강원대학교
AI 소프트웨어학과

머신러닝2

- 지도학습 –

Support Vector Machine

Support Vector Machine

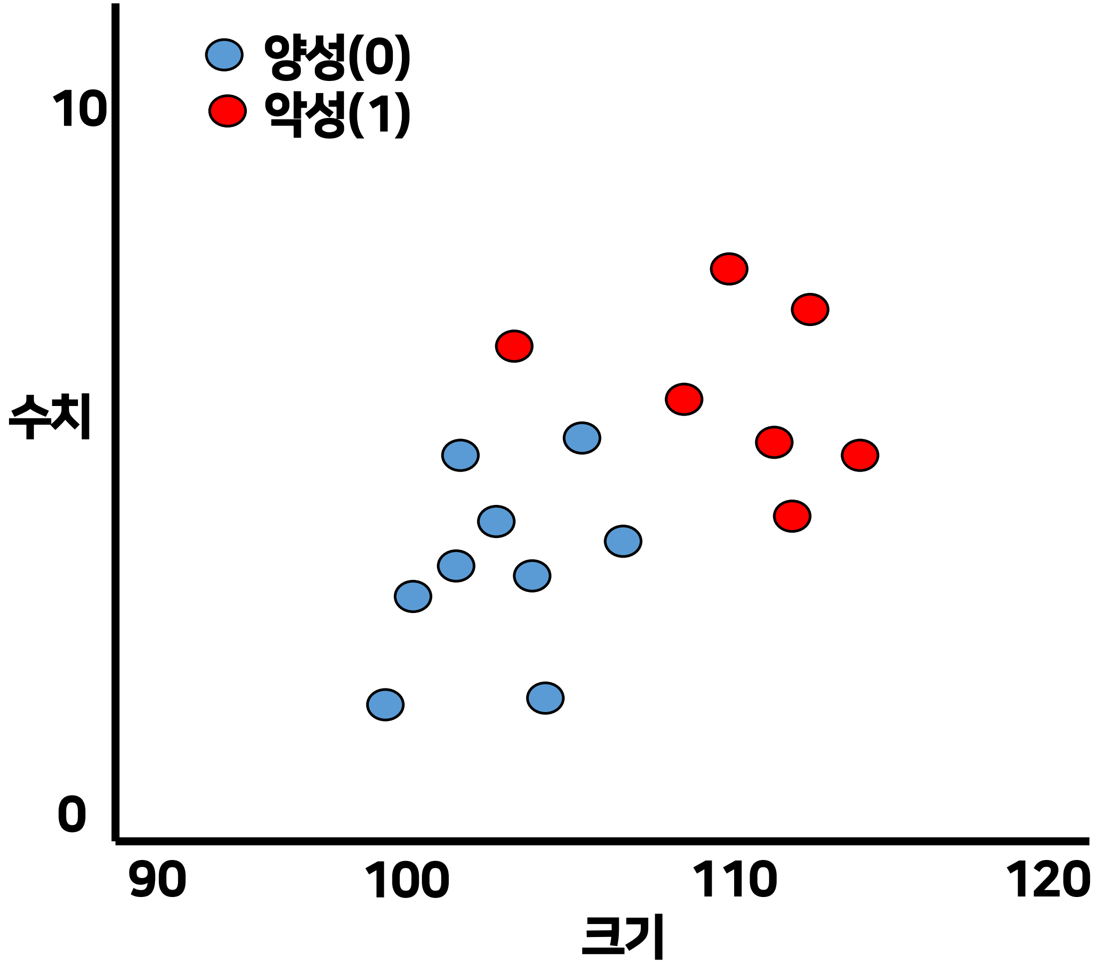
기계학습의 방법(지도학습)

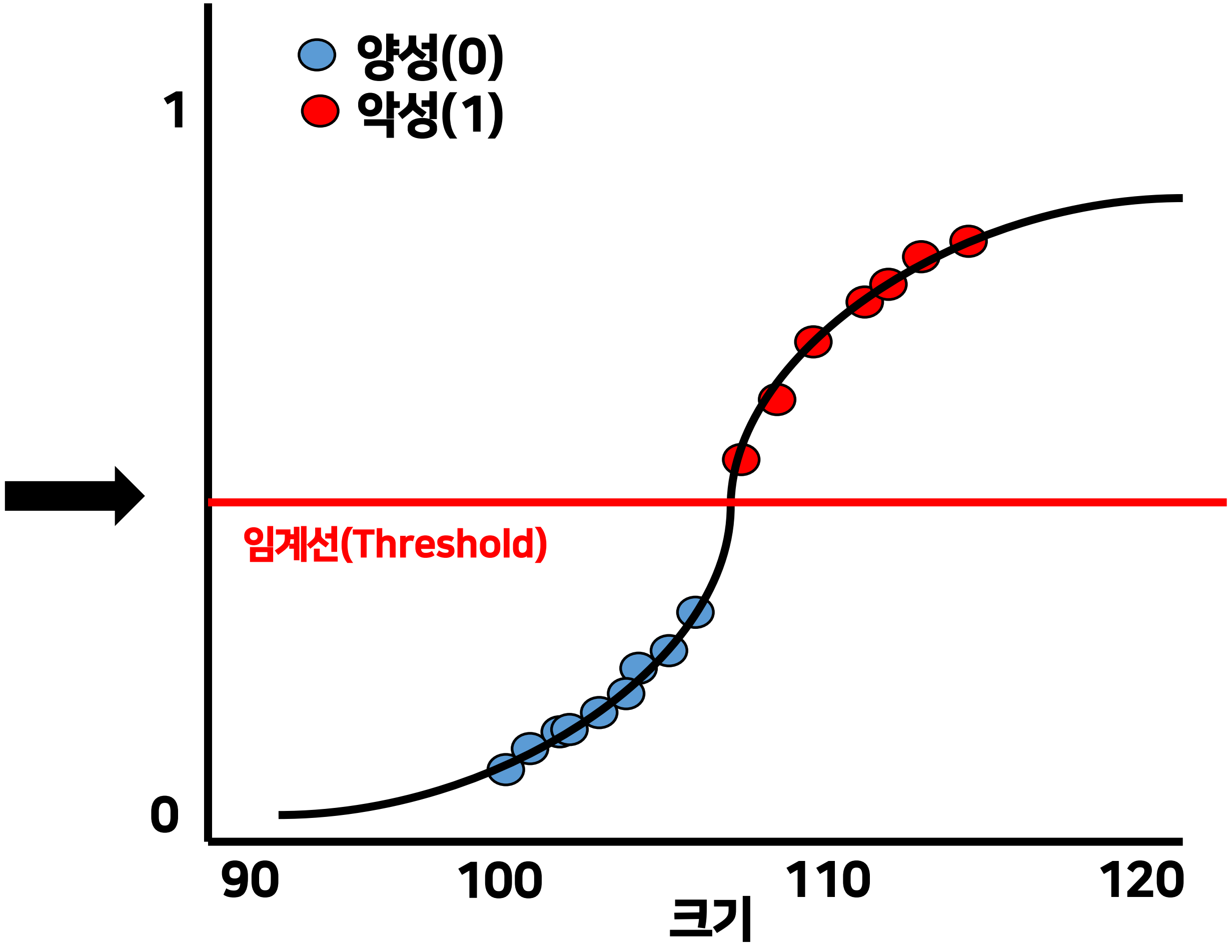
- 다중 회귀분석(Regression)
- 로지스틱 회귀분석(Logistic Regression)
- 신경망(Atificial Neural Network)
- 서포트 백터 머신(Support Vector Machine)
- 의사결정나무(Decision Tree)
- 앙상블(Ensemble)
- K-근접 이웃기법(k-Nearest Neighbor)

01 지도학습

Support Vector Machine

어떤 선이 가장 잘 두 집단을 구분할 수 있을까?

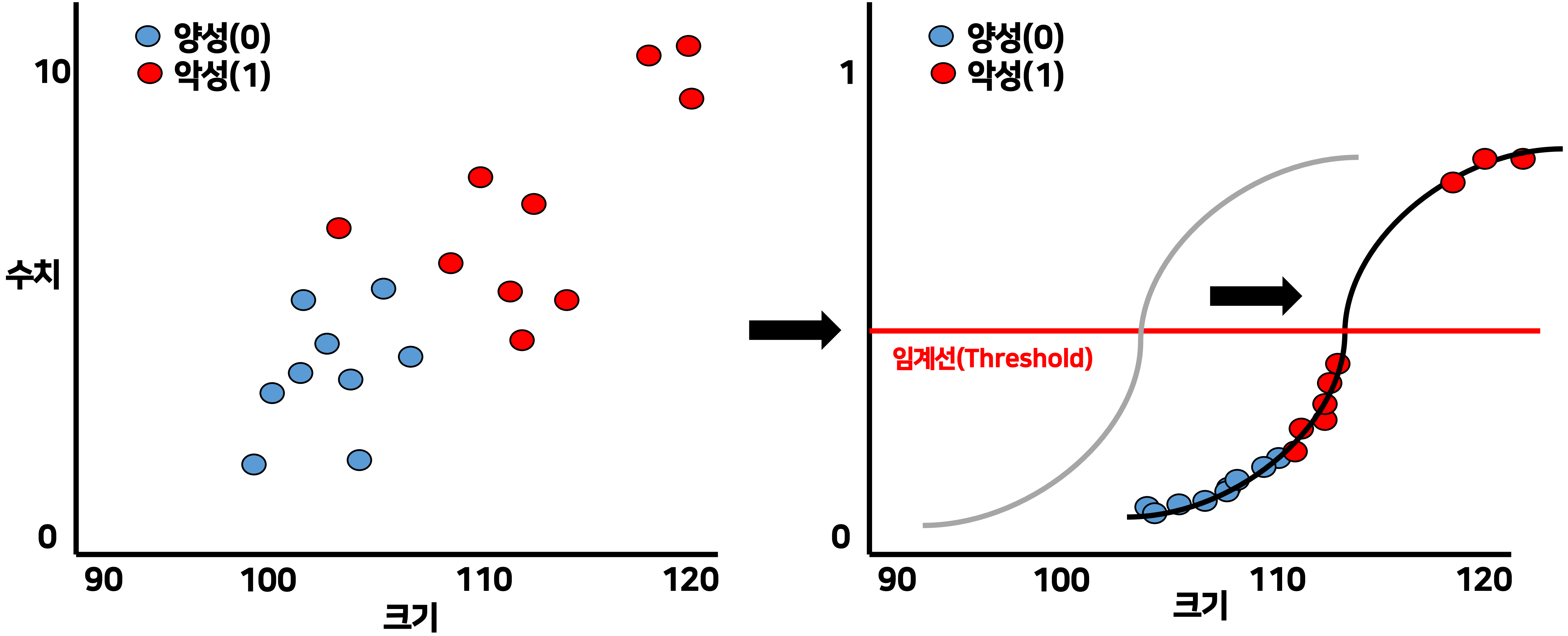




01 지도학습

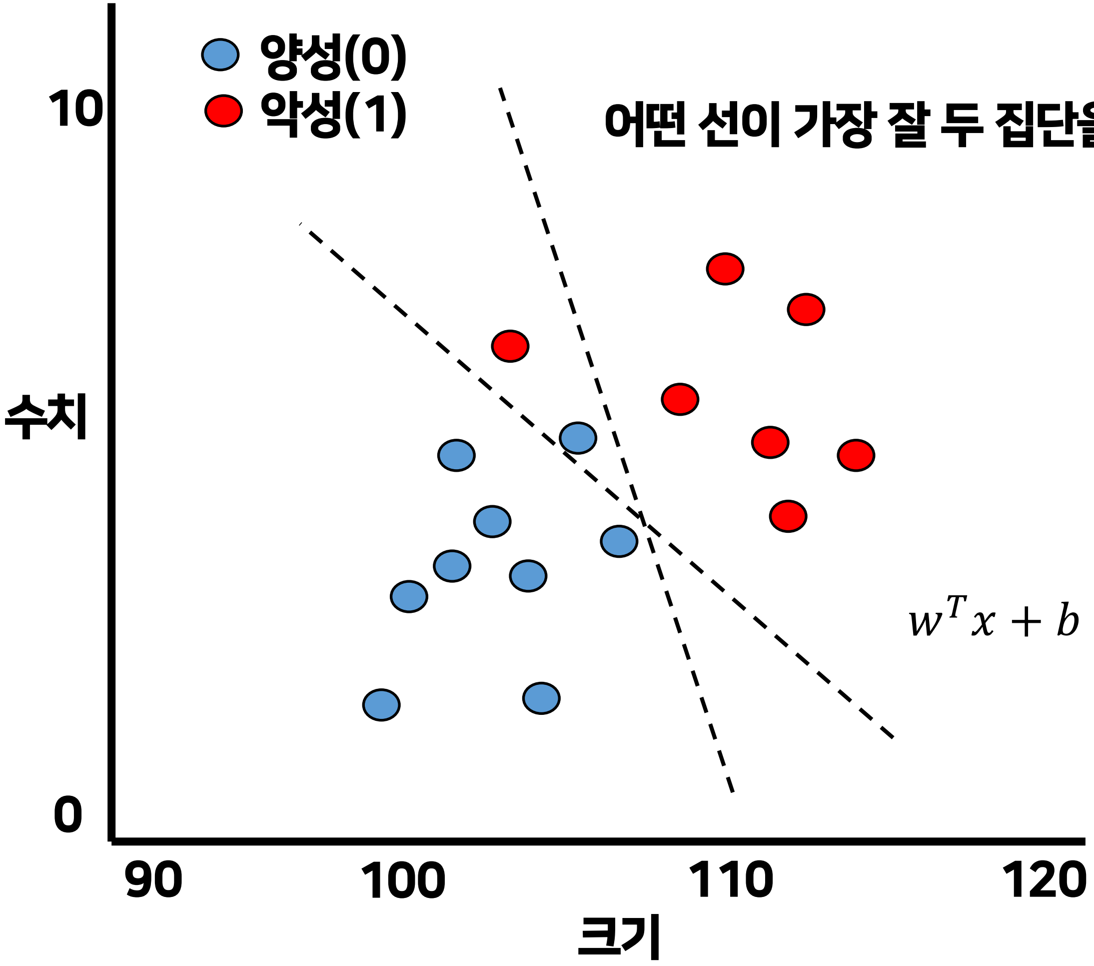
Support Vector Machine

어떤 선이 가장 잘 두 집단을 구분할 수 있을까?



01 지도학습

Support Vector Machine



어떤 선이 가장 잘 두 집단을 구분할 수 있을까?

$$w^T x + b = 0$$

W : 기울기
b : y절편

Support Vector Machine

서포트 벡터 머신(SVM)

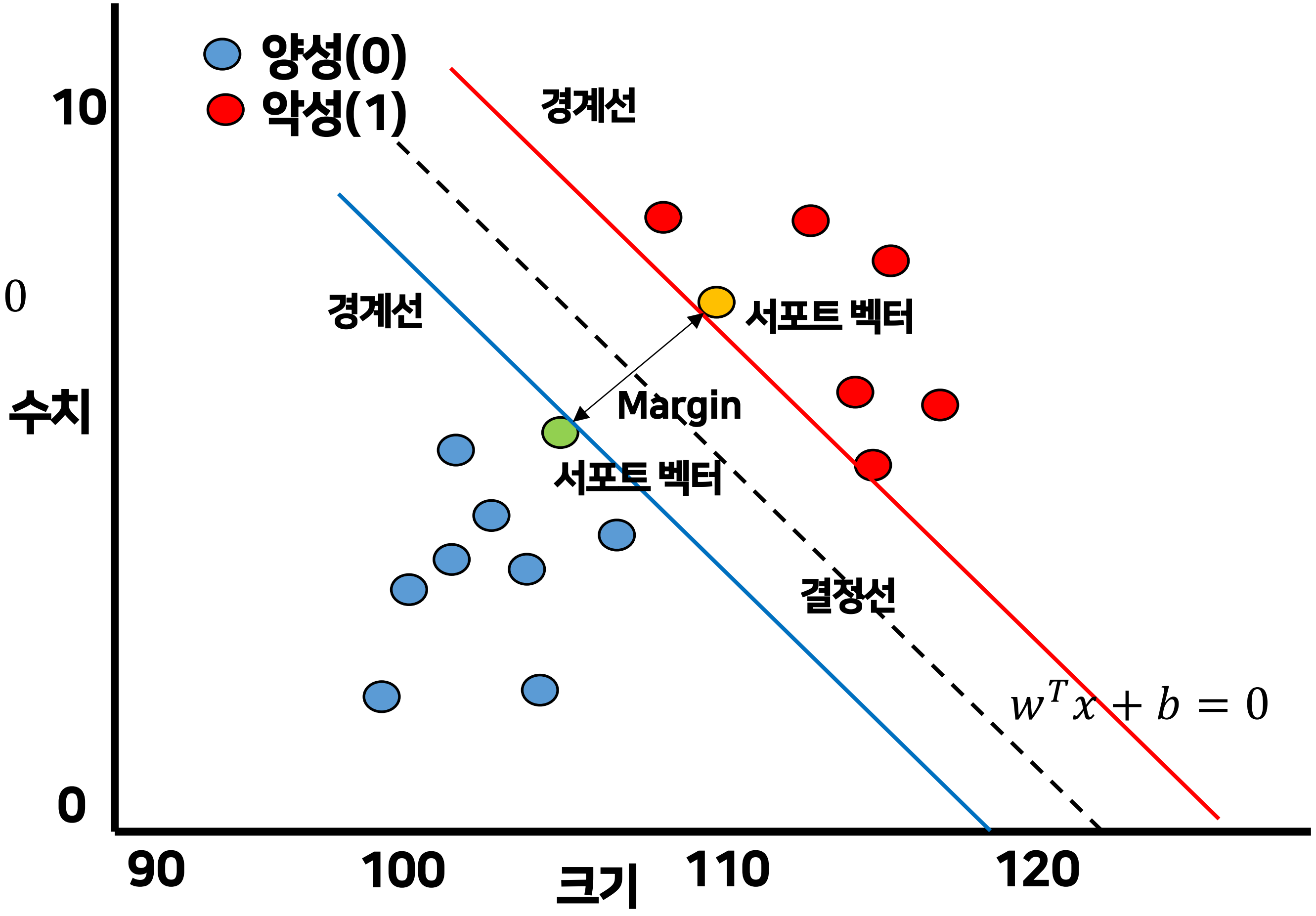
- 퍼셉트론의 확장된 개념
- 퍼셉트론 학습 : 분류오차의 최소화$(y - \hat{y})$
- SVM의 학습 : Margin(초평면=결정경계)의 최대화

SVM 장점

- 구조적이며 매번 수행해도 결과가 어느정도 비슷함
- 고차원 분류문제에 좋은 성능을 보이는 모델
- 오류 데이터 영향이 적음(서포트 벡터 값만 사용)
- 과적합되는 경우가 적음
- 신경망보다 사용하기 쉬움

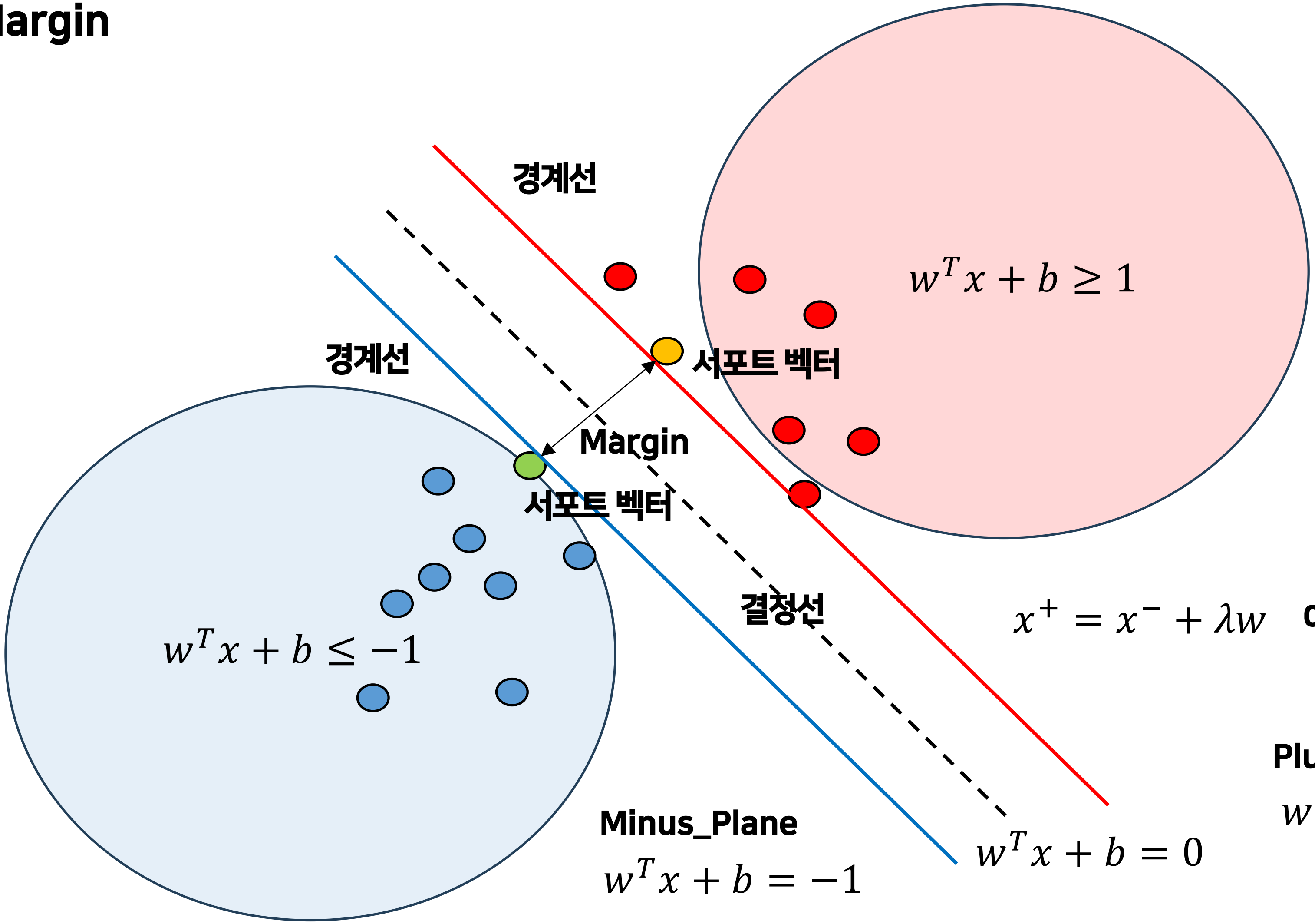
서포트 벡터 머신(SVM)

- 퍼셉트론의 확장된 개념
- 퍼셉트론 학습 : 분류오차의 최소화$(y - \hat{y})$
- SVM의 학습 : Margin(초평면=결정경계)의 최대화
- Margin을 최대화 하는 결정선(Hyperplane)찾자
- 결정선은 $w^T x + b = 0$ → $w_1 x_1 + w_2 x_2 + b = 0$
- 내적을 구하기 위해



Support Vector Machine

Geometric Margin



$x^+ = x^- + \lambda w$ 이동의 폭을 λ 로 하자

Support Vector Machine

Geometric Margin

$w^T x^+ + b = 1$

$w^T (x^- + \lambda w) + b = 1$

$w^T x^- + b + \lambda w^T w = 1$

$-1 + \lambda w^T w = 1$

$\lambda = \frac{2}{w^T w}$

$w = hyper\ Plane$ 기울기

$(x^+ = x^- + \lambda w)$

$w^T x^- + b = -1$

$$\begin{aligned} Margin &= distance(x^+, x^-) \\ &= ||x^+ - x^-||_2 \\ &= ||(x^- + \lambda w) - x^-||_2 \\ &= ||\lambda w||_2 \\ &= \lambda \sqrt{w^T w} \\ &= \frac{2}{w^T w} * \sqrt{w^T w} \\ &= \frac{2}{w^T w} = \frac{2}{||w||_2} \end{aligned}$$

The vector norm $||W||_p$ for p=1,2,3, … P norm

$$||w||_p = (\sum_i |w_i|^p)^{1/p}$$

$$L_2\ norm = ||w||_2 = (\sum_i |w_i|^2)^{\frac{1}{2}} = \sqrt{w_1^2 + w_2^2 + \cdots w_n^2} = \sqrt{w^T w}$$

벡터의 길이를 측정하는데 사용되는 함수

Support Vector Machine

Geometric Margin

$$max\ Margin = max\frac{2}{||w||_2} \Leftrightarrow min\frac{1}{2}||w||_2^2 \qquad ||w||_2 = \sqrt{w_1^2 + w_2^2 + \cdots w_n^2}$$

W의 L2 norm이 제곱근을 포함하고 있기 때문에 계산이 어려운(편의를 위해 아래와 같은 형태로 목적함수를 변경)

$$max\frac{2}{||w||_2} \Leftrightarrow min\frac{1}{2}||w||_2^2 \Rightarrow ||w||_2^2 = w_1^2 + w_2^2 + \cdots w_n^2$$

수치의 단순화, 알고리즘을 일관되게 동작하기 위해서

Geometric Margin

$$w^T x + b = -1 \qquad w^T x + b = 0 \qquad w^T x + b = 1$$

Distance(Margin)

$$max \frac{2}{||w||_2} \Leftrightarrow min \frac{1}{2} ||w||_2^2$$

$$||w||_2 = (\sum_i |w_i|^2)^{\frac{1}{2}} = \sqrt{w_1^2 + w_2^2 + \cdots w_n^2} = \sqrt{w^T w}$$

$case\ 1{:}\ ||w||_2 = 2$

$$\frac{2}{||w||_2} = \frac{2}{2} = 1 \Leftrightarrow \frac{1}{2}||w||_2^2 = \frac{1}{2}(2^2) = 2$$

$case\ 2{:}\ ||w||_2 = 1$

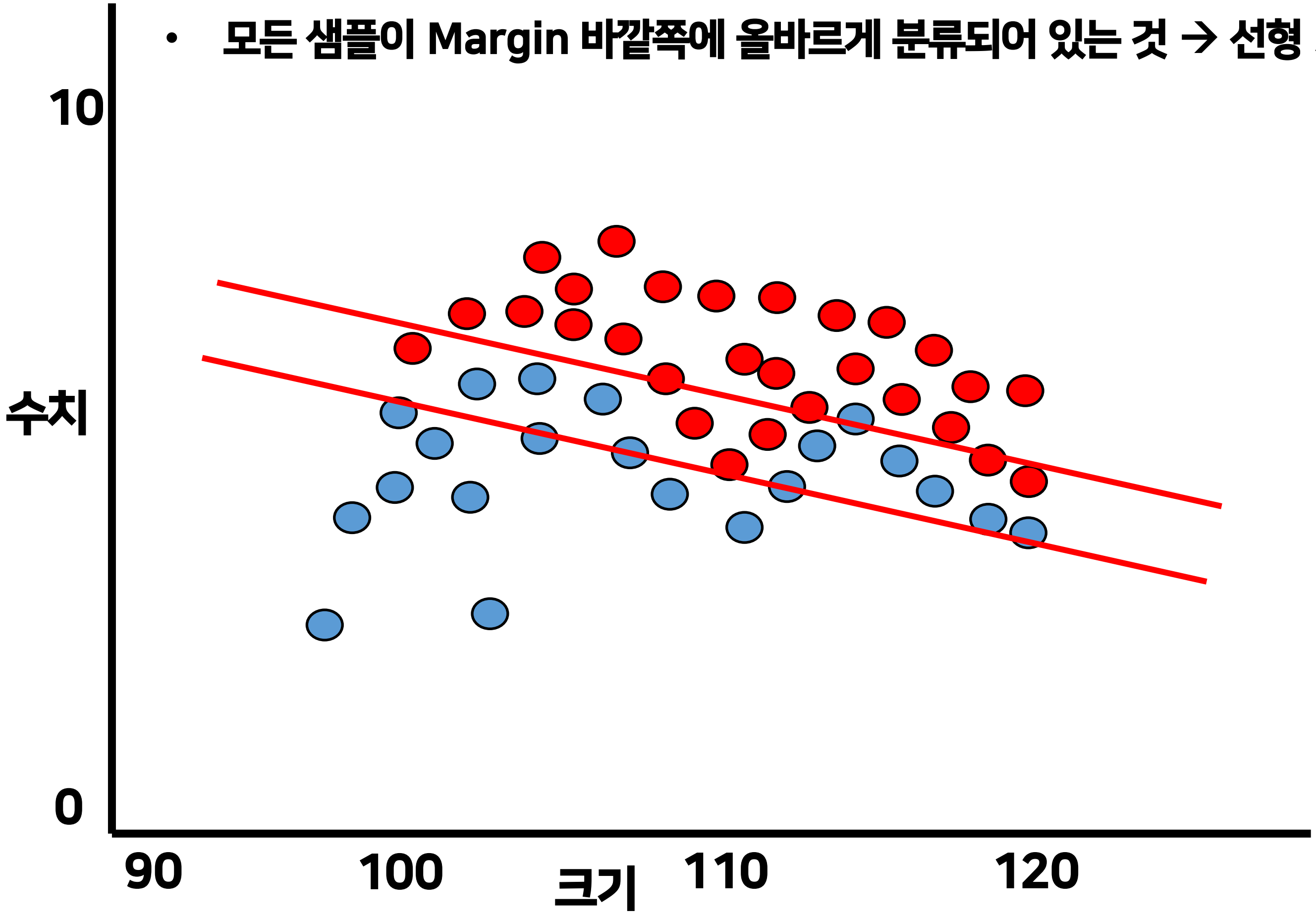
$$\frac{2}{||w||_2} = \frac{2}{1} = 2 \Leftrightarrow \frac{1}{2}||w||_2^2 = \frac{1}{2}(1^2) = 0.5$$

Support Vector Machine

Original Problem : Hard Margin

$$min \frac{1}{2} ||w||_2^2$$

- 모든 샘플이 Margin 바깥쪽에 올바르게 분류되어 있는 것 → 선형 SVM



Original Problem

$$min \frac{1}{2}||w||_2^2$$

L2 norm를 최소화 하고, 1보다는 큰 마진을 가지는 w, b를 구하는 것이 목표

$$y_i(w^T x_i + b) \geq 1, i = 1,2,3,4 \ldots, \mathrm{n}$$

Lagrangian Multiplier(마진을 최대화 또는 최소화) → 가중치 업데이트

$$\max_{\alpha} \min_{w,b} \mathcal{L}(w, b, \alpha) = \frac{1}{2}||w||_2^2 - \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - 1)$$

$$\alpha_i \geq 0, i = 1,2,3, 4, \ldots, n$$

- **SVM에서 제약 조건 : 포인트를 잘못 분류하지 마라(다른 영역을 침범하지 마!)**
- **최적화 문제, 특히 제약 조건을 처리할 때 사용되는 강력한 수학적 도구**
- **하나 이상의 제약 조건이 적용되는 함수의 최대값 또는 최소값을 찾는 데 도움 (목적식과 제약식을 하나로 표현하기 위해)**

Lagrangian Multiplier(마진을 최대화) → 가중치 업데이트

$$\max_{\alpha} \min_{w,b} \mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||_2^2 - \sum_{i=1}^{n} \alpha_i (y_i (w^T x_i + b) - 1)$$

α_i **: 각 w가 데이터 포인트에 미치는 영향**

$$\alpha_i \geq 0, i = 1,2,3,4,\dots,n$$

Convex, continuous이기 때문에 미분하면 =0의 최소값을 가짐

$$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

Lagrangian Multiplier

$$\frac{1}{2}||w||_2^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1)$$

$$\frac{\partial \mathcal{L}(w, b, \alpha)}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{1}{2}||w||_2^2 = \frac{1}{2} w^T w \quad = \frac{1}{2} w^T \sum_{j=1}^{n} \alpha_j y_j x_j$$

$$= \frac{1}{2} \sum_{j=1}^{n} \alpha_i y_i (w^T x_j)$$

$$= \frac{1}{2} \sum_{j=1}^{n} \alpha_i y_i (\sum_{i=1}^{n} \alpha_i y_i x_i^T x_j)$$

$$= \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Lagrangian Multiplier

$$\frac{1}{2}||w||_2^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1)$$

$$= -\sum_{i=1}^{n} \alpha_i y_i (w^T x_i - b) + \sum_{i=1}^{n} \alpha_i$$

$$= -\sum_{i=1}^{n} \alpha_i y_i w^T x_i - b \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i$$

$$= -\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, x_i^T x_j + \sum_{i=1}^{n} \alpha_i$$

$$\frac{\partial \mathcal{L}(w, b, \alpha)}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}(w, b, \alpha)}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Support Vector Machine

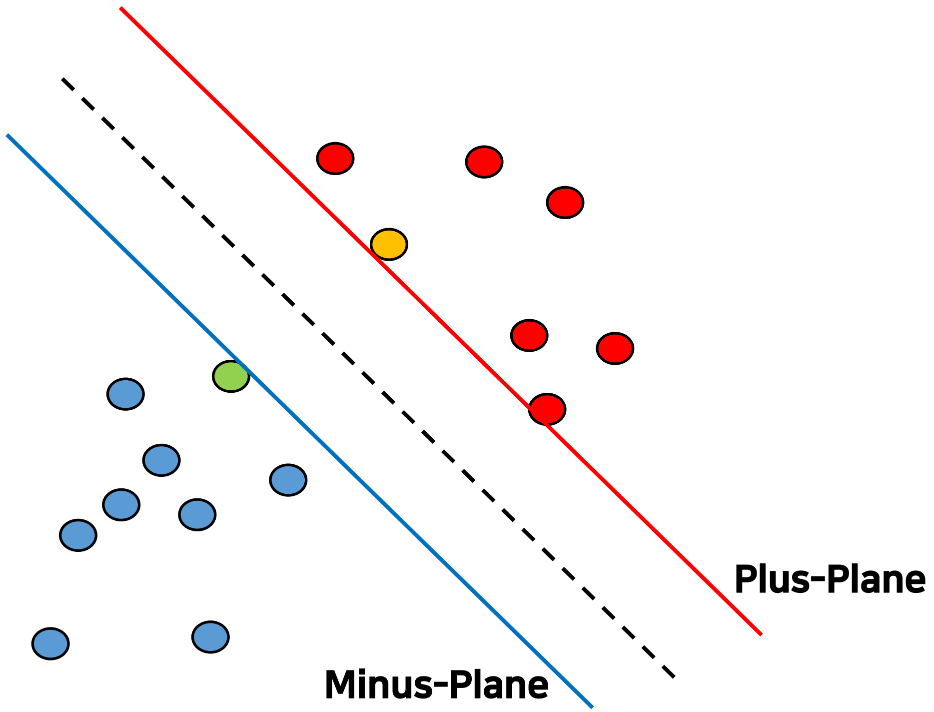
$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j\, x_i^T x_j$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0, i = 1,2,3,4, \dots, n$$

$$\alpha_i(y_i(w^T x_i + b) - 1 = 0$$

$$Plus\_Plane : \ \alpha_i > 0 \ and \ y_i(w^T x_i + b) - 1 = 0$$

$$Minus\_Plane : \ \alpha_i = 0 \ and \ y_i(w^T x_i + b) - 1 \neq 0$$



Support Vector Machine

$y = ax + b$

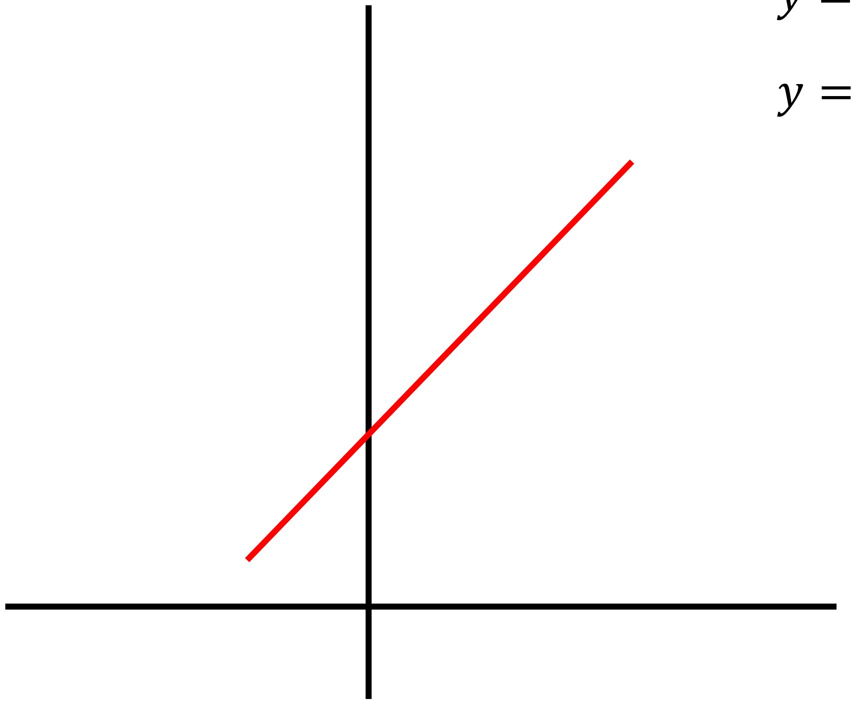
⇩

$w^T x + b = 0$

항상 그래프의 직선은 기울기와 절편의 값을 가짐

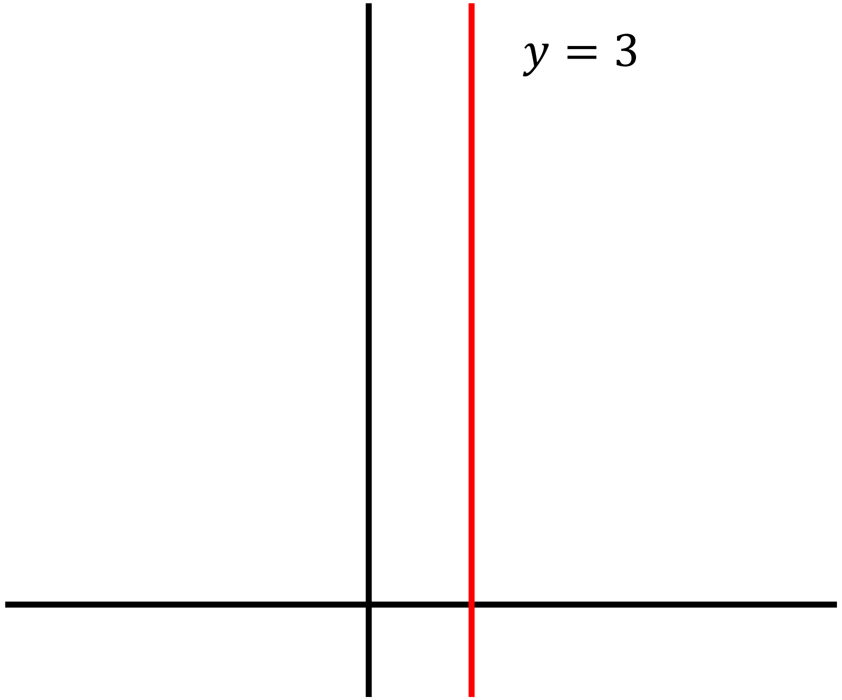
$y = ax + b$

$y = 0.5x + 1$



수직선을 사용하여 분리하면 기울기가 무한히 커짐
수직선으로 사용해 분리할 수 없음

$y = 3$

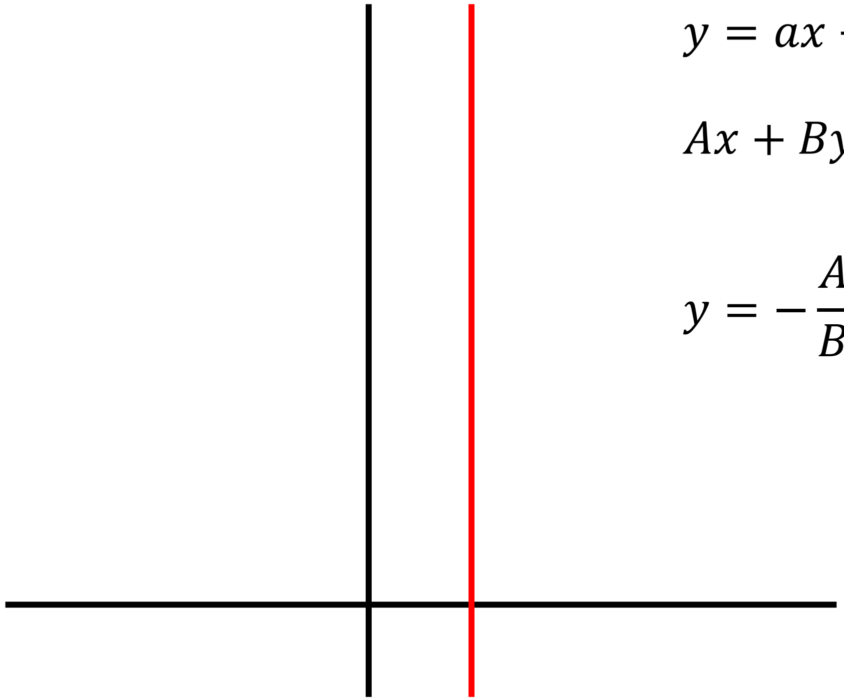


Support Vector Machine

$$y = ax + b$$

$$w^T x + b = 0$$

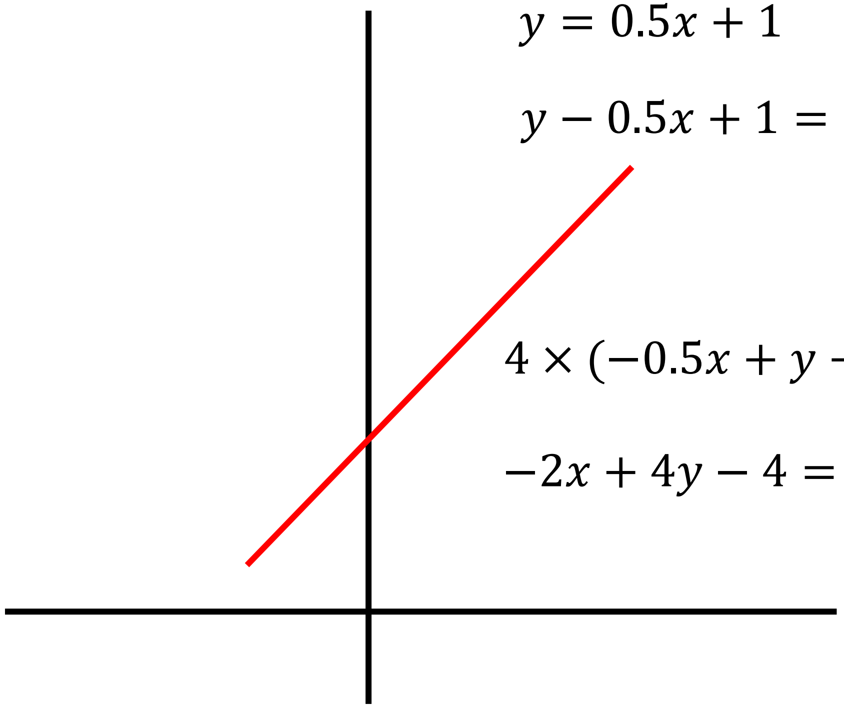
해당 방적식을 일반형식으로 변환하면



$$y = ax + b$$

$$Ax + By + C = 0$$

$$y = -\frac{A}{B}x - \frac{C}{B}$$



$$y = 0.5x + 1$$

$$y - 0.5x + 1 = 0 \qquad -0.5x + y + 1 = 0$$

$$4 \times (-0.5x + y + 1) = 0 \times 4$$

$$-2x + 4y - 4 = 0$$

Support Vector Machine

$$y = ax + b$$

$$\Downarrow$$

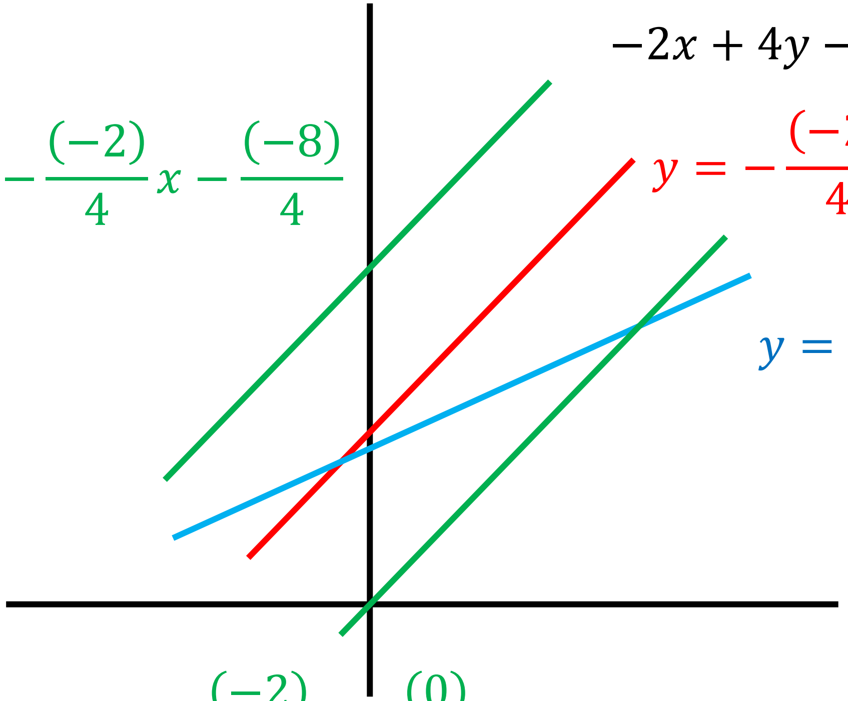
$$w^T x + b = 0$$

SVM도 이러한 원리와 마찬가지로 최적의 a,b를 찾음

$$-2x + 4y - 4 = 0$$

$$y = -\frac{(-2)}{4}x - \frac{(-8)}{4}$$

$$y = -\frac{(-2)}{4}x - \frac{(-4)}{4}$$



$$y = -\frac{(-1)}{4}x - \frac{(-4)}{4}$$

$$y = -\frac{(-2)}{4}x - \frac{(0)}{4}$$

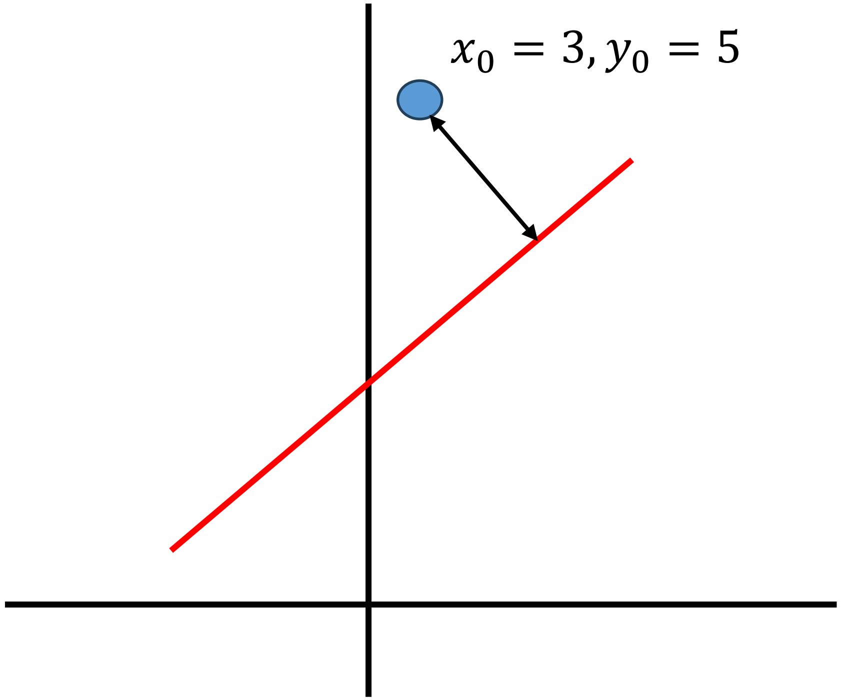
$$-2x + 4y - 4 = 1$$
$$-2x + 4y - 4 = 0$$
$$-2x + 4y - 4 = -1$$

Support Vector Machine

$$y = ax + b$$

$$\Downarrow$$

$$w^T x + b = 0$$



$$Ax + By + C = 0$$

$$-2x + 4y - 4 = 0$$

$$d = \frac{|Ax_0 + By_0 + C|}{\sqrt{A^2 + B^2}}$$

$$d = \frac{|(-2) * 3 + 4 * 5 + (-4)|}{\sqrt{(-2)^2 + 4^2}} = \frac{10}{\sqrt{20}} = 2.236$$

01 지도학습

Support Vector Machine

$$y = ax + b$$

$$\Downarrow$$

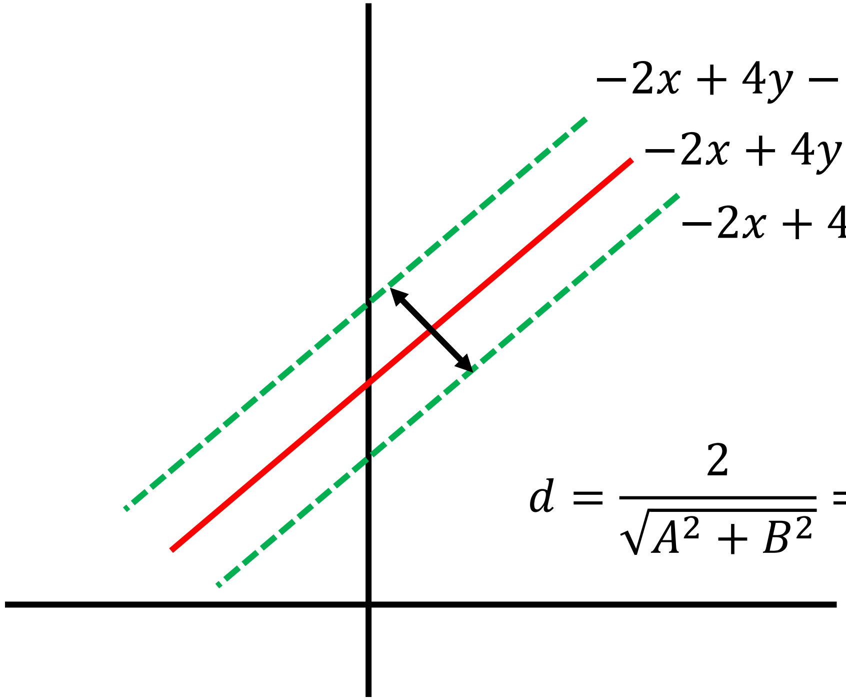
$$w^T x + b = 0$$

$$Ax + By + C = 0$$

$$-2x + 4y - 4 = 1$$

$$-2x + 4y - 4 = 0$$

$$-2x + 4y - 4 = -1$$

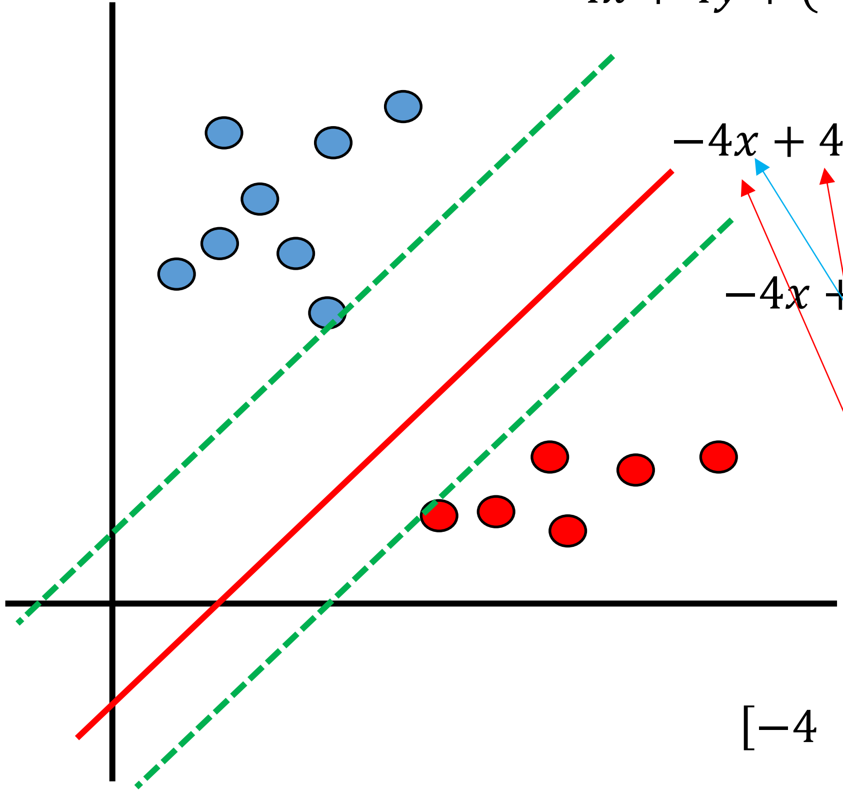


$$d = \frac{2}{\sqrt{A^2 + B^2}} = \frac{2}{||w||}$$

$$-4x + 4y + (-4) = 1$$

$$-4x + 4y + 4 = 0$$

$$-4x + 4y + 12 = -1$$



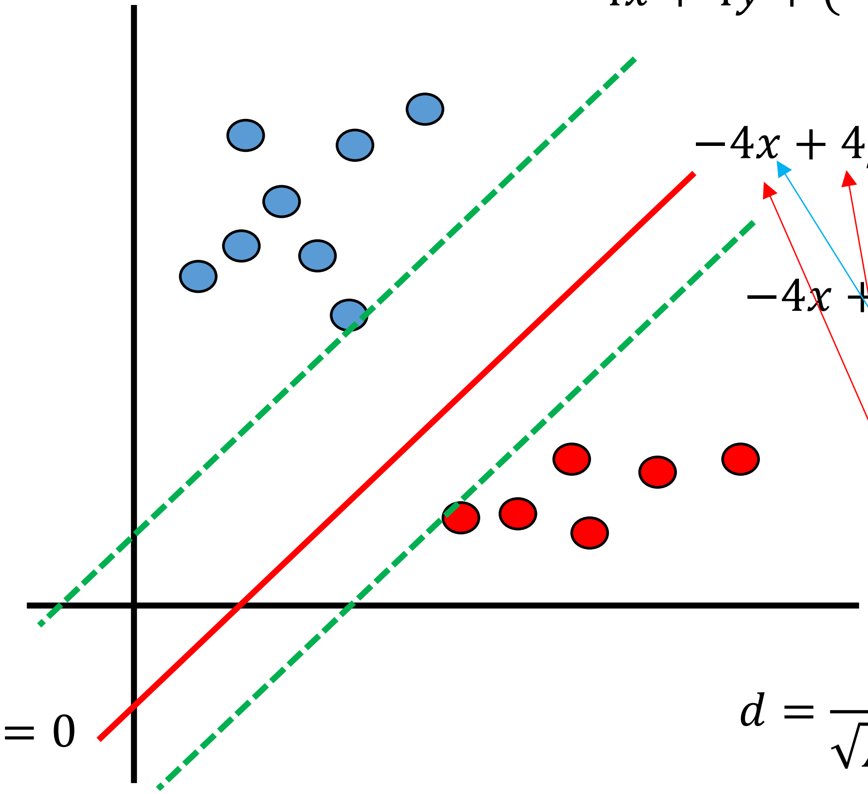
$$\boxed{w}^T \boxed{x} + b = 0$$

$$\begin{bmatrix} -4 & 4 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} + b = 0$$

Support Vector Machine

$$y = ax + b$$

$$w^T x + b = 0$$



$$-4x + 4y + (-4) = 1$$

$$-4x + 4y + 4 = 0$$

$$-4x + 4y + 12 = -1$$

$$\boxed{w}^T \boxed{x} + b = 0$$

$$w^T x + b = 1$$

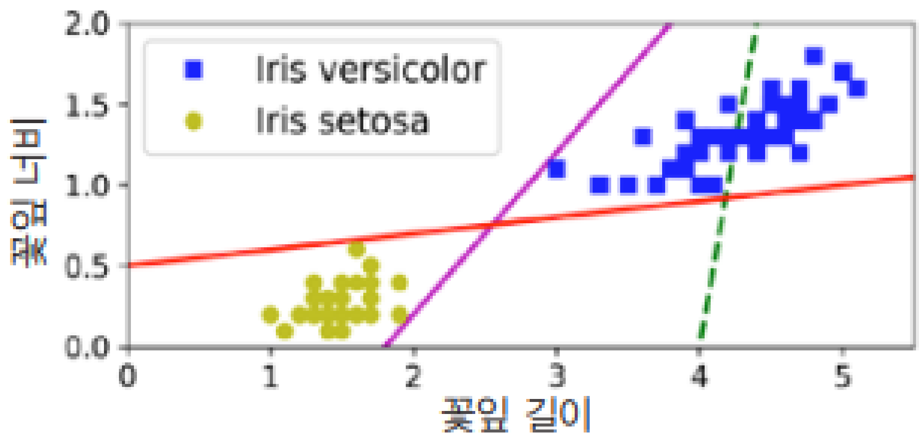
$$w^T x + b = 0$$

$$w^T x + b = -1$$

$$d = \frac{2}{\sqrt{A^2 + B^2}} = \frac{2}{||w||}$$

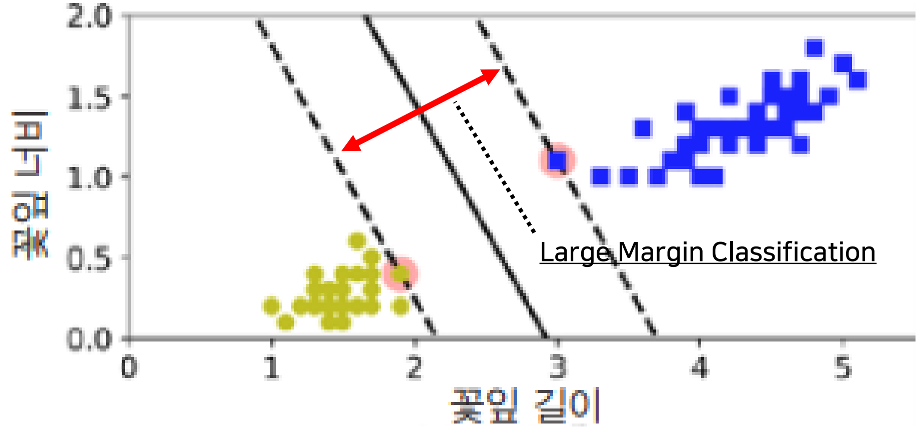
$$max \frac{2}{||w||} \; such\ that\ w^T x + b \begin{array}{l} \geq 1\ if\ y = +1 \\ \leq -1\ if\ y = -1 \end{array}$$

Support Vector Machine - Large Margin Classification



3개의 선형 결정 경계를 만든 모델

- 초록색 점선 결정 경계는 분류 X
- 나머지 2개의 결정 경계는 분류 O
- 하지만, 결정 경계가 샘플에 너무 가까워 새로운 샘플에 잘 작동하지 않을 것



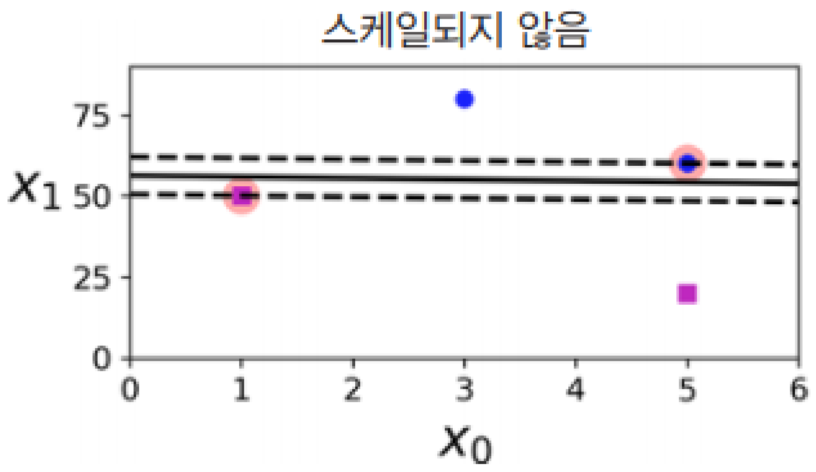
SVM 분류기

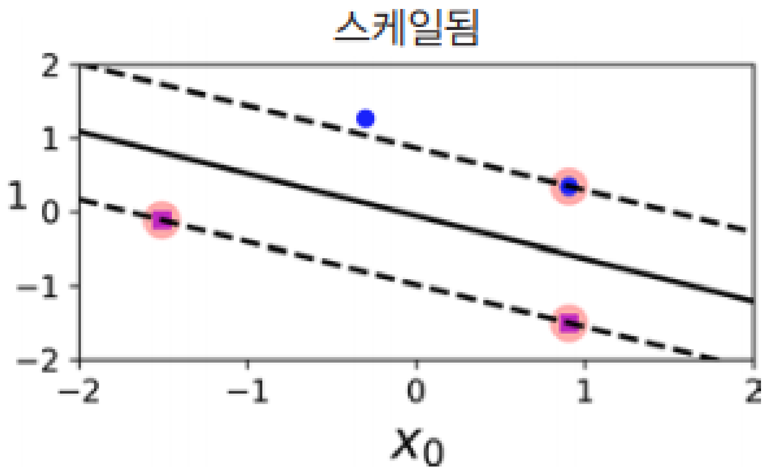
- 실선 = SVM 분류기의 결정 경계
- 가장 가까운 훈련 샘플로부터 멀리 떨어져 있음

→ Support Vector와 수직 거리로 가장 먼 거리를 "Large Margin"이라고 함

Support Vector Machine - Large Margin Classification

- **SVM은 특성의 스케일에 민감!**
- X_1 **이** X_0 **보다 훨씬 큰 값일 경우, Large Margin은 거의 수평**
- **따라서** X_1'**의 스케일을** X_0 **과 비슷하게 조정하여 분류기에 적합한 결정 경계를 설정**





01 지도학습

Support Vector Machine - Soft Margin Classification

Hard Margin Classification 이란?

- 모든 샘플이 Margin 바깥쪽에 올바르게 분류되어 있는 것 → 선형 SVM

Hard Margin Classification의 문제점

- 데이터가 선형적으로 구분될 수 있어야 제대로 작동
- 이상치에 민감

- Hard Margin Classification의 문제점을 해결하기 위해 Margin 폭을 넓게 유지하는 것과 마진 오류 사이에 적절한 균형 필요
- 이를 "Soft Margin Classification"라고 함

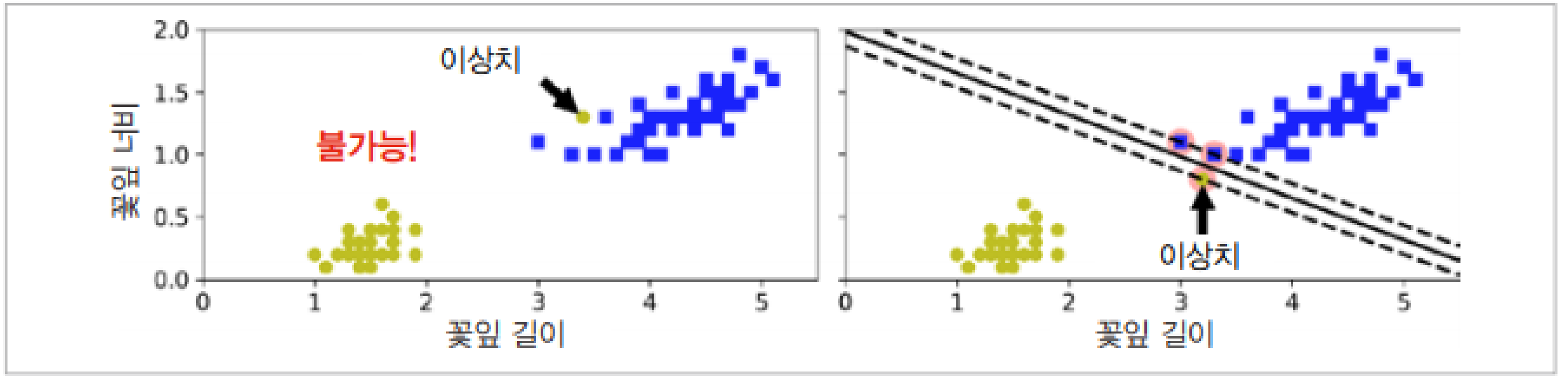
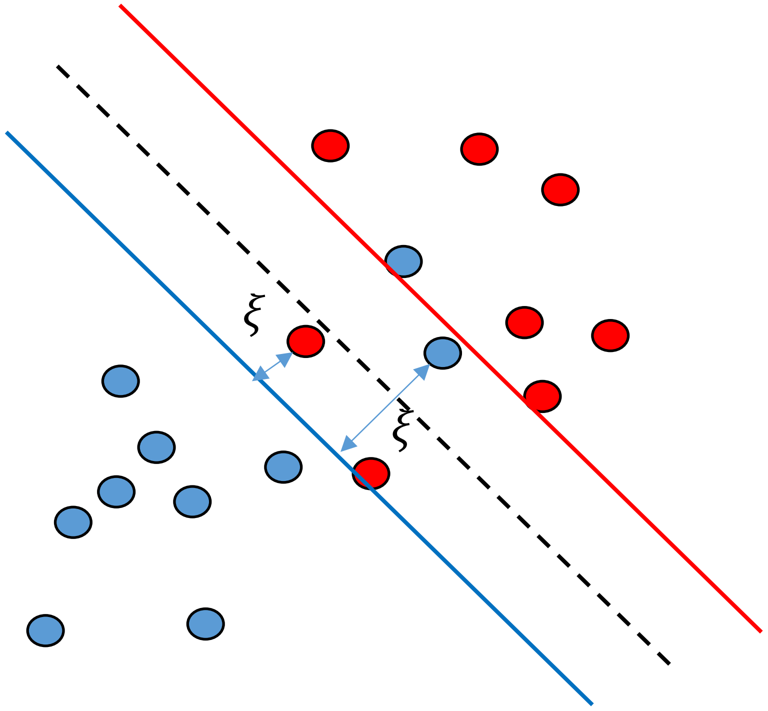


그림 5-3 이상치에 민감한 하드 마진

Support Vector Machine - Soft Margin Classification

Soft Margin Classification -Hyperparameter

- 어느정도의 Error는 이해해주자
- ξ(크사이)를 허용하자

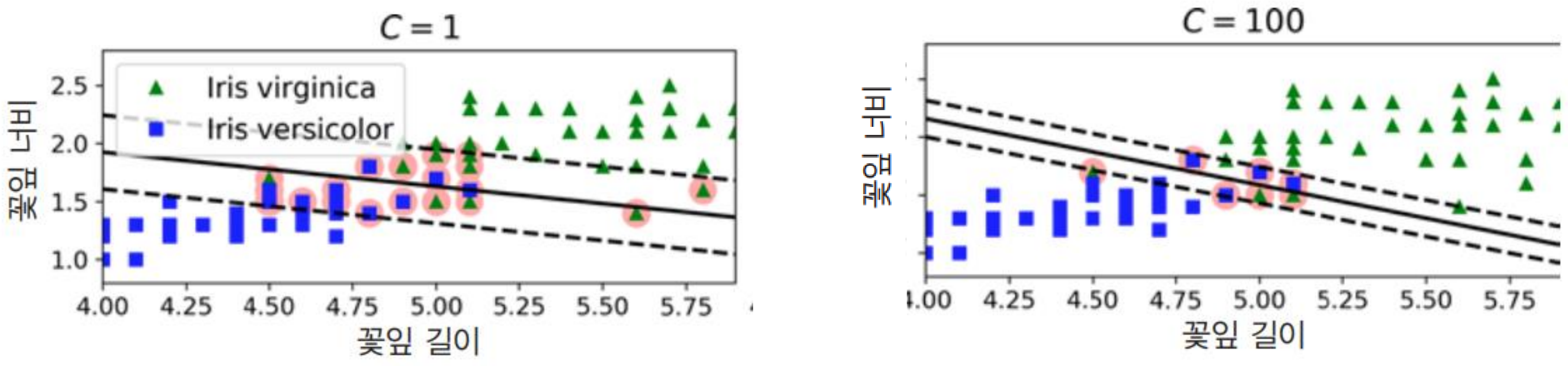


Support Vector Machine - Soft Margin Classification

Soft Margin Classification -Hyperparameter

- 어느정도의 Error는 이해해주자
- ξ(크사이)를 허용하자
- Regularization Parameter C값이 클 수록 에러를 많이 허용함(마진의 범위가 커지므로)

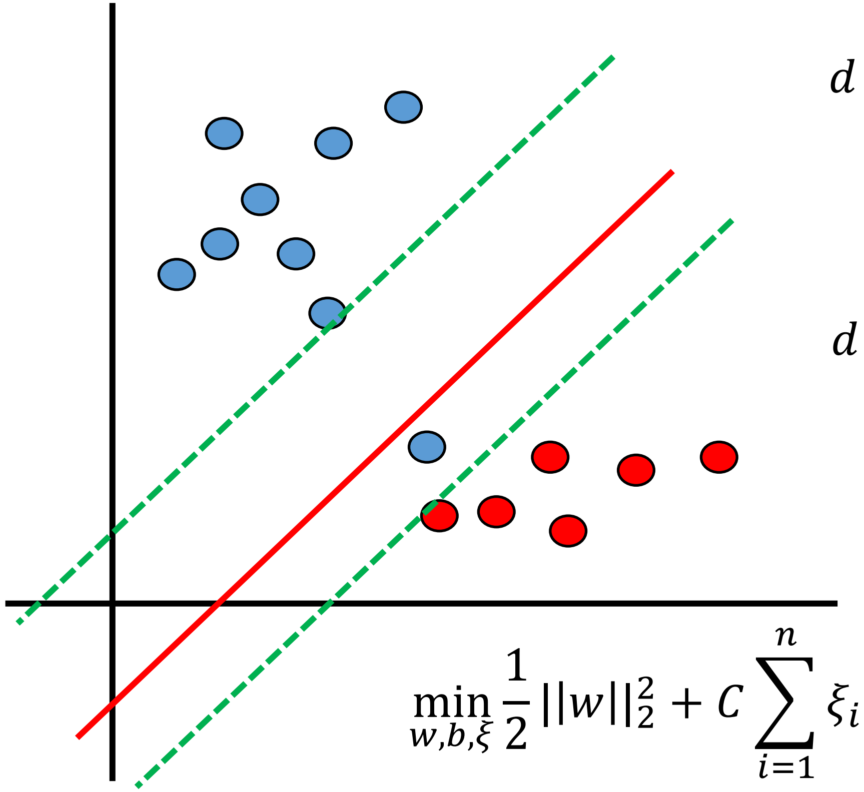
$$\min_{w,b,\xi} \frac{1}{2}||w||_2^2 + C\sum_{i=1}^{n}\xi_i$$



Support Vector Machine - Soft Margin Classification

$$y = ax + b$$

$$w^T x + b = 0$$



$$\min_{w,b,\xi} \frac{1}{2} ||w||_2^2 + C \sum_{i=1}^{n} \xi_i$$

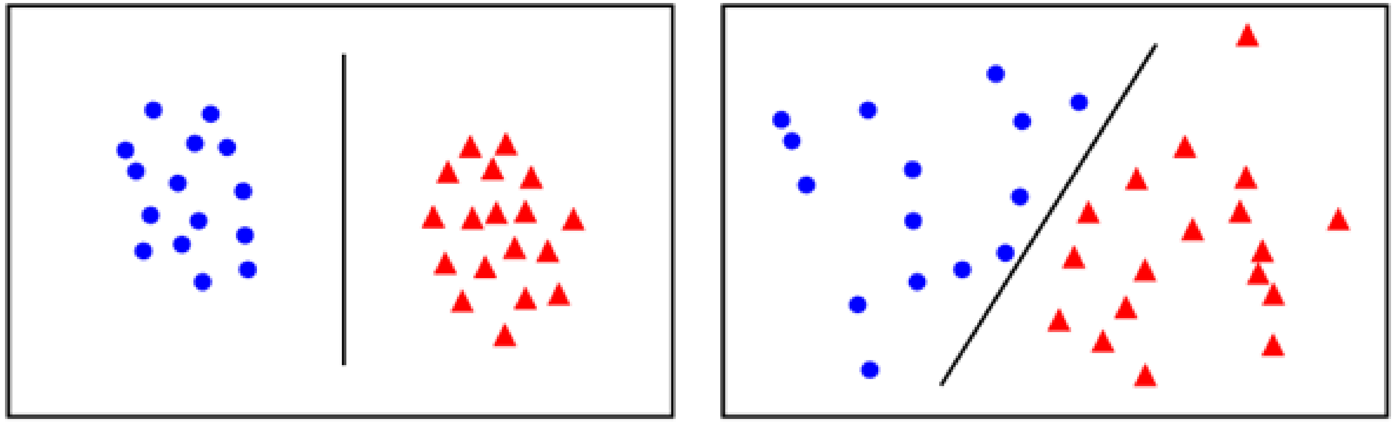
$$d = \frac{2}{\sqrt{A^2 + B^2}} = \frac{2}{||w||}$$

$$d = \frac{\xi}{||w||}$$

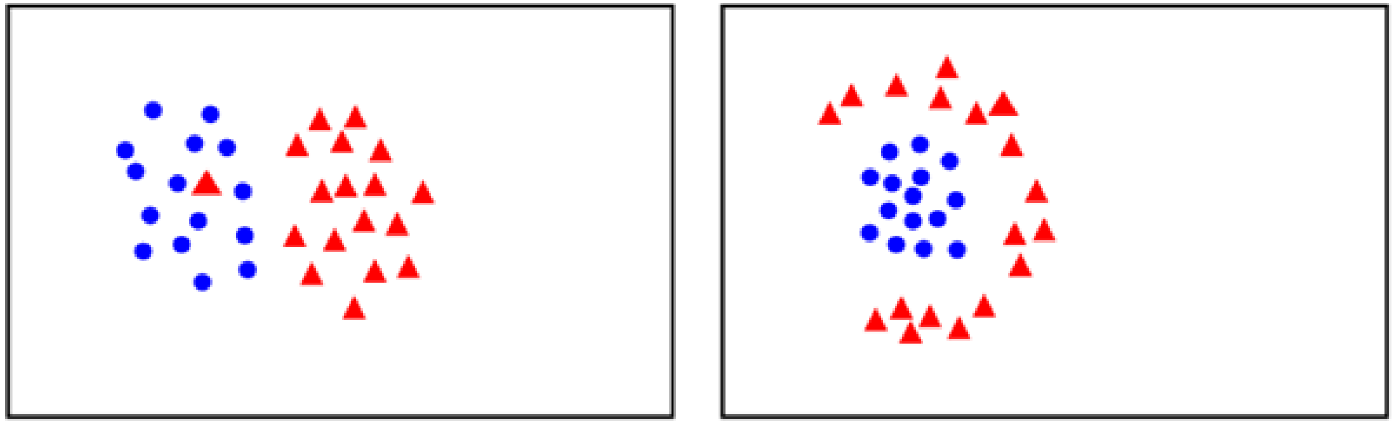
01 지도학습

Support Vector Machine - Soft Margin Classification

linearly
separable



not
linearly
separable

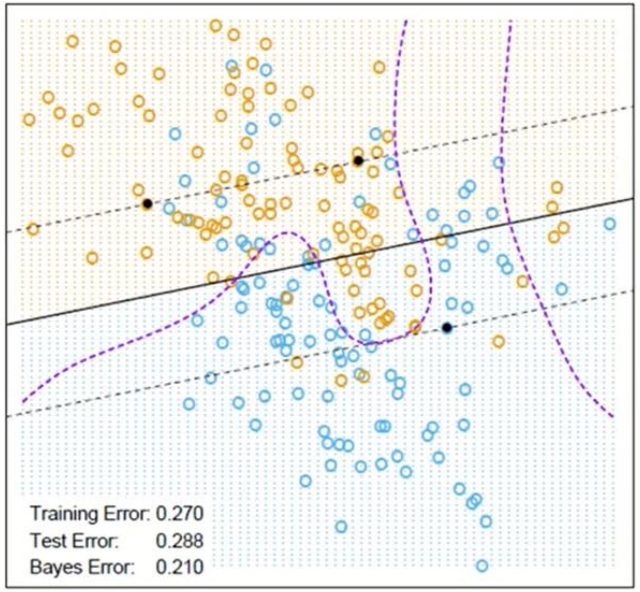


Support Vector Machine - Soft Margin Classification

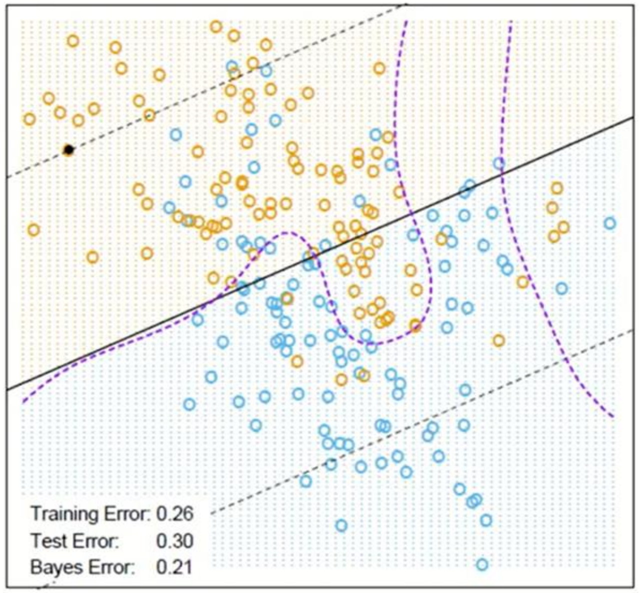
- Soft Margin Classification: 마진오류와 margin 폭을 넓게 유지해야 함

SVM 모델에서 Cost(C) hyperparameter 조절하여 Soft Margin 분류 진행

- Cost(C) hyperparameter 을 사용하여 조절
- C=0.01 일 경우, 마진은 넓어졌지만 마진 안에 샘플이 많이 포함되어 있음 → overfitting
- C=10,000 일 경우, 마진오류는 적어졌지만, 마진이 좁아 짐 → underfitting



$C = 10000$



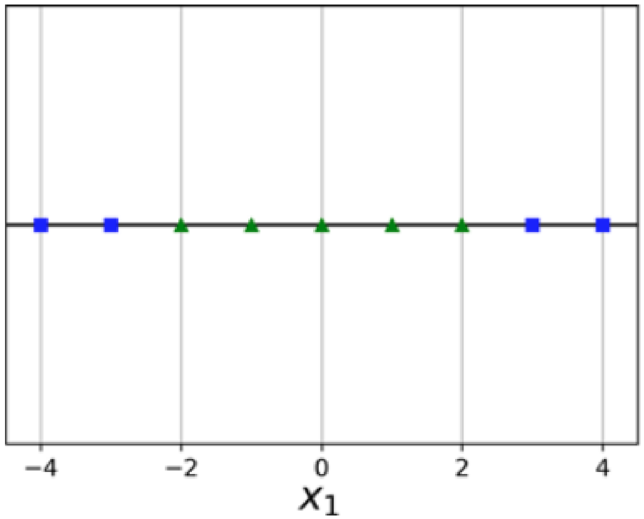
$C = 0.01$

Support Vector Machine – 비선형 SVM 분류방법

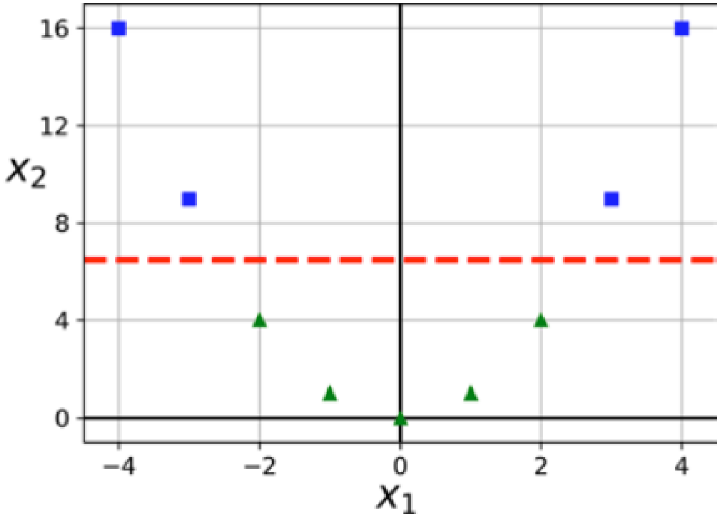
- 일반적인 선형 SVM으로 분류할 수 없고 특정 방법을 사용하여 비선형적인 dataset에 대한 최적의 초평면을 찾아 분류하는 것

1) 다항 특성 추가

➢ 다항 특성과 같은 특성을 추가하고 선형 SVM을 이용하여 분류



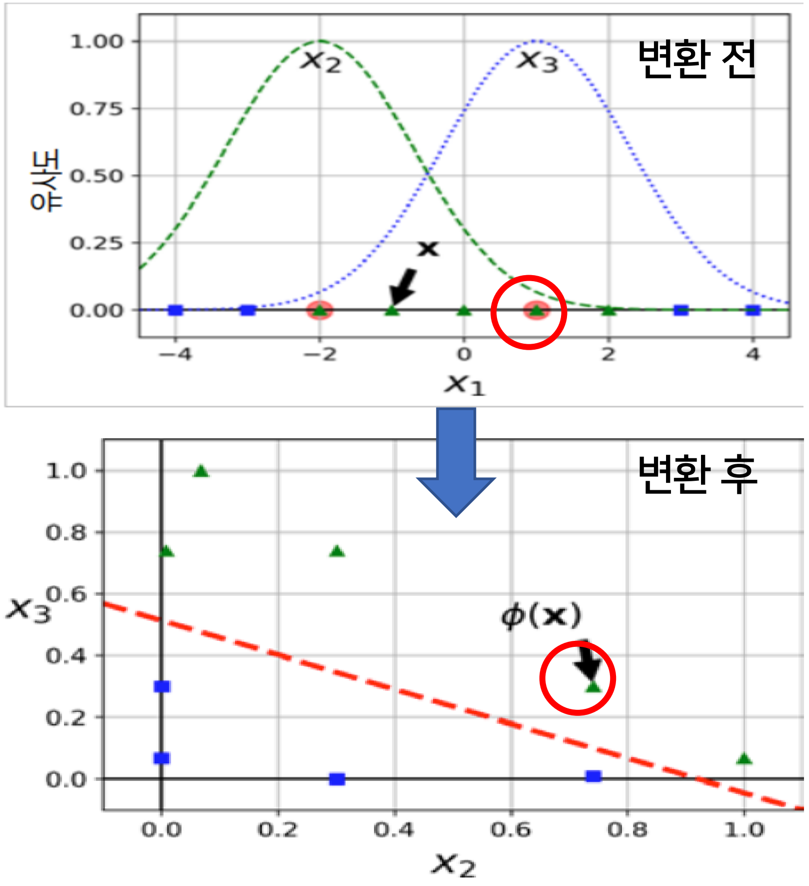
→ 하나의 특성 X_1 만을 가진 간단한 데이터셋은 선형적으로 구분이 안됨



→ $X_2 = (X_1)^2$ 을 추가하여 만들어진 2차원 dataset은 완벽하게 선형적 구분 가능

Support Vector Machine – 비선형 SVM 분류방법

2) 유사도 특성 (RBF 함수)



- 가우시안 방사 기저 함수(Radial Basis Function)를 활용하여 선형 분류로 변환

1) 1차원 dataset에 2개의 랜드마크 X_2=-2, X_3 =1을 추가

2) γ = 0.3 인 가우시안 방사 기저 함수(RBF)를 유사도 함수로 정의

이 함수 값은 0(랜드마크에서 아주 멀리 떨어진 경우) ~ 1(랜드마크와 같은 위치일 경우)

까지 변화하며 종 모양으로 표현

- RBF 식: $\phi_\gamma(\mathbf{x}, \ell) = \exp\left(-\gamma \|\mathbf{x} - \ell\|^2\right)$

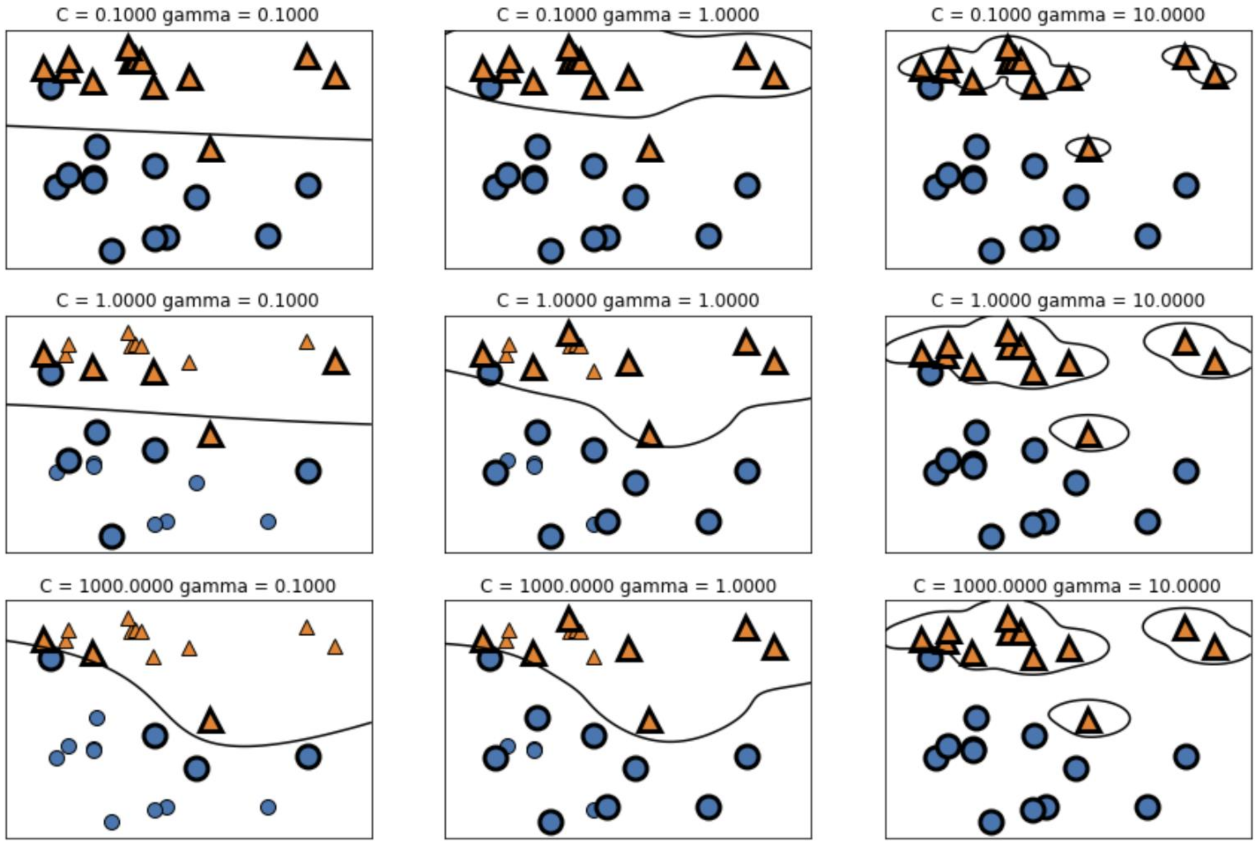
3) ex. 빨간색 표시된 $X_1 = -1$ 은 RBF에 대입하면 X_2=0.74, X_3 =0.3

따라서 본래의 특성은 없어지고 변환 후 그래프에서 볼 수 있듯이

선형적으로 구분이 가능

Support Vector Machine – 비선형 SVM 분류방법

C와 Gamma에 대한 적절한 수치로 모델을 학습하는 것이 매우 중요함



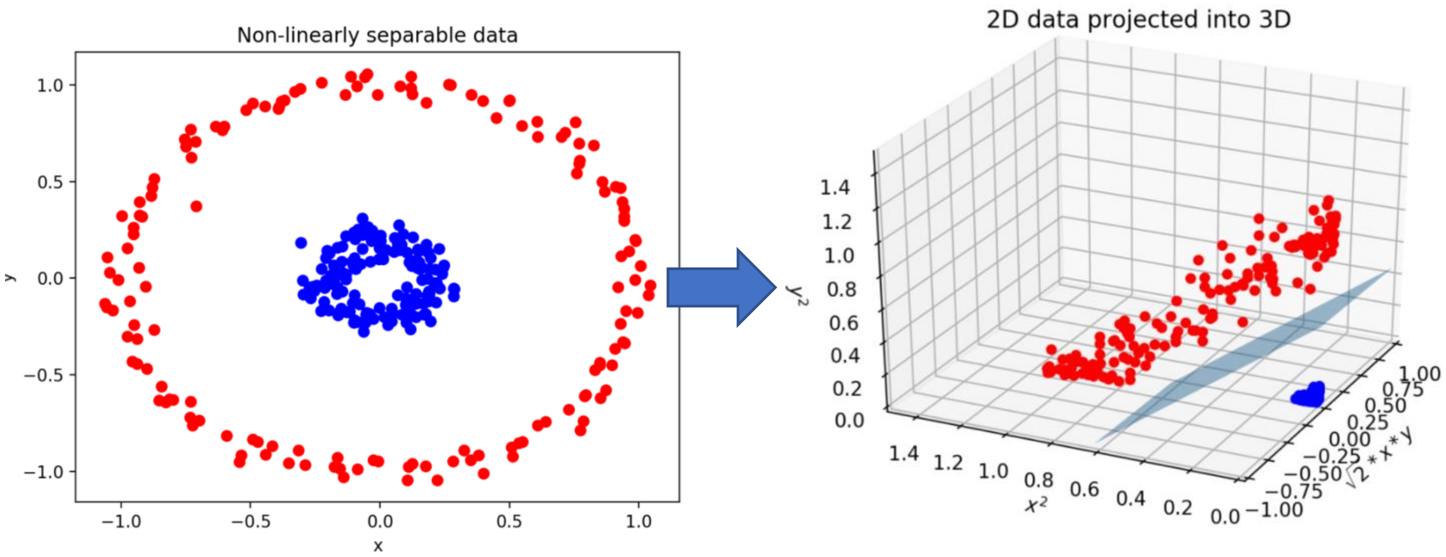
Support Vector Machine – 비선형 SVM 분류방법

3) 다항식 커널

- **다항 특성 추가 방법의 단점을 해결하고자 Kernel trick 이라는 수학적 기교를 적용함**

  ***Kernel trick은 실제로는 특성을 추가하지 않으면서 다항식 특성을 많이 추가한 것과 같은 결과를 얻을 수 있음**

- **단순 선형으로 해결되지 않을 경우 다항식 커널을 사용하면, 2차원에서 3차원으로 표현됨 (더 높은 차수도 가능)**
- **데이터를 더 높은 차원으로 변형하여 나타냄으로써 hyperplane의 결정 경계를 얻을 수 있음**



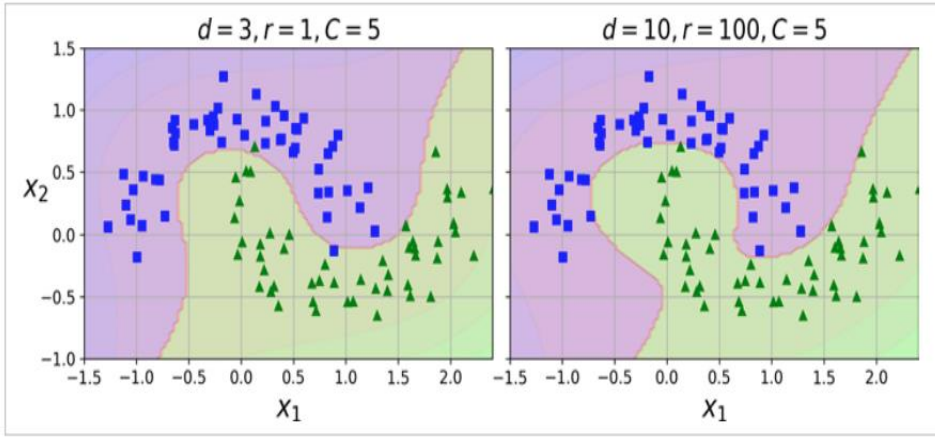


그림 5-7 다항식 커널을 사용한 SVM 분류기

Support Vector Machine

서포트벡터머신

```
library(e1071)  # SVM 모델
library(caret)

# 데이터 불러오기
df <- read.csv("UniversalBank.csv")

# 종속변수에 대한 데이터 요소로 변환
df$Personal.Loan <- ifelse(df$Personal.Loan == 1, "High", "Low")
df$Personal.Loan <- as.factor(df$Personal.Loan)

#데이터 나누기 학습 80%, 테스트 20%
trainIndex <- createDataPartition(df$Personal.Loan, p = 0.8, list = FALSE)
train_data <- df[trainIndex,]
test_data <- df[-trainIndex,]
```

서포트벡터머신

```
cost_value <- 1      # Regularization parameter C
gamma_value <- 0.1   # Gamma for RBF kernel

# SVM모델 생성
svm_model <- svm(train_data$Personal.Loan ~ ., data = train_data,
                 type = 'C-classification',
                 kernel = 'radial', #Radial Basis Function
                 cost = cost_value,
                 gamma = gamma_value,
                 probability = TRUE) #확률값으로 도출

summary(svm_model)


type = 'one-classification', kernel = 'radial' # One-classification(특정 클래스만 학습해 이상치 탐지

type = 'eps-regression', kernel = 'radial' # eps-regression(연속형 데이터를 예측하는 회귀문제 사용)
```

서포트벡터머신

```
install.packages("pROC")
library(pROC)

predicted_probs <- predict(svm_model, test_data, probability = TRUE)
predicted_probs <- attr(predicted_probs, "probabilities")[, "High"] #객체의 속성을 추출하는 함수

predicted_classes <- ifelse(predicted_probs > 0.5, "High", "Low")
predicted_classes <- as.factor(predicted_classes)
predicted_classes

conf_matrix <- confusionMatrix(predicted_classes, test_data$Personal.Loan)
print(conf_matrix)

roc_curve <- roc(test_data$Personal.Loan, predicted_probs)
plot(roc_curve)
auc(roc_curve)
```