Example

- $\mathbf{x_1}^T = (2,3)$, $\mathbf{x_2}^T = (4,1)$, $\mathbf{x_3}^T = (5,1)$
- $y_1 = +1$, $y_2 = -1$, $y_3 = -1$

$$L(\mu) = \sum_{i=1}^{N} \mu_i - \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \mu_i \mu_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}}{2}$$

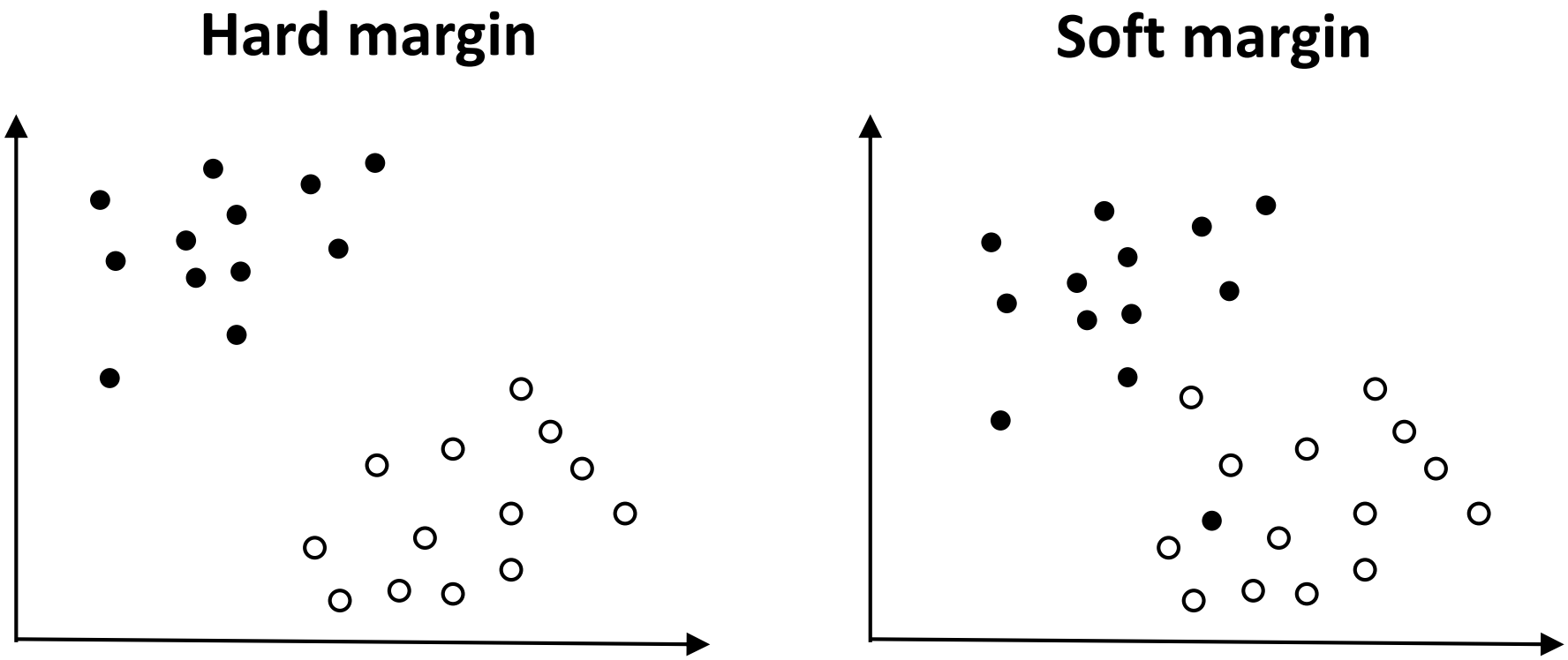
Support Vector Machine w/ Soft Margin & Kernel Trick

Hanwool Jeong

hwjeong@kw.ac.kr

Soft Margin vs. Hard Margin

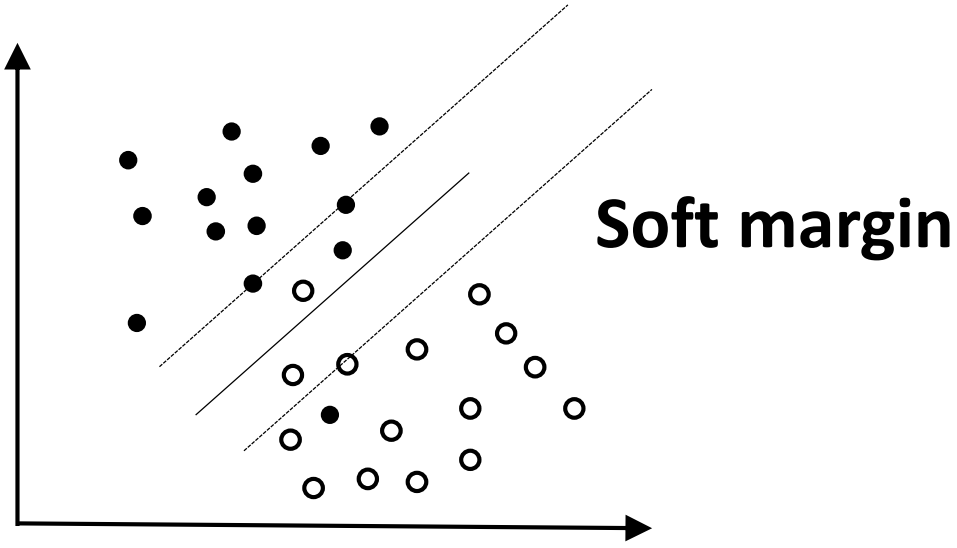
- There can be outliers



What is Optimal Boundary?

- We can categorize the samples into three types according to value of $y_i(\mathbf{w}^T\mathbf{x_i} + b)$
  1) $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$ 2) $0 < y_i(\mathbf{w}^T\mathbf{x_i} + b) < 1$ 3) $y_i(\mathbf{w}^T\mathbf{x_i} + b) < 0$
- A slack variable ξ is defined using $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i$.

➔ What is the meaning of $\sum_{i=1}^{n} \xi_i$ magnitude?



Problem Defining

- Maximize the number of samples 1) while the minimizing the number of samples of 2) & 3)
- We can revise the hard margin example into

Minimize $\frac{\|w\|^2}{2}$

$g(\mathbf{x_i}) = 1 - y_i(\mathbf{w}^T\mathbf{x_i} + b) \leq 0$

Minimize $\frac{\|w\|^2}{2} + C\sum_{i=1}^{N} \xi_i$

subject to $0 \leq \xi_i$

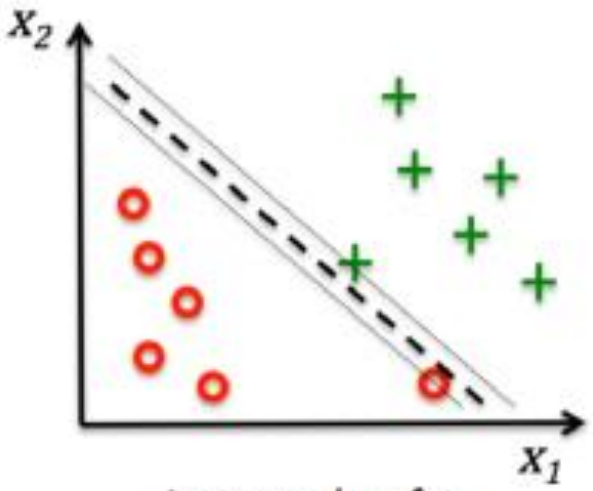
$g(\mathbf{x_i}) = 1 - \xi_i - y_i(\mathbf{w}^T\mathbf{x_i} + b) \leq 0$

- What is the meaning of C?
- We can repeat the procedure by the Lagrange aux. function of

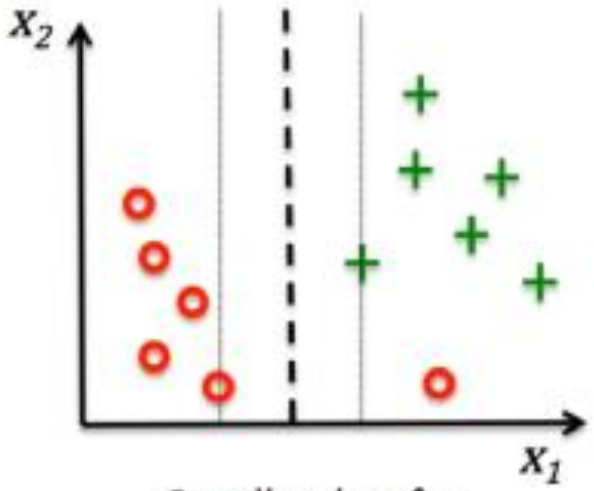
$$L = \frac{\|w\|^2}{2} + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \mu_i \{y_i(\mathbf{w}^T\mathbf{x_i} + b) - 1+\xi_i\}$$

Impact of C

- C = 0 ignores the second term just maximizing margin and accuracy is not considered.
- With larger C accuracy becomes more taken into account.

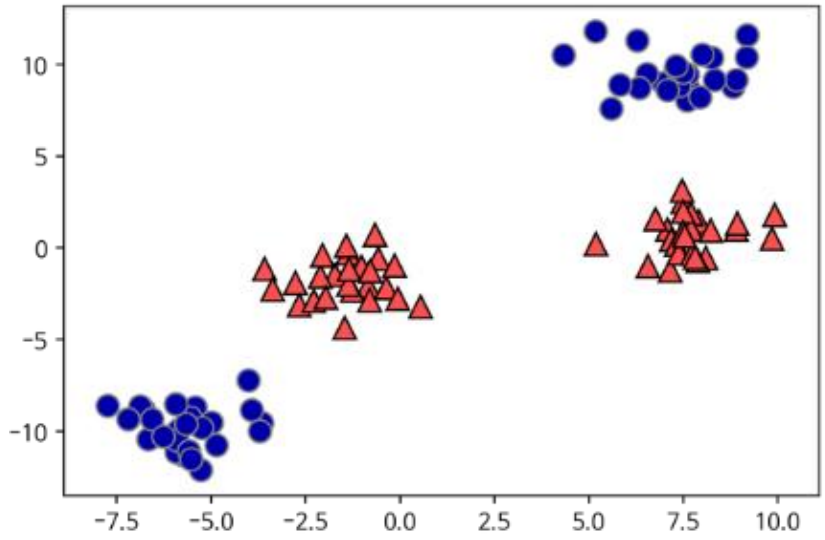


Large value for parameter *C*



Small value for parameter *C*

Linearly Separable vs. Inseparable



Kernel Trick!

- Revisit Lagrange auxiliary function:

$$L(\mu)= \sum_{i=1}^{N} \mu_i - \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \mu_i \mu_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}}{2}$$

- Using kernel trick, the following Lagrange auxiliary function should be maximized:

$$L(\mu) = \sum_{i=1}^{N} \mu_i - \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \mu_i \mu_j y_i y_j \Phi(\mathbf{x_i})^T \Phi(\mathbf{x}_j)}{2}$$

$$= \sum_{i=1}^{N} \mu_i - \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \mu_i \mu_j y_i y_j K(\mathbf{x}_i \bullet \mathbf{x}_j)}{2}$$

➔ We can exploit the advantage of high dimension space without severe complexity increase.

Prediction?

- For prediction for x_{new}, we can use

$$\mathbf{w^*} \text{ (in high dim)} = \sum_{i=1}^{N} \mu_i y_i \Phi(\mathbf{x_i})$$

$$b = (1/N_{S.V}) \sum_{j=S.V.} \{y_i - \mathbf{w}^{*T}\Phi(\mathbf{x_j})\}$$

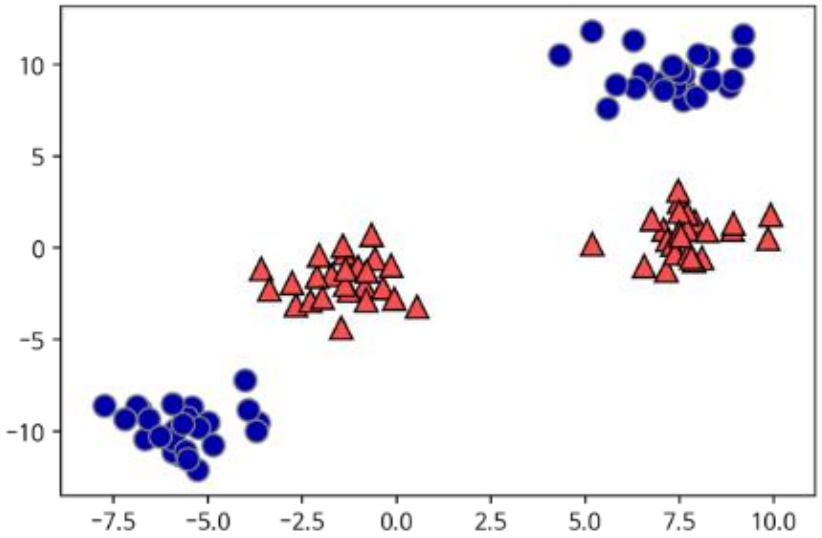
Then apply to below.

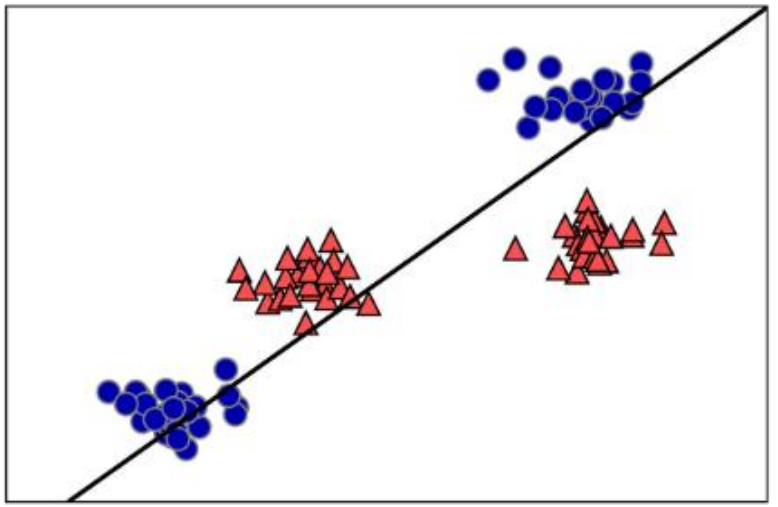
$$\text{sign}(\mathbf{w}^{*T}\Phi(\mathbf{x_{new}}) + b^*)$$

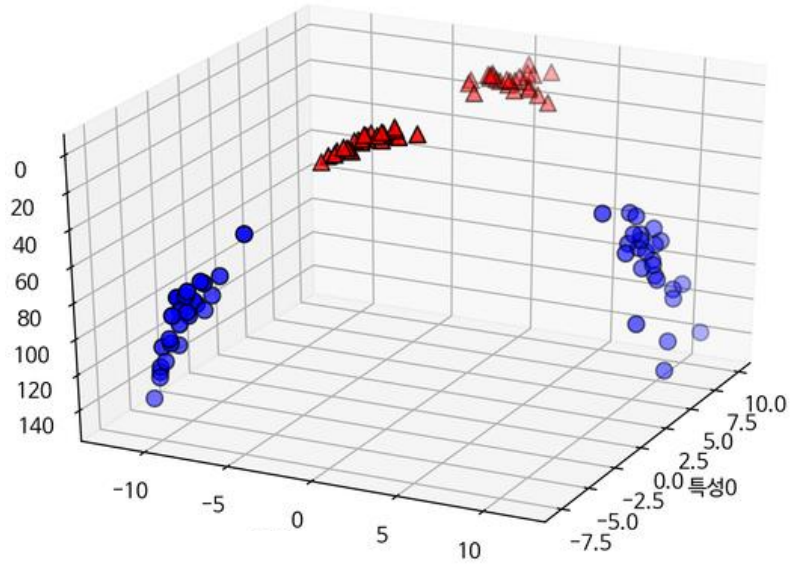
- Do we need the mapping function and high dimension calculation?
- No!

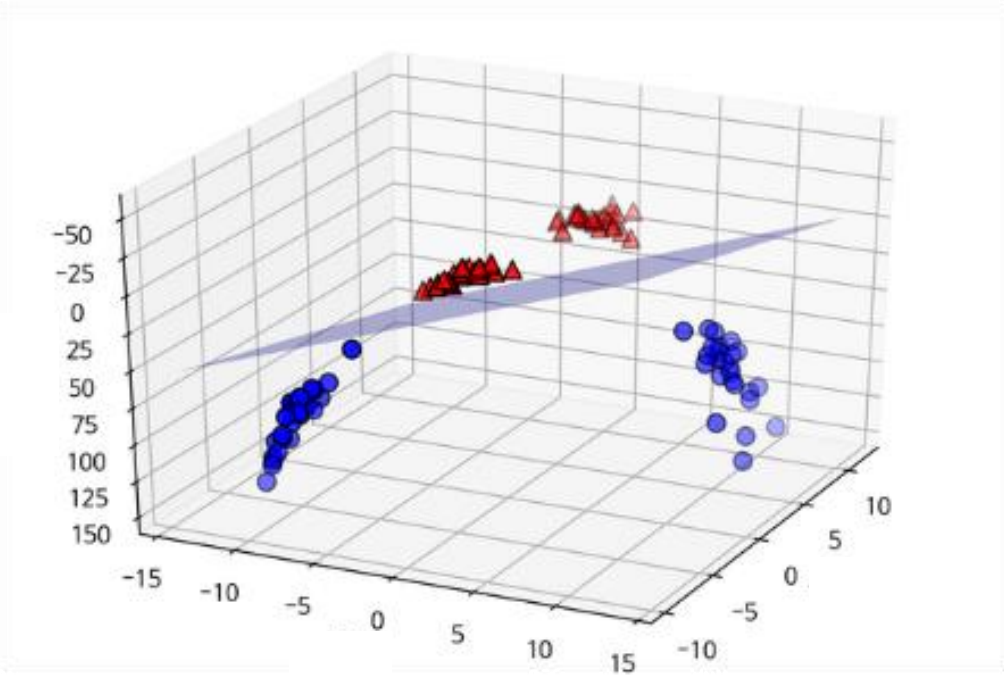
$$\sum_{i=1}^{N} \mu_i y_i \Phi(\mathbf{x_i}) \bullet \Phi(\mathbf{x_{new}}) + (1/N_{S.V}) \sum_{k=S.V.} \{y_i - \sum_{i=1}^{N} \mu_i y_i \, \Phi(\mathbf{x_i})\Phi(\mathbf{x_k})\}$$

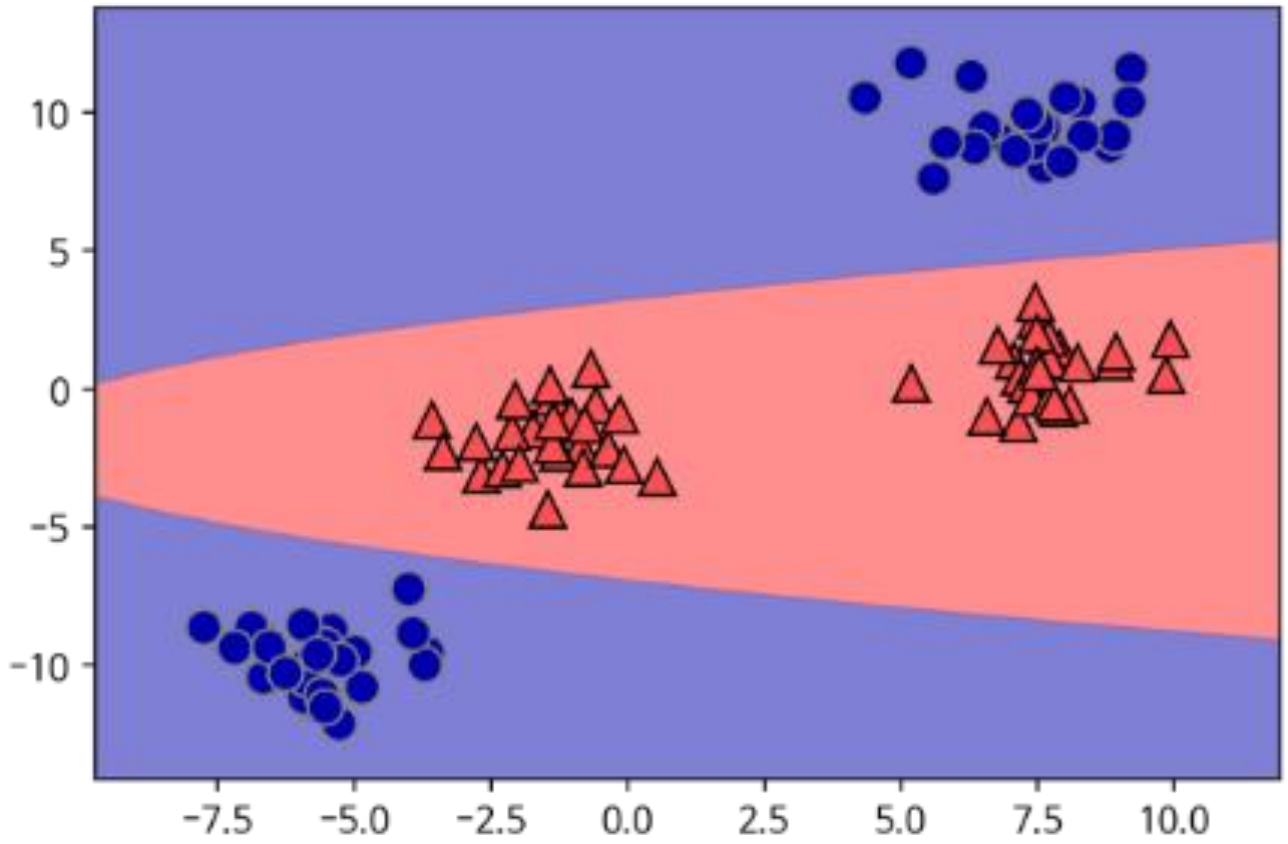
$$= \sum_{i=1}^{N} \mu_i y_i K(\mathbf{x_i}, \mathbf{x_{new}}) + (1/N_{S.V}) \sum_{k=S.V.} \{y_i - \sum_{i=1}^{N} \mu_i y_i \, K(\mathbf{x_i}, \mathbf{x_k})\}$$


10
5
0
−5
−10
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0



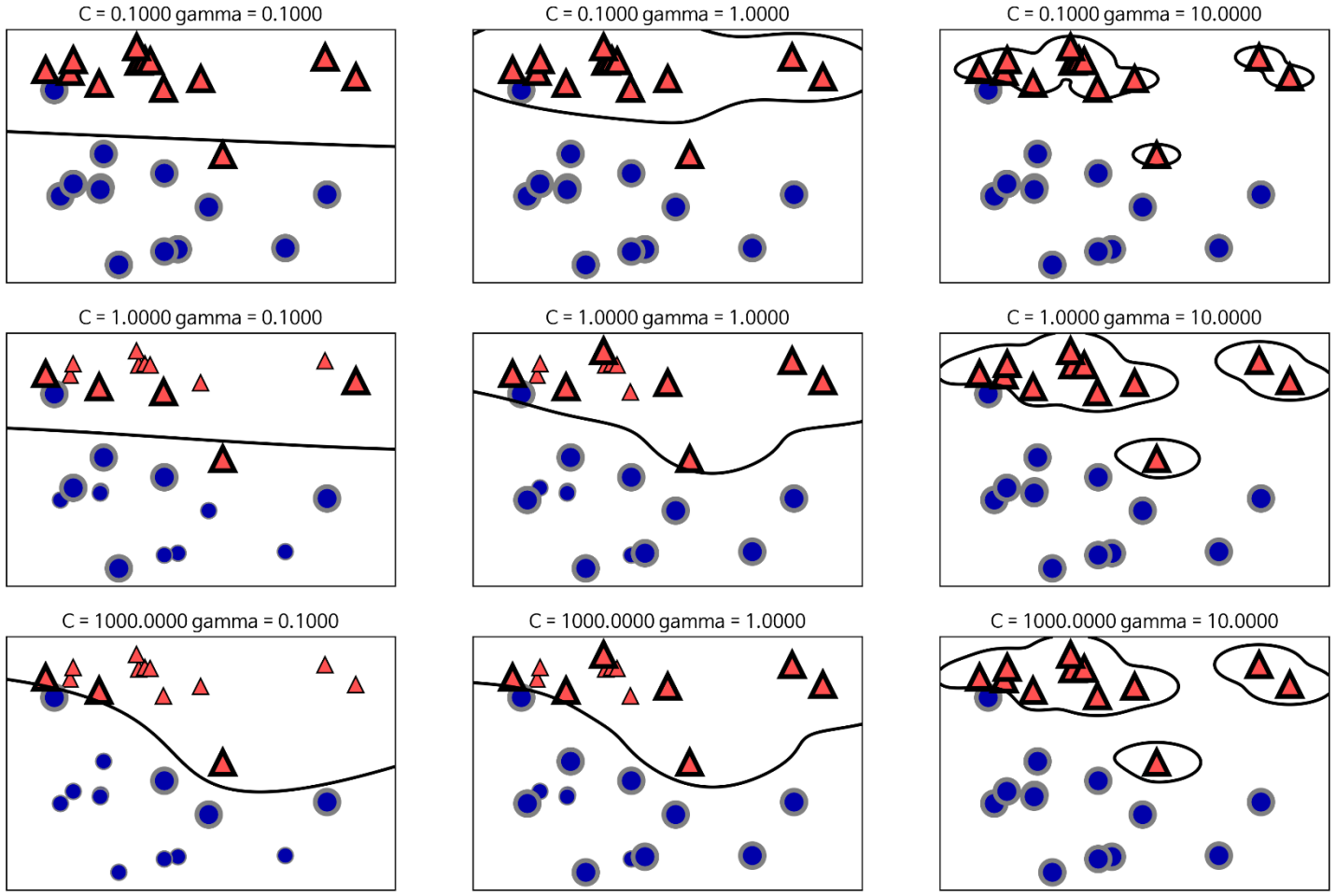

0
20
40
60
80
100
120
140
−10
−5
0
5
10
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
특성0


−50
−25
0
25
50
75
100
125
150
−15
−10
−5
0
5
10
15
10
5
0
−5
−10


10
5
0
−5
−10
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Applying Soft Margin & Kernel Trick

$$k_{rbf}(x_1, x_2) = exp(-\gamma||x_1 - x_2||^2)$$



Checkpoints

- ✓Implementing soft margins in SVM
- ✓Applying kernel trick for SVM
- ✓Coming up next : python coding for SVM