



Decision Trees

Qinliang Su (苏勤亮)

Sun Yat-sen University


suqliang@mail.sysu.edu.cn

Outline

- Introduction
- Criteria to Choose Expanding Attributes
- Decision Tree Learning

What is Decision Tree?

- Given a dataset below, we want to predict the outcome based on the attributes



| | Type | Length | Director | Famous actors | Liked? |
|---|----------|--------|----------|---------------|--------|
| 1 | Comedy | Short | Adamson | No | Yes |
| 2 | Animated | Short | Lasseter | No | No |
| 3 | Drama | Medium | Adamson | No | Yes |
| 4 | Animated | Long | Lasseter | Yes | No |
| 5 | Comedy | Long | Lasseter | Yes | No |
| 6 | Drama | Medium | Singer | Yes | Yes |
| 7 | Animated | Short | Singer | No | Yes |
| 8 | Comedy | Long | Adamson | Yes | Yes |
| 9 | Drama | Medium | Lasseter | No | Yes |

- Different from the previous classifiers, decision tree classifies data into different categories by building a tree of attributes

- Structure of a decision tree (DT)

- **Internal nodes** correspond to attributes

- **Leaf nodes** correspond to the outcome

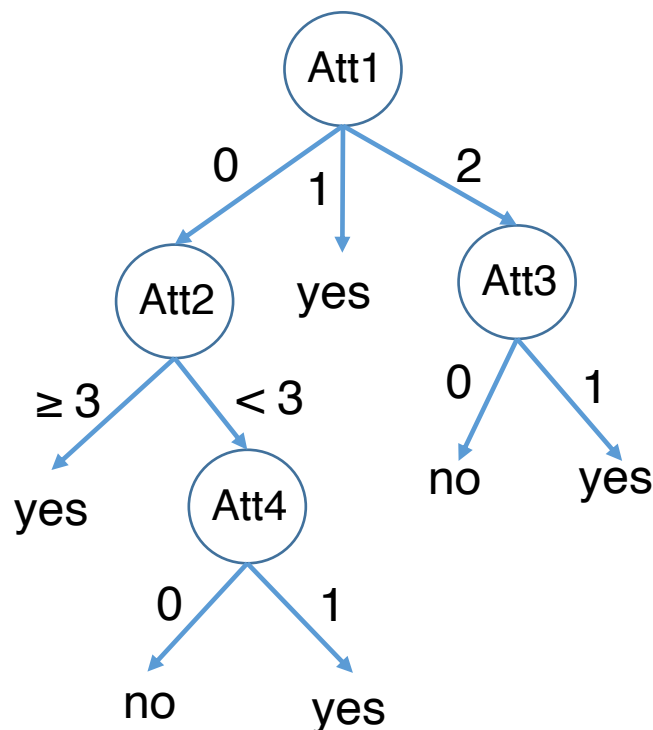
- **Edges** correspond to the attribute values

- Learning a decision tree

- Which attribute to use at each node?

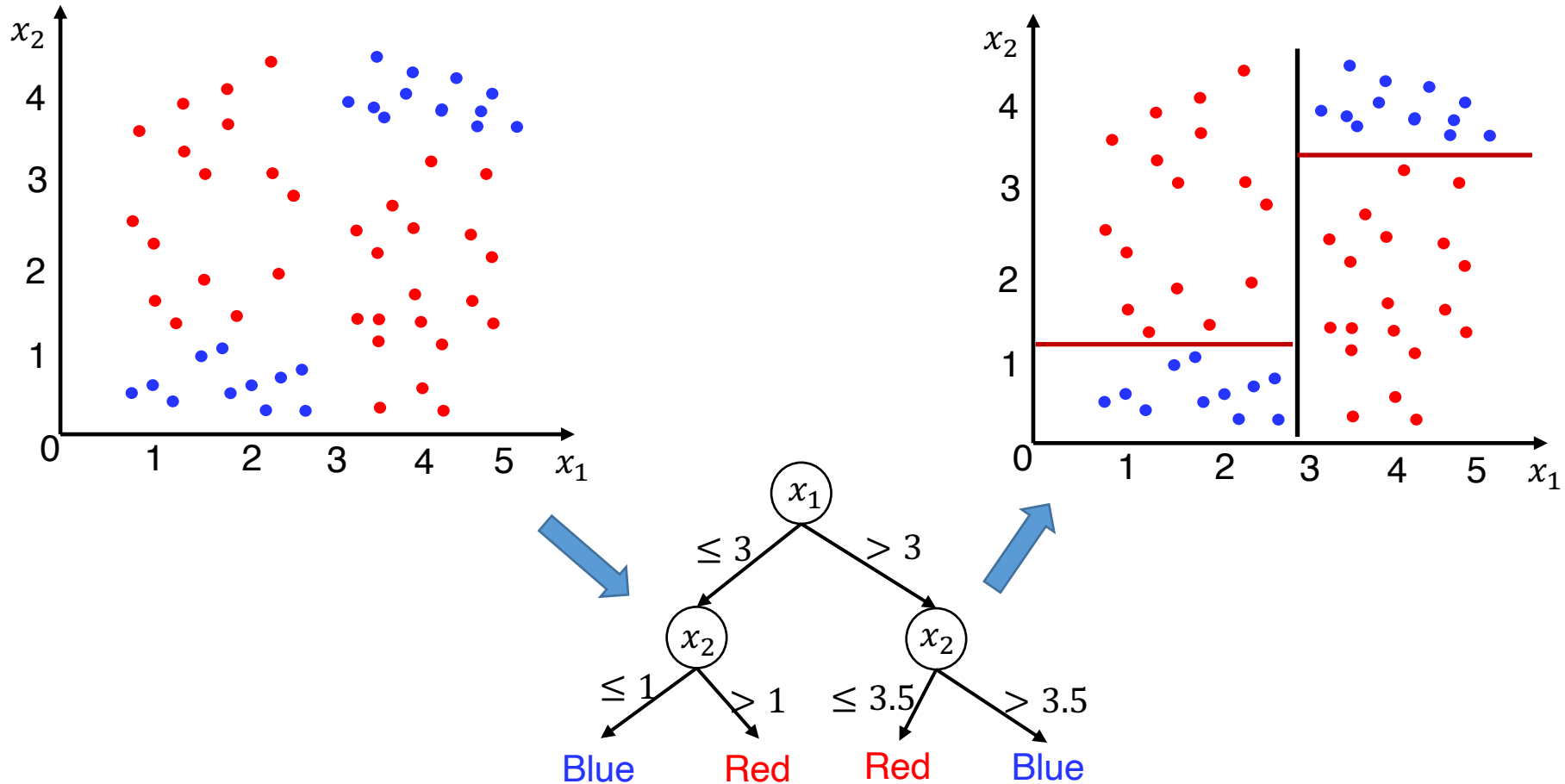
- How to partition the attribute values?

- When to stop expanding the tree?

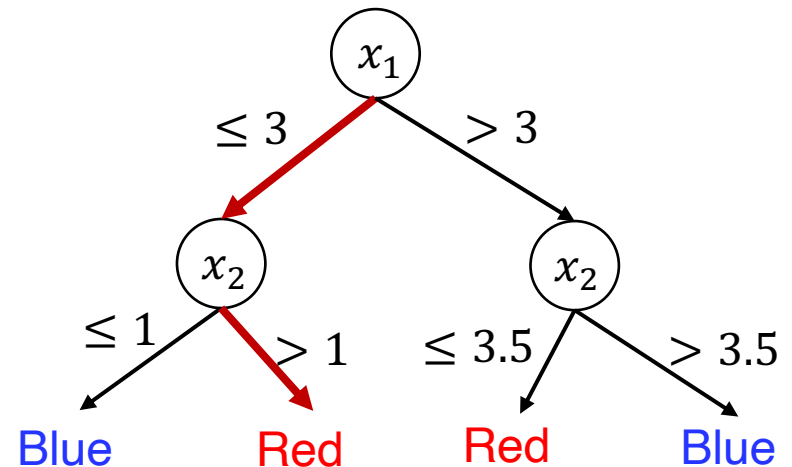
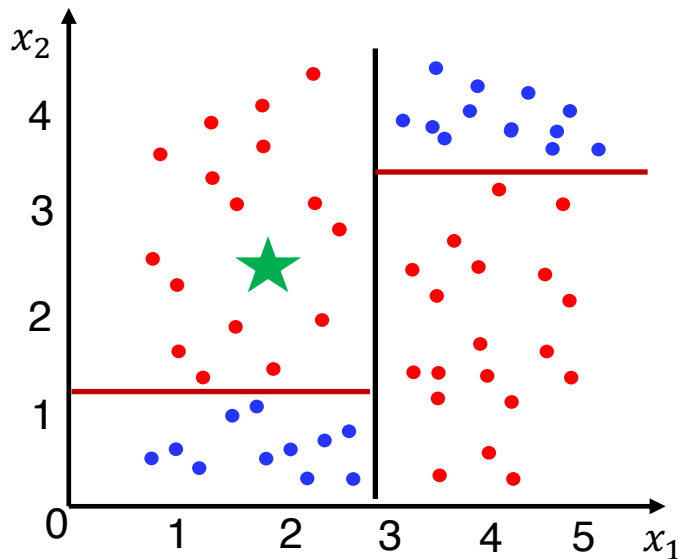


Decision Tree Learning: Example 1

- Given a dataset with two attributes x_1 and x_2 , as shown below, we want to build a decision tree to classify the instance



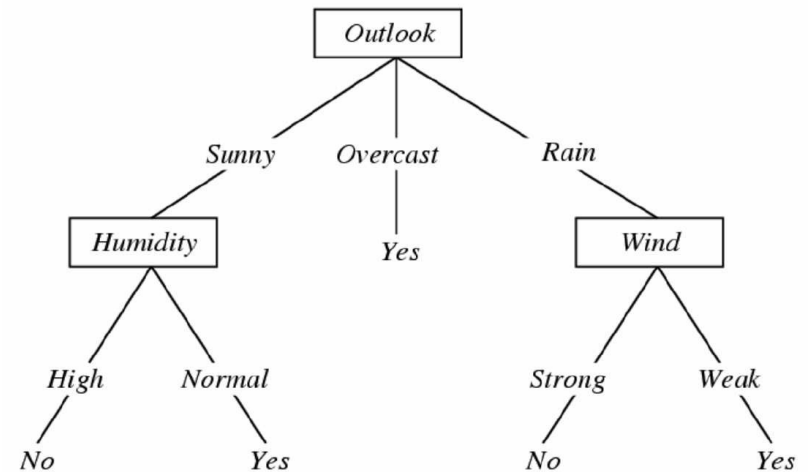
- From the example, to build the tree, we need to decide
 - which attribute to use at each step
 - how to partition the attribute value
- For a new coming data, a path from the root to the leaves can be found based on its attribute values, from which its category can be read out



Decision Tree Learning: Example 2

- Given a dataset below, build a decision tree to decide whether to play or not based on the four attribute values

| day | outlook | temperature | humidity | wind | play |
|-----|----------|-------------|----------|--------|------|
| 1 | sunny | hot | high | weak | no |
| 2 | sunny | hot | high | strong | no |
| 3 | overcast | hot | high | weak | yes |
| 4 | rain | mild | high | weak | yes |
| 5 | rain | cool | normal | weak | yes |
| 6 | rain | cool | normal | strong | no |
| 7 | overcast | cool | normal | strong | yes |
| 8 | sunny | mild | high | weak | no |
| 9 | sunny | cool | normal | weak | yes |
| 10 | rain | mild | normal | weak | yes |
| 11 | sunny | mild | normal | strong | yes |
| 12 | overcast | mild | high | strong | yes |
| 13 | overcast | hot | normal | weak | yes |
| 14 | rain | mild | high | strong | no |



- Questions**

- Why the outlook is chosen first, and then humidity and wind?
- Why not further expanding after the humidity and wind?
- Why the 'temperature' attribute does not appear in the tree?

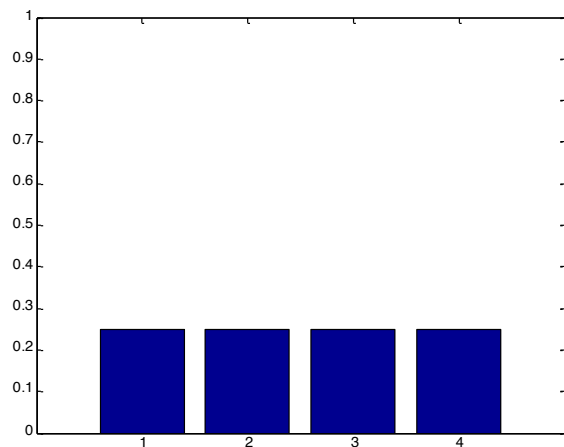
Outline

- Introduction
- Criteria to Choose Expanding Attributes
- Decision Tree Learning

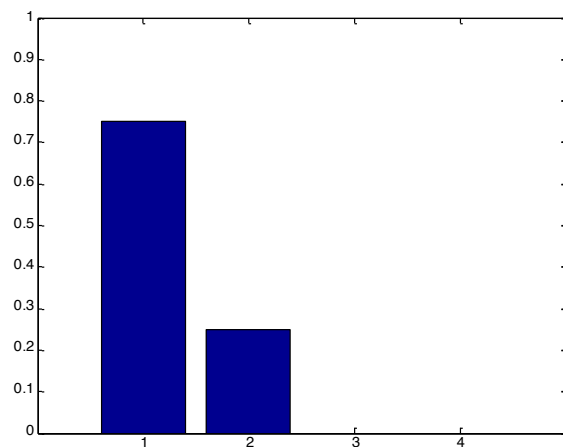
Entropy

- Among all attributes, choose the one that *contains most information* to expand
- How to measure the amount of information that an attribute contains?

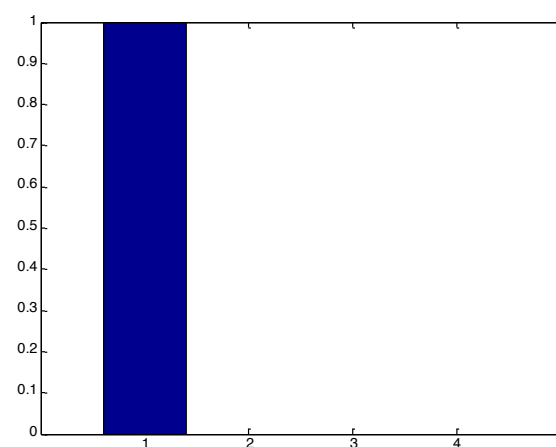
we need to measure *the amount of uncertainties* of a random variable first



(a)



(b)



(c)

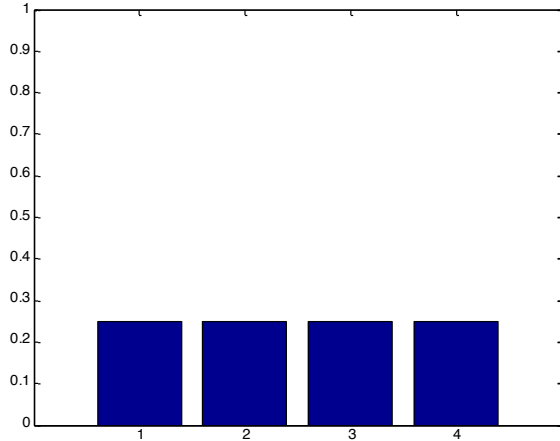
- The amount of uncertainties of a random variable in mathematics is measured by *entropy*
- **Definition:** Given a random variable Z that follows a distribution $p(z)$, its entropy is defined as

$$H(Z) = - \sum_{z \in \mathcal{C}} p(z) \log_2 p(z)$$

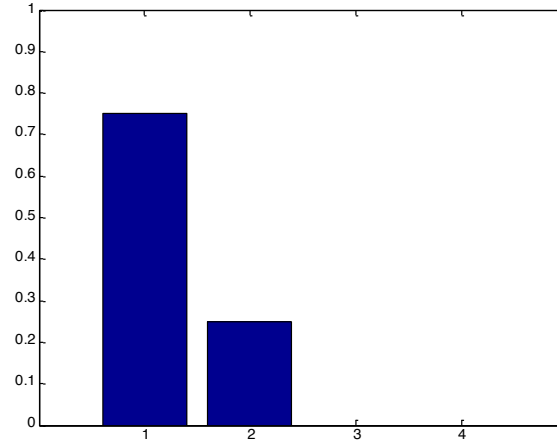
- \mathcal{C} is the set of possible values of random variable Z

- Example

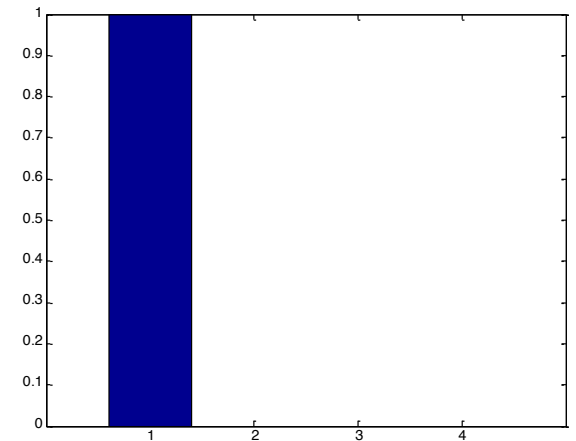
Which distribution below has the largest entropy?



(a)



(b)



(c)

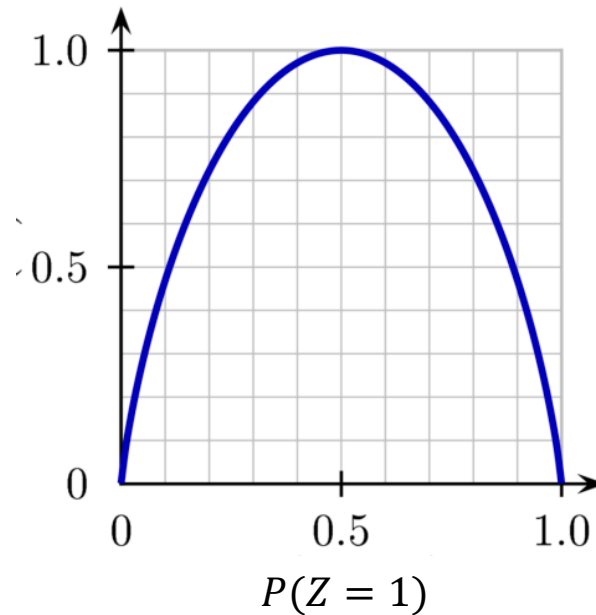
a) $H(Z) = -4 \times 0.25 \log_2 0.25 = 2$ bits

b) $H(Z) = -0.75 \log_2 0.75 - 0.25 \log_2 0.25 \approx 0.8133$ bits

c) $H(Z) = -1 \log_2 1 = 0$ bits

Distribution a) has the largest entropy, while c) has the smallest

- The entropy of a Bernoulli random variable as a function of the probability $P(Z = 1)$



- The entropy is consistent with our intuition, that is,

The more flat the distribution is, the larger the uncertainty will be

Conditional Entropy

- **Conditional entropy** $H(Z|Y)$: the entropy of random variable Z after knowing the values of random variable Y

$$H(Z|Y = y) = - \sum_{z \in \mathcal{C}} p(z|y) \log p(z|y)$$

$$H(Z|Y) = \sum_{y \in \mathcal{T}} P(Y = y) H(Z|Y = y)$$

| Y | Z |
|---|---|
| t | t |
| t | t |
| t | t |
| t | t |
| f | t |
| f | f |

➤ Example

$$p(Z = t|Y = t) = 1 \text{ and } p(Z = f|Y = t) = 0 \quad \Rightarrow \quad H(Z|Y = t) = 0$$

$$p(Z = t|Y = f) = 0.5 \text{ and } p(Z = f|Y = f) = 0.5 \quad \Rightarrow \quad H(Z|Y = f) = 1$$

$$p(Y = t) = 4/6 \text{ and } p(Y = f) = 2/6$$

$$\Rightarrow \quad H(Z|Y) = \frac{4}{6} \times 0 + \frac{2}{6} \times 1 = \frac{2}{6}$$

- The conditional entropy $H(Z|Y)$ is different from the entropy $H(Z)$

For the given example, it can be shown that

$$\begin{aligned} H(Z) &= -p(z = t) \log p(z = t) - p(z = f) \log p(z = f) \\ &= -\frac{5}{6} \log_2 \frac{5}{6} - \frac{1}{6} \log_2 \frac{1}{6} \\ &= -\frac{5}{6} \log_2 \frac{5}{6} - \frac{1}{6} \log_2 \frac{1}{6} \\ &\approx 0.65 \end{aligned}$$

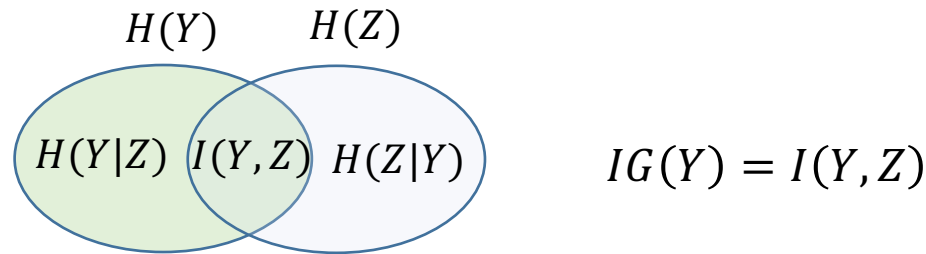
Obviously, it is larger than the conditional entropy $H(Z|Y) \approx 0.33$

Actually, the inequality $H(Z) \geq H(Z|Y)$ always holds

Information Gain

- The information gain of a random variable Y is the *amount of decreased entropy* of Z after knowing its values

$$IG(Y) = H(Z) - H(Z|Y)$$



- As for the example given above, the information gain of random variable Y is

$$IG(Y) = 0.65 - 0.33 = 0.32$$

- The information gain of Y means *the amount of uncertainties that can be reduced on average* if its value is known

Outline

- Introduction
- Criteria to Choose Expanding Attributes
- Decision Tree Learning

Choosing the Root Node

- The *entropy* of the outcome variable 'liked'

$$P(\text{Like} = \text{yes}) = 2/3 \text{ and } P(\text{Like} = \text{no}) = 1/3 \quad \Rightarrow \quad H(\text{Like}) = 0.91$$

- The *conditional entropy* of the outcome given attributes Type, Length, Director and Actors

$$H(\text{Like}|\text{Type}) = 0.61$$

$$H(\text{Like}|\text{Length}) = 0.61$$

$$H(\text{Like}|\text{Director}) = 0.36$$

$$H(\text{Like}|\text{Actor}) = 0.85$$

| | Type | Length | Director | Famous actors | Liked? |
|---|----------|--------|----------|---------------|--------|
| 1 | Comedy | Short | Adamson | No | Yes |
| 2 | Animated | Short | Lasseter | No | No |
| 3 | Drama | Medium | Adamson | No | Yes |
| 4 | Animated | Long | Lasseter | Yes | No |
| 5 | Comedy | Long | Lasseter | Yes | No |
| 6 | Drama | Medium | Singer | Yes | Yes |
| 7 | Animated | Short | Singer | No | Yes |
| 8 | Comedy | Long | Adamson | Yes | Yes |
| 9 | Drama | Medium | Lasseter | No | Yes |

- The information gain

$$IG(\textit{Type}) = H(\textit{Like}) - H(\textit{Like}|\textit{Type}) = 0.3$$

$$IG(\textit{Length}) = H(\textit{Like}) - H(\textit{Like}|\textit{Length}) = 0.3$$

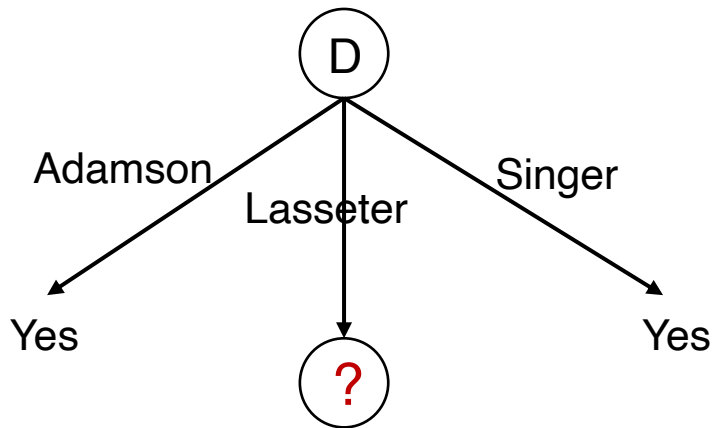
$$IG(\textit{Actor}) = H(\textit{Like}) - H(\textit{Like}|\textit{Actor}) = 0.06$$

$$IG(\textit{Director}) = H(\textit{Like}) - H(\textit{Like}|\textit{Director}) = 0.55$$

⇒ Director should be the root node

| | Type | Length | Director | Famous actors | Liked? |
|---|----------|--------|----------|---------------|--------|
| 1 | Comedy | Short | Adamson | No | Yes |
| 2 | Animated | Short | Lasseter | No | No |
| 3 | Drama | Medium | Adamson | No | Yes |
| 4 | Animated | Long | Lasseter | Yes | No |
| 5 | Comedy | Long | Lasseter | Yes | No |
| 6 | Drama | Medium | Singer | Yes | Yes |
| 7 | Animated | Short | Singer | No | Yes |
| 8 | Comedy | Long | Adamson | Yes | Yes |
| 9 | Drama | Medium | Lasseter | No | Yes |

- Build the tree



| | Type | Length | Director | Famous actors | Liked? |
|---|----------|--------|----------|---------------|--------|
| 1 | Comedy | Short | Adamson | No | Yes |
| 2 | Animated | Short | Lasseter | No | No |
| 3 | Drama | Medium | Adamson | No | Yes |
| 4 | Animated | Long | Lasseter | Yes | No |
| 5 | Comedy | Long | Lasseter | Yes | No |
| 6 | Drama | Medium | Singer | Yes | Yes |
| 7 | Animated | Short | Singer | No | Yes |
| 8 | Comedy | Long | Adamson | Yes | Yes |
| 9 | Drama | Medium | Lasseter | No | Yes |

- Since all outcomes from the branches of **Adamson** and **Singer** is Yes, we don't need to further expand the two branches
- The problem is how to choose the attribute for the branch of Lasseter

Continue to Expand

- After choosing the director of Lasseter, the remaining data is

| | Type | Length | Director | Famous actors | Liked? |
|---|----------|--------|----------|---------------|--------|
| 2 | Animated | Short | Lasseter | No | No |
| 4 | Animated | Long | Lasseter | Yes | No |
| 5 | Comedy | Long | Lasseter | Yes | No |
| 9 | Drama | Medium | Lasseter | No | Yes |

- Re-computing the entropy and conditional entropy gives

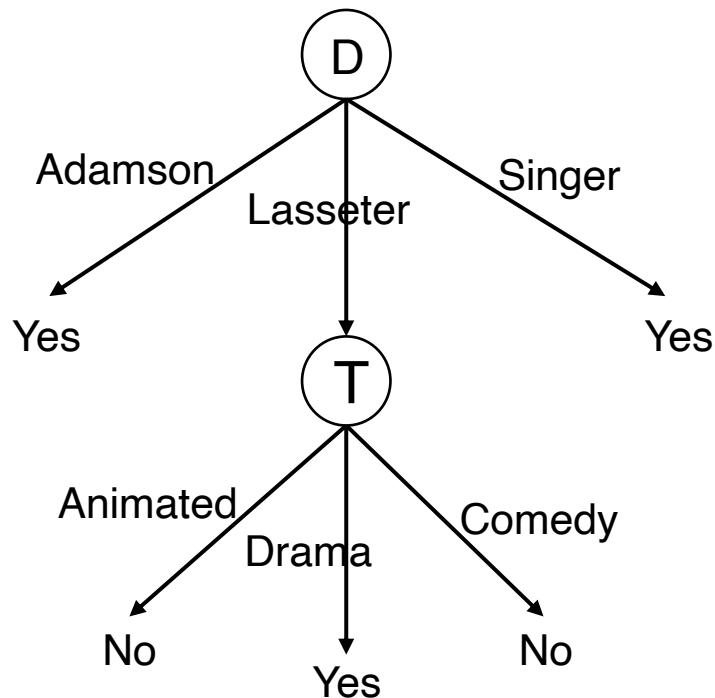
$$H(Like) = 0.81 \quad H(Like|Type) = 0 \quad H(Like|Length) = 0 \quad H(Like|Actor) = 0.5$$

- Thus, the information gains are

$$IG(Type) = 0.81 \quad IG(Length) = 0.81 \quad IG(Actor) = 0.31$$

Thus, we should choose the attribute of Type or Length to expand

- Build the tree



| | Type | Length | Director | Famous actors | Liked? |
|---|----------|--------|----------|---------------|--------|
| 2 | Animated | Short | Lasseter | No | No |
| 4 | Animated | Long | Lasseter | Yes | No |
| 5 | Comedy | Long | Lasseter | Yes | No |
| 9 | Drama | Medium | Lasseter | No | Yes |

This is the final decision tree!!

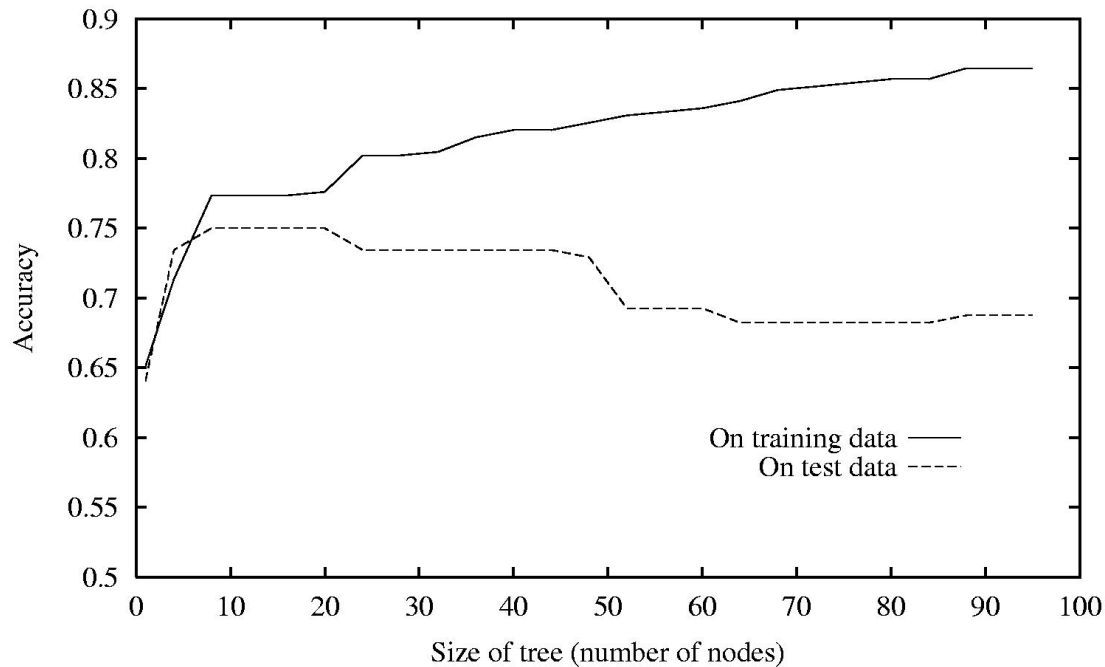
- We stop further expanding the tree because there is only one possible outcome *w.r.t.* every root-leaf path in the training data

Stop Expanding Criteria

- The tree cannot expand forever and should stop at some point. There are some stopping criteria as elaborated below
 - All the remaining instances have the same label
 - We run out of all the attributes
 - The depth of tree reaches the maximum limit
 - The information gain is smaller than a threshold
 -

Overfitting Issue

- If the tree is too large or too complex, it will work very well on the training data, but may perform poorly on the testing data



Tree Pruning

- To control the complexity of decision trees, we can
 - prune the branches as we learn the trees
 - prune the branches after learning the trees
- Basing on a validation dataset,
 - prune the nodes that doesn't hurt the accuracy on the **validation set**
 - Greedily remove the node that improves the **validation accuracy** least
 - Stop when the **validation set** accuracy starts to deteriorate