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

* **Regression** – the Problem of < quantity of interest y : Response / dependent variable
several observed variables $x_1, x_2, \dots, x_n$: 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!
(손해 보지 않는 선택을 하겠다!!!)
최선의

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.

- 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.

가정) $\mathcal{H}$는 벡터 θ에 의해 완전히 결정된다

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.

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

loss가 가장 작은 곳을 찾자

1.1 Linear Regression

- **Linear Model Assumption**

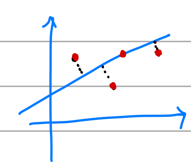
$y_i = \langle x_i, \theta^* \rangle + z_i$, x_i : feature vector, θ^*: (우리가 찾는) parameter, z_i : 노이즈

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

- **Loss function** 데이터를 잘 설명하는 θ를 찾자 → 그럼 모델이 얼마나 잘 예측하는데? → 평가 지표 (metrics)를 만들어 보자!

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) 미분 가능 (최적화)



떨어진 만큼의 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 $\propto \sigma_{min}(X)^{-2}$ 즉, X의 singular value 에 민감. → multicollinearity possible

1.2 Least Squares

☆

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

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

• 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

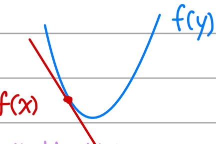
→ Prop.1을 Convex Optimization 관점에서 증명해보자.

왜? 다른모델(logistic Reg, …)에도 확장가능.

• Definition 1: What is 볼록 convex?

어떤 미분가능한 함수 f가 다음 조건을 만족하면 convex 하다고 함.

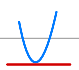
$$f(y) \ge f(x) + \langle y-x, \nabla f(x)\rangle$$



→ 그래프가 접선보다 위에있는 함수

Proposition 2. Optimality Condition (최적성 조건)

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



기울기=0.

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

$\sim f(y) \ge f(x^*)$ □

• Least Square 에 적용해보자.

• function $f(\theta) = \frac{1}{n}\|X\theta - y\|_2^2$

• $f(\theta)$ is convex, diff-ble

• Gradient) $n\hat R(\theta) = \langle X\theta - y, X\theta - y\rangle$

$= \langle\theta, X^TX\theta\rangle - 2\langle\theta, X^Ty\rangle + \langle y, y\rangle$

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

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

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

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

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

* 그럼 결론?

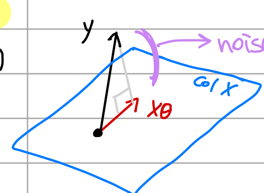
"gradient = 0"인 경우 ⇒ 해! 라고 지칭가능.

• Linear Algebraic Proof (기하학적) $\hat\theta_{LS}$ 의 의미,,

→ y를 Xθ로 가장 가깝게 근사한 Projection (직교 투영)

• Xθ는 X의 column vector 위에 존재

• 우리는 y에 가장 가까운 벡터 $X\hat\theta$를 찾고싶어요



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

Norm 2

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

$X = U\Sigma V^T$

$\hat\theta_{LS} = (X^TX)^{-1}X^Ty$ $U^TU = I$

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

$= V\Sigma^T\Sigma V^T = V\Sigma^2V^T$ $V^{-1} = V^T$

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

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

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

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

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

• SVD (특이값 분해)

$$X = \underbrace{U}_{\text{출력}}\overset{\text{스케일}}{\Sigma}\underbrace{V^T}_{\text{입력축 회전}}$$

입력공간을 회전해서 크기 조정을해서 출력공간으로 회전하자.

1) V^T : orthogonal matrix → 회전

$V^T = V^{-1}$ 역행렬이 전치행렬

역할 : 입력공간을 new basis $v_1, v_2, \cdots, v_d$ 로 회전.

2) Σ : diagonal matrix 곱셈같은거.

$$\begin{bmatrix} \sigma_1 & & & & 0 \\ & \sigma_2 & & & \\ & & \sigma_3 & & \\ & & & \ddots & \\ 0 & & & & \sigma_n \end{bmatrix}$$

- $\sigma_i > 0$: 스케일 크기
- 역할 : 회전된 입력을 각 방향으로 늘리거나 줄이기.

ex) x 방향 2배, y 방향 0.5배.. $\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 2 & & \\ & 0.5 & \\ & & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix}$

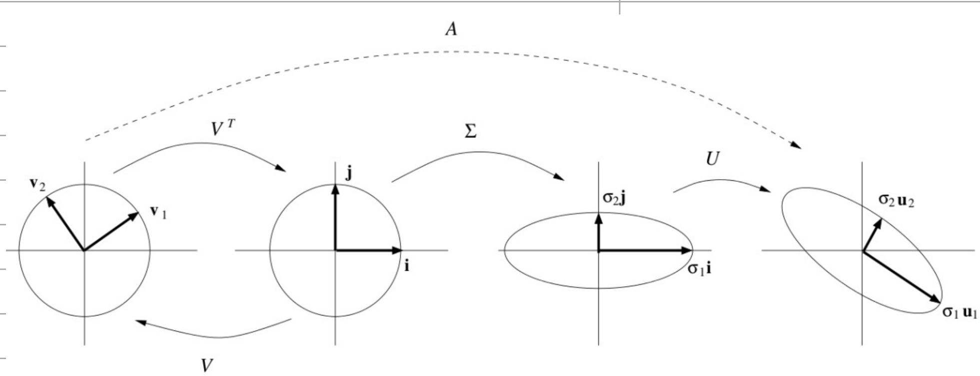
3) U : 출력 공간의 회전

- $U \in \mathbb{R}^{n\times d}$, $u_1, u_2, \cdots, u_d$ orthogonal
- $UU^T = I$: $u_1, u_2, \cdots, u_d$ 서로 직교
- 역할 : 스케일된 벡터를 출력 공간에서 새 방향으로 회전

ex) $U^TU = \begin{bmatrix} u_1^T \\ u_2^T \\ u_3 \end{bmatrix}\begin{bmatrix} u_1^T & u_2^T & u_3^T \end{bmatrix} = \begin{bmatrix} u_1^Tu_1 & u_1^Tu_2 & u_1^Tu_3 \\ u_2^Tu_1 & u_2^Tu_2 & u_2^Tu_3 \\ u_3^Tu_1 & u_3^Tu_2 & u_3^Tu_3 \end{bmatrix} = I$ □

각 열벡터 서로 직교
$= u_1 \perp u_2,\ u_2 \perp u_3,\ u_3 \perp u_1$

ex)



1.3 Analysis of LS estimate & Co-linearity

그럼 우리가 지금까지 구한 해 $\hat{\theta}_{LS}$ 는 항상 좋은가?

→ 아닐 수도 있음. ∵ 현실) features 상관관계 ↑ ⇒ co-linearity 문제

⇒ $\hat{\theta}_{LS}$ 의 Variance (분산) ↑, overfitting 발생

어떻게 해결?

→ Ridge Regression.

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

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

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

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

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

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

SVD 기반 해석) shrinkage 관점.

- X를 SVD, $X = U\Sigma V^T$.
- $$X\hat{\theta}_{ridge} = \sum_{i=1}^{d} \underbrace{\boxed{\frac{\sigma_i^2}{\sigma_i^2 + \lambda}}}_{\text{shrink factor} \in (0,1)} \langle y, u_i \rangle u_i$$

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

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

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

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

→ 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)

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
예측오차	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

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)

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

$*$ $y = h(x) + z$

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

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

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 ...**

- Bias 가 크다 → 모델이 단순해서 진짜 패턴 못 따라감 (underfitting)
- Variance 가 크다 → 데이터셋이 조금 바뀌면 예측이 크게 바뀜 (overfitting)
- Noise 는 피할 수 없음

* 왜 중요한가? 모델 선택 & Hyperparameter 튜닝의 근거.

Next chapter