Lab 6

Decision Tree

CS429 - Introduction to Big Data Analysis

Lab Instructor: Nguyễn Đình Thảo

Table of Contents

Building a Decision Tree

Pseudocode

Algorithm: *build_tree*()
Input: training examples E , attributes A
Output: decision tree T

```
if E is empty
    return failure
end if
if A is empty
    return a leaf node with the class label
end if
If all examples in E have same class label
    return a leaf node with that class label
end if

find attribute a_j ∈ A with highest Information Gain as split node
create a tree T_{a_j} with node a_j as split node

for each value v_i of attribute a_j do
    T_{v_i} = build_tree( E_{a_j,v_i} , A )
    add T_{v_i} as child of T_{a_j}
end for

return tree T_{a_j}
```

Finding the best split node

There are a variety of impurity measures for finding the best split node in a decision tree. In this assignment, we use $Information\ Gain$, which is computed as follows.
Given a set of training examples E , set of attributes $A = \{a_1, a_2, ..., a_n\}$, set of class label $C = \{c_1, c_2, ..., c_m\}$,
Information Gain of an attribute a_j can be computed as:

$$IGain(E, a_j) = Entropy(E) - \sum_{v \in Values(a_j)} \frac{|E_v|}{|E|} . Entropy(E_v)$$

where

- $Values(a_j)$ is the set of all possible values of attribute a_j
- E_v is set of training examples which has value v for attribute a_j
- $|E|, |E_v|$ are the number of training examples in E and E_v
- $Entropy(E) = -\sum_{i=1}^{|C|} p_i log_2(p_i)$ is the entropy before splitting
  - $|C|$ number of class labels
  - p_i is the fraction of training examples with class label c_i
- $Entropy(E_v)$ is the same as $Entropy(E)$ but computed on E_v only

Example

Given sample data as below

Outlook	Temperature	Humidity	Wind	Play Golf
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

We have:

- $A = \{Outlook, Temperature, Humidity, Wind\}$

- $C = \{Yes, No\}$ as set of class labels, i.e. to play golf or not

Now, select the first attribute to split using $Information\ Gain$:

$$Entropy(E) = -\sum_{i=1}^{2} p_i log_2(p_i) = -\sum_{yes} p_{yes} log_2(p_{yes}) - \sum_{no} p_{no} log_2(p_{no}) = -(\frac{9}{14}) log_2(\frac{9}{14}) - (\frac{5}{14}) log_2(\frac{5}{14}) = 0.94$$

$$IGain(E,\ Outlook) = Entropy(E) - \frac{|E_{Sunny}|}{|E|} Entropy(E_{Sunny}) - \frac{|E_{Overcast}|}{|E|} Entropy(E_{Overcast}) - \frac{|E_{Rain}|}{|E|} Entropy(E_{Rain})$$
$$= 0.94 - \frac{5}{14}(-\frac{2}{5}log_2(\frac{2}{5}) - \frac{3}{5}log_2(\frac{3}{5})) - \frac{4}{14}(-\frac{4}{4}log_2(\frac{4}{4}) - \frac{0}{4}log_2(\frac{0}{4})) - \frac{5}{14}(-\frac{3}{5}log_2(\frac{3}{5}) - \frac{2}{5}log_2(\frac{2}{5}))$$
$$= 0.246$$

Similarly, we have:

$IGain(E,\ Temperature) = 0.029$

$IGain(E,\ Humidity) = 0.151$

$IGain(E,\ Wind) = 0.048$

Hence, $Outlook$ should be chosen as the first split node.

Next for each value of $Outlook$, we need to repeat the above computation with the set of attribute $\{Temperature,\ Humidity,\ Wind\}$.

$$Entropy(E_{Sunny}) = -\sum_{i=1}^{2} p_i log_2(p_i) = -\sum_{yes} p_{yes} log_2(p_{yes}) - \sum_{no} p_{no} log_2(p_{no}) = -(\frac{2}{5}) log_2(\frac{2}{5}) - (\frac{3}{5}) log_2(\frac{3}{5}) = 0.97$$

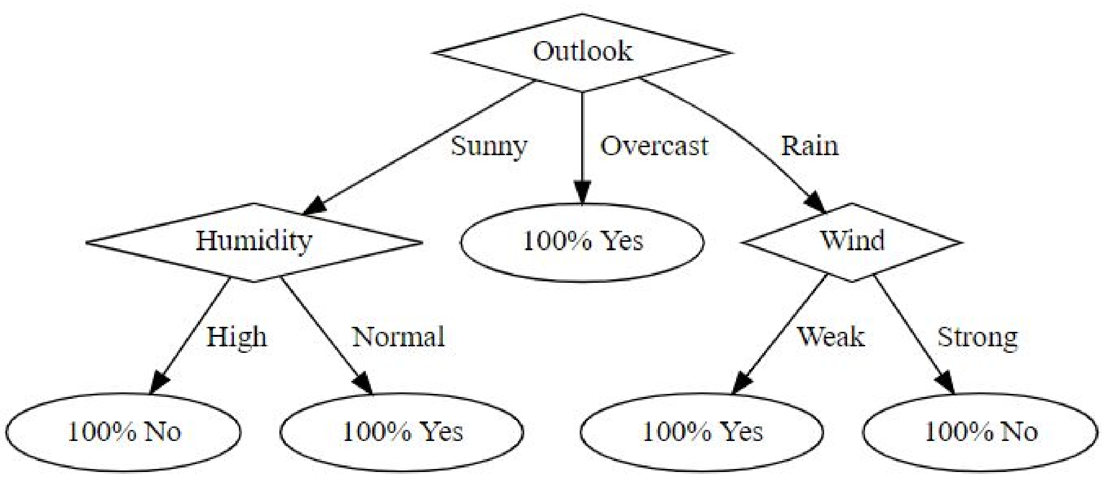
$IG(E_{Sunny},\ Temperature) = 0.57$

$IG(E_{Sunny},\ Humidity) = 0.96$

$IG(E_{Sunny},\ Wind) = 0.019$

Hence, the next split node on $Outlook = Sunny$ branch should be $Humidity$.

Keep going on with the computation and the final decision tree will look like this



Implementation

You need to implement the provided pseudocode of the decision tree algorithm based on the jupyter notebook **Lab 6.ipynb**. Assuming that the training dataset is too large to be stored in memory, all computations have to be performed distributedly using PySpark. Only the decision tree is stored and visualized on the local machine.

Input

Training data can be found in file ***golf.data.*** Each line is a training example.

Output

A complete decision tree, which

- Can be visualized
- Can predict on new examples

Submission

Submit your jupyter notebook with the naming format: **<your studentID>_lab6.ipynb**