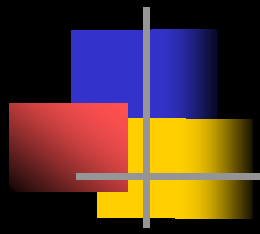
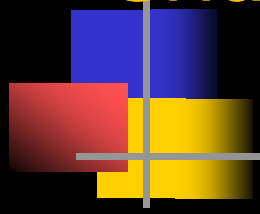


Chapter 7

Supervised machine learning for continuous data

Professor Jung Jin Lee
Soongsil University, Korea
New Uzbekistan University, Uzbekistan



Chapter 7 Supervised machine learning for continuous data

7.1 Bayes classification model

7.2 Logistic regression model

7.3 Nearest neighbor classification model

7.4 Neural network model

7.5 Support vector machine model

7.6 Ensemble model

7.1 Bayes classification model

Bayes classification rule by posterior probability

'If $P(G_1|X) \geq P(G_2|X)$, classify data as G_1, otherwise classify as G_2'

'If $\frac{P(X|G_1)}{P(X|G_2)} \geq \frac{P(G_2)}{P(G_1)}$, classify data as G_1, otherwise classify as G_2'

Bayes Classification - multiple groups

'If $P(G_k)f_k(\boldsymbol{x}) \geq P(G_i)f_i(\boldsymbol{x})$ for all $k \neq i$, classify $\boldsymbol{x}$ into group G_k'

7.1 Bayes classification model

[Example 7.1.1] (Bayes classification with one continuous variable)

- A survey of customers at a computer store showed the prior probabilities of the purchasing group (G_1) and the non-purchasing group (G_2) are P(G_1) = 0.4 and P(G_2) = 0.6, respectively.
- Suppose that the likelihood distribution of the age in the purchasing group is a normal distribution N(35,2^2), and the non-purchasing group is a normal distribution N(25,2^2).
- If a customer who visited this store on a certain day is 30 years old, classify the customer using the Bayes classification model whether he will purchase the product or not.

7.1 Bayes classification model

<Answer of Example 7.1.1>

- Functional form of likelihood probability distribution of each group.

$$P(x|G_1) = f_1(x) = \frac{1}{\sqrt{2\pi}\,2}\, exp\{-\frac{(x-35)^2}{2\times 2^2}\}$$

$$P(x|G_2) = f_2(x) = \frac{1}{\sqrt{2\pi}\,2}\, exp\{-\frac{(x-25)^2}{2\times 2^2}\}$$

- Bayes classification rule;

$$\text{If } \frac{f_1(x)}{f_2(x)} = exp\{-\frac{(x-35)^2}{2\times 2^2} - \frac{(x-25)^2}{2\times 2^2}\} \geq \frac{P(G_2)}{P(G_1)} = \frac{0.6}{0.4}, \text{ classify } x \text{ into } G_1, \text{ else } G_2.$$

- Classification rule;

  If x > 30.16, classify x int G_1, else G_2.

7.1 Bayes classification model

- Bayes classification with multivariate normal distribution
- Likelihood distribution of G_1 group is a multivariate normal distribution N(μ_1, Σ_1), and G_2 group is N(μ_2, Σ_2).
- Quadratic classification function

$$\text{If } d^Q(\boldsymbol{x}) = -\frac{1}{2} ln \frac{|\Sigma_2|}{|\Sigma_1|} - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)'\Sigma_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)'\Sigma_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) \geq ln \frac{P(G_2)}{P(G_1)}, \text{ classify } x \text{ into } G_1, \text{ else } G_2.$$

- If $\Sigma_1 = \Sigma_2 = \Sigma$, linear classification function

$$\text{If } d^L(\boldsymbol{x}) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)' \Sigma^{-1} [\, \boldsymbol{x} - \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2) \,] \geq ln \frac{P(G_2)}{P(G_1)}, \text{ classify } x \text{ into } G_1, \text{ else } G_2.$$

- Sample classification function

$$\text{If } d^L(\boldsymbol{x}) = (\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)' S^{-1} [\, \boldsymbol{x} - \frac{1}{2}(\bar{\boldsymbol{x}}_1 + \bar{\boldsymbol{x}}_2) \,] \geq ln \frac{P(G_2)}{P(G_1)}, \text{ classify } x \text{ into } G_1, \text{ else } G_2.$$

7.1 Bayes classification model

[Example 7.1.2] Consider a survey of 20 customers at a computer store on age, monthly income, and purchasing status. Assume that these continuous variables are multivariate normal distributions with the same covariance.

- Find a Bayes classification function and classify a customer who is 33 years old and has a monthly income of 200, whether he will purchase a computer or not.

Table 7.1.1 Survey of customers on age, income, and purchasing status

Number	Age	Income (unit 10,000 won)	Purchase
1	25	150	Yes
2	34	220	No
3	27	210	No
4	28	250	Yes
5	21	100	No
6	31	220	No
7	36	300	Yes
8	20	100	No
9	29	220	No
10	32	250	Yes
11	37	400	Yes
12	24	120	No
13	33	350	No
14	30	180	Yes
15	38	350	Yes
16	32	250	No

7.1 Bayes classification model

<Answer of Example 7.1.2>

- Sample mean and covariance matrix;

$$\bar{\boldsymbol{x}}_1 = \begin{bmatrix} 27.250 \\ 200.000 \end{bmatrix}, \quad \bar{\boldsymbol{x}}_2 = \begin{bmatrix} 33.125 \\ 291.250 \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} 31.621 & 470.105 \\ 470.105 & 9129.211 \end{bmatrix}$$

$$\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2 = \begin{bmatrix} -5.875 \\ -91.25 \end{bmatrix}, \quad \boldsymbol{S}^{-1} = \begin{bmatrix} 0.134895 & -0.006946 \\ -0.006946 & 0.000467 \end{bmatrix}, \quad (\bar{\boldsymbol{x}}_1 - \bar{\boldsymbol{x}}_2)'\boldsymbol{S}^{-1} = \begin{bmatrix} -0.15865 \\ -0.00182 \end{bmatrix}$$

- Sample linear classification function;

$$\text{If } (-0.15865) \times x_1 + (-0.00182) \times x_2 + 5.64297 \geq 0, \text{ classify } \boldsymbol{x} = (x_1, x_2) \text{ into } G_1, \text{ or } G_2.$$

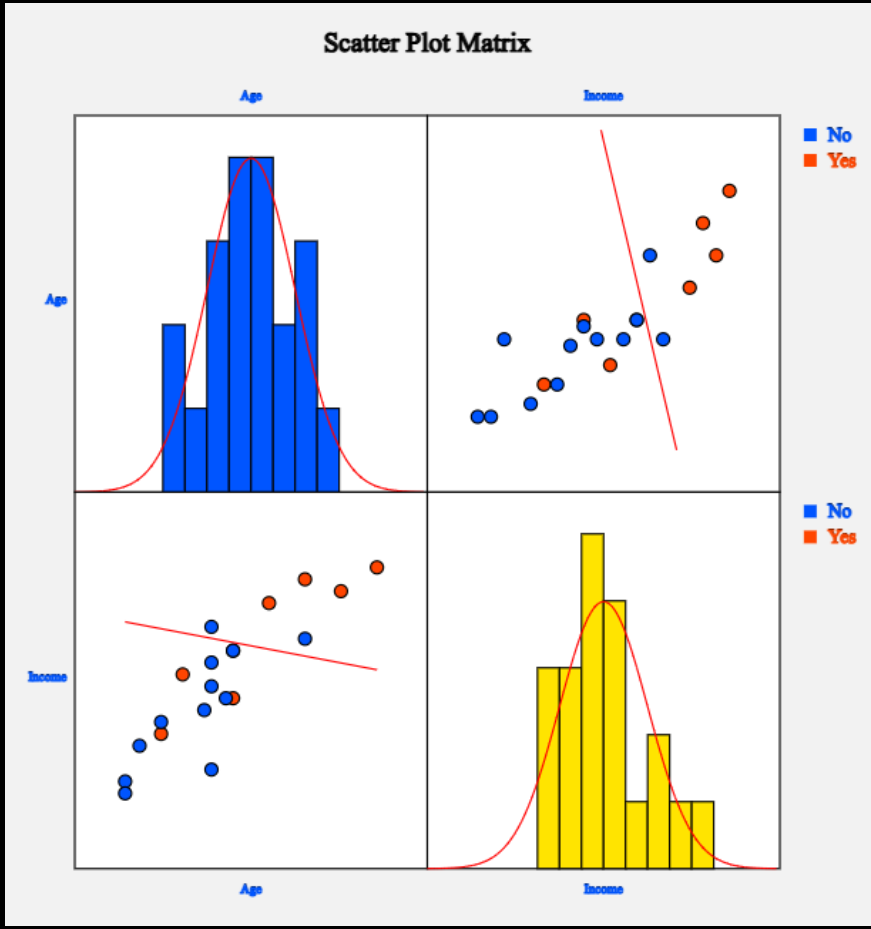
- If customer's age is 33 and income is 200, classification function;

  (−0.15865)∗(33)+(0.00182)∗(200)+(5.64297)=0.04251

  The customer is classified into the non-purchasing group.

7.1 Bayes classification model

<Answer of Example 7.1.2>



Training Data Classification Cell % Row %	Decision Purchase : No	Decision Purchase : Yes	Total
Purchase : No	10 50.00 % 83.33 %	2 10.00 % 16.67 %	12 60.00 % 100.00 %
Purchase : Yes	4 20.00 % 50.00 %	4 20.00 % 50.00 %	8 40.00 % 100.00 %
Total	14 70.00 %	6 30.00 %	20 100.00 %
Accuracy	70.00%	Misclassification Rate	30.00%

7.1 Bayes classification model

❖ **Variable selection**

- When there are many variables, selecting only the variables that increase accuracy.
- Stepwise classification analysis is selecting appropriate variables stepwise and classifying them.
- Select variables that can best explain group variables with high discriminatory power, e.g., high F values in ANOVA.
- Forward selection adds variables with high discriminatory power one by one.
- Backward elimination selects all variables and then removes variables with low discriminatory power.

7.1 Bayes classification model

❖ **Characteristics of Bayes classification**

- The risk of model overfitting is low and robust.
- Perform stable classification even when incomplete data, outliers, and missing values exist.

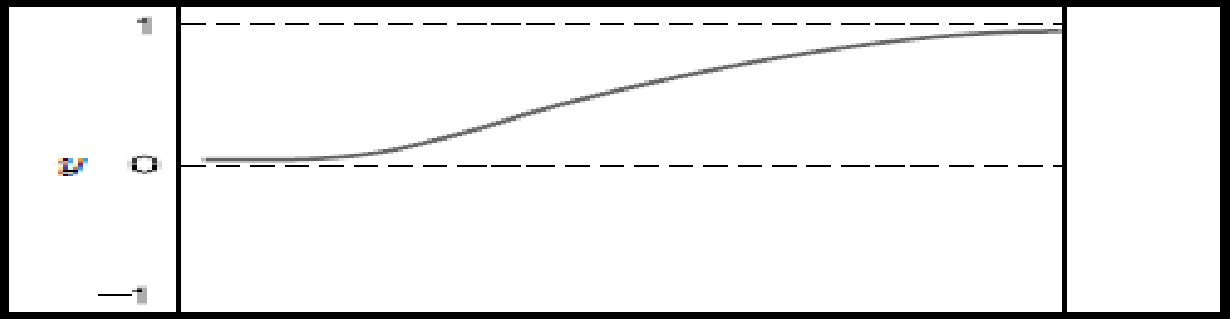
7.2 Logistic regression model

- Linear Regression Model
  - $Y = \alpha + \beta X + \varepsilon$
  - Explain Y (response variable) using independent variable X
  - Apply the regression model to classification analysis
    => predicting Y with groups 0 or 1 with X
    => predicted value of Y can be below 0 or above 1

- Logistic Regression Model
  - Regression model for the log odds ratio of Y probability in each X
  - $\text{Log}(P(Y=1) / P(Y=0)) = \alpha + \beta X$

7.2 Logistic regression model

- Logistic Regression Model
  - odds ratio(P(Y=1) / P(Y=0)) can have values in (0, ∞)
    => Log(odds ratio) can have values (-∞, ∞)
    => regression model is possible
  - Log(P(Y=1) / P(Y=0)) = α+ βX
    => Log(P(Y=1) / (1 - P(Y=1))) = α+ βX
    => P(Y=1) = exp(α+ βX) / (1 + exp(α+ βX))
    => P(Y = 1) : posterior probability



7.2 Logistic regression model

- Logistic multiple regression model

$$\log \frac{P(Y=1|x_1,x_2,\cdots,x_m)}{1-P(Y=1|x_1,x_2,\cdots,x_m)} = \beta_0+\beta_1 x_1+\cdots+\beta_m x_m$$

$$P(Y=1|x_1,\cdots,x_m) = \frac{\exp(\beta_0+\beta_1 x_1+\cdots+\beta_m x_m)}{1+\exp(\beta_0+\beta_1 x_1+\cdots+\beta_m x_m)}$$

- If the posterior probability is close to 1, classify Group 1, else Group 0.

7.2 Logistic regression model

- Incremental odds ratio
  - If X_i increases 1 unit and all other variables are constants,

$$\text{Incremental odds ratio} = \frac{exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_i(x_i + 1) + \cdots + \beta_m x_m)}{exp(\beta_0 + \beta_1 x_1 + \cdots + \beta_i x_i + \cdots + \beta_m x_m)} = exp(\beta_i)$$

  - If X_i increases 1 unit,
    exp(βi) > 1 if βi > 0 and exp(βi) < 1 if βi < 0
  - When X_i increases 1 unit, if the incremental odds ratio is 2, the probability of group 1 classification becomes twice.

7.2 Logistic regression model

- Variable selection
  - Akaike Information Criteria (AIC)
  - Forward selection method
  - Backward elimination method
  - Stepwise method

7.2 Logistic regression model

[Example 7.2.1] Using the survey data in Example 7.1.2, find a logistic regression model with product purchase as the target variable and age, monthly income, as independent variables.

Table 7.1.1 Survey of customers on age, income, and purchasing status

Number	Age	Income (unit 10,000 won)	Purchase
1	25	150	Yes
2	34	220	No
3	27	210	No
4	28	250	Yes
5	21	100	No
6	31	220	No
7	36	300	Yes
8	20	100	No
9	29	220	No
10	32	250	Yes
11	37	400	Yes
12	24	120	No
13	33	350	No
14	30	180	Yes
15	38	350	Yes
16	32	250	No
17	28	240	No
18	22	220	No
19	39	450	Yes
20	26	150	No

7.2 Logistic regression model

<Answer of Example 7.2.1>

R output;

Coefficients:		
(Intercept)	Age	Income
-7.629959	0.223517	0.001918

It implies;

$$ln\ \frac{P(Y=1 \mid \boldsymbol{X}=(X_1,X_2))}{1-P(Y=1 \mid \boldsymbol{X}=(X_1,X_2))} = -7.629959 + 0.223517X_1 + 0.001918X_2$$

The posterior probability of the customer who is 20 years old and has an income of 200 is as follows. Therefore, he is classified into group 0.

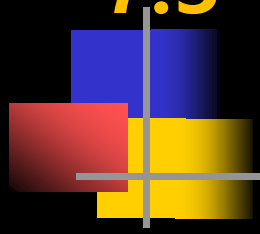
$$\begin{aligned} P(Y=1 \mid \boldsymbol{X}=(20,200)) &= \frac{exp(-7.629959 + 0.223517 \times 20 + 0.001918 \times 200)}{1 + exp(-7.629959 + 0.223517 \times 20 + 0.001918 \times 200)} \\ &= \frac{0.062286}{1 + 0.062286} \\ &= 0.058634 \end{aligned}$$

7.2 Logistic regression model

❖ Characteristics of logistic regression

- Logistic regression is a model for log odds ratio of group 1 and 0.
- P(Y=1) can be used for classification.
- Select variables that have much discriminatory power in case of many independent variables.

7.3 Nearest neighbor classification model

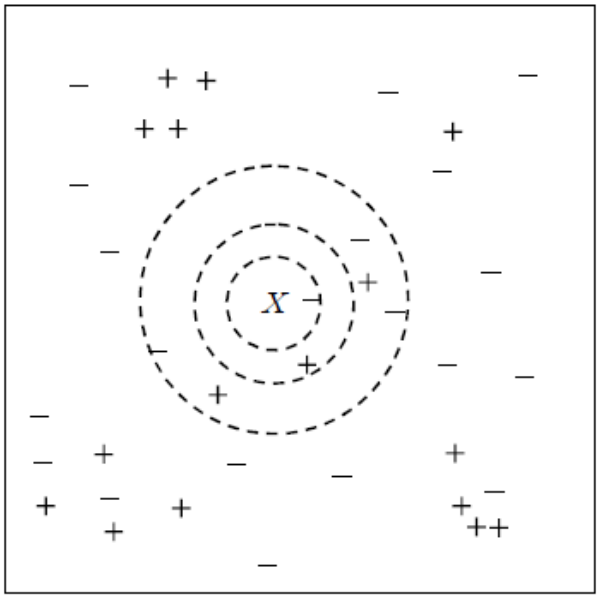


- The nearest neighbor classification model establishes a model when there is data to be classified, a lazy learner.
- K-nearest neighbor classification stores all training data in the computer.
- When data is to be classified, it finds a set of k data most similar to the variable values of the data and classifies the data into a group with a majority vote in that set.
- The similarity between the data and the training data uses various proximity measures.

7.3 Nearest neighbor classification model

[Algorithm for the K-nearest neighbor classification}

Step 1	Let $\boldsymbol{x}$ be the test data, and $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ be the training data.
Step 2	**for** test data $\boldsymbol{x}$ **do**
Step 3	**for** i = 1 to n **do**
Step 4	Calculate the distance $d(\boldsymbol{x}, \boldsymbol{x}_i)$ between $\boldsymbol{x}$ and $\boldsymbol{x}_i$
Step 5	**end for**
Step 6	Find the training data set $D_{\boldsymbol{x}}$ that is the k nearest neighbor of $\boldsymbol{x}$
Step 7	Classify $\boldsymbol{x}$ into the majority group of $D_{\boldsymbol{x}}$, that is $y = argmax_v \sum_{(\boldsymbol{x}_i, y_i) \in D_{\boldsymbol{x}}} I(v = y_i)$
Step 8	**end for**



7.3 Nearest neighbor classification model

[Example 7.3.1] Using the survey data in Example 7.1.2, classify a customer whose age is 33 years old and has a monthly income of 190, and determine whether he will buy a product or not, using the 5-nearest neighbor classification model.

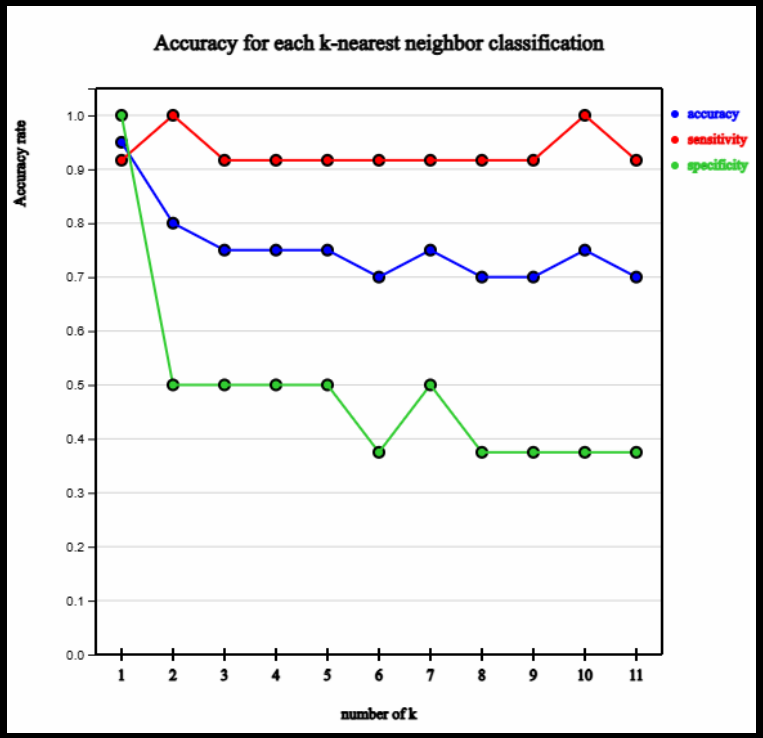
Table 7.3.1 Standardized data of age and income, and squared Euclid distance of the customer

Number	Age	Income (unit 10,000 won)	Purchase	Standardized Age	Standardized Income	Squared Euclid Distance of customer
1	25	150	Yes	-0.818	-0.905	2.199
2	34	220	No	0.782	-0.173	0.130
3	27	210	No	-0.462	-0.277	1.182
4	28	250	Yes	-0.285	0.141	1.185
5	21	100	No	-1.529	-1.429	5.441
6	31	220	No	0.249	-0.173	0.225
7	36	300	Yes	1.138	0.665	1.610
8	20	100	No	-1.707	-1.429	6.232
9	29	220	No	-0.107	-0.173	0.605
10	32	250	Yes	0.427	0.141	0.426
11	37	400	Yes	1.316	1.711	5.337
12	24	120	No	-0.996	-1.219	3.098
13	33	350	No	0.605	1.188	2.804
14	30	180	Yes	0.071	-0.591	0.296
15	38	350	Yes	1.494	1.188	3.595
16	32	250	No	0.427	0.141	0.426
17	28	240	No	-0.285	0.037	1.064
18	22	220	No	-1.352	-0.173	3.925
19	39	450	Yes	1.672	2.235	8.543
20	26	150	No	-0.640	-0.905	1.725

7.3 Nearest neighbor classification model

❖ Selection of k on nearest neighbor classification

- We can search for a k value that shows better accuracy, sensitivity, or specificity in the nearest neighbor classification.
- k is selected when there is no significant increase in accuracy..



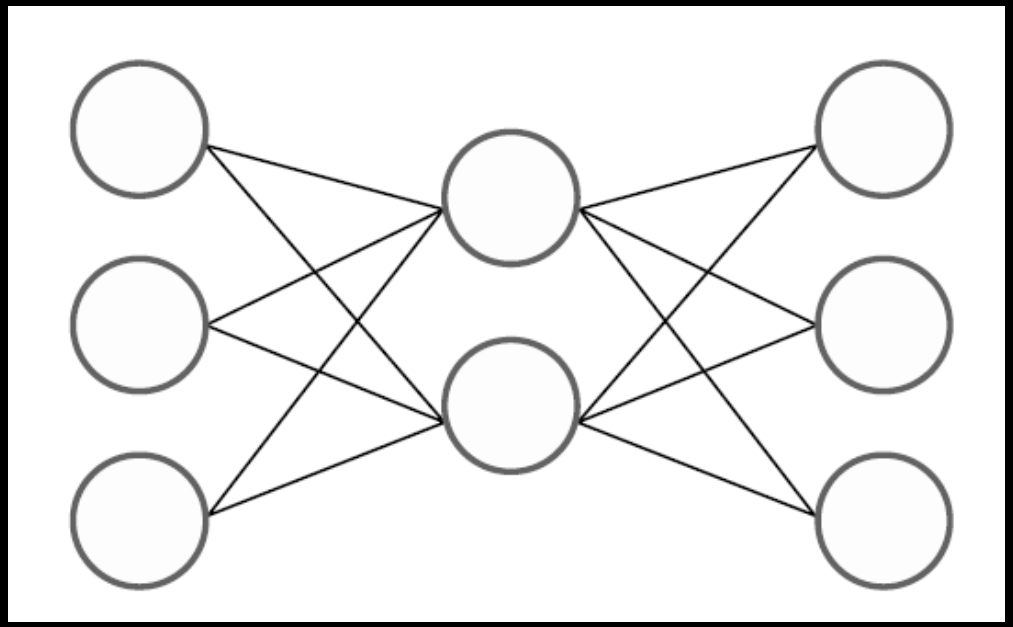
7.3 Nearest neighbor classification model

❖ Characteristics of nearest neighbor classification

- The nearest neighbor classification method only requires measuring the similarity between data and training data.
- If the training data increases, calculating the similarity measure takes a lot of time and effort
- If k is small and much noise in the data, the classification may not be accurate.
- The decision boundary is not a function
  => more flexible than other models.
  => too dependent on the training data
  => cause stability problem for classification.

7.4 Artificial neural network

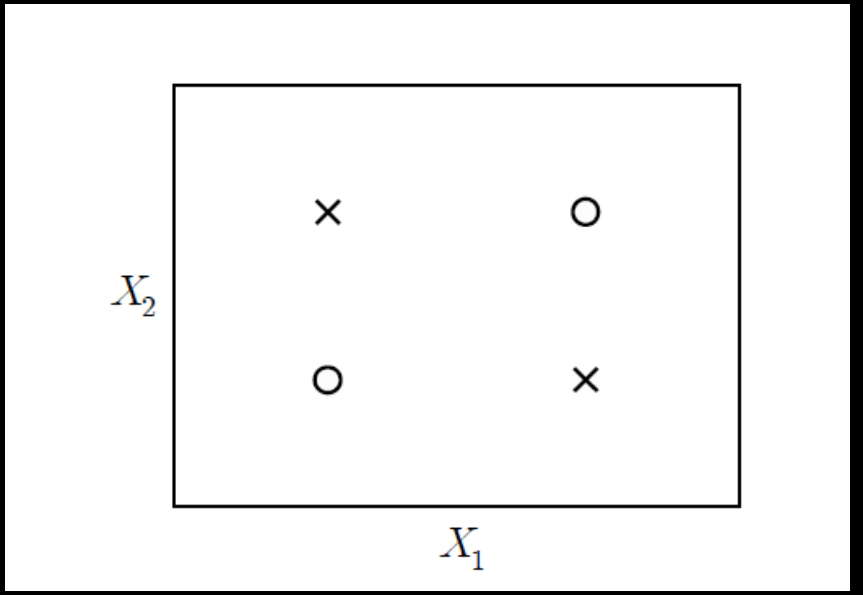
- The artificial neural network model imitates the way the human brain makes decisions and classifies them.
- Neural network connects multiple nodes into a network



7.4 Artificial neural network

- The artificial neural network model uses a generalized nonlinear function as a classification function.

- The data cannot be separated into two groups o and x by a single straight line and can only be separated by two straight lines or nonlinear functions.



7.4 Artificial neural network

❖ Single-layer neural network

[Example 7.4.1] Suppose y is a group variable where there are two groups, denoted '+1' and '-1', and there are three binary variables X_1, X_2, X_3 which have values either 0 or 1. If two or more of the three binary variables have the value 1, classify them as the group '+1'; if they have one or fewer 1 value, classify them as the group '-1'. and classify this data.

- Create a single-layer neural network model that can perform such classification and classify this data.

Table 7.4.1 Possible values of three binary variables x_1, x_2, x_3 and their group y

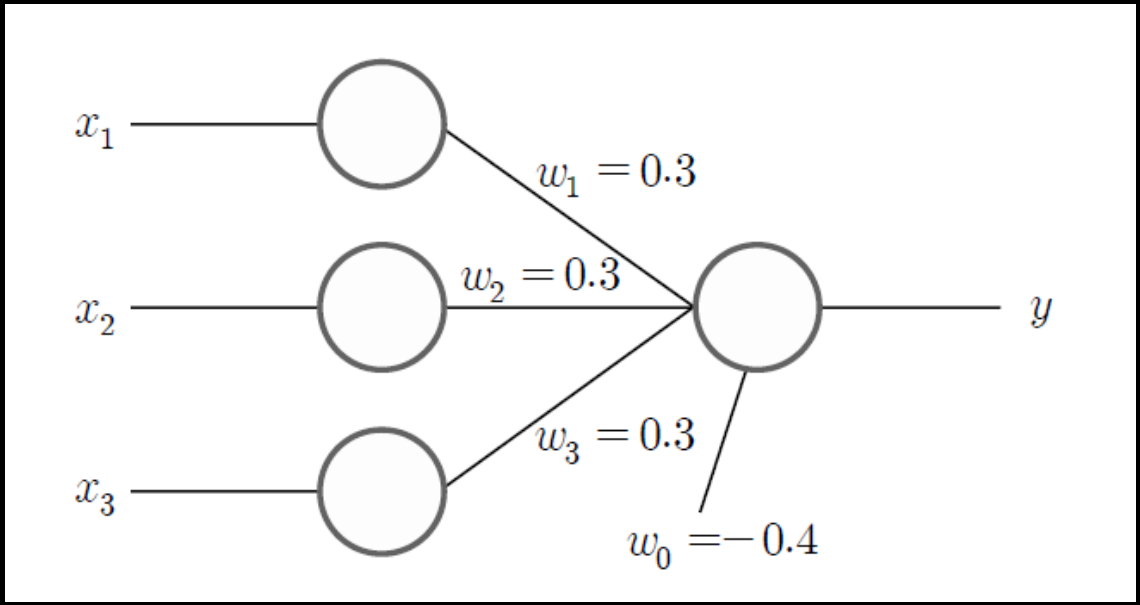
Number	x_1	x_2	x_3	y
1	0	0	0	-1
2	0	0	1	-1
3	0	1	0	-1
4	0	1	1	+1
5	1	0	0	-1
6	1	0	1	+1
7	1	1	0	+1
8	1	1	1	+1

7.4 Artificial neural network

❖ Single-layer neural network

<Answer of Example 7.4.1>

$$\hat{y} = \begin{cases} +1 & if \;\; 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0 \\ -1 & if \;\; 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0 \end{cases}$$



7.4 Artificial neural network

❖ Single-layer neural network

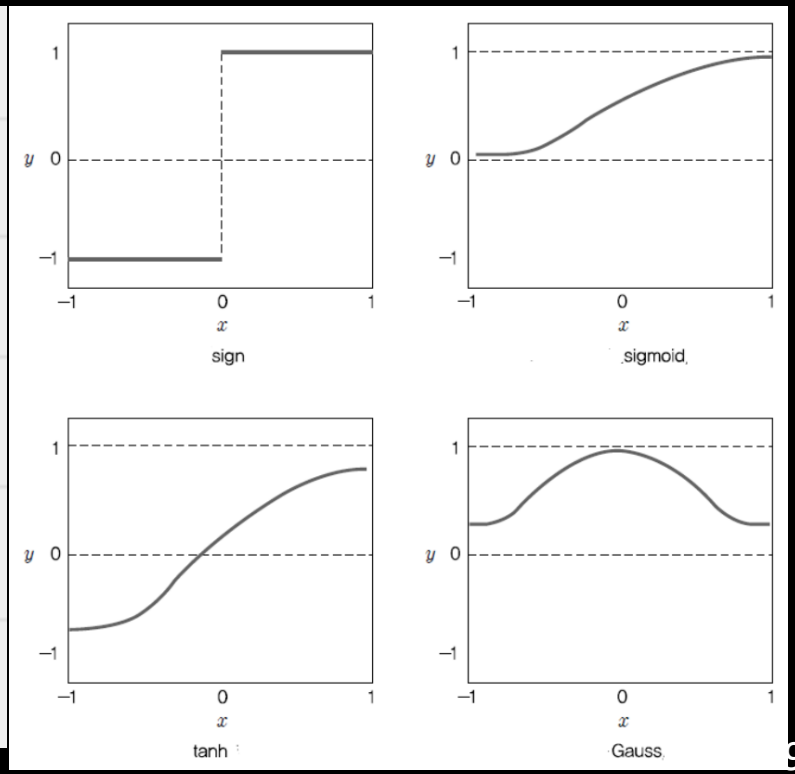
- $\hat{y} = \text{sign}\,(\,w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n\,)$

  Combination function; $(\,w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n\,)$

  Activation function; $\text{sign}\,(\,x\,)$

Table 7.4.2 Examples of activation functions

Name	Activation fuction	Range
sign function	$y = sign(x)$	-1, +1
sigmoid fuction	$y = \frac{1}{1+e^{-x}}$	(0, 1)
tanh function	$y = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	(-1, 1)
Gauss function	$y = e^{-\frac{x^2}{2}}$	(0, 1)



7.4 Artificial neural network

❖ Single-layer neural network

[Learning algorithm for the single-layer neural network]

Step 1	Let $D = \{(x_{i1}, x_{i2}, \ldots, x_{im}, y_i),\ i = 1, 2, \ldots, n\}$ be the training data
Step 2	$w_1^{(0)}, w_2^{(0)}, \ldots, w_m^{(0)}$ be the initial estimated value of the coefficients and λ is the learning rate
Step 3	**for** i = 1 to n **do**
Step 4	**for** j = 1 to m **do**
Step 5	Estimate $y_i^{(i)}$ using $w_1^{(i-1)}, w_2^{(i-1)}, \ldots, w_m^{(i-1)}$
Step 6	$w_j^{(i)} = w_j^{(i-1)} + \lambda(y_i - y_i^{(i)})x_{ij}$
Step 7	**end for**
Step 8	**end for**

7.4 Artificial neural network

❖ Single-layer neural network

[Example 7.4.2] For the single-layer neural network of Example 7.4.1, train the neural network with the initial values; w_0 = - 0.4, w_1 = 0.2, w_2 =0.1, w_3 = 0.1 and the learning rate λ = 0.1.

<Answer>

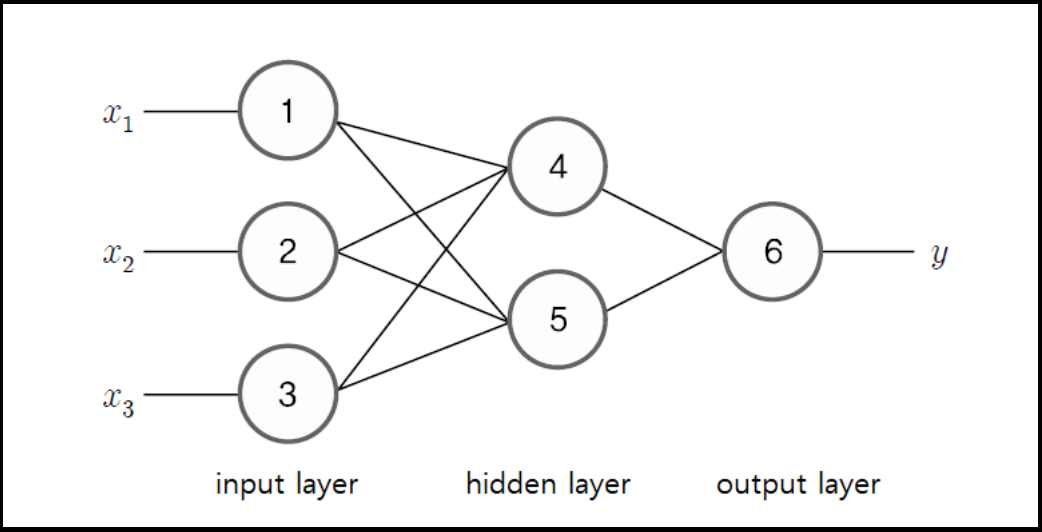
Table 7.4.3 Application of learning algorithm to the sigle-layer neural network

iteration	data				linear combination function	activation function	modified coefficients		
i	x_{i1}	x_{i2}	x_{i3}	y_i	$w^{(i)} = w_0 + w_1^{(i-1)}x_1 + w_2^{(i-1)}x_2 + w_3^{(i-1)}x_3$	$\hat{y}_i = sign(w^{(i)})$	$w_1^{(i)}$	$w_2^{(i)}$	$w_3^{(i)}$
1	0	0	0	-1	-0.4	-1	0.2	0.1	0.1
2	0	0	1	-1	-0.3	-1	0.2	0.1	0.1
3	0	1	0	-1	-0.3	-1	0.2	0.1	0.1
4	0	1	1	+1	-0.2	-1	0.2	0.3	0.3
5	1	0	0	-1	-0.2	-1	0.2	0.3	0.3
6	1	0	1	+1	0.1	+1	0.2	0.3	0.3
7	1	1	0	+1	0.1	+1	0.2	0.3	0.3
8	1	1	1	+1	0.4	+1	0.2	0.3	0.3

7.4 Artificial neural network

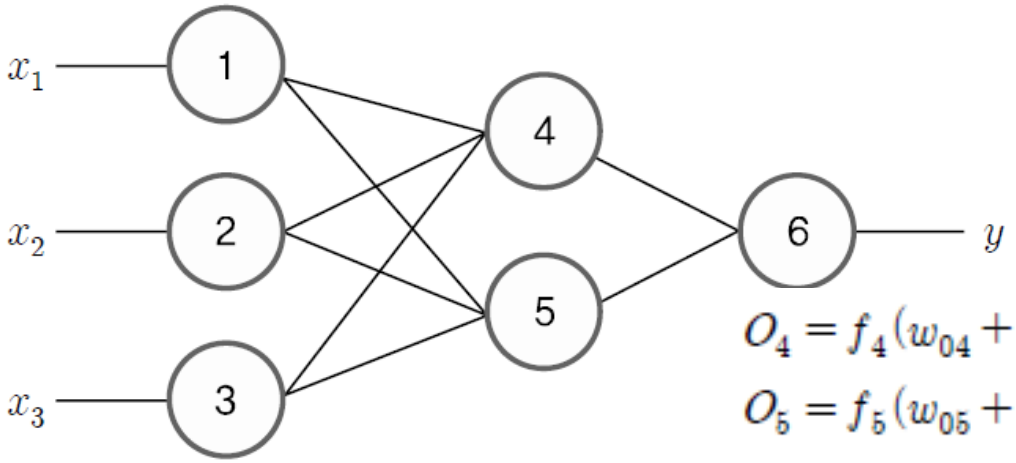
❖ Multilayer neural network

- A multilayer neural network consists of an input layer consisting of input nodes, a hidden layer that is a set of intermediate nodes that synthesize the nodes of the input layer, and an output layer that synthesizes the nodes of the hidden layer.



7.4 Artificial neural network

❖ Multilayer neural network



$$O_4 = f_4(w_{04} + w_{14}x_1 + w_{24}x_2 + w_{34}x_3)$$

$$O_5 = f_5(w_{05} + w_{15}x_1 + w_{25}x_2 + w_{35}x_3)$$

$$\hat{y} = f_6(w_{06} + w_{46}O_4 + w_{56}O_5)$$

$$\hat{y} = f_6(w_{46}f_4(w_{14}x_1 + w_{24}x_2 + w_{34}x_3 + w_{04}) + w_{56}f_5(w_{15}x_1 + w_{25}x_2 + w_{35}x_3 + w_{05}) + w_{06})$$

7.4 Artificial neural network

❖ Multilayer neural network

- How many hidden layers should there be?
- How many nodes should each hidden layer have?
- Is there a nonlinear function represented by these hidden layers and hidden nodes?

- Theorem 7.4.1 Approximation of a continuous function
  When a continuous function f(x) is defined on $[0,1]^m$, this function can be expressed as follows.

$$f(\boldsymbol{x}) = \sum_{k=1}^{2m+1} \Theta_k \left[\sum_{j=1}^{m} \phi_{jk}(x_j) \right]$$

7.4 Artificial neural network

❖ Design of multilayer neural network

1) Data preparation

- The variable values are usually converted to be between 0 and 1.

2) Number of input nodes

- If the variable is binomial or continuous data, assign one input node to each variable.
- If the variable is categorical, assign one input node to each categorical value.

3) Number of output nodes

- If there are two groups, one output node is sufficient.
- If there are K groups, assign K output nodes.

7.4 Artificial neural network

❖ Design of multilayer neural network

4) Number of hidden layers and number of hidden nodes

- A problem of determining the nonlinear function of the neural network model.
- If the number of hidden layers and hidden nodes increases, the model may be overfitted.
- => A small number of hidden layers and hidden nodes.
- Usually, after setting the number of hidden layers and hidden nodes sufficiently, we reduce them one by one and select a model with high accuracy.
- A model selection criteria such as AIC (Akaike information criteria) can be used.

7.4 Artificial neural network

❖ Design of multilayer neural network

5) Selection of activation function

- The sigmoid function, which is useful for the estimation algorithm of weight coefficients, is often used.
- The activation function is known to affect the algorithm speed during the training process of a neural network but does not have a significant effect on the results.

6) Initial value problem

- Generate initial values randomly between -1 and 1.
- Since there is a possibility that a given initial value will find a local solution, experiment several times to find the same weight coefficients by trying various initial values.

7.4 Artificial neural network

❖ Design of multilayer neural network

7) Interpretation of output variables

- If there are two groups and one output node, it can be classified based on an appropriate boundary value.
- If there are multiple groups, the number of output nodes is usually the same as the number of groups, and the group is classified based on the value of the output node that is large (or small).

8) Sensitivity analysis

- Change the value of the input variable from the minimum to the maximum and examine the change in the output value.

7.4 Artificial neural network

❖ Learning of multilayer neural network

- Find the weight coefficients that minimize the error sum of squares.

$$E(w) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$\frac{\partial E(w)}{\partial w_{jk}} = -2 \sum_{i=1}^{n} (y_i - \hat{y}_i) \frac{\partial \hat{y}_i}{\partial w_{jk}}$$

$$w_{jk} \leftarrow w_{jk} - \lambda \frac{\partial E(w)}{\partial w_{jk}}$$

$$w_{jk} \leftarrow w_{jk} - \lambda E_k O_j$$

7.4 Artificial neural network

❖ Learning of multilayer neural network

[Example 7.4.3] (Learning algorithm of the multilayer neural network) For the multilayer neural network model in Figure 7.4.5, let the input data be group 1 and the variable values be (x_1, x_2, x_3) = (1, 0, 1). Let us find the weight coefficient and bias of the model equation using the back-propagation algorithm of the gradient descent method.

The same sigmoid function, f(x), is used for all activation functions, and the initial values of the weight coefficient and bias are set as follows using a random number between (-1,1). Let the learning rate be λ = 0.1.

Table 7.4.4 Initial values of the weight coefficients for the multilayer neural network in Figure 7.4.5

w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	w_{04}	w_{05}	w_{06}
-0.51	-0.99	0.35	-0.45	0.39	0.19	0.27	0.71	-0.75	-0.09	0.18

7.4 Artificial neural network

❖ Learning of multilayer neural network

<Answer of Example 7.4.3>

- In the neural network of Figure 7.4.5, the output values O_1, O_2, O_3 of nodes ①, ②, and ③ are the values of the input variables x_1, x_2, x_3.
- The output values of nodes ④, ⑤, and ⑥ are as follows, using the given initial weight coefficients.

$$
\begin{aligned}
O_4 &= f(w_{14}x_1 + w_{24}x_2 + w_{34}x_3 + w_{04}) \\
&= f(-0.51 \times 1 + 0.35 \times 0 + 0.39 \times 1 - 0.75 \\
&= f(-0.87) = 0.2953 \\
O_5 &= f(w_{15}x_1 + w_{25}x_2 + w_{35}x_3 + w_{05}) \\
&= f(-0.99 \times 1 - 0.45 \times 0 + 0.19 \times 1 - 0.09 \\
&= f(-0.89) = 0.2911 \\
O_6 &= f(w_{46}O_4 + w_{56}O_6 + w_{06}) \\
&= f(0.27 \times 0.2953 + 0.71 \times 0.2911 + 0.18 \\
&= f(0.4664) = 0.6145
\end{aligned}
$$

7.4 Artificial neural network

❖ Learning of multilayer neural network

<Answer of Example 7.4.3>

- The backward step of the back-propagation algorithm estimates the error E_6 of node ⑥ and then estimates the errors of nodes ④ and ⑤.

$$\begin{aligned} E_6 &= O_6(1-O_6)(y_i - O_6) \\ &= 0.6145 \times (1-0.6145) \times (1-0.6145) = 0.0913 \end{aligned}$$

- O_6 (1- O_6) term is the rate of change from the differentiation of the sigmoid function, and y is the actual group value. The meaning of the error E_6 is the estimation error, y_i - O_6, multiplied by the error change rate O_6 (1- O_6).

7.4 Artificial neural network

❖ Learning of multilayer neural network

<Answer of Example 7.4.3>

- The error E_5 of hidden node ⑤ is calculated by multiplying the error change rate O_5 (1- O_5) by the error weighted sum of all nodes connected to node ⑤, which is called back-propagation of the error.

$$E_5 = O_5(1-O_5)\sum_k w_{5k}E_k$$

- In this problem, since there is only node ⑥ connected to node ⑤, the errors E_5 and E_4 are as follows.

$$\begin{aligned} E_5 &= O_5(1-O_5)w_{56}E_6) \\ &= 0.2911 \times (1-0.2911) \times 0.71 \times 0.0913 = 0.0134 \\ E_4 &= O_4(1-O_4)w_{46}E_6) \\ &= 0.2953 \times (1-0.2953) \times 0.27 \times 0.0913 = 0.0051 \end{aligned}$$

7.4 Artificial neural network

❖ Learning of multilayer neural network

<Answer of Example 7.4.3>

$$
\begin{aligned}
w_{46} &\leftarrow w_{46} + \lambda E_6 O_4 = 0.27 + 0.1 \times 0.0913 \times 0.2953 = 0.2727 \\
w_{56} &\leftarrow w_{56} + \lambda E_6 O_5 = 0.71 + 0.1 \times 0.0913 \times 0.2911 = 0.7127 \\
w_{14} &\leftarrow w_{14} + \lambda E_4 x_1 = -0.51 + 0.1 \times 0.0051 \times 1 = -0.5095 \\
w_{15} &\leftarrow w_{15} + \lambda E_5 x_1 = -0.99 + 0.1 \times 0.0134 \times 1 = -0.9887 \\
w_{24} &\leftarrow w_{24} + \lambda E_4 x_2 = +0.35 + 0.1 \times 0.0051 \times 0 = 0.3500 \\
w_{25} &\leftarrow w_{25} + \lambda E_5 x_2 = -0.99 + 0.1 \times 0.0134 \times 0 = -0.4500 \\
w_{34} &\leftarrow w_{34} + \lambda E_4 x_3 = +0.39 + 0.1 \times 0.0051 \times 1 = 0.3905 \\
w_{35} &\leftarrow w_{35} + \lambda E_5 x_3 = +0.19 + 0.1 \times 0.0134 \times 1 = 0.1913 \\
w_{04} &\leftarrow w_{04} + \lambda E_4 = -0.75 + 0.1 \times 0.0051 = -0.7495 \\
w_{05} &\leftarrow w_{06} + \lambda E_5 = -0.09 + 0.1 \times 0.0134 = -0.0887 \\
w_{06} &\leftarrow w_{06} + \lambda E_6 = +0.18 + 0.1 \times 0.0913 = 0.1891
\end{aligned}
$$

7.4 Artificial neural network

❖ Deep learning

- If there are many hidden layers in a multilayer neural network, the back-propagation algorithm often cannot find the weight coefficients successfully because of vanishing gradients where data disappears and learning does not proceed well.
- In 2006, Professor Geoffrey Hinton of the University of Toronto solved the vanishing gradient problem through the pretraining of neural networks and dropout data,
- The neural network model that applied this algorithm is called as deep learning.

7.4 Artificial neural network

❖ Characteristics of neural network model

1) Neural network models show somewhat better results than other models when the number of variables is large and the input and output variables have complex nonlinear function forms.
2) Since it is not easy to explain why the neural network model was classified that way for the classification results, this model is sometimes called a black box.
3) Multilayer neural networks require at least one hidden layer. Determining the appropriate number of hidden layers and nodes is crucial to avoid overfitting the model.

7.4 Artificial neural network

❖ Characteristics of neural network model

4) Neural networks do not show a sensitive response, even if the training data contains noise.

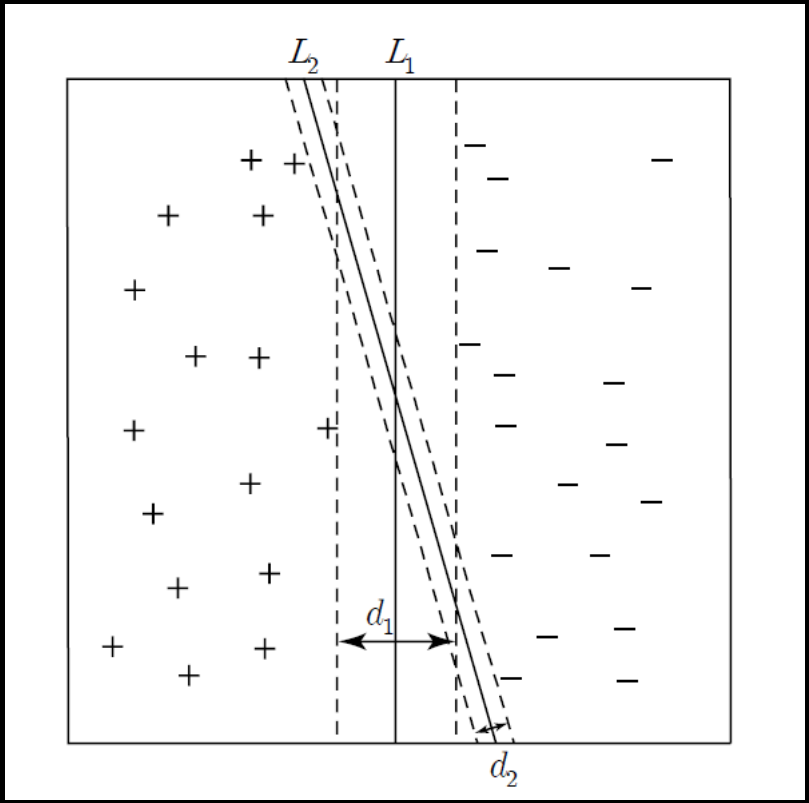
5) It is necessary to investigate whether it is a local solution by various methods, such as changing the initial value or analyzing the data's sensitivity.

6) The training process of a neural network is a very time-consuming when the number of hidden layers and nodes is large. However, after training, test data can be classified quickly.

7.5 Support vector machine

❖ Linear support vector machine (SVM)

- If two straight lines are used to classify the test data, the distance d_1 is larger than d_2, so L_1 has less possibility of misclassification.
- Linear SVM is a method to determine a linear classification function so that the distance, such as d_1 is maximized.



7.5 Support vector machine

❖ Linear support vector machine (SVM)

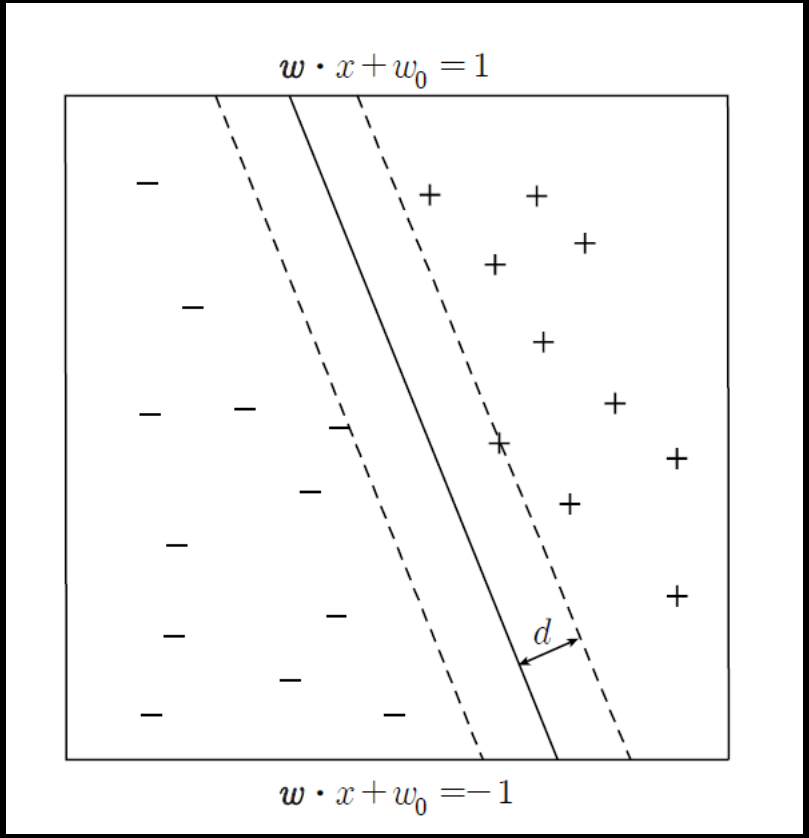
- Linear classification function

$$\text{if } \boldsymbol{w} \cdot \boldsymbol{x} + w_0 \geq 1, \quad \text{classify } \boldsymbol{x} \text{ into } y = 1$$
$$\text{if } \boldsymbol{w} \cdot \boldsymbol{x} + w_0 \leq -1, \quad \text{classify } \boldsymbol{x} \text{ into } y = -1$$

$$y_i (\boldsymbol{w} \cdot \boldsymbol{x}_i + w_0) \geq 1, \quad i = 1, 2, \ldots, n$$

- If x_1 lies in the hyperplane $w \cdot x_i + w_0 = 1$, $w \cdot x_1 + w_0 = 1$.
- If x_2 lies in the hyperplane $w \cdot x_i + w_0 = -1$, $w \cdot x_2 + w_0 = -1$.
- $w \cdot (x_1 - x_2) = 2$

$$\boldsymbol{w} \cdot (\boldsymbol{x}_1 - \boldsymbol{x}_2) = ||\boldsymbol{w}|| \, ||\boldsymbol{x}_1 - \boldsymbol{x}_2|| \cos\theta$$



7.5 Support vector machine

❖ Linear support vector machine (SVM)

- Shortest distance $\theta = 0$

$$||\boldsymbol{w}|| \times d = 2, \text{ that is } d = \frac{2}{||\boldsymbol{w}||}$$

- Linear SVM

Find $\boldsymbol{w}, w_0$ which minimizes $\frac{||\boldsymbol{w}||^2}{2^2}$

subject to

$y_i (\boldsymbol{w} \cdot \boldsymbol{x}_i + w_0) \geq 1, \quad i = 1, 2, \ldots, n$

If $\boldsymbol{w}^* \cdot \boldsymbol{x} + w_0^* \geq 0$, classify $\boldsymbol{x}$ into $'+1'$ group, else $'-1'$group.

7.5 Support vector machine

❖ Linear support vector machine (SVM)

[Example 7.5.1] When there are eight data for two variables (x_1, x_2)= (age, income) and group variable y (+1: purchase, -1: non-purchase) as in Table 7.5.1, find the classification equation using a linear support vector model.

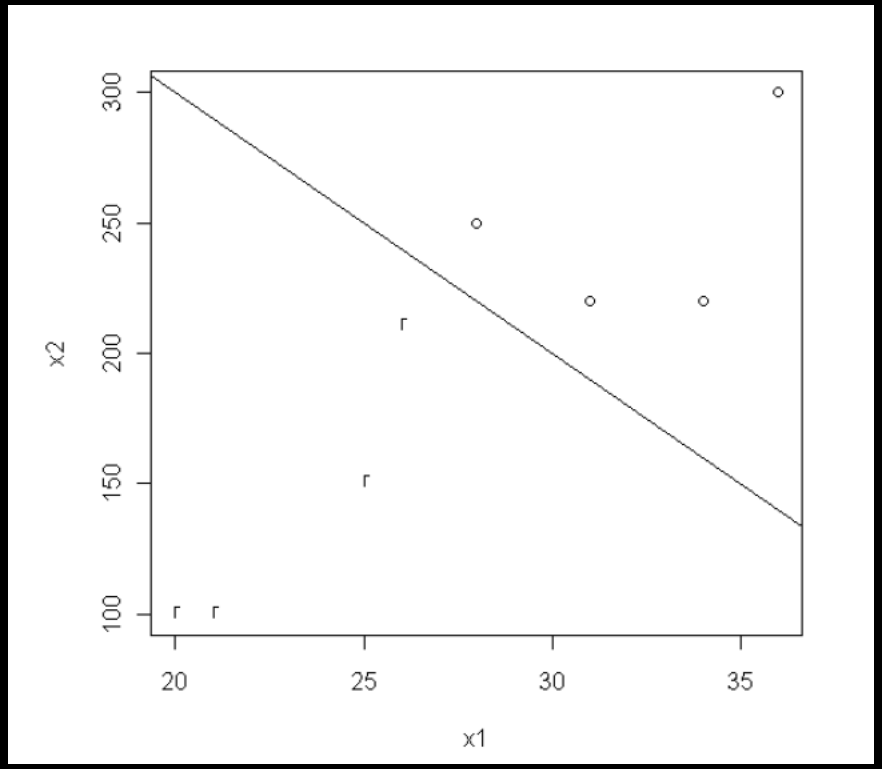
Table 7.5.1 Eight data with two variables and their group

number	Age x_1	Income x_2	Group y
1	25	150	-1
2	34	220	+1
3	26	210	-1
4	28	250	+1
5	21	100	-1
6	31	220	+1
7	36	300	+1
8	20	100	-1

7.5 Support vector machine

❖ Linear support vector machine (SVM)

<Answer of Example 7.5.1>



- If we draw a scatter plot for the data in Table 7.5.1, linear separation is possible.

- If we find the solution to the quadratic programming of the linear support vector, the classification function is

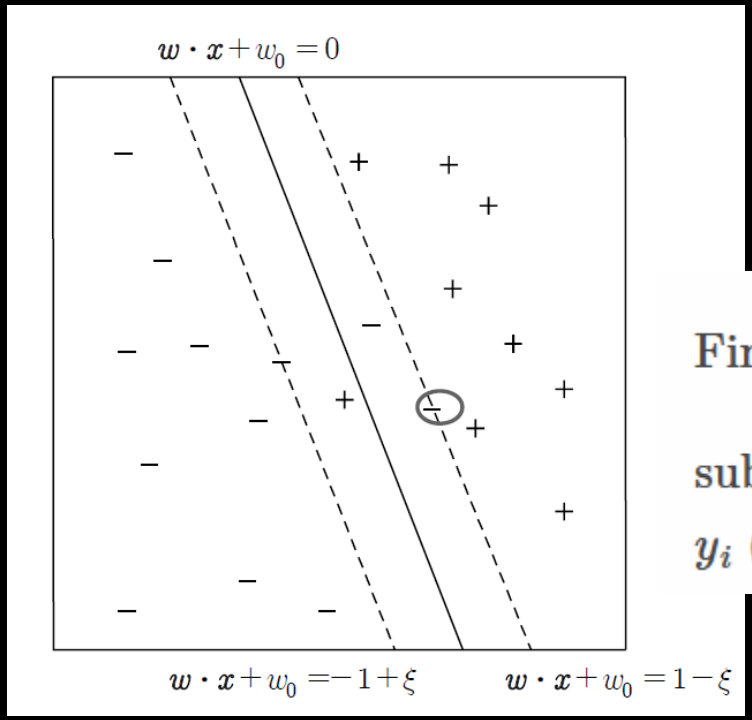
  $0.333\ x_1 + 0.033\ x_2 - 16.667 = 0$

- The decision rule is;

  If $0.333\ x_1 + 0.033\ x_2 - 16.667 \geq 0$, classify '+1' group, else '-1' group.

7.5 Support vector machine

❖ Linearly not separable case



$$if \ \boldsymbol{w} \cdot \boldsymbol{x} + w_0 \geq 1 - \xi, \quad y = 1$$
$$if \ \boldsymbol{w} \cdot \boldsymbol{x} + w_0 \leq -1 + \xi, \quad y = -1$$

Find $\boldsymbol{w}, w_0, \xi_i$ which minimize $\frac{||\boldsymbol{w}||^2}{2^2} + C(\sum_{i=1}^{n} \xi_i)^k$

subject to

$$y_i \ (\boldsymbol{w} \cdot \boldsymbol{x}_i + w_0) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n$$

7.5 Support vector machine

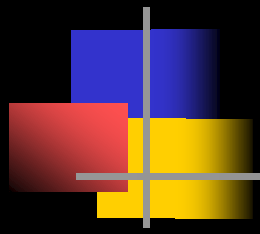
❖ Nonlinear support vector machine

Find $\boldsymbol{w}$ which minimize $\dfrac{||\boldsymbol{w}||^2}{2^2}$

subject to

$y_i\ (\boldsymbol{w} \cdot \Phi(\boldsymbol{x}_i)) \geq 1, \quad i = 1, 2, \ldots, n$

7.6 Ensemble model



- The ensemble model combines the results of multiple classification models to increase classification accuracy.
- A classification model using training data is a classifier.
- The ensemble model creates multiple classifiers from the training data and applies each classifier to classify unknown data.
- Determines the final group by a majority vote of the resulting groups

7.6 Ensemble model

- Suppose five classifiers classify two groups, and each has a misclassification rate of 5%.
- If the five classifiers are independent models, the ensemble model will misclassify if more than half of the classifiers are misclassified.
- The misclassification rate of the ensemble model is as follows.

$$e_{ensemble} = \sum_{i=3}^{5} \binom{5}{i} (0.05)^i (1-0.05)^{5-i} = 0.0001$$

7.6 Ensemble model

- Creating multiple classifiers;

A. Adjust the number of data
   - bagging
   - boosting

B. Control the number of variables
   - random forest

C. Control group names

D. Adjust classification model assumptions

7.6 Ensemble model

❖ Bagging

- Bagging (bootstrap aggregating) generates a classifier for each sampling with replacement repeatedly from the training data and then ensembles the results.
- The same data can be extracted multiple times in a sample, and some may not be extracted in all samples.
- When there are n data, if n samples are repeatedly extracted by sampling with replacement, the probability that each data will be extracted again is $1-(1-1/n)^n$ = $1-\frac{1}{e}$ = 0.632.

7.6 Ensemble model

❖ Bagging

[Bagging algorithm]

Step 1	Let R be the number of bootstrap samples, and n be the sample size
Step 2	for k = 1 to R do
Step 3	Generate bootstrap samples D_k of size n
Step 4	Create classifier C_k using bootstrap samples D_k
Step 5	end for
Step 6	Classify an unknown data $\boldsymbol{x}$ into the majority vote of all classifiers, that is, $$C^*(\boldsymbol{x}) = argmax_y \sum_{k=1}^{R} I(C_k(\boldsymbol{x}) = y)$$

7.6 Ensemble model

❖ Bagging

[Example 7.6.1] A survey of 10 people who visited a store showed monthly income x and purchasing status y (purchasers have a value of 1, and non-purchasers have a value of -1).

Table 7.6.1 Ten customer data with income x and purchase status y

x	100	120	160	180	186	190	210	250	270	300
y	1	1	1	-1	-1	-1	-1	1	1	1

- We want to use a simple decision tree classifier such that
  'If x ≤ c, classify x into purchaser group 1,
  otherwise classify into non-purchaser group -1'.
- It is called a decision stump, and c is determined to minimize entropy.
- Classify this data using the bagging method.

7.6 Ensemble model

❖ Bagging

<Answer of Example 7.6.1>

Sample 1	x	100	120	120	160	180	180	186	190	270	270	If $x \leq 170$, then y = 1, else y = -1
	y	1	1	1	1	-1	-1	-1	-1	1	1	
Sample 2	x	100	120	160	180	186	250	270	300	300	300	If $x \leq 300$, then y = 1, else y = -1
	y	1	1	1	-1	-1	1	1	1	1	1	
Sample 3	x	100	120	160	180	180	186	210	210	250	270	If $x \leq 170$, then y = 1, else y = -1
	y	1	1	1	-1	-1	-1	-1	-1	1	1	
Sample 4	x	100	100	120	180	180	186	186	210	250	270	If $x \leq 150$, then y = 1, else y = -1
	y	1	1	1	-1	-1	-1	-1	-1	1	1	
Sample 5	x	100	100	120	186	190	190	190	300	300	300	If $x \leq 153$, then y = 1, else y = -1
	y	1	1	1	-1	-1	-1	-1	1	1	1	
Sample 6	x	120	180	186	190	210	210	210	250	270	300	If $x \leq 230$, then y = -1, else y = 1
	y	1	-1	-1	-1	-1	-1	-1	1	1	1	
Sample 7	x	100	180	180	190	210	250	270	270	270	300	If $x \leq 230$, then y = -1, else y = 1
	y	1	-1	-1	-1	-1	1	1	1	1	1	
Sample 8	x	100	120	186	186	186	210	210	250	270	300	If $x \leq 230$, then y = -1, else y = 1
	y	1	1	-1	-1	-1	-1	-1	1	1	1	
Sample 9	x	100	160	180	180	190	210	210	250	300	300	If $x \leq 230$, then y = -1, else y = 1
	y	1	1	-1	-1	-1	-1	-1	1	1	1	
Sample 10	x	100	100	100	100	160	160	250	250	270	270	If $x \leq 50$, then y = -1, else y = 1
	y	1	1	1	1	1	1	1	1	1	1	

7.6 Ensemble model

❖ Bagging

<Answer of Example 7.6.1>

Table 7.6.3 Classification results of each data by bagging 10 classifier

Classifier of each sample	Income data x									
	100	120	160	180	186	190	210	250	270	300
Classifier 1	1	1	1	-1	-1	-1	-1	-1	-1	-1
Classifier 2	1	1	1	1	1	1	1	1	1	1
Classifier 3	1	1	1	-1	-1	-1	-1	-1	-1	-1
Classifier 4	1	1	1	-1	-1	-1	-1	-1	-1	-1
Classifier 5	1	1	1	-1	-1	-1	-1	-1	-1	-1
Classifier 6	-1	-1	-1	-1	-1	-1	-1	1	1	1
Classifier 7	-1	-1	-1	-1	-1	-1	-1	1	1	1
Classifier 8	-1	-1	-1	-1	-1	-1	-1	1	1	1
Classifier 9	-1	-1	-1	-1	-1	-1	-1	1	1	1
Classifier 10	1	1	1	1	1	1	1	1	1	1
Total	2	2	2	-6	-6	-6	-6	2	2	2
Sign of Total	1	1	1	-1	-1	-1	-1	1	1	1
Actual group y	1	1	1	-1	-1	-1	-1	1	1	1

7.6 Ensemble model

❖ Boosting

- In bagging, data is resampled with the same probability.
- Boosting extracts data by weighting depending on whether it is classified correctly in the previous stage.
- The classification results modify the probability that each data is selected in the next bootstrap sampling.
- ‘How do we modify the probability of extracting data in each boosting round?’
- ‘How do we synthesize the classifiers determined in each boosting round to make the final classification?’.

7.6 Ensemble model

❖ AdaBoosting algorithm

Step 1	Let $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ be the set of training data
Step 2	Let the initial probability being selected be $p_i^{(1)} = \frac{1}{n}, i = 1, 2, \ldots, n.$
Step 3	Let R be the number of bootstrap samples.
Step 4	for k = 1 to Rdo
Step 5	Generate bootstrap samples D_k of size n using $p_i^{(k)}$
Step 6	Create classifier C_k using bootstrap samples D_k
Step 7	Apply C_k to each data of D whether it classifies correctly or not
Step 8	calculates the misclassification rate $\epsilon_k = \frac{1}{n}\left[\sum_{i=1}^{n} p_i \, I\{C_k(\boldsymbol{x}_i) \neq y_i\}\right]$

7.6 Ensemble model

❖ AdaBoosting

Step 9	If $\epsilon_k > 0.5$ then
Step 10	Set again initial probability $p_i^{(1)} = \frac{1}{n}, i = 1, 2, \ldots, n$
Step 11	Go back to Step 4
Step 12	end if
Step 13	$\alpha_k = \frac{1}{2} ln \frac{1-\epsilon_k}{\epsilon_k}$
Step 14	$p_i^{(k+1)} = \frac{p_i^{(k)}}{Z_k} \times e^{-\alpha_k} \quad if\ C_k(\boldsymbol{x}_i) = y_i$ $= \frac{p_i^{(k)}}{Z_k} \times e^{\alpha_k} \quad if\ C_k(\boldsymbol{x}_i) \neq y_i$ Z is a constant that makes the sum of probability becomes 1.
Step 15	end for
Step 16	Classify an unknown data $\boldsymbol{x}$ into the weighted majority vote of each classifier, $C^*(\boldsymbol{x}) = argmax_v \sum_{k=1}^{R} \alpha_k I(C_k(\boldsymbol{x}) = v)$

7.6 Ensemble model

❖ AdaBoosting

[Example 7.6.2] (AdaBoosting)
Classify the data in Table 7.6.1 of Example 7.6.1 below using the Adaboosting ensemble method. The classifier in each round uses a minimum entropy decision stump.

Table 7.6.1 10 customer data with income x and purchase status y

x	100	120	160	180	186	190	210	250	270	300
y	1	1	1	-1	-1	-1	-1	1	1	1

<Answer of Example 7.6.2>

Table 7.6.4 (Sample 1) 10 bootstrap samples for AdaBoosting and classifier C_1

number	Bootstrap sample										Classifier C_1
Sample 1	x	100	180	186	190	190	210	216	210	250	300	If $x \leq 230$, then y = -1, else y = 1
	y	1	-1	-1	-1	-1	-1	-1	-1	1	1	

7.6 Ensemble model

❖ AdaBoosting

<Answer of Example 7.6.2>

Table 7.6.5 (Sample 2) 10 bootstrap samples for AdaBoosting and classifier C_2

number	Bootstrap sample										Classifier C_2
Sample 2	x	100	120	120	120	120	120	160	160	160	160	If $x \leq 50$, then y = -1, else y = 1
	y	1	1	1	1	1	1	1	1	1	1	

7.6 Ensemble model

❖ AdaBoosting

<Answer of Example 7.6.2>

Table 7.6.6 Process of updating the new probability of selection using C_2

Selction probability $p_i^{(2)}$	0.311	0.311	0.311	0.010	0.010	0.010	0.010	0.010	0.010	0.010
x_i	100	120	160	180	186	190	210	250	270	300
y_i	1	1	1	-1	-1	-1	-1	1	1	1
Classification by $C_2(x_i)$	1	1	1	1	1	1	1	1	1	1
$I(C_2(x_i) \neq y_i)$	0	0	0	1	1	1	1	0	0	0
$p_i^{(2)}\ I(C_2(x_i) \neq y_i)$	0	0	0	0.010	0.010	0.010	0.010	0	0	0
	$\epsilon_2 = 0.004, \quad \alpha_2 = \frac{1}{2}\ ln\ \frac{1-\epsilon_2}{\epsilon_2}$ = 2.758									
$e^{-\alpha_2} \quad if\ C_2(x_i) = y_i$ $e^{\alpha_2} \quad if\ C_2(x_i) \neq y_i$	0.063	0.063	0.063	15.78	15.78	15.78	15.78	0.063	0.063	0.063
$p_i^{(2)}$ × above row	0.02	0.02	0.02	0.158	0.158	0.158	0.158	0.006	0.006	0.006
	$Z_2 = 0.6922$									
New selction probability $p_i^{(3)}$	0.028	0.028	0.028	0.228	0.228	0.228	0.228	0.001	0.001	0.001

7.6 Ensemble model

❖ AdaBoosting

<Answer of Example 7.6.2>

Table 7.6.7 (Sample 3) 10 bootstrap samples for AdaBoosting and classifier C_3

number	Bootstrap sample											Classifier C_3
Sample 3	x	120	120	180	180	180	180	186	190	190	210	If $x \leq 150$, then y = 1, else y = -1
	y	1	1	-1	-1	-1	-1	-1	-1	-1	-1	

7.6 Ensemble model

❖ AdaBoosting

<Answer of Example 7.6.2>

Table 7.6.8 Process of updating the new probability of selection using C_3

Selction probability $p_i^{(3)}$	0.028	0.028	0.028	0.228	0.228	0.228	0.228	0.001	0.001	0.001
x_i	100	120	160	180	186	190	210	250	270	300
y_i	1	1	1	-1	-1	-1	-1	1	1	1
Classification by $C_3(x_i)$	1	1	1	-1	-1	-1	-1	-1	-1	-1
$I(C_3(x_i) \neq y_i)$	0	0	0	0	0	0	0	1	1	1
$p_i^{(3)}\ I(C_3(x_i) \neq y_i)$	0	0	0	0	0	0	0	0.001	0.001	0.001
	$\epsilon_3 = 0.0003, \quad \alpha_3 = \frac{1}{2}\, ln\, \frac{1-\epsilon_3}{\epsilon_3}$ = 4.0557									
$e^{-\alpha_3} \quad if\ C_3(x_i) = y_i$ $e^{\alpha_3} \quad if\ C_3(x_i) \neq y_i$	0.017	0.017	0.017	0.017	0.017	0.017	0.017	57.73	57.73	57.73
$p_i^{(3)} \times$ above row	0.0005	0.0005	0.0005	0.004	0.004	0.004	0.004	0.058	0.058	0.058
	$Z_3 = 0.1904$									
New selction probability $p_i^{(4)}$	0.003	0.003	0.003	0.021	0.021	0.021	0.021	0.303	0.303	0.303

7.6 Ensemble model

❖ AdaBoosting

<Answer of Example 7.6.2>

Table 7.6.9 Final classification results using AdaBootstrap

x_i	100	120	160	180	186	190	210	250	270	300	Classifier importance
$C_1(x_i)$ classification result	-1	-1	-1	-1	-1	-1	-1	1	1	1	$\alpha_1 = 1.738$
$C_2(x_i)$ classification result	1	1	1	1	1	1	1	1	1	1	$\alpha_2 = 2.758$
$C_3(x_i)$ classification result	1	1	1	-1	-1	-1	-1	-1	-1	-1	$\alpha_3 = 4.055$
Weighted sum of classification result	5.08	5.08	5.08	-3.04	-3.04	-3.04	-3.04	0.44	0.44	0.44	
Final classification result (sign)	1	1	1	-1	-1	-1	-1	1	1	1	

7.6 Ensemble model

❖ Random forest

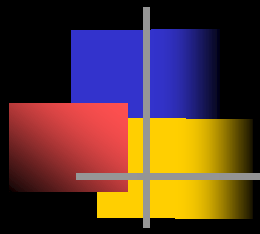
- Random forest is an ensemble method designed to combine the classification results of several decision trees.
- It is also used when there are many variables.
- Each decision tree is created using a subset of variables independently selected from all variables.
- The generation of the subset of variables can use random or probability distributions. The bagging method is a special case of the random forest method.

7.6 Ensemble model

❖ Random forest

[Random forest algorithm]

Step 1	Let $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$ be the set of training data, and m be the number of variables.
Step 2	Let R be the number of random forest samples.
Step 3	for k = 1 to Rdo
Step 4	Generate random forest samples D_k with the subset of all variables.
Step 5	Create classifier C_k using random forest samples D_k
Step 6	end for
Step 7	Classify an unknown data $\boldsymbol{x}$ by the majority vote of each classifier.



Summary

- Bayes classification model for continuous data
- Logistic regression model
- K-nearest neighbor classification model
- Neural network model
- Support vector machine model
  - Linear support vector machine
  - Nonlinear support vector machine
- Ensemble model
  - Bagging
  - AdaBoosting
  - Random forest

Thank you !!!