Principles of Knowledge Discovery in Databases

Fall 1999

Chapter 7: Data Classification

Dr. Osmar R. Zaïane

University of Alberta

Source:
Dr. Jiawei Han

Summary of Last Chapter

- What is association rule mining?
- How do we mine single-dimensional boolean associations?
- How do we mine multilevel associations?
- How do we mine multidimensional associations?
- Can we constrain the association mining?

Course Content

- Introduction to Data Mining
- Data warehousing and OLAP
- Data cleaning
- Data mining operations
- Data summarization
- Association analysis
- **Classification and prediction**
- Clustering
- Web Mining
- Similarity Search
- *Other topics if time permits*

Chapter 7 Objectives

Learn basic techniques for data classification and prediction.

Realize the difference between supervised classification, prediction and unsupervised classification of data.

Data Classification Outline

- What is classification of data and prediction?
- How do we classify data by decision tree induction?
- What are neural networks and how can they classify?
- What is Bayesian classification?
- Are there other classification techniques?
- How do we predict continuous values?

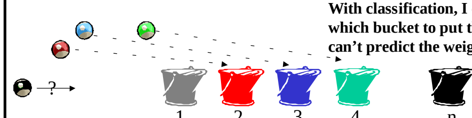
What is Classification?

The goal of data classification is to organize and categorize data in distinct classes.

- A model is first created based on the data distribution.
- The model is then used to classify new data.
- Given the model, a class can be predicted for new data.

Classification = prediction for discrete and nominal values

With classification, I can predict in which bucket to put the ball, but I can't predict the weight of the ball.



What is Prediction?

The goal of prediction is to forecast or deduce the value of an attribute based on values of other attributes.

- A model is first created based on the data distribution.
- The model is then used to predict future or unknown values.

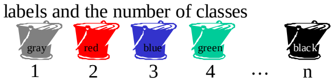
In Data Mining

If forecasting discrete value ➔ **Classification**

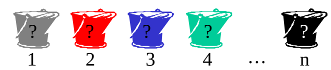
If forecasting continuous value ➔ **Prediction**

Supervised and Unsupervised

Supervised Classification = Classification
➔ We know the class labels and the number of classes



Unsupervised Classification = Clustering
➔ We do not know the class labels and may not know the number of classes



Preparing Data Before Classification

Data transformation:
•Discretization of continuous data
•Normalization to [-1..1] or [0..1]

Data Cleaning:
•Smoothing to reduce noise

Relevance Analysis:
•Feature selection to eliminate irrelevant attributes

Application

- **Credit approval**
- **Target marketing**
- **Medical diagnosis**
- **Defective parts identification in manufacturing**
- **Crime zoning**
- **Treatment effectiveness analysis**
- **Etc.**

Classification is a three-step process

1. Model construction (**Learning**):

- Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the **class label**.
- The set of all tuples used for construction of the model is called **training set**.
- The model is represented in the following forms:
  - **Classification rules, (IF-THEN statements),**
  - **Decision tree**
  - **Mathematical formulae**

Classification is a three-step process

2. Model Evaluation (**Accuracy**):

Estimate accuracy rate of the model based on a **test set**.

- 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.

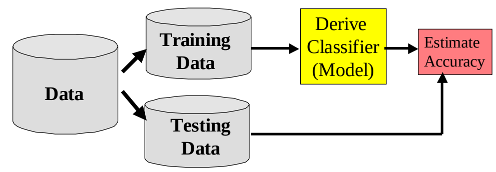
Classification is a three-step process

3. Model Use (**Classification**):

The model is used to classify unseen objects.

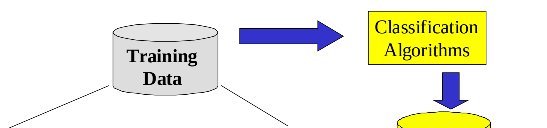
- Give a class label to a new tuple
- Predict the value of an actual attribute

Classification with Holdout



- •Holdout
- •Random sub-sampling
- •K-fold cross validation
- •Bootstrapping
- • …

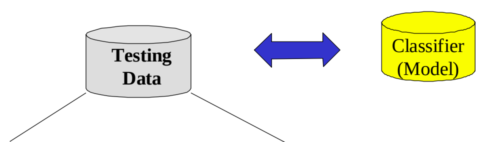
1. Classification Process (Learning)



Name	Income	Age	Credit rating
Bruce	Low	<30	bad
Dave	Medium	[30..40]	good
William	High	<30	good
Marie	Medium	>40	good
Anne	Low	[30..40]	good
Chris	Medium	<30	bad

IF Income = 'High'
OR Age > 30
THEN CreditRating = 'Good'

2. Classification Process (Accuracy Evaluation)

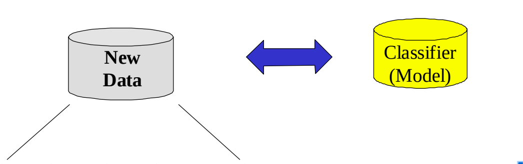


Name	Income	Age	Credit rating
Tom	Medium	<30	bad
Jane	High	<30	bad
Wei	High	>40	good
Hua	Medium	[30..40]	good

How accurate is the model?

IF Income = 'High'
OR Age > 30
THEN CreditRating = 'Good'

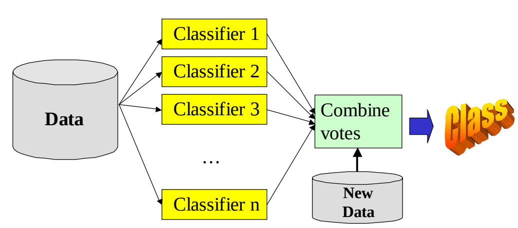
3. Classification Process (Classification)



Name	Income	Age	Credit rating
Paul	High	[30..40]	?

Credit Rating?

Improving Accuracy



Composite classifier

Classification Methods

- ❖ Decision Tree Induction
- ❖ Neural Networks
- ❖ Bayesian Classification
- ❖ Association-Based Classification
- ❖ K-Nearest Neighbour
- ❖ Case-Based Reasoning
- ❖ Genetic Algorithms
- ❖ Rough Set Theory
- ❖ Fuzzy Sets
- ❖ Etc.

Evaluating Classification Methods

- Predictive accuracy
  - Ability of the model to correctly predict the class label
- Speed and scalability
  - Time to construct the model
  - Time to use the model
- Robustness
  - Handling noise and missing values
- Scalability
  - Efficiency in large databases (not memory resident data)
- Interpretability:
  - The level of understanding and insight provided by the model
- Form of rules
  - Decision tree size
  - The compactness of classification rules

Data Classification Outline

- What is classification of data and prediction?
- How do we classify data by decision tree induction?
- What are neural networks and how can they classify?
- What is Bayesian classification?
- Are there other classification techniques?
- How do we predict continuous values?

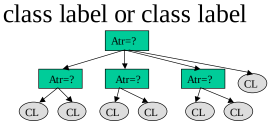
What is a Decision Tree?

A decision tree is a flow-chart-like tree structure.

Atr=? • Internal node denotes a test on an attribute

→ • Branch represents an outcome of the test
  - All tuples in branch have the same value for the tested attribute.

CL • Leaf node represents class label or class label distribution.

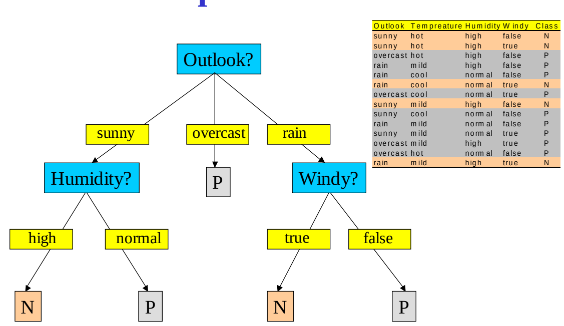


Training Dataset

- An Example from Quinlan's ID3

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

A Sample Decision Tree



Decision-Tree Classification Methods

- The basic top-down decision tree generation approach usually consists of two phases:
  1. **Tree construction**
     - At the start, all the training examples are at the root.
     - Partition examples are recursively based on selected attributes.
  2. **Tree pruning**
     - Aiming at removing tree branches that may reflect noise in the training data and lead to errors when classifying test data ➔ improve classification accuracy.

Decision Tree Construction

Recursive process:

- Tree starts a single node representing all data.
- If sample are all same class then node becomes a leaf labeled with class label.
- Otherwise, ***select attribute*** that best separates sample into individual classes. How?
- Recursion stops when:
  - Sample in node belong to the same class (majority);
  - There are no remaining attributes on which to split;
  - There are no samples with attribute value.

Choosing the Attribute to Split Data Set

- The measure is also called ***Goodness function***
- Different algorithms may use different goodness functions:
  - **information gain** (ID3/C4.5)
    - assume all attributes to be categorical.
    - can be modified for continuous-valued attributes.
  - **gini index**
    - assume all attributes are 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.

Information Gain (ID3/C4.5)

- Assume that there are two classes, *P* and *N*.
  - Let the set of examples *S* contain *x* elements of class *P* and *y* elements of class *N*.
  - The amount of information, needed to decide if an arbitrary example in *S* belong to *P* or *N* is defined as:

$$I(S_P,S_N) = -\frac{x}{x+y}\log_2\frac{x}{x+y} - \frac{y}{x+y}\log_2\frac{y}{x+y}$$

In general

$$I(s_1,s_2,\ldots,s_n) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

- Assume that using attribute A as the root in the tree will partition *S* in sets $\{S_1, S_2, \ldots, S_v\}$.
  - If S_i contains x_i examples of *P* and y_i examples of *N*, the information needed to classify objects in all subtrees S_i :

$$E(A) = \sum_{i=1}^{v} \frac{x_i + y_i}{x+y} I(S_{Pi}, S_{Ni})$$

In general

$$E(A) = \sum_{i=1}^{v} \frac{S_{1i} + S_{2i} + \ldots + S_{ni}}{s} I(s_{1i}, s_{2i}, \ldots, s_{ni})$$

Information Gain -- Example

- The attribute *A* is selected such that the ***information gain***

  $gain(A) = I(S_P,S_N) - E(A)$

  is maximal, that is, *E(A)* is minimal since $I(S_P,S_N)$ is the same to all attributes at a node.
- In the given sample data, attribute *outlook* is chosen to split at the root :

  gain(outlook) = 0.246

  gain(temperature) = 0.029

  gain(humidity) = 0.151

  gain(windy) = 0.048

> Information gain measure tends to favor attributes with many values. Other possibilities: Gini Index, χ^2, etc.

Gini Index

- If a data set *S* contains examples from *n* classes, gini index, *gini*(*S*) is defined as

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

where p_j is the relative frequency of class *j* in *S*.

- If a data set *S* is split into two subsets S_1 and S_2 with sizes N_1 and N_2 respectively, the *gini* index of the split data contains examples from *n* classes, the *gini* index *gini*(*S*) is defined as

$$gini_{split}(S) = \frac{N_1}{N} gini(S_1) + \frac{N_2}{N} gini(S_2)$$

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

Example for gini Index

- Suppose there two attributes: *age* and *income*, and the class label is buy and not buy.
- There are three possible split values for age: 30, 40, 50.
- There are two possible split values for income: 30K, 40K
- We need to calculate the following gini index
  - gini *age = 30* (S),
  - gini *age = 40* (S),
  - gini *age = 50* (S),
  - gini *income = 30k* (S),
  - gini *income = 40k* (S)
- Choose the minimal one as the split attribute

Primary Issues in Tree Construction

- **Split criterion**:
  - Used to select the attribute to be split at a tree node during the tree generation phase.
  - Different algorithms may use different goodness functions: information gain, gini index, etc.
- **Branching scheme**:
  - Determining the tree branch to which a sample belongs.
  - binary splitting (gini index) versus many splitting (information gain).
- **Stopping decision**: When to stop the further splitting of a node, e.g. impurity measure.
- **Labeling rule**: a node is labeled as the class to which most samples at the node belong.

How to construct a tree?

- Algorithm
  - greedy algorithm
    - make optimal choice at each step: select the best attribute for each tree node.
  - top-down recursive divide-and-conquer manner
    - from root to leaf
    - split node to several branches
    - for each branch, recursively run the algorithm

Example for Algorithm (ID3)

- All attributes are categorical
- Create a node N;
  - if samples are all of the same class C, then return N as a leaf node labeled with C.
  - if attribute-list is empty then return N as a left node labeled with the most common class.
- Select split-attribute with highest information gain
  - label N with the split-attribute
  - for each value A_i of split-attribute, grow a branch from Node N
  - let S_i be the branch in which all tuples have the value A_i for split- attribute
    - if S_i is empty then attach a leaf labeled with the most common class.
    - Else recursively run the algorithm at Node S_i
- Until all branches reach leaf nodes

How to use a tree?

- Directly
  - test the attribute value of unknown sample against the tree.
  - A path is traced from root to a leaf which holds the label.
- Indirectly
  - decision tree is converted to classification rules.
  - one rule is created for each path from the root to a leaf.
  - IF-THEN rules are easier for humans to understand.

Avoid Over-fitting in Classification

- A tree generated may over-fit the training examples due to noise or too small a set of training data.
- Two approaches to avoid over-fitting:
  - (Stop earlier): Stop growing the tree earlier.
  - (Post-prune): Allow over-fit and then post-prune the tree.
- Approaches to determine the correct final tree size:
  - Separate training and testing sets or use cross-validation.
  - Use all the data for training, but apply a statistical test (e.g., chi-square) to estimate whether expanding or pruning a node may improve over entire distribution.
  - Use Minimum Description Length (MDL) principle: halting growth of the tree when the encoding is minimized.
- Rule post-pruning (C4.5): converting to rules before pruning.

Continuous and Missing Values in Decision-Tree Induction

- Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals.

Temperature	40	48	60	72	80	90
play tennis	No	No	Yes	Yes	Yes	No

- Sort the examples according to the continuous attribute *A*, then identify adjacent examples that differ in their target classification, generate a set of candidate thresholds midway, and select the one with the maximum gain.
- Extensible to split continuous attributes into multiple intervals.
- Assign missing attribute values either
  - Assign the most common value of *A(x)*.
  - Assign probability to each of the possible values of *A*.

Alternative Measures for Selecting Attributes

- Info gain naturally favours attributes with many values.
- One alternative measure: gain ratio (Quinlan'86) which is to penalize attribute with many values.

$$SplitInfo(S,A) \equiv -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

$$GainRatio(S,A) = \frac{Gain(S,A)}{SplitInfo(S,A)}$$

  - Problem: denominator can be 0 or close which makes GainRatio very large.
- Distance-based measure (Lopez de Mantaras'91):
  - define a distance metric between partitions of the data.
  - choose the one closest to the perfect partition.
- There are many other measures. Mingers'91 provides an experimental analysis of effectiveness of several selection measures over a variety of problems.

Tree Pruning

- A decision tree constructed using the training data may have too many branches/leaf nodes.
  - Caused by noise, over-fitting.
  - May result poor accuracy for unseen samples.
- Prune the tree: merge a subtree into a leaf node.
  - Using a set of data different from the training data.
  - At a tree node, if the accuracy without splitting is higher than the accuracy with splitting, replace the subtree with a leaf node, label it using the majority class.
- Issues:
  - Obtaining the testing data.
  - Criteria other than accuracy (e.g. minimum description length).

Pruning Criterion

- Use a separate set of examples to evaluate the utility of post-pruning nodes from the tree.
  - CART uses cost-complexity pruning.
- Apply a statistical test to estimate whether expanding (or pruning) a particular node.
  - C4.5 uses pessimistic pruning.
- Minimum Description Length (no test sample needed).
  - SLIQ and SPRINT use MDL pruning.

Pruning Criterion --- MDL

- Best binary decision tree is the one that can be encoded with the fewest number of bits
  - Selecting a scheme to encode a tree
  - Comparing various subtrees using the cost of encoding
  - The best model minimizes the cost
- Encoding schema
  - One bit to specify whether a node is a leaf (0) or an internal node (1)
  - *loga* bits to specify the splitting attribute
  - Splitting the value for the attribute:
    - categorical --- log(*v*-1) bits
    - numerical --- log 2^{v-2}

PUBLIC: Integration of Two Phases

- Most decision tree classifiers have two phases:
  - Splitting
  - Pruning
- PUBLIC: Integration of two phases (Rastogi & Shim'98)
  - A large portions of the original tree are pruned during the pruning phase, why not use top-down methods to stop growing the tree earlier?
- Before expanding a node in building phase, a lower bound estimation on the minimum cost subtree rooted at the node is computed.
- If a node is certain to be pruned according to the estimation, return it as a leaf; otherwise, go on splitting it.

Classification and Databases

- Classification is a classical problem extensively studied by Statisticians and AI researchers, especially machine learning community.
- Database researchers re-examined the problem in the context of large databases.
  - most previous studies used small size data, and most algorithms are memory resident.
- Recent data mining research contributes to:
  - Scalability
  - Generalization-based classification
  - Parallel and distributed processing

Classifying Large Dataset

- Decision trees seem to be a good choice
  - relatively faster learning speed than other classification methods.
  - can be converted into simple and easy to understand classification rules.
  - can be used to generate SQL queries for accessing databases
  - has comparable classification accuracy with other methods
- Classifying data-sets with millions of examples and a few hundred even thousands attributes with reasonable speed.

Scalable Decision Tree Methods

- Most algorithms assume data can fit in memory.
- Data mining research contributes to the scalability issue, especially for decision trees.
- Successful examples
  - **SLIQ** (EDBT'96 -- Mehta et al.'96)
  - **SPRINT** (VLDB96 -- J. Shafer et al.'96)
  - **PUBLIC** (VLDB98 -- Rastogi & Shim'98)
  - **RainForest** (VLDB98 -- Gehrke, et al.'98)

Previous Efforts on Scalability

- Incremental tree construction (Quinlan'86)
  - using partial data to build a tree.
  - testing other examples and those misclassified ones are used to rebuild the tree interactively.
- Data reduction (Cattlet'91)
  - reducing data size by sampling and discretization.
  - still a main memory algorithm.
- Data partition and merge (Chan and Stolfo'91)
  - partitioning data and building trees for each partition.
  - merging multiple trees into a combined tree.
  - experiment results indicated reduced classification accuracy.

SLIQ -- A Scalable Classifier

- A fast scalable classifier by IBM Quest Group (Mehta et al.'96)
  - a disk-based algorithm
  - decision-tree based algorithm
- Issues in scalability
  - selecting the splitting attribute at each tree node
  - selecting splitting points for the chosen attribute
    - more serious for numeric attributes
  - fast tree pruning algorithm

SLIQ (I)

- Pre-sorting and breadth-first tree growing to reduce the costing of evaluating goodness of splitting numeric attributes.
  - build an index (attribute list) for each attribute to eliminate resorting data at each node of attributes
  - class list keeps track the leaf nodes to which samples belong
  - class list is dynamically modified during the tree construction phase
  - only class list and the current attribute list is required to reside in memory

SLIQ (II)

- Fast subsetting algorithm for determining splits for category attributes.
  - The evaluation of all the possible subsets of a categorical attribute can be prohibitively expensive, especially if the cardinality of the set is large.
  - If cardinality is small, all subsets are evaluated.
  - If cardinality exceeds a threshold, a greedy algorithm is used.
- Using inexpensive MDL-based tree pruning algorithm for tree pruning.

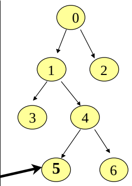
SLIQ (III) --- Data Structures

Credit Rating	rid
excellent	0
good	1
excellent	2
fair	3
fair	4
good	5
excellent	6

Age	rid
25	6
25	3
34	0
39	2
39	1
42	5
45	4

Disk Resident--Attribute List

rid	Send add	node
0	no	2
1	no	6
2	yes	3
3	no	2
4	yes	5
5	yes	5
6	no	2



Memory Resident--Class list

SPRINT (I)

- Removes all memory restrictions by using attribute list data structure
- SPRINT outperforms SLIQ when the class list is too large for memory, but needs a costly hash join to connect different attribute lists
- Designed to be easily parallelized

SPRINT (II) --- Data Structure

Age	Send Add	rid
25	no	6
25	no	3
34	no	0
39	yes	2
39	no	1
42	yes	5
45	yes	4

Credit Rating	Send Add	rid
excellent	no	0
good	no	1
excellent	yes	2
fair	no	3
fair	yes	4
good	yes	5
excellent	no	6

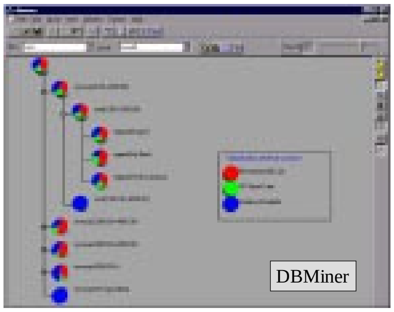
RainForest

- Gehrke, Ramakrishnan, and Ganti (VLDB'98)
- A generic algorithm that separates the scalability aspects from the criteria that determine the quality of the tree.
- Based on two observations:
  - Tree classifiers follow a greedy top-down induction schema.
  - When evaluating each attribute, the information about the class label distribution is enough.
  - AVC-list (attribute, value, class label) data structure.

Data Cube-Based Decision-Tree Induction

- Integration of generalization with decision-tree induction (Kamber et al'97).
- Classification at primitive concept levels
  - E.g., precise temperature, humidity, outlook, etc.
  - Low-level concepts, scattered classes, bushy classification-trees
  - Semantic interpretation problems.
- Cube-based multi-level classification
  - Relevance analysis at multi-levels.
  - Information-gain analysis with dimension + level.

Presentation of Classification Rules



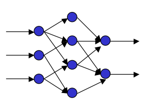
Data Classification Outline

- What is classification of data and prediction?
- How do we classify data by decision tree induction?
- What are neural networks and how can they classify?
- What is Bayesian classification?
- Are there other classification techniques?
- How do we predict continuous values?

What is a Neural Network?

A neural network is a data structure that supposedly simulates the behaviour of neurons in a biological brain.

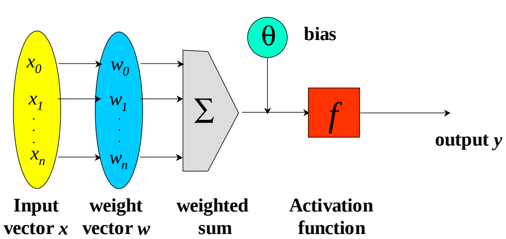
A neural network is composed of layers of units interconnected. Messages are passed along the connections from one unit to the other. Messages can change based on the *weight* of the connection and the value in the node.



Neural Networks

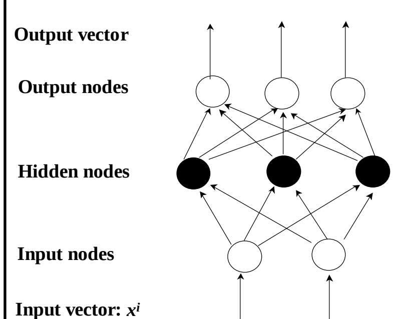
- Advantages
  - prediction accuracy is generally high.
  - robust, works when training examples contain errors.
  - output may be discrete, real-valued, or a vector of several discrete or real-valued attributes.
  - fast evaluation of the learned target function.
- Criticism
  - long training time.
  - difficult to understand the learned function (weights).
  - not easy to incorporate domain knowledge.

A Neuron

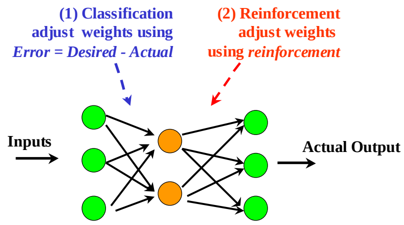


- The n-dimensional input vector x is mapped into variable y by means of the scalar product and a nonlinear function mapping.

Multi Layer Perceptron



Learning Paradigms



Learning Algorithms

- **Back propagation for classification**
- Kohonen feature maps for clustering
- Recurrent back propagation for classification
- Radial basis function for classification
- Adaptive resonance theory
- Probabilistic neural networks

Major Steps for Back Propagation Network

- Constructing a network
  - input data representation
  - selection of number of layers, number of nodes in each layer.
- Training the network using training data
- Pruning the network
- Interpret the results

Constructing the Network

- The number of input nodes: corresponds to the dimensionality of the input tuples.
  - age 20-80: 6 intervals
  - [20, 30) → 000001, [30, 40) → 000011, …., [70, 80) → 111111
- Number of hidden nodes: adjusted during training
- Number of output nodes: number of classes

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:
  - Initial weights are set randomly.
  - Input tuples are fed into the network one by one.
  - Activation values for the hidden nodes are computed.
  - Output vector can be computed after the activation values of all hidden node are available.
  - Weights are adjusted using error (desired output - actual output) and propagated backwards.

Network Pruning

- Fully connected network will be hard to articulate
- *n* input nodes, *h* hidden nodes and *m* output nodes lead to $h(m+n)$ links (weights)
- Pruning: Remove some of the links without affecting classification accuracy of the network.

Extracting Rules from a Trained Network

- Cluster common activation values in hidden layers.
- Find relationships between activation values and the output classes.
- Find the relationship between the input and activation values.
- Combine the above two to have rules relating the output classes to the input.

Data Classification Outline

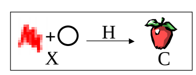
- What is classification of data and prediction?
- How do we classify data by decision tree induction?
- What are neural networks and how can they classify?
- What is Bayesian classification?
- Are there other classification techniques?
- How do we predict continuous values?

What is a Bayesian Classifier?

- It is a statistical classifier based on Bayes theorem.
- It uses probabilistic learning by calculating explicit probabilities for hypothesis.
- A naïve Bayesian classifier, that assumes total independence between attributes, is commonly used for data classification and learning problems. It performs well with large data sets and exhibits high accuracy.
- The model is incremental in the sense that each training example can incrementally increase or decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.

Bayes Theorem

- Given a data sample *X* with an unknown class label, *H* is the hypothesis that *X* belongs to a specific class *C.*
- The *posteriori probability* of a hypothesis *H, P(H|X), probability of X conditioned on H,* follows the Bayes theorem: $P(H|X)=\frac{P(X|H)P(H)}{P(X)}$



- P(H)=P( ) P(X) = P(+) P(X|H)=P(+ if)
- Practical difficulty: requires initial knowledge of many probabilities, significant computational cost.

Naïve Bayes Classifier

- Suppose we have n classes $C_1, C_2,\ldots,C_n$. Given an unknown sample X, the classifier will predict that $X=(x_1,x_2,\ldots,x_n)$ belongs to the class with the highest posteriori probability:

$X \in C_i$ if $P(C_i|X) > P(C_j|X)$ for $1 \le j \le n, j \ne i$

Maximize $\frac{P(X|C_i)P(C_i)}{P(X)}$ ➔ maximize $P(X|C_i)P(C_i)$

- $P(C_i) = s_i/s$
- $P(X|C_i)= \prod^{n} P(x_k|C_i)$ where $P(x_k|C_i) = s_{ik}/s_i$
- Greatly reduces the computation cost, only count the class distribution.
- Naïve: class conditional independence

Naïve Bayesian Classifier Example

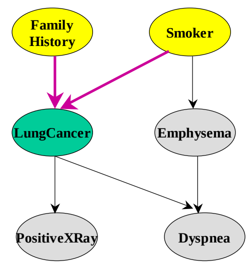
- Given a training set, we can compute the probabilities

Outlook	P	N		Humidity	P	N
sunny	2/9	3/5		high	3/9	4/5
overcast	4/9	0		normal	6/9	1/5
rain	3/9	2/5				
Tempreature				Windy		
hot	2/9	2/5		true	3/9	3/5
mild	4/9	2/5		false	6/9	2/5
cool	3/9	1/5				

Belief Network

- Allows class conditional dependencies to be expressed.
- It has a directed acyclic graph (DAG) and a set of conditional probability tables (CPT).
- Nodes in the graph represent variables and arcs represent probabilistic dependencies. (child dependent on parent)
- There is one table for each variable X. The table contains the conditional distribution P(X|Parents(X)).

Bayesian Belief Networks Example



Bayesian Belief Networks

	(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 for the variable LungCancer

Bayesian Belief Networks

Several cases of learning Bayesian belief networks:

- When both network structure and all the variables are given then the learning is simply computing the CPT.
- When network structure is given but some variables are not known or observable, then iterative learning is necessary (compute gradient *ln*P(S|H), take steps toward gradient and normalize).
- Many algorithms for learning the network structure exist.

Data Classification Outline

- What is classification of data and prediction?
- How do we classify data by decision tree induction?
- What are neural networks and how can they classify?
- What is Bayesian classification?
- Are there other classification techniques?
- How do we predict continuous values?

Other Classification Methods

- **Associative classification**: Association rule based condSet ➔ class
- **Genetic algorithm**: Initial population of encoded rules are changed by *mutation* and *cross-over* based on *survival* of accurate once (*survival*).
- **K-nearest neighbor classifier**: Learning by analogy.
- **Case-based reasoning**: Similarity with other cases.
- **Rough set theory**: Approximation to equivalence classes.
- **Fuzzy sets**: Based on fuzzy logic (truth values between 0..1).

Data Classification Outline

- What is classification of data and prediction?
- How do we classify data by decision tree induction?
- What are neural networks and how can they classify?
- What is Bayesian classification?
- Are there other classification techniques?
- How do we predict continuous values?

Prediction

Prediction of continuous values can be modeled by statistical techniques.

- Linear regression
- Multiple regression
- Polynomial regression
- Poisson regression
- Log-linear regression
- Etc.

Linear Regression

- Linear regression:

Approximate data distribution by a line Y = α + βX

Y is the *response variable* and X the *predictor variable*.

α and β are regression coefficients specifying the intercept and the slope of the line. They are calculated by least square method:

$$\beta = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$$

$$\alpha = \bar{y} - \beta\bar{x}$$

Where $\bar{x}$ and $\bar{y}$ are respectively the average of $x_1, x_2, \ldots, x_s$ and $y_1, y_2, \ldots, y_s$.

- Multiple regression: $Y = \alpha + \beta_1 X_1 + \beta_2 X_2$.
  - Many nonlinear functions can be transformed into the above.