Linear & Ridge Regression → ML에서 갑자기 왜 Regression? ∑ Linear Regression, closed-form 해 有 (계산 버음)

*** Regression** – the Problem of <
- quantity of interest y : Response / dependent variable
- several observed variables $x_1, x_2, \cdots, x_n$: covariates, features, independent variables

ex) y: house
x_1: price
x_2: room number
x_3: age of house
⋮

$x_1, x_2, x_3 \cdots$ 컨디션을 **잘 예측** 해서 조건에 부합하는 best 집을 찾아보자!
(손해보지 않는 선택을 하겠다!!!)
회선의

1. 문제 세팅 (How the response determined?)

- data : $D = \{(y_1, x_1), (y_2, x_2), \cdots, (y_n, x_n)\}$, $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$
- function : $\left[\begin{array}{l} y_i = \langle x_i, \theta^* \rangle + z_i \\ y_i = h^*(x_i) + z_i \end{array}\right]$ z_i = noise , h^* = true underlying assumption

잘 예측 한 함수. 규칙. etc.

가정) $\mathcal{H}$는 벡터 θ에 의해 완전히 결정된다

- goal : 어떻게 하면 잘 예측 할수 있을까? ⇒ θ^*, h^* 찾기

* 문제 θ^*, h^* 에 어떤 힌트도 없음 → suppose : the function h is lies in hypothesis class $\mathcal{H}$,
the set of function $\mathcal{H}$ is parameterized by a vector θ.

e.g) Linear function : $\mathcal{H}_{linear} = \{ h_\theta(x) = \langle x, \theta \rangle : \theta \in \mathbb{R}^d \}$ 의미) 함수 h는 θ에 의해 결정된다.

θ는 수많은 data로 학습, 각 케이스에 대해 잘 or 잘 못 예측했니 비교해서 공통점을 찾아보자!

• Loss function (손실함수)

* how well the prediction $h_\theta(x)$ describes output y.

$$\hat{\theta} = \underset{\theta}{\arg\min} \sum_{i=1}^{n} \text{loss}(h_\theta(x_i), y_i)$$

loss가 가장 작은 곳을 찾자

1.1 Linear Regression

• Linear Model Assumption

$y_i = \langle x_i, \theta^* \rangle + z_i$, x_i : feature vector, θ^* : (우리가 찾는) parameter, z_i : 노이즈

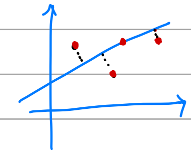
$\theta_0 +$ ⇒ $y_i = \theta_0 + \langle x_i, \theta^* \rangle + z_i$ $= \begin{bmatrix} \theta_0 \\ \theta \end{bmatrix} + \underbrace{[\ \ ]}_{x}\underbrace{[\ \ ]}_{\theta} + \underbrace{[\ \ ]}_{z_i} = \langle \tilde{\theta}, \tilde{x}_i \rangle + z_i$

• Loss function 데이터를 잘 설명하는 θ를 찾자 → 그럼 모델이 얼마나 잘 예측하는데? → 평가 지표 (metrics)를 만들어보자!

sum of squared errors $\hat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\langle x_i, \theta \rangle - y_i\right)^2$

$= \frac{1}{n} \|y - X\theta\|^2$

해석) 예측값과 실제값 사이의 오차를 제곱해서 더한 값
1) 오차 크면 큰 페널티 2) 미분가능 (최적화)



떨어진 안중의 제곱.
≈ l_2 – norm
(euklid)
점~선간의 거리.

• 추가)

- 기하학적 해석 : $X\hat{\theta}_{LS}$ 는 y를 X의 열공간에 직교 투영
- 추정값의 성질 : 1) Unbiased $E[\hat{\theta}_{LS}] = \theta^*$
  2) Variance ∝ $\sigma_{min}(X)^{-2}$ 즉, X의 singuar value 에 민감. → multicollinearity possible

1.2 Least Squares

$$\hat{R}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \langle x_i, \theta\rangle\right)^2 = \frac{1}{n}\|y - X\theta\|_2^2$$

- suppose) 1) $X \in \mathbb{R}^{n\times d}$ matrix
  2) full column rank → 열들이 선형독립! → X^TX invertible possible If not) ≠ singular matrix
  = 역행렬 없음
  → 계산 불가 → 해 없음.

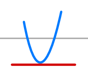
Proposition 1: If X has full rank, $\hat{\theta}_{LS} = (X^TX)^{-1}X^Ty$

1.2.1 Convex Optimization based on proof of Prop.1

→ Prop.1을 Convex Optimization 관점에서 증명해보자.
왜? 다른모델(logistic Reg, …)에도 확장가능.

Proposition 2. Optimality Condition (최적성 조건)

function f is convex & differentiable,
Consider a point x^* obeying $\nabla f(x^*) = 0$
} x^* is global minimizer



수식) $f(y) \geq f(x^*) + \underbrace{\langle y - x, \nabla f(x^*)\rangle}_{=0}$

$\rightsquigarrow f(y) \geq f(x^*)$ □

- Least Square 에 적용해보자.

- function $f(\theta) = \frac{1}{n}\|X\theta - y\|_2^2$
- $f(\theta)$ is convex, diff-ble
- Gradient) $n\hat{R}(\theta) = \langle X\theta - y, X\theta - y\rangle$
  $= \langle\theta, X^TX\theta\rangle - 2\langle\theta, X^Ty\rangle + \langle y, y\rangle$

- Linear Algebraic Proof (기하학적) $\hat{\theta}_{LS}$ 의 의미„

→ y를 $X\theta$로 가장 가깝게 근사한 Projection (직교 투영)
- $X\theta$는 X의 column vector 위에 존재
- 우리는 y에 가장 가까운 벡터 $X\hat{\theta}$를 찾고싶어요

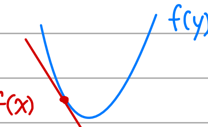
수식)
- $X \in \mathbb{R}^{n\times d}$. has full column rank,
- $U \in \mathbb{R}^{n\times d}$, 직교 열 벡터 column orthogonal
- $\Sigma \in \mathbb{R}^{d\times d}$, 대각행렬 singular value $\sigma_1, \sigma_2, \dots, \sigma_d > 0$
- $V \in \mathbb{R}^{d\times d}$, 정규직교행렬 . orthogonal

$X = U\Sigma V^T$

☆
1. Linear Regression 해가 존재하나?
2. 있다면 어떻게 구하는데?
3. 이게 왜 좋은/나쁜 해일까?

- Definition 1: What is 볼록 convex?
어떤 미분가능한 함수 f가 다음 조건을 만족하면 convex 하다고 함.

$f(y) \geq f(x) + \langle y - x, \nabla f(x)\rangle$



$\langle y-x, \nabla f(x)\rangle + f(x)$
→ 그래프가 접선보다 위에있는 함수

$\nabla n\hat{R}(\theta) = 2X^TX\theta - 2X^Ty$

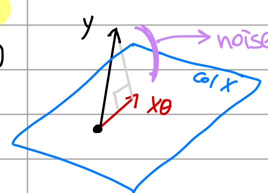
$\nabla\hat{R}(\hat{\theta}_{LS}) = 0$

$\Rightarrow 0 = \cancel{2}X^TX\hat{\theta}_{LS} - \cancel{2}X^Ty$

$\Rightarrow X^TX\hat{\theta}_{LS} = X^Ty$

$\Rightarrow \hat{\theta}_{LS} = (X^TX)^{-1}X^Ty$ □

*그럼 결론?
"gradient = 0"인 경우 ⇒ 해! 라고 지칭가능.



→ noise $= y - X\theta$의 길이 $= \|y - X\theta\|_2$ 최소화
Norm 2

$U^TU = I$

$\hat{\theta}_{LS} = (X^TX)^{-1}X^Ty$

- $X^TX = (U\Sigma V^T)^T(U\Sigma V^T) = (V\Sigma^TU^T)(U\Sigma V^T)$
  $= V\Sigma^T\Sigma V^T = V\Sigma^2V^T$

$V^{-1} = V^T$

$\Rightarrow (X^TX)^{-1} = (V\Sigma^2V^T)^{-1} = V\Sigma^{-2}V^T$

- $X^Ty = (U\Sigma V^T)^Ty = V\Sigma^TU^T$

$\Rightarrow \hat{\theta}_{LS} = V\Sigma^{-2}V^T \cdot V\Sigma^TU^Ty = V\Sigma^{-1}U^Ty$

- $X\hat{\theta}_{LS} = U\Sigma V^T \cdot V\Sigma^{-1}U^Ty = UU^Ty$

$X\hat{\theta}_{LS} = UU^Ty$ 의 의미) UU^T는 X의 colum space 에 대한 projection matrix

Ridge Regression

Idea) LR에서 $\hat{\theta}_{LS}$가 과하게 변하거나, 과하게 적합되는 문제를 해결하기 위해 해를 좀 더 제약해 안정적인 추정을 해보자

2.1 Ridge Regression Estimate

- $\hat{\theta}_{LS} = \text{argmin} \|y - X\theta\|_2^2$ (기존)
- 아주 큰 $\|\theta\|$는 예측을 불안하게 해. → $\|\theta\|$를 컨트롤해보자
  → $\|\theta\|$에 자체 패널티

$\lambda > 0$: reg. parameter
(크면 더 강한 제약)

$$\hat{\theta}_{ridge} = \underset{\theta}{\text{argmin}} \underbrace{\|y - X\theta\|_2^2}_{\text{fitting error}} + \underbrace{\lambda \|\theta\|_2^2}_{\text{regularization penalty}}$$

- 여전히 convex, diff-able ⇒ ∃ closed-form 해.

$$\hat{\theta}_{ridge} = (X^TX + \lambda I)^{-1} X^T y$$

SVD 기반 해석) shrinkage 관점.

- X를 SVD, $X = U\Sigma V^T$.
- $$X\hat{\theta}_{ridge} = \sum_{i=1}^{d} \boxed{\frac{\sigma_i^2}{\sigma_i^2 + \lambda}} \langle y, u_i \rangle u_i$$

shrink factor $\in (0, 1)$

→ 해석) 각 방향 u_i에 대해 shrink factor를 곱한다.
 즉, data가 약한 방향 (작은 σ_i)에서는 bias를 많이 준다
 → 노이즈에 덜 민감 (더 정확해짐)

$$\begin{aligned}
X\hat{\theta}_{ridge} &= X(X^TX + \lambda I)^{-1} X^T y \\
&= U\Sigma V^T (V\Sigma^T U^T U \Sigma V^T + \lambda VV^T)^{-1} V\Sigma U^T y \\
&= U\Sigma V^T (V\Sigma^2 V^T + \lambda VV^T)^{-1} V\Sigma U^T y \\
&= U\Sigma V^T (V(\Sigma^2 + \lambda I)V^T)^{-1} V\Sigma U^T y \\
&= U\Sigma V^T V(\Sigma^2 + \lambda I)^{-1} V^T V\Sigma U^T y \\
&= U\Sigma \underbrace{(\Sigma^2 + \lambda I)}_{\text{diagonal matrix with } \sigma_i^2 + \lambda}{}^{-1} \Sigma U^T y
\end{aligned}$$

- ⇒ $(\Sigma^2 + \lambda I)^{-1}$ = diagonal matrix with $\frac{1}{\sigma_i^2 + \lambda}$
- Σ = diagonal matrix with σ_i

⇒ $\Sigma(\Sigma^2 + \lambda I)^{-1}\Sigma = \text{diag}\left(\frac{\sigma_1^2}{\sigma_1^2 + \lambda}, \cdots, \frac{\sigma_i^2}{\sigma_i^2 + \lambda}\right)$

⇒ $UU^T y = \sum u_i \langle u_i, y \rangle = \sum \langle y, u_i \rangle u_i$

2.2 Bias-Variance-Trade off

- Ridge Reg. 추정값에 bias를 일부 도입 → Variance 줄임.

factor	Least Squares	Ridge Regression
Bias	low (0)	high
Variance	大	小
예측 오차	overfitting possible	control with λ

최적 λ 선택방법
- $\lambda = 0$: 일반 LS (Bias 0, Var ↑)
- $\lambda \to \infty$: $\hat{\theta}_{ridge} \to 0$ (극단적 단순화)
- 보통 cross-validation 사용

$P_{data}(y|X)$: true underlying distribution (observation)

$P_{model}(y|X,\theta)$: Parametric family of distribution

Optimization: Maximum Likelihood Estimate

- A method of estimating the parameters of a statistical model given observations by finding the parameter values that maximize the likelihood of making the observations given the parameters

  $\theta_{ML} = \arg\max_\theta P_{model}(y|X,\theta)$

- Approach

$$\boldsymbol{\theta}_{ML} = \arg\max_{\theta} \prod_{i=1}^{n} p_{model}(y_i|\boldsymbol{x_i},\boldsymbol{\theta})$$

iid → training samples are independent, generated by same probability distribution (i.i.d)

- We can replace the product by applying the logarithmic property $log_c(ab) = log_c(a) + log_c(b)$:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\theta} \sum_{i=1}^{n} \log p_{model}(y_i|\boldsymbol{x_i},\boldsymbol{\theta})$$

what shape does our probability distribution have?

$y_i = \underbrace{N(x_i\theta, \sigma^2)}_{\text{Gaußian}} = \underbrace{x_i\theta}_{\text{mean}} + N(0,\sigma^2)$

* 뒤장 *

- Assuming Gaussian distribution, we get the same result for the optimization as for Linear least squares

$$\boldsymbol{\theta} = (\boldsymbol{X^T X})^{-1}\boldsymbol{X^T y}$$

$* \ P(y_i|x_i,\theta) = (2\pi\sigma^2)^{-\frac{1}{2}} \cdot e^{-\frac{1}{2\sigma^2}(y_i - x_i\theta)^2}$

Assuming $y_i = N(x_i\theta, \sigma^2) = x_i\theta + N(0,\sigma^2)$

mean

cf) Gaußian:

$P(y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(y_i - \mu)^2} \quad y_i \sim N(\mu, \sigma^2)$

original optimization problem

$\theta_{ML} = \arg\max_\theta \sum^n \log P_{model}(y_i|x_i,\theta)$

$\log\left[(2\pi\sigma^2)^{-\frac{1}{2}} \cdot e^{-\frac{1}{2\sigma^2}(y_i - x_i\theta)^2}\right]$

$\sum_{i=1}^n \log\left[(2\pi\sigma^2)^{-\frac{1}{2}} \cdot e^{-\frac{1}{2\sigma^2}(y_i - x_i\theta)^2}\right]$

$= \sum_{i=1}^n -\frac{1}{2}\log(2\pi\sigma^2) + \sum_{i=1}^n \left(-\frac{1}{2\sigma^2}\right)(y_i - x_i\theta)^2$

$= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_i - X\theta)^T(y - X\theta)$

$\frac{\partial J(\theta)}{\partial \theta} \stackrel{!}{=} 0 \Rightarrow \theta = (X^TX)^{-1}X^Ty$

어떤 라면을 먹었는데 너무 맛있어 근데 어떤 브랜드인지 몰라.
각 브랜드마다 스프 비율이 다르고, 어떤 스프비율이 가장 맛있는 건지 비율을 찾아보자!

→ 여러번 라면을 시식하면서 data를 만들었어.

번호	스프비율	느낀 맛
⋮	⋮	⋮

목표: 진짜 스프 비율을 찾자. = 맛(y)이 스프(x)에 따라 어떻게 정해졌는지 추정해보자.

① Least squares ② Maximum Likelihood ....

① Leas Squares (기하학관점)

- 정답 θ는 모르지만
- θ를 넣어서 예측한 맛 $\hat{y}_i = x_i\theta$ 이 실제 맛 y_i 와 얼마나 비슷한지 보자.
  → 오차를 제곱해서 전부 더한게 가장 작은 게 best.

Interpretation

맛(y)이 얼마나 예측과 가까운지를 기준으로 θ를 찾자

② Maximum Likelihood

- data는 확률 model에서 나왔을 거야. $y_i \sim N(\theta x_i, \sigma^2)$
- "어떤 θ를 사용했을때 실제 맛(y_i)이 나올 확률"이 가장 높은 걸 찾자.

$\hat{\theta}_{MLE} = \arg\max_\theta \prod_{i=1}^n N(y_i | \theta x_i, \sigma^2)$

$P(A|B) = \frac{P(A \cap B)}{P(B)}$ ⇒ B가 일어났을때 그안에서 A가 일어날확률

* Interpretation *

지금 맛 본 결과들이 실제로 나올 확률이 가장 높은 θ를 찾자.
= 가장 그럴듯한 θ를 찾자.

- why? $y = ax + b$, ...

LS가 직관적인데 "왜 그 공식을 사용하는지"의 근거 부족
MLE는 "data가 왜 이렇게 나왔는가"를 확률적으로 설명가능

3.3 Bias-Trade-off

Ridge Regression 이 왜 일반화 성능이 좋은지 수학적으로 알아보자.
예측 오차를 $Bias^2$ + Variance + Noise 로 분해해서 어떻게 Variance를 줄여주는지 분석해보자

- 문제 설정 - Prediction Risk of $\hat{\theta}$

→ 새로운 샘플 $x \in \mathbb{R}^d$ 에 대해 예측할때

- Model : $y = h(x) + z$, $z \sim N(0, \sigma^2)$
- Dataset : $D = \{(y_1, x_1), \cdots (y_n, x_n)\}$
- 예측값 : $\hat{y} = \hat{h}(x) = \langle \hat{x}, \hat{\theta}_{ridge} \rangle$
- 실제값 : $y = \langle x, \theta^* \rangle + z$, $z \sim N(0, \sigma^2)$

(예측 리스크)
Prediction Risk $R(\hat{h}) = E_{x,y}\left[(\hat{h}(x) - y)^2\right]$
+ Dataset random (학습)

$$\Rightarrow E_D\left[R(\hat{h}_D)\right] = E_{x,y,D}\left[(\hat{h}_D(x) - y)^2\right]$$

- Goal : $E_{x,y,D}$ 를 Bias, Variance, Noise 로 분해.

- 수식)

$$E_D[R(\hat{h}_D)] = E_{x,y,D}[(\hat{h}_D(x) - y)^2] \overset{*}{=} E_D[(\hat{h}_D(x) - h(x) - z)^2]$$

$y = h(x) + z$

$$= E_D[(\hat{h}_D(x) - h(x))^2] + \underbrace{2E[(\hat{h}_D(x) - h(x))z]}_{} + E[z^2]$$

$$= E_D[(\hat{h}_D(x) - h(x))^2] + E[z^2]$$

z has zero mean.

$$E_D[(\hat{h}_D(x) - h(x))^2] = E_D[(\hat{h}_D(x) - E[\hat{h}_D(x)] + E[\hat{h}_D(x)] - h(x))^2]$$

$$= E_D[(h_D(x) - E[\hat{h}_D(x)])^2] - 2\underbrace{(E_D[\hat{h}_D(x)] - E_D[h(x)])}_{=0,\ \because \text{ 기댓값} \approx \text{평균}}(E[\hat{h}_D(x)] - h(x)) + E[(E[\hat{h}_D(x)] - h(x))^2]$$

$$= E_D[(h_D(x) - E[\hat{h}_D(x)])^2] + E[(E[\hat{h}_D(x)] - h(x))^2]$$

Thus we have,

$$E_D[R(\hat{h})] = E_D[(\hat{h}_D(x) - h(x))^2] + E[z^2]$$

$$= \underbrace{E[(E[\hat{h}_D(x)] - h(x))^2]}_{Bias^2} + \underbrace{E_D[(h_D(x) - E[\hat{h}_D(x)])^2]}_{Variance} + \underbrace{E[z^2]}_{Noise}$$

$Bias^2$ = 평균모델과 진짜함수 $h(x)$ 간의 거리

Variance : 학습 Dataset 변화에 따른 예측의 흔들림

Noise : 예측 불확실성

*직관적 이해...

- Bias가 크다 → 모델이 단순해서 진짜 패턴 못 따라감 (underfitting)
- Variance가 크다 → 데이터셋이 조금 바뀌면 예측이 크게 바뀜 (overfitting)
- Noise는 피할 수 없음

*왜 중요한가? 모델 선택 & Hyperparameter 튜닝의 근거.

Next chapter