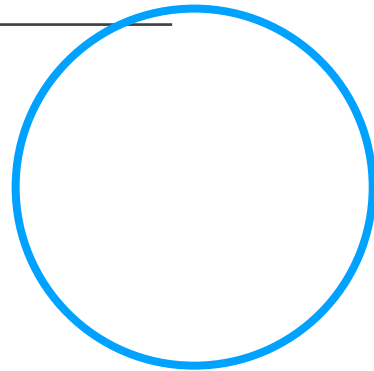


ESC 21FALL Week 2

-Doby is free! 🐝

SEP 16, 2021

SOOYON KIM 🧑💻 + JAEHYUN LEE 🧑💻



Goal for today!

[ESL CHAPTER 12] SUPPORT VECTOR
MACHINES AND FLEXIBLE DISCRIMINANTS

HYPERPLANE ?

MARGIN?

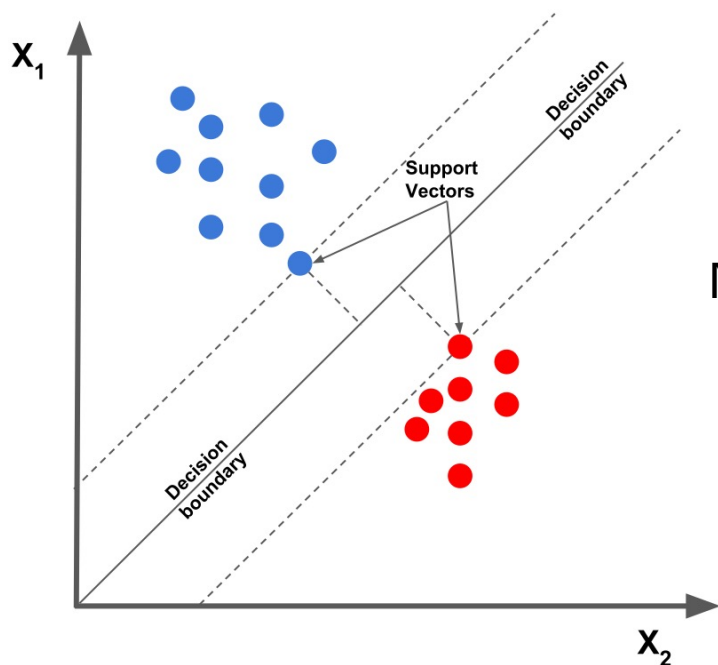
SUPPORT VECTOR MACHINE?

BEFORE WE START...

Flexible Discriminants ?

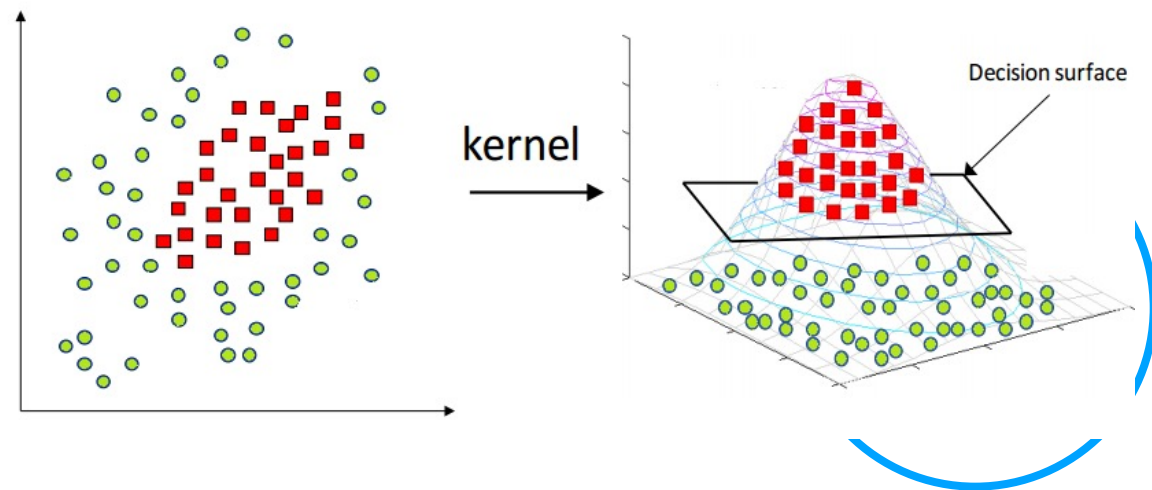
⚙️ Classification mechanism: finding decision boundaries!

Simplest case : linear boundary



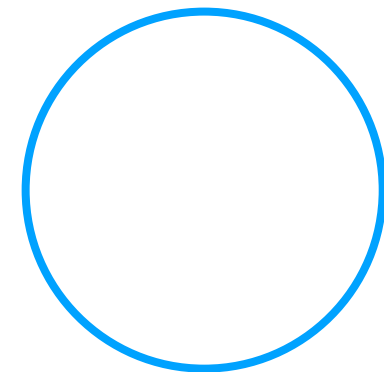
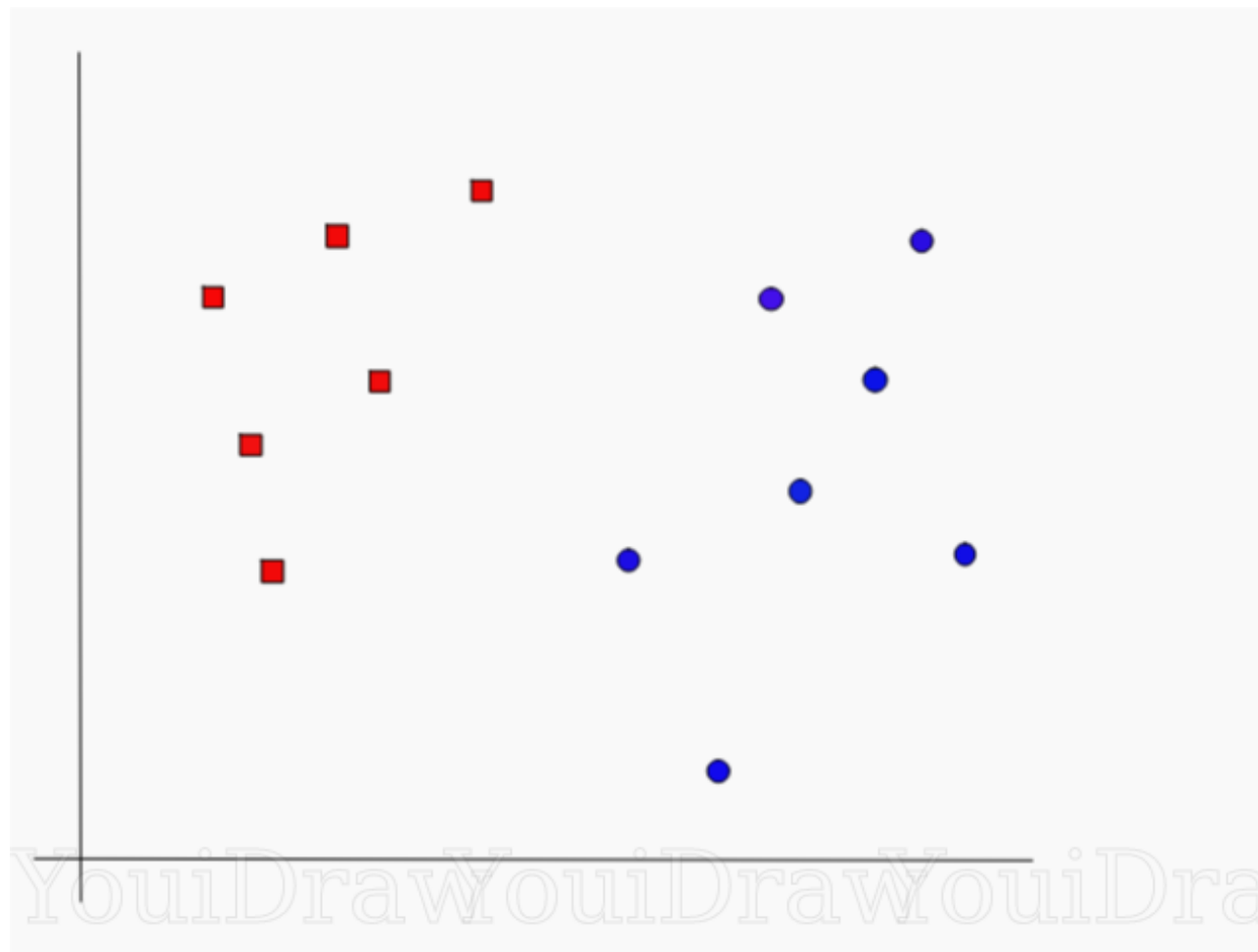
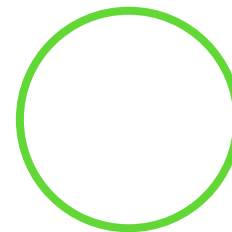
But what if...
Non-separable case?
classes overlap ?

non-linear boundary!
Extend feature space to non-linear
world





NOT A BIG DEAL ?



Optimal Margin Classifier ➡ Soft Margin Classifier ➡ Support Vector Machine
(Maximal Margin Classifier) (Support Vector Classifier)

OPTIMAL MARGIN CLASSIFIER

Optimal Margin Classifier ?

📈 separates two classes and maximizes the distance to the closest point from either class (a.k.a. Maximal Margin Classifier)

📈 한마디로, 제일 안전빵인 **hyperplane**을 찾겠다는 의도

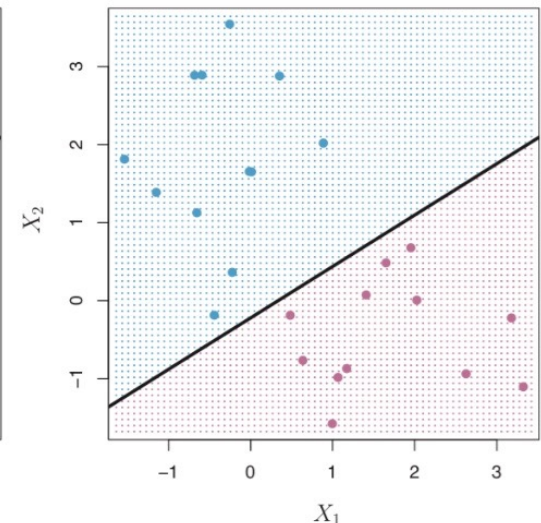
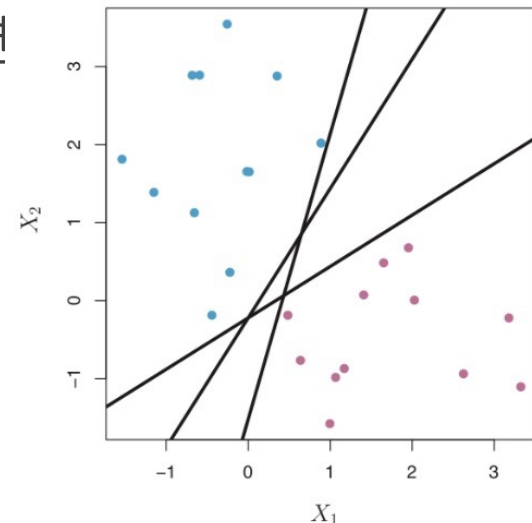
“separate data with a large gap (margin)”

Hyperplane ?

📖 p차원 공간에서 전체 공간을 두 부분으로 나누는 평면

$$L: f(x) = \beta_0 + \beta^T x = 0$$

$$f(X) = \beta_0 + X\beta = 0$$



OPTIMAL MARGIN CLASSIFIER

Hyperplane의 특징

1. β 의 의미 ?

🔍 hyperplane의 방향 (β is orthogonal to the hyperplane)

2. β_0 의 의미?

🔍 원점과 hyperplane 간의 거리를 결정 ($\beta^T x_0 = -\beta_0$)

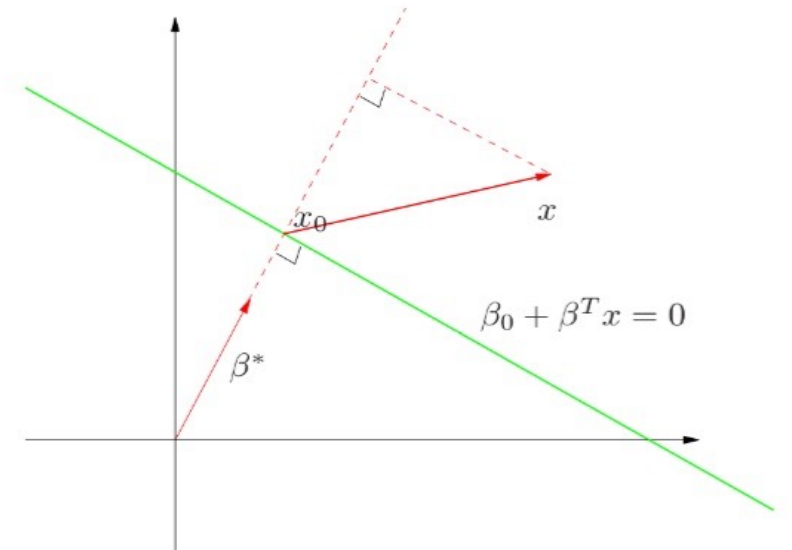
3. 임의의 벡터 x 와 Hyperplane 간의 거리

$$\beta^{*T} (x - x_0) = \frac{1}{\|\beta\|} (\beta^T x + \beta_0) = \frac{1}{\|f'(x)\|} f(x)$$

➡ 만약 두 observation x, x_0 가 모두 L위에 있었다면 $\beta^T (x - x_0) = 0$ 이 될 것!

➡ $f(x)$ is proportional to the signed distance from x to the hyperplane defined by $f(x)=0$

➡ 만약에 linearly separable한 hyperplane이 존재한다면, gradient descent 방법





OPTIMAL MARGIN CLASSIFIER



Margin ?

- 임의의 관측치 X_i 와 Hyperplane 사이의 거리 : **M**

$y_i \in \{-1, 1\}$ 의 예측에서 X 의 hyperplane을 이용한 classifier를 생각해보자!

g : classification rule!

$$y(X) = G(\beta^T X + \beta_0) \textbf{ where } G(z) = \begin{cases} 1, & z \geq 0 \\ -1, & z < 0 \end{cases}$$

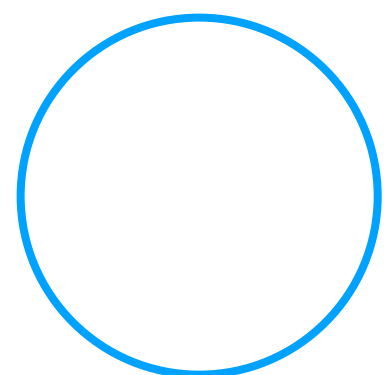
📌 $G(\beta^T X + b) = \text{signf}(x)$ 이 평면에서 의미하는게 뭘까 ?

➡ 직선보다 위에 있는지 아래에 있는지!

$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|=1} M \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq M, \quad i = 1, \dots, N. \end{aligned}$$

➡ 모든 점들이 최소한 M만큼은 hyperplane에서 떨어져있다!

💡 the distance of an obs. from the hyperplane = our confidence that the obs. was correctly classified!!



OPTIMAL MARGIN CLASSIFIER

Objective function

$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|=1} M \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq M, \quad i = 1, \dots, N. \end{aligned}$$

좀 더 최적화하기 쉽게 형태를 바꿔보면 다음과 같다.

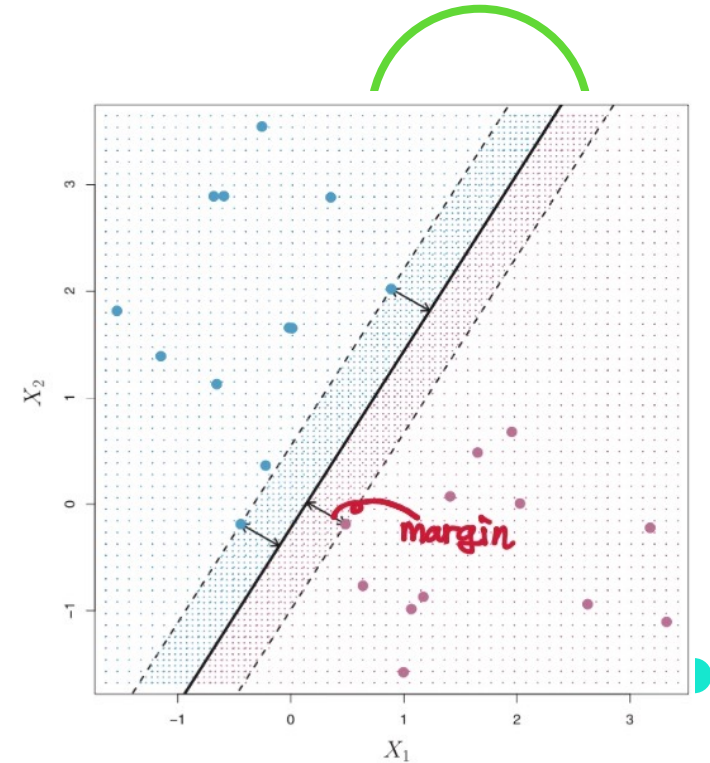
$$\begin{aligned} & \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \quad (\text{set } \|\beta\| = \frac{1}{M}) \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq M, \quad i = 1, \dots, N. \end{aligned}$$

➡ 위의 모든 점들이 최소한 M만큼은 떨어져있도록 하겠다는 조건은 두께가 $1/\|\beta\|$ 인 slab을 linear decision boundary 주위에 형성한다.

margin 위에 있는 data point = support vector

Why "support" vector?

- they "support" the maximal margin hyperplane in the sense that if these points were moved slightly then the maximal margin hyperplane would move as well (directly influence maximal margin hyperplane, whereas others do not...)



OPTIMAL MARGIN CLASSIFIER

Solving Optimization problem ?

$$L_P = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i (x_i^T \beta + \beta_0) - 1].$$

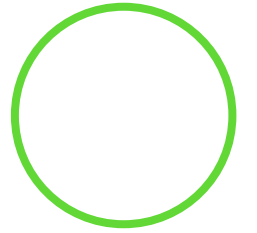
- Lagrangian method + Duality
- solutions satisfy KKT condition

~~수치해석을 들어도 이해하기 어려워요,,~~

그래서 결론은 ? How does optimal beta appear ?

- optimal한 해는 결론적으로 모든 data point들을 다 고려하지 않는다.
- solution vector β 는 *support point*라고 하는 특정한 x_i 들의 선형결합꼴로 나타난다.
- 훈련, 즉 optimal solution을 찾는 과정이 끝나고 나면 대부분의 훈련 샘플은 필요없어지고 최종 모델은 오직 support vector와 관련이 있게 된다!
- 만약 data가 separable하지 않다면? 이 문제에 대해 feasible solution은 없다.. 대안을 찾아야함

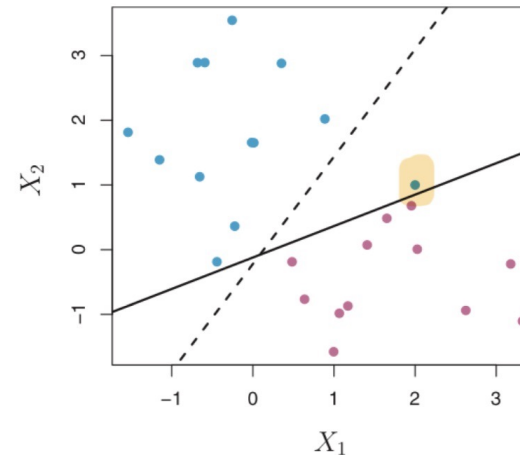
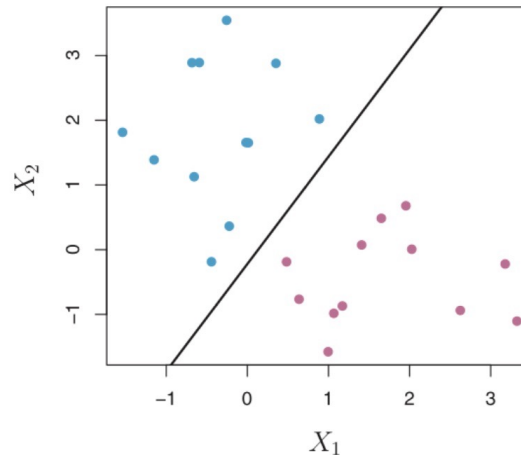
SUPPORT VECTOR CLASSIFIER



- ↑ 앞선 maximal margin classifier의 가장 큰 맹점은 training set이 sample space (or feature space)에서 선형분리가 가능 (linearly separable) 하다고 “가정”했다는 것이다.

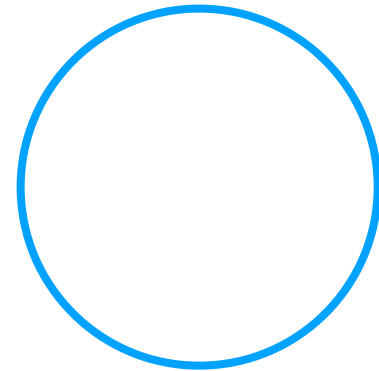
But in reality, ... 🧑

- ↑ 또다른 단점은 maximal margin classifier는 training set에 과적합되는 경향이 있다는 것이다... 🧑🧑



Solution ?

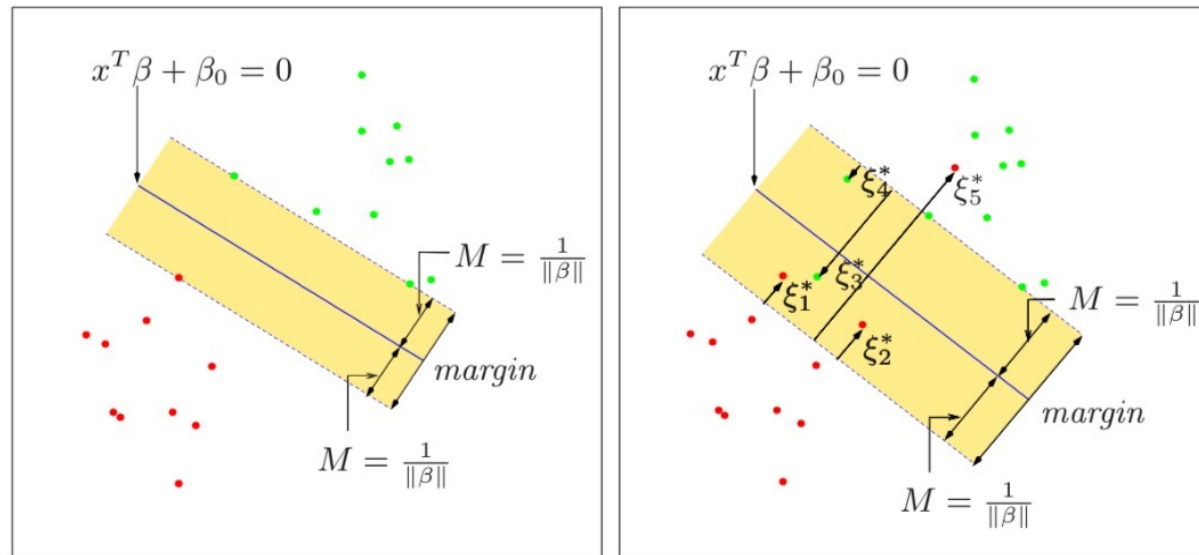
🖋️ 우리의 모델에게 약간의 오류를 허용해주자! (soft margin)



SUPPORT VECTOR CLASSIFIER

Then how should we change our objective ?

- define slack variables : 원래의 올바른 margin으로부터 얼마나 잘못된 위치에 있는지



Optimization problem

$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|=1} M \\ & \xi = (\xi_1, \xi_2, \dots, \xi_N) \\ & y_i(x_i^T \beta + \beta_0) \geq M - \xi_i, \text{ or } y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i) \end{aligned}$$

SUPPORT VECTOR CLASSIFIER

Optimization problem

$$\max_{\beta, \beta_0, \|\beta\|=1} M$$

$$\xi = (\xi_1, \xi_2, \dots, \xi_N)$$

$$y_i(x_i^T \beta + \beta_0) \geq M - \xi_i, \text{ or } y_i(x_i^T \beta + \beta_0) \geq M(1 - \xi_i)$$

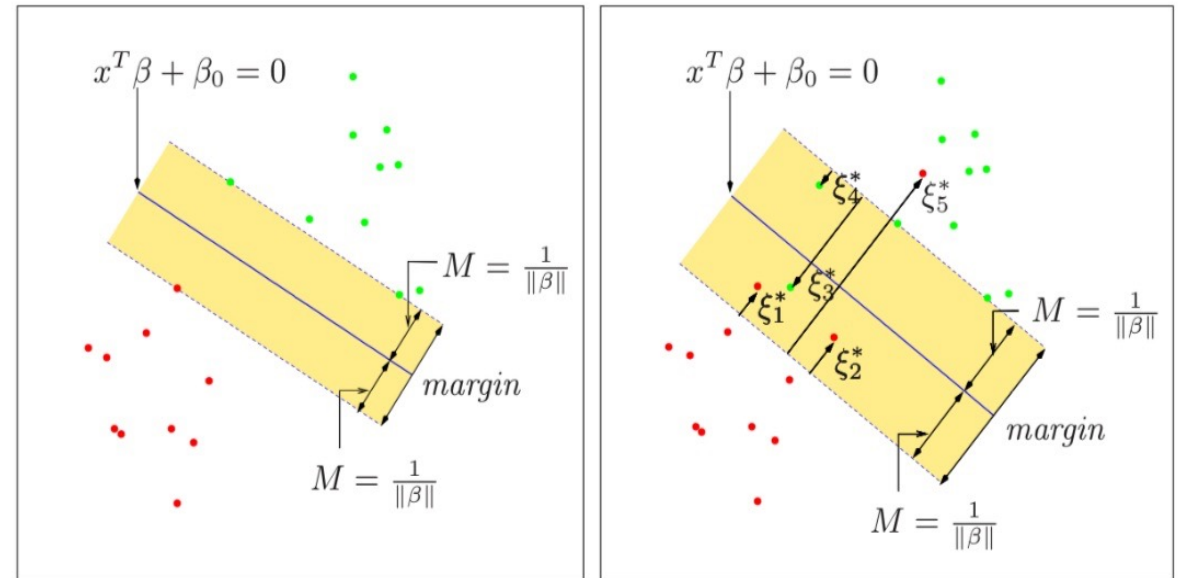
Left : overlap in actual distance from the margin

Right : overlap in relative distance, which changes with the width of margin M

⚙ proportional amount by which the prediction $f(x)$

is on the wrong side of its margin.

(그렇지만 오른쪽의 constraint를 사용해야 convex problem이 되기 때문에 오른쪽 제약식을 사용한다.)

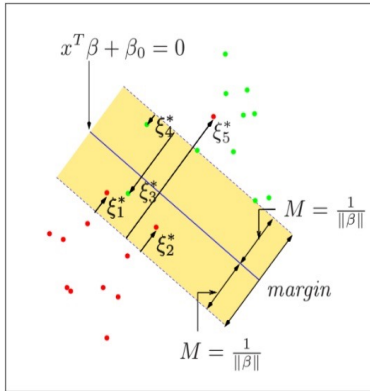


SUPPORT VECTOR CLASSIFIER

Bounding slack variables

$\xi_i > 1$: misclassification

so bounding $\sum \xi_i \leq K$ Means we are allowing the total number of training misclassifications at K



아까처럼 equivalent 한 형태로 최소화 문제로 변환하면

$$\min \|\beta\| \text{ subject to } \begin{cases} y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \\ \xi_i \geq 0, \sum \xi_i \leq C \end{cases}$$

위의 objective에 따르면, class boundary 안쪽에 잘 있는 점들은 boundary 형성에 별 영향을 주지 않는다는 것을 알 수 있다. (constraint에 영향을 주지 않으니까..)

이걸 !!또!! 바꾸면 다음과 같다. Lagrangian method로 풀기 위해서...

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i$$

$$\text{subject to } \xi_i \geq 0, y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i$$



SUPPORT VECTOR CLASSIFIER



Slack variable  Bias-Variance Tradeoff

$$\min ||\beta|| \text{ subject to } \begin{cases} y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \\ \xi_i \geq 0, \sum \xi_i \leq C \end{cases}$$

C : nonnegative tuning parameter

➡ bounds the sum of slack variables (i.e. 모든 slack variable=0이라면 maximal margin classifier!)

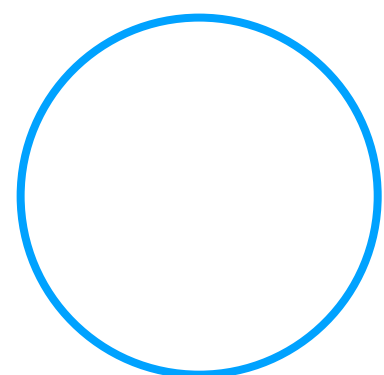

➡ as C increases, we become more tolerant and width becomes wider

!!bias-variance tradeoff!!

C small : tend to overfit data => low bias, high variance

C large : relaxed => high bias, low variance

Then, *support vectors* : Observations that lie directly on the margin,
or on the wrong side of the margin for their class

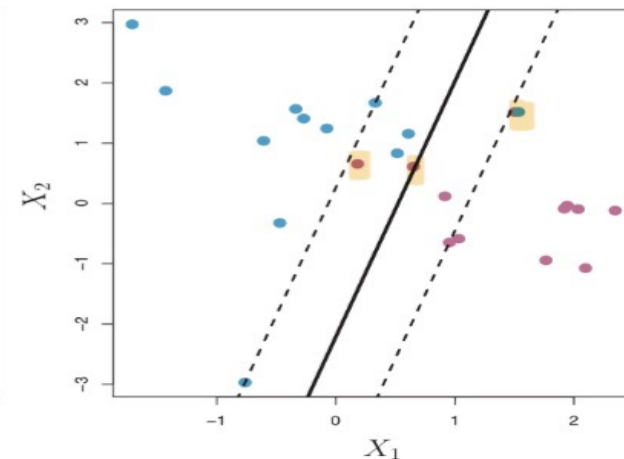
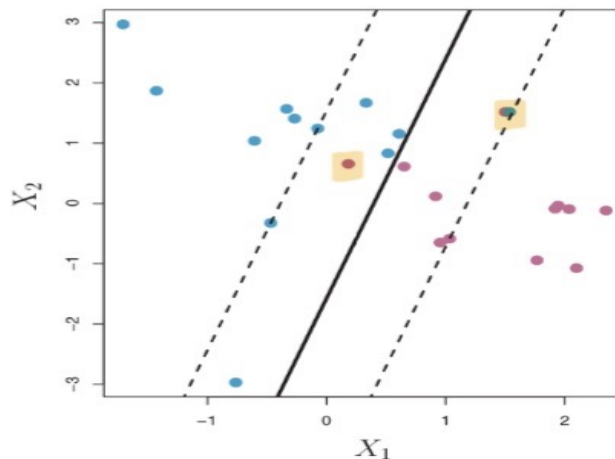
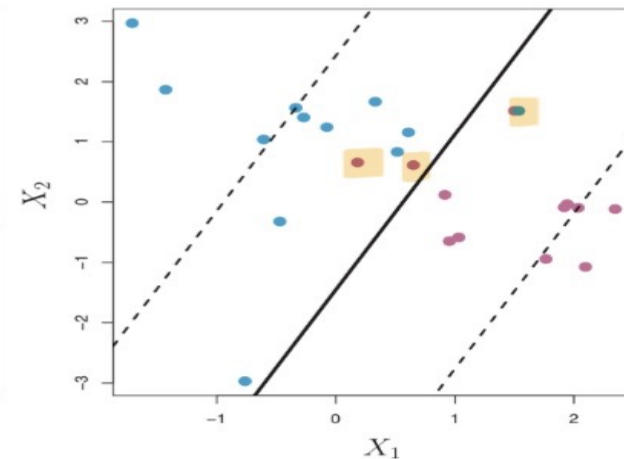
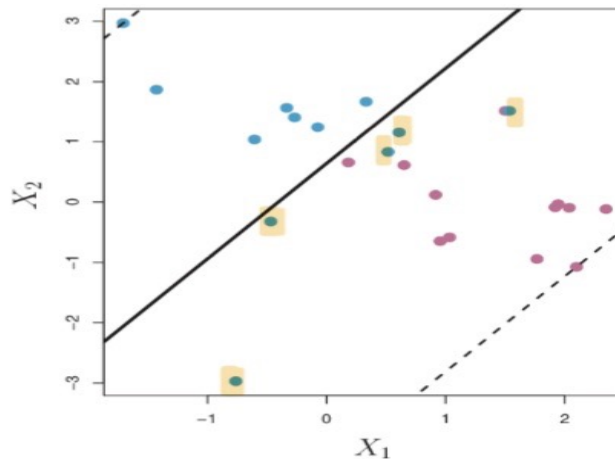


SUPPORT VECTOR CLASSIFIER

Slack variable Bias-Variance Tradeoff

- observation들과 연관지어 생각해보면,
C를 늘려서 margin의 넓이가 늘어난다는 건
그만큼 margin을 violate하는 observation들,
즉 support vector들을 많이 사용한다는 것이다.

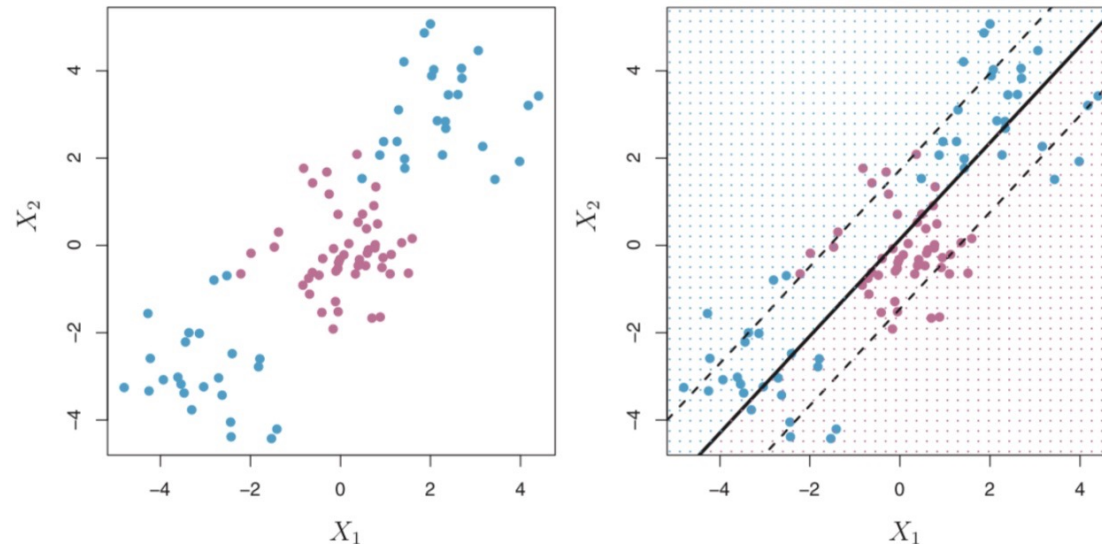
(hyperplane을 정할때 그만큼 많은 obs 들을
고려한단 의미)



SUPPORT VECTOR !! MACHINE !!

💡 How to produce non-linear decision boundaries (automatically) ?

Where are we ? Optimal margin classifier (Maximal margin classifier) => Support Vector Classifier (Soft margin classifier) => Support Vector Machines (Now!)



⬆ Being able to work only on linear decision boundary is poor in many cases...

Solution?

Think of how we dealt beyond linearity before! Fit higher order terms!



SUPPORT VECTOR !! MACHINE !!



Is there any efficient way?

SVM : extension of the support vector classifier that results from enlarging the feature space using **kernels**

Solution to support vector classifier includes only the *inner products* of the obs. (Not the obs. itself...!)

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^p x_{ij} x_{i'j}$$

$$f(x) = \beta_0 + \sum \alpha_i \langle x, x_i \rangle$$

(linear support vector classifier는 위와 같이 표현될 수 있는데,

이것만 보면 모든 obs.를 다 사용하는게 아닌가 싶지만 사실 support vector가 아닌 애들에 대해서는 $\alpha = 0$ 이 된다.)

➡ 결론적으로, linear classifier의 coefficient들을 계산하기 위해 우리가 필요한 것은 내적이다.



SUPPORT VECTOR !! MACHINE !!

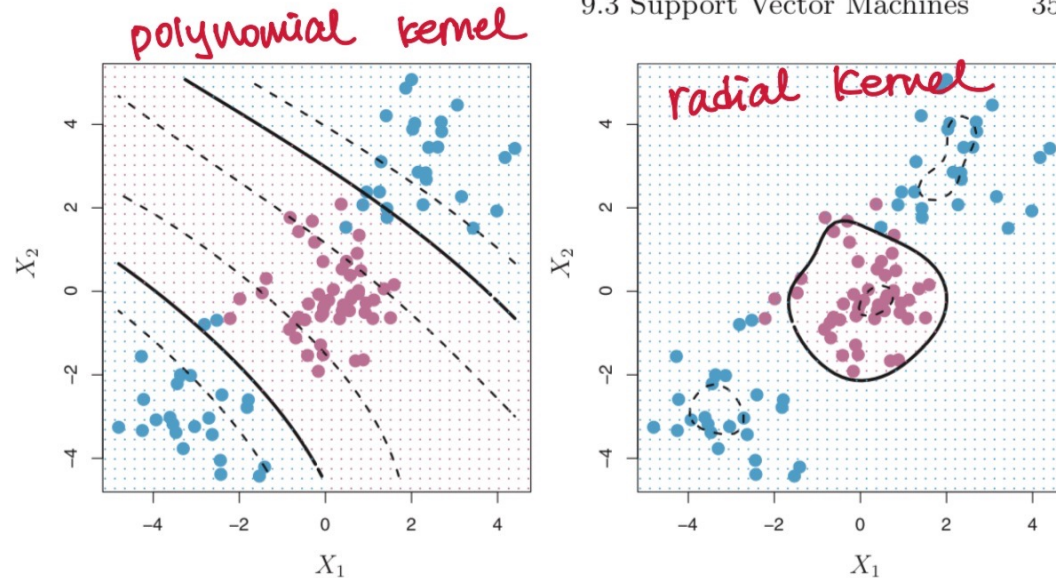
💡 그런데, 저번 주에 kernel이 넓게 말하면 similarity metric이라고 했던 걸 기억해봅시다!!
따지고 보면 저 위의 내적도 kernel로 일반화 가능

(A kernel is a function that quantifies the similarity of two observations!)

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

9.3 Support Vector Machines

353

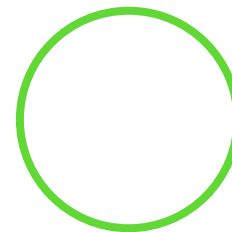


So, what is the advantage of using kernel?

- we can efficiently / easily move to high-dimensional feature space!



SUPPORT VECTOR !! MACHINE !!



misc. SVMs with More than Two Classes

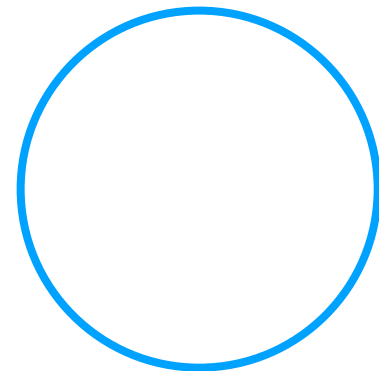
- *one-versus-one* vs. *one-versus-all*
- K개 class에 대해 classification을 하고 싶다면?!

one-versus-one

➡ K(K-1)/2개 SVM, 그 중 가장 frequent한 class로 예측 (kind of voting!)

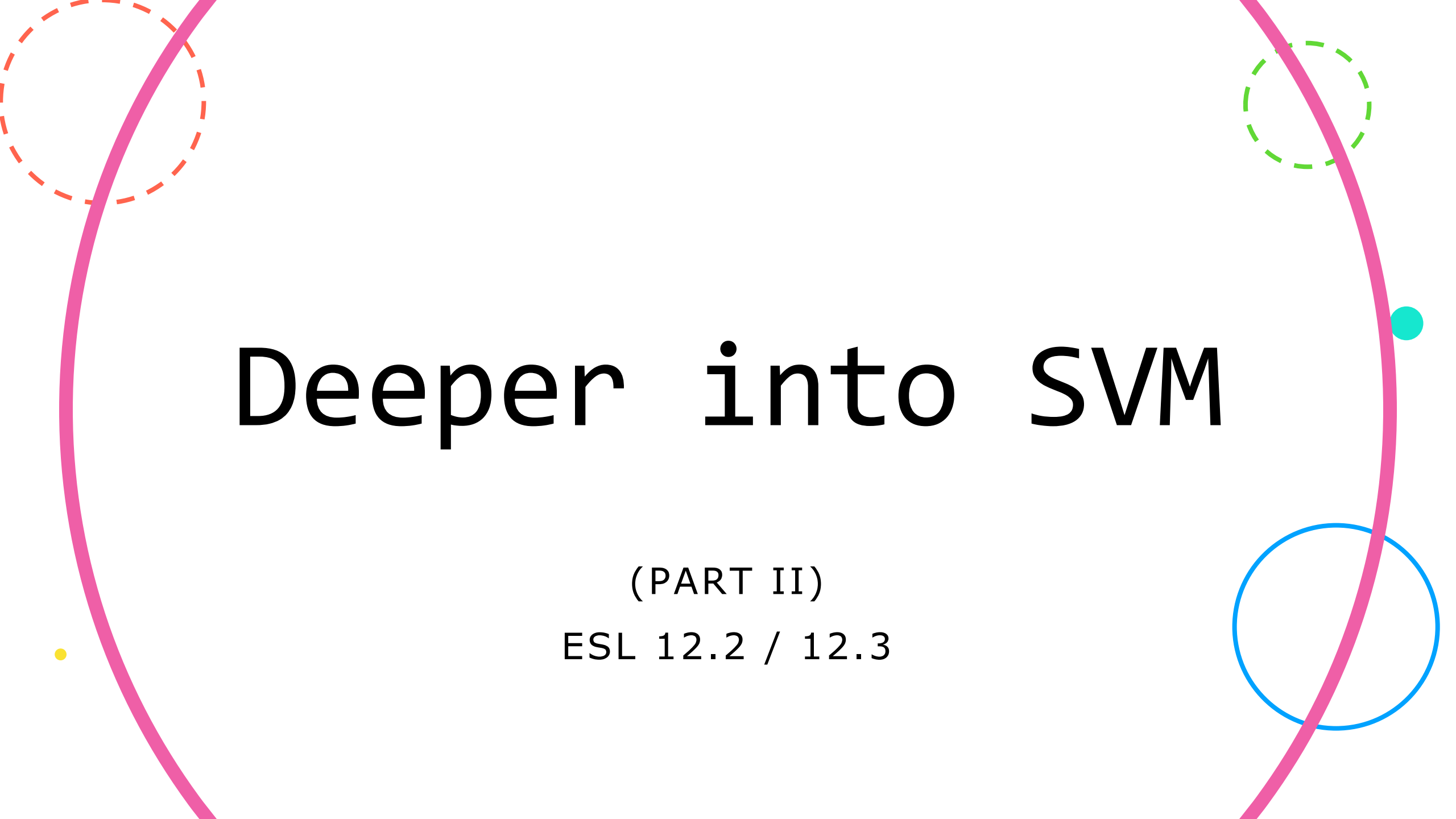
One-versus-all

➡ fit K SVMs, comparing one of the K classes to remainina K-1 classes



Break Time





Deeper into SVM

(PART II)

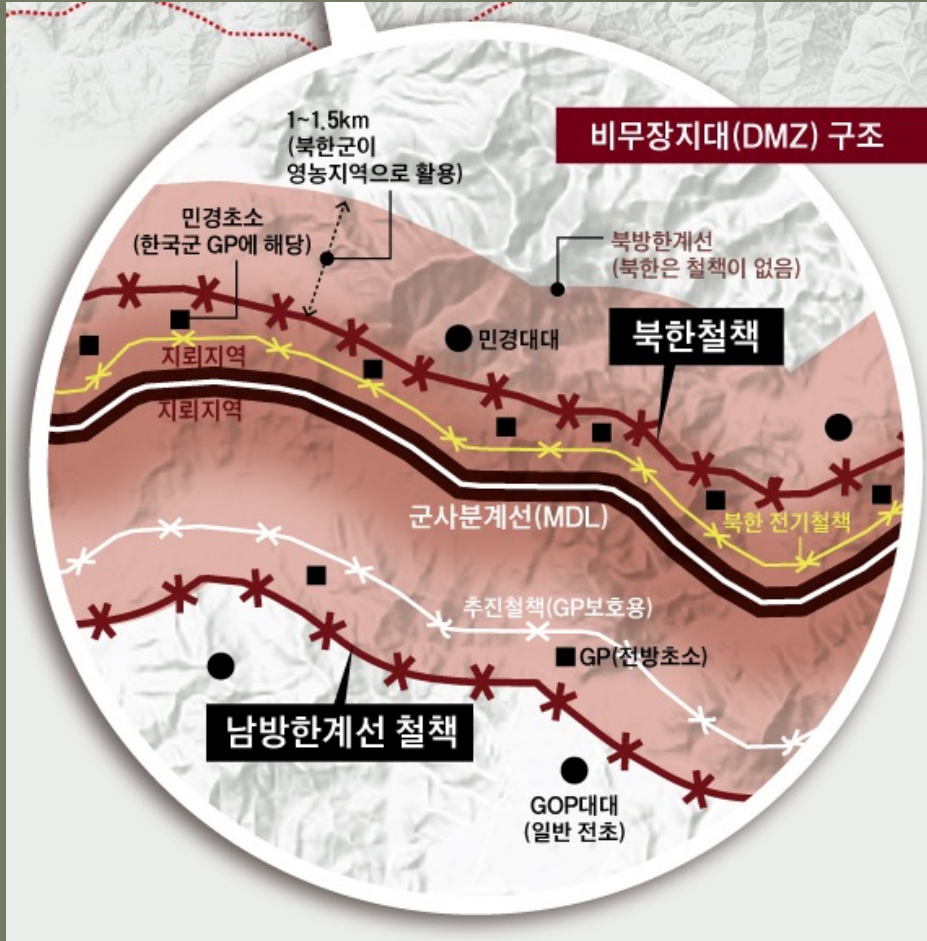
ESL 12.2 / 12.3



U.S. ★ V. ★ M.

PART. 2

SVM의 비유적 이해



SVM	남북한 경계선
Hyperplane	군사분계선 (MDL)
Slab	DMZ
Boundary of the slab	북방한계선, 남방한계선
Margin	2km
Support vector	GOP
Slack variables	GP
$\xi (xi)$	"봐주는 거리"



THE TWO PERSPECTIVES



Perspective M

$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|=1} M \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq M, \quad i = 1, \dots, N. \end{aligned} \quad (4.45)$$

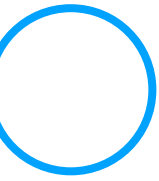
Perspective β

$$\begin{aligned} & \max_{\beta, \beta_0} M \\ & \text{subject to } \frac{1}{\|\beta\|} y_i(x_i^T \beta + \beta_0) \geq M \end{aligned}$$

$$\begin{aligned} & \max_{\beta, \beta_0} M \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1 \end{aligned}$$

$$\begin{aligned} & \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1 \end{aligned}$$

※ 현재까지 모든 수식은 동일한 식이다!





LAGRANGE FUNCTIONS



Lagrange primal function:

$$L_P = \frac{1}{2} \|\beta\|^2 - \sum_{i=1}^N \alpha_i [y_i (x_i^T \beta + \beta_0) - 1]. \quad (4.49)$$

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i, \quad (4.50)$$

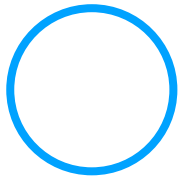
$$0 = \sum_{i=1}^N \alpha_i y_i, \quad (4.51)$$

After substitution, we obtain the Lagrangian Wolfe dual objective function:

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^N \alpha_i \alpha_k y_i y_k x_i^T x_k$$

subject to $\alpha_i \geq 0$ and $\sum_{i=1}^N \alpha_i y_i = 0.$ (4.52)

Additional Karush-Kuhn-Tucker condition:

$$\alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0 \quad \forall i. \quad (4.53)$$




PARAMETER ALPHA

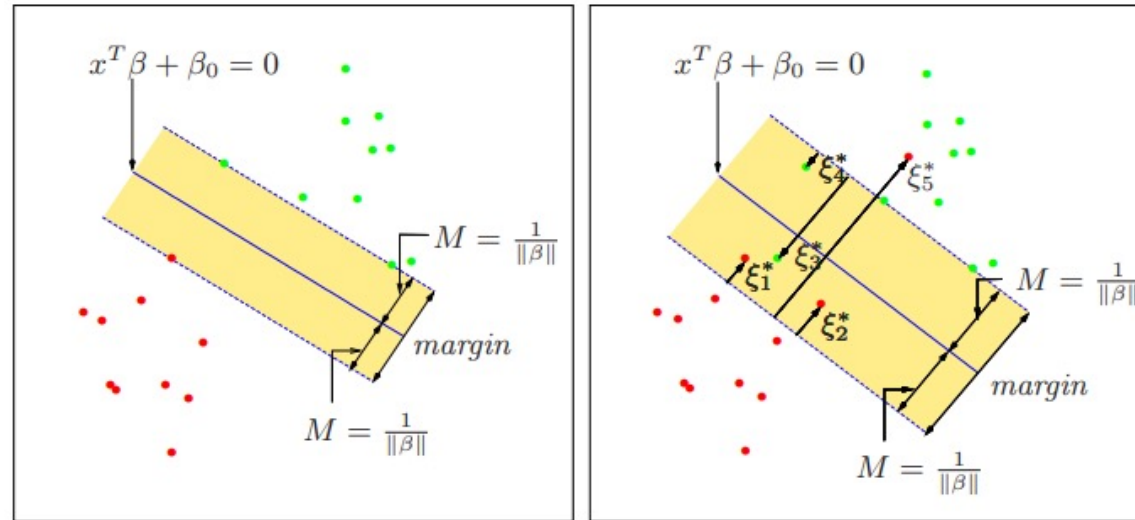
Parameter α

- if $\alpha_i > 0$, then $y_i(x_i^T \beta + \beta_0) = 1$, or in other words, x_i is on the boundary of the slab;
- if $y_i(x_i^T \beta + \beta_0) > 1$, x_i is not on the boundary of the slab, and $\alpha_i = 0$.

... contains information of the whether we concern the variable or not!



SLACK VARIABLES



Perspective M

$$\begin{aligned} & \max_{\beta, \beta_0, \|\beta\|=1} M \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq M - \xi_i, \\ & \xi_i \geq 0, \sum_{i=1}^N \xi_i \leq \text{constant} \end{aligned}$$

Perspective β

$$\begin{aligned} & \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \sum \xi_i \leq \text{constant}' \end{aligned}$$



LAGRANGE FUNCTIONS



Lagrange function for the last constraint only:

$$\begin{aligned} \min_{\beta, \beta_0} \quad & \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i \\ \text{subject to} \quad & \xi_i \geq 0, \quad y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \quad \forall i, \end{aligned} \quad (12.8)$$

Lagrange primal function:

$$L_P = \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i(x_i^T \beta + \beta_0) - (1 - \xi_i)] - \sum_{i=1}^N \mu_i \xi_i, \quad (12.9)$$

$$\beta = \sum_{i=1}^N \alpha_i y_i x_i, \quad (12.10)$$

$$0 = \sum_{i=1}^N \alpha_i y_i, \quad (12.11)$$

$$\alpha_i = C - \mu_i, \quad \forall i, \quad (12.12)$$




...CONTINUED

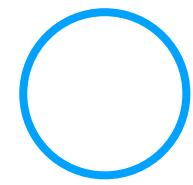
After substitution, we obtain the Lagrangian Wolfe dual objective function:

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{i'=1}^N \alpha_i \alpha_{i'} y_i y_{i'} x_i^T x_{i'}, \quad (12.13)$$

The followings are constraints:

$$\begin{aligned} 0 &\leq \alpha_i \leq C, \\ \sum_{i=1}^N \alpha_i y_i &= 0, \\ \alpha_i [y_i (x_i \beta + \beta_0) - (1 - \xi_i)] &= 0, \\ \mu_i \xi_i &= 0, \\ y_i (x_i^T \beta + \beta_0) - (1 - \xi_i) &\geq 0 \end{aligned}$$

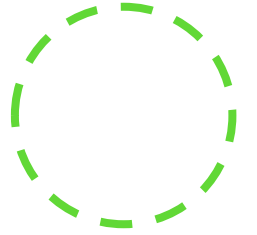
- Q. Parameter α , C 의 특징은?



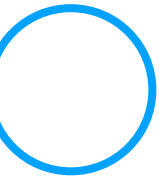


KERNEL INTRO

문제적남자!



사고실험: 일직선 위의 분류분석





TRANSFORMATION AND KERNEL FUNCTION

Transformation: $x \rightarrow h(x)$

$$L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{i'=1}^N \alpha_i \alpha_{i'} y_i y_{i'} \langle h(x_i), h(x_{i'}) \rangle. \quad (12.19)$$

$$\begin{aligned} f(x) &= h(x)^T \beta + \beta_0 \\ &= \sum_{i=1}^N \alpha_i y_i \langle h(x), h(x_i) \rangle + \beta_0. \end{aligned} \quad (12.20)$$

Kernel function: $K(x, x')$

$$K(x, x') = \langle h(x), h(x') \rangle \quad (12.21)$$


KERNEL CANDIDATES

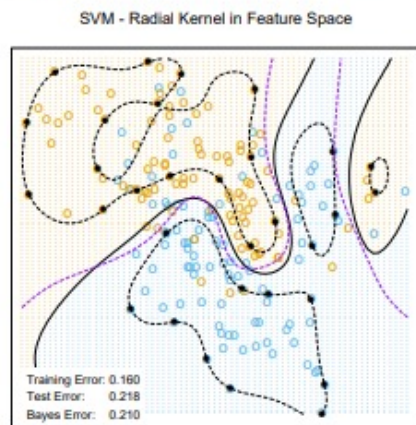
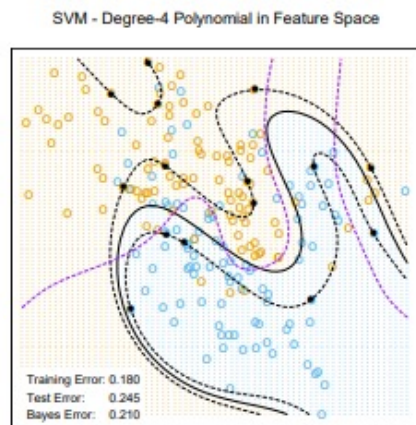


FIGURE 12.3. Two nonlinear SVMs for the mixture data. The upper plot uses a 4th degree polynomial kernel, the lower a radial basis kernel (with $\gamma = 1$). In each case C was tuned to approximately achieve the best test error performance, and $C = 1$ worked well in both cases. The radial basis kernel performs the best (close to Bayes optimal), as might be expected given the data arise from mixtures of Gaussians. The broken purple curve in the background is the Bayes decision boundary.

Candidates:

$$\begin{aligned} d\text{th-Degree polynomial: } K(x, x') &= (1 + \langle x, x' \rangle)^d, \\ \text{Radial basis: } K(x, x') &= \exp(-\gamma \|x - x'\|^2), \\ \text{Neural network: } K(x, x') &= \tanh(\kappa_1 \langle x, x' \rangle + \kappa_2). \end{aligned} \quad (12.22)$$

Example: $d=2$

$$\begin{aligned} K(X, X') &= (1 + \langle X, X' \rangle)^2 \\ &= (1 + X_1 X'_1 + X_2 X'_2)^2 \\ &= 1 + 2X_1 X'_1 + 2X_2 X'_2 + (X_1 X'_1)^2 + (X_2 X'_2)^2 + 2X_1 X'_1 X_2 X'_2. \end{aligned} \quad (12.23)$$

Then $M = 6$, and if we choose $h_1(X) = 1$, $h_2(X) = \sqrt{2}X_1$, $h_3(X) = \sqrt{2}X_2$, $h_4(X) = X_1^2$, $h_5(X) = X_2^2$, and $h_6(X) = \sqrt{2}X_1 X_2$, then $K(X, X') = \langle h(X), h(X') \rangle$. From (12.20) we see that the solution can be written

$$\hat{f}(x) = \sum_{i=1}^N \hat{\alpha}_i y_i K(x, x_i) + \hat{\beta}_0. \quad (12.24)$$

The kernel property of the support vector machine is not unique!

SVM과 차원의 저주

사고실험 1.

A. $\triangle - \triangle - \triangle - \triangle - \triangle - \triangle - \triangle - \triangle - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge$

B. $\blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \triangle - \triangle - \triangle - \triangle - \triangle - \triangle - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge - \blacklozenge$

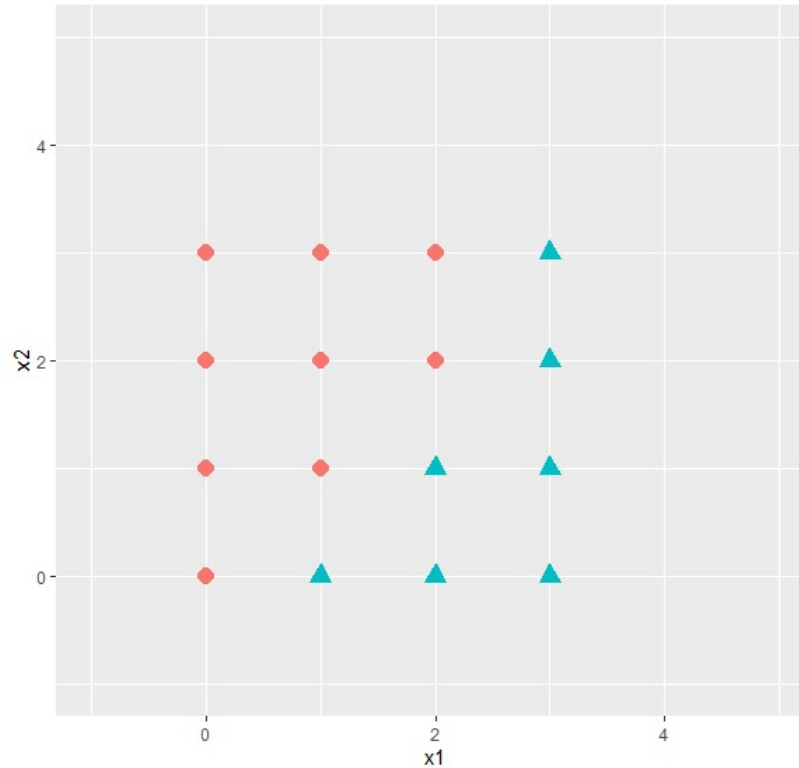
어느 것이 분류가 더 쉬운가?

→ 인간에게는 쉽지만, 컴퓨터에게는 까다로운 질문.

빠르고 (computational) 정확한 (reduced test error) 모델을 찾는 것이 컴퓨터의 목표이다.

...CONTINUED

사고실험 2.



느리더라도 고차원의 성능이 좋지 않나?

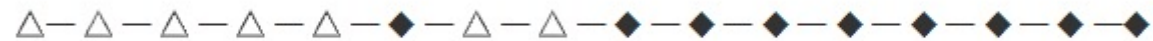
→ 그렇지 않다.

오히려 training data 내 노이즈의 영향을 크게 받는 과적합 문제가 발생할 수 있다.



...CONTINUED

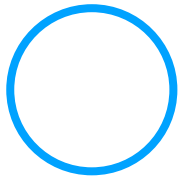
C , the cost of slack variables.



C 가 작다면?

- 하나를 slack variable로 처리하고
중간에서 자르면 된다!
- high bias, low variance
- underfitting

C 가 크다면?

- 복잡한 kernel 을 이용해서 margin을
확보해야 한다.
 - low bias, high variance
 - overfitting
- 



LOSS + PENALTY




Recall:

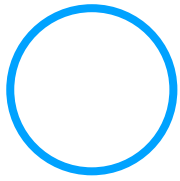
$$\begin{aligned} & \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \\ & \text{subject to } y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i, \\ & \xi \geq 0, \sum \xi_i \leq \text{constant}' \end{aligned}$$

Rephrase:

$$\begin{aligned} & \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \\ & \text{subject to } [1 - y_i(x_i^T \beta + \beta_0)]_+ \leq \xi_i, \\ & \sum \xi_i \leq \text{constant}' \end{aligned}$$



Rephrase:

$$\begin{aligned} & \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 \\ & \text{subject to } \sum_{i=1}^N [1 - y_i(x_i^T \beta + \beta_0)]_+ \leq \text{constant}' \end{aligned}$$




...CONTINUED

Final optimization problem:

$$\min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N [1 - y_i f(x_i)]_+$$

Which is equivalent to:

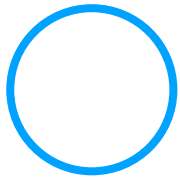
$$\min_{\beta_0, \beta} \sum_{i=1}^N [1 - y_i f(x_i)]_+ + \frac{\lambda}{2} \|\beta\|^2 \quad (12.25)$$

→ **Loss + Penalty** form!

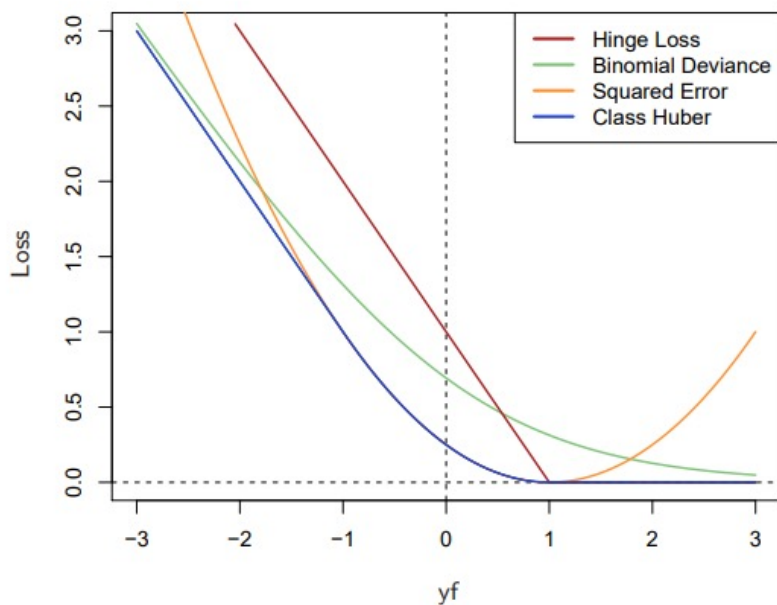
Compare it with:

$$\begin{aligned} \min_{\beta, \beta_0} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^N \xi_i \\ \text{subject to } \xi_i \geq 0, y_i(x_i^T \beta + \beta_0) \geq 1 - \xi_i \quad \forall i, \end{aligned} \quad (12.8)$$

→ The solutions are equal when $C = 1/\lambda$.



LOSS FUNCTIONS



$$\text{Risk function } R[y, f(x)] = E_y[L[y, f(x)]]$$

Loss Function	$L[y, f(x)]$	Minimizing Function
Binomial Deviance	$\log[1 + e^{-yf(x)}]$	$f(x) = \log \frac{\Pr(Y = +1 x)}{\Pr(Y = -1 x)}$
SVM Hinge Loss	$[1 - yf(x)]_+$	$f(x) = \text{sign}[\Pr(Y = +1 x) - \frac{1}{2}]$
Squared Error	$[y - f(x)]^2 = [1 - yf(x)]^2$	$f(x) = 2\Pr(Y = +1 x) - 1$
“Huberised” Square Hinge Loss	$-4yf(x), \quad yf(x) < -1$ $[1 - yf(x)]_+^2 \quad \text{otherwise}$	$f(x) = 2\Pr(Y = +1 x) - 1$



PENALTY



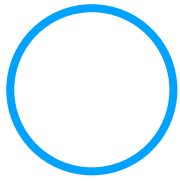
Ridge Regression:

$$L(\beta) = \min_{\beta} \sum_{i=1} (y_i - \hat{y}_i)^2 + \lambda \|\beta\|^2$$

Compare it with SVM:

$$\min_{\beta_0, \beta} \sum_{i=1}^N [1 - y_i f(x_i)]_+ + \frac{\lambda}{2} \|\beta\|^2 \quad (12.25)$$

→ Loss function 만 다르고 같은 R2 penalty 를 공유한다!

- Ridge Regression: Squared Error
 - LDA also uses squared error but without the penalty term.
 - SVM: SVM Hinge Loss
- 

SVM & LOGISTIC REGRESSION

Again the **loss functions** are different!

- SVM: SVM Hinge Loss
- Logistic Regression: Binomial Deviance, the negative (-) binomial log-likelihood loss function

Logistic Regression

$$\begin{aligned}Pr(Y = 1|x) &= \frac{1}{1 + e^{-f(x)}} \\Pr(Y = 0|x) &= \frac{1}{1 + e^{f(x)}}\end{aligned}$$

$$\begin{aligned}Y_{new} &= 2(Y - 0.5), Y \in \{0, 1\} \\&\text{so that } Y_{new} \in \{-1, 1\}\end{aligned}$$

$$\begin{aligned}Pr(Y_{new} = 1|x) &= \frac{1}{1 + e^{-f(x)}} \\Pr(Y_{new} = -1|x) &= \frac{1}{1 + e^{f(x)}}\end{aligned}$$

Binomial Likelihood

$$Pr(Y = 1|x)^Y \times Pr(Y = 1|x)^{1-Y}$$

Loss = Negative (-) binomial log-likelihood

$$\begin{aligned}Loss &= -Y \log Pr(Y = 1|x) - (1 - Y) \log Pr(Y = 0|x) \\&= -\frac{Y_{new} + 1}{2} \log Pr(Y_{new} = 1|x) - \frac{1 - Y_{new}}{2} \log Pr(Y_{new} = -1|x) \\&= -\frac{Y_{new} + 1}{2} \log \frac{1}{1 + e^{-f(x)}} - \frac{1 - Y_{new}}{2} \log \frac{1}{1 + e^{f(x)}} \\&= \frac{Y_{new} + 1}{2} \log (1 + e^{-f(x)}) + \frac{1 - Y_{new}}{2} \log (1 + e^{f(x)}) \\&= \log (1 + e^{-Y \cdot f(x)}) \because Y \in \{-1, 1\}\end{aligned}$$

Risk minimizer f

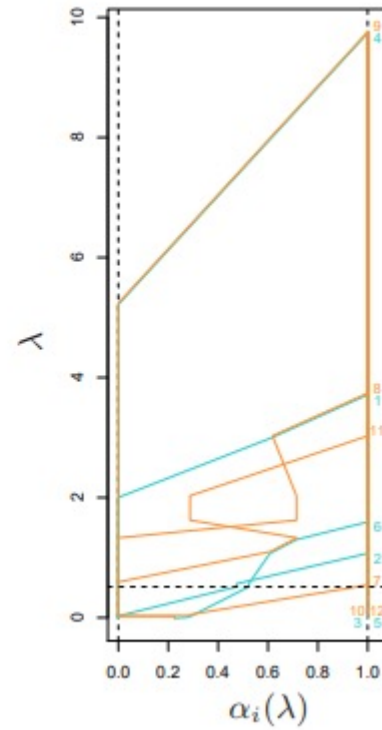
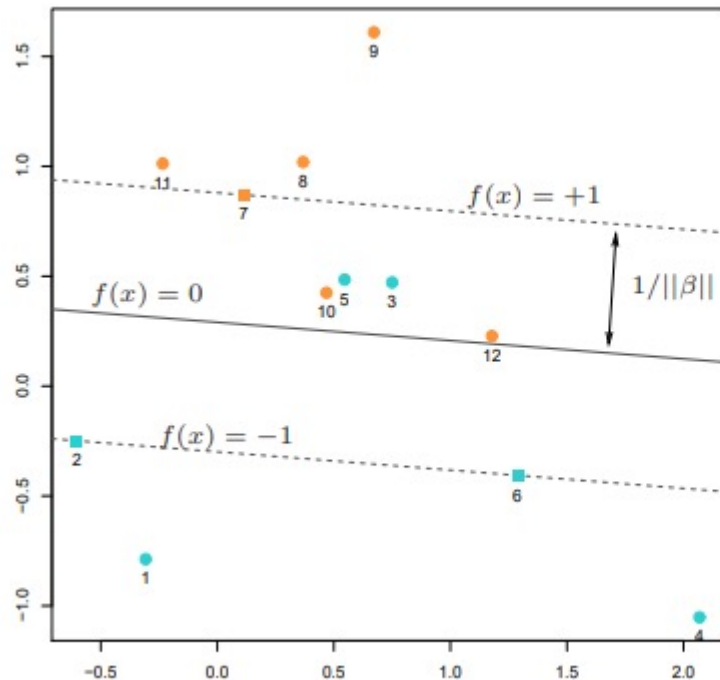
$$\begin{aligned}\hat{f}(x) &= \log \frac{\hat{Pr}(Y = +1|x)}{\hat{Pr}(Y = -1|x)} \\&= \hat{\beta}_0 + \sum_{i=1}^N \hat{\alpha}_i K(x, x_i),\end{aligned}\tag{12.31}$$

$$\hat{Pr}(Y = +1|x) = \frac{1}{1 + e^{-\hat{\beta}_0 - \sum_{i=1}^N \hat{\alpha}_i K(x, x_i)}}.\tag{12.32}$$

Rule of Thumb

- Use SVM when the classes are well separated!
- Use logistic regression in more overlapping regimes.

PATH ALGORITHM FOR PARAMETERS




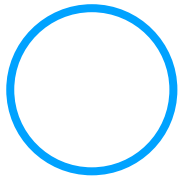
Optimization function and the corresponding solution

$$\min_{\beta_0, \beta} \sum_{i=1}^N [1 - y_i f(x_i)]_+ + \frac{\lambda}{2} \|\beta\|^2 \quad (12.25)$$

$$\beta_\lambda = \frac{1}{\lambda} \sum_{i=1}^N \alpha_i y_i x_i. \quad (12.33)$$



...CONTINUED

- α 's all lie in $[0, 1]$
 - $\alpha = 0 \leftarrow yf > 1$: the observations are correctly classified outside their margins.
 - $\alpha = 1 \leftarrow yf < 1$: the observations inside their margins.
 - $\alpha = 0.XX \leftarrow yf = 1$: the observations sitting on their margins.
 - λ is large $\rightarrow ||\beta||$ is small $\rightarrow M = 1 / ||\beta||$ is large \rightarrow The margin is wide.
 - We decrease λ from an initial large value.
 - Some observations inside the margin comes out. ($\alpha = 1 \rightarrow \alpha = 0$)
 - Meanwhile it smoothly changes the α on the margins. $\alpha(\lambda)$
- 
- 



- the end -

숙제는 내일 밤 깃헙을 확인해주세요!