Learning Outcomes

- Understand Classification
- Understand Prediction
- Classification issues
- Classification Methods
  - Bayesian classification
  - Association rules
  - Back-Propagation (NN)
- Prediction

2

Classification vs. Prediction

- Classification
  - predicts categorical class labels
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it to classify new data
- Prediction
  - Numeric prediction: models continuous-valued functions,
  - Example: predicts how much a given customer will spend during sales
- Typical Applications
  - credit approval, target marketing
  - medical diagnosis
  - treatment effectiveness analysis

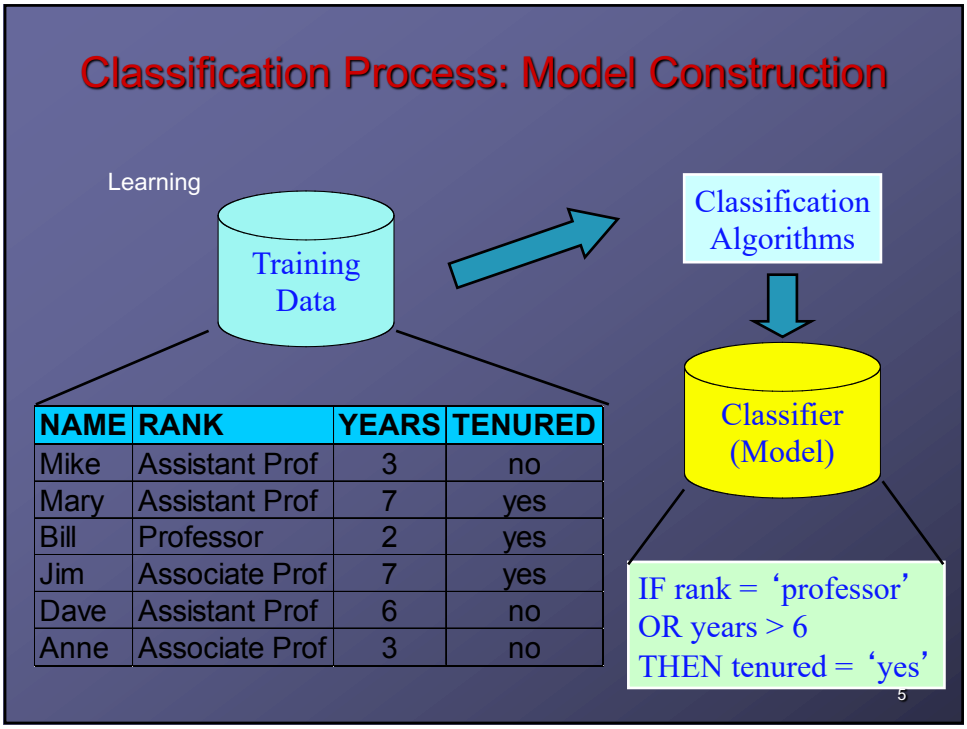
3

Classification

- Model Construction (1st Step)
  - The model should describe 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: training set
  - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage (2nd Step)
  - For classifying future or unknown objects
  - Estimate the model accuracy
    - 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 over-fitting will occur

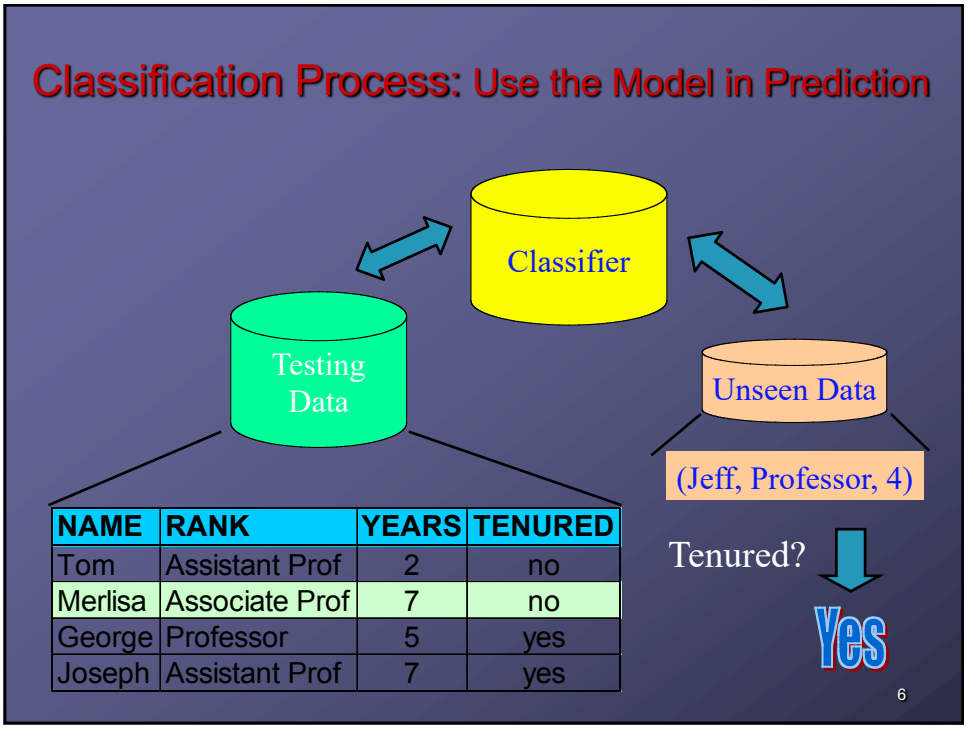
4

Classification Process: Model Construction



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

Classification Process: Use the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

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 are unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

7

Classification Issues (1): Data Preparation

- Data cleaning
  - Pre-process data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
- Data transformation
  - Generalise and/or normalise data

8

Issues (2): Evaluating Classification Methods

- Accuracy of the prediction
- Speed and scalability
  - time to construct the model
  - time to use the model
  - efficiency in disk-resident databases
- Robustness
  - handling noise and missing values
- Interpretability
  - understanding and insight provided by the model

9

Classification by Decision Tree Induction

- Decision tree
  - A flow-chart-like tree structure, where
  - Internal node denotes a test on an attribute
  - Branch represents an outcome of the test
  - Leaf nodes represent class labels or class distribution
- Decision tree generation consists of two phases
  - Tree construction
    - At start, all the training examples are at the root
    - Partition examples recursively based on selected attributes
  - Tree pruning
    - Identify and remove branches that reflect noise or outliers
- Use of decision tree: Classifying an unknown sample
  - Test the attribute values of the sample against the decision tree

10

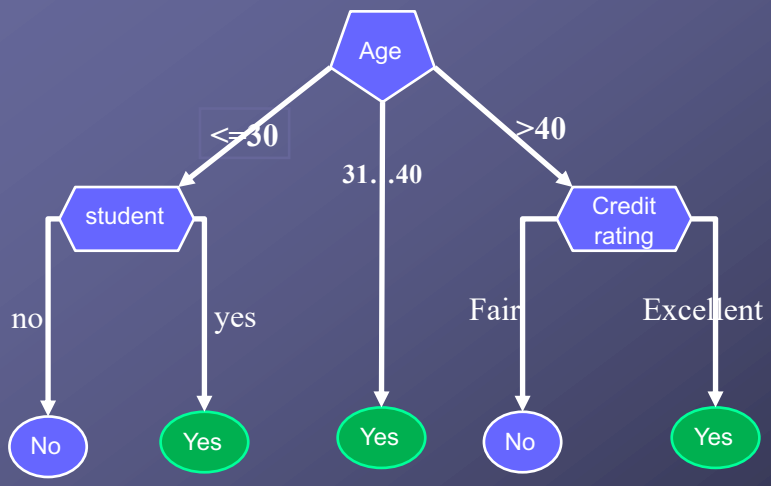
Training Dataset

Example : Who will buy a computer?

Age	Income	Student	Credit rating	Buy 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

11

Output: A Decision Tree for "*buy computer*"



12

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 discretised 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

13

Attribute Selection Measure

- Information gain
  - All attributes are assumed to be categorical
  - Can be modified for continuous-valued attributes
- Gini index
  - All attributes are assumed to be continuous-valued
  - Assume there exist several possible split values for each attribute
  - May need other tools, such as clustering, to get the possible split values
  - Can be modified for categorical attributes

14

Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Assume there are two classes, *P* and *N*
  - Let the set of examples *S* contain *p* elements of class *P* and *n* elements of class *N*
  - The amount of information, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

$$I(p,n) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

15

Information Gain in Decision Tree Induction

- Type equation here.Assume that using attribute A a set *S* will be partitioned into sets $\{S_1, S_2, \ldots, S_v\}$
  - If S_i contains p_i examples of *P* and n_i examples of *N*, the entropy, or the expected information needed to classify objects in all sub-trees S_i is

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I(p_i, n_i)$$

- The encoding information that would be gained by branching on *A* $Gain(A) = I(p,n) - E(A)$

16

Attribute Selection by Information Gain ComputationType equation here.

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for *age*:

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

$$E(age) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = \mathbf{0.69}$$

Hence

$$Gain(age) = I(9,5) - E(age) = \mathbf{0.250}$$

Similarly

$$Gain(student) = \mathbf{0.151}$$
$$Gain(credit\ rating) = \mathbf{0.048}$$
$$Gain(income) = \mathbf{0.029}$$

17

Gini Index

- Gini Index
  - If a data set *T* contains examples from *n* classes, gini index, *gini*(*T*) is
  - Measures the impurity of the data
  - where p_j is the relative frequency of class *j* in *T*

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

- Gini Split
  - If *T* is split into T_1 and T_2 of size N_1 and N_2 respectively, the *gini* index of the split data contains examples from *n* classes, and is
- Attribute split

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

  - The attribute that provides the smallest $gini_{split}(T)$ is chosen to split the node
  - (*need to enumerate all possible splitting points for each attribute*).

18

Extracting Classification Rules from Trees

- Represent the knowledge in the form of IF ()-THEN () rules
- One rule is created for each path from the root to a leaf
- Each attribute-value pair along a path forms a conjunction
- The leaf node holds the class prediction
- Rules are easier for humans to understand
- Example

IF *age* = "<=30" AND *student* = "*no*" THEN *buys_computer* = "*no*"
IF *age* = "<=30" AND *student* = "*yes*" THEN *buys_computer* = "*yes*"
IF *age* = "31…40" THEN *buys_computer* = "*yes*"
IF *age* = ">40" AND *credit_rating* = "*excellent*" THEN *buys_computer* = "*yes*"
IF *age* = ">40" AND *credit_rating* = "*fair*" THEN *buys_computer* = "*no*"

19

Avoid Overfitting in Classification

- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result in poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Pre-pruning: 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
  - Post-pruning: 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"

20

Determining the Final Tree Size

- Separate training (2/3) and testing (1/3) sets
- Use cross validation, e.g., 10-fold cross validation
- Use all the data for training
  - but apply a statistical test (e.g., χ^2) to estimate whether expanding or pruning a node may improve the entire distribution
- Use minimum description length (MDL) principle:
  - halting growth of the tree when the encoding is minimised

21

Enhancements to basic decision tree induction

- Allow 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

22

Classification in Large Datasets

- 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 decision tree induction in data mining?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods

23

Scalable DTI Methods in Data Mining

- SLIQ
  - builds an index for each attribute and only class list and the current attribute list reside in memory
- SPRINT
  - constructs an attribute list data structure
- PUBLIC
  - integrates tree splitting and tree pruning: stop growing the tree earlier
- RainForest
  - separates the scalability aspects from the criteria that determine the quality of the tree
  - builds an AVC-list (attribute, value, class label)

24

Bayesian Classification

- Probabilistic learning
  - Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems
- Incremental
  - Each training example can incrementally increase/decrease the probability that a hypothesis is correct
  - Prior knowledge can be combined with observed data
- Probabilistic prediction
  - Predict multiple hypotheses, weighted by their probabilities
- Standard
  - Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

25

Bayesian Theorem

- Posteriori Probability
  - Given training data *D, posteriori probability of a hypothesis h, P(h|D)* follows the Bayes theorem

$$\boldsymbol{P(h|D) = \frac{P(D|h)P(h)}{P(D)}}$$

- MAP (maximum posteriori) hypothesis

$$h_{MAP} \equiv \arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} P(D|h)P(h).$$

- Practical difficulty
  - require initial knowledge of many probabilities, significant computational cost

26

Bayes Theorem (2)

- Example
  - Consider a football game between two rival teams: T_0 and T_1. Suppose T_0 wins 65% of the time and T_1 wins the remaining matches. Among the games won by T_0, only 30% of them come from playing on T_1's field. On the other hand, 75% of the victories for T_1 are obtained while playing at home.
  - If T_1 is to host the next match against T_0, which team will most likely emerge as the winner?
- Use Bayes Theorem
  - X: random variable representing the team hosting the match
  - Y: random variable representing the winner of the match
  - The objective if to calculate P(Y = 1 | X = 1) or P(Y = 0 | X = 1)
- Answer
  - P(X,Y) = P(Y|X)*P(X) = P(X|Y)*P(Y)
  - Calculate P(Y = 0), P(Y = 1), P(X = 1|Y = 1), P(X = 1|Y = 0)
  - P(Y = 1|X = 1) = …

27

Naïve Bayes Classifier (I)

- Simplified assumption
  - Attributes are conditionally independent

$$P(C_j|V) \approx \prod_{i=1}^{N} P(v_i|C_j)$$

- The Cost
  - Greatly reduces the computation cost, only count the class distribution

28

Naive Bayesian Classifier (II)

- Given a training set, we can compute the probabilities

Outlook	Y	N
sunny	2/9	3/5
overcast	4/9	0
rain	3/9	2/5
Tempreature		
hot	2/9	2/5
mild	4/9	2/5
cool	3/9	1/5

Humidity	Y	N
high	3/9	4/5
normal	6/9	1/5
Windy		
true	3/9	3/5
false	6/9	2/5

29

Bayesian classification

- The classification problem may be formalised using a-posteriori probabilities
  - P(C|X) = prob. that the sample tuple $X = \langle x_1, \ldots, x_k \rangle$ is of class C
  - E.g. P(class = N | outlook = sunny, windy = true,…)
- Idea
  - assign to sample X the class label C such that P(C|X) is maximal

30

Estimating a-posteriori probabilities

- Bayes theorem
  - $P(C|X) = P(X|C) \cdot P(C) / P(X)$
  - $P(X)$ is constant for all classes
- $P(C)$ = relative frequency of class C samples
  - C such that $P(C|X)$ is maximum = C such that $P(X|C) \cdot P(C)$ is maximum
- Problem
  - Computing $P(X|C)$ is unfeasible!

31

Naïve Bayesian Classification

- Naïve assumption (attribute independence)
  - $P(x_1,\ldots,x_k|C) = P(x_1|C) \cdot \ldots \cdot P(x_k|C)$
- If i-th attribute is categorical then
  - $P(x_i|C)$ is estimated as the relative frequency of samples having value x_i as i-th attribute in class C
- If i-th attribute is continuous then
  - $P(x_i|C)$ is estimated through a Gaussian density function
- Computationally easy in both cases

32

Play-tennis example: estimating $P(x_i|C)$

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	Y
rain	mild	high	false	Y
rain	cool	normal	false	Y
rain	cool	normal	true	N
overcast	cool	normal	true	Y
sunny	mild	high	false	N
sunny	cool	normal	false	Y
rain	mild	normal	false	Y
sunny	mild	normal	true	Y
overcast	mild	high	true	Y
overcast	hot	normal	false	Y
rain	mild	high	true	N

P(y) = 9/14

P(n) = 5/14

outlook	
P(sunny|y) = 2/9	P(sunny|n) = 3/5
P(overcast|y) = 4/9	P(overcast|n) = 0
P(rain|y) = 3/9	P(rain|n) = 2/5
temperature	
P(hot|y) = 2/9	P(hot|n) = 2/5
P(mild|y) = 4/9	P(mild|n) = 2/5
P(cool|y) = 3/9	P(cool|n) = 1/5
humidity	
P(high|y) = 3/9	P(high|n) = 4/5
P(normal|y) = 6/9	P(normal|n) = 2/5
windy	
P(true|y) = 3/9	P(true|n) = 3/5
P(false|y) = 6/9	P(false|n) = 2/5

33

Play-Tennis example: classifying X

- An unseen sample X = < rain, hot, high, false >
  - $P(X|y)\cdot P(y) = P(rain|y)\cdot P(hot|y)\cdot P(high|y)\cdot P(false|y)\cdot P(y)$
  - $= 3/9\cdot 2/9\cdot 3/9\cdot 6/9\cdot 9/14 = 0.010582$
  - $P(X|n)\cdot P(n) = P(rain|n)\cdot P(hot|n)\cdot P(high|n)\cdot P(false|n)\cdot P(n) =$
  - $2/5\cdot 2/5\cdot 4/5\cdot 2/5\cdot 5/14 = 0.018286$
- Sample X is classified in class n (do not play)

34