Instructions:
1. Please answer all questions in Part A.
2. You can optionally answer the bonus question in Part B. You can obtain additional marks for the bonus question if you answer it correctly.
3. The total marks in Part A are 100.
4. The total marks in Part B are 20.
5. The total marks you can obtain in this assignment are 100 only.
   - If you answer the bonus question in Part B correctly, you can obtain additional marks.
   - But, if the total marks you obtain from Part A and Part B are over 100, your marks will be truncated to 100 only.
6. You can use a coupon to waive any question you want in Part A only (not Part B) and obtain full marks for this question.
7. You can waive at most one question in Part A only (not Part B) in each assignment.
8. You can also answer the question you will waive. We will also mark it but will give full marks to this question.
9. The coupon is non-transferrable. That is, the coupon with a unique ID can be used only by the student who obtained it in class.
10. Please staple the coupon to the submitted assignment.
11. Please write down the question no. you want to waive on the coupon.

Part A (Compulsory Questions)

Q1 [20 Marks]

Consider the density-based subspace clustering. The size of a subspace is defined to be the total number of dimensions for this subspace. For example, subspace \{A, B\} is of size 2.

(a) When the size of the subspace is larger, it is less likely that a grid unit with respect to the subspace is dense. Please explain it.

(b) In order to overcome the weakness described in (a), instead of setting a fixed density threshold for the subspace of any size, we use a smaller density threshold for the subspace of larger size. Specifically, let \(T_i\) be the density threshold for the subspace of size \(i\). If \(i < j\), then \(T_i > T_j\). Let Condition 1 be “\(T_i > T_j\) for any \(i < j\)”.

Let Condition 2 be “for any \(i\) and \(j\), \(T_i = T_j\)”. We know that if Condition 2 is satisfied, then the original Apriori-like algorithm studied in class can find all subspaces containing dense units.

(i) Under Condition 1, is it always true that we can still adopt the Apriori-like algorithm? If yes, please describe how to adopt the algorithm. Otherwise, please give reasons why it cannot be adopted.

(ii) Suppose that we modify Condition 1 to the following form. Let Condition 1 be “\(T_i = \alpha T_{i+1}\) for each positive integer \(i\)” where \(\alpha\) is a positive real number at least 1 and is given by users. Assume that we adopt this new form of Condition 1. If we set \(\alpha\) to some values, we cannot adopt the Apriori-like algorithm. If we set \(\alpha\) to other values, we can adopt the Apriori-like algorithm. What is the greatest possible value of \(\alpha\) such that we can adopt the Apriori-like algorithm? Please explain it.
In order to overcome the weakness described in (a), we use the same number of grid units for any subspace, says N, and thus the density threshold is set to a fixed value T for any subspace. For example, if N = 4, the total number of grid units with respect to a subspace of size 2 is 4 (Figure 1a) and the total number of grid units with respect to a subspace of size 1 is also 4 (Figure 1b). Let n be the total number of dimensions. Suppose we set N = 2^n. Same as before, we want to find all subspaces containing dense units under Condition 2. Can we still adopt the Apriori-like algorithm? If yes, please describe how to adopt the algorithm. Otherwise, please give reasons why it cannot be adopted.

Q2 [20 Marks]

Consider the entropy-based subspace clustering. The size of a subspace is defined to be the total number of dimensions for this subspace. For example, subspace \{A, B\} is of size 2.

(a) Is the following statement true? If yes, please give a formal proof. If no, please give a counter example.

“When the size of the subspace is larger, it is less likely or equally likely that the subspace has a good clustering.”

(b) Suppose that c is a positive real number where we do not know the exact value. Similarly, d is also another positive real number where d is equal to c+5.

(i) Consider the four 2-dimensional data points:

\[(a:7 + c, 7 + c),\ (b:9 + c, 9 + c),\ (c:6 + c, 10 + c),\ (d:10 + c, 6 + c)\]

We can make use of the KL-Transform to find a transformed subspace containing a cluster. Let L be the total number of dimensions in the original space and K be the total number of dimensions in the projected subspace. Suppose that L = 2 and K = 1. Please illustrate with the above example.

(ii) Consider the four 2-dimensional data points:

\[(a:7 – d, 7 – d),\ (b:9 – d, 9 – d),\ (c:6 – d, 10 – d),\ (d:10 – d, 6 – d)\]

We can make use of the KL-Transform to find a transformed subspace containing a cluster. Let L be the total number of dimensions in the original space and K be the total number of dimensions in the projected subspace. Suppose that L = 2 and K = 1.

Can we make use of the answers in part (b)(i) to perform the KL-Transform? If yes, please write down each transformed data point. If no, please write down the reasons why we cannot make use of the answers of part (b)(i).

(iii) Consider the four 2-dimensional data points:

\[(a:7c, 7c),\ (b:9c, 9c),\ (c:6c, 10c),\ (d:10c, 6c)\]

We can make use of the KL-Transform to find a transformed subspace containing a cluster. Let L be the total number of dimensions in the original space and K be the total number of dimensions in the projected subspace. Suppose that L = 2 and K = 1.
Can we make use of the answers in part (b)(i) to perform the KL-Transform? If yes, please write down each transformed data point. If no, please write down the reasons why we cannot make use of the answers of part (b)(i).

Q3 [20 Marks]

The following shows a history of customers with their incomes, ages and an attribute called “Play_PokemonGO” indicating whether they played Pokemon GO. We also indicate whether they will buy a Nintendo Switch or not in the last column.

<table>
<thead>
<tr>
<th>No.</th>
<th>Income</th>
<th>Age</th>
<th>Play_PokemonGO</th>
<th>Buy_NintendoSwitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>high</td>
<td>young</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>2</td>
<td>high</td>
<td>old</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>young</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>4</td>
<td>high</td>
<td>old</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>medium</td>
<td>young</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>6</td>
<td>medium</td>
<td>young</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>7</td>
<td>medium</td>
<td>old</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>8</td>
<td>medium</td>
<td>old</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

(a) We want to train a CART decision tree classifier to predict whether a new customer will buy a Nintendo Switch or not. We define the value of attribute Buy_NintendoSwitch to be the label of a record.

(i) Please find a CART decision tree according to the above example. In the decision tree, whenever we process a node containing at most 3 records, we stop to process this node for splitting.

(ii) Consider a new young customer whose income is medium and he played Pokemon GO. Please predict whether this new customer will buy a Nintendo Switch or not.

(b) What is the difference between the C4.5 decision tree and the ID3 decision tree? Why is there a difference?
### Q4 [20 Marks]

We have the following Bayesian Belief Network.

```
<table>
<thead>
<tr>
<th>Family History (FH)</th>
<th>Smoker (S)</th>
<th>PositiveXRay (PR)</th>
<th>Lung Cancer (LC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FH = Yes</td>
<td>S = Yes</td>
<td>PR = Yes</td>
<td>LC = Yes</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6</td>
<td>0.85</td>
<td>0.7</td>
</tr>
<tr>
<td>FH = Yes</td>
<td>S = Yes</td>
<td>PR = Yes</td>
<td>LC = Yes</td>
</tr>
<tr>
<td>0.7</td>
<td></td>
<td></td>
<td>0.85</td>
</tr>
<tr>
<td>FH = Yes</td>
<td>S = No</td>
<td>PR = Yes</td>
<td>LC = No</td>
</tr>
<tr>
<td>0.45</td>
<td></td>
<td></td>
<td>0.45</td>
</tr>
<tr>
<td>FH = No</td>
<td>S = Yes</td>
<td>PR = Yes</td>
<td>LC = Yes</td>
</tr>
<tr>
<td>0.55</td>
<td></td>
<td></td>
<td>0.85</td>
</tr>
<tr>
<td>FH = No</td>
<td>S = No</td>
<td>PR = Yes</td>
<td>LC = No</td>
</tr>
<tr>
<td>0.2</td>
<td></td>
<td></td>
<td>0.45</td>
</tr>
</tbody>
</table>
```

Suppose that there is a new person. We know that
(1) he has his family history
(2) he is a non-smoker
(3) his result of X-Ray is positive

We would like to know whether he is likely to have Lung Cancer.

(a) Please use Bayesian Belief Network classifier with the use of Bayesian Belief Network to predict whether he is likely to have Lung Cancer.

(b) Although Bayesian Belief Network classifier does not have an independent assumption among all attributes (compared with the naïve Bayesian classifier), what are the disadvantages of this classifier?

### Q5 [20 Marks]

In the view materialization method discussed in class, there are two implicit assumptions.

- The first assumption is that the objective is to find the fixed number of views for materialization. However, in general, different views are of different sizes and the same number of views may not occupy the same amount of storage. In a more realistic problem setting, we are given available memory of fixed size $X$.
- The second assumption is that the probability that each possible view is queried is equal. However, in general, the probability that each possible view is queried is not equal.

In order to relax the above two assumptions, we assume that we are given available memory of fixed size $X$ and the probability that each view $v$ is queried denoted by $p(v)$. Please propose a solution for this problem setting.
Part B (Bonus Question)

Note: The following bonus question is an **OPTIONAL** question. You can decide whether you will answer it or not.

Q6 [20 Additional Marks]
We are given \( l \) attributes, namely \( A_1, A_2, \ldots, A_l \), and \( n \) data points. Suppose that \( n = 2^k \) where \( k \) is a positive integer \( \geq 2 \). Given a data point \( x \), we denote the value of attribute \( A \) for \( x \) to be \( x.A \). Assume that two different data points have different attribute values for each attribute.

In the density-based subspace clustering, we learnt that each (grid) unit can be represented by \( p \) intervals if we consider \( p \) attributes only where \( p \leq l \). “\( A_2 = [1, 10], A_6 = [21, 30] \)” is an example of the representation of a unit if we consider two attributes, \( A_2 \) and \( A_6 \). The number of attributes involved for this unit is 2. Besides, the length of the interval “\( A_i = [a, b] \)” is defined to be \( b-a \) where \( a \) and \( b \) are two real numbers. The volume of a unit is defined to be the product of the lengths of all intervals involved for this unit. Recall that in this density-based subspace clustering, we want to find all subspaces which contain dense units. Formally, given a subspace \( S \) in the result, there exists a unit such that the attributes involved for this unit are the attributes involved for \( S \) and this unit is dense.

However, in this density-based subspace clustering, the length of each interval along an attribute is fixed. Motivated by this observation, we want to define some intervals with varying lengths along an attribute according to the data distribution. In this way, we can define a unit based on these intervals.

According to the data, we can generate a set of intervals according to a function \( \text{Generate} \) (to be described). According to these intervals, we can define units. Before we describe function \( \text{Generate} \), we give some concepts as follows.

We split a set \( G \) of points into two parts according to an attribute \( A \) such that there exist four numbers \( y_1, y_2, y_3, y_4 \in A \) where
- \( y_1 < y_2 < y_3 < y_4 \)
- \( y_1 = \min_{x \in G} x.A \)
- \( y_2 \) is equal to the attribute value of a data point in \( G \)
- \( y_3 \) is also equal to the attribute value of a data point in \( G \)
- \( y_4 = \max_{x \in G} x.A \)
- the total number of data points in \( G \) which attribute values on \( A \) are at most \( y_2 \) is equal to the total number of data points in \( G \) which attribute values on \( A \) are at least \( y_3 \)
- there are no data points in \( G \) which attribute values on \( A \) are greater than \( y_2 \) and smaller than \( y_3 \)

Let \( \text{Left}(G, A) \) be a set of data points in \( G \) which attribute values on \( A \) are at most \( y_2 \) (described above).
Let \( \text{Right}(G, A) \) be a set of data points in \( G \) which attribute values on \( A \) are at least \( y_3 \) (described above).
Let \( \text{SplitValue}(G, A) \) be \( \frac{(y_2-y_1)+(y_4-y_3)}{2} \) (where \( y_1, y_2, y_3 \) and \( y_4 \) were described above).
Now, we define a function **Generate** which takes a set $G$ of data points as an input and outputs a set of intervals.

**Function Generate**($G$)

if $|G| \leq 4$

$X \leftarrow \emptyset$

for each $i \in [1, l]$ do

$y_1 \leftarrow \min_{x \in G} x.A_i$

$y_4 \leftarrow \max_{x \in G} x.A_i$

interval$_i \leftarrow [y_1, y_4]$

$X \leftarrow X \cup \text{"Ai = interval$_i"}$

else

$A_i \leftarrow$ the attribute A which has the greatest value of SplitValue($G, A$) (among all attributes)

$G_{\text{left}} \leftarrow \text{Left}(G, A_i)$

$G_{\text{right}} \leftarrow \text{Right}(G, A_i)$

$X_{\text{left}} \leftarrow \text{Generate}(G_{\text{left}})$

$X_{\text{right}} \leftarrow \text{Generate}(G_{\text{right}})$

$X \leftarrow X_{\text{left}} \cup X_{\text{right}}$

**return** $X$

(a) Is it possible that the intervals along a single attribute which are generated by function **Generate** are overlapping? Why?

(b) Under this new definition of intervals, we have a new definition of “dense” as follows.

Suppose that we define that a unit is dense if this unit contains at least 4 data points and there exists an interval of this unit where the length of this interval is smaller than or equal to a support threshold $t$ where $t$ is a non-negative real number and a user parameter. Can we still adopt the Apriori-like algorithm for finding all subspaces containing dense units? If yes, please describe how to adopt the algorithm. Otherwise, please give reasons why it cannot be adopted.

(c) This part is independent of part (b).

Under this new definition of intervals, we have a new definition of “dense” as follows.

Suppose that we define that a unit is dense if this unit contains at least 4 data points and the volume of this unit is smaller than or equal to $t^s$ where $t$ is a non-negative real number and a user parameter, and $s$ is the number of attributes involved for this unit. Can we still adopt the Apriori-like algorithm for finding all subspaces containing dense units? If yes, please describe how to adopt the algorithm. Otherwise, please give reasons why it cannot be adopted.

(d) What are the advantages of using this approach for subspace clustering compared with the density-based subspace clustering you learnt in class?