14기 정규세션

ToBig's 13기 이유민

SVM

Support Vector Machine

오늘 학습할 내용은?

- 머신러닝
  - 지도학습
    - 분류모델
      - KNN 모델
      - SVM
    - 예측모델
  - 비지도학습
    - 군집모델

Contents

Unit 01 | Support Vector Machine

Unit 02 | Soft Margin SVM

Unit 03 | Non-Linear SVM

Unit 04 | Summary

Contents

Unit 01 | Support Vector Machine

Unit 02 | Soft Margin SVM

Linear

Unit 03 | Non-Linear SVM

Unit 04 | Summary

Support Vector Machine

: 주로 바이너리 분류를 위해 사용하는 기법 (회귀에 사용하는 경우 : SVR)

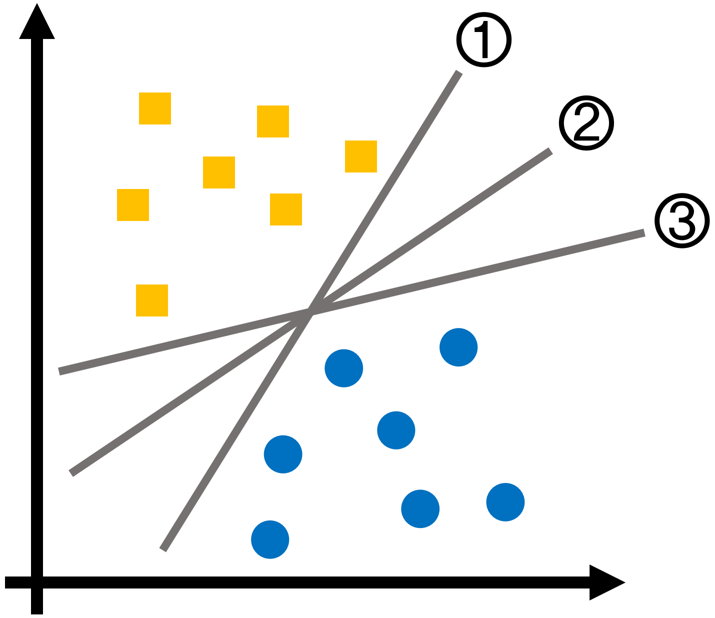
: 딥러닝 이전에 높은 성능으로 주목받은 모델

SVM의 분류

선형 여부	분류
선형	Linear svm
비선형	Non-linear svm

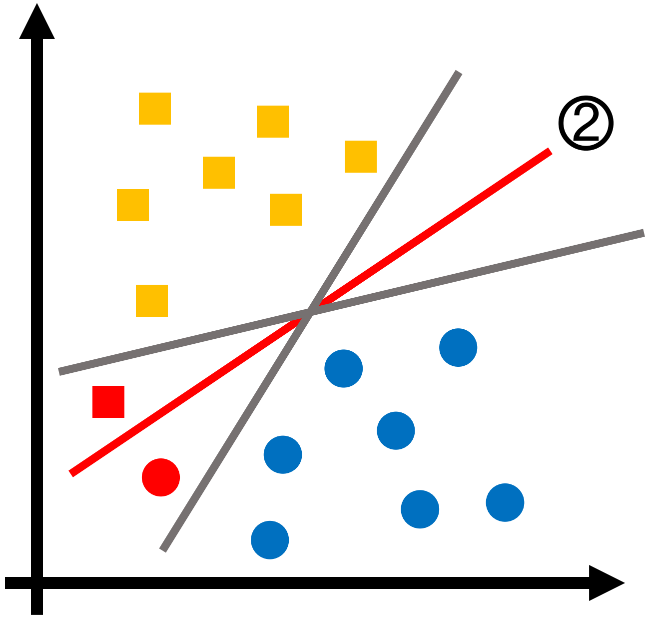
오분류 허용 여부	분류
X	Hard margin svm
O	soft margin svm

1. SVM?



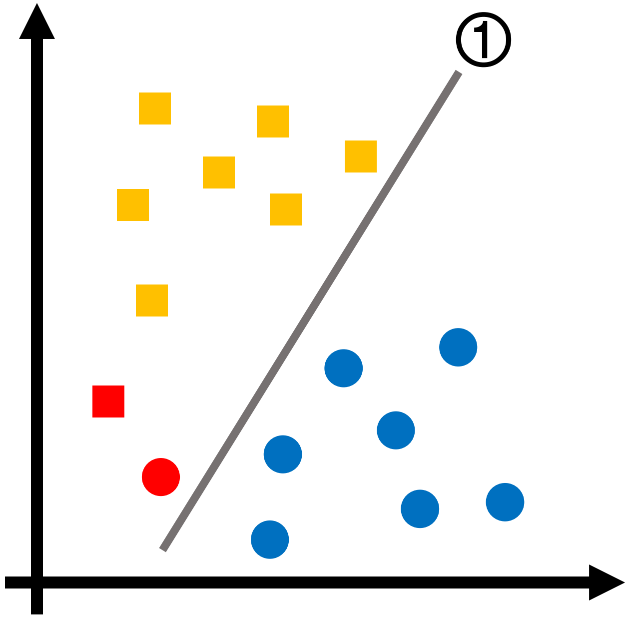
Support Vector Machine 관점에서,
데이터를 가장 잘 나누는 선은?

1. SVM?



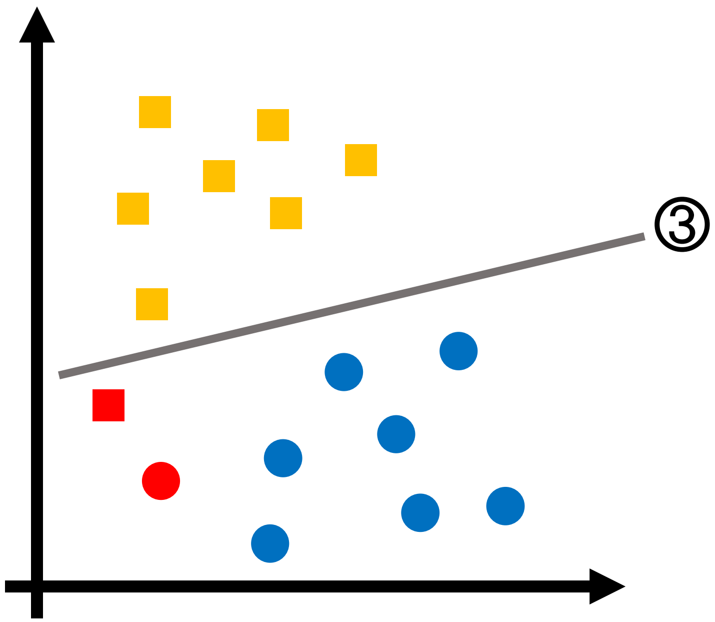
정답은 2번! 이유는?

1. SVM?



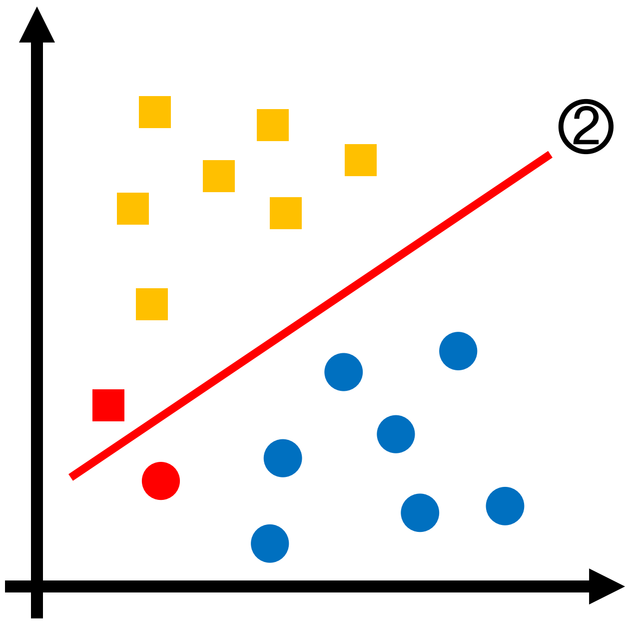
새로운 데이터가 추가되었을 때
1은 동그라미를 잘못 분류

1. SVM?



새로운 데이터가 추가되었을 때
3은 네모를 잘못 분류

1. SVM?



2번 분류기는?

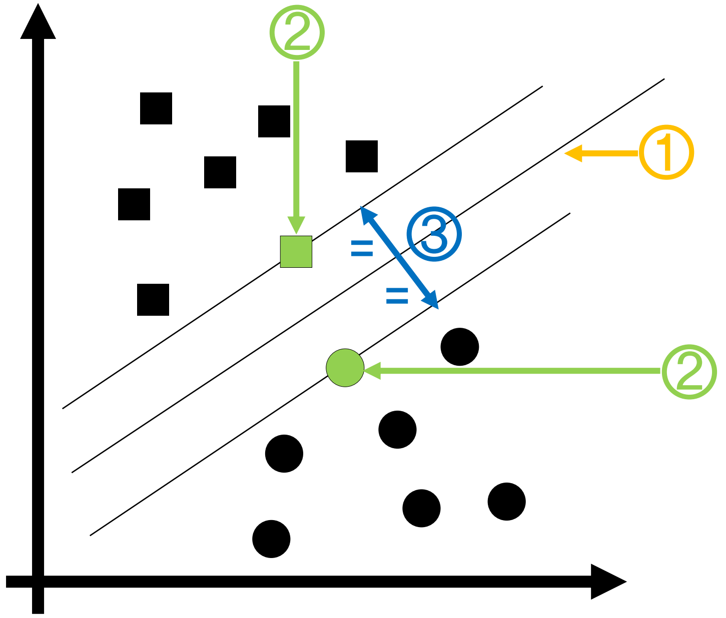
: Robust 하다 (이상치의 영향을 작게 받는다)

원래 있던 데이터들과 경계선이 충분히 떨어져 있어서 모호한 데이터도 잘 구분한다!

다시 말해,

결정경계와 데이터의 거리(여백)가 커서 좋아!

2. 정리 – 용어



① Hyperplane

: 여러 데이터를 나누는 기준이 되는 경계(초평면)

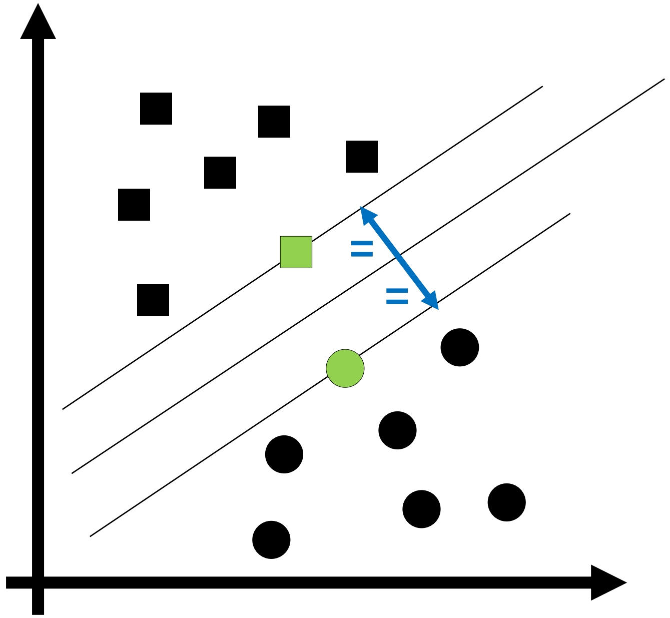
② Support Vector

: Hyperplane과 가장 가까운 '데이터'

③ Margin

: 결정경계와 서포트 벡터 사이의 거리 X 2

1. SVM – 정리 1

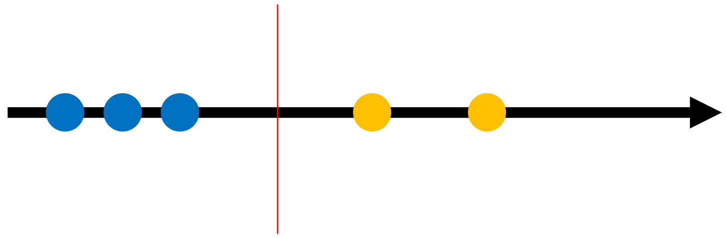


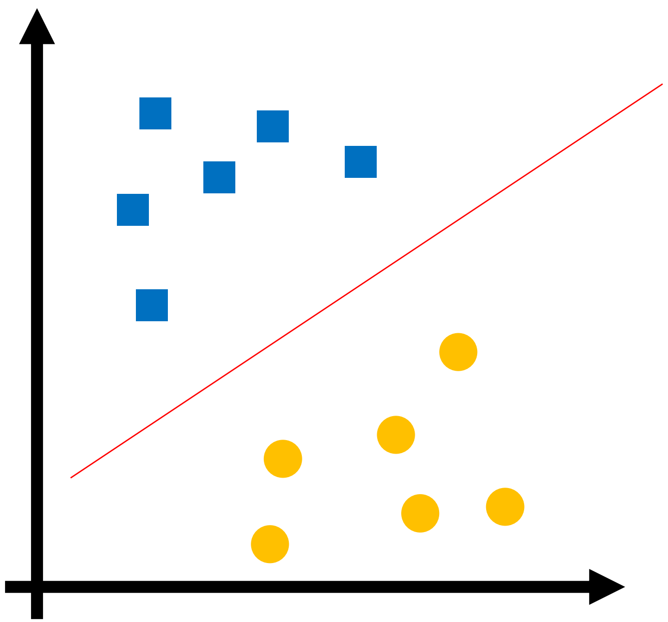
SVM의 발상?

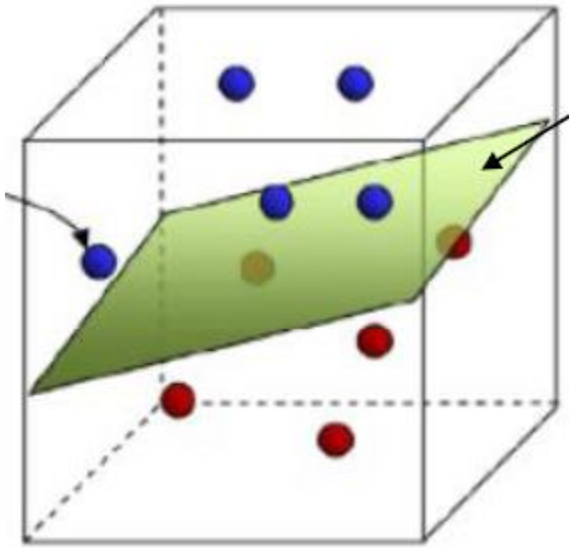
: 여백(마진)을 구하는 방법을 공식화하고
이 마진을 최대화하는 결정초평면
(decision hyperplane)을 찾자!

이제 수식으로 살펴볼까요?

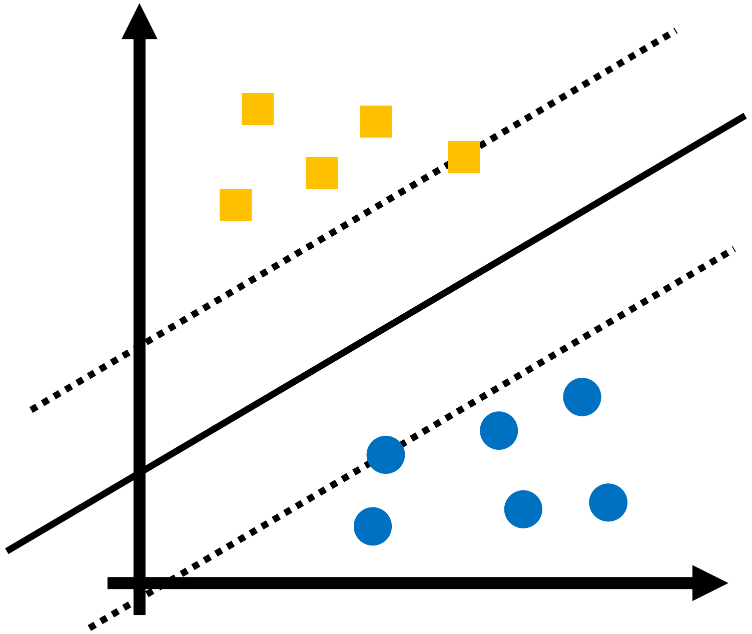
1-1. 여러 차원에서 분류되는 모습







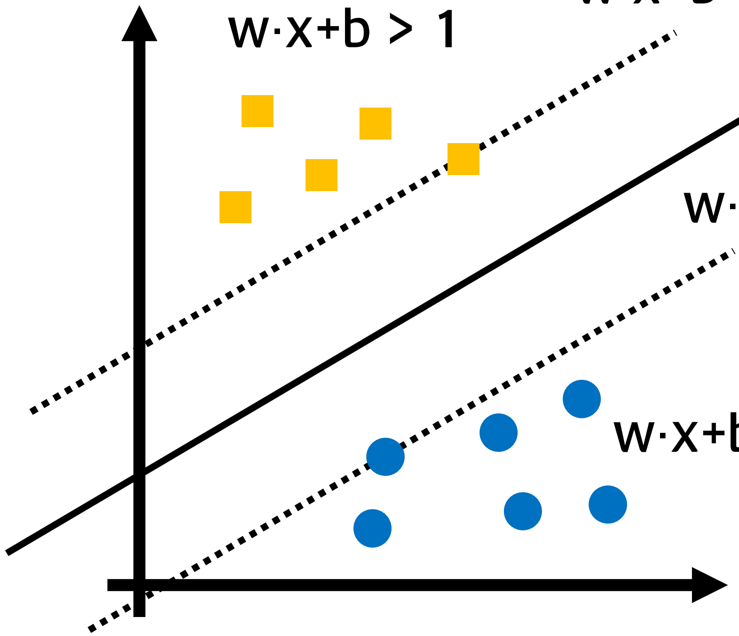
3. 수식으로 살펴보자



모든 plane은 $w \cdot x + b = 0$ 으로 표현할 수 있다!

Ex) $y = ax + b$ 라 하면,

3-1. y=±1 분류 문제



편의상, y=±1 분류 문제

: (+1) class라면 +1 이상,

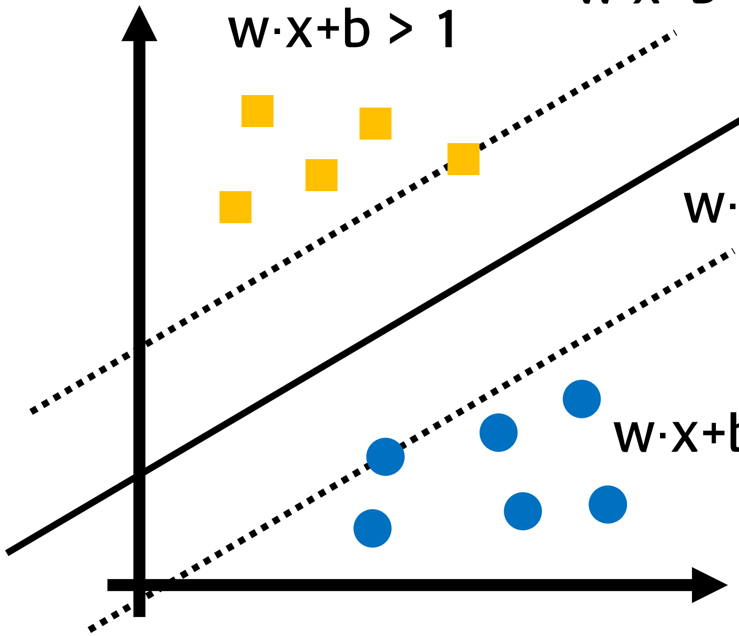
(-1) class라면 -1 이하의 값을 갖도록 하자

즉,

$$w \cdot x_+ + b \geq +1$$

$$w \cdot x_- + b \leq -1$$

3-2. 우리의 결정규칙

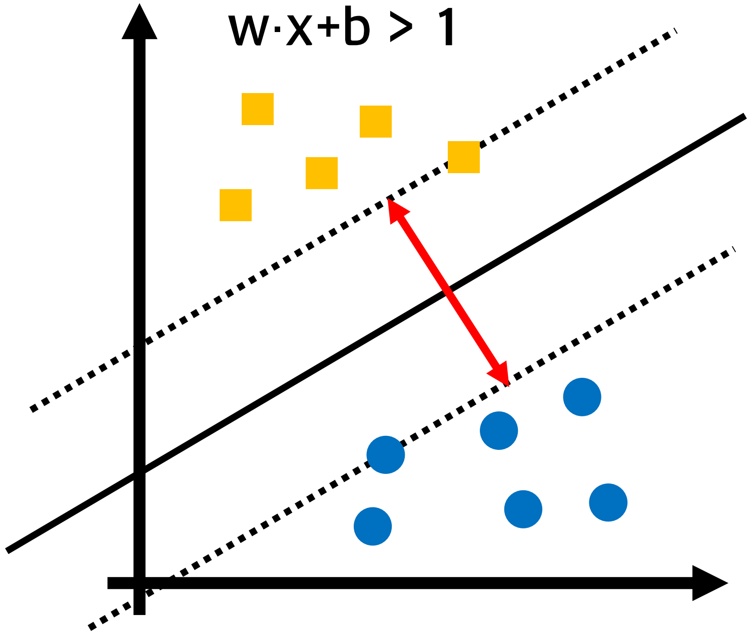


수식으로 나타내면?

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

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

3-3. margin



길의 너비(margin)

$$= (x_{+} - x_{-}) \cdot \frac{\boldsymbol{w}}{||\boldsymbol{w}||}$$

$$= \frac{\mathbf{1} - \boldsymbol{b} - (-\mathbf{1} - \boldsymbol{b})}{||\boldsymbol{w}||}$$

$$= \frac{\mathbf{2}}{||\boldsymbol{w}||}$$

지금까지의 수식을 정리하면,

1. 제약식(조건)

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

: 모든 데이터들은 결정경계 안에 잘 들어가 있어야 함

2. 목적식

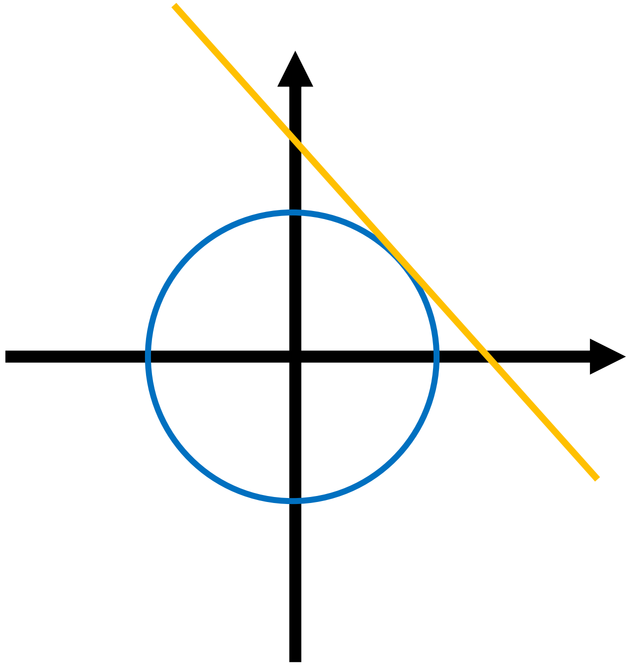
: $\frac{2}{||w||}$ 가 최대가 되게 하고, 동시에 위의 제약식을 만족하는 W와 b를 찾자 > margin이 최대

: $\max(\frac{2}{||w||})$ > $\min(||w||)$ > $\min(\frac{||w||^2}{2})$ (계산의 편의를 위한 식 변형)

3-4. 라그랑주 승수법 1

: $\phi(x,y) = 0$이라는 제약이 있을 때, 최적화 문제를 푸는 방법

> "목적식(f) + 제약식(ϕ)"을 <u>하나의 식으로 표현</u> 가능



목적식 f와 제약식 ϕ의 그라디언트 방향이 같을 때, f의 최적값

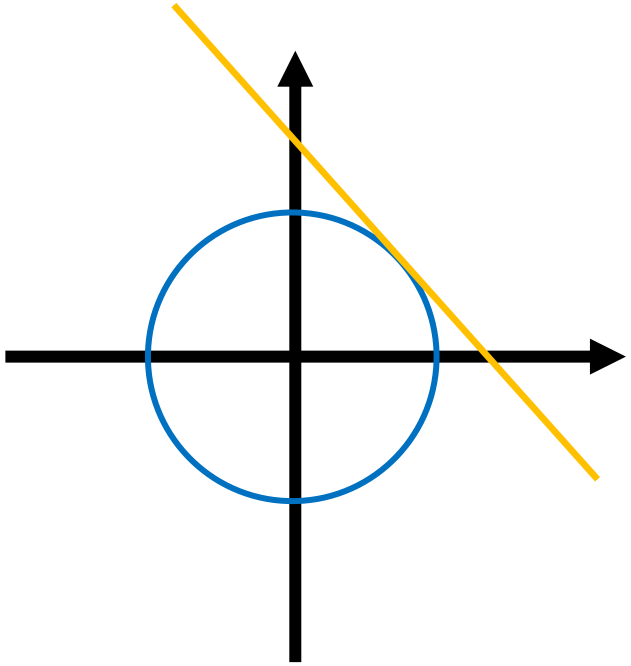
> $\nabla_{x,y} f = \lambda \nabla_{x,y} \phi$

> Let $L(x,y) = f(x,y) - \lambda\phi(x,y)$

> Then, $\nabla_{x,y,\lambda} L(x,y,\lambda) = 0$ 을 푸는 문제가 된다!

3-4. 라그랑주 승수법 2

우리의 제약식과 목적식은?



1. 제약식 : $y_i\,(w \cdot x_i + b) \geq 1$
2. 목적식 : $\frac{||w||^2}{2}$ 의 최소화

> 문제는? 제약식이 '부등식' (라그랑주 승수법은 등식)

3-5. KKT condition (Karush-Kuhn-Tucker 조건)

연립 부등식의 경우 라그랑주 승수법을 사용하되,
최적값이기 위한 필요충분조건인 KKT 조건이 붙는다.
어려우니 조건만 간단히 짚고 넘어갈게요!

1. 라그랑주 승수를 제외한 변수에 대한 편미분 값은 0이 되어야 한다.
2. 라그랑주 승수는 0보다 크거나 같아야 한다.
3. 라그랑주 승수와 제약식 중 하나는 무조건 0이 되어야 한다(즉, 둘의 곱은 항상 0).

> 조건을 전개해서 정리하자

3-6. 결론 – 의미를 중심으로!

위 식을 이용해서 정리하면

$\Sigma\alpha_i y_i = 0$ $\alpha_i \geq 0$ for all α_i 를 만족하고 동시에

$L = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j x_i^T x_j$ 를 최대화 하는 α 를 찾으면

하이퍼 플레인의 방향은 $\hat{w} = \Sigma\alpha_i y_i x_i$ 이고

분류결과는 $\mathrm{sgn}(\Sigma(\hat{\alpha}_i y_i x_i^T x_j + b)$ 이 되니까

: <u>W가 최대가 되는 w와 b를 찾는 문제는 > α를 찾는 문제가 된다 (컴퓨터가 풀기 쉬움)</u>

Contents

Unit 01 | Support Vector Machine이란?

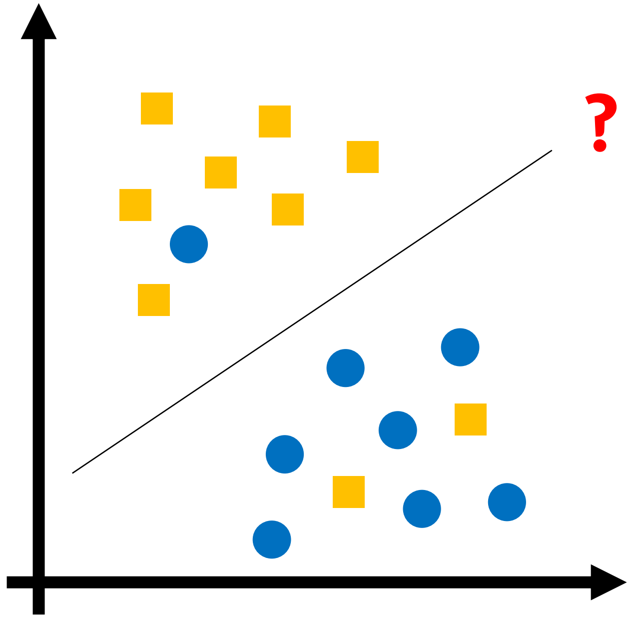
Unit 02 | Soft Margin SVM

Unit 03 | Non-Linear SVM

Unit 04 | Summary

Unit 02 | Soft Margin SVM

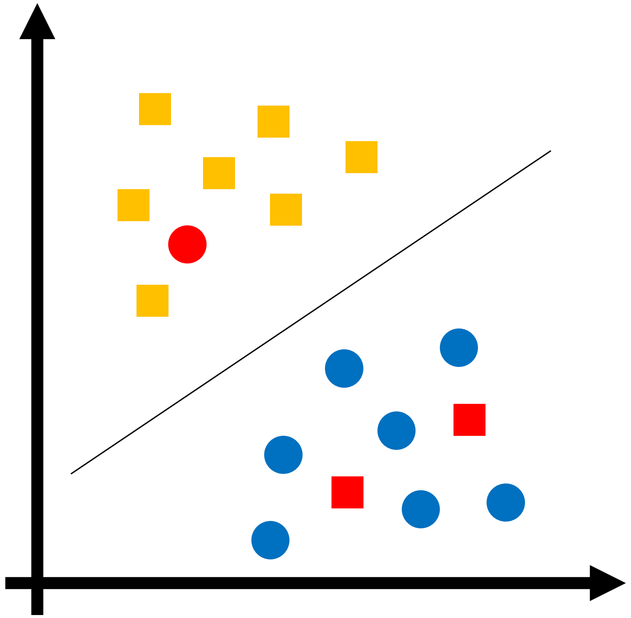
현실적으로 모든 데이터가 깔끔한 경계로 나뉘지는 않는다!
이런 애들은 어떻게 해 줘야 할까?



1. Soft Margin SVM

> 우리가 지금까지 했던 건 Hard Margin SVM
: error를 허용하지 않고 분류

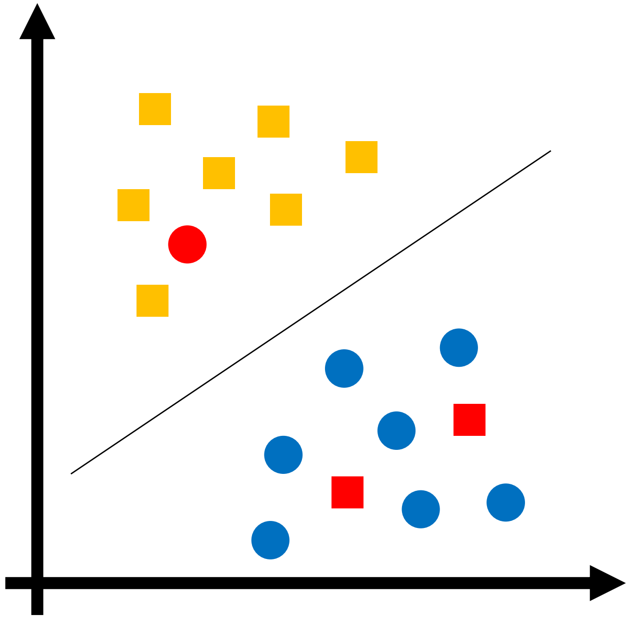
> Soft Margin SVM
: error(오분류)를 허용하되,
패널티를 줘서 전체 error를 최소화!



2. Penalty

> Penalty를 주는 방법

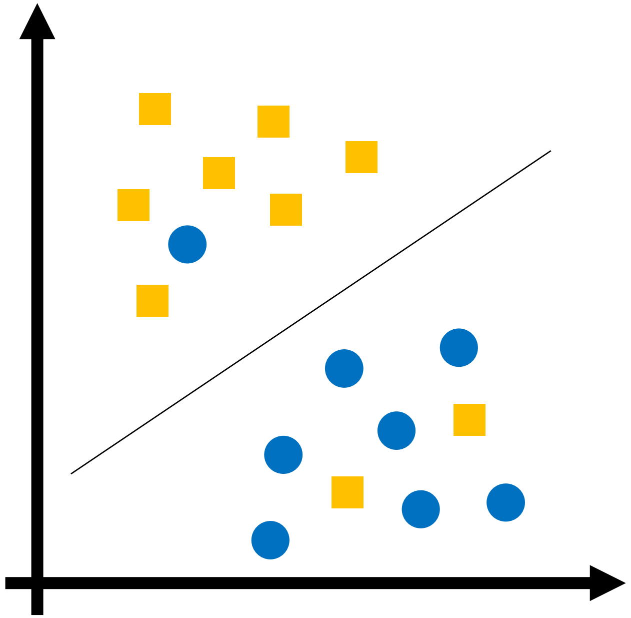
1. 0-1 Loss
2. Hinge Loss



2-1. 0-1 Loss

: error가 발생한 개수만큼 패널티 계산

: min$||w|| + C\#\text{error}$



2-2. Hinge Loss

: 오분류 정도에 따라 error의 크기를 다르게 하자

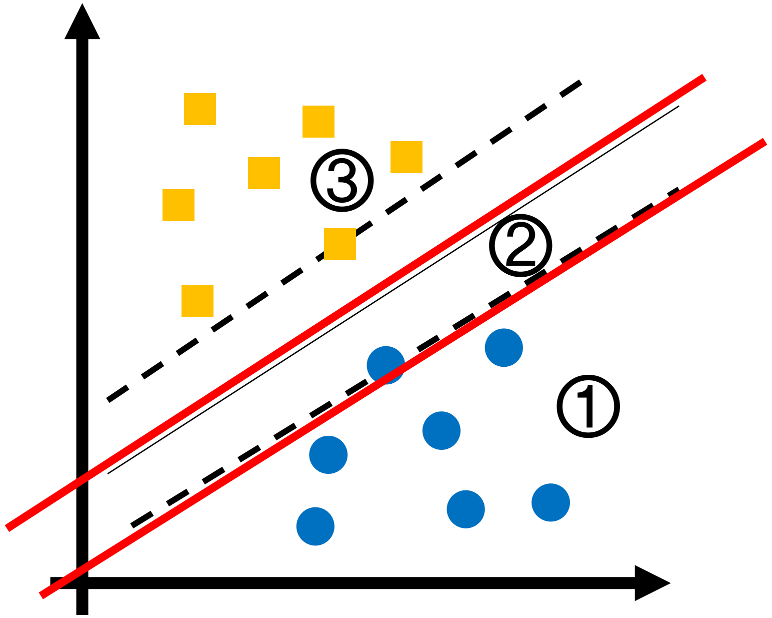
실제로는 ● 인 데이터가 분류기의 각 영역에 있을 때

① error 없음 (좋은 분류) > ξj=0

② 작은 error > $0 \leq \xi j \leq 1$

③ 큰 error (잘못된 분류) > ξj > 1

> slack variable(ξj) 사용



2-2. Hinge Loss

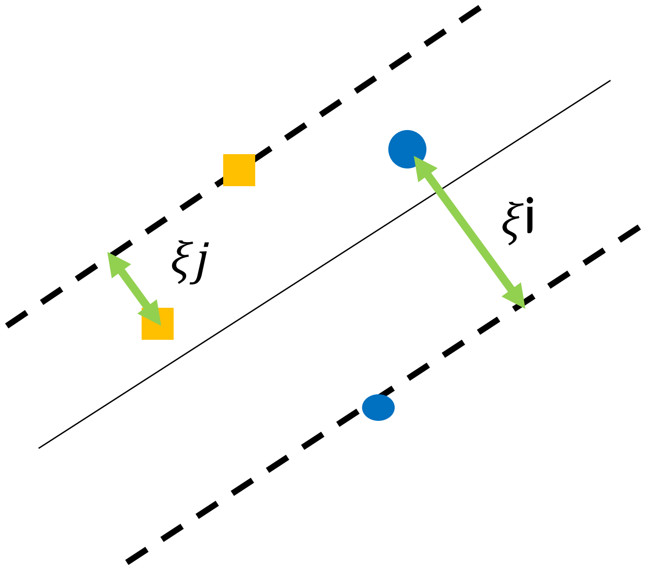
> 기존 목적함수에 error항을 추가

: argmin||w|| + $c\Sigma\xi j$

> 여기서 c는 하이퍼 파라미터 (추가자료 참고)

C가 크다면 > error를 줄이는 데에,

C가 작다면 > W를 줄이는 데에 신경써주자!



Contents

Unit 01 | Support Vector Machine이란?

Unit 02 | Soft Margin SVM

Unit 03 | Non-Linear SVM

Unit 04 | Summary

1. 구분할 수 있는 linear line이 없는데 > linear svm 불가능
2. Outlier를 무시하지 못 하는 경우에는? > soft margin svm 불가능

Unit 03 | Non-Linear SVM

1차원에서 선형 경계로 분류 불가능한 data를
> 2차원으로 매핑하면 분류 가능!

1. Kernel

: 저차원 데이터를 고차원 데이터로 매핑하는 작업

: 관측치 x들을 높은 차원으로 매핑해서 분류하자!

: SVM을 original space가 아닌 고차원의 feature space에서 학습시키면 좋지 않을까?

https://www.youtube.com/watch?v=3liCbRZPrZA 전체
https://www.youtube.com/watch?v=-Z4aojJ-pdg 9:45~10:05

1. Kernel

목적함수를 최적화하는 수식에 내적이 있다!

: $\max L(\alpha_i) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathrm{x_i}^{\mathrm{T}}\mathrm{x_j}$

=>$\max L(\alpha_i) = \Sigma\alpha_i - ½\Sigma\sigma\alpha_i\alpha_j y_i y_j \phi(\mathrm{x_i})^{\mathrm{T}}\phi(\mathrm{x_j})$

그런데 고차원으로 매핑하니 내적 부분의 차원이 증가한다

> 연산량이 너무 많지 않아?

2. Kernel Trick

> 고차원으로 매핑한 '데이터' 말고 '내적값'을 알자!
: 고차원으로 매핑하되, 연산은 간단하게 할 수 있게 된다

: $\max L(\alpha_i) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathrm{x_i}^T\mathrm{x_j}$ => $\max L(\alpha_i) = \Sigma\alpha_i - ½\Sigma\sigma\alpha_i\alpha_j y_i y_j \phi(\mathrm{x_i})^T\phi(\mathrm{x_j})$

> 매핑한 함수/데이터는 필요없다. 매핑한 내적값(최종결과)만 알면 된다!
> 그 값을 쉽게 알 수 있는 'Trick'

2. Kernel Trick

: 고차원으로 보낸 뒤 벡터의 내적 연산 == 내적을 한 후 고차원으로 보내는 연산

: 예시로 다시 보면?

2차원 벡터 $\mathbf{x}=[x_1\ \ x_2]$, $K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i^T\mathbf{x}_j)^2$ 일 때 $K(\mathbf{x}_i,\mathbf{x}_j)=\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$를 보이자

$$K(\mathbf{x}_i,\mathbf{x}_j)=(1+\mathbf{x}_i^T\mathbf{x}_j)^2 = 1+x_{i1}^2x_{j1}^2+2\,x_{i1}x_{j1}\,x_{i2}x_{j2}+x_{i2}^2x_{j2}^2+2x_{i1}x_{j1}+2x_{i2}x_{j2}=$$

$$=[1\ \ x_{i1}^2\ \ \sqrt{2}\,x_{i1}x_{i2}\ \ x_{i2}^2\ \ \sqrt{2}x_{i1}\ \ \sqrt{2}x_{i2}]^T[1\ \ x_{j1}^2\ \ \sqrt{2}\,x_{j1}x_{j2}\ \ x_{j2}^2\ \ \sqrt{2}x_{j1}\ \ \sqrt{2}x_{j2}]$$

$$=\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) \quad \text{where } \varphi(\mathbf{x})=[1\ \ x_1^2\ \ \sqrt{2}\,x_1x_2\ \ x_2^2\ \ \sqrt{2}x_1\ \ \sqrt{2}x_2]$$

2. Kernel Trick

> kernel을 쓰는 이유?
: 고차원으로 매핑도 하고, 연산도 간단하게 할 수 있으니까!

> Kernel Trick을 만족하는 조건도 있지만, 복잡하니 생략하고 자주 쓰는 kernel만 알자

3. 자주 쓰는 Kernel

Linear	$K(x_i,x_j)= x_i{}^T x_j$
quadratic	$K(x_i,x_j)= (1+ x_i{}^T x_j)^2$
Polynomial of power p	$K(x_i,x_j)= (1+ x_i{}^T x_j)^p$
Sigmoid	$K(x_i,x_j)= \tanh(\beta_0 x_i{}^T x_j + \beta_1)$
Gaussian (radial-basis function network)	$K(\mathbf{x}_i,\mathbf{x}_j) = \exp\left(-\frac{\lVert \mathbf{x}_i-\mathbf{x}_j \rVert^2}{\sigma^2}\right)$

: 가장 많이 사용하는 kernel은 가우시안 커널! (RBF kernel)

:각 함수마다 파라미터가 조금씩 다름!

4. RBF kernel (= 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, 테일러 급수에 의해

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

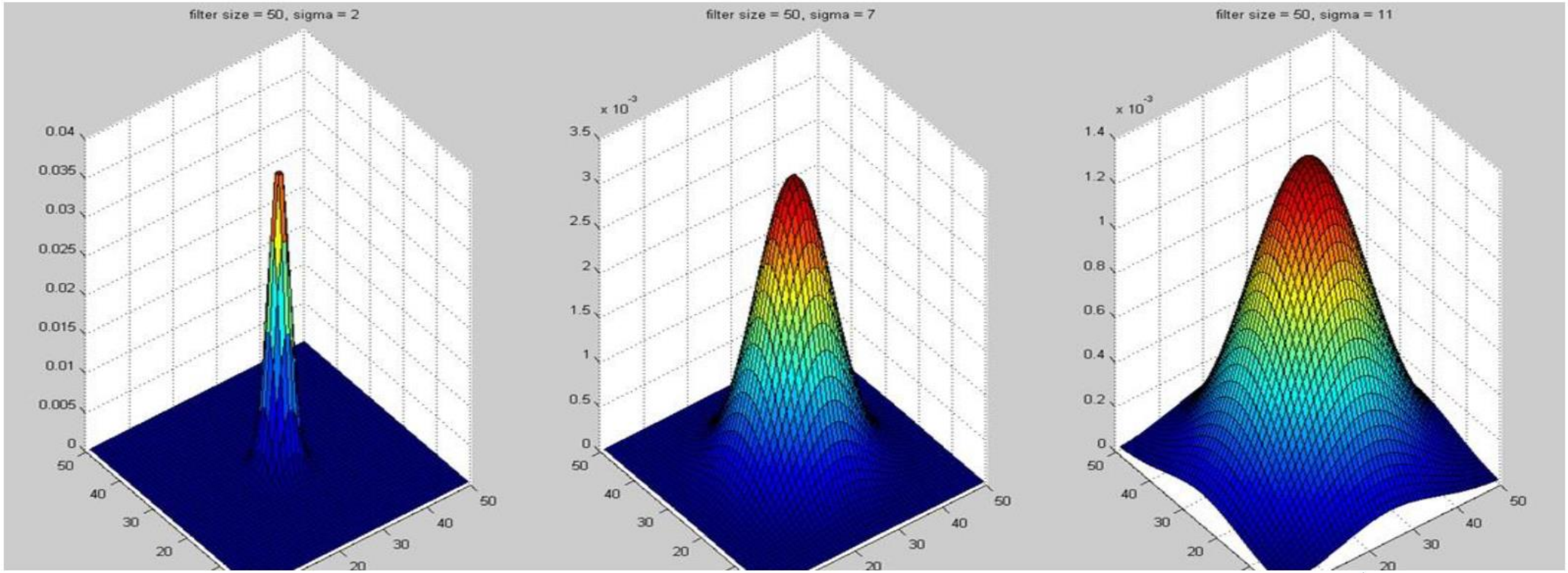
> 무한대 차원의 Feature Space로 매핑!

4-1. RBF 커널 - Gamma Parameter Tuning

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

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

4-1. RBF 커널 – Gamma Parameter Tuning



Gamma 감소 = 표준편차 증가

4-1. RBF 커널 - Gamma Parameter Tuning

: 감마가 클수록

훨씬 인접한 것들만 같은 영역으로 본다

= 엄청 인접하지 않으면 엄청 먼 곳으로 인식한다

= 원래 차원으로 돌아왔을 때 경계가 아주 촘촘하다

= Hyper plane이 훨씬 더 굴곡지다

= Ovefitting의 가능성이 높다 !

Contents

Unit 01 | Support Vector Machine이란?

Unit 02 | Soft Margin SVM

Unit 03 | Non-Linear SVM

Unit 04 | Summary

1. Hard Margin과 Soft Margin

: Hard Margin : 에러를 허용하지 않음 (현실적이지 못함)
: Soft Margin : 에러를 허용

: 에러를 허용하는 방법은 0-1 Loss 와 Hinge Loss
: 에러와 마진 둘 중 무엇을 줄일 것인지를 결정하는 hyper parameter는 C

2. Linear SVM과 Non-linear SVM

> Linear SVM

: 선형으로 분류

: Feature가 많아 kernel을 하게 되면 차원이 높아지고 연산량이 많아지니 Linear가 효과적.

> Non-Linear SVM

: Feature가 많지 않을 땐 그걸로 분류하기 쉽지 않기 때문에 차원을 올려줌

: 이 때 차원을 올려주는 방법에 따라 다양한 parameter가 존재하지만
 가장 많이 쓰이는 건 Gaussian Kernel에서 Gamma

3. 정리

1) SVM 알고리즘 중 일반적으로 널리 사용되는 건 RBF 커널SVM
2) 좋은 성능을 얻으려면 매개변수인 C와 gamma를 잘 조정해줘야 함
3) C는 데이터 샘플들이 다른 클래스에 놓이는 것을 허용하는 정도를, gamma는 결정경계의 곡률을 결정
4) 두 값 모두 커질수록 알고리즘의 복잡도는 증가
5) 일반적으로 grid search로 경험적으로 최적의 매개변수 값들을 찾아가는데, 당연하지만 내용을 어느정도 숙지하고 있다면 훨씬 더 빠르게 좋은 성능을 내는 매개변수 값들을 찾아낼 수 있을 것!

Unit 04 | Summary

-Multi Class SVM

1. One VS one

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

N개의 클래스에 대해 서로서로 Classifier를 가지고 있어야 하기 때문에 $\frac{n(n-1)}{2}$ 개의 Classifier가 필요.

2. One vs rest

클래스가 N개 있으면 모든 Class에 대해 1 : N-1로 binary 분류하여 이 클래스가 맞는지 아닌지를 판단하고 투표로 결정한다.

이때 N 개의 Classifier가 필요하다.

Reference

1. 투빅스 12기 박진혁님, 11기 심은선님, 10기 박규리님 svm 강의자료
2. 투빅스 13기 이지용님 Optimization 강의자료
3. 서울시립대학교 박창이 교수님 SVM 강의
4. 고려대학교 김성범 교수님 SVM 강의
5. KAIST 문일철 교수님 SVM 강의
6. Lec. 16 Learning: Support Vector Machines, Patrick Winston MIT OCW 6.034 Fall 2010
7. 앤드류응(Andrew Ng) 교수님의Machine Learning (명)강의
   https://www.coursera.org/learn/machine-learning#syllabus #강추

과제 설명

Assignment 1.

참고자료

참고자료 1. Grid Search의 활용

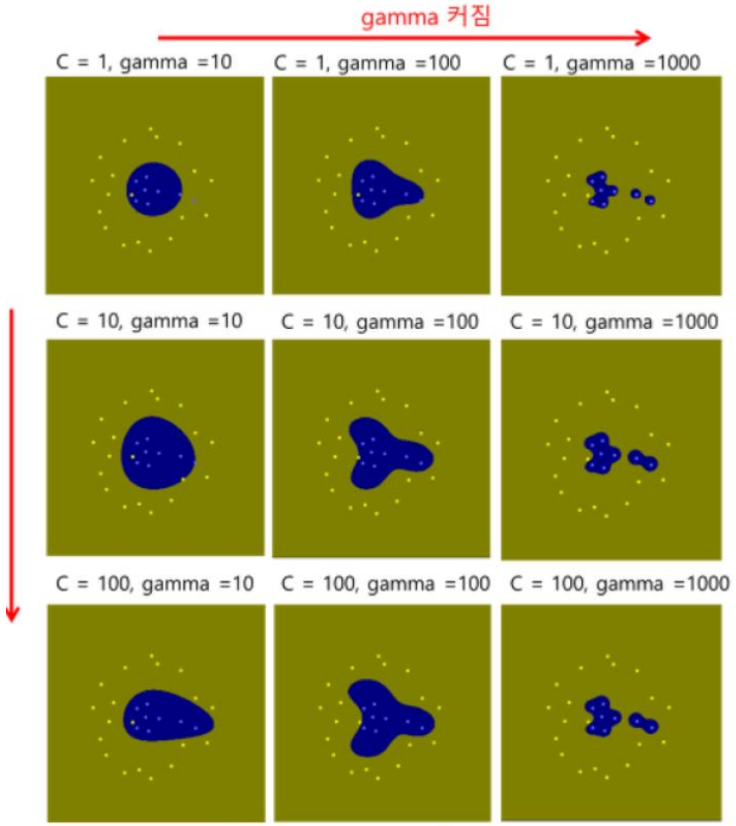


그림9. C와 gamma의 영향

하이퍼 파라미터가 두 개나 있어서 감이 안 올 때는?

C : 데이터가 다른 클래스에 놓이는 걸 허용하는 정도

γ : 결정경계의 곡률을 결정 (kernel모델의 파라미터)

> Overfitting 위험 : C가 크고 Gamma가 클 때

> underfitting 위험 : C가 작고 Gamma가 작을 때

참고자료

참고자료 2.

앤드류응: Andrew Ng 교수님께서는

1) 피쳐가 데이터 개수에 비해 많으면 커널 없는 SVM 즉, linear kernel을 쓰고(피쳐 개수10000개정도)

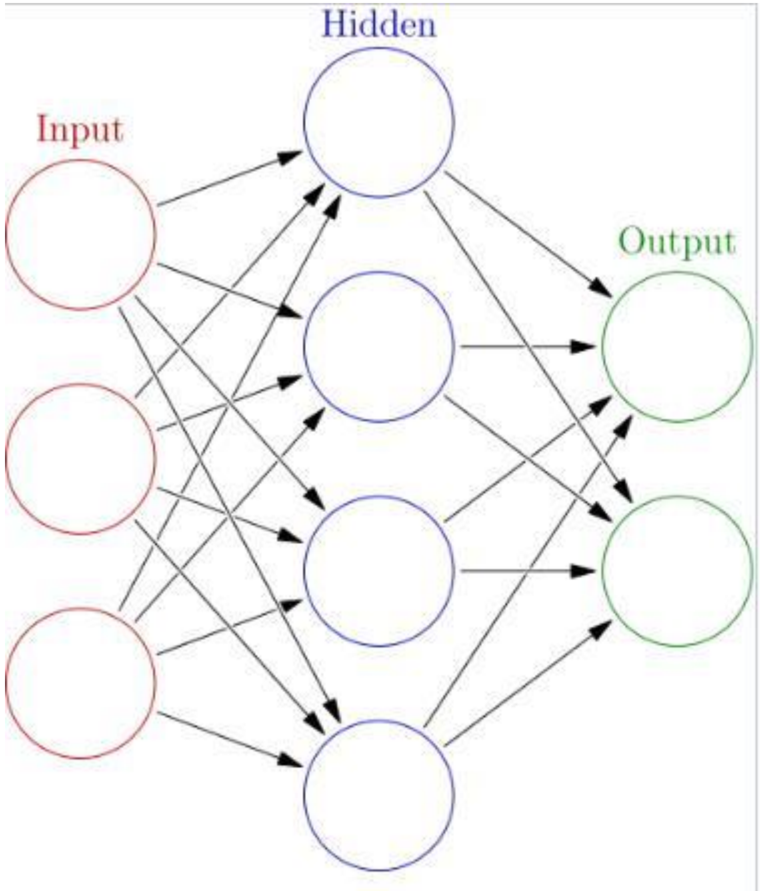
2) 피쳐가 적고 데이터 개수는 보통이면 가우시안 커널을 적용한 SVM 즉, RBF커널 SVM을 쓰라고 하셨고(피쳐 개수 1000개정도, 데이터개수10 -10000개정도)

3) 피쳐가 적고 데이터가 많으면 커널 없는 SVM이나 로지스틱 회귀를 쓰라고 하셨으나, 이 세번째 부분은 커널 있는 SVM은 데이터가 많아지면 연산이 많아져 부담이 되기 때문이라고 하셨다.

따라서 연산의 부담이 좀 많아도 괜찮으면 RBF 커널도 무방할 듯 하다

(지극히 교수님의 경험에 의한 개인적인 생각!)

참고자료 3. SVM과 Neural Network



Hidden layer를 여러 개로 만들어 차원을 확장시키고
Activation Function을 사용하면(Non-linear)기본적으로
SVM kernel과 원리는 같다.

대부분 NN이 성능이 더 좋기 때문에 요즘은 NN이 주로 사용 된다.
또한 Neural Network와 학습 방법이 유사한 * SGD Classifier를
이용해서 값을 찾는 방법이 있는데 이는 사이킷런에서 라이브러리
를 제공한다.

그럼에도 불구하고 SVM이 조금 더 직관적인 해석이 가능하며
좀 더 알고리즘 최적화도 효율적이므로 쓰이는 경우가 있다!

참고자료

참고자료 4. 사이킷런 SVM parameter

1. Kernel

Decision Boundary의 모양 결정

Kernel 선택 가능 (Linear, Polynomial, Sigmoid, RBF 등)

2. C

: Decision Boundary 일반화 VS training data의 정확한 분류 사이의 trade-off 조정

: C가 크면 정확하게 구분하는 데에, 작으면 smooth한 결정경계를 만드는 데에 초점을 맞춤

: C 있음: soft / C 없음: hard margin svm

3. Margin

: 결정경계의 굴곡에 영향을 주는 데이터의 범위(reach)를 정의

Q & A

들어주셔서 감사합니다.