IIT ROORKEE

NPTEL ONLINE
CERTIFICATION COURSE

Attribute selection Measures in CART : II

Dr. A. Ramesh
DEPARTMENT OF MANAGEMENT STUDIES

Agenda

- Attribute selection measures:
  - Gain Value
  - Gain ratio
  - Gini Index

Gain Ratio

- The information gain measure is biased toward tests with many outcomes
- That is, it prefers to select attributes having a large number of values
- For example, consider an attribute that acts as a unique identifier, such as product ID.
- A split on product ID would result in a large number of partitions (as many as there are values), each one containing just one tuple

Gain Ratio

- Because each partition is pure, the information required to classify dataset D based on this partitioning would be $Info_{\text{product ID}}(D) = 0$
- Information Gain = Info D-$(Info_{\text{product ID}}(D))$ = maximum
- Therefore, the information gained by partitioning on this attribute is maximal
- Clearly, such a partitioning is useless for classification
- Gain ratio is an extension to information gain which attempts to overcome this bias

Split information

- It applies a kind of normalization to information gain using a "split information" value defined analogously with Info(D) as:

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

- D_j= single partion
- D = Data set
- This value represents the potential information generated by splitting the training data set, D, into v partitions, corresponding to the v outcomes of a test on attribute A

Gain ratio

- Gain ratio differs from information gain, which measures the information with respect to classification that is acquired based on the same partitioning
- The gain ratio is defined as

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

- The attribute with the maximum gain ratio is selected as the splitting attribute

Gain Ratio example

- Consider the previous example for computation of gain ratio for the attribute income
- A test on income splits the data of the following Table into three partitions, namely low, medium, and high, containing four, six, and four tuples,respectively

4

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Han, J., Pei, J. and Kamber, M., 2011. *Data mining: concepts and techniques*. Elsevier.

Calculate Entropy for high

- High :

High	Class: buys computer
Yes	2
No	2

- Calculate Entropy for high:

$$= -(2/4)\log_2(2/4) - (2/4)\log_2(2/4)$$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Calculate Entropy for ‘medium’

- Medium:

Medium	Class: buys computer
Yes	4
No	2

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

- Calculate Entropy for Medium:

$$= -(4/6)\log_2(4/6) - (2/6)\log_2(2/6)$$

Calculate Entropy for 'low'

- Low :

Low	Class: buys computer
Yes	3
No	1

- Calculate Entropy for Low:

$$= - (3/4)\log_2(3/4) - (1/4)\log_2(1/4)$$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Calculate Entropy for buying class D

- Calculate information:
- $= -p_y \log_2 (p_y) - p_n \log_2 (p_n)$
- Where p_y is probability of yes and p_n is probability of no

$$Info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits.}$$

Gain of income

- The expected information needed to classify a tuple in D if the tuples are partitioned according to income is:
- $\text{Info}_{income}(D) = (4/14)\,(-(2/4)\log_2(2/4) - (2/4)\log_2(2/4)) +$

$$(6/14)\,(-(4/6)\log_2(4/6) - (2/6)\log_2(2/6)) +$$

$$(4/14)\,(-(1/4)\log_2(1/4) - (3/4)\log_2(3/4))$$

$$= 0.911 \text{ bits}$$

Gain of income : $\text{Info}(D) - \text{Info}_{income}(D)$

$$= 0.94 – 0.911 = \boxed{0.029}$$

income
low – 4
Medium – 6
high – 4

Gain-Ratio(income)

- Calculation of split ratio:

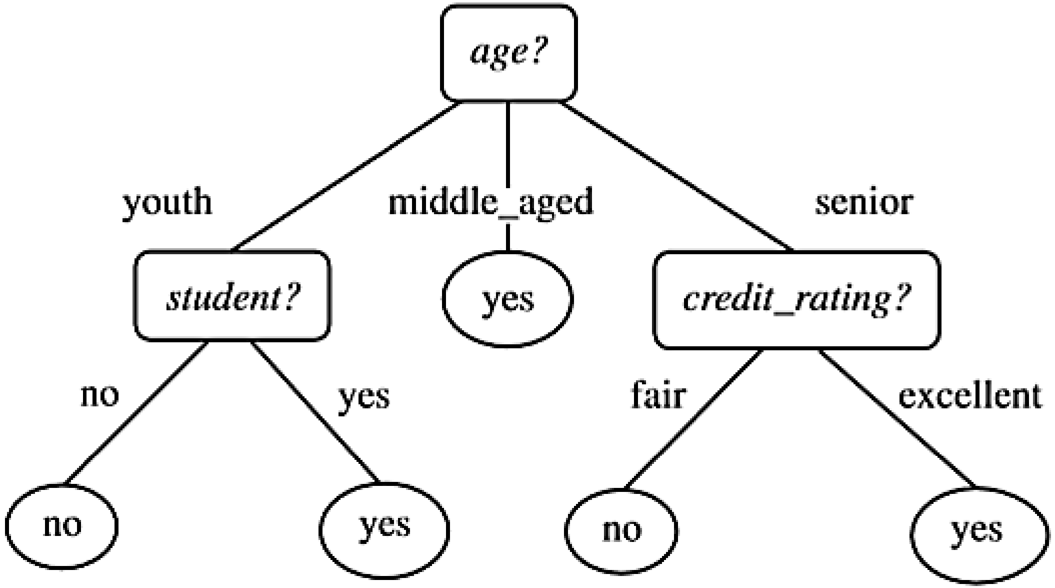
$$SplitInfo_A(D) = -\frac{4}{14}\times\log_2\left(\frac{4}{14}\right) - \frac{6}{14}\times\log_2\left(\frac{6}{14}\right) - \frac{4}{14}\times\log_2\left(\frac{4}{14}\right)$$

$$\boxed{= 0.926.}$$

- Therefore, Gain-Ratio(income) = 0.029/0.926 $\boxed{= 0.031}$

Interpretation

- Further we calculate the same for the rest 3 criteria (age, student, credit rating)
- The one with maximum Gain ratio value will results in the maximum reduction in impurity of the tuples in D and is returned as the splitting criterion


age?
youth
middle_aged
senior
student?
yes
credit_rating?
no
yes
fair
excellent
no
yes
no
yes

Decision tree using Gini index

- Let's take the Introduction of a decision tree using Gini index
- Let D be the training data of the following table

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Han, J., Pei, J. and Kamber, M., 2011. *Data mining: concepts and techniques*. Elsevier.

Example

- In this example, each attribute is discrete-valued
- Continuous-valued attributes have been generalized
- The class label attribute, buys computer, has two distinct values (namely, {yes, no}); therefore, there are two distinct classes (that is, $m = 2$)
- Let class C_1 correspond to 'yes' and class C_2 correspond to 'no'.
- There are nine tuples of class 'yes' and five tuples of class 'no'.
- A (root) node N is created for the tuples in D

Calculation of Gini(D)

- We first use the following Equation for Gini index to compute the impurity of D:

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

$$= \quad Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459.$$

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Gini index for income attribute

- Lets calculate Gini index for income attribute
- To find the splitting criterion for the tuples in D, we need to compute the Gini index for each attribute
- Let's start with the attribute income and consider each of the possible splitting subsets
- Income has three possible values, namely {low, medium, high}, then the possible subsets are {low, medium, high}, {low, medium}, {low, high}, {medium, high}, {low}, {medium}, {high}, and {}
- Power set and empty set will not be used for splitting

Gini index for income attribute

- Consider the subset{low, medium}
- This would result in 10 tuples in partition D1 satisfying the condition "income ∈{low, medium}"
- The remaining four tuples of D (high) would be assigned to partition D2

L M D1 H D2

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Tuples in partition D1

- Low + Medium:

Medium + Low	Class: buys computer
Yes	3+4 =7
No	1+ 2 = 3

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Tuples in partition D2

- High : (D4)

High	Class: buys computer
Yes	2
No	2

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Gini index for income attribute

- The Gini index value computed based on this partitioning is

$$Gini_{income \in \{low,medium\}}(D)$$

$$= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$

$= (10/14)\ (1- (7/10)^2 – (3/10)^2) +$

$(4/14)\ (1- (2/4)^2 – (2/4)^2)$

$= 0.443 = Gini_{\text{income} \in \{high\}}$

Gini index for income attribute

- Consider the subset{high, medium}
- This would result in 10 tuples in partition D1 satisfying the condition "income ∈{high, medium}"
- The remaining four tuples of D (low) would be assigned to partition D_2

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

H/M
D_1
L
D_2

Tuples in partition D1

- High + Medium:

Medium + high	Class: buys computer
Yes	2+4
No	2 + 2

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Tuples in partition D2

- Low :

Low	Class: buys computer
No	1
Yes	3

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Gini index for income attribute

- The Gini index value computed based on this partitioning is

$\text{Gini}_{\text{income} \in \{\text{high, medium}\}}$

$$= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$

$= (10/14)\,(1- (6/10)^2 – (4/10)^2) +$

$(4/14)\,(1- (1/4)^2 – (3/4)^2)$

$= 0.45 = \text{Gini}_{\text{income} \in \{\text{low}\}}$

H, M

D_1

L

D_2

Gini index for income attribute

- Consider the subset{high, low}
- This would result in 8 tuples in partition D1 satisfying the condition "income ∈{high, low}"
- The remaining six tuples of D (medium) would be assigned to partition D_2

H L D1

M D2

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Tuples in partition D1

- High + low:

high + low	Class: buys computer
Yes	2+3
No	2 + 1

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Tuples in partition D2

- Medium:

Low	Class: buys computer
No	2
Yes	4

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Gini index for income attribute

- The Gini index value computed based on this partitioning is

$\text{Gini}_{\text{income} \in \{\text{high, low}\}}$

$$= (8/14)\,(1 - (5/8)^2 - (3/8)^2) +$$

$$(6/14)\,(1 - (2/6)^2 - (4/6)^2)$$

$$= 0.458 = \text{Gini}_{\text{income} \in \{\text{medium}\}}$$

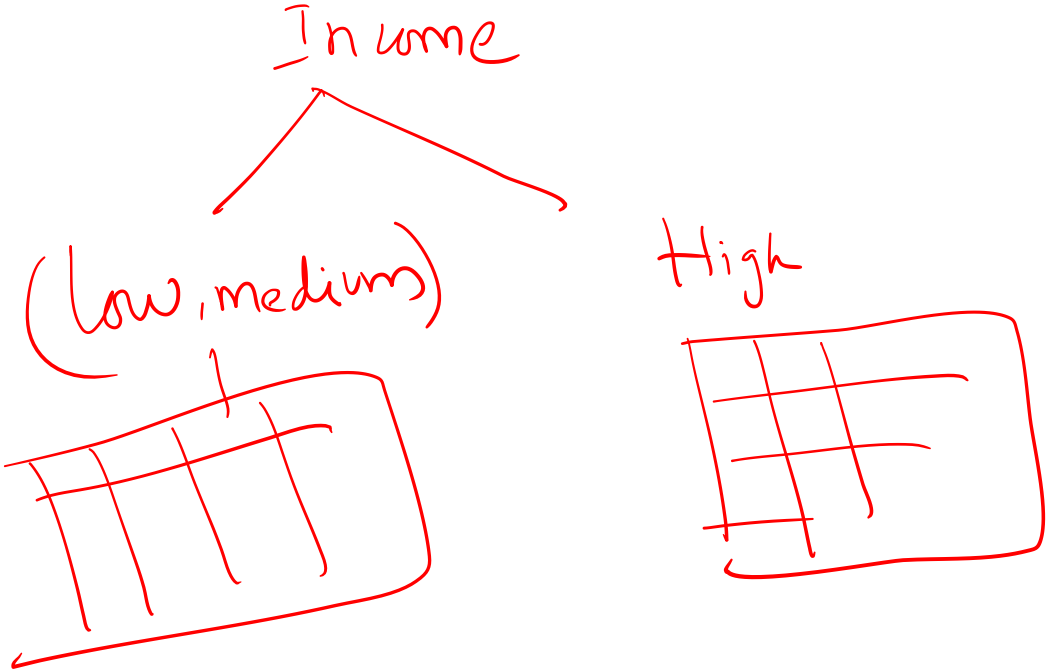
M

Gini Index values

	Gini Index values
$Gini_{income \in \{high, low\}}$	0.458
$Gini_{income \in \{high, medium\}}$	0.45
$Gini_{income \in \{medium, low\}}$	0.443

Interpretation

- The best binary split for attribute income is on {medium, low} (or {high}) because it minimizes the Gini index
- The splitting subset {medium,low} therefore give the minimum Gini index for attribute income
- Reduction in impurity = 0.459 – 0.443 = 0.016
- Further we calculate the same for the rest 3 criteria (age, student, credit rating)
- The one with minimum Gini index value will results in the maximum reduction in impurity of the tuples in D and is returned as the splitting criterion


Income
(low, medium)
High

Thank You