DATA ANALYTICS

Class # 20

Decision Tree Induction – CART & C4.5

Dr. Sreeja S R

Assistant Professor

Indian Institute of Information Technology

IIIT Sri City

ALGORITHM CART

CART ALGORITHM

- It is observed that information gain measure used in ID3 is biased towards test with many outcomes, that is, it prefers to select attributes having a large number of values.

- L. Breiman, J. Friedman, R. Olshen and C. Stone in 1984 proposed an algorithm to build a binary decision tree also called CART decision tree.
  - CART stands for **Classification and Regression Tree**
  - In fact, invented independently at the same time as ID3 (1984).
  - ID3 and CART are two cornerstone algorithms spawned a flurry of work on decision tree induction.

- CART is a technique that generates a **binary decision tree;** That is, unlike ID3, in CART, for each node only two children is created.

- ID3 uses Information gain as a measure to select the best attribute to be splitted, whereas CART do the same but using another measurement called **Gini index** . It is also known as **Gini Index of Diversity** and is denote as γ.

GINI INDEX OF DIVERSITY

Definition 20.1: **Gini Index**

Suppose, D is a training set with size $|D|$ and $C = \{c_1, c_2, \ldots, c_k\}$ be the set of k classifications and $A = \{a_1, a_2, \ldots, a_m\}$ be any attribute with m different values of it. Like entropy measure in ID3, CART proposes Gini Index (denoted by G) as the measure of impurity of D. It can be defined as follows.

$$G(D) = 1 - \sum_{i=1}^{k} p_i^2$$

where p_i is the probability that a tuple in D belongs to class c_i and p_i can be estimated as

$$p_i = \frac{|C_{i,D}|}{D}$$

where $|C_{i,D}|$ denotes the number of tuples in D with class c_i.

GINI INDEX OF DIVERSITY

Note

- $G(D)$ measures the "impurity" of data set D.
- The smallest value of $G(D)$ is zero
  - which it takes when all the classifications are same.
- It takes its largest value $= 1 - \frac{1}{k}$
  - when the classes are evenly distributed between the tuples, that is the frequency of each class is $\frac{1}{k}$.

GINI INDEX OF DIVERSITY

Definition 20.2: **Gini Index of Diversity**

Suppose, a binary partition on A splits D into D_1 and D_2, then the weighted average Gini Index of splitting denoted by $G_A(D)$ is given by

$$G_A(D) = \frac{|D_1|}{D}.G(D_1) + \frac{|D_2|}{D}.G(D_2)$$

This binary partition of D reduces the impurity and the reduction in impurity is measured by

$$\gamma(A, D) = G(D) - G_A(D)$$

GINI INDEX OF DIVERSITY AND CART

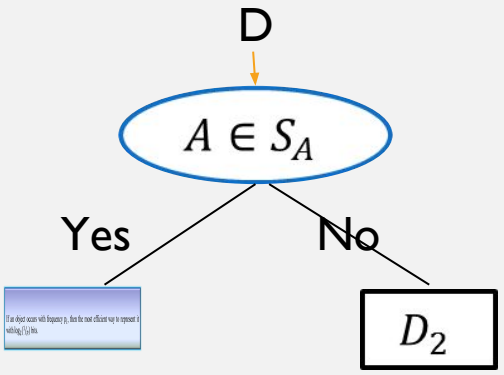
- This $\gamma(A, D)$ is called the Gini Index of diversity.
- It is also called as "impurity reduction".
- The attribute that maximizes the reduction in impurity (or equivalently, has the minimum value of $G_A(D)$) is selected for the attribute to be splitted.

N-ARY ATTRIBUTE VALUES TO BINARY SPLITTING

- The CART algorithm considers a binary split for each attribute.
- We shall discuss how the same is possible for attribute with more than two values.
- **Case 1: Discrete valued attributes**
- Let us consider the case where A is a discrete-valued attribute having m discrete values $a_1, a_2, \ldots, a_m$.
- To determine the best binary split on A, we examine all of the possible subsets say 2^A of A that can be formed using the values of A.
- Each subset $S_A \in 2^A$ can be considered as a binary test for attribute A of the form "$A \in S_A$?".

N-ARY ATTRIBUTE VALUES TO BINARY SPLITTING

- Thus, given a data set D, we have to perform a test for an attribute value A like

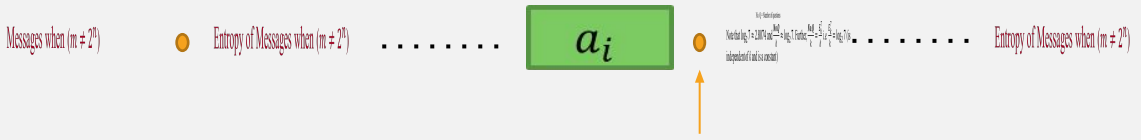


- This test is satisfied if the value of A for the tuples is among the values listed in S_A.
- If A has m distinct values in D, then there are 2^m possible subsets, out of which the empty subset { } and the power set $\{a_1, a_2, \ldots, a_n\}$ should be excluded (as they really do not represent a split).
- Thus, there are $2^m - 2$ possible ways to form two partitions of the dataset D, based on the binary split of A.

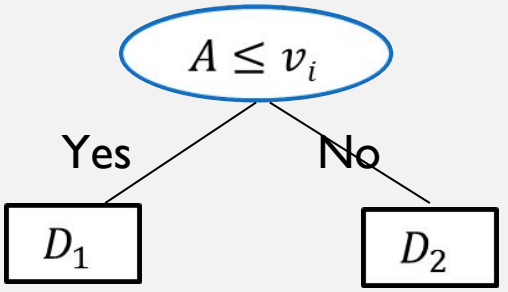
N-ARY ATTRIBUTE VALUES TO BINARY SPLITTING

Case2: Continuous valued attributes

- For a continuous-valued attribute, each possible split point must be taken into account.
- The strategy is similar to that followed in ID3 to calculate information gain for the continuous –valued attributes.
- According to that strategy, the mid-point between a_i and a_{i+1} , let it be v_i, then



$$v_i = \frac{a_i + a_{i+1}}{2}$$



N-ARY ATTRIBUTE VALUES TO BINARY SPLITTING

- Each pair of (sorted) adjacent values is taken as a possible split-point say v_i.
- D_1 is the set of tuples in D satisfying $A \leq v_i$ and D_2 in the set of tuples in D satisfying $A > v_i$.
- The point giving the minimum Gini Index $G_A(D)$ is taken as the split-point of the attribute A.

Note

- The attribute A and either its splitting subset S_A (for discrete-valued splitting attribute) or split-point v_i (for continuous valued splitting attribute) together form the splitting criteria.

CART ALGORITHM : ILLUSTRATION

Example 20.1 : CART Algorithm

Suppose we want to build decision tree for the data set EMP as given in the table below.

Age
Y : young
M : middle-aged
O : old

Salary
L : low
M : medium
H : high

Job
G : government
P : private

Performance
A : Average
E : Excellent

Class : Select
Y : yes
N : no

Tuple#	Age	Salary	Job	Performance	Select
1	Y	H	P	A	N
2	Y	H	P	E	N
3	M	H	P	A	Y
4	O	M	P	A	Y
5	O	L	G	A	Y
6	O	L	G	E	N
7	M	L	G	E	Y
8	Y	M	P	A	N
9	Y	L	G	A	Y
10	O	M	G	A	Y
11	Y	M	G	E	Y
12	M	M	P	E	Y
13	M	H	G	A	Y
14	O	M	P	E	N

CART ALGORITHM : ILLUSTRATION

For the EMP data set,

$$G(EMP) = 1 - \sum_{i=1}^{2} p_i^2$$

$$= 1 - \left[\left(\frac{9}{14}\right)^2 + \left(\frac{5}{14}\right)^2\right]$$

$$= \mathbf{0.4592}$$

Now let us consider the calculation of $G_A(EMP)$ for **Age**, **Salary**, **Job** and **Performance**.

CART ALGORITHM : ILLUSTRATION

Attribute of splitting: Age

The attribute age has three values, namely Y, M and O. So there are 6 subsets, that should be considered for splitting as:

$$\underset{age_1}{\{Y\}}, \quad \underset{age_2}{\{M\}}, \quad \underset{age_3}{\{O\}}, \quad \underset{age_4}{\{Y,M\}}, \quad \underset{age_5}{\{Y,O\}}, \quad \underset{age_6}{\{M,O\}}$$

$$G_{age_1}(D) = \frac{5}{14} * \left(1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2\right) + \frac{9}{14}\left(1 - \left(\frac{6}{14}\right)^2 - \left(\frac{8}{14}\right)^2\right) = \mathbf{0.4862}$$

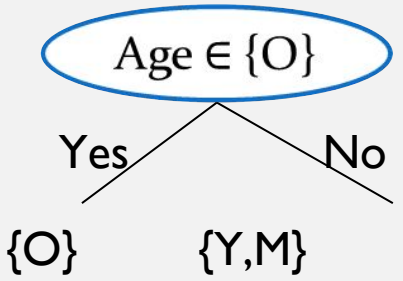
$G_{age_2}(D) = \mathbf{?}$

$G_{age_3}(D) = \mathbf{?}$

$G_{age_4}(D) = G_{age_3}(D)$

$G_{age_5}(D) = G_{age_2}(D)$

$G_{age_6}(D) = G_{age_1}(D)$



The best value of Gini Index while splitting attribute Age is $\gamma(Age_3, D) = \mathbf{0.3750}$

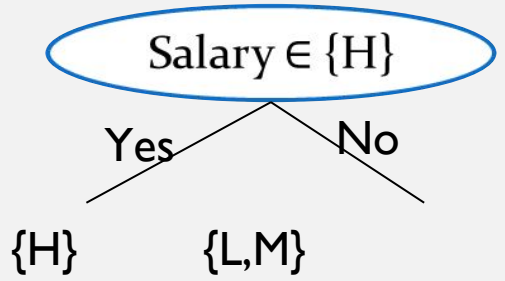
CART ALGORITHM : ILLUSTRATION

Attribute of Splitting: Salary

The attribute salary has three values namely L, M and H. So, there are 6 subsets, that should be considered for splitting as:

$\{L\}$	$\{M,H\}$	$\{M\}$	$\{L,H\}$	$\{H\}$	$\{L,M\}$
sal_1	sal_2	sal_3	sal_4	sal_5	sal_6

$$G_{sal_1}(D) = G_{sal_2}(D) = 0.3000$$
$$G_{sal_3}(D) = G_{sal_4}(D) = 0.3150$$
$$G_{sal_5}(D) = G_{sal_6}(D) = 0.4508$$



$$\gamma(salary_{(5,6)}, D) = 0.4592 - 0.4508 = 0.0084$$

CART ALGORITHM : ILLUSTRATION

Attribute of Splitting: job

Job being the binary attribute , we have

$$G_{job}(D) = \frac{7}{14} G(D_1) + \frac{7}{14} G(D_2)$$

$$= \frac{7}{14}\left[1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2\right] + \frac{7}{14}\left[1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2\right] = ?$$

$$\gamma(job, D) = ?$$

CART ALGORITHM : ILLUSTRATION

Attribute of Splitting: Performance

Job being the binary attribute , we have

$G_{Performance}(D) = ?$

$\gamma(performance, D) = ?$

Out of these $\gamma(salary, D)$ gives the minimum value and hence, the attribute **Salary** would be chosen for splitting subset $\{M, H\}$ or $\{L\}$.

Note:

It can be noted that the procedure following "information gain" calculation (i.e. $\propto (A, D)$) and that of "impurity reduction" calculation (i.e. $\gamma(A, D)$) are near about.

CALCULATING Γ USING FREQUENCY TABLE

- We have learnt that splitting on an attribute gives a reduction in the average Gini Index of the resulting subsets (as it does for entropy).
- Thus, in the same way the average weighted Gini Index can be calculated using the same frequency table used to calculate information gain $\alpha(A, D)$, which is as follows.

The G(D_j) for the j^{th} subset D_j

$$G(D_j) = 1 - \sum_{i=1}^{k} \left(\frac{f_{ij}}{|D_j|} \right)^2$$

CALCULATING Γ USING FREQUENCY TABLE

The average weighted Gini Index, $G_A(D_j)$ (assuming that attribute has m distinct values is)

$$G_A(D_j) = \sum_{j=1}^{k} \frac{|D_j|}{|D_1|} . G(D_j)$$

$$= \sum_{j=1}^{m} \frac{|D_j|}{|D|} - \sum_{j=1}^{m} \sum_{i=1}^{k} \frac{|D_j|}{|D|} . \left(\frac{f_{ij}}{|D_j|} \right)^2$$

$$= 1 - \frac{1}{|D|} \sum_{j=1}^{m} \frac{1}{D_j} . \sum_{i=1}^{k} {f_{ij}}^2$$

The above gives a formula for m-attribute values; however, it an be fine tuned to subset of attributes also.

ILLUSTRATION: CALCULATING Γ USING FREQUENCY TABLE

Example 20.2 : Calculating γ using frequency table of OPTH

Let us consider the frequency table for OPTH database considered earlier. Also consider the attribute A_1 with three values 1, 2 and 3. The frequency table is shown below.

	1	2	3
Class 1	2	1	1
Class 2	2	2	1
Class 3	4	5	6
Column sum	8	8	8

ILLUSTRATION: CALCULATING Γ USING FREQUENCY TABLE

Now we can calculate the value of Gini Index with the following steps:

1. For each non-empty column, form the sum of the squares of the values in the body of the table and divide by the column sum.

2. Add the values obtained for all columns and divided by $|D|$, the size of the database.

3. Subtract the total from 1.

As an example, with reference to the frequency table as mentioned just.

$$A_1 = 1 = \frac{(2^2+2^2+4^2)}{24} = 3.0 \qquad A_1 = 2 = \frac{(1^2+2^2+5^2)}{24} = 3.75$$

$$A_1 = 3 = \frac{(1^2+1^2+6^2)}{24} = 4.75 \qquad \text{So, } G_{A1}(D) = 1 - \frac{1+3.75+4.75}{24} = 0.5208$$

Illustration: Calculating γ using Frequency Table

Thus, the reduction in the value of Gini Index on splitting attribute A_1 is

$$\gamma(A_1, D) = 0.5382 - 0.5208 = 0.0174$$

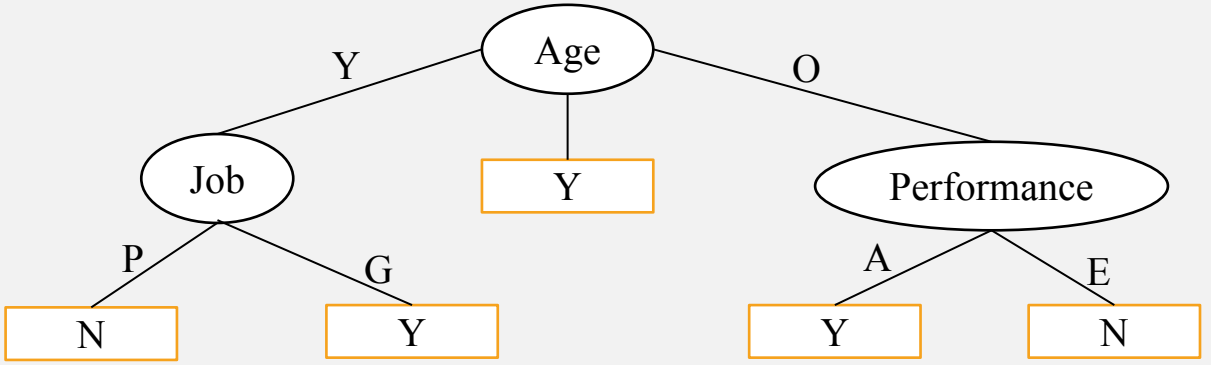
where $G(D) = 0.5382$

The calculation can be extended to other attributes in the OTPH database and is left as an exercise.

DECISION TREES WITH ID3 AND CART ALGORITHMS

Example 20.3 : Comparing Decision Trees of EMP Data set

Compare two decision trees obtained using ID3 and CART for the EMP dataset. The decision tree according to ID3 is given for your ready reference (subject to the verification)



Decision Tree using ID3

?

Decision Tree using CART

ALGORITHM C4.5

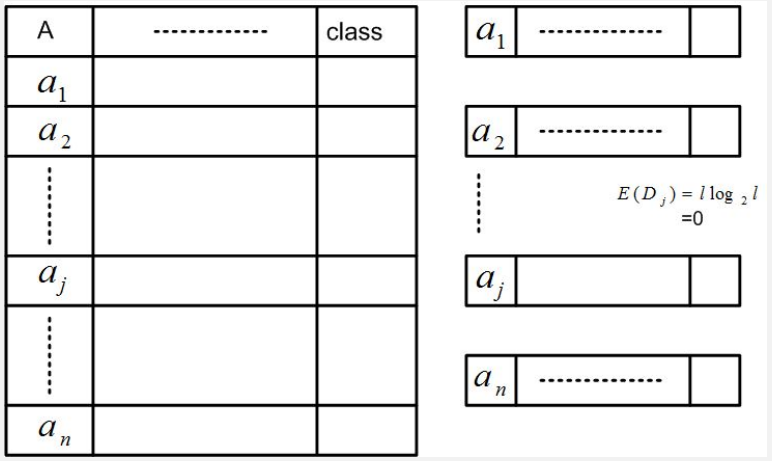
ALGORITHM C4.5 : INTRODUCTION

- J. Ross Quinlan, a researcher in machine learning developed a decision tree induction algorithm in 1984 known as ID3 (Iterative Dichotometer 3).
- Quinlan later presented C4.5, a successor of ID3, addressing some limitations in ID3.
- ID3 uses information gain measure, which is, in fact biased towards splitting attribute having a large number of outcomes.
- For example, if an attribute has distinct values for all tuples, then it would result in a large number of partitions, each one containing just one tuple.
  - In such a case, note that each partition is pure, and hence the purity measure of the partition, that is $E_A(D) = 0$

ALGORITHM C4.5 : INTRODUCTION

Example 20.4 : Limitation of ID3

In the following, each tuple belongs to a unique class. The splitting on A is shown.



$$E_A(D) = \sum_{j=1}^{n} \frac{|D_j|}{|D|} . E(D_j) = \sum_{j=1}^{n} \frac{1}{|D|} . 0 = 0$$

Thus, $\alpha(A, D) = E(D) - E_A(D)$ is maximum in such a situation.

ALGORITHM: C 4.5 : INTRODUCTION

- Although, the previous situation is an extreme case, intuitively, we can infer that ID3 favours splitting attributes having a large number of values
  - compared to other attributes, which have a less variations in their values.

- Such a partition appears to be useless for classification.

- This type of problem is called **overfitting problem**.

Note:

Decision Tree Induction Algorithm ID3 may suffer from overfitting problem.

ALGORITHM: C 4.5 : INTRODUCTION

- The overfitting problem in ID3 is due to the measurement of information gain.
- In order to reduce the effect of the use of the bias due to the use of information gain, C4.5 uses a different measure called **Gain Ratio**, denoted as β.
- Gain Ratio is a kind of normalization to information gain using a **split information**.

ALGORITHM: C4.5 : GAIN RATIO

Definition 20.3: **Gain Ratio**

The gain ratio can be defined as follows. We first define split information $E_A^*(D)$ as

$$E_A^*(D) = -\sum_{j=1}^{m} \frac{|D_j|}{|D|} . \log \frac{|D_j|}{|D|}$$

Here, m is the number of distinct values in A.

The gain ratio is then defined as $\beta(A, D) = \frac{\alpha(A,D)}{E_A^*(D)}$, where $\alpha(A, D)$ denotes the information gain on splitting the attribute A in the dataset D.

PHYSICAL INTERPRETATION OF $E_A^*(D)$

Split information $E_A^*(D)$

- The value of split information depends on
  - the number of (distinct) values an attribute has and
  - how uniformly those values are distributed.
- In other words, it represents the potential information generated by splitting a data set D into m partitions, corresponding to the m outcomes of on attribute A.
- Note that for each outcome, it considers the number of tuples having that outcome with respect to the total number of tuples in D.

PHYSICAL INTERPRETATION OF $E_A^*(D)$

Example 20.5 : Split information $E_A^*(D)$

- To illustrate $E_A^*(D)$, let us examine the case where there are 32 instances and splitting an attribute A which has a_1, a_2, a_3 and a_4 sets of distinct values.
  - Distribution 1 : Highly non-uniform distribution of attribute values

Frequency	32	0	0	0

$$E_A^*(D) = -\frac{32}{32}log_2(\frac{32}{32}) = -log_2 1 = 0$$

  - Distribution 2

Frequency	16	16	0	0

$$E_A^*(D) = -\frac{16}{32}log_2(\frac{16}{32}) - \frac{16}{32}log_2(\frac{16}{32}) = log_2 2 = 1$$

PHYSICAL INTERPRETATION OF $E_A^*(D)$

Distribution 3

Frequency	16	8	8	0

$$E_A^*(D) = -\frac{16}{32}log_2(\frac{16}{32}) - \frac{8}{32}log_2(\frac{8}{32}) - \frac{8}{32}log_2(\frac{8}{32}) = 1.5$$

- Distribution 4

Frequency	16	8	4	4

$$E_A^*(D) = 1.75$$

- Distribution 5: Uniform distribution of attribute values

Frequency	8	8	8	8

$$E_A^*(D) = (-\frac{8}{32}log_2(\frac{8}{32}))*4 = -log_2(\frac{1}{4}) = 2.0$$

PHYSICAL INTERPRETATION OF $E_A^*(D)$

- In general, if there are m attribute values, each occurring equally frequently, then the split information is $log_2 m$.
- Based on the Example 20.5, we can summarize our observation on split information as under:
  - Split information is 0 when there is a single attribute value. It is a trivial case and implies *the minimum possible value of split information*.
  - For a given data set, when instances are uniformly distributed with respect to the attribute values, split information increases as the number of different attribute values increases.
  - The maximum value of split information occur when there are many possible attribute values, all are equally frequent.

Note:

- Split information varies between 0 and $log_2 m$ (both inclusive)

PHYSICAL INTERPRETATION OF $\beta(A, B)$

- Information gain signifies how much information will be gained on partitioning the values of attribute A
  - Higher information gain means splitting of A is more desirable.
- On the other hand, split information forms the denominator in the gain ratio formula.
  - This implies that higher the value of split information is, lower the gain ratio.
  - In turns, it decreases the information gain.
- Further, information gain is large when there are many distinct attribute values.
  - When many distinct values, split information is also a large value.
  - This way split information reduces the value of gain ratio, thus resulting a balanced value for information gain.
- Like information gain (in ID3), the attribute with the maximum gain ratio is selected as the splitting attribute in C4.5.

CALCULATION OF β USING FREQUENCY TABLE

- The frequency table can be used to calculate the gain ratio for a given data set and an attribute.
- We have already learned the calculation of information gain using Frequency Table.
- To calculate gain ratio, in addition to information gain, we are also to calculate split information.
- This split information can be calculated from frequency table as follows.
- For each non-zero column sum say s_j contribute $|D_j|$ for the j-th column (i.e., the j-th value of the attribute). Thus the split information is

$$E_A^*(D) = -\sum_{j=1}^{m} \frac{s_j}{|D|} log_2 \frac{s_j}{|D|}$$

If there are m-columns in the frequency table.

Practice:

Using Gain ratio as the measurement of splitting attributes, draw the decision trees for OPTH and EMP data sets. Give calculation of gain ratio at each node.

SUMMARY OF DECISION TREE INDUCTION ALGORITHMS

- We have learned the building of a decision tree given a training data.
  - The decision tree is then used to classify a test data.

- For a given training data *D*, the important task is to build the decision tree so that:
  - All test data can be classified accurately

  - The tree is balanced and with as minimum depth as possible, thus the classification can be done at a faster rate.

- In order to build a decision tree, several algorithms have been proposed. These algorithms differ from the chosen splitting criteria, so that they satisfy the above mentioned objectives as well as the decision tree can be induced with minimum time complexity. We have studied three decision tree induction algorithms namely ID3, CART and C4.5. A summary of these three algorithms is presented in the following table.

TABLE 20.6

Algorithm	Splitting Criteria	Remark
ID3		

Algorithm	Splitting Criteria	Remark
CART		

Algorithm	Splitting Criteria	Remark
C4.5		

In addition to this, we also highlight few important characteristics of decision tree induction algorithms in the following.

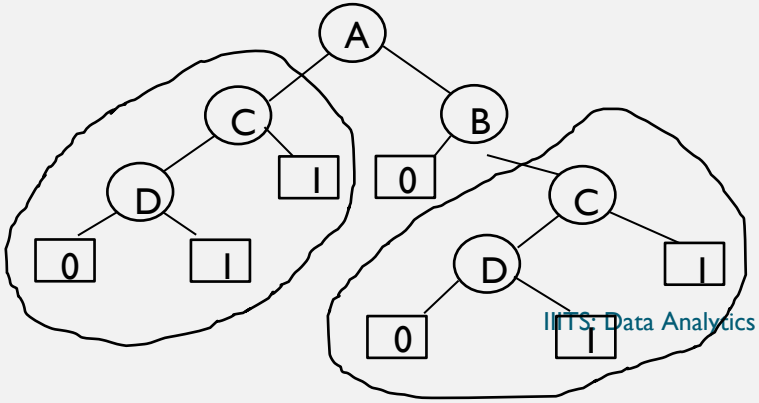
NOTES ON DECISION TREE INDUCTION ALGORITHMS

1. **Optimal Decision Tree:** Finding an optimal decision tree is an NP-complete problem. Hence, decision tree induction algorithms employ a heuristic based approach to search for the best in a large search space. Majority of the algorithms follow a greedy, top-down recursive divide-and-conquer strategy to build decision trees.
2. **Missing data and noise:** Decision tree induction algorithms are quite robust to the data set with missing values and presence of noise. However, proper data pre-processing can be followed to nullify these discrepancies.
3. **Redundant Attributes:** The presence of redundant attributes does not adversely affect the accuracy of decision trees. It is observed that if an attribute is chosen for splitting, then another attribute which is redundant is unlikely to chosen for splitting.
4. **Computational complexity:** Decision tree induction algorithms are computationally inexpensive, in particular, when the sizes of training sets are large, Moreover, once a decision tree is known, classifying a test record is extremely fast, with a worst-case time complexity of O(d), where d is the maximum depth of the tree.

Notes on Decision Tree Induction algorithms

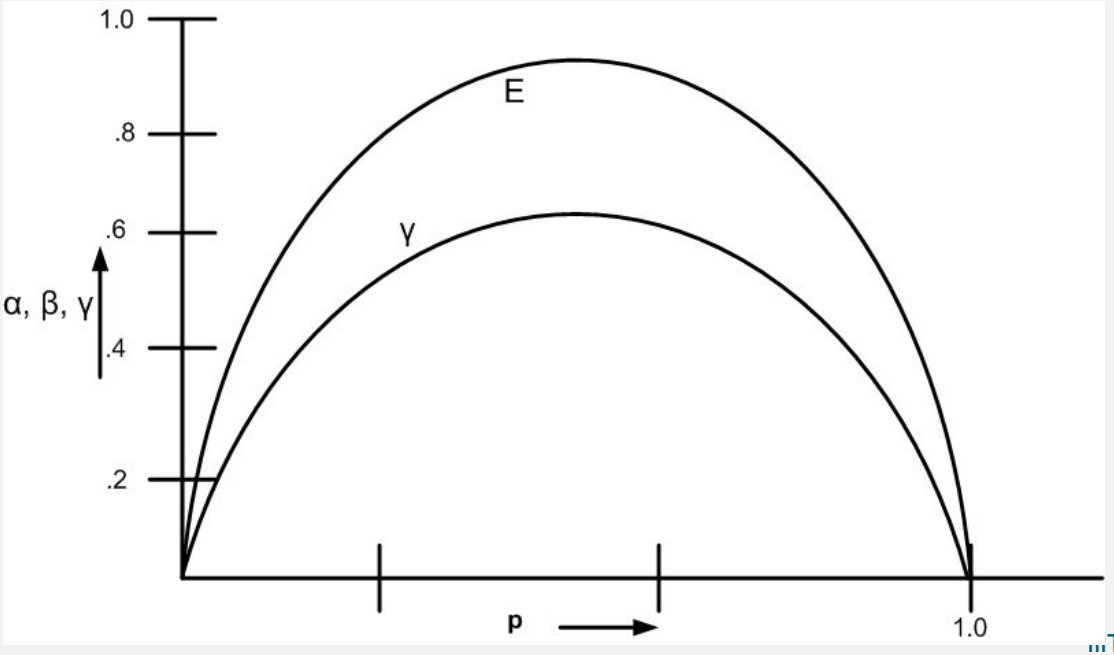
5. **Data Fragmentation Problem:** Since the decision tree induction algorithms employ a top-down, recursive partitioning approach, the number of tuples becomes smaller as we traverse down the tree. At a time, the number of tuples may be too small to make a decision about the class representation, such a problem is known as the data fragmentation. To deal with this problem, further splitting can be stopped when the number of records falls below a certain threshold.

6. **Tree Pruning:** A sub-tree can replicate two or more times in a decision tree (see figure below). This makes a decision tree unambiguous to classify a test record. To avoid such a sub-tree replication problem, all sub-trees except one can be pruned from the tree.



Notes on Decision Tree Induction algorithms

7. **Decision tree equivalence:** The different splitting criteria followed in different decision tree induction algorithms have little effect on the performance of the algorithms. This is because the different heuristic measures (such as information gain (α), Gini index (γ) and Gain ratio (β) are quite consistent with each other); also see the figure below.



REFERENCE

- The detail material related to this lecture can be found in

Data Mining: Concepts and Techniques, (3rd Edn.), Jiawei Han, Micheline Kamber, Morgan Kaufmann, 2015.

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison-Wesley, 2014

Any question?