Homework 4

Steven Xu

Due @ 5pm on November 9, 2018

Part 1. Let $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{w}$, where $\mathbf{y} \in \mathbb{R}^n, \mathbf{X} \in \mathbb{R}^{n\times p}$, $\boldsymbol{\beta} \in \mathbb{R}^p$, and w_i are i.i.d. random vectors with zero mean and variance σ^2. Recall that the ridge regression estimate is given by

$$\hat{\boldsymbol{\beta}}_\lambda = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2.$$

1. Show that the variance of $\hat{\boldsymbol{\beta}}_\lambda$ is given by

$$\sigma^2 \mathbf{W}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{W},$$

where $\mathbf{W} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}$.

Answer:

First we need to show that $(X^\mathsf{T}X + \lambda\mathbf{I}) \succ 0$

Let $\mathbf{z} \neq \mathbf{0}, \mathbf{z}(^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X})\mathbf{z} = (\mathbf{X}\mathbf{z})^\mathsf{T}(\mathbf{X}\mathbf{z}) = ||\mathbf{X}\mathbf{z}||_2^2 \geq 0$

Thus $\mathbf{X}^\mathsf{T}\mathbf{X}$ is positive semi-definite.

Then immediately we have $\mathbf{z}^\mathsf{T}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})\mathbf{z} > 0 \ \ \forall \mathbf{z} \neq \mathbf{0}$

Therefore $(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}) \succ 0$

$$f(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\mathsf{T}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \frac{\lambda}{2}\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\beta}$$

$$f(\boldsymbol{\beta}) = \frac{1}{2}(\mathbf{y}^\mathsf{T}\mathbf{y} - 2\mathbf{y}^\mathsf{T}\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\beta}) + \frac{\lambda}{2}\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\beta}$$

$$\frac{\partial f}{\partial \boldsymbol{\beta}} = \frac{1}{2}(-2\mathbf{X}^\mathsf{T}\mathbf{y} + 2\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta} = 0$$

$$(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} = \mathbf{X}^\mathsf{T}\mathbf{y}$$

$$\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$$

$$\frac{\partial^2 f}{\partial \boldsymbol{\beta}^2} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}) \succ 0$$

$$Var(\hat{\boldsymbol{\beta}}_\lambda) = \mathbf{W}\mathbf{X}^\mathsf{T} var(\mathbf{y})\mathbf{X}\mathbf{W}^\mathsf{T}$$

Since M is symmetric and $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \ \boldsymbol{\sigma}^\mathbf{2}\mathbf{I})$

$$Var(\hat{\boldsymbol{\beta}}_\lambda) = \sigma^2\mathbf{W}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{W}$$

Where $\mathbf{W} = (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}$

2. Show that the bias of $\hat{\boldsymbol{\beta}}_\lambda$ is given by

$$-\lambda \mathbf{W}\boldsymbol{\beta}$$

Answer:

$$\begin{aligned}
Bias(\hat{\boldsymbol{\beta}}_\lambda) &= \mathbb{E}(\hat{\boldsymbol{\beta}}_\lambda) - \boldsymbol{\beta} \\
&= (\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta} \\
&= [(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X} - \mathbf{I}]\boldsymbol{\beta} \\
&= [(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I} - \lambda\mathbf{I}) - \mathbf{I}]\boldsymbol{\beta} \\
&= [\mathbf{I} - \lambda(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1} - \mathbf{I}]\boldsymbol{\beta} \\
&= -\lambda(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\boldsymbol{\beta} \\
&= -\lambda\mathbf{W}\boldsymbol{\beta}
\end{aligned}$$

3. A natural question is how to choose the tuning parameter λ. Several classes of solutions See Efron paper.

The degrees of freedom of a linear estimator $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ is given by $\text{tr}(\mathbf{S})$. Ridge regression provides a linear estimator of the observed response $\mathbf{y}$ where $\mathbf{S} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}$. Show that the degrees of freedom of the ridge estimator is given by

$$\sum_i \frac{\sigma_i^2}{\sigma_i^2 + \lambda},$$

where σ_i is the ith singular value of $\mathbf{X}$.

Answer:

$$
\begin{aligned}
tr(S) &= tr\left\{\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\right\} \\
&= tr\left\{(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X}\right\} \\
&= tr\left\{(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I}-\lambda\mathbf{I})\right\} \\
&= tr\left\{\mathbf{I}-\lambda(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}\right\} \\
&= \sum_{i=1}^{n} 1-\lambda\times tr\left\{(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}\right\} \\
&= \sum_{i=1}^{n} 1-\lambda\times tr\left\{(\mathbf{V}\mathbf{\Sigma}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}+\lambda\mathbf{V}\mathbf{V}^\mathsf{T})^{-1}\right\} \\
&= \sum_{i=1}^{n} 1-\lambda\times tr\left\{(\mathbf{V}\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}+\lambda\mathbf{V}\mathbf{V}^\mathsf{T})^{-1}\right\} \\
&= \sum_{i=1}^{n} 1-\lambda\times tr\left\{[\mathbf{V}(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}+\lambda\mathbf{I})\mathbf{V}^\mathsf{T}]^{-1}\right\} \\
&= \sum_{i=1}^{n} 1-\lambda\times tr\left\{[\mathbf{V}^\mathsf{T}(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}+\lambda\mathbf{I})^{-1}\mathbf{V}]\right\} \\
&= \sum_{i=1}^{n} 1-\lambda\times tr\left\{(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}+\lambda\mathbf{I})^{-1}\mathbf{V}\mathbf{V}^\mathsf{T}\right\} \\
&= \sum_{i=1}^{n} 1-\lambda\times tr\left\{(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}+\lambda\mathbf{I})^{-1}\right\}
\end{aligned}
$$

We know that $\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}+\lambda\mathbf{I}$ is diagonal with i^{th} diagonal entry equals to $\sigma_i^2+\lambda$.

$$
\begin{aligned}
\therefore tr\left\{(\mathbf{\Sigma}^\mathsf{T}\mathbf{\Sigma}+\lambda\mathbf{I})^{-1}\right\} &= \sum_{i=1}^{n}\frac{1}{\sigma_i^2+\lambda} \\
\therefore tr(S) &= \sum_{i=1}^{n}(1-\frac{\lambda}{\sigma_i^2+\lambda}) = \sum_{i=1}^{n}\frac{\sigma_i^2}{\sigma_i^2+\lambda}
\end{aligned}
$$

Part 2. Ridge Regression.

You will next add an implementation of the ridge regression to your R package.

Please complete the following steps.

Step 0: Make a file called `ridge.R` in your R package. Put it in the R subdirectory, namely we should be able to see the file at github.ncsu.edu/unityidST758/unityidST758/R/ridge.R

Step 1: Write a function `ridge_regression` that computes the ridge regression coefficient estimates for a sequence of regularization parameter values λ.

It should return an error message

- if the response variable $\mathbf{y} \in \mathbb{R}^n$ and the design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ are not conformable
- if the tuning parameters are negative

Please use the `stop` function.

```
#' Ridge Regression
#'
#' \code{ridge_regression} returns the ridge regression coefficient estimates
#' for a sequence of regularization parameter values.
#'
#' @param y response variables
#' @param X design matrix
#' @param lambda vector of tuning parameters
#' @export
# ridge_regression <- function(y, X, lambda) {
#
# }
```

Step 3: Write a unit test function `test-ridge` that

- checks the error messages for your `ridge_regression` function
- checks the correctness of the estimated regression coefficients produced by `ridge_regression` function. Given data $(\mathbf{y}, \mathbf{X})$, recall that $\mathbf{b}$ is the ridge estimate with regularization parameter λ if and only if

$$(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda \mathbf{I})\mathbf{b} = \mathbf{X}^\mathsf{T}\mathbf{y}.$$

Step 4: Construct three poorly conditioned multiple linear regression problems, with design matrices with condition numbers of 100, 1000, and 10000. Write in this Markdown file, using nice notation the problem set up.

For an arbitrary full column rank design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$ be the singular value decomposition of $\mathbf{X}$. Now since the definition of condition number is $\kappa(\mathbf{X}) = \frac{\sigma_{max}(X)}{\sigma_{min}(X)}$, where $\sigma_{max}(X)$, $\sigma_{min}(X)$ are the largest and smallest non-zero sigular value respectively. Then we can modify $\mathbf{\Sigma}$ such that $\frac{\sigma^*_{max}(X)}{\sigma^*_{min}(X)} = K$, where $\sigma^*_{max}(X)$ and $\sigma^*_{min}(X)$ are modified singular values and K our desired condition number. Then the reconstructed $\mathbf{X}$ would be a ill-conditioned design matrix.

$$\mathbf{X} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1.001 \end{pmatrix}, \ \mathbf{y} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

We can see that X is extremely close to singular.

$$\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T} = \mathbf{X}$$

$$\mathbf{\Sigma} = \begin{pmatrix} 2.45 & 0 \\ 0 & 0.00058 \\ 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0.058 & 0 \\ 0 & 0.00058 \\ 0 & 0 \end{pmatrix} \; or \; \begin{pmatrix} 0.58 & 0 \\ 0 & 0.00058 \\ 0 & 0 \end{pmatrix} \; or \; \begin{pmatrix} 5.8 & 0 \\ 0 & 0.00058 \\ 0 & 0 \end{pmatrix} = \mathbf{\Sigma}^*$$

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}^*\mathbf{V}^\mathsf{T}$$

Step 5: Solve the three regression problems you constructed in Step 4.

```
set.seed(1234)
X <- matrix(c(1,1,1,1,1,1.001),nrow=3,ncol=2, byrow = F)
y = matrix(c(1,2,1),ncol=1)
s = svd(X)
d = s$d
u = s$u
v = s$v
n = length(d)
d[1] <- d[2]*1e2
b_hat = matrix(0,ncol=3,nrow=n)
X_1e2 = u%*%diag(d)%*%t(v)
d[1] <- d[2]*1e3
X_1e3 = u%*%diag(d)%*%t(v)
d[1] <- d[2]*1e4
X_1e4 = u%*%diag(d)%*%t(v)
b_hat[,1] = solve(t(X_1e2)%*%X_1e2,t(X_1e2)%*%y)
b_hat[,2] = solve(t(X_1e3)%*%X_1e3,t(X_1e3)%*%y)
b_hat[,3] = solve(t(X_1e4)%*%X_1e4,t(X_1e4)%*%y)
```

Step 6: Solve a perturbed linear regression problem, i.e. add noise to the design matrix and response variable. Solve the perturbed systems and report the relative error between the solutions to the perturbed systems and the solutin you obtained in Step 5. How does this relative error compare to the worst case bounds derived in class?

$$\Delta\mathbf{X} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 2 \times 10^{-6} & 2 \times 10^{-6} \end{pmatrix}, \; \Delta\mathbf{Y} \sim N_3(\mathbf{0}, 10^{-6} \times \mathbf{I})$$

Since $\kappa(X) = 10^2, 10^3, 10^4$, I chose $\epsilon = 5 \times 10^{-5}$ so that $\epsilon\kappa(X) = \frac{1}{2}$. Then if all the conditions are met, we should have the inequality $\frac{||\tilde{\beta}-\hat{\beta}||_2}{||\hat{\beta}||_2} \leq 4\epsilon\kappa(X)$

```
X_noise = matrix(c(0,0,2e-6,0,0,-2e-6),ncol=2)
y_noise = rnorm(3,0,1e-6)
X_pert_1e2 = X_1e2+X_noise
X_pert_1e3 = X_1e3+X_noise
X_pert_1e4 = X_1e4+X_noise
y_pert = y+y_noise
cat("Size of perturbation")
```

```
## Size of perturbation
```

```
c(max(norm(X_noise,'2')/norm(X_1e2,'2'),norm(y_noise,'2')/norm(y,'2')),max(norm(X_noise,'2')/norm(X_1e3
```

```
## [1] 4.899796e-05 4.899796e-06 6.720599e-07
```

```
b_hat_pert = matrix(0,ncol=3,nrow=n)
b_hat_pert[,1] = solve(t(X_pert_1e2)%*%X_pert_1e2,t(X_pert_1e2)%*%y_pert)
b_hat_pert[,2] = solve(t(X_pert_1e3)%*%X_pert_1e3,t(X_pert_1e3)%*%y_pert)
b_hat_pert[,3] = solve(t(X_pert_1e4)%*%X_pert_1e4,t(X_pert_1e4)%*%y_pert)
re_err1 = numeric(3)
for(i in 1:3){
  re_err1[i] = norm(matrix((b_hat_pert[,i]-b_hat[,i]),ncol=1),'2')/norm(matrix(b_hat[,i],ncol=1),'2')
}
cat("relative error")
```

```
## relative error
```

```
re_err1
```

```
## [1] 0.004006632 0.004012891 0.004012955
```

The maximum perturbation for the three ill-conditioned matrices are $4.9 \times 10^{-5}, 4.9 \times 10^{-6}, 1.16 \times 10^{-6}$ which are all less than 5×10^{-5}. Therefore the conditions are met. Relative erros are all about 0.004, comparing to the worst bounds $0.02, 0.2, 2$. We see that for Matrix with smaller condition number the bound is tighter.

Step 7: Write a function `leave_one_out` that computes the following leave-one-out (LOO) prediction error estimate:

$$\text{LOO}(\lambda) = \frac{1}{n} \sum_{k=1}^{n} (y_k - \hat{y}_k^{-k}(\lambda))^2,$$

where

$$y_k - \hat{y}_k^{-k}(\lambda) = \frac{y_k - \hat{y}_k(\lambda)}{1 - h_k(\lambda)},$$

and $h_k(\lambda)$ is the kth diagonal entry of the matrix $\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}^\mathsf{T}$.

```
#' Leave One Out
#'
#' \code{leave_one_out} returns the leave-one-out
#' for a sequence of regularization parameter values.
#'
#' @param y response variables
#' @param X design matrix
#' @param lambda vector of tuning parameters
#' @export
# leave_one_out <- function(y, X, lambda) {
# }
```

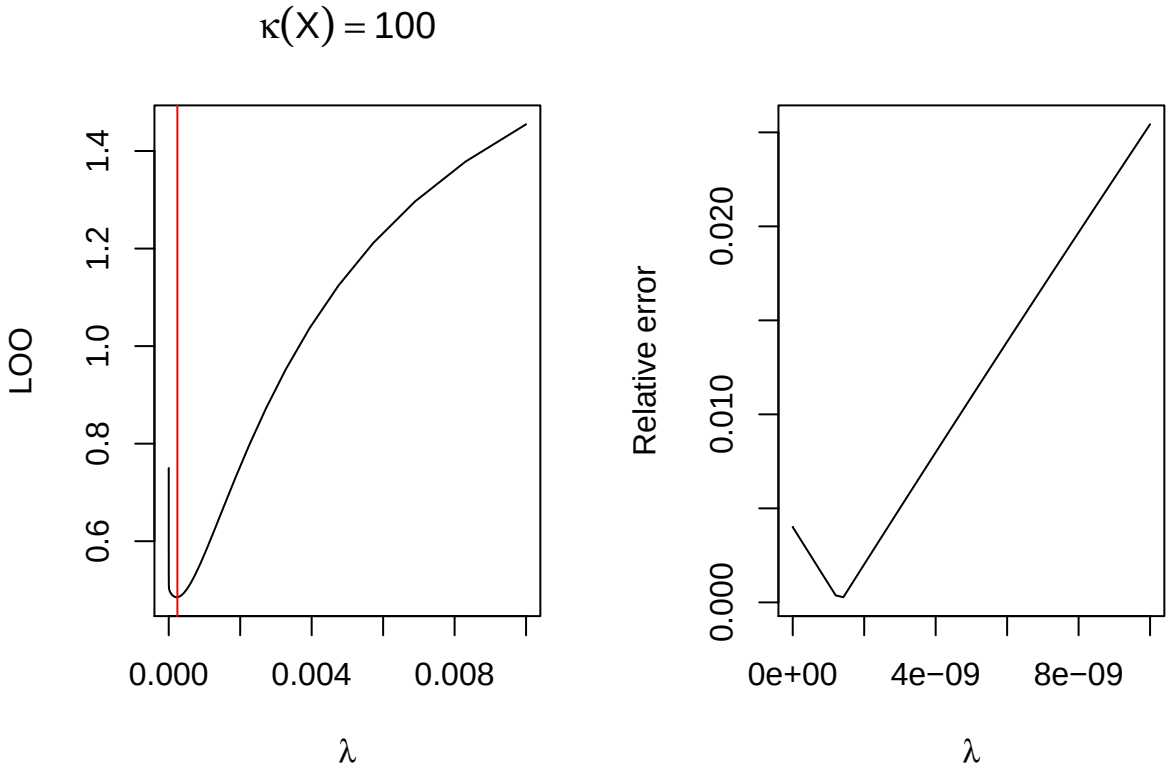
Step 8: Solve ridge penalized versions of the perturbed multiple linear regression problems for several values of the tuning parameter λ. Please highlight the one that minimizes the LOO prediction error. Plot the relative error for the three problems as a function of λ.

```
library(sgxuST758)
```

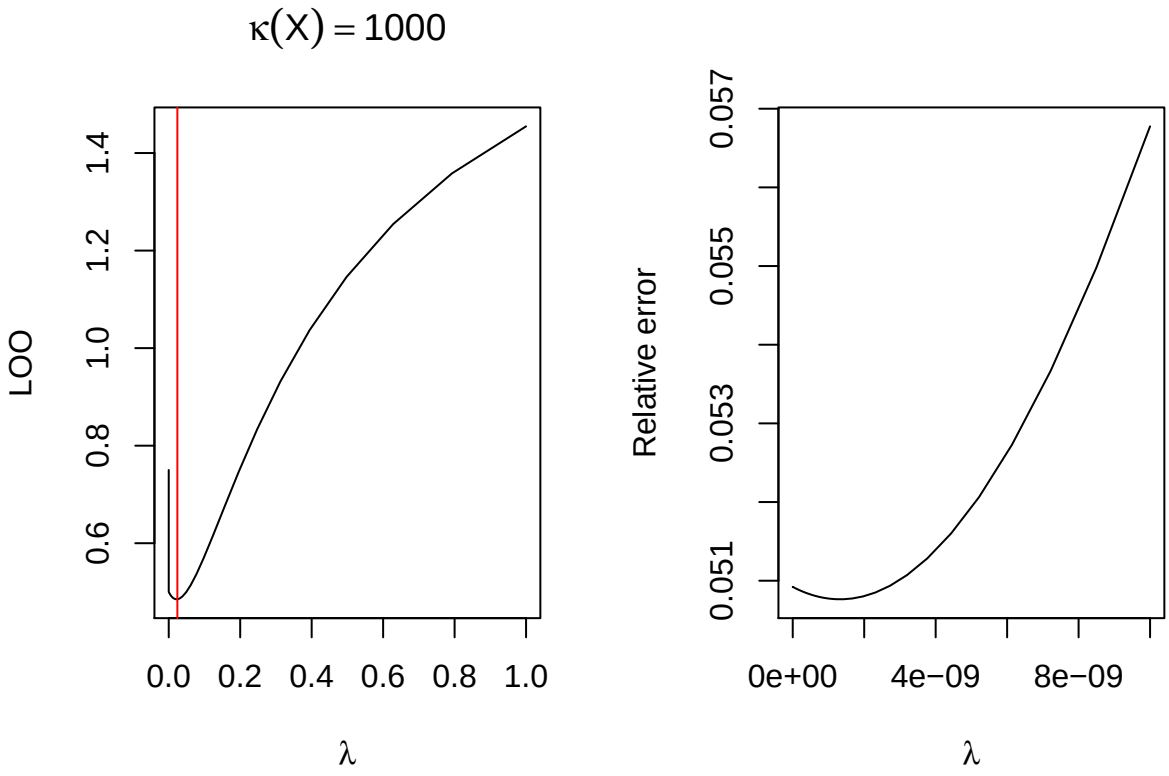
```
##
## Attaching package: 'sgxuST758'
```

```
## The following object is masked from 'package:base':
##
##     sweep
```

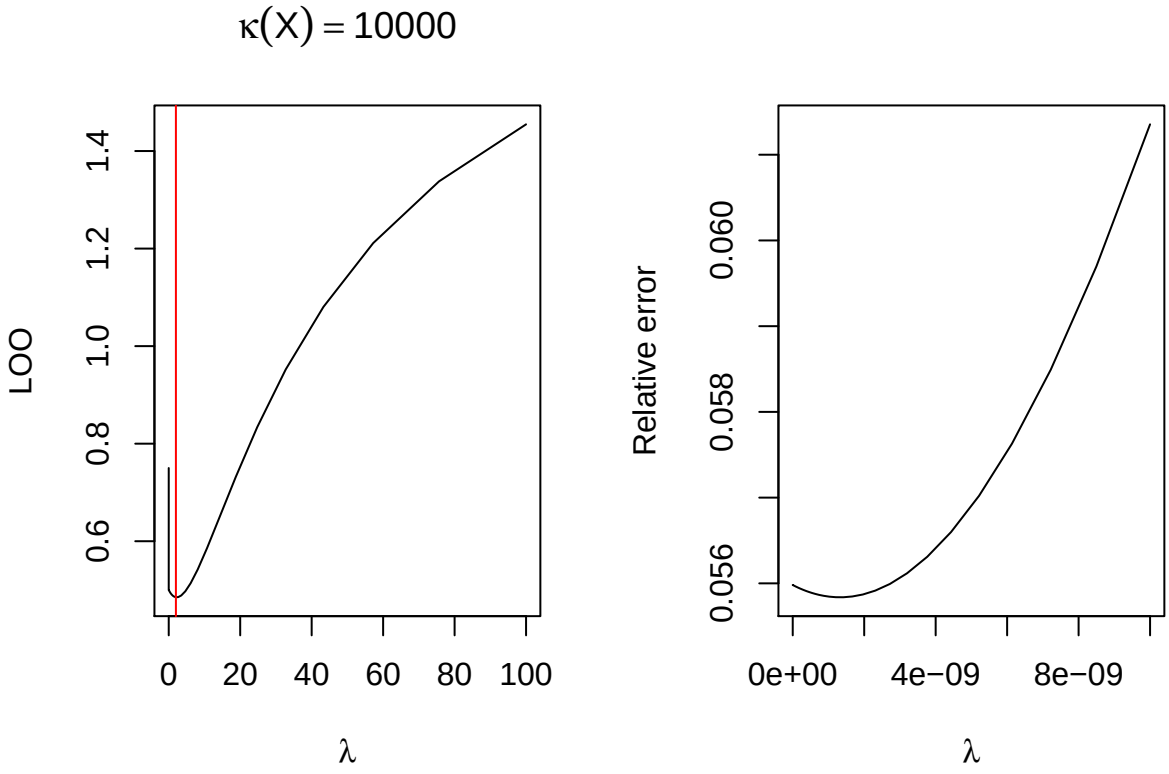
```r
par(mfrow=c(1,2))
lambda<-10^seq(-10,-2,length.out = 100)
lambda1<-10^seq(-15,-8,length.out = 100)

re_err = numeric(length(lambda))
b_lam = ridge_regression(y_pert,X_pert_1e2,lambda1)
for(i in 1:length(lambda1)){
  re_err[i] = norm(matrix((b_lam[,i]-b_hat[,1]),ncol=1),'2')/norm(matrix(b_hat[,1],ncol=1),'2')
}
loo_vec<-leave_one_out(y_pert,X_pert_1e2,lambda)
plot(lambda,loo_vec,type="l",xlab=expression(lambda),ylab="LOO",main=expression(kappa(X)==100))
l_min = which(loo_vec == min(loo_vec))
abline(v=lambda[l_min],col=2)
plot(lambda1,re_err,type="l",xlab=expression(lambda),ylab="Relative error")
```




```r
b_lam = ridge_regression(y_pert,X_pert_1e3,lambda1)
lambda<-10^seq(-10,0,length.out = 100)
for(i in 1:length(lambda1)){
  re_err[i] = norm(matrix((b_lam[,i]-b_hat[,1]),ncol=1),'2')/norm(matrix(b_hat[,1],ncol=1),'2')
}
loo_vec<-leave_one_out(y_pert,X_pert_1e3,lambda)
plot(lambda,loo_vec,type="l",xlab=expression(lambda),ylab="LOO",main=expression(kappa(X)==1000))
l_min = which(loo_vec == min(loo_vec))
abline(v=lambda[l_min],col=2)
plot(lambda1,re_err,type="l",xlab=expression(lambda),ylab="Relative error")
```




```
b_lam = ridge_regression(y_pert,X_pert_1e4,lambda1)
lambda<-10^seq(-10,2,length.out = 100)
for(i in 1:length(lambda1)){
  re_err[i] = norm(matrix((b_lam[,i]-b_hat[,1]),ncol=1),'2')/norm(matrix(b_hat[,1],ncol=1),'2')
}
loo_vec<-leave_one_out(y_pert,X_pert_1e4,lambda)
plot(lambda,loo_vec,type="l",xlab=expression(lambda),ylab="LOO",main=expression(kappa(X)==10000))
l_min = which(loo_vec == min(loo_vec))
abline(v=lambda[l_min],col=2)
plot(lambda1,re_err,type="l",xlab=expression(lambda),ylab="Relative error")
```



We can see that for larger condition number, larger value of λ is required to reach the minimizing point for LOO.