Exploring Modern Regression

From Classic Through Deep Learning and Beyond

DMQA Open Seminar

한경석

hanks6125@korea.ac.kr

2024.11.22

Index

Introduction

○ **한경석 (Han Kyungseok)**

- 고려대학교 산업경영공학과 석사 과정 (2024.03 ~ present)
- Data Mining & Quality Analytics Lab **(김성범 교수님)**

○ Research Interest

- Regression
- Deep Learning

○ E-Mail

- hanks6125@korea.ac.kr

Background

Regression?

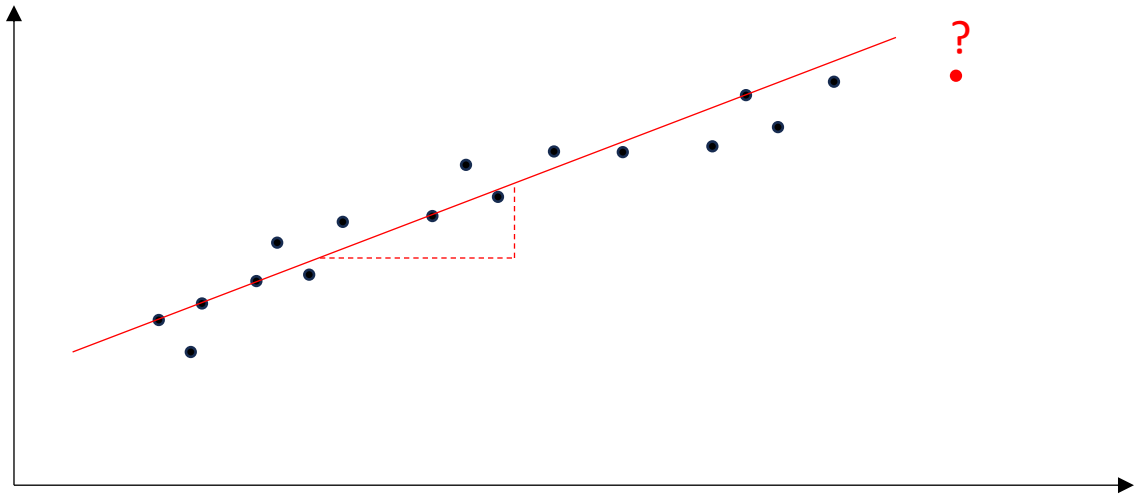
○ **Regress** : **Return to former** or less developed state , 되돌아감, 퇴보

○ 독립변수와 종속 변수 사이의 **관계**를 모델링하는 방법

○ 일반적으로 Classification이 Discrete classes를 예측하는 것과 달리 Regression은 **Continuous Target**을 예측

○ 회귀 모델은 다양한 분야에서 중요한 역할을 수행할 수 있음

- Prediction : 주어진 과거의 데이터로부터 미래의 값을 예측
- Quantifying Relationships : 독립변수가 변할 때, 종속 변수가 얼마나 변할지를 정량적으로 확인
- Detecting trends and patterns : 회귀 Line (혹은 Curve) 으로부터 추세나 패턴을 확인



Background

OLS (Ordinary Least Squares)

○ Regression 에서의 가장 **핵심 요소는 OLS**

- **최소자승법**은 1805년 Legendre에 의해 최초로 제안되었고, 1821년 Gauss가 이를 더욱 발전시켜 **일반최소자승법**, 즉 **OLS**를 공식화함.
- OLS를 사용하면 불편 추정량(Unbiased Estimator)를 얻을 수 있음. 즉, 입력 데이터와 출력 데이터간의 관계를 선형으로 가정할 때, 선형 회귀 모델의 계수 β̂ 을 추정할 수 있고 이 계수의 기대값은 아래를 만족

$$E[\hat{\beta}] = \beta$$

β : 모집단의 계수

Background

Assumption Needed

○ 이와 같은 불편추정량을 얻으려면 Gauss-Markov 정리에 의해 아래 6가지 가정을 만족해야함

① Linearity : X와 y는 선형관계이다. 즉 $y = X^2$ 과 같은 비선형 관계가 아니다.

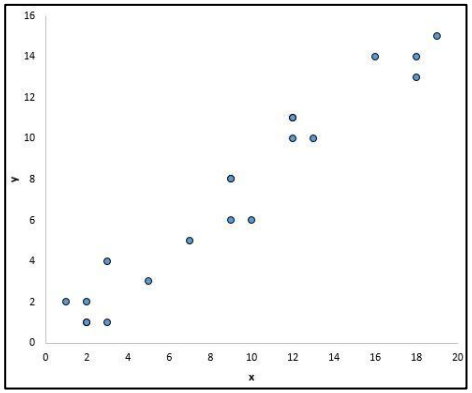
② Constant Error Variance : 잔차는 등분산이다. 즉 특정 구간에서 오차가 크거나 작게 나타나지 않고 일정하다.

③ Independent Errors : 잔차는 독립적이어야 함. 즉, 첫번째 샘플의 잔차가 두번째 샘플에 영향을 주면 안됨

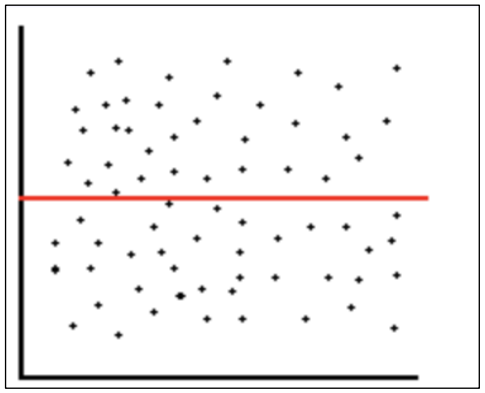
④ No Multicollinearity : 독립변수는 서로에게 영향을 주면 안됨.

⑤ Normality : 잔차는 평균 0인 정규분포를 따른다.

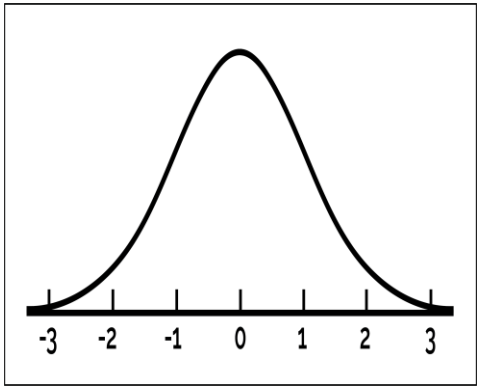
⑥ Exogeneity : 독립 변수와 오차항은 서로 상관이 없어야함, 즉 $E[X \mid \varepsilon] = 0$



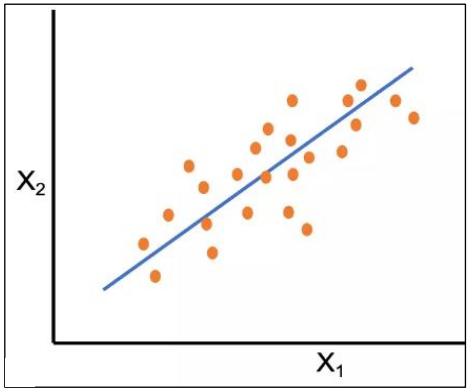
[Linearity]



[Constant Error variance]



[Normality]



[No Multicollinearity]

Background

Assumption Needed

$$\boldsymbol{E[\hat{\beta}] = \beta}$$

Normal Equation의 해 $\hat{\beta} = (X^\intercal X)^{-1} X^\intercal Y$

$= (X^\intercal X)^{-1} X^\intercal (X\beta + \epsilon)$

$= (X^\intercal X)^{-1} X^\intercal X\beta + (X^\intercal X)^{-1} X^\intercal \epsilon$

$= \beta + (X^\intercal X)^{-1} X^\intercal \epsilon \qquad \because A^{-1}A = AA^{-1} = I$

$E[\hat{\beta}] = E[\beta + (X^\intercal X)^{-1} X^\intercal \epsilon]$

$= \beta + E[(X^\intercal X)^{-1} X^\intercal \epsilon]$

$= \beta + E[(X^\intercal X)^{-1} X^\intercal] * E[\epsilon] \qquad \because$ 잔차와 독립변수 x는 독립

$= \beta + E[(X^\intercal X)^{-1} X^\intercal] * 0 \qquad \because$ 잔차의 기대값은 0

$\therefore \boldsymbol{E[\hat{\beta}] = \beta}$

Classic

Regression Models

○ Ridge

- High Variance를 극복하고자 함
- 회귀 계수에 L2 Penalty를 부여하여 Variance를 줄이고자 함

$$J(\theta) = \frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2 + \frac{\lambda}{2}||\theta_i||^2$$

○ Lasso

- Ridge's poor performance on outliers
- 회귀 계수에 L1 Penalty를 부여, Ridge와 달리 일부 계수를 0으로 만들어버리므로 자동으로 feature selection을 수행하게 됨.
- Ridge는 모든 계수가 살아 있으므로 outlier에 영향을 받음

$$J(\theta) = \frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2 + \lambda||\theta_i||$$

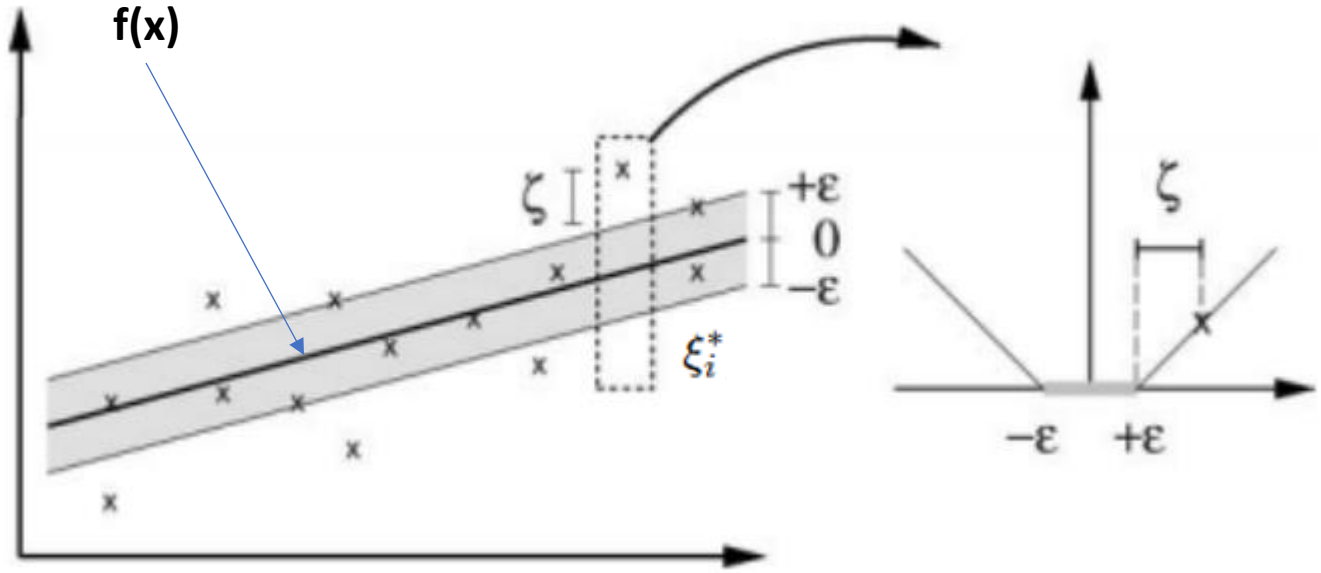
Classic

Regression Models

○ Support Vector Regression

- SVM의 아이디어에서 출발, 초평면이 곧 회귀선(Regression Line)임.
- 허용 오차 ±ε 내에 있는 오차는 0으로 간주하고, 벗어나는 데이터만 손실함수에 기여하도록 함

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$



Classic

Regression Models

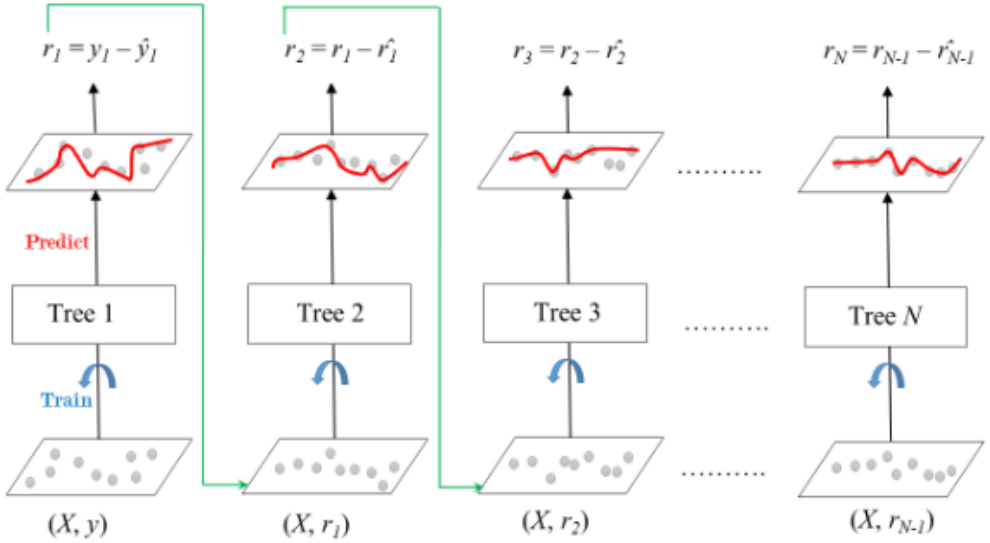
○ Random Forest Regression

- 여러 개의 결정 트리를 조합하는 Ensemble 모델. 각각의 트리로부터 예측값을 얻고, 평균값을 사용
- Regression 문제는 분산을 불순도로 삼고, 부모 노드와 자식 노드간의 분산 감소량이 크도록 학습함

$$\text{Reduction} = \text{Variance}_{\text{parent}} - (w_{\text{left}} \cdot \text{Variance}_{\text{left}} + w_{\text{right}} \cdot \text{Variance}_{\text{right}})$$

○ Boosted Regression Tree

- 오류를 점진적으로 줄여가는 Tree 기반 모델.



Classic

Regression Models

○ **ElasticNet**

- L1, L2 penalty를 결합

○ **Least Angle Regression**

- Iterative하게 X_k 를 추가하면서 상관관계가 큰 변수만을 선택

○ **RANSAC (RANdom SAmple Consensus) regression**

- Iterative model. threshold ε 를 정해놓고, 일부데이터로 모델 M을 Fitting 하여 ε 내에 있는 Data를 Inlier에 포함. 가장 많은 Inlier를 포함하는 모델 M을 찾음

○ **Huber Regression**

- 이상치에 민감하지 않도록 손실함수를 변환

$$HuberLoss = \begin{cases} \frac{1}{2}a^2 & ,if|a| \leq \delta \\ \delta(|a| - \frac{1}{2}\delta) & otherwise \end{cases}$$

Classic

Regression Models

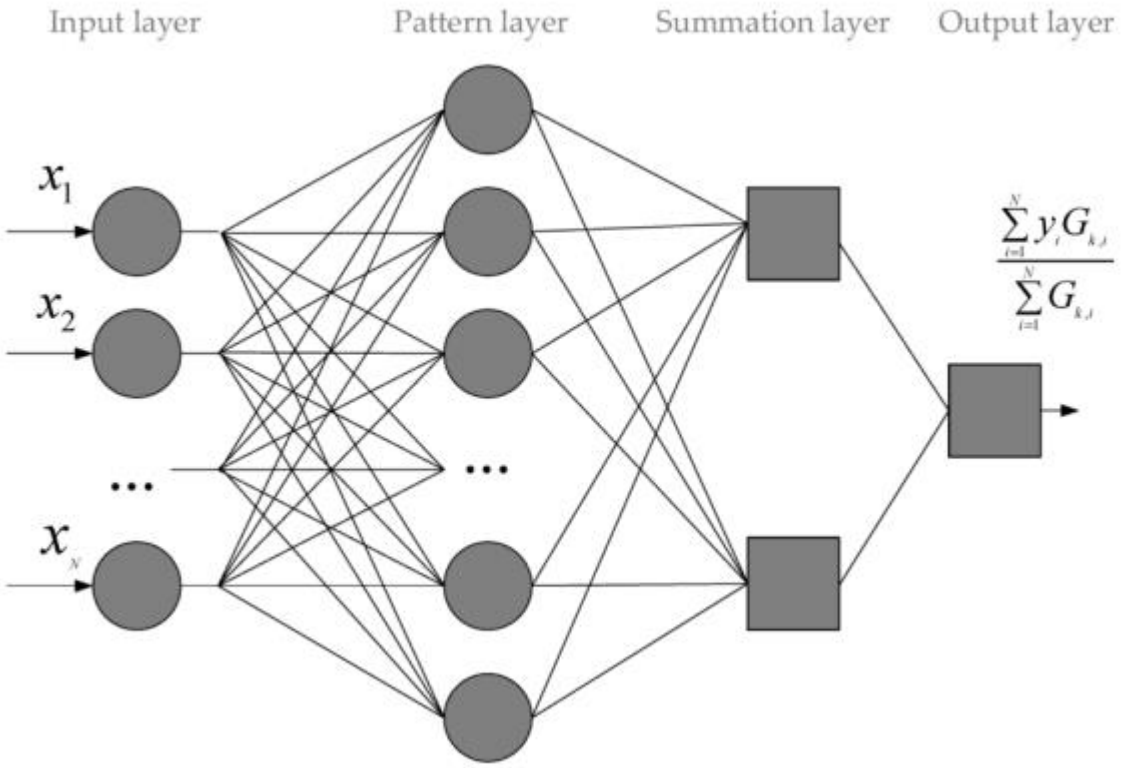
구분	개요
Multivariate Adaptive Regression Splines	데이터를 다양한 구간으로 나누어 비선형 관계를 추정
Polynomial Regression	독립 변수에 비선형 변환을 적용, 비선형성을 해결
Weighted Least squares	오차가 등분산성을 위반할 때, 오차에 가중치를 부여해서 해결
Generalized Least squares	오차간 상관관계가 존재할 때, 선형 변환을 통해 해결
Bayesian Regression	사전 분포를 활용해 계수를 추정, 데이터의 사전 지식을 반영
Quantile Regression	분위수에 따라 손실함수를 설정하여 데이터의 다양한 분포를 반영
Ordinal Regression	서열 데이터를 예측하는 회귀방법

Relationship with Deep Learning

General Regression Neural Network

○ **1991년, Donald F. Specht에 의해 제안된 뉴럴 네트워크**

- 회귀, 분류에 사용가능, Online dynamic system에 적합한 Structure
- 학습 parameter가 없음 **(Non-parametric)**

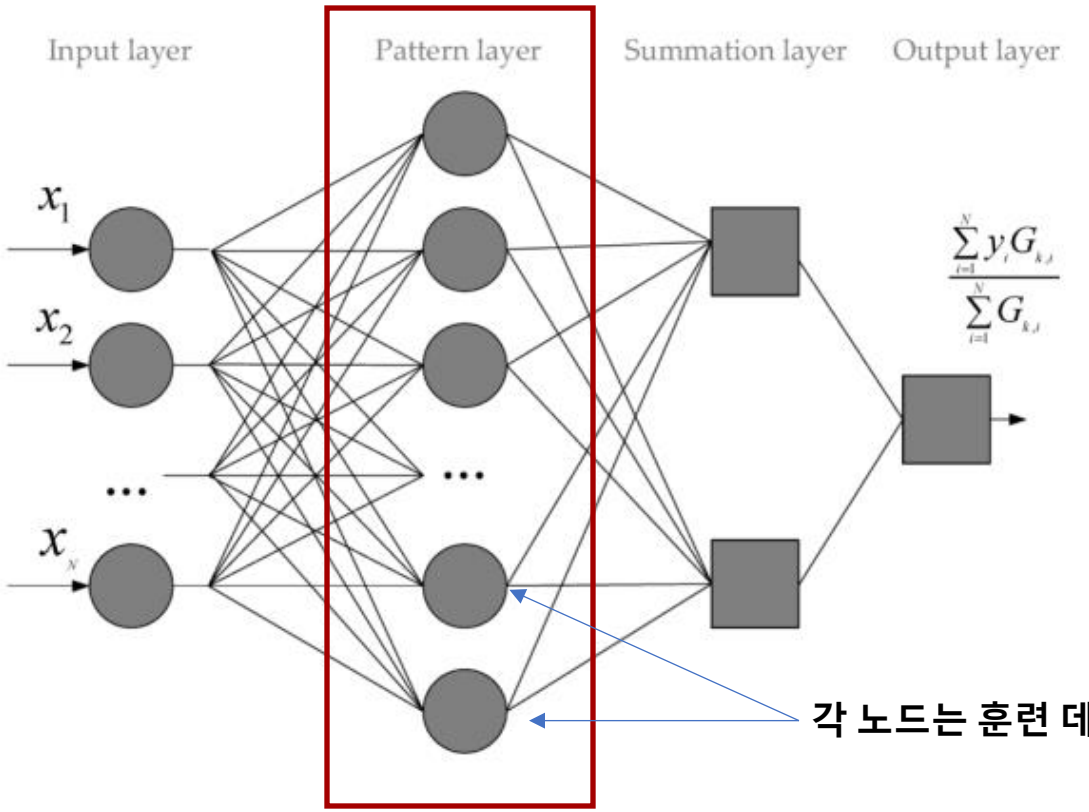


Relationship with Deep Learning

General Regression Neural Network

○ GRNN은 학습 데이터를 메모리에 전부 가지고 있는 상태에서 시작

○ 새로운 데이터 x 가 들어오면 메모리에 저장된 학습 데이터와의 유사도를 측정



각 노드는 훈련 데이터의 특정 Point를 의미

Relationship with Deep Learning

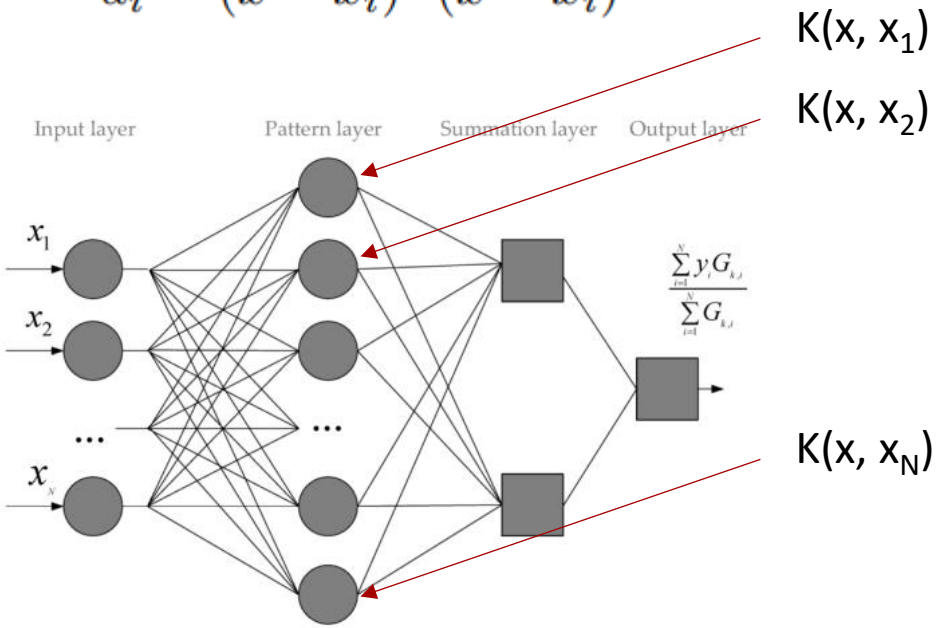
General Regression Neural Network

○ 유사도는 아래의 **커널 함수를 사용**

- 즉 새로운 데이터 포인트 x와 학습데이터의 i번째 데이터간의 유클리드 거리에 따라, 거리가 가까우면 큰 값을, 멀면 작은 값을 가지도록 함

$$K(x, x_i) = e^{-d_i/2\sigma^2}$$

$$d_i = (x - x_i)^T (x - x_i)$$



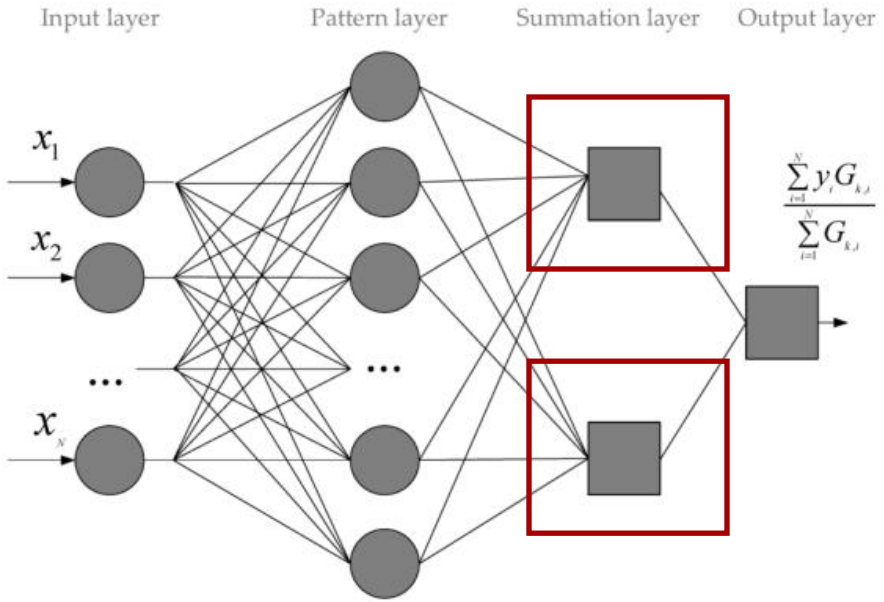
Relationship with Deep Learning

General Regression Neural Network

○ Summation layer는 2개의 노드로 이루어지며, 각각 분자 분모 노드임

분자 노드 : $S = \sum_{i=1}^{N} y_i K(x, x_i)$

분모 노드 : $D = \sum_{i=1}^{N} K(x, x_i)$

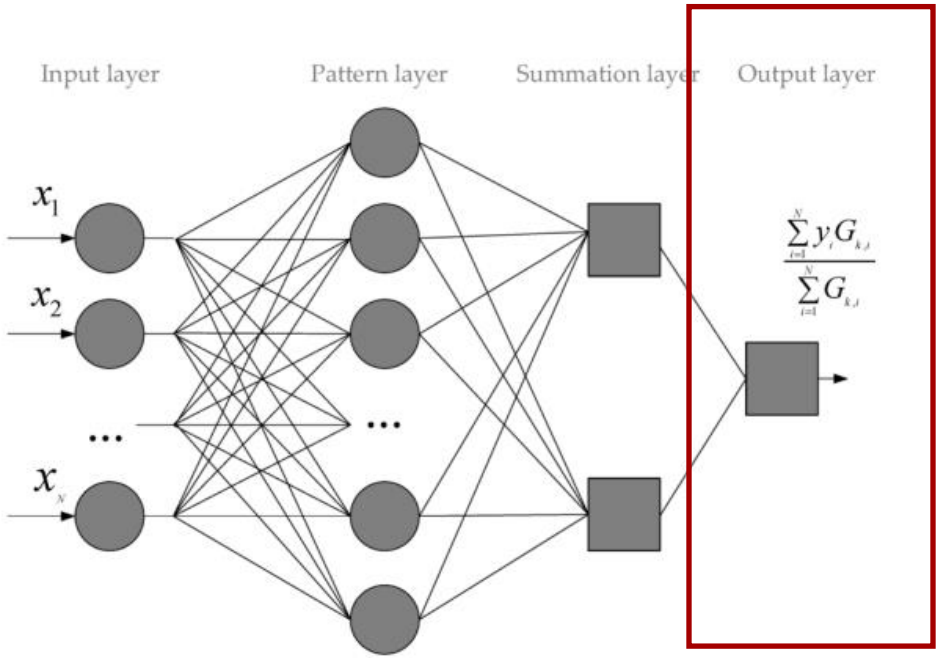


Relationship with Deep Learning

General Regression Neural Network

○ 최종 출력은 아래와 같이 학습 데이터와 새로운 입력간의 유사도를 기반으로 한 가중평균임

$$\hat{y}(x) = \frac{S}{D} = \frac{\Sigma_{i=1}^{N} y_i K(x, x_i)}{\Sigma_{i=1}^{N} K(x, x_i)}$$



Varying Coefficient Model

개요

○ 일반 선형 모델은 해석 가능성은 높지만, **비선형 효과와 변수간 상호작용의 포착이 어려움**

○ 딥러닝은 높은 예측 성능을 제공하지만 **해석 가능성이 부족**

VCM

○ 일반적인 회귀 모델을 확장하여, 회귀 계수 β를 **특정 변수에 의존하는 함수로 모델링**

- 변수간 상호작용이나 비선형 효과를 유연하게 학습

GLM(General Linear Model) : $E[Y|X=x] = \mu(x;\beta_0,\beta) = g^{-1}(\beta_0 + \beta^\top x)$

g는 Link 함수로, **x의 선형 조합을 비선형으로 연결하는 함수**. 대표적인 예로 logit 함수가 있음

$$g(\mu) = \eta = \beta_0 + \beta^\top X$$
$$\mu = g^{-1}(\eta)$$

Logistic 회귀에서

$$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right), \qquad \mu = \frac{1}{1+e^{-\eta}}, \qquad \eta = \beta_0 + \beta^\top X$$

VCM(General Linear Model) : $E[Y|X=x, Z=z] = \mu(x;\beta_0,\beta(z)) = g^{-1}(\beta_0 + \beta(z)^\top x)$

여기서 β(z) 는 z 변수에 따라 달라지는 회귀 계수의 함수임
z는 X 의 부분집합 혹은 외부 변수임

Varying Coefficient Model

계수 함수

○ 각 변수 x_j 에 대한 계수 함수 $\beta_j(z)$ 는 아래와 같음

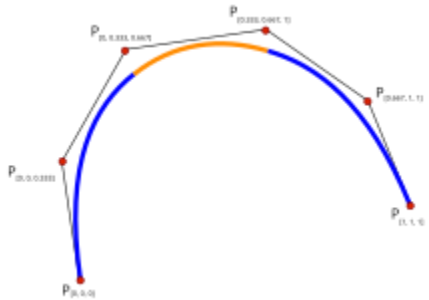
$$\beta(z) = (\beta_1(z), \beta_2(z), \ldots, \beta_p(z))^\top$$

- z와 x_j 의 관계를 알고 있는 경우

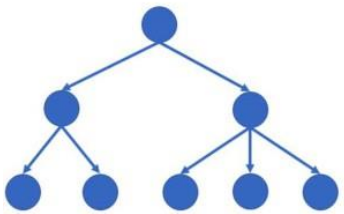
$$\beta_j(z) = a + b \cdot z$$

$$\beta_j(z) = \sin(z) \quad \text{또는} \quad \beta_j(z) = z^2$$

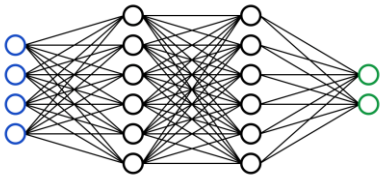
- 모델이 함수 형태를 추정해야하는 경우



Spline



Tree

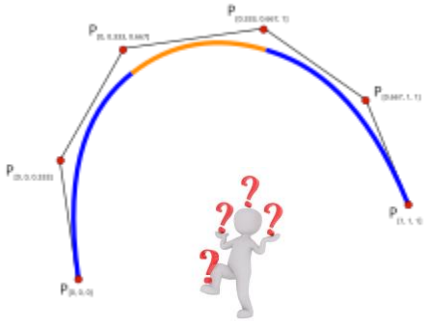


Neural net

Varying Coefficient Model

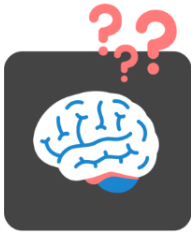
한계

○ β(z)라는 다차원 매개변수 함수를 학습해야하는데, 아래와 같은 문제점이 있음



무슨 함수 선택하지?

계산 비용



해석 난해

Cyclic Boosted VCM

개요

○ VCM에 Boosting과 Cyclic 학습을 결합하여, $\beta_j(z)$를 유연하고 효율적으로 학습

알고리즘

For $j = 1, \ldots, p$: 각 Feature를 순차적으로

If $k \leq \kappa_j$: 최대 Tree 갯수

(i) **Calculate** partial derivatives, for $i = 1, \ldots, n$:

i번째 y값과 예측평균 μ의 Loss의 Gradient를 구해서

$$g_{ij} = x_{ij} \frac{\partial}{\partial \mu} \mathcal{L}(\mu; y_i) \Big|_{\mu=\mu(\boldsymbol{x}_i;\widehat{\beta}_0,\widehat{\boldsymbol{\beta}}(\boldsymbol{z}_i))} \cdot \frac{\partial}{\partial z} u^{-1}(v) \Big|_{v=\widehat{\beta}_0+\widehat{\boldsymbol{\beta}}(\boldsymbol{z}_i)^\top \boldsymbol{x}_i}.$$

(ii) **Fit** tree regions

Tree 개선

$$\widehat{\mathcal{A}}^{(k,j)} = \arg\min_{\mathcal{A}} \sum_{\mathcal{A}_l \in \mathcal{A}} \sum_{i:\boldsymbol{z}_i \in \mathcal{A}_l} \left(g_{ij}^{(k)} - \bar{g}_l^{(k)} \right)^2$$

A는 z를 나누는 Tree,
이 Tree는 x_j 와 y 와의 관계를 z 공간에서 설명하기 위한 Tree임

Cyclic Boosted VCM

알고리즘

l번째 Tree

(iii) **Adjust** terminal node values, for $l = 1, \ldots, |\widehat{\mathcal{A}}^{(k,j)}|$:

$$\widehat{\gamma}_l^{(k,j)} = \underset{\gamma \in \mathbb{R}}{\arg\min} \sum_{i:\boldsymbol{z}_i \in \widehat{\mathcal{A}}_l^{(k,j)}} \mathcal{L}\left(u^{-1}\left(\widehat{\beta}_0 + \widehat{\boldsymbol{\beta}}(\boldsymbol{z}_i)^\top \boldsymbol{x}_i + \gamma x_{ij}\right), y_i\right)$$

Loss를 최소화하는 보정값 gamma를 찾음

(iv) **Update** coefficient function

$$\widehat{\beta}_j(\boldsymbol{z}) \leftarrow \widehat{\beta}_j(\boldsymbol{z}) + \epsilon_j \sum_{\mathcal{A}_l \in \widehat{\mathcal{A}}^{(k,j)}} \mathbb{1}_{\{\boldsymbol{z} \in \mathcal{A}_l\}} \widehat{\gamma}_l^{(k,j)}$$

계수 함수 업데이트

Cyclic Boosted VCM

$$g_{ij} = x_{ij} \frac{\partial}{\partial \mu} \mathcal{L}(\mu; y_i)\Big|_{\mu=\mu(\boldsymbol{x}_i;\hat{\beta}_0,\hat{\boldsymbol{\beta}}(\boldsymbol{z}_i))} \cdot \frac{\partial}{\partial z} u^{-1}(v)\Big|_{v=\hat{\beta}_0+\hat{\boldsymbol{\beta}}(\boldsymbol{z}_i)^\top \boldsymbol{x}_i}.$$

알고리즘

X1	X2	X3	X4	...	X_{j-2}	X_{j-1}	X_j	Y
x_{11}	x_{12}	x_{13}	x_{14}		x_{1j-2}	x_{1j-1}	x_{1j}	y_1
x_{21}	x_{22}	x_{23}	x_{24}		x_{2j-2}	x_{2j-1}	x_{2j}	y_2
. .	. .	. .	. .		. .	. .	. .	

Z
z_1
z_2
. .

z = [0, 40]

A_1 [0, 20) A_2 [20, 40]

A_1 : 1,3,5번 샘플

A_2 : 2,4,6번 샘플

g
g_{11}
g_{21}
. .

$$\hat{\mathcal{A}}^{1,1} = \underset{\mathcal{A}}{\arg\min} \sum_{\mathcal{A}_l \in \mathcal{A}} \sum_{i:\boldsymbol{z}_i \in \mathcal{A}_l} \left(g_{ij}^{(k)} - \bar{g}_l^{(k)} \right)^2$$

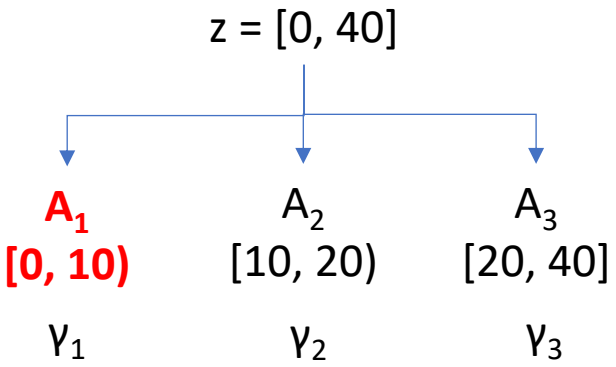
0~20 구간에서 가능한 모든 분기를 쳐서
Loss가 더 낮아지는 새로운 분기를 찾으라는 의미

g_{11} g_{31} g_{51} $(g_{11} + g_{31} + g_{51})/3$

만약 $A^{1,1}$ 의 Loss를 더 낮추는 새로운 분기 t = 10을 찾으면

Cyclic Boosted VCM

알고리즘

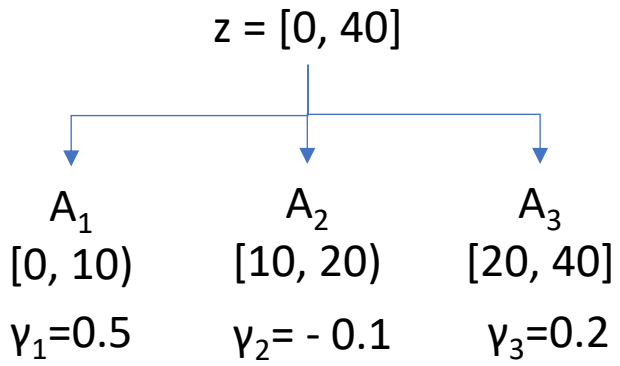


(iii) **Adjust** terminal node values, for $l = 1, \ldots, |\widehat{\mathcal{A}}^{(k,j)}|$:

$$\widehat{\gamma}_l^{(k,j)} = \underset{\gamma \in \mathbb{R}}{\arg\min} \sum_{i:\boldsymbol{z}_i \in \widehat{\mathcal{A}}_l^{(k,j)}} \mathcal{L}\left(u^{-1}\left(\widehat{\beta}_0 + \widehat{\boldsymbol{\beta}}(\boldsymbol{z}_i)^\top \boldsymbol{x}_i + \gamma x_{ij}\right), y_i\right)$$

Cyclic Boosted VCM

알고리즘



(iv) **Update** coefficient function

$$\widehat{\beta}_j(\boldsymbol{z}) \leftarrow \widehat{\beta}_j(\boldsymbol{z}) + \epsilon_j \sum_{\mathcal{A}_l \in \widehat{\mathcal{A}}^{(k,j)}} \mathbb{1}_{\{\boldsymbol{z} \in \mathcal{A}_l\}} \widehat{\gamma}_l^{(k,j)}$$

$$\beta_0 = 1, \epsilon_j = 0.1$$

$$\beta_j(z) = \begin{cases} \beta_0(=1) + 0.1 * 0.5 = 1.05 & \text{if } z \in [0,10) \\ \beta_0(=1) + 0.1 * -0.1 = 0.09 & \text{if } z \in [10,20) \\ \beta_0(=1) + 0.1 * 0.2 = 1.02 & \text{if } z \in [20,40) \end{cases}$$

Cyclic Boosted VCM

Contribution

○ **순차적인 업데이트는 아래와 같은 이점이 있음**

- 계산 효율성 높음
- **각 독립 변수 x_j 의 기여를 분리 하여 추적 가능하므로 결과를 해석할 수 있음 : Feature selection 가능**
- **Feature별로 Loss가 개선되지 않으면 Early Stopping도 가능함**

실험

○ **Simulation Data로 실험**

- X는 8차원 (독립변수 8개, X2와 X8은 상관계수 0.5, 그 외에는 독립)
- 모든 X는 평균 0, constant variance를 가지는 Normal distribution
- True Regression Function에 따라 20만개의 데이터 생성

$$\mu(\boldsymbol{x}) = \boldsymbol{\beta}(\boldsymbol{x})^\top \boldsymbol{x}$$

$$Y_i \mid \boldsymbol{X} = \boldsymbol{x}_i \sim \mathcal{N}(\mu(\boldsymbol{x}_i), 1)$$

Attention	Expression
$\beta_1(\boldsymbol{x})$	0.5
$\beta_2(\boldsymbol{x})$	$-\frac{1}{4}x_2$
$\beta_3(\boldsymbol{x})$	$\frac{1}{2}\mathrm{sgn}(x_3)\sin(2x_3)$
$\beta_4(\boldsymbol{x})$	$\frac{1}{4}x_5$
$\beta_5(\boldsymbol{x})$	$\frac{1}{4}x_4$
$\beta_6(\boldsymbol{x})$	$\frac{1}{8}x_5^2$
$\beta_7(\boldsymbol{x})$	0
$\beta_8(\boldsymbol{x})$	0

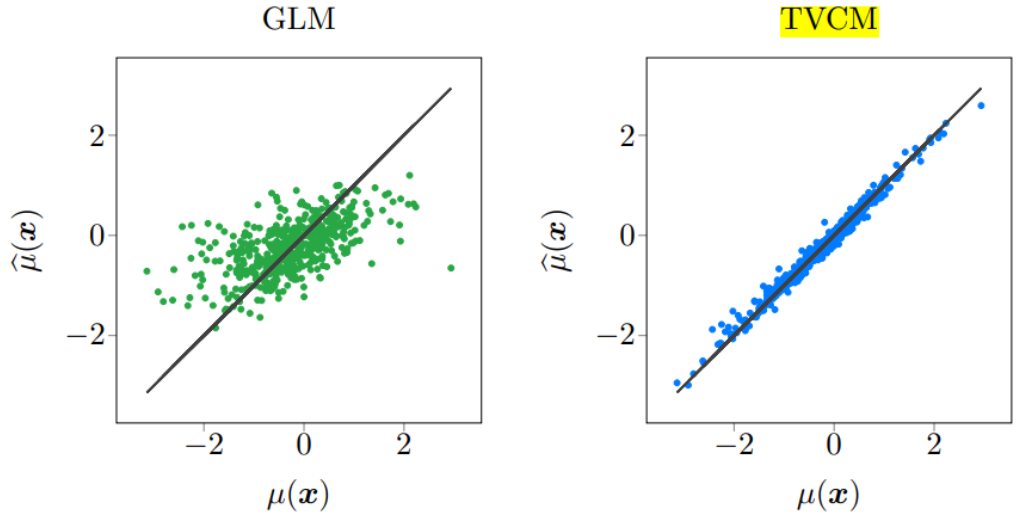
Table 1: True coefficient functions $\beta_j(\boldsymbol{x})$ for the simulated data

Cyclic Boosted VCM

실험 결과

	Train	Test
True	1.002	0.996
Intercept	1.791	1.792
GLM	1.524	1.527
TVCM	1.008	1.013
LocalGLMnet	1.002	1.005

Table 2: MSE results for the simulated data example.



Robustness in Regression

Introduction

Question 1. *In the context of linear regression, if a model has "small" standard risk, how "small" can its adversarial risk be? Is it possible to be robust while being accurate?*

선형 회귀의 맥락에서, 모델의 표준 리스크, 즉 일반적인 상황에서의 예측 오류가 작다면,
Adversarial Risk, 즉 적대적 공격을 받았을 때의 오류는 얼마나 작게 만들 수 있는가?
즉, **높은 정확도를 유지하면서 동시에 강건성을 확보할 수 있을까?**

Robustness in Regression

Goal

○ 회귀 모델에서 **정확도와 강건성의 Tradeoff**를 **정량적으로 규명**하는 것을 목표로 함

Contributions

○ **강건성 계산식 제공** : 공격 강도에 따라 강건성 최적치를 산출하는 공식

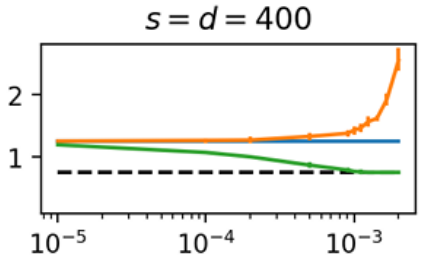
○ **강건성 조건 구체화** : 정확도 손실이 없는 상태에서 강건성을 확보하는 상황과 조건 규명

○ **상전이 다이어그램** : 정확도와 강건성의 상관관계를 시각화

$$\begin{cases} d^{1-\delta}, & \text{if } 0 \le \delta < 1, \\ \log d, & \text{if } \delta = 1, \\ 1, & \text{if } \delta > 1. \end{cases}$$

[강건성 계산식 제공]

[강건성 조건 구체화]



[Diagram]

Robustness in Regression

Key Definition

○ Standard Risk

- 모델이 일반적인 상황에서 예측할 때의 오류. 즉, 공격이 없는 상태에서의 예측 정확도

$$E(w) := \mathbb{E}[(f_w(x) - y)^2] = \|w - w_0\|_\Sigma^2 + \sigma^2$$

w : 추정한 가중치, w_0 : 정답 가중치, σ^2 : 레이블 오차 ~ $N(0, \sigma^2)$

○ Adversarial Risk

- 공격을 받았을 때 예측이 얼마나 잘못되는지를 측정. 적대적 공격은 x에 작은 변동 δ를 추가

$$E(w, r) := \mathbb{E}\left[\sup_{\|\delta\| \le r} (f_w(x + \delta) - y)^2\right]$$

r : 공격 강도, y : 관측 값

- 공격강도 ||δ||≤r 제약 조건 하에서, δ를 변경하여 모델의 예측오류를 최대화할 수 있는 상황을 찾음
- 즉, Adversarial Risk는 특정 공격 강도 r 내에서 모델이 직면할 수 있는 최악의 예측 오류값임
  · 이러한 상황의 모델은 **최악의 강건성**을 가진 상태임

Robustness in Regression

Key Definition

○ Optimal Adversarial Risk

- 공격강도 r 에서 강건성이 최선, 즉 예측 에러가 가장 낮은 상태

$$E_{opt}(r) := \min_{w \in \mathbb{R}^d} E(w, r)$$

$$w_{opt}(r)$$

○ 정규화된 Adversarial Risk

- Ridge Regression 형태로 나타내며, 정규화된 리스크는 공격강도 r이 커지면 정규화 항이 커지므로, 강건성을 강화하게 됨.

$$E(w, r) := \mathbb{E}\left[\sup_{\|\delta\| \leq r} (f_w(x + \delta) - y)^2\right]$$

$$\overline{E}(w, r) := \sigma^2 + \|w - w_0\|_\Sigma^2 + r^2 \|w\|_\star^2$$

Robustness in Regression

Robustness via Regularization

정규화 Risk : $\overline{E}(w, r) := \sigma^2 + \|w - w_0\|_\Sigma^2 + r^2\|w\|_\star^2$

$$w^{prox}(\lambda) := \arg\min_{w \in \mathbb{R}^d} \|w - w_0\|_\Sigma^2 + \lambda\|w\|_\star^2$$

Robustness in Regression

Robustness via Regularization

$$w^{prox}(\lambda) := \arg\min_{w \in \mathbb{R}^d} \|w - w_0\|_\Sigma^2 + \lambda \|w\|_\star^2$$

- 본 논문에서는 듀얼 놈을 positive definite matrix B로 놓고 위 목적식을 풀이함 즉,

$$w^{prox}(\lambda) := \arg\min_{w \in R^d} ||w - w_0||_\Sigma^2 + \lambda ||w||_B^2$$

$i)\ ||w - w_0||_\Sigma^2 = (w - w_0)^T \Sigma (w - w_0)$

$$\nabla_w \big((w - w_0)^T \Sigma (w - w_0)\big) = 2\Sigma(w - w_0)$$

$ii)\ \lambda ||w||_B^2 = \lambda * w^T \mathrm{Bw}$

$$\nabla_w (\lambda * w^T \mathrm{Bw}) = 2\lambda \mathrm{Bw}$$

$$\therefore 2\Sigma(w - w_0) + 2\lambda \mathrm{Bw} = 0$$

$$\Sigma w - \Sigma w_0 + 2\lambda B w = 0$$
$$(\Sigma + 2\lambda B) w = \Sigma w_0$$

$$\therefore w = (\Sigma + 2\lambda B)^{-1} \Sigma w_0$$

위와 같이 Closed-form의 해를 얻음.

Robustness in Regression

Robustness via Regularization

○ λ = r^2 이므로, proximal w를 구하면 Attacker's norm 과 상관 없이 아래의 최적 Risk를 얻음

$$E_{opt}(r) \approx E(w^{prox}(\lambda), r) \approx ||w^{prox}(\lambda) - w_0||_{\Sigma}^2 + r^2 ||w^{prox}(\lambda)||_p^2$$

- 즉 proximal w를 구하면, λ = r^2 이고, 이 때 **모델의 강건성**을 최대로 유지할 수 있음.
- **그러나 이 말은 공격 강도 r 에서 모델의 강건성이 좋다는 것이지, 모델 정확하다는 뜻은 아님**

Robustness in Regression

Accuracy vs Robustness Tradeoffs

○ 논문의 목적은, 강건성을 유지하면서 정확도를 유지하는 데 있음

- 즉, 공격에 의한 강건성은 유지 하면서도, Standard Error ε 을 초과하고 싶지 않음

○ 따라서, 공격 강도 r ≥ 0, 허용 오차 ε ≥ 0, 그리고 W_ϵ 을 ε-accurate model 이라고 할 때,

$$\begin{aligned}\mathcal{W}_\epsilon &:= \{w \in \mathbb{R}^d \mid \Delta(w) \leq \epsilon^2\} \\ &= \{w \in \mathbb{R}^d \mid \|w - w_0\|_\Sigma \leq \epsilon \|w_0\|_\Sigma\}\end{aligned}$$

$$E_{opt}(r, \epsilon) := \min_{w \in \mathcal{W}_\epsilon} E(w, r)$$

Robustness in Regression

Accuracy vs Robustness Tradeoffs

○ ε-accurate Wε 를 찾기 위해 ε의 기준을 설정해야 함. 따라서 임계허용오차 ε_{FL} 을 도입

$$\epsilon_{FL}(r) := \sqrt{\Delta\left(w^{prox}(r^2)\right)} = \frac{\sqrt{\left|\left|w^{prox}(r^2) - w_0\right|\right|_{\Sigma}^2}}{\left|\left|w_0\right|\right|_{\Sigma}}$$

i) $0 \le \varepsilon \le \varepsilon_{FL}(r)$ 의 경우, 즉 임계허용오차보다 허용 오차를 작게 설정한 경우엔,

$$\|w^{prox}(\lambda) - w_0\|_{\Sigma}^2 \quad = \quad \epsilon^2 \|w_0\|_{\Sigma}^2 \qquad w = (\Sigma + 2\lambda B)^{-1}\Sigma w_0$$

를 풀어서 $\lambda_{opt}(r, \varepsilon)$ 을 구할 수 있고, 이 때의 해는 $[0, r^2)$ 에 있음.

Robustness in Regression

Accuracy vs Robustness Tradeoffs

○ **ε-accurate Wε 를 찾기 위해 ε의 기준을 설정해야 함. 따라서 임계허용오차 ε_{FL} 을 도입**

$$\epsilon_{FL}(r) := \sqrt{\Delta\left(w^{prox}(r^2)\right)} = \frac{\sqrt{\left|\left|w^{prox}(r^2) - w_0\right|\right|_{\Sigma}^2}}{\left|\left|w_0\right|\right|_{\Sigma}}$$

ii) 반면, **$\varepsilon \geq \varepsilon_{FL}(r)$** 의 경우, **$\lambda_{opt}(r, \varepsilon) = r^2$** 으로 바로 놓으면 됨.
즉, 오차가 임계치보다 높은 상태이므로 정규화 측면에서의 강건성 확보만 생각하면 된다는 것임.
이와 같은 상태를 Free Lunch 라고 함.

○ **Free Lunch**

- **허용 오차 ε 이 임계값 ε_{FL} 이상일 때 성립**
- **정규화 파라미터 $\lambda = r^2$ 으로 설정, 공격강도 r에 대해 최적의 강건성을 확보하면서 성능 저하 없음**

Robustness in Regression

Conclusion

○ **이 논문은 Robustness 와 Accuracy 간의 관계를 최적화 하고자 함**

- **적대적 리스크를 최소화하면서도 모델의 Accuracy를 유지할 수 있는 조건에 대해 연구함**

○ **Main Idea**

- 공격 강도 r 과 허용 오차 ε 에 대해 정규화 파라미터 λ를 조정하는 방법을 제시
- 공격 강도 r 일 때의 허용 오차 임계값 ε_{FL} 제시
- 허용 오차 ε 이 임계값 ε_{FL} 대비 작거나 클 때의 최적 λ_{opt} 선택 방법 제시
- **$\varepsilon \geq \varepsilon_{FL}$ 이고, $\lambda = r^2$ 일 때 Free Lunch 조건이 성립**하며 **강건성과 정확성 모두 확보 가능**함

○ **Contributions**

- 회귀 모델에서의 다양한 공격강도와 허용 오차 수준에서의 강건성을 정략적으로 평가
- 정규화 전략을 통한 적대적 리스크 관리
- 안정적이면서도 정확한 모델 설계에 지침 제공

Summary

고맙습니다.