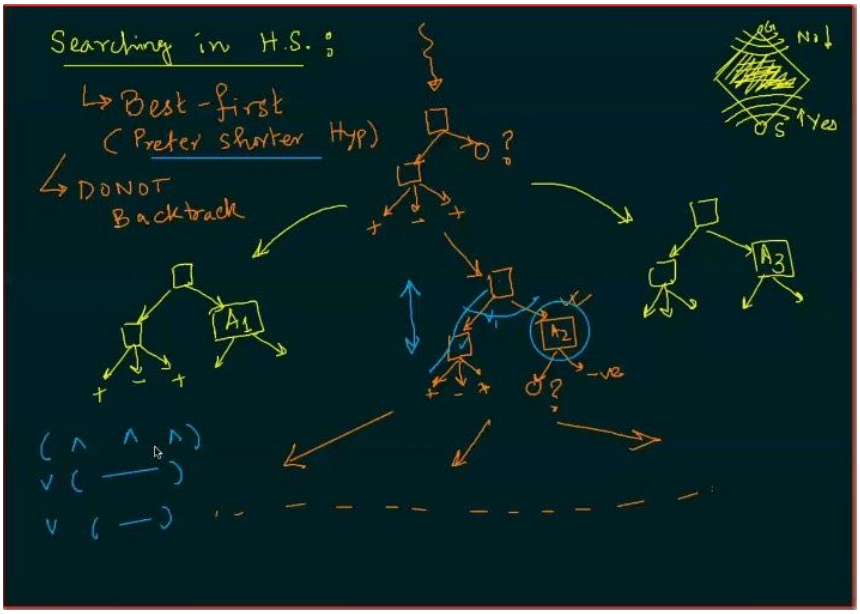
Lecture Scribe
For
Machine Learning(CS60050), Spring 2020-2021
Decision Tree
Date: 15th January, 2021

Hypothesis space in Decision Tree learning is set of Decision Trees.
The target is to find the hypothesis(h) which approximates to the unknown function(f).

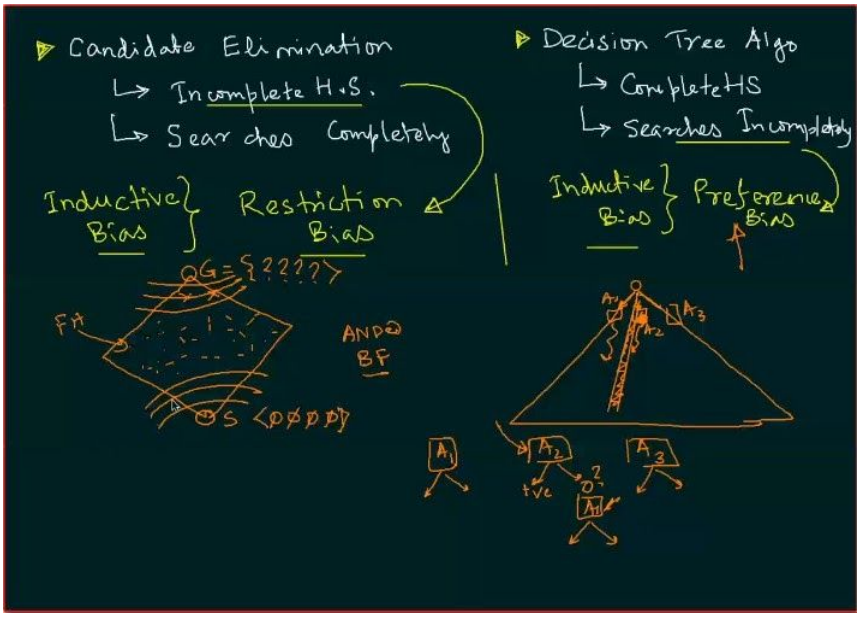
(Refer Slide Time: 09:47)



How does searching take place in hypothesis space to obtain the final hypothesis?

In decision tree learning, attributes are added up to the tree based on information gain. The next attribute that is added can be selected in different ways. It could be either A_1 or A_2 or A_3. For different selection of attributes different learning trees are obtained. It searches for the next attribute that can make a better classification tree. It uses Best-first search for that. Best-first search prefers shorter hypotheses. Shorter hypothesis meaning height of the tree is shorter. It is because of higher the depth , longer the formula in AND in respect to OR. By preferring the next attribute, it tries to restrict the height of the tree. This kind of search does not allow backtracking to alter the attribute previously selected .

(Refer Slide Time: 19:42)

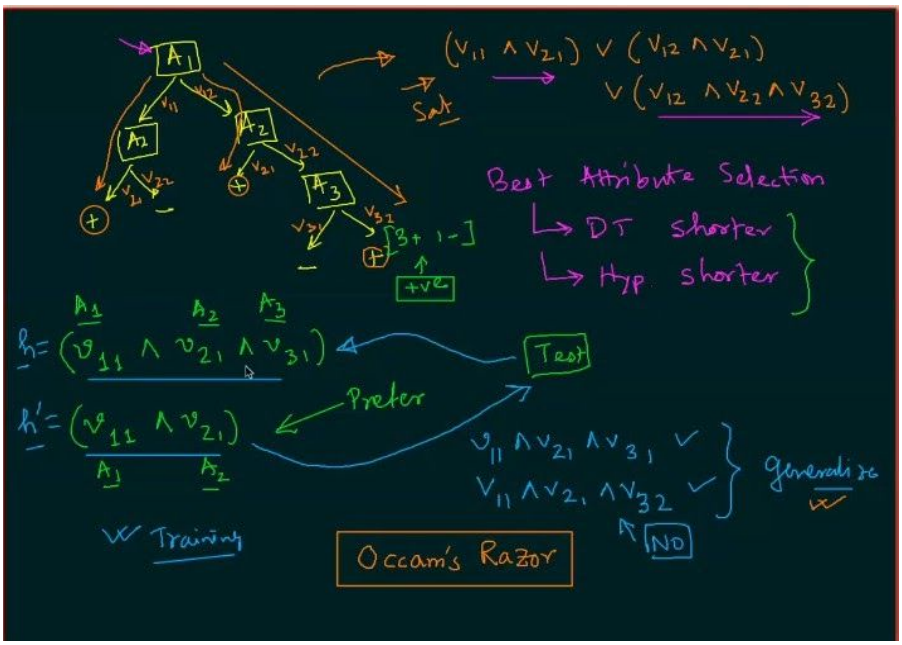


Comparison of Learning Algorithms:

We compare the Candidate Elimination and Decision tree algorithm in the following table.

Candidate Elimination	Decision Tree
It results in incomplete hypothesis space. Because the hypothesis space does not allow a disjunctive formula.	It results in complete hypothesis space. It allows all kinds of formula(AND, OR).
To obtain the final hypothesis, it searches hypothesis space completely.	To obtain the final hypothesis, it searches hypothesis space incompletely.
It has inductive bias since the hypothesis is formed only using Boolean AND operation. It is restricted bias. Hypothesis space is restricted in a incomplete manner	It has preference bias. Complete hypothesis space is searched incompletely based on preference in terms of information gain.

(Refer Slide Time: 31:47)



Why do best-first search prefer decision trees with lesser depth?

Best attribute selection makes Decision tree shorter that implies a shorter hypothesis is obtained. Shorter hypothesis makes the tree more generalized to test data. It can be understood well with an example. Let us consider, hypothesis 1(h_1) and hypothesis 2(h_2).

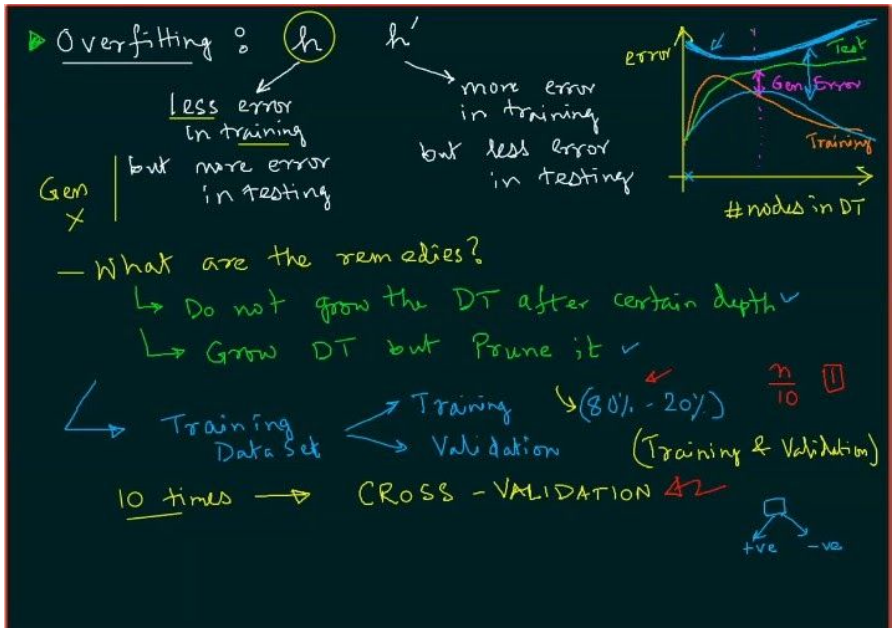
h_1= (V_{11} AND V_{21} AND V_{31}), where V_{ij} is attribute value and 'AND' is boolean AND operation.

h_2= (V_{11} AND V_{21}), where V_{ij} is attribute value and 'AND' is boolean AND operation .

AND terms come from the depth of the tree and OR terms come from branching.
Now, if we pass a test data to h_1, it only passes (V_{11} AND V_{21} AND V_{31}).
But, h_2 can pass (V_{11} AND V_{21} AND V_{31}) as well as (V_{11} AND V_{21} AND V_{32}). That means h_2 is more generalized than h_1. Also, the hypothesis that we get is good fit with training data and we haven't seen test data. So, it is preferred that the hypothesis be more generalized.

NOTE: Shorter hypothesis always gives a generalized hypothesis, it is not proven theoretically. Because (V_{11} AND V_{21} AND V_{32}) could be negative in test data. It is a conjecture. This conjecture is known as **Occam's Razor**. In the decision tree, we follow Occam's Razor to make shorter hypothesis preferences. Occam's Razor has its pros and cons. But, it generalizes unseen data in the correct sense.

(Refer Slide Time: 42:03)



Overfitting: The phenomena that the hypothesis is too much fitted with training data is known as overfitting. Consider two hypotheses- h and h'.

Hypothesis h is giving less error in training data, but more error in testing data.
Hypothesis h' is giving more error in training, but less error in testing.

Then we say formally hypothesis h is overfitting the data.

The graph in the right hand side is showing two curves for error with the depth in the decision tree corresponding to training data and testing data. The gap between the curve is called generalization error. At first, the generalization error was minimal when the depth of the tree was less. Then it increases with the increment of depth.

What is the solution to the overfitting problem?

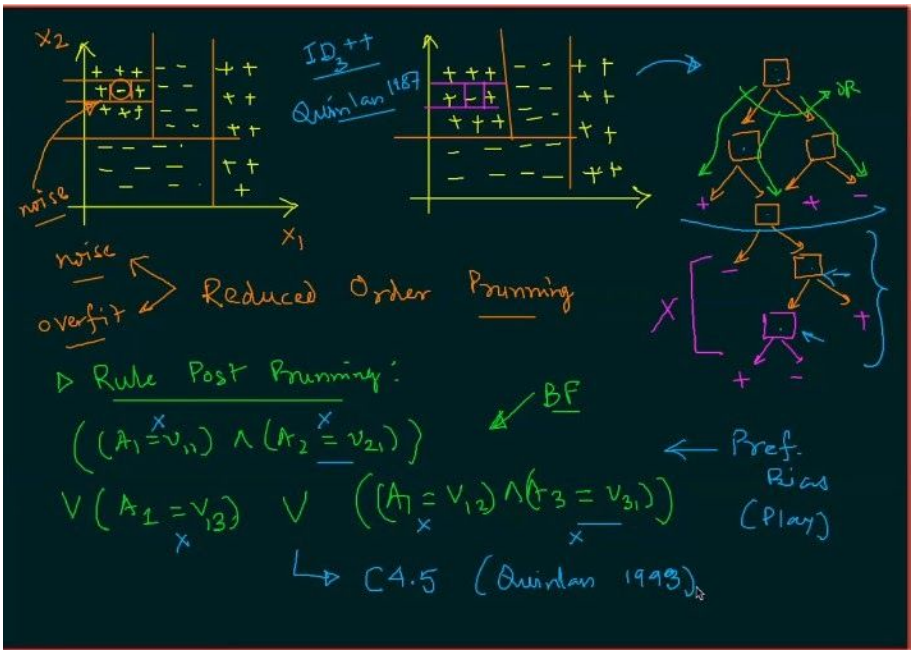
1. Do not let the decision tree grow after a certain depth.
   or
2. Let the decision tree grow , and then prune it.

How to know if the model is overfitting as we don't have test data?

We could do one of the following:

1. Split the whole dataset in training and validation dataset into 80:20 ratio. Then check the training and validation errors. This is called the **training and validation approach**.
2. Divide the dataset into 10 parts. Use 9/10 th as training and 1/10th as validation dataset. Then check the training and validation errors. Repeat this experiment 10 times using different 9/10th and 1/10 th parts as training and validation dataset. This method is called **cross-validation**.

(Refer Slide Time: 49:14)



Types of Pruning:

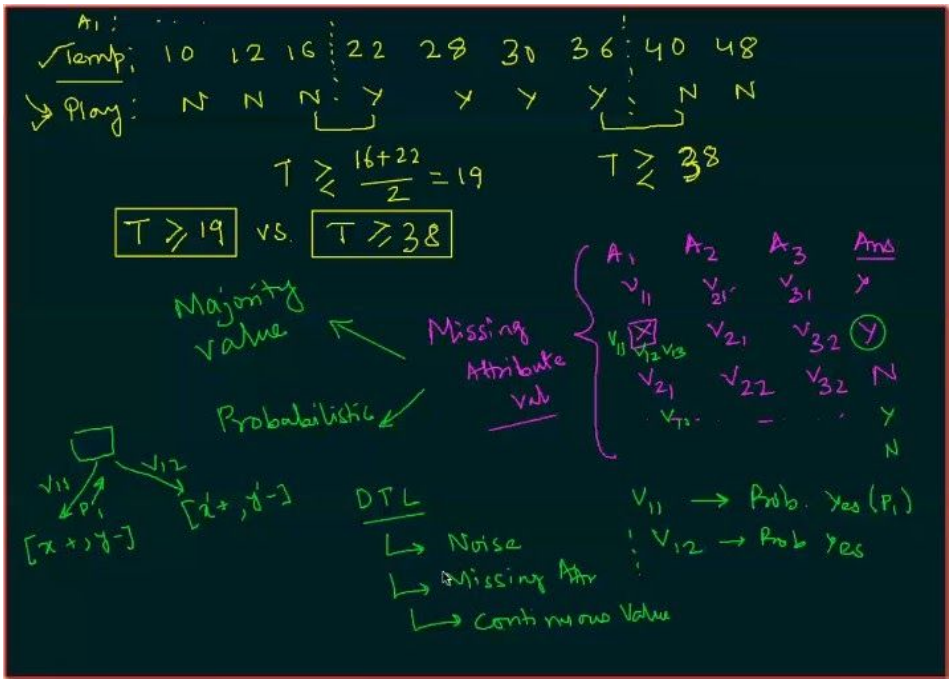
Pruning is useful when there is noise in data or it needs to reduce overfitting.
There are two types of pruning- (a) Reduced order pruning, (b) Rule post pruning.

Reduced Order Pruning:

In reduced order pruning the size of the tree is reduced by eliminating the attributes that are in lower depth.

Rule Post Pruning:
Unlike the reduced order pruning, this method can eliminate any attribute and any brunch to make the tree more generalized.

(Refer Slide Time: 55:07)



How continuous-valued attributes are handled?

Temp is the continuous-valued attribute. One criteria is to take the average of boundary values where 'yes' changes to 'no' or 'no' changes to 'yes'. Then we get two candidates- one is Temp>=19 and another is Temp>=38. Then calculate the information gain for these two candidates and select the higher one as an attribute node in the tree. In summary, we discretize the interval where we will get no yes boundary and split the interval to obtain the candidate attribute. Then we calculate information gain to decide the final attribute for the node.

How missing attribute value is handled?

Attribute value for A_1 in the 2nd row is missing as shown with X sign. We could do one of the following to handle this particular case.

1. **Majority Value Approach:** Replace the missing value with majority value in A_1 which says yes as yes is the level of the instance. In example it is V_{11}.
2. **Probabilistic Approach:** Calculate probability of getting yes for different values of A_1. For each branch from attribute A_1, incorporate the probability value that is previously computed. For each branch, compute the number of positive and negative samples

assuming the missing value is assigned to the branch value. Then multiply this probability value and then sum it over to calculate information gain for A_1.

Conclusion:

Decision tree can handle **noise**, **missing attributes**, **continuous-valued variables** in attributes. Therefore, Decision tree is one of the most preferred learning algorithms while learning the boolean concept.

Scribe prepared by Salma Mandi(20CS92R03)