DWDM : Unit V

Classification and Prediction

UNIT – V: Classification and Prediction: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Classification by Back propagation, Classification Based on Concepts from Association Rule Mining, k-nearest neighbor classifier, Prediction, Classifier Accuracy.

Classification and Prediction

1. What is classification? What is prediction?
2. Issues regarding classification and prediction
3. Classification by decision tree induction
4. Bayesian Classification
5. Classification by Neural Networks
6. Classification based on concepts from association rule mining
7. Other Classification Methods
8. Prediction
9. Classification accuracy
10. Summary

Classification vs. 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.

- Prediction:
  - models continuous-valued functions, i.e., predicts unknown or missing values

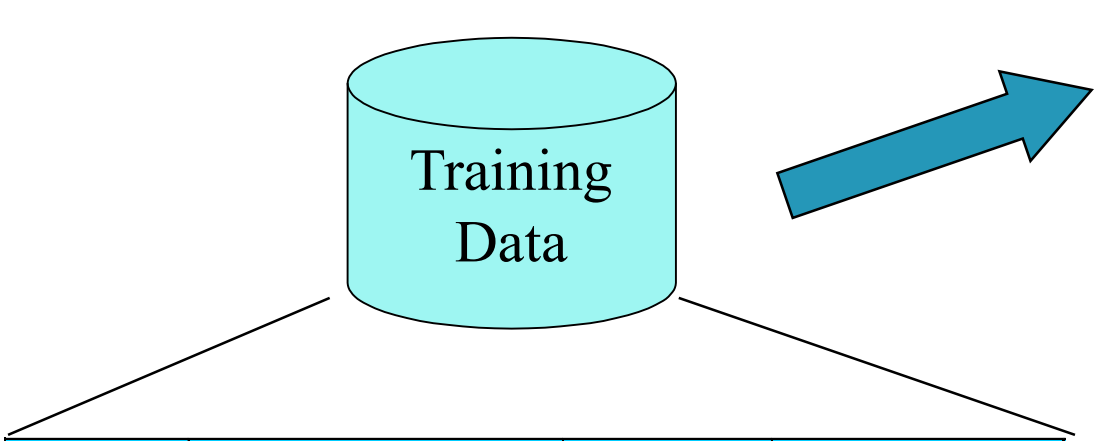
- Typical Applications
  - credit approval
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis

Classification—A Two-Step Process

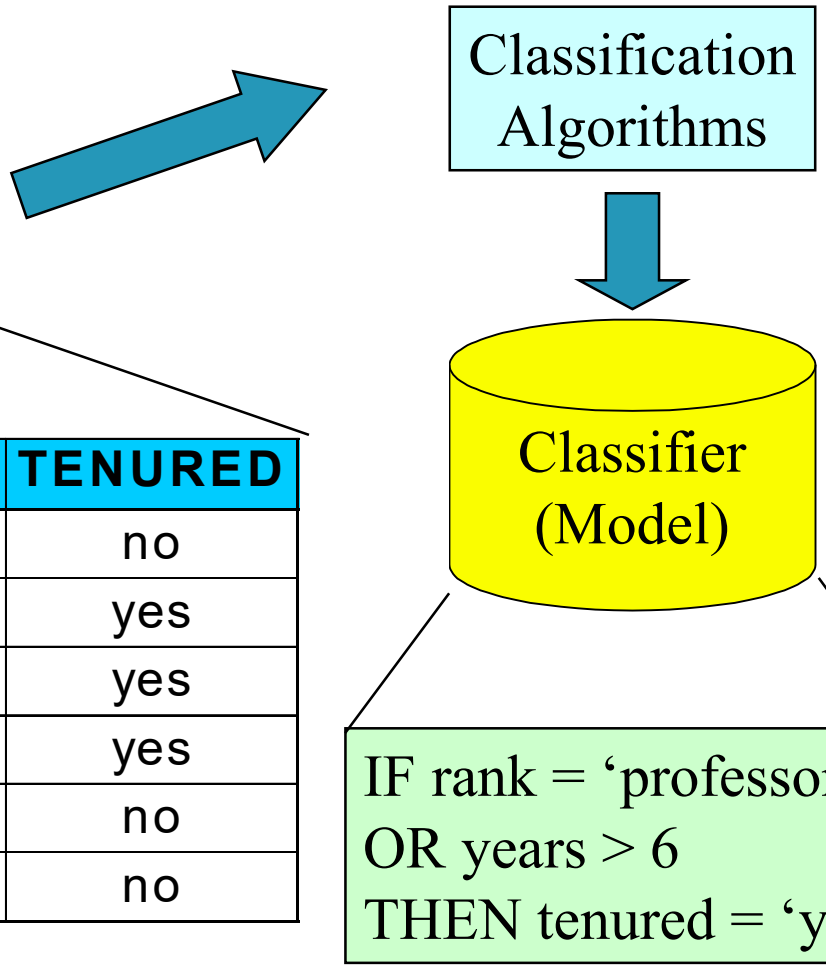
1. Model construction: describing a set of predetermined classes
   - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
   - The set of tuples used for model construction is training set
   - The model is represented as classification rules, decision trees, or mathematical formulae

2. 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 = % of test set samples that are correctly classified by the model
     - Test set is independent of training set, otherwise over-fitting will occur
   - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

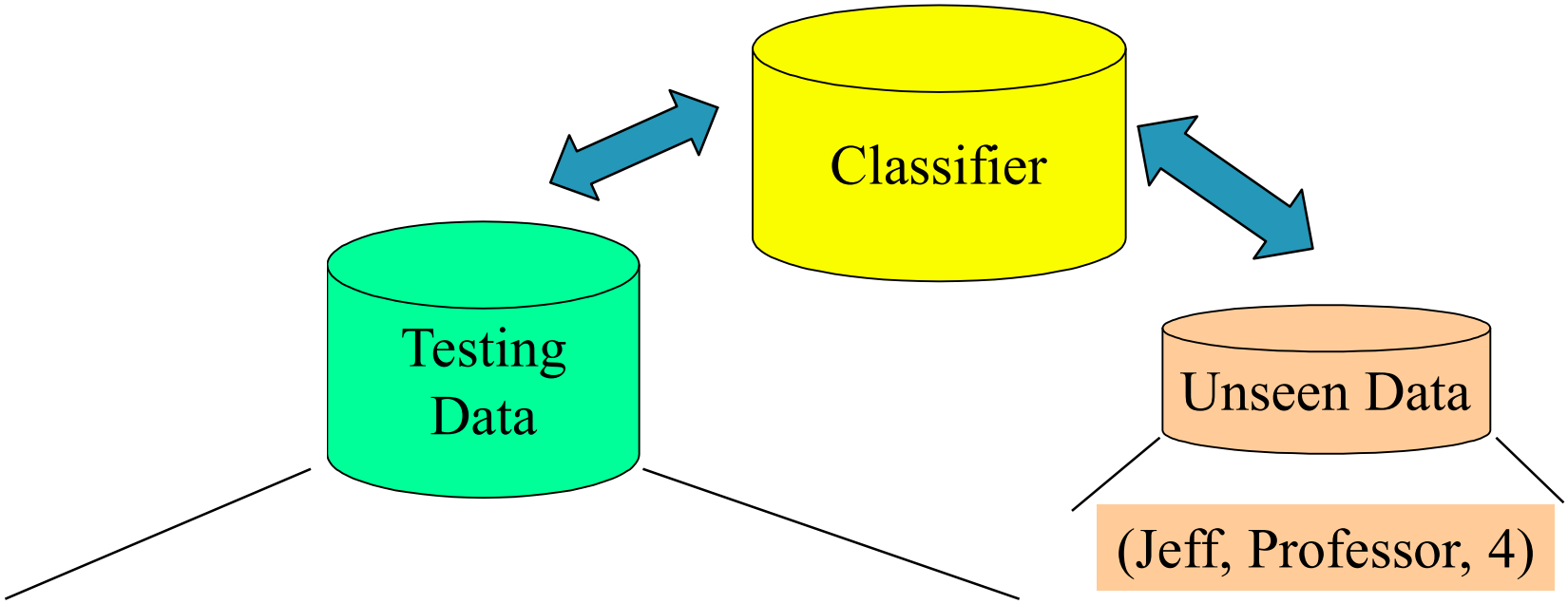
Classification Process (1): Model Construction



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 (2)
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)	Unsupervised learning (clustering)
Supervision: The training data are accompanied by labels indicating the class of the observations	The class labels of training data is unknown
New data is classified based on the training set	Given: a set of measurements, observations, etc. Aim: Establish the existence of classes or clusters in the data

Issues Regarding Classification and Prediction: Data Preparation and Comparing Classification Methods

A : Data Preparation	B : Comparing Classification Methods
1. Data Cleaning : Preprocess data in order to reduce noise and handle missing values	1. Predictive accuracy : ability of the model to correctly predict the class label of new/unseen data.
	2. Speed time to construct the model time to use the model
2. Relevance Analysis (Feature Selection) : Remove the irrelevant or redundant attributes. This avoids slow-down, misleading of the learning step.	3. Robustness : handling noise and missing values
	4. Scalability : efficiency of the model for large databases
3. Data Transformation : Generalize and/or normalize data. Concept hierarchies can be used.	5. Goodness of rules: decision tree size compactness of classification rules
	6. Interpretability: understanding the insight provided by the model

Data Preparation helps improve efficiency, accuracy and scalability of classification and prediction.

Classification by Decision Tree Induction

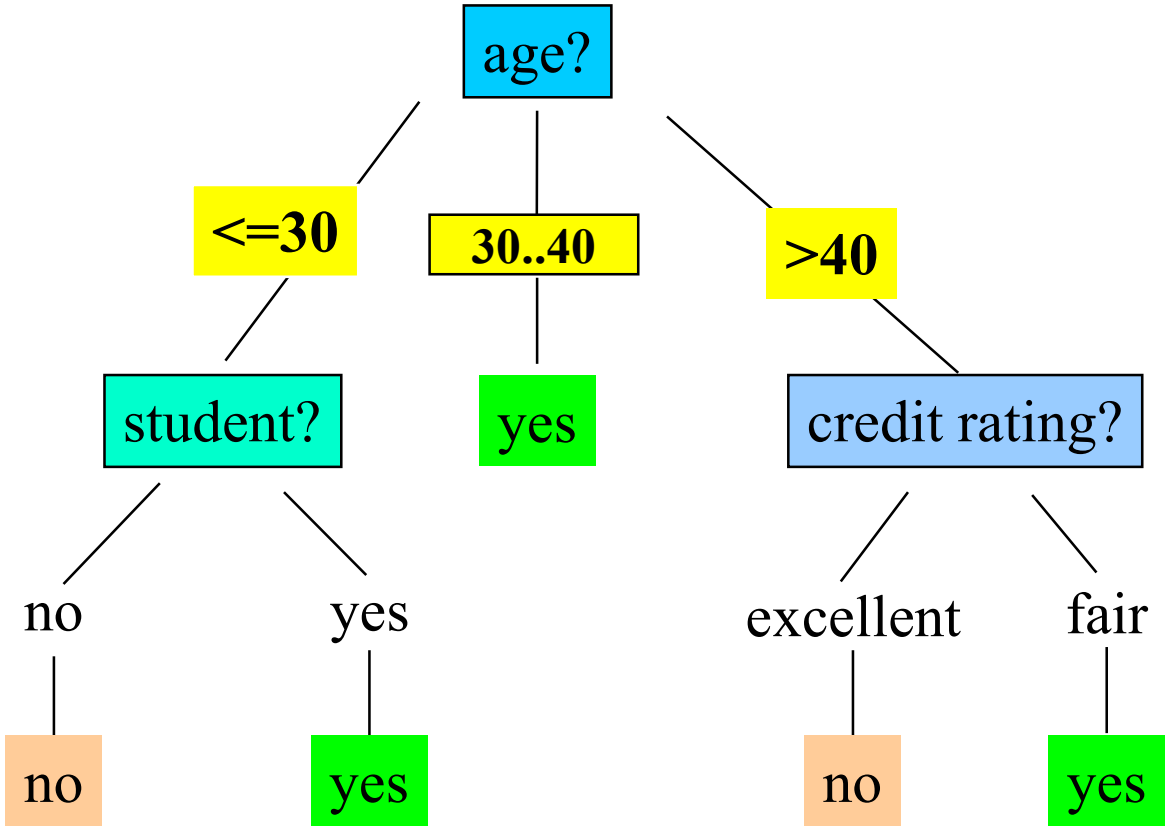
Training Dataset

This follows an example from Quinlan's ID3

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

Classification by Decision Tree Induction

Output: A Decision Tree for "*buys_computer*"



Basic algorithm for inducing a decision tree from training tuples

Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition, *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split-point* or *splitting subset*.

Output: A decision tree.

Basic algorithm for inducing a decision tree from training tuples (contd..)

Method:

```
(1)  create a node N;
(2)  if tuples in D are all of the same class, C, then
(3)      return N as a leaf node labeled with the class C;
(4)  if attribute_list is empty then
(5)      return N as a leaf node labeled with the majority class in D; // majority voting
(6)  apply Attribute_selection_method(D, attribute_list) to find the "best" splitting_criterion;
(7)  label node N with splitting_criterion;
(8)  if splitting_attribute is discrete-valued and
         multiway splits allowed then // not restricted to binary trees
(9)      attribute_list ← attribute_list − splitting_attribute; // remove splitting_attribute
(10) for each outcome j of splitting_criterion
     // partition the tuples and grow subtrees for each partition
(11)     let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)     if Dj is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate_decision_tree(Dj, attribute_list) to node N;
     endfor
(15) return N;
```

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

- Select the attribute with the highest information gain
- S contains s_i tuples of class C_i for i = {1, …, m}
- information measures: info required to classify any arbitrary tuple

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

- entropy of attribute A with values $\{a_1, a_2, \ldots, a_v\}$

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj})$$

- information gained by branching on attribute A

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A)$$

Attribute Selection by Information Gain Computation

- 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
30…40	4	0	0
>40	3	2	0.971

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

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes's and 3 no's. Hence

$$Gain(age) = I(p,n) - E(age) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

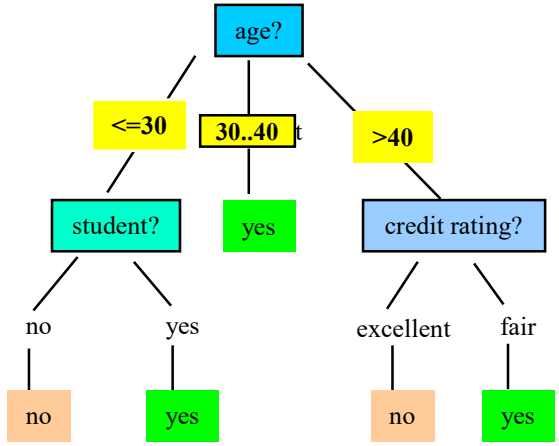
Tree Pruning: Avoid Overfitting in Classification

- 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"

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* = "<=30" AND *credit_rating* = "*fair*" THEN *buys_computer* = "*no*"

Approaches to Determine the Final Tree Size

- Separate training (2/3) and testing (1/3) sets
- Use minimum description length (MDL) principle
  - halting growth of the tree when the encoding is minimized

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 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

Scalable Decision Tree Induction Methods in Data Mining Studies

- 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)

Bayesian Classification

- Bayesian classifiers are statistical classifiers.
- They can predict class membership probabilities:- probability that a given sample belongs to a particular class.
- Have high accuracy and speed for large databases.
- class conditional independence : Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes.
- It is made to simplify the computations involved, and in this sense, is considered “naive".

Bayesian Theorem Basics

- Given training data *X, posteriori probability of a hypothesis H, P(H|X)* follows the Bayes theorem

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

- Informally, this can be written as
  posterior =likelihood x prior / evidence

- Let
  - X- a data sample whose class label is unknown
  - H- hypothesis that X belongs to class C

- For classification problems, determine P(H/X): the probability that the hypothesis holds given the observed data sample X

- P(H): prior probability of hypothesis H (i.e. the initial probability before we observe any data, reflects the background knowledge)

- P(X): probability that sample data is observed

- P(X|H) : probability of observing the sample X, given that the hypothesis holds

Bayesian Classification: Practical difficulty

- require initial knowledge of many probabilities, significant computational cost

Naive Bayesian classification

- Suppose
  - 1. Each data sample, X = (x1,x2,...,xn), depicting n measurements for n attributes, respectively A1,A2,...,An
  - 2. There are m classes, C1,C2,...,Cm.

- Given : unknown data sample, X
- the naive Bayesian classier assigns an unknown sample X to the class Ci iff

$$P(C_i|X) > P(C_j|X) \text{ for } 1 \leq j \leq m, j \neq i.$$

By Bayes theorem $P(Ci/X) = \frac{P(X/Ci) * P(Ci)}{P(X)}$

Probability of X conditioned on Ci is given by

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i).$$

i.e., P(X/Ci) = P(x1/Ci) * P(x2/Ci) *…… *P(xn/Ci)

where unknown data sample, X=(x1,x2,…..xn)

Training dataset

Class:
C1:buys_computer= 'yes'
C2:buys_computer= 'no'

Data sample(unknown)
X =(age<=30,
Income=medium,
Student=yes
Credit_rating=
Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…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

Naïve Bayesian Classifier: Example
Compute P(X/Ci) for each class

- Data sample X =(age<=30, Income=medium, Student=yes Credit_rating= Fair)

- P(age="<30" | buys_computer="yes") = 2/9=0.222
  P(age="<30" | buys_computer="no") = 3/5 =0.6
  P(income="medium" | buys_computer="yes")= 4/9 =0.444
  P(income="medium" | buys_computer="no") = 2/5 = 0.4
  P(student="yes" | buys_computer="yes)= 6/9 =0.667
  P(student="yes" | buys_computer="no")= 1/5=0.2
  P(credit_rating="fair" | buys_computer="yes")=6/9=0.667
  P(credit_rating="fair" | buys_computer="no")=2/5=0.4

P(buys_computer="yes")=9/14=0.643
P(buys_computer="no")=5/14=0.357

P(X|Ci) : P(X|buys_computer="yes")= 0.222 x 0.444 x 0.667 x 0.667 =0.044
P(X|buys_computer="no")= 0.6 x 0.4 x 0.2 x 0.4 =0.019

P(X|Ci)*P(Ci) : P(X|buys_computer="yes") * P(buys_computer="yes")=0.028
P(X|buys_computer="no") * P(buys_computer="no")=0.007

X belongs to class "buys_computer=yes"

Bayesian Classification: Why?

- 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

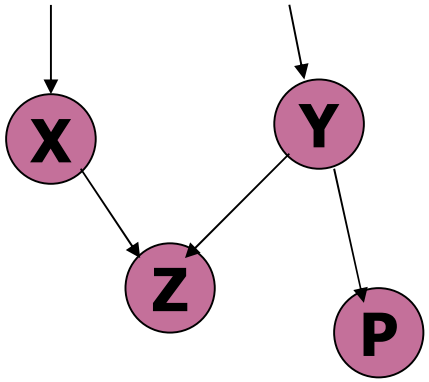
Naïve Bayesian Classifier: Comments

- Advantages :
  1. Easy to implement
  2. Good results obtained in most of the cases
- Disadvantages
  1. Assumption: class conditional independence , therefore loss of accuracy
  2. Practically, dependencies exist among variables
     1. E.g., hospitals: patients: Profile: age, family history etc
     2. Symptoms: fever, cough etc., Disease: lung cancer, diabetes etc
  3. Dependencies among these cannot be modeled by Naïve Bayesian Classifier

- How to deal with these dependencies?
  - Bayesian Belief Networks

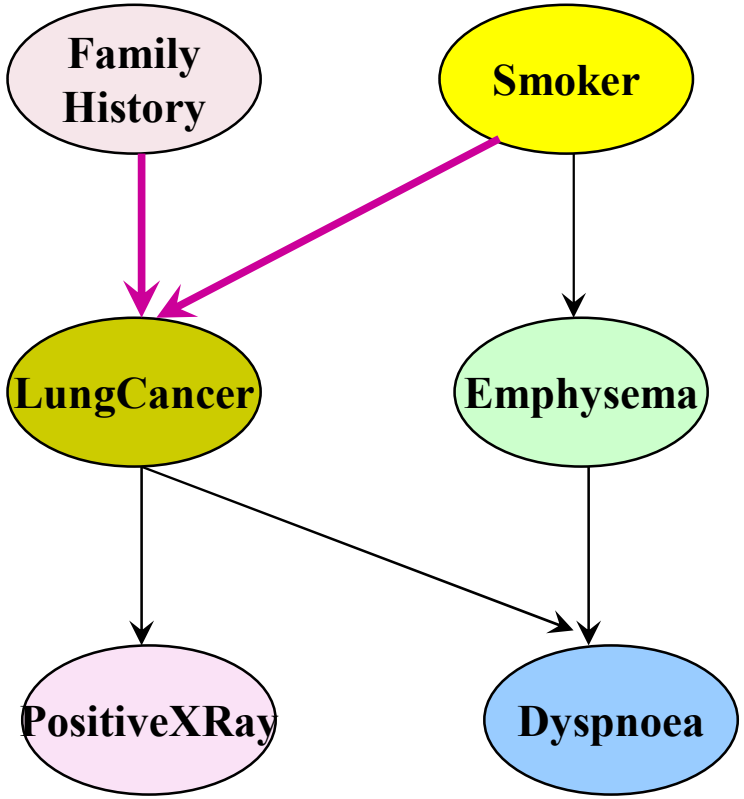
Bayesian Networks

- Bayesian belief network allows a *subset* of the variables conditionally independent
- A graphical model of causal relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution



- ❑Nodes: random variables
- ❑Links: dependency
- ❑X,Y are the parents of Z, and Y is the parent of P
- ❑No dependency between Z and P
- ❑Has no loops or cycles

Bayesian Belief Network: An Example



	(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

The **Conditional Probability Table (CPT)** for the variable LungCancer: Shows the conditional probability for each possible combination of its parents

$$P(z1,\ldots, zn) = \prod_{i=1}^{n} P(z_i \mid Parents(Z_i))$$

Bayesian Belief Networks

Where tuple(z1,…,zn) correspond to attributes Z1,Z2,…,Zn)

Learning Bayesian Networks

- Several cases
  1. Given both the network structure and all variables observable: learn only the CPTs
  2. Network structure known, some hidden variables: method of gradient descent, analogous to neural network learning
  3. Network structure unknown, all variables observable: search through the model space to reconstruct graph topology
  4. Unknown structure, all hidden variables: no good algorithms known for this purpose

Classification by Neural Networks (Backpropagation)

“What is backpropagation?"

- Its a neural network learning algorithm.

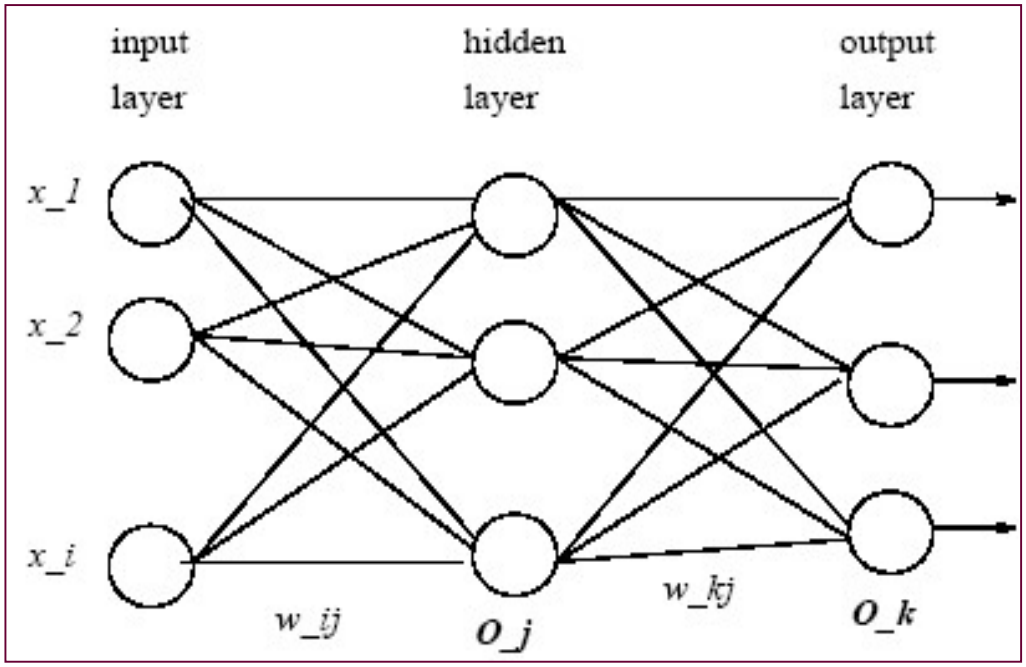
Neural Network

- Its a set of connected input/output units where each connection has a weight associated with it.
- During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input samples.

Multilayer Feed-forward Neural Network

- A training sample, X = (x1; x2; ...; xi), is fed to the input layer.
- The weighted outputs of these units are, in turn, fed simultaneously to a second layer, known as a hidden layer.
- The hidden layer's weighted outputs can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used.
- The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given samples.

Weighted connections exist between each layer, where wij denotes the weight from a unit j in one layer to a unit i in the previous layer.



Benefits and drawbacks of Neural Networks

- The network is feed-forward in that none of the weights cycle back to an input unit or to an output unit of a previous layer.

Benefits:

- include their high tolerance to noisy data
- able to classify patterns on which they have not been trained.

Disadvantages:

- involve long training times, so more suitable for applications where this is feasible.
- Need to know network topology or “structure".
- poor interpretability, as its difficult for humans to interpret the symbolic meaning behind the learned weights.

Defining a network topology

- Before training can begin, the user must decide on the network topology by specifying
  1. the number of units in the input layer,
  2. the number of hidden layers (if more than one),
  3. the number of units in each hidden layer, and
  4. the number of units in the output layer.

- Normalizing the input values for each attribute in the training samples to fall between 0 and 1.0.

Backpropagation

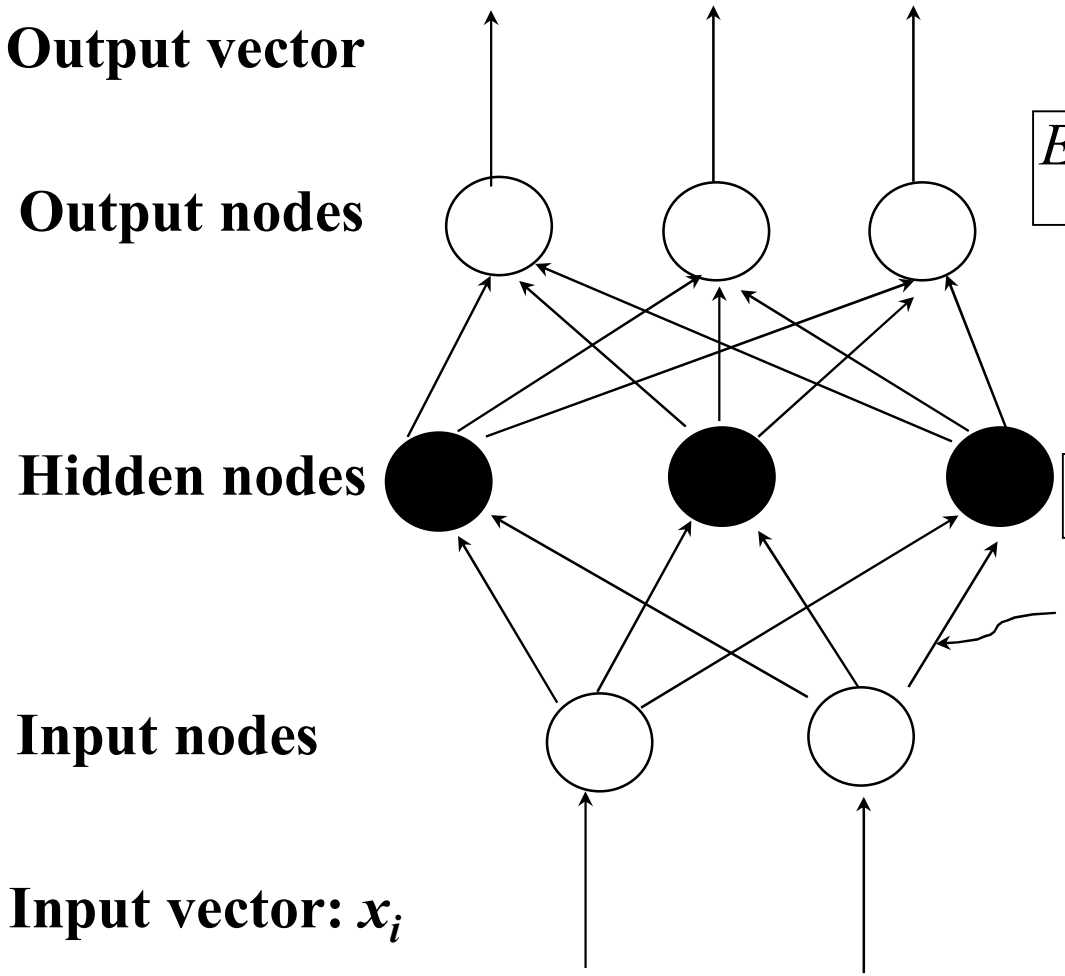
“What is backpropagation?"

- Its a neural network learning algorithm.

How does it work?

- Backpropagation learns by iteratively processing a set of training samples, comparing the network's prediction for each sample with the actual known class label.
- For each training sample, the weights are modified so as to minimize the error between the network's prediction and the actual class.
- These modifications are made in the “backwards" direction, i.e., from the output layer, through each hidden layer down to the first hidden layer (hence the name backpropagation).

Multi-Layer Perceptron



$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

$$\theta_j = \theta_j + (l)Err_j$$

$$w_{ij} = w_{ij} + (l)Err_j O_i$$

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

$$O_j = \frac{1}{1+e^{-I_j}}$$

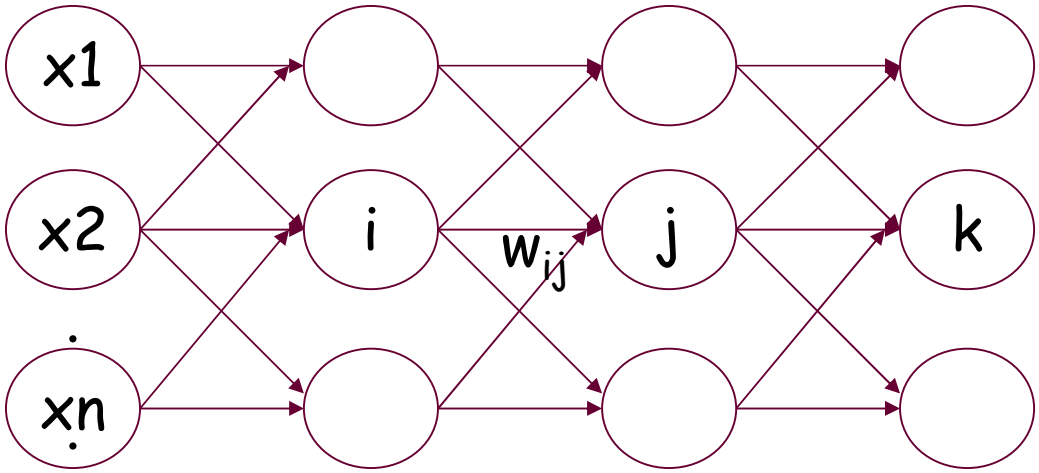
$$I_j = \sum_i w_{ij} O_i + \theta_j$$

- Given a unit j in a hidden or output layer, the net input,

  Ij, to unit j is:

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

θ_j =Bias(threshold)



Given the net input Ij to unit j, then Oj, the output of unit j, is computed as:

$$O_j = \frac{1}{1 + e^{-I_j}}$$

Backpropagate the error. The error is propagated backwards by updating the weights and biases to reflect the error of the network's prediction. Tj=true o/p

$$Err_j = O_j(1 - O_j)(T_j - O_j) \Rightarrow Err_j = O_j(1 - O_j)\sum_k Err_k w_{jk}$$

Network Training

- The ultimate objective of training
  - obtain a set of weights that makes almost all the tuples in the training data classified correctly

- Steps
  - Initialize weights with random values
  - Feed the input tuples into the network one by one
  - For each unit
    - Compute the net input to the unit as a linear combination of all the inputs to the unit
    - Compute the output value using the activation function
    - Compute the error
    - Update the weights and the bias

Association-Based Classification

- Several methods for association-based classification
  - ARCS: Quantitative association mining and clustering of association rules
    - It beats C4.5 in (mainly) scalability and also accuracy
  - Associative classification:
    - It mines high support and high confidence rules in the form of "cond_set => y", where y is a class label
  - CAEP (Classification by Aggregating Emerging Patterns)
    - Emerging patterns (EPs): the itemsets whose support increases significantly from one class to another
    - Mine Eps based on minimum support and growth rate

Other Classification Methods

- k-nearest neighbor classifier
- case-based reasoning
- Genetic algorithm
- Rough set approach
- Fuzzy set approaches

Instance-Based (Lazy Learners) Methods

- Instance-based learning:
  - Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified

- Typical approaches
  - *k*-nearest neighbor approach
    - Instances represented as points in a Euclidean space.

  - Case-based reasoning
    - Uses symbolic representations and knowledge-based inference

  - Locally weighted regression
    - Constructs local approximation

Remarks on Lazy vs. Eager Learning

- K Nearest Neighbor & Case based reasoning : lazy evaluation
- Decision-tree and Bayesian classification: eager evaluation

Key differences

Lazy Learning	Eager Learning
Store training examples and delay the processing until a new instance must be classified	Constructs generalization model before receiving new samples to classify.
Faster at training	Slower at training
Expensive computational costs when neighbors are more.	Less expensive. Assigns different weights to attributes.
Take more time for predicting Accuracy	Faster at classification as it builds model early.

The *k*-Nearest Neighbor Algorithm

- All instances correspond to points in the n-D space.
- The nearest neighbor are defined in terms of Euclidean distance.
- The target function could be discrete- or real- valued.
- For discrete-valued, the *k*-NN returns the most common value among the k training examples nearest to *X*.
- Euclidean distance between 2 points: X=(x1,x2,...,xn) Y=(y1,y2,...,yn)

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Case-Based Reasoning

- Uses: lazy evaluation + analyze similar instances
- Instances are not "points in a Euclidean space"

- Methodology
  - Instances represented by rich symbolic descriptions, or as cases.
  - Earlier similar cases are seen. Multiple retrieved cases may be combined
  - Tight coupling between case retrieval, knowledge-based reasoning, and problem solving

- Research issues
  - Indexing based on syntactic similarity measure, and when failure, backtracking, and adapting to additional cases

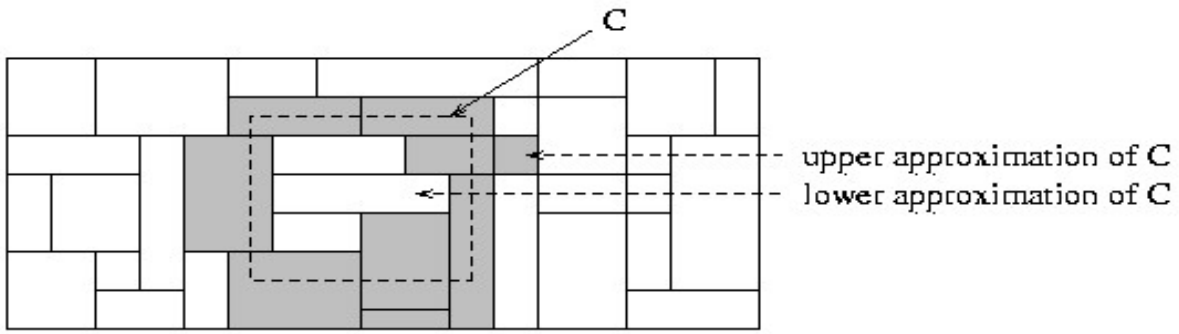
Genetic Algorithms

based on an analogy to biological evolution

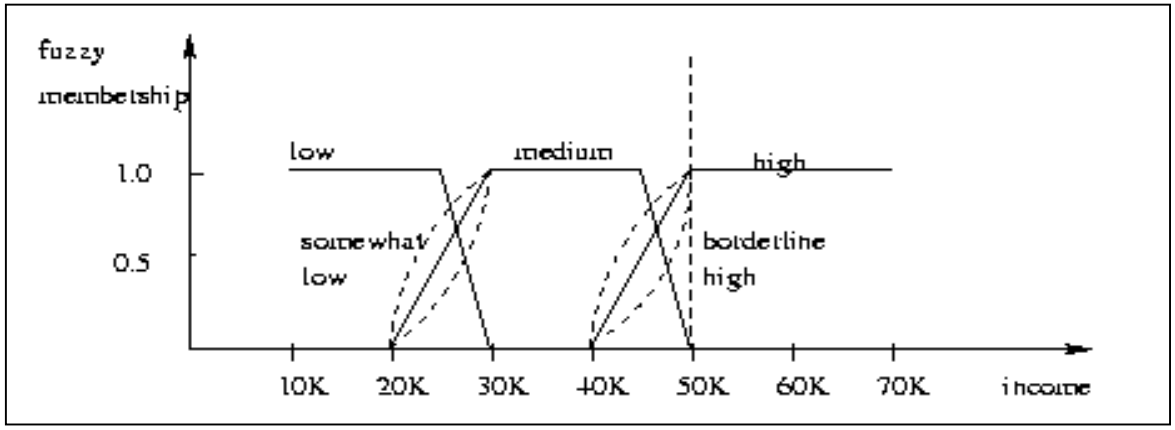
- Each rule is represented by a string of bits
- An initial population is created consisting of randomly generated rules
  - e.g., IF A_1 and Not A_2 then C_2 can be encoded as 101
- Based on the notion of survival of the fittest, a new population is formed to consists of the fittest rules and their offsprings
- The fitness of a rule is represented by its classification accuracy on a set of training examples
- Offsprings are generated by crossover and mutation

Rough Set Approach

- Rough sets are used to approximately or "roughly" define equivalent classes
- A rough set for a given class C is approximated by two sets: a lower approximation (certain to be in C) and an upper approximation (cannot be described as not belonging to C)
- Finding the minimal subsets (reducts) of attributes (for feature reduction) is NP-hard but a discernibility matrix is used to reduce the computation intensity



Fuzzy Set Approaches



- Fuzzy logic uses truth values between 0.0 and 1.0 to represent the degree of membership (such as using fuzzy membership graph)
- Attribute values are converted to fuzzy values
  - e.g., income is mapped into the discrete categories {low, medium, high} with fuzzy values calculated
- For a given new sample, more than one fuzzy value may apply
- Each applicable rule contributes a vote for membership in the categories
- Typically, the truth values for each predicted category are summed

What Is Prediction?

- Prediction is similar to classification
  - First, construct a model
  - Second, use model to predict unknown value
    - Major method for prediction is regression
      - Linear and multiple regression
      - Non-linear regression

- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions

Regress Analysis and Log-Linear Models in Prediction

- Linear regression: $Y = \alpha + \beta X$
  - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of Y_1, Y_2, ..., X_1, X_2, ...

- Multiple regression: $Y = b0 + b1 X1 + b2 X2$.
  - Many nonlinear functions can be transformed into the above.

- Non-linear models:
  - Polynomial regression can be modeled by adding polynomial terms to the basic linear model.
  - By applying transformations to the variables, we can convert the nonlinear model into a linear one that can then be solved by the method of least squares.
  - X1=X, X2=X^2, X3=X^3

$$Y = \alpha + \beta_1 X + \beta_2 X^2 + \beta_3 X^3$$

  - The above equation can be converted as

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3,$$

Classification Accuracy: Estimating Classifier accuracy

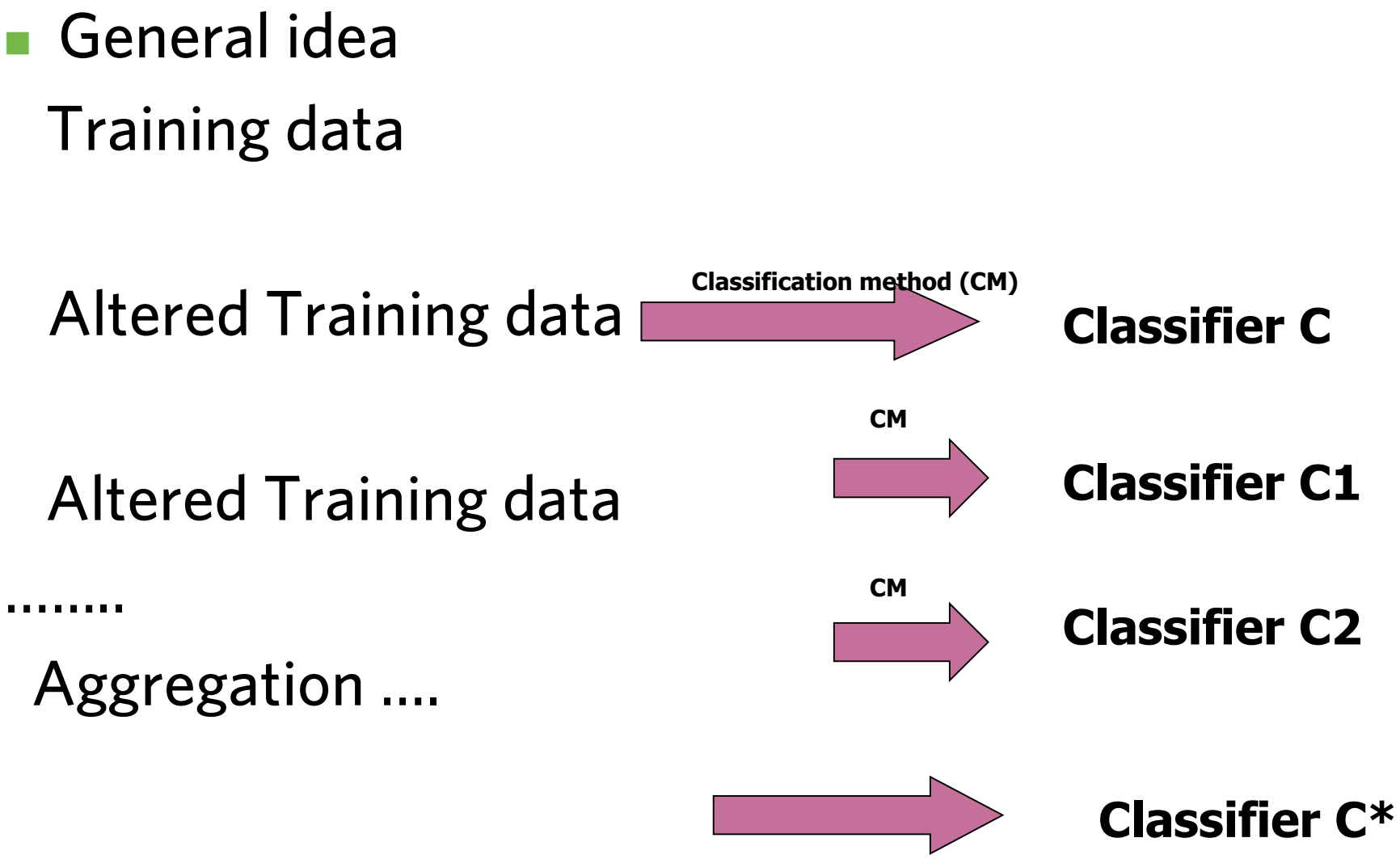
1. Holdout --Partition: Training-and-testing
   - use two independent data sets, e.g., training set (2/3), test set(1/3)
   - used for data set with large number of samples

2. K-fold Cross-validation
   - divide the data set into *k* subsamples
   - use ***k-1*** subsamples as training data and one sub-sample as test data
   - Useful for data set with moderate size

3. Bootstrapping (leave-one-out)
   - for small size data

Increasing Accuracy: Bagging and Boosting

- General idea



Bagging

- Given a set S of s samples
- Generate a bootstrap sample T from S. Cases in S may not appear in T or may appear more than once.
- Repeat this sampling procedure, getting a sequence of k independent training sets
- A corresponding sequence of classifiers C1,C2,…,Ck is constructed for each of these training sets, by using the same classification algorithm
- To classify an unknown sample X,let each classifier predict or vote
- The Bagged Classifier C* counts the votes and assigns X to the class with the “most” votes

Boosting Technique — Algorithm

- Assign every example an equal weight *1/N*
- *For t = 1, 2, …, T Do*
  - Obtain a hypothesis (classifier) $h^{(t)}$ under $w^{(t)}$
  - Calculate the error of *h(t)* and re-weight the examples based on the error . Each classifier is dependent on the previous ones. Samples that are incorrectly predicted are weighted more heavily
  - Normalize $w^{(t+1)}$ to sum to 1 (weights assigned to different classifiers sum to 1)
- Output a weighted sum of all the hypothesis, with each hypothesis weighted according to its accuracy on the training set

Summary

- Classification is an extensively studied problem (mainly in statistics, machine learning & neural networks)
- Classification is probably one of the most widely used data mining techniques with a lot of extensions
- Scalability is still an important issue for database applications: thus combining classification with database techniques should be a promising topic
- Research directions: classification of non-relational data, e.g., text, spatial, multimedia, etc..