INTRODUCTION TO DATA ANALYTICS

Class # 19

Decision Tree Induction – ID3

Dr. Sreeja S R

Assistant Professor

Indian Institute of Information Technology

IIIT Sri City

Entropy Calculation

Lemma 19.1: **Entropy calculation**

The minimum number of *yes/no* questions needed to identify an unknown object from $m = 2^n$ equally likely possible object is n.

If m is not a power of 2, then the entropy of a set of m distinct objects that are equally likely is $\log_2 m$

ENTROPY IN MESSAGES

- We know that the most conventional way to code information is using binary bits, that is, using 0s and 1s.
- The answer to a question that can only be answered *yes/no* (with equal probability) can be considered as containing one **unit of information**, that is, one bit.
- In other words, the unit of information can also be looked at as the amount of information that can be **coded** using only 0s and 1s.

ENTROPY IN MESSAGES

Example 19.7: Information coding

- If we have **two** possible objects say **male** and **female,** then we use the coding

  0 = female

  1 = male $\quad m = 2(= 2^n, n = 1)$

- We can encode **four** possible objects say **East**, **West**, **North**, **South** using two bits, for example

  00 : North

  01 : East $\quad m = 4(= 2^n, n = 2)$

  10 : West

  11 : South

- We can encode **eight** values say eight different colours, we need to use **three** bits, such as

  000 : Violet

  001 : Orange

  010 : Red

  011 : White

  100 : Yellow

  101 : Indigo $\quad m = 8(= 2^n, n = 3)$

  110 : Blue

  111 : Green

Thus, in general, to code *m* values, each in a distinct manner, we need *n* bits such that $m = 2^n$.

ENTROPY IN MESSAGES

- In this point, we can note that to identify an object, if it is encoded with bits, then we have to ask questions in an alternative way. For example
  - Is the first bit 0?
  - Is the second bit 0?
  - Is the third bit 0? and so on
- Thus, we need n questions, if m objects are there such that $m = 2^n$.
- The above leads to (an alternative) and equivalent definition of entropy

Definition 19.1: Entropy

The entropy of a set of m distinct values is the number of bits needed to encode all the values in the most efficient way.

MESSAGES WHEN $(m \neq 2^n)$

- In the previous discussion, we have assumed that m, the number of distinct objects is exactly a power of 2, that is $m = 2^n$ for some $n \geq 1$ and all m objects are equally likely.
- This is mere an assumption to make the discussion simplistic.
- In the following we try to redefine the entropy calculation in more general case, that is, when m $\neq 2^n$ and not necessarily m objects are equally probable. Let us consider a different instance of *yes/no* question game, which is as follows.

Example 19.8: Name game

- There are seven days: Sun, Mon, Tue, Wed, Thu, Fri, Sat.
- We are to identify a sequence of $k \geq 1$ such values (each one chosen independently of the others, that is, repetitions are allowed).
- We denote the minimum number of yes/no questions needed to identify a sequence of k unknown values drawn independently from m possibilities as E_k^m, the entropy in this case.
- In other words, E_k^m is the number of questions required to discriminate amongst m^k distinct possibilities.

MESSAGES WHEN $(m \neq 2^n)$

- Here, $m = 7$ (as stated in the game of sequence of days) and $k = 6$ (say).
- An arbitrary sequence may be {Tue, Thu, Tue, Mon, Sun, Tue}, etc. There are $7^6 = 117649$ possible sequences of six days.
- From our previous understanding, we can say that the minimum number of yes/no questions that is required to identify such a sequence is $\log_2 117649 = 16.8443$.
- Since, this is a non integer number, and the number of question should be an integer, we can say *17* questions are required. Thus,

$$E_6^7 = \lceil \log_2 7^6 \rceil$$

- In general,

$$E_k^m = \lceil \log_2 m^k \rceil$$

- Alternatively, the above can be written as,

$$\lceil \log_2 m^k \rceil \leq E_k^m \leq \lceil \log_2 m^k \rceil + 1$$

- Or

$$\lceil \log_2 m \rceil \leq \frac{E_k^m}{k} \leq \left\lceil \log_2 m \right\rceil + \frac{1}{k}$$

ENTROPY OF MESSAGES WHEN $(m \neq 2^n)$

Note that here $\frac{E_k^m}{k}$ is the average number of questions needed to determine each of the values in a sequence of k values. By choosing a large enough value of k, that is, a long enough sequence, the value of $\frac{1}{k}$ can be made as small as we wish. Thus, the average number of questions required to determine each value can be made arbitrarily close to $\log_2 m$. This is evident from our earlier workout, for example, tabulated below, for $m = 7$.

$$E_k^m = \lceil \log_2 m^k \rceil$$

k	m^k	$\log_2 m^k\rceil$	No. Q	$\frac{No.Q}{k}$
6	117649	16.84413	17	2.8333
21		58.95445	59	2.8095
1000		2807.3549	2808	2.8080
…..	…..	…..	…..	…..

No. Q = Number of questions

Note that $\log_2 7 \approx 2.8074$ and $\frac{No.Q}{k} \approx \log_2 7$. Further, $\frac{No.Q}{k} = \frac{E_k^7}{k}$ *i.e.* $\frac{E_k^7}{k} = \log_2 7$ (is independent of k and is a constant!)

ENTROPY OF MESSAGES WHEN $(m \neq 2^n)$

Lemma 19.3: Entropy Calculation

The entropy of a set of m distinct objects is $\log_2 m$ even when m is not exactly a power of 2.

- We have arrived at a conclusion that $E = \log_2 m$ for any value of m, irrespective of whether it is a power of 2 or not.

Note: E is not necessarily be an integer always.

- Next, we are to have our observation, if all m objects are not equally probable.
- Suppose, p_i denotes the frequency with which the i^{th} of the m objects occurs, where $0 \leq p_i \leq 1$ for all p_i such that

$$\sum_{i=1}^{m} p_i = 1$$

DISCRIMINATING AMONGST M VALUES ($m \neq 2^n$)

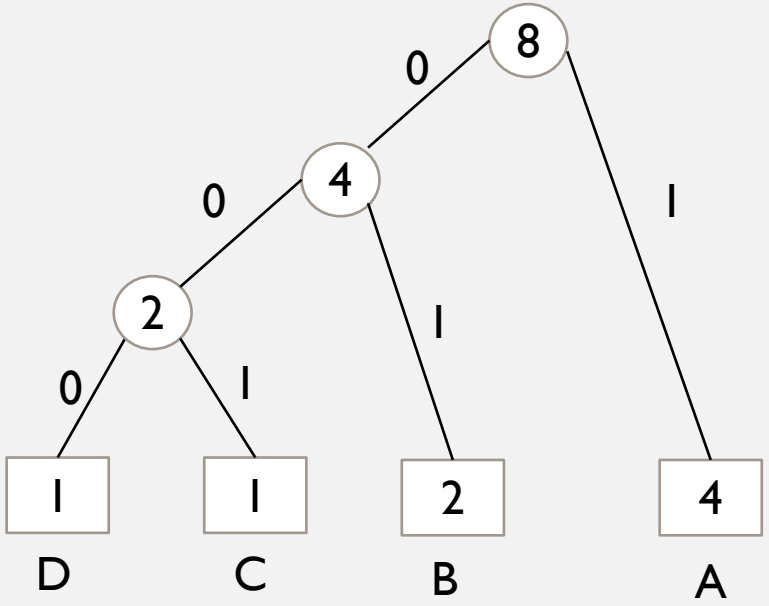
Example 19.8: Discriminating among objects

- Suppose four objects A, B, C *and* D which occur with frequencies $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{8}$, respectively. (A B C D A A A B)
- Thus, in this example, $m = 4$ and $p_1 = \frac{1}{2}$, $p_2 = \frac{1}{4}$, $p_3 = \frac{1}{8}$ and $p_4 = \frac{1}{8}$.
- Using standard 2-bit encoding, we can represent them as

$$A = 00,$$
$$B = 01,$$
$$C = 10,$$
$$D = 11.$$

- Also, we can follow variable length coding (also called Huffman coding) as an improved way of representing them.

HUFFMAN CODING



- The Huffman coding of A, B, C *and* D with their frequencies $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{8}$ are shown below.

  A = 1
  B = 01
  C = 001
  D = 000

DISCRIMINATING AMONGST M VALUES ($m \neq 2^n$)

- With the above representation say, if A is to be identified, then we need to examine only one question, for B it is 2 and for C and D both, it is 3.
- Thus, on the average, we need

$$\frac{1}{2}\times 1 + \frac{1}{4}\times 2 + \frac{1}{8}\times 3 + \frac{1}{8}\times 3 = 1.75\ bits$$

- This is the number of yes/no questions to identify any one of the four objects, whose frequency of occurrences are not uniform.
- This is simply in contrast to 2-bit encoding, where we need 2-bits (questions) on the average.

DISCRIMINATING AMONGST M VALUES ($m \neq 2^n$)

- It may be interesting to note that even with variable length encoding, there are several ways of encoding. Few of them are given below.

1)	2)	3)
$A = 0$	$A = 01$	$A = 101$
$B = 11$	$B = 1$	$B = 001$
$C = 100$	$C = 001$	$C = 10011$
$D = 101$	$D = 000$	$D = 100001$

- The calculation of entropy in the observed cases can be obtained as:

  1) 1.75 2) 2 3) 3.875

- Anyway, key to finding the most efficient way of encoding is to **assign a smallest number of bits to the object with highest frequency** and so on.
- The above observation is also significant in the sense that it provides a **systematic way of finding a sequence of well-chosen question in order to identify an object at a faster rate.**

INFORMATION CONTENT

Based on the previous discussion we can easily prove the following lemma.

Lemma 19.4: Information content

If an object occurs with frequency p, then the most efficient way to represent it with $\log_2(1/p)$ bits.

Example 19.9: Information content

- A which occurs with frequency $\frac{1}{2}$ is represented by 1-bit, B which occurs with frequency $\frac{1}{4}$ represented by 2-bits and both C and D which occurs with frequency $\frac{1}{8}$ are represented by 3 bits each.

ENTROPY CALCULATION

We can generalize the above understanding as follows.

- If there are m objects with frequencies $p_1, p_2 \ldots\ldots, p_m$, then the average number of bits (i.e. questions) that need to be examined a value, that is, entropy is the frequency of occurrence of the i^{th} value multiplied by the number of bits that need to be determined, summed up values of i from 1 to m.

Theorem 19.4: Entropy calculation

If p_i denotes the frequencies of occurrences of m distinct objects, then the entropy E is

$$E = \sum_{i=1}^{m} p_i \log(1/p_i) \text{ and } \sum_{i=1}^{m} p_i = 1$$

Note:

- If all are equally likely, then $p_i = \frac{1}{m}$ and $E = \log_2 m$; it is the special case.

ENTROPY OF A TRAINING SET

- If there are k classes $c_1, c_2 \ldots\ldots, c_k$ and p_i for $i = 1\ to\ k$ denotes the number of occurrences of classes c_i divided by the total number of instances (i.e., the frequency of occurrence of c_i) in the training set, then entropy of the training set is denoted by

$$E = -\sum_{i=1}^{m} p_i \log_2 p_i$$

Here, E is measured in "bits" of information.

Note:

- The above formula should be summed over the non-empty classes only, that is, classes for which $p_i \neq 0$
- E is always a positive quantity
- E takes it's minimum value (zero) if and only if all the instances have the same class (i.e., the training set with only **one** non-empty class, for which the probability 1).
- Entropy takes its maximum value when the instances are equally distributed among k possible classes. In this case, the maximum value of E is $log_2\ k$.

ENTROPY OF A TRAINING SET

Example 19.10: OPTH dataset

Consider the OTPH data shown in the following table with total 24 instances in it.

Age	Eye sight	Astigmatic	Use Type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	1
2	2	2	1	3
2	2	2	2	1
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	3
3	2	1	1	3
3	2	1	2	3
3	2	2	1	3
3	2	2	2	3

A coded forms for all values of attributes are used to avoid the cluttering in the table.

ENTROPY OF A TRAINING SET

Specification of the attributes are as follows.

Age	Eye Sight	Astigmatic	Use Type
1: Young	1: Myopia	1: No	1: Frequent
2: Middle-aged	2: Hypermetropia	2: Yes	2: Less
3: Old			

Class: 1: Contact Lens 2:Normal glass 3: Nothing

In the OPTH database, there are 3 classes and 4 instances with class 1, 5 instances with class 2 and 15 instances with class 3. Hence, entropy E of the database is:

$$E = -\frac{4}{24}\log_2\frac{4}{24} - \frac{5}{24}\log_2\frac{5}{24} - \frac{15}{24}\log_2\frac{15}{24} = 1.3261$$

Note:

- The entropy of a training set implies the number of yes/no questions, on the average, needed to determine an unknown test to be classified.

- It is very crucial to decide the series of questions about the value of a set of attribute, which collectively determine the classification. Sometimes it may take one question, sometimes many more.

- Decision tree induction helps us to ask such a series of questions. In other words, we can utilize entropy concept to build a better decision tree.

How entropy can be used to build a decision tree ?

DECISION TREE INDUCTION TECHNIQUES

- Decision tree induction is a top-down, recursive and divide-and-conquer approach.
- The procedure is to choose an attribute and split it into from a larger training set into smaller training sets.

- Different algorithms have been proposed to take a good control over
  1. Choosing the best attribute to be splitted, and
  2. Splitting criteria

- Several algorithms have been proposed for the above tasks. In this lecture, we shall limit our discussions into three important of them
  - **ID3**
  - **C 4.5**
  - **CART**

ALGORITHM ID3

ID3: DECISION TREE INDUCTION ALGORITHMS

- Quinlan [1986] introduced the ID3, a popular short form of **I**terative **D**ichotomizer 3 for decision trees from a set of training data.

- In ID3, each node corresponds to a splitting attribute and each arc is a possible value of that attribute.

- At each node, the splitting attribute is selected to be the most informative among the attributes not yet considered in the path starting from the root.

ALGORITHM ID3

- In ID3, entropy is used to measure how informative a node is.
  - It is observed that splitting on any attribute has the property that average entropy of the resulting training subsets will be less than or equal to that of the previous training set.
- ID3 algorithm defines a measurement of a splitting called **Information Gain** to determine the goodness of a split.
  - The attribute with the largest value of information gain is chosen as the splitting attribute and
  - it partitions into a number of smaller training sets based on the distinct values of attribute under split.

DEFINING INFORMATION GAIN

- We consider the following symbols and terminologies to define information gain, which is denoted as α.
- $D \equiv$ denotes the training set at any instant
- $|D| \equiv$ denotes the size of the training set D
- $E(D) \equiv$ denotes the entropy of the training set D
- The entropy of the training set D

$$E(D) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

  - where the training set D has $c_1, c_2, \ldots, c_k$, the k number of distinct classes and
  - p_i, $0 < p_i \leq 1$ is the probability that an arbitrary tuple in D belongs to class c_i ($i = 1, 2, \ldots, k$).

DEFINING INFORMATION GAIN

- p_i can be calculated as

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

- where $C_{i,D}$ is the set of tuples of class c_i in D.

- Suppose, we want to partition D on some attribute A having m distinct values $\{a_1, a_2, \ldots, a_m\}$.

- Attribute A can be considered to split D into m partitions $\{D_1, D_2, \ldots, D_m\}$, where D_j $(j = 1,2, \ldots, m)$ contains those tuples in D that have outcome a_j of A.

DEFINING INFORMATION GAIN

Definition 19.4: **Weighted Entropy**

The weighted entropy denoted as $E_A(D)$ for all partitions of D with respect to A is given by:

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

Here, the term $\frac{|D_j|}{|D|}$ denotes the weight of the j-th training set.

More meaningfully, $E_A(D)$ is the expected information required to classify a tuple from D based on the splitting of A.

DEFINING INFORMATION GAIN

- Our objective is to take A on splitting to produce an exact classification (also called pure), that is, all tuples belong to one class.

- However, it is quite likely that the partitions is impure, that is, they contain tuples from two or more classes.

- In that sense, $E_A(D)$ is a measure of impurities (or purity). A lesser value of $E_A(D)$ implying more power the partitions are.

Definition 19.5: Information Gain

Information gain, $\alpha(A, D)$ of the training set D splitting on the attribute A is given by

$$\alpha(A, D) = E(D) - E_A(D)$$

In other words, $\alpha(A, D)$ gives us an estimation how much would be gained by splitting on A. The attribute A with the highest value of α should be chosen as the splitting attribute for D.

INFORMATION GAIN CALCULATION

Example 19.11 : Information gain on splitting OPTH

- Let us refer to the OPTH database discussed earlier.
- Splitting on **Age** at the root level, it would give three subsets D_1, D_2 and D_3 as shown in the tables in the following three slides.
- The entropy $E(D_1), E(D_2)$ and $E(D_3)$ of training sets D_1, D_2 and D_3 and corresponding weighted entropy $E_{Age}(D_1)$, $E_{Age}(D_2)$ and $E_{Age}(D_3)$ are also shown alongside.
- The Information gain $\alpha\ (Age, OPTH)$ is then can be calculated as **0.0394**.
- Recall that entropy of OPTH data set, we have calculated as *E(OPTH)* = **1.3261**

  (see Slide #17)

INFORMATION GAIN CALCULATION

Example 19.11 : Information gain on splitting OPTH

Training set: D_1(Age = 1)

Age	Eye-sight	Astigmatism	Use type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1

$$E(D_1) = -\frac{2}{8}log_2(\frac{2}{8}) - \frac{2}{8}log_2(\frac{2}{8}) - \frac{4}{8}log_2(\frac{4}{8}) = \mathbf{1.5}$$

$$E_{Age}(D_1) = \frac{8}{24} \times 1.5 = \mathbf{0.5000}$$

CALCULATING INFORMATION GAIN

Training set: $D_2(\text{Age} = 2)$

Age	Eye-sight	Astigmatism	Use type	Class
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3

$$E(D_2) = -\frac{1}{8}log_2(\frac{1}{8}) - \frac{2}{8}log_2(\frac{2}{8}) - \frac{5}{8}log_2(\frac{5}{8})$$
$$= \mathbf{1.2988}$$

$$E_{Age}(D_2) = \frac{8}{24} \times 1.2988 = \mathbf{0.4329}$$

CALCULATING INFORMATION GAIN

Training set: D_3(Age = 3)

Age	Eye-sight	Astigmatism	Use type	Class
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

$$E(D_3) = -\frac{1}{8}log_2(\frac{1}{8}) - \frac{1}{8}log_2(\frac{1}{8}) - \frac{6}{8}log_2(\frac{6}{8}) = \mathbf{1.0613}$$

$$E_{Age}(D_3) = \frac{8}{24} \times 1.0613 = \mathbf{0.3504}$$

$$\alpha\,(Age, D) = 1.3261 - (0.5000 + 0.4329 + 0.3504) = \mathbf{0.0394}$$

INFORMATION GAINS FOR DIFFERENT ATTRIBUTES

- In the same way, we can calculate the information gains, when splitting the OPTH database on Eye-sight, Astigmatic and Use Type. The results are summarized below.
- Splitting attribute: Age

$$\alpha(Age, OPTH) = 0.0394$$

- Splitting attribute: Eye-sight

$$\alpha(Eye\_sight, OPTH) = 0.0395$$

- Splitting attribute: Astigmatic

$$\alpha(\text{Astigmatic}, OPTH) = 0.3770$$

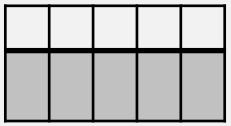
- Splitting attribute: Use Type

$$\alpha(\text{Use Type}, OPTH) = 0.5488$$

DECISION TREE INDUCTION : ID3 WAY

- The ID3 strategy of attribute selection is to choose to split on the attribute that gives the greatest reduction in the weighted average entropy
  - The one that maximizes the value of information gain
- In the example with OPTH database, the larger values of information gain is $\alpha(\text{Use Type}, OPTH) = 0.5488$
  - Hence, the attribute should be chosen for splitting is "Use Type".
- The process of splitting on nodes is repeated for each branch of the evolving decision tree, and the final tree, which would look like is shown in the following slide and calculation is left for practice.

DECISION TREE INDUCTION : ID3 WAY



$E(OPTH) = 1.3261$

$OPTH$

Age ✗ $\alpha = 0.0394$

Eye-sight ✗ $\alpha = 0.395$

Use Type ✓ $\alpha = 0.5488$

Astigmatic ✗ $\alpha = 0.3770$

$D1$ $Frequent(1)$

$E(D_1) = ?$

Age	Eye	Ast	Use	Class
1	1	1	1	3
1	1	2	1	3
1	2	1	1	3
1	2	2	1	3
2	1	1	1	3
2	2	1	1	3
2	2	2	1	3
3	1	1	1	3
3	1	2	1	3
3	2	1	1	3
3	2	2	1	3

? Age $\alpha = ?$

? Eye-sight $\alpha = ?$

? Astigmatic $\alpha = ?$

$D2$ $Less(2)$

$E(D_2) = ?$

Age	Eye	Ast	Use	Class
1	1	1	2	2
1	1	2	2	1
1	2	1	2	2
1	2	2	2	1
2	1	1	2	2
2	1	2	2	1
2	2	1	2	2
3	1	1	2	3
3	1	2	2	3
3	2	1	2	2
3	2	2	2	3

? Age $\alpha = ?$

? Eye-sight $\alpha = ?$

? Astigmatic $\alpha = ?$

FREQUENCY TABLE : CALCULATING A

- Calculation of entropy for each table and hence information gain for a particular split appears tedious (at least manually)!
- As an alternative, we discuss **a short-cut method** of doing the same using a special data structure called **Frequency Table.**
- **Frequency Table**: Suppose, $X = \{x_1, x_2, \ldots, x_n\}$ denotes an attribute with $n - different$ attribute values in it. For a given database D, there are a set of k $classes$ say $C = \{c_1, c_2, \ldots c_k\}$. Given this, a frequency table will look like as follows.

FREQUENCY TABLE : CALCULATING A

Class \ X	x_1	x_2	.....	x_i	.....	x_n
c_1			.....		.....	
c_2			.....		.....	
⋮	⋮	⋮	.....	⋮	.....	⋮
c_j			.....	f_{ij}	.....	
⋮	⋮	⋮	.....	⋮	.....	⋮
c_k			.....		.....	

- Number of rows = Number of classes
- Number of columns = Number of attribute values
- f_{ij} = Frequency of x_i for class c_j

Assume that $|D| = N$, the number of total instances of D.

CALCULATION OF A USING FREQUENCY TABLE

Example 19.12 : OTPH Dataset

With reference to OPTH dataset, and for the attribute Age, the frequency table would look like

	Age=1	Age=2	Age=3	Row Sum
Class 1	2	1	1	4
Class 2	2	2	1	5
Class 3	4	5	6	15
Column Sum	8	8	8	24

N=24

Column Sums

CALCULATION OF A USING FREQUENCY TABLE

- The weighted average entropy $E_X(D)$ then can be calculated from the frequency table following the
  - Calculate $V = f_{ij} \log_2 f_{ij}$ for all $i = 1,2, \dots, k$ *(Entry Sum)* $j = 1, 2, \dots, n$ and $v_{ij} \neq 0$
  - Calculate $S = s_i \log_2 s_i$ for all $i = 1, 2, \dots, n$ *(Column Sum)* in the row of column sum
  - Calculate $E_X(D) = (-V + S)/N$

Example 19.13: OTPH Dataset

For the frequency table in **Example 18.12**, we have

$$V = 2\log 2 + 1\log 1 + 1\log 1 + 2\log 2 + 2\log 2 + 1\log 1 + 4\log 4 + 5\log 5 + 6\log 6$$
$$S = 8\log 8 + 8\log 8 + 8\log 8$$

$$\boldsymbol{E_{Age}(OPTH) = 1.2867}$$

PROOF OF EQUIVALENCE

- In the following, we prove the equivalence of the short-cut of entropy calculation using Frequency Table.
- Splitting on an attribute A with n values produces n subsets of the training dataset D (of size $|D| = N$). The $j-th$ subset ($j = 1,2, \dots, n$) contains all the instances for which the attribute takes its $j-th$ value. Let N_j denotes the number of instances in the $j-th$ subset. Then

$$\sum_{j=1}^{n} N_j = N$$

- Let f_{ij} denotes the number of instances for which the classification is c_i and attribute A takes its $j-th$ value. Then

$$\sum_{i=1}^{k} f_{ij} = N_j$$

PROOF OF EQUIVALENCE

Denoting E_j as the entropy of the $j-th$ subset, we have

$$E_j = -\sum_{i=1}^{k} \frac{f_{ij}}{N_j} \log_2 \frac{f_{ij}}{N_j}$$

Therefore, the weighted average entropy of the splitting attribute A is given by

$$E_A(D) = \sum_{j=1}^{n} \frac{N_j}{N} . E_j$$

$$= -\sum_{j=1}^{n} \sum_{i=1}^{k} \frac{N_j}{N} . \frac{f_{ij}}{N_j} . \log_2 \frac{f_{ij}}{N_j}$$

$$= -\sum_{j=1}^{n} \sum_{i=1}^{k} \frac{f_{ij}}{N_j} . \log_2 \frac{f_{ij}}{N_j}$$

PROOF OF EQUIVALENCE

$$= -\sum_{j=1}^{n}\sum_{i=1}^{k}\frac{f_{ij}}{N_j}.\log_2 f_{ij} + \sum_{j=1}^{n}\sum_{i=1}^{k}\frac{f_{ij}}{N_j}.\log_2 N_j$$

$$= -\sum_{j=1}^{n}\sum_{i=1}^{k}\frac{f_{ij}}{N_j}.\log_2 f_{ij} + \sum_{j=1}^{1}\frac{N_j}{N}.\log_2 N_j$$

$$\because \sum_{i=1}^{k} f_{ij} = N_j$$

$$= \left(-\sum_{j=1}^{n}\sum_{i=1}^{k} f_{ij}.\log_2 f_{ij} + \sum_{j=1}^{n} N_j \log_2 N_j\right)\Big/ N$$

$$= (-V + S)/N$$

where $V = \sum_{j=1}^{n}\sum_{i=1}^{k} f_{ij}.\log_2 f_{ij}$ (Entries sum)

$and\ S = \sum_{j=1}^{n} N_j \log_2 N_j$ $(Column\ Sum)$

Hence, the equivalence is proved.

LIMITING VALUES OF INFORMATION GAIN

- The Information gain metric used in ID3 always should be positive or zero.
- It is always positive value because information is always gained (i.e., purity is improved) by splitting on an attribute.
- On the other hand, when a training set is such that if there are k classes, and the entropy of training set takes the largest value i.e., $\log_2 k$ (this occurs when the classes are balanced), then the information gain will be zero.

LIMITING VALUES OF INFORMATION GAIN

Example 19.14: Limiting values of Information gain

Consider a training set shown below.

Data set *Table A*

X	Y	Class
1	1	A
1	2	B
2	1	A
2	2	B
3	2	A
3	1	B
4	2	A
4	1	B

X *Table X*

	1	2	3	4
A	1	1	1	1
B	1	1	1	1
$C.Sum$	2	2	2	2

Frequency table of X

Y *Table Y*

	1	2
A	2	2
B	2	2
$C.Sum$	4	4

Frequency table of Y

LIMITING VALUES OF INFORMATION GAIN

- Entropy of Table A is

$$E = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = \log 2 = 1 \text{ (The maximum entropy).}$$

- In this example, whichever attribute is chosen for splitting, each of the branches will also be balanced thus each with maximum entropy.
- In other words, information gain in both cases (i.e., splitting on *X* as well as *Y*) will be zero.

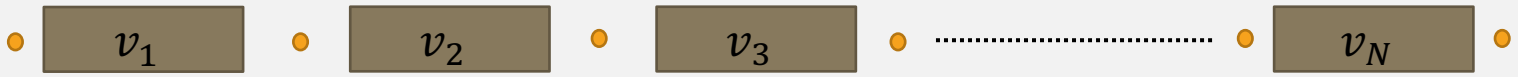
Note:

- The absence of information gain does not imply that there is no profit for splitting on the attribute.
- Even if it is chosen for splitting, ultimately it will lead to a final decision tree with the branches terminated by a leaf node and thus having an entropy of zero.
- Information gain can never be a negative value.

SPLITTING OF CONTINUOUS ATTRIBUTE VALUES

- In the foregoing discussion, we assumed that an attribute to be splitted is with a finite number of discrete values. Now, there is a great deal if the attribute is not so, rather it is a continuous-valued attribute.
- There are two approaches mainly to deal with such a case.

1. **Data Discretization**: All values of the attribute can be discretized into a finite number of group values and then split point can be decided at each boundary point of the groups.

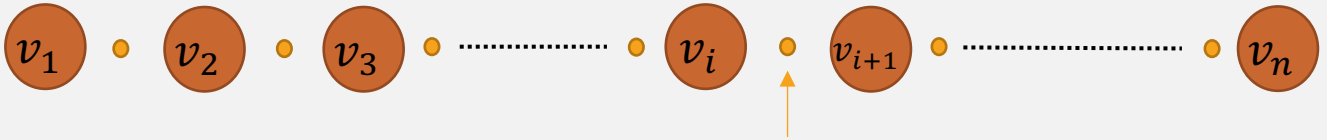


So, if there are $n - groups$ of discrete values, then we have $(n + 1)$ split points.

SPLITTING OF CONTINUOUS ATTRIBUTE VALUES

2. Mid-point splitting: Another approach to avoid the data discretization.

- It sorts the values of the attribute and take the distinct values only in it.
- Then, the mid-point between each pair of adjacent values is considered as a split-point.



- Here, if n-distinct values are there for the attribute A, then we choose $n-1$ split points as shown above.
- For example, there is a split point $s = \frac{v_i + v_{(i+1)}}{2}$ in between v_i and $v_{(i+1)}$
- For each split-point, we have two partitions: $A \leq s$ *and* $A > s$, and finally the point with maximum information gain is the desired split point for that attribute.

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?