CS412: Introduction to Data Mining
Assignment 4
Li Miao

Problem 1

a. Monotone and succinct
When a pattern satisfies C by having v, then all its super patterns also satisfy C, so it is monotone
Any sets satisfy C has a common subset includes value v, so it is succinct

b. Monotone and succinct
For a pattern satisfy C that max(S) >= v, its super patterns also satisfy C, so it is monotone
Since there exists A1 that contain all values larger than v, so S contains a subset belonging to A1, so it is succinct.

c. Anti-monotone and succinct
For a pattern violate C that max(S)<=v, its super patterns also violate C, so it is anti-monotone
Let A1 contains all values less than v, so set S contains a subset belonging to A1, so it is succinct.

d. Both are Strongly convertible
avg(S) >= v
When sorting values in ascending order, monotone
When sorting values in descending order, anti-monotone

avg(S) <=v
When sorting values in ascending order, anti-monotone
When sorting values in descending order, monotone

Problem 2

a. T. Within the level wise framework, no direct pruning based on the constraint can be made. There is an example on note that if we use avg(x) >=v, then we would prune an itemset that should be kept.
b. F. It is not efficient when database cannot fit in main memory. But we could use integration of physical and pseudo-projection.
c. T. It is lossless because it is the property of closed pattern.
d. T. X^2 + X*(X-1)/2 = (3/2)*X^2 – (1/2) * X. The equation holds.
e. T. If both monotone and anti-monotone, it should be trivial. So nontrivial cannot be monotone and anti-monotone at the same time.

Problem 3

1)

C1

Candidate	Support
<b>	3
<c>	3
<d>	3
<e>	3
<f>	3
<g>	1

L1

Candidate	Support
<b>	3
<c>	3
<d>	3
<e>	3
<f>	3

2)

C2

	<b>	<c>	<d>	<e>	<f>
<b>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<db>	<dc>	<dd>	<de>	<df>
<e>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fb>	<fc>	<fd>	<fe>	<ff>

	<b>	<c>	<d>	<e>	<f>
<b>		<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>			<(cd)>	<(ce)>	<(cf)>
<d>				<(de)>	<(df)>
<e>					<(ef)>
<f>					

L2

	<b>	<c>	<d>	<e>	<f>
<b>			<bd>	<be>	<bf>
<c>			<cd>	<ce>	<cf>
<d>					<df>
<e>					<ef>
<f>					

3)

C3

<bdf>	<bef>	<cdf>	<cef>

L3

<bdf>	<bef>	<cdf>	<cef>

C4 and L4 are both empty, no elements. So the program terminates.

4)
P1

<b>
<c>
<d>
<e>
<f>

5)
<b> projected DB

<(_c)(de)f>
<cdef>
<(_c)dbef>

<c> projected DB

<(de)f>
<def>
<dbef>

<d> projected DB

<(_e)f>
<ef>
<bef>

<e> projected DB

<f>
<f>
<f>

<f> projected DB
Empty. No elements.

6)
Prefix <b>: <bd>,<be>,<bf>
<bd> projected DB

<(_e)f>
<ef>
<bef>

Prefix <bd>:<bdf> no projected DB, stop

<be> projected DB

<f>
<f>
<f>

Prefix <be>: <bef> no projected DB, stop

<bf> projected DB: none, stop

Prefix <c>: <cd>, <ce>, <cf>
<cd> projected DB

(_e)f
(ef)
(bef)

Prefix<cd>: <cdf> no projected DB

<ce> projected DB

<f>
<f>
<f>

Prefix<ce>: <cef> no projected DB, stop
<cf> projected DB: none, stop

Prefix <d>: <df>
<df> projected DB: none, stop

Prefix <e>: <ef>
<ef> projected DB: none, stop

7)
GSP is like Apriori-based approach, and it would scan the database at every step to generate new length candidates, and there will be too many candidates.
PrefixSpan is like FP-tree approach, go deeply first but not widely. Only scan database once. No candidate sequence needs to be generated and projected databases keep shrinking.

8)
When the minimum support is too small and the sequential patterns need to mine are too long, the PrefixSpan will outperform GSP because GSP would generate too many candidates and scan database over and over. It would cost a lot of time. In this case, PrefixSpan would be much faster and more efficient than GSP.

Problem 4
a)
Info(D) = I(6,6) = 1
Info_gpa(D) = 3/12*I(3,0) + 5/12*I(3,2) + 4/12*I(0,4) = 0.4045627
Info_univ(D) = 5/12*I(3,2)+3/12*I(2,1)+4/12*I(1,3) = 0.9045627
Info_published(D) = 5/12*I(3,2) + 7/12*I(3,4) = 0.9792792
Info_recom(D) = 8/12*I(5,3) + 4/12*I(1,3) = 0.9067154

Gain_gpa = 0.5954373
Gain_univ = 0.09543725
Gain_published = 0.02072084
Gain_recom = 0.09328462

Therefore GPA should be the first split variable.
For GPA = 4.0, all get accepted = yes
For GPA = 3.5, all get accepted = no
We only need to continue split GPA = 3.7

id	GPA	univ	published	recommendation	accpeted
4	3.7	Top-10	yes	good	yes
5	3.7	Top-20	no	good	yes
6	3.7	Top-30	yes	good	yes
7	3.7	Top-30	no	good	no
8	3.7	Top-10	no	good	no

Info(D) = I(3,2) = 0.9709506
Info_univ = 2/5*I(1,1) + 1/5*I(1,0) + 2/5*I(1,1) = 0.8
Info_published = 2/5*I(2,0) + 3/5*I(1,2) = 0.5509775
Info_recom = I(3,2) = 0.9709506

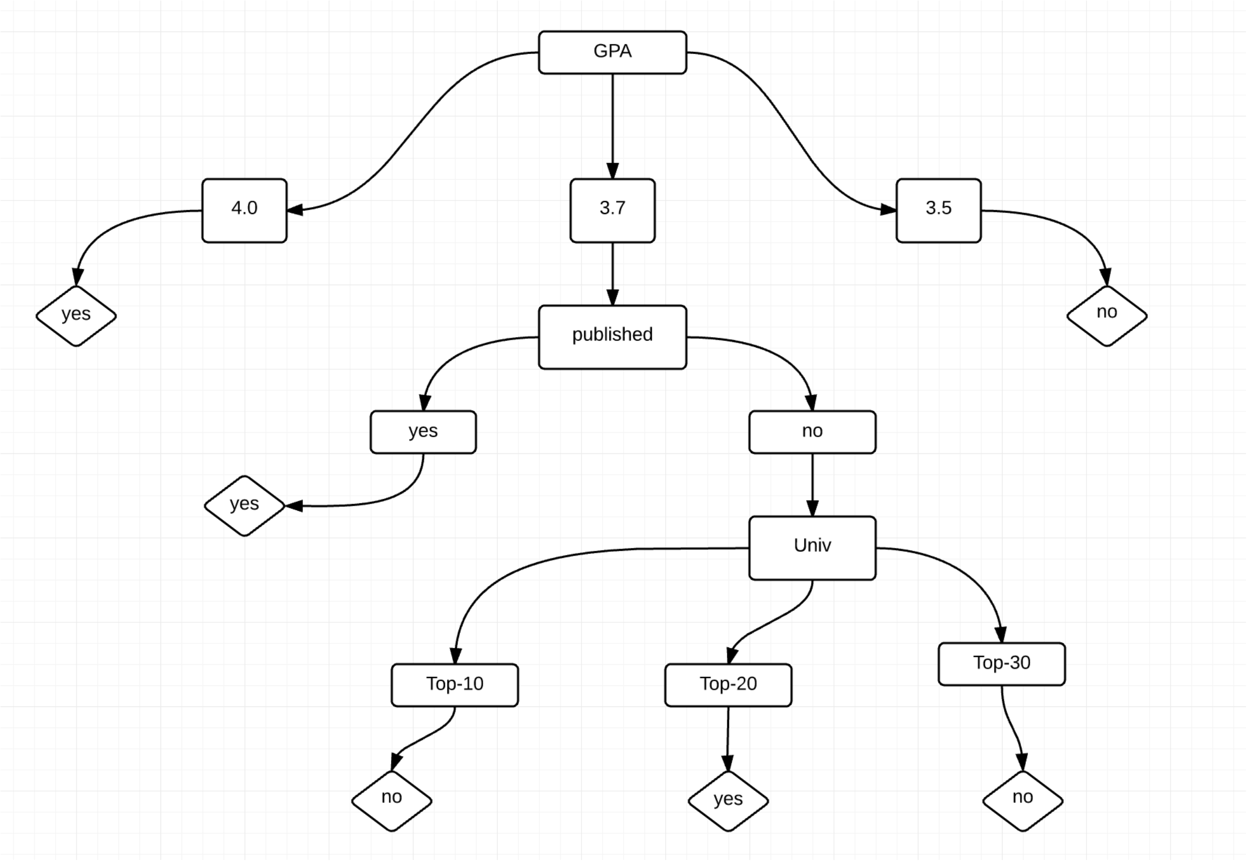
Gain_univ = 0.1709506
Gain_published = 0.4199731
Gain_recom = 0

So this time published is chosen to be the split variable.
For published = yes, all have accepted yes
So we continue when published = no

id	univ	Recommendation	accepted
5	Top-20	good	yes
7	Top-30	good	no
8	Top-10	good	no

Info(D) = I(2,1) = 0.9182958
Info_univ = 0
Info_recom = I(2,1) = 0.9182958
Gain_univ = 0.9182958
Gain_recom = 0

At this time, univ is the split variable. And all data are classified. So the program stop.



b)
For the test data, the prediction based on the tree above is following

id	accepted	prediction
1	yes	yes
2	yes	yes
3	yes	no
4	no	no
5	no	no

Precision = 2/(2+0) = 1, Recall = 2/(2+1) = $\frac{2}{3}$

c)
The time complexity is O(n*m^2). There are m attributes, so for the tree the worst case we have to consider m levels. At each level, we consider one attribute. And for each attribute, we need consider n observations. So totally, n*m + n*(m-1) + … = O(n* m^2)

d)
1.
If GPA is 4.0, then the instance has class “yes”
If GPA is 3.5, then the instance has class “no”
If GPA is 3.7 and published is yes, then the instance has class “yes”
If GPA is 3.7 and published is no and university is top-10, then the instance has class “no”

If GPA is 3.7 and published is no and university is top-20, then the instance has class “yes”
If GPA is 3.7 and published is no and university is top-30, then the instance has class “no”

2.
Yes, it is possible to construct a decision tree from a set of rules. Because each rule can convert to a root to leaf path, so we can identify the root first, and find all internal nodes and grow the tree as long as we have all the rules.

Problem 5

a)
Let a = 1.5
For our h1, if x > 1.5, label +1, if x <= 1.5, label -1
We only misclassify one observation. And satisfy the requirement for breaking ties.

b)

id	x	y	label	prediction	weight
1	1.0	0.5	+1	-1	$\frac{1}{6}$
2	2.2	1.0	+1	+1	$\frac{1}{6}$
3	2.7	2.0	+1	+1	$\frac{1}{6}$
4	0.5	1.5	-1	-1	$\frac{1}{6}$
5	1.2	2.3	-1	-1	$\frac{1}{6}$
6	1.5	2.7	-1	-1	$\frac{1}{6}$

Therefore, $\varepsilon_1 = \frac{1}{6}$

c)
$\varepsilon_1/(1-\varepsilon_1) = 0.2$

id	x	y	label	prediction	weight	Normalized weight
1	1.0	0.5	+1	-1	$\frac{1}{6}$	$\frac{1}{2}$
2	2.2	1.0	+1	+1	$\frac{1}{30}$	$\frac{1}{10}$
3	2.7	2.0	+1	+1	$\frac{1}{30}$	$\frac{1}{10}$
4	0.5	1.5	-1	-1	$\frac{1}{30}$	$\frac{1}{10}$
5	1.2	2.3	-1	-1	$\frac{1}{30}$	$\frac{1}{10}$
6	1.5	2.7	-1	-1	$\frac{1}{30}$	$\frac{1}{10}$

d)

For h2, if y<=1, label +1, if y > 1, label -1

id	x	y	label	prediction	weight
1	1.0	0.5	+1	+1	$\frac{1}{2}$
2	2.2	1.0	+1	+1	$\frac{1}{10}$
3	2.7	2.0	+1	-1	$\frac{1}{10}$
4	0.5	1.5	-1	-1	$\frac{1}{10}$
5	1.2	2.3	-1	-1	$\frac{1}{10}$
6	1.5	2.7	-1	-1	$\frac{1}{10}$

So, $\varepsilon_2 = \frac{1}{10}$

$\varepsilon_2 \ / \ (1- \varepsilon_2 \ ) = \frac{1}{9}$

e)

id	x	y	label	prediction	weight	Normalized weight
1	1.0	0.5	+1	+1	$\frac{1}{18}$	$\frac{5}{18}$
2	2.2	1.0	+1	+1	$\frac{1}{90}$	$\frac{1}{18}$
3	2.7	2.0	+1	-1	$\frac{1}{10}$	$\frac{1}{2}$
4	0.5	1.5	-1	-1	$\frac{1}{90}$	$\frac{1}{18}$
5	1.2	2.3	-1	-1	$\frac{1}{90}$	$\frac{1}{18}$
6	1.5	2.7	-1	-1	$\frac{1}{90}$	$\frac{1}{18}$

For h3, if y <=2, label +1, if y>2, label -1

$\varepsilon_3 = \frac{1}{18}$

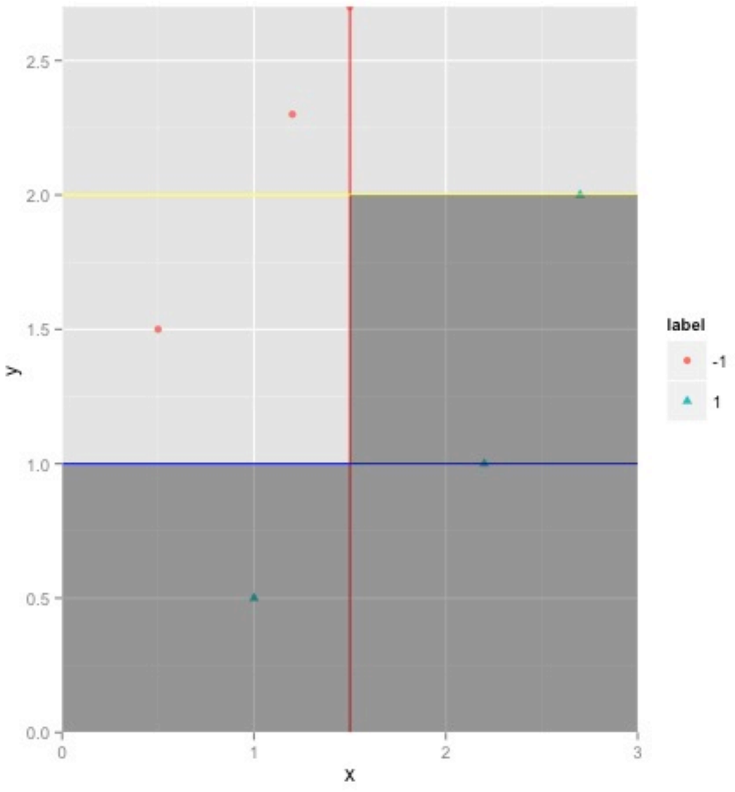
$\varepsilon_3 \ / \ (1- \varepsilon_3 \ ) = 1/17$

id	x	y	label	prediction	weight
1	1.0	0.5	+1	+1	$\frac{5}{18}$
2	2.2	1.0	+1	+1	$\frac{1}{18}$

3	2.7	2.0	+1	+1	$\frac{1}{2}$
4	0.5	1.5	-1	+1	$\frac{1}{18}$
5	1.2	2.3	-1	-1	$\frac{1}{18}$
6	1.5	2.7	-1	-1	$\frac{1}{18}$

f)
a1 = log(5)
a2 = log(9)
a3 = log(17)
For h’, if M1(x)*a1 + M2(x)*a2 + M3(x)*a3 > 0, label +1, otherwise, label -1
Following includes the prediction of h’

id	x	y	label	prediction
1	1.0	0.5	+1	+1
2	2.2	1.0	+1	+1
3	2.7	2.0	+1	+1
4	0.5	1.5	-1	-1
5	1.2	2.3	-1	-1
6	1.5	2.7	-1	-1



Problem 6

a)

P(accepted = “yes”) = 6/12 = 0.5

P(accepted = “no”) = 6/12 = 0.5

b)

P(GPA=4.0 | yes) = 3/6 = 0.5

P(GPA=3.7 | yes) = 3/6 = 0.5

P(GPA=2.5 | yes) = 0

P(univ = top-10 | yes) = 3/6 = 0.5

P(univ = top-20 | yes) = 2/6 = 0.33

P(univ = top-30 | yes) = 1/6 = 0.167

P(published = yes | yes) = 3/6 = 0.5

P(published = no | yes) = 0.5

P(recom = good | yes) = 5/6 = 0.833

P(recom = normal | yes) = 1/6 = 0.167

c)

P(GPA=4.0 | no) = 0

P(GPA=3.7 | no) = 2/6 = 0.33

P(GPA=2.5 | no) = 4/6 = 0.67

P(univ = top-10 | no) = 1/3 = 0.33

P(univ = top-20 | no) = 1/6 = 0.167

P(univ = top-30 | no) = 1/2 = 0.5

P(published = yes | no) = 0.33

P(published = no | no) = 0.67

P(recom = good | no) = 0.5

P(recom = normal | no) = 0.5

d)

For the student with GPA 3.7, top20 univ, published = yes and good recommendation

P(X1 and yes) = P(X1 | yes) * P(yes) = ½ * 1/3 * ½ * 5/6 * ½ = 0.03472222

P(yes | X1) = P(X1 and yes) / P(X1) = 0.03472222/0.04166666 = 0.8823529

P(X1 | yes) = ½ * 1/3 * ½ * 5/6 = 0.06944444

P(X1 | no) = 1/3 * 1/6 * ½ * 1/3 = 0.009259259

For student with GPA 3.7, top30 univ, published = no and normal recommendation

P(X2 and yes) = P(X2 | yes) * P(yes) = ½ * 1/6 * ½ * 1/6 * ½ = 0.00347

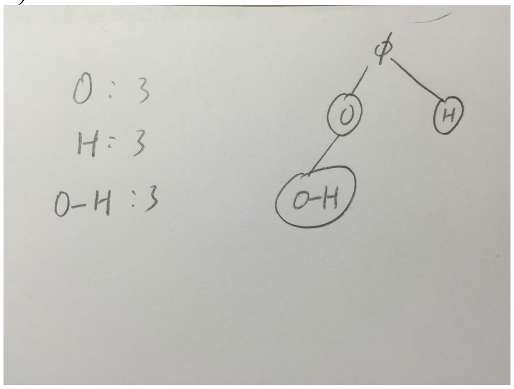
P(yes | X2) = P(X2 and yes) / P(X2) = 0.00347/0.03125 = 0.1111111

P(X2|yes) = ½ * 1/6 * ½ * 1/6 = 0.006944444
P(X2 | no) = 1/3 * ½ * 2/3 * ½ = 0.05555556

e)
The time complexity should be O(nm). For all m attributes, we need to consider all n observations. Therefore it should be O(nm).

f)
Decision Tree
Pro: Interpretability. Easy to interpret
Con: The high classification error rate while training set is small
Naïve Bayes
Pro: Super simple and accurate when the independence assumption hold
Con: The independence assumption. When we draw samples from a population that not fully representative, we could get 0 probability for classification task when we use NB. And this will affect the probability estimate.

Problem 7
a)



b)
O :3
H : 3
C : 2
C-O : 2
C=O: 2
O-H : 3
C-O-H: 2
O=C-O:2
O=C-O-H:2

c)
O-H and O=C-O-H are the closed patterns.
From the three acids, O-H make them to be acids. And for organic acetic acid, it has O=C-O-H but sulfuric does not, so it determines the acidity of organic/inorganic.