Seoul National University
Dept. of Civil and Environmental Engineering

Instructor: Junho Song
junhosong@snu.ac.kr

M1586.002500 Information Engineering for CE Engineers

In-Class Material: Class 15

Linear Model Selection and Regularization (ISL Chapter 6)

1. Ridge regression

(a) In linear model, the least-squares-fitting estimates $\beta_0, \beta_1, \ldots, \beta_p$ by minimizing RSS

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

(b) **Ridge regression** is very similar to least squares, except that the coefficients are estimated by minimizing the following objective function

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

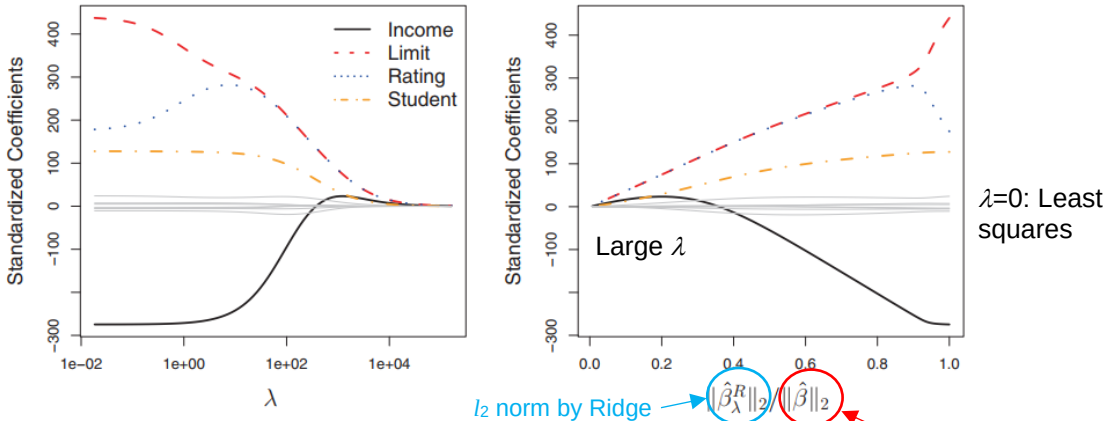
where $\lambda \geq 0$ is a *tuning parameter*.

(c) **Shrinkage penalty**: $\lambda\sum_{j=1}^{p}\beta_j^2$
Ridge regression seeks coefficient estimates that make the shrinkage penalty small as well as RSS → it has the effect of shrinking the estimates of β_j towards zero

(d) For best, selecting a good value for λ is critical → ***selecting tuning parameter***

(e) Why/how does Ridge regression improve over Least Squares? → Ridge regression's advantage over least squares is rooted in the *bias-variance trade-off*:
λ increases → bias increases (variance decreases)

(f) In general, ridge regression works well in the following situations
→ The least squares estimate have low bias and high variance
→ The number of variables is larger than the number of observations $p > n$



λ=0: Least squares

The Standardized ridge regression coefficients for Credit data

Instructor: Junho Song
junhosong@snu.ac.kr

(g) it is important to standardize the predictors as follows *before* the Ridge regression $\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}}$

While the standard least squares coefficients are *scale equivalent*, those by Ridge regression is NOT scale invariant because of the shrinkage penalty term.

(h) Computational advantage of Ridge regression over best subset selection (2^p models): construct a single model for each λ.

```
library(ISLR)
Hitters=na.omit(Hitters)

x=model.matrix(Salary~.,Hitters)[,-1]
#Creating x in which all qualitative variables are automatically
   transformed into dummy variables
y=Hitters$Salary

########################################
# Ridge fittings for a range of lambda #
########################################
library(glmnet)
grid=10^seq(10,-2,length=100)
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
# Perform ridge regression (alpha=0: ridge, alpha=1: lasso)
# lambda = 10^2 ~ 10^-2
# Note: glmnet, by default, automatically standardizes the predictors
# If you want to avoid standardization, use "standardize = FALSE"
dim(coef(ridge.mod))

#case 1: larger lambda(=11498)
ridge.mod$lambda[50]
coef(ridge.mod)[,50]
sqrt(sum(coef(ridge.mod)[-1,50]^2)) #beta_norm decreases

#case 2: smaller lambda(=705)
ridge.mod$lambda[60]
coef(ridge.mod)[,60]
sqrt(sum(coef(ridge.mod)[-1,60]^2)) #beta_norm increases

#can use 'predict' to obtain ridge regression for new lambda(=50)
predict(ridge.mod,s=50,type="coefficients")[1:20,]
```

2. The Lasso

(a) The penalty term in Ridge regression will shrink *all* of the coefficients towards zero, but it will *not* set any of them *exactly to zero* (unless λ is infinite)
→ all p predictors exist in the final models, which is an obvious disadvantage for model interpretation

(b) The *Lasso* is a recent alternative to ridge regression that overcomes the disadvantage
The coefficients are estimated by minimizing the following objective function

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

Instructor: Junho Song
junhosong@snu.ac.kr

(c) The difference between Lasso and Ridge regression is penalty term
penalty of the Lasso → l_1 penalty
penalty of Ridge regression → l_2 penalty

(d) The l_1 penalty has the effect of forcing some of the coefficient estimates to be *exactly zero* when the tuning parameter λ is sufficiently large (the Lasso performs "variable selection" as well)
→ the Lasso are generally much easier to interpret than Ridge regression (the Lasso yields *sparse* models)

(e) As in Ridge regression, selecting a good value for λ is critical → ***selecting tuning parameter***

(f) One can show that the Lasso and Ridge regression coefficient estimates solve the following optimization problems (for every value of λ, there is some s to give same coefficient estimates)

Ridge regression

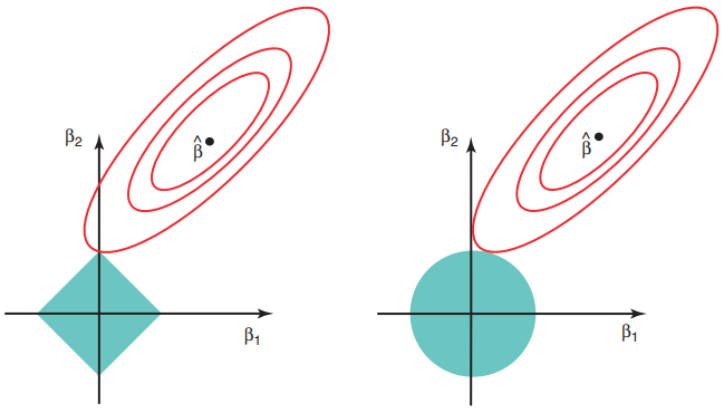
$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}\beta_j^2 \le s$$

The Lasso

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

Best Subset Selection

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \quad \text{subject to} \quad \sum_{j=1}^{p}I(\beta_j \ne 0) \le s$$



Contours of the error and constraint function for the lasso (left) and ridge regression (right)

(g) Comparing the Lasso and Ridge regression
Model Interpretation: the Lasso > Ridge regression
Prediction accuracy: the Lasso (?) Ridge regression → depends on the problems

Instructor: Junho Song
junhosong@snu.ac.kr

(h) The Lasso implicitly assumes that a number of the coefficients are truly equal to zero
Usually, if all p predictors were related to the response the Ridge regression outperforms the Lasso

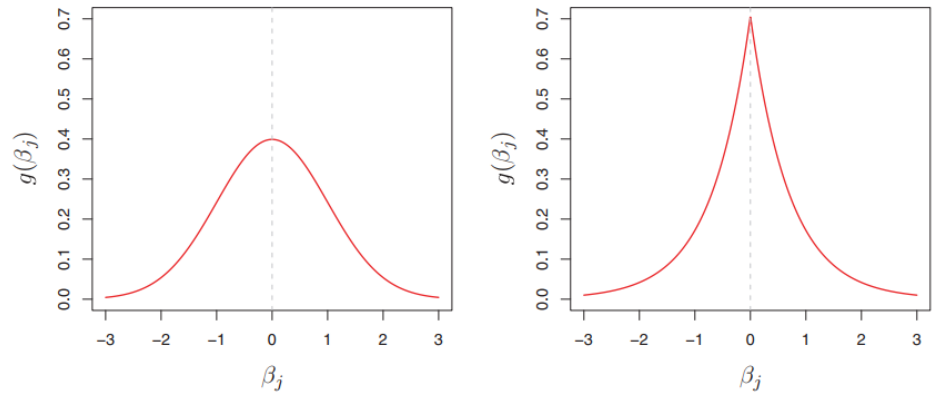
(i) Bayesian interpretation for Ridge regression and the Lasso

$$p(\beta|X,Y) \propto f(Y|X,\beta)p(\beta|X) = f(Y|X,\beta)p(\beta) \ \ (\because X \ is \ fixed)$$
$$Y = \beta_0 + X_1\beta_1 + \cdots + X_p\beta_p + \epsilon$$

Assume ϵ is independent and normally distributed, and $p(\beta) = \prod_{j=1}^{p} g(\beta_j)$

If g is a Gaussian distribution with mean of zero and standard deviation of λ,
→ the posterior mode for β = Ridge regression solution (It is also the posterior mean)

If g is a double-exponential (Laplace) distribution with mean of zero and scale parameter of λ,
→ the posterior mode for β = the Lasso solution (However, it is not the posterior mean)



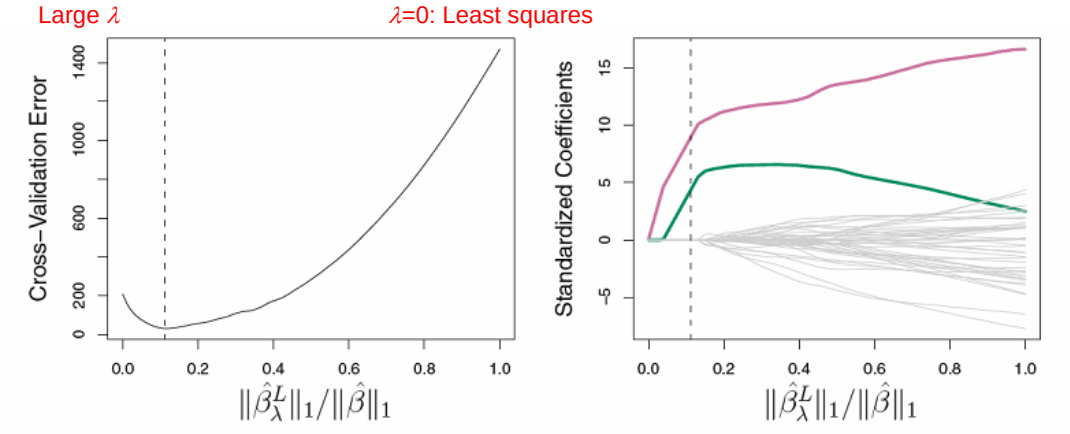
Gaussian prior (left) and double-exponential prior (Right)

3. Selecting the tuning parameter

(a) Implementing Ridge regression and the Lasso requires a method for selecting a value for the tuning parameter λ (or equivalently the value of the constraint s in another formulations for two approaches)

(b) Cross-validation provides a simple way to tackle this problem → we can choose a grid of λ values, and compute the cross-validation error for each value of λ, then select the tuning parameter value for which the cross-validation error is the smallest

Instructor: Junho Song
junhosong@snu.ac.kr



Cross validation error and standardized coefficients for the lasso

(c) The left-hand panel of the figure above displays the cross-validation error of the Lasso, while the right-hand panel displays the coefficient estimates of the Lasso

(d) The vertical dashed line indicate the point at which the cross-validation error is smallest

The two colored lines in the right-hand panel represent the two predictors related to the response, while the grey lines represent the unrelated predictors (these often are called as *signal* and *noise* variables respectively)

(e) Cross-validation together with the Lasso has correctly identified the two *signal* variables in the model

In contrast, the least squares solution assigns coefficient estimates to both signal and noise variables

Ridge regression using cross-validation (In case the Lasso, change alpha to 1)

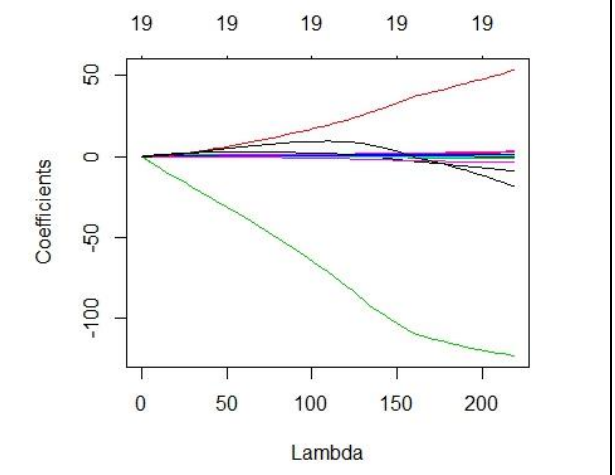
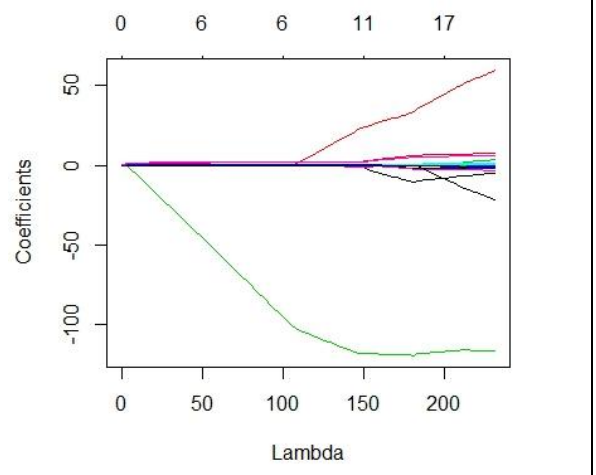
```
# Test MSE of Ridge regression models
set.seed(1)
train=sample(1:nrow(x),nrow(x)/2) # alternative way to sample
test=(-train)
y.test=y[test]

ridge.mod = glmnet(x[train,],y[train],alpha=0,lambda=grid,thres=1e-12)

#case 1: using lambda(=0), i.e. Least Squares
ridge.pred=predict(ridge.mod,s=0,newx=x[test,])
mean((ridge.pred-y.test)^2)
#case 2: using lambda(=4)
ridge.pred = predict(ridge.mod,s=4,newx=x[test,])
mean((ridge.pred-y.test)^2) # MSE for test set
#case 3: using lambda(=10^10)
ridge.pred=predict(ridge.mod,s=1e10,newx=x[test,])
mean((ridge.pred-y.test)^2)

# Tuning lambda by cross validation
set.seed(1)
cv.out = cv.glmnet(x[train,],y[train],alpha=0) # 10-fold CV(default)
plot(cv.out)
bestlam = cv.out$lambda.min
```

Instructor: Junho Song
junhosong@snu.ac.kr

```
# Test MSE
ridge.pred = predict(ridge.mod, s=bestlam, newx)
test_MSE =mean((ridge.pred-y.test)^2) # better than lambda=4 case

# Coefficients for full data
out = glmnet(x,y,alpha=0)
plot(out)
predict(out,type="coefficients",s=bestlam)[1:20,]
```

	Ridge regression	The Lasso
Selected λ	212	16.78
Test MSE	96,016	100,740

Coefficients estimates using full data for all λ

Ridge regression	The Lasso
	

Coefficients estimates using full data for selected λ

Ridge regression

(Intercept)	AtBat	Hits	HmRun	Runs	RBI
9.88487157	0.03143991	1.00882875	0.13927624	1.11320781	0.87318990
Walks	Years	CAtBat	CHits	CHmRun	CRuns
1.80410229	0.13074383	0.01113978	0.06489843	0.45158546	0.12900049
CRBI	CWalks	LeagueN	DivisionW	PutOuts	Assists
0.13737712	0.02908572	27.18227527	-91.63411282	0.19149252	0.04254536
Errors	NewLeagueN				
-1.81244470	7.21208394				

The Lasso

(Intercept)	AtBat	Hits	HmRun	Runs	RBI
19.5223995	0.0000000	1.8701714	0.0000000	0.0000000	0.0000000
Walks	Years	CAtBat	CHits	CHmRun	CRuns
2.2187934	0.0000000	0.0000000	0.0000000	0.0000000	0.2072852
CRBI	CWalks	LeagueN	DivisionW	PutOuts	Assists
0.4127984	0.0000000	1.7591970	-103.5051402	0.2206884	0.0000000
Errors	NewLeagueN				
0.0000000	0.0000000				

Seoul National University
Dept. of Civil and Environmental Engineering

Instructor: Junho Song
junhosong@snu.ac.kr

M1586.002500 Information Engineering for CE Engineers
In-Class Material: Class 16
Linear Model Selection and Regularization (ISL Chapter 6)

1. Dimension Reduction Methods

(a) All variance-control methods introduced in this chapter (linear model selection and regularization) use the original predictors, $X_1, X_2, \dots X_p$.

(b) Can we transform the original predictors to $Z_1, Z_2, \dots, Z_M$ where $M < p$, and then fit least squares using the transformed variables? → **Dimension Reduction** methods

(c) **Transformation** of predictors: linear combinations of original predictors X_j

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j, \qquad m = 1, \dots, M$$

where $M < p$. In a matrix-form, the transformation is described as $\mathbf{Z} = \mathbf{\Phi X}$

Can be considered as a special case of the original linear regression model

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i$$
$$= \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \qquad i = 1, \dots, n$$

with constraint

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

The dimension of problems is *reduced* from $p + 1$ to $M + 1$

If the constants $\phi_{1m}, \phi_{2m}, \dots, \phi_{pm}$ are chosen wisely, then such dimension reduction approaches using θ_m can outperform least squares regression using β_j because the constraint $\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm}$ introduces additional b_____ but reduces v_____.

(d) Two common steps of all dimension reduction methods

① The transformed predictors $Z_1, Z_2, \dots, Z_M$ are obtained

② The model is fitted using these $M + 1$ predictors

Instructor: Junho Song
junhosong@snu.ac.kr

2. Principal Components Analysis (PCA)

(a) A popular approach for deriving a low-dimensional set of features from a large set of original variables (a popular tool for unsupervised learning in Chapter 10)

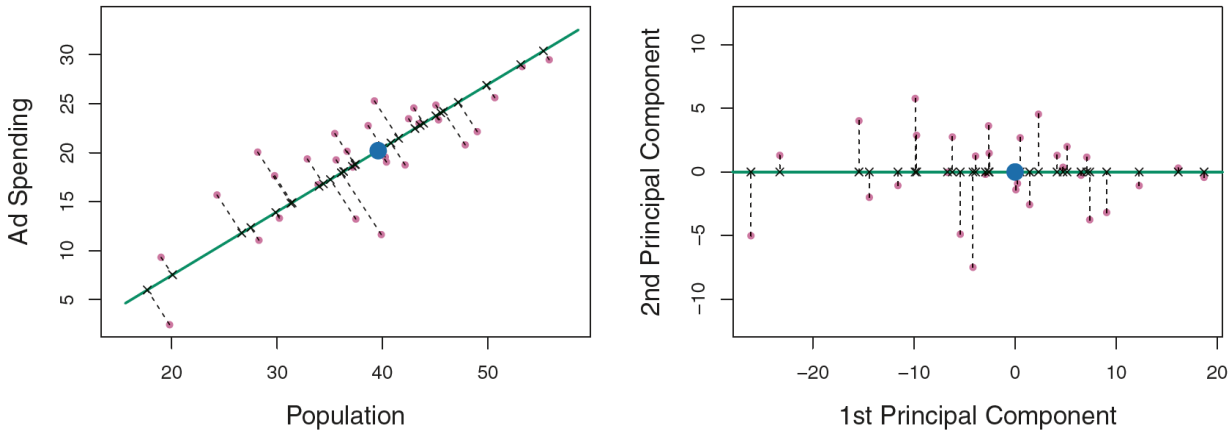
(b) **First principal component**: the direction along which the observations vary the most

$$Z_1 = \sum_{j=1}^{p} \phi_{j1}(X_j - \bar{X}_j)$$

where ϕ_{j1} is the **principal component loading**, which is defined as

$$\mathbf{\Phi}_1 = (\phi_{11},, \dots, \phi_{p1})^{\mathrm{T}} = \underset{\|\mathbf{\Phi}_1\|=1}{\operatorname{argmax}}(\mathrm{Var}[Z_1]) = \underset{\|\mathbf{\Phi}_1\|=1}{\operatorname{argmax}}\left(\mathbf{\Phi}_1^{\mathrm{T}} \Sigma_{\mathbf{XX}} \mathbf{\Phi}_1\right)$$

The first principal component vector is the line that is as close as possible to the data, i.e. minimizing the distance between the data and line



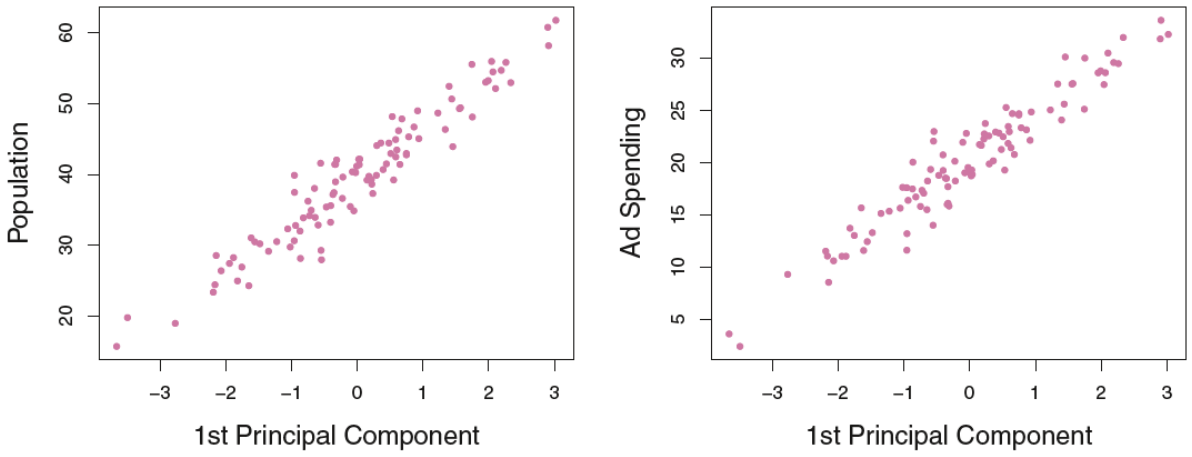
The first principal component (green) and first principal component score (dashed)

(c) Principal component score: the value of the first principal component Z_1 at each point. The principal component score at the ith point is

$$z_{i1} = \sum_{j=1}^{p} \phi_{j1}(x_{ij} - \bar{X}_j)$$

Principal component scores are single number summaries of the predictor combinations (showing strong correlation with the predictors)

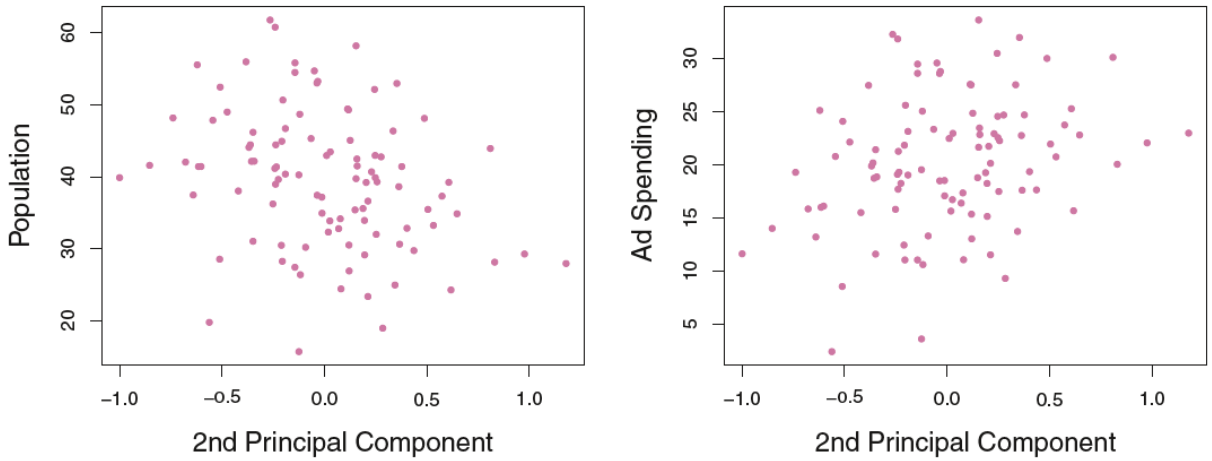
Instructor: Junho Song
junhosong@snu.ac.kr



(d) Further components $(m \leq p)$

$$\boldsymbol{\Phi}_m = \left(\phi_{1m}, , \ldots, \phi_{pm}\right)^{\mathrm{T}} = \underset{\|\boldsymbol{\Phi}_m\|=1}{\operatorname{argmax}}(\operatorname{Var}[Z_m]) = \underset{\|\boldsymbol{\Phi}_m\|=1}{\operatorname{argmax}}\left(\boldsymbol{\Phi}_m^{\mathrm{T}} \Sigma_{\hat{\boldsymbol{X}}_m \hat{\boldsymbol{X}}_m} \boldsymbol{\Phi}_m\right)$$

where $\hat{\boldsymbol{X}}_m = \boldsymbol{X}\left(\boldsymbol{I} - \sum_{k=1}^{m-1} \boldsymbol{\Phi}_k \boldsymbol{\Phi}_k^{\mathrm{T}}\right)$



(e) Each succeeding component is **orthogonal** to the preceding components

$$\boldsymbol{\Phi}_i \perp \boldsymbol{\Phi}_j \ (i \neq j)$$

(f) Standardization

In order to prevent high-variance variables from playing a larger role, the standardization is needed to ensure that all variables are on the same scale.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_{ij} - \bar{x}_j\right)^2}}$$

Instructor: Junho Song
junhosong@snu.ac.kr

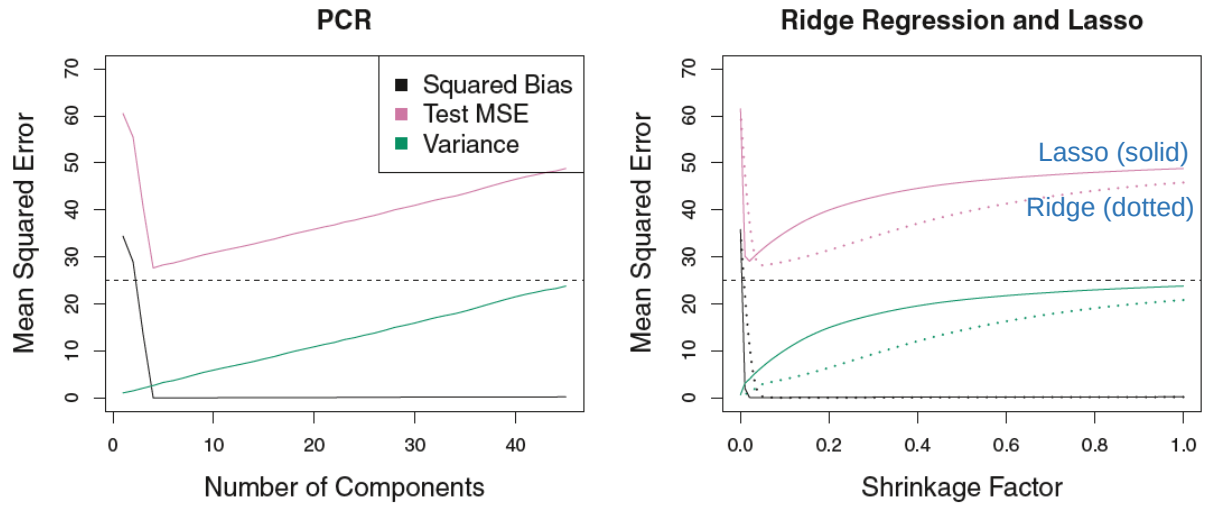
3. [Method 1] Principal Component Regression (PCR)

(a) Constructing the first M principal components $(Z_1, Z_2, \dots, Z_M)$ and then using these components as the predictors in a linear regression model

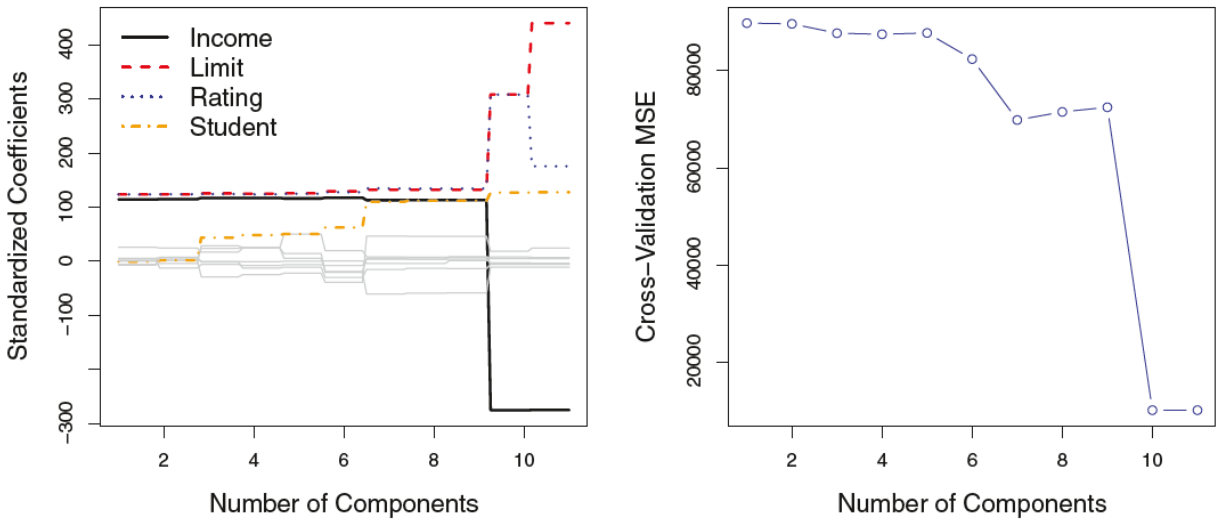
(b) Assumption: a small number of principal components suffice to explain most of the variability in the data, as well as the relationship with the response

(c) If the assumption underlying PCR holds, a least squares model with M principal components will lead to better results than a least squares model with p original predictors

(d) The optimal value of M, i.e. the optimal number of principal components used in regression can be chosen by cross-validation



Mean squared error by PCR, Ridge and Lasso for a simulated data set



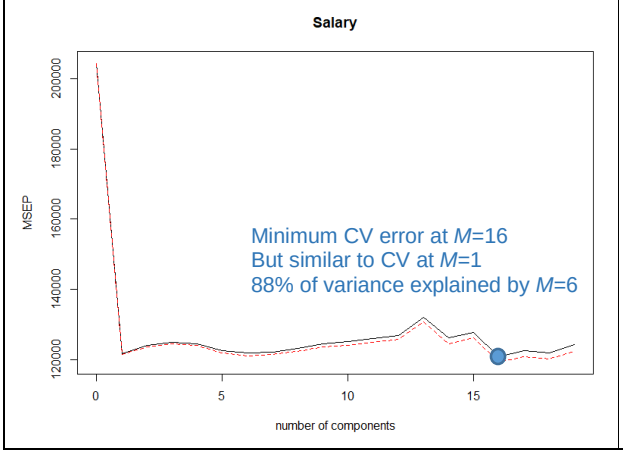
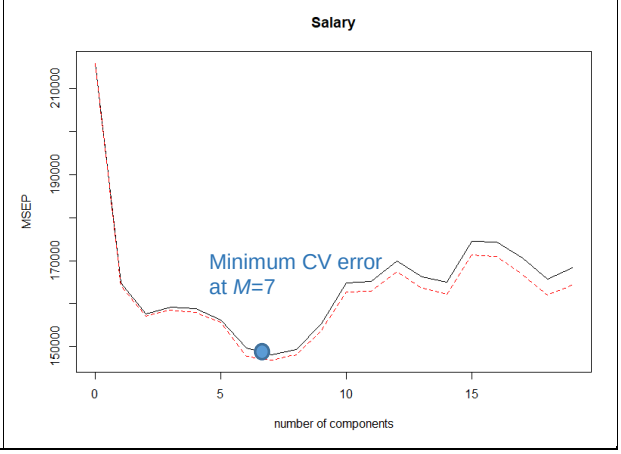
The cross-validation MSE obtained using PCR on the Credit data

(e) PCR with $M = p$ is equivalent to performing least squares

Instructor: Junho Song
junhosong@snu.ac.kr

Principal Component Regression

```
library(ISLR)
Hitters=na.omit(Hitters)

x=model.matrix(Salary~.,Hitters)[,-1]
y=Hitters$Salary
set.seed(1)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]

library(pls) # to use "pcr" command

# (1) PCR with 10-fold Cross Validation
set.seed(2)
pcr.fit = pcr(Salary~.,data=Hitters,scale=TRUE,validation="CV")
# default: 10-fold CV
summary(pcr.fit) # CV score is 'root' mean square
validationplot(pcr.fit,val.type="MSEP") # plotting the MSE

# (2) Fit PCR model with train data set
set.seed(1)
pcr.fit = pcr(Salary~.,data=Hitters,subset=train,scale=TRUE,
   validation="CV")
validationplot(pcr.fit,val.type="MSEP")
# minimum test error at M=7

pcr.pred = predict(pcr.fit,x[test,],ncomp=7)
mean((pcr.pred-y.test)^2)

# fitting PCR on the full data set (using M=7)
pcr.fit = pcr(y~x,scale=TRUE,ncomp=7)
summary(pcr.fit)
```

Cross-validation using whole set	Cross-validation using train data
	

Instructor: Junho Song
junhosong@snu.ac.kr

4. [Method 2] Partial Least Squares (PLS)

(a) PCR identified the principal directions in an unsupervised way, since the response is not used in the process

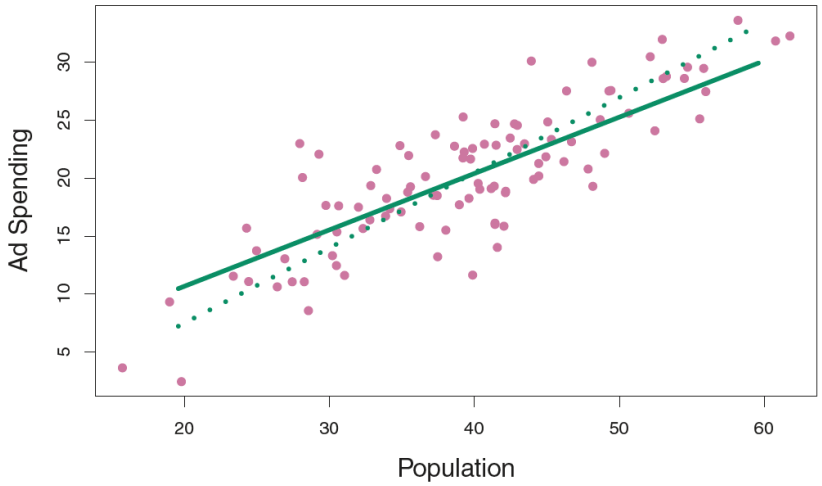
(b) Drawback of PCR: there is no guarantee that the identified best directions for the original predictors will also be the best directions to use for predicting the response

(c) Partial Least Squares (PLS): a supervised alternative to PCR

$$\underset{z_{i1}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \beta_0 - z_{i1})^2 = \underset{\phi_{j1}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2$$

(d) The first direction of PLS is same as the simple linear regression of Y onto X_j

$$Z_1 = \sum_{j=1}^{p} \phi_{j1} X_j = \sum_{j=1}^{p} \beta_j X_j$$



The first PLS direction (solid) and first PCR direction (dotted) for the Advertising data

(e) The way to identify other directions of PLS

① Each of the variables for Z_1 can be adjusted by regressing each variable on Z_1 and taking residuals

② Z_2 can be computed by using this orthogonalized data in exactly the same fashion as Z_1

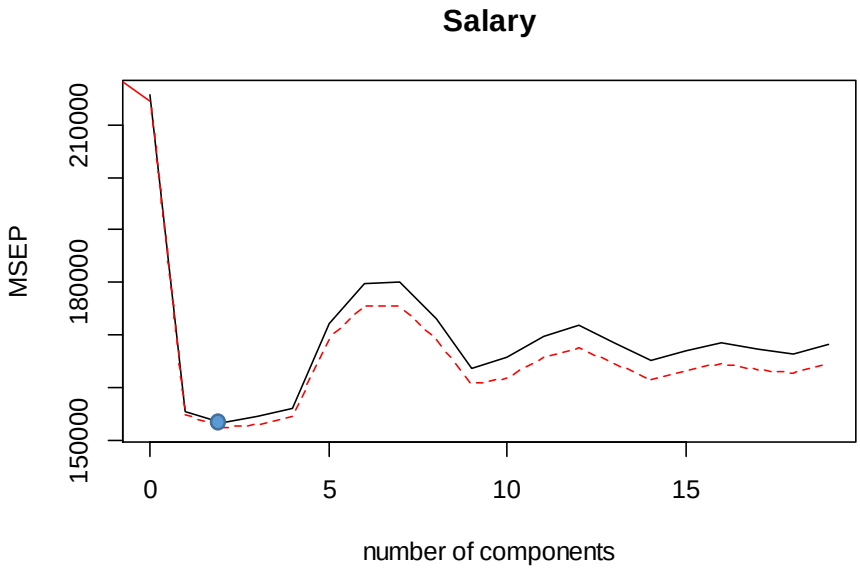
③ Repeat the iterative approach M times

(f) As with PCR, the number of M of PLS directions is a tuning parameter that is typically chosen by cross-validation

Instructor: Junho Song
junhosong@snu.ac.kr

Partial Least Squares

```
# (3) PLS regression (with train set)
set.seed(1)
pls.fit = plsr(Salary~.,data=Hitters,subset=train,scale=TRUE,
   validation="CV")
summary(pls.fit)
validationplot(pls.fit,val.type="MSEP")

# evaluating the test set MSE using M with the lowest cross-validation
   error
pls.pred = predict(pls.fit,x[test,],ncomp=2)
mean((pls.pred-y.test)^2)

# performing PLS using the full data set using M
pls.fit=plsr(Salary~.,data=Hitters,scale=TRUE,ncomp=2)
summary(pls.fit)
```



	Principal Component Regression	Partial Least Squares
Number of Components	7	2
Test MSE	96,556.22	101,417.5
Percentage of Variance of response explained by principal components	46.69%	46.40%