CS 412 – Introduction to Data Mining

Assignment 4

Problem 1: Selection of attributes in Decision Tree Classifier

(a)

Information Gain for the Univ attribute is given by:

Gain(Univ) = Info (D) - Info $_{Univ}$ (D)

Now, Info (D) = $-\sum_i p_i \, log(p_i)$

= - [P (Accepted = Yes) *log (P (Accepted = yes)) + P (Accepted = No) *log (P (Accepted = No))]

Consider

P (accepted = Yes) = $\frac{No.\ of\ accepted\ students}{Total\ no.of\ applications} = \frac{\#(Accepted=Yes)}{Total\ Applications} = \frac{6}{12} = 0.5$

P (accepted = No) = $\frac{No.\ of\ not\ accepted\ students}{Total\ no.of\ applications} = \frac{\#(Accepted=No)}{Total\ Applications} = \frac{6}{12} = 0.5$

Hence,

Info (D) = $-\sum_i p_i \, log(p_i)$ = - [0.5 log (0.5) + 0.5log (0.5)] = 1

Next,

To compute: Info $_{Univ}$ (D), consider the following table:

Univ	Accepted = Yes (P_i)	Accepted = No (N_i)	Info (P_i, N_i)
Top 10	3	2	$-(\frac{3}{5}\log\left(\frac{3}{5}\right) + \frac{2}{5} * \log\left(\frac{2}{5}\right)) = 0.9709$
Top 20	2	1	$-(\frac{2}{3}\log\left(\frac{2}{3}\right) + \frac{1}{3} * \log\left(\frac{1}{3}\right)) = 0.918$
Top 30	1	3	$-(\frac{1}{4}\log\left(\frac{1}{4}\right) + \frac{3}{4} * \log\left(\frac{3}{4}\right)) = 0.8112$

Then, **Info $_{Univ}$ (D) = $\sum_{i=1}^{v} \frac{|D_j|}{|D|}$ Info (D_j)**

$= \frac{5}{12} Info(Univ = Top\ 10) + \frac{3}{12} Info(Univ = Top\ 20) + \frac{4}{12} Info(Univ = Top\ 30) =$

$\frac{5}{12} 0.9709 + \frac{3}{12} 0.918 + \frac{4}{12} 0.8112$

$= 0.904515$

Hence, **Gain(Univ) = Info (D) - Info $_{Univ}$ (D) = 1 – 0.904515 = 0.095485**

(b)

Reduction in Impurity for the gini Index for the Published attribute is given by:

ΔGini(Published) = Gini (D) – Gini$_{Published}$ (D)

Now,

Gini (D) = $1 - \sum_i p_i^2 = 1 -$ [P (Accepted= Yes)]^2 - P (Accepted = No)]^2

Consider

P (Accepted = Yes) = $\frac{\#(Accepted=Yes)}{Total\ Applications} = \frac{6}{12}$ = 0.5

P (Accepted = No) = $\frac{\#(Accepted=No)}{Total\ Applications} = \frac{6}{12}$ = 0.5

Hence,

Gini (D) = $1 - \sum_i p_i^2 = 1 -$ [P (Accepted= Yes)]^2 - P (Accepted = No)]^2

= 1 – 0.5^2 – 0.5^2 = **0.5**

Next,

To compute: Gini $_{Published}$ (D), consider the following table:

Published	Accepted = Yes (P_i)	Accepted = No (N_i)	Gini (P_i, N_i)
Yes	3	2	1- $(\frac{3}{5})^2 - (\frac{2}{5})^2 = 0.48$
No	3	4	$1 - (\frac{3}{7})^2 - (\frac{4}{7})^2 = 0.4897$

P (Published = Yes) = $\frac{\#(Published=Yes)}{Total\ Applications} = \frac{5}{12}$ = 0.4166

P (Published = No) = $\frac{\#(Published=No)}{Total\ Applications} = \frac{7}{12} = 0.5833$

Then, Gini $_{\textbf{Published}}$ **(D) =** $\sum_{i=1}^{v} \frac{|D_j|}{|D|}$ **Gini** (D_j)

$= \frac{5}{12} Gini(Published = Yes) + \frac{7}{12} Gini(Published = No) = \frac{5}{12} 0.48 + \frac{7}{12} 0.4897 = 0.48525$

Hence,

Gini Index for published attribute Gini $_{\text{Published}}$ (D) = 0.48525

Reduction in Impurity in Gini Index is given by:

ΔGini(Published) = Gini (D) – Gini$_{\text{Published}}$ (D) = 0.5 - 0.48525 = 0.01475

Problem 2: Naïve Bayes' Algorithm:

(a)

P (accepted = Yes) = $\frac{No.\ of\ accepted\ students}{Total\ no.of\ applications} = \frac{\#(Accepted=Yes)}{Total\ Applications} = \frac{6}{12} = 0.5$

P (accepted = No) = $\frac{No.\ of\ not\ accepted\ students}{Total\ no.of\ applications} = \frac{\#(Accepted=No)}{Total\ Applications} = \frac{6}{12} = 0.5$

(b)

P (GPA = 4.0| accepted = Yes) = $\frac{\#(GPA=4.0\ and\ Accepted=Yes)}{\#(Accepted=Yes)} = \frac{3}{6} = 0.5$

P (GPA = 3.7| accepted = Yes) = $\frac{\#(GPA=3.7\ and\ Accepted=Yes)}{\#(Accepted=Yes)} = \frac{3}{6} = 0.5$

P (GPA = 3.5| accepted = Yes) = $\frac{\#(GPA=3.5\ and\ Accepted=Yes)}{\#(Accepted=Yes)} = \frac{0}{6} = 0$

P (University = Top 10| accepted = Yes) = $\frac{\#(\text{ University = Top 10 } and\ Accepted=Yes)}{\#(Accepted=Yes)} = \frac{3}{6} = 0.5$

$$P(\text{University} = \text{Top 20} \mid \text{accepted} = \text{Yes}) = \frac{\#(\text{University} = \text{Top 20 } and \text{ } Accepted=Yes)}{\#(Accepted=Yes)} = \frac{2}{6} = 0.33$$

$$P(\text{University} = \text{Top 30} \mid \text{accepted} = \text{Yes}) = \frac{\#(\text{University} = \text{Top 30 } and \text{ } Accepted=Yes)}{\#(Accepted=Yes)} = \frac{1}{6} = 0.16$$

$$P(\text{Published} = \text{Yes} \mid \text{accepted} = \text{Yes}) = \frac{\#(\text{ Published} = \text{Yes } and \text{ } Accepted=Yes)}{\#(Accepted=Yes)} = \frac{3}{6} = 0.5$$

$$P(\text{Published} = \text{No} \mid \text{accepted} = \text{Yes}) = \frac{\#(\text{Published} = \text{No } and \text{ } Accepted=Yes)}{\#(Accepted=Yes)} = \frac{3}{6} = 0.5$$

$$P(\text{Recommendation} = \text{Good} \mid \text{accepted} = \text{Yes}) = \frac{\#(\text{ Recommendation} = \text{Good } and \text{ } Accepted=Yes)}{\#(Accepted=Yes)} = \frac{5}{6} = 0.83$$

$$P(\text{Recommendation} = \text{Normal} \mid \text{accepted} = \text{Yes}) = \frac{\#(\text{Recommendation} = \text{Normal } and \text{ } Accepted=Yes)}{\#(Accepted=Yes)} = \frac{1}{6} = 0.16$$

(c)

$$P(\text{GPA} = 4.0 \mid \text{accepted} = \text{No}) = \frac{\#(GPA=4.0 \text{ } and \text{ } Accepted=No)}{\#(Accepted=No)} = \frac{0}{6} = 0$$

$$P(\text{GPA} = 3.7 \mid \text{accepted} = \text{No}) = \frac{\#(GPA=3.7 \text{ } and \text{ } Accepted=No)}{\#(Accepted=No)} = \frac{2}{6} = 0.33$$

$$P(\text{GPA} = 3.5 \mid \text{accepted} = \text{No}) = \frac{\#(GPA=3.5 \text{ } and \text{ } Accepted=No)}{\#(Accepted=No)} = \frac{4}{6} = 0.66$$

$$P(\text{University} = \text{Top 10} \mid \text{accepted} = \text{No}) = \frac{\#(\text{ University} = \text{Top 10 } and \text{ } Accepted=No)}{\#(Accepted=No)} = \frac{2}{6} = 0.33$$

$$P(\text{University} = \text{Top 20} \mid \text{accepted} = \text{No}) = \frac{\#(\text{University} = \text{Top 20 } and \text{ } Accepted=No)}{\#(Accepted=No)} = \frac{1}{6} = 0.66$$

$$P(\text{University} = \text{Top 30} \mid \text{accepted} = \text{No}) = \frac{\#(\text{University} = \text{Top 30 } and \text{ } Accepted=No)}{\#(Accepted=No)} = \frac{3}{6} = 0.5$$

$$P(\text{Published} = \text{Yes} \mid \text{accepted} = \text{No}) = \frac{\#(\text{ Published} = \text{Yes } and \text{ } Accepted=No)}{\#(Accepted=No)} = \frac{2}{6} = 0.33$$

P (Published = No| accepted = No) = $\frac{\#(\text{Published = No } and\ Accepted=No)}{\#(Accepted=No)} = \frac{4}{6} = 0.66$

P (Recommendation = Good| accepted = No) =

$\frac{\#(\text{ Recommendation = Good } and\ Accepted=No)}{\#(Accepted=No)} = \frac{3}{6} = 0.5$

P (Recommendation = Normal | accepted = No) =

$\frac{\#(\text{Recommendation = Normal } and\ Accepted=No)}{\#(Accepted=No)} = \frac{3}{6} = 0.5$

(d)

Consider

P_1 = P (Accepted = Yes| GPA=3.7, university= Top-20, Published=yes, Recommendation=good)

=

P (GPA=3.7, university= Top-20, Published=yes, Recommendation=good| Accepted = Yes) * P (Accepted = Yes)

Since Naïve Bayes' assumes that attributes are independent, we have:

P_1 = P (GPA=3.7| Accepted = Yes) *P (university= Top-20| Accepted = Yes) * P (Published=yes| Accepted = Yes) * P (Recommendation=good| Accepted = Yes) * P (Accepted = Yes)

$= \frac{3}{6} * \frac{2}{6} * \frac{3}{6} * \frac{5}{6} * \frac{1}{2} = 0.03472$

Consider

P_2 = P (Accepted = Yes| GPA=3.7, University=top-30, Publication=no, Recommendation=normal) =

P (GPA=3.7, University=top-30, Publication=no, Recommendation=normal | Accepted = Yes) * P (Accepted = Yes)

Since Naïve Bayes' assumes that attributes are independent, we have:

P_2 = P (GPA=3.7 | Accepted = Yes) * P (University=top-30 | Accepted = Yes) * P (Publication=no| Accepted = Yes) * P (Recommendation=normal | Accepted = Yes) * P (Accepted = Yes)

$$= \frac{3}{6} * \frac{1}{6} * \frac{3}{6} * \frac{1}{6} * \frac{1}{2} = 0.0034722$$

Problem 3: GSP Implementation

Consider the following sequence database D:

Customer Id	Shopping Sequence
1	a(bc)(de)f
2	bc(ad)ef
3	a(bc)d(ab)ef

Minimum Support = 3

To implement GSP, we will be using the Apriori Pruning Principle to reduce the candidate space that if a sequence is infrequent, its' super sequence must be infrequent.

(a)

We scan the database once and count support for all candidates to compute the Length 1 sequential Pattern candidates as:

C_1 :

Candidate	Support
a	3
b	3
c	3
d	3
e	3
f	3

Note that while calculating the support, even if the same customer bought an item more than once, it adds only one to the candidates' support since membership is a Boolean value.

We count the numbers of customers having bought a particular item as the support of the item.

Since, Min support = 3, no candidate gets pruned and hence list L_1 is as follows:

L_1 :

Candidate	Support
a	3
b	3
c	3
d	3
e	3
f	3

Hence, we get 6 candidates in L_1.

(b)

Next, to compute Length 2 sequential Pattern candidates, we combine candidates in L_1 to attain C_2:

Note that since order matters, we consider [ab] and [ba] as separate candidates and since a customer may buy the same item twice, we consider candidates like [aa] as well.

Also, a customer may buy some items in the same visit, we consider candidates of form [(ab)] as well where 'a' and 'b' were bought in the same visit.

Hence, we get 6*6 + $^6C_2$ = 51 candidates in C_2.

C_2 :

Candidate	Support
aa	1
ab	2
ac	2
ad	2
ae	3
af	3
ba	2
bb	1
bc	1
bd	3
be	3
bf	3
ca	2

cb	1
cc	0
cd	3
ce	3
cf	3
da	1
db	1
dc	0
dd	0
de	2
df	3
ea	0
eb	0
ec	0
ed	0
ee	0
ef	3
fa	0
fb	0
fc	0
fd	0
fe	0
ff	0
(ab)	1
(ac)	0
(ad)	1
(ae)	0
(af)	0
(bc)	2
(bd)	0
(be)	0
(bf)	0
(cd)	0
(ce)	0
(cf)	0
(de)	1
(df)	0
(ef)	0

According to the Apriori pruning principle, since say [ab] is infrequent hence any super sequence with [ab] is infrequent as well. Hence, we may simply delete it from the set of patterns we need to consider.

Since, Min support = 3, candidate gets pruned and after pruning list L_2 is as follows:
L_2 :

Candidate	Support
ae	3
af	3
bd	3
be	3
bf	3
cd	3
ce	3
cf	3
df	3
ef	3

(c)

Next, to compute Length 3 sequential Pattern candidates, we combine candidates in L_2 to attain C_3:

Now, we trim the head (1st element) and tail (last element) of the remaining candidates in L_2 and merge the ones which are same after the head prune of the first candidate and tail prune of the second candidate.

Consider [ae] and [ef], we merge them to form [aef]. Hence, we get C_3 as follows:

Candidate	Trim head	Trim tail
ae	e	a
af	f	a
bd	d	b
be	e	b
bf	f	b
cd	d	c
ce	e	c
cf	f	c
df	f	d
ef	f	e

C_3:

Candidate	Support
aef	3
bdf	3
bef	3
cdf	3
cef	3

Since, Min support = 3, no candidate gets pruned and hence list L_3 is as follows:

L_3 :

Candidate	Support
aef	3
bdf	3
bef	3
cdf	3
cef	3

Next, to compute Length 4 sequential Pattern candidates, we combine candidates in L_3 to attain C_4:
Now, we trim the head (1st element) and tail (last element) of the remaining candidates in L_3 and merge the ones which are same after the head prune of the first candidate and tail prune of the second candidate.

Consider [ae] and [ef], we merge them to form [aef]. Hence, we get C_3 as follows:

Candidate	Trim head	Trim tail
aef	ef	ae
bdf	df	bd
bef	ef	be
cdf	df	cd
cef	ef	ce

Since none of the trimmed sequence sets after head trim and tail trim match, we cannot join any length 3 sequences to form length 4 sequences.

Hence, the algorithm terminates here.

Hence, we have the following results:

C_1	L_1	C_2	L_2	C_3	L_3	C_4	L_4
a	a: 3	aa	ae:3	aef	aef: 3	None	None
b	b: 3	ab	af:3	bdf	bdf: 3		
c	c: 3	ac	bd:3	bef	bef: 3		
d	d: 3	ad	be:3	cdf	cdf: 3		
e	e: 3	ae	bf:3	cef	cef: 3		
f	f: 3	af	cd:3				
		ba	ce:3				
		bb	cf:3				
		bc	df:3				
		bd	ef:3				
		be					
		bf					
		ca					
		cb					
		cc					
		cd					
		ce					
		cf					
		da					
		db					
		dc					
		dd					
		de					
		df					
		ea					
		eb					
		ec					
		ed					
		ee					
		ef					
		fa					
		fb					
		fc					
		fd					
		fe					
		ff					
		(ab)					
		(ac)					
		(ad)					
		(ae)					

		(af)					
		(bc)					
		(bd)					
		(be)					
		(bf)					
		(cd)					
		(ce)					
		(cf)					
		(de)					
		(df)					
		(ef)					