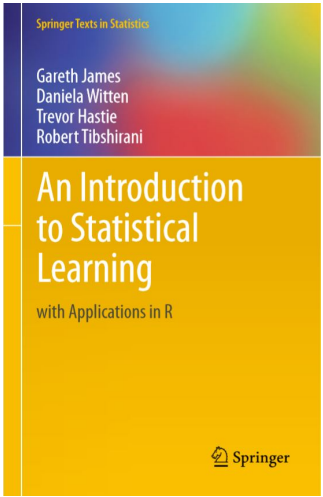
쉽게 배우는 머신러닝

Ch.6 Linear Model Selection and Regularization

훈 러닝 (Hun Learning)

January 12, 2020

TEXTBOOK



- **An Introduction to Statistical Learning : with Applications in R**
- 목차:
  1. Intro
  2. Statistical Learning
  3. Linear Regression
  4. Classification
  5. Resampling Methods
  6. Linear Model Selection and Regularization
  7. Moving Beyond Linearity
  8. Tree-based Methods
  9. Support Vector Machines
  10. Unsupervised Learning

LINEAR MODEL FRAMEWORK

이번 챕터에서는 우선 선형 모델에 대해 얘기해보자. 7장과 8장에서 비선형 모델에 대한 얘기도 한다. 그 전에 일단 선형 모델부터 좀 더 "개선"하는 방법을 짚고 간다.

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + ... + \beta_{p-1} X_{i,p-1}$$

다 좋은데 될 수 있으면 p의 개수를 좀 줄이고 싶다. 왜?

- **Model Interpretability:** 당연히 간단한게 더 보기 좋잖아?
- **Prediction Accuracy:** p가 n에 비교해서 클수록 안습한 상황이 벌어진다. p가 크면 매번 다른 데이터로 fitting할때마다 베타가 왔다갔다하고, p가 n보다 크면 무수히 많은 fitting이 가능함.
  - Linear Model은 n차원 벡터 Y를 p개의 n차원 벡터 X_i의 선형결합으로 나타내는 모형. 이때 OLS는 Y를 $dim = p$인 $col(X)$에 직교투영하는 방법이다. 이때 만일 p가 n에 거의 가까우면 어쩔 수 없이 데이터에 따라 predictor 간에 상관관계가 높은 결과가 나올 수 있다. 때문에 데이터에 따라 $\beta = (X^T X)^{-1} X^T Y$가 널뛰기할 것.
  - 아예 n보다 p가 같거나 크면? $col(X)$이 $dim \geq n$이므로 X로 Y를 perfect fit 가능. 심지어 그렇게 해주는 β가 무수히 많음

LINEAR MODEL FRAMEWORK

선형모델에서 p의 개수를 줄여주는 방법은?

1. **Subset Selection:**
   p개 중에서 일단 몇 개 쳐내보고 OLS fit을 보자. 나아지면 쳐내고, 아니면 다른거 쳐내보고.
2. **Shrinkage:**
   일단 p개로 다 fit을 하긴 한다. 그러나 OLS 말고 다른 방식으로 fit해서 β가 줄어들게(shrink) 해보자!
3. **Dimension Reduction:**
   p개의 predictor를 선형결합해 M개로 만들고, $M < p$인 M차원 공간에 Y를 직교투영한다. 이렇게 탄생한 M개의 변수를 "latent variable"이라고 한다.

차례차례 알아보자.

SUBSET SELECTION

Best Subset Selection

- 변수가 p개이면 가능한 모든 모델의 수는 2^p이다. 이걸 모두 다 비교하는게 Best Subset Selection
  1. 1부터 p까지 숫자 k에 대해, $\binom{p}{k}$개의 모델 중에서 RSS, R^2 (tranining error)가 가장 작은 놈을 뽑는다.
  2. 뽑은 놈들 중에서 CV error(test error), C_p, AIC, BIC, Adjusted R^2가 작은 놈을 뽑는다.
- 문제는 p가 크면 사실상 불가능하다는 거. $2^{20} =?$ R^2 백만 개 구하고 있을거야?

Stepwise Selection

- 때문에 타협한게 stepwise 방법. 일단 아무것도 없는 상태에서 하나씩 넣어보거나(forward), 아니면 다 넣은 상태에서 하나씩 빼보거나(backward).
  1. 개개 변수 하나를 넣거나(forward) 뺐을 때(backward)의 training error의 개선을 보면서 "변수 k개 짜리 모델 중에서는 M_k 모델이 제일 낫구나" 정하고
  2. M_0부터 M_p까지 중에서 가장 CV error, C_p, AIC, BIC, Adjusted R^2이 낮은 M_k을 뽑는 것.
- **Hybrid Approach:** Forward는 한번 넣은 애는 계속 들어가있으니, 이걸 개선한답시고 forward에서 새로 변수 하나를 추가했을 때마다 가장 fit 기여가 적은 기존 변수를 빼는 방법.

SUBSET SELECTION

"좋은" Model을 고를 때 RSS와 R^2는 training error이므로 적절한 기준이 아니다. 뭘 써야 할까?

- Cross Validation 방법으로 test error 추정
- training error을 적절히 수정해 변수 개수에 따른 패널티를 추가한 척도를 쓴다.
  C_p, AIC, BIC, Adjusted R^2 등등

1. Mallow's C_p

$$C_p = \frac{RSS_p}{\hat{\sigma}^2_{all\ p}} - n + 2p$$

- $\hat{\sigma}^2_{all\ p}$는 true σ^2의 unbiased estimator ($\hat{\sigma}^2_{all\ p} = \frac{RSS_{all\ p}}{n - all\ p}$)
- **직관:** p개로 fitting한 $\hat{Y}_{i(p)}$가 Y_i의 unbiased estimator라면, $RSS_p \approx RSS_{all\ p} = (n - all\ p)\sigma^2$일 것!
- **Model Selection:** 변수를 추가하면 RSS_p가 감소하지만 $2p$가 패널티로 작용. 모든 p가 다 들어가면 $C_p = all\ p$. p개로 fitting했을 때 C_p가 모든 변수 개수 $all\ p$보다 작으면 "좋은" 모델(더 적은 변수로 unbiased를 달성했다는 거니까.)

2. Akaike's Information Criterion

- **AIC는 "penalized likelihood":** 모델의 파라미터 (β) 추정 방식이 MLE일 경우, 최대화된 Log-Likelihood에 변수 추가에 따른 패널티 항을 추가한 것이 AIC.

$$AIC = -2\ LogLikelihood + 2p$$

$$BIC = -2\ LogLikelihood + \log(n)p$$

- 선형회귀에서의 MLE:

$$L(\beta, \sigma^2|X, Y) = (\frac{1}{\sqrt{2\pi\sigma^2}})^n exp(-\frac{1}{2}\sum(\frac{Y_i - X_i^T\beta}{\sigma})^2)$$

$$\log L(\beta, \sigma^2|X, Y) = -\frac{n}{2}log(2\pi) - nlog\sigma - \frac{1}{2\sigma^2}\|Y - X\beta\|^2$$

- MLE Estimate는 베타는 OLS와 같으며 $\hat{\sigma}^2_{MLE} = \frac{RSS}{n}$ ($\hat{\sigma}^2_{OLS} = \frac{RSS}{n-p}$)

2. Akaike's Information Criterion

- 이걸 넣고 최대화된 Log-Likelihood를 보면

$$logL(\hat{\beta}, \hat{\sigma}^2|X, Y) = -n \log RSS + some\ constant$$

- 상수항을 무시하고 AIC와 BIC를 계산하면 회귀분석에서의 AIC와 BIC에 대한 식을 얻는다.

$$\begin{aligned} AIC &= -2p \log L(\hat{\beta}, \hat{\sigma}^2|X, Y) + 2p \\ &= 2n \log RSS + 2p \\ BIC &= 2n \log RSS + log(n)p \end{aligned}$$

- **Model Selection:** 변수 개수가 늘어나면서 주어진 데이터에 대한 Likelihood는 최대화되면서 RSS는 감소하지만 패널티 항에 AIC 증가(BIC는 이런 패널티가 더 크다). 여러 모델 중에서 AIC 값이 가장 낮은 모델을 고르자.

3. Adjusted R-squared

- R^2의 모티브는 전체 분산 중에서 회귀분석 식이 설명하는 분산의 비율

$$R^2 = 1 - \frac{\sum(Y_i - \hat{Y}_i)^2}{\sum(Y_i - \bar{Y})^2} = 1 - \frac{RSS}{TSS}$$

- 그러나 R^2은 암묵적으로 표본에서 구한 RSS와 TSS의 자유도가 모두 n이라고 가정하는데, 이러면 표본분산이 모분산의 unbiased estimator가 될 수 없다. 때문에 이를 조정한 것이 *Adjusted* R^2

$$\textit{Adjusted } R^2 = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

- **Model Selection:** 변수 p의 개수가 늘수록 RSS는 감소하지만 만일 추가한 변수가 RSS의 감소에 별 영향이 없다면 분자가 커지기만 한다. 가장 *Adjusted* R^2가 큰 모델을 선택하자.

SUBSET SELECTION

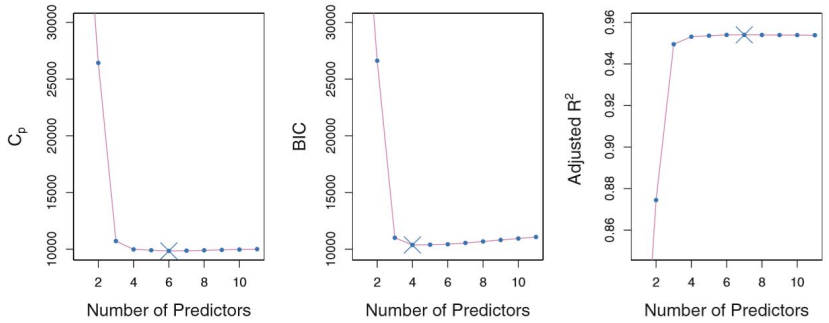


FIGURE 6.2. *C_p, BIC, and adjusted R^2 are shown for the best models of each size for the* `Credit` *data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.*

SUBSET SELECTION

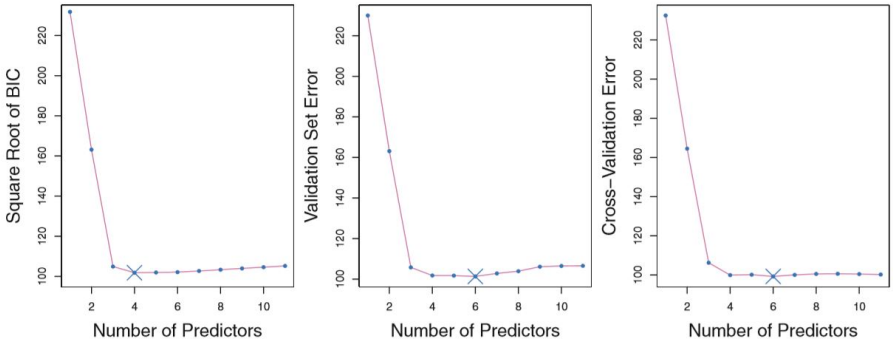


FIGURE 6.3. *For the* `Credit` *data set, three quantities are displayed for the best model containing d predictors, for d ranging from* 1 *to* 11. *The overall* best *model, based on each of these quantities, is shown as a blue cross.* Left: *Square root of BIC.* Center: *Validation set errors.* Right: *Cross-validation errors.*

SHRINKAGE METHODS: RIDGE AND LASSO

모든 변수에다가 fitting을 하긴 하는데, RSS가 조금 높아도 되니 β가 "쪼그라드는" 방법은 없을까?
Regularization

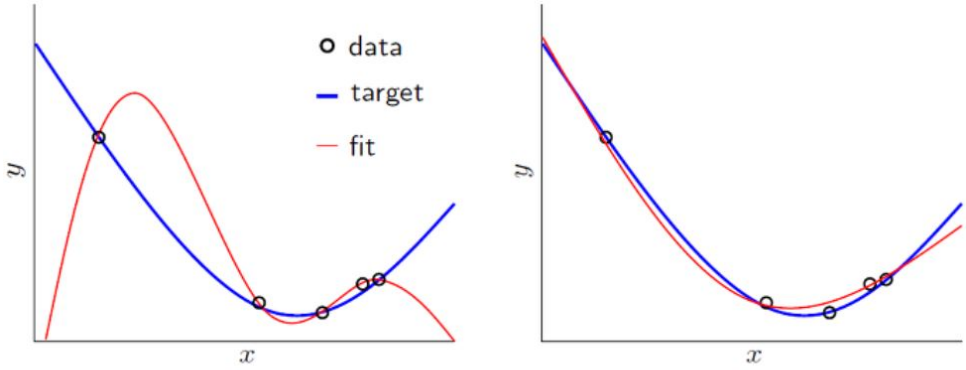
- **오캄의 면도날:** 중세 유명론의 대가 윌리엄 오브 오캄(William of Ockham, ca.1285-1349) 선생님께서는 다음과 같이 말씀하셨지요. (Principle of Parsimony)

  "*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 ϵ이 섞여있다. 만일 이를 고려하지 않고 ϵ까지 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

Regularization



(a) without regularization (b) with regularization

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

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 β의 크기에 상관없이 그냥 RSS가 가장 작은 β를 뱉어낸다. 여기에 β의 크기를 작게 하도록 Regularization term을 추가하면?

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

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

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 $(1/\sqrt{2}, 1/\sqrt{2})$)

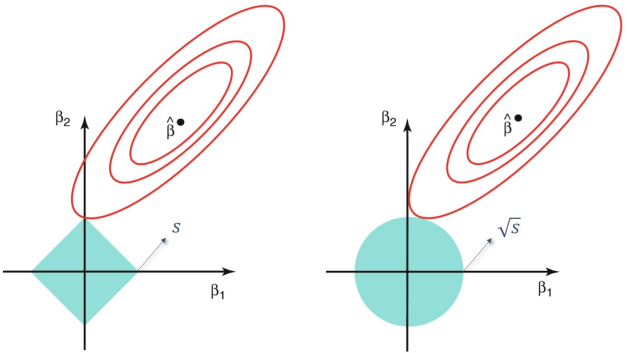


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

SHRINKAGE METHODS: RIDGE AND LASSO

Regularization: Lasso promotes sparsity of coefficients!
Lasso 방식을 쓸 때 모서리 해가 더 잘 나온다!

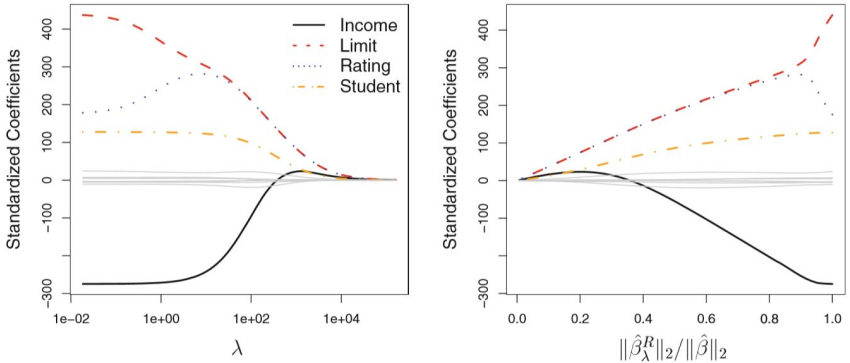


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

SHRINKAGE METHODS: RIDGE AND LASSO

Regularization: Lasso promotes sparsity of coefficients!
Lasso 방식을 쓸 때 모서리 해가 더 잘 나온다!

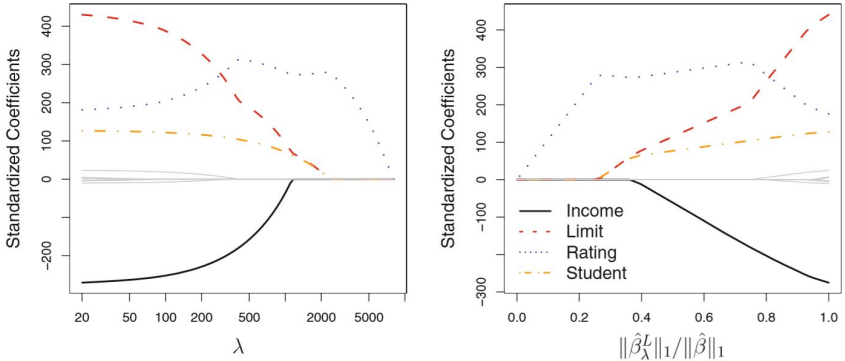


FIGURE 6.6. *The standardized lasso coefficients on the* `Credit` *data set are shown as a function of* λ *and* $\|\hat{\beta}_\lambda^L\|_1/\|\hat{\beta}\|_1$.

SHRINKAGE METHODS: RIDGE AND LASSO

Regularization: Lasso promotes sparsity of coefficients!

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

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

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

- 이를 만족하는 해는 다음과 같다.

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

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

- 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.

Regularization: Lasso promotes sparsity of coefficients!

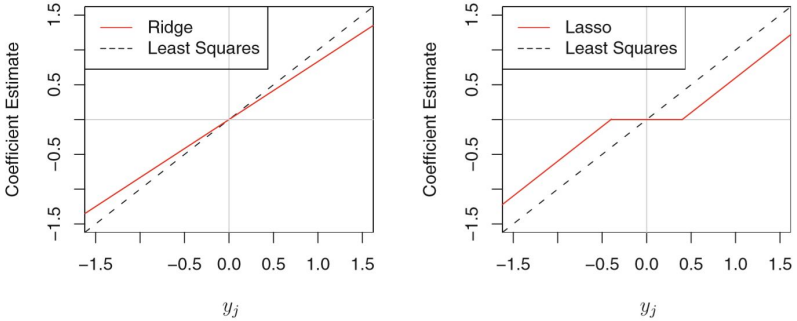


FIGURE 6.10. *The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and $\mathbf{X}$ a diagonal matrix with 1's on the diagonal.* Left: *The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates.* Right: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

Regularization: in Bayesian Lens

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

- 모수 θ인 확률분포를 따르는 확률변수 y를 생각해보자.

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

- θ를 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.

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

Regularization: in Bayesian Lens

- θ를 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 θ가 취할 수 있는 공간을 임의로 두 영역으로 분리해, H_0하의 단 하나의 값에서의 $P(y|\theta_{H_0})$를 보는 것)
  - 이에 반해 베이지안은
    1) θ에 대한 나의 사전 믿음과,
    2) 주어진 데이터에서 어떤 θ 값이 얼마나 likely한지를 종합적으로 판단해,
    3) 데이터에 의해 수정된 θ에 대한 사후 믿음, 즉 **확률 분포 전체**를 보여준다.[2]

[2]참조 링크: http://bjlkeng.github.io/posts/probabilistic-interpretation-of-regularization/id1

Regularization: in Bayesian Lens

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

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

Compare this with

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

Linear Regression의 맥락에서 생각한다면, (σ^2를 unknown but constant로 가정했을 때) β에 어떤 prior를 주냐에 따라 Ridge 혹은 Lasso!

Regularization: in Bayesian Lens

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

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

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

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 β가 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 λ가 높아진다(높은 기준!).

Regularization: in Bayesian Lens

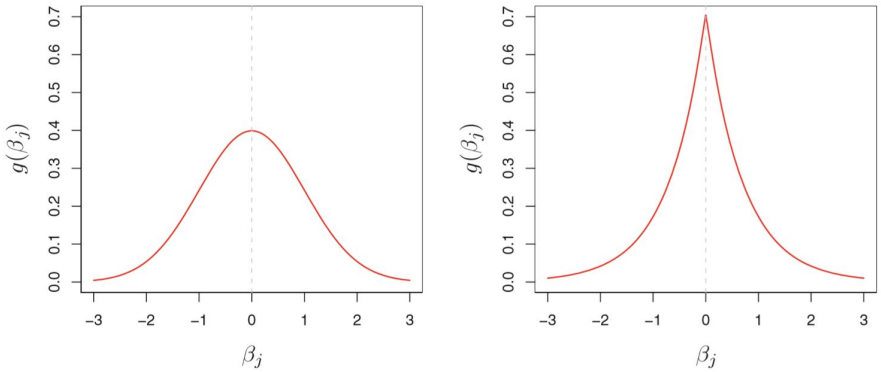


FIGURE 6.11. Left: *Ridge regression is the posterior mode for β under a Gaussian prior.* Right: *The lasso is the posterior mode for β under a double-exponential prior.*

Regularization: Selecting tuning parameter

λ는 어떻게 고르냐? CV 최소화하는 지점을 보면 되지!

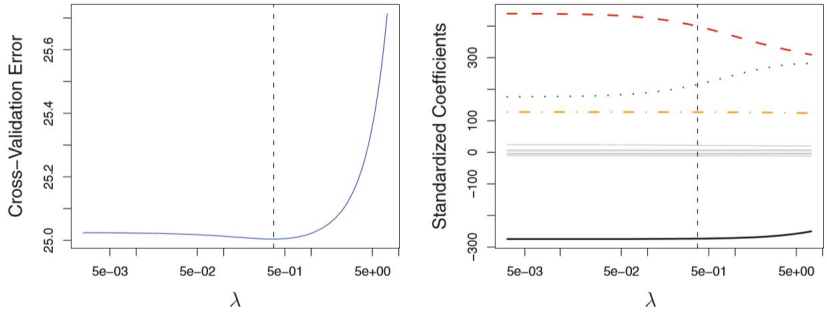


FIGURE 6.12. Left: *Cross-validation errors that result from applying ridge regression to the* `Credit` *data set with various value of* λ. Right: *The coefficient estimates as a function of* λ. *The vertical dashed lines indicate the value of* λ *selected by cross-validation.*

Regularization: Selecting tuning parameter

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

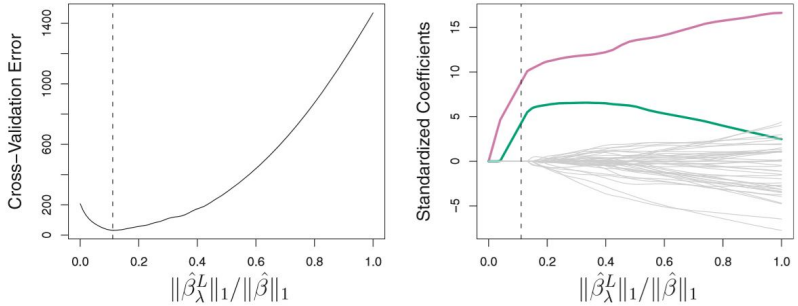


FIGURE 6.13. Left: *Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9.* Right: *The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

DIMENSION REDUCTION METHOD: PCA

PCA 직관 얻기

잠시 $Y_{n\times 1}$은 저리 치워두고, $X_{n\times p}$에 대해서만 생각해보자. 쉽게 말하면 $X_{n\times p}$은 n개의 p차원 벡터 뭉치, 즉 "p차원으로 표현된 데이터 더미"이다.

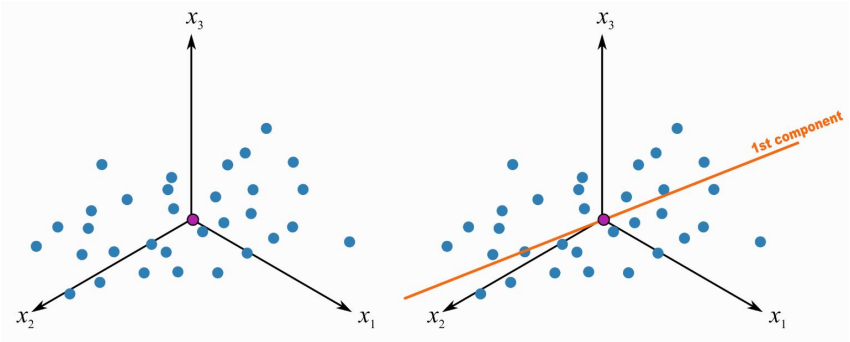
근데 말이다, 굳이 p차원이 다 필요할까? 어차피 우리에게 중요한 것은 $X_{n\times p}$가 담고 있는 "퍼짐"이다. 이 퍼짐, 즉 데이터의 분포를 p차원보다 더 낮은 차원으로 고스란히 담아낼 수 있다면? 그렇게 축소된 차원에서 $Y_{n\times 1}$에 대해 선형회귀를 할 수 있지 않을까? $\rightarrow$ **차원 축소!**

그렇다면 p차원의 데이터 더미를 어떻게 줄일 수 있을까? **데이터를 잘 설명하는 방향으로 새로운 축 (Principal Component)을 그려보는 것이 어떨까?**

1. 먼저 데이터가 가장 퍼져있는 쪽으로 선을 그어보고 (1^{st} Principal Component)
2. 거기에 **직교**하면서 그 다음으로 데이터가 퍼져있는 쪽으로 선을 긋기를 반복하고 (k^{th} PC)
3. 나머지 데이터 변동이 미미한 축들을 제낀다!

PCA 직관 얻기: 그림

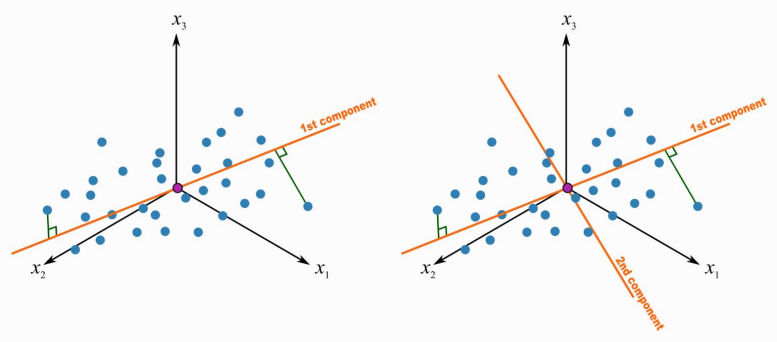
그림으로 설명해보자.[3]예컨대 $X_{n\times 3}$ 데이터는 아래와 같이 그릴 수 있다(편의를 위해 모두 centered 되었다고 가정). 여기에 원점을 지나는 직선을 긋되, 모든 데이터를 이 직선에 수직으로 내리꽂았을 때 찍히는 직선 상의 점들의 분산이 가장 크도록 그어보자!



[3]https://learnche.org/pid/latent-variable-modelling/principal-component-analysis/geometric-explanation-of-pca

PCA 직관 얻기: 그림

이제 원점을 지나면서 아까 그은 직선에 직교하는 선들을 고려해보자. 이 중에서, 아까와 똑같이, 모든 데이터를 이 직선에 수직으로 내리꽂았을 때 찍히는 직선 상의 점들의 분산이 가장 큰 놈을 선택하자!

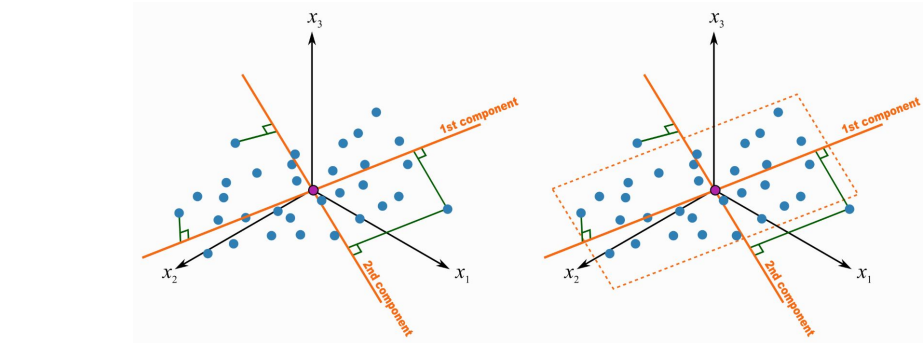


[3]https://learnche.org/pid/latent-variable-modelling/principal-component-analysis/geometric-explanation-of-pca

DIMENSION REDUCTION METHOD: PCA

PCA 직관 얻기: 그림

그럼 우리에겐 두 개의 직선, 즉 평면이 생긴다. 이 평면은 3차원의 데이터의 분산을 가장 잘 설명하는 방식으로 만들어졌기 때문에, 조금의 오차를 감수한다면 3차원 공간의 정보를 2차원 평면으로 담을 수 있는 것이다!



[3] https://learnche.org/pid/latent-variable-modelling/principal-component-analysis/geometric-explanation-of-pca

PCA 수식으로 보이기

이제 직관을 얻었으니 수학적으로 이걸 어떻게 하는지 함 보자.
(선형대수 배경이 많이 필요하니, 이해가 안 된다면 직관만 얻고 넘어가도 무방하다.)
아래와 같은 행렬식을 생각해보자. W_i는 서로 직교하는 unit vector라고 하자.
나중에 자세히 설명할테니 그냥 일단 함 보자.

$$\begin{aligned}\mathbf{T}_{n\times p} &= \mathbf{X}_{n\times p}\mathbf{W}_{p\times p} \\ &= \mathbf{X}[W_1, W_2, ..., W_p] \\ &= [\mathbf{X}W_1, \mathbf{X}W_2, ..., \mathbf{X}W_p]\end{aligned}$$

이걸 원소별로 풀어쓰면 다음과 같다.

$$\begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{np} \end{bmatrix} = \left[\begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1p} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline x_{n1} & x_{n2} & \cdots & x_{np} \end{array}\right] \left[W_1| \quad \cdots \quad |W_p\right]$$

DIMENSION REDUCTION METHOD: PCA

PCA 수식으로 보이기

$$\begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{np} \end{bmatrix} = \left[\begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{array}\right] \left[W_1 | \quad \cdots \quad | W_p\right]$$

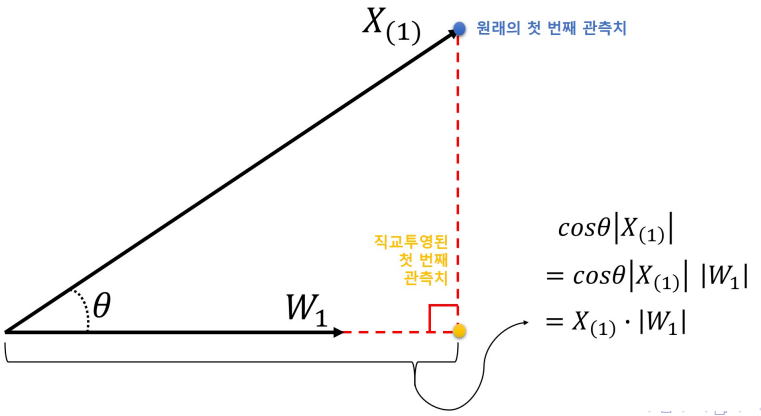
여기서 $\mathbf{T}_{n\times p}$의 첫째 열 t_{11}, t_{21}, t_{31} ... 들의 의미는 뭘까?

$$\begin{aligned} t_{11} &= X_{(1)} \bullet W_1 \\ t_{21} &= X_{(2)} \bullet W_1 \\ &\vdots \\ t_{n1} &= X_{(n)} \bullet W_1 \end{aligned}$$

즉 얘네들은 n개의 관측치 $X_{(i)}$들을 W_1에 직교투영하였을 때의 좌표인 것이다! 왜?

PCA 수식으로 보이기

두 벡터 A,B에 대해 dot product는 $A \bullet B = cos\theta \|A\| \|B\|$임을 알면, 다음의 그림을 이해할 수 있을 것이다. 즉 $X_{(1)} \bullet W_1$는 원점에서 출발한 W_1이 가리키는 축 위에 직교투영된 점의 좌표인 것이다!



PCA 첫 번째 PC 찾기

우리가 하고 싶은 것은, 전체 n개의 관측치 뭉치 $\mathbf{X}_{n\times p}$를 직교투영하였을 때의 분산이 가장 큰 직선 W_1을 찾는 것이다. 이를 수식으로 나타내면

$$\max_{W_1} \sum_{i=1}^{n} t_{i1}^2 = \max_{W_1} \|T_1\| = \max_{W_1} \|XW_1\| = \max_{W_1} W_1^T X^T X W_1$$

$$\text{s.t. } \|W_1\| = 1$$

위 조건에 대한 Lagrange Multiplier식을 W_1에 대해 미분하면 다음과 같다.

$$\mathcal{L}(W_1, \lambda_1) = W_1^T X^T X W_1 - \lambda_1 (W_1^T W_1 - 1) = 0$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = X^T X W_1 - \lambda_1 W_1 = 0$$

$$(\leftrightarrows X^T X W_1 = \lambda_1 W_1)$$

즉 W_1은 $X^T X$의 고유벡터, 그 중에서도 이때 목적함수의 극대화를 위해 가장 큰 고유값 λ_1에 대응하는 고유벡터인 것이다!

PCA 두 번째 PC 찾기

두번째 Principal Component W_2도 마찬가지이지만, W_1와 직교해야한다는 조건이 추가된다!

$$\max_{W_2} \|XW_2\| = \max_{W_2} W_2^T X^T X W_2$$
$$\text{s.t. } \|W_1\| = 1$$
$$\text{s.t. } W_2^T W_1 = 0$$

위 조건에 대한 Lagrange Multiplier식을 W_2에 대해 미분하면 다음과 같다.

$$\mathcal{L}(W_1, \lambda_2, \alpha) = W_2^T X^T X W_2 - \lambda_2(W_2^T W_2 - 1) - \alpha W_2^T W_1 = 0$$
$$\frac{\partial \mathcal{L}}{\partial W_2} = X^T X W_2 - \lambda_1 W_2 - \alpha W_1 = 0$$
$$\leftrightarrows W_1^T X^T X W_2 - \lambda_1 W_1^T W_2 - \alpha W_1^T W_1 = 0$$
$$= \lambda_1 \cdot 0 - 0 \cdot \lambda_1 - \alpha = 0$$

때문에 두 번째 조건식($W_2^T W_1 = 0$)이 날라가버린다.

PCA 두 번째 PC 찾기

두번째 조건식이 없으므로 W_2는 X^TX의 두 번째로 큰 고유값에 해당하는 고유벡터이다.

$$\max_{W_2} \|XW_2\| = \max_{W_2} W_2^T X^T X W_2$$
$$\text{s.t. } \|W_1\| = 1$$

위 조건에 대한 Lagrange Multiplier식을 W_1에 대해 미분하면 다음과 같다.

$$\mathcal{L}(W_2, \lambda_2) = W_2^T X^T X W_2 - \lambda_2 (W_2^T W_2 - 1) = 0$$
$$\frac{\partial \mathcal{L}}{\partial W_2} = X^T X W_2 - \lambda_2 W_2 = 0$$
$$(\leftrightarrows X^T X W_2 = \lambda_2 W_2)$$

PCA의 의미

결론은, $\mathbf{X_{n\times p}}$가 주어졌을 때 이 데이터 더미의 Principal Component는 바로 $(\mathbf{X}^T\mathbf{X})_{p\times p}$의 고유벡터인 것이다. 이게 무슨 뜻인가?

- $\mathbf{X}^T\mathbf{X}$의 관점: Sample Covariance Matrix
  $S = \frac{1}{n-1}(\mathbf{X}^T\mathbf{X})_{p\times p}$는 $\mathbf{X_{n\times p}}$의 표본 공분산행렬이다. 즉 전체 $\mathbf{X_{n\times p}}$의 분산이 가장 큰 직선 방향은 S의 Principal Eivenvector이며, 그 퍼진 정도는 S의 Principal Eigenvalue로 볼 수 있다!

$$\text{Sample Covariance Matrix} = \begin{bmatrix} \frac{\sum(X_{i1}X_{i1})}{n-1} & \frac{\sum(X_{i1}X_{i2})}{n-1} & \cdots & \frac{\sum(X_{i1}X_{ip})}{n-1} \\ \frac{\sum(X_{i2}X_{i1})}{n-1} & \frac{\sum(X_{i2}X_{i2})}{n-1} & \cdots & \frac{\sum(X_{i2}X_{ip})}{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\sum(X_{ip}X_{i1})}{n-1} & \frac{\sum(X_{ip}X_{i2})}{n-1} & \cdots & \frac{\sum(X_{ip}X_{ip})}{n-1} \end{bmatrix}$$

DIMENSION REDUCTION METHOD: PCA

PCA의 의미 (SVD를 모르면 Skip)

- **X**의 관점: Singular Value Decomposition
  $\mathbf{T} = \mathbf{XW}$에서 $\mathbf{W}$가 Orthogonal Matrix이므로, $\mathbf{TW^T} = \mathbf{X}$ 으로 쓸 수 있으며, SVD에 의해 $\mathbf{T} = \mathbf{U\Sigma}$ 이다. 즉 1st PC에서의 X의 좌표 XW_1는 $\mathbb{R}^n$의 직교기저 U의 첫 번째 기저를 singular value만큼 연장한 $U\Sigma_1$와 같다는 것!

$$\textbf{SVD of } \mathbf{X_{n\times p}} = \mathbf{U_{n\times n}\Sigma_{n\times p}W^T_{p\times p}}$$

where

$\mathbf{U}$: *Orthonormal Basis of* $\mathbb{R}^n$

$(U_i = XV_i/\sigma_i$ *if available*$)$

$\mathbf{\Sigma}$: *Diagonal Matrix with Eigenvalues of* X^TX

$\mathbf{W}$: *Orthonormal Basis of* $\mathbb{R}^p$

$(W_i =$ *Eigenvector of* $X^TX)$

DIMENSION REDUCTION METHOD: PCA

PCA Regression: 중요한 M개만 추려내자!

- $\mathbf{T} = \mathbf{XW}$에서 p개의 PC를 뽑아냈다. 이 중에서 **실제로 Y와 연관이 있는지 없는지와 상관 없이** 분산이 큰 (즉, $\|XW_i\|$이 큰) $M(< p)$개의 PC에 대해서만 OLS fit을 하자.
- **가정:** "components with small variance are unlikely to be important in regression"
  즉 직교투영된 데이터가 많이 퍼진 PC들만이 Y와 연관이 있을 것이다.
- **PCA를 하기 이전에 주의할 점은 데이터를 모두 standardized 하여 같은 unit로 만들어야 한다는 것. 이는 Ridge와 Lasso의 경우도 마찬가지.** PCA의 경우 변수마다 스케일이 다르면 특정 방향으로 X의 분포가 찌그러져 늘어난다. Ridge와 Lasso의 경우도 X_i의 scale에 따라 그에 따른 β_i의 크기도 영향을 받으니 제대로 패널티가 들어가지 않을 수 있다.

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \bar{x}_j}{s_j}$$

- 또한 PCR은 Lasso, Ridge와 같은 **변수 선택법이 아니다!** M개만 쓴다 해도, $T_{n\times M} = X_{n\times p}W_{p\times M}$을 보면, 선택된 M개의 PC도 결국 X의 선형결합이기 때문. 다만 데이터에 명시적으로 나오지 않은 **"Latent Variables"**을 구성하는 것.

PCA Regression: 중요한 M개만 추려내자!
교재의 예시. 2개의 설명변수에 대해 PCA를 진행하였다.

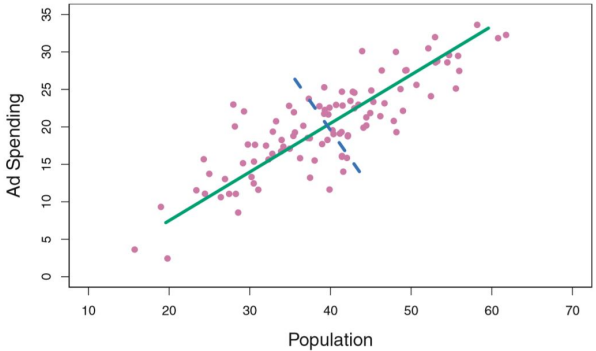


FIGURE 6.14. *The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

PCA Regression: 중요한 M개만 추려내자!

1st PC에 비해 2nd PC가 데이터의 변동을 잘 설명하지 못함. 이럴 때 1st PC에 대해서만 fit한다.

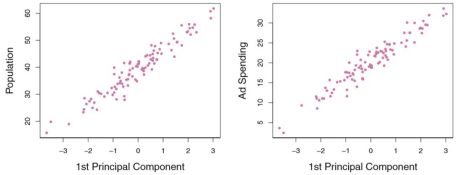


FIGURE 6.16. *Plots of the first principal component scores z_{i1} versus* pop *and* ad. *The relationships are strong.*

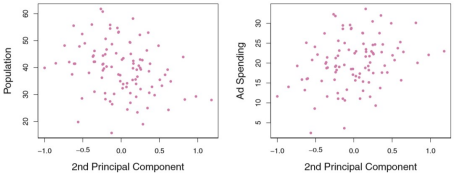


FIGURE 6.17. *Plots of the second principal component scores z_{i2} versus* pop *and* ad. *The relationships are weak.*

DIMENSION REDUCTION METHOD: PCA

PCA Regression: 중요한 M개만 추려내자!

PCR의 예시. PC를 추가할수록 모델의 분산은 늘어난다. **PCR을 하는 가정은 소수의 PC만으로 Y를 잘 설명, 즉 Bias를 줄일 수 있다는 것**. 이 경우 데이터의 설명에 5개의 PC만 필요하므로 Bias가 확 준다.

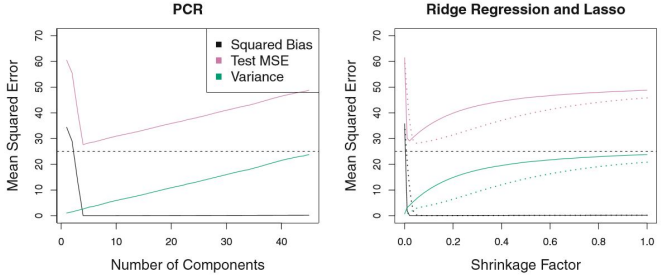


FIGURE 6.19. *PCR, ridge regression, and the lasso were applied to a simulated data set in which the first five principal components of X contain all the information about the response Y. In each panel, the irreducible error $Var(\epsilon)$ is shown as a horizontal dashed line.* Left: *Results for PCR.* Right: *Results for lasso (solid) and ridge regression (dotted). The x-axis displays the shrinkage factor of the coefficient estimates, defined as the ℓ_2 norm of the shrunken coefficient estimates*

DIMENSION REDUCTION METHOD: PCA

PCA Regression: 중요한 M개만 추려내자!
이 경우 데이터의 설명에 많은 PC가 필요하도록 되어있기에 Bias가 잘 안 줄어든다.

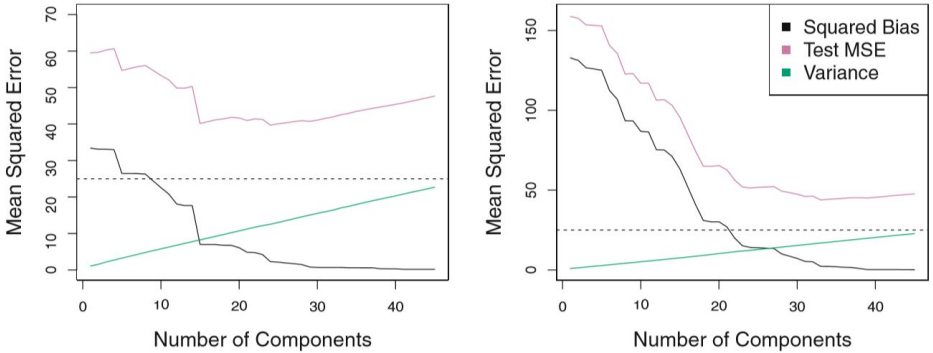


FIGURE 6.18. *PCR was applied to two simulated data sets.* Left: *Simulated data from Figure 6.8.* Right: *Simulated data from Figure 6.9.*

DIMENSION REDUCTION METHOD: PCA

PCA Regression: 중요한 M개만 추려내자!

그러면 M 개수는 어떻게 정하나? Cross-Validation 결과 보고 정한다!

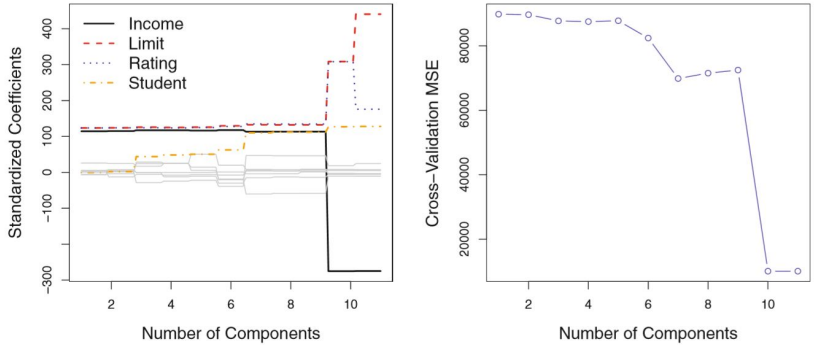


FIGURE 6.20. Left: *PCR standardized coefficient estimates on the* `Credit` *data set for different values of* M. Right: *The ten-fold cross validation MSE obtained using PCR, as a function of* M.

CONSIDERATIONS IN HIGH DIMENSIONS

설명변수가 많다고 무조건 좋은가? (Curse of Dimensionality)

- 설명변수 개수(p)가 표본 수(n)에 비해 상대적으로 많은 경우를 생각해보자. 일단 데이터가 있으니 이걸 모델에 다 집어넣으면 뭔가 더 정교한 모델이 나올거같지만 아니다. 물론 원칙적으로 Y의 설명에 큰 기여를 하는 설명변수를 모델에 포함하면 모델의 Bias가 줄어든다. 그러나 변수를 추가하면 언제나 모델의 분산이 증가한다는 것을 명심해야 한다. 그러니 별 설명도 못하는 애들을 잔뜩 포함시켜봤자 분산만 높이는 꼴이 되는 것!
- 특히 n에 비해 p가 상대적으로 많으면 Multicollinearity 문제가 심각해진다. 실제로는 서로 관련이 없지만 관측치가 적다 보니, 즉 우리가 볼 수 있는 차원이 제한적이다 보니 마치 상관관계가 높은 (두 벡터가 비슷한 방향으로 가는) 것처럼 보이기 때문.
- 그렇게 될 경우 p개의 n차원 벡터로 n차원 공간을 span하기가 굉장히 쉬워져, 주어진 sample에서는 fit이 참 잘 나온다. 이러면 σ^2의 추정치인 MSE가 실제보다 낮아져 사실상 의미가 없다. 때문에 R^2는 물론 MSE에 의존하는 AIC, BIC, C_p 이런 것들도 의미가 없어진다.
- **때문에 차원이 높은 경우는 모델의 성능을 보고할 때 다른 sample에서의 test MSE나 Cross-Validation Error를 써야한다!**