Machine Learning

Introduction of Machine Learning & Regression Model

Present by Sangmin Bae

Contents

Part One

Introduction

What is Machine Learning?

머신 러닝(Machine Learning), **인공 지능**(Artificial Intelligence), **딥러닝**(Deep Learning)

인공 지능(AI) 머신 러닝(ML) 딥러닝(DL)

What is Machine Learning?

머신 러닝(Machine Learning) : 단어 그대로 **기계**를 **학습**한다.

1) **머신**이란?

   인간이 제공한 데이터를 표현할 수 있는 **모델** (= **함수**)

2) **학습**이란?

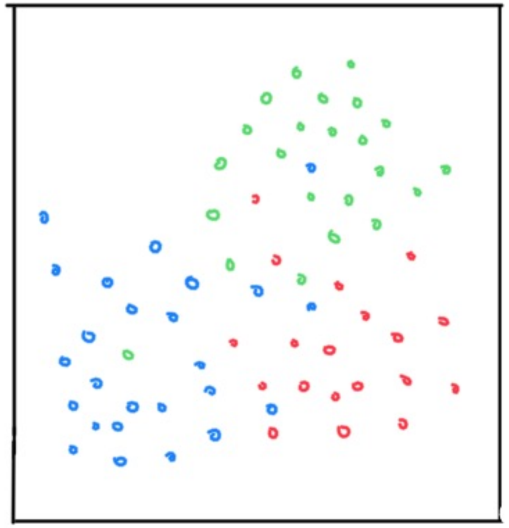
   데이터를 가장 잘 표현할 수 있는 **모델을 찾는 것** (= 모델의 파라미터 **최적화**)

3) 어떻게?

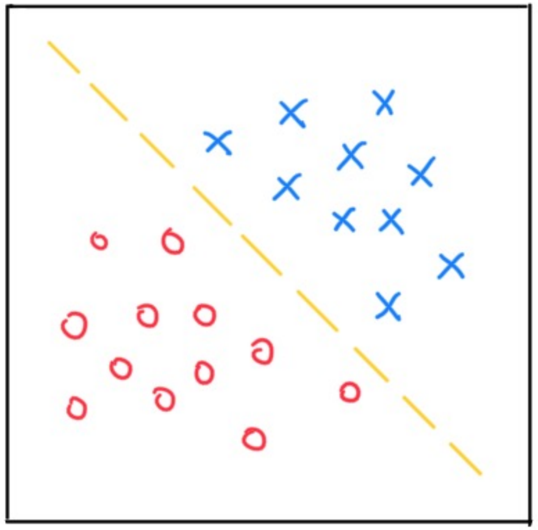
   통계적인 방법 혹은 **경사하강법**을 이용해 최적의 파라미터를 찾음

What is Machine Learning?

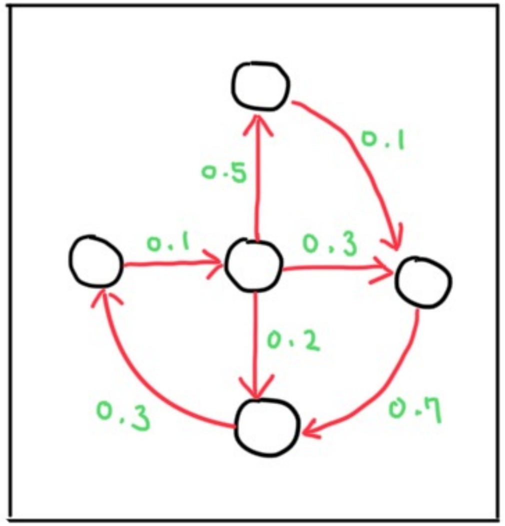
어떤 형태의 데이터가 머신에게 주어지는지에 따라 다음의 세부 분야들로 분류됨



비지도 학습



지도 학습



강화 학습

Lecture Contents (1)

Supervised Learning

1. 회귀
    - Linear and Nonlinear Regression
    - Gradient Descent
    - Bias and Variance Trade-off
2. 분류
    - Logistic and Softmax Regression
    - Support Vector Machine (SVM)
    - Decision Tree
    - Linear Discriminant Analysis (LDA)

Lecture Contents (2)

Supervised Learning

3. 앙상블 학습
    - Bagging
    - Boosting

Lecture Contents (2)

Unsupervised Learning

1. 차원 축소
   - Principal Component Analysis (PCA)
   - Singular Value Decomposition (SVD)
   - LDA, t-SNE, UMAP
2. 군집화
   - K-Means
   - Mean Shift
   - Gaussian Mixture Model
   - DBSCAN

ML vs. DL

머신 러닝(ML)

딥러닝(DL)

Part Two

Regression

Regression vs. Classification

회귀 (Regression)

1. 입력값 : 연속값(실수형), 이산값(범주형) 등 모두 가능
2. 출력값 : **연속값(실수형)**
3. 모델 형태 : 일반적인 함수 형태 (eg. $y = w_1x + w_0$)

분류 (Classification)

1. 입력값 : 연속값(실수형), 이산값(범주형) 등 모두 가능
2. 출력값 : **이산값(범주형)**
3. 모델 형태 : 이진 분류라면 시그모이드(sigmoid) 함수, 다중 분류라면 소프트맥스(softmax) 함수 꼭 포함

Notations (1)

데이터의 구성

- 데이터는 **피처(feature)**와 **라벨(label)**로 구성됨
- 이는 **독립 변수**와 **종속 변수**로도 불림
- 라벨은 y로 표기하며, 라벨의 유무로 지도학습, 비지도학습 구분

	혈압	몸무게	나이	지병
길동	130	34	14	x
철수	120	76	30	x
…	…	…	…	…
영희	150	50	51	o

Feature (=attribute, 피처)

- 데이터 **X**의 **특징**, 혹은 **항목**을 의미
- **N** : 데이터 샘플 갯수, **D** : 피처의 갯수
- ex) 혈압, 몸무게, 나이

$$\boldsymbol{X} = \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{bmatrix} \quad \boldsymbol{x}_1 = \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{N1} \end{bmatrix}, \cdots, \boldsymbol{x}_D = \begin{bmatrix} x_{1D} \\ x_{2D} \\ \vdots \\ x_{ND} \end{bmatrix}$$

Notations (2)

Parameter (=weight, 파라미터, 가중치)

- 주어진 데이터(입력값) 말고, 모델이 가지고 있는 **학습 가능한(learnable)** 파라미터

  ex) $w_0, w_1, \cdots, w_D$

Hyperparameter (하이퍼 파라미터)

- 모델 학습에 있어, **인간이 정해야하는 변수들**
- 학습률, 배치 크기 등등

Notations (3)

Input (입력값) vs. Output (출력값)

- Input : 모델(함수)에 입력되는 값으로 데이터의 **피처** 부분 ($\mathbf{x}$로 표기)
- Output : 모델로부터 출력되는 **예측값** ($\hat{\mathbf{y}}$로 표기)

선형 모델 vs. 비선형 모델

- Linear regression (선형 회귀) : 파라미터를 **선형 결합식**으로 표현 가능한 모델

  ex) $\mathrm{y} = \mathrm{w}_0 + \mathrm{w}_1\mathrm{x}_1 + \mathrm{w}_2\mathrm{x}_2 + \cdots + \mathrm{w}_D\mathrm{x}_D, \mathrm{y} = \mathrm{w}_0 + \mathrm{w}_1\mathrm{x} + \mathrm{w}_2\mathrm{x}^2$
- Nonlinear regression (비선형 회귀) : 선형 결합식으로 표현 불가능한 모델

  ex) $\log(\mathrm{y}) = w_0 + w_1\log(\mathrm{x}),\ \mathrm{y} = \max(\mathrm{x}, 0)$

Basic Math for ML (1)

함수

- **두 집합 사이의 관계, 혹은 규칙**
- $\mathbf{y} = \mathbf{f}(\mathbf{x})$ 의 식으로 표현, 이 때의 $\mathbf{x}$는 입력값, $\mathbf{y}$는 출력값

일차 함수

- $\mathbf{y}$가 $\mathbf{x}$에 대한 **일차식**으로 표현된 경우
- $\mathbf{y} = \mathbf{ax} + \mathbf{b}\ (\mathbf{a} \neq \mathbf{0})$
- $\mathbf{a}$를 **기울기**, $\mathbf{b}$를 **절편**이라고 표현

Basic Math for ML (2)

이차 함수

- y가 x에 대한 **이차식**으로 표현된 경우
- $\mathbf{y = a(x - p)^2 + q \ (a \neq 0)}$

Basic Math for ML (3)

순간 변화율

- $\mathbf{x}$의 값이 **미세하게 변화**했을 때, $\mathbf{y}$의 변화율

$$\lim_{\Delta x \to 0} \frac{f(a+\Delta x)-f(a)}{\Delta x}$$

- 어떤 $\mathbf{x}$ 값(= a)에서의 그래프와 맞닿는 **접선의 기울기**

Basic Math for ML (4)

미분

- 함수 $f(x)$를 미분한다는 것은 함수의 순간 변화율을 구한다는 뜻
- $\mathbf{f'(x)}$ 또는 $\mathbf{\frac{d}{dx}f(x)}$ 로 표기
- Ex. $\mathbf{f(x) = ax,\ \ f(x) = x^a}$

함수의 최솟값

- **함수의 최솟값에서의 미분값(순간 변화율)은 항상 0 임**
- 이를 바탕으로 파라미터의 최적값을 구할 수 있음

Basic Math for ML (5)

지수함수

- $\mathbf{y = a^{x}}\ (\mathbf{a \neq 1, a > 0})$
- $\mathbf{a}$를 **밑**, $\mathbf{x}$를 **지수**라고 부름
- 한 쪽은 0으로 수렴, 다른 쪽은 ∞로 발산

Basic Math for ML (6)

자연 상수

- $\mathbf{e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n}$
- '자연 로그의 밑' 또는 '오일러의 수' 등으로 불림
- π 처럼 수학에서 중요하게 사용되는 무리수 ($\approx \mathbf{2.718281828} \cdots$)
- **100%의 성장률**을 가지고 **1회 연속 성장** 할 때 가질 수 있는 **최대 성장량**
- $\mathbf{\frac{d}{dx} e^x = e^x}$

Basic Math for ML (7)

시그모이드 함수 (sigmoid function)

- 이진 분류 문제를 위한 비선형 함수
- $\mathbf{y} = \frac{\mathbf{1}}{\mathbf{1+e^{-x}}}$
- 함수의 출력값이 항상 0이상 1이하며, 중앙 출력값은 0.5임

소프트맥스 함수 (softmax function)

- **다중 분류 문제**를 위한 비선형 함수
- $\mathbf{y_i} = \frac{\mathbf{e}^{\mathbf{x}_i}}{\sum_{k=1}^{K} e^{x_k}}$ ($\mathbf{k}$ 는 클래스 갯수)

Basic Math for ML (8)

로그 함수

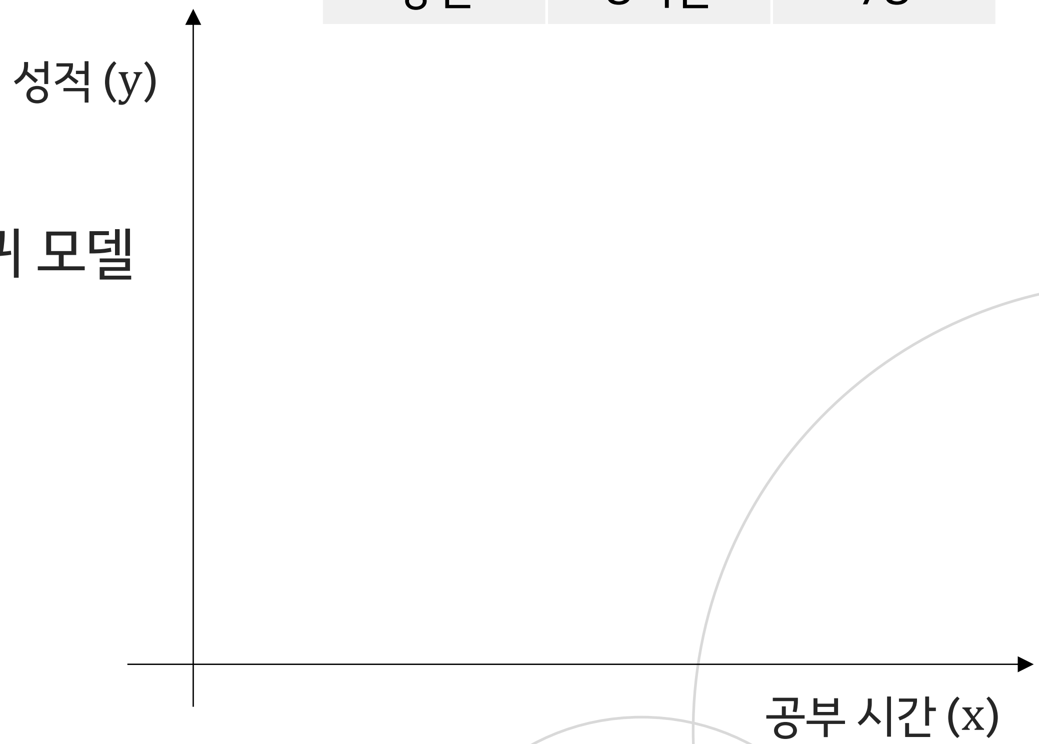
- $\mathbf{y = log_a x}$
- 지수 함수와 **역함수**의 관계
- 로그 함수의 밑이 e 일 때, $y = \ln x$

Linear Regression

	공부 시간	성적
길동	2시간	40
철수	4시간	58
영희	6시간	64
상민	8시간	78

단순 선형 회귀 (simple linear regression)

- **피처의 종류가 한 개**인 데이터에 대한 회귀 모델
- $y = w_0 + w_1 x$



다중 선형 회귀 (multiple linear regression)

- **피처의 종류가 여러 개**인 데이터에 대한 회귀 모델
- $y = w_0 + w_1 x_1 + \cdots + w_D x_D$

다항 회귀 (polynomial regression)

- 독립 변수(피처)의 **차수를 높인** 회귀 모델
- $y = w_0 + w_1 x + w_2 x^2 + w_m x^m$

How to find optimal parameters?

(Remind) Parameter (=weight, 파라미터, 가중치)

- 주어진 데이터(입력값) 말고, 모델이 가지고 있는 **학습 가능한(learnable)** 파라미터

  ex) $w_0, w_1, \cdots, w_D$

Optimal (최적의) 이란 뜻은 데이터를 가장 잘 표현한다는 말과 동치

→ 모델 **예측값($\hat{y}$) 과 실제값(y) 의 차이가 가장 적은** 모델

→ **손실 함수값을 최소**로 만드는 모델 파라미터

Part Three

Optimization

Advanced math for ML

편미분

- **원하는 변수에 대해서만 미분**하는 것
- 그 외의 모든 것들은 **상수 취급**
- $\frac{\partial y}{\partial x}$
- Ex. $f(x, y) = x^2 + xy + 3$

연쇄법칙 (chain rule)

- $\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$
- Ex. $y = \ln(u), u = 2x + 4$

Loss function

평균 제곱 오차 (mean squared error, MSE)

- **회귀 문제**에서의 대표적인 손실 함수
- **오차의 제곱의 평균**
- $L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

Least Square Method (1)

최소 제곱법 (least square method)

- 최적의 파라미터를 구할 수 있는 한 방법으로, **데이터에 대한 오차를 최소화**하도록 함
- 기울기 a 와 절편 b 의 일차 함수 ($\mathbf{L} = \sum_{i=1}^{N}\left(\mathbf{y_i} - (\mathbf{ax_i} + \mathbf{b})\right)^2$)
- 풀이 방법 1
  - $\mathbf{0} = \frac{\partial \mathbf{L}}{\partial \mathbf{a}} = \sum_{i=1}^{N} 2\left(y_i - (ax_i + b)\right)(-x_i) = 2(a\sum_{i=1}^{N} x_i^2 + b\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} x_i y_i)$
  - $\mathbf{0} = \frac{\partial \mathbf{L}}{\partial \mathbf{b}} = \sum_{i=1}^{N} 2\left(y_i - (ax_i + b)\right)(-1) = 2(a\sum_{i=1}^{N} x_i + b\sum_{i=1}^{N} 1 - \sum_{i=1}^{N} y_i)$

➢ $\mathbf{a}^* = \frac{\sum_{i=1}^{N}(\mathbf{x}-\bar{\mathbf{x}})(\mathbf{y}-\bar{\mathbf{y}})}{\sum_{i=1}^{N}(\mathbf{x}-\bar{\mathbf{x}})^2}$

➢ $\mathbf{b}^* = \bar{\mathbf{y}} - \mathbf{a}^*\bar{\mathbf{x}}$

Least Square Method (2)

최소 제곱법 (least square method)

- 최적의 파라미터를 구할 수 있는 한 방법으로, **데이터에 대한 오차를 최소화**하도록 함
- 기울기 a 와 절편 b 의 일차 함수 ($\mathbf{L} = \sum_{\mathbf{i}=\mathbf{1}}^{N} \left(\mathbf{y_i} - (\mathbf{a}\mathbf{x_i} + \mathbf{b}) \right)^{\mathbf{2}}$)
- 풀이 방법 2
  - $\|\mathbf{Y} - \mathbf{W}\mathbf{X}\|^2$ 행렬에 대한 편미분
  - $-2\mathrm{X}^{\mathrm{T}}(\mathrm{Y} - \mathrm{W}\mathrm{X}) = 0$

➢ $\mathbf{W} = \left(\mathbf{X}^{\mathbf{T}}\mathbf{X}\right)^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{Y}$

(Remind) Linear Regression

단순 선형 회귀 (simple linear regression)

- **피처의 종류가 한 개**인 데이터에 대한 회귀 모델
- $y = w_0 + w_1 x$

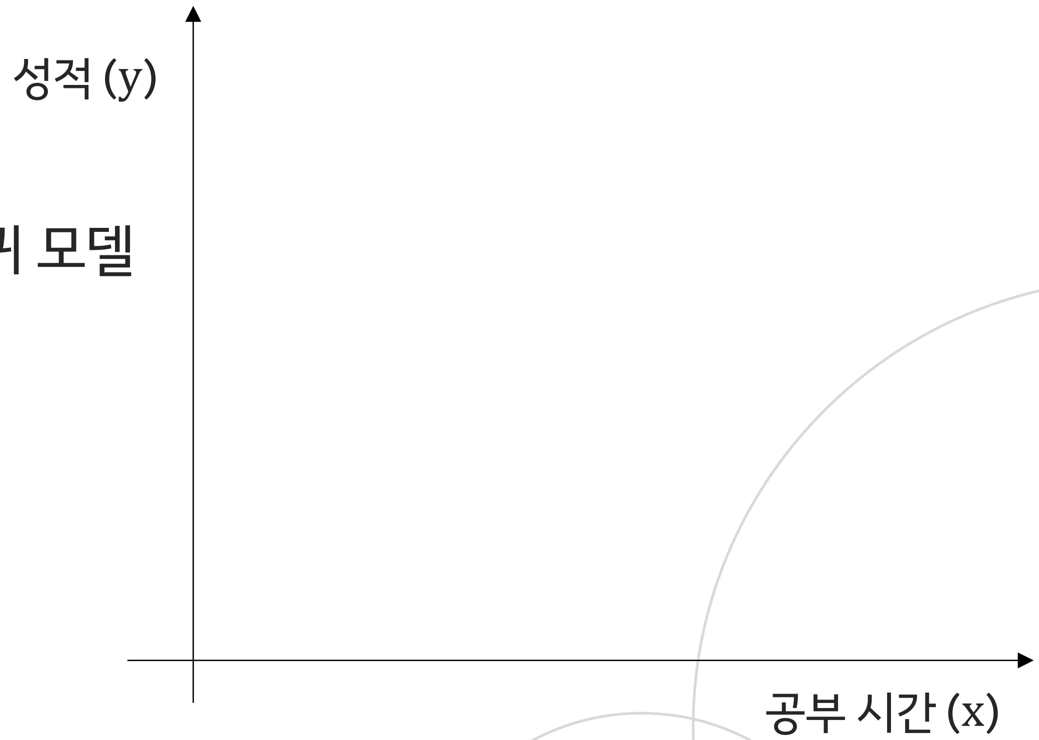
➢ $a^* = \frac{\sum_{i=1}^{N}(x-\bar{x})(y-\bar{y})}{\sum_{i=1}^{N}(x-\bar{x})^2}$

➢ $b^* = \bar{y} - a^* \bar{x}$

다중 선형 회귀 (multiple linear regression)

- **피처의 종류가 여러 개**인 데이터에 대한 회귀 모델
- $y = w_0 + w_1 x_1 + \cdots + w_D x_D$

	공부 시간	성적
길동	2시간	40
철수	4시간	58
영희	6시간	64
상민	8시간	78



Gradient Descent (1)

복잡한 함수의 경우..

- 다중 선형 회귀, 다항 회귀, 비선형 함수
- **최소 제곱법으로 해결 어려움**
- 어떻게 최적의 파라미터를 찾을 수 있을까?

경사 하강법 (gradient descent)

- **손실 함수의 값을 최소화시키는 방향**으로 파라미터를 업데이트하자!

Gradient Descent (2)

경사 하강법 (gradient descent)

- **손실 함수의 값을 최소화시키는 방향**으로 파라미터를 업데이트하자!
- **함수의 최솟값은 무조건 순간 변화율이 0이다!**
- 손실 함수에 대한 미분값이 0이 되는 방향으로 파라미터의 업데이트 방향을 결정

슈도 코드 (pseudo code)

1. 현재 파라미터에서의 **손실 함수에 대한 미분값**을 구함
2. **미분값의 반대 방향**으로 파라미터값을 업데이트
3. 미분값이 0이 될 때까지 1~2번을 **에폭(epoch)**만큼 반복

Gradient Descent (3)

학습률 (learning rate)

- 계산한 미분값을 그대로 사용해 업데이트 하지않고, **학습률 x 미분값**을 사용함
- 학습률이 크면?
- 학습률이 작으면?

Gradient Descent (4)

학습률 스케줄러 (learning rate scheduler)

- 일반적으로 학습률을 큰 값에서 작은 값으로 변화시킴
- Multi-step scheduler
- Cosine annealing sheduler

Example for Gradient descent

(Remind) 선형 회귀 문제

- 기울기 a 와 절편 b 의 일차 함수 ($\mathbf{L} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \left(\mathbf{y_i} - (\mathbf{ax_i} + \mathbf{b}) \right)^{\mathbf{2}}$)
- 풀이 방법 1
  - $\mathbf{0} = \frac{\partial \mathbf{L}}{\partial \mathbf{a}} = \sum_{i=1}^{N} 2\left(y_i - (ax_i + b) \right)(-x_i) = 2(a\sum_{i=1}^{N} x_i^2 + b\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} x_i y_i)$
  - $\mathbf{0} = \frac{\partial \mathbf{L}}{\partial \mathbf{b}} = \sum_{i=1}^{N} 2\left(y_i - (ax_i + b) \right)(-1) = 2(a\sum_{i=1}^{N} x_i + b\sum_{i=1}^{N} 1 - \sum_{i=1}^{N} y_i)$

Advanced Gradient Descent (1)

확률적 경사 하강법 (stochastic gradient descent)

- 데이터가 굉장히 많을 때, **전체 데이터셋을 활용한 경사 하강법**은 계산 비용이 매우 큼
- $W = (X^T X)^{-1} X^T Y$
- **1개의 데이터만으로 업데이트**하고, 이를 n번 반복

미니 배치 확률적 경사 하강법 (mini batch stochastic gradient descent)

- 둘의 절충안으로, **배치 (batch)** 개념 도입
- 배치의 크기를 조절해, 학습 속도와 정확도를 조절

Advanced Gradient Descent (2)

이 밖의 다양한 경사 하강법 방법

1. Momentum
2. Nesterov Accelerated Gradient
3. Adagrad (Adaptive Gradient) : 변수마다 업데이트 크기를 조절하는 방식
4. RMSProp
5. Adam

Part Four

Bias & Variance

Training data vs. Test data

데이터의 분할

- 입력된 데이터는 **학습 데이터**와 **평가 데이터**로 나눌 수 있음
- 학습 데이터는 **모델 학습에 사용**되는 모든 데이터셋
- 평가 데이터는 **오직 모델의 평가만을 위해 사용**되는 데이터셋
- **평가 데이터는 절대로 모델 학습에 사용되면 안됨**

평가 데이터

- 학습 데이터와 평가 데이터는 같은 분포를 가지는가?
- 평가 데이터는 어느 정도 크기를 가져야 하는가?

Bias and Variance Trade-off (1)

모델의 복잡도

- **모델의 파라미터 수**가 많아질수록(선형에서 비선형 모델로 갈수록), **복잡도**가 증가함
- 모델이 복잡해질수록, 학습 데이터를 더 완벽하게 학습함
- 학습과 평가 데이터의 분포가 같다는 가정하에 어떤 복잡도가 좋은가?
    1. 학습 데이터가 많은 상황 (Under-fitting)
    2. 학습 데이터가 적은 상황 (Over-fitting, 과적합)

Bias and Variance Trade-off (2)

편향(bias)과 분산(variance)

- 편향과 분산은 알고리즘이 가지고 있는 에러의 종류
- $\mathbf{MSE}(\hat{\theta}) \equiv \mathbf{E}_{\theta}\left((\hat{\theta}-\theta)^2\right) = \mathbf{E}\left((\hat{\theta}-\mathbf{E}(\hat{\theta})+\mathbf{E}(\hat{\theta})-\theta)^2\right)$

$$= \mathbf{E}\left(\left(\hat{\theta}-\mathbf{E}(\hat{\theta})\right)^2 + 2((\hat{\theta}-\mathbf{E}(\hat{\theta}))(\mathbf{E}(\hat{\theta})-\theta)) + \left(\mathbf{E}(\hat{\theta})-\theta\right)^2\right)$$

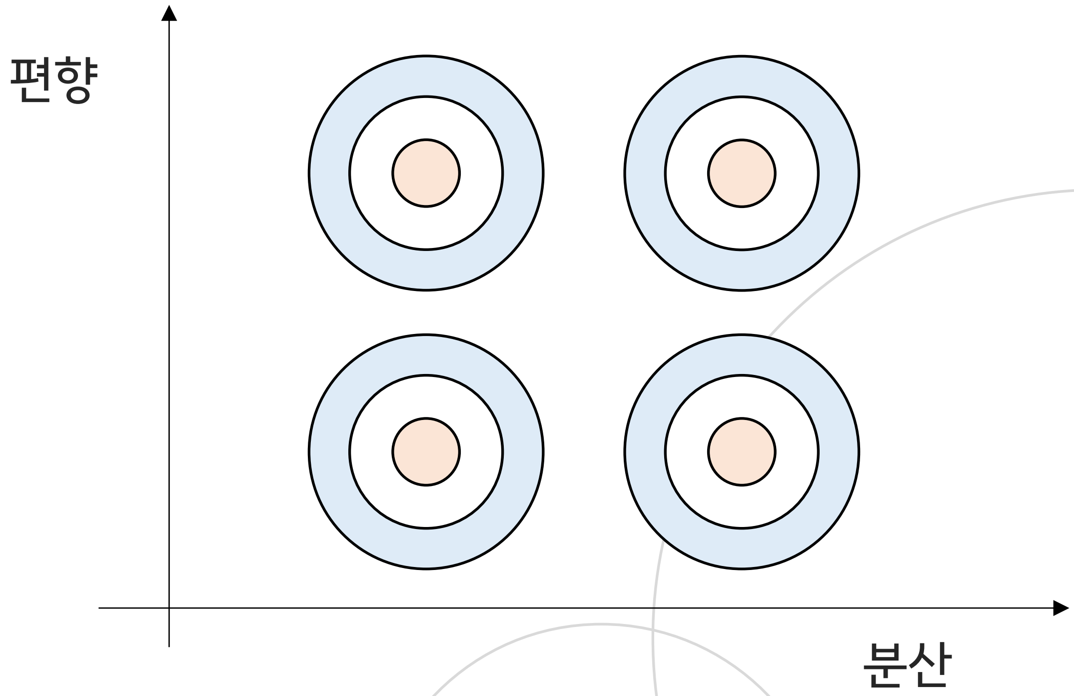
$$= \mathbf{E}\left(\left(\hat{\theta}-\mathbf{E}(\hat{\theta})\right)^2\right) + 2\left(\mathbf{E}(\hat{\theta})-\theta\right)\mathbf{E}\left(\hat{\theta}-\mathbf{E}(\hat{\theta})\right) + \left(\mathbf{E}(\hat{\theta})-\theta\right)^2$$

$$= \mathbf{E}\left(\left(\hat{\theta}-\mathbf{E}(\hat{\theta})\right)^2\right) + \left(\mathbf{E}(\hat{\theta})-\theta\right)^2 = \mathbf{Var}_{\theta}(\hat{\theta}) + \mathbf{Bias}_{\theta}(\hat{\theta},\theta)^2$$

Bias and Variance Trade-off (3)

편향(bias)과 분산(variance)

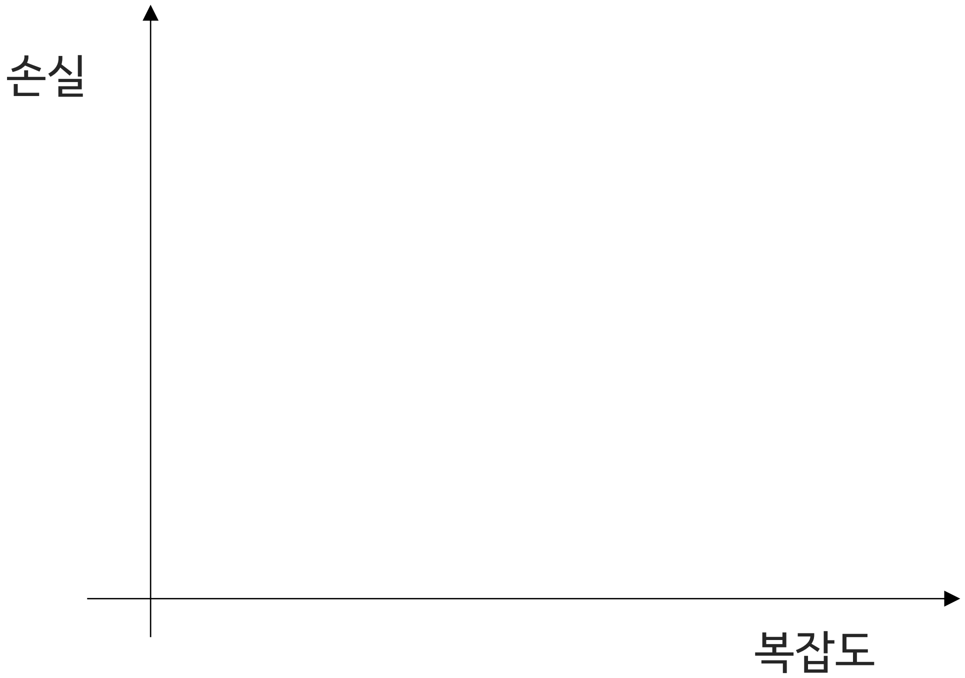
- $\mathbf{MSE}(\hat{\boldsymbol{\theta}}) = \mathbf{E}_{\boldsymbol{\theta}}\left((\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^2\right) = \mathbf{E}\left(\left(\hat{\boldsymbol{\theta}} - \mathbf{E}(\hat{\boldsymbol{\theta}})\right)^2\right) + \left(\mathbf{E}(\hat{\boldsymbol{\theta}}) - \boldsymbol{\theta}\right)^2 = \mathbf{Var}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}) + \mathbf{Bias}_{\boldsymbol{\theta}}(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})^2$
- 분산은 over-fitting과 관련 있는 개념
- 편향은 under-fitting과 관련 있는 개념

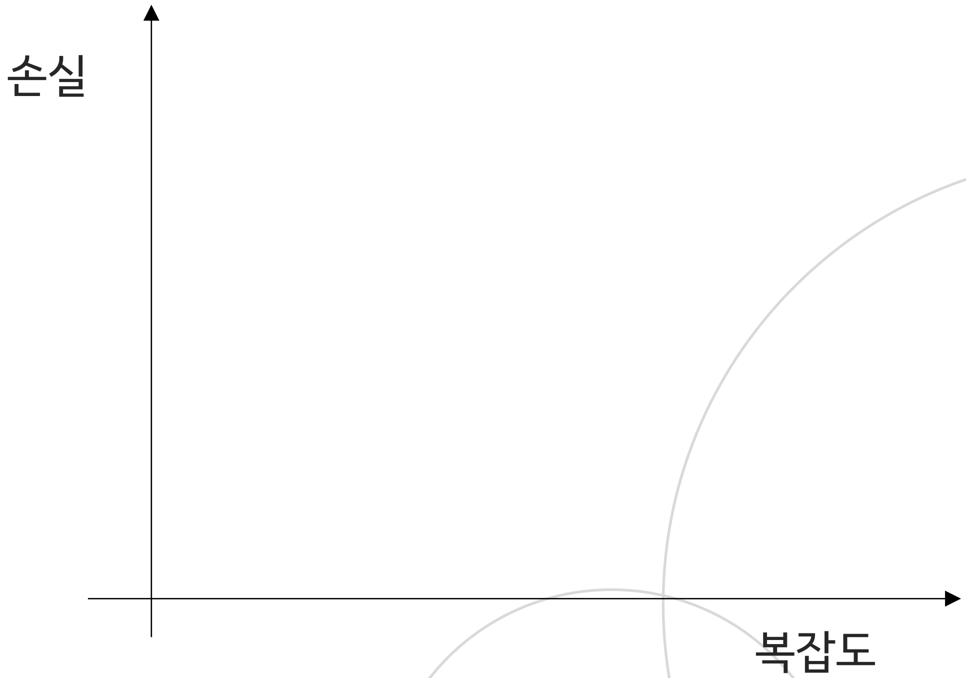


Bias and Variance Trade-off (4)

편향(bias)과 분산(variance)

- $\mathbf{MSE}(\hat{\theta}) = \mathbf{E}_{\theta}\left((\hat{\theta} - \theta)^2\right) = \mathbf{E}\left(\left(\hat{\theta} - \mathbf{E}(\hat{\theta})\right)^2\right) + \left(\mathbf{E}(\hat{\theta}) - \theta\right)^2 = \mathbf{Var}_{\theta}(\hat{\theta}) + \mathbf{Bias}_{\theta}(\hat{\theta}, \theta)^2$
- Training error vs. Test error





How to solve that trade-off?

해결방안

- 일반적으로 모델의 복잡도를 키우고, **과적합을 막는 방법론**을 사용
  1. **검증 데이터셋** 활용
  2. K-fold cross validation
  3. **정규화 손실 함수**

Validation data

검증 데이터셋

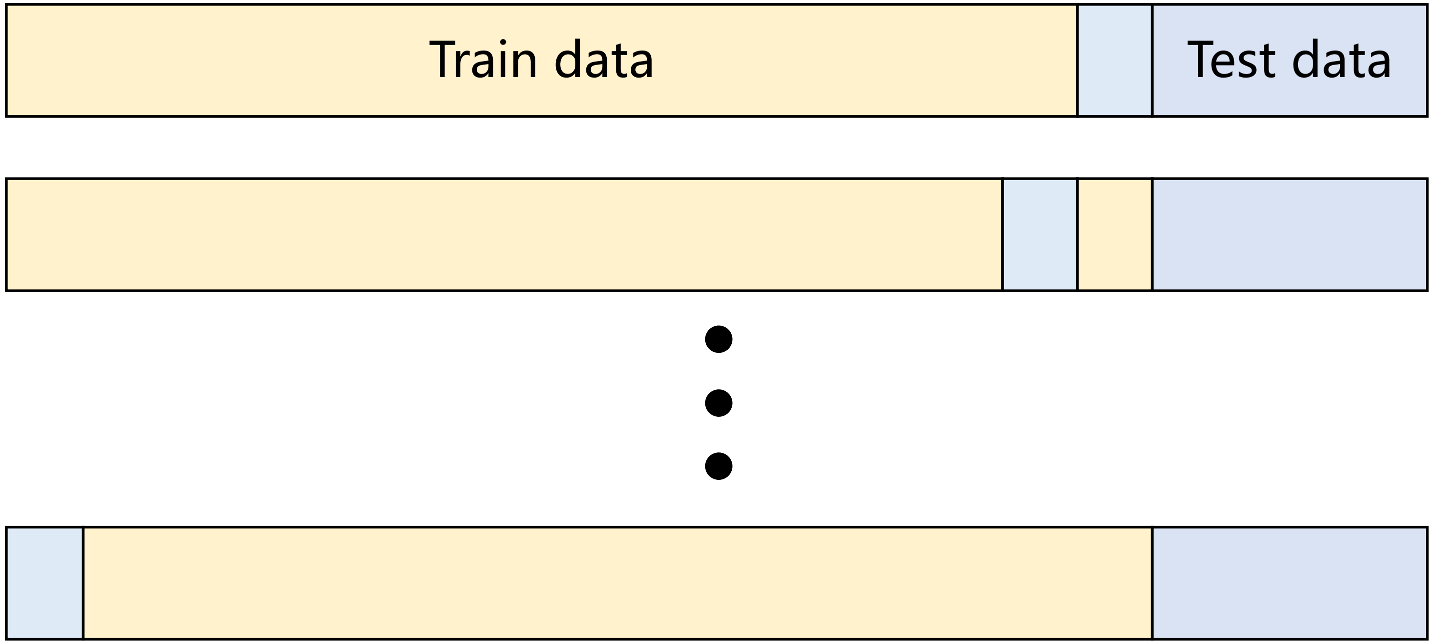
- 모델 **학습의 정도를 검증**하기 위한 데이터셋
- 모델 학습에 직접적으로 참여하지 못함
- **학습 중간에 계속해서 평가**를 하고, 가장 좋은 성능의 파라미터를 저장해 둠

Train data	Valid data	Test data

Leave-One-Out Cross-Validation

LOOCV

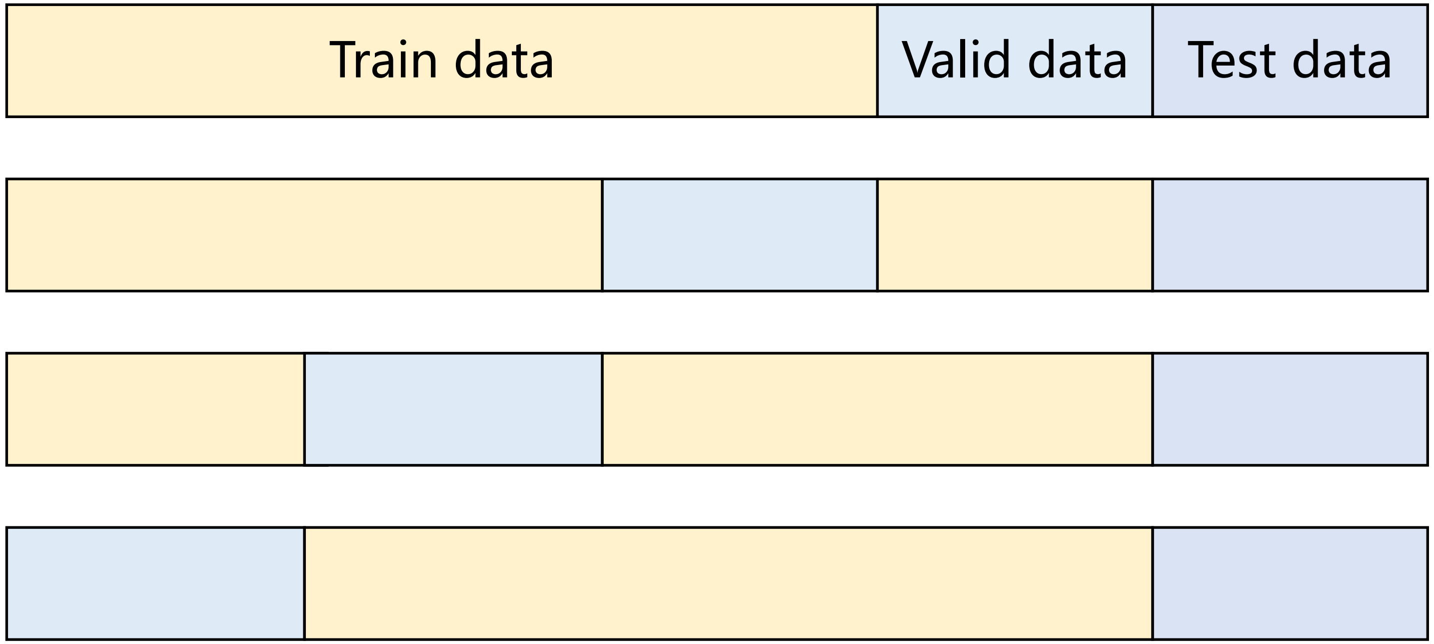
- 랜덤으로 생성된 **검증 데이터셋 하나는 편향된 결과**를 줄 수도 있음
- **검증 데이터셋 샘플들은 모델이 학습할 수 없음**
- 간단하게 **모든 데이터 샘플 한 개마다 검증**을 진행할 수 있음



K-fold cross validation (1)

K-fold 교차 검증

- LOOCV의 경우 **계산 비용이 매우 큰 단점**이 있음
- 이러한 문제를 해결하기 위해, **K개의 파트로 나누어 검증**을 진행하는 방법론
- Ex. 4-fold 교차 검증



K-fold cross validation (2)

Question

- **K 의 값이 커지면**, 어떤 것이 바뀔까?
  1. 학습 데이터의 수 ↑↓
  2. Bias 에러값 ↑↓, Variance 에러값 ↑↓
  3. 계산 비용 ↑↓

Regularization

정규화 손실 함수

- **모델의 복잡도가 커진다 == 모델의 파라미터 수가 많아진다**
- 모델의 복잡도가 커질수록, **과적합(over-fitting)**이 발생할 가능성이 커진다
- 복잡도가 큰 모델을 정의하고, 그 중 **중요한 파라미터만 학습**하면 안될까?
- **필요없는 파라미터 값을 0으로 만들자!**

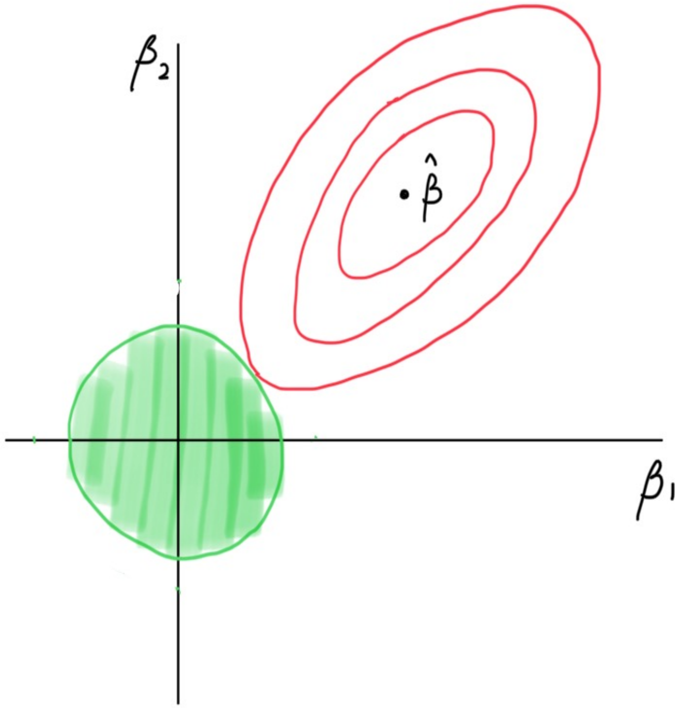
정규화 종류

- Ridge 회귀 (L2 regression)
- Lasso 회귀 (L1 regression)

Ridge Regression

Ridge Regression

- $\mathbf{L} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \left(\mathbf{y_i} - \left(\boldsymbol{\beta}_0 + \sum_{\mathbf{j=1}}^{\mathbf{D}} \boldsymbol{\beta}_{\mathbf{j}} \mathbf{x}_{\mathbf{ij}} \right) \right)^2 + \boldsymbol{\lambda} \sum_{\mathbf{j=1}}^{\mathbf{D}} \boldsymbol{\beta}_{\mathbf{j}}^2$
- MSE 손실을 줄이지 못하면 페널티 항의 손실값이 더 크게 작용함
- **λ (람다)**는 정규화의 영향을 조절하는 하이퍼파라미터
- 정규화 식이 **제곱의 합**으로 표현됨

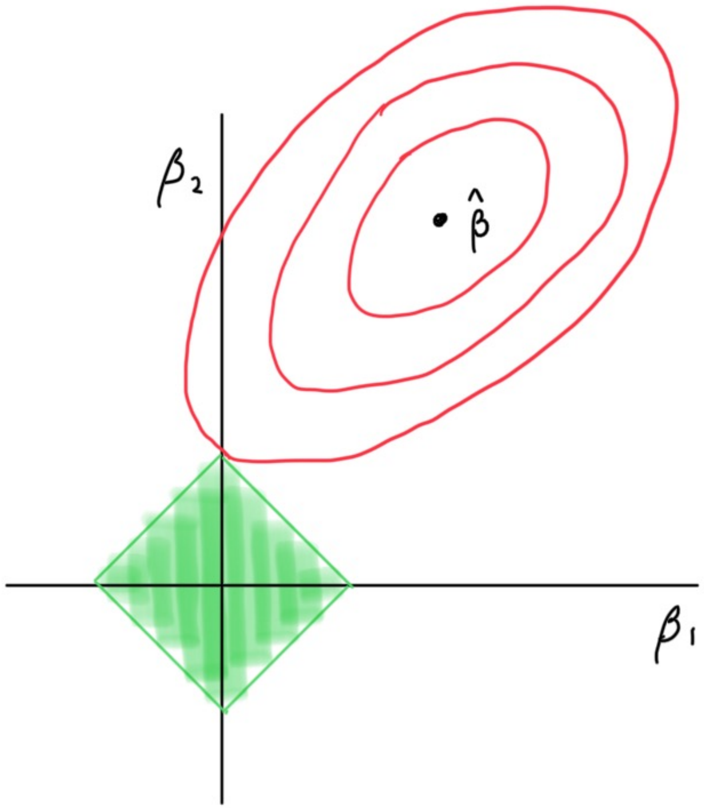


$$\beta_1^2 + \beta_2^2 \leq s$$

Lasso Regression

Lasso Regression

- $\mathbf{L} = \sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{D}\beta_j x_{ij}\right)\right)^2 + \lambda \sum_{j=1}^{D}|\beta_j|$
- MSE 손실을 줄이지 못하면 페널티 항의 손실값이 더 크게 작용함
- **λ (람다)**는 정규화의 영향을 조절하는 하이퍼파라미터
- 정규화 식이 **절댓값의 합**으로 표현됨



$|\boldsymbol{\beta}_1| + |\boldsymbol{\beta}_2| \le s$

Regularization

Question

- λ가 커지면, **Bias** 에러값↑↓, **Variance** 에러값↑↓
- 파라미터의 **희소성(sparsity)** 정도 : Ridge 정규화 Lasso 정규화
- 0의 값을 가진 파라미터가 더 많아지게 하는 방법?
- 위의 방법은 좋은 효과를 줄까?

Thank you

Introduction of Machine Learning & Regression Model