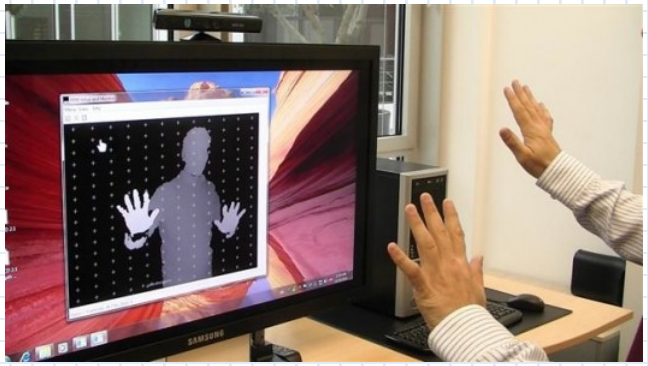


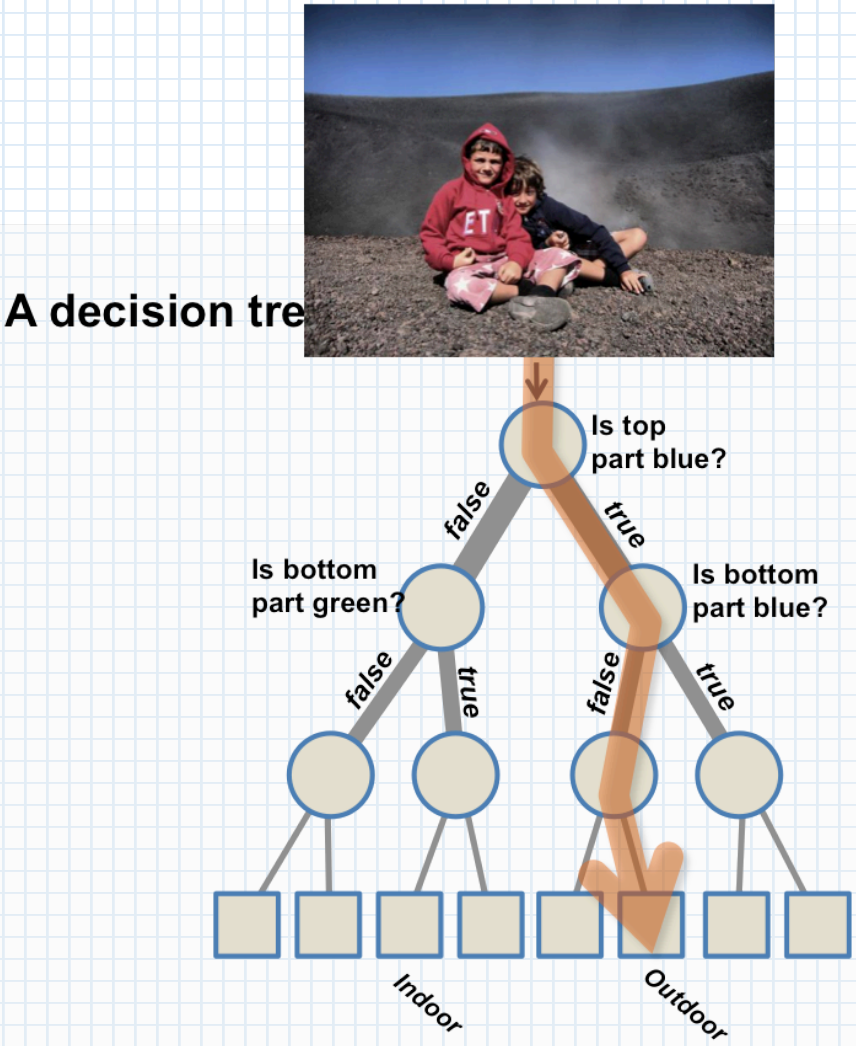
CSL465/603 - Decision Tree Learning

Fall 2016

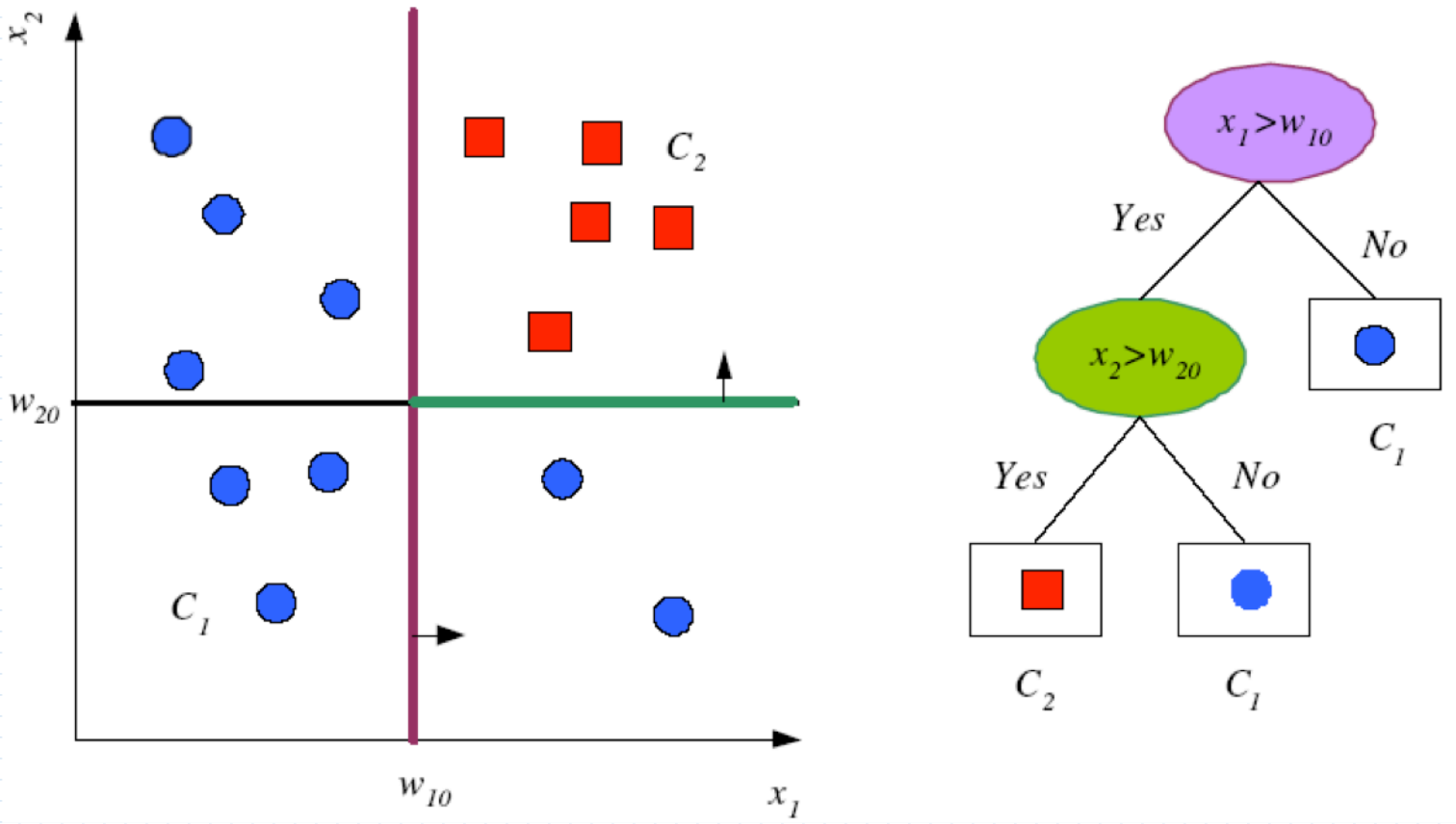
Narayanan C Krishnan

ckn@iitrpr.ac.in

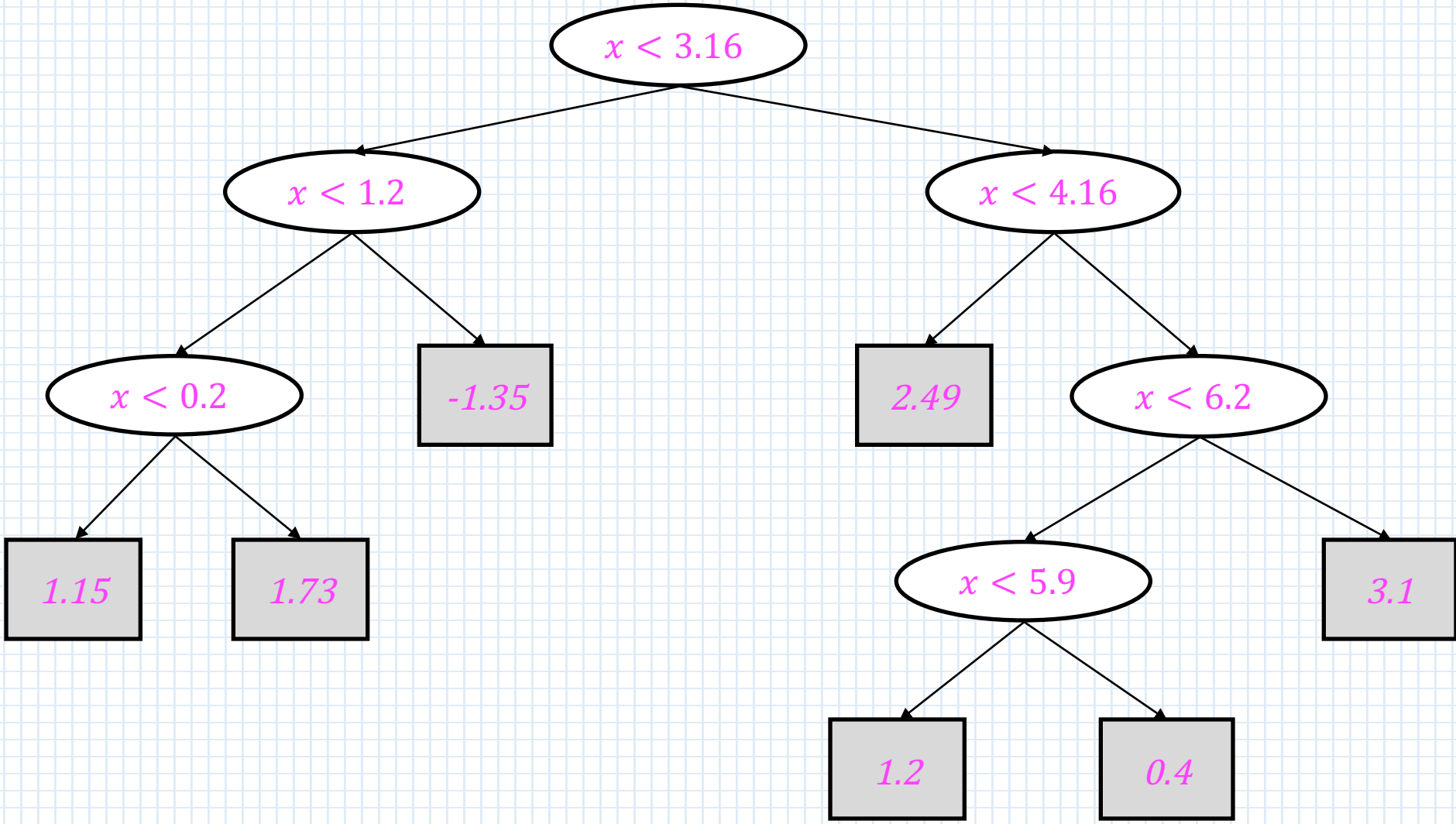
Decision Tree - Example



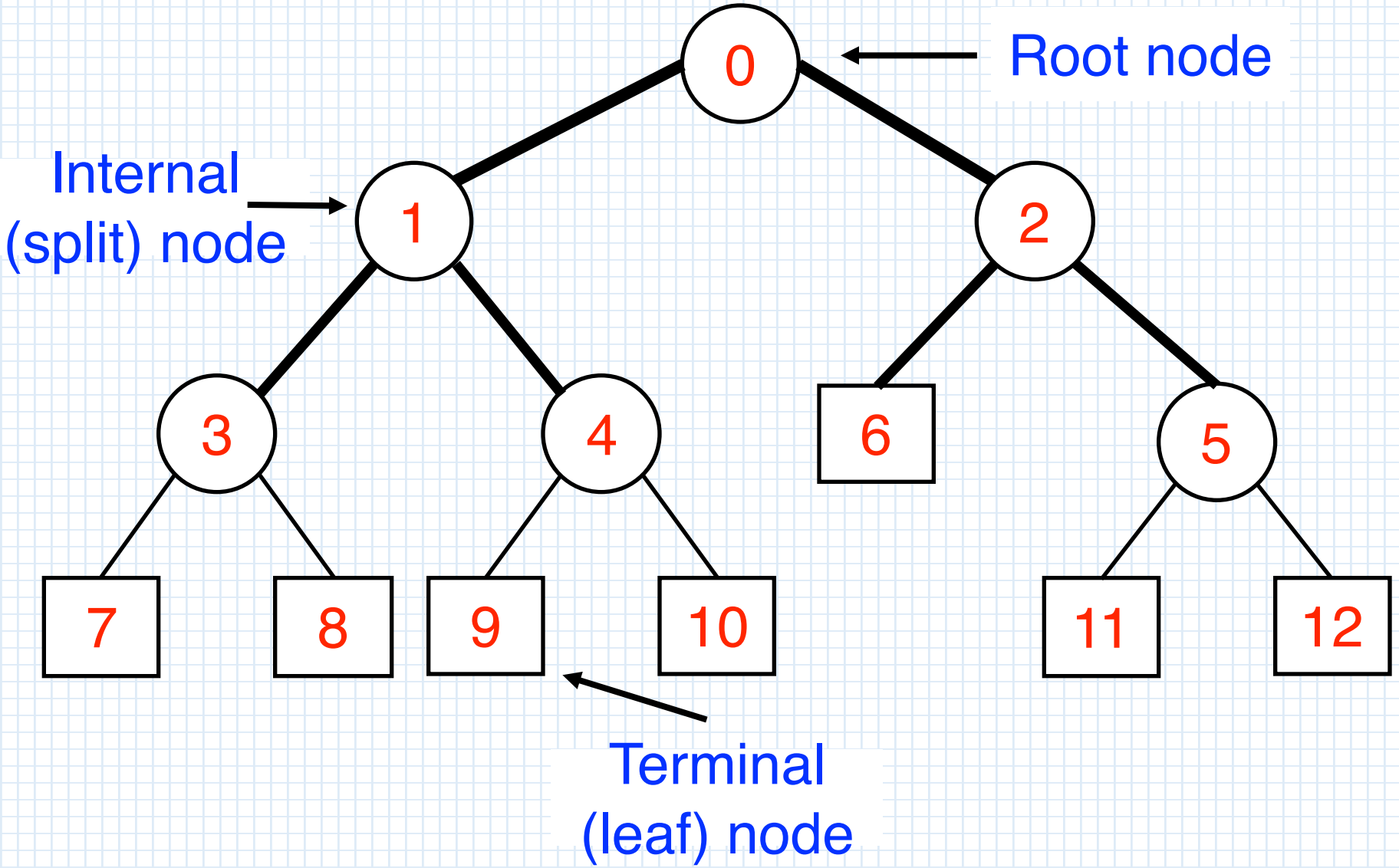
Classification Tree



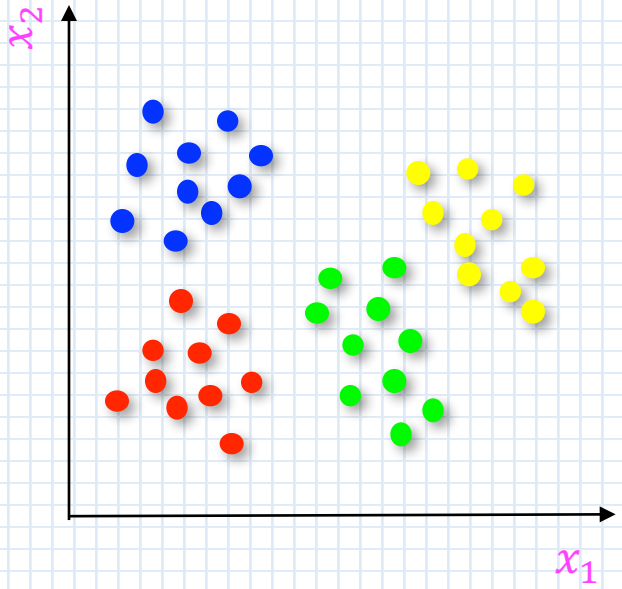
Regression Tree

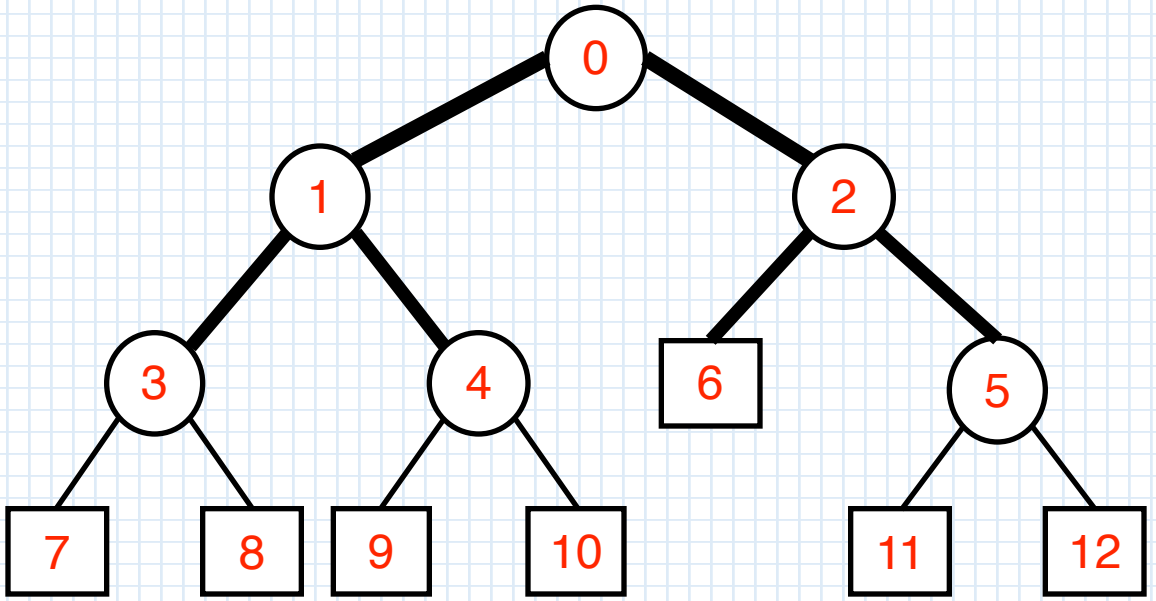


Decision Tree - Structure



Learning Classification Tree



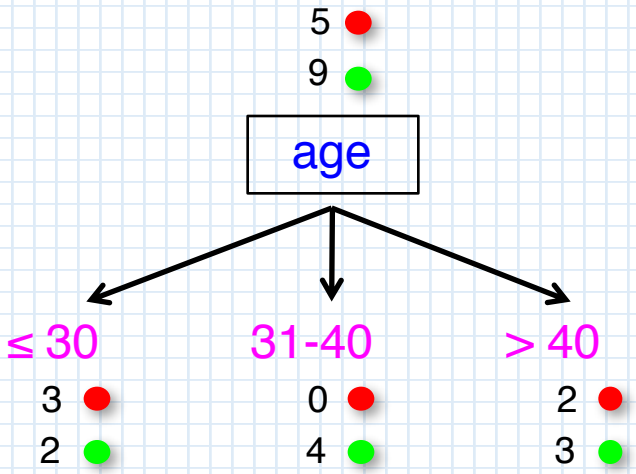


Example Dataset

example	age	income	student	Credit rating	Buys computer
x_1	≤ 30	high	no	fair	no
x_2	≤ 30	high	no	excellent	no
x_3	$31-40$	high	no	fair	yes
x_4	>40	medium	no	fair	yes
x_5	>40	low	yes	fair	yes
x_6	>40	low	yes	excellent	no
x_7	$31-40$	low	yes	excellent	yes
x_8	≤ 30	medium	no	fair	no
x_9	≤ 30	low	yes	fair	yes
x_{10}	>40	medium	yes	fair	yes
x_{11}	≤ 30	medium	yes	excellent	yes
x_{12}	$31-40$	medium	no	excellent	yes
x_{13}	$31-40$	high	yes	fair	yes
x_{14}	>40	medium	no	excellent	no

Tree Construction (1)

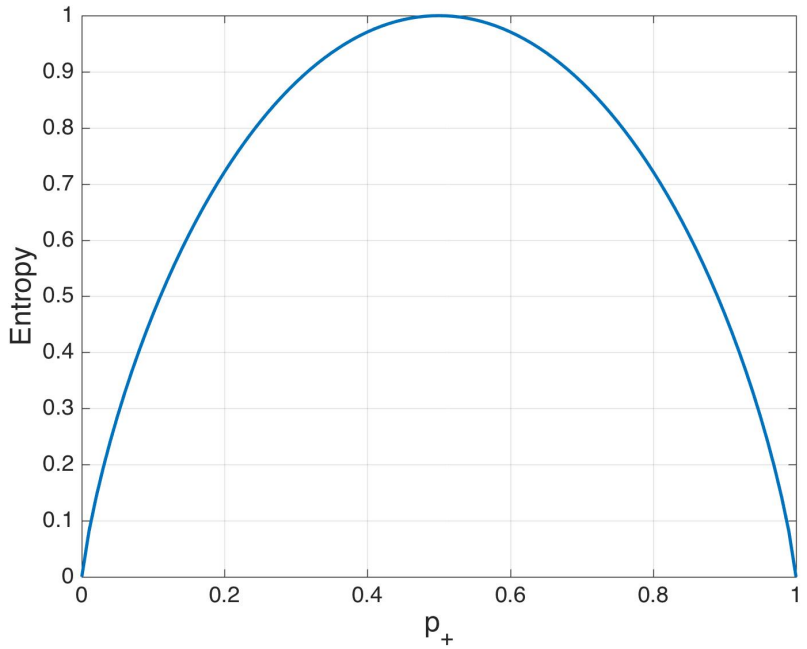
example	age	Buys computer
x_1	≤ 30	no
x_2	≤ 30	no
x_8	≤ 30	no
x_9	≤ 30	yes
x_{11}	≤ 30	yes
x_3	$31 - 40$	yes
x_7	$31 - 40$	yes
x_{12}	$31 - 40$	yes
x_{13}	$31 - 40$	yes
x_4	> 40	yes
x_5	> 40	yes
x_6	> 40	no
x_{10}	> 40	yes
x_{14}	> 40	no



Entropy - 2-class (1)

- I be the set of training examples
- p_+ be the proportion of positive examples in I
- p_- be the proportion of negative examples in I
- Entropy measure the impurity of I

$$\text{Entropy}(I) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$



Entropy - 2-class (2)

- $\text{Entropy}(I)$ represents the expected number of bits to encode the class (+ or -) of a randomly drawn member of I
  - Derived from principles of Information Theory
    - Optimal length code assigns $-\log_2 p$ bits to a message having a probability p
  - The expected number of bits to encode + or – of a random member of I

$$\text{Entropy}(I) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Entropy - 2-class (2)

- $\text{Entropy}(I)$ represents the expected number of bits to encode the class (+ or -) of a randomly drawn member of I
  - Derived from principles of Information Theory
    - Optimal length code assigns $-\log_2 p$ bits to a message having a probability p
  - The expected number of bits to encode + or – of a random member of I

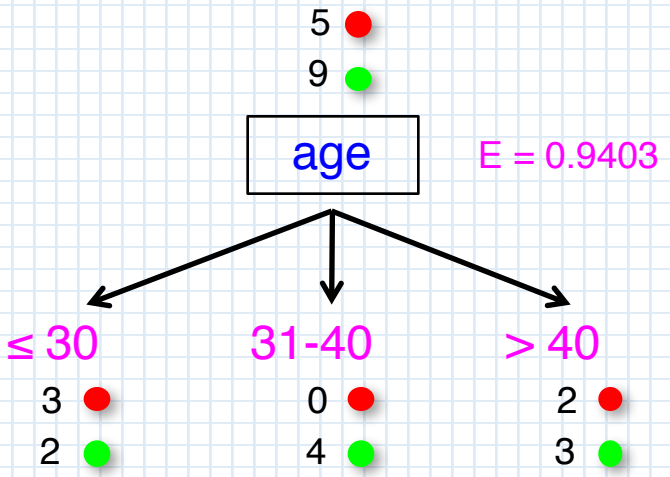
$$\text{Entropy}(I) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

- Entropy if there are K classes?
- Sometimes H is used to refer to entropy

Tree Construction (2)

example	age	Buys computer
x_1	≤ 30	no
x_2	≤ 30	no
x_8	≤ 30	no
x_9	≤ 30	yes
x_{11}	≤ 30	yes
x_3	$31 - 40$	yes
x_7	$31 - 40$	yes
x_{12}	$31 - 40$	yes
x_{13}	$31 - 40$	yes
x_4	> 40	yes
x_5	> 40	yes
x_6	> 40	no
x_{10}	> 40	yes
x_{14}	> 40	no

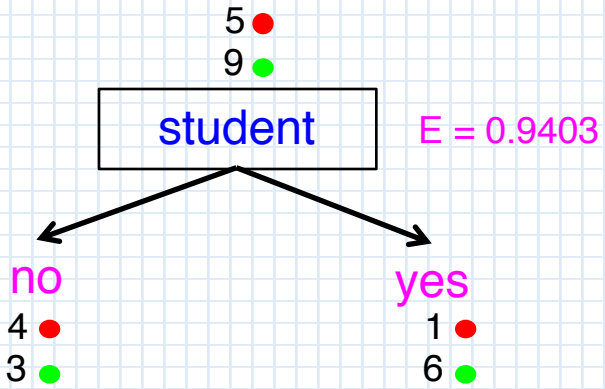
- Entropy for Age



Tree Construction (3)

example	student	Buys computer
x_1	no	no
x_2	no	no
x_3	no	yes
x_4	no	yes
x_8	no	no
x_{12}	no	yes
x_{14}	no	no
x_5	yes	yes
x_6	yes	no
x_7	yes	yes
x_9	yes	yes
x_{10}	yes	yes
x_{11}	yes	yes
x_{13}	yes	yes

- Entropy for Student



Information Gain

- $\text{Gain}(I, x_i)$ – expected reduction in entropy due to sorting on attribute x_i

$$\text{Gain}(I, x_i) = \text{Entropy}(I) - \sum_{v \in \text{Values}(x_i)} \frac{|I_v|}{|I|} \text{Entropy}(I_v)$$

- $\text{Values}(x_i)$ – all possible values that attribute x_i can take.
- $I_v \subset I$ – data points in I that take the value v for the attribute x_i

Mutual Information and Information Gain

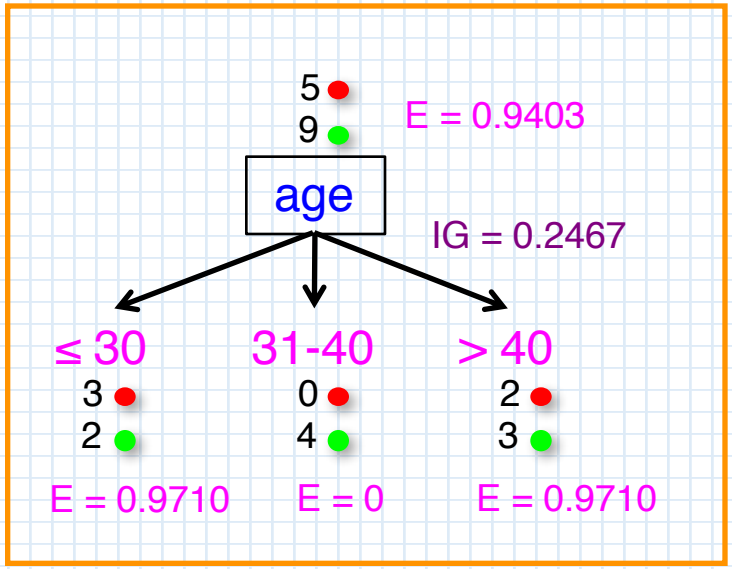
- Mutual Information – measure of mutual dependence of two random variables
- If X and Y are discrete random variables, then mutual information is defined as

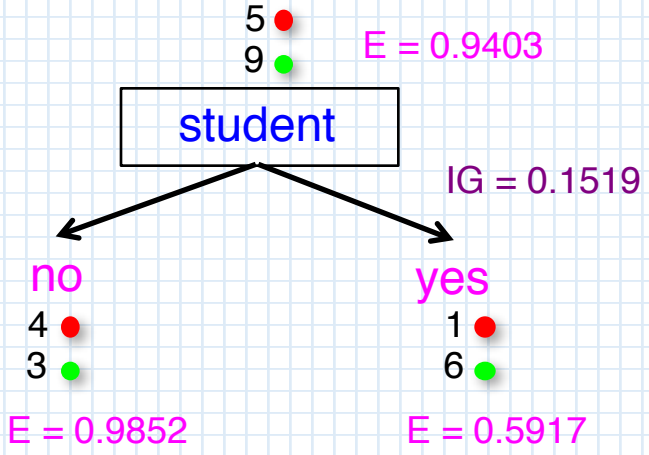
$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

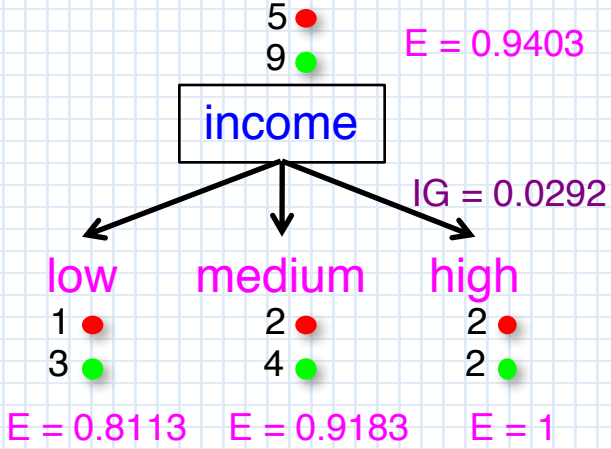
- In context of decision trees, mutual information and information gain are synonymous

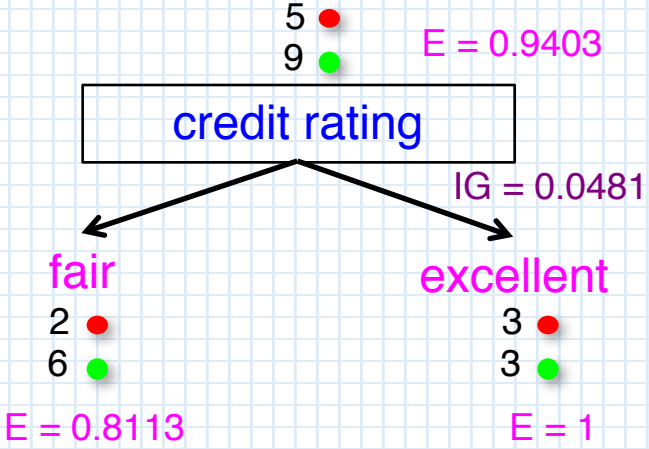
$$I(X;Y) = H(Y) - H(Y|X)$$

Tree Construction (4)



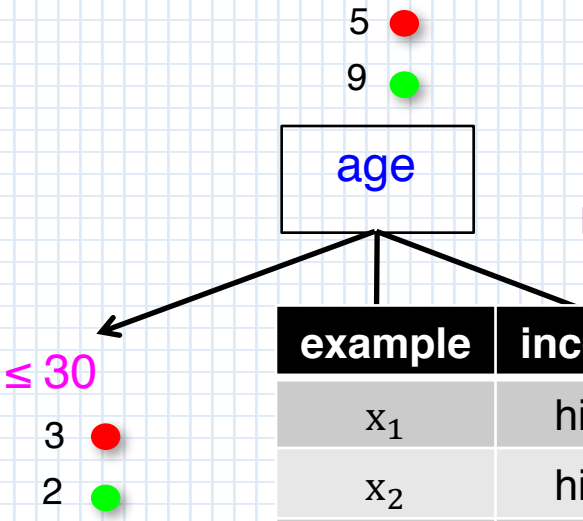






Tree Construction (5)

- Selecting the next split

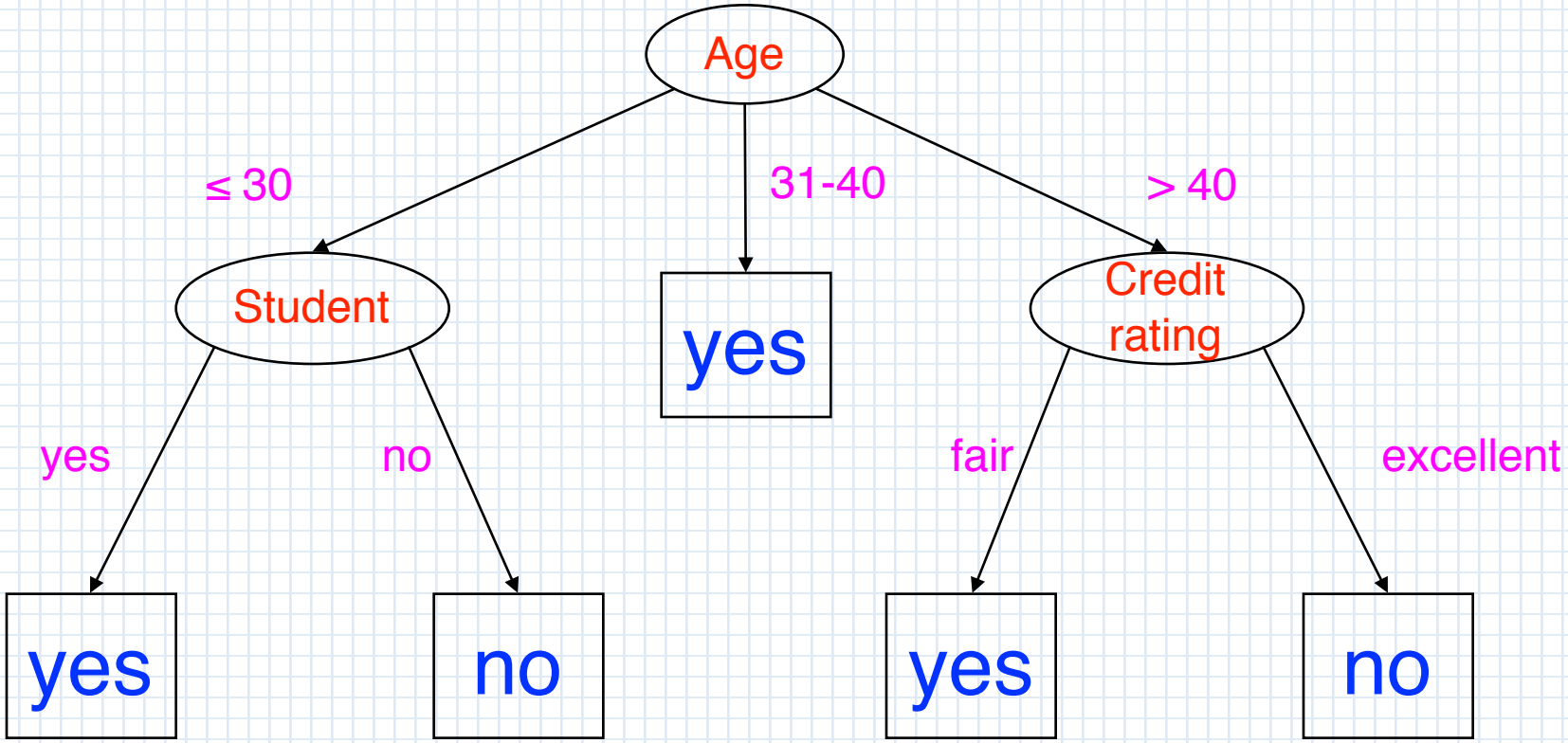


example	income	student	Credit rating	Buys computer
x_1	high	no	fair	no
x_2	high	no	excellent	no
x_8	medium	no	fair	no
x_9	low	yes	fair	yes
x_{11}	medium	yes	excellent	yes

Putting it all together – ID3 Algorithm

- $\mathrm{ID3}(\mathrm{Examples}, \mathrm{Target}-\mathrm{Attribute}, \mathrm{Attributes})$:
- Create a **Root** node for the tree
- If all the $\mathrm{Examples}$ are positive, return single-node tree **Root**, with **positive label**
- If all the $\mathrm{Examples}$ are negative, return single-node tree **Root**, with **negative** label
- Otherwise Begin
  - A – attribute from attributes that best classified $\mathrm{Examples}$
  - The decision attribute for the **Root** is A
  - For each possible value v for attribute A
    - Add a new tree branch below **Root**, corresponding to the test $A = v$
    - Let $\mathrm{Examples}_v$ be the subset of $\mathrm{Examples}$ that have v as the value for attribute A
    - If $\mathrm{Examples}_v$ is empty
      - Then below this new branch add a leaf node with the most frequently occurring $\mathrm{Target}-\mathrm{Attribute}$ in $\mathrm{Examples}$
      - Else, below this new branch add the subtree $\mathrm{ID3}(\mathrm{Examples}_v, \mathrm{Target}-\mathrm{Attribute}, \mathrm{Attributes}-\{A\})$

A Decision Tree for the Dataset



Analyzing Decision Trees

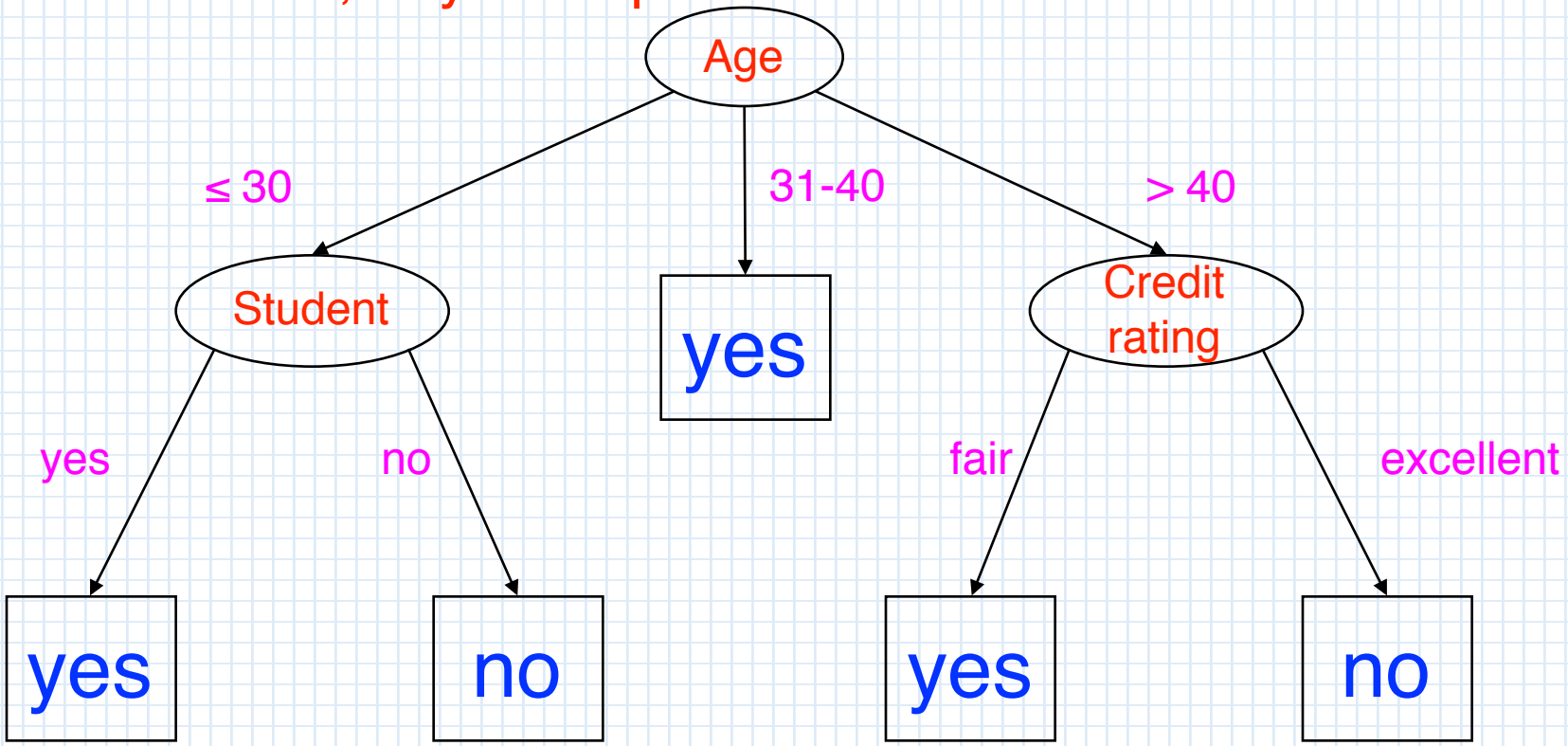
- Hypothesis Space - Growing
  - The space is complete – any discrete valued function can be represented as a decision tree, relative to the available attributes
  - Target function is surely present in the space search
- Outputs only a single hypothesis
  - Consistent with the training set
  - Cannot determine alternate decision trees that are consistent with the training set
  - Cannot resolve between competing hypotheses
- No backtracking
  - Greedy algorithm
- Statistically-based search choices
  - Use all 'available' training data to make decisions on how to refine the current hypothesis

Bias in Decision Tree Learning

- True bias is hard to estimate due to the complex search strategy.
- Approximations
  - Shorter trees are preferred over larger trees
  - Trees that place high information gain attributes close to the root are preferred over those that do not.
- Bias is exhibited in the preference of one hypothesis over another. There is no bias in the hypothesis space (which is complete)

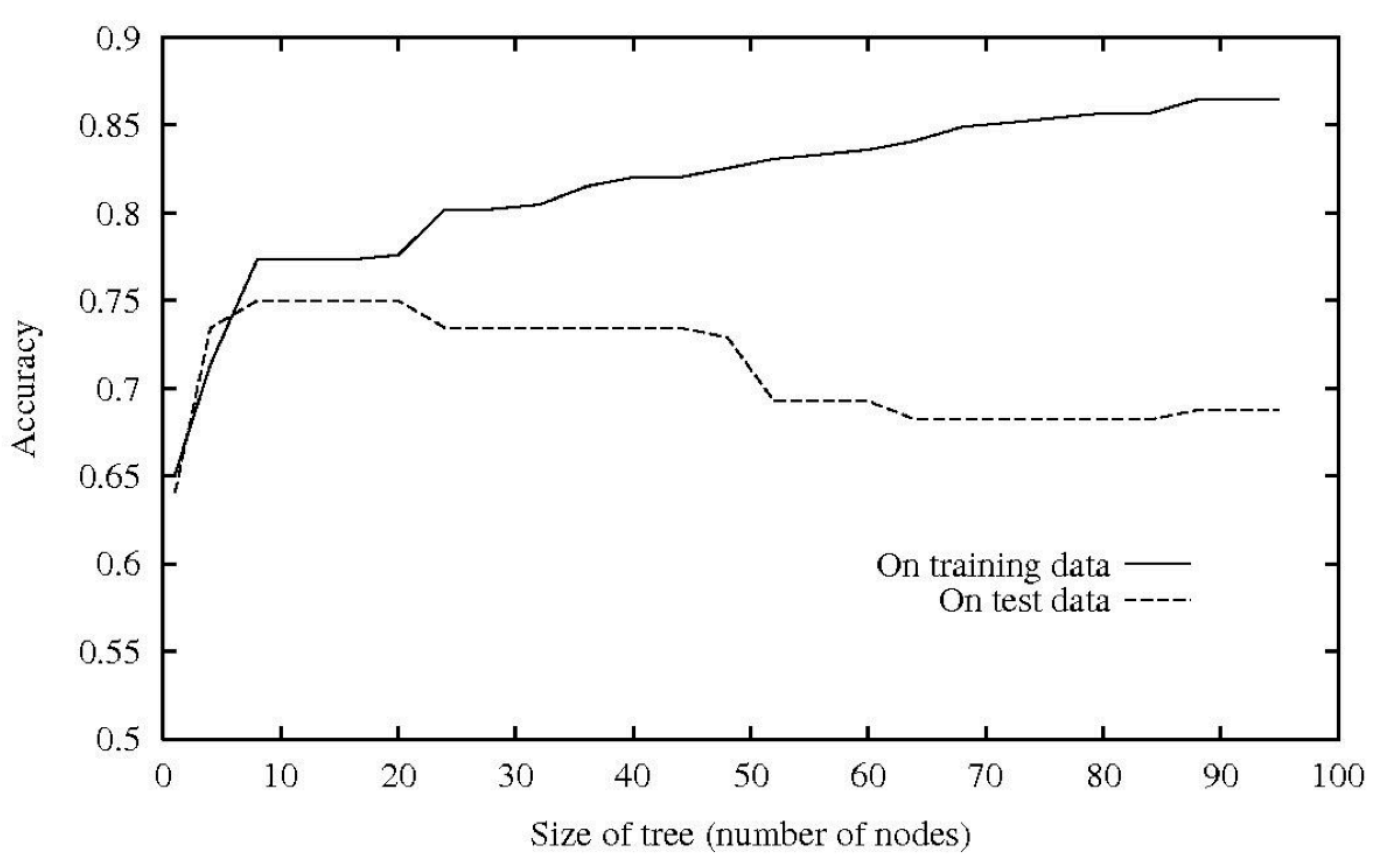
Overfitting in Decision Trees (1)

- Consider adding a noisy training example
  - Age - <= 30, income – high, student – yes, creditrating – excellent, buys computer- no



Overfitting in Decision Trees (2)

- ID3 - applied to the task of learning which medical patients have a form of diabetes



Add/Drop

- Current strength ~ 70
- Last day to add/drop
  - Those who have added the course during the past few days and are having doubts … about continuing, please contact me after the class
- Check the lab that has been posted
  - This is at the lower end of complexity
  - Progressively it will get harder
- Take this course if you are really committed to learning the subject
- Grades will not come easily

Analyzing Decision Trees

- Hypothesis Space – Growing Complexity
  - The space is complete – any discrete valued function can be represented as a decision tree, relative to the available attributes
  - Target function is surely present in the space search
- Outputs only a single hypothesis
  - Consistent with the training set
  - Cannot determine alternate decision trees that are consistent with the training set
  - Cannot resolve between competing hypotheses
- No backtracking
  - Greedy algorithm
- Statistically-based search choices
  - Use all 'available' training data to make decisions on how to refine the current hypothesis

Overfitting – Formal Definition

- Consider the error of hypothesis h over
  - Training data – $E(\mathrm{h}|\mathrm{X})$
  - Entire distribution P of the data – $E_P(\mathrm{h})$
- Hypothesis $\mathrm{h} \in H$ overfits the training data if there is an alternative hypothesis $\mathrm{h}' \in H$ such that
  - $E(\mathrm{h}|\mathrm{X}) < E(\mathrm{h}'|\mathrm{X})$
  - $E_P(\mathrm{h}) > E_P(\mathrm{h}')$

Avoiding Overfitting (1)

- Two (three) strategies to prevent overfitting
  - Stop growing when data split is not statistically significant
  - Grow full tree, then post prune
  - *Using ensembles – Random/Decision Forests*

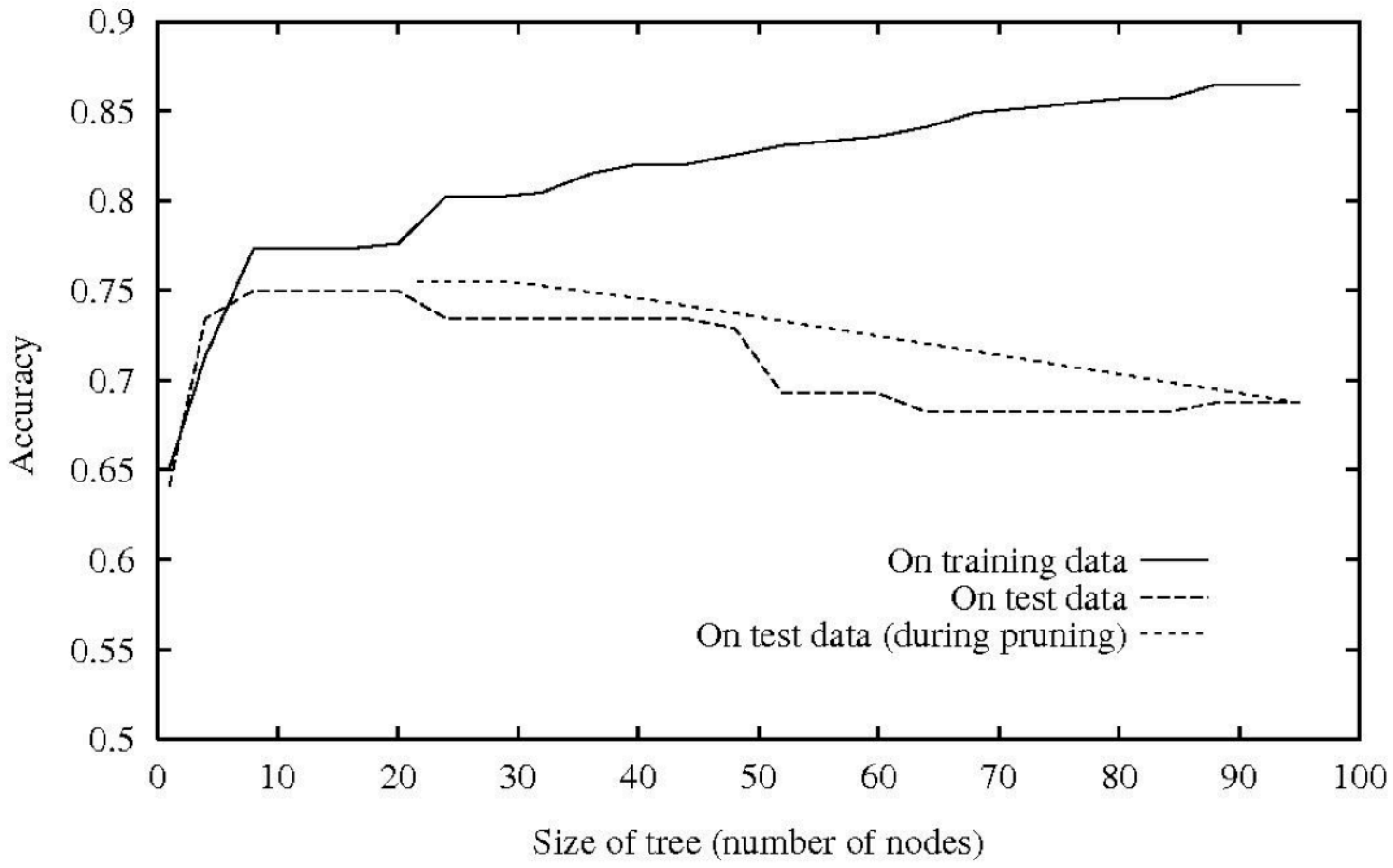
- How to select "best" tree
  - Measure performance over training data
  - Measure performance over separate validation data
  - Add complexity penalty to performance measure

Avoiding Overfitting (2)

- Reduced Error Pruning
  - Split data into training and validation set
  - Do until further pruning is harmful
    - Evaluate the impact on the validation set of pruning each possible subtree (node, plus those below it)
    - Greedily remove the subtree that results in maximum improvement over the validation set
  - Produces the smallest version of the most accurate subtree

Avoiding Overfitting (3)

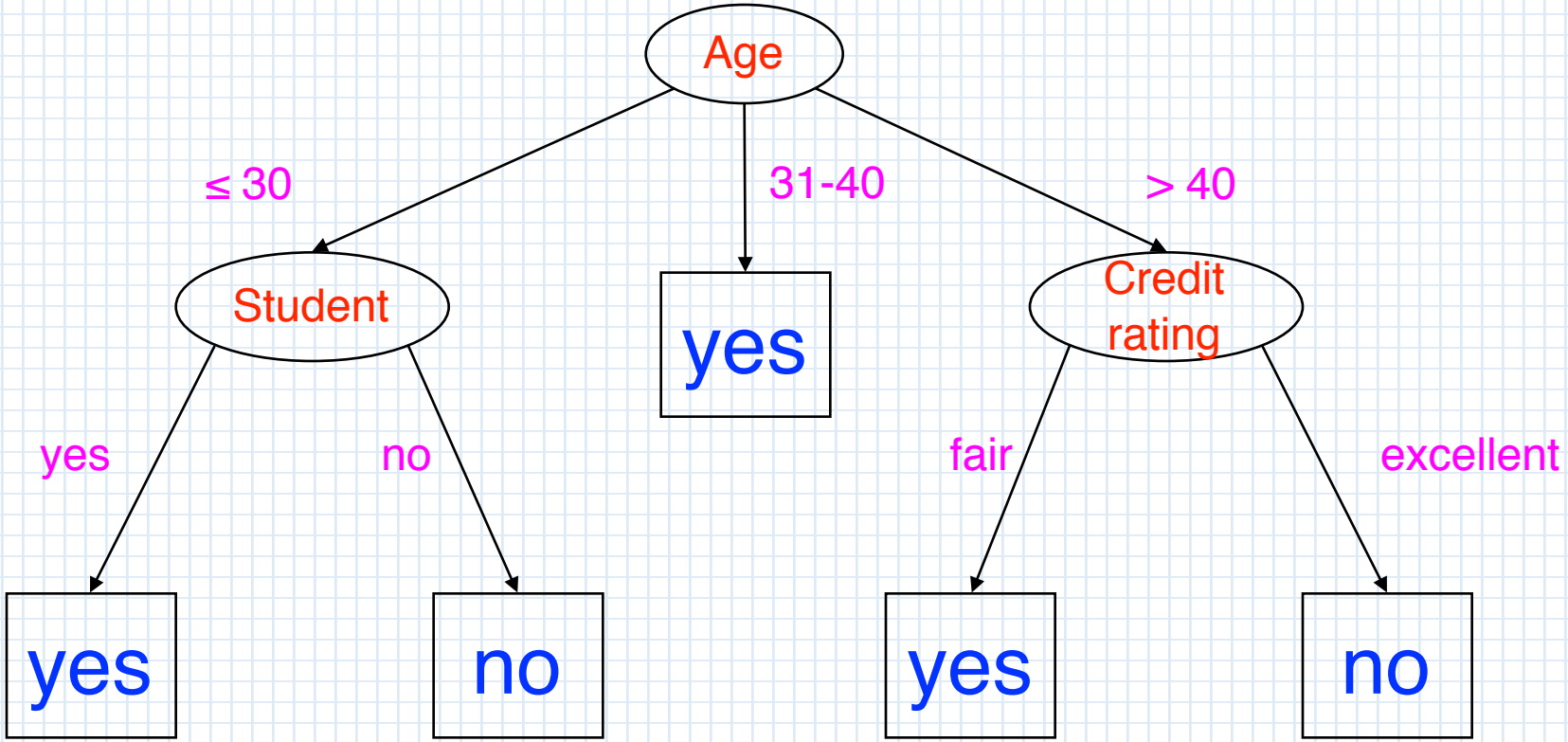
- Reduced Error Pruning - Impact



Avoiding Overfitting (4)

- Rule Post-Pruning
  - Infer the decision tree from the training set
  - Convert the learned tree into an equivalent set of rules
  - Prune each rule by removing any preconditions that result in improving estimated performance
    - Check the preconditions against an held out validation set
  - Sort the pruned rules by their estimated performance
- Most frequently used method (C4.5)

Converting a Decision Tree to Rules



Miscellaneous Issues in Decision Tree Learning (1)

- Real-Valued Features
  - Quantization – creating a discrete feature set to test real values
    - Example – credit rating = 820, new feature – credit rating >745
  - Sort the values and try splits on mid points

Credit rating	189	245	256	370	377	394	605	720
Buys computer	no	no	yes	no	yes	yes	no	no

Miscellaneous Issues in Decision Tree Learning (2)

- Alternative measures for selecting attributes
  - Problem with information gain

Miscellaneous Issues in Decision Tree Learning (3)

example	age	Buys computer
x_1	20	no
x_2	21	no
x_3	23	yes
x_4	34	yes
x_5	25	yes
x_6	41	no
x_7	37	yes
x_8	18	no
x_9	27	yes
x_{10}	45	yes
x_{11}	29	yes
x_{12}	35	yes
x_{13}	36	yes
x_{14}	46	no

Miscellaneous Issues in Decision Tree Learning (4)

- Alternative measures for selecting attributes
  - Problem with information gain
- *Gain Ratio* – incorporates a term called split information that is sensitive to how broadly and uniformly the attribute splits the data

$$\text{SplitInformation}(I, x_i) = -\sum_{c=1}^{C} \frac{|I_c|}{|I|} \log_2 \frac{|I_c|}{|I|}, C = |Values(x_i)|$$

$$\text{Gain Ratio}(I, x_i) = \frac{\text{Information Gain } (I, x_i)}{\text{Split Information}(I, x_i)}$$

Exercise – Compute the Gain Ratio for Age attribute

Miscellaneous Issues in Decision Tree Learning (5)

- Alternative measures for selecting attributes
  - Problem with information gain
  - *Gain Ratio*
- *Cost sensitive information gain*
  - Applications where each attribute has an associated cost
    - Medical Diagnosis – MRI Scan costs ₹ 5000
  - Goal is to learn a consistent tree with low expected cost
  - Modify information gain to include cost

$$\frac{\text{Information Gain}(I, x_i)}{\text{Cost}(x_i)}$$

Miscellaneous Issues in Decision Tree Learning (6)

- Handling Training Examples with Missing Attribute values
- Node n tests the attribute x_i
  - Assign the most common value of x_i among other examples at node n
  - Assign the most common value of x_i among other examples with the same target value at node n
  - Assign probability p_v to each possible value v of x_i
    - Assign fraction p_v of the example to each descendant from the node

Advantages and Disadvantages of Trees

- ✅ Easy to explain – results in set of rules
- ✅ Closely mirrors human decision making process
- ✅ Visualization is easy for interpretation even by an non-machine learning expert.
- ✅ Number of hyper-parameters to by tuned is almost none
- ❌ Trees are prone to overfitting
- ❌ Prediction accuracies are low compared to other approaches
  - Aggregation of many decision trees can substantially improve the performance

Decision (Random) Forests

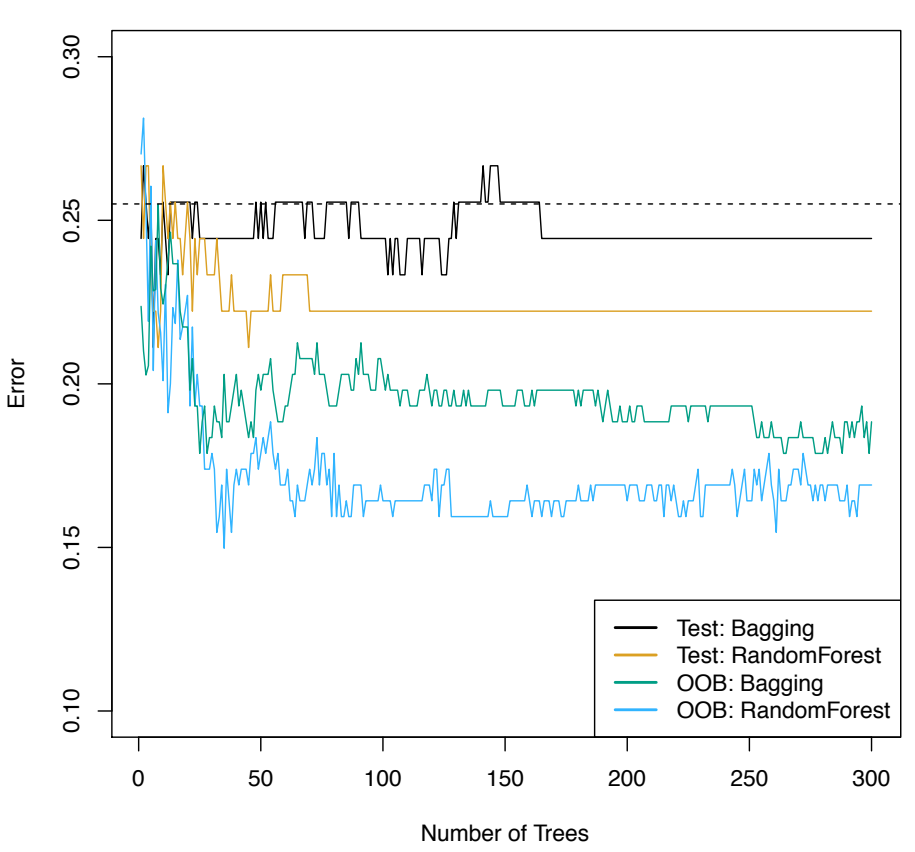
- Ensemble of decision trees
  - Reduce overfitting, by averaging the output of multiple trees
- How to grow multiple trees?
  - Use a random subset of the training data – Instance Bagging
  - Use a random subset of features – Feature Bagging

Instance Bagging (1)

- Also referred to as Bootstrap aggregation
- General procedure for reducing the variance of a statistical learning method
- Principle
  - Given a set of n independent observations $h_1, h_2, \ldots, h_n$ each with variance σ^2, the variance of the mean $\bar{h}$ of the observations is given by σ^2/n Prove it as an exercise
- Averaging a set of observations reduces variance
  - However not practical because we do have access to multiple training sets
  - Solution – Bootstrap, by taking repeated samples from the training dataset

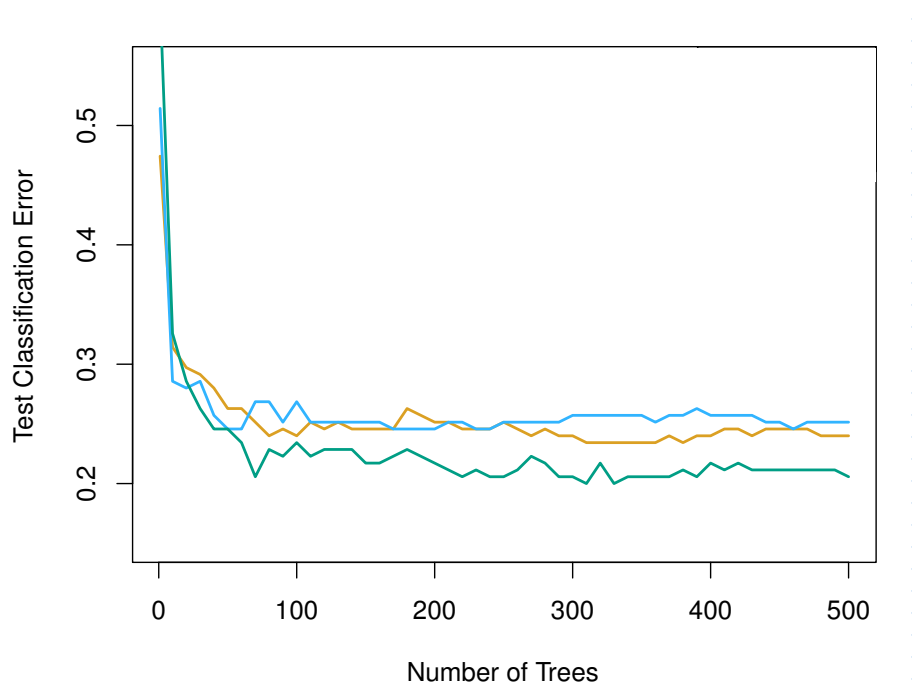
Instance Bagging (2)

- Generate K different bootstrapped training datasets
  - Sampling with replacement
- Learn a decision tree on each of the K datasets – h_K
- Final prediction is the majority vote (*average*) of all predictions
- *OOB – out of bag examples*



Feature Bagging – Random Forests

- Provide an improvement over instance bagging
  - Decorrelates trees – in turn reduces variance when we average the trees
- Build a number of decision trees on bootstrapped training samples with the change
  - Each tree is constructed using a random subset of m attributes (instead of D)
  - Typically $m \approx \sqrt{D}$



Summary

- Popular concept learning method
- Uses information theoretic heuristic for the construction of the tree
- The model is completely expressive
- Inductive Bias is towards shorter trees