PREPARED BY: RAJARAJESWARI VAIDYANATHAN **CWID**: A20362220 **COURSE**: CS422 – DATA MINING

PROBLEMS AND SOLUTIONS FROM CHAPTERS 2, 3, 4

Chapter 2 Problems:

Hamming distance:

- Hamming distance is the number used to represent the distance between two binary strings of equal length.

For example: String 1: 10101010 String 2: 10011001

- Compare the first 2 bits in both the strings. If they are same then record '0'. If not, then record '1'.
- Record: 00110011
- Now adding all the '1's in a record together gives the Hamming distance,
- Hamming distance = 0+0+1+1+0+0+1+1 = 4.

Jaccard co-efficient:

- If in a store, each asymmetric binary attribute corresponds to an item in a store, then '1' indicates that the item was purchased and '0' indicates that the item was not purchased.
- Since the number of products not purchased by any customer always outcomes/overcomes the products that are purchased, a similarity measure will indicate that all transactions are similar.
- Hence Jaccard co-efficient is frequently used to handle objects consisting of asymmetric binary attributes.

Problem 18:

a.) Compute the Hamming distance and the Jaccard similarity between the following two binary vectors.

Solution:

x = 0101010001
y = 0100011000

- Hence to calculate the Hamming distance between x and y binary vectors, compute the record value of both.
- Record: 0001001001 (comparing bits in x and y respectively and record 0 if they are same else, record 1).
- Now adding all the bits in the record string.
- Hamming distance = 0 + 0 + 0+ 1 + 0+ 0+ 1 + 0 + 0+ 1 = 3.

Answer: Hamming distance between x and y binary vectors of same length = 3

Jaccard Co-efficient for x and y attributes.

x = 0101010001

y = 0100011000

- f_{00} = 5 bits (5 bits corresponds to '00' binary bits from x and y),
- f_{01} = 1 bit corresponds to '01' binary bits from x and y
- f_{10} = 2 bits corresponds to '10' binary bits from x and y
- f_{11} = 2 bits corresponds to '11' binary bits from x and y

Hence Jaccard co-efficient is calculated in the below formula.

$$J = \frac{\text{Number of matching presences}}{\text{Number of attributes not involved in 00 matches'}} = \frac{f11}{f\,01 + f\,10 + f11}$$

Hence substituting the values of f11, f10, f01 in the above formula.

$$J = \frac{2}{2 + 2 + 1} = \frac{2}{5} = 0.4$$

Answer: Jaccard co-efficient of x and y asymmetric binary attributes = 0.4

Problem 18:

c.) Suppose that you are comparing how similar two organisms of different species are in terms of the number of genes they share. Describe which measure, Hamming or Jaccard, you think would be more appropriate for comparing the genetic makeup of two organisms. Explain. (Assume that each animal is represented as a binary vector, where each attribute is 1 if a particular gene is present in the organism and 0 otherwise.)

Solution:

2 types of comparisons arise.

- When comparing two organisms of the same species then the vital genes to compare in them are the "alike" genes.
- When comparing genes in same species then the vital genes to compare in them are the ones that are ONLY different.

For our problem, we would be requiring to calculate the alike genes in two organisms for comparing the genetic makeup.

Consider an example below when comparing 2 organisms represented by a binary vector with an attribute '1' indicating that the gene is present in the species and "0" indicating that the gene is not present.

PREPARED BY: RAJARAJESWARI VAIDYANATHAN **CWID**: A20362220 **COURSE**: CS422 – DATA MINING

Let us take the x and y vectors.

x = 01010100011 and y = 01000110001

Observations:

- There are about 11 genes to be compared between 2 organisms.
- There are 3 genes that are alike of the whole 11 genes (1-1 matches which are shown in "red") which are the only genes considered to be important in our case.
- Genes that are not present in either organism are considered that both organisms (x and y organisms in our case) are different.
- Hence Jaccard is more suitable for comparing the genes of 2 organisms as we would want to know how many genes that both the organisms share -> alike.

Answer: This approach of handling genes that consists of asymmetric binary attributes are calculated by Jaccard method and this approach of comparison is known as Jaccard co-efficient.

Problem 18:

d.) If you wanted to compare the genetic makeup of two organisms of the same species, e.g., two human beings, would you use the Hamming distance, the Jaccard coefficient, or a different measure of similarity or distance? Explain. (Note that two human beings share > 99.9% of the same genes.)

Solution:

- When genetic makeup of humans are considered, we would be most importantly considering the genes that are different in those species.
- Thus Hamming distance would fit in more appropriately here to identify the genes that are different in the total number of genes present in humans.

For example let's consider x and y human vectors:

x = 00100001010101010001 and y = 00110001010101010011

- From the above vectors, say '0' corresponds to the genes not present and '1' corresponds to the genes that are present.
- From the above 20 genes (from x human vector), we could arrive at a conclusion that only 2 genes are different resulting in 18 genes which are the same in both the humans.
- Hence Hamming distance is calculated = 2 bits (that are different).

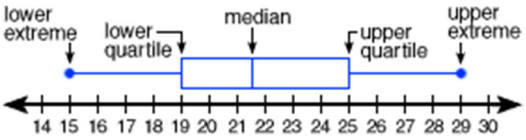
Answer: Thus Hamming distance would be used to measure the difference or similarity

PREPARED BY: RAJARAJESWARI VAIDYANATHAN CWID: A20362220 COURSE: CS422 – DATA MINING

Chapter 3 problems:

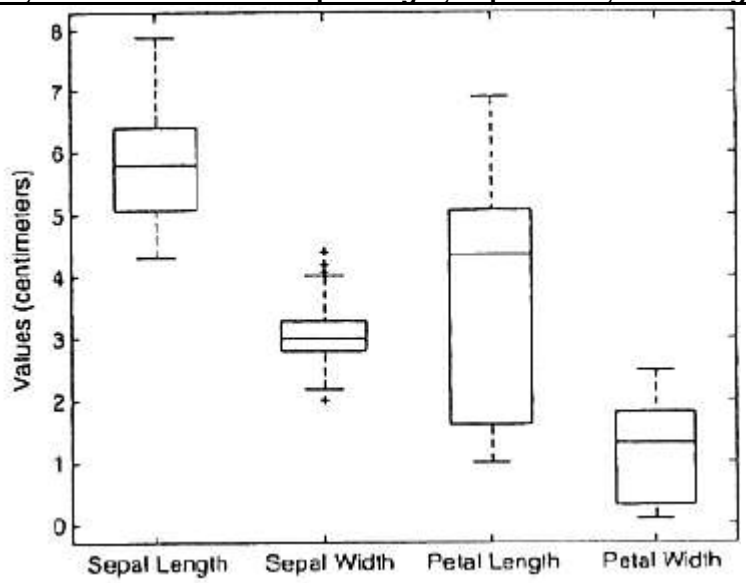
BOX PLOTS:

The below diagram represents the box plot for any attribute taken.



Problem 8:

In fig 3.11 -> IRIS dataset, and its attributes – Sepal length, Sepal width, Petal length and Petal width.



Here, for Sepal Length attribute, we can calculate the median, lower quartile, upper quartile.

Let's consider the values in cm with respect to the above box plots for Sepal Length.

X = {5, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6.0, 6.1, 6.2, 6.3, 6.4}

Calculating the median = (15 terms + 1) / 2

- Median = 16^{th} term / 2
  = 8^{th} term

Median = 5.7 (8^{th} term in the X set).

Calculating the Lower quartile = Number of elements below the median – 7 elements below the median

- X on lower quartile = {5.0, 5.1, 5.2, 5.3, 5.4, 5.5, 5.5}
- Lower quartile = (7 terms + 1) / 2
  = 8^{th} term / 2
  = 4^{th} term

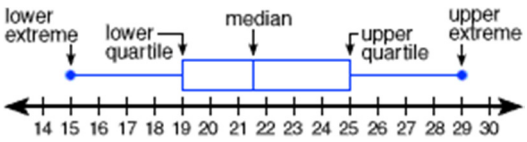
Lower quartile = 5.3

Calculating the Upper quartile = Number of elements above the median – 7 elements above the median

- X on Upper quartile = {5.8, 5.9, 6.0, 6.1, 6.2, 6.3, 6.4}
- Upper quartile = (7 terms + 1) / 2
  = 8^{th} term / 2
  = 4^{th} term

Upper quartile = 6.1

Hence the box plot is arrived based on the above calculations.



X = {lower extreme, lower quartile, median, upper quartile, upper extreme}

X = {5.0, 5.3, 5.7, 6.1, 6.4}.

With the help of the above, we can identify and differentiate if the values of an attribute is symmetrically distributed or not.

a) Length of whiskers and outliers is an indication to some points but they do not involve many points in a set and hence are ignored as they will mislead us in analyzing the data.

b) Hence, if a line is representing the median of the data, then the data is symmetrically distributed if 75% of the data lies between the first and third quartiles.

c) In our case, for Sepal Length they are symmetrically distributed as 75% of the data lies between the first and third quartile {5.0 to 5.7 -> 75% of the values pointed from the whole set of values 5.0 to 6.4}

d) The same is applicable for Sepal width as they are too symmetrically distributed (75% of the values lies between 1^{st} and 2^{nd} quartile)

e) Also, from the box plot for IRIS dataset, we could conclude that Petal length are not symmetrically distributed and are skewed as the percentage of data split is being scattered and represents some of the class attributes itself.

Answer:

From the fig 3.11 (box plot for IRIS dataset) it is shown that 75% of the data lies between the first and third quartile for Sepal length and width which indicates that they are symmetrically distributed.

Also, Petal length seems to be skewed (not symmetrically distributed) and so as the Petal width.

This box plot is used to calculate the sales between 2 companies with respect to their high and low values in a set by calculating the median, lower quartile, higher quartile, figuring it and analyzing the ups and downs.

PREPARED BY: RAJARAJESWARI VAIDYANATHAN **CWID**: A20362220 **COURSE**: CS422 – DATA MINING

CHAPTER 4 PROBLEMS:

INFORMATION GAIN, ENTROPY AND GINI INDEX:

Entropy:

- Entropy is defined as the average of the information contained in each set.
- This will be used to decide the split based on a particular attribute.
- If any particular attribute's entropy is high, then the decision tree algorithm makes use of this attribute to decide its branches (children).

Entropy is given by the formula = - { $\sum p \log_2 p$ }, where P is the proportion of the positive and negative samples.

Information gain:

- When entropy is used in impurity measure, the difference in the entropy is known as Information gain-> info.
- This is based on the decrease in entropy after a dataset is split on an attribute. A decision tree is constructed based on the attribute that returns the highest information gain.

Information gain is given by the formula = $\sum p \log_2 p$ Entropy original gained before split – {$\sum$(Pi)/P * E(P)} → entropy after the split for positive and negative samples multiplied by the probability of their occurrences)

GINI index:

- Used to calculate impurity measure instead of entropy

Gini index is given by the formula = 1 - $\sum p^2$

Problem 5:

a.) **Calculate the information gain when splitting on A and B. Which attribute will the decision tree algorithm choose to make decision?**

A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

Solution:

Contingency table after splitting on Attribute A:

When	A = T	A = F
+	4	0
-	3	3

When A = T, there are 4 positive values and 3 negative values. Similarly when A = F there are '0' positive values and '3' negative values.

Overall entropy is calculated before the split as shown below.

Thus Eorig = - {4/10 log 4/10 + 6/10 log 6/10}

= - {0.4 log 0.4 + 0.6 log 0.6}

= -0.4 (-1.321) – 0.6 (-0.736)

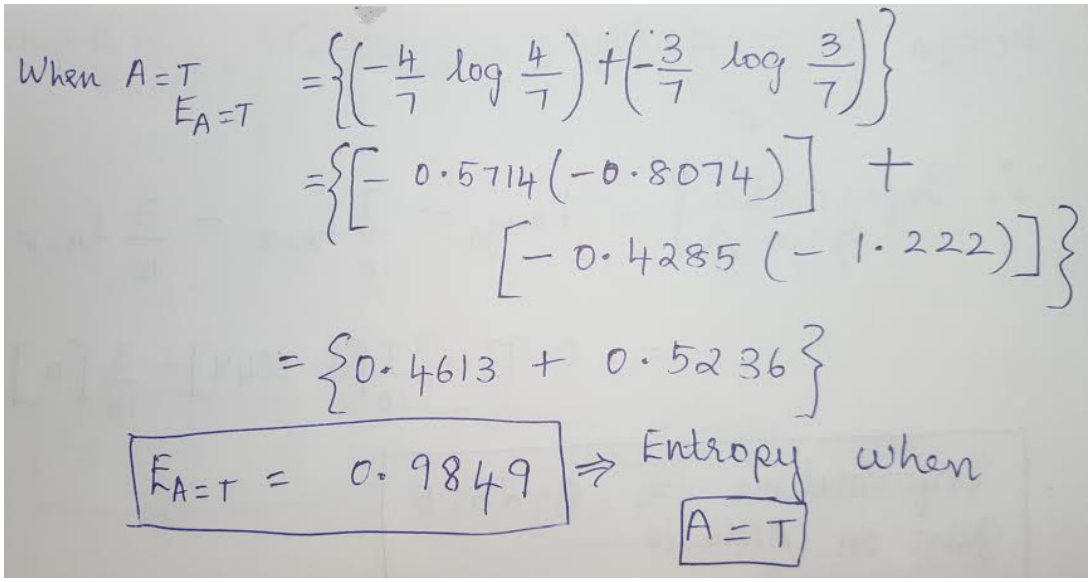
= 0.5284 + 0.4416

Eorig	**= 0.97**

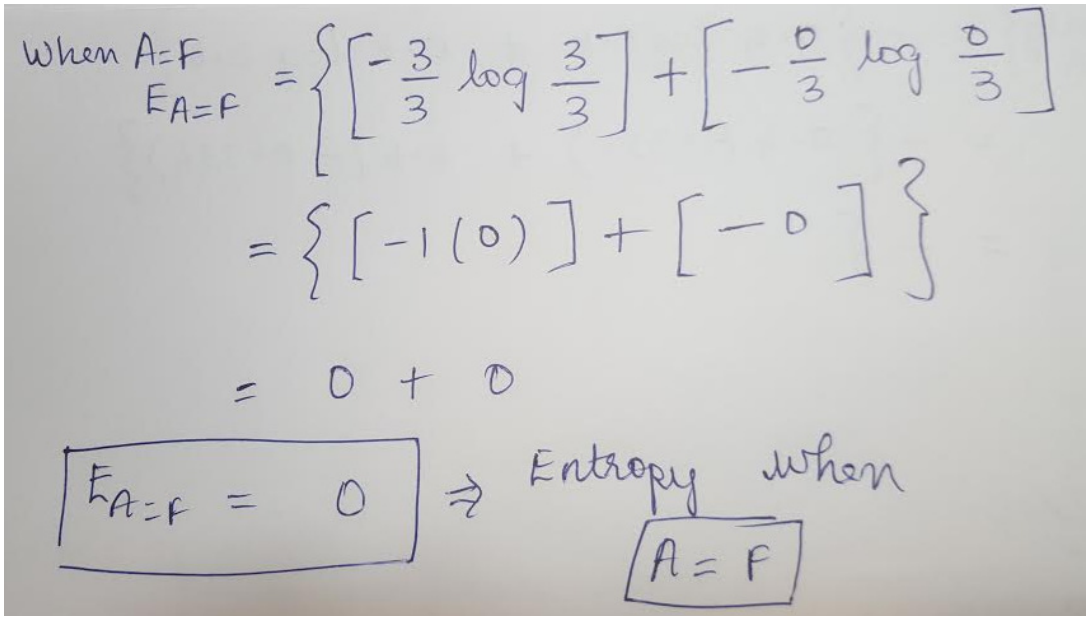
Where, 4 positive values and 6 negative values on the whole.

Now, the information gain after splitting on Attribute A.

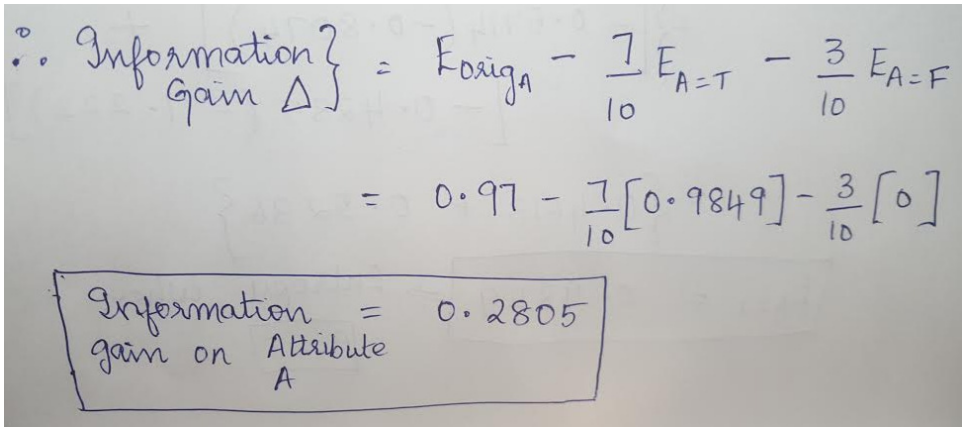
To achieve this, Entropy is calculated when A= T and when A = F as shown below.



PREPARED BY: RAJARAJESWARI VAIDYANATHAN CWID: A20362220 COURSE: CS422 – DATA MINING



Information gain calculated on Attribute A.



Similarly we calculate the information gain on Attribute B.

Contingency table after splitting on Attribute B:

When	B = T	B = F
+	3	1
-	1	5

When B = T, there are 3 positive values and 1 negative value. Similarly when B = F there are '1' positive value and '5' negative values.

Original Value of the entropy is calculated before the split on B

Thus Eorig = - {4/10 log 4/10 + 6/10 log 6/10}

= - {0.4 log 0.4 + 0.6 log 0.6}

= -0.4 (-1.321) – 0.6 (-0.736)

= 0.5284 + 0.4416

Eorig = 0.97

Entropy calculations on the split when B = T and when B = F

When B = T,

$E_{B=T}$ = - {3/4 log 3/4 + 1/4 log 1/4}

= - {0.75 log 0.75 + 0.25 log 0.25}

= - 0.75 (-0.415) – 0.25 (-2)

= 0.3112 + 0.5

E when $_{B=T}$ = 0.8112

When B = F,

$E_{B=F}$ = - {1/6 log 1/6 + 5/6 log 5/6}

= - {0.167 log (0.167) + 0.834 log (0.834)}

= - 0.167 (-2.582) – 0.834 (-0.2618)

E when $_{B=F}$ = 0.6501

Information gain:

Information gain = Eorig on B - (4/10 * E when $_{B=T}$) - (6/10 * E when $_{B=F}$)

= 0.97 – 4/10 (0.8112) – 6/10 (0.6501)

= 0.97 – 0.3244 – 0.3900

= 0.256

Answer:

Thus information gain on Attribute A = 0.2805 and Information gain on Attribute B = 0.256

Hence the decision tree will split the tree based on Attribute "A" as the information gain is higher.

Problem 5:

b.) Calculate the gain in the Gini index when splitting on *A* and *B*. Which attribute would the decision tree induction algorithm choose?

Solution:

The Gain in GINI before splitting is

$G_{orig} = 1 - \sum p^2$
$= 1 – (4/10)^2 – (6/10)^2$
$= 1 – (0.4)^2 – (0.6)^2$
$= 0.48$

Gain obtained via GINI before the split $G_{orig} = 0.48$

The gain in gini after splitting on A when A = T is

$G_{A=T} = 1 - \sum p^2$
$= 1 – (4/7)^2 – (3/7)^2$
$= 1 – (0.5714)^2 – (0.4285)^2$
$= 0.4898$

Gain obtained when $G_{A=T} = 0.4898$

The gain in gini after splitting on A when A = F is

$G_{A=F} = 1 - \sum p^2$
$= 1 – (3/3)^2 – (0/3)^2$
$= 1 – (1)^2 – (0)^2$
$= 0$

Gain obtained when $G_{A=F} = 0$

PREPARED BY: RAJARAJESWARI VAIDYANATHAN CWID: A20362220 COURSE: CS422 – DATA MINING

Information gain obtained with the help of both Entropy.

Information gain = G_{orig} - (7/10) $G_{A=T}$ – (3/10)$G_{A=F}$

= 0.4898 – (7/10) 0.48 – (3/10) 0

= 0.1538

Information gain obtained on splitting attribute A = 0.1538

The gain in gini after splitting on B when B = T is

$G_{B=T} = 1 - \sum p^2$
$= 1 - (1/4)^2 - (3/4)^2$
$= 1 - (0.25)^2 - (0.75)^2$
$= 0.3750$

Gain obtained when $G_{B=T} = 0.3750$

The gain in gini after splitting on B when B = F is

$G_{B=F} = 1 - \sum p^2$
$= 1 - (1/6)^2 - (5/6)^2$
$= 1 - (0.166)^2 - (0.833)^2$
$= 0.2778$

Gain obtained when $G_{B=F} = 0.2778$

Information gain obtained with the help of both Entropy.

Information gain = G_{orig} - (4/10) $G_{B=T}$ – (6/10)$G_{B=F}$

= 0.4898 – (4/10) 0.3750 – (3/10) 0.2778

= 0.1663

Information gain obtained on splitting attribute B = 0.1663

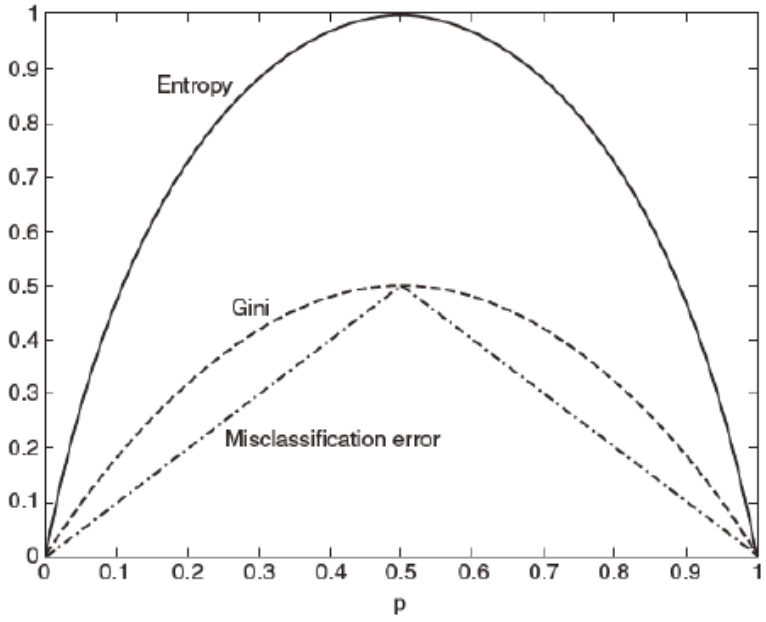
Answer: From the above calculations, we can conclude that attribute B will be chosen to do the split on the tree to arrive at the branches as the information gain obtained on Attribute B is higher than that of Attribute A.

PREPARED BY: RAJARAJESWARI VAIDYANATHAN CWID: A20362220 COURSE: CS422 – DATA MINING

Problem 5:

c.) Figure 4.13 shows that entropy and the Gini index are both monotonously increasing on the range [0, 0.5] and they are both monotonously decreasing on the range [0.5, 1]. Is it possible that information gain and the gain in the Gini index favor different attributes? Explain.

Solution:



In the above fig, even though the measures have similar range and monotonous behavior, their respective gains which are the scaled differences of the measures, do not necessarily behave in the same way.

- For example, the gain is calculated on the correct instances reported for the dataset.
- In our case, let's take IRIS dataset where the attribute Petal length has all correct instances for the class Setosa and hence entropy would be highly dependent on this particular attribute.
- Say for example, when I change some records and make it to be reported as incorrect instances, then the resulting decision tree will again calculate entropy and see the Petal Length has more incorrect instances and the entropy has become poor.
- But this time, the entropy of Petal width attribute is more and hence the decision tree will take the Petal width as the attribute and split its tree accordingly.

Though entropy and Gini shows the measures of the angles to be the same and both decreases at the same fraction of records, the information gained on Entropy and Gini for the same fraction of records differ in values as we derived from a.) and b.) in above calculations.

Information gain is biased towards choosing attributes with a large number of values.

PREPARED BY: RAJARAJESWARI VAIDYANATHAN CWID: A20362220 COURSE: CS422 – DATA MINING

CHAPTER 4:

Problem 7:

The following table summarizes a data set with three attributes *A*, *B*, *C* and two class labels +, –. Build a two-level decision tree.

A	B	C	Number of Instances	
			+	–
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

a.) According to the classification error rate, which attribute would be chosen as the first splitting attribute? For each attribute, show the contingency table and the gains in classification error rate.

Solution:

Calculating the original error rate for the whole data without doing any partition on any attribute.

E orig = 1 – max (50/100, 50/100)
= 1 – (50/100)
= 50/100
= 0.5

Gain obtained when Eorig = 0.5

After splitting on Attribute A, the gain in error rate is

When	A = T	A = F
+	25	25
-	0	50

When A = T, error rate = 1 – max (25/25, 0/25)
= 1 – {25/25)}
= 1 – {1}
= 0

Gain obtained when $E_{A=T} = 0$

When A = F, error rate = 1 – max (25/75, 50/75)
= 1 – {50/75}
= 1/3
= 0.33

Gain obtained when $E_{A=F}$ = 0.33

Gain in error rate thus calculated

Gain in error rate = Eorg – 25/100 * $E_{A=T}$ – 75/100 * $E_{A=F}$
= 0.5 – 1/4 * 0 – 3/4 * 0.33
= 0.25

Gain in error rate is obtained on attribute A = 0.25

After splitting on Attribute B, the gain in error rate is

When	B = T	B = F
+	30	20
-	20	30

When B = T, error rate = 1 – max (30/50, 20/30)
= 1 – {30/50}
= 2/5
= 0.4

Gain obtained when $E_{B=T}$ = 0.4

When B = F, error rate = 1 – max (20/50, 30/50)
= 1 – {30/50}
= 2/5
= 0.4

Gain obtained when $E_{B=F}$ = 0.4

Gain in error rate thus calculated

Gain in error rate = Eorg – 50/100 * $E_{B=T}$ – 50/100 * $E_{B=F}$
= 0.5 – 1/2 * 0.4 – 1/2 * 0.4
= 0.1

Gain in error rate is obtained on attribute B = 0.1

After splitting on Attribute C, the gain in error rate is

When	C = T	C = F
+	25	25
-	25	25

When C = T, error rate = 1 – max (25/50, 25/50)
= 1 – {25/50}
= 1 – {0.5}
= 0.5

Gain obtained when $E_{C=T}$ = 0.5

When C = F, error rate = 1 – max (20/50, 25/50)
= 1 – {25/50}
= 1 – {0.5}
= 0.5

Gain obtained when $E_{C=F}$ = 0.5

Gain in error rate thus calculated

Gain in error rate = Eorg – 50/100 * $E_{C=T}$ – 50/100 * $E_{C=F}$
= 0.5 – 1/2 * 0.5 – 1/2 * 0.5
= 0

Gain in error rate is obtained on attribute C = 0

Answer:

Gain in error rate calculated in attribute A = 0.25

Gain in error rate calculated in attribute B = 0.1

Gain in error rate calculated in attribute C = 0

The algorithm thus chooses attribute "A" because it has the highest gain.

PREPARED BY: RAJARAJESWARI VAIDYANATHAN CWID: A20362220 COURSE: CS422 – DATA MINING

Problem 7:

b.) Repeat for the two children of the root node.

Solution:

When A = T, the child node is pure meaning there are no incorrect instances reported when A = T as shown in the below column (indicated by color).

When	A = T	A = F
+	25	25
-	0	50

Total number of attributes in a dataset:

A	B	C	Number of Instances	
			+	−
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

Hence for A = F, child node the distribution of the training instances becomes as shown below.

B	C	Class label when "+"	Class label when "-"
T	T	0	20
F	T	0	5
T	F	25	0
F	F	0	25

Now the classification error of the child A = F, (child node becomes)

When A = F, error rate = 1 – max (25/75, 50/75)
= 1 – {50/75}
= 1 – {2/3}
= 0.33

Gain obtained when $E_{A=F\ (original\ obtained)} = 0.33$

Now splitting on attribute B:

When	B = T	B=F
+	25	0
-	20	30

When B = T, error rate = 1 – max (25/45, 20/45)
= 1 – {25/45}
= 20/45
= 0.44

Gain obtained when $E_{B=T}$ = 0.44

When B = F, error rate = 1 – max (0/30, 30/30)
= 1 – {30/30}
= 1 – (1)
= 0

Gain obtained when $E_{B=F}$ = 0

Gain in error rate thus calculated

Gain in error rate = Eorg – 45/75 * $E_{B=T}$ – 30/75 * $E_{B=F}$
= 0.33 – 0.6 * 0.44 – 0.4 * 0
= 0.0667

Gain in error rate is obtained on attribute B = 0.0667

After splitting on Attribute C, the gain in error rate is

When	C = T	C = F
+	0	25
-	25	25

When C = T, error rate = 1 – max (0/25, 25/25)
= 1 – {25/25}
= 1 – {1}
= 0

Gain obtained when $E_{C=T}$ = 0

When C = F, error rate = 1 – max (25/50, 25/50)
= 1 – {25/50}
= 1 – {1/2}
= 0.5

Gain obtained when $E_{C=F}$ = 0.5

Gain in error rate thus calculated

Gain in error rate = Eorg – 25/75 * $E_{C=T}$ – 50/75 * $E_{C=F}$
= 0.33 – 1/3 * 0 – 2/3 * 0.5
= 0.33 – 0.33
= 0

Gain in error rate is obtained on attribute C = 0

Answer:

Gain in error rate calculated in attribute B = 0.0667

Gain in error rate calculated in attribute C = 0

The algorithm thus chooses attribute "B" because it has the highest gain compared to the attribute "C"

Problem 7:

c.) How many instances are misclassified on the resulting decision tree?

Solution:

From the above problem, we know that the algorithm chooses attribute B because it has the highest gain compared to C.

Hence when looking at the attribute B as shown below we could see that there are 20 instances which were misclassified totally (marked in color).

When	B = T	B=F
+	25	0
-	**20**	30

Answer: These 20 instances are misclassified when B = T (as the class that is being reported) when actually the predicted class for these 20 instances should be when B = F.

Hence the error rate on the whole for B (resulting decision tree attribute) = 20/100 = 0.2

PREPARED BY: RAJARAJESWARI VAIDYANATHAN CWID: A20362220 COURSE: CS422 – DATA MINING

Problem 7:

d.) Repeat parts (a), (b) and (c) using C as the split attribute.

Solution:

When C = T child node, the error rate before splitting is shown as below

After splitting on Attribute C, the gain in error rate is

When	C = T	C = F
+	25	25
-	25	25

When C = T, error rate = 1 – max (25/50, 25/50)
= 1 – {25/50}
= 1 – {1/2}
= 0.5

Gain obtained when $E_{C=T} = 0.5$

Training set:

A	B	C	Number of Instances	
			+	–
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

Now after splitting on Attribute A from the training set when C = T, the gain in error rate becomes

When	A = T	A = F
+	25	0
-	0	25

When A = T, error rate = 1 – max (25/25, 0/25)
= 1 – {25/25}
= 1 – {1}
= 0

Gain obtained when $E_{A=T} = 0$

When A = F, error rate = 1 – max (0/25, 25/25)
= 1 – {25/25}
= 1 – {1}
= 0

Gain obtained when $E_{A=F} = 0$

Gain in error rate thus calculated

Gain in error rate = Eorg – 25/50 * $E_{A=T}$ – 25/50 * $E_{A=F}$
= 0.5 – 1/2 * 0 – 1/2 * 0
= 0.5

Gain in error rate is obtained on attribute A = 0.5

Now, when splitting on Attribute B

When	B = T	B = F
+	5	20
-	20	5

When B = T, error rate = 1 – max (5/25, 20/25)
= 1 – {20/25}
= 5/25
= 1/5
= 0.2

Gain in error obtained when $E_{B=T} = 0.2$

When B = F, error rate = 1 – max (20/25, 5/25)
= 1 – {20/25}
= 5/25
= 1/5
= 0.2

Gain in error obtained when $E_{B=F} = 0.2$

Gain in error rate thus calculated

Gain in error rate = Eorg – 25/50 * $E_{B=T}$ – 25/50 * $E_{B=F}$
= 0.5 – 1/2 * 0.2 – 1/2 * 0.2
= 0.3

Gain in error rate is obtained on attribute B = 0.3

Thus when C = T,

Error gain obtained on attribute A = 0.5

Error gain obtained on attribute B = 0.3

Thus the algorithm chooses Attribute A to do the split when C = T.

When C = F, then the error rate again calculated before splitting

When	C = T	C = F
+	25	25
-	25	25

When C = F, error rate = 1 – max (25/50, 25/50)
= 1 – {25/50}
= 1 – {1/2}
= 0.5

Gain obtained when $E_{C=F}$ = 0.5

Training set:

A	B	C	Number of Instances	
			+	–
T	T	T	5	0
F	T	T	0	20
T	F	T	20	0
F	F	T	0	5
T	T	F	0	0
F	T	F	25	0
T	F	F	0	0
F	F	F	0	25

Now after splitting on Attribute A from the training set when C = F, the gain in error rate becomes

When	A = T	A = F
+	0	25
-	0	25

When A = T, error rate = 1 – max (0/0, 0/0)
= 1 – {0/0}
= 0

Gain obtained when $E_{A=T}$ = 0

When A = F, error rate = 1 – max (25/50, 25/50)
= 1 – {25/50}
= 1 – {1/2}
= 0.5

Gain obtained when $E_{A=F}$ = 0.5

Gain in error rate thus calculated

Gain in error rate = Eorg – 0/50 * $E_{A=T}$ – 50/50 * $E_{A=F}$
= 0.5 – 0 * 0 – 1 * 0.5
= 0

Gain in error rate is obtained on attribute A = 0

Now, when splitting on Attribute B

When	B = T	B = F
+	25	0
-	0	25

When B = T, error rate = 1 – max (25/25, 0/25)
= 1 – {25/25}
= 0

Gain in error obtained when $E_{B=T}$ = 0

When B = F, error rate = 1 – max (0/25, 25/25)
= 1 – {25/25}
= 1 – 1
= 0

Gain in error obtained when $E_{B=F}$ = 0

Gain in error rate thus calculated

Gain in error rate = Eorg – 25/50 * $E_{B=T}$ – 25/50 * $E_{B=F}$
= 0.5 – 1/2 * 0 – 1/2 * 0
= 0.5

Gain in error rate is obtained on attribute B = 0.5

Thus when C = F,

Error gain obtained on attribute A = 0

Error gain obtained on attribute B = 0.5

When C = F, the algorithm chooses Attribute B as it has higher gain = 0.5

When C = T, the algorithm chooses Attribute A as it has higher gain = 0.5

Answer: Therefore, the overall error rate when C = T and C = F, is ZERO.

Problem 7:

e.) Use the results in Parts c.) and d.) to conclude about the greedy nature of the decision tree induction algorithm.

Solution:

We know that there are 20 instances misclassified in B when A = F (child node), when A = T, there is no split as shown below (in color)

When	B = T	B=F
+	25	0
-	**20**	30

Also, from d.) we know that there are "40" instances misclassified when C = T and "0" instances misclassified when C = F.

When C = T, there are 40 misclassified instances totally

When	B = T	B = F
+	5	**20**
-	**20**	5

When C = F, there are '0' misclassified instances totally.

When	B = T	B = F
+	25	**0**
-	**0**	25

Answer: Thus the greedy heuristic does not always lead to the best tree on which we can depend upon for the split.