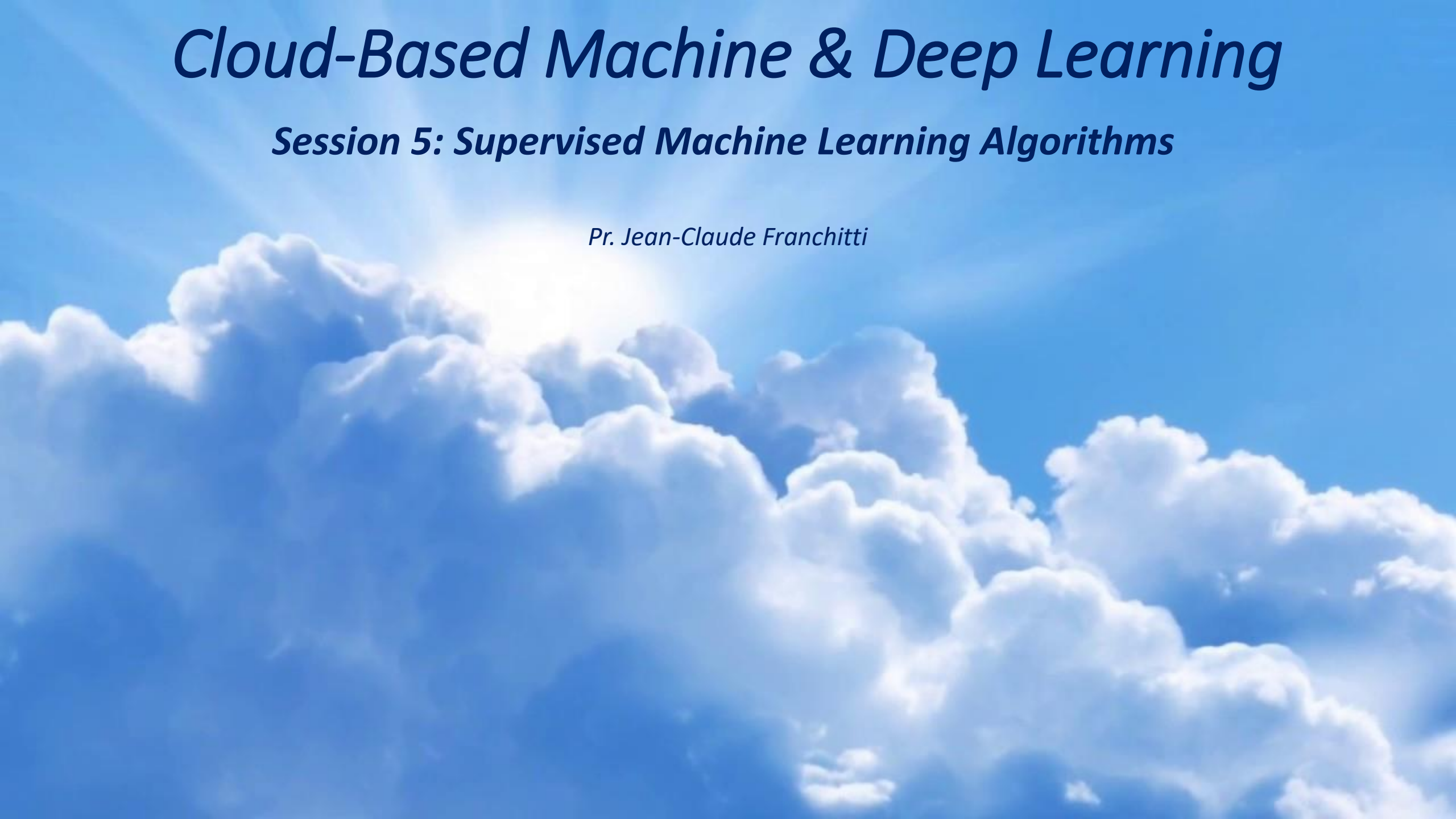
Cloud-Based Machine & Deep Learning

Session 5: Supervised Machine Learning Algorithms

Pr. Jean-Claude Franchitti

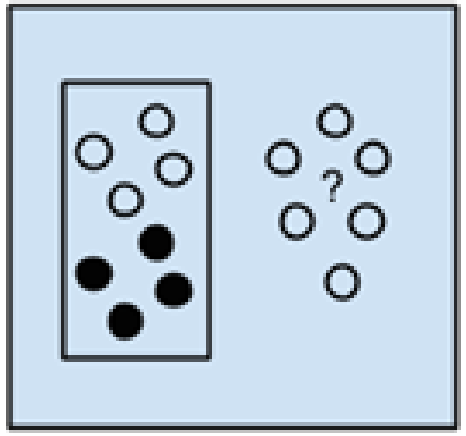


Basic Concept of Machine Learning

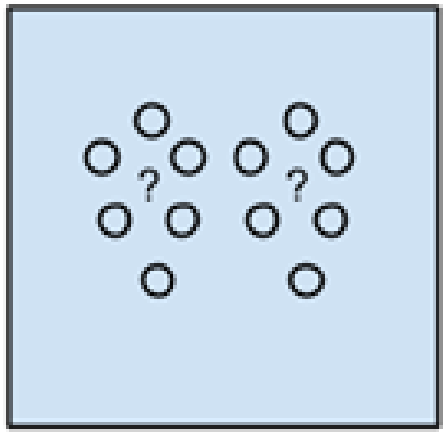
- Machine Learning is a scientific discipline that explores the construction and study of algorithms that can learn from data.
- Machine learning algorithms operate by building a model from executing example inputs and using that model to make predictions or decisions.
- Machine Learning is a subfield of computer science stemming from research into artificial intelligence. It has strong ties to statistics and mathematical optimization.

Taxonomy of Machine Learning Algorithms

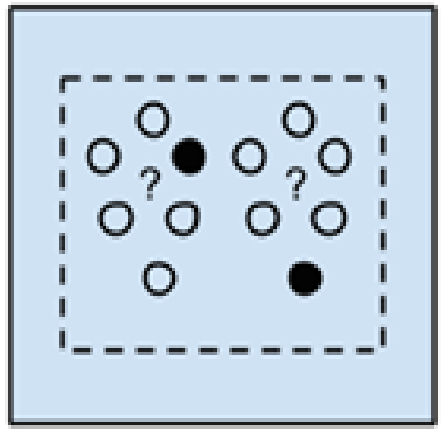
Machine Learning Based on Learning Styles



Supervised Learning Algorithms
(a)



Unsupervised Learning Algorithms
(b)

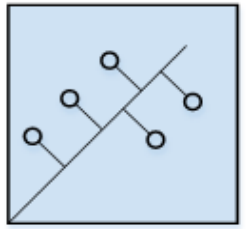


Semi-supervised Learning Algorithms
(c)

Figure 4.1 Machine learning algorithms grouped by different learning styles

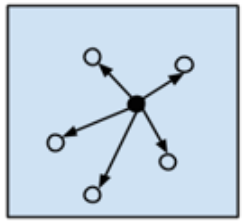
Taxonomy of Machine Learning Algorithms

Machine Learning Based on Similarity Testing



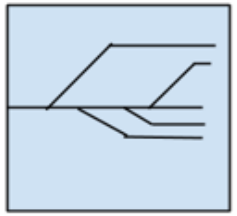
Regression Algorithms
(a)

- This model offers a supervised approach using statistical learning. The regression process is iteratively refined using an error criterion to make better predictions. This method minimizes the error between predicted value and actual experience in input data.



Instance-based
Algorithms
(b)

- This models a decision problem with instances or critical training data, as highlighted by the solid dots in. Figure(b) The data instance is built up with a database of reliable examples. A similarity test is conducted to find the best match to make a prediction. This method is also known as memory-based learning.

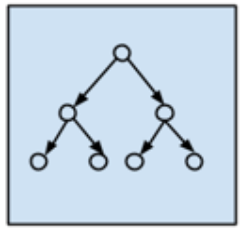


Regularization
Algorithms
(c)

- This method extends from the regression method that regulates the model to reduce complexity. This regularization process acts in favor of simpler models that are also better for generalization. Figure(c) shows how to sort the best prediction model among various design options.

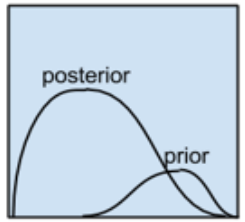
Taxonomy of Machine Learning Algorithms

Machine Learning Based on Similarity Testing



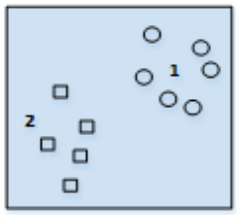
Decision tree Algorithms (d)

- The model is based on observation of the data's target values along various feature nodes in a tree-structured decision process. Various decision paths fork in the tree structure until a prediction decision is made at the leave node, hierarchically. Decision trees are trained on given data for better accuracy in solving classification and regression problems.



Bayesian Algorithms (e)

- The model is often applied in pattern recognition, feature extraction and regression applications. A Bayesian network is shown in Figure(e), which offers a directed acyclic graph (DAG) model represented by a set of statistically independent random variables. Both prior and posterior probabilities are applied in making predictions.

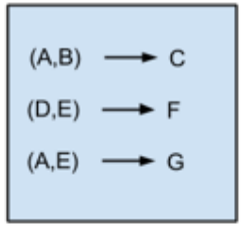


Clustering Algorithms (f)

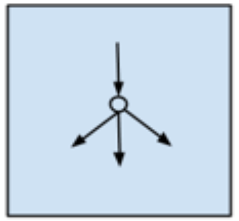
- This is a method based on grouping similar data objects as clusters. Two clusters are shown in Figure(f). Like regression, this method is unsupervised and modeled by using centroid-based clustering and/or hierarchal clustering. All clustering methods are based on similarity testing.

Taxonomy of Machine Learning Algorithms

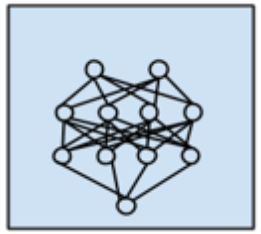
Machine Learning Based on Similarity Testing



Association Rule Learning Algorithms (g)



Artificial Neural Network Algorithms (h)

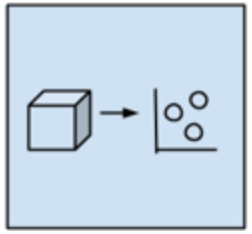


Deep Learning Algorithms (i)

- These are unsupervised with training data. Instead, the method generates inference rules that best explain observed relationships between variables in the data. These rules, as shown in Figure(g), are used to discover useful associations in large multidimensional datasets.
- These are cognitive models inspired by the structure and function of biological neurons. The ANN tries to model the complex relationships between inputs and outputs. They form a class of pattern matching algorithms that are used for solving deep learning.
- These extend from artificial neural networks by building much deeper and complex neural networks, as shown in Figure(i). Deep learning networks are built of multiple layers of interconnected artificial neurons. They are often used to mimic human brain processes in response to light, sound and visual signals. This method is often applied to semi-supervised learning problems, where large datasets contain very little labeled data.

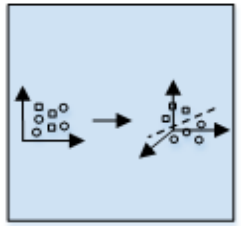
Taxonomy of Machine Learning Algorithms

Machine Learning Based on Similarity Testing



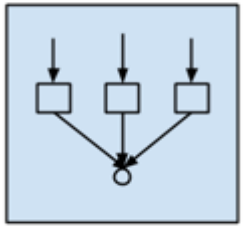
Dimensional Reduction Algorithms
(j)

- These exploit the inherent structure in the data in an unsupervised manner. The purpose is to summarize or describe data using less information. This is done by visualizing multi-dimensional data with principal components or dimensions. Figure(j) shows the reduction from a 3-D e space to a 2-D data space.



Support Vector Machine Algorithms
(k)

- These are often used in supervised learning methods for regression and classification applications. Figure(k) shows how a hyperplane (a surface in a 3-D space) is generated to separate the training sample data space into different subspaces or categories.



Ensemble Algorithms
(l)

- These are models composed of multiple weaker models that are independently trained. The prediction results of these models are combined in Figure(l), which makes the collective prediction more accurate. Much effort is put into what types of weak learners to combine and the ways in which to combine them effectively.

Taxonomy of Machine Learning Algorithms

Supervised Machine Learning Algorithms

Table 4.1 Supervised machine learning algorithms

ML Algorithm Classes	Algorithm Names
Regression	Linear, Polynomial, Logistic, Stepwise, OLSR (Ordinary Least Squares Regression), LOESS (Locally Estimated Scatterplot Smoothing), MARS (Multivariate Adaptive Regression Splines)
Classification	KNN (k-nearest Neighbor), Trees, Naïve Bayesian, SVM (Support Vector Machine), LVQ (Learning Vector Quantization), SOM (Self-Organizing Map), LWL (Locally Weighted Learning)
Decision Trees	Decision trees, Random Forests, CART (Classification and Regression Tree), ID3 (Iterative Dichotomiser 3), CHAID (Chi-squared Automatic Interaction Detection), ID3 (Iterative Dichotomiser 3), CHAID (Chi-squared Automatic Interaction Detection)
Bayesian Networks	Naïve Bayesian, Gaussian, Multinomial, AODE (Averaged One-Dependence Estimators), BBN (Bayesian Belief Network), BN (Bayesian Network)

Taxonomy of Machine Learning Algorithms

Unsupervised Machine Learning Algorithms

Table 4.2 Some unsupervised machine learning algorithms

ML Algorithm Classes	Algorithm Names
Association Analysis	A priori, Association Rules, Eclat, FP-Growth
Clustering	Clustering analysis, k-means, Hierarchical Clustering, Expectation Maximization (EM), Density-based Clustering
Dimensionality Reduction	PCA (principal Component Analysis), Discriminant Analysis, MDS (Multi-Dimensional Scaling)
Artificial Neural Networks (ANNs)	Perception, Back propagation, RBFN (Radial Basis Function Network)

Regression Methods for Machine Learning

Basic Concepts of Regression Analysis

- Regression analysis performs a sequence of parametric or non-parametric estimations. The method finds the causal relationship between the input and output variables.
- The estimation function can be determined by experience using a priori knowledge or visual observation of the data.
- Regression analysis is aimed to understand how the typical values of the output variables change, while the input variables are held unchanged.
- Thus regression analysis estimates the average value of the dependent variable when the independent variables are fixed.

Regression Methods for Machine Learning

Basic Concepts of Regression Analysis

- Most regression methods are parametric in nature and have a finite dimension in the analysis space. We will not deal with nonparametric regression analysis, which may be infinite-dimensional.

- The accuracy or the performance of regression methods depends on the quality of the dataset used. In a way, regression offers an estimation of continuous response variables, as opposed to the discrete decision values used in classification.

Regression Methods for Machine Learning

Formulation of A Regression Process

To model a regression process, the unknown parameters are often denoted as β, which may appear as a scalar or a vector. The independent variables are denoted by an input vector X and the out is the dependent variable as Y. When multiple dimensions are involved, these parameters are vectors in form. A regression model establishes the approximated relation between X, β. and Y as follows:

$$Y = f(X, \beta)$$

- The function $f(X, \beta)$ is approximated by an expected value $E(Y \mid X)$. The regression function f is based on the knowledge of the relationship between a continuous variable Y and a vector X. If no such knowledge is available, an approximated form is chosen for f .
- Consider k components in the vector of unknown parameters β, *also known as weights*. We have three models to relate the inputs to the output, depending on the relative magnitude between the number N of observed data points of the form (X, Y) and the dimension k of the sample space.

Regression Methods for Machine Learning

Three Cases To be Modeled in A Regression Process

- When $N < k$, most classical regression analysis methods can be applied. Since the defining equation is underdetermined, there are not enough data to recover the unknown parameters β.

- When $N = k$ and the function f is linear, the equations $Y = f(X, \beta)$ can be solved exactly without approximation, because there are N equations to solve N components in β. The solution is unique as long as the X components are linearly independent. If f is nonlinear, many solutions may exist or no solution at all.

- In general, we have the situation that $N > k$ data points. This implies that there is enough information in the data that can estimate a unique value for β under an overdetermined situation.

Regression Methods for Machine Learning

Three Basic Assumptions For Regression Analysis

- The sample is representative of the data space involved. The error is a random variable with a mean of zero conditional on the explanatory variables.
- The independent variables are measured with no error. The predictors are linearly independent.
- The errors are uncorrelated and the variance of the error is constant across observations. If not, the weighted least squares methods may be used.

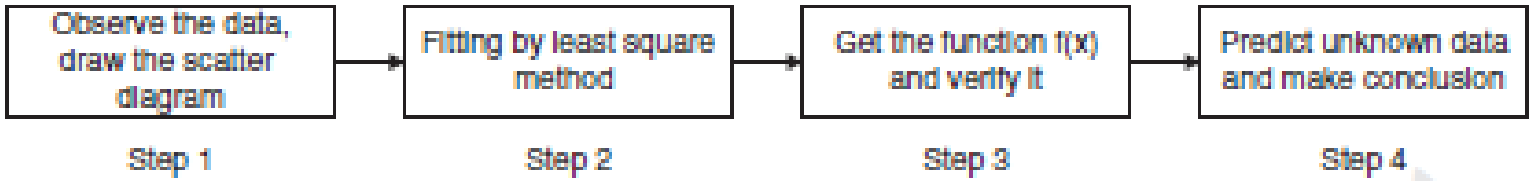


Figure 4.3 *Major steps in linear regression*

Regression Methods for Machine Learning

Unitary Linear Regression Analysis

Consider a set of data elements in a 2-D sample space, (*x1, y1), (x2, y2),⋯, (xn, yn). All* points are mapped into a scatter diagram. If they can be covered, approximately, by a straight line, then we obtain the following linear regression expression:

$$y = ax + b + \varepsilon$$

where *x stands for explanatory variable, y stands for explained variable, a and b are corresponding* coefficients, and *ε is the random error, which follows independent normal* distribution with the same distribution with mean *E(ε)and variance Var(ε). Then we* need to work out the expectation by using a linear regression expression:

$$y = ax + b$$

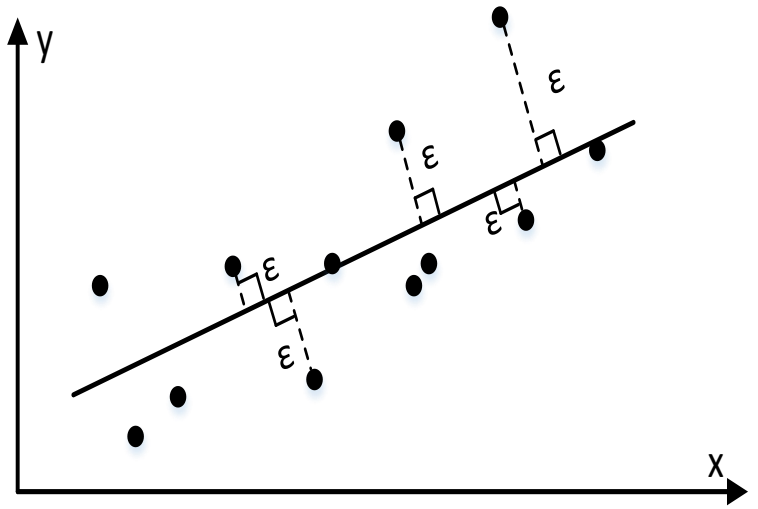


Figure 4.4 Unitary linear regression analysis.

Regression Methods for Machine Learning

Unitary Linear Regression Analysis

The main task for regression analysis is to conduct estimations for coefficient a and b through observations on n groups of samples. The common method is the least square method, and its objective function is given by:

$$\min Q(\hat{a},\hat{b}) = \sum_{i=1}^{n}[\mathrm{y}_i - \mathrm{E}(\mathrm{y}_i)]^2 = \sum_{i=1}^{n}(\mathrm{y}_i - \hat{a}x_i - \hat{b})^2$$

To minimize the sum of squares, we need to calculate the partial derivative of *Q for* $\hat{a},\hat{b}$,and make them zero, as shown below:

$$\begin{cases} \dfrac{\partial Q}{\partial \hat{b}} = \sum_{i=1}^{n}(y_i - \hat{a}x_i - \hat{b}) = 0 \\ \dfrac{\partial Q}{\partial \hat{a}} = \sum_{i=1}^{n}(y_i - \hat{a}x_i - \hat{b})x_i = 0 \end{cases} \xrightarrow{solve} \begin{cases} \hat{a} = \dfrac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \\ \hat{b} = \overline{y} - \hat{a}\overline{x} \end{cases}$$

Regression Methods for Machine Learning

Unitary Linear Regression For Binary Classification

When we calculate the regression equation $y = \hat{a}x + \hat{b}$, we may work out the estimated value of the dependent variable for each sample in the training dataset; the formula is $\hat{y} = \hat{a}x + \hat{b}$, thus it assumes two possible values:

$$class = \begin{cases} 1 & y_i > \hat{y}_i \\ 0 & y_i < \hat{y}_i \end{cases} \quad i = 1, 2, \cdots, n$$

The initial data (x_0, y_0) is used for classification. First, we determine $\hat{y}_0$ by using the dependent variable x_0 , then we compare $y_0, \hat{y}_0$ to determine to which class it belongs. For multivariate linear regression, as studied below, this method is also applied to classify a dataset.

Regression Methods for Machine Learning

Multivariate Linear Regression Analysis

model of multivariate linear regression analysis

$$\begin{cases} y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m + \varepsilon \\ \varepsilon - N(0, \sigma^2) \end{cases}$$

By working out the expectation for the structure above, we get the multivariate linear regression the relationship equation (substituted *y for E(y)) as:*

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m$$

Matrix form: $y = X\beta$ $X = [1, x_1, \cdots x_m], \beta = [\beta_0, \beta_1, \cdots, \beta_m]^T$

Objective is given as: $\min Q = \sum_{i=1}^{n} \varepsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_1 - \cdots - \beta_m x_{im})^2$

Final regression equation: $y = X\hat{\beta} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_m x_m$

Regression Methods for Machine Learning

Example : Healthcare Data Analysis with Linear Regression

With the improvement of the economy, more people are concerned about their health condition. As an example, obesity is reflected by the weight index. A fat person is more likely to have high blood pressure or diabetes. Using a linear regression model, we predict the relationship. Table 4.4 shows the dataset for body weight index and blood pressure of some people who received a health examination at a hospital in Wuhan, China. We conduct a preliminary judgment on what is the datum of blood pressure of a person with a body weight index of 24.

Table 4.3 Data sheet for body weight index and blood pressure

id	Body Weight Index	Blood Pressure(mmHg)	id	Body Weight Index	Blood Pressure(mmHg)
1	20.9	123	8	21.4	126
2	21.5	123	9	21.4	124
3	19.6	123	10	25.3	129
4	26	130	11	22.4	124
5	16.8	119	12	26.1	133
6	25.9	131	13	23	129
7	21.6	127	14	16	118

Regression Methods for Machine Learning

- First, determine distribution of the data points, and draw a scatter diagram for body weight index–blood pressure with MATLAB, as shown in Figure 4.5.
- All data points are almost on or below the straight line, and they are linearly distributed. Therefore, the data space is modeled by a unitary linear regression process. By the least square method we get $\begin{cases} \hat{a} = 1.32 \\ \hat{b} = 96.58 \end{cases}$
- Therefore we have: *y =1.32x + 96.58.*
- A significance test is needed to verify whether the model will fit well with the current data.

$$\begin{cases} avrerr = 1.17 \\ R^2 = 0.90 \end{cases}$$

- The mean residual is much less than the mean value 125.6 of blood pressure, and the coefficient of determination is close to 1. Then a prediction is made through calculation

y = 1.32 × 24 + 96.58 = 128.

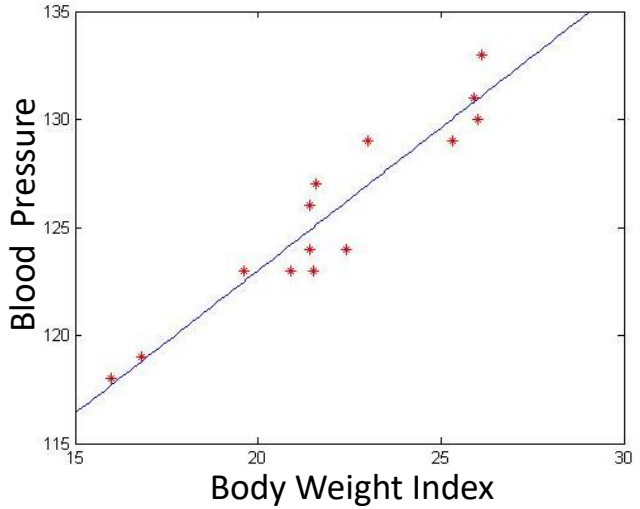


Figure 4.5 The relation between body weight and blood pressure

Regression Methods for Machine Learning

Logistic Regression For Classification

- Logistic regression is a linear regression analysis model in a broad sense, and may be used for prediction and classification. It is commonly used in fields such as data mining, automatic diagnosis for diseases and economical prediction.
- The logistic model may only be used to solve problem of dichotomy. As for logistic regression classification, the principle is to conduct classification to sample data with a logistic function, known as a sigmoid function defined by

$$f(x) = 1 / (1 + e^{-x})$$

- The output range of the sigmoid function is in the range (0, 1). In this sense, the sigmoid function is a probability density function for the sample data shown in Figure 4.6.

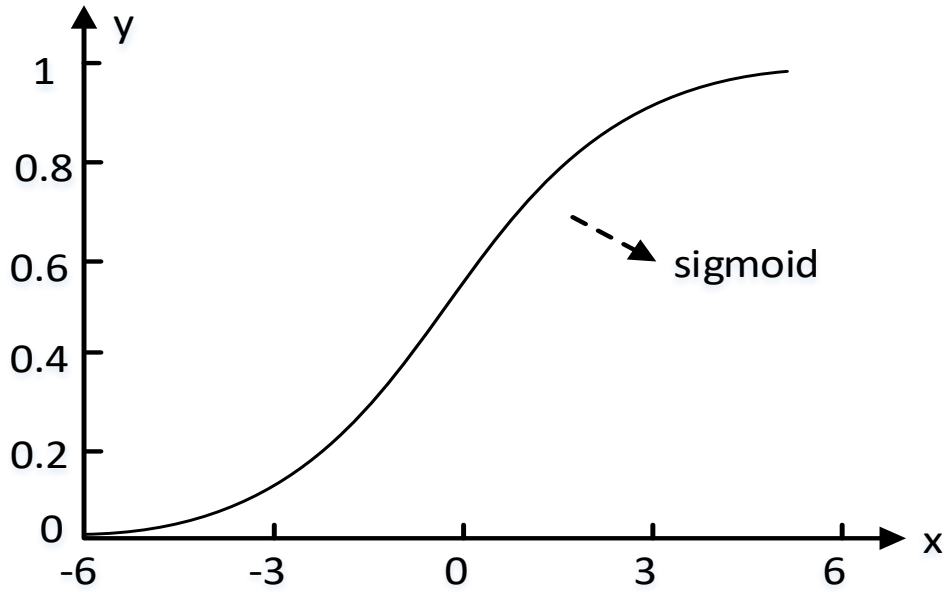


Figure 4.6 The curve of the sigmoid function applied in the regression method.

Regression Methods for Machine Learning

Logistic Regression For Classification

The basic idea of logistic regression is to consider vector *x* with m independent input variables. Each dimension of *x* stands for one attribute (feature) of the sample data (training data). In logistic regression, multiple features of the sample data are combined into one feature by using linear function.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_m x_m$$

We need to figure out the probability of the feature with designated data and apply the sigmoid function to act on that feature. We obtain the logistic regression as plotted in Fig. 7.6.

$$\begin{cases} P(Y=1 \mid x) = \pi(x) = \dfrac{1}{1+e^{-z}} \\ z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_m x_m \end{cases} \rightarrow \begin{cases} x \in 1, \, if \;\; P(Y=1 \mid x) > 0.5 \\ x \in 0, \, if \;\; P(Y=0 \mid x) < 0.5 \end{cases}$$

Logistic Regression For Classification

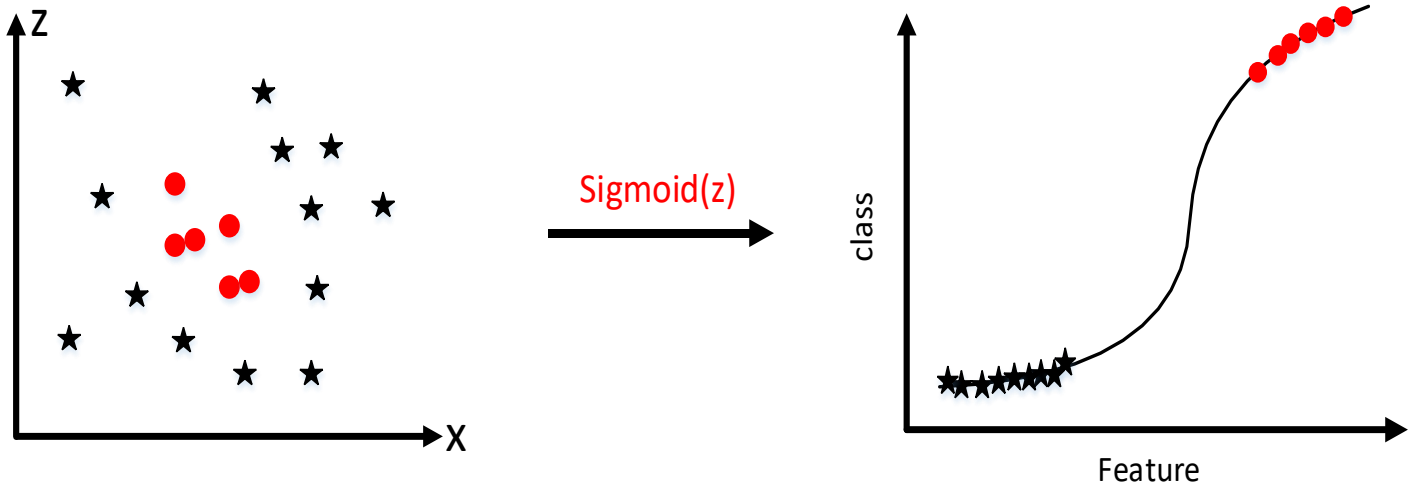


Figure 4.6 Principle of using logistic regression for classification purposes

Supervised Classification Methods

- Four Supervised Classification Methods
  - decision tree
  - rule-based classifier
  - nearest neighbor classifier
  - support-vector machines
- Three steps in building a classification model

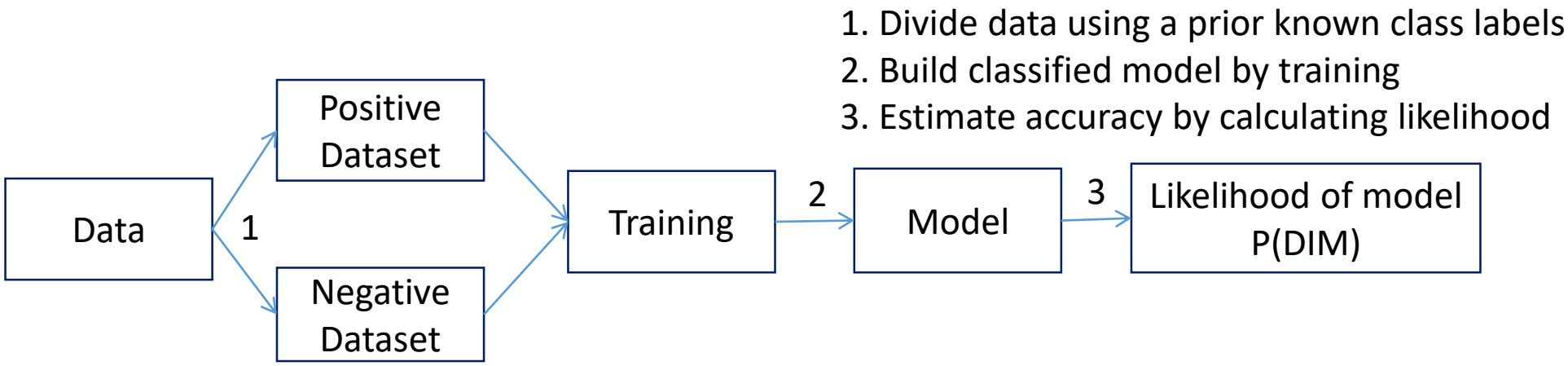


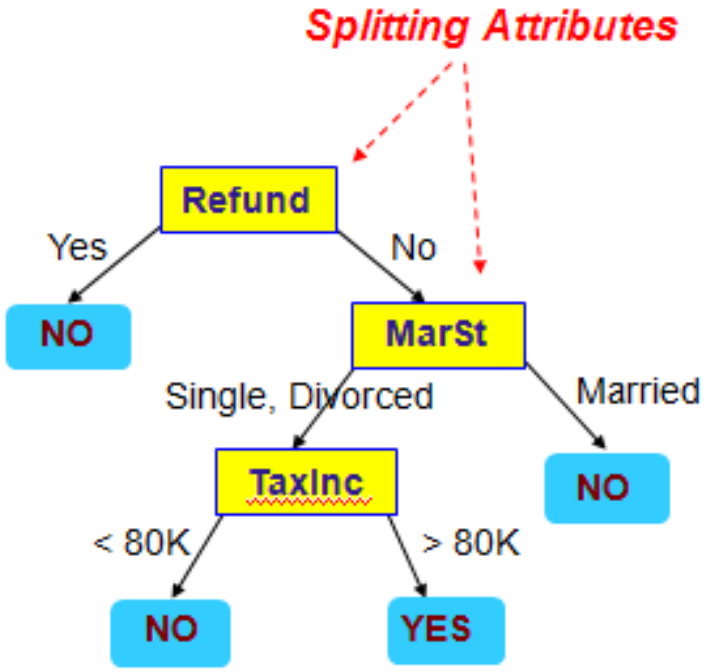
Figure 4.7 Three steps in building a classification model through sample data training.

Decision Trees for Machine Learning

Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

Decision Trees for Machine Learning

- Basic Concepts of Regression Analysis

  A decision tree offers a predictive model in both data mining and machine learning. We will concentrate on the machine learning using of decision trees. The goal is to create a model that predicts the value of an output target variable at the leaf nodes of the tree, based on several input variables or attributes at the root and interior nodes of that tree.

- Many Algorithms
  - ID3
  - C4.5
  - CART
  - SLIQ,SPRINT

Decision Trees for Machine Learning

ID3 Algorithm Tagging

The core idea of the ID3 algorithm takes the information gain of the attribute as the measure, and splits the attribute with the largest information gain after splitting, to make the output partition on each branch belong to the same class as far as possible. The measure standard of information gain is entropy, which depicts the purity of any example set. Given a training set *S of positive and negative examples, the entropy function of S is defined as*

$$Entropy(S) = -p_+ \log_2^{p_+} - p_- \log_2^{p_-}$$

where *p+ represents positive examples and p- represents negative examples. If the target* attribute possesses *m different values, then the entropy of S relative to classifications of m classes is defined by*

$$Entropy(S) = \sum_{i=1}^{m} -p_i \log_2^{p_i}$$

Decision Trees for Machine Learning

ID3 Algorithm Tagging

The measure standard of the effectiveness of training data is defined as the entropy, which is the standard for measuring training example set purity, and the above measure standard is called the “information gain”. The information gain of an attribute shows the decrease of expected entropy caused by segmented examples. We define the gain *Gain(S, A) of an attribute A in set S as*

$$Gain(S, A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where *V(A) is the range of A, S is the sample set and Sv is the sample set with A value* equal to *v*

Decision Trees for Machine Learning

Example: Decision Tree Prediction using the ID3 Algorithm

Given a training set *D with 500 samples, where the data format is shown in Table 4.4*

id	Annual income($)	Age	Marital status	Class load
1	70K	18	Single	No
2	230K	35	Divorce	Yes
3	120K	28	Married	Yes
4	200K	30	Married	Yes

Class label attribute “load” has two different values (i.e. {yes, no}), therefore there are two different categories (i.e. m = 2). Suppose category C1 corresponds to “yes”, and category C2 corresponds to “no”. There are 300 tuples in category “yes”, and 200 tuples in category “no”. And (root) node N is created for tuples in D. The information gain of each attribute must be calculated in order to find the split criterion of those tuples. The entropy value is used to classify the tuples in *D as*

$$Entropy(D) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

Decision Trees for Machine Learning

Then we calculate the expected information demand of each attribute. For the income attribute of equal or greater than 80 K, there are 250 *"yes" tuples and 100 "no" tuples*

$$Entropy_{income}(D) = \frac{7}{10} \times (-\frac{5}{7}\log_2\frac{5}{7} - \frac{2}{7}\log_2\frac{2}{7}) + \frac{3}{10} \times (-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}) = 0.8797$$

$$Gain(D, income) = Entropy(D) - Entropy_{income}(D) = 0.9710 - 0.8797 = 0.0913$$

Similarly, information of age and marital status can be calculated.

$$Entropy_{age}(D) = \frac{1}{2} \times (-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}) + \frac{1}{2} \times (-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}) = 0.8464$$

$$Gain(D, age) = Entropy(D) - Entropy_{age}(D) = 0.9710 - 0.8464 = 0.1246$$

$$Gain(D, marry) = 0.9710 - 0.9510 = 0.02$$

From the above calculation, the information gain is the largest when using age attribute, therefore this attribute is selected to be the classification criterion.

Rule-based Classification

Example of a Rule-based Classification

The rule-based classifier is a technique to use a set of “if then…” rules to classify records. Consider the use of three prediction rules:

1) *r1: (Body temperature = Cold blood) → Non-mammalian*
2) *r2: (Body temperature = Constant temperature) ∧ (Viviparity = Yes) → Mammalian*
3) *r3: (Body temperature = Constant) ∧ (Viviparity = No) → Non-mammalian*

Rule-based Classification

- Two properties to improve the applicability of the rules
  - Mutual exclusion rule

    If there are no rules triggered by the same record in the rule set R, it is said that rules in the rule set R are mutually exclusive. This property ensures that daily records are covered by one rule at most in R. The above rule set is a mutually exclusive one.

  - Exhaustive rule

    If for any combination of property values, there is a rule in R to cover it, it is said that the rule set R is with exhaustive coverage. This property ensures that daily records are covered by one rule at least in R.

Rule-based Classification

- Two solutions of determine the classification result
  - Ordered rules

    This kind of rule set is ordered from large to small in accordance with the rule priority, which is defined generally with precision, coverage and so on. When classifying, rules are scanned in sequence until a rule covering the record is found, and this rule will be the classification result of this record. General rule-based classifiers adopt this method.

  - Unordered rules

    In this case, all rules are equal to each other. The rules are scanned successively, and after a record occurs, each will be chosen, and the one getting the most votes will be the final classification result of the record.

Rule-based Classification

Rule Extraction with Direct Rule

- The sequential coverage algorithm is often used to directly extract rules from data, and the growth of rules is usually in a greedy manner based on some kind of evaluation measure.
- The algorithm extracts a class of rules at a time from the record containing more than one training data.

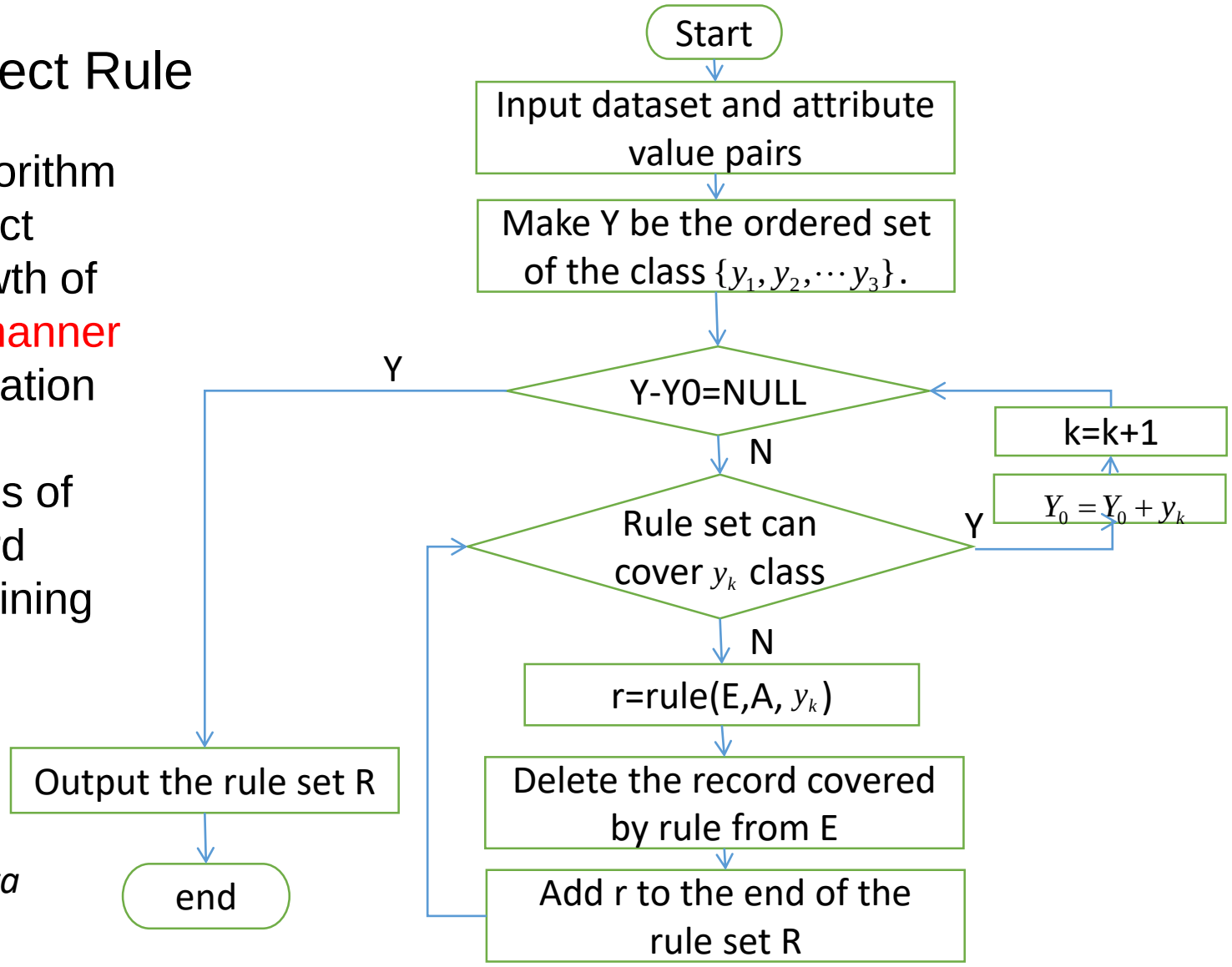
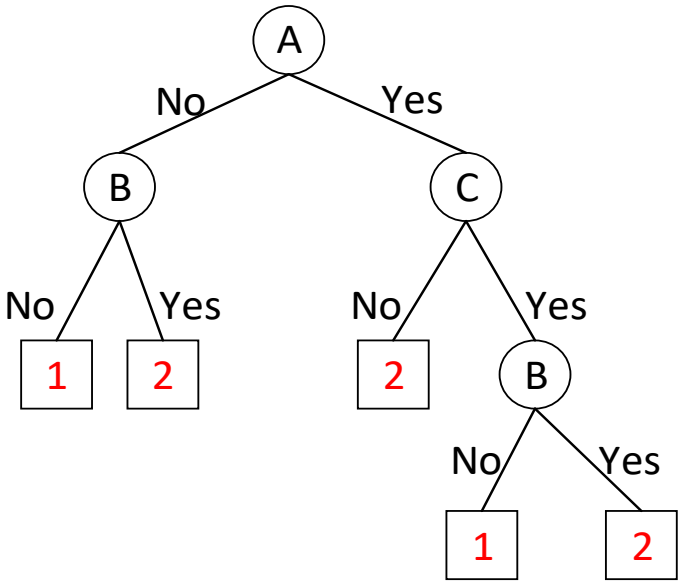


Figure 4.8 Sequential coverage and data flow for rule extraction

Rule-based Classification

Rule Extraction from Decision Tree

Rule extraction from decision tree modeling is a common indirect method for rule extraction. In principle, each path of the decision tree from its root node to its leaf node may express a classification rule.



Rule set:

$$r_1 : (A = No, B = No) \rightarrow 1$$

$$r_2 : (A = No, B = Yes) \rightarrow 2$$

$$r_3 : (A = Yes, C = No) \rightarrow 2$$

$$r_4 : (A = Yes, C = Yes, B = No) \rightarrow 1$$

$$r_5 : (A = Yes, C = Yes, B = Yes) \rightarrow 2$$

Figure 4.9 Rule set generated from using decision tree.

Rule-based Classification

Example Diabetes Prediction using Rule-Based Classification

Table 4.10 shows a dataset of blood glucose (high, low), weight (overweight, normal), lipid content and diabetic (*yes, no*) from physical examination of some people in Wuhan, based on which the corresponding rule sets may be constituted, and convenient to classify people into two categories, i.e. the diabetic and the normal.

Table 4.10 Physical examination dataset for diabetes

ID	Blood glucose	Weight	Blood lipid content (mmol/L)	Diabetic(Yes or No)
1	Low	Overweight	2.54	No
2	High	Normal	1.31	No
3	High	Overweight	1.13	No
4	Low	Normal	2.07	No
5	High	Overweight	2.34	**Yes**
6	High	Normal	0.55	No
7	Low	Overweight	2.48	No
8	High	Overweight	3.12	**Yes**
9	High	Normal	1.14	No
10	High	Overweight	8.29	Yes

The Nearest Neighbor Classifier

The Rote classifier

a kind of passive learning method, will not classify the test data until it matches a certain training dataset instance completely. The method has an apparent disadvantage that most of the test data instances cannot be classified, because no training dataset matches them.

The nearest neighbor classifier

find all the training dataset instances which have the most similar properties as the test sample. the nearest neighbor classifier considers each sample as a n-dimensional point, *and determines* the nearest neighbor between two given points.

Euclidean distance: $d(x, y) = \sum_{k=1}^{n} | x_k - y_k |$

The Nearest Neighbor Classifier

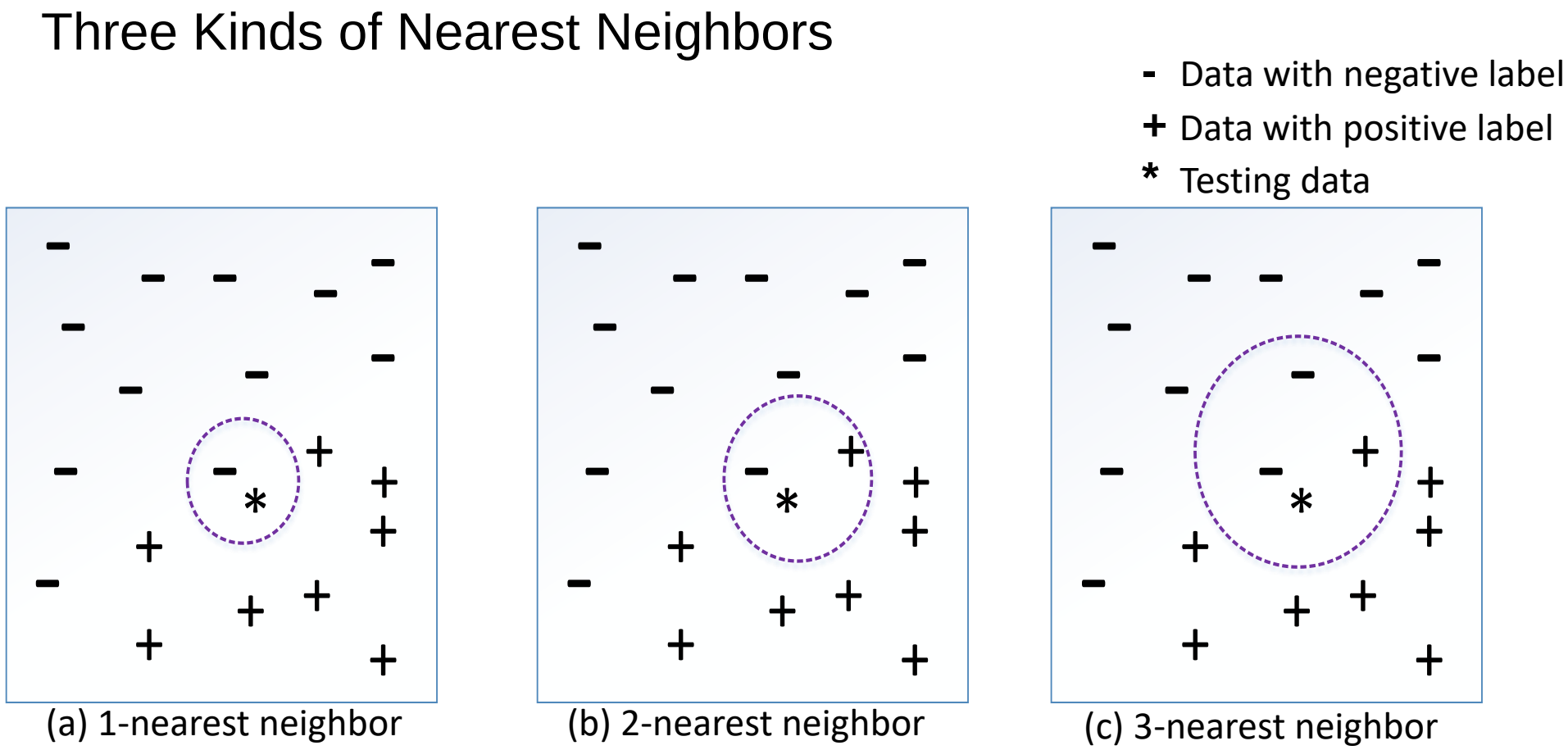


Figure 4.10 Instance of three kinds of nearest neighbors

The Nearest Neighbor Classifier

Two Methods of Choosing the Class Label

- majority voting

$$y = \arg\max_{v} \sum_{(x_i, y_i) \in D_x} I(v = y_i)$$

- weighted distance voting

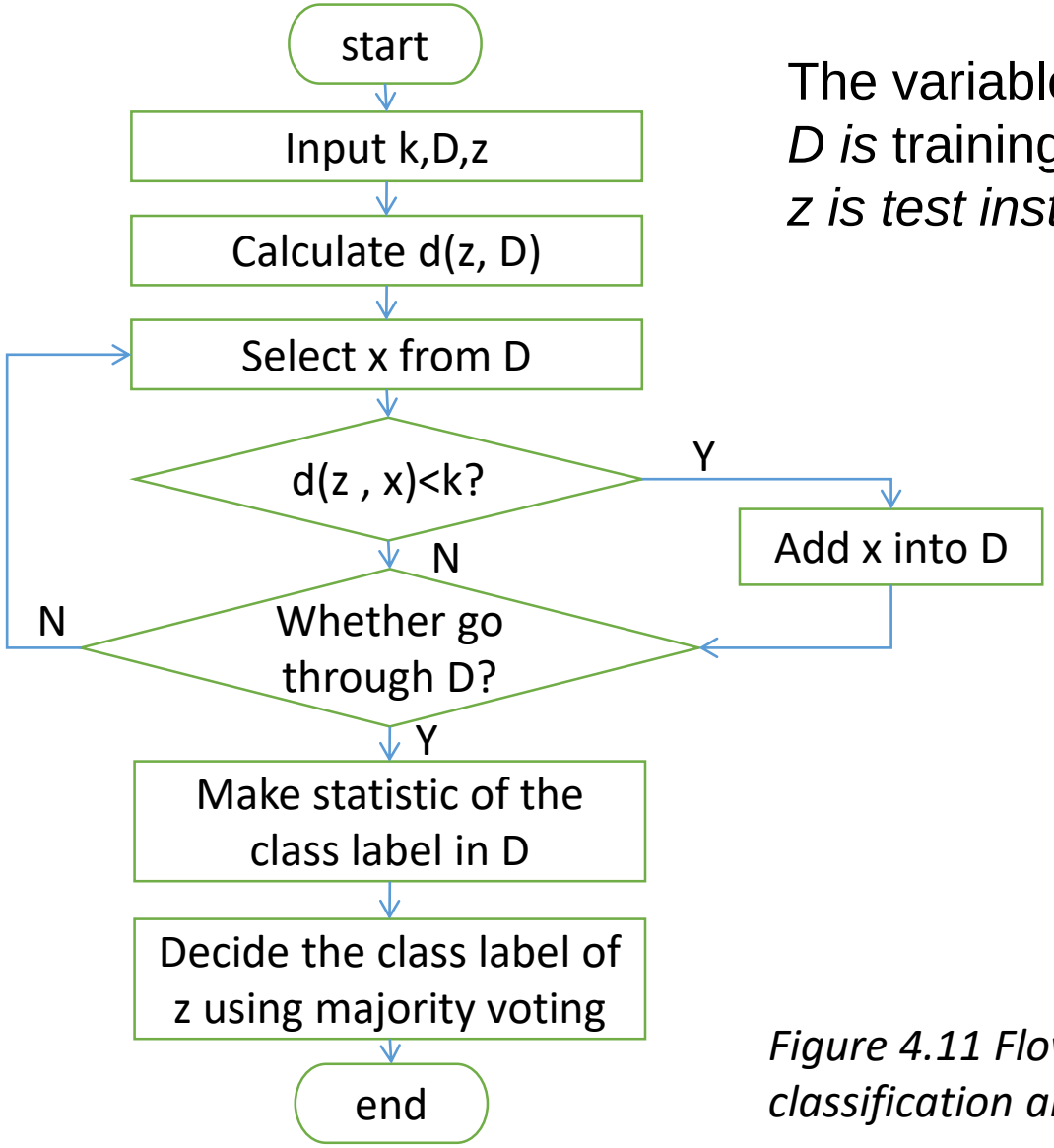
  Some of the nearest neighbor samples are very important

$$y = \arg\max_{v} \sum_{(x_i, y_i) \in D_x} w_i \times I(v = y_i)$$

I(·) is an indicator function defined as:

$$I(y_i) = \begin{cases} 1 & y_i = v \\ 0 & y_i \neq v \end{cases}$$

The Nearest Neighbor Classifier



The variable *k represents distance threshold*
D is training dataset
z is test instance.

Figure 4.11 Flow chart for the nearest neighbor classification algorithm

Support Vector Machines

Find A Linear Hyperplane(decision boundary) that will separate the data

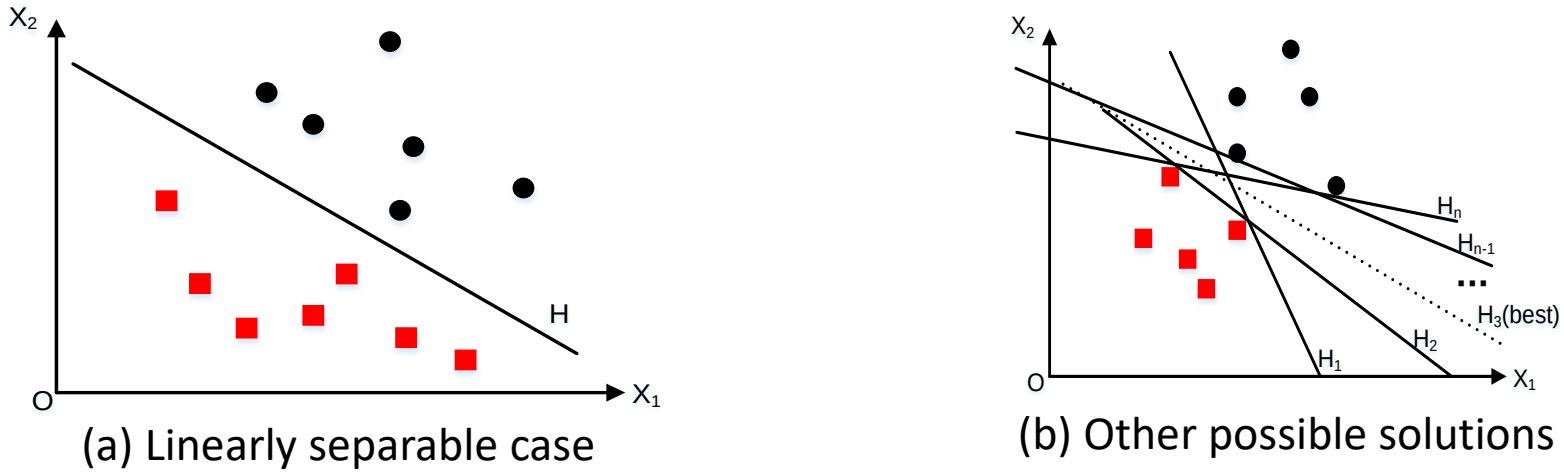


Figure 4.12 The concept of using SVM to classify between two classes of sample data.

How to find the “best” line, i.e. the one with the minimum classification error?

Support Vector Machines

Definition of Maximal Margin Hyperplane

Consider those squares and circles nearest to the decision boundary, as shown in Figure 4.20; adjust parameters *w and b, and two parallel hyperplanes H1 and H2 can be* represented by

$$H_1: \ w^T x + b = 1$$
$$H_2: \ w^T x + b = -1$$

The margin of the decision boundary is given by the distance between those two hyperplanes. To calculate the margin, make *x1 the data point on H1, and x2 the data* point on *H2, and insert x1 and x2 into the above formula, then margin d can be obtained* by subtracting the formulas:

$$w(x_1 - x_2) = 2 \qquad d = \frac{2}{\| w \|}$$

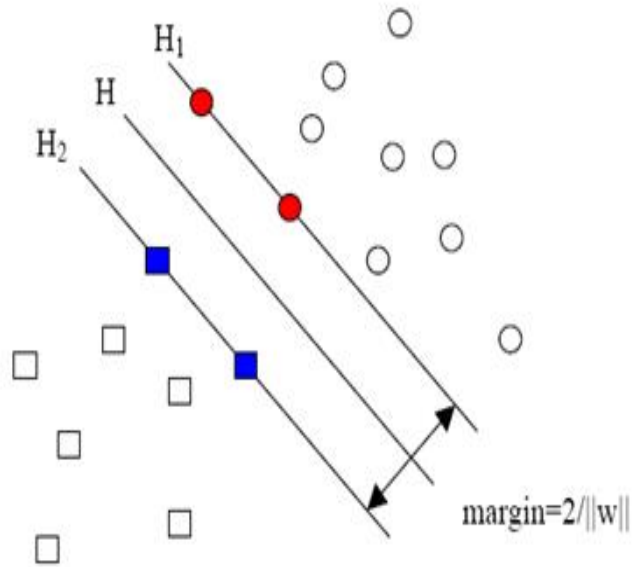


Figure 4.12 Linearly separating hyperplane with maximized margin from each class.

Support Vector Machines

Formal SVM Model

We want to maximize: $\text{Margin} = \frac{2}{\| \vec{w} \|^2}$

Which is equivalent to minimizing: $L(w) = \frac{\| \vec{w} \|^2}{2}$

But subjected to the following constraints:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases}$$

This is a constrained optimization problem

Numerical approaches to solve it (e.g., quadratic programming)

Support Vector Machines

Non-Linear Hyperplanes

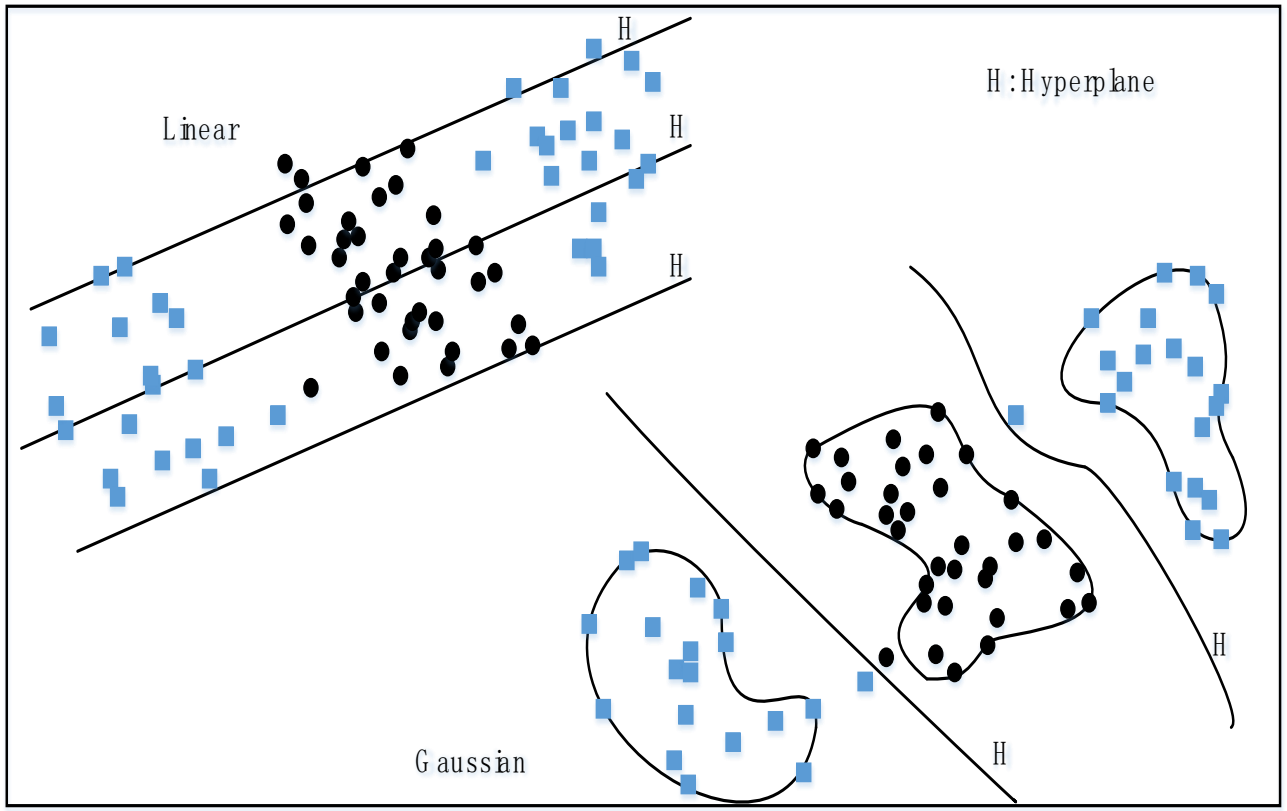


Figure 4.13 Nonlinear support vector machine

Support Vector Machines

Non-Linear Hyperplanes

What if the problem is not linearly separable?

Introduce slack variables

Need to minimize: $L(w) = \frac{\| \vec{w} \|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k \right)$

Subject to:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

Bayesian Network and Ensemble Methods

Bayesian Classifiers

- A probabilistic framework for solving classification problems

- Conditional Probability: $P(C \mid A) = \frac{P(A,C)}{P(A)}$

$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

- Bayes theorem: $P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$

Bayesian Network and Ensemble Methods

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A_1, A_2,…,A_n)
  - Goal is to predict class C
  - Specifically, we want to find the value of C that maximizes P(C| A_1, A_2,…,A_n)
- Can we estimate P(C| A_1, A_2,…,A_n) directly from data?

Bayesian Network and Ensemble Methods

Bayesian Classifiers

- Approach:
  - compute the posterior probability P(C | A_1, A_2, …, A_n) for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}$$

  - Choose value of C that maximizes
    P(C | A_1, A_2, …, A_n)
  - Equivalent to choosing value of C that maximizes
    P(A_1, A_2, …, A_n|C) P(C)
- How to estimate P(A_1, A_2, …, A_n | C)?

Bayesian Network and Ensemble Methods

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
  - $P(A_1, A_2, \ldots, A_n \mid C) = P(A_1 \mid C_j)\, P(A_2 \mid C_j) \ldots P(A_n \mid C_j)$
  - Can estimate $P(A_i \mid C_j)$ for all A_i and C_j
  - New point is classified to C_j if $P(C_j) \prod P(A_i \mid C_j)$ is maximal

Bayesian Network and Ensemble Methods

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero

Probability estimation:

$$\text{Original}: P(A_i \mid C) = \frac{N_{ic}}{N_c}$$

$$\text{Laplace}: P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c}$$

$$\text{m-estimate}: P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m}$$

c: number of classes

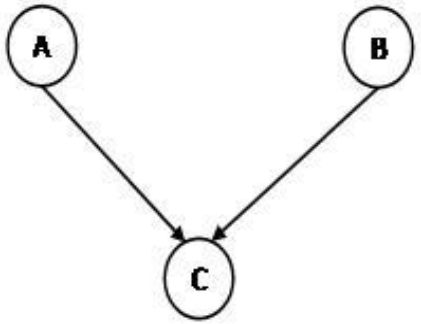
p: prior probability

m: parameter

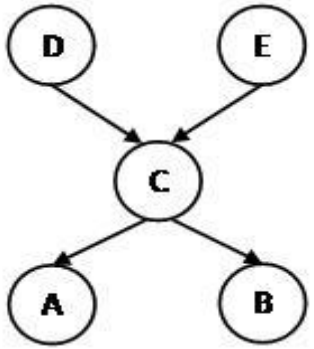
Bayesian Network and Ensemble Methods

Bayesian Belief Networks

- The Bayesian belief network is a graphical representation of the relationship among attributes
  - a directed acyclic graph, representing the dependencies between the variables;
  - a probability table, connecting each node and its parent node directly.
- The example of Bayesian Belief Networks



(a) Three variables



(b) Five variable

Figure 4.14 Two Bayesian belief networks with two different numbers of variables.

Bayesian Network and Ensemble Methods

Algorithm: Use of Bayesian Belief Network for Predictive Analytics

- Input:
  - *d The number of variables*
  - *T General order of variables.*
- Output:
  - Bayesian belief network topology
- Procedure:
  1. Consider $T = (X_1, X_2, \cdots, X_d)$ as one general order of variables
  2. for I = 1 to d
  3. Make as $X_{T(i)} i^{th}$ highest variable in *T*
  4. Make $C(X_{T(i)})$ as the set of variables before $X_{T(i)}$
  5. Eliminate all the variables in $C(X_{T(i)})$ with no impact on X_i with future knowledge
  6. Draw an arc between remaining variables of $C(X_{T(i)})$ and $X_{T(i)}$
  7. end
  8. Output the drawn topological graph, namely Bayesian belief network topology.

Random Forests and Ensemble Methods

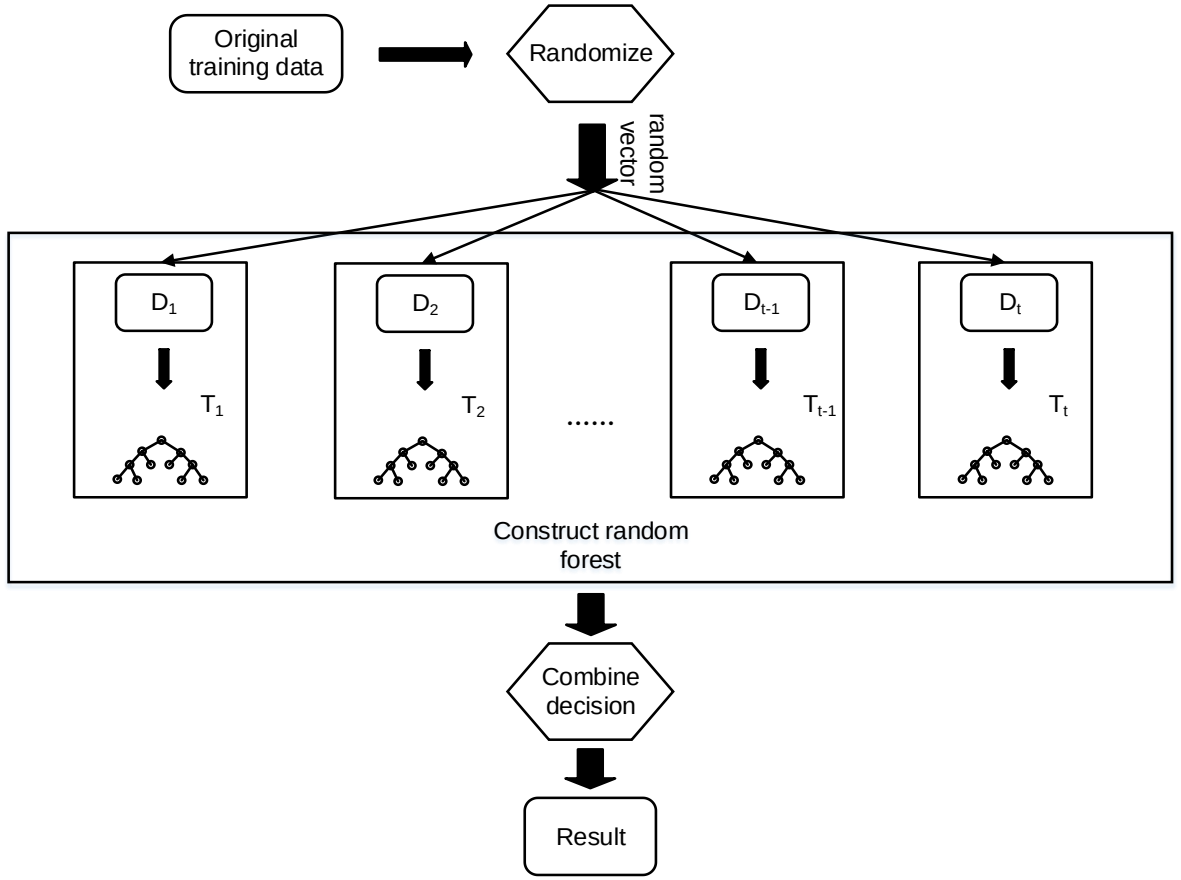


Figure 4.15 The process of using random forest for ensemble decision making

Bayesian Network and Ensemble Methods

Random Forests and Ensemble Methods

- the number of the original attributes, d, is too small, it is difficult to select a random independent set of attributes to construct the decision tree
  - using L input attributes of the linear combination to create a new attribute
  - using created new attributes to form a random vector
  - construct the multiple pieces of decision tree
- This random forest decision-making method is known as Forest-RC

Bayesian Network and Ensemble Methods

Algorithm : Use of Random Forests for Decision Making in Classification

- Input:
  - *R: forecast sample*
  - *L: attribute matrix*
- Output:
  - The decision results
- Procedure:
  1. Calculate the vector dimension F
  2. Create F-dimension random attribute vector to constitute the collection, C
  3. The decision tree is constructed according to the elements C, and a random forest is established
  4. Make decision in every decision tree
  5. Calculate and output the final results with the most votes
  6. End.