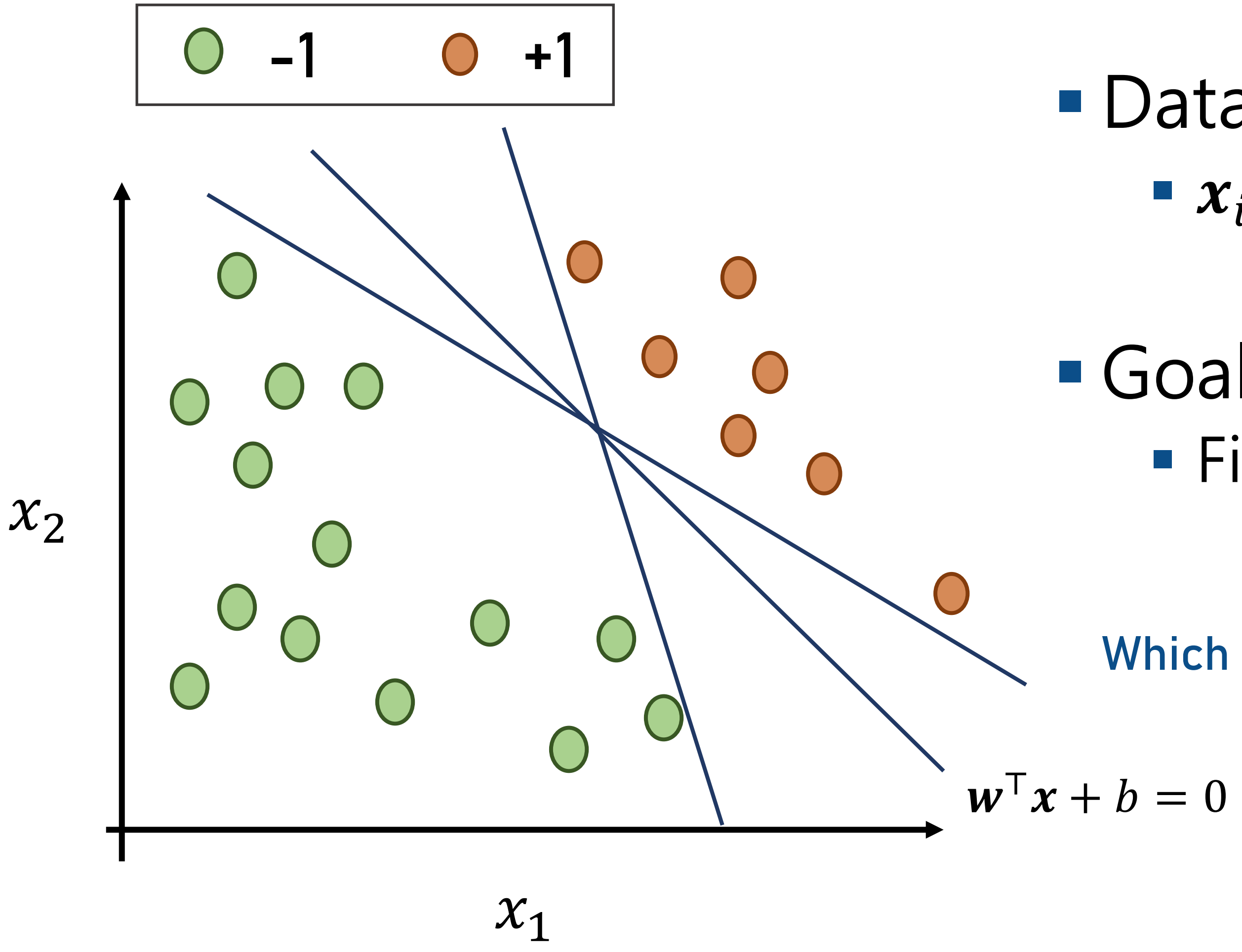
Support Vector Machine (SVM) 2

한양대학교 ERICA
소프트웨어융합대학
COLLEGE OF COMPUTING
인공지능학과
Department of
Artificial Intelligence

정 우 환 (whjung@hanyang.ac.kr)

Fall 2021

Linear SVM

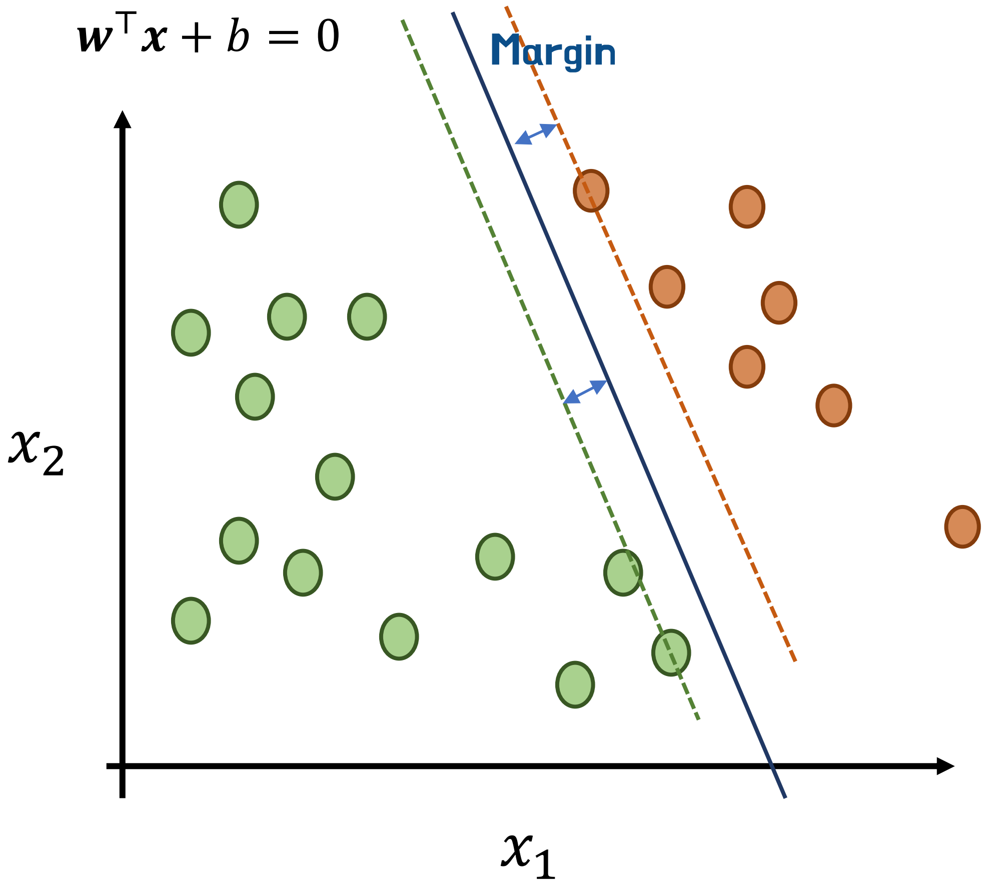


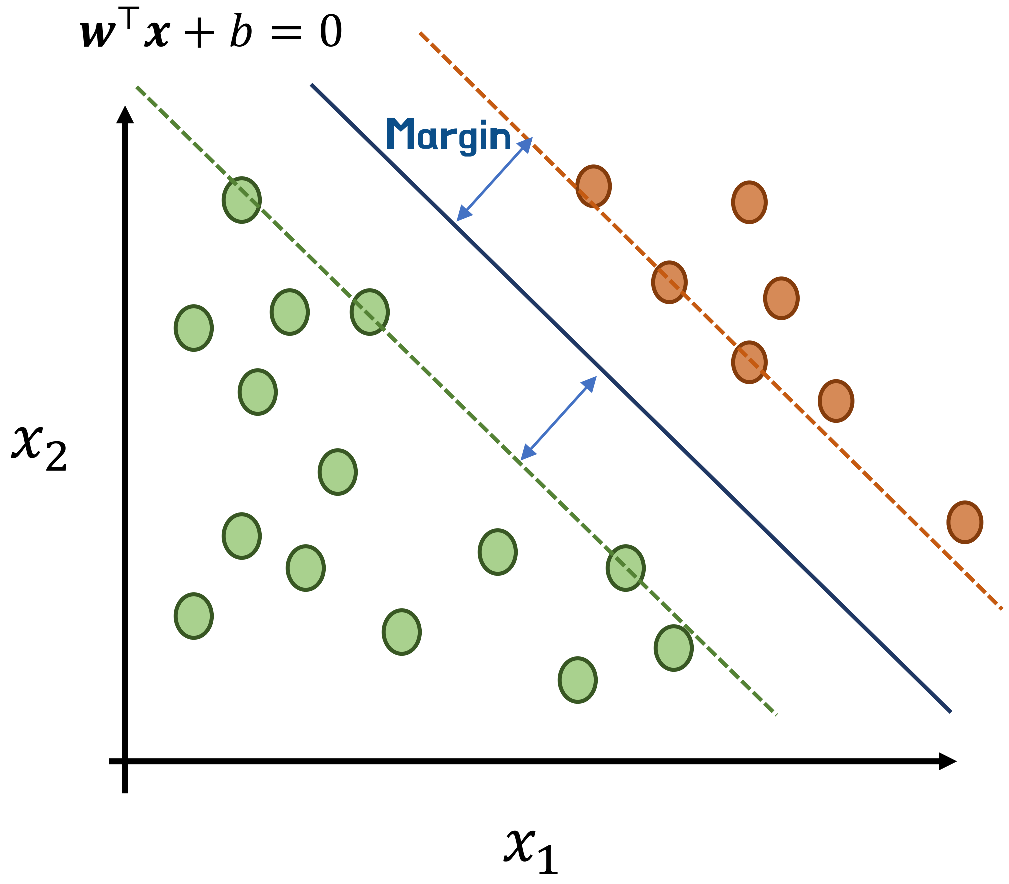
- Data: $< \boldsymbol{x}_i, y_i >$ for $i = 1, \dots n$
  - $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{-1, +1\}$
- Goal
  - Finding a good separating hyperplane

Which is best?

Margin

Maximizing margin over the training set
= Minimizing generalization error





Constrained Optimization Problem

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$subject\ to\ y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1,2,\dots,n$$

- Learnable parameter는 $\boldsymbol{w}, b$
- Loss function $\frac{1}{2}\|\boldsymbol{w}\|_2^2$
  - Margin $\frac{1}{\|\boldsymbol{w}\|_2}$ 을 최대화
- Constraint $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$
  - Training data를 완벽하게 separating
  - 두 boundary ($\boldsymbol{w}^\top \boldsymbol{x}_i + b = \pm 1$) 사이에 데이터가 없음

Lagrangian Formulation

Original Problem

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

$$subject\ to\ y_i(w^T x_i + b) \geq 1, i = 1,2, \ldots \ldots, n$$

Lagrangian multiplier를 이용하여 Lagrangian primal문제로 변환

Lagrangian Primal

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

$$subject\ to\ \alpha_i \geq 0, i = 1,2, \ldots \ldots, n$$

Seoung Bum Kim
https://youtu.be/qFg8cDnqYCI

Lagrangian Formulation

Lagrangian Primal

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

$subject\ to\ \alpha_i \geq 0, i = 1,2, \dots \dots, n$

Convex, continuous이기 때문에 미분 = 0에서 최소값을 가짐

w −

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

② $$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial b} = 0 \longrightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

Seoung Bum Kim
https://youtu.be/qFg8cDnqYCI

Lagrangian Formulation

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

① $\frac{1}{2}\|w\|_2^2 = \frac{1}{2}w^T w$

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

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

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

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

RECALL

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

Seoung Bum Kim
https://youtu.be/qFg8cDnqYCI

Lagrangian Formulation

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

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

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

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

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

RECALL

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

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

Seoung Bum Kim
https://youtu.be/qFg8cDnqYCI

Lagrangian Dual

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

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

$$where \ \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Quadratic programming formulation
- Convex optimization을 통해 풀 수 있음

Original formulation

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

$$subject\ to\ y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1, i = 1,2, \dots, n$$

Lagrangian dual

$$\underset{\alpha}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i{}^\top \boldsymbol{x}_j$$

$$subject\ to\ \alpha_i \geq 0, i = 1,2, \dots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

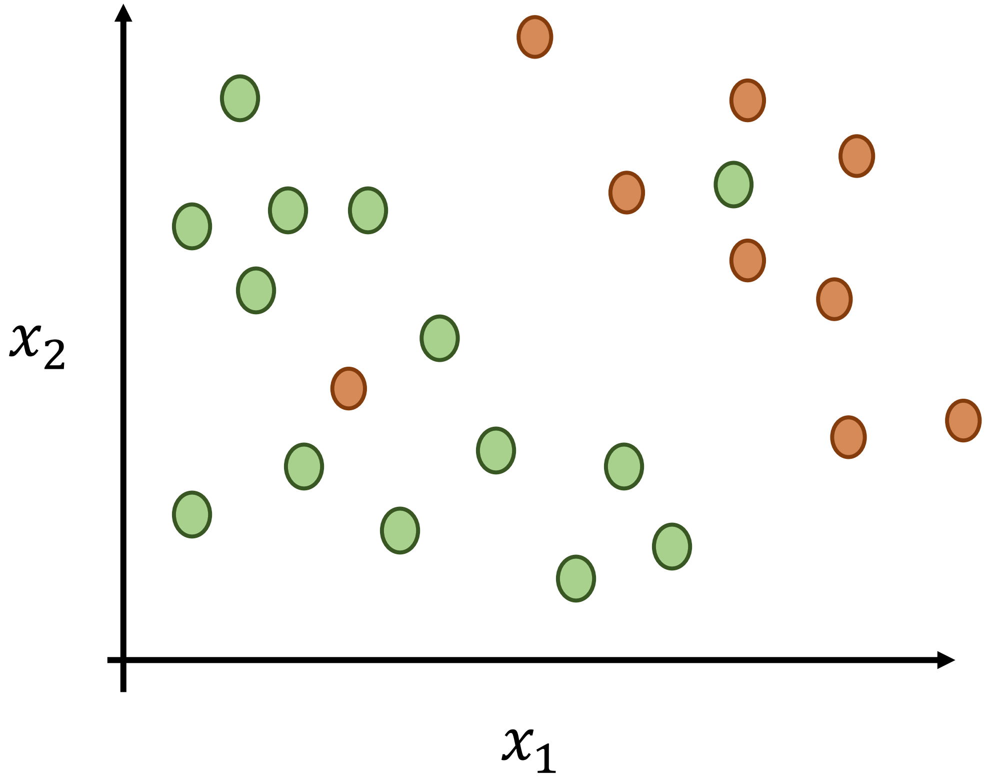
Linear SVM classifier

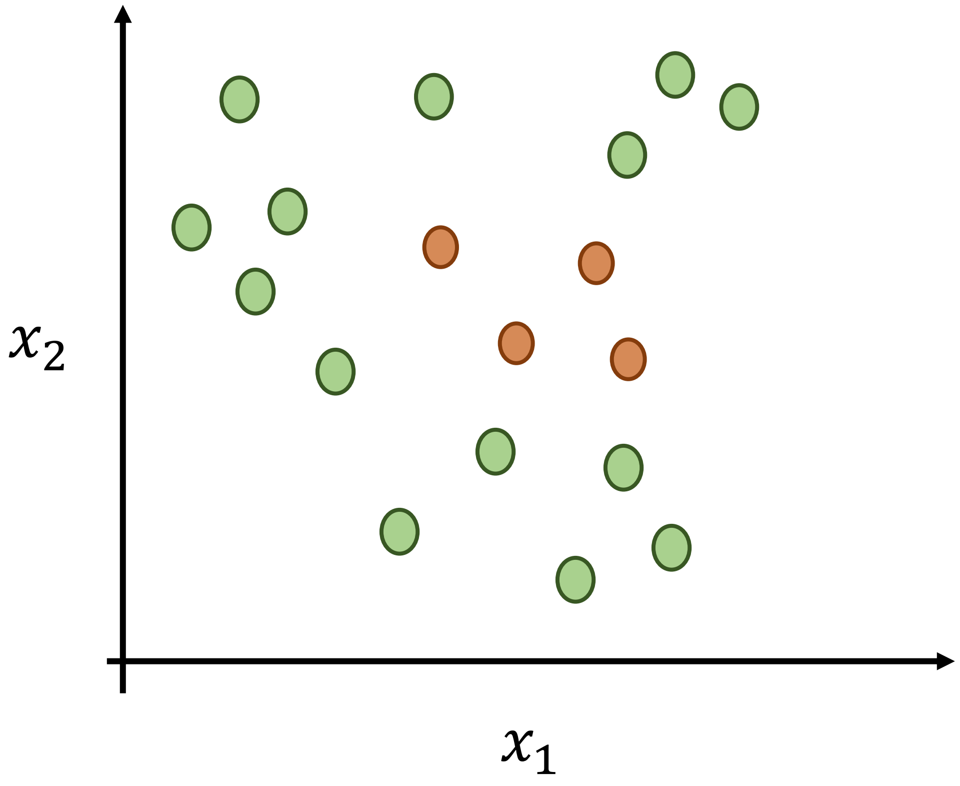
$$\boldsymbol{x} \longrightarrow \boxed{f} \longrightarrow \hat{y}$$

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{w}^*, b^*) \;=\; sign(\boldsymbol{w}^{*\top} \boldsymbol{x} + b^*)$$

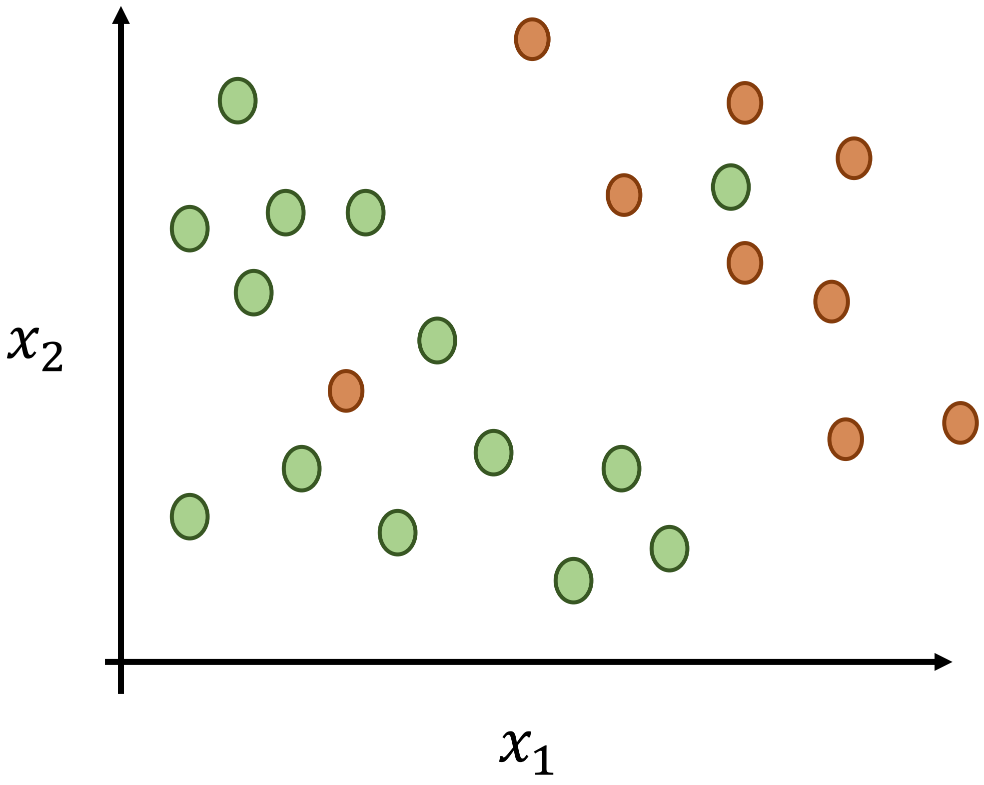
where $\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i$

Linearly Non-separable Data

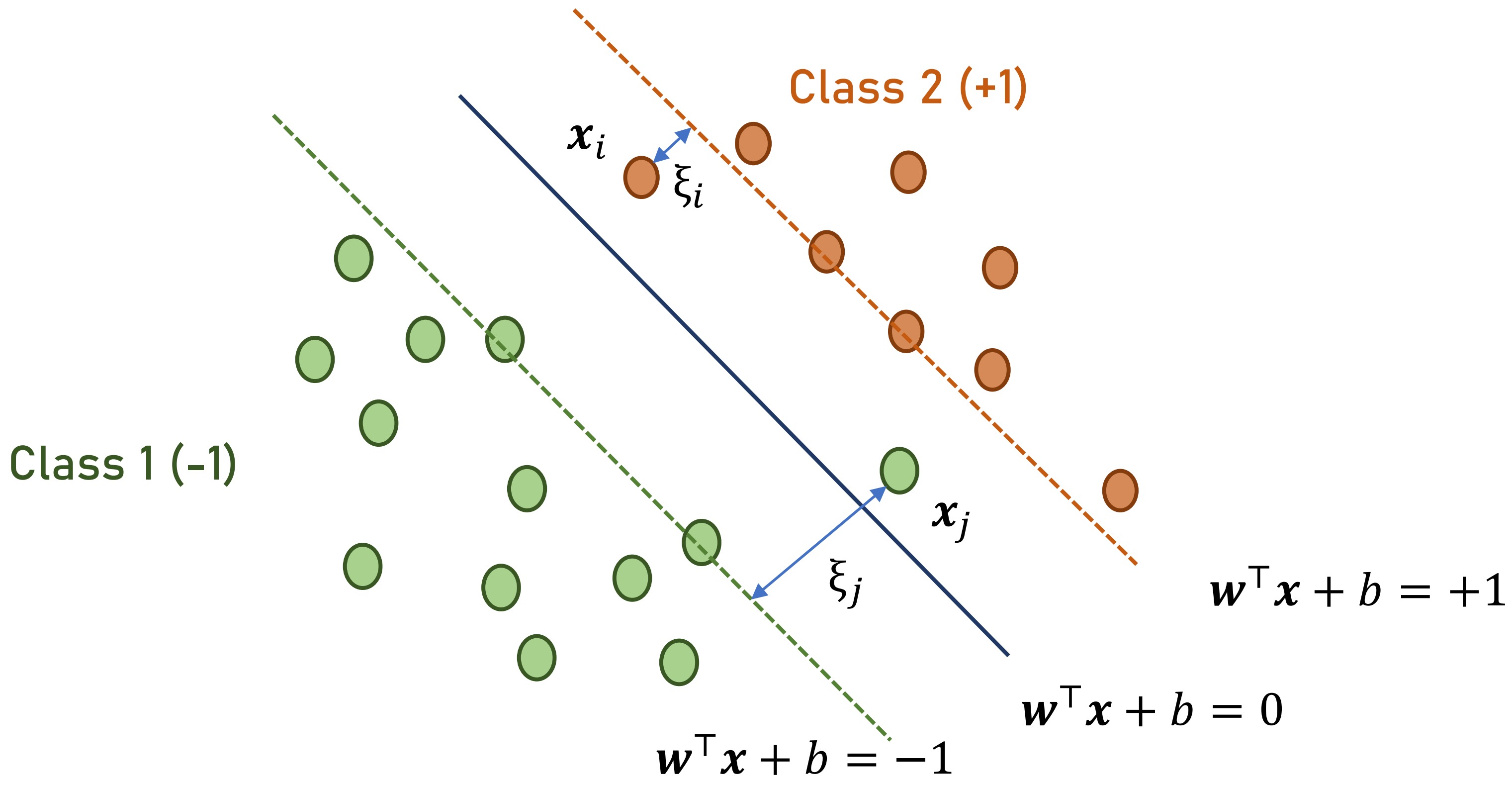




Soft Margin SVM



Soft Margin



Class 2 (+1)
$\boldsymbol{x}_i$
ξ_i
Class 1 (-1)
$\boldsymbol{x}_j$
ξ_j
$\boldsymbol{w}^\top \boldsymbol{x} + b = +1$
$\boldsymbol{w}^\top \boldsymbol{x} + b = 0$
$\boldsymbol{w}^\top \boldsymbol{x} + b = -1$

Optimization Problem

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i$$

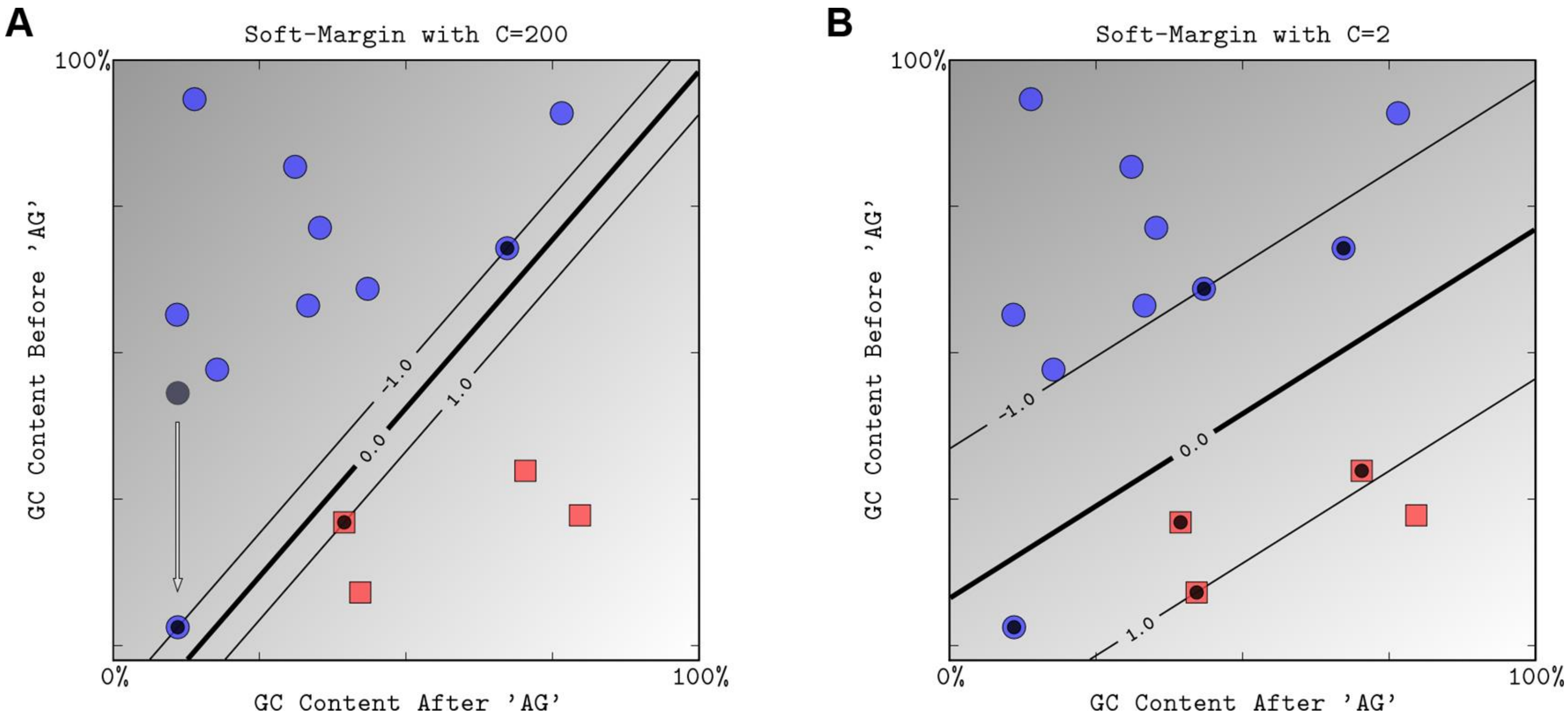
$$\textit{subject to} \quad y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i \qquad \text{For } i = 1,2,\dots,n$$

$$\xi_i \geq 0 \qquad \text{For } i = 1,2,\dots,n$$

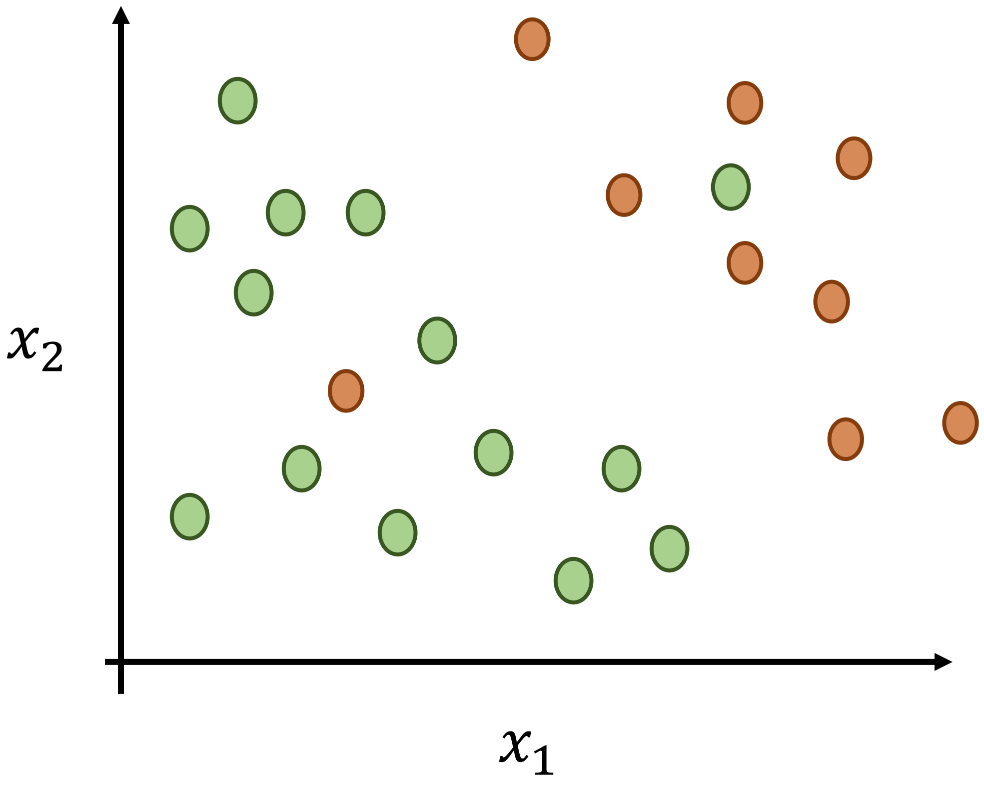
- $C\sum_{i=1}^{n}\xi_i$: 예외의 최소화
  - C를 활용해 허용할 training error를 결정
  - C ↑: training error를 적게허용
  - C ↓: training error를 많이허용
- Linearly separable 하지 않더라도 해가 존재

Seoung Bum Kim
https://youtu.be/qFg8cDnqYCI

Soft-margin for Linearly Separable Problem

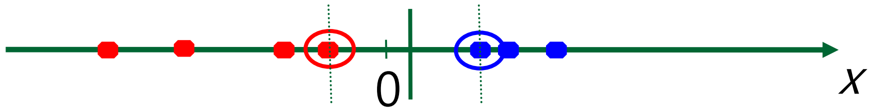


Kernel Trick

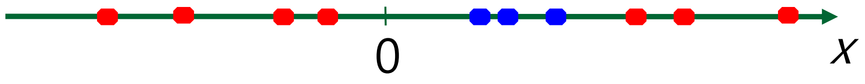


Non-linearly Separable Problems

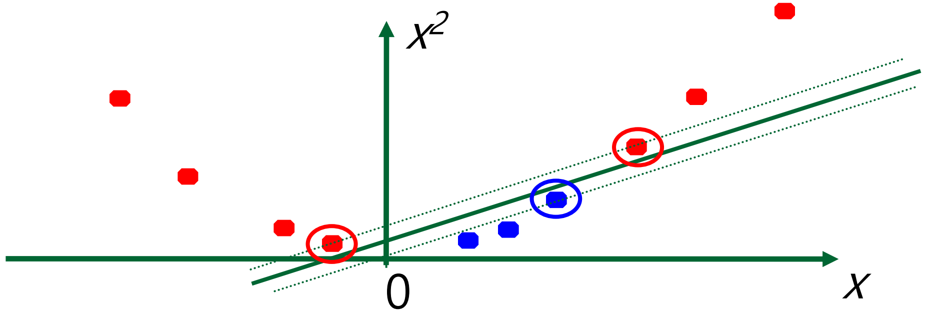
- Datasets that are linearly separable with noise work out great:



- But what are we going to do if the dataset is just too hard?



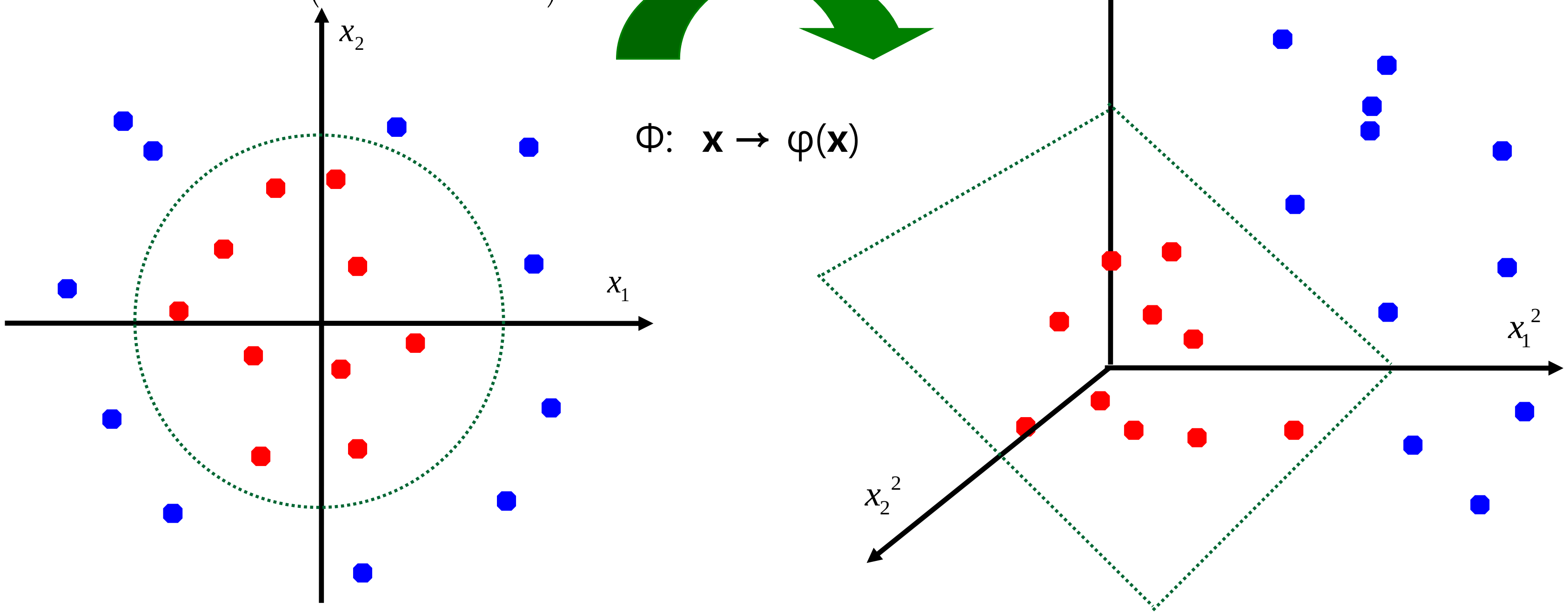
- How about… mapping data to a higher-dimensional space:



Non-linearly Separable Problems

- General idea: the original input space(x) can be mapped to some higher-dimensional feature space(φ(x)) where the training set is separable:

$$\phi(\mathbf{x}) = \left(x_1^2, x_2^2, \sqrt{2x_1x_2}\right)$$



If data are mapped into higher a space of sufficiently high dimension, then they will in general be linearly separable;
N data points are in general separable in a space of N-1 dimensions or more!!!

Kernel Mapping

Linear SVM formulation (Lagrangian dual)

$$\underset{\alpha}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i{}^{\top} \boldsymbol{x}_j$$

$$subject\ to\ \alpha_i \geq 0, i = 1,2, \dots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

SVM formulation (transformation)

$$\underset{\alpha}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^{\top} \phi(\boldsymbol{x}_j)$$

$$subject\ to\ \alpha_i \geq 0, i = 1,2, \dots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Kernel Mapping

SVM formulation (transformation)

$$\underset{\alpha}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_j)$$

$$subject\ to\ \alpha_i \geq 0, i = 1,2, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

SVM formulation (kernel)

$$\underset{\alpha}{\text{minimize}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$subject\ to\ \alpha_i \geq 0, i = 1,2, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

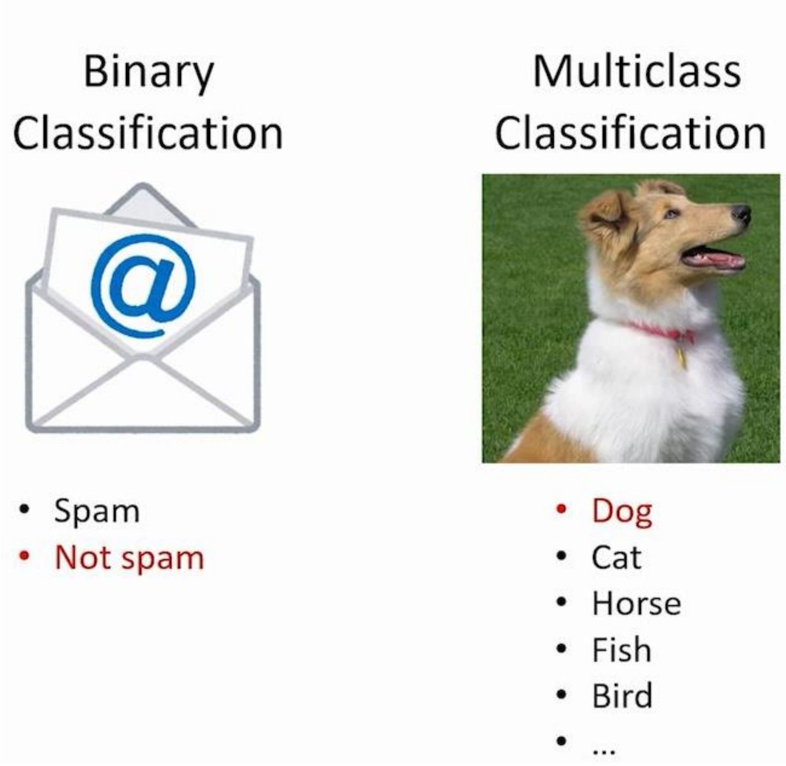
Kernel Functions

- Linear
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j$
  - Mapping Φ: $\boldsymbol{x} \to \varphi(\boldsymbol{x})$, where $\varphi(\boldsymbol{x})$ is $\boldsymbol{x}$ itself
- Polynomial of power p
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\boldsymbol{x}_i^\top \boldsymbol{x}_j + 1\right)^P$
- Gaussian (radial-basis function):
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}}$
- Sigmoid (Neural net style)
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\left(\kappa\, \boldsymbol{x}_i^\top \boldsymbol{x}_j - \delta\right)$

SVM for Multi-class Classification

One-to-One

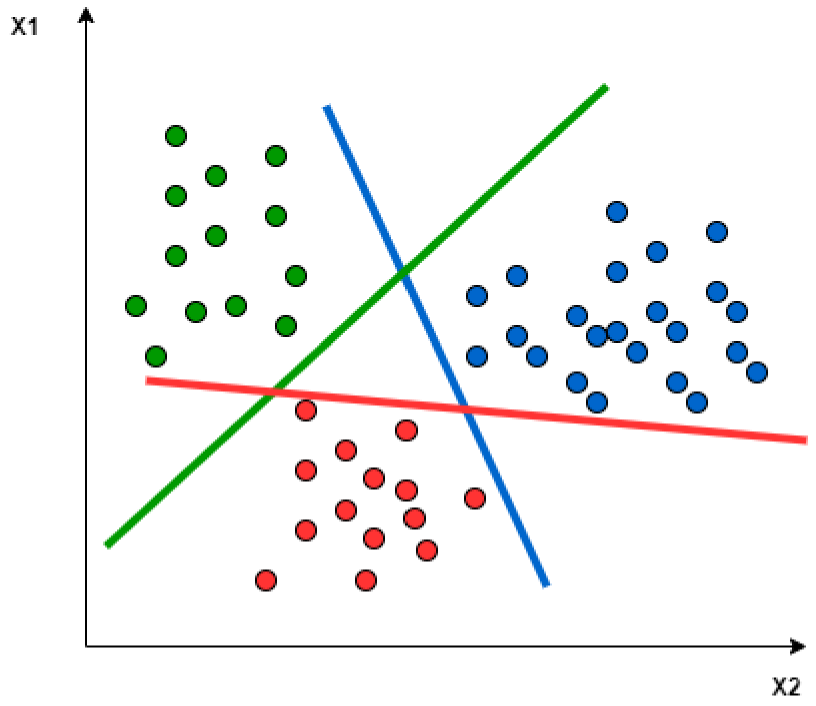
One-to-Rest



그림출처: Ertuğrul Demir https://www.kaggle.com/c/lish-moa/discussion/180500

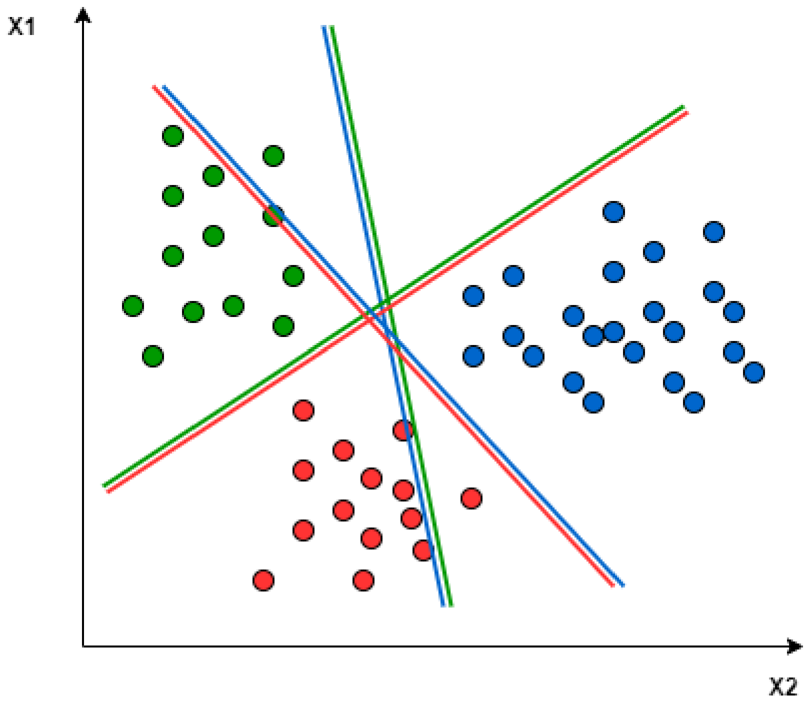
One-to-Rest

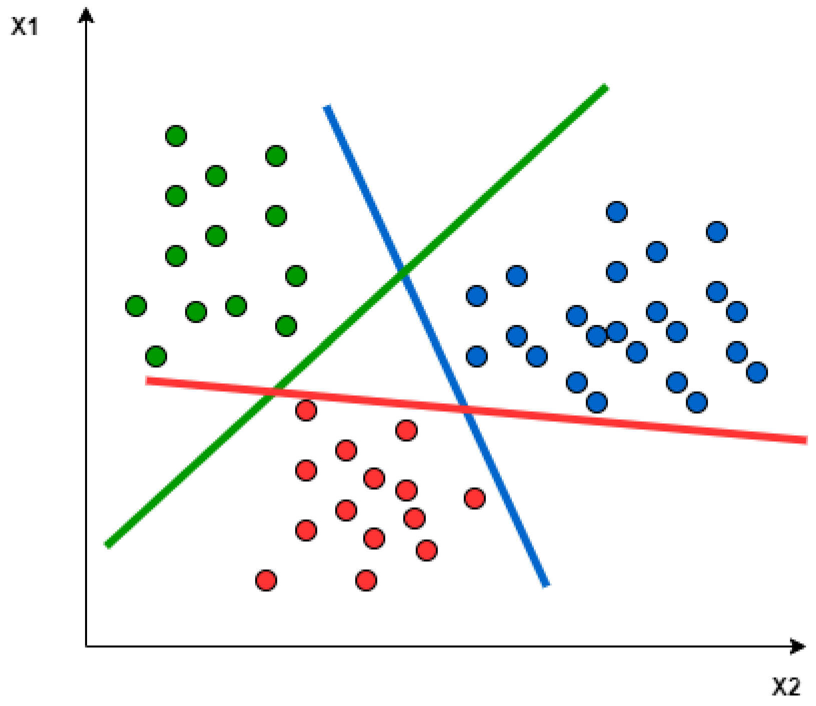
- Splitting the multi-class dataset into multiple binary classification problems
  - Example) Multi-class problem: 'red', 'blue', 'green'
    - **Binary Classification 1**: red vs [blue, green]
    - **Binary Classification 2**: blue vs [red, green]
    - **Binary Classification 3**: green vs [red, blue]
- Number of datasets (models): $\# \, classes$
- Predictions are made using the model with the highest confidence



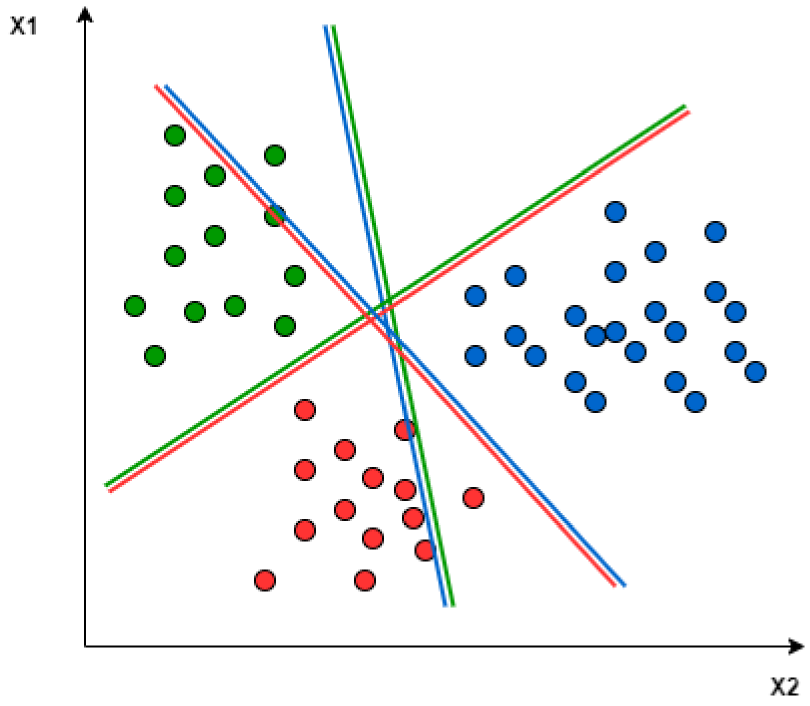
One-to-One

- Splitting the multi-class dataset into multiple binary classification problems
  - Example) Multi-class problem: 'red', 'blue', 'green'
    - **Binary Classification 1**: red vs. blue
    - **Binary Classification 2**: red vs. green
    - **Binary Classification 3**: red vs. yellow
    - **Binary Classification 4**: blue vs. green
    - **Binary Classification 5**: blue vs. yellow
    - **Binary Classification 6**: green vs. yellow
- Number of datasets (models): $\frac{n_{class}(n_{class}-1)}{2}$
- Prediction
  - Voting





One-to-Rest



One-to-One

https://www.baeldung.com/cs/svm-multiclass-classification

SVM vs. Neural Network

- **SVM**
  - Deterministic algorithm
  - Nice generalization properties
  - Hard to learn – learned in batch mode using quadratic programming techniques
  - Using kernels can learn very complex functions

- **Neural Network**
  - Nondeterministic algorithm
  - Generalizes well but doesn’t have strong mathematical foundation
  - Can easily be learned in incremental fashion
  - To learn complex functions—use multilayer perceptron (nontrivial)

References

- Andrew W. Moore's slides:
  - http://www.cs.cmu.edu/~awm/tutorials
- Seoung Bum Kim's slides:
  - https://youtu.be/qFg8cDnqYCI
- Kyuseok Shim's slides