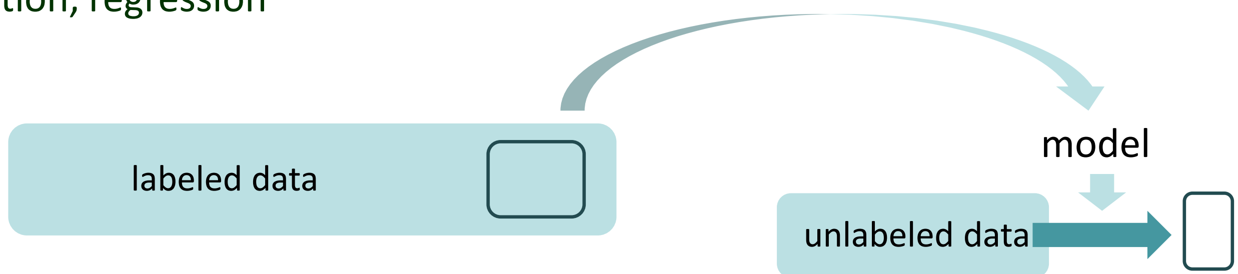
Classification

Young-Rae Cho, Ph.D.
Associate Professor
Division of Software / Division of Digital Healthcare
Yonsei University – Mirae Campus

Supervised vs. Unsupervised Learning

❑ **Supervised Learning**

- Training data (observations, measurement, etc.) are given
- Training data include class labels predefined
- Find rules or models of class labels of training data
- New data are classified based on the rules or models
- Example: classification, regression



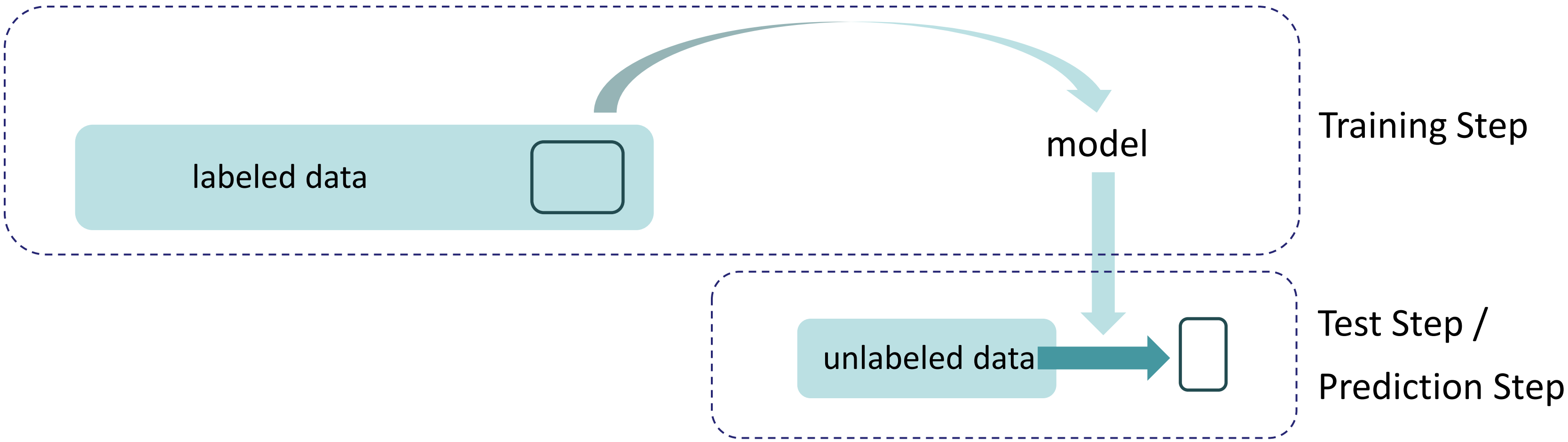
❑ **Unsupervised Learning**

- No training data are given
- New data are classified without any training data
- Example: clustering, pattern mining

Classification vs. Regression

❑ Classification

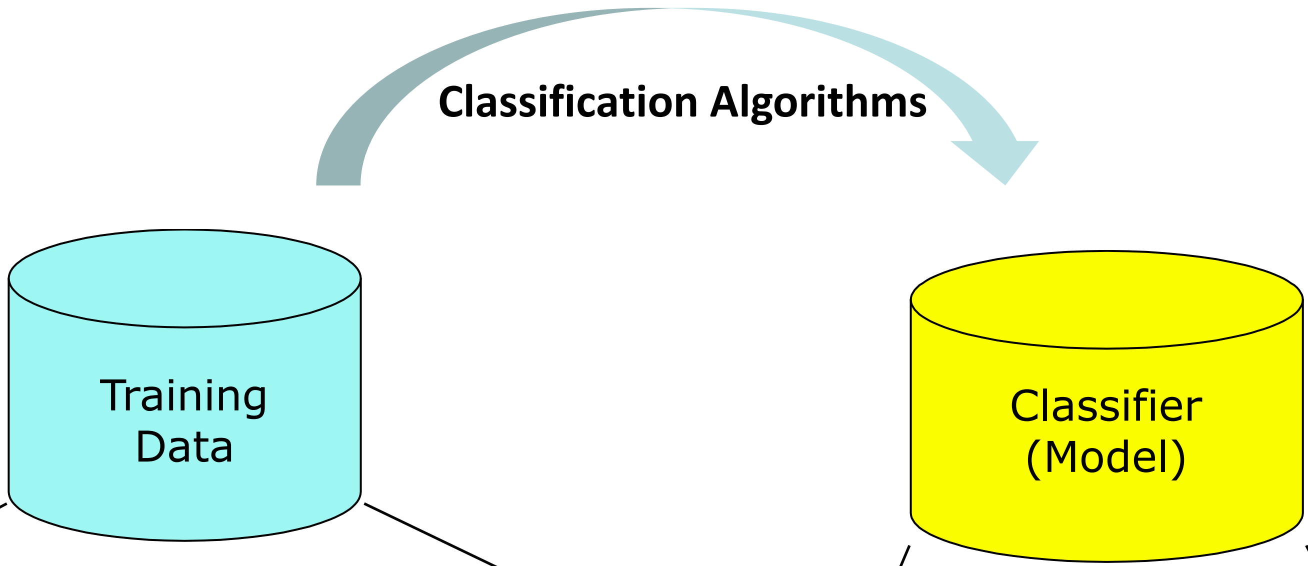
- Training class labels in attributes of a training data set
- Predicts class labels of a new data set based on the rules or models of class labels of the training data set



❑ Regression

- Modeling continuous-valued functions for a data set
- Predicts unknown or missing values in the data set

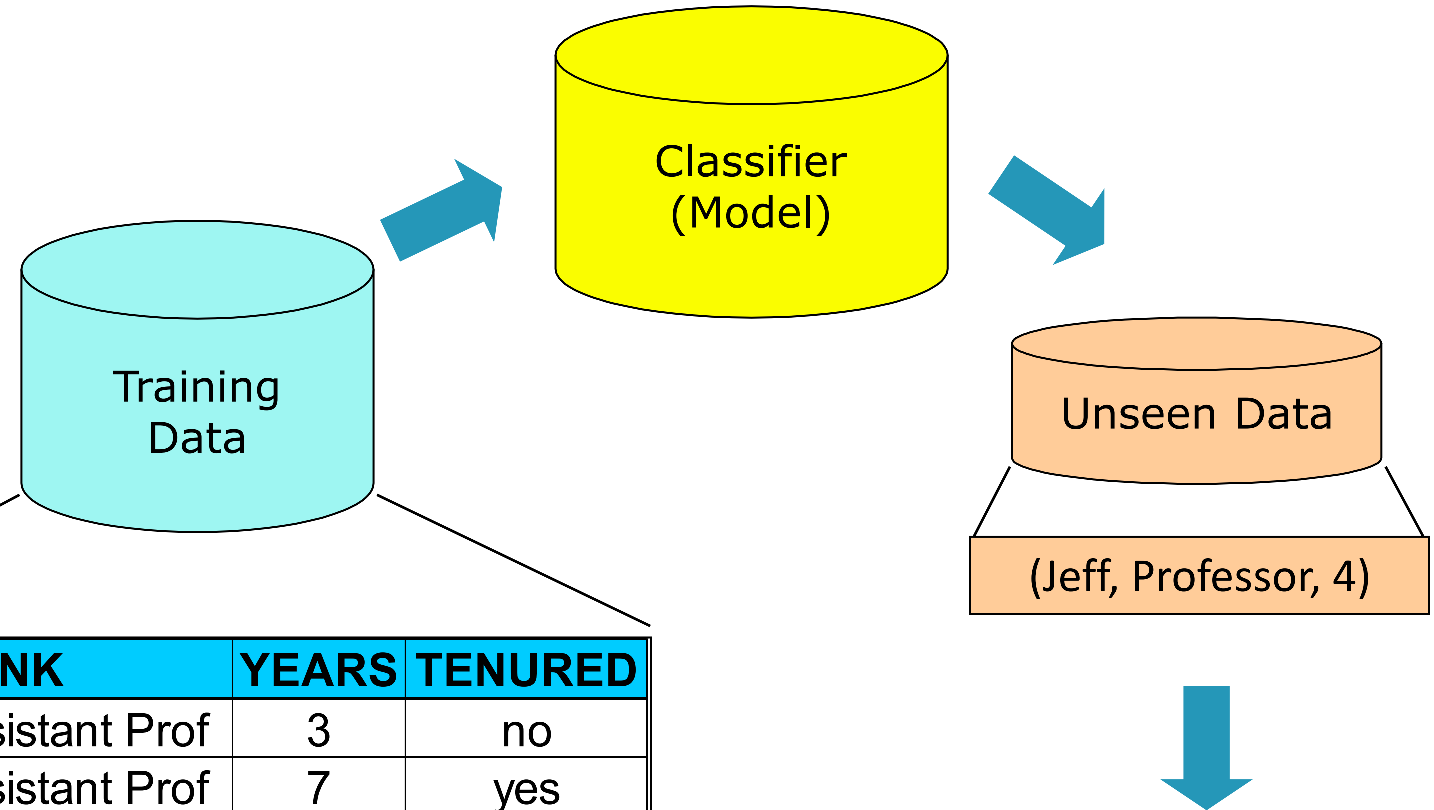
Classification Step 1: Training



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Classification Step 2: Prediction



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Tenured? Yes

Issues in Classification

- **Accuracy**
  - Training accuracy and prediction accuracy
- **Efficiency**
  - Training time and prediction time
- **Robustness**
  - Handling noise and missing values
- **Scalability**
  - Efficient memory usage in disk-resident databases
- **Interpretability**
  - Understanding of classifying models

Overview

1. <u>Decision Tree Induction</u>
2. Bayesian Classification
3. k-Nearest Neighbor Learning
4. Rule-Based Classification
5. Pattern-Based Classification
6. Classification Accuracy Measures

Decision Tree Induction

❑ Decision Tree Structure

- Each non-leaf node represents ??
  - Attributes should be categorical (if continuous, discretize the values)
    → Each attribute should have a finite number of values
- Each leaf node represents ??
- Each edge represents ??

❑ Decision Tree Construction

- A decision tree is constructed in a top-down recursive manner
- An attribute is selected by an information-theoretic measure
- The training data are recursively partitioned on the selected attribute at each round

❑ Classification Process

- The new data are classified by tracing the decision tree from the root

Decision Tree Construction

❑ Process

1) Put all data at the root node
2) Recursively, select an attribute and partition the data-set into subsets as child nodes, until having a stopping condition

How to select an attribute at each step ?

❑ Stopping Conditions

- If all data samples for a given node in the tree belong to the same class
- If there are no remaining attributes for further partitioning (majority voting is employed for classifying data in the leaf node)
- There are no data samples left

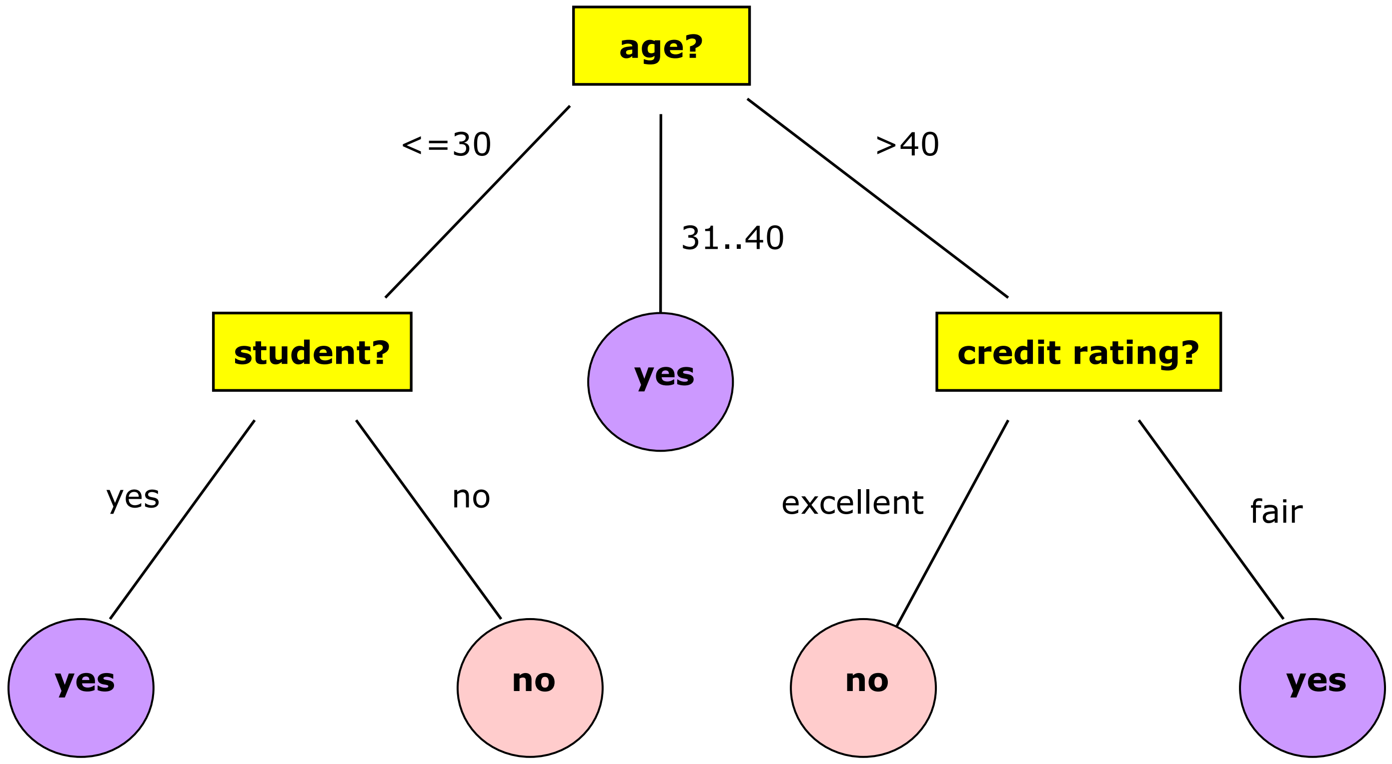
Example of Training Data

❑ Training Data Set

age	income	student	credit_rate	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

Example of Decision Tree

❑ **Output Decision Tree for "buys_computer"**



ID3 Algorithm

Main Idea

- Attribute selection measure during decision tree construction

  → Select the attribute with <u>the highest information gain</u>
- Let p_i be the probability that an arbitrary record in D belongs to class C_i
- Expected information (entropy):

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

- Information after using an attribute A to split D into v partitions

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

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

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

Example of Information Gain

- **Information**
  - 9 “yes”es and 5 “no”s, in buy_computer
  - Info(D) =
- **Information after Splitting by “age”**
  - $\text{Info}_{age}(D)$ =
- **Information Gain by “age”**
  - Gain(age) = Info(D) - $\text{Info}_{age}(D)$ =
- **Information Gain by other attributes**
  - Gain(income) =
  - Gain(student) =
  - Gain(credit_rating) =

age	buys_computer
<=30	no
<=30	no
31~40	yes
> 40	yes
> 40	yes
> 40	no
31~40	yes
<=30	no
<=30	yes
> 40	yes
<=30	yes
31~40	yes
31~40	yes
> 40	no

C4.5 Algorithm

❑ Main Idea

- An extension of the ID3 algorithm
- Information gain measure in ID3 is biased towards attributes with a large number of values
- Uses gain ratio to overcome the problem (normalizing information gain)

  → Select the attribute with the highest gain ratio
- Split information for normalization of information gain

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

- **Gain Ratio**(A) = Gain(A) / SplitInfoA(D)

Example of Gain Ratio

- **Split Information by "age"**
  - $SplitInfo_{age}(D)$ =

- **Gain Ratio by "age"**
  - GainRatio(age) =

- **Gain Ratio by other attributes**
  - GainRatio(income) =
  - GainRatio(student) =
  - GainRatio(credit_rating) =

age	buys_computer
<=30	no
<=30	no
31~40	yes
> 40	yes
> 40	yes
> 40	no
31~40	yes
<=30	no
<=30	yes
> 40	yes
<=30	yes
31~40	yes
31~40	yes
> 40	no

CART (Classification and Regression Trees)

Main Idea

- Attribute selection during decision tree construction

  → Select the attribute with the greatest difference of Gini index
- Gini index: a measure of inequality

$$Gini(D) = 1 - \sum_{j=1}^{m} p_j^2$$

- If a data set D is split on the attribute A into two subsets D1 and D2,

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

- **ΔGini(A)** by the binary split on A

$$\Delta Gini(A) = Gini(D) - Gini_A(D)$$

Example of Gini Index

- **Gini Index in "buy_computer"**
  - 9 "yes"es and 5 "no"s, in buy_computer
  - Gini(D) =
- **Gini Index after Splitting by "age"**
  - $Gini_{age}(D)$ =
- **ΔGini by "age"**
  - ΔGini(age) =
- **Gini Index after Splitting by other attributes**
  - ΔGini(income) =
  - ΔGini(student) =
  - ΔGini(credit_rating) =

age	buys_computer
<=30	no
<=30	no
31~40	yes
> 40	yes
> 40	yes
> 40	no
31~40	yes
<=30	no
<=30	yes
> 40	yes
<=30	yes
31~40	yes
31~40	yes
> 40	no

Problems of Attribute Selection

- **Information Gain**
  - Biased towards the attributes with a large number of values
- **Gain Ratio**
  - Biased towards the unbalanced splits in which one partition is much larger than the others
- **Gini Index**
  - Biased to multi-valued attributes

Summary of Decision Tree Induction

- **Strength**
  - Simple and easy to understand classification rules
  - Able to use SQL queries to access databases

- **Weakness**
  - Not able to handle continuous attributes

    → Partition the continuous attribute values into a discrete set of intervals
  - Overfitting
  - Limitation of scalability – restriction of the training data size

    → Scalable algorithms: SLIQ, SPRINT, RainForest

Overfitting

- **Underfitting vs. Overfitting**
  - Underfitting: the classifier performs poorly on the training data
  - Overfitting: the classifier performs well on the training data,
    but performs poorly on classifying new data

- **Example of Overfitting**
  - Too many branches of decision tree by reflecting anomalies due to noise or outliers

- **Solving Overfitting**
  - Prepruning: Halt tree construction early
    - Stop splitting a node if the result is 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
    - Inefficient

RainForest

❑ Main Idea

- Create AVC-set / AVC-group, which fit in memory, by scanning database

❑ AVC (Attribute-Value, Class-label)

- AVC-set of an attribute X : the projection of the training dataset on X and class labels where counts of individual class labels are aggregated
- AVC-group of a node n : the set of AVC-sets of all predictor attributes at n

age	buy_computer	
	yes	no
<=30	3	2
31..40	4	0
>40	3	2

❑ Reference

- Gehrke, J., et al., “RainForest – a framework for fast decision tree construction of large datasets” In Proceeding of VLDB (1998)

Overview

1. Decision Tree Induction
2. <u>Bayesian Classification</u>
3. k-Nearest Neighbor Learning
4. Rule-Based Classification
5. Pattern-Based Classification
6. Classification Accuracy Measures

Bayesian Classification

❑ Main Idea

- A statistical classifier: performs probabilistic prediction

  → Outputs the probability of class membership

- Utilizes the Bayesian Theorem

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

  - H : a hypothesis
  - X : an evidence
  - P(H|X) : posterior probability
  - P(H) : prior probability
  - P(X|H) : likelihood

- Assumes that the effect of an attribute value on a given class is independent of the values of the other attributes

Bayesian Classification – cont'

❑ Classification Components

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

- X : sample data (class label is unknown)
- H : a hypothesis that X belongs to class C
- P(H|X) : the probability that the hypothesis holds given X
- P(H) : initial probability that any random data belongs to class C
- P(X) : probability that the sample data is observed
- P(X|H) : probability of observing the sample X, given that the hypothesis holds

❑ Classification Process

- Predicts X belongs to C_i iff the probability $P(C_i|X)$ is the highest among all the $P(C_k|X)$ for all k classes
- Practical difficulty: requires initial knowledge of many probabilities,
  → significant computational cost

Naïve Bayesian Classifier

Computational Efficiency

- Classification is to derive the maximum posterior probability, $P(C_i|X)$

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

- Suppose P(X) is constant for all classes, maximize

$$P(C_i|X) = P(X|C_i)P(C_i)$$

Assumption of Conditional Independence

- Attributes are conditionally independent between attributes

$$P(X \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)$$

- If x_k is categorical, $P(x_k|C_i)$ is the number of objects in C_i having value x_k divided by the number of objects of C_i
- If x_k is continous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

Example of Training Data

❑ Training Data Set

age	income	student	credit_rate	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

Example of Classification Results

- **Test Data (No Class Label)**
  - X = (age<=30 , income=medium, student=yes, credit_rating=fair)

- **Hypothesis that X belongs to buys_computer = "yes"**
  - $P(C_i)$ = P(buys_computer = "yes") =
  - $P(X|C_i)$
    - P(age = "<=30" | buys_computer = "yes") =
    - P(income = "medium" | buys_computer = "yes") =
    - P(student = "yes" | buys_computer = "yes") =
    - P(credit_rating = "fair" | buys_computer = "yes") =
  - $P(C_i|X) = P(X|C_i) \times P(C_i)$
    - P(buys_computer = "yes" | X) =

Example of Classification Results – cont'

- **Hypothesis that X belongs to buys_computer = "no"**
  - $P(C_i)$ = P(buys_computer = "no") =
  - $P(X|C_i)$
    - P(age = "<=30" | buys_computer = "no") =
    - P(income = "medium" | buys_computer = "no") =
    - P(student = "yes" | buys_computer = "no") =
    - P(credit_rating = "fair" | buys_computer = "no") =
  - $P(C_i|X) = P(X|C_i) \times P(C_i)$
    - P(buys_computer = "no" | X) =

Summary of Naïve Bayesian Classifier

- **Strength**
  - Easy to implement
  - Good results in most of the cases

- **Weakness**
  - Assumption of conditional independence of attributes

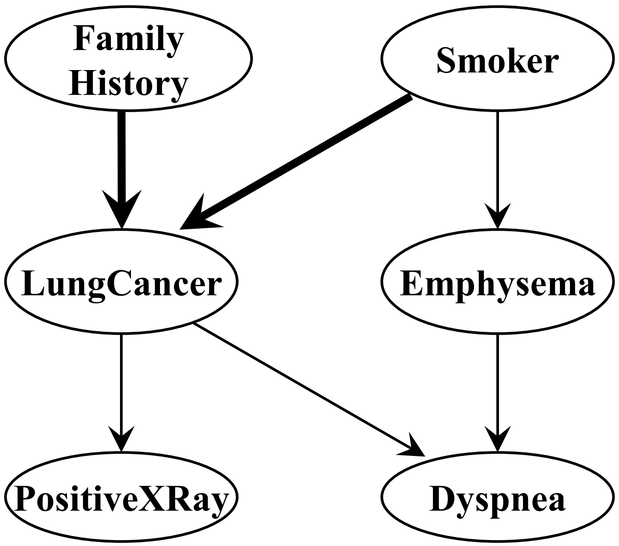
    → Loss of accuracy
  - In practice, dependencies exist between attributes

    → Dealing with dependencies: Bayesian belief networks

Bayesian Belief Networks

- **Main Idea**
  - Represents dependency among attributes by training data in Bayesian networks

- **Bayesian Network**
  - Directed acyclic graph (DAC)



  - Conditional probability table

	FH,S	FH,~S	~FH,S	~FH,~S
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

Overview

1. Decision Tree Induction
2. Bayesian Classification
3. k-Nearest Neighbor Learning
4. Rule-Based Classification
5. Pattern-Based Classification
6. Classification Accuracy Measures

k-Nearest Neighbor Learning (kNN)

Main Idea

- Lazy learning (or, instance-based learning)
  - → Store the training data and wait until it is given the data for prediction
  - → Less time in training but more time in predicting
- All instances (data objects) correspond to points in the n-D space
- The nearest neighbors are found by a distance function
- The distance function can be defined for numerical or categorical values

Learning Process

- Searches the k closest neighbor instances of the unknown instance
- For categorical values, the unknown instance is assigned the most common class among k neighbors
- For numerical values, the unknown instance is assigned the mean of k neighbors

Distance Functions

- **Numerical Attributes**
  - Minkowski distance, $$d = \left(\sum_{i=1}^{n} | x_i - y_i |^p \right)^{1/p}$$
    - Euclidean distance when p=2, and Manhattan distance, when p=1

- **Categorical Attributes**
  - Jaccard coefficient, $$d = \frac{|X \Delta Y|}{|X \cup Y|} = 1 - \frac{|X \cap Y|}{|X \cup Y|}$$
    - XΔY: the symmetric difference between X and Y

- **Boolean Attributes**
  - If symmetric, $$d = \frac{r+s}{q+r+s+t}$$
  - If asymmetric, $$d = \frac{r+s}{q+r+s}$$

contingency table

	1	0	sum
1	q	r	q+r
0	s	t	s+t
sum	q+s	r+t	p

Summary of kNN

- **Strength**
  - Robust to noisy data by averaging k neighbors

- **Weakness**
  - Consider all attributes equally by a distance function
    - → Might be dominated by irrelevant attributes
    - → Might need to eliminate irrelevant attributes
  - Need pre-determined the k value
    - → Small k makes sensitive to noise
    - → Large k makes inaccurate
    - → Might need to weight each of the k neighbors according to their distance

Overview

1. Decision Tree Induction
2. Bayesian Classification
3. k-Nearest Neighbor Learning
4. Rule-Based Classification
5. Pattern-Based Classification
6. Classification Accuracy Measures

Rule-Based Classification

Main Idea

- Find rules in the form of IF-THEN rules

  e.g., IF age < 30 AND student = yes, THEN buy_computer = yes

  e.g., IF student = yes AND income = low, THEN buy_computer = no

How to find rules ?

Learning Process

- Training step: generating a set of rules
- Prediction step: classifying a new data by the rules applied

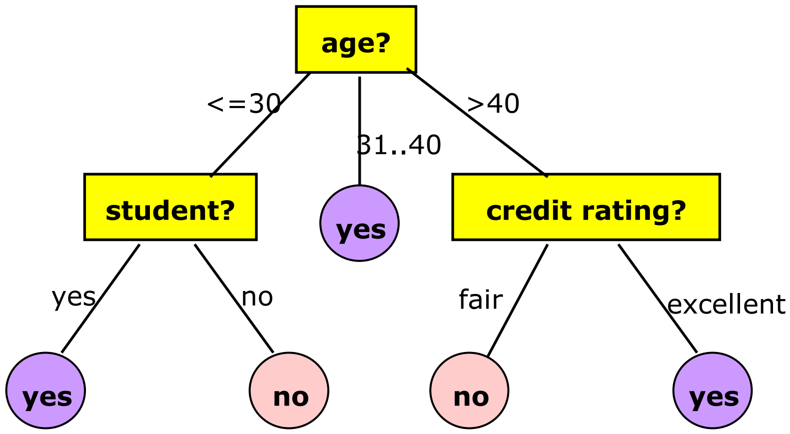
Issue

- If more than one rule are triggered, need conflict resolution
  - Attribute size ordering: decreasing order of the number of attributes in the rules
  - Rule-based ordering: decreasing order of rule quality

Rule Extraction from Decision Tree

❑ Main Idea

- Each rule can be created by each path from the root to a leaf
- Each attribute-value pair along a path forms a conjunction with “AND”
- Rules are mutually exclusive



❑ Examples

- IF age = young AND student = yes, THEN buys_computer = yes
- IF age = young AND student = no, THEN buys_computer = no
- IF age = mid-age, THEN buys_computer = yes
- IF age = old AND credit_rating = excellent, THEN buys_computer = yes
- IF age = old AND credit_rating = fair, THEN buys_computer = no

Rule Extraction by Sequential Covering

Main Idea

- Each rule is learned sequentially

Sequential Covering Algorithm

1) Learn a rule, and remove the data covered by the rule
2) Repeat (1) until reaching a termination condition
3) Repeat (1) and (2) for each class

Rule Learning

- Starts with the most general rule possible, and grows the rule in a general-to-specific manner
- Adds new attributes into the rule by selecting the one that most improves the rule quality

Termination Condition

- There are no more training data
- It does not reach the rule quality threshold

Rule Quality Measures

- **Coverage & Accuracy**
  - n_{covers} = the number of data objects covered by the rule R
  - $n_{correct}$ = the number of data objects correctly classified by R
  - coverage(R) = $n_{covers}/|D|$ where D is the training data set
  - accuracy(R) = $n_{correct}/n_{covers}$

- **FOIL Gain**
  - FOIL (First Order Inductive Learning)
  - Similar to information gain
  - pos = the number of positive data objects covered by the rule R
  - pos' = the number of positive data objects covered by the new rule R'
  - $$FOIL\_Gain = pos' \times \left(\log_2 \frac{pos'}{pos' + neg'} - \log_2 \frac{pos}{pos + neg} \right)$$

Overview

1. **Decision Tree Induction**

2. **Bayesian Classification**

3. **k-Nearest Neighbor Learning**

4. **Rule-Based Classification**

5. **Pattern-Based Classification**

6. **Classification Accuracy Measures**

Pattern-Based Classification

- **Main Idea**
  - Frequent patterns and their corresponding association rules are generated and analyzed for classification
  - Also called associative classification
  - Search for strong associations between frequent patterns and class labels
  - Each pattern is represented as conjunctions of attribute-value pairs based on its support and confidence

- **Methods**
  - CBA (Classification by Association)
  - CMAR (Classification based on Multiple Association Rules)
  - CPAR (Classification based on Predictive Association Rules)

CBA (Classification By Association)

❑ Main Idea

- Mining all possible association rules by their support and confidence in the form of "$p_1 \wedge p_2 \dots \wedge p_n$" → "$A_{class} = C$", called Class Association Rule (CAR)
- Difference between Association Rule Mining and CBA
  - Association Rule Mining: target is not predetermined
  - CBA: only one predetermined target
- Building a classifier with the rules according to decreasing precedence of their confidence

❑ Classification Rules

1) Find all covered CARs from the training data
2) Classify the test data with the highest confidence CAR
3) If some CARs tie, use the highest support CAR, and then the majority class

❑ References

- Liu, B., Hsu, W. and Ma, Y., "Integrating classification and association rule mining", In Proceedings of KDD (1998)

Overview

1. Decision Tree Induction
2. Bayesian Classification
3. k-Nearest Neighbor Learning
4. Rule-Based Classification
5. Pattern-Based Classification
6. Classification Accuracy Measures

Evaluation of Classification Methods

Holdout Method

- Randomly partitions the given data into a training set and a test set

Random Sampling

- Repeats the holdout method k times
- Estimates the overall accuracy by averaging the accuracy from each round

k-Fold Cross-Validation

- Randomly partitions the given data into k mutually exclusive subsets, each approximately equal size
- Measures accuracy k times using the i-th subset as a test set and the others as a training set

Leave-One-Out Cross-Validation

- k-fold cross-validation where k is the total size of data set
- One sample is left out as a test set for each round

Classification Accuracy Measures

❑ Accuracy Measures

- Confusion matrix

		Predicted class	
		C'_i	$\sim C'_i$
Actual class	C_i	true positive	false negative
	$\sim C_i$	false positive	true negative

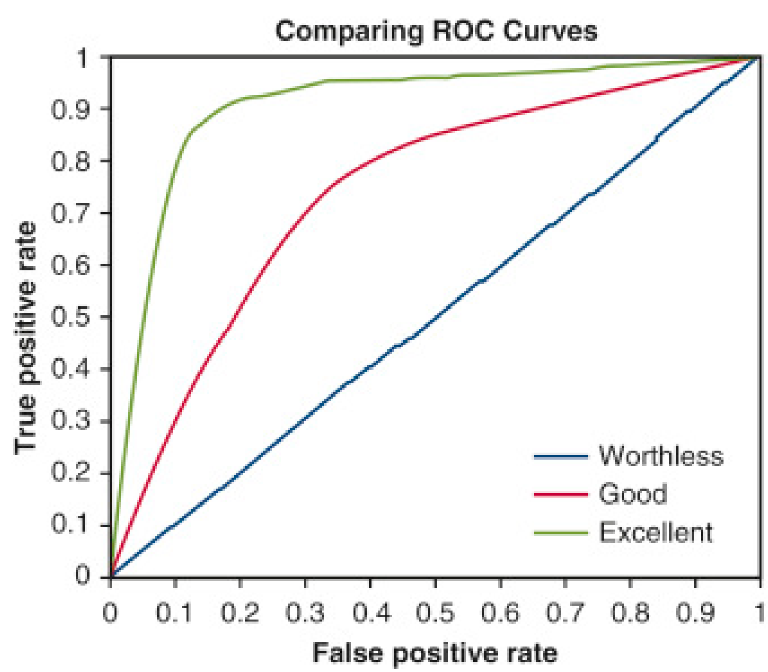
- Sensitivity (true positive rate, recall) =
- Specificity (true negative rate) =
- Positive predictive value (precision) =
- Negative predictive value =
- Accuracy = sensitivity × (tp+fn)/total + specificity × (fp+tn)/total

  =
- Error rate =

Classification Accuracy Measures – cont’

- **ROC Curve**
  - Receiver Operating Characteristic Curve
  - A graphic plot of true positive rate (sensitivity) vs. false positive rate (1-specificity)
  - A tool to show optimality of a classifier
  - The closer to the diagonal line, the less accurate the classifier is

- **AUC**
  - The area under the ROC curve
  - Represents classification accuracy



Questions?

❑ Lecture Slides on the Course Website, "https://ads.yonsei.ac.kr/faculty/data_mining"