

Data Mining:


Concepts and Techniques

(3rd ed.)

Jiawei Han, Micheline Kamber, and Jian Pei
University of Illinois at Urbana-Champaign &
Simon Fraser University

©2011 Han, Kamber & Pei. All rights reserved.

Classification: Basic Concepts

- Classification: Basic Concepts 
- Decision Tree Induction
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy:
Ensemble Methods
- Summary

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
 - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
 - New data is classified based on the training set
- Unsupervised learning (clustering)
 - The class labels of training data is unknown
 - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

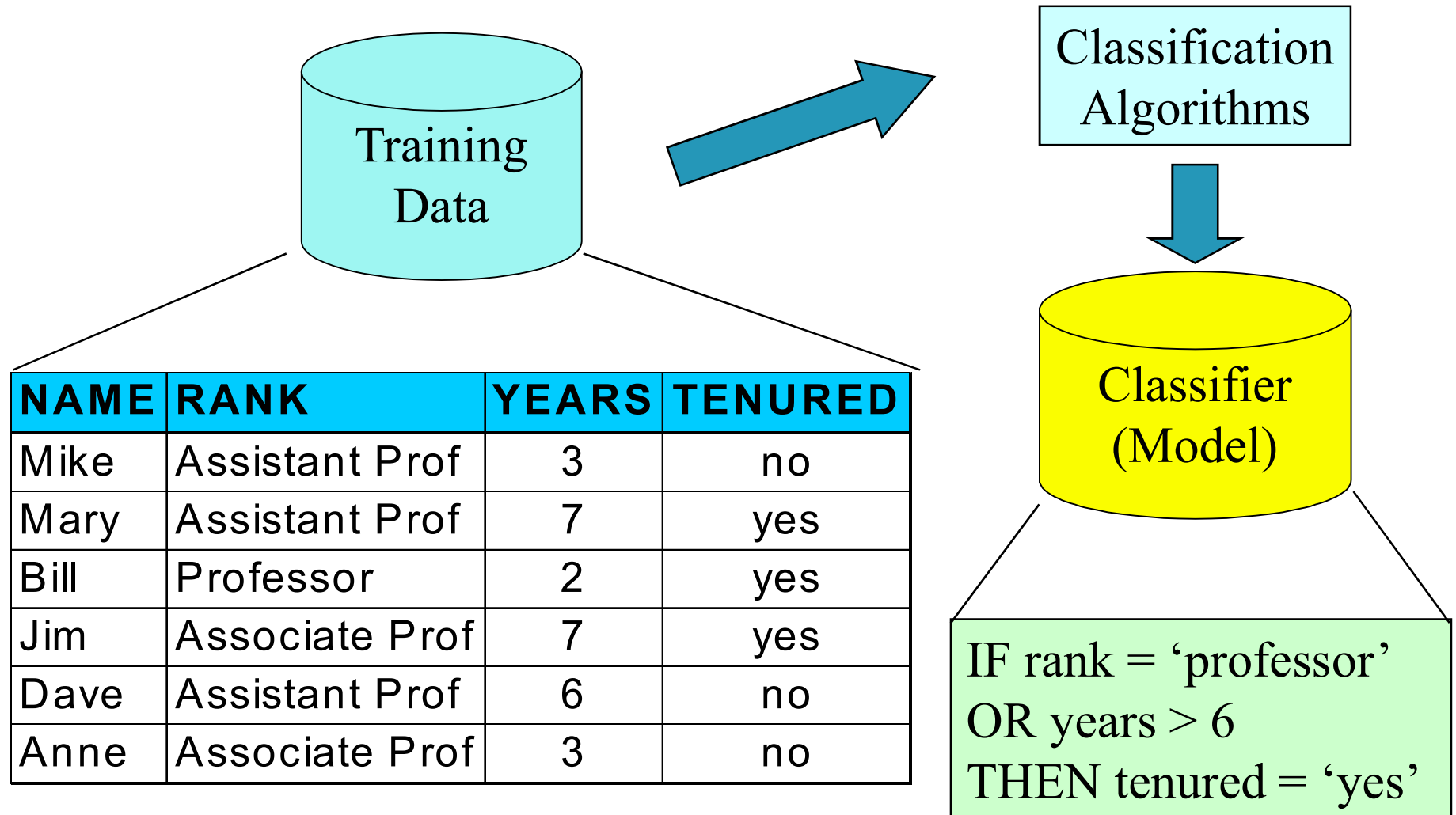
Prediction Problems: Classification vs. Numeric Prediction

- **Classification**
 - predicts categorical class labels (discrete or nominal)
 - classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute and uses it in classifying new data
- ✓ **Numeric Prediction (Regression) – already discussed**
 - models continuous-valued functions, i.e., predicts unknown or missing values
- **Typical applications**
 - Credit/loan approval:
 - Medical diagnosis: if a tumor is cancerous or benign
 - Fraud detection: if a transaction is fraudulent
 - Web page categorization: which category it is

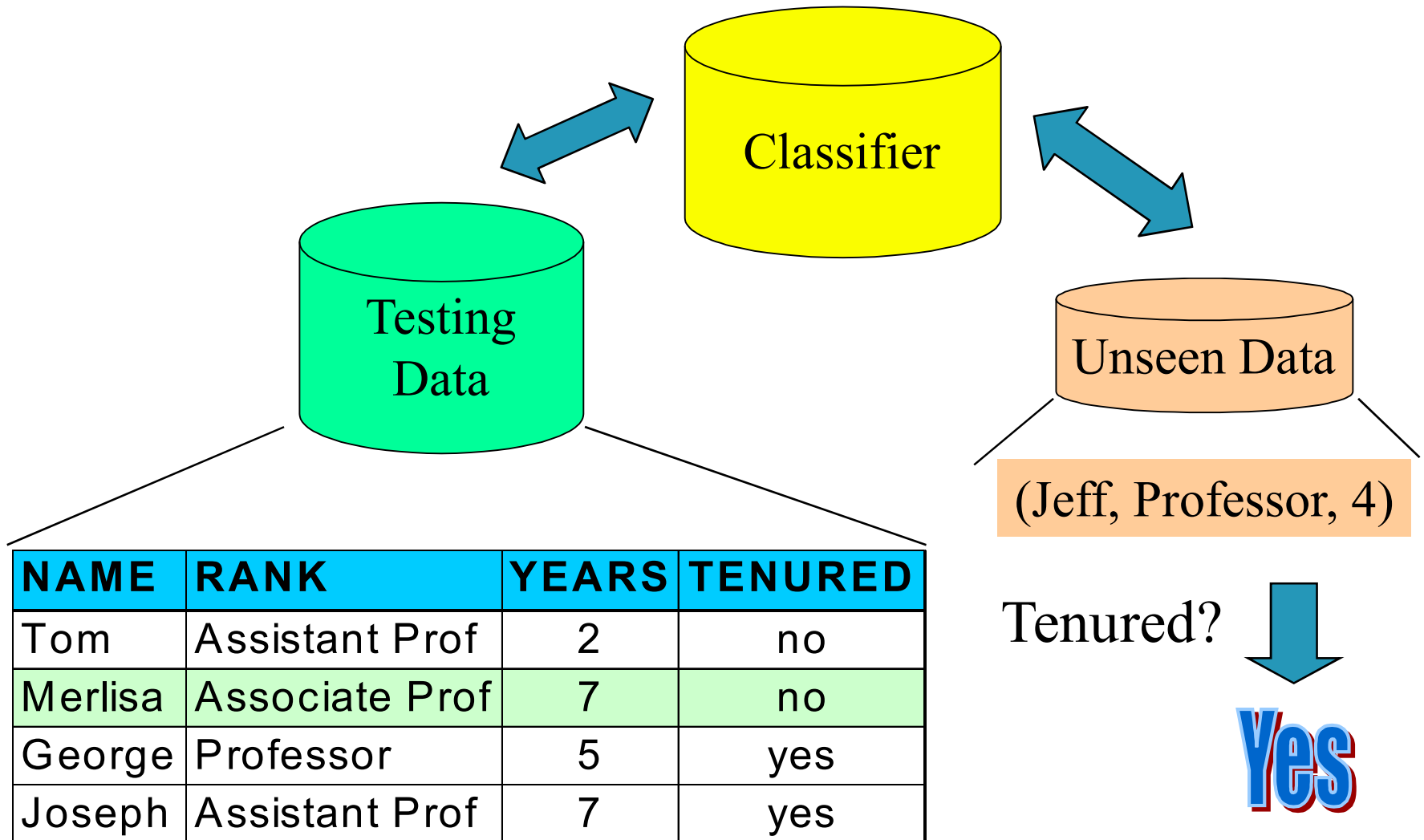
Classification—A Two-Step Process

- **Model construction**: describing a set of predetermined classes
 - Each tuple/sample is assumed to belong to a predefined class, as determined by the **class label attribute**
 - The set of tuples used for model construction is **training set**
 - The model is represented as classification rules, decision trees, or mathematical formulae
- **Model usage**: for classifying future or unknown objects
 - **Estimate accuracy** of the model
 - The known label of test sample is compared with the classified result from the model
 - **Accuracy** rate is the percentage of test set samples that are correctly classified by the model
 - **Test set** is independent of training set (otherwise overfitting)
 - If the accuracy is acceptable, use the model to **classify new data**
- Note: If *the test set* is used to select models, it is called **validation (test) set**


Process (1): Model Construction



Process (2): Using the Model in Prediction



Chapter 8. Classification: Basic Concepts

- Classification: Basic Concepts
- Decision Tree Induction 
- Bayes Classification Methods
- Model Evaluation and Selection
- Techniques to Improve Classification Accuracy:
Ensemble Methods
- Summary

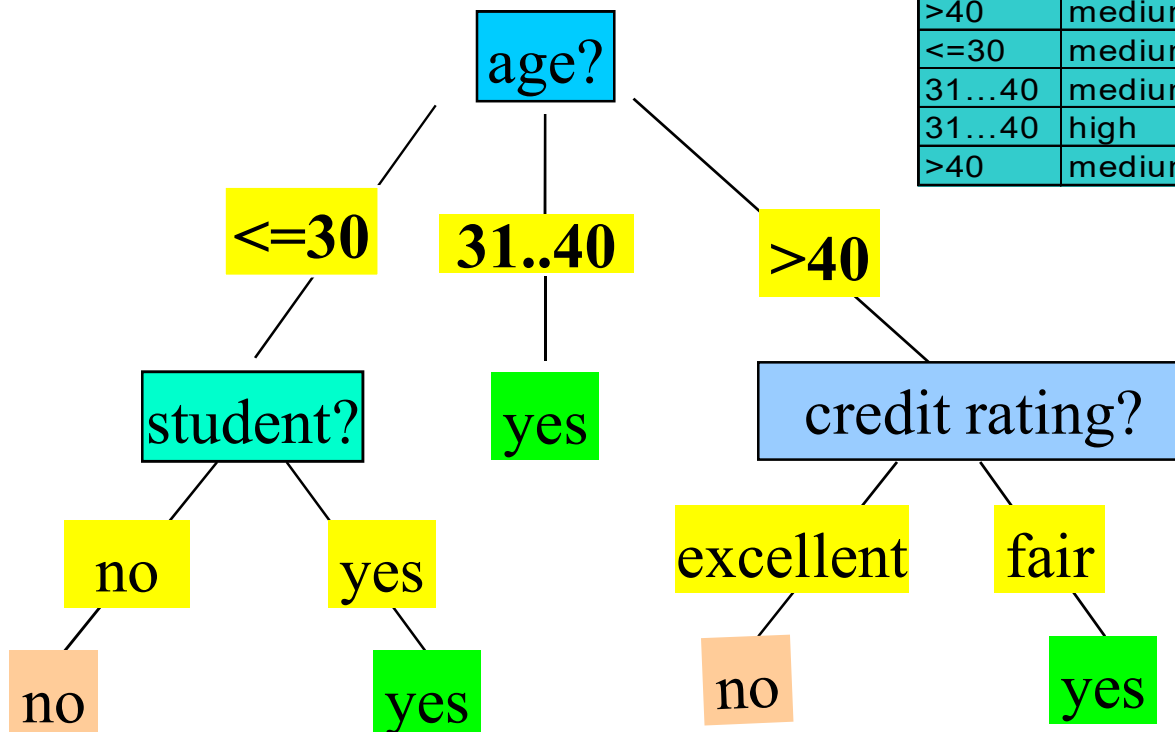
Let's Play



Decision Tree Induction: An Example

- ❑ Training data set:
Buys_computer
- ❑ The data set follows an example of Quinlan's ID3
- ❑ Resulting tree:

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

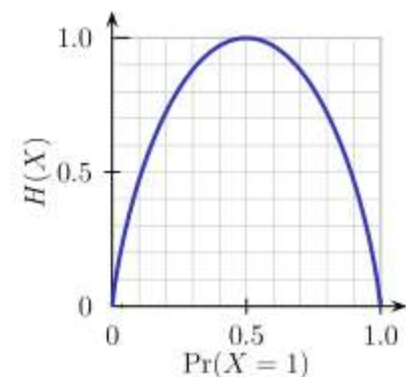


Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
 - Tree is constructed in a **top-down recursive divide-and-conquer manner**
 - At start, all the training examples are at the root
 - Attributes are categorical (if continuous-valued, they are discretized in advance)
 - Examples are partitioned recursively based on selected attributes
 - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., **information gain**)
- Conditions for stopping partitioning
 - All samples for a given node belong to the same class
 - There are no remaining attributes for further partitioning – **majority voting** is employed for classifying the leaf
 - There are no samples left

Brief Review of Entropy

- Entropy (Information Theory)
 - A measure of uncertainty associated with a random variable
 - Calculation: For a discrete random variable Y taking m distinct values $\{y_1, \dots, y_m\}$,
 - $H(Y) = -\sum_{i=1}^m p_i \log(p_i)$, where $p_i = P(Y = y_i)$
 - Interpretation:
 - Higher entropy \Rightarrow higher uncertainty
 - Lower entropy \Rightarrow lower uncertainty
- Conditional Entropy
 - $H(Y|X) = \sum_x p(x)H(Y|X = x)$



m = 2

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i , estimated by $|C_{i,D}|/|D|$

- **Expected information** (entropy) needed to classify a tuple in D :

$$Info(D) = -\sum_{i=1}^m p_i \log_2(p_i)$$

- **Information** needed (after using A to split D into v partitions) to classify D :

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$

- **Information gained** by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Buys_computer data

	age	income	student	credit_rating	buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31...40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31...40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31...40	medium	no	excellent	yes
13	31...40	high	yes	fair	yes
14	>40	medium	no	excellent	no

■ Class P: buys_computer = “yes”

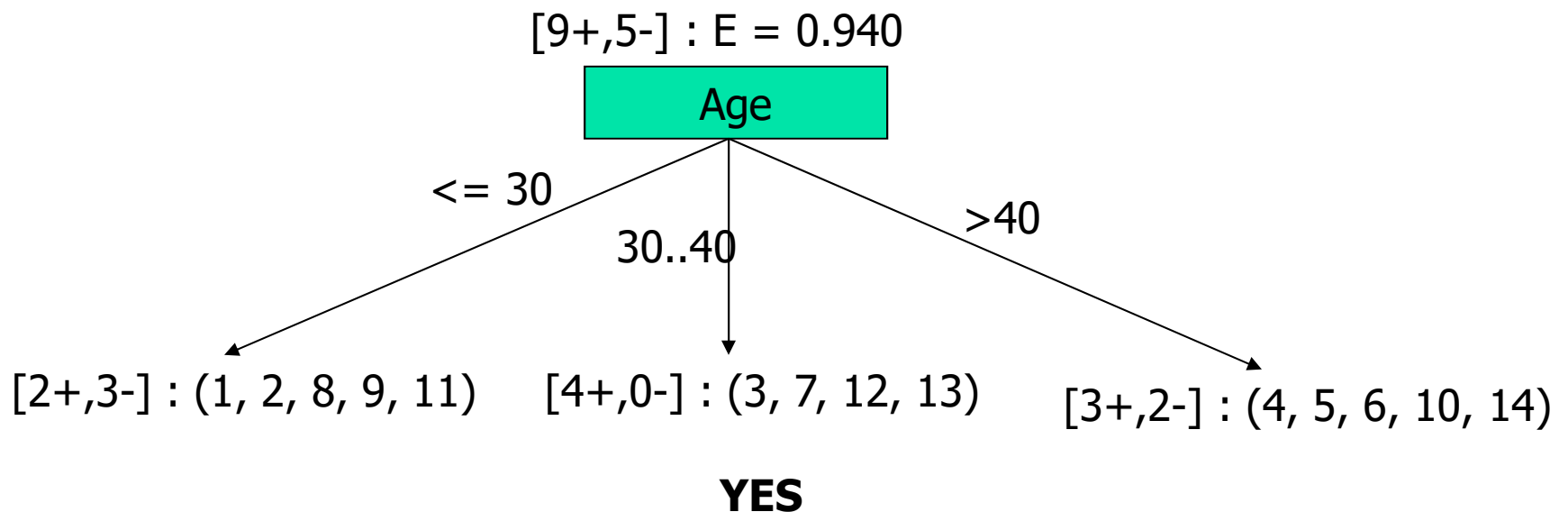
■ Class N: buys_computer = “no”

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

The Buys_Computer Example

$$\begin{aligned} \text{Info}_{age}(D) &= \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) \\ &\quad + \frac{5}{14} I(3,2) = 0.694 \end{aligned}$$

$$\text{Gain}(age) = \text{Info}(D) - \text{Info}_{age}(D) = 0.246$$



The Buys_Computer Example (cont'd)

- Similarly, compute the Gain for the other attributes, i.e. Student, Income and Credit_rating.
- Best Attribute for root?
 - $Gain(Age) = 0.246$
 - $Gain(Student) = ?$
 - $Gain(Credit_rating) = ?$
 - $Gain(Income) = ?$

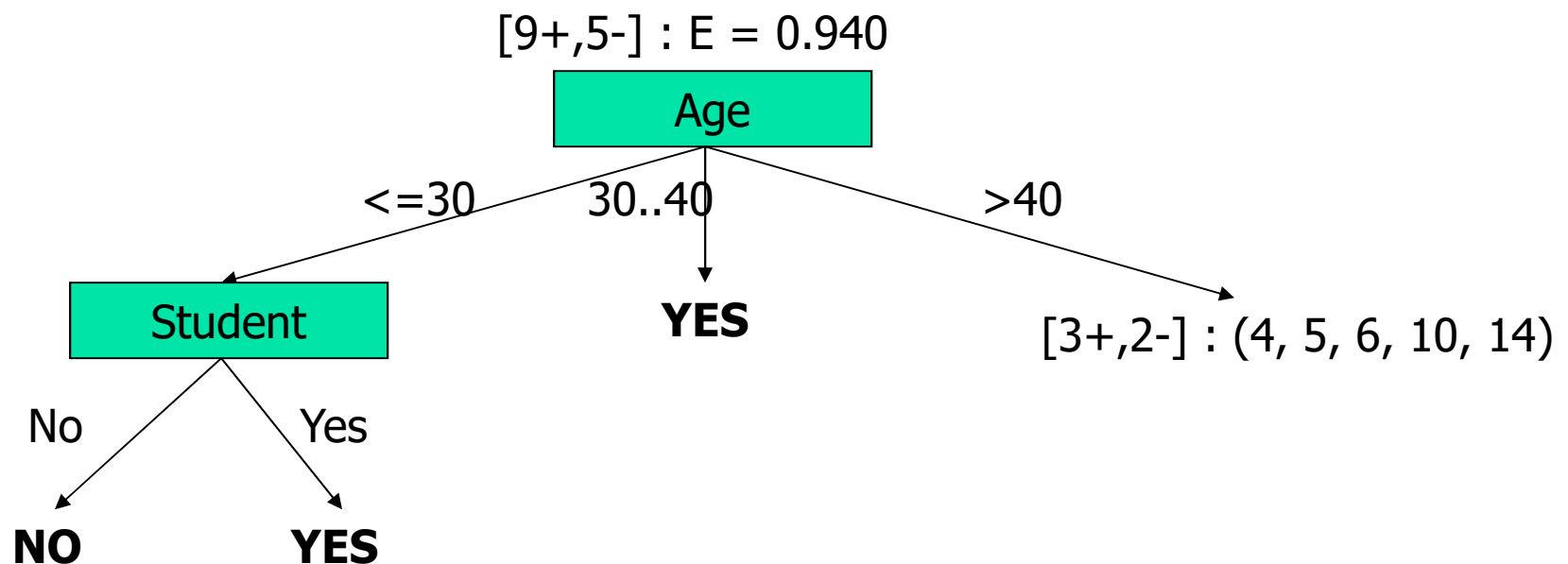
The Buys_Computer Example (cont'd)

- Similarly, compute the Gain for the other attributes, i.e. Student, Income and Credit_rating.
- Best Attribute for root?
 - **$\text{Gain}(\text{Age}) = 0.246$**
 - $\text{Gain}(\text{Student}) = 0.151$
 - $\text{Gain}(\text{Credit_rating}) = 0.048$
 - $\text{Gain}(\text{Income}) = 0.029$

The Buys_Computer Example (cont'd)

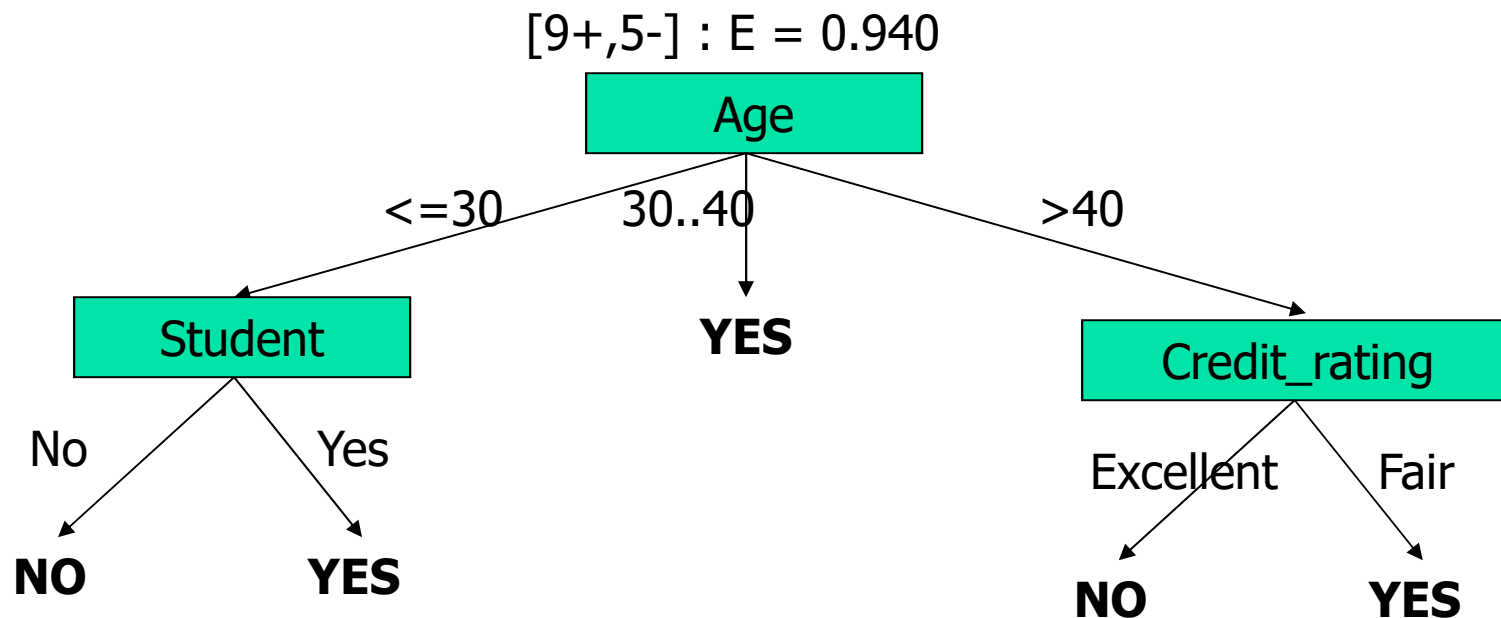
- Best Attribute for sub-tree Age≤30?

- ***Gain(Student) = 0.971***
- *Gain(Credit_rating) = 0.020*
- *Gain(Income) = 0.571*



The Buys_Computer Example (cont'd)

- Best Attribute for sub-tree Age>40?
 - $\text{Gain}(D, \text{Student}) = 0.020$
 - $\text{Gain}(D, \text{Credit_rating}) = 0.971$
 - $\text{Gain}(D, \text{Income}) = 0.020$



Weka Data Mining Tool

- Let's try: classification tab → J48

The screenshot shows the Weka Explorer window with the 'Classify' tab selected. The 'Current relation' is 'breast-cancer' with 10 attributes and 286 instances. The 'Selected attribute' is 'age', which is a nominal attribute with 6 distinct values. The 'Attributes' list on the left includes 'age', 'menopause', 'tumor-size', 'inv-nodes', 'node-caps', 'deg-malig', 'breast', 'breast-quad', and 'irradiat'. The 'Status' bar at the bottom shows 'OK'. A bar chart visualization is displayed at the bottom right, showing the distribution of the 'age' attribute across the dataset.

Current relation

Relation: breast-cancer
Instances: 286
Attributes: 10
Sum of weights: 286

Selected attribute

Name: age
Missing: 0 (0%)
Distinct: 6
Type: Nominal
Unique: 1 (0%)

No.	Label	Count	Weight
1	10-19	0	0.0
2	20-29	1	1.0
3	30-39	36	36.0
4	40-49	90	90.0
5	50-59	96	96.0

Attributes

All None Invert Pattern

No.	Name
1	<input checked="" type="checkbox"/> age
2	<input type="checkbox"/> menopause
3	<input type="checkbox"/> tumor-size
4	<input type="checkbox"/> inv-nodes
5	<input type="checkbox"/> node-caps
6	<input type="checkbox"/> deg-malig
7	<input type="checkbox"/> breast
8	<input type="checkbox"/> breast-quad
9	<input type="checkbox"/> irradiat

Remove

Status

OK

Log x 0

Computing Information-Gain for Continuous-Valued Attributes

- Let attribute A be a continuous-valued attribute
- Must determine the *best split point* for A
 - Sort the value A in increasing order
 - Typically, the midpoint between each pair of adjacent values is considered as a possible *split point*
 - $(a_i + a_{i+1})/2$ is the midpoint between the values of a_i and a_{i+1}
 - The point with the *minimum expected information requirement* for A is selected as the split-point for A
- Split:
 - D1 is the set of tuples in D satisfying $A \leq \text{split-point}$, and D2 is the set of tuples in D satisfying $A > \text{split-point}$

Dealing with Continuous Attributes

- Entropy only needs to be evaluated between points of different classes

Income (\$k)	7.5	8.3	8.9	10.7	12	13.2	15.4	21	24.7
Class	no	no	no	yes	yes	yes	yes	no	no

Gain Ratio for Attribute Selection (C4.5)

- Information gain measure is biased towards attributes with a large number of values
- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|} \right)$$

- $GainRatio(A) = Gain(A)/SplitInfo(A)$

- Ex. $SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2 \left(\frac{4}{14} \right) - \frac{6}{14} \times \log_2 \left(\frac{6}{14} \right) - \frac{4}{14} \times \log_2 \left(\frac{4}{14} \right) = 1.557$

- $gain_ratio(income) = 0.029/1.557 = 0.019$

- The attribute with the maximum gain ratio is selected as the splitting attribute

Comparing Attribute Selection Measures

- The two measures, in general, return good results but
 - **Information gain**
 - biased towards multivalued attributes
 - **Gain ratio:**
 - tends to prefer unbalanced splits in which one partition is much smaller than the others

Overfitting and Tree Pruning

- Overfitting: An induced tree may overfit the training data
 - Too many branches, some may reflect anomalies due to noise or outliers
 - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
 - Prepruning: *Halt tree construction early*
 - Do not split a node if this would result in the goodness measure falling below a threshold
 - Difficult to choose an appropriate threshold
 - Postpruning: *Remove branches* from a “fully grown” tree—get a sequence of progressively pruned trees
 - Use a set of data different from the training data to decide which is the “best pruned tree”

Enhancements to Basic Decision Tree Induction

- Allow for **continuous-valued attributes**
 - Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals
- Handle **missing attribute values**
 - Assign the most common value of the attribute
 - Assign probability to each of the possible values
- **Attribute construction**
 - Create new attributes based on existing ones that are sparsely represented
 - This reduces fragmentation, repetition, and replication

Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why is decision tree induction popular?
 - relatively faster learning speed (than other classification methods)
 - convertible to simple and easy to understand classification rules
 - comparable classification accuracy with other methods

BOAT (Bootstrapped Optimistic Algorithm for Tree Construction)

- Use a statistical technique called *bootstrapping* to create several smaller samples (subsets), each fits in memory
- Each subset is used to create a tree, resulting in several trees
- These trees are examined and used to construct a new tree T'
 - It turns out that T' is very close to the tree that would be generated using the whole data set together