Problem 1 Penalized likelihood and soft thresholding

(A) --Prove that the quadratic term in the objective $S_\lambda(y) = \arg\min_\theta \frac{1}{2}(y-\theta)^2 + \lambda|\theta|$ is the negative likelihood of a Gaussian distribution with mean *θ* and variance 1.

Since the Gaussian (*θ*, 1) distribution had the following probability density function:

$$f(y\,|\,\theta,1) = (2\pi\sigma^2)^{-1/2}\exp[-\frac{(x-\theta)^2}{2\sigma^2}] = (2\pi)^{-1/2}\exp[-\frac{(x-\theta)^2}{2}]$$

The likelihood function is $L(\theta\,|\,y,1) = (2\pi)^{-1/2}\exp[-\frac{(y-\theta)^2}{2}]$ (for a single y)

Then the log likelihood function is $l(\theta\,|\,y,1) = \log[L(\theta\,|\,y,1)] = -\frac{1}{2}\log(2\pi) - \frac{1}{2}(y-\theta)^2$, and the first term is not dependent on *θ*, so we drop it.

Therefore we have the negative log likelihood function $\frac{1}{2}(y-\theta)^2$ of the Gaussian (*θ*, 1), which is the quadratic term of the objective minimization framework $S_\lambda(y)$.

--Prove that $S_\lambda(y) = sign(y)\cdot(|y|-\lambda)_+$, where $a_+ = \max(a,0)$ is the positive part of *a*.

Take derivative of $S_\lambda(y)$, we have

$$\frac{\partial S_\lambda(y)}{\partial\theta} = \frac{\partial}{\partial\theta}[\frac{1}{2}(y-\theta)^2 + \lambda|\theta|] = -(y-\theta) + \lambda\frac{|\theta|}{\theta} = -(y-\theta) + \lambda\cdot sign(\theta) \qquad (*)$$

We will separate this problem into three parts: (1) *θ* > 0; (2) *θ* < 0; (3) *θ* = 0.

(1) If *θ* > 0, set function (*) equal to zero, which can be written as $-(y-\theta)+\lambda = 0$

$$\theta = y - \lambda > 0 \Rightarrow y > \lambda$$

(2) If *θ* < 0, set function (*) equal to zero, which can be written as $-(y-\theta)-\lambda = 0$

$$\Rightarrow \theta = y + \lambda < 0 \Rightarrow y < -\lambda$$

(3) If *θ* = 0, we need to employ the subdifferential or subgradient:

A vector *g* is a subgradient of a convex function *f* at $x \in dom\ f$ if

$$f(y) \geq f(x) + g^T(y-x) \quad \forall y \in dom\ f$$

The subdifferential $\partial f(x)$ (always a closed convex set) of *f* at *x* is the set of all subgradients:

$$\partial f(x) = \{g\,|\,g^T(y-x) \leq f(y) - f(x), \forall y \in dom\ f\}$$

Based on the above, we have $|\theta| \geq |0| + \tau\cdot(\theta - 0) \Rightarrow |\theta| \geq \tau\theta \Rightarrow -1 \leq \tau \leq 1$

So for *θ* = 0, $\frac{\partial S_\lambda(y)}{\partial\theta} = -y + \lambda\tau \overset{set}{=} 0 \Rightarrow \tau = y/\lambda \in [-1,1]$

$\Rightarrow y \in [-\lambda, \lambda]$, which means that $\theta = 0$ is optimal for $y \in [-\lambda, \lambda]$.

Therefore, $$S_\lambda(y) = \arg\min_\theta \frac{1}{2}(y-\theta)^2 + \lambda|\theta| = \begin{cases} y-\lambda, & y>\lambda \\ y+\lambda, & y<-\lambda \\ 0, & -\lambda \le y \le \lambda \end{cases}$$

$$= sign(y)\cdot(|y|-\lambda)_+$$

--Compare soft-thresholding with hard-thresholding:

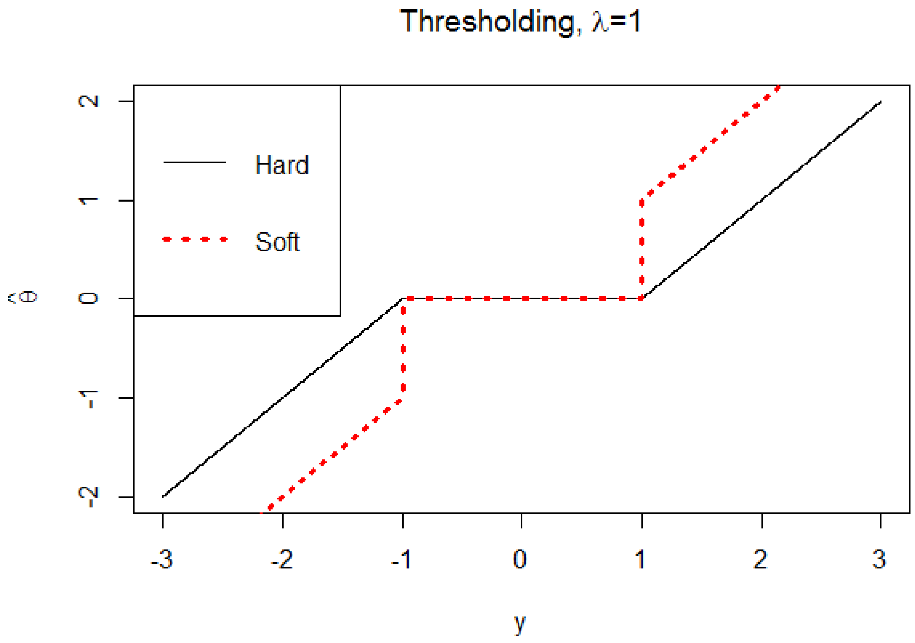

Figure 1: Comparison of soft thresholding and hard thresholding

(B) Toy examples:

i) Plot $\hat{\theta}(y_i)$ versus θ_i across a discrete grid of different λ values (0, 1, 2, 3, 4, 5) and use 80% sparsity across n = 100 data points, where the remaining points are given θ = 1, …, 20. We observe that soft-thresholding function both selects certain $\theta_i^{'}s$ by sparsifying the estimate, as well as shrinks the nonzero estimates towards 0 (Figure 2).

ii) Plot the mean-squared error of the estimate as a function of λ:

$$MSE(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[\hat{\theta}(y_i) - \theta_i]^2$$

Since each curve comes from a different randomly chosen vector θ and generated data, I've scaled each MSE to minimize to 1. The plot in Figure 3 shows how MSE changes with λ for different levels of sparsity in θ. Clearly, the λ that minimizes MSE increase as θ becomes more sparse. The bigger λ, the more aggressive the shrinkage affect.

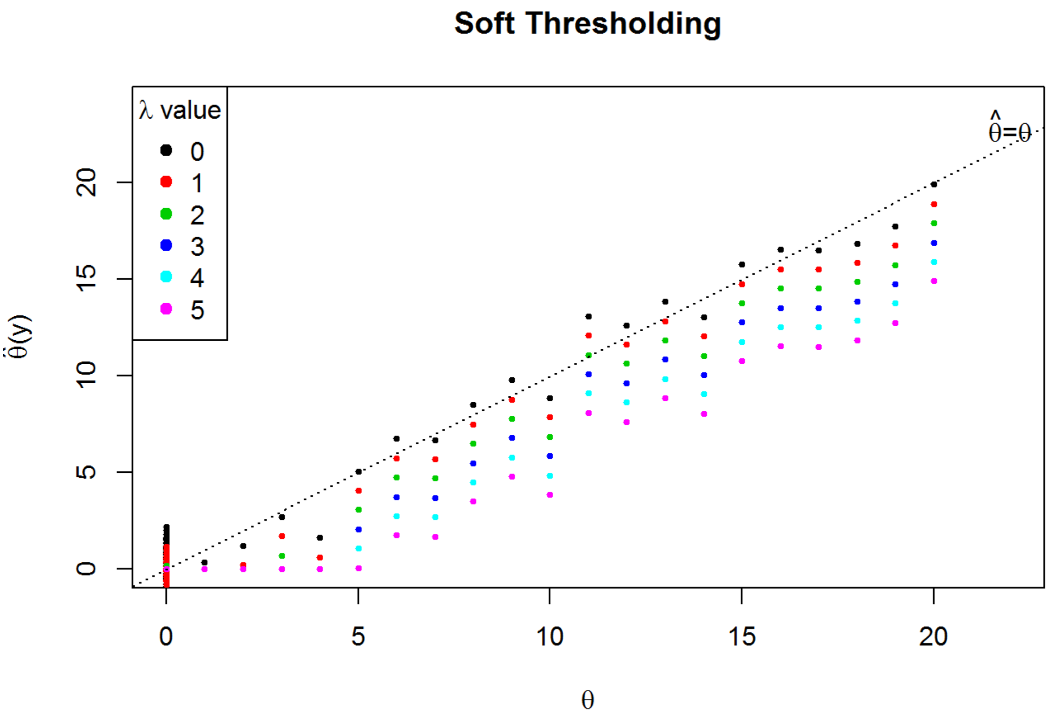

Figure 2: soft-thresholding function selects certain $\theta_i^{'}s$ by sparsifying the estimate and shrinks the nonzero estimates towards 0

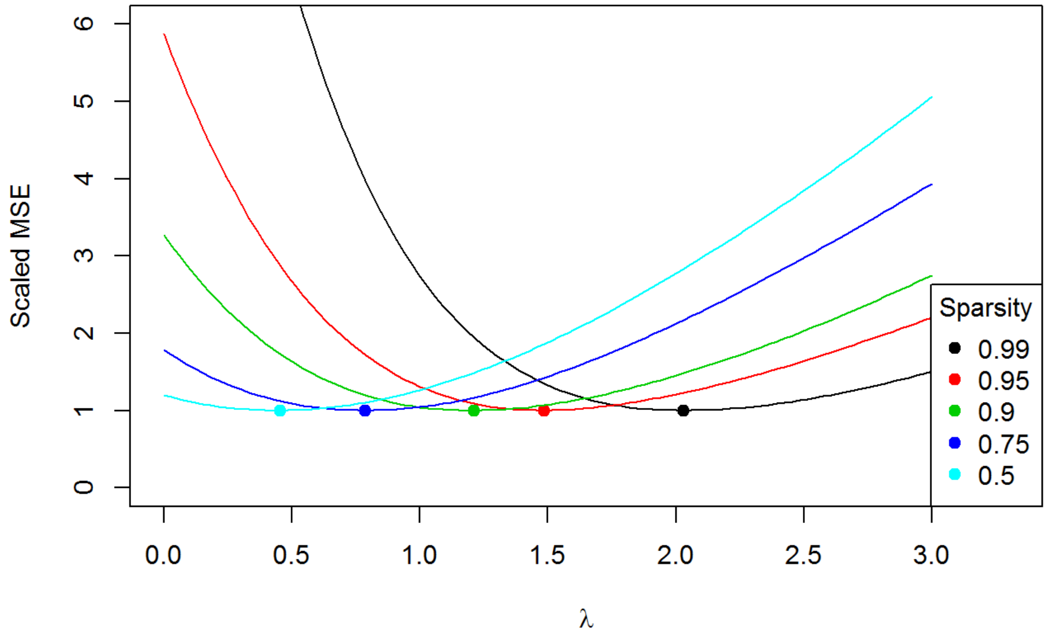

Figure 3: MSE changes with λ for different levels of sparsity in θ

Problem 2 The Lasso

Consider the standard linear regression model

$$y = X\beta + e$$

where y is an n-vector of responses, X is an $n \times p$ features matrix whose ith row x_i is the vector of features for observation i, and e is a vector of errors/residuals.

The Lasso involves estimating β as the solution to the penalized least squares problem

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2} \| y - X\beta \|_2^2 + \lambda \| \beta \|_1$$

where $\| \beta \|_1$ is the L_1 norm of coefficient vector: $\| \beta \|_1 = \sum_{j=1}^{p} | \beta_j |$. When λ is very small, the Lasso solution should be very close to the OLS solution, and all of the coefficients are in the model. As λ grows, the regularization term has greater effect and we will see fewer variables in your model (because more and more coefficients will be zero valued).

Leave the intercept in a Lasso fit unpenalized and write the objective as

$$\frac{1}{2n} \| y - (\alpha 1 + X\beta) \|_2^2 + \lambda \| \beta \|_1$$

where α is a scalar and 1is a vector of all 1's.

(A) Fit the Lasso model across a range of λ values and plot the solution path $\hat{\beta}_\lambda$ as a function of λ.

Also, track the in-sample mean-squared prediction error of the fit across the solution path.

$$MSE(\hat{\beta}_\lambda) = \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T \hat{\beta}_\lambda)^2 = \frac{1}{n} \| y - X\hat{\beta}_\lambda \|_2^2$$

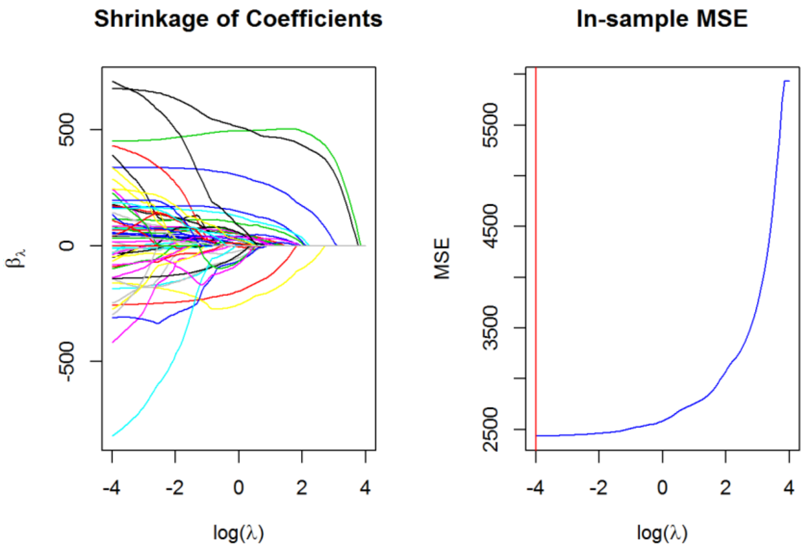

Figure 4: log (λ) v.s. $\hat{\beta}_\lambda$ and in-sample mean-squared prediction errors (MSE)

The in-sample MSE decreases as λ goes to zero. In other words, in sample fit does not want shrinkage, it wants all 64 coefficients.

(B) Randomly split the datasets into two equal parts (training and test datasets) and use 10-fold cross-validation across training data to find the best λ. ***glmnet*** packages have a ***cv.glmnet*** to choose best tuning parameters (in a chosen sequence of λ) which gives the minimum MSE. From the R output, the best $\lambda = 0.935$ with 35 parameters in the models (as shown in Figure 5). To avoid simply choosing part of λ (90 out of 100) in ***cv.glmnet***, code up the 10-folds cross-validation across all λ's. The resulting best $\lambda = 0.960$ with 34 parameters in the models and other parameters shrink to zero.

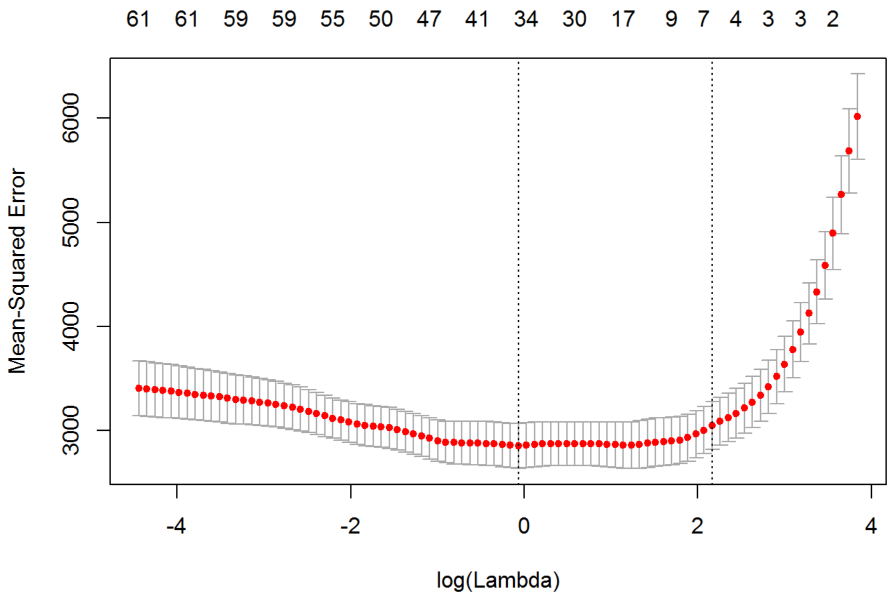

Figure 5: ***cv.glmnet*** chooses the best $\lambda = 0.935$ with 35 parameters (MSE =2808.36)

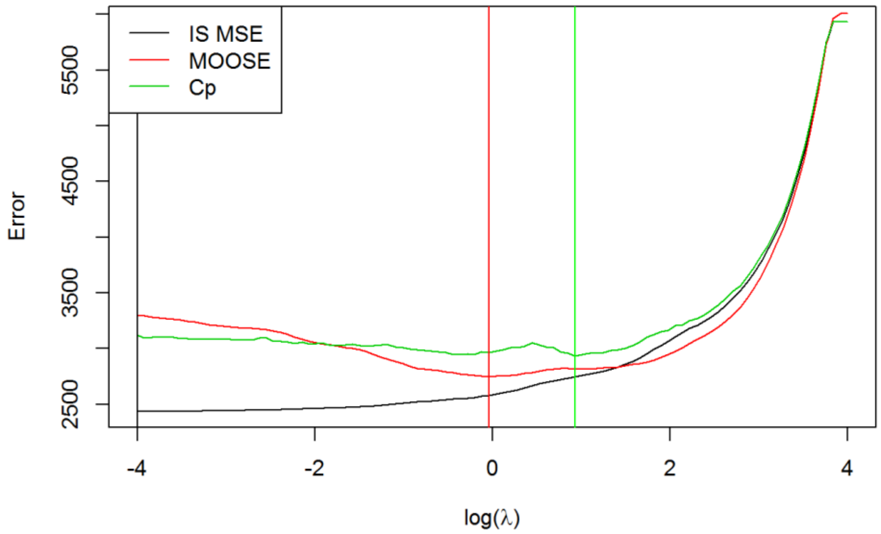

Figure 6: Comparison of MSE, MOOSE and C_p with respective best λ

(C) The C_p statistic (Mallow's C_p is defined as

$$C_p(\hat{\beta}_\lambda) = MSE(\hat{\beta}_\lambda) + 2 \cdot \frac{s_\lambda}{n} \hat{\sigma}^2$$

Where s_λ is the degrees of freedom of the fit (i.e. the number of nonzero parameters selected at that particular value of λ), and $\hat{\sigma}^2$ is an estimate of the residual variance. Figure 6 shows the comparison between MSE, MOOSE, and C_p. The C_p statistic closely follows the cross-validated MOOSE when λ increases. The minimum from C_p with best λ = 2.533, which is different from that of MOOSE with best λ = 0.960, implies 16 non-zero coefficients. The in-sample MSE clearly prefers over-fitting.