Supervised Learning

Aug 04 2022/ 김예진

0. 목차

0. 개괄

지도 학습이란?

학습

입력값 INPUT

특징(feature)
독립 변수(independent variable)

신경망 모델
통계 모형
확률 모형

출력값 OUTPUT

레이블(label)
종속 변수(dependent variable)
타겟(target)

사전에 입력데이터와 출력 데이터가 **이미 존재하는** 환경에서,

주어진 입력값을 가지고 일종의 변환(혹은 모델)을 거쳐

최종 출력값을 **추정하도록 학습(fitting!)**

지도 학습이란?

입력값 INPUT

특징(feature)
독립 변수(independent variable)

예측

신경망 모델
통계 모형
확률 모형

?

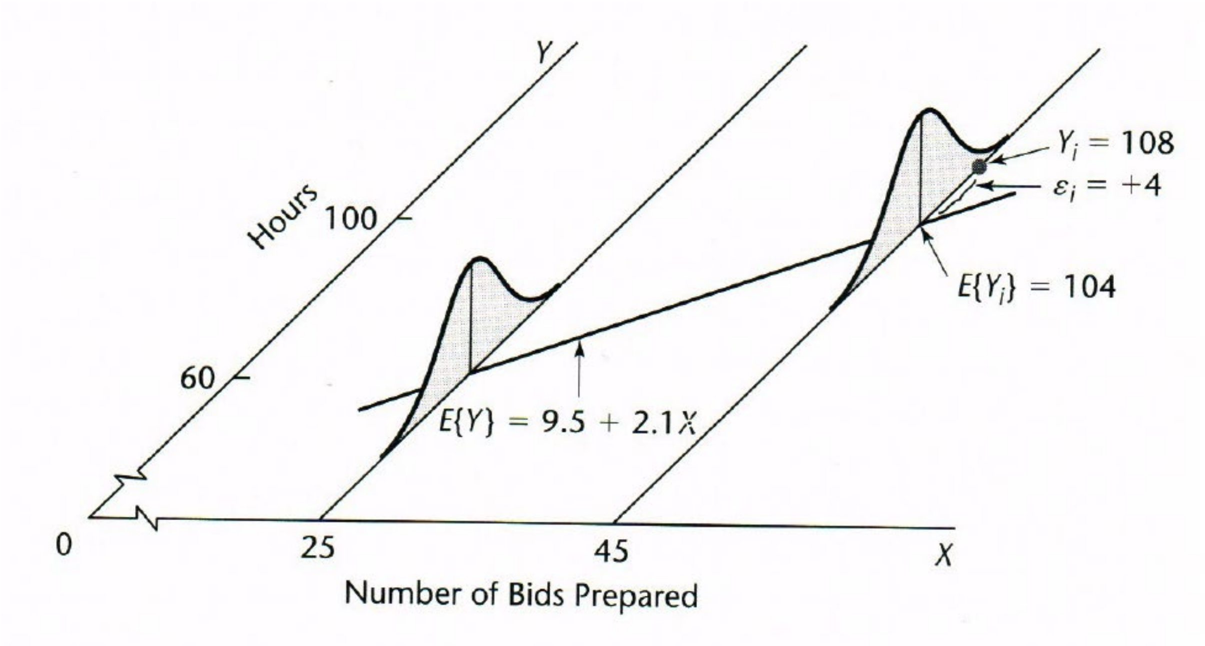
새로운 입력 데이터만 가지고 있는 환경에서,

새로운 입력값에 일종의 변환(혹은 모델)을 거쳐

최종 출력값을 **예측(predicting!)**

1. Basic Model

Linear Regression



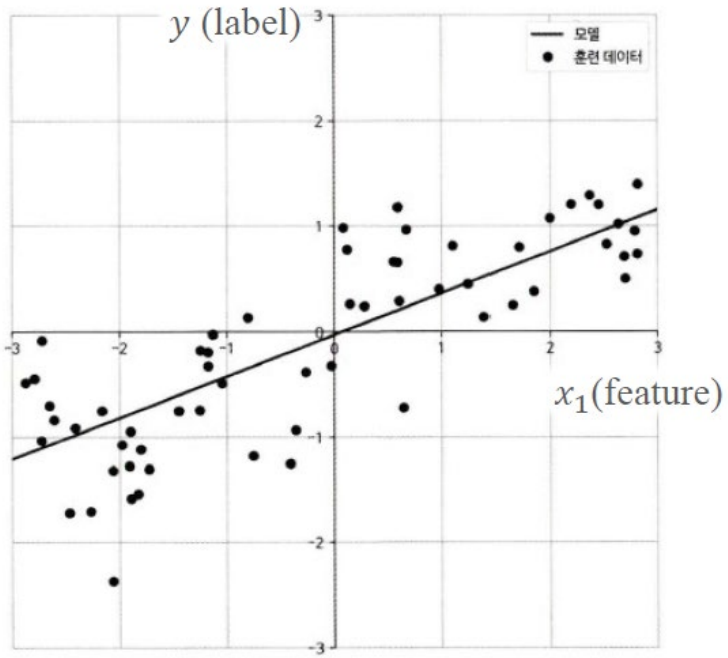
Model equation form

$$y_i = \beta_0 + \beta_1 x_{1,i} + \epsilon_i$$

$$\epsilon_i : iid$$

1. Basic Model

Linear Regression



$$\hat{y} = \beta_0 + \beta_1 x_1$$

입력 데이터가 1개인 경우,

Y절편이 β_0, 기울기가 β_1인 **1차 함수**로

출력 데이터를 **선형 관계**로 표현할 수 있습니다.

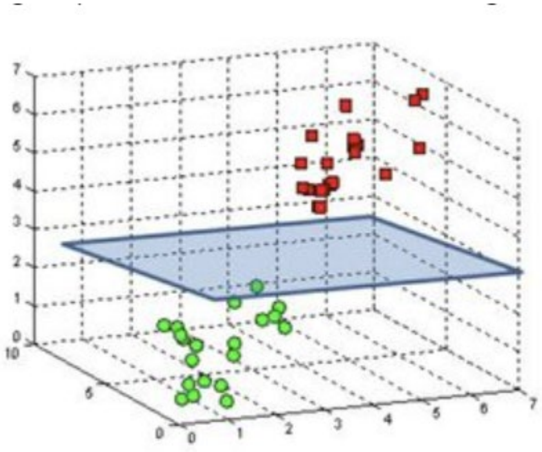
이를 시각화한다면,

X축에 입력 데이터가, Y축에 출력 데이터일 때

데이터의 **산점도**와 **모델**을 확인할 수 있습니다.

Linear Regression

초평면(hyperplane) 예시



$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k = \boldsymbol{\beta}^T \boldsymbol{X}$$

입력 데이터가 k개인 경우,

출력 데이터를 **초평면(hyperplane)**으로 표현할 수 있습니다.

즉,

입력데이터가 많을 수록 그 데이터를 표현할 수 있는 값들($\boldsymbol{\beta}$)가 많아지기에,

적절히 예측하는 모델을 만들 수 있습니다.

https://www.quora.com/Support-Vector-Machines-What-is-an-intuitive-explanation-of-hyperplane

1. Basic Model

Linear Regression

입력데이터가 많을 수록 그 데이터를 표현할 수 있는 값들 = $\boldsymbol{\beta}$

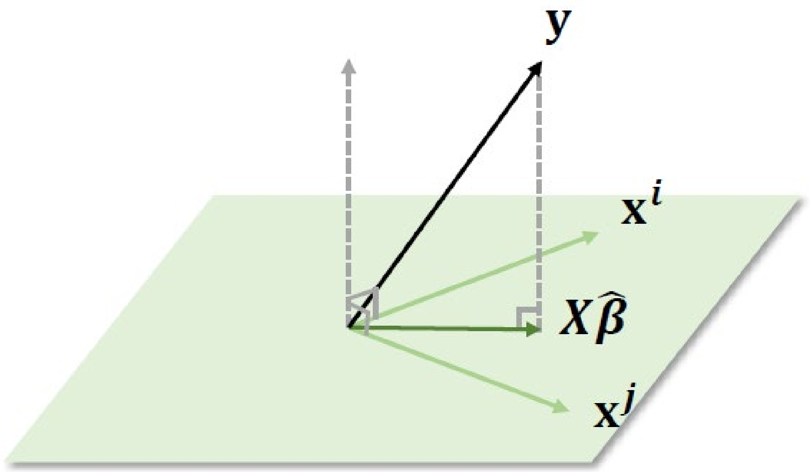
Parameter!

어떻게 beta를 구할 수 있을까요?

최소 자승법 / 특이값 분해 / 경사 하강법

Simple Linear Regression

최소자승법(Ordinary Least Squares Method, OLS)



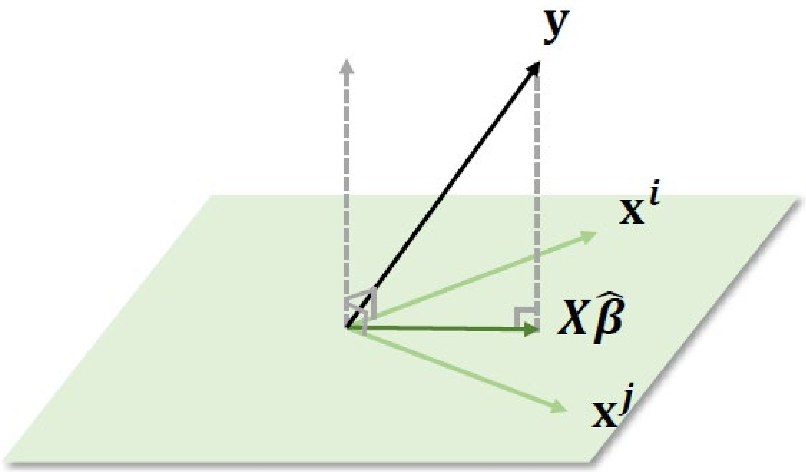
$$
\begin{aligned}
Q(\boldsymbol{\beta}) &= \sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 X_{i1} - \ldots - \beta_{p-1}X_{i,p-1})^2 \\
&= (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{b} &= \hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}
\end{aligned}
$$

1. Basic Model

Simple Linear Regression

최소자승법(Ordinary Least Squares Method, OLS)



$$
\begin{aligned}
Q(\boldsymbol{\beta}) &= \sum_{i=1}^{n} (Y_i - \beta_0 - \beta_1 X_{i1} - \ldots - \beta_{p-1} X_{i,p-1})^2 \\
&= (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{b} &= \hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \\
&= \boxed{(\mathbf{X}^T\mathbf{X})^{-1}} \mathbf{X}^T \mathbf{Y}
\end{aligned}
$$

1. Basic Model

Simple Linear Regression

특이값 분해(Singular Value Decomposition, SVD)

$X = U\Sigma V^T$라고 할 때

$X^+ = V\Sigma^{-1}U^T$에 대해서

$$b = \hat{\beta} = X^+Y$$

역행렬을 구할 수 없을 때 근사적으로 구할 수 있는 방식으로,

여러 패키지(scikit-learn 등)에서 쉽게 구할 수 있다!

1. Basic Model

MSE Loss

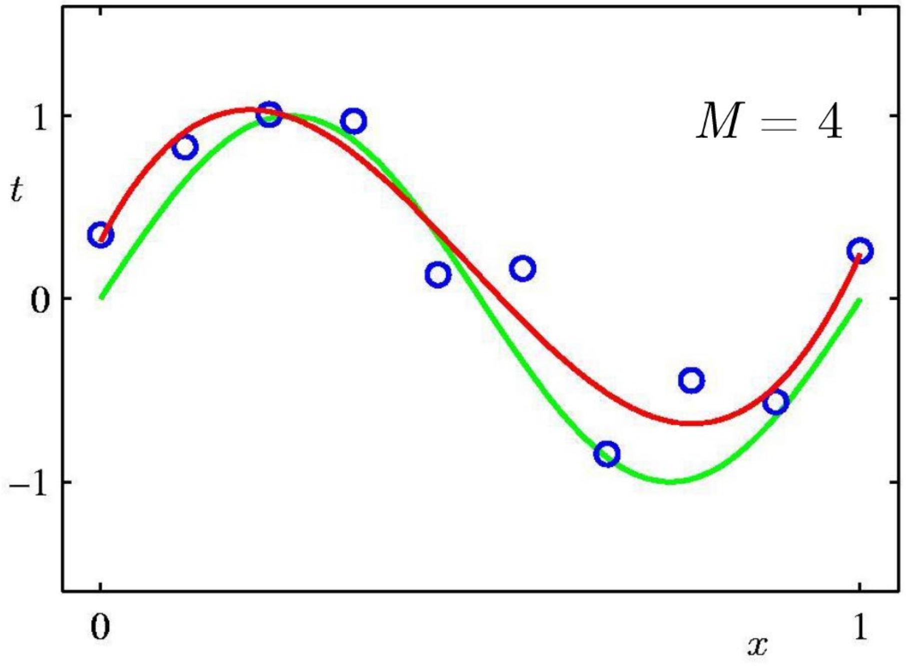
Mean Squared Error

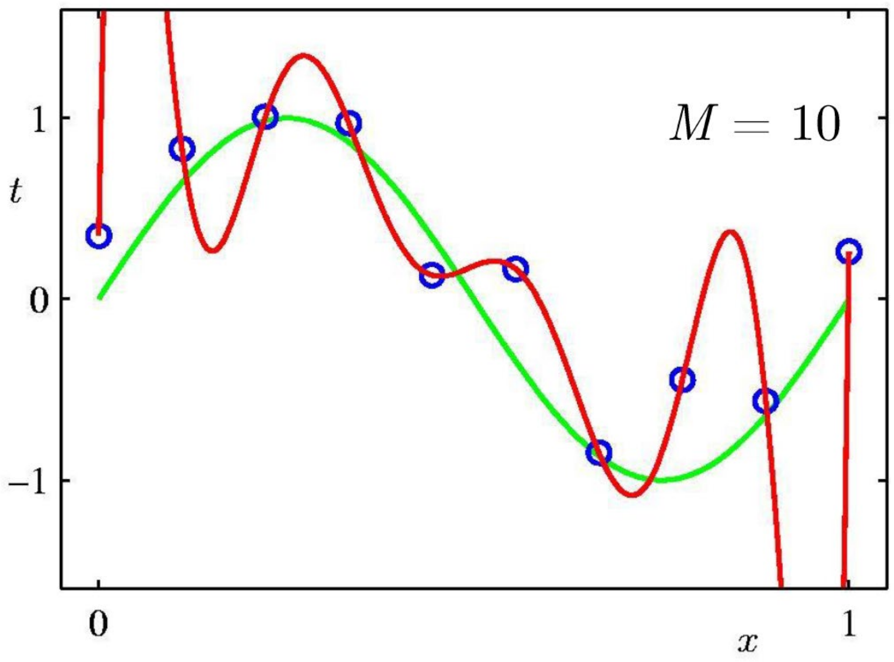
$$MSE = \frac{1}{N}\sum_{i}^{N}(y_i - \hat{y}_i)^2$$

실제 레이블 값(y_i)와 추정(예측) 값($\hat{y}_i$)가 얼마다 다른 지 알려주는 수치

2. Generalization

Underfit & Overfit





어떻게 **일반적**으로 좋은 모델을 만들 수 있을까요?

Underfit & Overfit

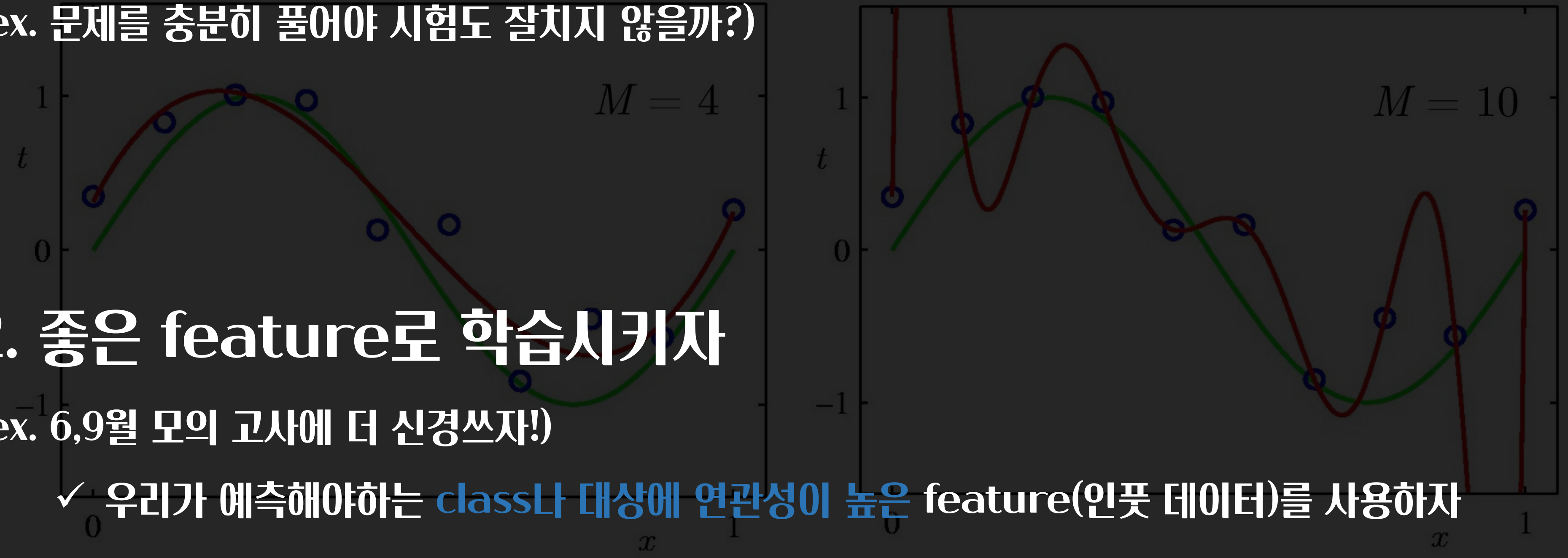
1. 많은 데이터를 확보하자!

(ex. 문제를 충분히 풀어야 시험도 잘치지 않을까?)

2. 좋은 feature로 학습시키자

(ex. 6,9월 모의 고사에 더 신경쓰자!)

✓ 우리가 예측해야하는 class나 대상에 연관성이 높은 feature(인풋 데이터)를 사용하자

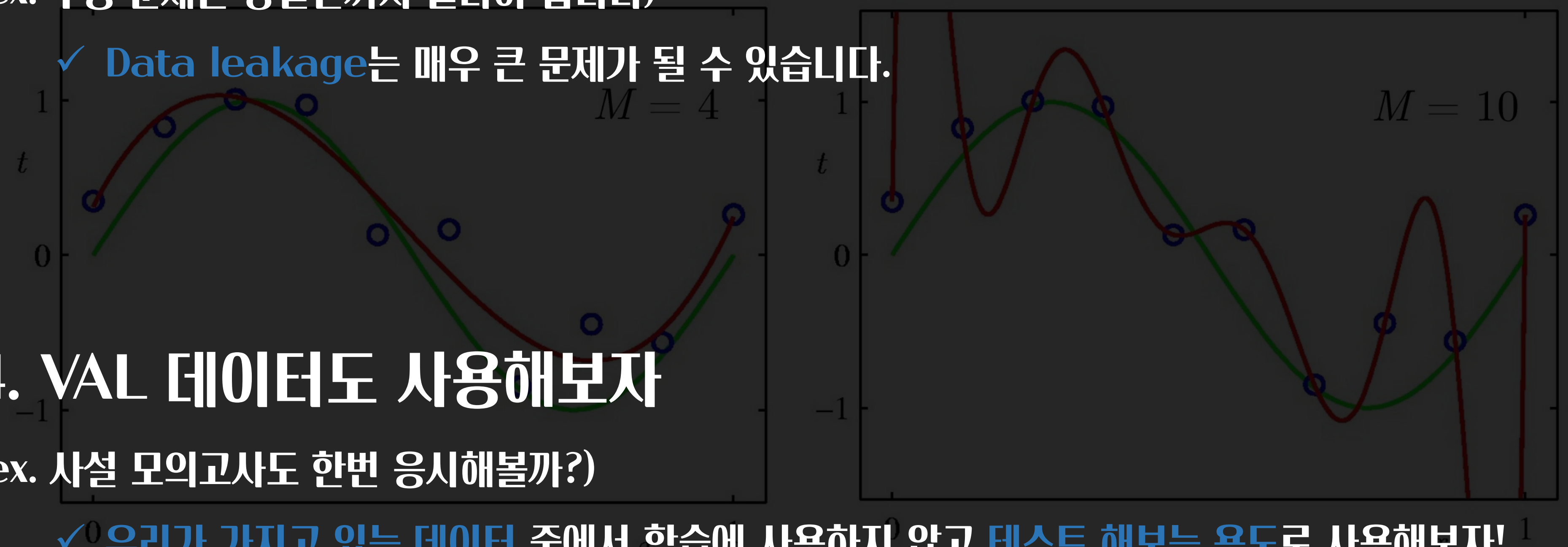


어떻게 **일반적**으로 좋은 모델을 만들 수 있을까요?

3. TEST 데이터와 TRAIN 데이터는 구분하자!

(ex. 수능 문제는 당일전까지 몰라야 합니다)

✓ Data leakage는 매우 큰 문제가 될 수 있습니다.



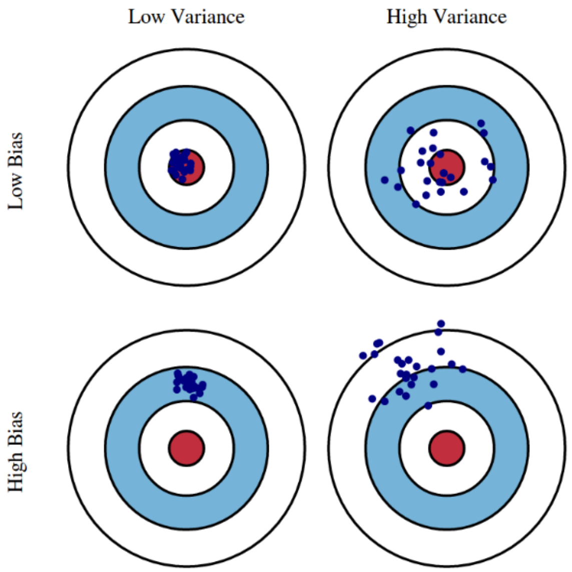
4. VAL 데이터도 사용해보자

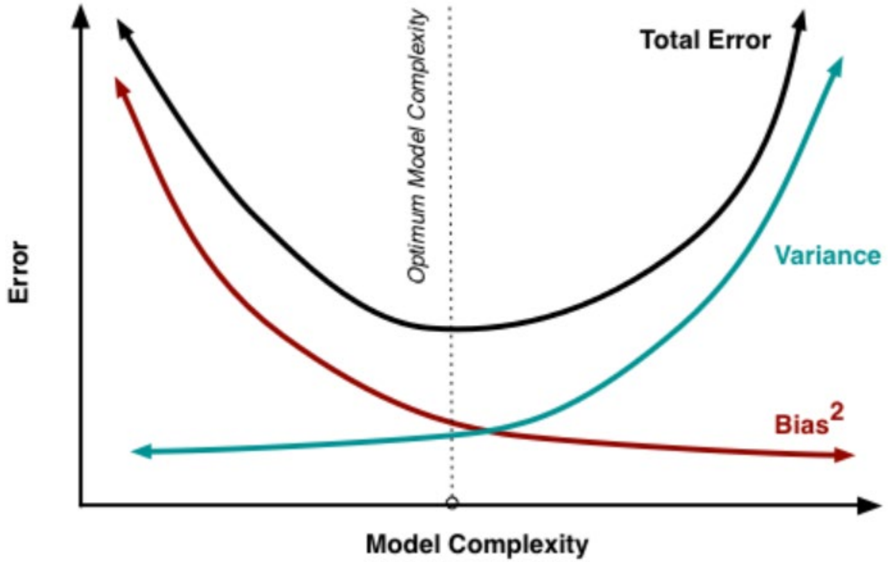
(ex. 사설 모의고사도 한번 응시해볼까?)

✓ 우리가 가지고 있는 데이터 중에서 학습에 사용하지 않고 테스트 해보는 용도로 사용해보자!

어떻게 **일반적**으로 좋은 모델을 만들 수 있을까요?

Bias-Variance Trade-off





Bias와 Variance는 무엇에 대한 것일까?

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

2. Generalization

Bias-Variance Trade-off

$$\underbrace{E_{\mathbf{x},y,D}\left[\left(h_D(\mathbf{x})-y\right)^2\right]}_{\text{Expected Test Error}} = \underbrace{E_{\mathbf{x},D}\left[\left(h_D(\mathbf{x})-\bar{h}(\mathbf{x})\right)^2\right]}_{\text{Variance}} + \underbrace{E_{\mathbf{x},y}\left[\left(\bar{y}(\mathbf{x})-y\right)^2\right]}_{\text{Noise}} + \underbrace{E_{\mathbf{x}}\left[\left(\bar{h}(\mathbf{x})-\bar{y}(\mathbf{x})\right)^2\right]}_{\text{Bias}^2}$$

데이터 셋에 따라 달라지는 모델 : $h_D(X)$

데이터 셋에 대해 평균적인 모델의 예측값 : $\bar{h}_D(X)$

인풋 데이터에 대해 평균적인 라벨값 : $\bar{y}$

Bias: 평균적인 모델 예측값과 평균적인 라벨들 간의 차이

Variance: 어떤 데이터 셋 구성하는지에 따라 달라지는 모델들의 예측값의 변동성

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html

Lasso & Ridge

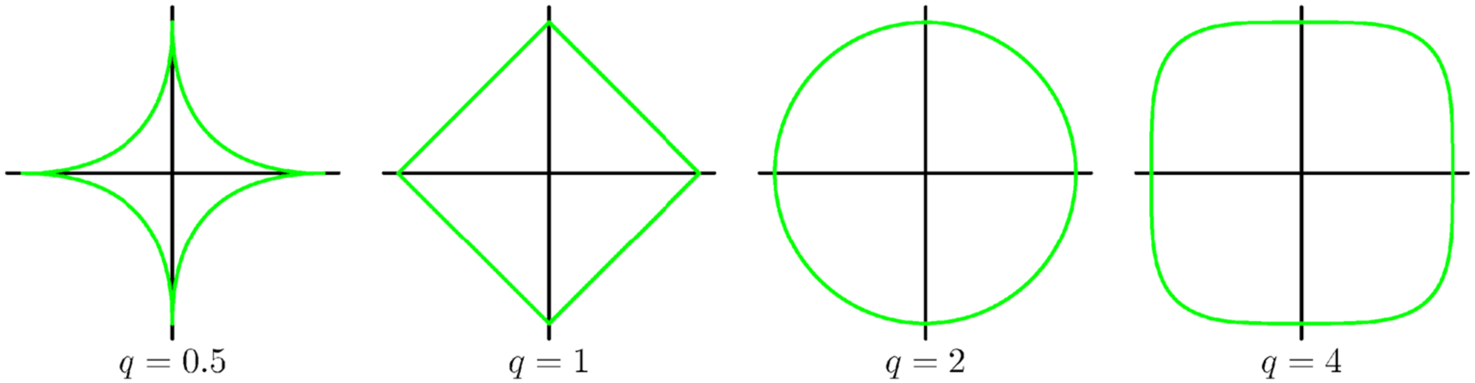
q=1 : Lasso Regression
q=2 : Ridge Regression

$$\sum_{i}^{N}(y_i - \hat{y}_i)^2 + \lambda \sum_{j}^{k} \|\beta_j\|^q$$

핵심은 Parameter의 크기를 적절한 수준으로 조절하는 항이 추가가 되며,

이러한 방식을 **Regularization**이라고 합니다!

Lasso & Ridge



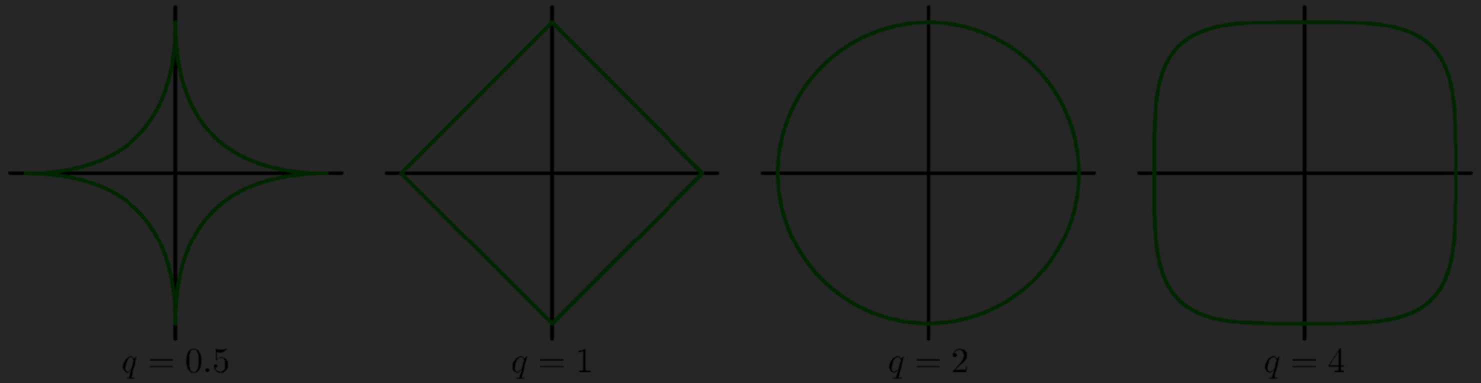
Lasso
“L1 regularization”

Quadratic
“L2 regularization”

추가 설명판:

3. Advanced Model

Lasso & Ridge

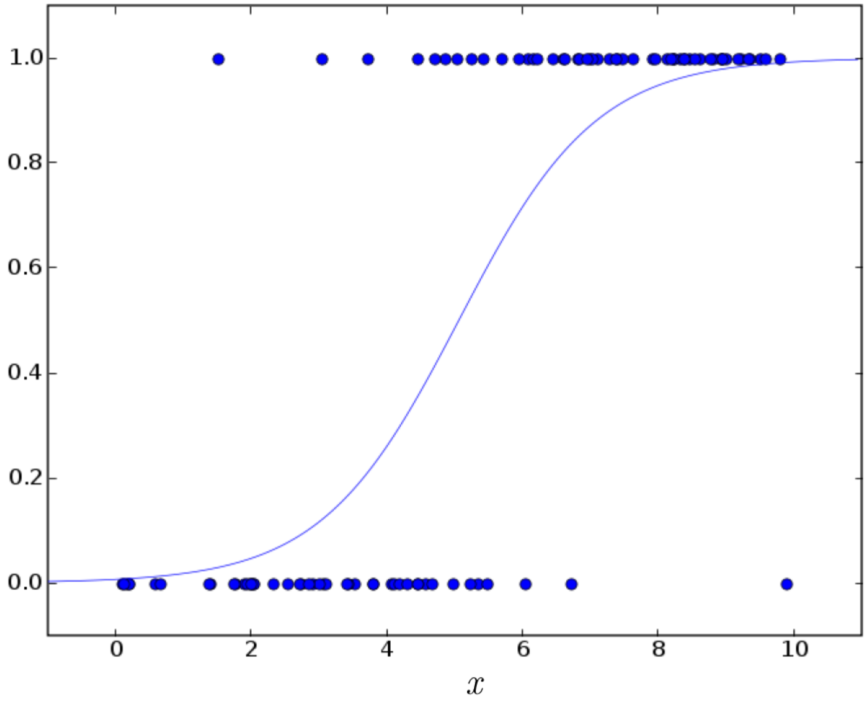


Lasso
"L1 regularization"

Quadratic
"L2 regularization"

4. Classification

Logistic Regression



Sigmoid

: $\sigma(x) = \frac{1}{1+e^{-x}}$

Logistic Regression

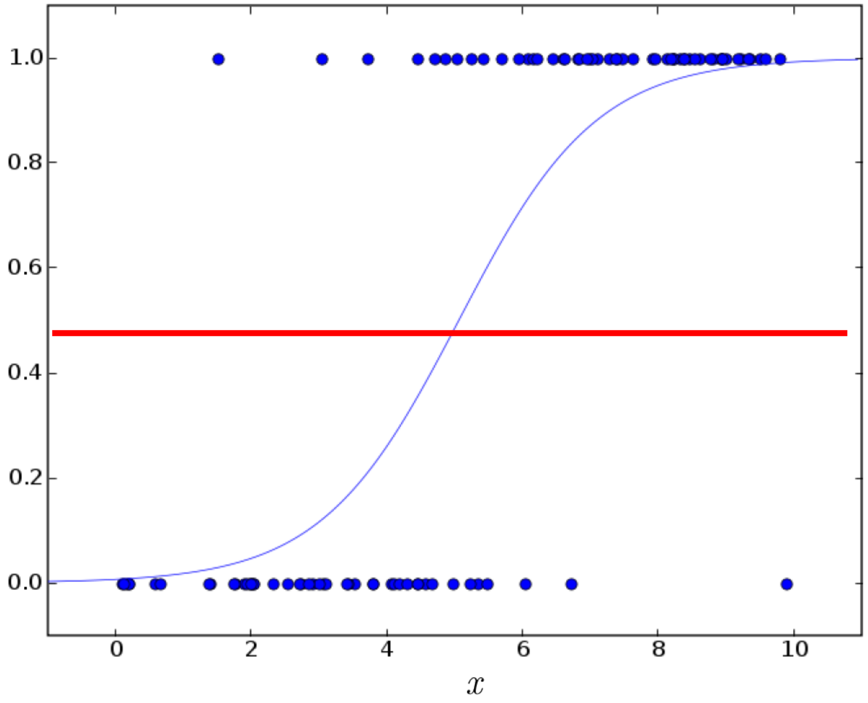
$y \sim B(n, p)$: Probability distribution form

✓ 오차항이 존재하지 않는다!

$$p = \frac{\exp(\boldsymbol{\beta X})}{1 + \exp(\boldsymbol{\beta X})}$$

4. Classification

Logistic Regression



Sigmoid

: $\sigma(x) = \frac{1}{1+e^{-x}}$

Logistic Regression

$y \sim B(n, p)$ **: Probability distribution form**

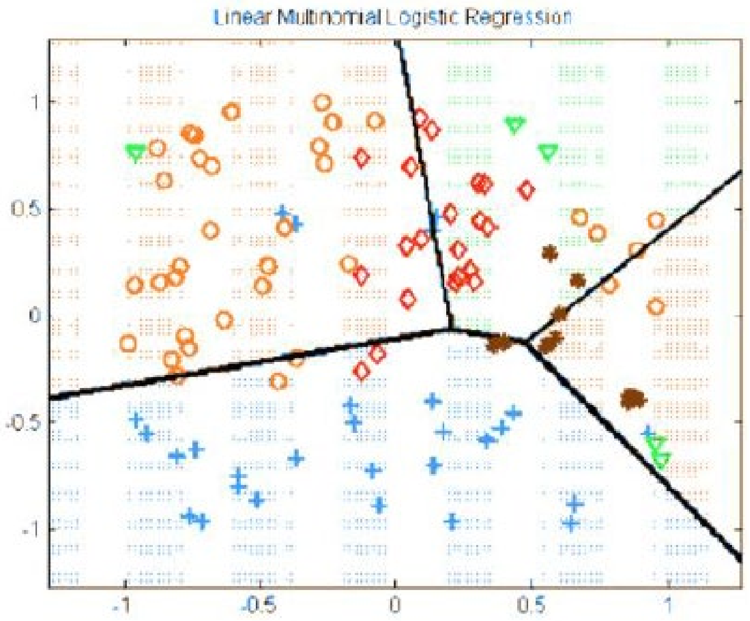
✓ **오차항이 존재하지 않는다!**

$$p = \frac{\exp(\boldsymbol{\beta}^T \boldsymbol{X})}{1 + \exp(\boldsymbol{\beta}^T \boldsymbol{X})}$$

$$\ln\left(\frac{p}{1-p}\right) = \boldsymbol{\beta}^T \boldsymbol{X}$$

4. Classification

Softmax → Multiclass!



$$p_i = \frac{\exp(\beta_i^T X)}{\sum_j \exp(\beta_j^T X)}$$

사실 softmax는

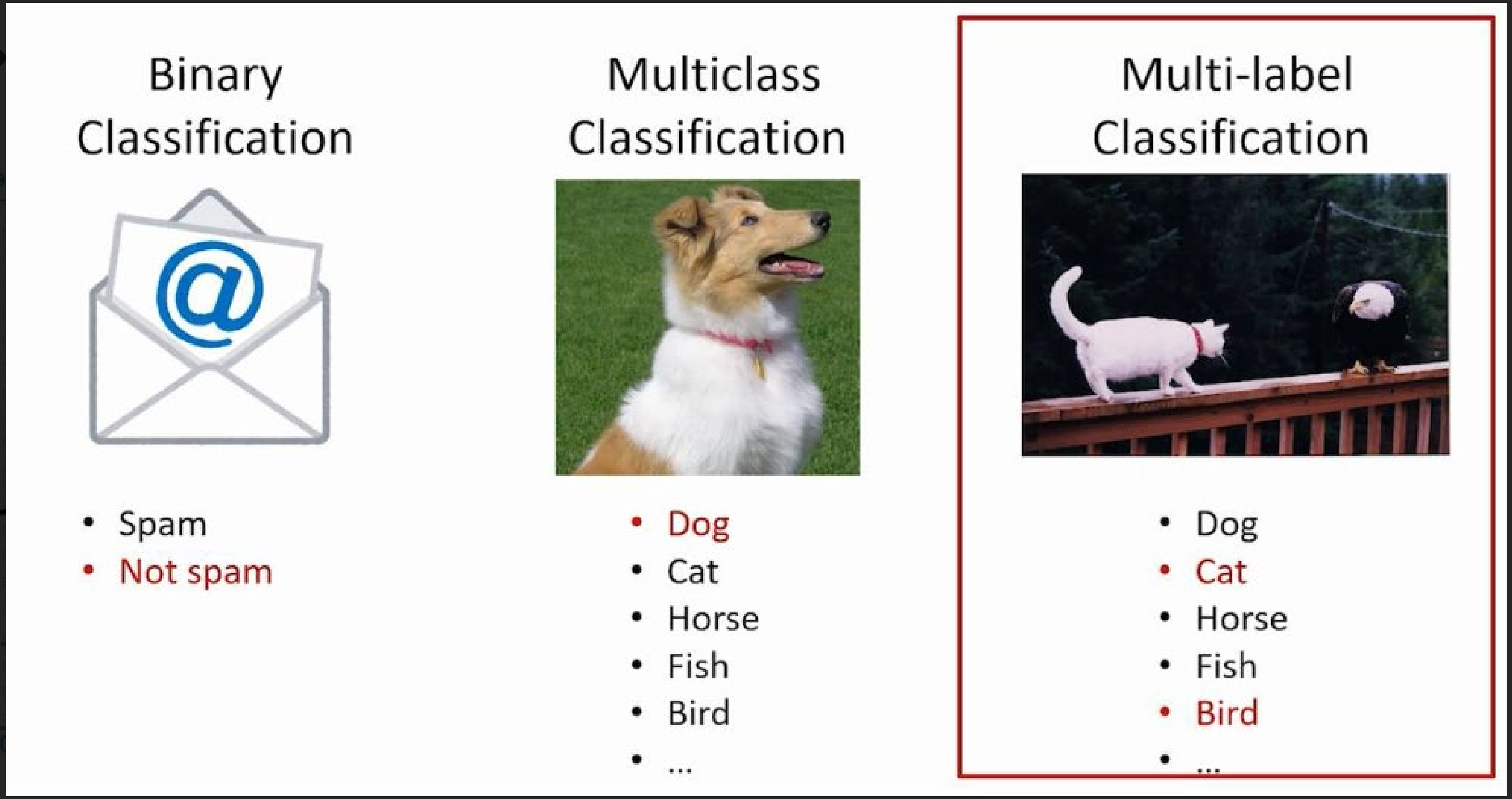
k개의 class에 대한 sigmoid로 생각할 수 있다!

Cf) multi-label과 multi-class는 다른 것입니다!

추가 설명판:

4. Classification

Softmax →



수 있다!

Cf) multi-label과 multi-class는 다른 것입니다!

https://medium.com/technovators/machine-learning-based-multi-label-text-classification-9a0e17f88bb4

4. Classification

CE Loss

Cross Entropy?!

$$H(p,q) = -E_p[\log q]$$

분포 p에 대해서 $\log q$에 대한 기댓값!

$$= -\sum_x p(x)[\log q(x)]$$

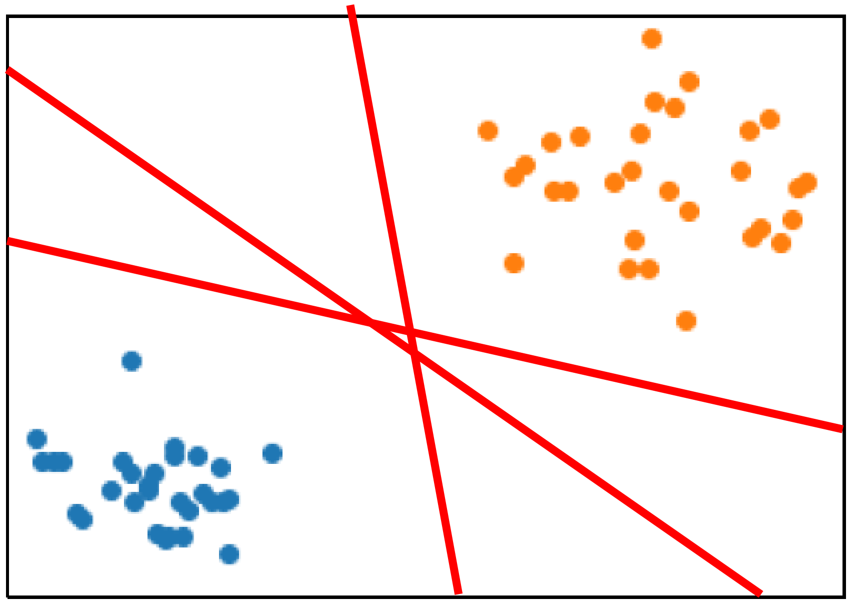
분포 p를 따르는 sample에 대해
$\log q$에 대의 평균값!

→ p와 q가 비슷하다면 CE Loss는 매우 작아진다!

5. SVM

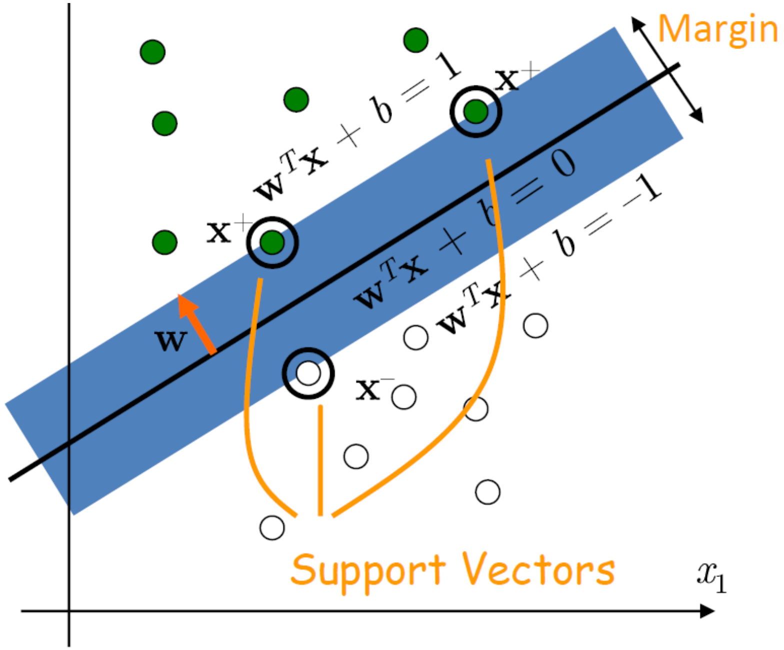
Overview

어떤 선(model)이 잘 분류하고 있는 것일까요?



5. SVM

Overview



Decision boundary : $\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b} = 0$

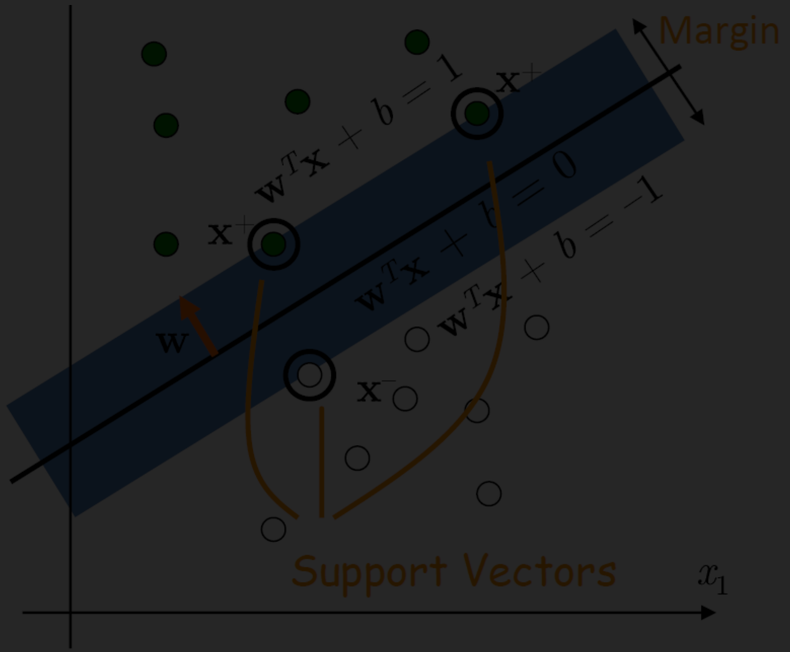
$$\hat{y} = \begin{cases} 0 & (\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b} < 0) \\ 1 & (\boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b} \geq 0) \end{cases}$$

Decision boundary와 가장 가까운 벡터 : **support vector**

Support vector와 decision boundary 사이 거리: **margin**

➔ SVM(Support Vector Machine)은 margin을 최대화!

Overview



Decision boundary : $\boldsymbol{w}^T\boldsymbol{x}+\boldsymbol{b}=0$

수식 주의!

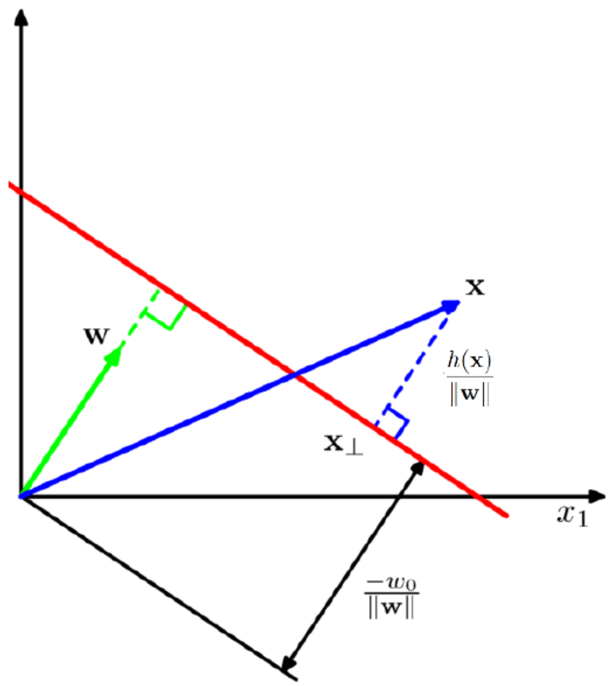
$$\hat{y}=\begin{cases}0 & (\boldsymbol{w}^T\boldsymbol{x}+\boldsymbol{b}<0)\\ 1 & (\boldsymbol{w}^T\boldsymbol{x}+\boldsymbol{b}\geq 0)\end{cases}$$

Decision boundary와 가장 가까운 벡터 : support vector

Support vector와 decision boundary 사이 거리: margin

➔ SVM(Support Vector Machine)은 margin을 최대화!

Problem Formulation



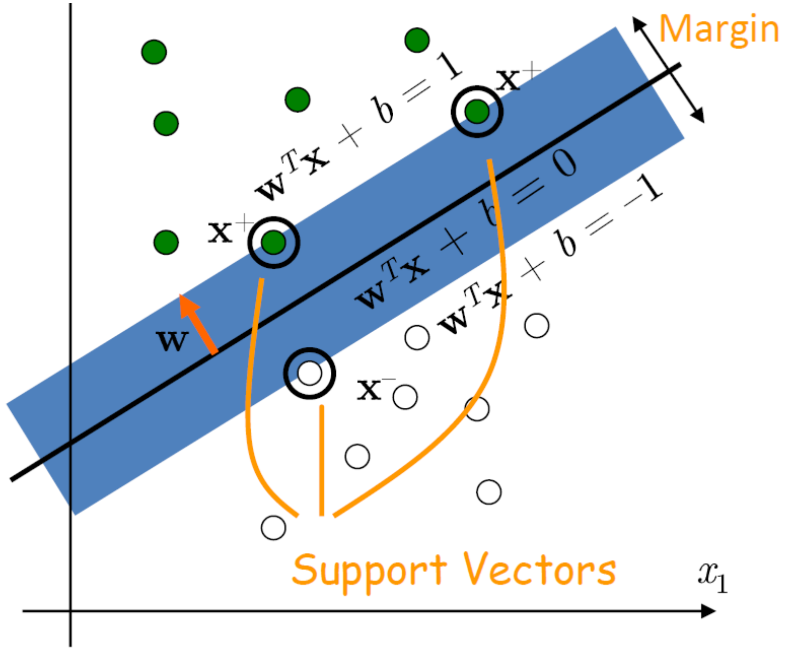
$$x - x_{\perp} = \delta \frac{w}{|w|}$$

$$w^T(x - x_{\perp}) = w^T \delta \frac{w}{|w|} = \delta w$$

$$w^T(x + b) - w^T(x_{\perp} + b) = w^T \delta \frac{w}{|w|} = \delta w$$

$$\delta = \frac{w^T x + b}{w} = y \frac{w^T x + b}{w}$$

Overview



$$\underset{w,b}{\operatorname{argmax}} \frac{1}{\|w\|} \min_n [y^{(n)} w^T x^{(n)} + b]$$

Support vector는

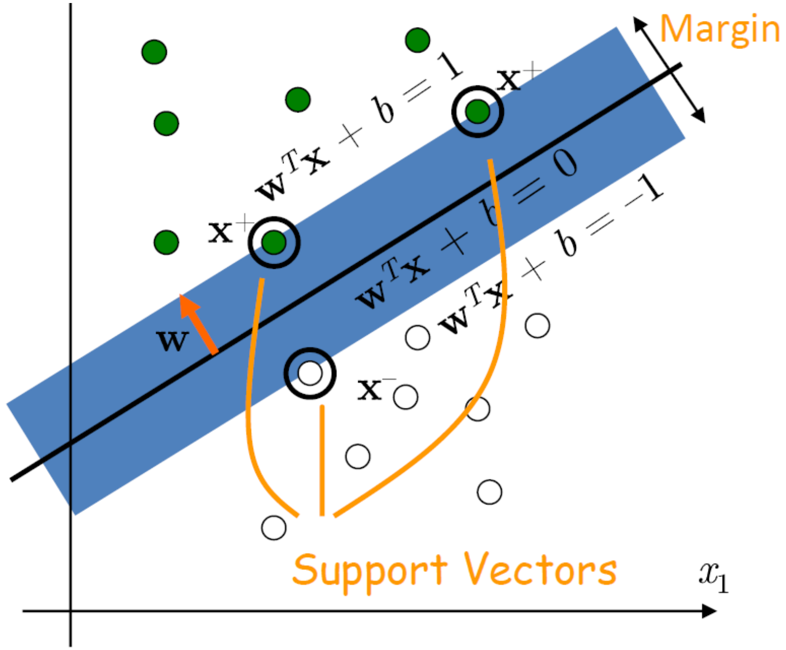
$$\min_n y^{(n)} [w^T x^{(n)} + b] = 1$$

을 만족하는 점으로 정의!

→ $w^T x^+ + b = 1$, $w^T x^- + b = -1$

$$\underset{w,b}{\operatorname{argmax}} \frac{1}{\|w\|} = \underset{w,b}{\operatorname{argmin}} \frac{1}{2}\|w\|^2$$

Overview



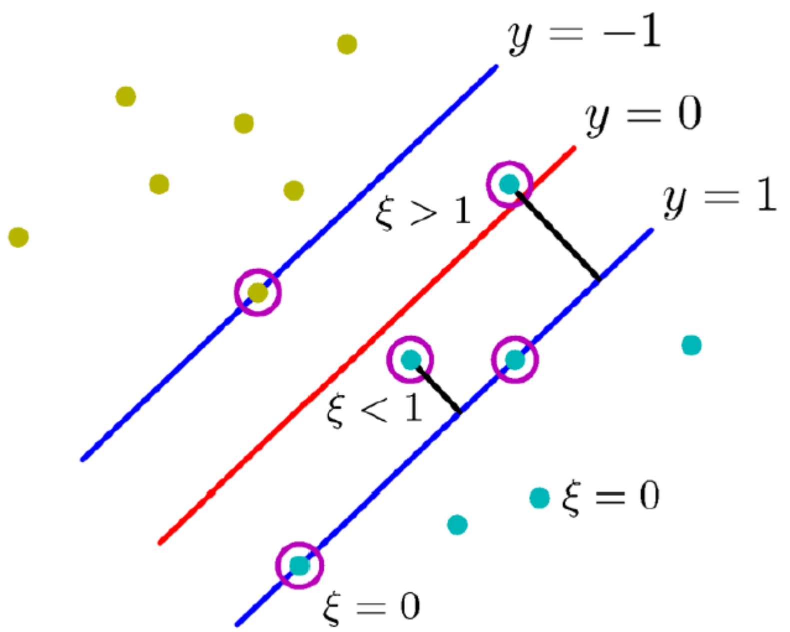
$$\underset{w,b}{\operatorname{argmin}} \frac{1}{2}\|w\|^2$$

$$y^{(n)}(w^T x^{(n)} + b) \geq 1$$

Constrained optimization problem

→ Lagrange multipliers(Convex Optimization)

Soft & Hard margin



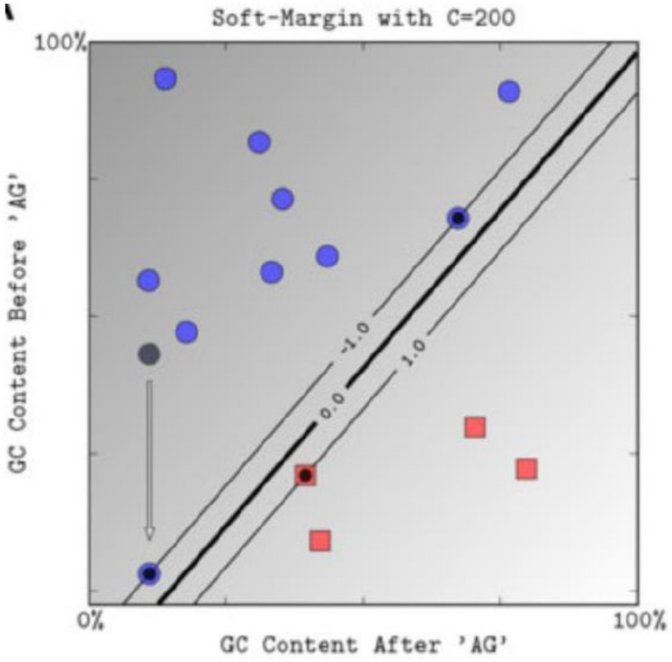
어느정도의 오차를 용인한다면?

오차 : slack variable $\xi^{(n)}$

$$\underset{w,b,\xi}{\operatorname{argmin}}[C\sum_{n}^{N}\xi^{(n)} + \frac{1}{2}\|w\|^2]$$

$$y^{(n)}(w^T x^{(n)} + b) \geq 1 - \xi^{(n)}$$

Soft & Hard margin



C가 큰 경우

→ 큰 패널티의 효과

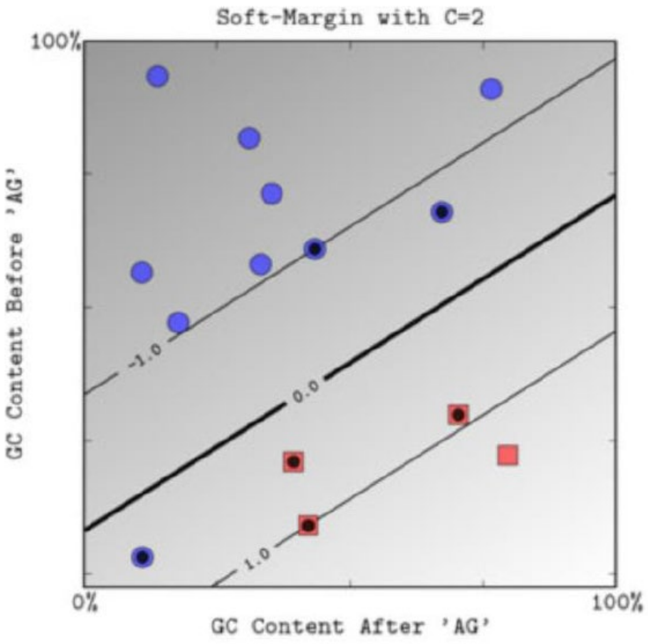
→ 오차가 적어지게

→ 좁은 margin을 가지기!

$$\underset{w,b,\xi}{\operatorname{argmin}}[C\sum_{n}^{N}\xi^{(n)}+\frac{1}{2}\|w\|^2]$$

$$y^{(n)}(w^Tx^{(n)}+b)\geq 1-\xi^{(n)}$$

Soft & Hard margin



C가 작은 경우

→ 작은 패널티의 효과

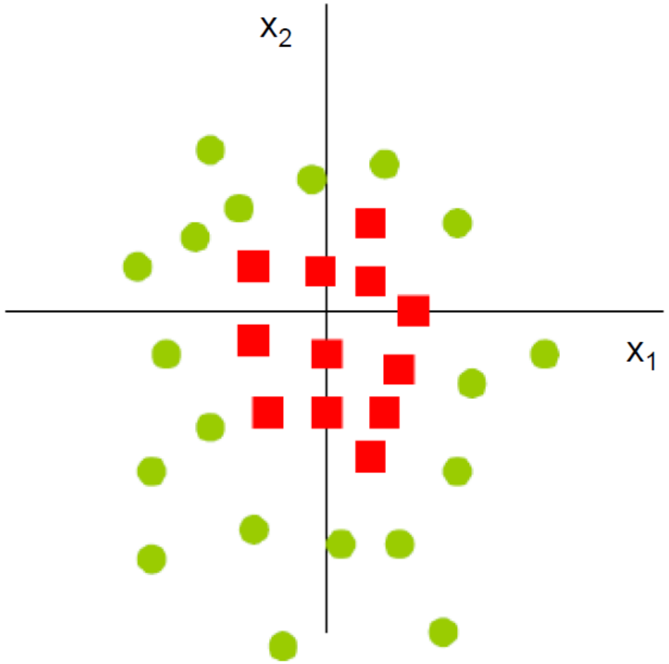
→ 오차가 커도 괜찮아

→ 넓은 margin을 가지기!

$$\underset{w,b,\xi}{\operatorname{argmin}}[C\sum_{n}^{N}\xi^{(n)}+\frac{1}{2}\|w\|^{2}]$$

$$y^{(n)}(w^{T}x^{(n)}+b)\geq 1-\xi^{(n)}$$

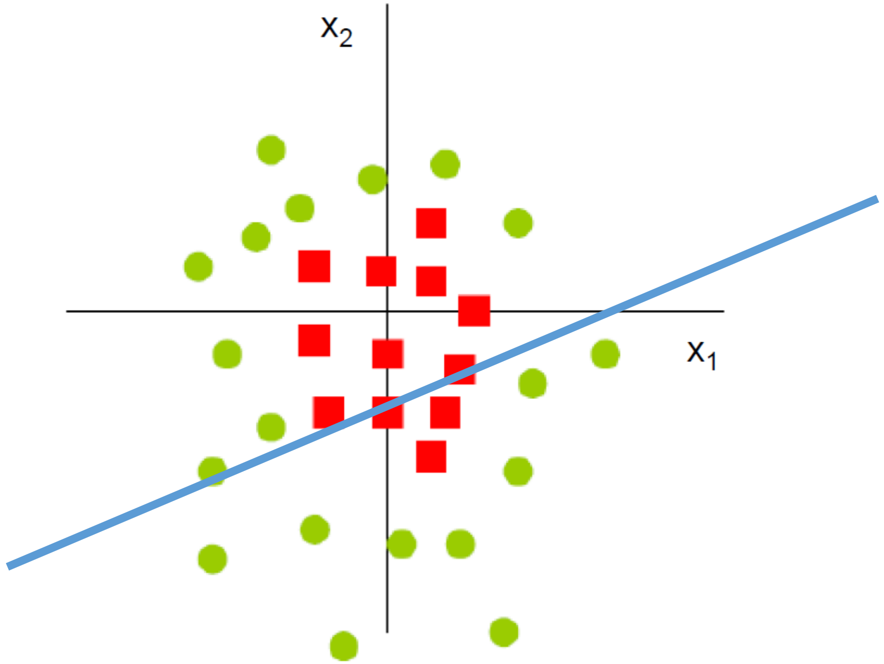
Nonlinearity



어떻게 구분할 수 있을까요?

+. Kernel

Nonlinearity

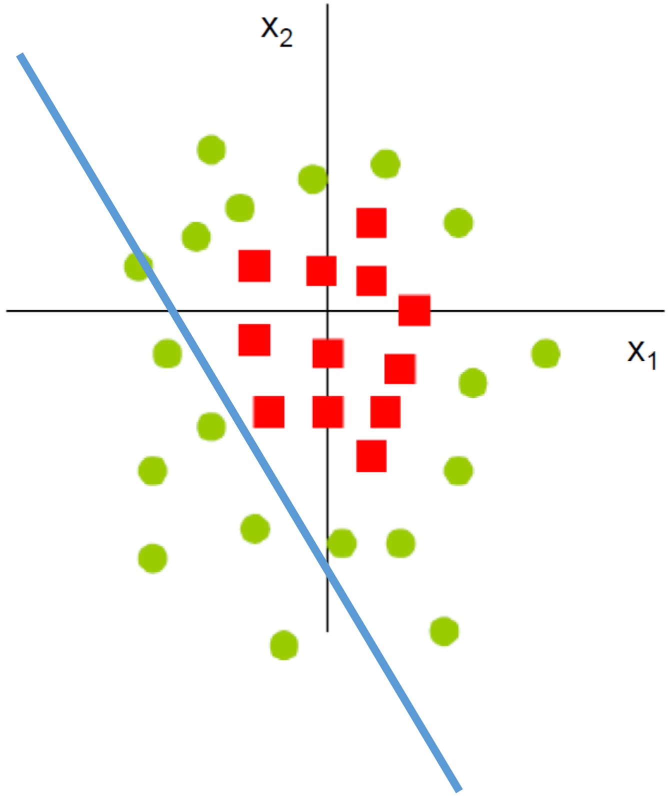


어떻게 구분할 수 있을까요?

이렇게?

+. Kernel

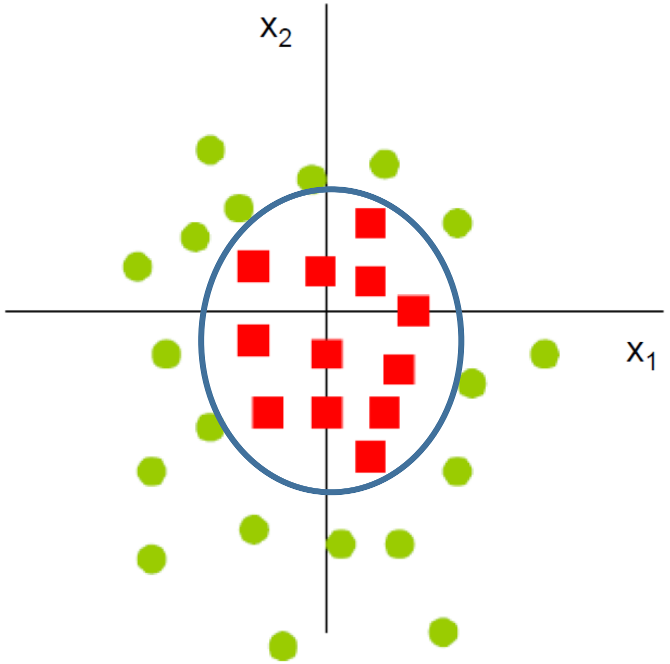
Nonlinearity



어떻게 구분할 수 있을까요?

이렇게?

Nonlinearity



어떻게 구분할 수 있을까요?

이렇게!

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2)$$

+. Kernel

Nonlinearity

$$(1, x_1) \rightarrow (1, x_1, x_1^2, x_1^3)$$

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2)$$

$$(x_1, x_2) \rightarrow (x_1^2 + 2x_1x_2 + x_2^2)$$

…

그런데 데이터와 변수가 매우 많다면 언제 다 계산할까?

Nonlinearity

$$\boldsymbol{x} = (x_1, x_2), \qquad \boldsymbol{z} = (z_1, z_2)$$

$$k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^T \boldsymbol{z})^2 = (\mathrm{x}_1\mathrm{z}_1 + \mathrm{x}_2\mathrm{z}_2)^2 = (\mathrm{x}_1^2\mathrm{z}_1^2 + 2\mathrm{x}_1\mathrm{x}_2\mathrm{z}_1\mathrm{z}_2 + \mathrm{x}_2^2\mathrm{z}_2^2)$$

$$= \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right) \cdot \left(z_1^2, \sqrt{2}z_1z_2, z_2^2\right)^T$$

Polynomial kernel

$$\boldsymbol{x} = (x_1, x_2), \qquad \boldsymbol{z} = (z_1, z_2)$$

$$k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^T \boldsymbol{z})^2 = (\mathrm{x_1 z_1 + x_2 z_2})^2 = (\mathrm{x_1^2 z_1^2 + 2x_1 x_2 z_1 z_2 + x_2^2 z_2^2})$$

$$= \boxed{\left(x_1^2, \sqrt{2} x_1 x_2, x_2^2\right)} \cdot \left(z_1^2, \sqrt{2} z_1 z_2, z_2^2\right)^T$$

$k(\boldsymbol{x}, \boldsymbol{z}) = \left(\boldsymbol{x}^T \boldsymbol{z}\right)^p$ **: 다항 커널**

Radial Basis Function **kernel**

$k(\boldsymbol{x}, z) = \exp\left(-\frac{|\boldsymbol{x}-\boldsymbol{z}|^2}{2\sigma^2}\right)$: **가우시안 커널(a.k.a. RBF 커널)**

→ **무한 차원의 특징을 사용하는 것과 같은 효과!**

Radial Basis Function kernel

$k(x,z) = \exp\left(-\frac{\boxed{|x-z|^2}}{2\sigma^2}\right)$: 가우시안 커널(a.k.a. RBF 커널)

→ 무한 차원의 특징을 사용하는 것과 같은 효과!

참고) 두 feature 간 거리도 kernel로 생각할 수 있다!

$$\begin{aligned} ||\phi(\mathbf{x}) - \phi(\mathbf{z})||^2 &= \langle \phi(\mathbf{x}) - \phi(\mathbf{z}), \phi(\mathbf{x}) - \phi(\mathbf{z}) \rangle \\ &= \langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle - 2\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle + \langle \phi(\mathbf{z}), \phi(\mathbf{z}) \rangle \\ &= \kappa(\mathbf{x}, \mathbf{x}) - 2\kappa(\mathbf{x}, \mathbf{z}) + \kappa(\mathbf{z}, \mathbf{z}) \end{aligned}$$

➔ Kernel에 몇가지 변환을 취한 것은 또 kernel이다!

증명!

Kernel

Radial B

$k(x, z) =$

→ 무한 차원으

$$e^{-\gamma(a-b)^2} = e^{-\gamma\left(a^2+b^2-2ab\right)} = e^{-\gamma\left(a^2+b^2\right)}e^{ab}$$

$$e^{ab} = 1 + \frac{1}{1!}ab + \frac{1}{2!}(ab)^2 + \frac{1}{3!}(ab)^3 + \cdots + \frac{1}{\infty!}(ab)^\infty$$

$$= \left(1, \sqrt{\frac{1}{1!}}a, \sqrt{\frac{1}{2!}}a^2, \dots, \sqrt{\frac{1}{\infty!}}a^\infty\right) \cdot \left(1, \sqrt{\frac{1}{1!}}b, \sqrt{\frac{1}{2!}}b^2, \dots, \sqrt{\frac{1}{\infty!}}b^\infty\right)$$

$s = \sqrt{e^{-\gamma\left(a^2+b^2\right)}}$ 라 하면

$$e^{-\gamma(a-b)^2} = \left(s, s\sqrt{\frac{1}{1!}}a, s\sqrt{\frac{1}{2!}}a^2, \dots, s\sqrt{\frac{1}{\infty!}}a^\infty\right) \cdot \left(s, s\sqrt{\frac{1}{1!}}b, s\sqrt{\frac{1}{2!}}b^2, \dots, s\sqrt{\frac{1}{\infty!}}b^\infty\right)$$

마무리하기 전에…

+) Convex Optimization

그 이후가 궁금하다면?

Dual Optimization & KKT condition

+) Kernel(커널)의 확장

SVM이나 기존의 Linear Regression을 kernel로 서술해보기!

6. Summary

Summary

- 지도 학습은 입력데이터에 상응하는 출력데이터가 있을 때 학습하는 방식입니다.
- Linear Regression에서 사용할 수 있는 방법은 OLS/SVM/GD가 있습니다
- 어떤 모델의 error는 ???의 variance와 ???와 ??? 간 편향과 데이터 자체의 noise로 생각할 수 있다.
- Logistic Regression은 Linear regression과 달리 확률 모형으로 모델링이 된다.
- SVM은 ???을 최대화 하는 분류기이며 분류할 때에 사용되는 점들은 ???이며 이것들로만 w와 b를 계산하게 된다.

Reference

Lecture Notes

- 6기 박준우님 세션 자료
- 이기복 교수님 통계적 머신러닝 강의안
- 강승호 교수님 이론 통계학(1) 강의안
- 회귀분석 수업 강의안

- https://www.quora.com/Support-Vector-Machines-What-is-an-intuitive-explanation-of-hyperplane
- https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote12.html
- https://medium.com/technovators/machine-learning-based-multi-label-text-classification-9a0e17f88bb4