ISL_Chapter 6. Linear Model Selection and Regularization

우현수(19.11.09)

6.1 Subset Selection
6.2 Shrinkage Methods
6.3 Dimension Reduction Methods
6.4 Considerations in High Dimensions

Introduction

$Y = \beta 0 + \beta 1X1 + \cdots + \beta pXp + E \qquad (6.1)$

simple linear model : still a good way of solving real-world problems
ways to improve simple linear model: replacing plain least squares with
alternative fitting procedures

-Prediction Accuracy:
예측치와 결과값이 선형관계라고 가정합니다.
n(data수) >> p(변수) ---> LSE(least squares estimates) low variance —-> perform well
n > p *not much larger n* —-> LSE overfitting —-> perform poorly
n < p *no unique least squares coefficient estimate* —-> infinite variance —-> cannot use

-Model Interpretability:
too many variables in multiple regression model —-> 종속변수에 영향을 미치지 못함
모델 해석을 쉽게 하기 위해 중요하지 않은 변수 제거 필요(계수 0으로)

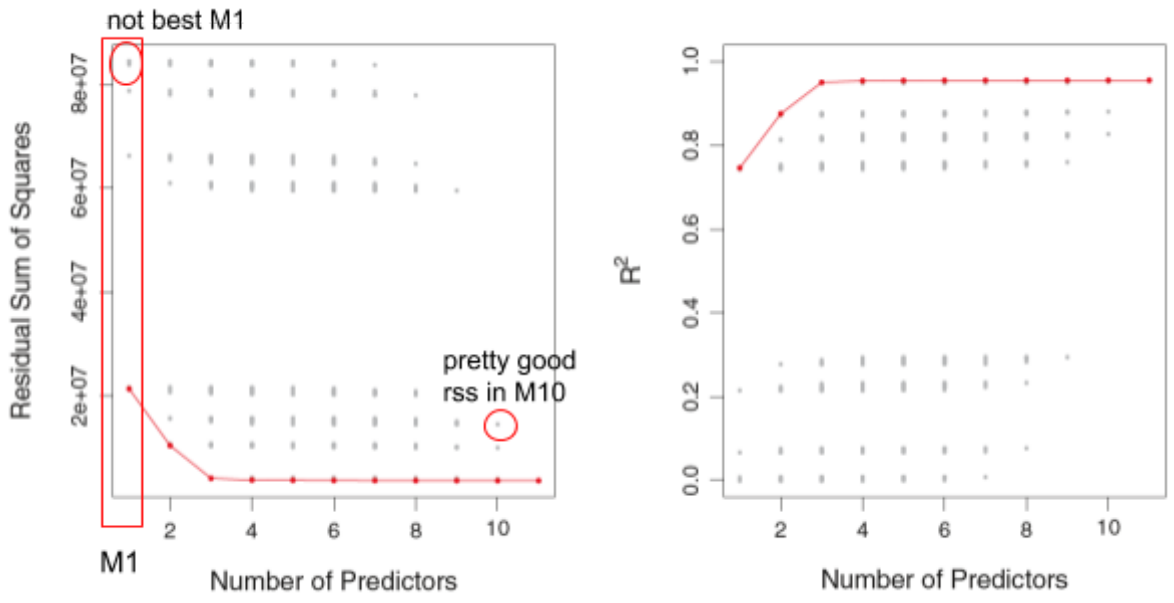
3 classes of methods
-subset selection: y에 영향을 주는 p predictors의 부분집합
-Shrinkage(Regularization): 모든 p-predictors에 적용. 분산을 줄이는 효과. Ridge, Lasso
-Dimension Reduction: 변수간의 조합. 모든 p-predictors를 M차원으로 투영. PCR

6.1 Subset Selection

$$\binom{p}{2} = \frac{p(p-1)}{2} = \frac{p!}{2!\,(p-2)\,!} = {}_pC_2$$

problem: best model from among the 2^p possibilities

-M0 : null model(no predictiors, sample mean)
-M1 ~ Mp : for k in range(p), fit all $\binom{p}{k}$ models that contain exact k predictors and pick the best model and call it Mk. Here, ‘best’ is the model having the **smallest RSS(largest R^2)**



왼쪽 figure의 많은 점: 전체 predictors에서 가능한 sub models들을 의미(그중 best가 선택)
*M10이 M4보다 좋은가? -> cross-validated prediction error, C_p, BIC, $adjusted\ R^2$

for other models that are not linear regression: deviance rather than RSS(generalization of RSS, RSS only applied to linear regression)
여기서는 편의를 위해 RSS를 쓰겠습니다.

-Stepwise Selection:
Forward Stepwise Selection, Backward Stepwise Selection

when p is large(over 40), best subset selection cannot be applied :
2^40 = 1,000,000,000,000, overfitting, high variance of the coefficient estimates

실제로 best subset selection을 고려할 때는 handful of predictors가 있을 때 씁니다.

위의 이유(computationa, statistical)로 stepwise 방법을 제시
: more restricted models

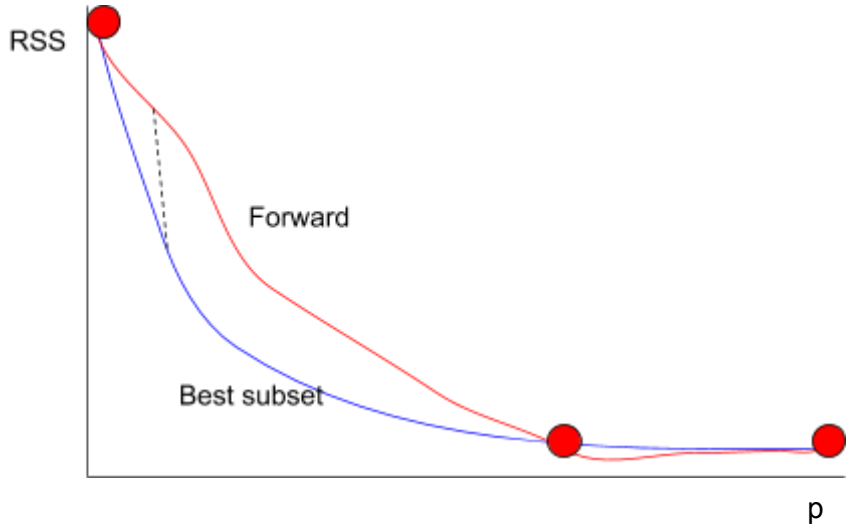
Forward Stepwise Selection VS Backward Stepwise Selection
: whether you start your model with no predictors or all predictors

Forward Stepwise Selection
: start a model with no predictors, add predictors to the model one by one til all of the predictors are in the model

best subset과 차이: 모든 2^p models를 고려하지 않고 nested된 구성

for k = 0, .., p-1: for M_k model, p-k models 중 best 하나를 추가한 것이 M_{k+1}
M0(no predictors)—>M1(one more predictor to M0)---> M2(consider all p-1 predictors to the M1 and choose the best model+ additional predictor) ….

computational advantage :1+ $\sum_{k=0}^{p-1}(p-k) = \frac{p(p+1)}{2}$ +1 (ex. p=20, only 211 models)
instead 2^p, $1+p+(p-1) \approx p^2$



그림ㅈㅅ

not guaranteed to find the best possible model
best model containing k predictors is **not a superset** of the best model containing k-1 predictors

# Variables	Best subset	Forward stepwise
One	rating	rating
Two	rating, income	rating, income
Three	rating, income, student	rating, income, student
Four	cards, income, student, limit	rating, income, student, limit

차이가 나는 이유: correlations between features, if no correlations, the two are exactly same

Backward Stepwise Selection
: 앞에서와 반대로 모든 p predictors에서 the least useful predictor를 하나씩 제거

for k = p, p-1...1: consider all k models containing all but one of the predictors in Mk, for a total of k-1 predictors

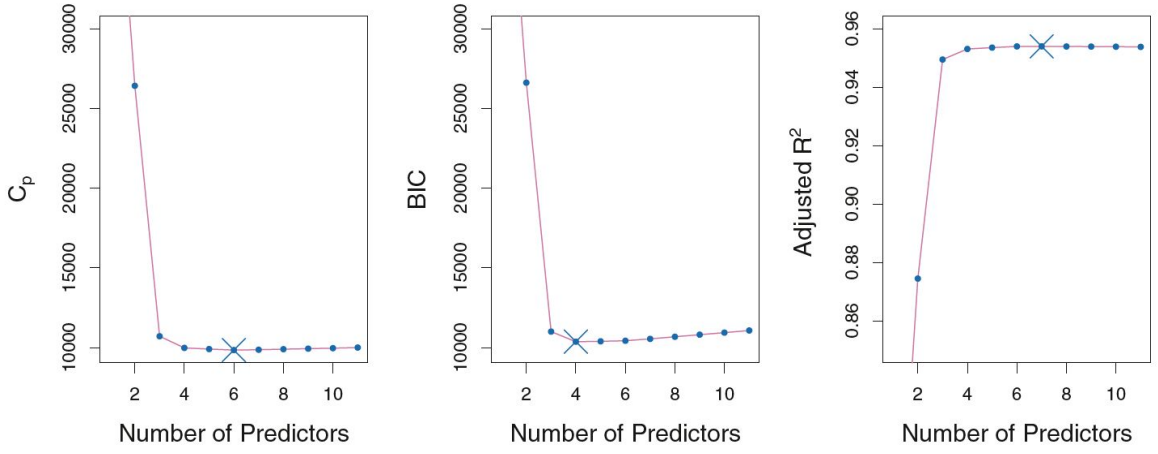
```
Forward stepwise selection
  R^2    Predictors
1 0.3215 ['CRBI']
2 0.4252 ['CRBI', 'Hits']
3 0.4514 ['CRBI', 'Hits', 'PutOuts']
4 0.4754 ['CRBI', 'Hits', 'PutOuts', 'Division_W']
5 0.4908 ['CRBI', 'Hits', 'PutOuts', 'Division_W', 'AtBat']
6 0.5087 ['CRBI', 'Hits', 'PutOuts', 'Division_W', 'AtBat', 'Walks']
7 0.5132 ['CRBI', 'Hits', 'PutOuts', 'Division_W', 'AtBat', 'Walks', 'CWalks']

Backrward stepwise selection
  R^2    Predictors
7 0.5136 ['AtBat', 'Hits', 'Walks', 'CRuns', 'CWalks', 'PutOuts', 'Division_W']
6 0.4997 ['AtBat', 'Hits', 'Walks', 'CRuns', 'PutOuts', 'Division_W']
5 0.4841 ['AtBat', 'Hits', 'Walks', 'CRuns', 'PutOuts']
4 0.4664 ['AtBat', 'Hits', 'CRuns', 'PutOuts']
3 0.4485 ['Hits', 'CRuns', 'PutOuts']
2 0.4148 ['Hits', 'CRuns']
1 0.3166 ['CRuns']
```

backward selection: n>p(full models can be fit)
forward selection: n<p, n>p 둘 다 가능

Choosing the Optimal Model
RSS, R^2 —> training error related—-> not suitable for selecting the best model
what we need is a model with low test error

Estimating test error: 2 approaches
-indirectly adjust: to the training error to account for the bias due to overfitting
-directly estimate : validation set, cv

<indirectly adjust>
$C_p,\ BIC,\ adjusted\ R^2$
: adjust training error for the model size, can be used to select among a set of models with different numbers of variables



중간 그래프를 보면 4 이후부터 상향하고 있습니다. 이 사실을 포함해 세 그래프를 모두 확인 후 알맞는 predictor를 정할 수 있습니다.

C_p: 모델이 갖는 오차와 변수의 갯수를 갖고 RSS 추정(n>p일 때 사용). 가장 작은 값을 선택

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

d = total number of parameters (including intercept)

$\widehat{\partial}^2$ = estimate of the variance of the error associated with each response measurement

2d$\widehat{\partial}^2$ 는 RSS를 제약하는 역할을 합니다. 모델에 들어가는 변수가 많을수록 모델 정확성이 떨어지게 됩니다.

*AIC(아카이케 정보 기준): 원래는 -2logL + 2d(L: maximized value of the likelihood function for the estimated model)

$$\mathrm{AIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

Y = β0 + β1X1 + · · · + βpXp + E (6.1)에 대한 AIC

-2logL = RSS/sigma^2

여기서 Cp와 proportional 한 관계로 같은 역할을 합니다. 선형모델 외 모델에서 AIC와 Cp는 같지 않습니다.

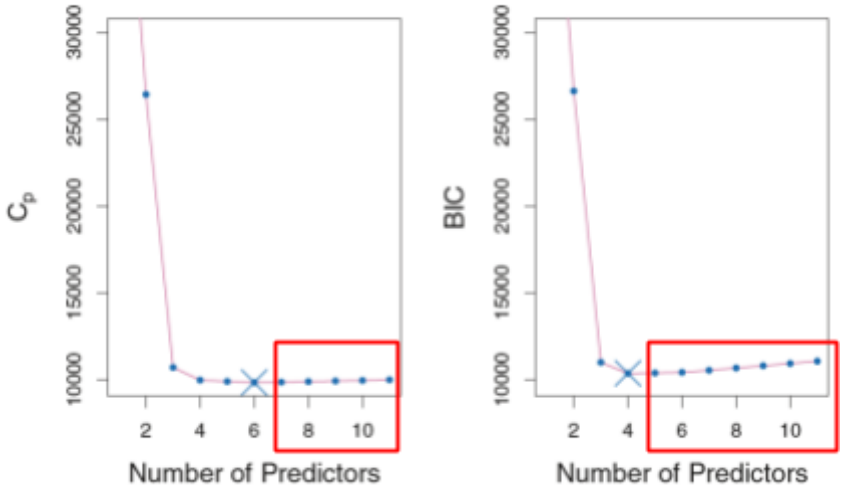
BIC(베이지안 정보기준): 형태는 Cp와 유사. 가장 작은 값을 선택

$$\mathrm{BIC} = \frac{1}{n\hat{\sigma}^2}\left(\mathrm{RSS} + \log(n)d\hat{\sigma}^2\right)$$

n은 number of observations

$$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

log n> 2 for any n > 7—-> BIC가 좀 더 큰 값을 가짐—-> heavier penalty on models with many variables —-> smaller model selection than Cp



$Adjusted\ R^2$: 모델 정확도를 위해 많이 쓰는 R^2가 변수와 관측치 수가 반영되지 않아 과적합(변수 갯수와 R^2가 비례)되는 문제가 있었는데 adjusted R^2에서는 변수의 수를 반영(pays a price)

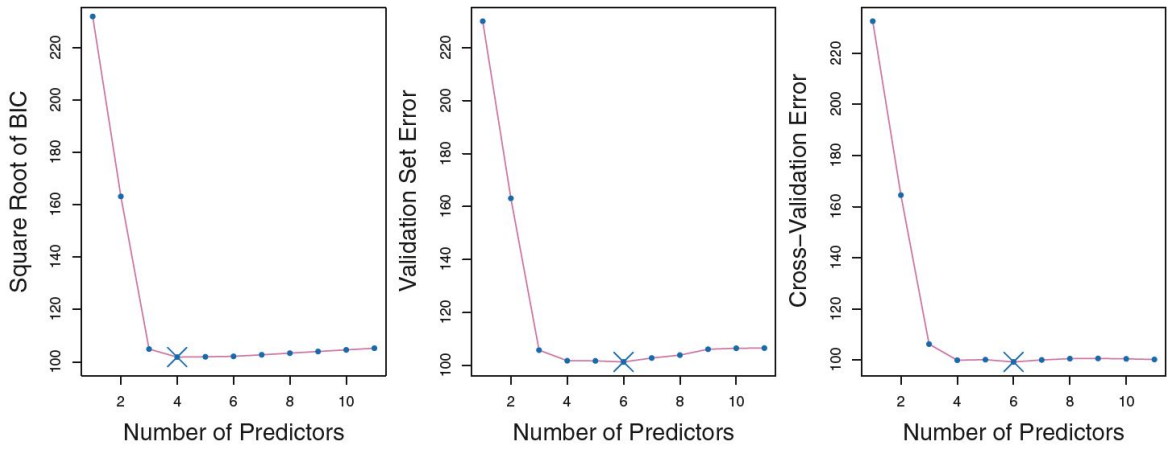
$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}$$

참고) $R^2 = 1 - \frac{RSS}{TSS}$

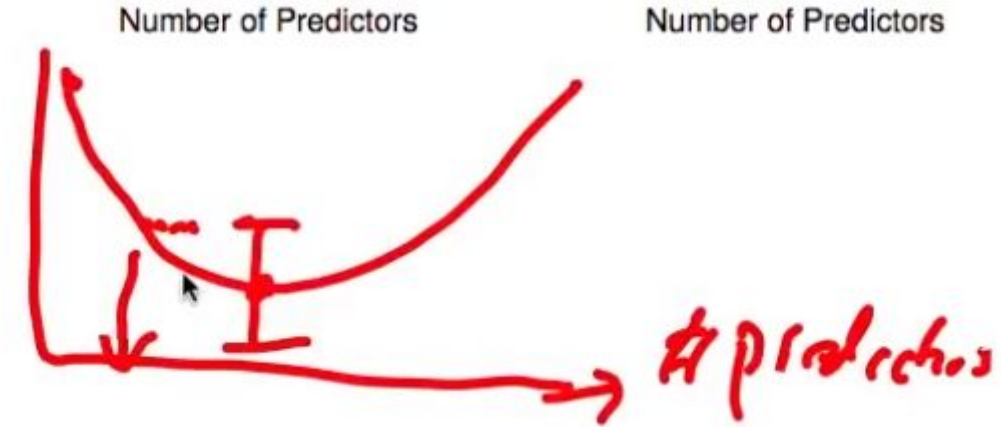
predictors numbers가 달라도 비교 가능합니다.

d: number of variables in the model we consider. when d is large, d+1 is large, ending up bigger RSS, which would result in smaller R^2.
변수가 많아도 페널티가 없었던 R^2와 다르게 adjusted R^2는 페널티를 부여합니다. adjusted R^2가 잘 나온다는 것은 그만큼 모델이 데이터를 잘 설명함을 의미합니다. 또한 predictor의 수에 관계없이 model들을 비교할 수 있습니다. n<p일 때도 사용 가능합니다.

<directly estimate>
validation and CV
: doesn't require an estimate of the error variance sigma^2(large numbers of features일 경우, d를 확정할 수 없을 때 유용)



one-standard-error-rule: fewer predictors(in the graph below, X is the number of predictors) are better and simpler model



6.2 Shrinkage Methods

-Ridge, Lasso

-Ridge regression: 계수에 대한 패널티 추가, L2 normalization

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

minimize the value in the parenthesis

In contrast, ridge regression coefficient estimates 베타햇^R are the values that minimize

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

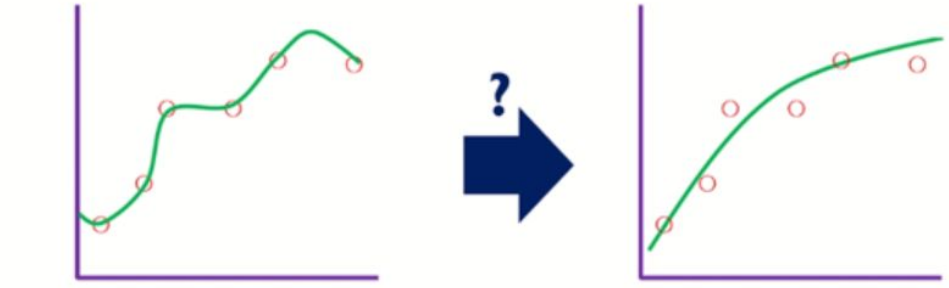
tuning parameter *
sum of the squares of coef

RSS 최소화

model fit + coef penalty

coef가 커지는 걸 제약
non zero coef에 더 큰
비용을 감수

lambda(>=0): tuning parameter
이 페널티가 shrinkage penalty입니다.
lambda가 작아지면 beta가 커지고 lambda가 커지면 beta가 작아집니다.

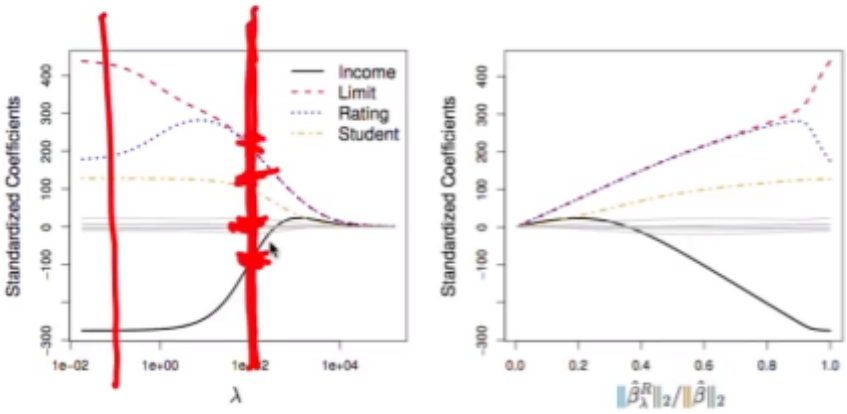


$\beta_0+\beta_1x+\beta_2x^2+\beta_3x^3+\beta_4x^4$ $\qquad$ $\beta_0+\beta_1x+\beta_2x^2$

$$\min_{\beta}\sum_{i=1}(y_i-\hat{y}_i)^2+5000{\beta_3}^2+5000{\beta_4}^2$$

$\beta_3\approx 0 \qquad \beta_4\approx 0$

beta3과 beta4에 5000이라는 큰 lambda값을 부여하면 최소화시킬 때 둘은 0이 되어야 최소 error값을 갖게 됩니다.



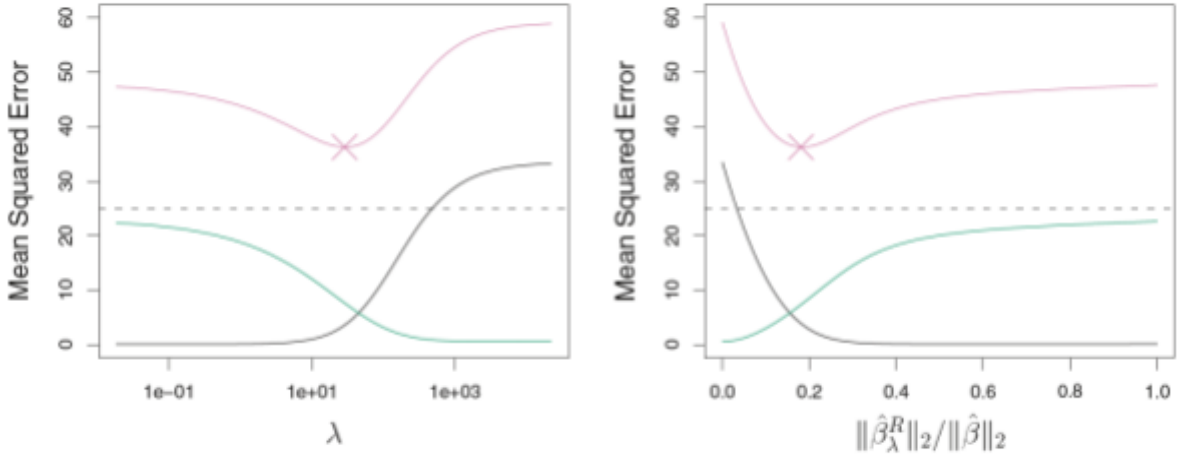
L2 norm of the squares of the coef / L2 norm of the least squares estimates

$\|B_1 \cdots B_p\|_2 = \sqrt{\sum B_j^2}$

L_2 norm

L2 norm ——> $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j{}^2}$

변수의 스케일링에 따라 베타값이 바뀌기 때문에 standardize predictors를 한 뒤에 ridge regression을 적용시켜줍니다.



lambda가 늘어날수록 bias가 일정수준 비슷하게 머물러있고 variance가 줄어듭니다.
ridge regression에서 coef를 shrinkage(zero)하는 것은 variance를 컨트롤합니다.

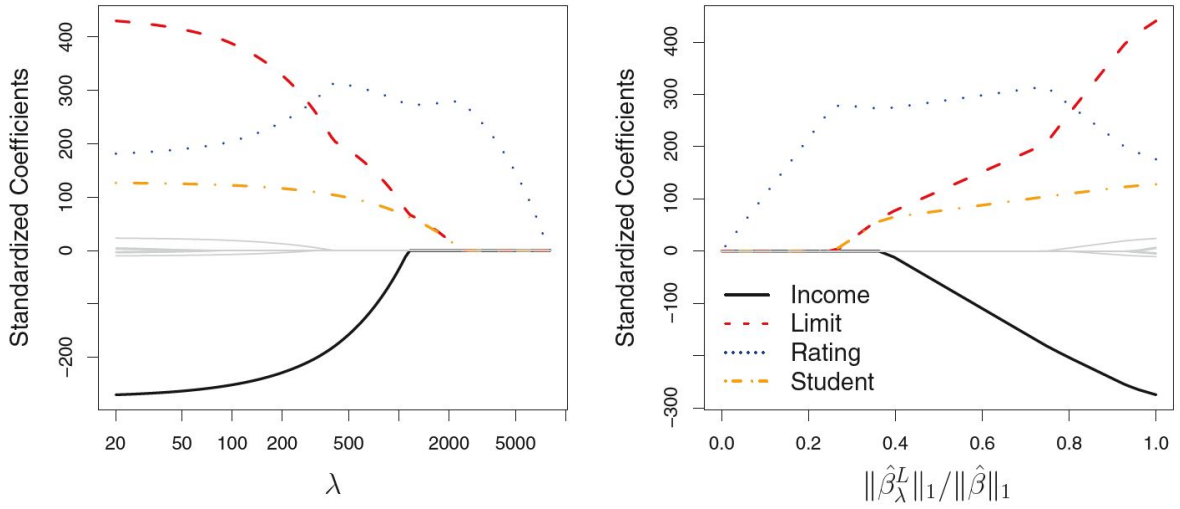
bias: black, variance: green, test error: purple

Ridge regression은 모든 p predictors를 최종 모델에 포함시킵니다. 즉 coef가 0으로 줄어들지만 0이 되진 않습니다.

-Lasso: 계수에 대한 패널티를 처리하는 방법이 다름(L1 norm)

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| \;=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

lasso 또한 coef estimates를 0으로 줄입니다. 튜닝 파라미터 람다가 충분히 크면 완전히 0으로 만듭니다.



왼쪽 그래프에서 회색선이 실제로 0이 되는 것을 확인할 수 있습니다.

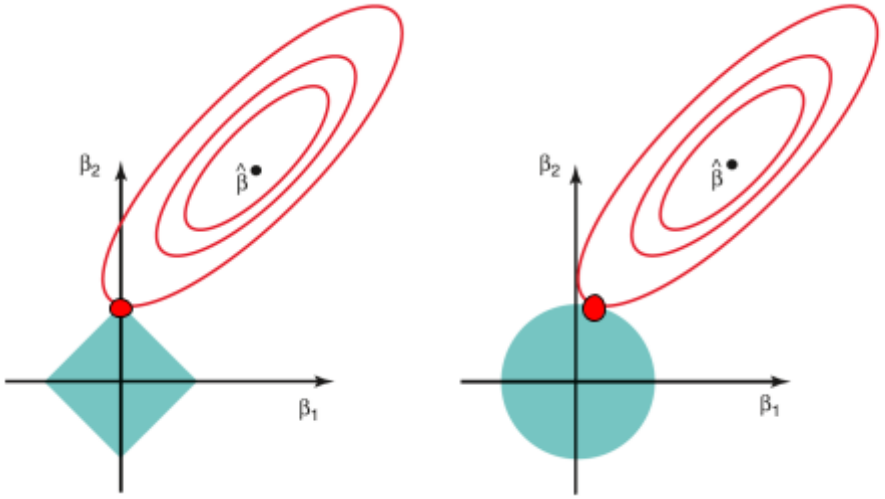
*importance of feature selection
예) 의료데이터 features(p) 30,000인 disease 진단 테스트 키트: lasso를 사용하면 p=6개로 확 줄일 수 있음. interpretable subset을 만들 수 있는 sparse model에 적합

-The variable Selection Property of the Lasso(s=budget)

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\}\quad\text{subject to}\quad\sum_{j=1}^{p}|\beta_j|\le s \tag{6.8}$$

and

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\}\quad\text{subject to}\quad\sum_{j=1}^{p}\beta_j^2\le s, \tag{6.9}$$



p = 2

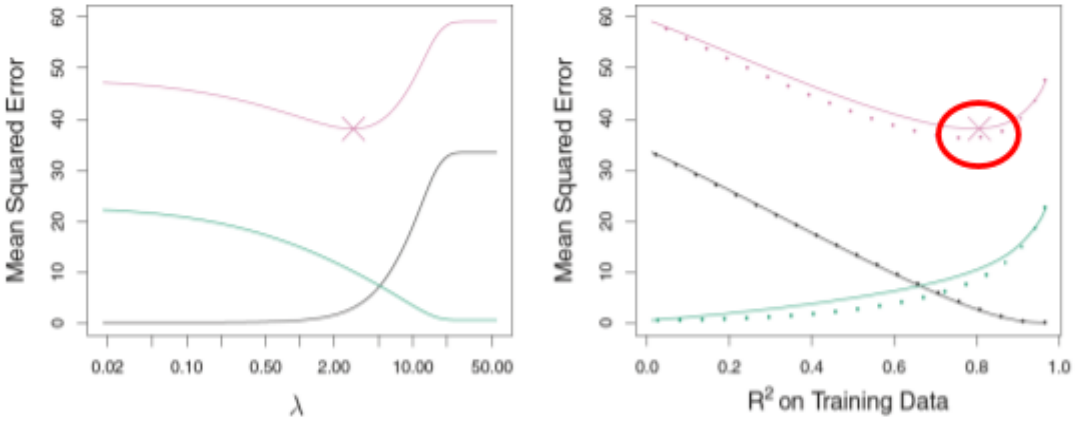
contours of RSS가 커지는 모양

budge의 모양이 녹색도형

solution은 둘이 만나는 지점

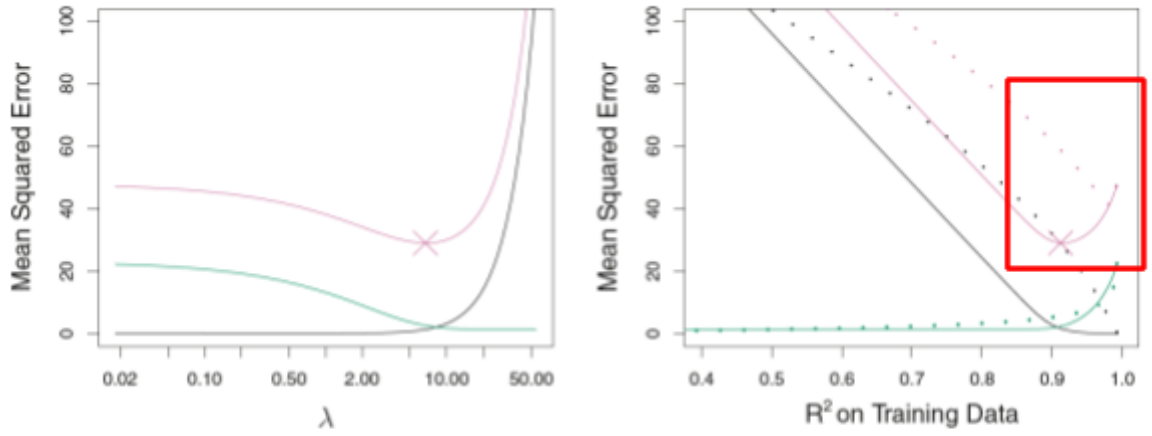
왼쪽이 Lasso(0이 됨), 오른쪽이 Ridge(0으로 가지만 0이 되진 않음)

추정식(등고선)과 초록색 도형(norm)



오른쪽: ridge가 좀 더 좋다. true model은 sparse하지 않음.
45 variables, non zero coef

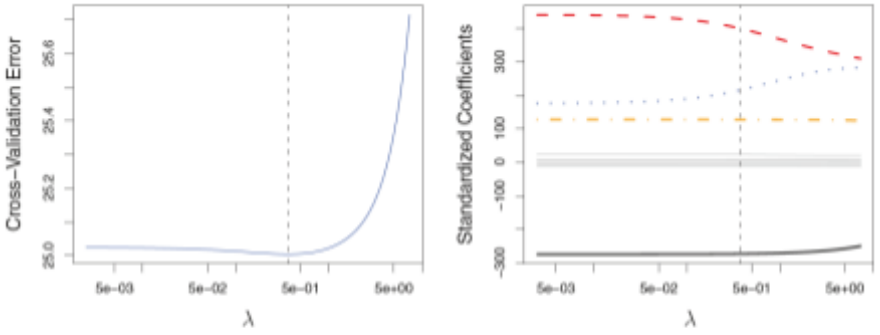
점선: Ridge
실선: Lasso
black: bias
green: variance
purple: test MSE



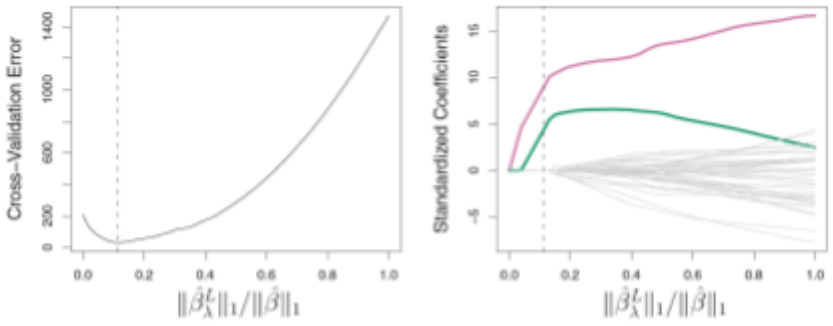
sparse model(only 2 predictors)
실선 Lasso의 결과가 좀 더 좋음.

true model : dense(Ridge), sparse(Lasso)

-Selecting the Tuning Parameter for Ridge , Lasso



CV error가 최소화되는 지점의
Standardized Coef값을
확인해봄



왼쪽: CV curve(U shape) 0.1쯤 최솟값

오른쪽: 같은 값으로 Standardized Coef를
살펴보니 2개 coef만 빼고 나머지는 모두 0인
것을 알 수 있음.

6.3 Dimension Reduction Methods(차원축소)
지금까지 원래 데이터의 predictors를 사용했지만 여기서는 transformed된 predictors를 사용합니다. 축소한 차원 M을 사용하므로 M <p 인 m predictors를 사용합니다.

X1, X2, X3...Xp —-> Z1, Z2, Z3...ZM

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, \ldots, n,$$

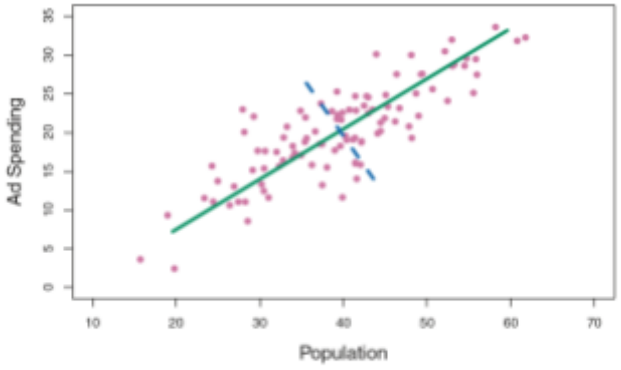
$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij},$$

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$$

linear combination의 베타가
세타와 파이의 새로운 조합으로 바뀜

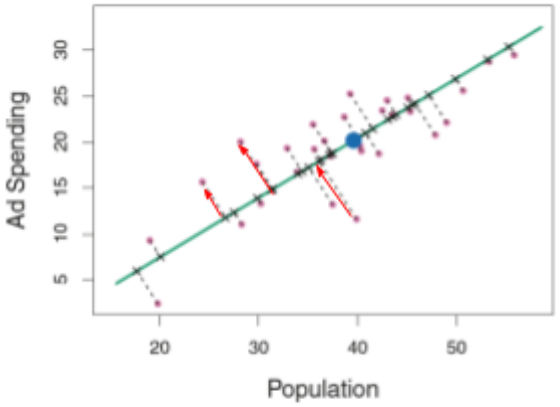
기존 linear model의 coef에 constraint가 가해진 것입니다. 이렇게 나온 모델은 low variance, low bias

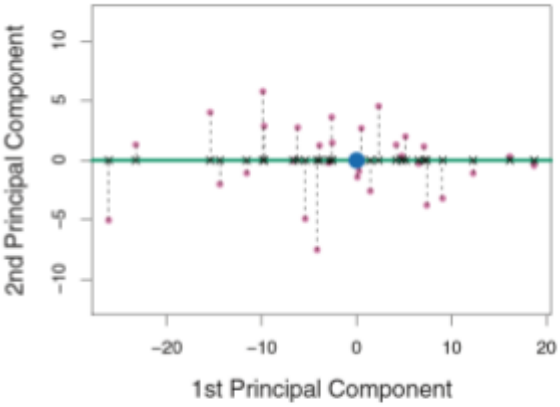
-PCA(주성분분석)



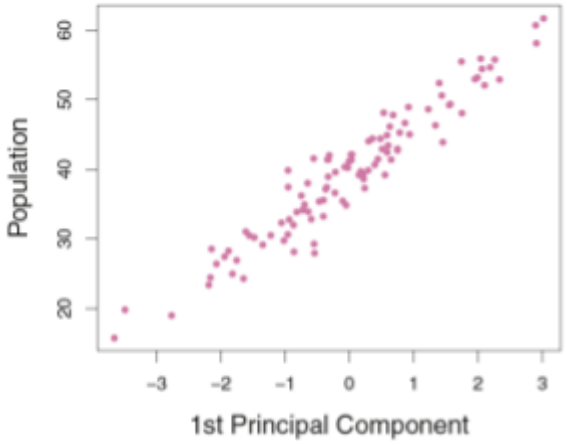
가장 주로 나타나는 direction은 초록색
: first component direction

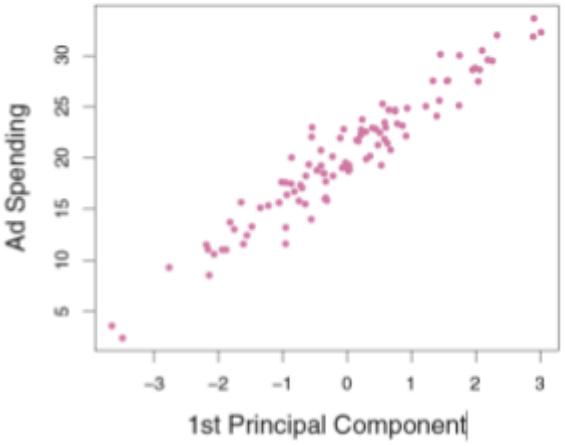
가장 uncorrelated direction은 파란색



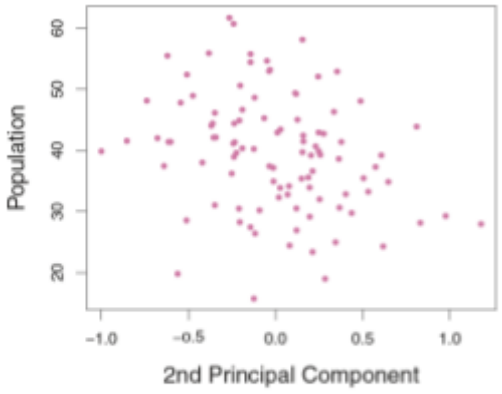


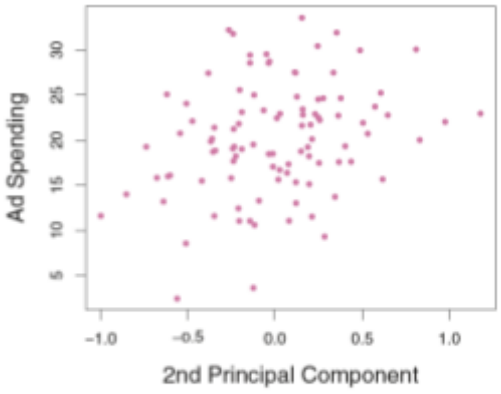
green line: first principal component,
dots distance small as possible



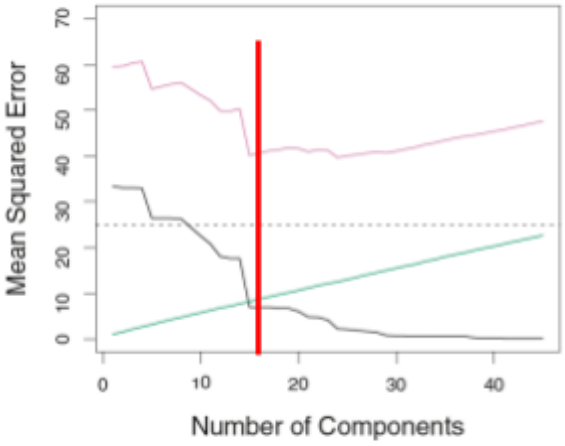


population, ad spending을 둘 다 쓰지 않고
1st principal component를 씀->model's new predictor

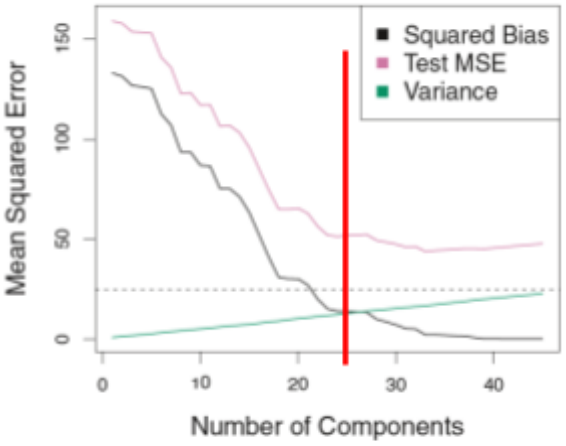




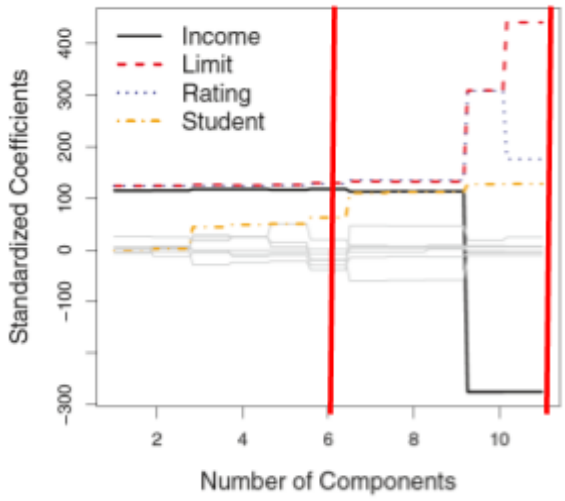
population, ad spending : 2nd principal
component little relationship

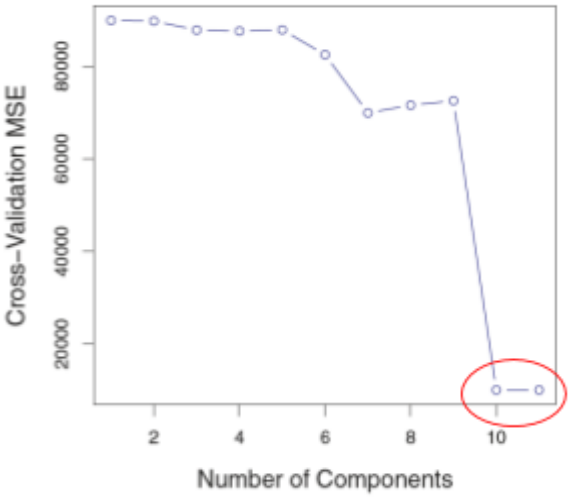


차원을 늘릴수록 bias를 줄어들지만
variance는 늘어남.
MSE = bias+variance (U shape)
18차원에서 가장 작음.



MSE가 줄어들다가 크게 변동이 없는
25차원에서 가장 simple





PCA에서 CV MSE는 원래 데이터가 가장
낮은 수치를 보임.

PCR(주성분 회귀분석)
: 종속변수와 독립변수의 관계가 낮을수록 변수 X만이 축소된 결과

PLS
: X와 Y의 연관성까지 고려

1. X, Y의 표준화
2. Y와 X_j의 단순회귀계수를 가중치로 활용한다. (이를 θ_{j1}로 표현한다.)
3. $Z_1 = \sum_{j=1}^{p} \theta_{j1} X_j$ 를 구한다.
4. $Z_2 = \sum_{j=1}^{p} \theta_{j2} X_j$ 를 구한다. 이때 θ_{j2} 는 Z_1, X_j 와의 단순회귀계수를 의미한다. Z_2는 Z_1로 설명되고 난 나머지를 의미하며 Z_1과 직교한다.
5. $Z_1 \dots Z_m$ 을 구한 후 Y와 회귀분석을 한다.
6. 어느정도 축소할지에 대해서는 Cross Validation을 이용해 가장 최적의 축소 변수 갯수를 구한다.

부록)
* 왜 변수가 데이터보다 많으면 LSE를 쓸 수 없을까.
cost function(비용함수) J와 가설함수 h를 아래와 같이 가정.

$$h(x) = \theta_0 + \theta_1 X + \theta_2 X^2 + \theta_3 X^3 + \theta_4 X^4 + \theta_5 X^5$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i - y^i) \right)^2$$

2개의 데이터(1,1,0,0,0,1), (0,1,1,1,1,3)밖에 없는 상태에서

$$J(\theta) = \frac{1}{2} \left\{ \left(\theta_0 + \theta_1 + \theta_2 - 1 \right)^2 + \left(\theta_0 + \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 - 3 \right)^2 \right\}$$

비용함수 J(Θ)를 0으로 만들려면 수없이 많은 경우가 생김(infinite함).

$\theta_0 + \theta_1 + \theta_2 - 1 = 0 \Rightarrow \theta_0 + \theta_1 + \theta_2 = 1$

$\theta_0 + \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 - 3 = 0 \Rightarrow \theta_0 + \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 = 3$