Support Vector Machines

Support Vector Machine (SVM)

The **S**upport **V**ector **M**achine (SVM) has been shown to be able to achieve good generalization performance for classification of high-dimensional datasets and its training can be framed as solving a quadratic programming problem.

• Usually, we try to maximize classification performance for the training data.

• But if the classifier is too fit for the training data, the classification ability for unknown data (i.e., the generalization ability) is degraded.

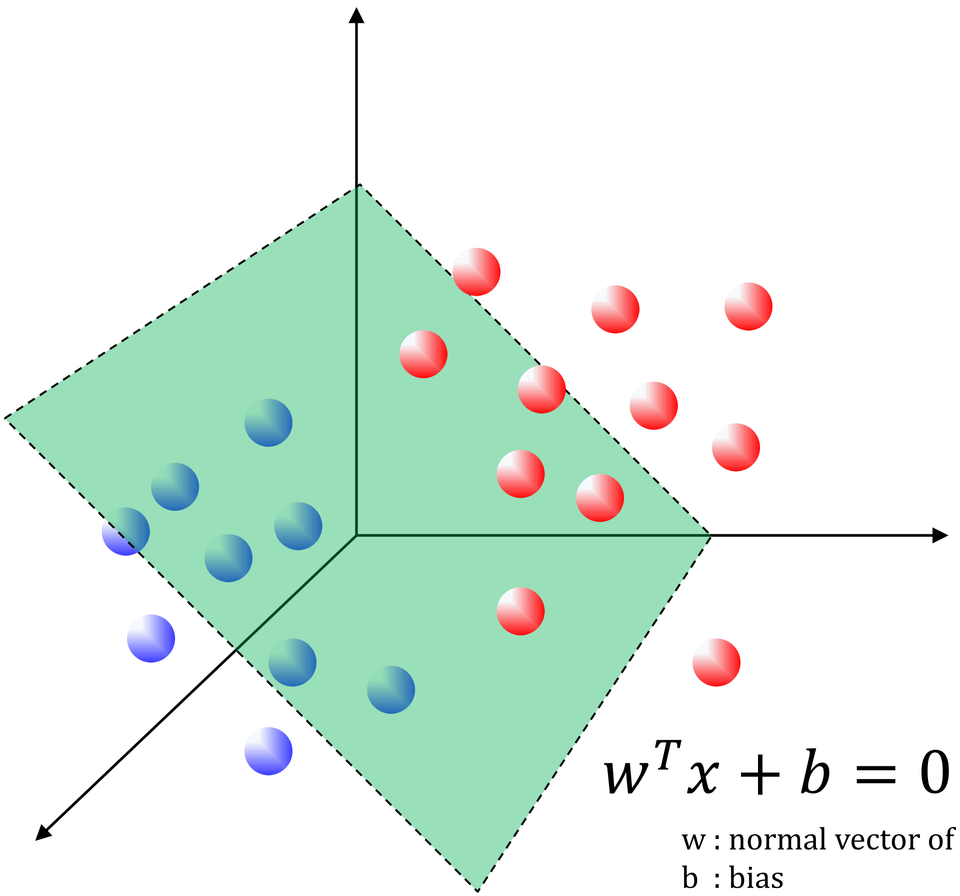
• There is a trade-off between the generalization ability and fitting to the training data.

• SVM is trained so that the direct decision function maximizes the generalization ability.

• SVM is based on statistical learning theory

[1] B.E. Boser *et al.* A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsburgh, 1992.
[2] V. Vapnik. The Nature of Statistical Learning Theory. 2nd edition, Springer, 1999.

Separating Hyperplane

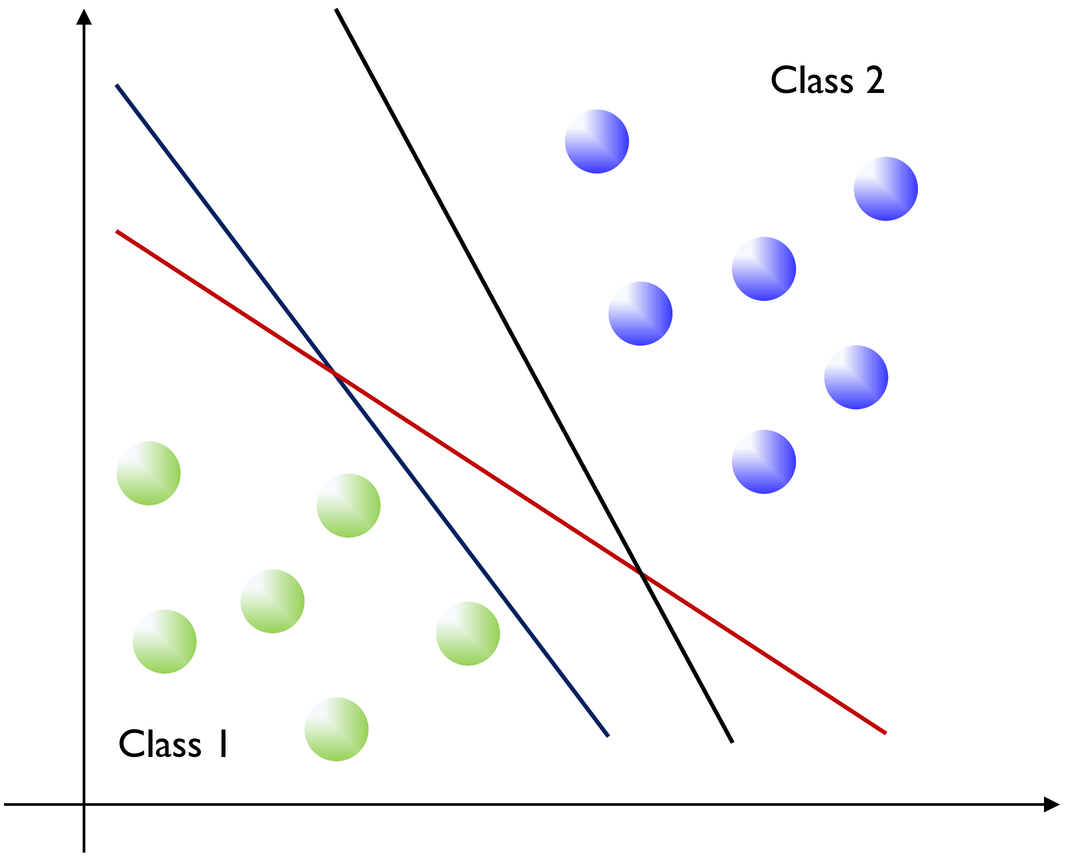


$$w^T x + b = 0$$

w : normal vector of the hyperplane
b : bias

목적: Training data (X, Y)를 가지고 w와 b를 찾자!

Separating Hyperplane



Two class classification 문제

두 class를 나누는 hyperplane은 무한히 많음

어떤 hyperplane이 가장 "좋은" hyperplane 인가?

"좋다"는 것의 기준은?

Separating Hyperplane

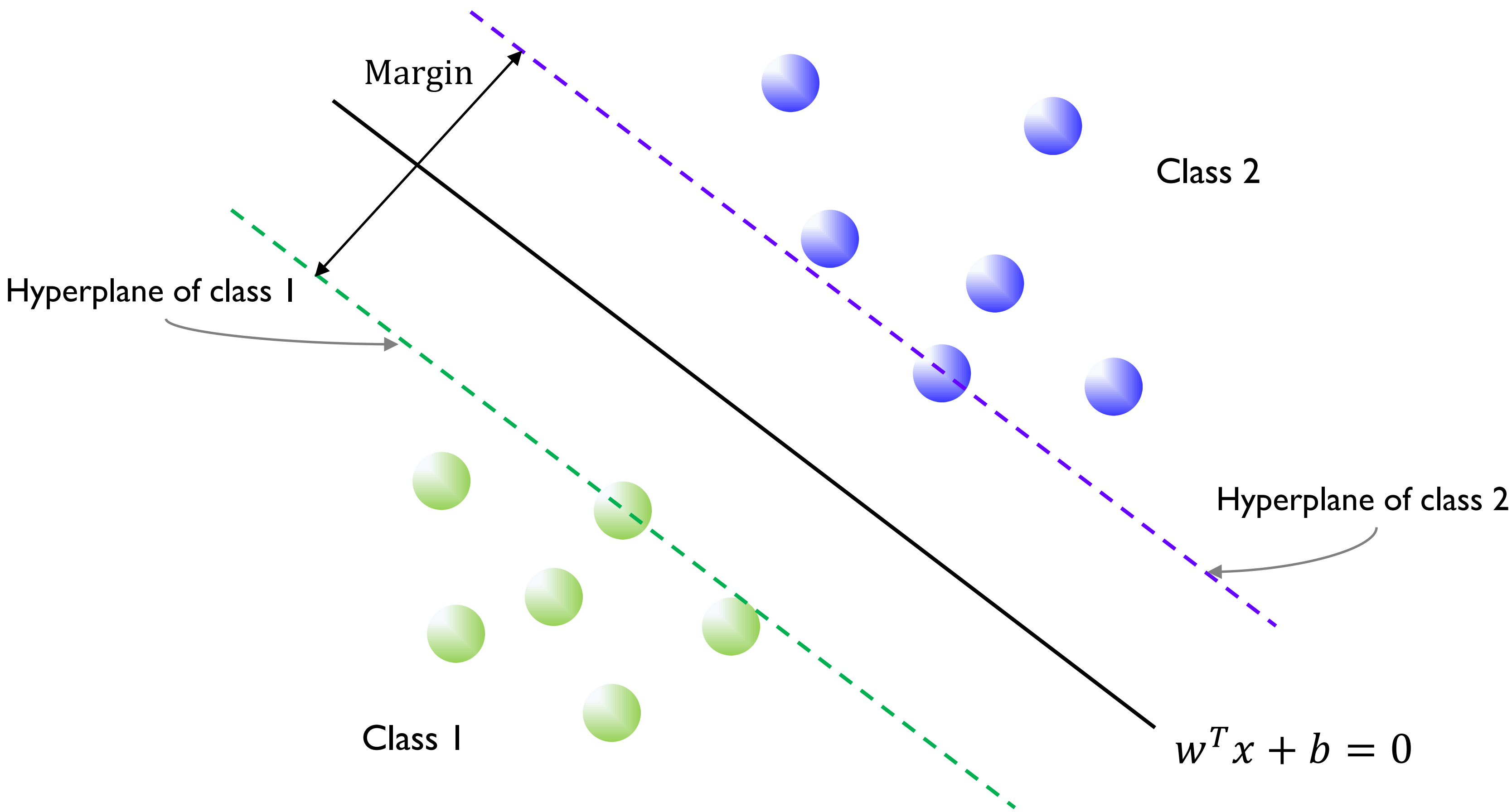
Maximizing margin over the training set

= minimizing generalization error

= good prediction performance

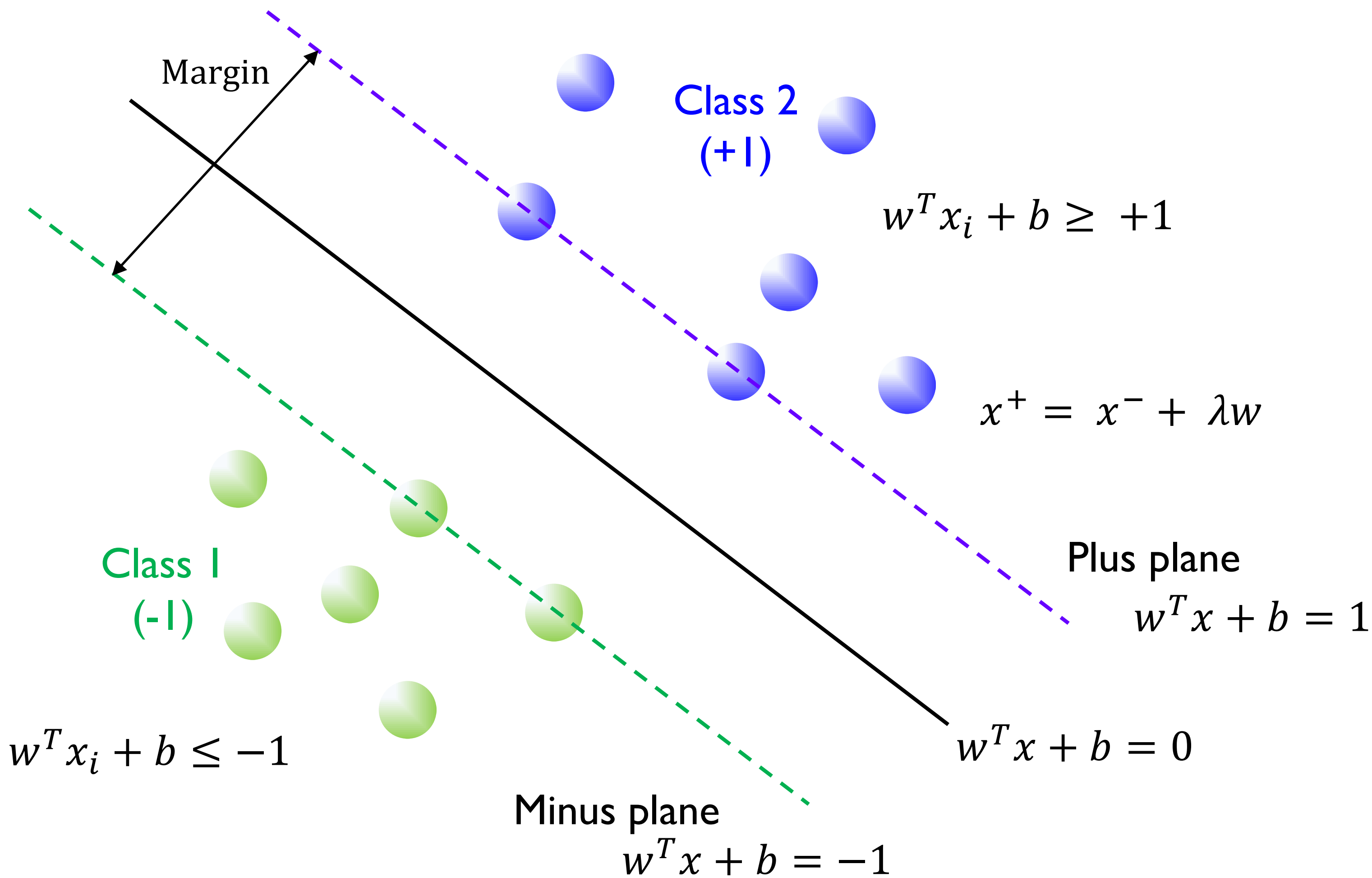
So, what is the **margin**?

Geometric Margin



- Margin: 각 클래스에서 가장 가까운 관측치 사이의 거리
- Margin은 w (기울기) 로 표현 가능

Geometric Margin



Geometric Margin

$w^T x^+ + b = 1$ $\quad x^+$가 plus plane 위의 점

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

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

$-1 + \lambda w^T w = 1$ $\quad x^-$ 는 minus plane 위의 점

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

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

The vector norm $\|W\|_p \ for\ p = 1,2,3, \ldots$

$$\|W\|_p = \left(\sum_i |w_i|^p\right)^{1/p}$$

L_2 norm

$$\|W\|_2 = \left(\sum_i |w_i|^2\right)^{1/2} = \sqrt{w_1^2 + w_2^2 + \cdots w_n^2} = \sqrt{W^T W}$$

Geometric Margin

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

w의 L_2 norm이 제곱근을 포함하고 있기 때문에 계산이 어려움
→ 계산상의 편의를 위해 다음과 같은 형태로 목적함수를 변경

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

Convex Optimization Problem

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

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

- Decision variable은 w 와 b
- Objective function은 separating hyperplane으로 부터 정의된margin의 역수
- Constraint는 training data를 <u>완벽하게</u> separating하는 조건
- Objective function is quadratic, and constraint is linear → quadratic programming → convex optimization → globally optimal solution exists (전역최적해 존재)
- Training data가 linearly separable한 경우에만 해가 존재함

Lagrangian Formulation

Original Problem

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

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

Lagrangian multiplier를 이용하여 Lagrangian primal문제로 변환

Lagrangian Primal

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

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

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

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

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

Lagrangian Formulation

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

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

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

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

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

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

RECALL

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

Lagrangian Formulation

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

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

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

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

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

RECALL

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

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

Lagrangian Dual

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

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

① ②

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

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

Lagrangian Dual

따라서 Lagrangian dual은 다음과 같은 quadratic programming formulation

quadratic

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

Inner product

linear

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

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

- Original problem formulation (primal formulation) 보다 풀기 쉬운 형태
- Objective function is quadratic and constraint is linear → quadratic programming → convex optimization → globally optimal solution exists (전역최적해 존재)
- Optimization 문제가 x들의 inner product만으로 표현됨 (nonlinear case로 확장했을 때 좋은 성질)
- Lagrangian dual의 decision variable은 α이며, quadratic optimization을 풀어 α에 대한 solution을 얻을 수 있음 $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots \dots, \alpha_n^*)$

Lagrangian Dual

(w,b,α)가 Lagrangian dual problem의 최적해가 되기 위한 조건

KKT (Karush-Kuhn-Tucker) conditions:

① Stationarity

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

② Primal feasibility $y_i(w^T x_i + b) \geq 1, i = 1,2, \dots \dots, n$

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

④ Complementary slackness $\alpha_i(y_i(w^T x_i + b) - 1) = 0$

Characteristics of the Solutions

$$\alpha_i(y_i(w^T x_i + b) - 1) = 0\ , i = 1,2, \dots \dots, n$$

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

x_i 가 plus-plane 또는 minus-plane (마진) 위에 있음 (**support vector**) 해당 $\alpha_i > 0$

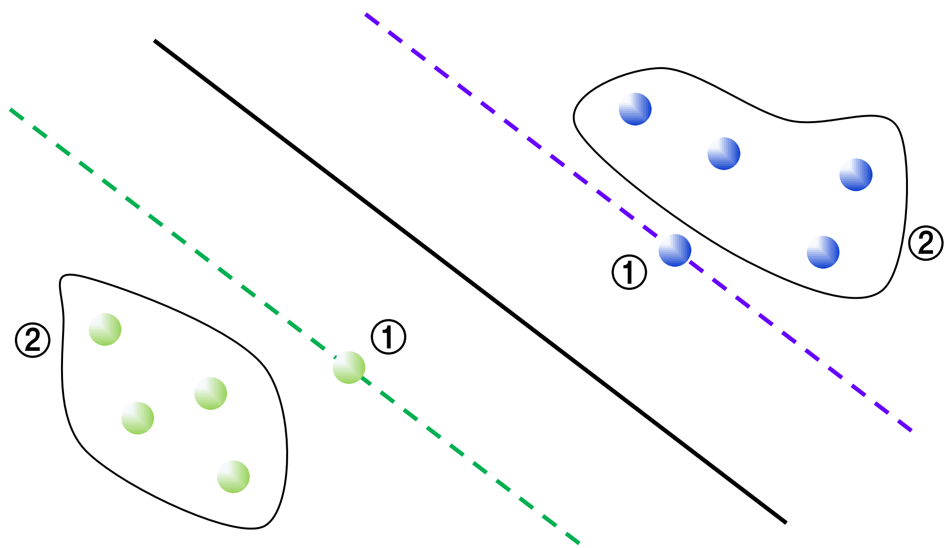
② $\alpha_i = 0\ and\ y_i(w^T x_i + b) - 1 \neq 0$

x_i 가 plus-plane 또는 minus-plane 위에 있지 않음 해당 $\alpha_i = 0$

Hyperplane을 구축하는데 영향을 미치지 않음

SVM이 outlier에 robust (강건)한 이유

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



Characteristics of the Solutions

x_i가 support vector인 경우에만 $\alpha_i^* \geq 0$이므로

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i = \sum_{i \in SV} \alpha_i^* y_i x_i$$

즉, support vector만 이용하여 optimal hyperplane (decision boundary) 을 구할 수 있다 (sparse representation!)

또한, 다음과 같이 임의의 support vector 하나를 이용하여 b^*를 구할 수 있다.

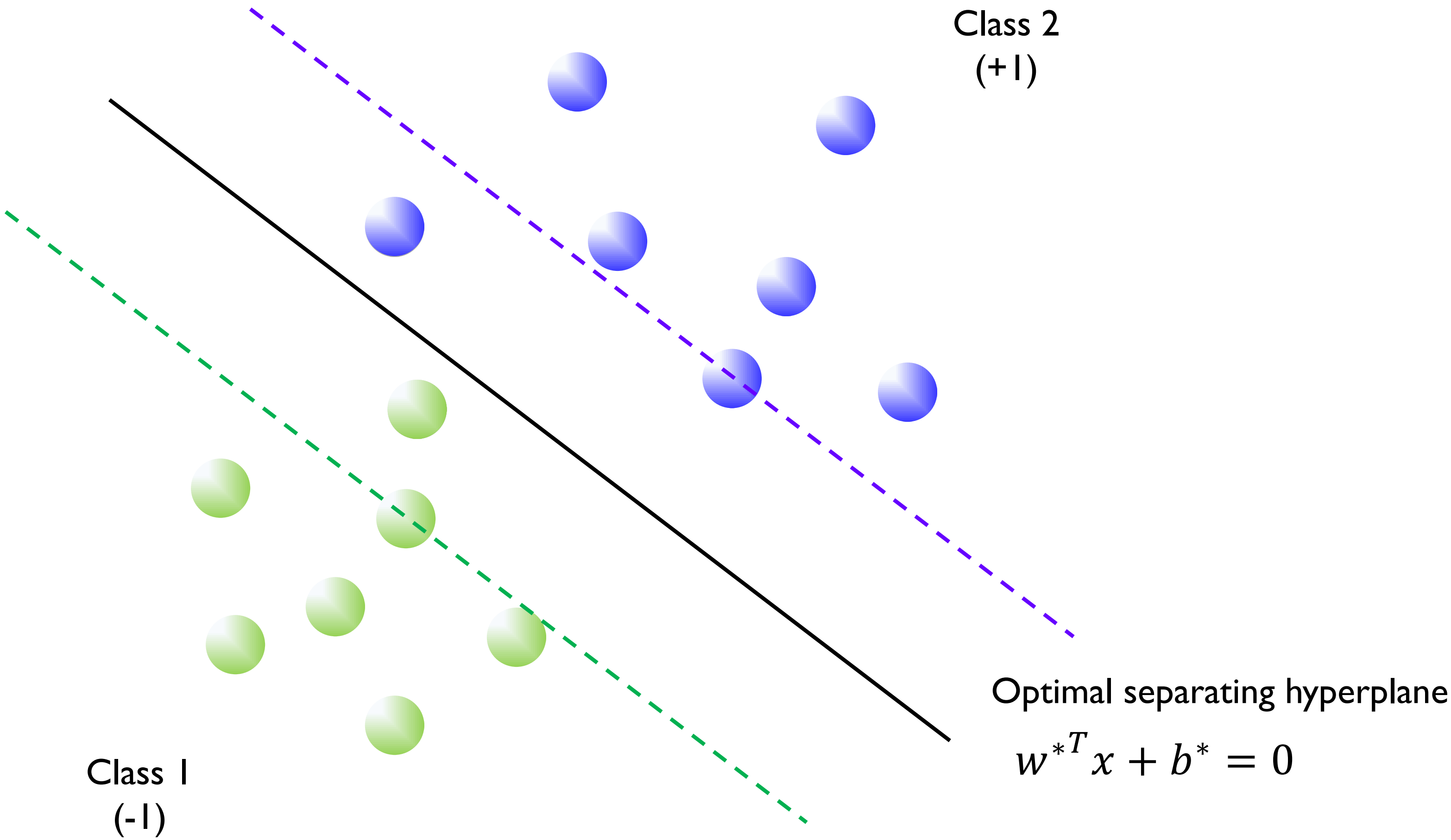
$$w^{*T} x_{sv} + b^* = y_{sv}$$

$$w^{*T} x_{sv} + b^* = \boxed{\sum_{i=1}^{n} \alpha_i^* y_i x_i^T x_{sv} + b^* = y_{sv}}$$

$$b^* = y_{sv} - \sum_{i=1}^{n} \alpha_i^* y_i x_i^T x_{sv}$$

Classifying New Data Points

Training data로부터 w^*와 b^*를 계산하였다면, 새로운 데이터에 대한 분류를 시행할 수 있음



Classifying New Data Points

새로운 데이터가 optimal separating hyperplane보다 밑에 있음

$$w^{*T} x_{new} + b^* < 0$$

→ class 1 (-1) 로 예측 $\hat{y}_{new} = -1$

새로운 데이터가 optimal separating hyperplane보다 위에 있음

$$w^{*T} x_{new} + b^* > 0$$

→ class 2 (+1) 로 예측 $\hat{y}_{new} = +1$

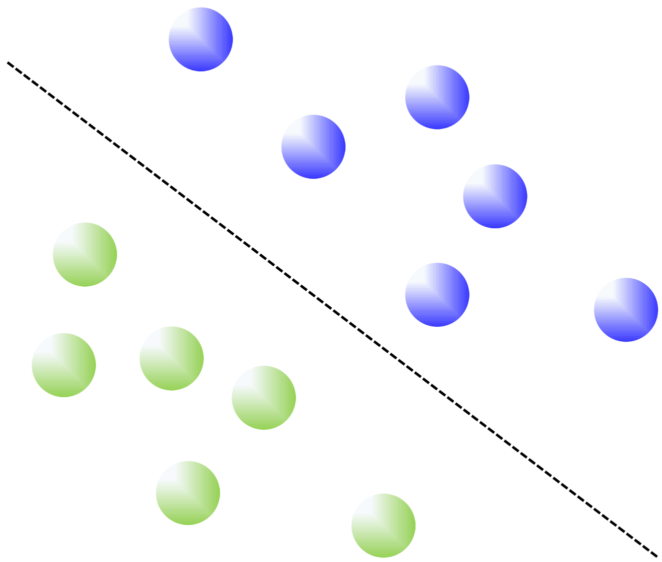
따라서 다음과 같이 새로운 데이터에 class를 부여할 수 있음

$$\hat{y}_{new} = \text{sign}(w^{*T} x_{new} + b^*)$$

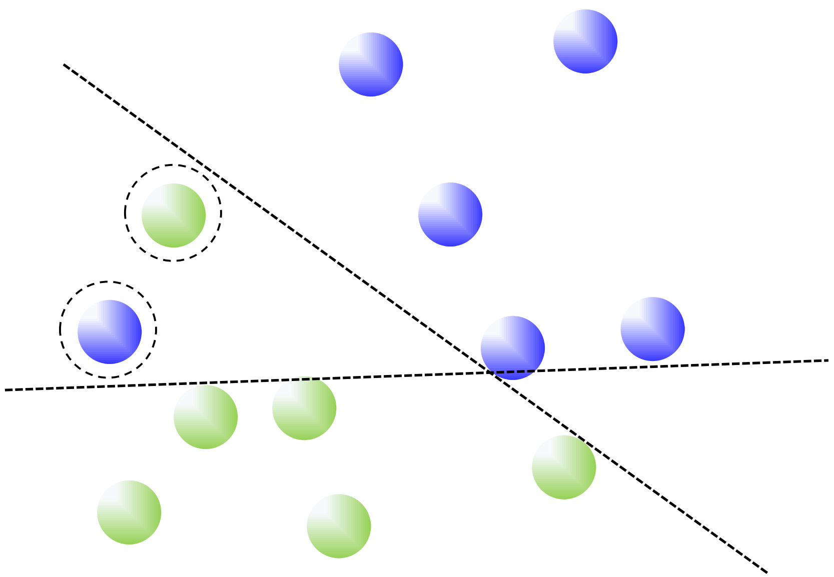
$$= \text{sign}(\sum_{i \in SV} \alpha_i^* y_i x_i^T x_{new} + \ b^*)$$

Linearly Nonseparable Case
(Soft Margin SVM)

Linearly Nonseparable Problems



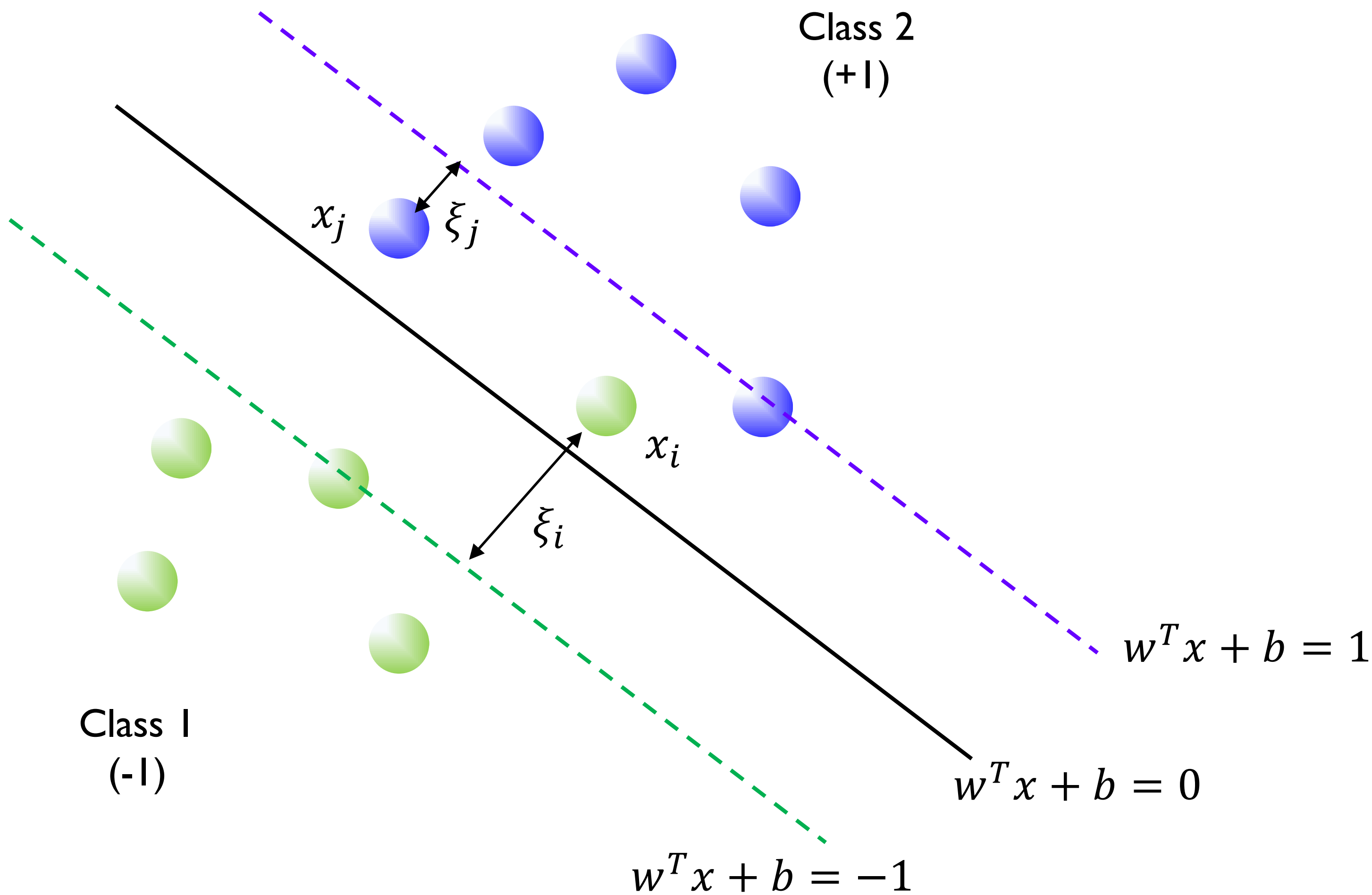
Linearly separable



Linearly nonseparable

Linear decision boundary를 이용하여 완벽하게 나누는 것은 불가능
→ Error 허용

Linearly Nonseparable Problems



Convex Optimization Formulation

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

$$\textit{subject to}\ y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1,2,\ldots\ldots,n$$

- Decision variable은 w, b
- Slack variable ξ_i ≥0 도입하여 training error 허용 → 그렇다고 마냥 크게 할 수 없음
- Objective function에 penalty 를 추가하여 억제
- C는 margin과 training error에 대한 trade-off를 결정하는 tuning parameter

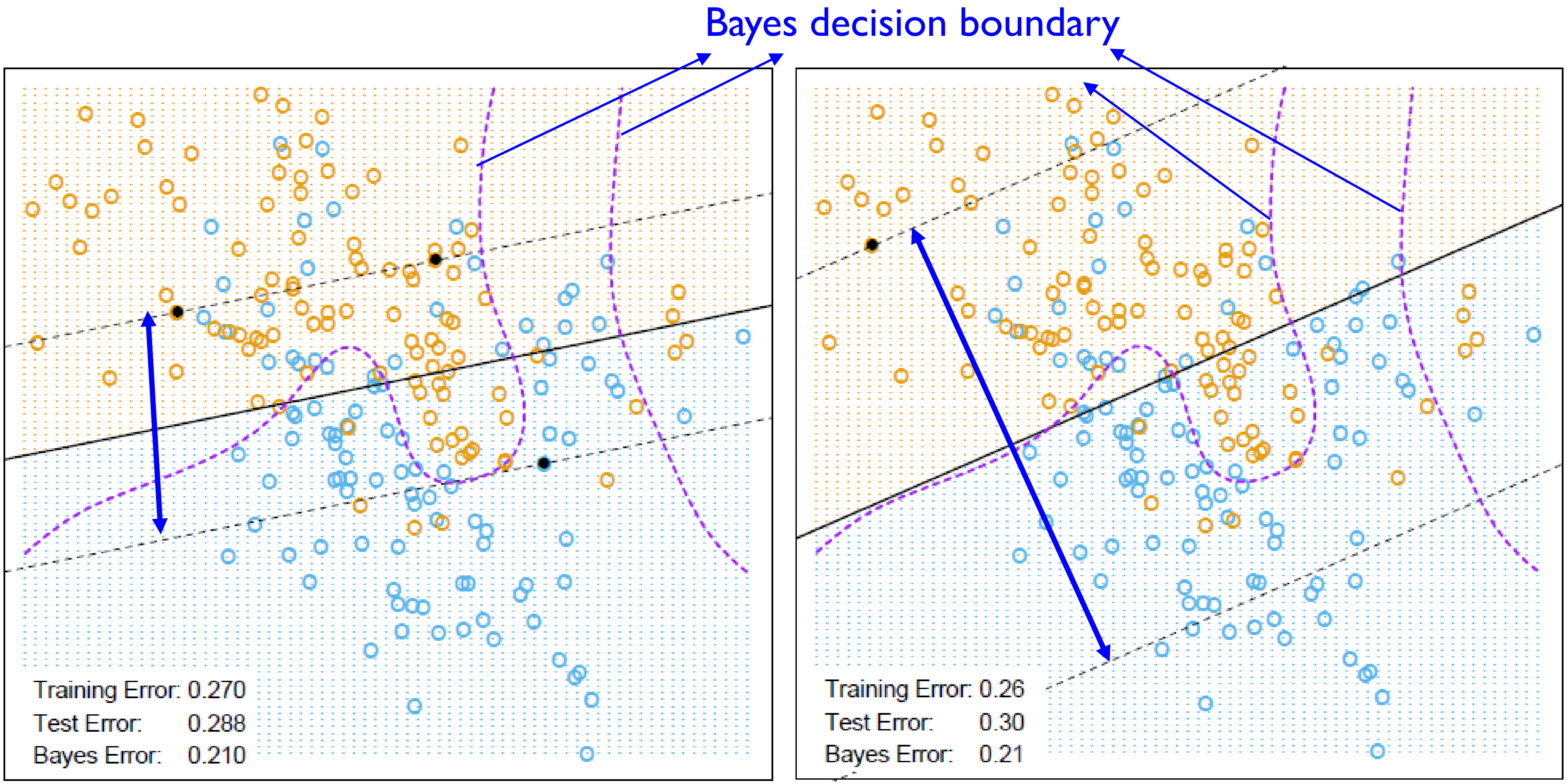
  C ↑ : training error를 많이 허용하지 않음 (training data에 최대한 fitting) → overfit

  C ↓ : training error 많이 허용 → underfit
- Training data가 linearly separable하지 않아도 해가 존재함

Soft Margin SVM Classifiers

C ↑ : training error를 많이 허용하지 않음 → overfit

C ↓ : training error 많이 허용 → underfit



$C = 10000$ $C = 0.01$

Lagrangian Formulation

Original Problem

$$minimize\ \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i$$

$$subject\ to\ y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1,2, \dots \dots, n$$

Lagrangian multiplier를 이용하여 Lagrangian primal 문제로 변환

Lagrangian Primal

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

$$subject\ to \qquad \alpha_{\mathrm{i}} \cdot \gamma_{\mathrm{i}} \geq 0, i = 1,2, \dots \dots, n$$

Lagrangian Formulation

Lagrangian Primal

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

subject to $\alpha_{\mathrm{i}} \cdot \gamma_{\mathrm{i}} \geq 0, i = 1,2, \ldots \ldots, n$

Convex, continuous이기 때문에 미분 = 0 에서 최소값을 가짐

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

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

③ $$\frac{\partial \mathcal{L}(w,b,\xi,\alpha,\gamma)}{\partial \xi_{\mathrm{i}}} = 0 \Longrightarrow C - \alpha_i - \gamma_i = 0, i = 1,2, \ldots \ldots, n$$

Lagrangian Dual

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

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

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}(\alpha_i + \gamma_i)\xi_i$$

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

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

RECALL

$$\frac{\partial\mathcal{L}(w,b,\xi,\alpha,\gamma)}{\partial\xi_i} = 0 \Longrightarrow C - \alpha_i - \gamma_i = 0,\ i = 1,2,\dots\dots,n$$

Lagrangian Dual

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

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

Now, let's maximize it !

$$where \sum_{i=1}^{n}\alpha_i y_i = 0 \; and \; C - \alpha_i - \gamma_i = 0\,, i = 1,2, \dots \dots, n$$

Note : $\alpha_i \geq 0, \gamma_i \geq 0, and\; C - \alpha_i - \gamma_i = 0$

$$\Longrightarrow \quad 0 \leq \alpha_i \leq C$$

Lagrangian Dual

따라서 Lagrangian dual은 다음과 같음

RECALL

Soft Margin SVM

$$\underset{\alpha}{maximize} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

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

$$0 \leq \alpha_i \leq C, i = 1,2, \ldots \ldots, n$$

Hard Margin SVM

$$\underset{\alpha}{maximize} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

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

$$0 \leq \alpha_i,\ i = 1,2, \ldots \ldots, n$$

Linearly separable case와 마찬가지로 decision variable은 α이며,
quadratic programming을 풀어 α에 대한 solution을 얻을 수 있음

Lagrangian Dual

(w,b,ξ,α, ɣ)가 Lagrangian dual problem의 해가 되기 위한 조건 (KKT condition)

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

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

③ $\frac{\partial \mathcal{L}(w,b,\xi,\alpha,\gamma)}{\partial \xi_{\mathrm{i}}} = 0 \Longrightarrow C - \alpha_i - \gamma_i = 0, i = 1,2, \dots \dots, n$

Complementary slackness

④ $\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) = 0, \quad \gamma_i \xi_i = 0, i = 1,2, \dots \dots, n$

Characteristics of the Solution

KKT condition으로부터 다음과 같은 정보를 얻을 수 있음:

$$\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) = 0,$$
$$\alpha_i = C - \gamma_i, \qquad \gamma_i \xi_i = 0, i = 1,2, \dots \dots, n$$

① $\alpha_i = 0 \Rightarrow \gamma_i = C$

$\Rightarrow \xi_i = 0$

$\Rightarrow (y_i(w^T + b) - 1) \neq 0$

$\Longrightarrow x_i$ 가 plus-plane 또는 minus-plane 위에 있지 않음

② $0 < \alpha_i < C \Rightarrow \gamma_i > 0$

$\Rightarrow \xi_i = 0, \gamma_i \xi_i = 0$

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

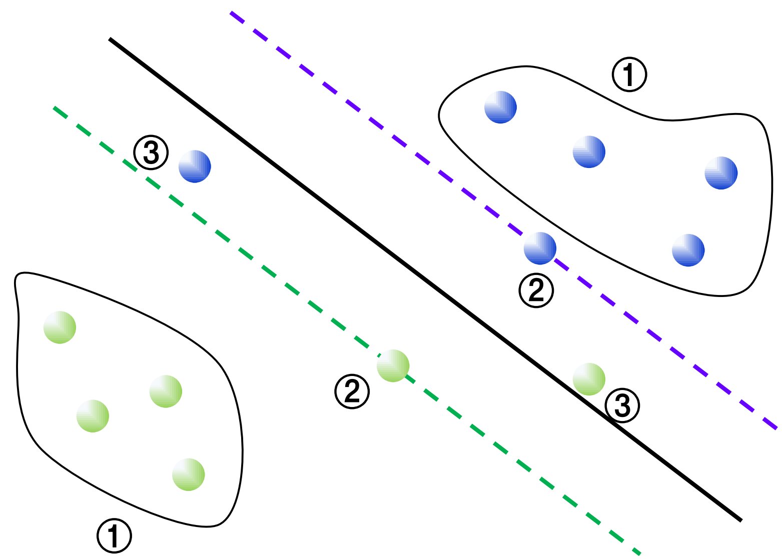
$\Longrightarrow x_i$ 가 plus-plane 또는 minus-plane 위에 있음
(**support vector**)

③ $\alpha_i = C \Rightarrow \gamma_i = 0$

$\Rightarrow \xi_i > 0$

$\Rightarrow \alpha_i(y_i(w^T + b) - 1) = -\alpha_i \xi_i \neq 0$

$\Longrightarrow x_i$ 가 plus-plane과 minus-plane 사이에 있음
(**support vector**)



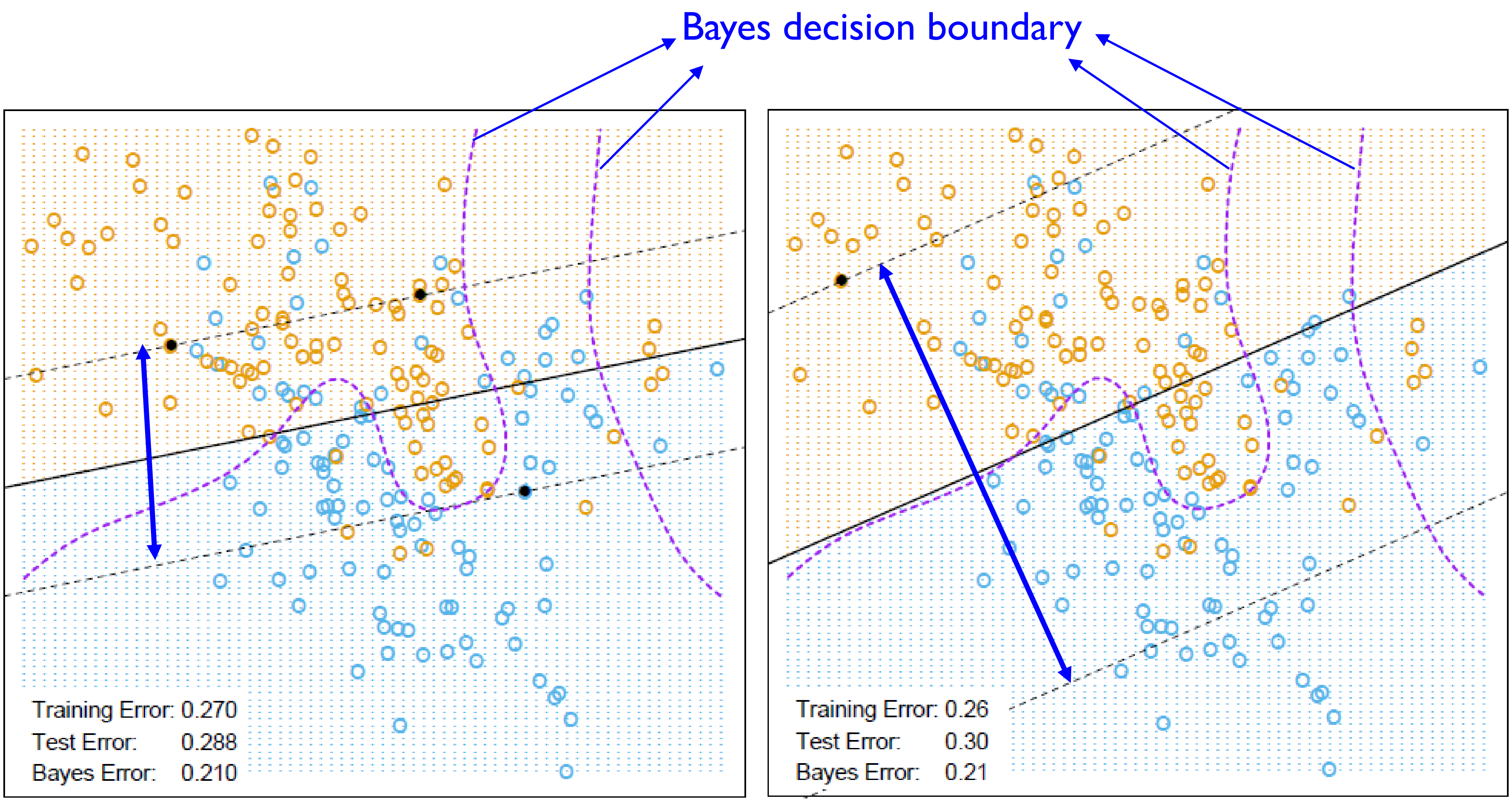
Solutions

x_i가 support vector인 경우에만 $\alpha_i^* > 0$ 이므로

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i = \sum_{i \in SV} \alpha_i^* y_i x_i$$

$$\hat{y}_{new} = sign(\textstyle\sum_{i \in SV} \alpha_i^* y_i x_i^T x_{new} + b^*)$$

Soft Margin SVM Classifiers

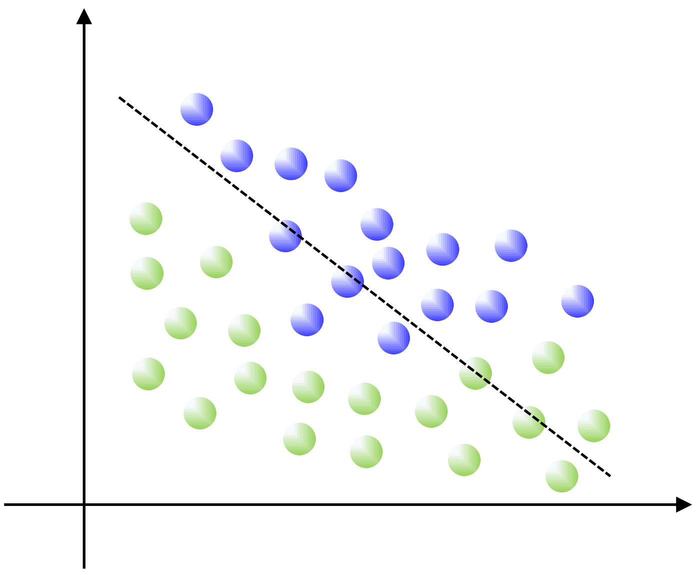


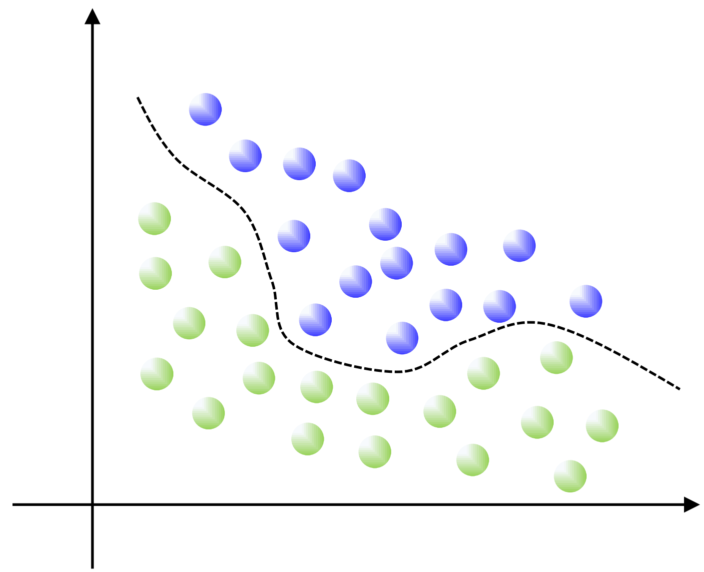
$C = 10000$ $C = 0.01$

Kernel Methods for Nonlinear Classification

Nonlinear Decision Boundary

Linear decision boundary → Nonlinear decision boundary



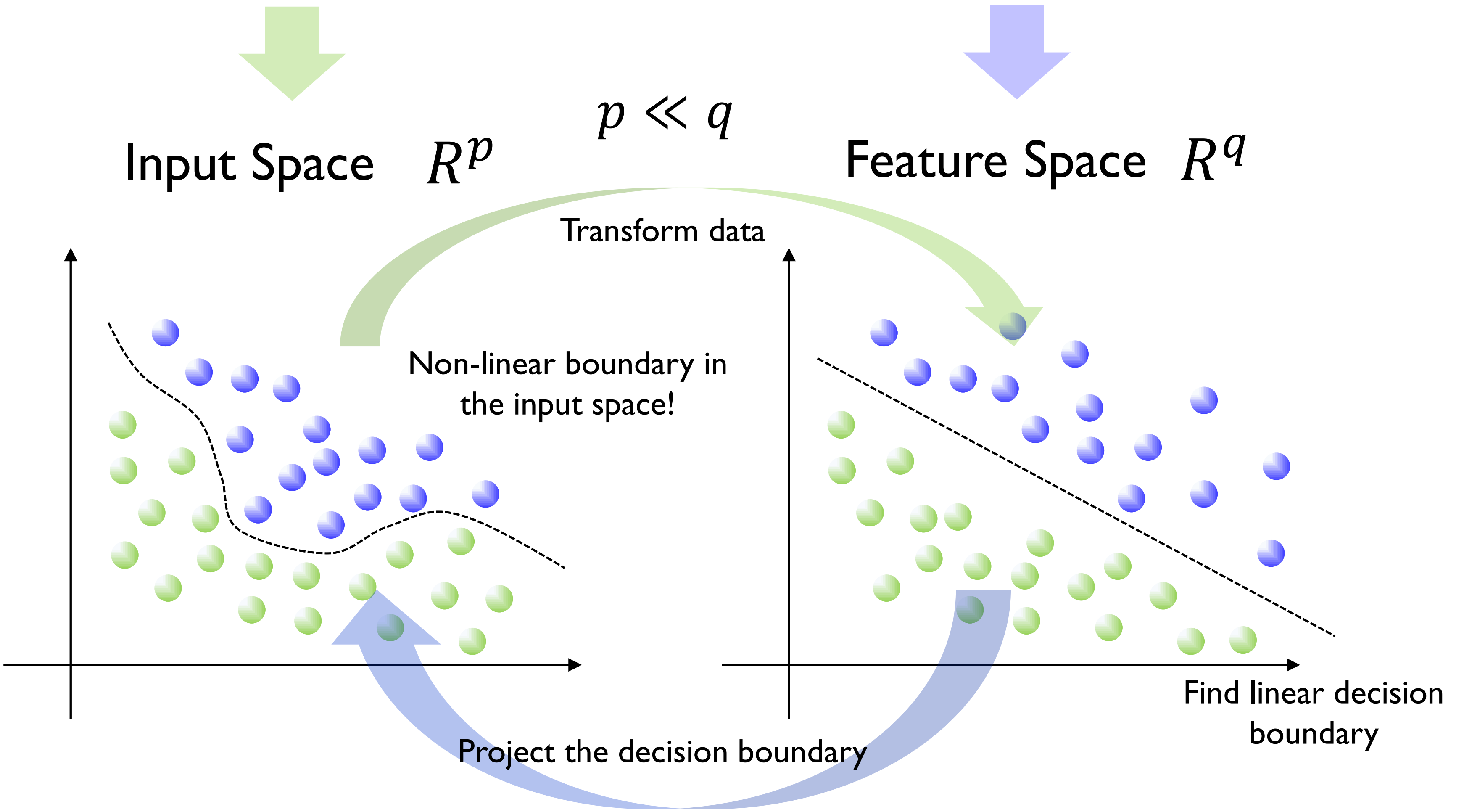


How?

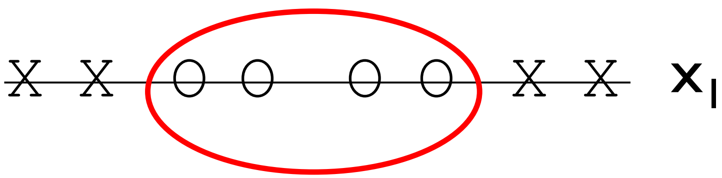
관측치 x들을 더 높은 차원으로 변환시켜 분류해보자!

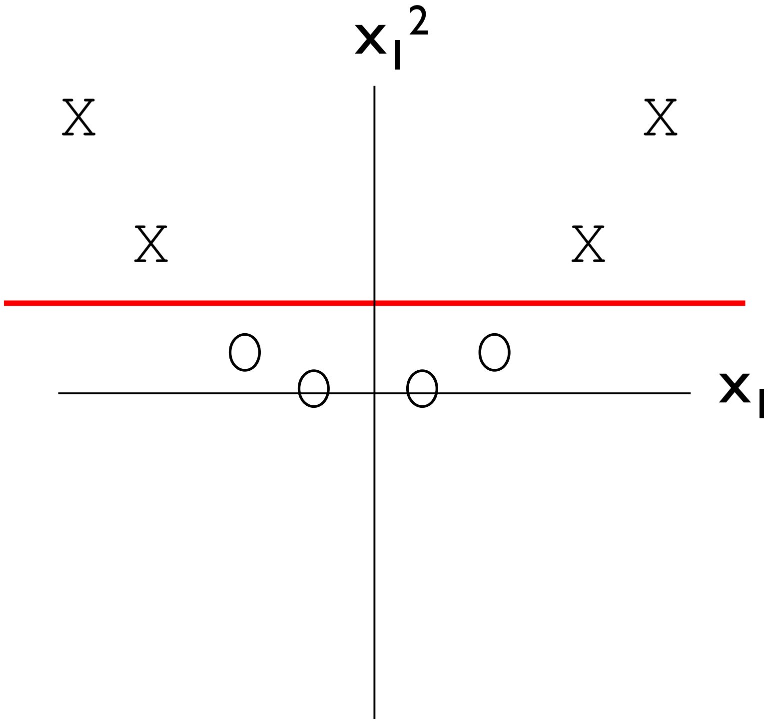
Transforming Data

$$x = (x_1, x_2, \ldots \ldots, x_n) \rightarrow \phi(x) = z = (z_1, z_2, \ldots \ldots z_n)$$

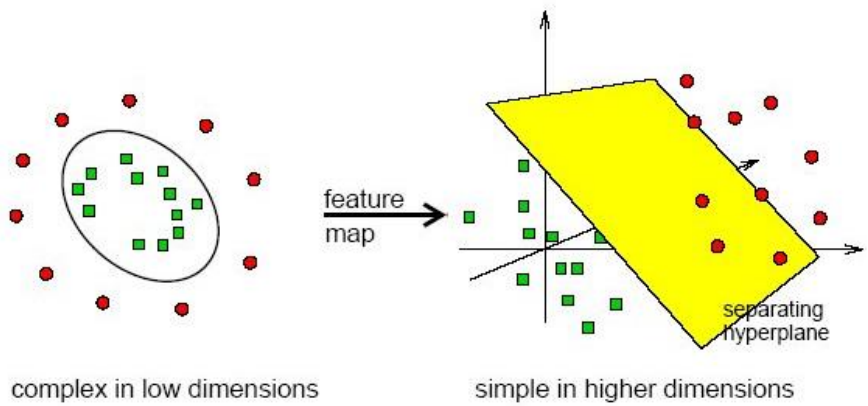


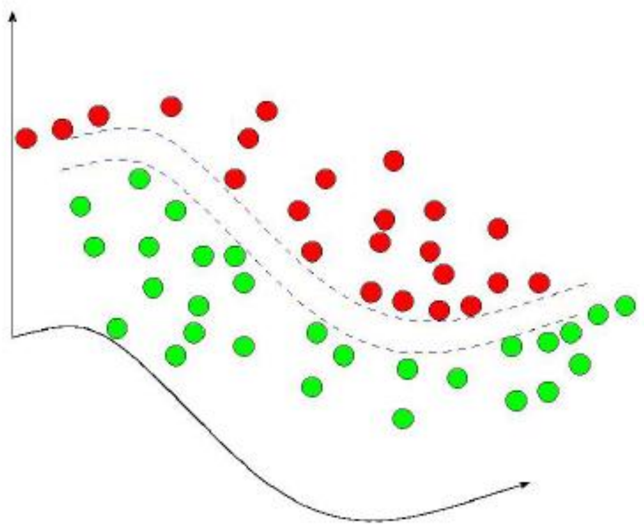
Transforming Data - Example





Transforming Data - Example





- Changing the representation of the data.
- Use another coordinates system such that the “curve” becomes a “line.

Mapping Original Space to Kernel Space

$$\phi : x \longmapsto z = \phi(x)$$

Example

$$\phi : (x_1, x_2) \longmapsto (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$$

2D (Original Space)

5D (Feature Space)

- SVM을 original space가 아닌 feature space에서 학습
- Original space에서 nonlinear decision boundary → Feature space에서 linear decision boundary
- 고차원 feature space에서는 관측치 분류가 더 쉬울 수 있음
- 고차원 feature space를 효율적으로 계산할 수 있는 방법이 있음

Kernel Mapping

SVM Lagrangian dual formulation

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

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

$$0 \leq \ \alpha_i \leq C, i = 1,2, \ldots \ldots, n$$

$$x_i \rightarrow \phi(x_i)$$

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$$

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

$$0 \leq \ \alpha_i \leq C, i = 1,2, \ldots \ldots, n$$

Kernel Mapping

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$$

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

$$0 \le \ \alpha_i \le C, i = 1,2, \dots \dots, n$$

Inner product of $\phi(x_i)^T\phi(x_j)$

$< \phi(x_i), \phi(x_j) >$

$x_i \to \phi(x_i)$

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

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

$$0 \le \ \alpha_i \le C, i = 1,2, \dots \dots, n$$

ϕ를 이용해서 직접 데이터를 변환할 필요 없이 **inner product**에 해당하는 $< \phi(x_i), \phi(x_j) >$만 정의해도 같은 효과를 얻을 수 있음

Kernel Mapping

$$\hat{y}_{new} = sign(\sum_{i \in SV} \alpha_i^* y_i x_i^T x_{new} + b^*)$$

$$x_i \rightarrow \phi(x_i)$$

$$\hat{y}_{new} = sign(\sum_{i \in SV} \alpha_i^* y_i \phi(x_i)^T \phi(x_{new}) + b^*) \; \frac{>0}{<0}$$

$$= sign(\sum_{i \in SV} \alpha_i^* y_i K\langle x_i^T x_{new} \rangle + b^*)$$

ϕ를 이용해서 직접 데이터를 변환할 필요 없이 **inner product**에 해당하는 $< \phi(x_i), \phi(x_j) >$만 정의해도 같은 효과를 얻을 수 있음

Kernel Mapping – Example

$X = (x_1, x_2)$
$Y = (y_1, y_2)$

$\emptyset(X) = \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$
$\emptyset(Y) = \left(y_1^2, y_2^2, \sqrt{2}y_1y_2\right)$

$\langle\emptyset(X),\emptyset(Y)\rangle = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$

Question: Can we compute $x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$ without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$?

$(X, Y)^2 = \langle(x_1, x_2), (y_1, y_2)\rangle^2$
$= \langle x_1y_1 + x_2y_2\rangle^2$
$= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$
$= \langle\emptyset(X),\emptyset(Y)\rangle$

$\rightarrow$ This can be obtained without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$

without knowing the explicit = implicit

$(X, Y)^2 = \langle(x_1, x_2), (y_1, y_2)\rangle^2 = \mathrm{K}(X, Y)$ (Kernel function)

Kernel Functions

- Linear kernel

$$K\langle x_1, x_2\rangle = \langle x_1, x_2\rangle$$

- Polynomial kernel

$$K\langle x_1, x_2\rangle = (a\langle x_1, x_2\rangle + b)^d$$

- Sigmoid kernel (Hyperbolic tangent kernel)

$$K\langle x_1, x_2\rangle = tanh(a\langle x_1, x_2\rangle + b)$$

- Gaussian kernel (Radial basis function (RBF) kernel)

$$K\langle x_1, x_2\rangle = \exp(\frac{-\|x_1 - x_2\|_2^2}{2\sigma^2})$$

Choosing Kernel Functions

- SVM 사용시 kernel을 결정하는 것은 어려운 문제 → 딱히 기준이 없음
- 사용하는 kernel에 따라 feature space의 특징이 달라지기 때문에 데이터의 특성에 맞는 kernel을 결정하는 것은 중요함
- 일반적으로는 RBF kernel, sigmoid kernel, low degree polynomial kernel 또는 등이 주로 사용됨

EOD