# ML IN PROBABILISTIC PERSPECTIVE

## Week 2. Linear Methods for Regression

강경훈

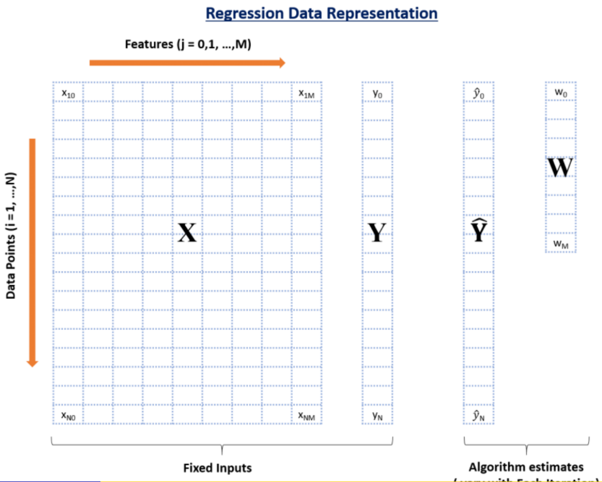
ESC, YONSEI UNIVERSITY

April 9, 2020

# Table of Contents

# LINEAR BASIS

Regression의 기본적인 셋팅. 여기서 $\mathbf{X}$를 $\boldsymbol{\Phi}$로 바꿔주는 걸 **Feature Extraction**이라고 함. (source)



# LINEAR BASIS

Polynomial Basis이든 Gaussian이든 시그모이달이든 푸리에든 결국 Design Matrix는 똑같이 생김.
이름에 겁먹을 필요 없다 어차피 Design Matrix만 정해지면 걍 OLS한다. 편-안

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}$$

$M$을 결정하는 것은 내가 얼마나 Basis를 많이 가져올거냐 결정하는 것. Polynomial이면 그게 차수이고, Gaussian이면 그게 정규분포의 중심일거고.. 핵심은 타겟 $t$는 빼놓고 설명변수만 조지는데, $\mathbf{X} = [x_1, x_2, ..., x_N]^T$를 $\mathbf{\Phi} = [\phi_0, \phi_2, ..., \phi_{M-1}]$로 바꿔주는 것. 왜 바꿔? 그래야 이쁘게 Non-linear한 선/면도 막 그리고 하니까.
자세한 예시는 필기와 코드로 때운다!

# Table of Contents

# LEAST SQUARES AND MLE

**Regression: Least Squares Approach**

알아두면 정신 건강에 도움이 되는 벡터 미분 법칙들.

**Vector Derivatives Rule 3**

Let $\alpha = \mathbf{y}^\mathbf{T}\mathbf{A}\mathbf{x}$ where $\mathbf{y} \in \mathbb{R}^{m\times 1}$, $\mathbf{x} \in \mathbb{R}^{n\times 1}$, and $\mathbf{A} \in \mathbb{R}^{m\times n}$. Then $\dfrac{\partial \alpha}{\partial \mathbf{x}} = \mathbf{y}^\mathbf{T}\mathbf{A}$ and $\dfrac{\partial \alpha}{\partial \mathbf{y}} = \mathbf{x}^\mathbf{T}\mathbf{A}^\mathbf{T}$

($\mathbf{A}$ is constant matrix.)

**Vector Derivatives Rule 4 (Quadratic formula)**

Let $\alpha = \mathbf{x}^\mathbf{T}\mathbf{A}\mathbf{x}$ where $\mathbf{x} \in \mathbb{R}^{n\times 1}$, and $\mathbf{A} \in \mathbb{R}^{n\times n}$. Then $\dfrac{\partial \alpha}{\partial \mathbf{x}} = \mathbf{x}^\mathbf{T}(\mathbf{A} + \mathbf{A}^\mathbf{T})$. ($\mathbf{A}$ is constant matrix.)

# LEAST SQUARES AND MLE

## Regression: Least Squares Approach

앞에 저것들만 알아도 당신은 벡터미분달인

### Example: Least Sqaures Estimators

- For a target vector $\mathbf{t}$ and a design matrix $\mathbf{\Phi}$, LSE $\mathbf{w}$ can be obatained by;

$$\mathbf{w} = \arg\min_{\mathbf{w}} \|\mathbf{t} - \mathbf{\Phi w}\|^2 = (\mathbf{t} - \mathbf{\Phi w})^T(\mathbf{t} - \mathbf{\Phi w})$$

Differentiate wrt $\mathbf{w}$ and we have

$$\frac{\partial}{\partial \mathbf{w}}(\mathbf{t^T t} - 2\mathbf{t^T \Phi w} + \mathbf{w^T \Phi^T \Phi w}) =^{set} 0$$

$$\therefore \mathbf{w}_{OLS} = (\mathbf{\Phi^T \Phi})^{-1}\mathbf{\Phi^T t}$$

# LEAST SQUARES AND MLE

**Regression: MLE Approach**

$$\textbf{Sampling Density} \quad p(t|x, \mathbf{w}, \beta) = \mathcal{N}(t|\mathbf{x}^T\mathbf{w}, \beta^{-1})$$

- 데이터를 전체 N개의 iid sample의 집합으로 가정한다면, 전체 데이터 셋의 분포는 다음과 같다.

$$\textbf{Joint Sampling Density} \quad p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{x_n}^T\mathbf{w}, \beta^{-1})$$

- $p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta)$에서 $\mathbf{X}$는 주어진 상수로 가정했다. $\mathbf{w}$, $\beta$를 알고 있으면 $\mathbf{t}$에 대한 pdf이지만, 거꾸로 생각해서 $\mathbf{t}$를 알고 있다면 이는 모수의 특정한 값 ($\mathbf{w}$, $\beta$)이 참일 때 주어진 데이터가 얼마나 "말이 되는지"를 알려주는 **Likelihood** 함수로 볼 수 있다.
- **Maximum Likelihood Principle:** 때문에 **Likelihood는 데이터마다 함수 형태가 다르다!** 그러나 데이터가 무수히 많아지면 결국 Likelihood는 참 모수의 값에서 극대화된다. 때문에 주어진 데이터로 그린 Likelihood를 최대화하는 지점 ($\mathbf{w}$, $\beta$)을 모수의 추정치로 삼을 수 있다.

# LEAST SQUARES AND MLE

## Regression: MLE Approach

- 최적화 문제의 장점은 목적함수에 단조변화함수를 맘껏 취해줄 수 있다는 것이다. 때문에 로그를 취해 $\prod$를 $\sum$으로 바꿔주면

$$\begin{aligned}\ln p(\mathbf{t}|\mathbf{X},\mathbf{w},\beta) &= -\frac{\beta}{2}\sum_{n=1}^{N}\{\mathbf{x_n^T w}-t_n\}^2+\frac{N}{2}\ln\beta-\frac{N}{2}\ln 2\pi \\ &= -\frac{\beta}{2}\|\mathbf{t}-\mathbf{Xw}\|^2+\frac{N}{2}\ln\beta-\frac{N}{2}\ln 2\pi\end{aligned}$$

- $\beta$의 값은 $\mathbf{w}$가 최소화되는 지점에는 영향을 미치지 않는다. 때문에 아까 본 error function을 최소화하는 문제와 똑같다. $\mathbf{w}_{ML}=\mathbf{X}(\mathbf{X^T X})^{-\mathbf{1}}\mathbf{t}$, $\beta$에 대해 미분하면 $\beta_{ML}^{-1}=\frac{1}{N}\|\mathbf{t}-\mathbf{Xw}\|^2$
- 이렇게 구한 추정치를 원래의 Likelihood에 넣으면, 우리는 새로운 데이터 $t$에 대한 predictive distribution, 일종의 확률모델을 얻는다. 이거 돌려서 예측하는 것.

**Predictive Distribution** $\quad p(t|x,\mathbf{w}_{ML},\beta_{ML})=\mathcal{N}(t|\mathbf{x}^T\mathbf{w}_{ML},\beta_{ML}^{-1})$

# LEAST SQUARES AND MLE

## Regression: Geometric Interpretation

### Least Squares Solution

- From the orthogonal condition of LS solution $(\mathbf{t} - \mathbf{M}\mathbf{w}) \cdot \mathbf{m} = 0 \quad {}^\forall \mathbf{m} \in col\mathbf{M}$, it follows;

$$
\begin{aligned}
(\mathbf{t} - \mathbf{M}\mathbf{w}) \cdot \mathbf{m} = 0 &\quad {}^\forall \mathbf{a} \in col\mathbf{M} \\
(\mathbf{t} - \mathbf{M}\mathbf{w}) \cdot \mathbf{M}\mathbf{b} = 0 &\quad {}^\forall \mathbf{b} \in \mathbb{R}^n \\
(\mathbf{M}^T\mathbf{t} - \mathbf{M}^T\mathbf{M}\mathbf{w}) \cdot \mathbf{b} = 0 &\quad {}^\forall \mathbf{b} \in \mathbb{R}^n \\
\mathbf{M}^T\mathbf{t} = \mathbf{M}^T\mathbf{M}\mathbf{w} &
\end{aligned}
$$

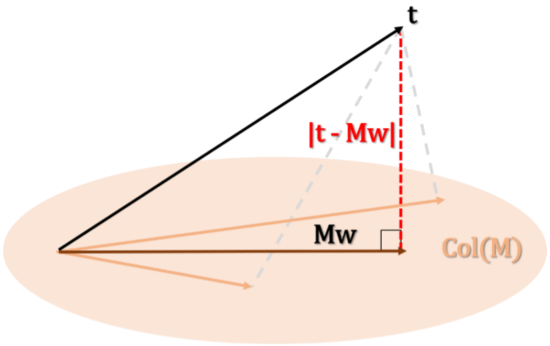
  That is, $\mathbf{w}$ is LS solution iff it satisfies the above equation, and there **always** exists at least one solution. (Pick any orthonormal basis $\{\mathbf{v}_1, ..., \mathbf{v}_p\}$ and we have $\mathbf{M}\mathbf{w} = \sum k_i \mathbf{v}_i$. Then since $(\mathbf{t} - \sum k_i \mathbf{v}_i) \cdot \mathbf{v}_j = 0$ so that we can write $k_j = \mathbf{t} \cdot \mathbf{v}_j$.)
- If $\mathbf{M}$ is non-singular, i.e. each column is linearly independent, then $\mathbf{M}^T\mathbf{M}$ is invertible and we have a unique LS solution; $\mathbf{w} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{t}$
- If $\mathbf{M}$ is singular, then there are infinitely many solutions.

# LEAST SQUARES AND MLE

## Regression: Geometric Interpretation

**Least Squares Solution**

$\mathbf{Mw}$ is the closest to $\mathbf{t}$ among all the other vectors in the plane $col\mathbf{M}$.



# Table of Contents

# SHRINKAGE METHODS: RIDGE AND LASSO

모든 변수에다가 fitting을 하긴 하는데, MSE가 조금 높아도 되니 $\beta$가 "쪼그라드는" 방법은 없을까?

**Regularization**

- **오캄의 면도날:** 중세 유명론의 대가 윌리엄 오브 오캄(William of Ockham, ca.1285-1349) 선생님께서는 다음과 같이 말씀하셨지요. (Principle of Parsimony)

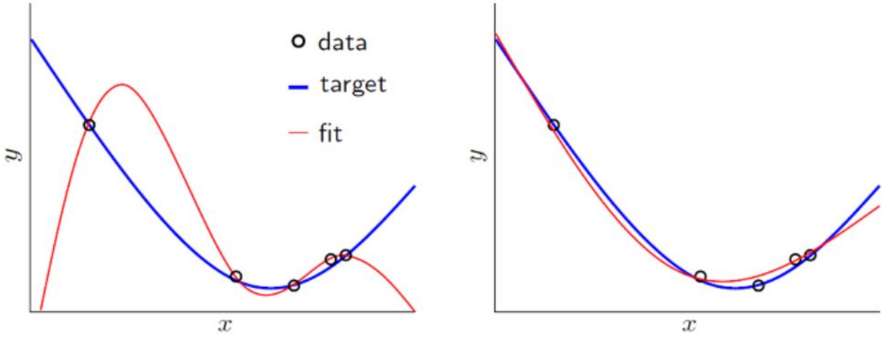
  *"Plurality should not be posited without necessity."*

  무슨말이냐? **쓸데없이 여러 가정을 넣지 말라. 즉 모델을 복잡하게 만들지 말라는 거다!**[1] 왜?
- 실제 데이터 형성 과정이 $Y = f(X) + \epsilon$라면, 우리가 얻는 데이터에는 $\epsilon$이 섞여있다. 만일 이를 고려하지 않고 $\epsilon$까지 포함해 모델을 fitting하면, 다음 슬라이드와 같은 불상사가 일어날 수 있다.
- 이를 방지하기 위해 일종의 Regularization 혹은 Penalty term을 추가하여 모델의 복잡도를 방지하고자 하는 방법이 Regularization이다!

[1]참조 동영상: `https://www.youtube.com/watch?v=iWtdGpSYEC0`

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization




(a) without regularization (b) with regularization

[1]이미지 출처: `https://enginius.tistory.com/476`, 이분도 어디 강의노트에서 가져오신듯

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization

- OLS를 예시로 들어보자. OLS의 회귀계수는 다음과 같이 구해진다.

$$\hat{\beta}_{OLS} = \arg\min_{\beta} \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2 = \arg\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i^T \beta)^2 = \arg\min_{\beta} \|Y - X\beta\|^2$$

- 위 식은 $\beta$의 크기에 상관없이 그냥 RSS가 가장 작은 $\beta$를 뱉어낸다. 여기에 $\beta$의 크기를 작게 하도록 Regularization term을 추가하면?

$$\hat{\beta}_{L1,Lasso} = \arg\min_{\beta} [ \sum_{i=1}^{N} (Y_i - X_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \ ] = \arg\min_{\beta} [ \ \|Y - X\beta\|^2 + \lambda \|\beta\|_1 \ ]$$

$$\hat{\beta}_{L2,Ridge} = \arg\min_{\beta} [ \sum_{i=1}^{N} (Y_i - X_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^2 \ ] = \arg\min_{\beta} [ \ \|Y - X\beta\|^2 + \lambda \|\beta\|_2^2 \ ]$$

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization

- 기하학적으로 이해해보자. Best Subset Model Selection 문제를 수식으로 표현하면 다음과 같다.

$$\hat{\beta}_{Best} = \arg\min_{\beta} \|Y - X\beta\|^2 \ \ s.t. \ \sum_{j=1}^{p} I(\beta_j \neq 0) \leq s$$

- 아쉽지만 위의 식은 computationally infeasible. 위의 조건을 조금 완화한 것이 바로 Ridge와 Lasso!

$$\hat{\beta}_{Lasso} = \arg\min_{\beta} \|Y - X\beta\|^2 \ \ s.t. \ \sum_{j=1}^{p} |\beta_i| \leq s$$

$$\hat{\beta}_{Ridge} = \arg\min_{\beta} \|Y - X\beta\|^2 \ \ s.t. \ \sum_{j=1}^{p} |\beta_i|^2 \leq s$$

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: Lasso promotes sparsity of coefficients!**

Lasso 방식을 쓸 때 모서리 해가 더 잘 나온다! (ex. compare $(1,0)$ vs $(1/\sqrt{2}, 1/\sqrt{2})$)

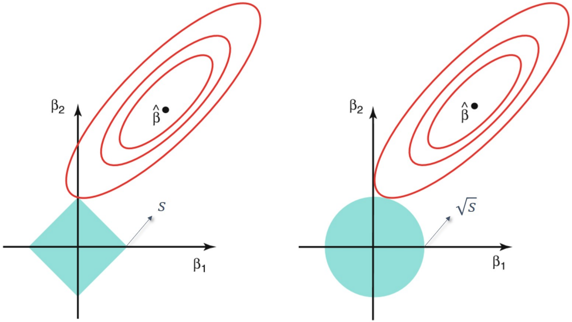



**FIGURE 6.7.** *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: Lasso promotes sparsity of coefficients!**

Lasso 방식을 쓸 때 모서리 해가 더 잘 나온다!

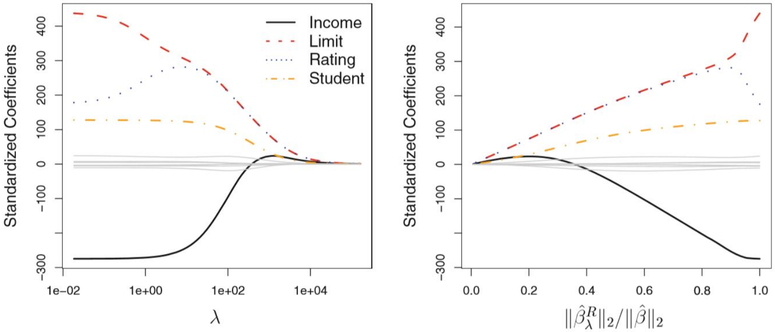



**FIGURE 6.4.** *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* $\lambda$ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: Lasso promotes sparsity of coefficients!**

Lasso 방식을 쓸 때 모서리 해가 더 잘 나온다!

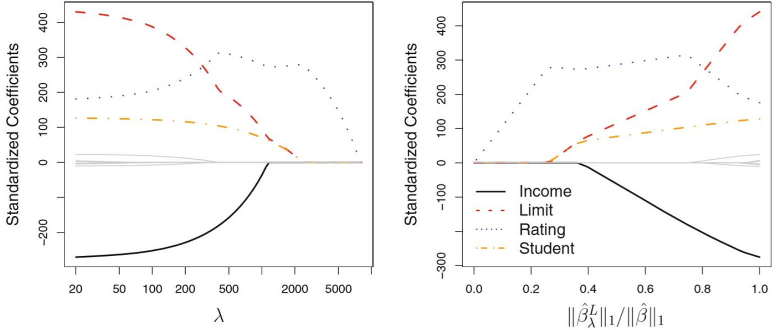



**FIGURE 6.6.** *The standardized lasso coefficients on the* Credit *data set are shown as a function of* $\lambda$ *and* $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$.

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: Lasso promotes sparsity of coefficients!**

- 더 직관적인 예를 위해 $X$가 $I_{n\times n}$인 경우를 보자. 이 Lasso와 Ridge 식은 다음과 같다.

$$Lasso: \quad \sum(y_i - \beta_i)^2 + \lambda\sum|\beta_i|$$

$$Ridge: \quad \sum(y_i - \beta_i)^2 + \lambda\sum\beta_i^2$$

- 이를 만족하는 해는 다음과 같다.

$$\beta_i^R = y_i/(1+\lambda)$$

$$\beta_i^L = \begin{cases} y_i - \lambda/2 & \text{if } y_i > \lambda/2 \\ y_i + \lambda/2 & \text{if } y_i < -\lambda/2 \\ 0 & \text{if } |y_i| \leq \lambda/2 \end{cases}$$

- 즉 Ridge는 모든 계수를 일정한 비율만큼 줄여주는 반면, Lasso는 같은 정도로 빼주고, 절댓값이 일정 이하이면 모조리 0으로 바꿈. 좀 더 일반적인 경우도 대충 이런 식이다.

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: Lasso promotes sparsity of coefficients!**

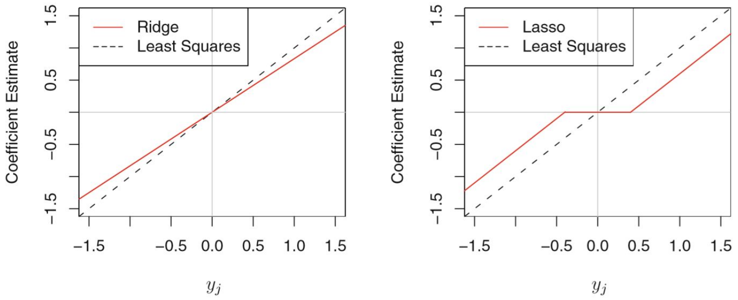



**FIGURE 6.10.** *The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and $\mathbf{X}$ a diagonal matrix with 1's on the diagonal.* Left: *The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates.* Right: *The lasso coefficient estimates are soft-thresholded towards zero.*

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: in Bayesian Lens**

여기까지만 알아도 되지만, 기왕 베이지안 배운거 베이지안 관점에서 Ridge와 Lasso를 이해해보자.

- 모수 $\theta$인 확률분포를 따르는 확률변수 y를 생각해보자.

$$\textbf{똑같은 식, 다른 해석} \begin{cases} Probability\ Density\ of\ y := P(y|\theta) \\ Likelihood\ of\ \theta\ (given\ y) := L(\theta|y) \end{cases}$$

- $\theta$를 어떻게 추정할까?
  - **Frequentist (MLE):** Likelihood $L(\theta|y)$을 최대화하는 단 하나의 값을 $\hat{\theta}_{MLE}$

    $$\textbf{Maximum Likelihood Estimator: } \hat{\theta}_{MLE} = \arg\max_{\theta} \log P(y|\theta)$$

  - **Bayesian (MAP):** $P(\theta|y)$를 Bayes Rule로 뜯어보면 다음과 같다.

$$\underbrace{P(\theta|y)}_{\text{posterior}} = \frac{\overbrace{P(y|\theta)}^{\text{likelihood}}\ \overbrace{P(\theta)}^{\text{prior}}}{\underbrace{P(y)}_{\text{evidence}}}$$

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization: in Bayesian Lens

- $\theta$를 어떻게 추정할까?
  - **Bayesian (MAP):** $P(\theta|y)$를 Bayes Rule로 뜯어보면 다음과 같다.

$$\underbrace{P(\theta|y)}_{\text{posterior}} = \frac{\overbrace{P(y|\theta)}^{\text{likelihood}}\overbrace{P(\theta)}^{\text{prior}}}{\underbrace{P(y)}_{\text{evidence}}}$$

  - 빈도론자는 주어진 데이터가 나올 확률을 가장 높여주는 **하나의 값**을 채택한다. (가설 검정도 결국 $\theta$가 취할 수 있는 공간을 임의로 두 영역으로 분리해, $H_0$하의 단 하나의 값에서의 $P(y|\theta_{H_0})$를 보는 것)
  - 이에 반해 베이지안은
    1) $\theta$에 대한 나의 사전 믿음과,
    2) 주어진 데이터에서 어떤 $\theta$ 값이 얼마나 likely한지를 종합적으로 판단해,
    3) 데이터에 의해 수정된 $\theta$에 대한 사후 믿음, 즉 **확률 분포 전체**를 보여준다.

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization: in Bayesian Lens

- $\theta$를 어떻게 추정할까?
  - **Bayesian (MAP):** 여기서 얻는 Posterior 분포 $P(\theta|y)$의 값이 최대가 되는 값을 $\hat{\theta}_{MAP}$

$$\textbf{Maximum A Posteriori Estimator: } \hat{\theta}_{MAP} = \arg\max_{\theta} P(\theta|y)$$
$$= \arg\max_{\theta} \frac{P(y|\theta)P(\theta)}{P(y)}$$
$$= \arg\max_{\theta} P(y|\theta)P(\theta)$$
$$= \arg\max_{\theta}[\ \log P(y|\theta) + \log P(\theta)\ ]$$

Compare this with

$$\textbf{Maximum Likelihood Estimator: } \hat{\theta}_{MLE} = \arg\max_{\theta} \log P(y|\theta)$$

Linear Regression의 맥락에서 생각한다면, ($\sigma^2$를 unknown but constant로 가정했을 때) $\beta$에 어떤 prior를 주냐에 따라 Ridge 혹은 Lasso!

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization: in Bayesian Lens

- Normal Prior: $\beta \sim MVN(0_p, \tau^2 I_p)$
  **의미: 나는 $\beta$가 0이라는 "종 모양"의 믿음을 가지고 있다.**

$$
\begin{aligned}
\hat{\beta}_{MAP} &= \arg\max_{\beta}\Big[\ \frac{1}{\sqrt{2\pi\sigma^2}} exp\Big(-\frac{1}{2\sigma^2}\|Y - X\beta\|^2\Big) + \frac{1}{\sqrt{2\pi\tau^2}} exp\Big(-\frac{1}{2\tau^2}\beta^T\beta\Big)\ \Big] \\
&= \arg\max_{\beta}\Big[\ -\frac{1}{2\sigma^2}\|Y - X\beta\|^2 - \frac{1}{2\tau^2}\beta^T\beta\ \Big] \\
&= \arg\min_{\beta}\Big[\ \|Y - X\beta\|^2 + \frac{\sigma^2}{\tau^2}\|\beta\|_2^2\ \Big] \\
&= \hat{\beta}_{L2,Ridge}
\end{aligned}
$$

즉 Ridge Regression은 Beta 사전 분포가 정규 분포일때 MAP Estimate라는 것!
또한 사전분포의 scale $\sigma^2$를 낮게 잡을수록(강한 믿음!) 실질적으로 $\lambda$가 높아진다(높은 기준!).

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization: in Bayesian Lens

- Laplacean Prior: $\beta_j \sim Laplace(0, b)$ $\quad (c.f.\ P(y|\mu, b) = \frac{1}{2b} exp(\frac{-|y-\mu|}{b}))$
  **의미: 나는 $\beta$가 0이라는 "뾰족한" 믿음을 가지고 있다.**

$$\hat{\beta}_{MAP} = \arg\max_{\beta}[\ \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{1}{2\sigma^2}\|Y - X\beta\|^2) + \prod_{j=1}^{p} \frac{1}{2b} exp(-\frac{|\beta_j|}{b})\ ]$$

$$= \arg\max_{\beta}[\ -\frac{1}{2\sigma^2}\|Y - X\beta\|^2 - \sum_{j=1}^{p} \frac{|\beta_j|}{b}\ ]$$

$$= \arg\min_{\beta}[\ \|Y - X\beta\|^2 + \frac{2\sigma^2}{b}\|\beta\|_1\ ]$$

$$= \hat{\beta}_{L1,Lasso}$$

즉 Ridge Regression은 Beta 사전 분포가 정규 분포일때 MAP Estimate라는 것!
또한 사전분포의 scale $b2$를 낮게 잡을수록(강한 믿음!) 실질적으로 $\lambda$가 높아진다(높은 기준!).

# SHRINKAGE METHODS: RIDGE AND LASSO

## Regularization: in Bayesian Lens

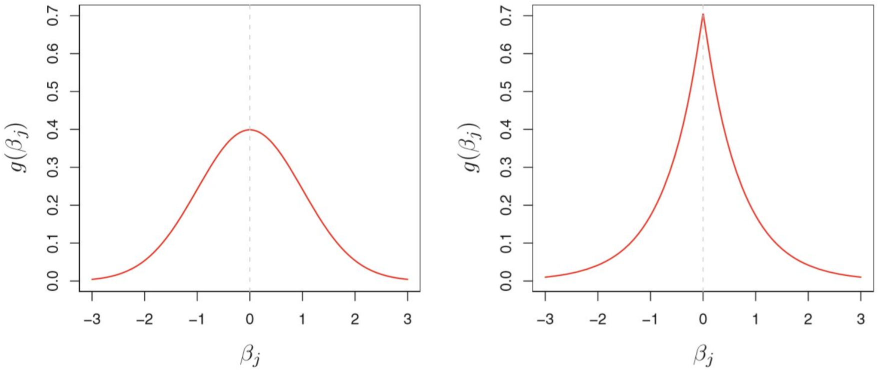



**FIGURE 6.11.** Left: *Ridge regression is the posterior mode for* $\beta$ *under a Gaussian prior.* Right: *The lasso is the posterior mode for* $\beta$ *under a double-exponential prior.*

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: Selecting tuning parameter**

$\lambda$는 어떻게 고르냐? CV 최소화하는 지점을 보면 되지!

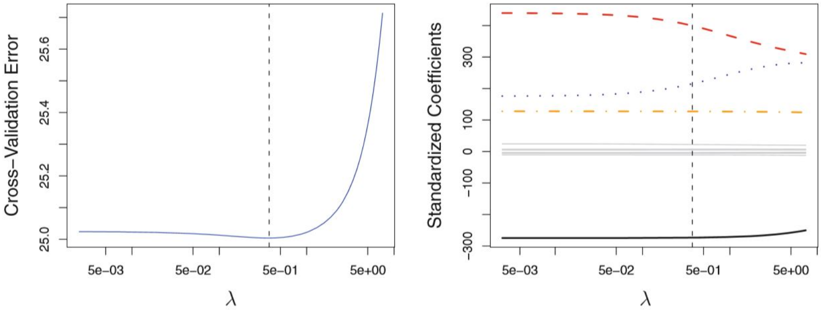



**FIGURE 6.12.** Left: *Cross-validation errors that result from applying ridge regression to the* Credit *data set with various value of* $\lambda$. Right: *The coefficient estimates as a function of* $\lambda$. *The vertical dashed lines indicate the value of* $\lambda$ *selected by cross-validation.*

# SHRINKAGE METHODS: RIDGE AND LASSO

**Regularization: Selecting tuning parameter**

p=45, n=50인 경우. 이처럼 p가 많을 때 쳐내는 용도로 Lasso가 좋다.

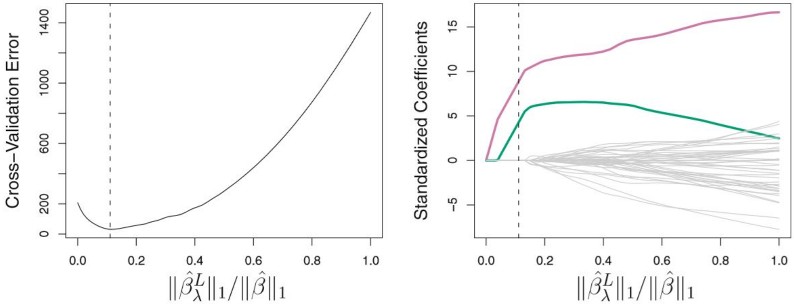



**FIGURE 6.13.** Left: *Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9.* Right: *The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

# Table of Contents

# GRADIENT DESCENT

한치 앞도 안 보이는 잠, 당신은 오로지 감에 의존하여 산을 내려가야 합니다. 어떻게 하시겠습니까? 당연히 경사가 낮은 쪽으로 내려가겠죠? **Gradient Descent** 끝. 아래 내용은 궁금하면 읽고 아님 skip

- Batch Gradient Descent:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\nabla\sum_{i=1}^{N}(t_i - \mathbf{w}_t^T\phi(\mathbf{x_i}))^2$$

- Stochastic Gradient Descent

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\nabla(t_i - \mathbf{w}_t^T\phi(\mathbf{x_i}))^2$$

- 왜 마이너스인가? **최솟값** 찾는 문제니까! $\nabla_{\mathbf{w}} f = \langle \mathbf{w}, \nabla f \rangle = \|\mathbf{w}\|\|\nabla f\| \cos\theta$인데, 우리는 $\nabla_{\mathbf{w}} f$의 값이 작아져 0이 되는 지점으로 가야하니, $\cos\theta = -1$이 되는 방향, 즉 그라디언트의 반대 방향으로 가야한다.

# GRADIENT DESCENT

## 그....그라디언트...? (1)

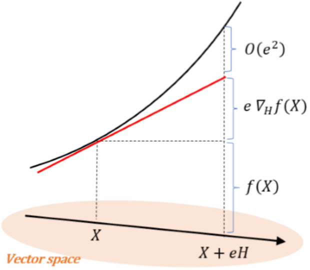
**n-dimensional Derivative**

- For $f(x) : \mathbb{R}^n \mapsto \mathbb{R}$, an infinitsimal change $e$ in the input $\mathbf{x}$ **in the direction of $\mathbf{H}$** leads to change in the output;

$$f(\mathbf{X} + e\mathbf{H}) = f(\mathbf{X}) + e\nabla_{\mathbf{H}} f(\mathbf{X}) + O(e^2)$$

- Rearrage, $e \to 0$, and we have nD **Directional** Derivative;

$$\nabla_{\mathbf{H}} f(\mathbf{X}) = \lim_{e \to 0} \frac{f(\mathbf{X} + e\mathbf{H}) - f(\mathbf{X})}{e}$$




$$\nabla_H f(X) = H \cdot \nabla f = H \cdot \frac{\partial f}{\partial X}$$

# GRADIENT DESCENT

## 그....그라디언트...? (2)

### Directional Derivative in Vectorspace

- With a gradient of a function $f; \mathbf{x} \mapsto \mathbb{R}$, once we specify a direction $\mathbf{b}$ (unit vector) of variation in the input $\mathbf{x}$, we have a **directional derivative of** $f$;

$$\begin{aligned}\nabla_{\mathbf{b}} f(\mathbf{x}) = f'(\mathbf{x}) \cdot \mathbf{b} &= \lim_{\epsilon \to 0} \frac{f(\mathbf{x} + \epsilon \mathbf{b}) - f(\mathbf{x})}{\epsilon} \\ &= \nabla f(\mathbf{x}) \cdot \mathbf{b} = \|\nabla f(\mathbf{x})\| \|\mathbf{b}\| \cos\theta\end{aligned}$$

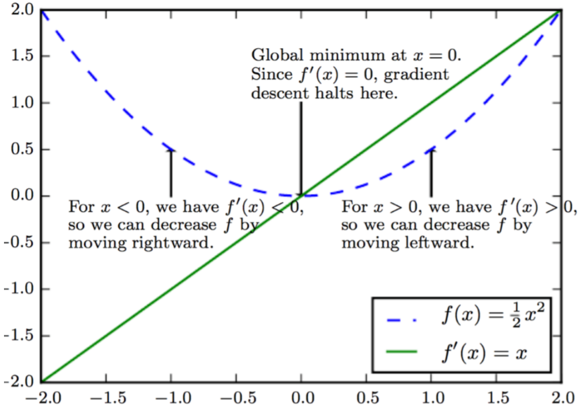
- Directional derivative is just a $\mathbb{R}^1$ derivative generalized to $\mathbb{R}^n$ space. The difference is that, unlike in $\mathbb{R}^1$ where we had only a number line, in higher dimensions we have to specify which direction $d\mathbf{x}$ is headed.

**1D derivatives:** $$\frac{df(x)}{dx} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon) - f(x)}{\epsilon}$$

- $\nabla f(\mathbf{x})$ is called **gradient** and represents a direction in $\mathbf{x}$ of the maximum change in $f(\mathbf{x})$ since $\cos\theta = 1 \iff \theta = 0.$

# GRADIENT DESCENT

심화 내용이 궁금하면 스리하리 교수님 **PPT**참조



# Table of Contents

# TEST MSE DECOMPOSTION

- 모델 Complexity를 정할 때 너무 단순하면 모델의 Bias가 높고, 너무 복잡하면 모델의 Variance가 높다는 말을 했다. 이제 이를 수식으로 보여보자.
- Loss function for Regression (Decision Theory)

$$L = (t - f)^2$$

$t \sim p(t)$는 타(확률변수), $f = f(x; D)$는 우리가 가진 데이터 $D$로 결정되는 t의 예측값. 이 loss는 그때 그때 어떤 데이터가 얻어지냐에 따라 다를 것이다. 우겟리는 **모든 데이터에 걸쳐 loss를 최소화하는** $f$를 고르고 싶다.

- Expected Loss

$$E[L] = E(t - f)^2$$

**우리가 구하는 MSE는 바로 Expected Loss의 추정량이다.** 그래서 이 MSE를 낮추려고 했던 것. 이 식을 한번 쪼개보자.

# TEST MSE DECOMPOSTION

- Expected Loss. ($E[t|x] = E_x t$)

$$\begin{aligned} E[L] = E(t-f)^2 &= E(t - E_x t + E_x t - f)^2 \\ &= E(t-E_x t)^2 + E(E_x t - f)^2 + 2E(t-E_x t)(E_x t - f) \\ &= Var(t|x) + E(f - E_x t)^2 \end{aligned}$$

($E[\cdot] = E[E[\cdot|x]]$ 쓰면 마지막 텀이 날라감)

때문에 우리는 $f = E_x t$로 고를 때, 즉 t를 t의 x에 대한 조건부 기대로 예측할때 가장 Loss가 최소화된다고 배웠다. 근데 이건 모분포 $p(x,t)$를 알 때나 가능하다. 그래서 Population Minimizer 라고 하는 거다.

- 실생활에서 우리에게 주어진 것은 하나의 제한된 표본이고, 그 제한된 표본으로 구한 불안정한 모델 $f$이다. 이 모델을 어떻게 잡느냐에 따라서 MSE가 바뀔건데, 어차피 못 없애는 $Var(t|x)$ 말고는 우리의 모델 선택에 영향을 끼치는 것은 $E(f - E_x t)^2$이다. 얘를 한번 쪼개보자.

# TEST MSE DECOMPOSTION

$$\begin{aligned} E(f - E_x t)^2 &= E(f - Ef + Ef - E_x t)^2 \\ &= E(f - Ef)^2 + E(Ef - E_x t)^2 + 2E(f - Ef)(Ef - E_x t) \\ &= E(f - Ef)^2 + (Ef - E_x t)^2 \end{aligned}$$

- $E(f - Ef)^2$: **Variance** 이게 바로 모델의 분산이다. 여기서 $E$은 **모든 데이터에 걸친 평균**을 의미한다. 즉 한 데이터에 모델 $f$를 fitting하면 어떤 예측값이 나오지만 그 예측값을 모든 데이터에 걸쳐 평균을 구하면 $Ef$이다.
- $(Ef - E_x t)^2$: **Bias** 이게 바로 모델의 편차이다. $Ef$는 나의 불완전한 모델을 모든 데이터에 돌렸을 때 얻은 예측의 평균, $E_x t$은 전지전능 population minimizer 정답. 때문에 데이터에 따라 달라지는게 아니니까 E 밖으로 나온다.

# TEST MSE DECOMPOSTION

$$
\begin{aligned}
E(f - E_x t)^2 &= E(f - Ef + Ef - E_x t)^2 \\
&= E(f - Ef)^2 + E(Ef - E_x t)^2 + 2E(f - Ef)(Ef - E_x t) \\
&= E(f - Ef)^2 + (Ef - E_x t)^2
\end{aligned}
$$

- 통계학의 진리(by 강승호 교수님): **정확한 모델은 없다.** 즉 $f$를 니가 아무리 잘 잡아도 인간이 감히 상상할 수 없이 하나님만이 아는 $E_x t$를 구할 수 없다. 그래도 지 딴에 모델을 정교하게 잡으면 계속 이 편차가 줄 것이다.
- 하지만 실상은 모델을 복잡하게 잡으면 과적화. 왜? **데이터가 제한되어 있으니까 그렇다!** 데이터가 늘면 늘수록 모델의 분산은 줄어든다. 데이터의 개수만큼 모델의 정확도가 높아지니까. 그러나 실제로는 데이터가 제한되어있으니 복잡한 모델이 쪼꼬만한 데이터의 에러까지 잡아 회를 떠놓으니 과적화가 되는 것이다.

# TEST MSE DECOMPOSTION

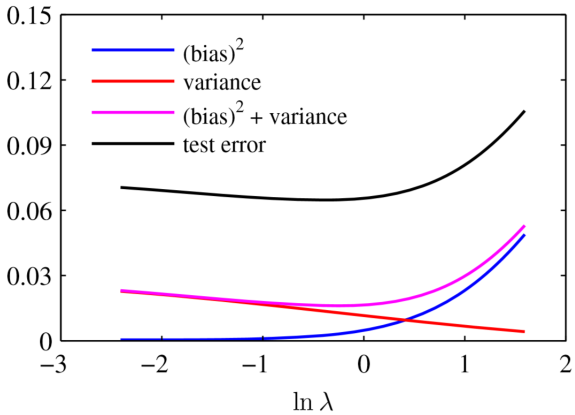



Figure: **Test MSE에 숨겨져 있는 것들**

# Table of Contents

# ASSIGNMENT

## 2주차 과제

- **(필수) Real 데이터로 다음을 해보기**
  1. sklearn 활용해 test set 추출 (test set은 데이터 분석에 **절대로** 사용하지 않는다!)
  2. sklearn 활용해 Polynomial Basis로 Feature Extraction (차수는 2 이상)
  3. sklearn 활용해 Ridge / Lasso Regression 해보기
  4. sklearn 활용해 k-cv sampling을 한 후, MSE가 최소화되는 정규화 계수 $\lambda$ 찾기
  5. 최종적으로 test MSE 보고 후, 어떤 feature가 선택되었으며, 왜 그랬는지 설명해보기 (1, 2문장)
- (선택, 후한 가산점) 위도와 경도가 있는 공간 데이터를 구하고, geopanda로 시각화해보기 (미국이 shp 파일 구하기 쉬울 거에요)

올려드린 코드를 따라해보셔도 되고, 본인이 스스로 더 좋은 코드를 만들어서 활용하셔도 됩니다.
중요한 것은 **한 번 해보기**입니다!