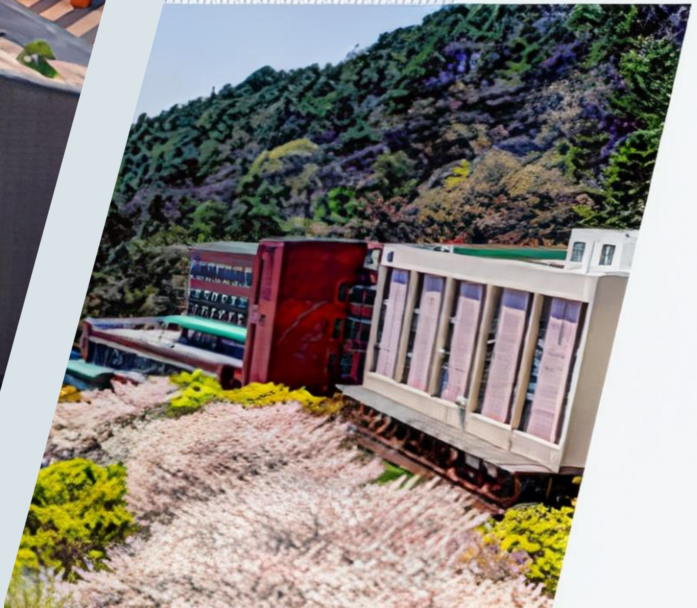
Dong-A Univ. (ISPL)

Support Vector Machine - 실습

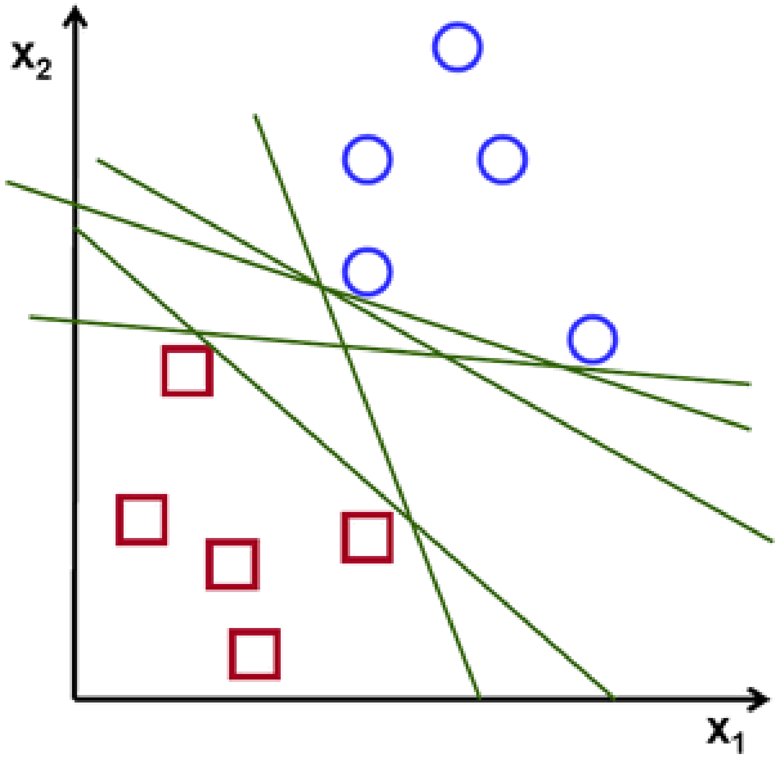
컴퓨터AI공학부

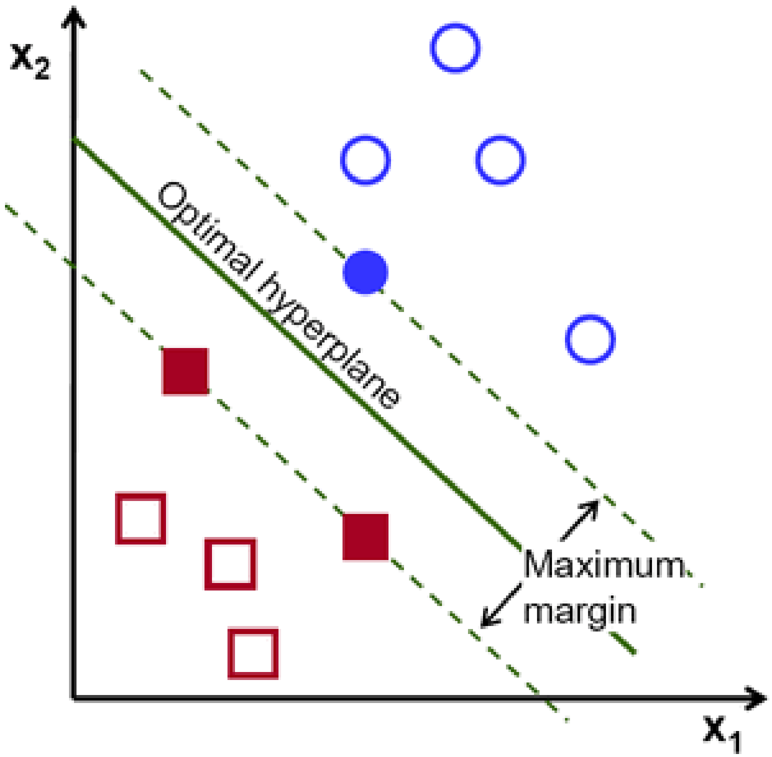
2024년 2학기 머신러닝



Review – Support Vector Machine

- 목적: Margin을 최대화하는 optimal separating hyperplane (decision boundary) 구하기



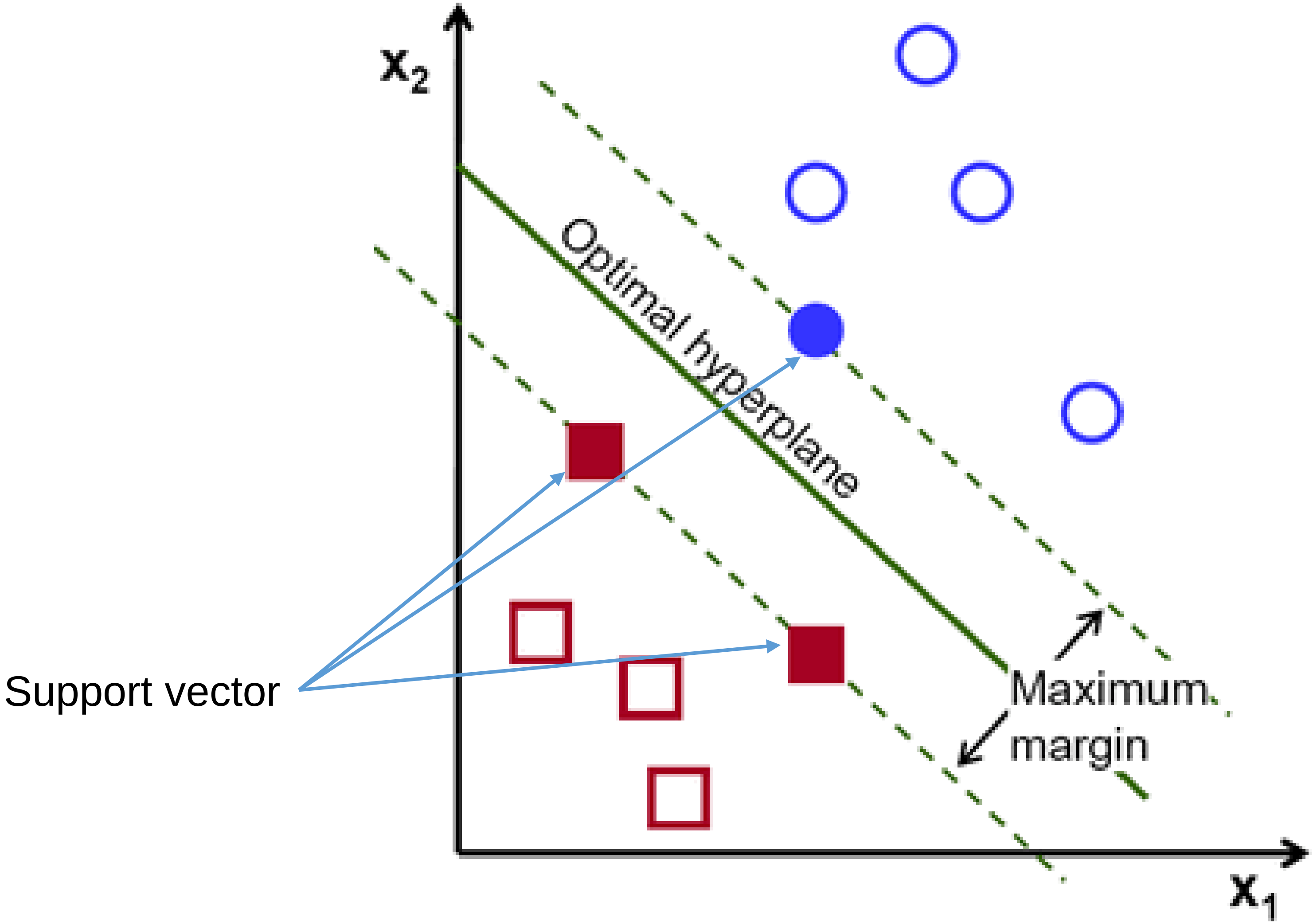


Ref.: https://towardsdatascience.com/

동아대학교
DONG-A UNIVERSITY

Review – Support Vector Machine

- 용어 정리



$$f(X) = W^T X + b$$

Review – Support Vector Machine

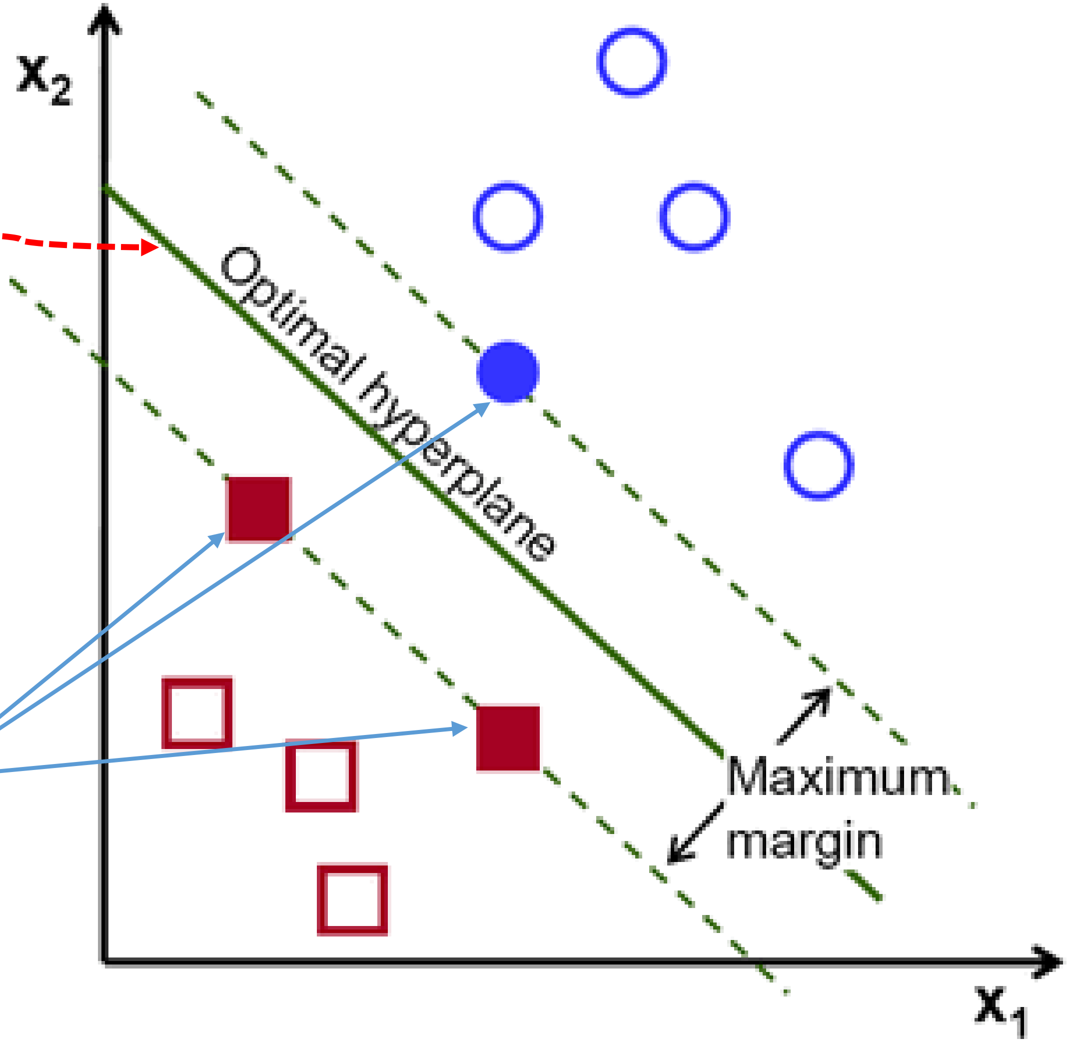
- 용어 정리

$$f(X) = W^T X + b$$

$$W^T X + b = 0$$

Optimal separating hyperplane

Support vector

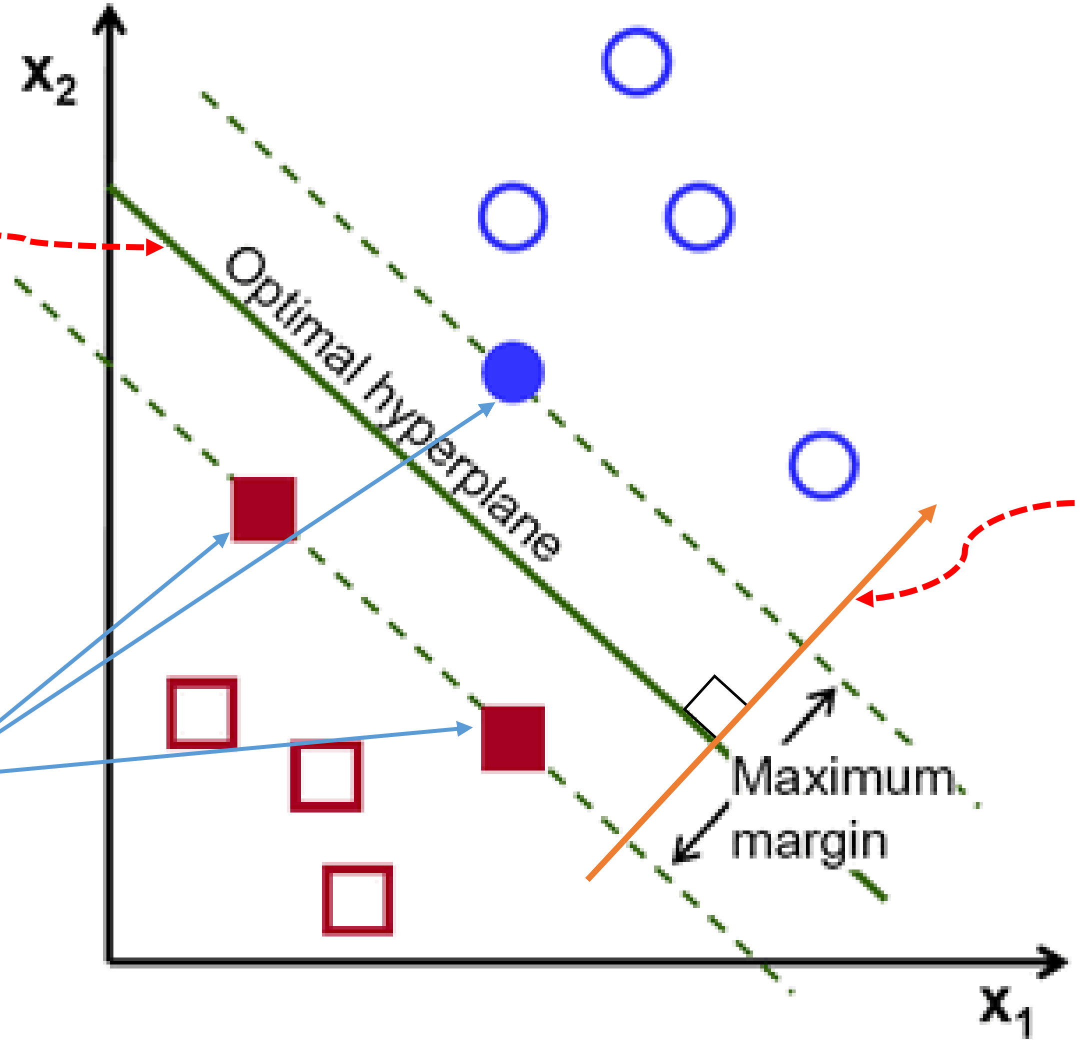


- **용어 정리**

$$f(X) = W^T X + b$$

$$W^T X + b = 0$$

Optimal separating hyperplane

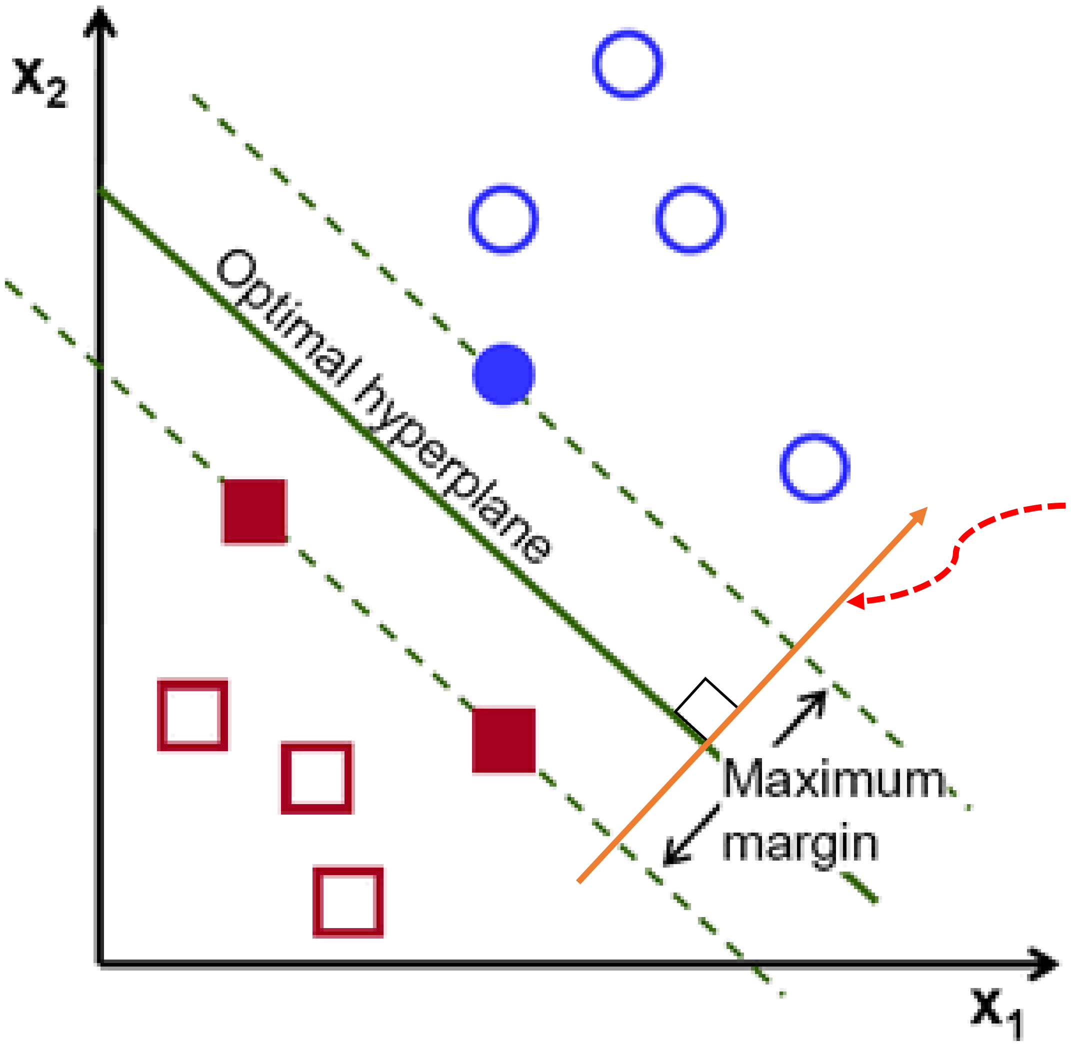


Support vector

$$\vec{W} \begin{cases} \text{Weight (vector)} \\ \text{Normal vector} \end{cases}$$

Review – Support Vector Machine

- 용어 정리



$\vec{W}$ $\begin{cases} \text{Weight (vector)} \\ \text{Normal vector} \end{cases}$

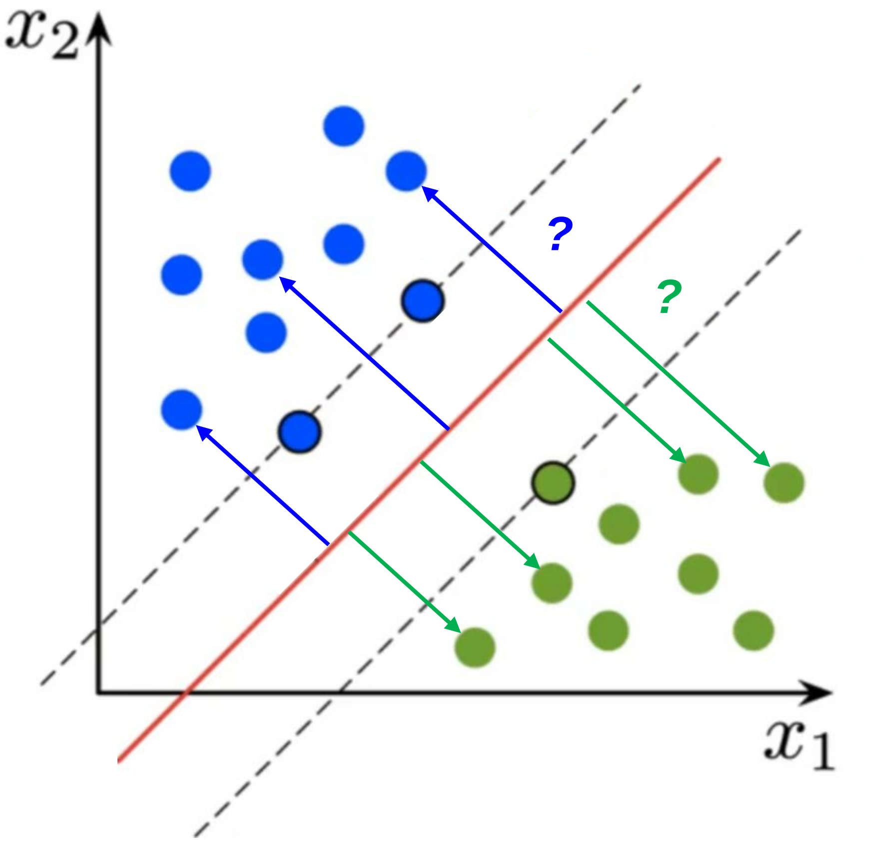
Margin 방향의 unit vector

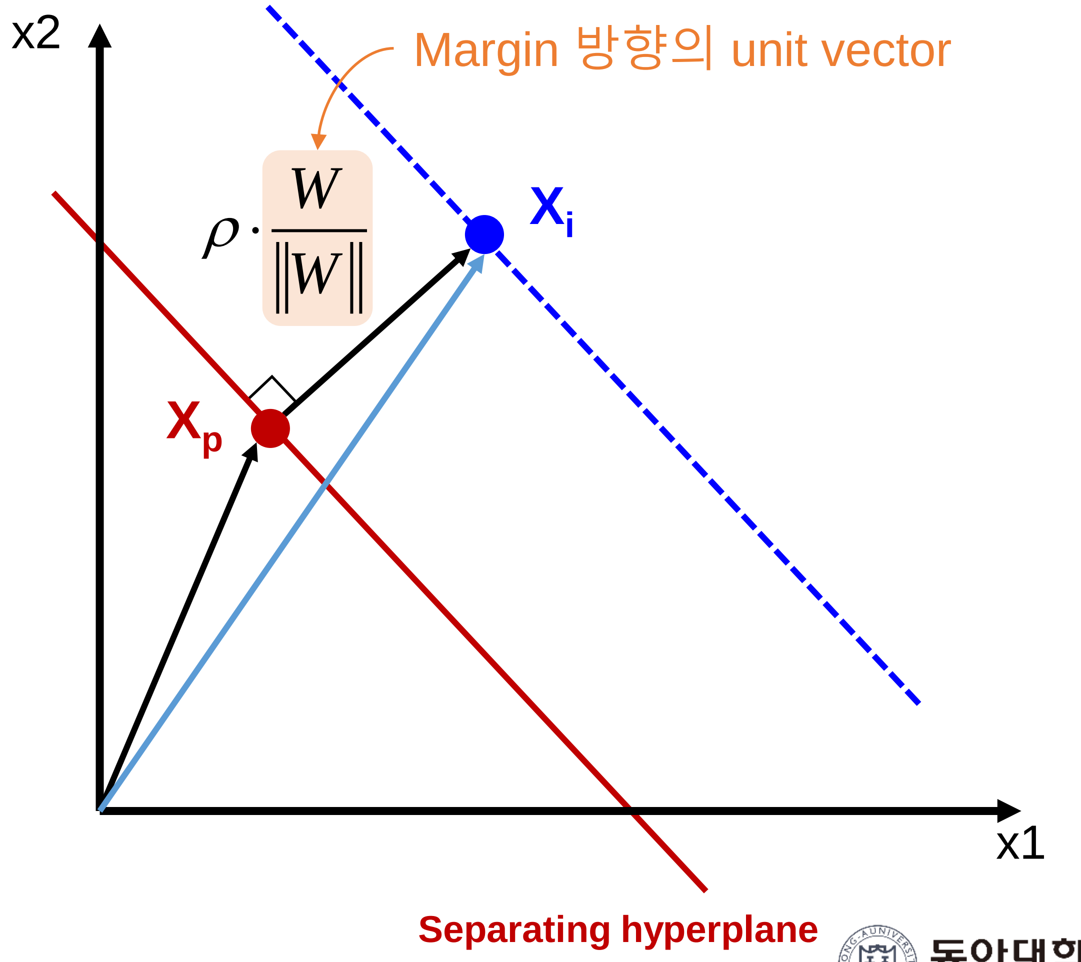
$$\frac{W}{\|W\|}$$

Ref.: https://towardsdatascience.com/

Review – Support Vector Machine

- 임의의 데이터 x_i 에 대해 separating hyperplane과의 거리: ρ

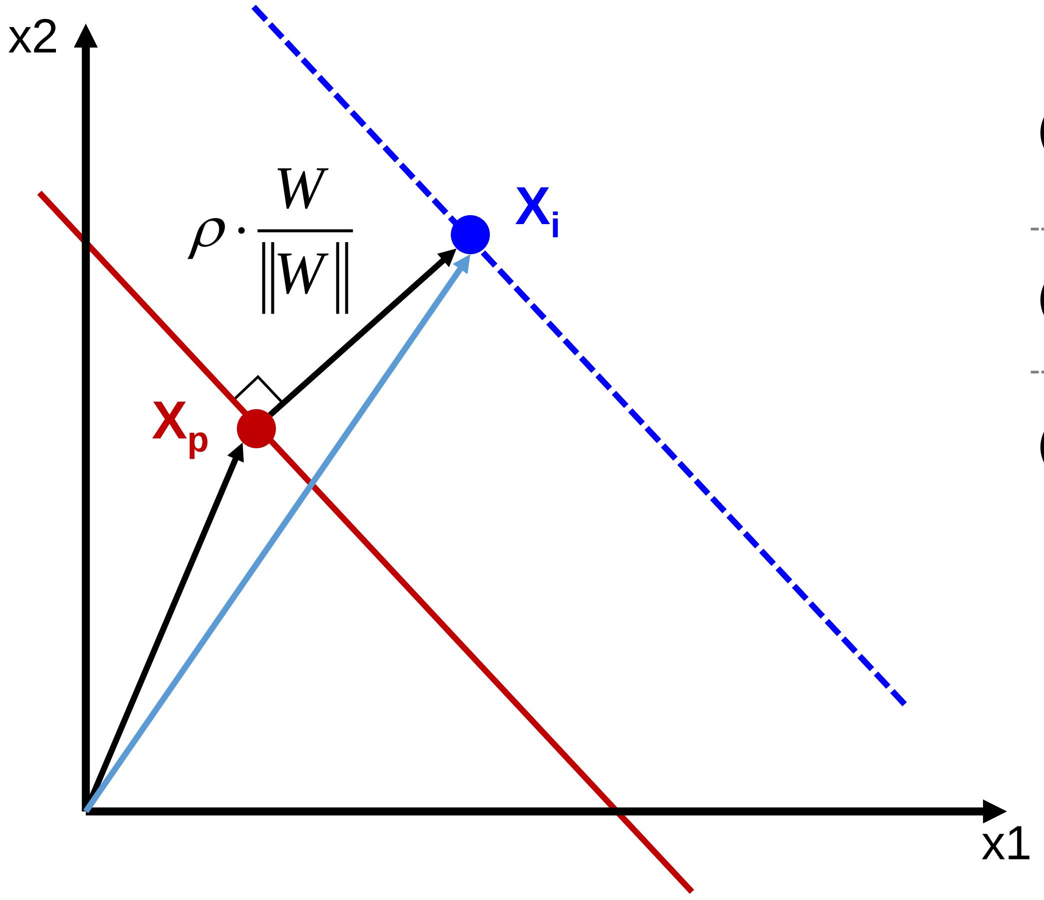


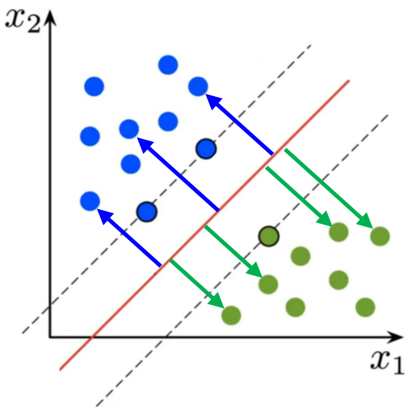


동아대학교 DONG-A UNIVERSITY

Review – Support Vector Machine

- 임의의 데이터 x_i 에 대해 separating hyperplane과의 거리: ρ





1. $$x_i = x_p + \rho \cdot \frac{W}{\|W\|}$$

2. $$f(x_p) = W^T x_p + b = 0$$

3. $$f(x_i) = W^T x_i + b$$
$$= W^T \cdot \left(x_p + \rho \cdot \frac{W}{\|W\|} \right) + b$$
$$= \underbrace{\left(W^T x_p + b \right)}_{0} + \rho \cdot \frac{\|W\|^2}{\|W\|}$$

4. $$\therefore \rho = \frac{f(x)}{\|W\|}$$

동아대학교 DONG-A UNIVERSITY

Review – Support Vector Machine

- **Binary classification**

1 : Optimal separating hyperplane

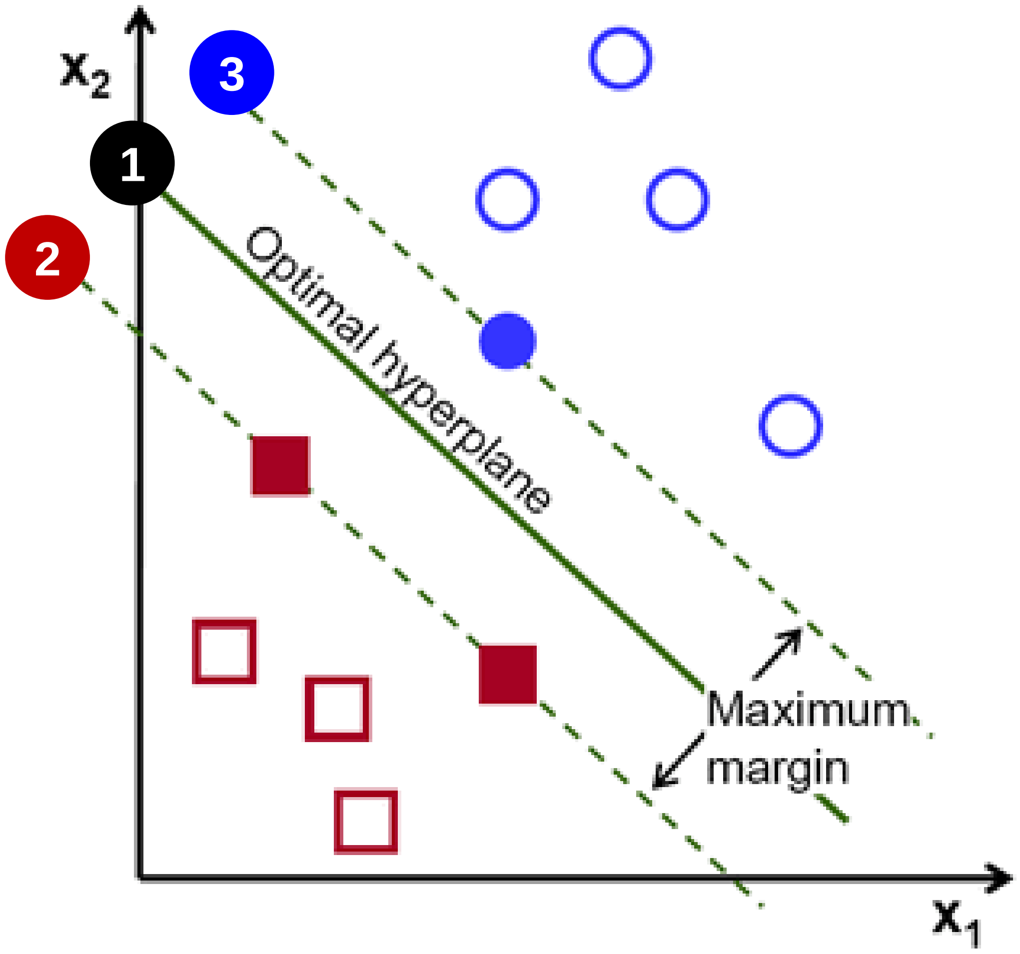
$$f(X) = W^T X + b = 0$$

2 : Support vector (negative)

$$f(X) = W^T X + b = -1$$

3 : Support vector (positive)

$$f(X) = W^T X + b = +1$$



Review – Support Vector Machine

- **Binary classification**

1 : Optimal separating hyperplane

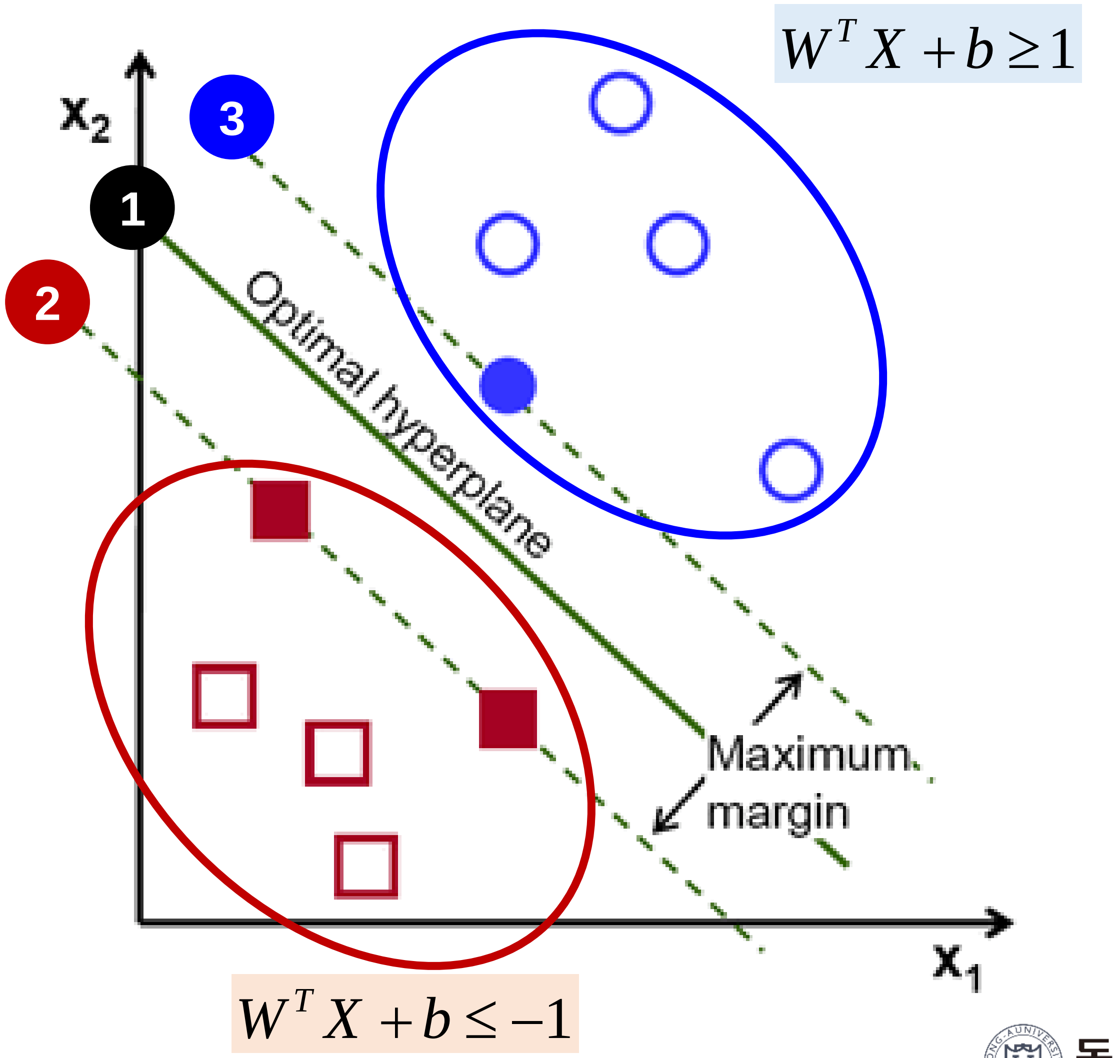
$$f(X) = W^T X + b = 0$$

2 : Support vector (negative)

$$f(X) = W^T X + b = -1$$

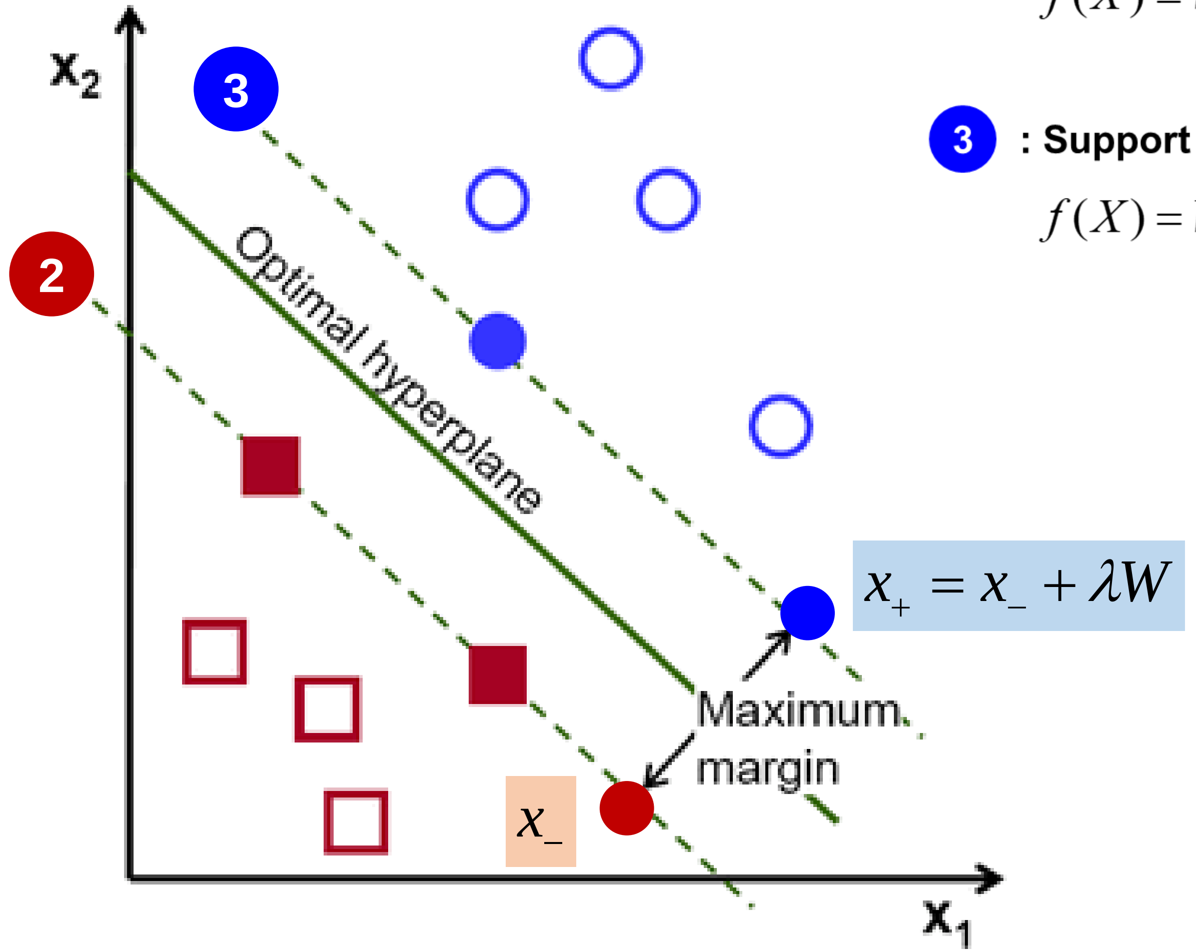
3 : Support vector (positive)

$$f(X) = W^T X + b = +1$$



Review – Support Vector Machine

- **Margin 계산**



2 : **Support vector (negative)**

$$f(X) = W^T X + b = -1$$

3 : **Support vector (positive)**

$$f(X) = W^T X + b = +1$$

$$W^T x_+ + b = 1$$

$$W^T (x_- + \lambda W) + b = 1$$

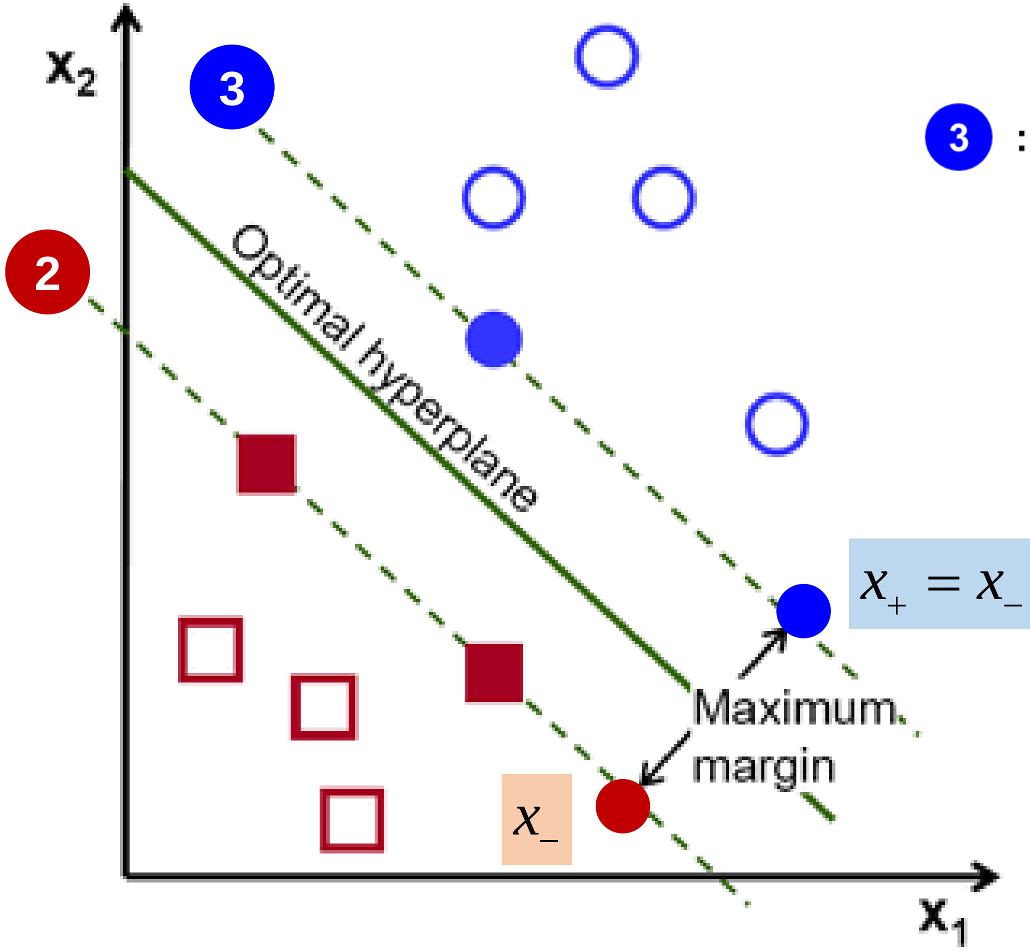
$$\underbrace{W^T x_- + b}_{-1} + W^T \lambda W = 1$$

$$\therefore \lambda = \frac{2}{W^T W}$$

동아대학교 DONG-A UNIVERSITY

Review – Support Vector Machine

- **Margin 계산**



$x_+ = x_- + \lambda W$

2 : **Support vector (negative)**

$$f(X) = W^T X + b = -1$$

3 : **Support vector (positive)**

$$f(X) = W^T X + b = +1$$

$$W^T x_+ + b = 1$$

$$W^T (x_- + \lambda W) + b = 1$$

$$W^T x_- + b + W^T \lambda W = 1$$

$$\therefore \lambda = \frac{2}{W^T W}$$

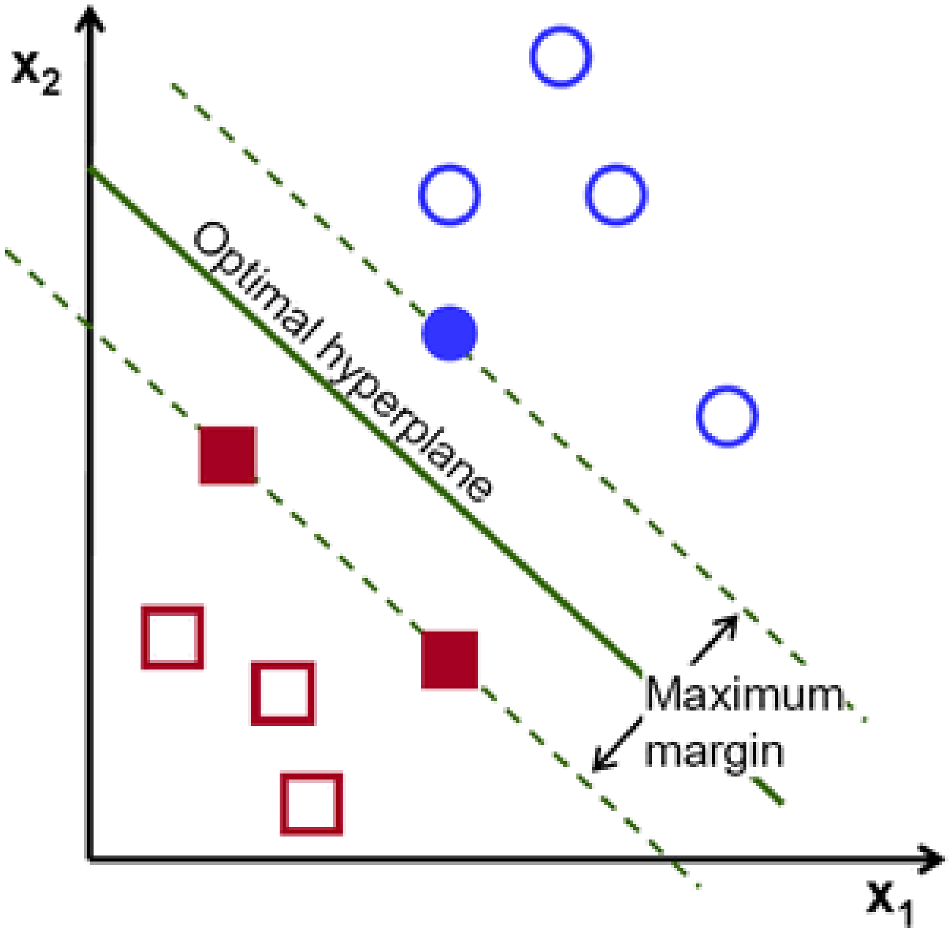
$$\text{Margin} = \text{distance}(x_+, x_-)$$

$$= \|x_+ - x_-\|_2 = \|\lambda W\|_2$$

$$= \frac{2}{W^T W} \cdot \sqrt{W^T W} = \boxed{\frac{2}{\|W\|_2}}$$

[실습] Support Vector Machine

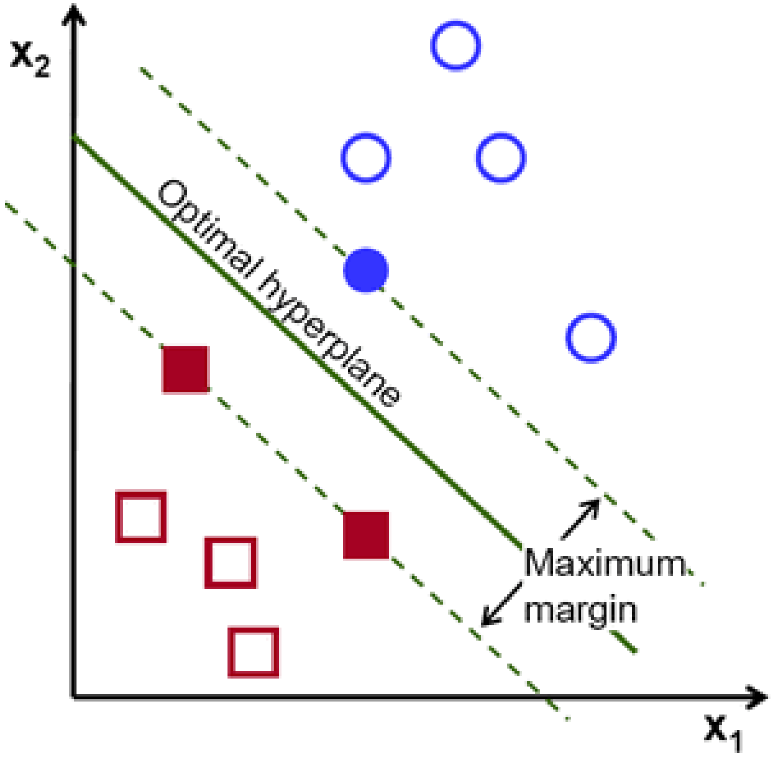
- 목적: Margin을 최대화하는 optimal separating hyperplane (decision boundary) 구하기



$$f(X) = W^T X + b$$

[실습] Support Vector Machine

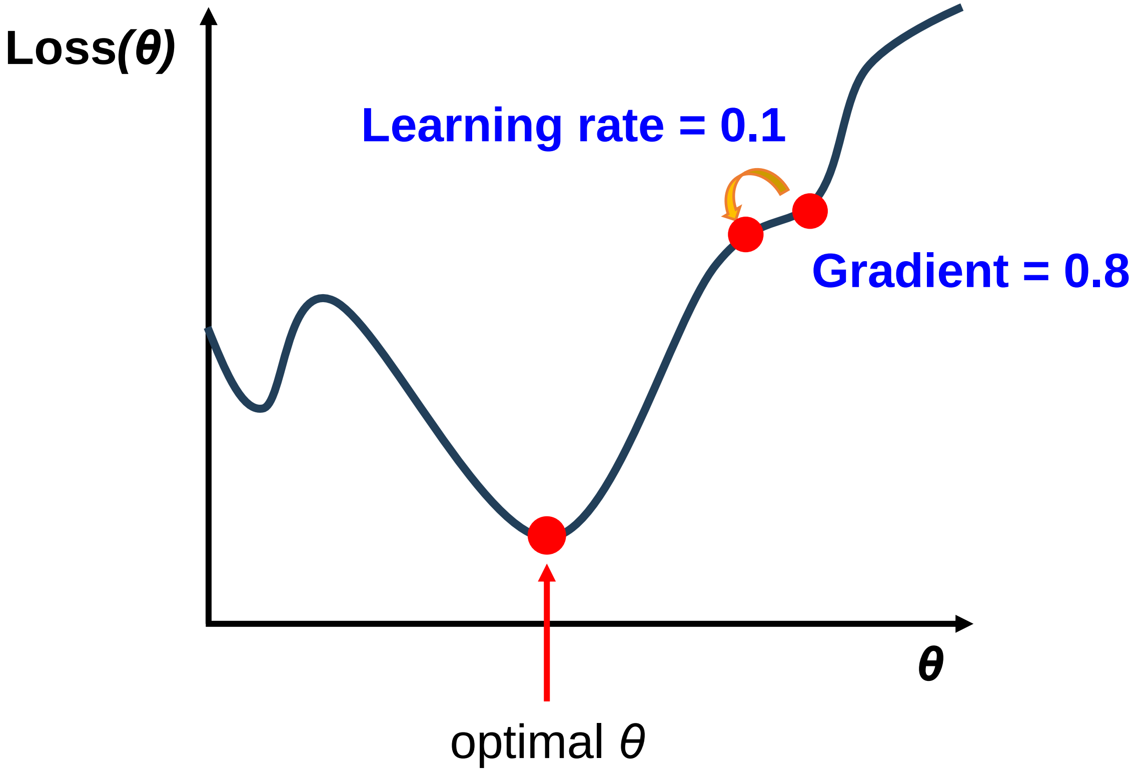
- **목적: Margin을 최대화하는 optimal separating hyperplane (decision boundary) 구하기**
- **Solution**
  - Quadratic Programming (2차 계획법)
  - Gradient Decent Method (GD) → Optimal W, b



$$f(X) = \boxed{W^T} X + \boxed{b}$$

[실습] Support Vector Machine

- **Solution**
  - Quadratic Programming (2차 계획법)
  - Gradient Decent Method (GD) → Optimal W, b



Gradient decent algorithm

① 현재 지점에서 미분을 이용해 gradient 계산

② Gradient에 learning rate를 곱하고 반대 방향으로 weight update

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial L}{\partial \theta_t}$$

$$= \theta_t - 0.08$$

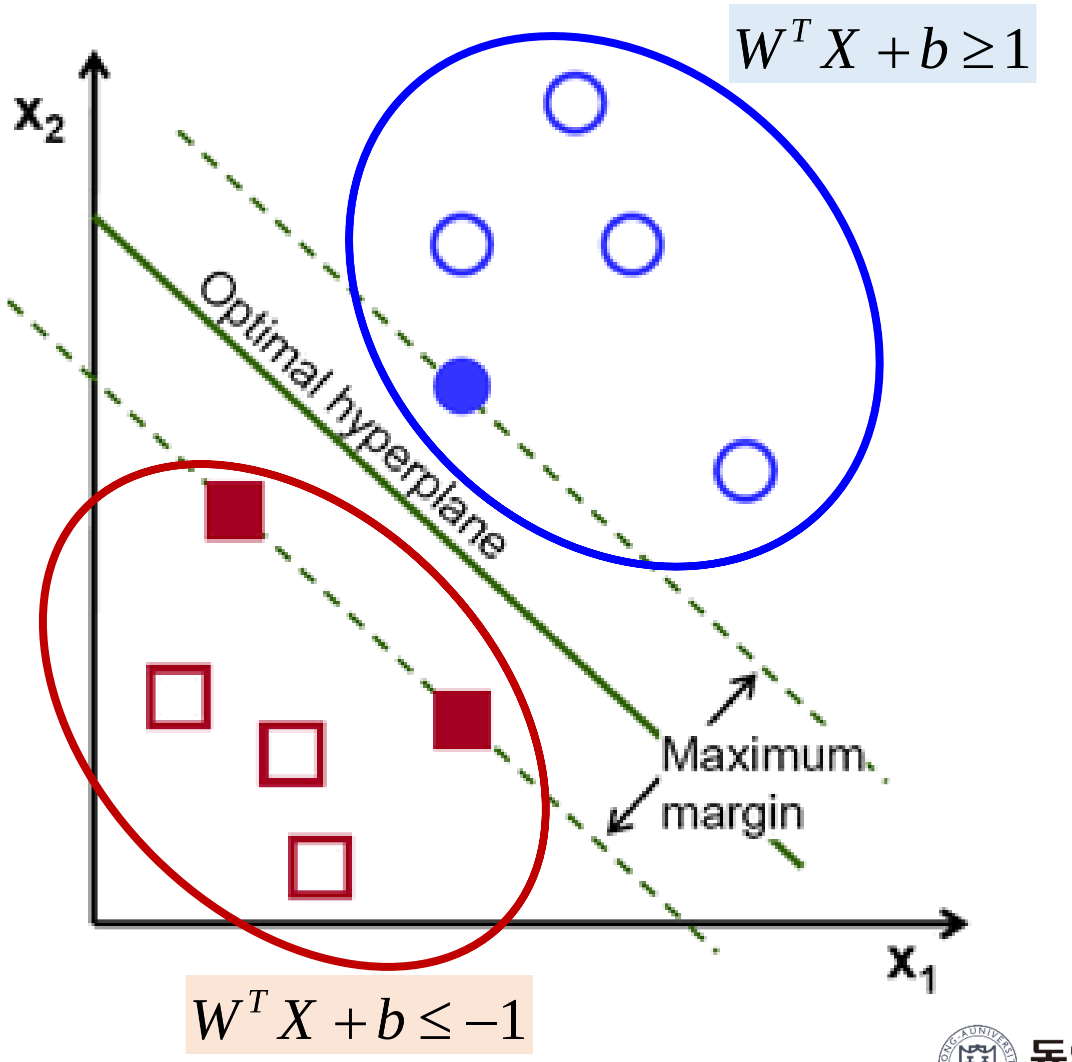
[실습] Support Vector Machine

- **Loss function (Cost function): Hinge loss**

Prediction		**Label**
$W^T X_i + b \geq 1$	→	$y_i = +1$
$W^T X_i + b \leq -1$	→	$y_i = -1$

$$\underbrace{y_i}_{\text{Label}}\underbrace{(W^T x_i + b)}_{\text{Prediction}} \geq 0$$

이 조건을 만족하는 경우 정상적으로 분류 성공



[실습] Support Vector Machine

- Loss function (Cost function): Hinge loss

Prediction		Label
$W^T X_i + b \geq 1$	→	$y_i = +1$
$W^T X_i + b \leq -1$	→	$y_i = -1$

$$\boxed{y_i (W^T x_i + b)} \geq 0$$

Label Prediction

이 조건을 만족하는 경우 정상적으로 분류 성공

Hinge loss

$$Loss = \max(0, 1 - \boxed{y_i (W^T x_i + b)})$$

[실습] Support Vector Machine

- **Loss function (Cost function): Hinge loss**

Prediction		Label
$W^T X_i + b \geq 1$	→	$y_i = +1$
$W^T X_i + b \leq -1$	→	$y_i = -1$

$$\boxed{y_i (W^T x_i + b)} \geq 0$$

Label **Prediction**

이 조건을 만족하는 경우 정상적으로 분류 성공

Hinge loss

$$Loss = \max(0, 1 - y_i (W^T x_i + b))$$

$y_i (W^T x_i + b) = -1$ → Loss = +2

$y_i (W^T x_i + b) = 0$ → Loss = +1

$y_i (W^T x_i + b) = +0.5$ → Loss = +0.5

$y_i (W^T x_i + b) = +1$ → Loss = 0

[실습] Support Vector Machine

- **Loss function (Cost function): Hinge loss → Gradient**

Hinge loss

$$Loss = \max(0, 1 - y_i(W^T x_i + b))$$

1 $y_i(W^T x_i + b) \geq 1$ ➡ $Loss = 0$

2 $otherwise$ ➡ $Loss = 1 - y_i(W^T x_i + b)$

[실습] Support Vector Machine

- Loss function (Cost function): **Hinge loss → Gradient**

Hinge loss

$$Loss = \max(0, 1 - y_i(W^T x_i + b))$$

1 $y_i(W^T x_i + b) \geq 1$ ➡ $Loss = 0$

2 $otherwise$ ➡ $Loss = 1 - y_i(W^T x_i + b)$

1 $y_i(W^T x_i + b) \geq 1$

$$\frac{\delta L}{\delta W} = 0 \qquad \frac{\delta L}{\delta b} = 0$$

Update 수행 X

2 $otherwise$

$$\frac{\delta L}{\delta W} = -y_i x_i \qquad \frac{\delta L}{\delta b} = -y_i$$

[실습] Support Vector Machine

- **Basecode 다운로드: LMS 강의 콘텐츠 13주차**

Support Vector Machine (GD Method)

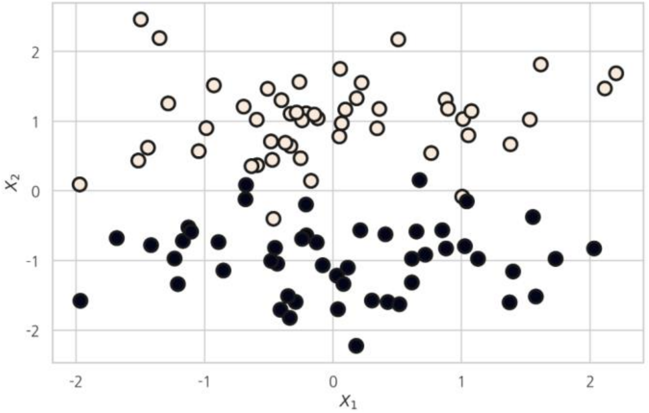
```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt

     from sklearn.datasets import make_blobs
```

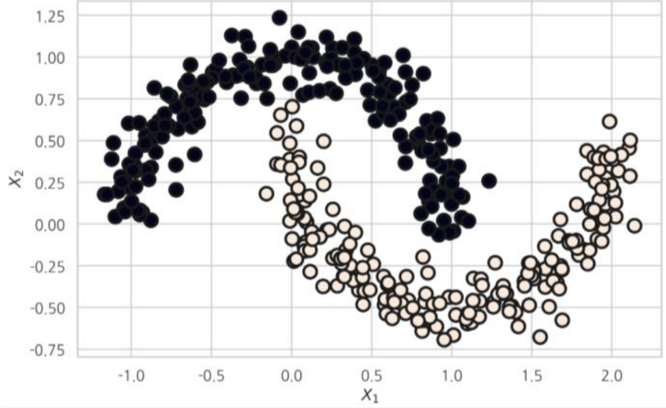
[실습] Support Vector Machine

- 데이터셋 생성: sklearn.datasets

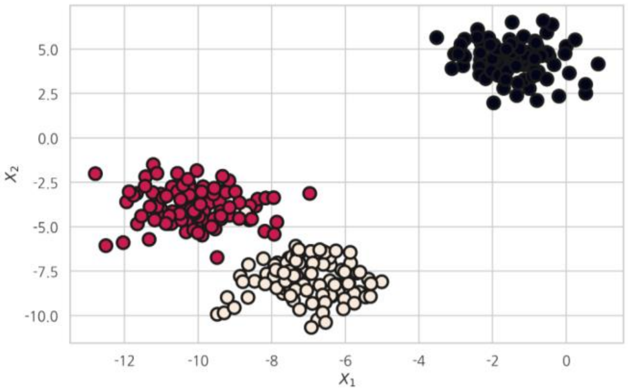
make_classification



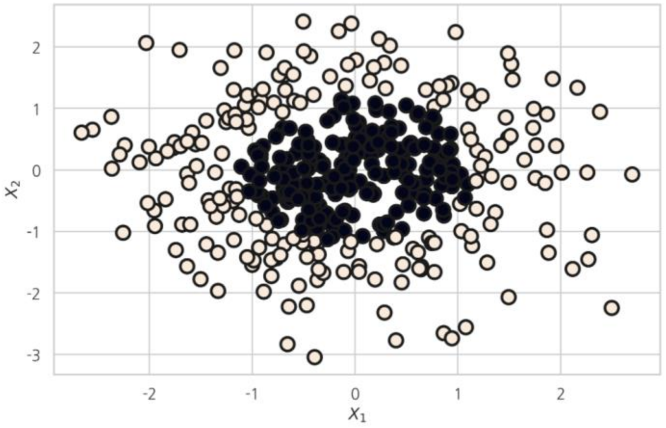
make_moons



make_blobs



make_gaussian_quantiles

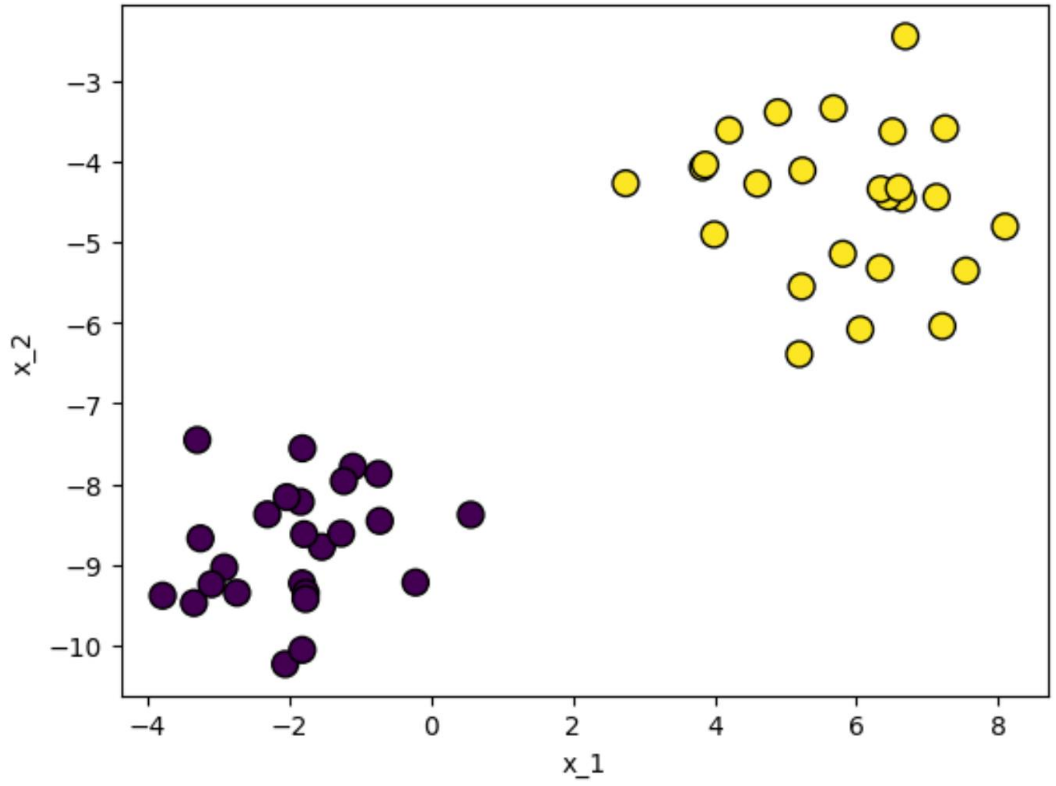


[실습] Support Vector Machine

- **데이터셋 생성: sklearn.datasets**

▾ Dataset

```
[ ] X, y = make_blobs(n_samples=50, n_features=2, centers=2, cluster_std=1.05, random_state=40)
    plt.scatter(X[:, 0], X[:, 1], marker='o', c=y, s=100, edgecolor="k", linewidth=1)
    plt.xlabel("x_1")
    plt.ylabel("x_2")
    plt.show()
```



[실습] Support Vector Machine

- **SVM 모델 작성 및 gradient decent 코드 작성**

Model

```
[ ]  class SVM:
         def __init__(self, learning_rate=0.001, n_iters=1000):
             # initialization


         def fit(self, X, y):
             # Update parameters


         def predict(self, X):
             # Prediction
```

[실습] Support Vector Machine

- **SVM 모델 training 및 도출된 W, b값 확인**

Prediction

```
[ ]  model = SVM()
     margin_log = model.fit(X, y)

     print(model.w, model.b)
```

```
[0.64070956 0.14828428] -0.12500000000000008
```

```
[ ]  margin = 2 / np.sqrt(np.dot(model.w.T, model.w))
     print(margin)
```

```
3.0411543613656318
```

$$\text{Margin} = \text{distance}(x_+, x_-)$$

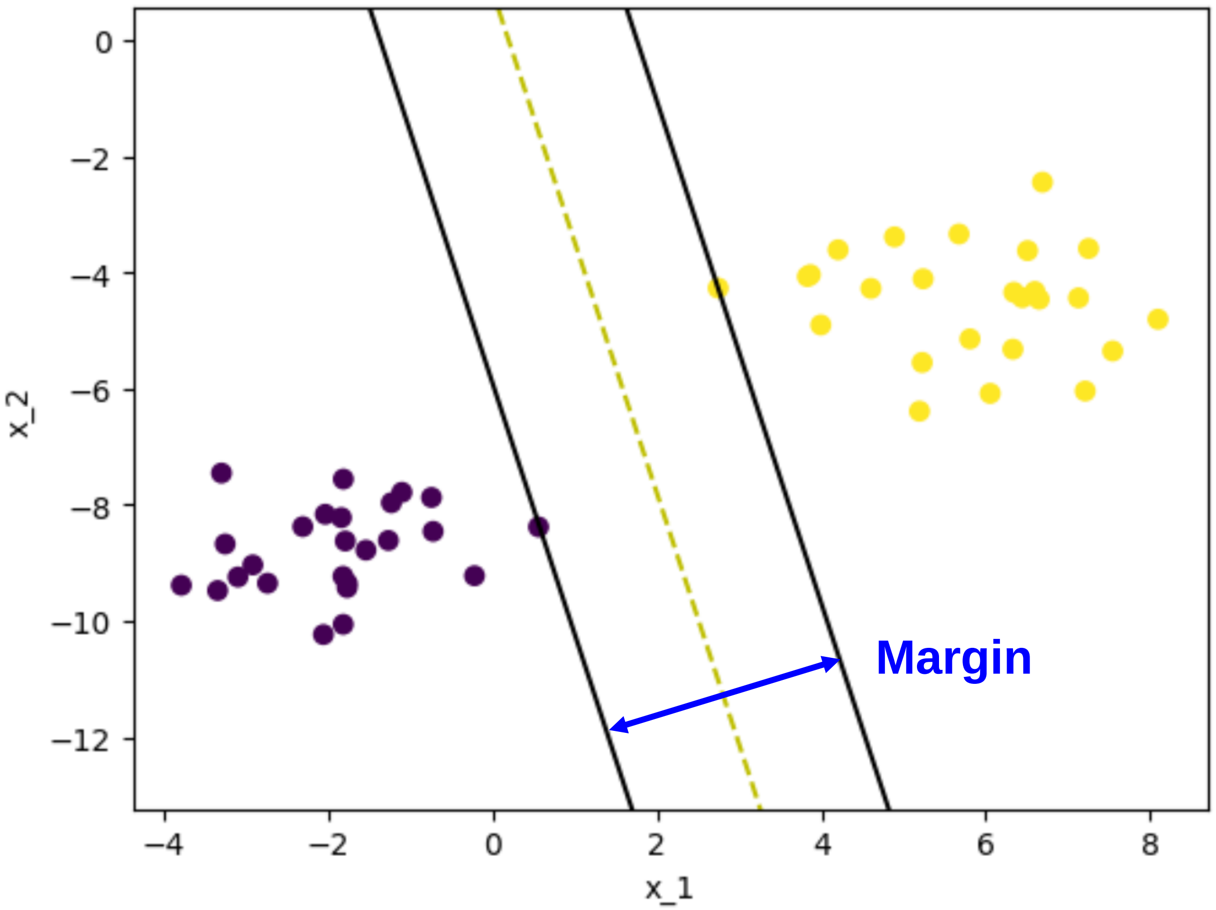
$$= \|x_+ - x_-\|_2 = \|\lambda W\|_2$$

$$= \frac{2}{W^T W} \cdot \sqrt{W^T W} = \boxed{\frac{2}{\|W\|_2}}$$

[실습] Support Vector Machine

- **Visualization**

```
[ ] visualize_svm()
```



Questions & Answers

Dongsan Jun (dsjun@dau.ac.kr)
Image Signal Processing Laboratory (www.donga-ispl.kr)
Dept. of Computer Engineering
Dong-A University, Busan, Rep. of Korea