

# Statistical Machine Learning

5주차

담당: 11기 명재성

1 / n

# Review

---

- Least Square Regression solves

$$\min_{\beta} (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta)$$

- OLS estimator is an Unbiased Estimator, MLE, and UMVUE.

$$\hat{\beta}_{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

$$E[\hat{\beta}_{OLS}] = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T E[\mathbf{Y}] = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{X} \beta = \beta$$

# Review

---

- Expected Prediction Error

$$E[(Y_0 - \hat{Y}_0)^2] = \sigma^2 + E[(\mu_0 - \hat{Y}_0)^2]$$

Irreducible error

Model error

where  $Y_0 = \mu_0 + \epsilon_0 = \mathbf{X}_0^T \boldsymbol{\beta} + \epsilon_0$

and  $\hat{Y}_0 = \mathbf{x}_0^T \hat{\boldsymbol{\beta}}$

# Review

---

- Model Error

$$\begin{aligned} E[(\mu_0 - \hat{Y}_0)^2] &= E[(\mu_0 - E[\hat{Y}_0] + E[\hat{Y}_0] - \hat{Y}_0)^2] \\ &= \underbrace{(\mu_0 - E[\hat{Y}_0])^2}_{\text{Bias}^2} + \underbrace{\text{Var}[\hat{Y}_0]}_{\text{variance}} \end{aligned}$$

- $\hat{\beta}_{OLS}$  has the smallest variance among all unbiased estimators.

# Review

---

- Ridge Regression solves

$$\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_2^2 \quad (L2 \text{ penalty})$$

- LASSO Regression solves

$$\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_1 \quad (L1 \text{ penalty})$$

# Review

---

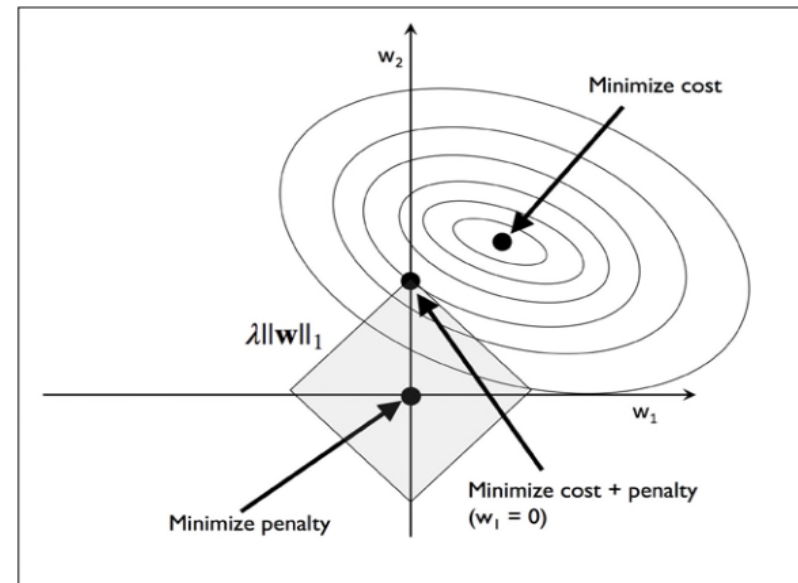
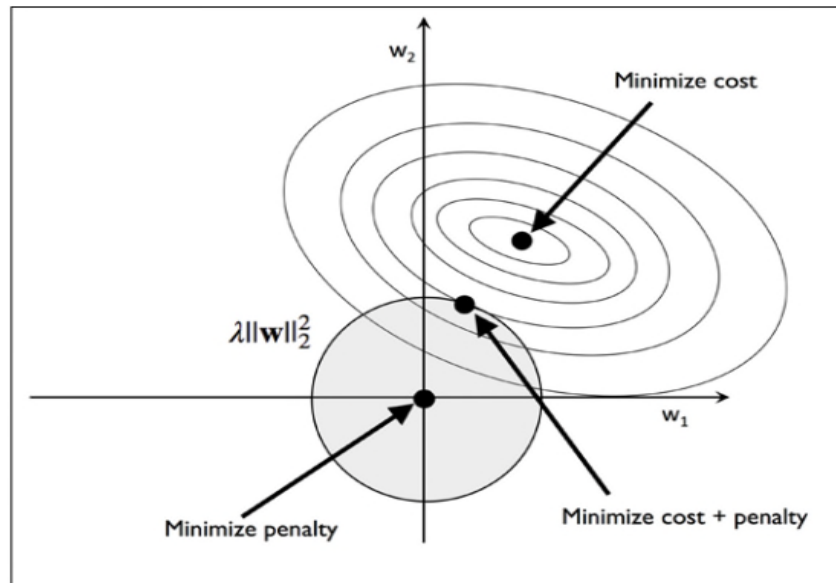
- Primal Problem

$$\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$
$$\text{subject to } \|\boldsymbol{\beta}\|_p^p - C \leq 0$$

- Dual Problem

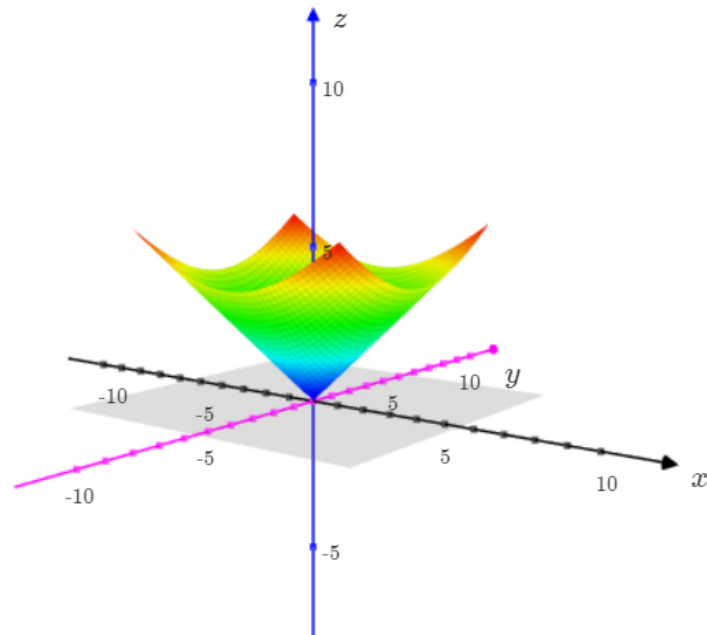
$$\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda(\|\boldsymbol{\beta}\|_p^p - C)$$

# Review



# Review

---





# Review

---

$$\hat{\beta}^{\lambda,p} = \underset{\beta}{\operatorname{argmin}} (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta) + \lambda(||\beta||_p^p - C)$$

$$\Leftrightarrow \underset{\beta}{\operatorname{argmin}} (\mathbf{Y} - \mathbf{X}\beta)^T (\mathbf{Y} - \mathbf{X}\beta) + \lambda||\beta||_p^p$$

- Although  $\hat{\beta}^{\lambda,p}$  is biased, it can achieve smaller variance so that its model error (MSE) is smaller than  $\hat{\beta}_{OLS}$  with a carefully selected  $\lambda$ .

# Review

---

- Regularized Logistic Regression solves

$$\min_{\boldsymbol{\beta}} - \sum_{i=1}^n [y_i(\boldsymbol{\beta}^T \mathbf{x}_i) - \log(1 + \exp(\boldsymbol{\beta}^T \mathbf{x}_i))] + \lambda \|\boldsymbol{\beta}\|_p^p$$

```
# Logistic regression
from sklearn.linear_model import LogisticRegression
Logit = LogisticRegression(C=1e2, random_state=1023) # C = 1/λ. 디폴트: L2, One-versus-Rest.
Logit.fit(X_train_std, y_train)
```

# Review

---

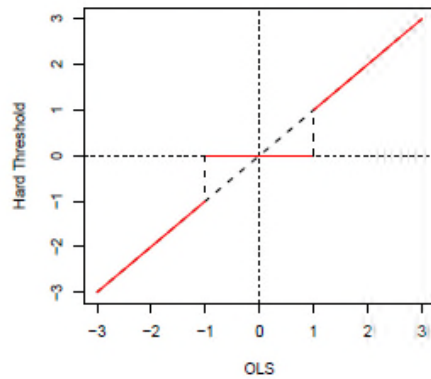
- One-dimensional Solution

$$\hat{\beta}_{OLS} = \frac{1}{n} \sum x_i y_i \quad \hat{\beta}_{Ridge} = \frac{\hat{\beta}_{OLS}}{1 + \lambda} \quad \hat{\beta}_{LASSO} = S_{\lambda}(\hat{\beta}_{OLS})$$

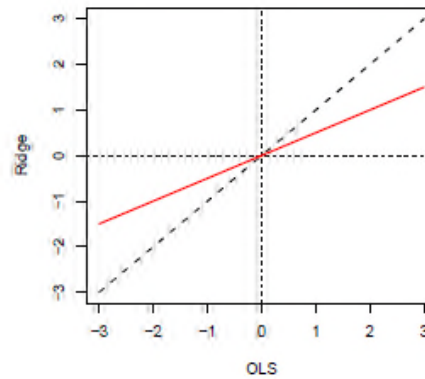
$$\times S_{\lambda}(x) = \text{sign}(x) (|x| - \lambda)_+$$

# Review

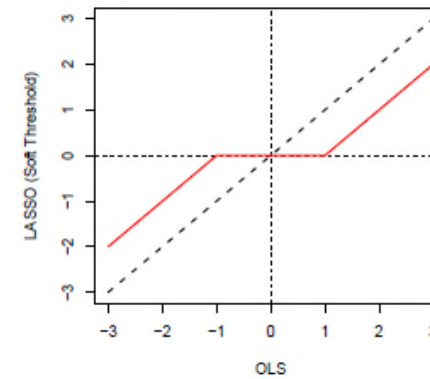
---



(a) Hard Thresh.



(b) Ridge Regression



(c) Lasso (Soft Thresh.)

# Review

---

- Elastic Net solves

$$\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \left[ \alpha \|\boldsymbol{\beta}\|_1 + \frac{1}{2} (1 - \alpha) \|\boldsymbol{\beta}\|_2^2 \right]$$

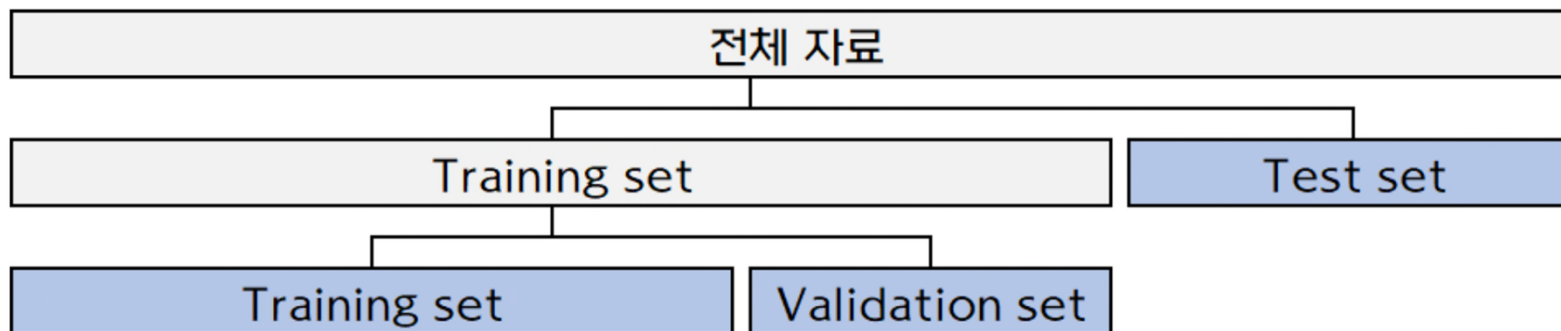
- One-dimensional Case

$$\hat{\beta}_{\text{Elastic net}} = \frac{S_{\lambda}(\hat{\beta}_{OLS})}{1 + \lambda(1 - \alpha)}$$

# Review

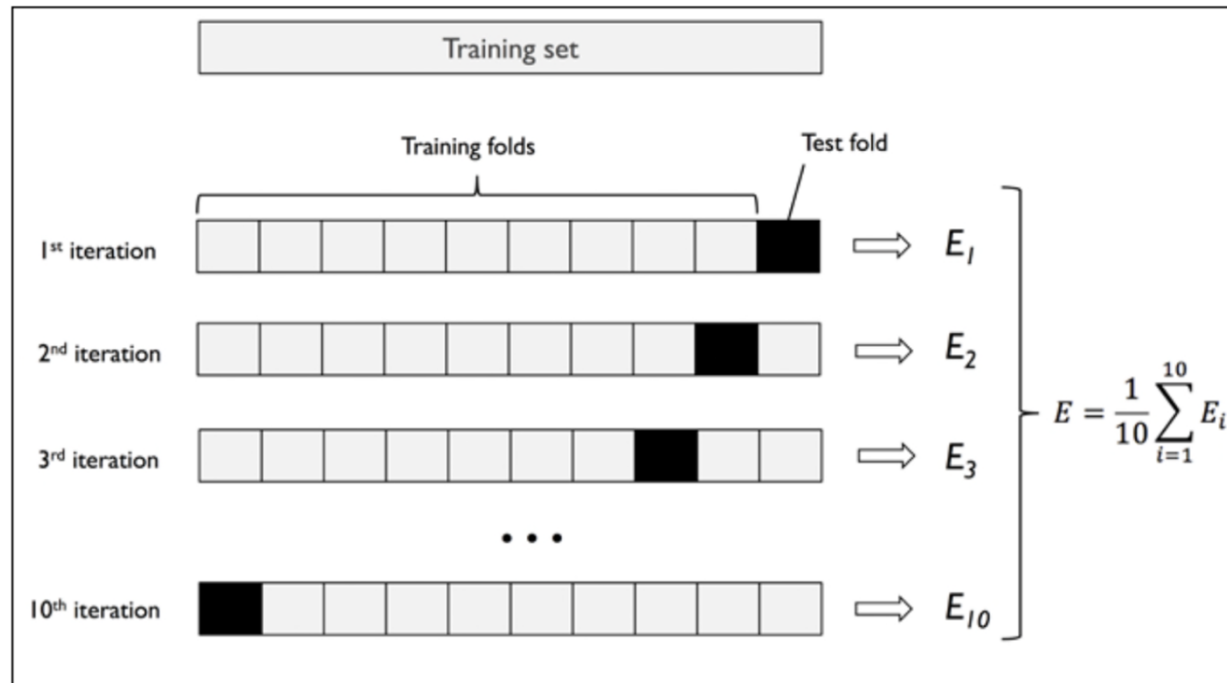
---

- Cross-Validation



# Review

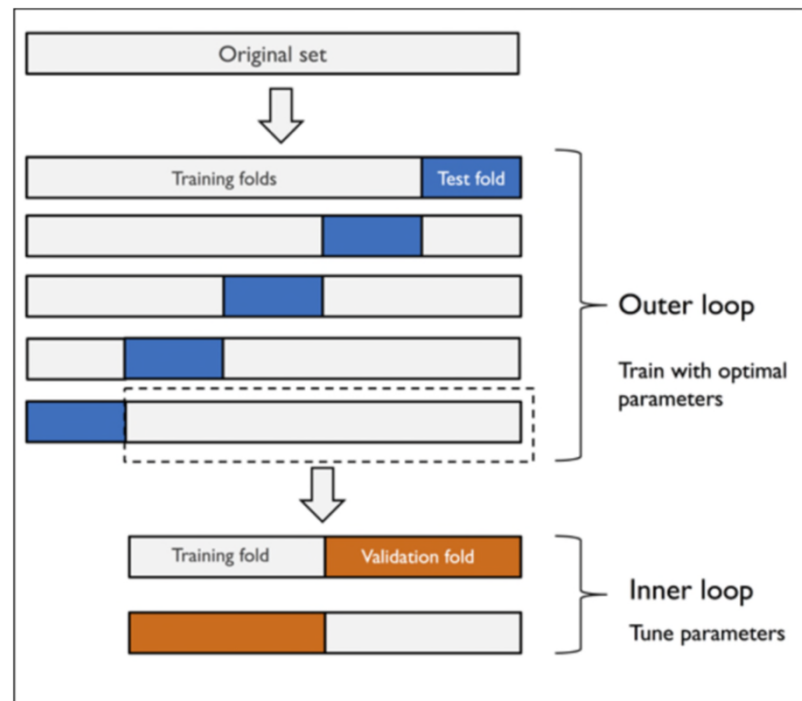
- $K = 10$



# Review

- $K_1 = 5$

$$K_2 = 2$$





# Review

---

```
[ ] # Decision tree
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.model_selection import GridSearchCV
    from sklearn.model_selection import KFold
    inner_cv=KFold(n_splits=3, shuffle=True, random_state=0)
    outer_cv=KFold(n_splits=5, shuffle=True, random_state=0)
    gs = GridSearchCV(estimator=DecisionTreeClassifier(random_state=0),
                      param_grid=[{'max_depth': [1, 2, 3, 4, 5, 6, 7, None]}],
                      scoring='accuracy', cv=inner_cv)
    scores = cross_val_score(gs, X, y, scoring='accuracy', cv=outer_cv)
    print('CV accuracy: %.3f +/- %.3f' % (np.mean(scores), np.std(scores)))
```

➡ CV accuracy: 0.942 +/- 0.012

# Review

---

`cv.glmnet {glmnet}`

R Documentation

## Cross-validation for glmnet

### Description

Does k-fold cross-validation for glmnet, produces a plot, and returns a value for `lambda` (and `gamma` if `relax=TRUE`)

### Usage

```
cv.glmnet(x, y, weights = NULL, offset = NULL, lambda = NULL,  
  type.measure = c("default", "mse", "deviance", "class", "auc", "mae",  
    "C"), nfolds = 10, foldid = NULL, alignment = c("lambda",  
    "fraction"), grouped = TRUE, keep = FALSE, parallel = FALSE,  
  gamma = c(0, 0.25, 0.5, 0.75, 1), relax = FALSE, trace.it = 0, ...)
```

# Lagrange Multiplier Theorem

---

- Primal Problem

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$\text{subject to } g_i(\mathbf{x}) \leq 0, \quad \text{for } i = 1, \dots, m$$

$$h_j(\mathbf{x}) = 0, \quad \text{for } j = 1, \dots, k$$

# Lagrange Multiplier Theorem

---

- Dual Problem

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) + \sum_i^m \alpha_i g_i(\mathbf{x}) + \sum_j^k \gamma_j h_j(\mathbf{x})$$

$$\alpha_i \geq 0, \quad \text{for } i = 1, \dots, m$$

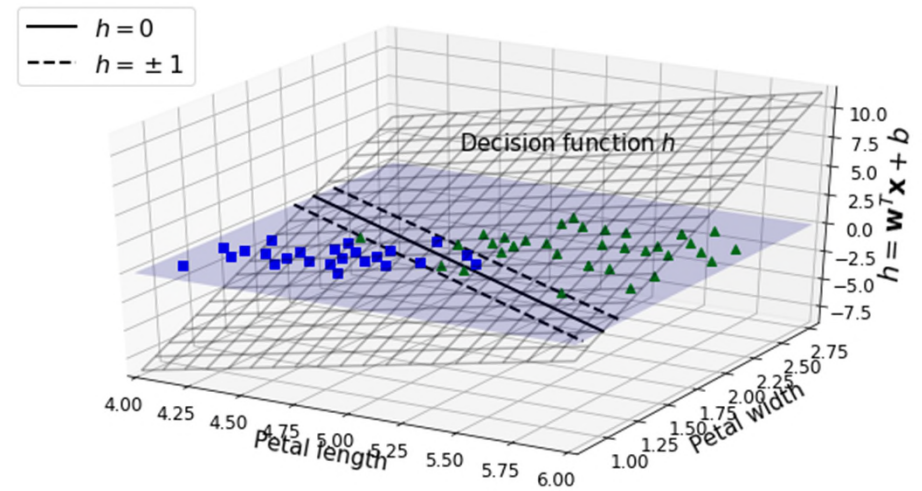
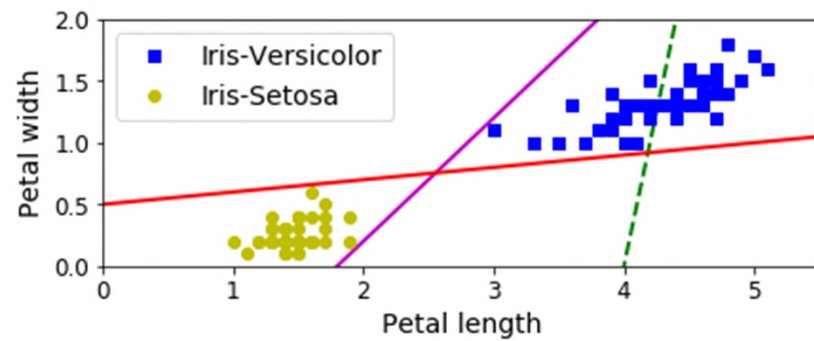
$$\gamma_j \geq 0, \quad \text{for } j = 1, \dots, k$$

# Karush-Kuhn-Tucker Conditions

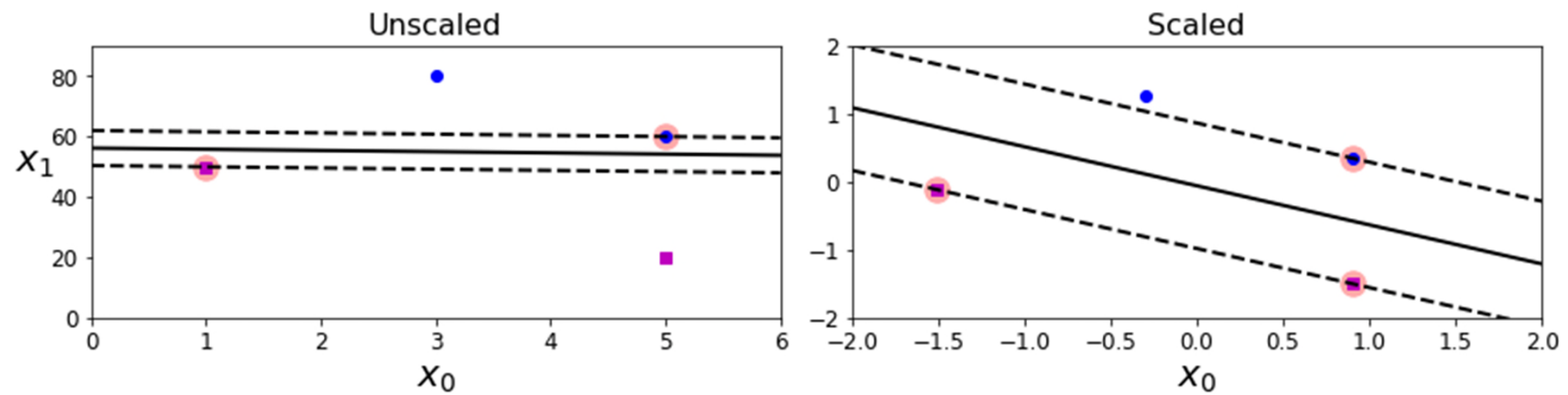
---

1.  $\nabla f(\mathbf{x}) + \sum_i^m \alpha_i \nabla g_i(\mathbf{x}) + \sum_j^k \gamma_j \nabla h_j(\mathbf{x}) = 0$  (Stationary)
2.  $\alpha_i g_i(\mathbf{x}) = 0$ , for  $i = 1, \dots, m$  (Complementary Slackness)
3.  $g_i(\mathbf{x}) \leq 0$ , for  $i = 1, \dots, m$  and  $h_j(\mathbf{x}) = 0$ , for  $j = 1, \dots, k$  (Primal Feasibility)
4.  $\alpha_i \geq 0$ , for  $i = 1, \dots, m$  (Dual Feasibility)

# Hyperplane

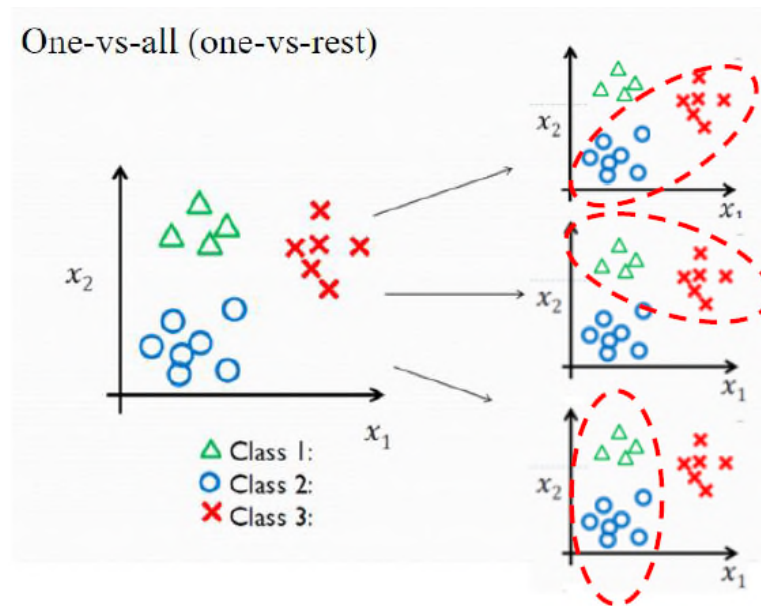


# Scaled? Unscaled?

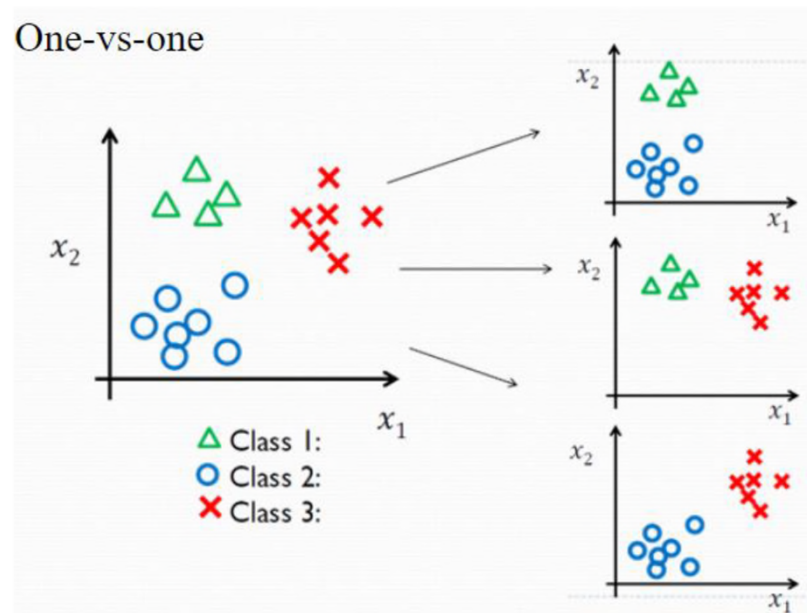


# OVR and OVO

One-vs-all (one-vs-rest)

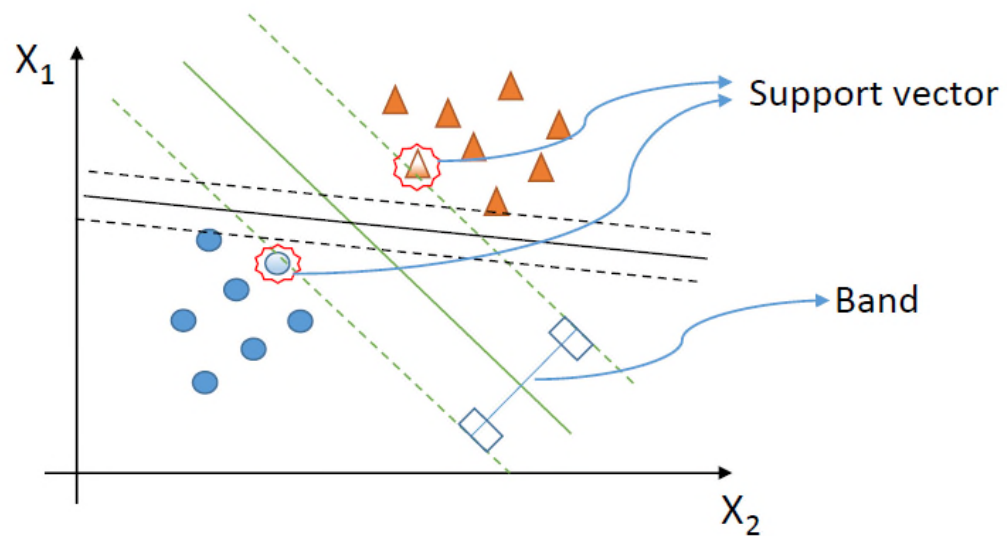


One-vs-one



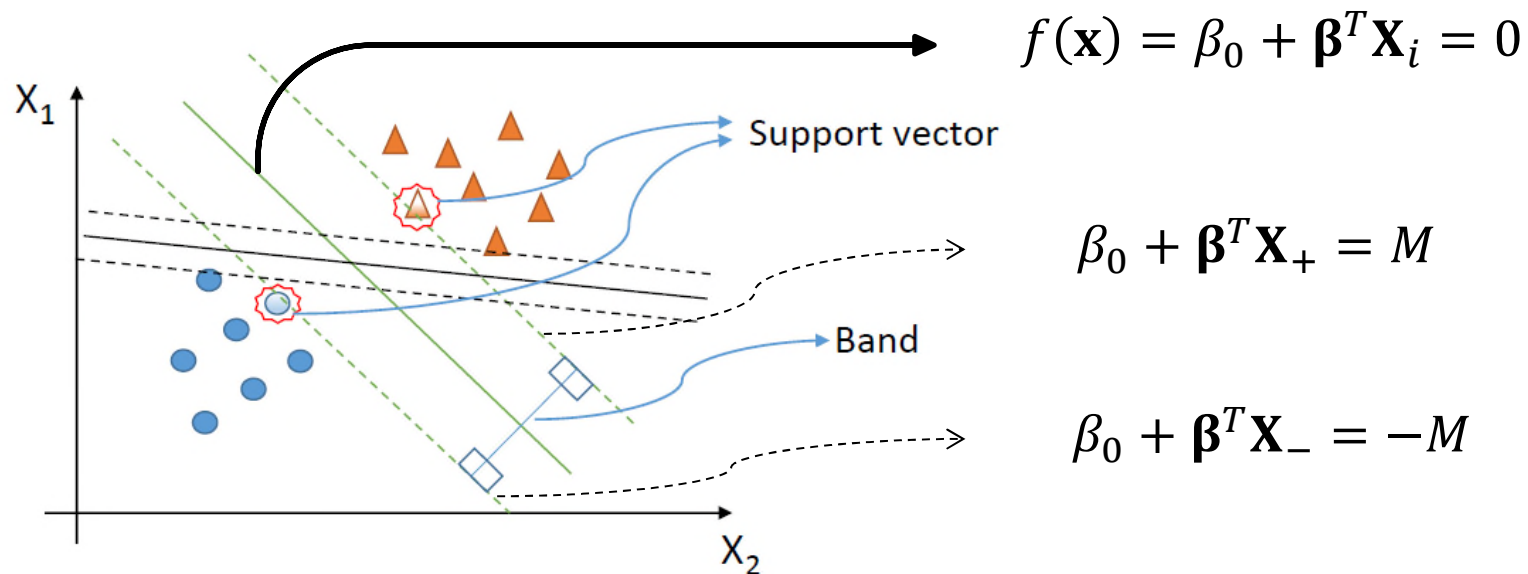


# Linear Support Vector Machine

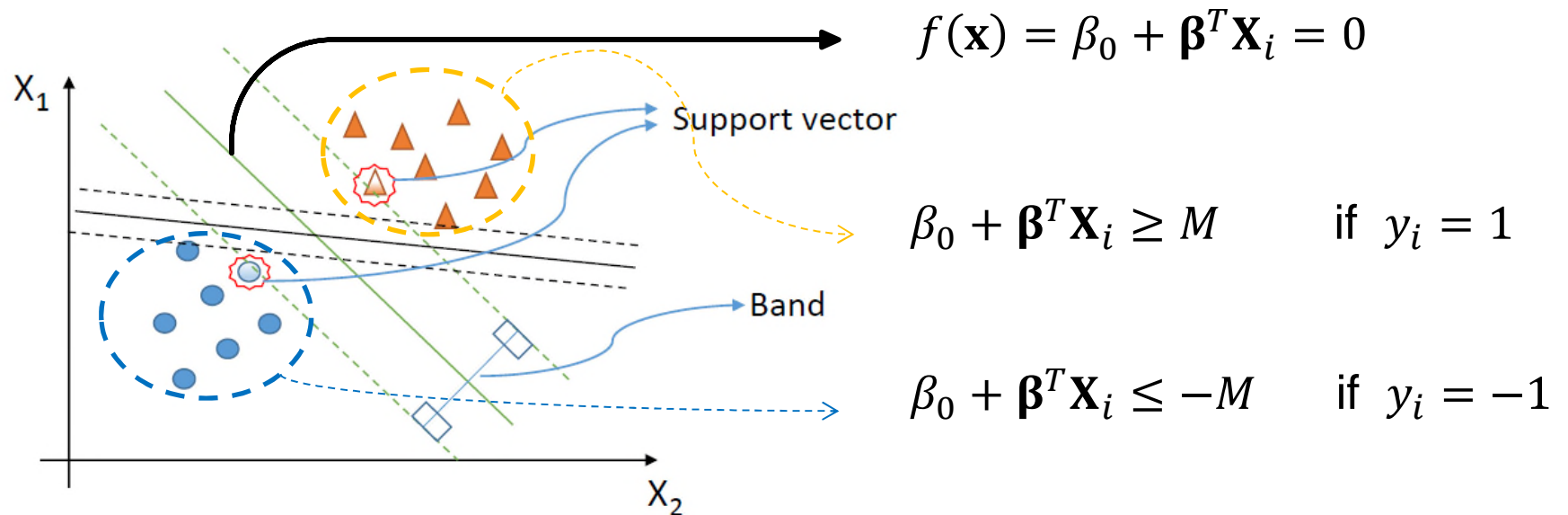


$$y = \{-1, 1\}$$

# Linear Support Vector Machine



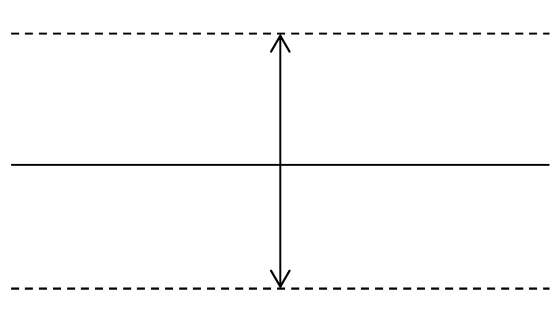
# Linear Support Vector Machine



# Linear Support Vector Machine

---

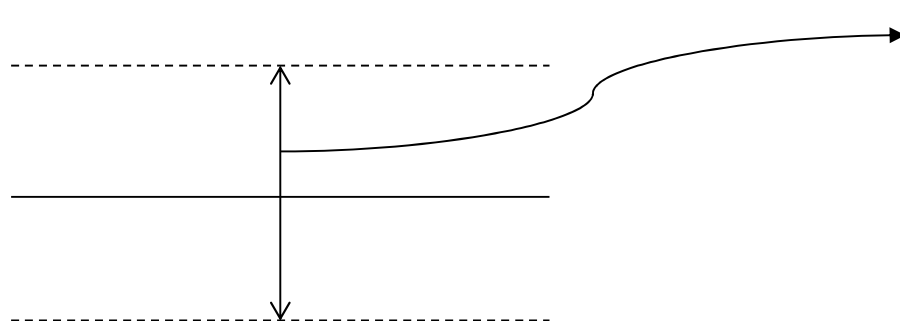
- We want to **maximize** the width of the band.


$$\begin{aligned} \beta_0 + \boldsymbol{\beta}^T \mathbf{X}_+ &= M \\ \beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i &= 0 \\ \beta_0 + \boldsymbol{\beta}^T \mathbf{X}_- &= -M \end{aligned}$$

# Linear Support Vector Machine

---

- We want to **maximize** the width of the band.



The diagram shows a 2D coordinate system with a solid horizontal line representing the decision boundary. Two dashed horizontal lines are parallel to it, one above and one below, representing the margins. A vertical double-headed arrow indicates the distance between these two dashed lines, which is the width of the band. A curved arrow points from this vertical distance to the equation  $\beta^T (\mathbf{X}_+ - \mathbf{X}_-) = 2M$ .

$$\beta^T (\mathbf{X}_+ - \mathbf{X}_-) = 2M$$
$$\max_{\beta_0, \beta} M$$

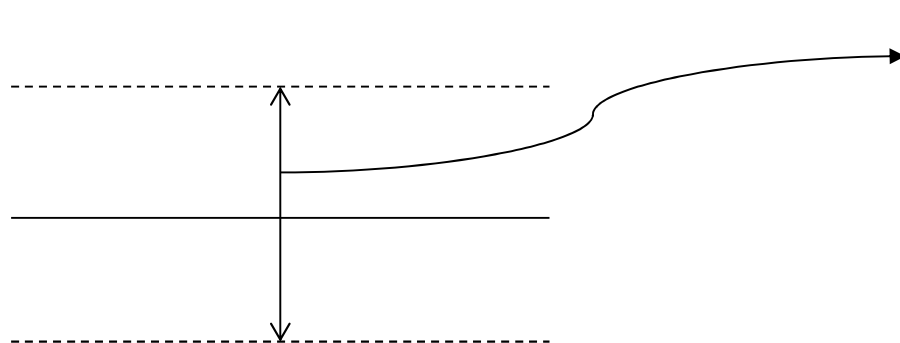
subject to  $y_i(\beta_0 + \beta^T \mathbf{X}_i) \geq M, \text{ for } i = 1, \dots, n$

$$\|\beta\| = 1$$

# Linear Support Vector Machine

---

- We want to **maximize** the width of the band.



The diagram shows a 2D coordinate system with a solid horizontal line representing the decision boundary. Above and below this line are two dashed horizontal lines representing the margins. A vertical double-headed arrow indicates the distance between the margins, which is the width of the band. A curved arrow points from the top margin line to the equation  $\frac{\boldsymbol{\beta}^T}{\|\boldsymbol{\beta}\|} (\mathbf{x}_+ - \mathbf{x}_-) = \frac{2M}{\|\boldsymbol{\beta}\|}$ .

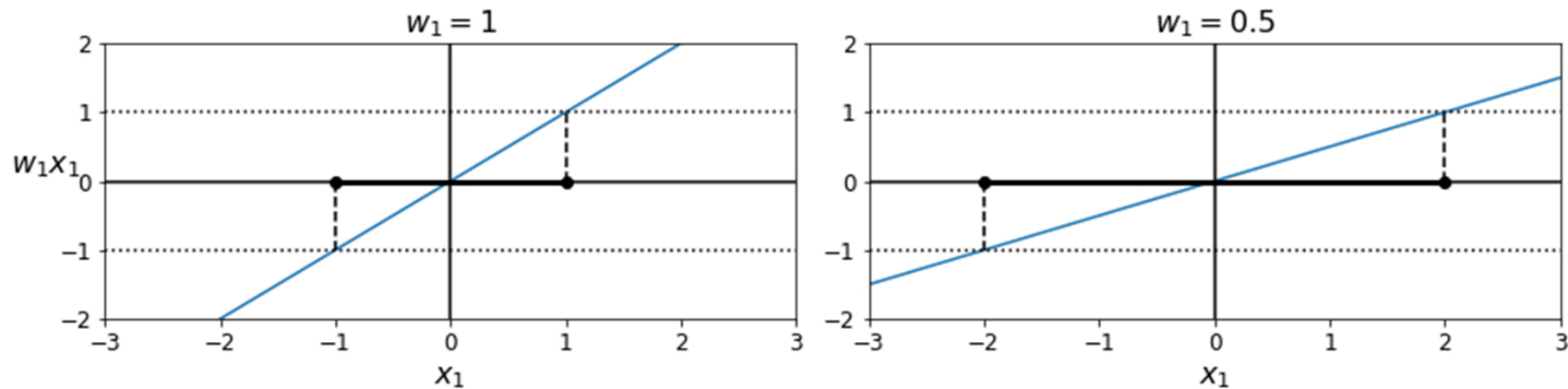
$$\frac{\boldsymbol{\beta}^T}{\|\boldsymbol{\beta}\|} (\mathbf{x}_+ - \mathbf{x}_-) = \frac{2M}{\|\boldsymbol{\beta}\|}$$
$$\max_{\beta_0, \boldsymbol{\beta}} M$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq M, \text{ for } i = 1, \dots, n$

$$\|\boldsymbol{\beta}\| = 1$$

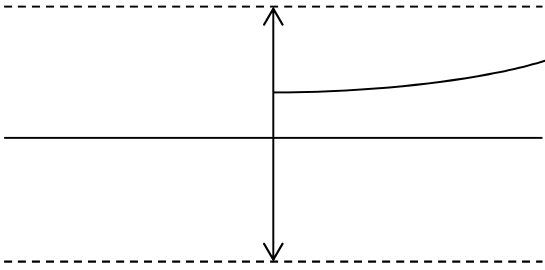
# Linear Support Vector Machine

- A smaller weight vector results in a larger margin



# Linear Support Vector Machine

- We want to **maximize** the width of the band.



The diagram shows a 2D coordinate system with a solid horizontal line representing the decision boundary. Above and below this line are two dashed horizontal lines representing the margins. A vertical double-headed arrow indicates the distance between these margins. A curved arrow points from the top margin line to the equation  $\frac{\boldsymbol{\beta}^T}{\|\boldsymbol{\beta}\|} (\mathbf{x}_+ - \mathbf{x}_-) = \frac{2M}{\|\boldsymbol{\beta}\|}$ .

$$\frac{\boldsymbol{\beta}^T}{\|\boldsymbol{\beta}\|} (\mathbf{x}_+ - \mathbf{x}_-) = \frac{2M}{\|\boldsymbol{\beta}\|}$$
$$\min_{\beta_0, \boldsymbol{\beta}} \|\boldsymbol{\beta}\|$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq M$ , for  $i = 1, \dots, n$

$$M = ?$$

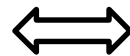


# Linear Support Vector Machine

---

- We want to **maximize** the width of the band.

$$\max_{\beta_0, \boldsymbol{\beta}} M$$



$$\min_{\beta_0, \boldsymbol{\beta}} ||\boldsymbol{\beta}||^2$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq M$ , for  $i = 1, \dots, n$

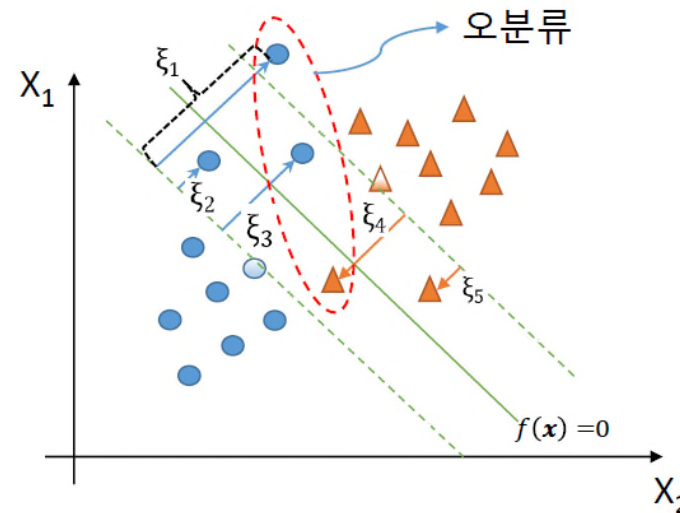
subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1$ , for  $i = 1, \dots, n$

and

$$||\boldsymbol{\beta}|| = 1$$

# Linear Support Vector Machine

- If the data are not perfectly separable, no solution exists.



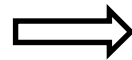
# Linear Support Vector Machine

---

- Hard Margin Classifier

$$\min_{\beta_0, \boldsymbol{\beta}} \|\boldsymbol{\beta}\|^2$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1$ , for  $i = 1, \dots, n$



- Soft Margin Classifier

$$\min_{\beta_0, \boldsymbol{\beta}} \|\boldsymbol{\beta}\|^2$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1 - \zeta_i$

and  $\zeta_i \geq 0$ ,

and  $\sum_i^n \zeta_i \leq \tilde{C}$ , for  $i = 1, \dots, n$

# Linear Support Vector Machine

---

- Primal Problem

$$\min_{\beta_0, \boldsymbol{\beta}} ||\boldsymbol{\beta}'||^2$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{X}_i) \geq 1 - \zeta_i$

and  $\zeta_i \geq 0$ ,

and  $\sum_{i=1}^n \zeta_i \leq \tilde{C}$ , for  $i = 1, \dots, n$

- Dual Problem

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}'||^2 + C \sum_{i=1}^n \zeta_i$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}'\mathbf{X}_i) \geq 1 - \zeta_i$

and  $\zeta_i \geq 0$ , for  $i = 1, \dots, n$

**C is not a Lagrange multiplier**

# Linear Support Vector Machine

---

- Primal Problem

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}'||^2 + C \sum_i^n \zeta_i$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq 1 - \zeta_i$

and  $\zeta_i \geq 0$ , for  $i = 1, \dots, n$

- Dual Problem

$$\Leftrightarrow \min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}'||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq 1 - \zeta_i$

for  $i = 1, \dots, n$

# Linear Support Vector Machine

---

- Primal Problem

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i$$

subject to  $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i) \geq 1 - \zeta_i, \quad \text{for } i = 1, \dots, n$

# Linear Support Vector Machine

---

- Dual Problem

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

- Taking derivative w.r.t  $\beta_0, \boldsymbol{\beta}, \zeta_i$   
(Stationary)

# Linear Support Vector Machine

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}'||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\begin{aligned} \text{(Stationary)} \left\{ \begin{array}{l} \frac{\partial}{\partial \beta_0} \mathcal{L}_p: \sum_i^n \alpha_i y_i = 0 \\ \frac{\partial}{\partial \boldsymbol{\beta}} \mathcal{L}_p: \boldsymbol{\beta} = \sum_i^n \alpha_i y_i \mathbf{x}_i \\ \frac{\partial}{\partial \zeta_i} \mathcal{L}_p: \alpha_i = C - \gamma_i \end{array} \right. \quad \text{(Complementary Slackness)} \left\{ \begin{array}{l} \alpha_i [y_i f(\mathbf{x}_i) - (1 - \zeta_i)] = 0 \\ \gamma_i \zeta_i = 0 \end{array} \right. \end{aligned}$$



# Linear Support Vector Machine

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \max_{\alpha_i} \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad \text{QP}$$

subject to  $0 \leq \alpha_i \leq C$

$$\text{and } \sum_i^n \alpha_i y_i = 0, \quad \text{for } i = 1, \dots, n$$

# Linear Support Vector Machine

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\Rightarrow \hat{\boldsymbol{\beta}} = \sum_i^n \hat{\alpha}_i y_i \mathbf{x}_i$$

$$\hat{\beta}_0 = y_i - \hat{\boldsymbol{\beta}}^T \mathbf{x}_k \quad \text{for any support vector } \mathbf{x}_k$$

$$\widehat{f(\mathbf{x}_i)} = \hat{\beta}_0 + \hat{\boldsymbol{\beta}}^T \mathbf{x}_k$$

## Linear Support Vector Machine

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \max_{\alpha_i} \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad \text{QP}$$

subject to  $0 \leq \alpha_i \leq C$

$$\text{and } \sum_i^n \alpha_i y_i = 0, \quad \text{for } i = 1, \dots, n$$

## Kernel Support Vector Machine

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}'||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \max_{\alpha_i} \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to  $0 \leq \alpha_i \leq C$

$$\text{and } \sum_i^n \alpha_i y_i = 0, \quad \text{for } i = 1, \dots, n$$

# Kernel Support Vector Machine

---

- Kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$$

*Linear Kernel*

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j))$$

*Gaussian Kernel  
(Radial Basis function)*

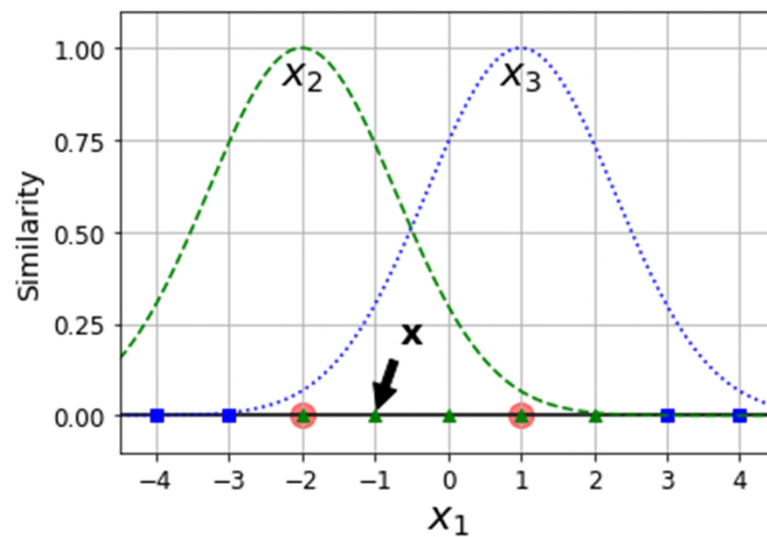
$$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma + \gamma \mathbf{x}_i^T \mathbf{x}_j)^p$$

*polynomial Kernel*

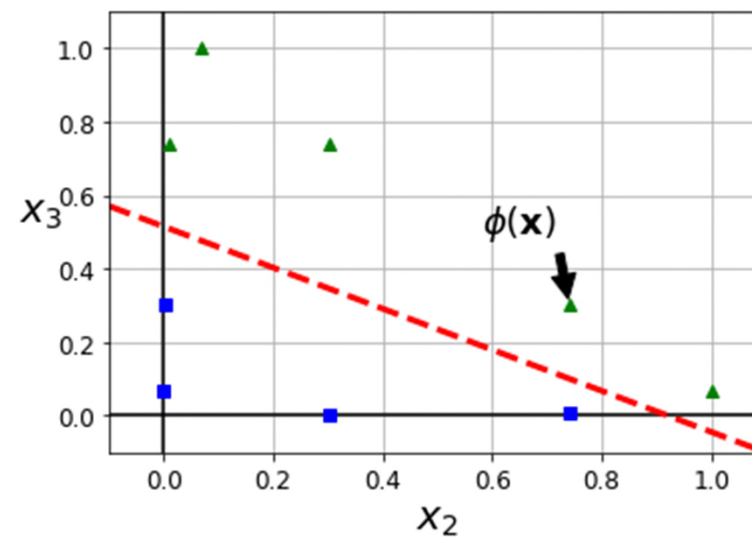
$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(k_1 \mathbf{x}_i^T \mathbf{x}_j + k_2)$$

*Sigmoid Kernel*

# Kernel Support Vector Machine



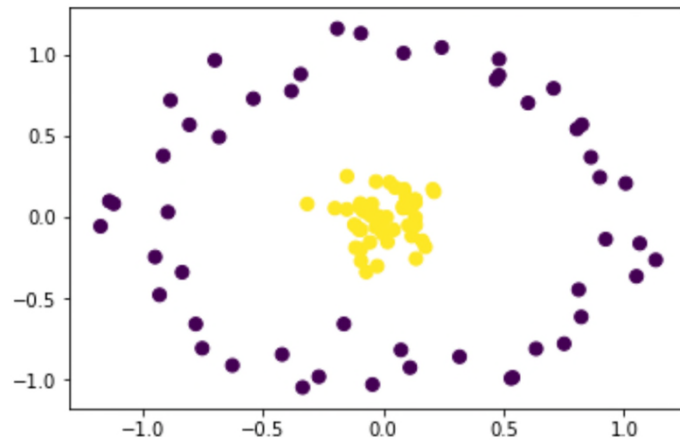
$x_1$



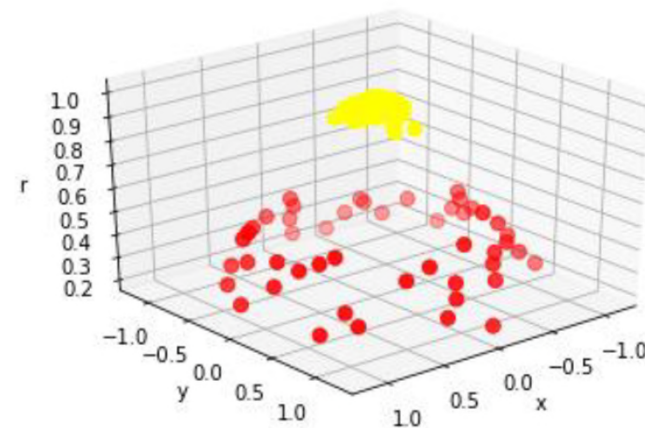
$$x_2 = \exp(-(x_1 - 2)^2)$$
$$x_3 = \exp(-(x_1 + 1)^2)$$

# Kernel Support Vector Machine

---



$(x_1, x_2)$



$(x_1, x_2, \exp(-(x_1^2 + x_2^2)))$

# Kernel Support Vector Machine

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}'||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \max_{\alpha_i} \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \mathbf{h}(\mathbf{x}_i)^T \mathbf{h}(\mathbf{x}_j)$$

subject to  $0 \leq \alpha_i \leq C$

$$\text{and } \sum_i^n \alpha_i y_i = 0, \quad \text{for } i = 1, \dots, n$$



# Kernel Trick

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \max_{\alpha_i} \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to } 0 \leq \alpha_i \leq C$$

$$\text{and } \sum_i^n \alpha_i y_i = 0, \quad \text{for } i = 1, \dots, n$$

## Kernel trick

- 특성함수의 생성 어려움 + 고차원 확장시 차원의 저주 문제 발생.
- 2차 다항커널 : 입력변수  $x_1$  과  $x_2$  이고  $i$  번째 관측치와  $j$  번째 관측치일때,

$$\begin{aligned}
 K(\mathbf{x}_i, \mathbf{x}_j) &= (1 + \mathbf{x}_i^T \mathbf{x}_j)^2 \\
 &= (1 + x_{i,1}x_{j,1} + x_{i,2}x_{j,2})^2 \\
 &= 1 + 2x_{i,1}x_{j,1} + 2x_{i,2}x_{j,2} + (x_{i,1}x_{j,1})^2 + (x_{i,2}x_{j,2})^2 + 2x_{i,1}x_{j,1}x_{i,2}x_{j,2}
 \end{aligned} \tag{7.11}$$

- 이때 다음과 같이 정의하면,

$$h_1(x_1, x_2) = 1, \quad h_2(x_1, x_2) = \sqrt{2}x_1, \quad h_3(x_1, x_2) = \sqrt{2}x_2, \quad h_4(x_1, x_2) = x_1^2, \quad h_5(x_1, x_2) = x_2^2, \quad h_6(x_1, x_2) = \sqrt{2}x_1x_2$$

$$\mathbf{h}(x_1, x_2) = (h_1(x_1, x_2), h_2(x_1, x_2), \dots, h_6(x_1, x_2))^T;$$

- 식 (7.11)은  $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2 = \mathbf{h}(\mathbf{x}_i)^T \mathbf{h}(\mathbf{x}_j)$  로 변형 가능.
- 특성함수를 정의하지 않고 커널 함수를 이용.
- 즉,  $\hat{\beta}$  이  $\mathbf{h}(\mathbf{x}_i)^T \mathbf{h}(\mathbf{x}_j)$  의 형태이면,  $K(\mathbf{x}_i, \mathbf{x}_j)$  를 직접 이용하여 추정.

**$\beta_0$ 와  $\beta$ 의 추정 - by kernel trick**

- 특성변수  $x$ 로 부터 basis함수  $h(x)$ 로 차원을 증대시키면 커널 SVM 목적함수.

$$L_k = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j h(x_i)^T h(x_j) \quad (7.12)$$

- 선형 SVM 식 (7.11)은  $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}^T x = \hat{\beta}_0 + \sum_{i=1}^n \hat{\alpha}_i y_i x_i^T x$ 로 변형 가능.

- $L_k$  최소화한 모수 추정치를  $\hat{\beta}_0^*$ 와  $\hat{\beta}^*$ 라 할 때 커널 SVM의 예측치

$$\hat{f}(x) = \hat{\beta}_0^* + \sum_{i=1}^n \hat{\alpha}_i^* y_i h(x_i)^T h(x) \quad (7.13)$$

- 식(7.12)와 식(7.13) 모두  $h(x_i)^T h(x_j)$ 의 형태임.
- 식(7.12)에  $h(x_i)^T h(x_j)$  대신 커널 함수  $K(x_i, x)$ 를 대체하여  $\hat{\beta}_0^*$ 와  $\hat{\beta}^*$ 를 추정.
- 식(7.13)도  $h(x_i)^T h(x_j)$ 를 이용하여 동일한 커널 SVM을 구함.

## Hinge Loss

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}||^2 + C \sum_i^n \zeta_i$$

$$\text{subject to } y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq 1 - \zeta_i$$

$$\text{and } \zeta_i \geq 0, \text{ for } i = 1, \dots, n$$

# Hinge Loss

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}||^2 + C \sum_i^n \zeta_i$$

$$\text{subject to } \zeta_i \geq 1 - y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i)$$

$$\text{and } \zeta_i \geq 0, \text{ for } i = 1, \dots, n$$

# Hinge Loss

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i$$

$$\text{subject to } \zeta_i \geq [1 - y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i)]_+ \quad \text{for } i = 1, \dots, n$$

# Hinge Loss

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \min_{\beta_0, \boldsymbol{\beta}} \|\boldsymbol{\beta}\|^2 + C \sum_i^n [1 - y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i)]_+$$

# Hinge Loss

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} \|\boldsymbol{\beta}\|^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$
$$\iff \min_{\boldsymbol{\beta}} \frac{1}{C} \|\boldsymbol{\beta}\|^2 + \sum_i^n [1 - y_i f(\mathbf{x}_i)]_+$$



# Hinge Loss

---

$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \min_{\boldsymbol{\beta}} \lambda ||\boldsymbol{\beta}||^2 + \sum_i^n [1 - y_i f(\mathbf{x}_i)]_+$$

$$\frac{1}{C} = \lambda$$

# Hinge Loss

---

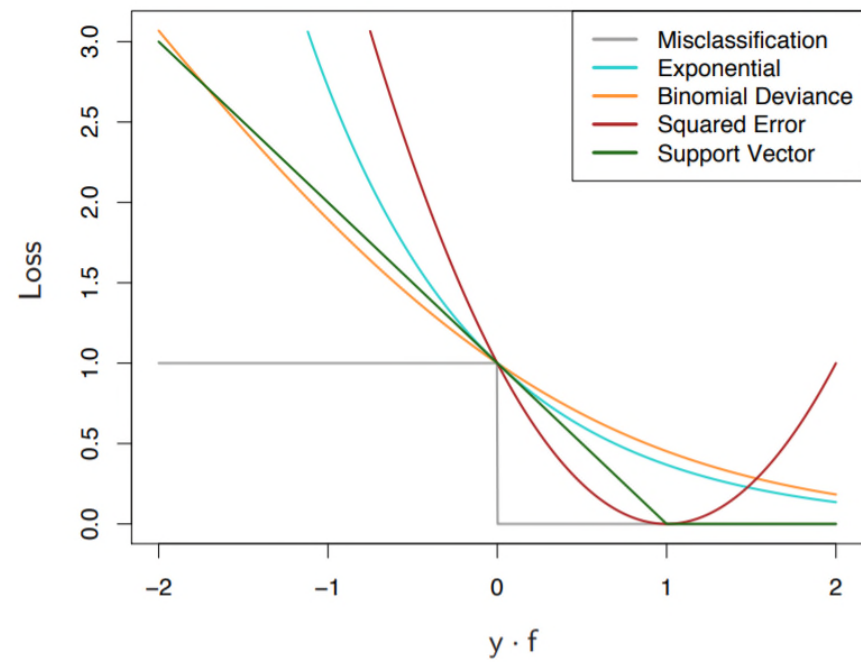
$$\min_{\beta_0, \boldsymbol{\beta}, \zeta_i} ||\boldsymbol{\beta}||^2 + C \sum_i^n \zeta_i - \sum_i^n \gamma_i \zeta_i - \sum_i^n \alpha_i [y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) - (1 - \zeta_i)]$$

$$\iff \min_{\boldsymbol{\beta}} \sum_i^n [1 - y_i f(\mathbf{x}_i)]_+ + \lambda ||\boldsymbol{\beta}||^2$$

$\Rightarrow$  Expression of “Loss + Penalty”

# Hinge Loss

---



# Grid Search for SVM

---

```
▶ ### Grid search에 의한 초모수 결정 (SVM) ###  
from sklearn.model_selection import GridSearchCV  
from sklearn.svm import SVC  
pipe_svc = make_pipeline(StandardScaler(), SVC(random_state=1))  
param_range = [0.0001, 0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0]  
param_grid = [{ 'svc__C': param_range, 'svc__kernel': ['linear'] },  
               { 'svc__C': param_range, 'svc__gamma': param_range,  
                 'svc__kernel': ['rbf'] },  
               { 'svc__C': param_range, 'svc__degree': [2,3,4,5],  
                 'svc__kernel': ['poly'] }]  
gs = GridSearchCV(estimator=pipe_svc, param_grid=param_grid,  
                  scoring='accuracy', cv=10)  
gs = gs.fit(X_train, y_train)  
print(gs.best_score_)  
print(gs.best_params_)  
  
clf = gs.best_estimator_  
clf.fit(X_train, y_train)  
clf.score(X_train, y_train)  
clf.score(X_test, y_test)
```

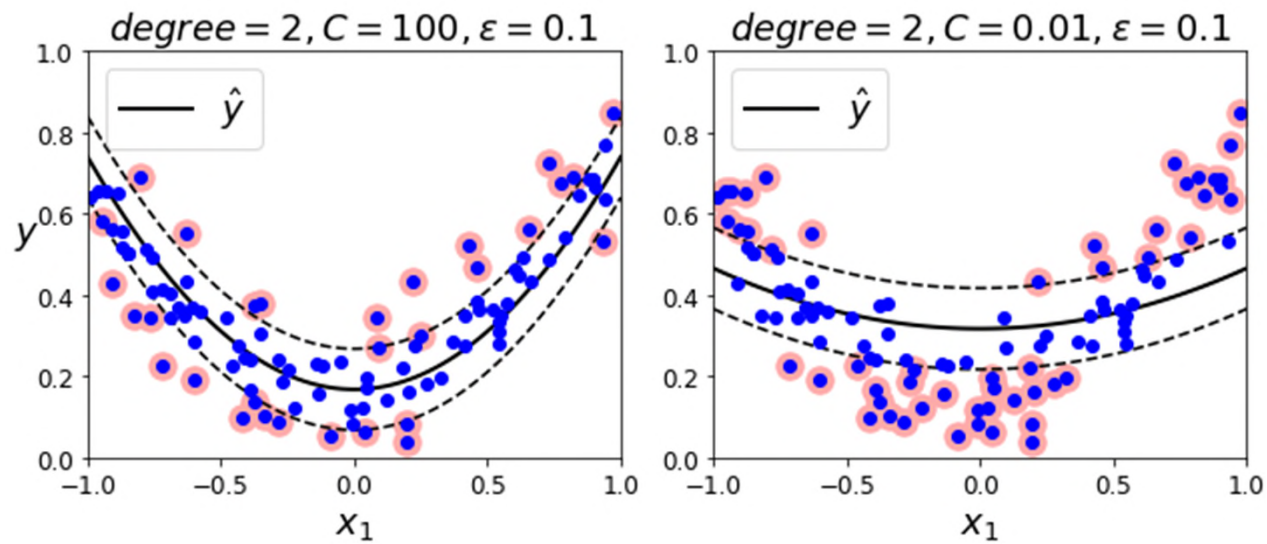
# Support Vector Regression

---

$$\min_{\boldsymbol{\beta}} \sum_i^n L_{\epsilon}[y_i - f(\mathbf{x}_i)] + \lambda \boldsymbol{\beta}^T \boldsymbol{\beta} \quad \text{Linear SVR}$$

$$\min_{\boldsymbol{\beta}} \sum_i^n L_{\epsilon}[y_i - f(\mathbf{x}_i)] + \lambda \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} \quad \text{Kernel SVR}$$

# Support Vector Regression



# Support Vector Regression

---

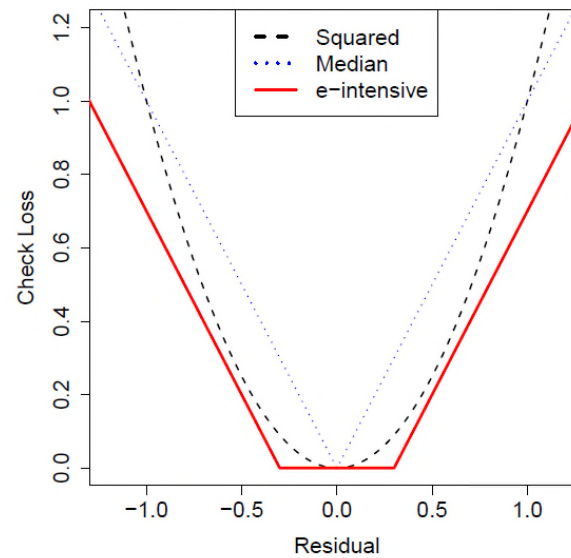


Figure:  $\epsilon$ -intensive loss for SVR.

## reference

자료

19-2 STAT424 통계적 머신러닝 - 박유성 교수님

교재

파이썬을 이용한 통계적 머신러닝 (2020) - 박유성

ISLR (2013) - G. James, D. Witten, T. Hastie, R. Tibshirani

The elements of Statistical Learning (2001) - J. Friedman, T. Hastie, R. Tibshirani

Hands on Machine Learning (2017) - Aurelien Geron