COMP1942 Exploring and Visualizing Data (Spring Semester 2013)
Midterm Examination (Question Paper)
Date: 22 March, 2013 (Fri)
Time: 9:00-10:15
Duration: 1 hour 15 minutes

Student ID:__________________ Student Name:__________________________________

Seat No. :__________________

Instructions:
(1) Please answer **all** questions in **Part A** and **Part B** in this paper.
(2) You can **optionally** answer the bonus question in **Part C** in this paper. You can obtain additional marks for the bonus question if you answer it correctly.
(3) The total marks in Part A and part B are 100.
(4) The total marks in Part C are 10.
(5) The total marks you can obtain in this exam are 100 only.
If you answer the bonus question in Part C correctly, you can obtain additional marks.
But, if the total marks you obtain from Part A, Part B and Part C are over 100, your marks will be truncated to 100 only.
(6) You can use a calculator.

Answer Sheet

Part	Question	Full Mark	Mark
A	Q1	20	
	Q2	20	
	Q3	20	
	Q4	20	
B	Q5-Q8	20	
Total (Parts A and B)		100	
C	Q9 (OPTIONAL)	10	
Total (Parts A, B and C)		100	

Part A (Compulsory Short Questions)

Q1 (20 Marks)
(a) (i)

support = 3

(ii)

confidence = 3/4
= 0.75

(iii)

Lift ratio = (3/4)/(3/5)
= 5/4
= 1.25

(iv)(1)

C_2 = {{P, R}, {P, S}, {P, T}, {R, S}, {R, T}, {S, T}}

(2)

C_3 = { {P, R, T} }

Q1 (Continued)
(b)

a
a, b
a, b
a, b
a, c
a, b, c
a, b, c
b
b
b
b, c

Q2 (20 Marks)
(a)

Consider the correlation between A and B.

B\A	**1**	**0**
1	2	0
0	1	1

$X_{AB}^2 = 1.33$

Consider the correlation between A and C.

C\A	**1**	**0**
1	1	1
0	2	0

$X_{AC}^2 = 1.33$

Consider the correlation between B and C.

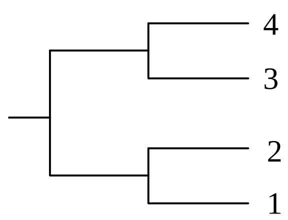
C\B	**1**	**0**
1	0	2
0	2	0

$X_{BC}^2 = 4$

For attribute A,
$X_{AB}^2 + X_{AC}^2 = 1.33 + 1.33 = 2.66$
For attribute B,
$X_{AB}^2 + X_{BC}^2 = 1.33 + 4 = 5.33$
For attribute C,
$X_{AC}^2 + X_{BC}^2 = 1.33 + 4 = 5.33$

We choose attribute B for splitting since it has the largest value.
We divide the data into two groups, namely {1, 2} and {3, 4}.

Dendrogram:



Q2 (Continued)
(b)

No. This is because we do not know the distance between cluster (a, b) and cluster (c d) and the distance between (a b) and e.

Q3 (20 Marks)

(a) (i)

Make initial guesses of the means $m_1, m_2, \dots, m_k$
Set the counts $n_1, n_2, \dots, n_k$ to zero
Until interrupted
  Acquire the next example x
  If m_i is closest to x
    Increment n_i
    Replace m_i by $m_i + 1/n_i\,(x - m_i)$

(ii)

x_j : the j-th example in cluster i
$m_i(t)$: the mean vector of cluster i containing t examples

Consider that x is the t-th example in cluster i

$$m_i(t-1) = \frac{x_1 + x_2 + \dots + x_{t-1}}{t-1}$$

$$\begin{aligned} m_i(t) &= \frac{x_1 + x_2 + \dots + x_{t-1} + x_t}{t} \\ &= \frac{m_i(t-1) \times (t-1) + x_t}{t} \\ &= \frac{t \times m_i(t-1) + x_t - m_i(t-1)}{t} \\ &= m_i(t-1) + \frac{1}{t}(x_t - m_i(t-1)) \end{aligned}$$

(b)

Class	# Cases	# Errors	% Error
Yes	5	3	60.00
No	7	2	28.57
Overall	12	5	41.67

Q4 (20 Marks)
(a)

Info(T) = $1-0.5^2-0.5^2=0.5$
For attribute Income,

Info(T_{high})= $1-1^2-0^2=0$

Info(T_{medium})= $1-(\frac{1}{5})^2-(\frac{4}{5})^2=0.32$

Info(Income, T)= $\frac{3}{8}Info(T_{high})+\frac{5}{8}Info(T_{medium})=0.2$

Gain(Income, T)= Info(T)-Info(Income, T)=0.3

For attribute Age,

Info(T_{young})=$1-0.5^2-0.5^2=0.5$

Info(T_{old})=$1-0.5^2-0.5^2=0.5$

Info(Age, T)=$\frac{1}{2}Info(T_{young})+\frac{1}{2}Info(T_{old})=0.5$

Gain(Age, T)= Info(T)-Info(Age, T)=0

For attribute Have_iPhone,

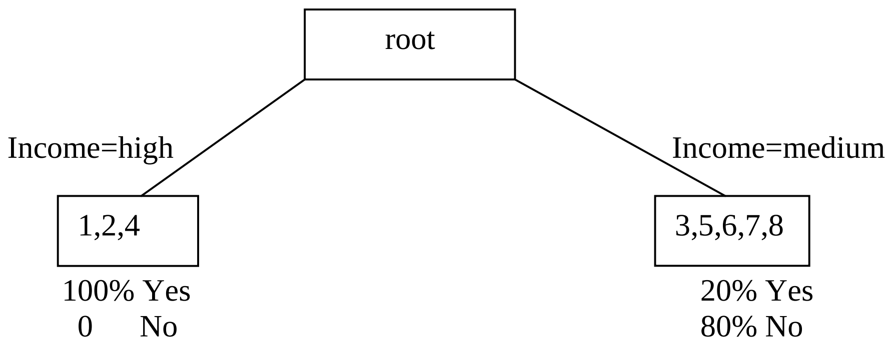
Info(T_{yes})=$1-1^2-0^2=0$

Info(T_{no})=$1-(\frac{1}{3})^2-(\frac{2}{3})^2=0.4444$

Info(Have_iPhone, T)= $\frac{1}{4}Info(T_{yes})+\frac{3}{4}Info(T_{no})=0.3333$

Gain(Have_iPhone, T)= Info(T)-Info(Have_iPhone, T)= 0.1667

We choose attribute Income for Splitting:



Consider the node for “Income=medium”

Info(T)= $1-(\frac{1}{5})^2-(\frac{4}{5})^2=0.32$

Q4 (Continued)

For attribute Age,

$\text{Info}(T_{young})=1-(\frac{1}{3})^2-(\frac{2}{3})^2=0.4444$

$\text{Info}(T_{old})=1-1^2-0^2=0$

$\text{Info(Age, T)}=\frac{3}{5}Info(T_{young})+\frac{2}{5}Info(T_{old})=0.26664$

Gain(Age, T)= Info(T)-Info(Age, T)= 0.05336

For attribute Have_iPhone,

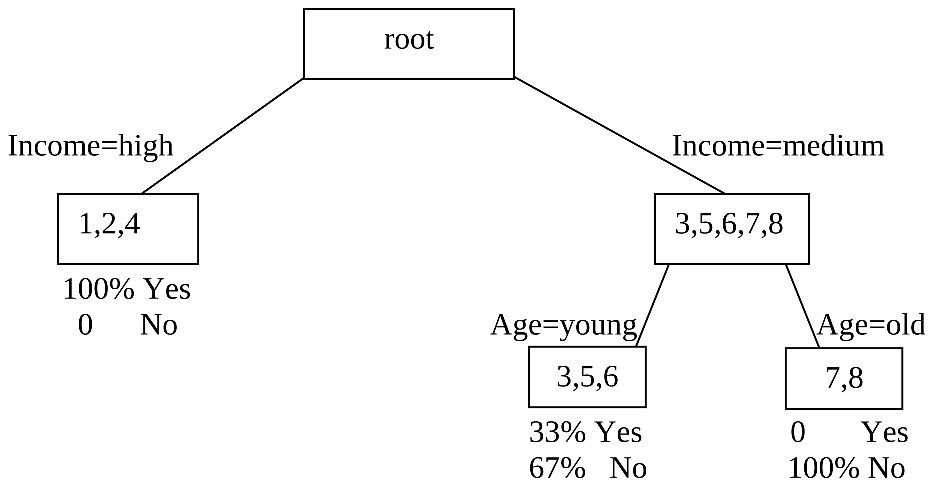
$\text{Info}(T_{yes})$=undefined

$\text{Info}(T_{no})=1-(\frac{1}{5})^2-(\frac{4}{5})^2=0.32$

Info(Have_iPhone, T)= $0\times Info(T_{yes})+1\times Info(T_{no})=0.32$

Gain(Have_iPhone, T)= Info(T)-Info(Have_iPhone, T)= 0

We choose attribute Age for Splitting:



Q4 (Continued)

(b) It is likely that he will not buy an iPad.

Part B (Compulsory Multiple-Choice (MC) Questions)

Note: For each question in this part, you just need to write down one of the five possible choices (i.e., A, B, C, D or E). The total scores in this part are 20 scores. Each question in this part weighs 5 scores.

Question	Answer
Q5	X
Q6	X
Q7	X
Q8	X

Part C (Bonus Question)

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

Q9 (10 Additional Marks)
(a)

Yes. It can be adapted.

First of all, we obtain the 2-sequence by scanning the database.
(NOTE: The 2-sequence contains two kinds of sequences – (1) the sequence contains only one *timestamp entry* (e.g. <{D, E}>) and (2) the sequence contains two or more timestamp entries (e.g. <{D}, {E}>).

The Apriori-like algorithm is described as follows.

1. k=2
2. Find all frequent 2-sequences and store them in L_k
3. repeat
4.   k=k+1
5.   Generate candidate k-sequences from L_{k-1} (which will be described later) and store them in C_k
6.   Scan the database and count the support of each candidate in C_k
7.   Find the k-sequence in C_k with support ≧ minsupport and store them in L_k
8. until L_k = empty set
9. return L_i for i=2,…k

e.g.
We obtain the following 2-sequences.
{<{A}, {D}>, <{A}, {E}>, <{A}, (G}>, <{D, E}> }

Next, we generate the candidate 3-sequences by the join-and-prune process.

The *join* step of the generation process is described as follows.
A sequence $s^{(1)}$ is joined with another sequence $s^{(2)}$ only if the subsequence obtained by dropping the first item in $s^{(1)}$ is identical to the subsequence obtained by dropping the last item in $s^{(2)}$. The resulting candidate is the sequence $s^{(1)}$, concatenated with the last item from $s^{(2)}$. The last item from $s^{(2)}$ can either be joined into the same timestamp element as the last item in $s^{(1)}$ or different timestamp elements depending on the following conditions.

1. If the last two items in $s^{(2)}$ belong the same timestamp element, then the last item in $s^{(2)}$ is part of the last timestamp element in $s^{(1)}$ in the joined sequence. (e.g. Suppose we have frequent sequences <{A},{B},{C}> and <{B},{C, D}> in L_{k-1}. Candidate <{A},{B},{C, D}> is obtained by joining <{A},{B},{C}> and <{B), {C, D}>).
2. If the last two items in $s^{(2)}$ belong to different timestamp elements, then the last item in $s^{(2)}$ becomes a separate timestamp element appended to the end of $s^{(1)}$ in the joined sequence. (e.g. Suppose we have frequent sequences <{A},{B},{C}> and <{B},{C},{D}> in L_{k-1}. Candidate <{A},{B},{C},{D}> is obtained by joining <{A},{B},{C}> and <{B},{C},{D}>).

e.g. In the running example, we obtain one candidate 3-sequence <{A}, {D,E}> (by joining <{A}, {D}> and <{D, E}>) after the join step.

Q9 (Continued)

The *prune* step of the generation process is described as follows.
A candidate k-sequence is pruned if at least one of its (k-1)-sequence is infrequent.
For example, <{A}, {D,E}> is a candidate 3-sequence. We need to check whether <{A}, {E}> is a frequent 2-sequence (NOTE: We do not need to check whether <{A}, {D}> and <{D, E}> are frequent 2-sequence because <{A}, {D,E}> was generated from these two frequent sequences). Since <{A}, {E}> is frequent, <{A}, {D,E}> is also considered as a candidate 3-sequence after the prune step.

Then, we do the *counting* step to count the support of each candidate in the set.
As the support of <{A}, {D,E}> is 2, then it is one of the final results.

We repeat the process until L_k is an empty set.

In our running example, all sequences with support at least 2 are {<{A}, {D,E}> }

(b)

No. It cannot be adapted.
This is because the Apriori property cannot be satisfied.

Consider the following example containing three sequences for three customers.

<{A}, {B}, {B}, {C}>
<{A}, {B}>
<{A}, {B}>

The support of a 2-sequence <{A}, {B}> is equal to 4/3 = 1.33.
The support of a 3-sequence <{A}, {B}, {C}> is equal to 2/1 = 2.
Since the 3-sequence <{A}, {B}, {C}> can be derived from 2-sequence <{A}, {B}> by appending one element at the end (with a new timestamp), and the support of this 3-sequence is larger than the support of this 2-sequence, the Apriori property cannot be satisfied.

Q9 (Continued)

End of Answer Sheet