Machine learning handbook

Classifier

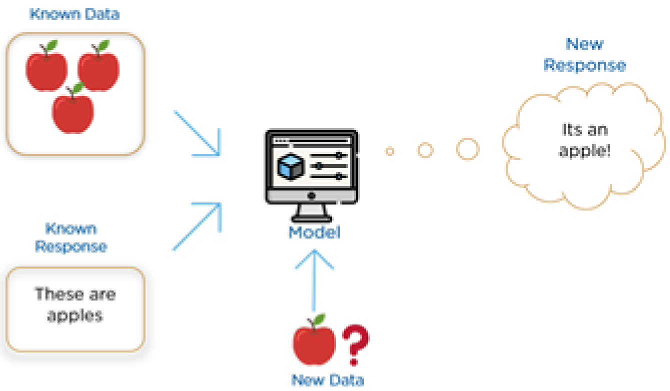
Taeyang Yang

Update: July 13, 2018

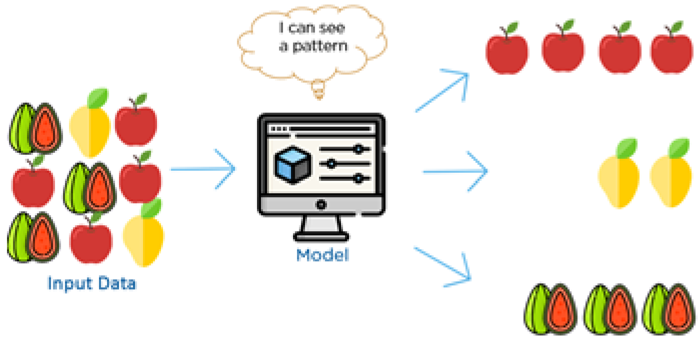
BCILAB, UNIST

Supervised & unsupervised learning

Supervised learning

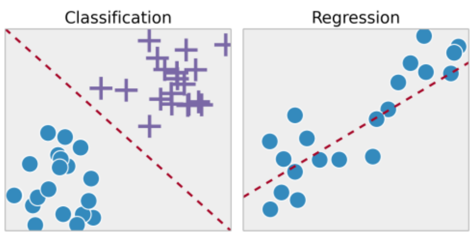


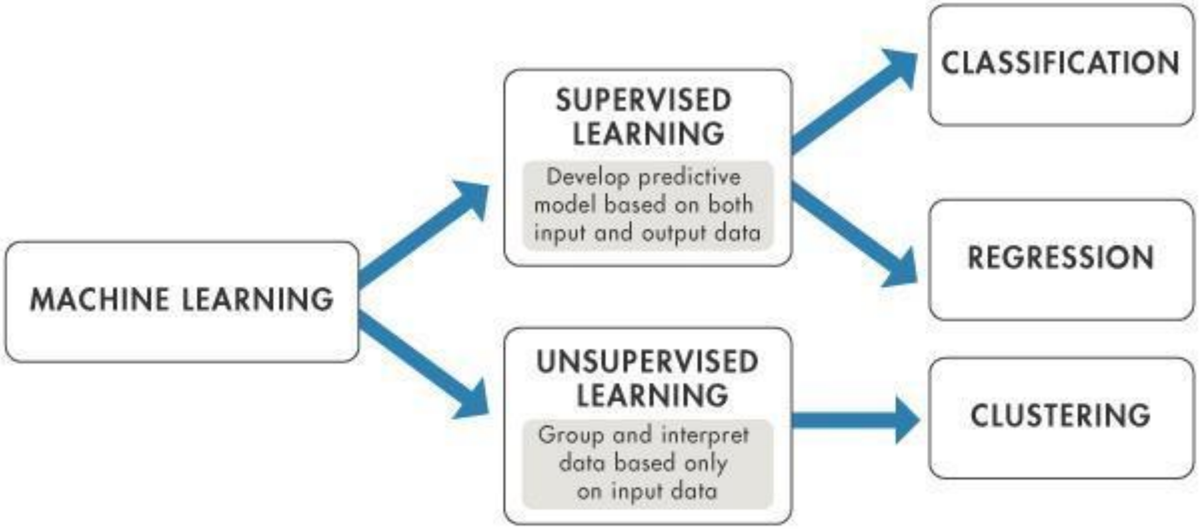
Unsupervised learning



https://www.quora.com/What-is-the-difference-between-supervised-and-unsupervised-learning-algorithms

Classification, Regression & Clustering





https://www.researchgate.net/figure/Supervised-Learning-versus-Unsupervised-Learning-Mathworks-nd_fig1_319937079

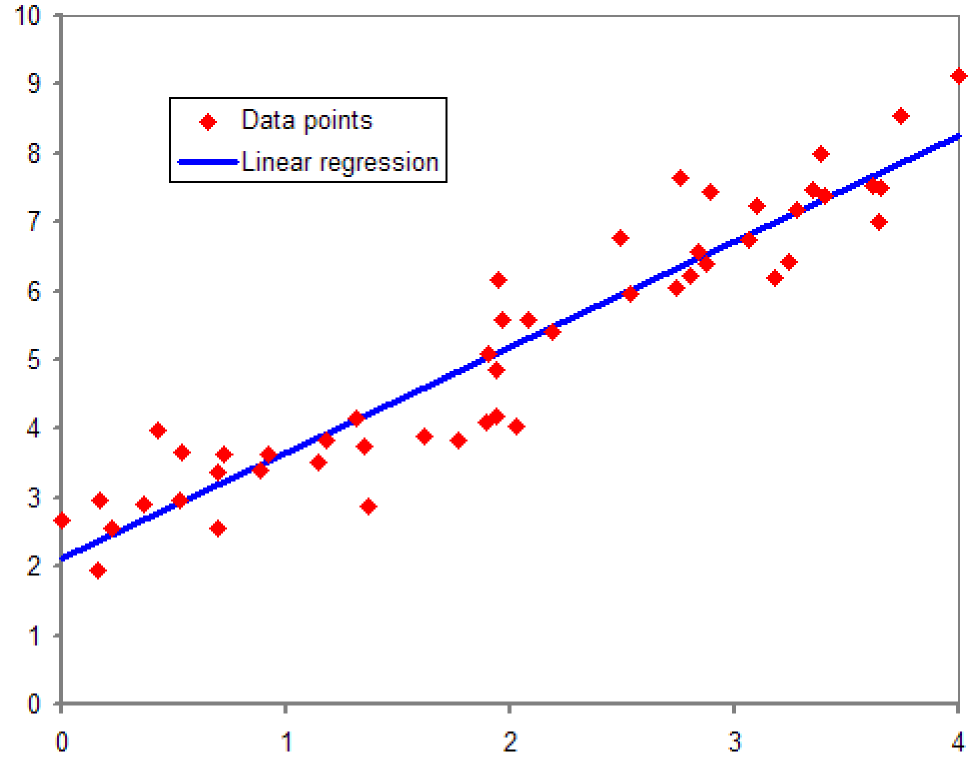
https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

1

Supervised learning

- Linear Regression
- Naïve Bayes (NB) classifier
- Linear Discriminant Analysis (LDA)
- Support Vector Machine (SVM)

Linear Regression (overview)



$$\hat{y} = ax + b$$

Linear Regression

- Same expression !

$$\boldsymbol{y} = \boldsymbol{a}\boldsymbol{x} + \boldsymbol{b}$$

$$\hat{y} = X\beta + \varepsilon$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{1p}} \\ \boldsymbol{x_{21}} & & \boldsymbol{x_{2p}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{n1}} & \cdots & \boldsymbol{x_{np}} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

- **Cost (Loss) function**

$$J(\beta) = MSE = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)^2$$
$$= \frac{1}{2m} \sum_{i=1}^{m} \left(X^{(i)}\beta + \varepsilon - y^{(i)} \right)^2$$

Linear Regression

- Cost (Loss) function

$$J(\beta) = MSE = \frac{1}{2m}\sum_{i=1}^{m}\left(X^{(i)}\beta + \varepsilon - y^{(i)}\right)^2$$

- Derivate

$$\frac{\partial J}{\partial \varepsilon} = \frac{1}{m}\sum_{i=1}^{m}\left(X^{(i)}\beta + \varepsilon - y^{(i)}\right)$$

$$\frac{\partial J}{\partial \beta} = \frac{1}{m}\sum_{i=1}^{m}\left(X^{(i)}\beta + \varepsilon - y^{(i)}\right)X^{(i)}$$

Linear Regression

- Solution 1: Normal equation, **derivate=0**

$$\varepsilon m + \beta \sum_{i=1}^{m} X^{(i)} = \sum_{i=1}^{m} y^{(i)}$$
$$\varepsilon \sum_{i=1}^{m} X^{(i)} + \beta \sum_{i=1}^{m} {X^{(i)}}^2 = \sum_{i=1}^{m} y^{(i)} X^{(i)}$$

- **Matrix expression**

$$\begin{bmatrix} \varepsilon \\ \beta \end{bmatrix}^T \begin{bmatrix} m & \sum_{i=1}^{m} X^{(i)} \\ \sum_{i=1}^{m} X^{(i)} & \sum_{i=1}^{m} {X^{(i)}}^2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} y^{(i)} \\ \sum_{i=1}^{m} y^{(i)} X^{(i)} \end{bmatrix}$$

Linear Regression

- Trick

$$\boldsymbol{w} = \begin{bmatrix} \varepsilon \\ \beta \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} 1 & \boldsymbol{x}_1^T \\ 1 & \boldsymbol{x}_2^T \\ & \vdots \\ 1 & \boldsymbol{x}_n^T \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

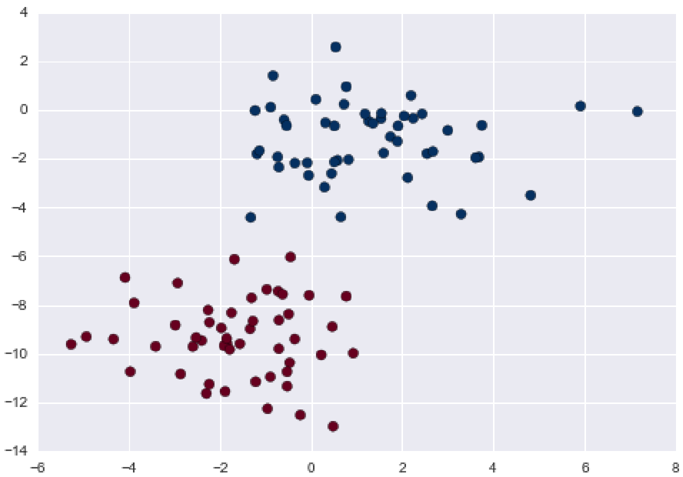
$$\boldsymbol{X}^T\boldsymbol{X} = \begin{bmatrix} m & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \end{bmatrix}, \boldsymbol{X}^T\boldsymbol{y} = \begin{bmatrix} \sum_{i=1}^m y_i \\ \sum_{i=1}^m y_i x_i \end{bmatrix}$$

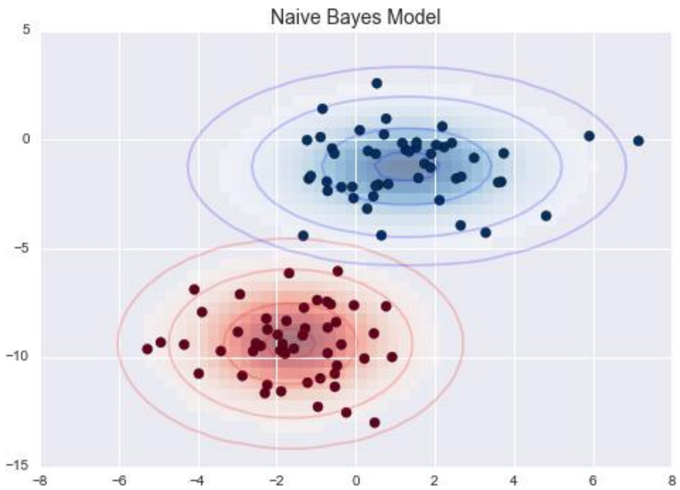
- Estimated weight

$$X^T X w = X^T y$$

$$\boldsymbol{w} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Naïve Bayes classifier (overview)





https://medium.com/@gp_pulipaka/applying-gaussian-na%C3%AFve-bayes-classifier-in-python-part-one-9f82aa8d9ec4

Naïve Bayes classifier

- **Bayes theorem-based model**
  - Bayes' theorem

Likelihood *Class prior probability*

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

Posterior probability *Predictor prior probability*

Naïve Bayes classifier

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

- **Joint probability**: chain rule of conditional probability

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2, \dots, x_n, C_k) \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k)p(x_3, \dots x_n, C_k) \\ &= \cdots \\ &= p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k) \dots p(x_n|C_k)p(C_k) \end{aligned}$$

- **Assumption of "naïve conditional independence"** :
  - Each x_i is conditionally independent of every other features x_j for $j \neq i$ given the category C_k

$$p(x_i|x_{i+1}, \dots, x_n, C_k) = p(x_i|C_k)$$

- **Posterior**

$$\begin{aligned} p(C_k|x_1, \dots x_n) \propto p(C_k, x_1, \dots, x_n) &= p(C_k)p(x_1|C_k)p(x_2|C_k) \dots p(x_n|C_k) \\ &= p(C_k)\prod_{i=1}^{n} p(x_i|C_k) \end{aligned}$$

Naïve Bayes classifier

- Maximum a posterior (MAP)

$$\hat{y} = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}}\, p(C_k) \prod_{i=1}^{n} p(x_i | C_k)$$

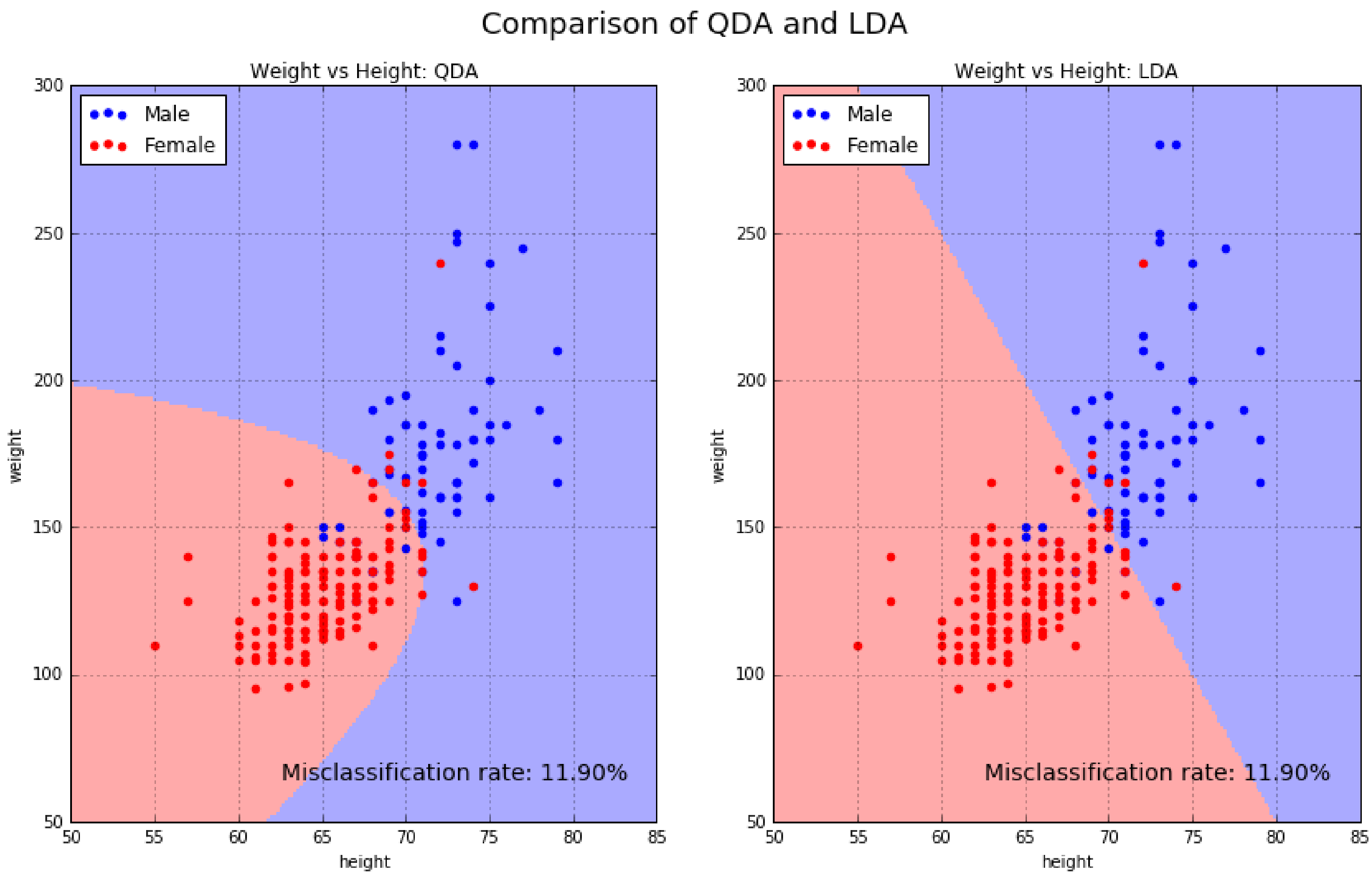
- What is $\boldsymbol{p(x_i|C_k)}$?
  - parameter estimation based on assumption of likelihood
    - Gaussian naïve Bayes → μ, σ

$$p(x = v | C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

    - Multinomial naïve Bayes → p_{ki}

$$p(X|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x^i}$$

Linear Discriminant Analysis (overview)



http://www.phillypham.com/Nearest%20Neighbors%20and%20Discriminant%20Analysis

Linear Discriminant Analysis

- 데이터 클래스를 가장 잘 분류하는 축으로 projection
  - 원래: P 차원의 데이터 벡터 X_i

$$N_1: the\ number\ of\ samples\ in\ class\ C_1$$
$$N_2: the\ number\ of\ samples\ in\ class\ C_2$$

  - 원 좌표계의 평균벡터

$$\overrightarrow{m_1} = \frac{1}{N_1} \sum_{n \in C_1} X_n$$
$$\overrightarrow{m_2} = \frac{1}{N_2} \sum_{n \in C_2} X_n$$

  - Projected된 평균벡터

$$\overrightarrow{\mu_1} = \vec{w}^T \overrightarrow{m_1}$$
$$\overrightarrow{\mu_2} = \vec{w}^T \overrightarrow{m_2}$$

Linear Discriminant Analysis

- 목표: Projected 시킨 좌표계에서 아래를 만족시키는 $\vec{w}$ 구하기
  - 클래스간 차이는 크고 = 평균 차이 max

$$\underset{\vec{w}}{\boldsymbol{argmax}}((\overrightarrow{\boldsymbol{\mu_1}} - \overrightarrow{\boldsymbol{\mu_2}})^2)$$

$$\overrightarrow{\mu_1} - \overrightarrow{\mu_2} = \vec{w}^T \overrightarrow{m_1} - \vec{w}^T \overrightarrow{m_2} = \vec{w}^T (\overrightarrow{m_1} - \overrightarrow{m_2})$$

  - 클래스내 차이는 적게 = 클래스 내 분산 차이 min

$$\underset{\vec{w}}{\boldsymbol{argmin}}(\boldsymbol{s_1^2} + \boldsymbol{s_2^2})$$

$$s_1^2 = \sum_{n \in C_1} (w^T X_n - \overrightarrow{\mu_1})^2$$

$$s_2^2 = \sum_{n \in C_1} (w^T X_n - \overrightarrow{\mu_1})^2$$

Linear Discriminant Analysis

- 목표: Projected 시킨 좌표계에서 아래를 만족시키는 $\vec{w}$ 구하기
  - 평균 차이는 최대화시키고, 클래스 내 분산 차이는 최소화시키는 방법?
    - 분수 !

$$\underset{\vec{w}}{argmax}\frac{(\overrightarrow{\mu_1}-\overrightarrow{\mu_2})^2}{s_1^2+s_2^2}$$

Linear Discriminant Analysis

- 분자항 전개

$$
\begin{aligned}
\overrightarrow{\mu_1} - \overrightarrow{\mu_2} &= (w^T\overrightarrow{m_1} - w^T\overrightarrow{m_2})^2 \\
&= (w^T(\overrightarrow{m_1} - \overrightarrow{m_2}))^2 \\
&= w^T(\overrightarrow{m_1} - \overrightarrow{m_2})(\overrightarrow{m_1} - \overrightarrow{m_2})^T w \\
&= w^T S_w w
\end{aligned}
$$

- **Between class scatter matrix**: 대칭행렬

$$S_w = (\overrightarrow{m_1} - \overrightarrow{m_2})(\overrightarrow{m_1} - \overrightarrow{m_2})^T$$

Linear Discriminant Analysis

- **분모항 전개**

$$s_1^2 + s_2^2 = \sum_{n \in C_1} (w^T X_n - \overrightarrow{\mu_1})^2 + \sum_{n \in C_2} (w^T X_n - \overrightarrow{\mu_1})^2$$
$$= \sum_{n \in C_1} (w^T X_n - w^T \overrightarrow{m_1})^2 + \sum_{n \in C_2} (w^T X_n - w^T \overrightarrow{m_2})^2$$
$$= \sum_{n \in C_1} (w^T (X_n - \overrightarrow{m_1}))^2 + \sum_{n \in C_2} (w^T (X_n - \overrightarrow{m_2}))^2$$
$$= \sum_{n \in C_1} w^T (X_n - \overrightarrow{m_1})(X_n - \overrightarrow{m_1})^T w + \sum_{n \in C_2} w^T (X_n - \overrightarrow{m_2})(X_n - \overrightarrow{m_2})^T w$$
$$= w^T \left[\sum_{n \in C_1} (X_n - \overrightarrow{m_1})(X_n - \overrightarrow{m_1})^T + \sum_{n \in C_2} (X_n - \overrightarrow{m_2})(X_n - \overrightarrow{m_2})^T \right] w$$

- **Within-class Scatter matrix**: 대칭행렬

$$S_W = \sum_{n \in C_1} (X_n - \overrightarrow{m_1})(X_n - \overrightarrow{m_1})^T + \sum_{n \in C_2} (X_n - \overrightarrow{m_2})(X_n - \overrightarrow{m_2})^T$$

Linear Discriminant Analysis

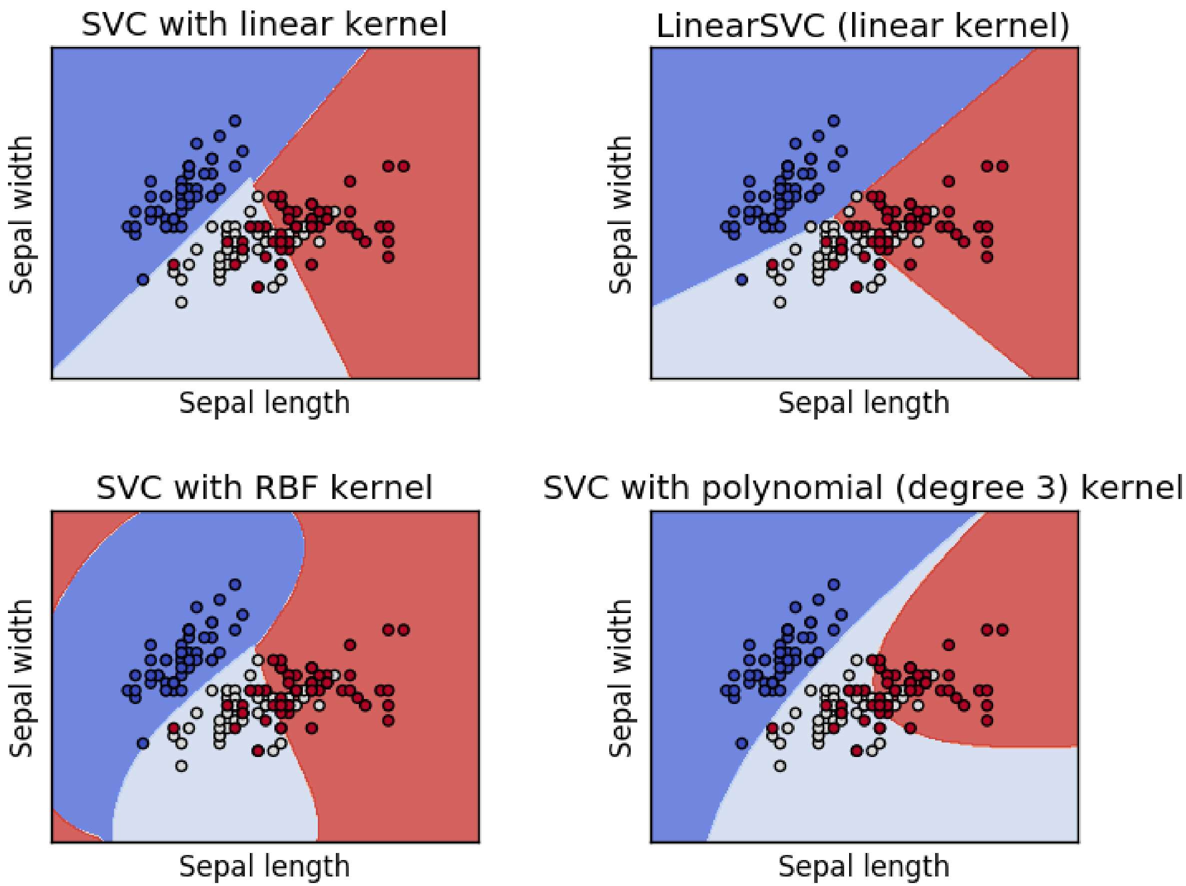
- “최대화”: $\vec{w}$에 대해 미분한 값 = 0

$$J(\vec{w}) = \frac{(\overrightarrow{\mu_1} - \overrightarrow{\mu_2})^2}{s_1^2 + s_2^2} = \frac{w^T S_B w}{w^T S_W w}$$

$$\frac{\partial J(\vec{w})}{\partial \vec{w}} = \frac{2S_B S_W w - w^T S_B w 2 S_W w}{(w^T s_W w)^2} = 0$$

- $\vec{w}$ 를 찾으면 됩니다 !
  - 어떻게? 아직 모르겠다 …

Support Vector Machine (overview)



Support Vector Machine

2

Unsupervised learning

- Random forest
- K-nearest neighborhood
- K-means clustering

Random forest

K-nearest neighborhood

K-means clustering