(g_i)$$

Algorithm 1: k-means(D, k)

```
   Data: $D$ is a dataset of $n$ $d$-dimensional points; $k$ is the number of clusters.
1  Initialize $k$ centers $C = [c_1, c_2, \ldots, c_k]$;
2  $canStop \leftarrow$ false;
3  while $canStop =$ false do
4      Initialize $k$ empty clusters $G = [g_1, g_2, \ldots, g_k]$;
5      for each data point $p \in D$ do
6          $c_x \leftarrow$ NearestCenter($p, C$);
7          $g_{c_x}$.append($p$);
8      for each group $g \in G$ do
9          $c_i \leftarrow$ ComputeCenter($g$);
10 return $G$;
```

$cost(g_i)$ is the sum of squared distances of all its constituent points to the center c_i, or

$$cost(g_i) = \sum_{p \in g_i} dist^2(p, c_i)$$

$dist()$ is the Euclidean distance. Now show that the cost of k clusters as evaluated at the end of each iteration (i.e., after Line 9 in the current algorithm) never increases.

(3) Prove that the cost of clusters obtained by k-means algorithm always converges to a local minima. You can make use of the previous conclusion even if you have not proved it.

Hint 2. *Show that the two loops (Lines 5–7 and Lines 8–9) never increases the cost.*

Submission

Please write down your answers in a file named `ass1.pdf`. You **must write down your name and student ID on the first page**.

You can submit your file by

```
give cs9318 ass1 ass1.pdf
```

Late Penalty. -10% per day for the first two days, and -30% for each of the following days.