Homework 4 : C4.5

Backgrounds of ID3 and C4.5

Jihoon Yang

Machine Learning Research Laboratory
Department of Computer Science & Engineering
Sogang University
Email: yangjh@sogang.ac.kr

Contents

- What is Tree ?
  - Tree Machine Learning Algorithm ?
  - Representation of Tree from Tabular Data
- Backgrounds of Information Theory
  - Entropy
  - Information Gain
- Learning Decision Tree (ID3) Classifier Process
- C4.5: An Advanced Version of ID3
  - Differences from ID3
  - Pseudo Code

What is Tree ?
- Tree Machine Learning Algorithm ?

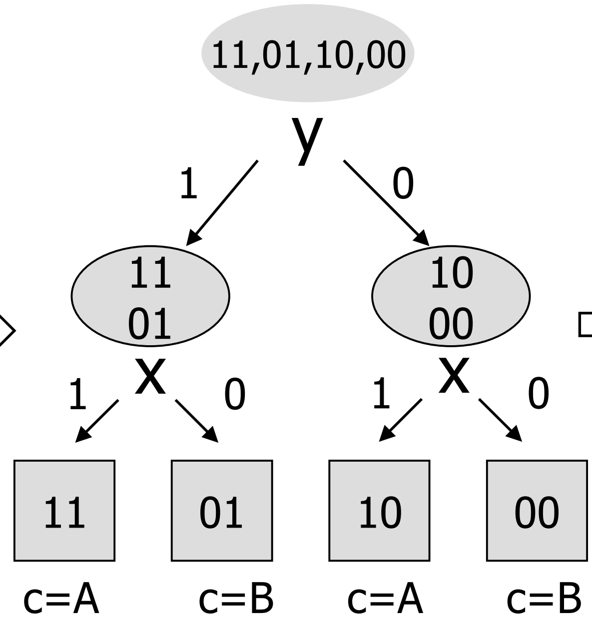
- **Tree-based ML is one of Supervised Learning algorithms.**
- **It is very similar to 'twenty questions' !!!**

Attributes Class

Examples

	x	y	c
1	1	1	A
2	0	1	B
3	1	0	A
4	0	0	B

Train Dataset



Learnt Tree Algorithm by the train dataset

x	y	Class
1	1	?

Test Dataset

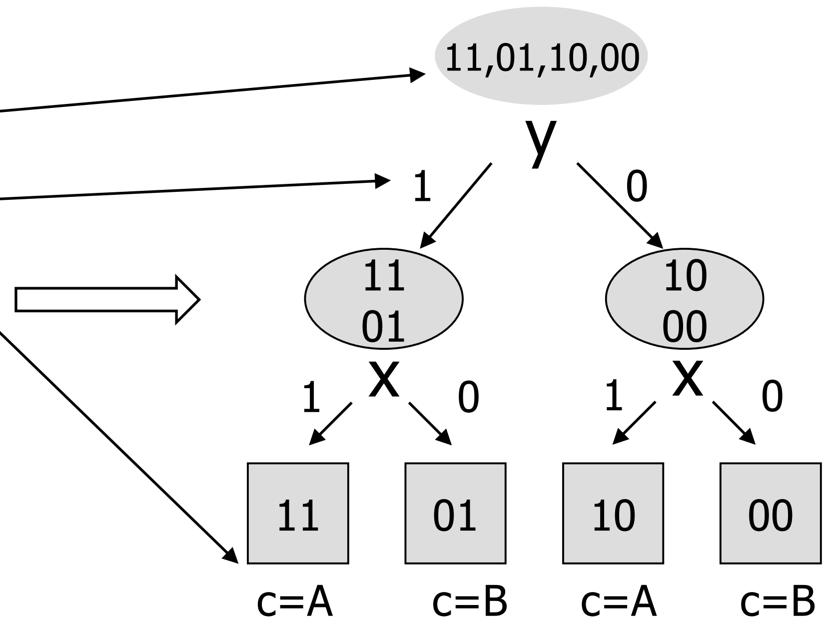
What is Tree ?
- Representation of Tree from Tabular Data

- **How to represent tree with the tabular-formed dataset ?**

Attributes (x, y) | Class (c)

Examples

	x	y	c
1	1	1	A
2	0	1	B
3	1	0	A
4	0	0	B



Each attribute → Each node
Each value of attribute → Each branch
Each class → Each leaf node

SOGANG UNIVERSITY

Backgrounds of Information Theory - Entropy

- Suppose we have a message that conveys the result of a random experiment with *m* possible discrete outcomes, with probabilities

$$p_1, p_2, \ldots, p_m$$

- The expected information content of such a message is called the entropy of the probability distribution

$$H(p_1, p_2, \ldots, p_m) = \sum_{i=1}^{m} p_i I(p_i)$$

$$I(p_i) = -\log_2 p_i \text{ provided } p_i \neq 0$$

$$I(p_i) = 0 \text{ otherwise}$$

Backgrounds of Information Theory - Entropy

Let $\vec{P} = (p_1 \; .... \; p_n)$ be a discrete probability distribution

The entropy of the distribution P is given by

$$H(\vec{P}) = \sum_{i=1}^{n} p_i \log_2\left(\frac{1}{p_i}\right) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

$$H\left(\frac{1}{2},\frac{1}{2}\right) = -\sum_{i=1}^{2} p_i \log_2(p_i) = -\left(\frac{1}{2}\right)\log_2\left(\frac{1}{2}\right) - \left(\frac{1}{2}\right)\log_2\left(\frac{1}{2}\right) = 1\,bit$$

$$H(0,1) = -\sum_{i=1}^{2} p_i \log_2(p_i) = -1I(1) - 0I(0) = 0\,bit$$

- **Which would be better: High entropy ? Low entropy ? And… Why ?**

SOGANG UNIVERSITY

Backgrounds of Information Theory - Information Gain

- The expected information gain is the change in information entropy H from a prior state to a state that takes some information as given:

$$IG(T,a) = H(T) - H(T\,|a)$$

, where $H(T\,|a)$ is the conditional entropy of T given the value of attribute a.

- **Which would be better: High IG ? Low IG ? And… Why ?**

Learning Decision Tree (ID3) Classifier Process

- How ID3 learns from the train dataset ?
- For example,

Instances –

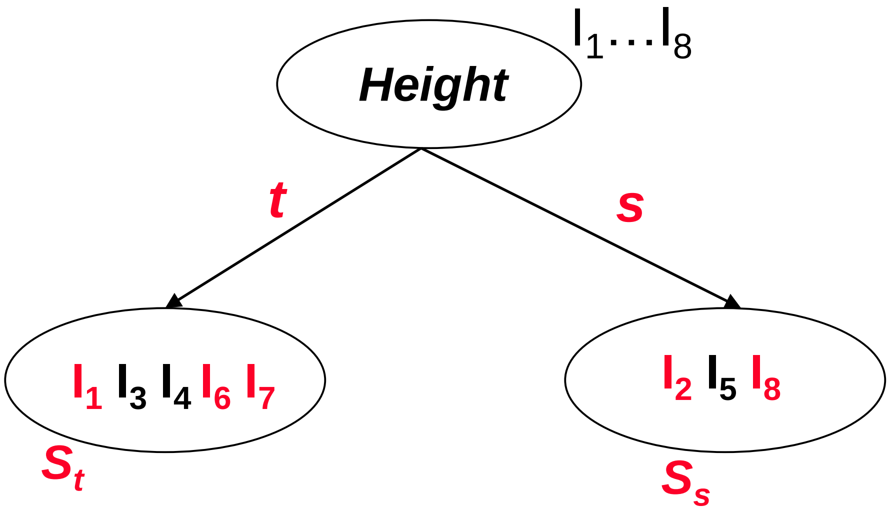
ordered 3-tuples of attribute values corresponding to

Height **(tall, short)**
Hair **(dark, blonde, red)**
Eye **(blue, brown)**

Training Data

Instance	Class label
I_1 (t, d, l)	+
I_2 (s, d, l)	+
I_3 (t, b, l)	–
I_4 (t, r, l)	–
I_5 (s, b, l)	–
I_6 (t, b, w)	+
I_7 (t, d, w)	+
I_8 (s, b, w)	+

Learning Decision Tree (ID3) Classifier Process



$$\hat{H}(X) = -\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8} = 0.954 bits$$

$$\hat{H}(X \mid Height = t) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 bits$$

$$\hat{H}(X \mid Height = s) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918 bits$$

$$\hat{H}(X \mid Height) = \frac{5}{8}\hat{H}(X \mid Height = t) + \frac{3}{8}\hat{H}(X \mid Height = s) = \frac{5}{8}(0.971) + \frac{3}{8}(0.918) = 0.95 bits$$

Similarly, $\hat{H}(X \mid Hair) = \frac{3}{8}\hat{H}(X \mid Hair = d) + \frac{4}{8}\hat{H}(X \mid Hair = b) + \frac{1}{8}\hat{H}(Hair = r) = 0.5 bits$ and

$$\hat{H}(X \mid Eye) = 0.607 bits$$

Hair is the most informative because it yields the largest reduction in entropy; Thus, we choose it as a root node !!!

Learning Decision Tree (ID3) Classifier Process

- The task of the learner then is to extract the needed information from the training set and store it in the form of a decision tree for classification

- ***Information gain* based decision tree learner**

Start with the entire training set at the root

Recursively add nodes to the tree

corresponding to tests that yield the greatest expected reduction in entropy (or the largest expected information gain)

until some termination criterion is met

(e.g. the training data at every leaf node has zero entropy)

C4.5: An Advanced Version of ID3 - Differences from ID3

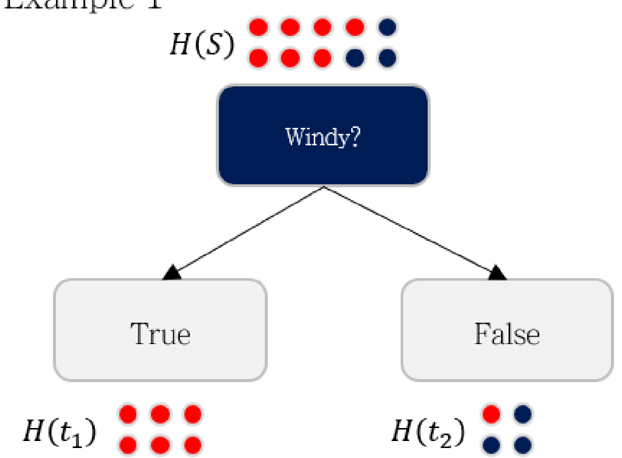
- **Normalized Information Gain**
- **Numerical Values**
- **Missing Values →** we do not consider this case in this Homework…
- **Prunning to prevent overfitting**

C4.5: An Advanced Version of ID3 - Differences from ID3

- **Normalized Information Gain**

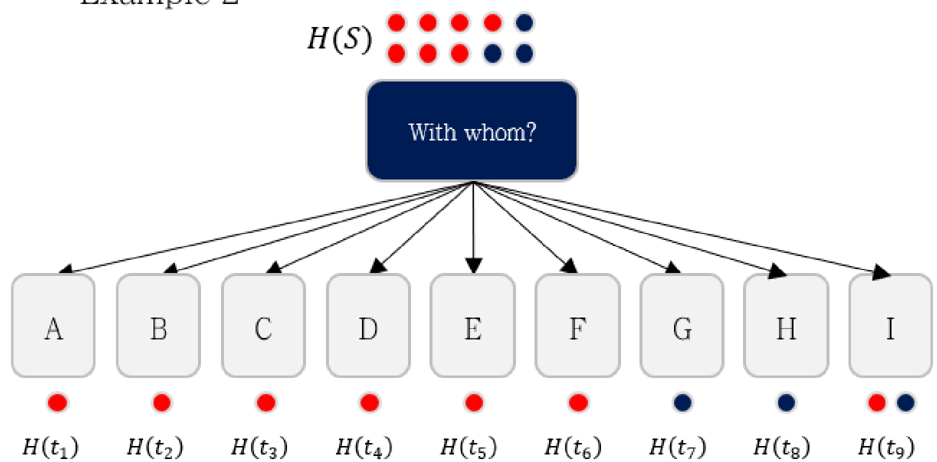
  - Why ?: Limitation of Information Gain

Example 1



$$
\begin{aligned}
Information\ Gain &= H(7,3) - (\frac{6}{10}H(6,0) + \frac{4}{10}H(1,3)) \\
&= 0.8813 - (\frac{6}{10} \times 0 + \frac{4}{10} \times 0.8113) \\
&= 0.5568
\end{aligned}
$$

Example 2



$$
\begin{aligned}
Information\ Gain &= H(7,3) - (\frac{1}{10}H(1,0) + \frac{1}{10}H(1,0) + \ldots + \frac{1}{10}H(0,1) + \frac{2}{10}H(1,1)) \\
&= 0.8813 - (\frac{1}{10} \times 0 + \ldots + \frac{2}{10} \times 1) \\
&= 0.6813
\end{aligned}
$$

reference : https://tyami.github.io/machine%20learning/decision-tree-3-c4_5/#%EA%B0%80%EC%A7%80%EC%B9%98%EA%B8%B0-pruning

C4.5: An Advanced Version of ID3 - Differences from ID3

- **Normalized Information Gain**
  - Thus, ‘Information Gain Ratio’

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv - \sum_{i=1}^{|Values(A)|} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

C4.5: An Advanced Version of ID3 - Differences from ID3

- **Numerical Values**

Attribute T	40	48	50	54	60	70
Class	N	N	Y	Y	Y	N

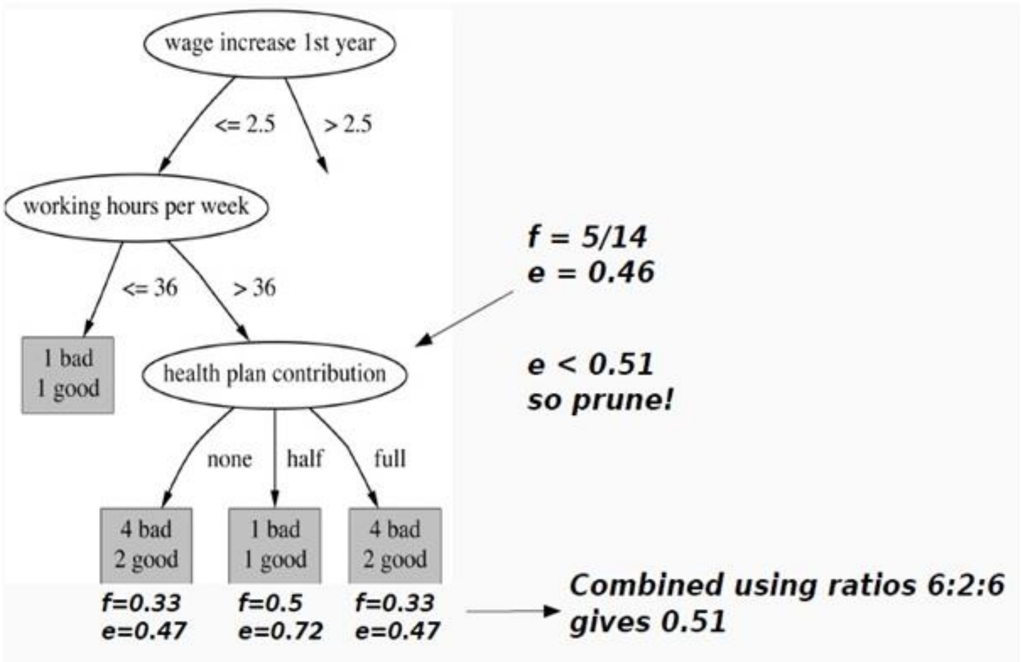
Candidate splits $T > \frac{(48+50)}{2}?$ $\quad T > \frac{(60+70)}{2}?$

$$E(S \mid T > 49?) = \frac{2}{6}(0) + \frac{4}{6}\left(-\left(\frac{3}{4}\right)\log_2\left(\frac{3}{4}\right) - \left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right)\right)$$

- Sort instances by value of numeric attribute under consideration
- For each attribute, find the test which yields the lowest entropy
- Greedily choose the best test across all attributes

C4.5: An Advanced Version of ID3 - Differences from ID3

- **Prunning to prevent overfitting**
  - Post-pruning : Conducting pruning after tree is completed



$$e = \frac{f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

$N = sample\ size$

$f = Error\ rate$

$z = z\ score\ (default\ z = 0.69)$

If parent node's e is smaller than weighted sum of the childrens' e, then do prune the parent node !!

C4.5: An Advanced Version of ID3 - Pseudo Code

SOGANG UNIVERSITY

Algorithm 2: C4.5 Algorithm

1. Check for **base cases**.
2. For each attribute *a*
   find the **normalized information gain** from splitting on *a*.
3. Let *a_best* be the attribute with the
   highest normalized information gain.
4. Create a **decision node** that splits on *a_best*.
5. Recur on the sublists obtained by splitting on *a_best*, and add
   those nodes as children of **node**.

- **Base cases → You don't have to worry about them !!!**
  - All the samples in the list belong to the same class. (→ Most cases !)
  - None of the features provide any information gain.
  - Instance of previously-unseen class encountered.

Guidelines for This Homework

- **The formats of train.txt is:**

<train 데이터 개수> <numeric attribute의 개수> <categoric attribute의 개수>
<첫번째 데이터의 첫번째 feature값> … <첫번째 데이터의 마지막 feature값> <첫번째 데이터의 class>
<두번째 데이터의 첫번째 feature값> … <두번째 데이터의 마지막 feature값> <두번째 데이터의 class>
…
<마지막 데이터의 첫번째 feature값> … <마지막 데이터의 마지막 feature값> <마지막 데이터의 class>

이 때, numeric attribut와 categorical attribute의 data type은 int이다. 그리고 순차적으로 numeric attribute와 categorical attribute가 있으며 class와 categoric attribute는 0 또는 1 값만을 갖는다. (즉 binary classification이며 각 node의 branch는 2개이다.)

Guidelines for This Homework

- **The formats of test.txt is:**

<test 데이터 개수> <numeric attribute의 개수> <categoric attribute의 개수>

<첫번째 test 데이터의 첫번째 feature값> … <첫번째 test 데이터의 마지막 feature값>

<두번째 test 데이터의 첫번째 feature값> … <두번째 test 데이터의 마지막 feature값>

…

<마지막 test 데이터의 첫번째 feature값> … <마지막 test 데이터의 마지막 feature값>

자료형과 attribute의 개수 및 순서는 train.txt와 같다.

Guidelines for This Homework

- **The formats of results.txt must be:**

<첫번째 test 데이터의 첫번째 feature값> … <첫번째 test 데이터의 마지막 feature값> <첫번째 data의 예측 class>

<두번째 test 데이터의 첫번째 feature값> … <두번째 test 데이터의 마지막 feature값> <두번째 data의 예측 class>

…

<마지막 test 데이터의 첫번째 feature값> … <마지막 test 데이터의 마지막 feature값> <마지막 data의 예측 class>

Guidelines for This Homework

- 과반수에서 개수가 같을 경우에는, label 0과 1중 0으로 leaf 노드를 생성한다.
- 구현 과정에서 중요하다고 생각하는 부분에 대한 설명과 sample run에 대한 내용이 담긴 보고서를 제출한다. ('어려웠지만 재미있었다.', '즐거웠다.', '힘들었지만 보람찼다.'와 같은 표현 금지)
- C4_5_학번.c, C4_5_학번.pdf 파일을 사이버캠퍼스에서 제출한다.

Due Date : 06. 02. 23:59