20기 정규세션

ToBig's 19기 강의자

이동우

SVM

Support Vector Machine

Contents

Unit 01 | SVM

Support Vector Machine

: 주로 '분류'를 하기 위해 사용되는 기법이나, 회귀 역시도 적용 가능(SVR)

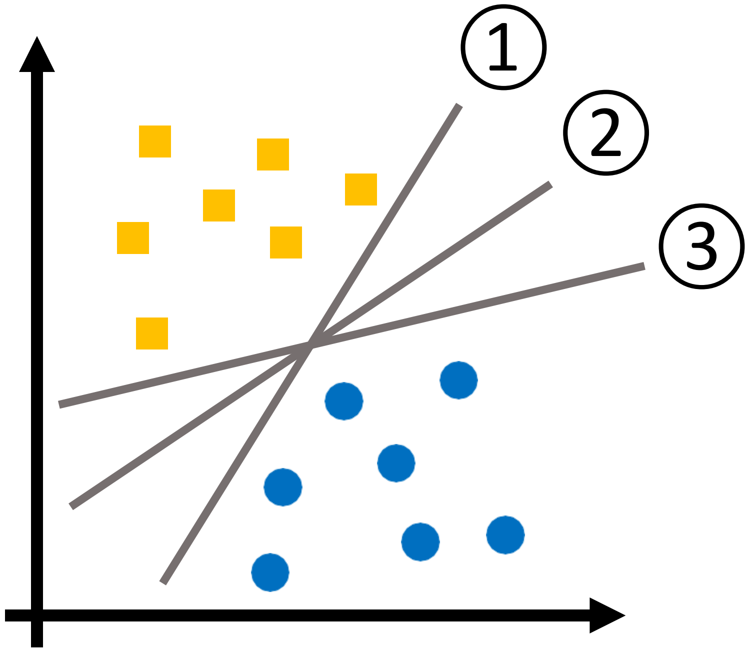
: 딥러닝 이전에 높은 성능으로 주목받은 모델

SVM의 분류

선형여부	분류
선형	Linear svm
비선형	Non-linear svm

오분류허용여부	분류
X	Hard margin svm
O	soft margin svm

1. SVM?

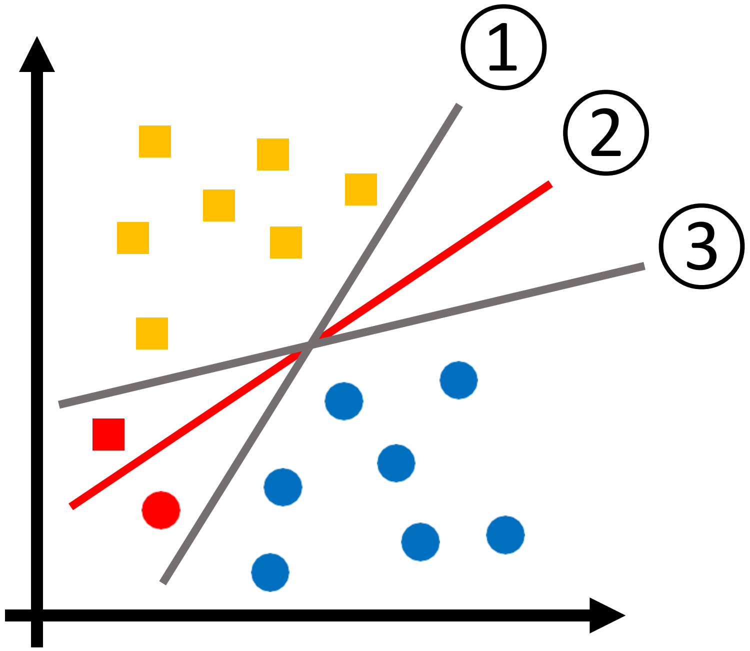


Binary Classification Problem;

데이터들을 가장 잘 나눈 선은
어떤 선일까요?

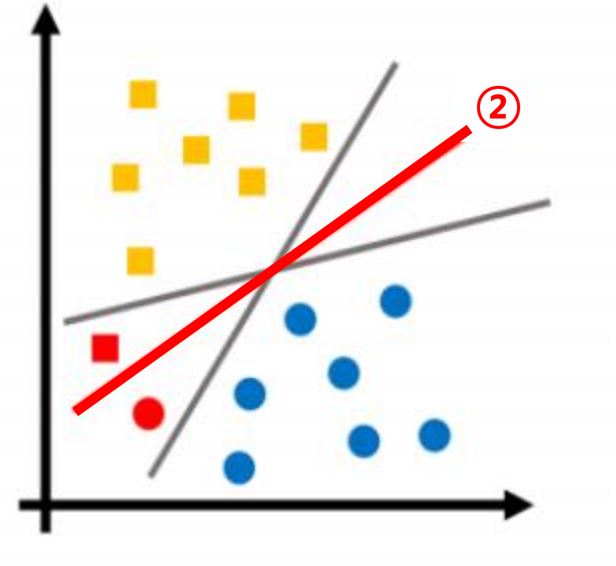
좋다라는 것의 기준은?

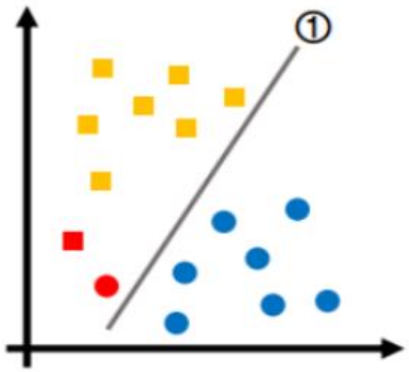
1.SVM?



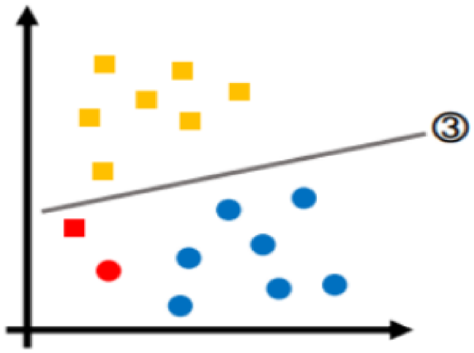
② 정답은 2번! 이유는?

SVM (support vector machine)





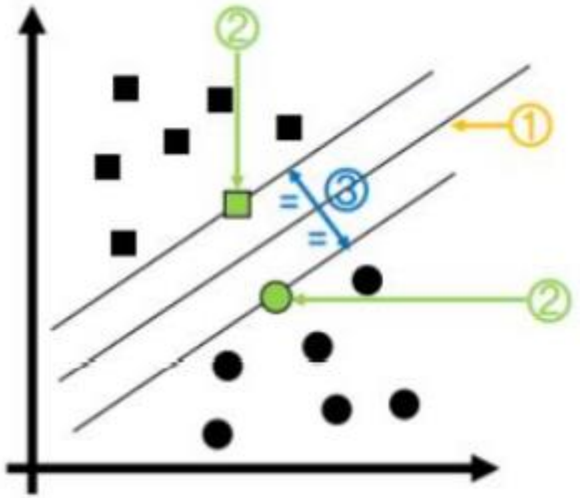
① 동그라미를 분류 하지 못함(X)



3.네모를 분류 하지 못함(X)

새로운 점이 추가 되었을 경우를 가정
'잘 나누었다' = 여유 있게 결정 경계를 생성했다
즉, 일반화된 성능을 잘 뽑아내느냐의 문제.

SVM (support vector machine) **관련 용어**



①Hyperplane

:여러 데이터를 나누는 기준이 되는 경계(초평면)

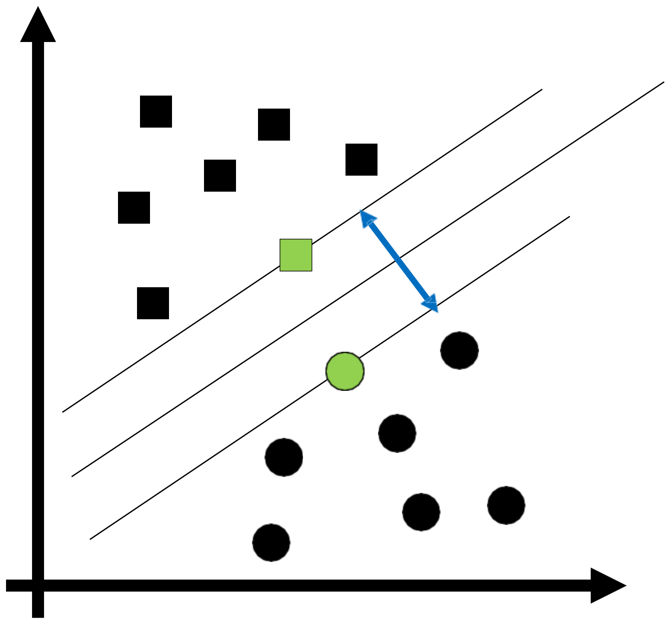
②Support Vector

:각 Class별로 Hyperplane과 가장 가까운 데이터

③Margin

:결정경계와 서포트 벡터 사이의 거리 X 2

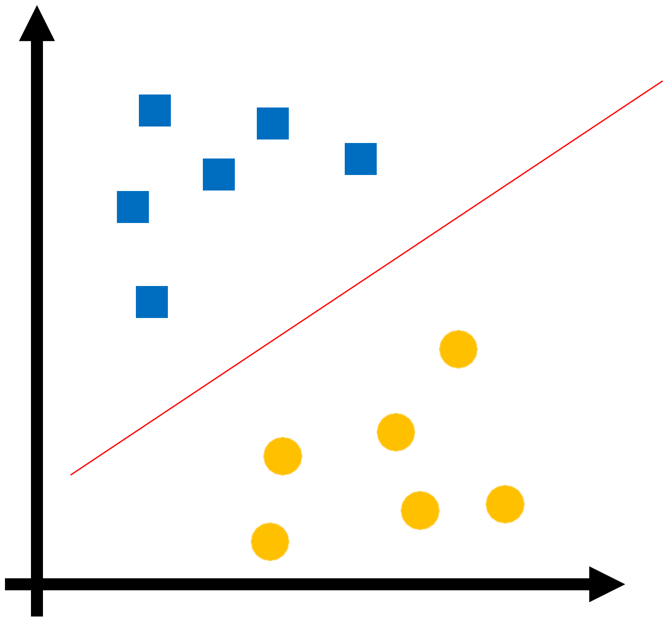
Ideas

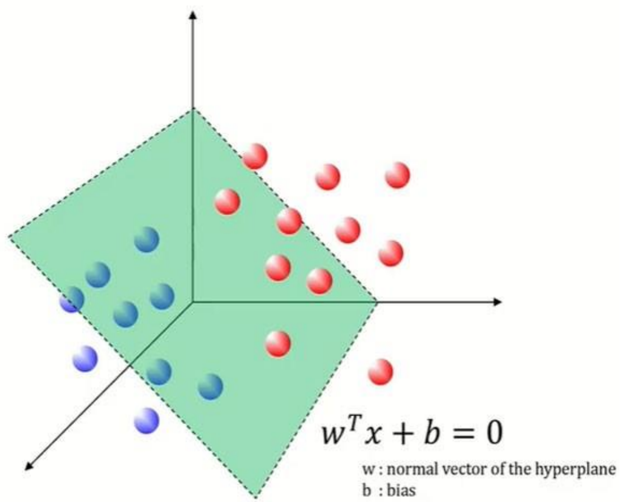


SVM:

: 마진을 구하는 방법을 공식화하고
이 마진을 최대화하는 결정 초평면
(decision hyperplane)을 찾는 것이 목표.

3-0. Hyperplane 수식 정의





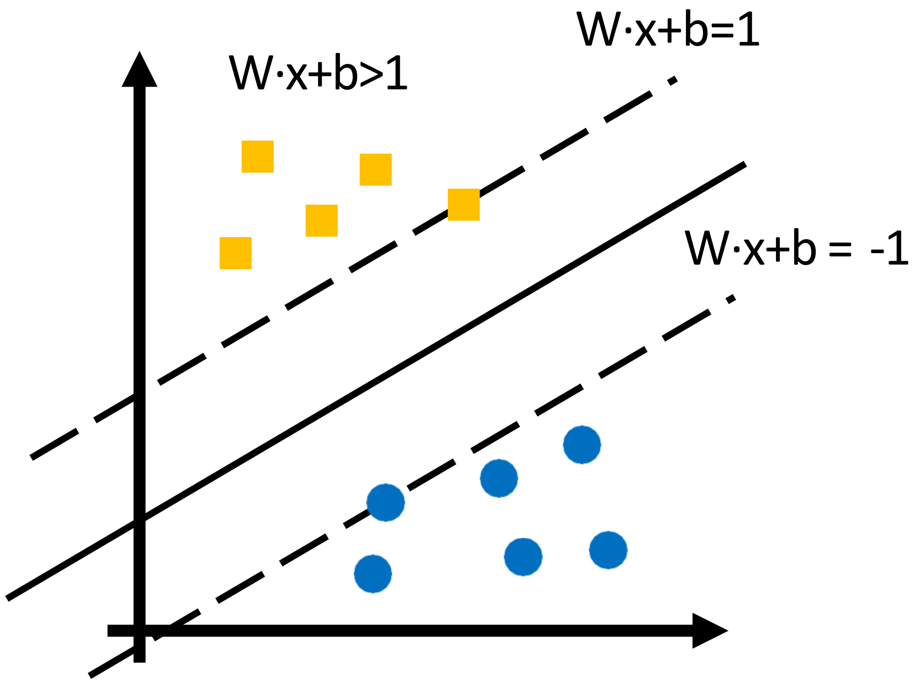
N차원에서의 분류를 수행하는 초평면은, N-1 차원으로의 Subspace로서 정의됨.

이 때 Hyperplane은 법선벡터 w와 실수 b를 활용하여,

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

으로서 표현이 가능.

3-1. y = ±1 분류문제 상정



편의상,y=±1 분류 문제를 푼다고 가정.

■ : (+1) class에 속하도록 하려면,

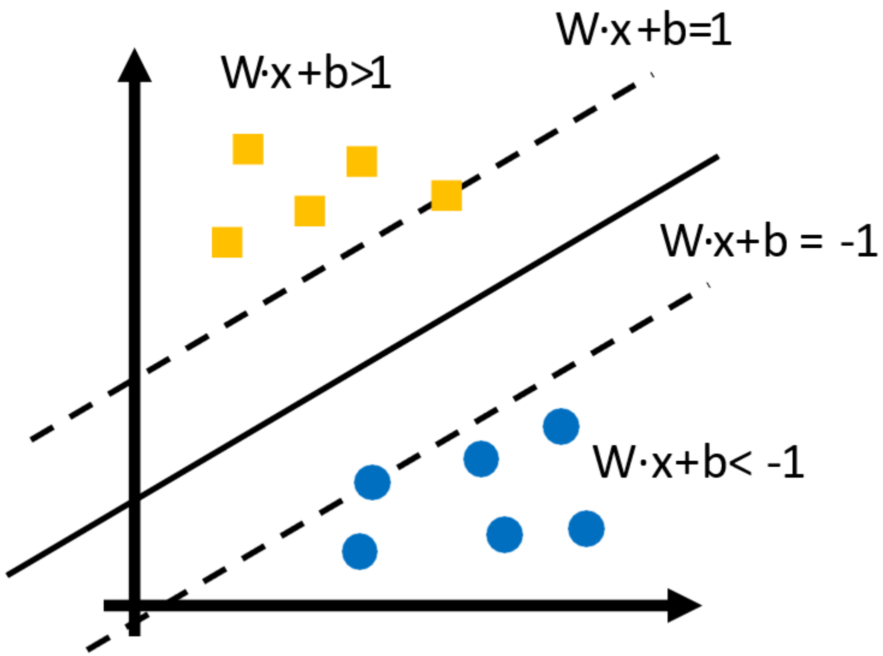
$$w^T x_+ + b \geq +1$$

● : (-1) class에 속하도록 하려면,

$$w^T x_- + b \leq -1$$

와 같이 설정.
이 때 하나의 식으로 줄여주기 위해,
이진변수를 도입!

3-2. 식의 간소화

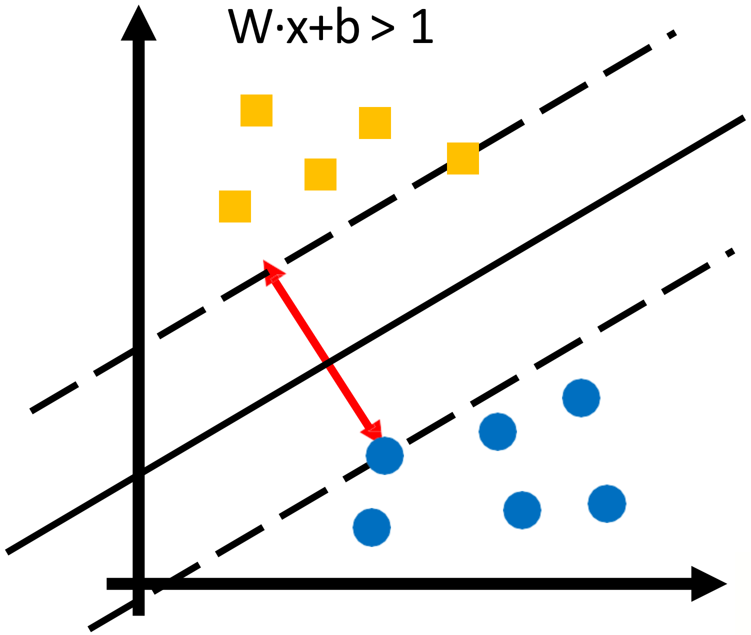


수식으로나타내면?

$$y_i\,(w \cdot x_i + b) \geq 1$$

단, $y_i = \begin{cases} +1 \text{ for } \blacksquare \text{ sample} \\ -1 \text{ for } \bullet \text{ sample} \end{cases}$

3-3. margin



$$x^+ = x^- + \lambda w$$

$w^T x^+ + b = 1$ $\quad x^+$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

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

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

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

길의 너비(margin)

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

지금까지의수식을정리하면,

1. 제약식(조건)

: $y_i\,(w \cdot x_i + b) \geq 1$

(이는, 모든 데이터가 선형적 hyperplane에 의해 '완벽히' 분류된다는 것을 가정한 것)

2. 목적식

: $\frac{2}{||w||}$ 가 최대가 되게 하고, 동시에 위의 제약식을 만족하는 W와 b를 찾자 -> margin이 최대

$:\max\left(\frac{2}{||w||}\right) > \min(||w||) > \min\left(\frac{||w||^2}{2}\right)$ 분수식 싫다 -> 2-norm 루트식 싫다 -> 루트 없는 형태

목적함수는 2차식이고, 제약식은 선형식이다 -> Quadratic(2차)계획법
-> convex optimization -> 전역최적해 존재

3-4. 라그랑주승수법

그렇다면, 앞의 Convex Optimization 문제를 어떻게 해결할 것인가?

-> Lagrangian 승수법: 최대 또는 최소값을 찾으려는 문제에서 해결방법으로 사용됨.

목적함수 $f(x, y)$와 제약 조건 $g(x, y) = 0$ 에 대해, 새로운 변수 λ 를 이용하여,

$$L(x, y, \lambda) = f(x, y) - \lambda g(x, y)$$

라는 보조 방정식을 만들고, 이 보조방정식의 해를 구해내는 문제로 변형시킬 수 있음.
즉, '제약식에 Lagrangian 승수를 곱한 항'을 최적화 하려는 목적식에 더하여, 제약이 있는 문제를 제약 없는 문제로 바꾸어 푸려는 기법.

Recall. 목적함수는 Maximizing the margin (= minimizing the squared second norm)
And 제약조건은 $y_i (w \cdot x_i + b) \geq 1$ -> 등호가 아닌데 어떻게 하지? => KKT 조건을 만족하면 가능.

3-4. 라그랑주승수법 -> Lagrangian Primal 꼴 도출.

Original Problem)

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

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

Using Lagrangian Multiplier α_i, Lagrangian Primal로 변환)

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

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

Unit 01 | SVM

Original Problem

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

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

Lagrangian multiplier를 이용하여 Lagrangian primal문제로 변환

Lagrangian Primal

① ②

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

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

Convex, continuous이기 때문에 미분 = 0에서 최소값을 가짐

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

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

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

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

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

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

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

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

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

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

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

$$\boxed{\max_{\alpha} \min_{w,b} L(w,b,\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j}$$

3-5. Dual 형태로 변환, KKT condition 적용

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

KKT (Karush-Kuhn-Tucker) conditions:

① Stationarity

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

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

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

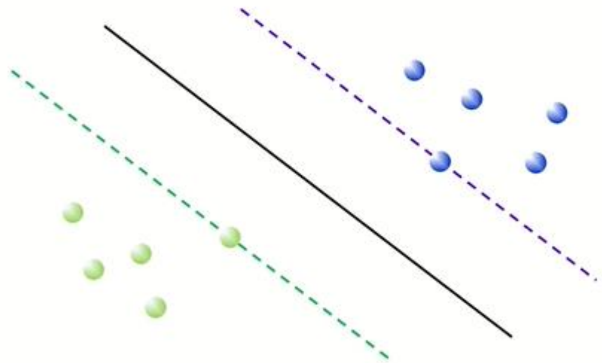
- 이와 같이 Dual 형태로 바꾸면, Primal Form보다 풀기 쉬운 형태
- 여전히, Objective function이 quadratic하고 constraint가 linear
- 향후 Kernel trick을 사용하기 더 용이(내적 꼴이 포함됨)
- Convex optimization form 유지, 전역 최적해 존재.
- 여기서의 decision variabl은 α임.

위 식에서 도출된 w, b, α가 Lagrangian Dual의 최적해가 되기 위한 조건으로서, 만족되어야 할 조건이 KKT.

1. Decision Variable에 대한 편미분 값은 0(라그랑주 승수 제외)
2. 라그랑주승수는0보다크거나같아야한다.
3. 라그랑주승수와제약식중하나는무조건0이되어야한다.

Complementary Slackness를 활용한 Dual Form에서의 Solution 도출

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



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

KKT 조건, 특히 Complementary Slackness를 만족하는 경우를 가지고 Decision variable W, b를 구한다.

1) W 구하기

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i = \sum_{i \in SV} \alpha_i^* y_i x_i$$ => Support Vector에 해당하는 애들만을 가지고 w*를 구한다.

2) b 구하기

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

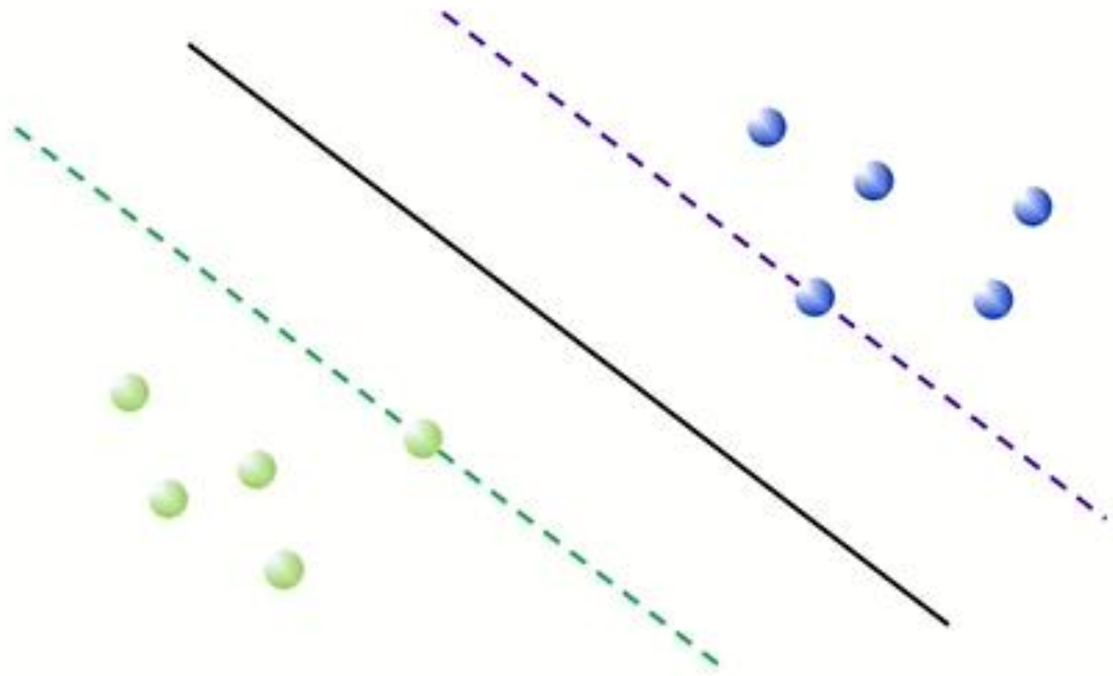
$$b^* = y_{sv} - \sum_{i=1}^{n} \alpha_i^* y_i x_i^T x_{sv}$$ =>임의의 Support vector를 대입하여, b를 구한다.

3) 분류 Hyperplane 구축, 새로운 데이터에 대해 분류 수행 가능

=> $\text{sign}(w^{*T} x_{new} + b^*) = sign\left(\sum_{i \in SV} \alpha_i y_i x_i^T x_{new} + b^*\right)$로서 class 부여

Unit 01 | SVM

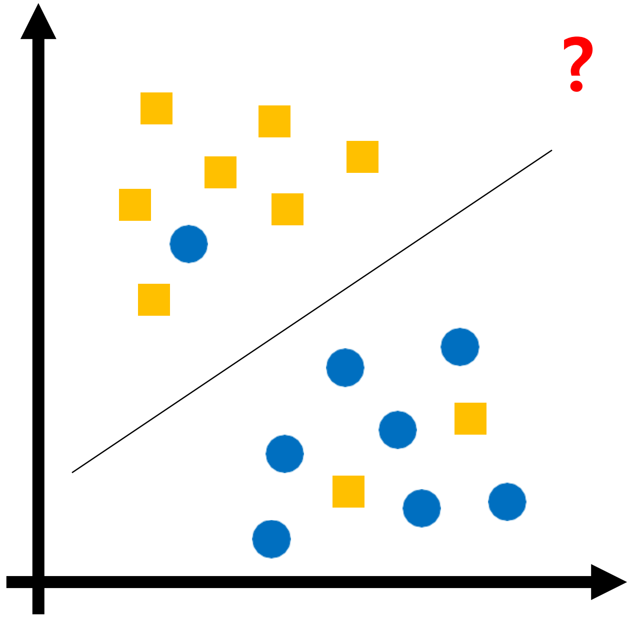
즉, 기존의 데이터들을 가지고 support vector가 위치한 margin plane을 구축하고
구해진 W와 b를 이용해 $w^T x + b = 0$ 인 초평면을 구축.
이후 초평면을 기준으로 부호를 통해 class를 결정.



Contents

Unit 02. Soft Margin SVM

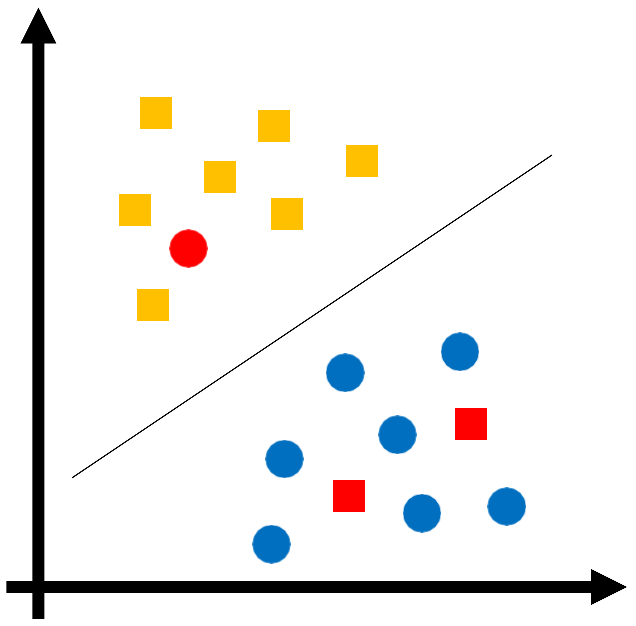
현실적으로 모든 데이터가 깔끔하게,
선형 경계로 나뉘지는 않는다!
이런 경우는 어떻게?



1. Soft Margin SVM

앞에서 진행했던 수식들을 돌이켜 보면,
한치의 error을 허용하지 않는 분류를 지향한 것.

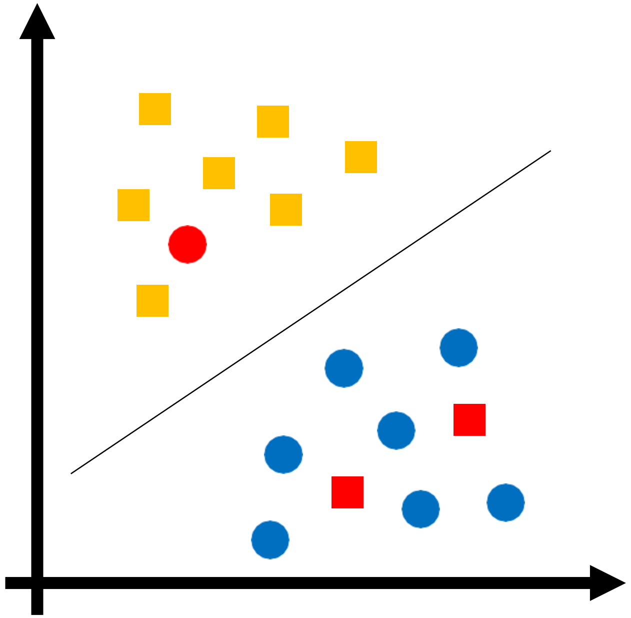
> Soft Margin SVM
:error(오분류)를 허용하되,패널티를
쥐서전체error를 최소화!



2. Penalty

> Penalty를 주는 방법

1. 0-1 Loss
2. Hinge Loss

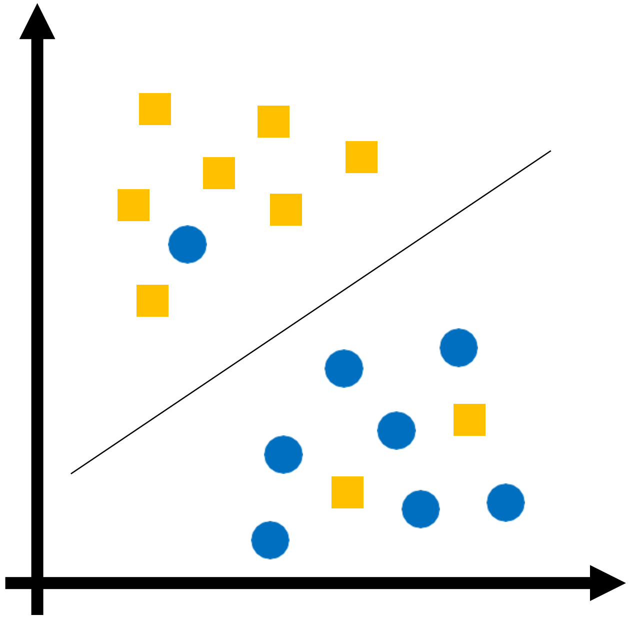


Unit 02. Soft Margin SVM

2-1.0-1 Loss

:error가 발생한 개수만큼 패널티 계산

:min $||w|| + C\#error$



2-2. Hinge Loss

:오분류 정도에 따라 error의 크기를 다르게 하자

실제로는 ● 인 데이터가 분류기의 각 영역에 있을 때

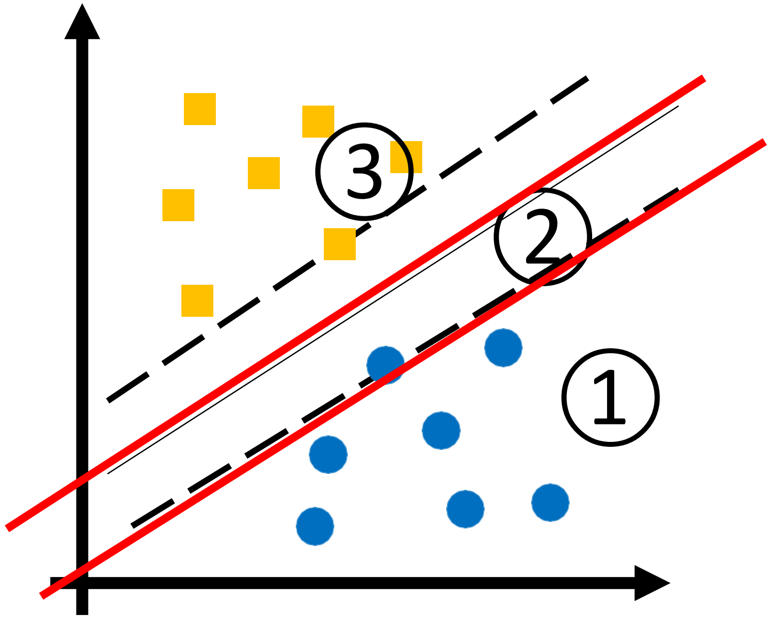
① error 없음 (좋은 분류) > $\xi j=0$

② 작은 error > $0 \leq \xi j \leq 1$

③ 큰 error (잘못된 분류) > $\xi j > 1$

> 데이터에 따라 error 크기 다르게 주기 위한

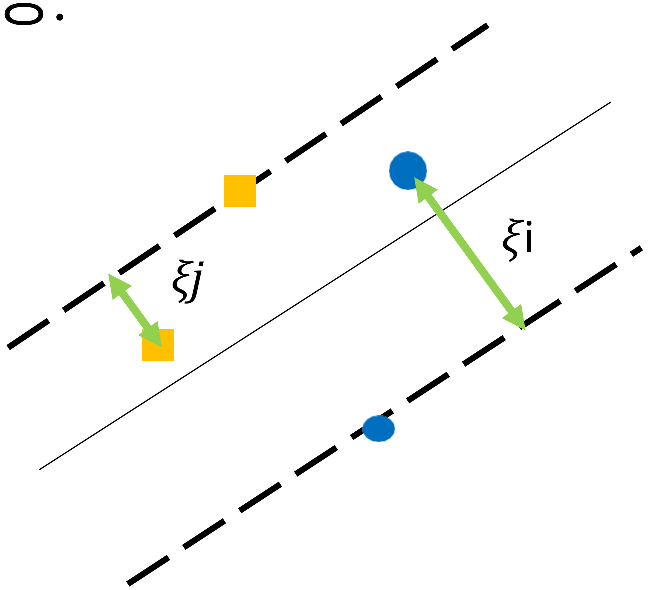
> slack variable(ξj) 사용



2-2. Hinge Loss

> 기존 목적함수에 error항을 추가, margin을 파고드는 것을 허용.

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



> 여기서 c는 하이퍼파라미터

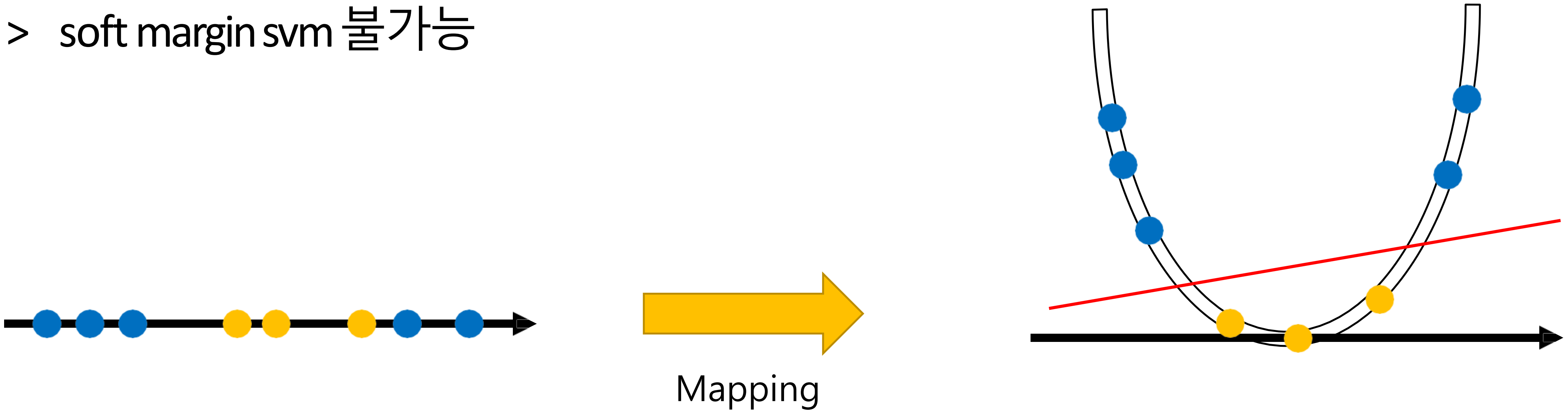
C가 크다면 > error를 줄이자. -> Hard Margin
C가 작다면 > error가 있어도 큰 영향x. W를 줄이는데 집중
작은 C는 Underfitting의 가능성, 큰 C는 overfitting 가능성 존재.

Contents

구분할수 있는linear line이 없는데
Outlier를 무시하지 못 하는 경우에는?

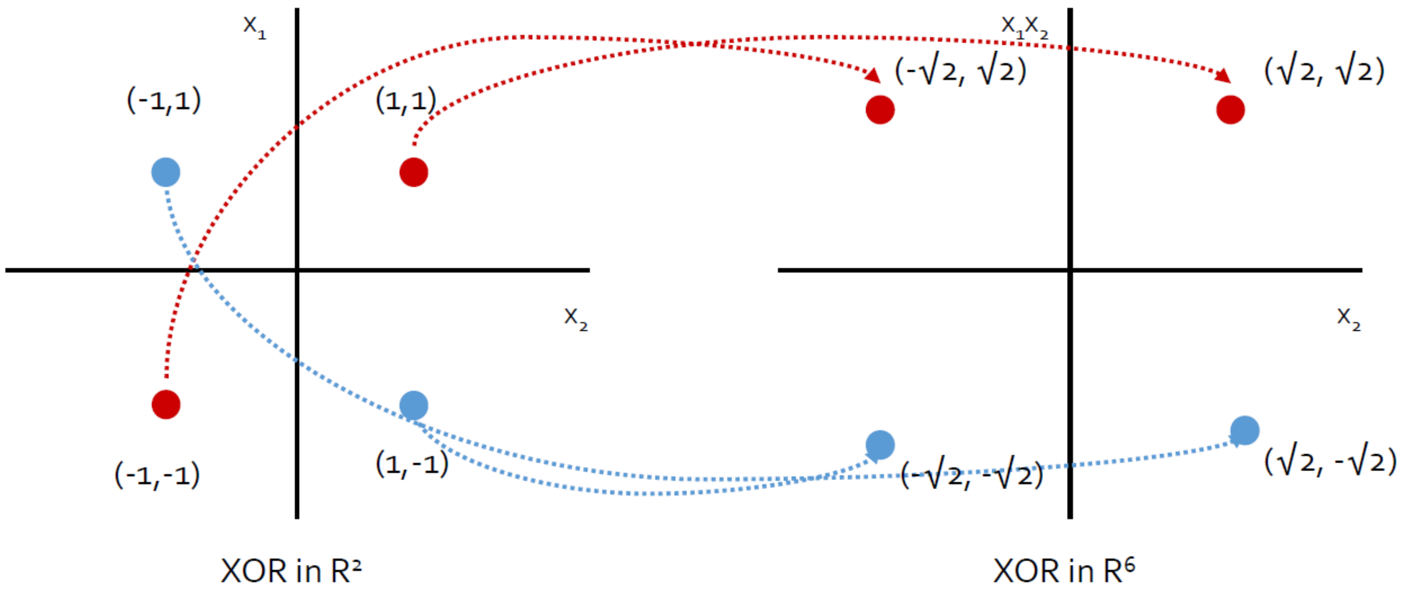
> linear svm 불가능
> soft margin svm 불가능

저차원에서선형경계로분류불가능한data를
고차원으로 매핑하면 분류 가능!



1. Mapping with Φ

Ex) $\Phi: (x_1, x_2) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, 1)$



선형 분리 불가

선형 분리 가능(고차원에서)

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

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

2. Using Kernels

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

Mapping 함수를 적용한 후, 그 값을 또 내적해야함.
->즉, 모든 관측치에 대해 고차원으로 매핑을 하고,
-> 이를 다시 내적해야함.

Kernel Trick:
⇒ 직접 mapping 함수를 이용해 데이터를 변환하는 것이 아닌, 그 내적 값만을 바로 구하는 Kernel을 대신 사용!
⇒ 내적 값인 Scalar 값을 바로 계산하여 활용, feature map 및 복잡한 mapping function 생각 안해도 됨
참고) Valid Kernel을 위한 조건(kernel matrix가 symmetric positive-semi definiite) 또한 존재, Mercer's Theorem
https://ratsgo.github.io/machine%20learning/2017/05/30/SVM3/

2.Using Kernels

예제)

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

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

수동 Mapping)

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

Polynomial Kernel) 2nd Order

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

⇒ 직접 mapping 함수를 구하지 않고
Kernel을 이용해 빠르게 내적값을 구함

- 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)

$$K\langle x_1, x_2\rangle = \exp\left(\frac{-\|x_1 - x_2\|_2^2}{2\sigma^2}\right) \Rightarrow e^{-\gamma(a-b)^2}$$

Unit 03. Non-Linear SVM

3. RBFkernel (=Gaussian Kernel)

:무한대의 차원으로 매핑하는 커널 (by 테일러급수)

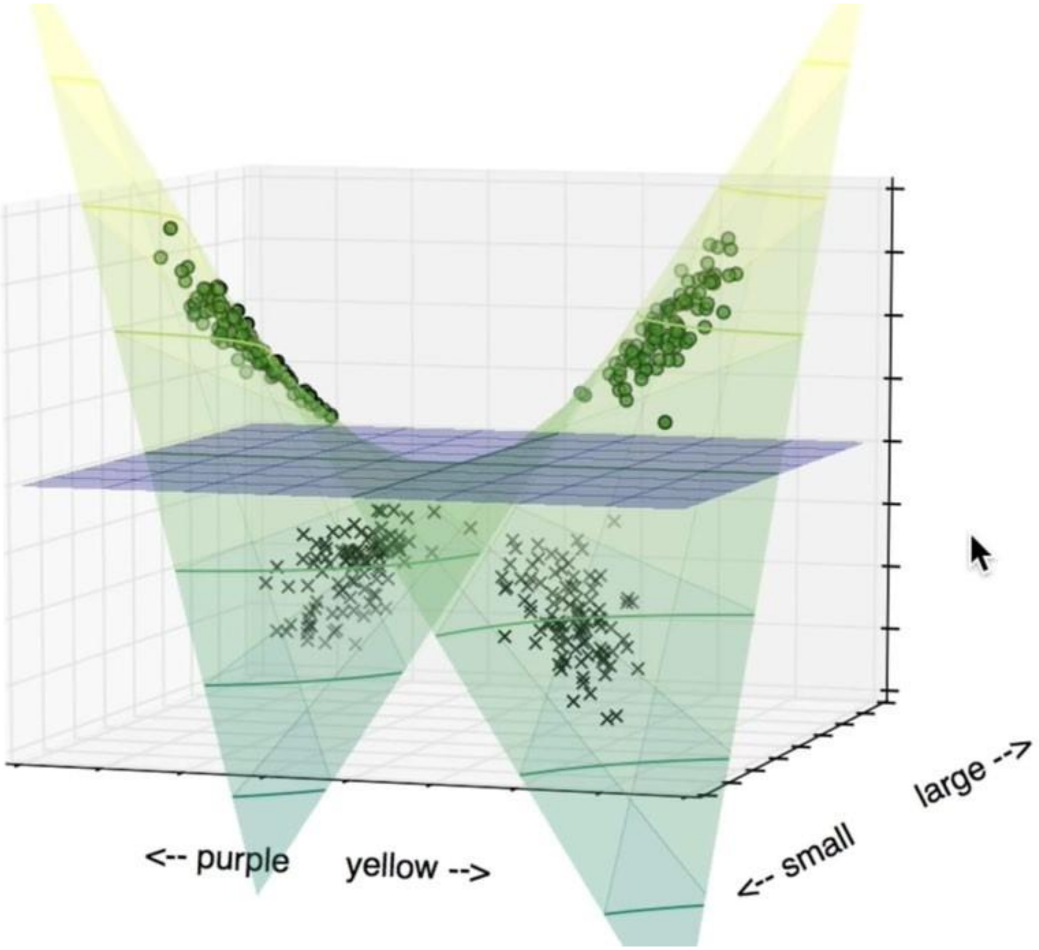
Ex) $2\sigma^2 = 1$ =>

$$K(x_1, x_2) = exp\left\{-(x_1 - x_2)^2\right\}$$
$$= exp(-x_1)exp(-x_2)exp(2x_1x_2)$$

여기서, 테일러급수에 의해

$$exp(2x_1x_2) = \sum_{k=0}^{\infty} \frac{2^k x_1^k x_2^k}{k!}$$

> 무한대 차원의 Feature Space로 매핑!

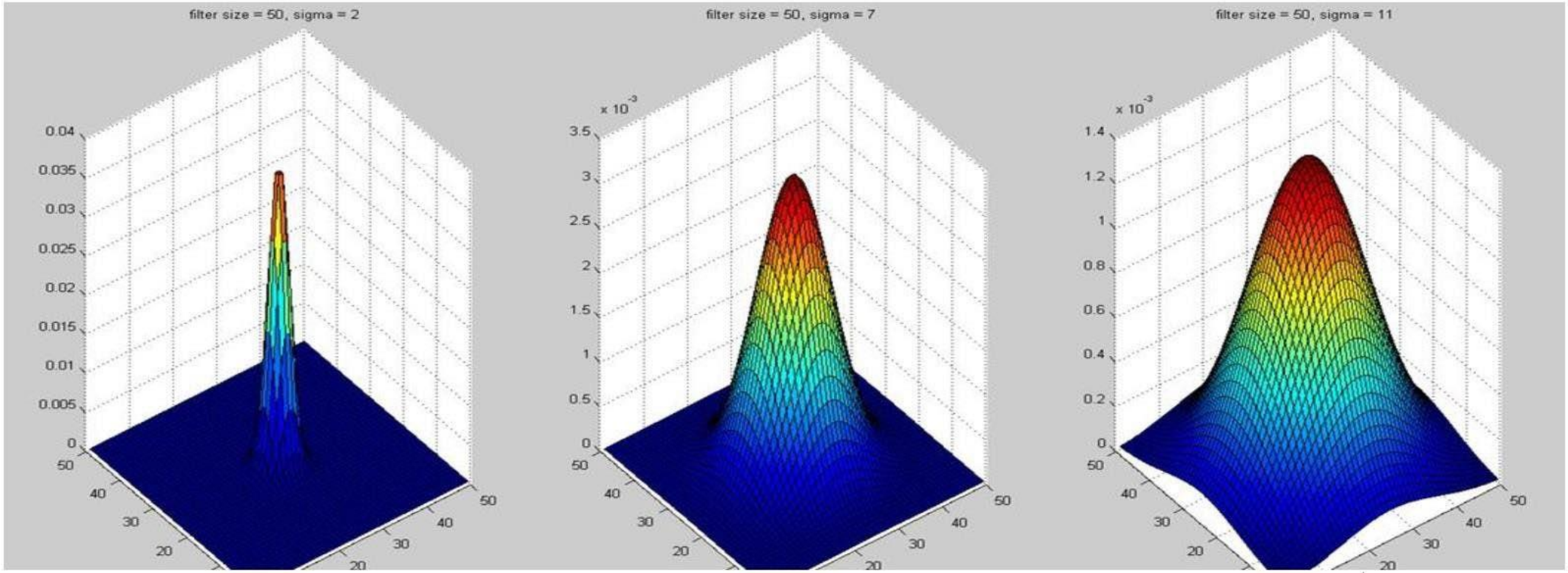


3. RBF Kernel-Gamma Parameter Tuning

$$K(x_1, x_2) = exp\left\{-\frac{\|x_1 - x_2\|_2^2}{2\sigma^2}\right\}, \quad \sigma \neq 0 \quad \Rightarrow \quad e^{-\gamma(a-b)^2}$$

⇒ $\gamma = \frac{1}{2\sigma^2}$ ⇒ 감마와 분산(표준편차)은 반비례 관계

3. RBF Kernel-Gamma Parameter Tuning



Gamma 감소 = 표준편차 증가

3. RBF Kernel-Gamma Parameter Tuning

:감마가 클수록
휠씬 인접한 것들만 같은 영역으로 본다
=엄청 인접하지 않으면 엄청 먼 곳으로 인식한다
=원래 차원으로 돌아왔을 때 경계가 아주 촘촘하다
=Hyper plane이 휠씬 더 굴곡지다
=Ovefitting의 가능성이 높다

…we get **0.11** when ***γ*** = **1**.

$$e^{-(2.5-4)^2} = e^{-(-1.5)^2} = e^{-2.25} \boxed{= 0.11}$$

When ***γ*** = **2**, we get **0.01**…

$$e^{-2(2.5-4)^2} = e^{-2(-1.5)^2} = e^{-2\times 2.25} \boxed{= 0.01}$$

Contents

Unit 04 | 확장/과제 설명

EFSVM: SVM의 변형

Class Imbalance problem일 경우에, 어떻게 대처해야 하나?

-> 숫자가 상대적으로 적은 Class를 분류하는 것이, 더 중요(e.g. Credit card fraud detection task)

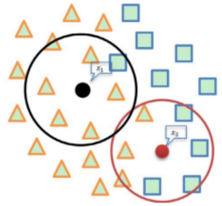
-> SVM의 decision surface 결정시, 다수 클래스의 영향력을 축소하는 데 목적을 두고 EFSVM을 설계.

-> 각 데이터의 '중요도'를 Entropy를 사용하여 변환

$$H_i = -p_{+i} ln(p_{+i}) - p_{-i} ln(p_{-i})$$

$$p_{+i} = \frac{num_{+i}}{k}$$
$$p_{-i} = \frac{num_{-i}}{k}.$$

» $k = 7$ 예시:



	num_+	num_-	p_+	p_-	H
x_1	6	1	6/7	1/7	0.41
x_2	4	3	4/7	3/7	0.68

-> K는 각 데이터의 nearest neighbor 숫자 parameter.

-> Entropy 값이 클수록 margin에 가까운 데이터, 즉 특정 class에 속한다고 확신하는 정도가 낮아짐.

EFSVM: SVM의 변형

Class Imbalance problem일 경우에, 어떻게 대처해야 하나?

-> Negative(다수) Class를 increasing order로 정렬 후, M개의 subset으로 나눔

-> H_{Sub_l} 이 l번째 subset의 Entropy라 할 때, $H_{Sub_1} < H_{Sub_2} < \cdots < H_{Sub_m}$

-> 각 Subset l에 대해 Entropy based Fuzzy Membership을 부여

$$FM_l = 1.0 - \beta * (l - 1), \ l = 1, 2, \ldots, m$$

$$0 \le \beta \le \frac{1}{m-1} \qquad s_i = \begin{cases} 1.0 & \text{if } y_i = +1 \\ FM_l & \text{if } y_i = -1 \& x_i \in Sub_l \end{cases}$$

https://www.sciencedirect.com/science/article/abs/pii/S0950705116303495?via%3Dihub

이 외에도 여러 SVM의 변형 존재, 여기서 한단계 더 발전된 형태: IEFSVM 등

Unit 04. 확장/과제 설명

Multi Class SVM

: 지금까지는, binary한 classification을 수행하는 SVM에 대해 살펴보았음.

⇒ Class가 여러 개인 경우에 대해서는 어떻게 진행할까?

1. One vs One:

클래스가 N개 있을 때, 모든 Class에 대해 1:1로 binary분류를 하고 제일 많이 승리한 것에 대해 투표로 결정.

e.g.) 클래스 A,B,C. 3개의 머신 필요

M1: A OR B. M2: B OR C. M3: C OR A.

=> 새로운 데이터가 들어오고, M1은 A, M2는C, M3는 A -> 이 데이터는 A!

Unit 04. 확장/과제 설명

Multi Class SVM

: 지금까지는, binary한 classification을 수행하는 SVM에 대해 살펴보았음.

⇒Class가 여러 개인 경우에 대해서는 어떻게 진행할까?

2. One vs Rest:

Class가 N개 있으면 모든 Class에 대해 1:N-1로 binary 분류하여 이 클래스가 맞는지 아닌 지를 판단하고 투표로 결정

e.g.) 클래스 A,B,C, 3개의 머신 필요.

M1: A OR NOT A M2: B OR NOT B M3: C OR NOT C

새로운 데이터에 대해, M1: A! M2: NOT B M3: NOT C -> 새로운 데이터는 A!

Unit 04 | 확장/과제 설명

Assignment 1. Multiclass SVM 구현

1. Multiclass SVM을 직접 구현하시는 것입니다. 기본적으로 사이킷런에 있는 SVM은 멀티클래스 SVM을 지원하지만 과제에서는 절대 쓰면 안됩니다! Iris 데이터는 총 세 개의 클래스가 있으므로 이 클래스를 one- hot인코딩한 뒤, 각각 binary SVM을 트레이닝하고 이 결과를 조합하여 multiclass SVM을 구현하시면 됩니다

2. 기본적으로 one vs one, one vs rest 방법이 있으며 둘 중 자유롭게 구현해주세요. 만약 투표결과가 동점으로 나온 경우(예를 들어, 각각의 SVM 결과가 A vs B 의 경우 A로 판별, B vs A 의 경우 B로 판별, C vs A 의 경우 C로 판별한 경우 투표를 통해 Class를 결정할 수 없음)

1) decision_function을 활용하시거나
2) 가장 개수가 많은 클래스를 사용하시거나
3) 랜덤으로 하나를 뽑거나 하는 방법 등을 이용해 동점자인 경우를 판별 해주시면 됩니다.

공식문서를 통해 사이킷런이 어떤 방법으로 구현했는지 참고하셔도 됩니다

3. 과제코드에는 iris 데이터를 로드하고 스케일링 부분까지 구현되어 있습니다.

4. Iris의 클래스는 3개입니다. Iris 데이터셋 뿐만 아니라 다른 데이터셋에도 적용 가능한, 클래스의 수와 무관한 Multiclass SVM을 만들어주세요.

-투빅스 14기 정재윤님 강의자료
-투빅스 17기 유현우님 강의자료
-투빅스 18기 손유진님 가의자료
-고려대학교 김성범 교수님 강의자료
-한양대학교 송재욱 교수님 강의자료
https://www.youtube.com/watch?v=qFg8cDnqYCI
-Patrick Winston, MIT OCW6.034 Fall 2010, Lec.16 Learning: Support Vector Machines
https://www.youtube.com/watch?v=Qc5IyLW_hns
-앤드류응(Andrew Ng)교수님의Machine Learning강의
https://www.youtube.com/watch?v=oGs1F7q1FQw
https://haawron.tistory.com/16
https://ratsgo.github.io/machine%20learning/2017/05/30/SVM3/

Q & A

들어주셔서 감사합니다.