STAT 5630, Fall 2019

Penalized Linear Regression

Xiwei Tang, Ph.D. <xt4yj@virginia.edu>

University of Virginia

September 12, 2019

Shrinkage Methods

Different Variable Selection Methods

- Best subset selection
  - Computationally expensive
  - Not feasible when p is large
- Forward/backward selection
  - No guarantee to find the best global submodel
  - The selection process is discrete (“add” or “drop”), often leads to high variance.
- Shrinkage (regularization, penalization) methods
  - A continuous process, does not suffer from high variability

Motivation

- The OLS estimator is the BLUE, but not necessary the ”best”.
- Recall that the prediction accuracy is

$$\text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

- Consider about bias-variance trade-off.
- Generally, by regularizing (shrinking, penalizing) the estimator in some way, its variance can be reduced; if the corresponding increase in bias is small, we have better prediction accuracy

Motivating Example

- Suppose we obtain the following estimated linear model:

  $$\hat{y} = 1 + x_1 + 2x_2 + 0.0001x_3 + 0.000001x_4 + 0.0000001x_5,$$

  and x_i's ($i = 1, \dots, 5$) are of the same scale. Q: Is this a good model?

Motivating Example

- Suppose we obtain the following estimated linear model:

  $$\hat{y} = 1 + x_1 + 2x_2 + 0.0001x_3 + 0.000001x_4 + 0.0000001x_5,$$

  and x_i's ($i = 1, \ldots, 5$) are of the same scale. Q: Is this a good model?

- Removing x_3, x_4 and x_5 may lead to a biased model, however, it would also reduce the variance.

Motivating Example

- Suppose we obtain the following estimated linear model:

$$\hat{y} = 1 + x_1 + 2x_2 + 0.0001x_3 + 0.000001x_4 + 0.0000001x_5,$$

  and x_i's ($i = 1, \ldots, 5$) are of the same scale. Q: Is this a good model?

- Removing x_3, x_4 and x_5 may lead to a biased model, however, it would also reduce the variance.

- Recall the model selection criterion

Goodness-of-fit + Complexity-Penalty

Shrinkage Method

- Consider a penalized loss function:

$$RSS + p_\lambda(\boldsymbol{\beta})$$

- RSS (sum of squared residuals) measures Goodness of Fit
- $p_\lambda(\cdot)$ is a penalty function and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$, which controls model complexity
- $\lambda \geq 0$ is a tuning parameter which controls the penalization level.

Shrinkage Methods: Some Penalties

- ℓ_2 penalty: Ridge regression
- ℓ_1 penalty: Lasso
- Connecting the two: Bridge, elastic net
- Bias reduction: adaptive Lasso, SCAD, MCP
- Penalties for special data structures: grouped lasso, fused lasso

Ridge Regression

Ridge Regression

Penalizing the square of the coefficients $\|\boldsymbol{\beta}\|^2 = \sum_{j=1}^{p} \beta_j^2$

$$\widehat{\boldsymbol{\beta}}^{\text{ridge}} = \underset{\boldsymbol{\beta}}{\arg\min}\ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2$$

- Hoerl and Kennard (1970); Tikhonov (1943)
- $\lambda \geq 0$ is a tuning parameter (penalty level)
- The coefficients $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ are shrunken towards 0.

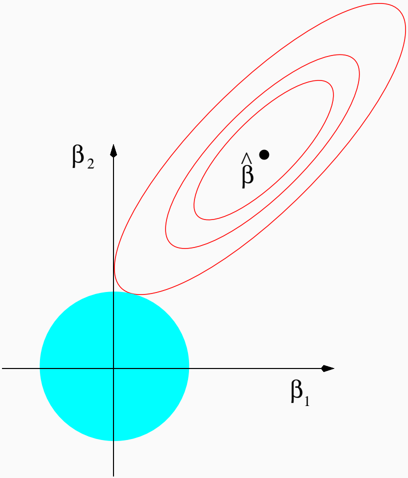
Ridge Regression

An equivalent formulation is given by

$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} \beta_j^2 \leq s \end{aligned}$$

- One-to-one correspondence between the parameters λ and s
- Ridge regression is usually employed to deal with highly correlated covariates
    - It's likely to obtain very large coefficients' estimates when there are many highly correlated variables
    - Ridge regression alleviate this problem by imposing a size constraint

Linear Regression



Ridge constrained solution

Ridge Regression for Correlated Variables

```
> library(MASS)
> set.seed(1)
> n = 30
>
> # highly correlated variables
> X = mvrnorm(n, c(0, 0), matrix(c(1,0.999, 0.999, 1), 2,2))
> y = rnorm(n, mean=1 + X[,1] + X[,2])
>
> # compare parameter estimates
> summary(lm(y~X))$coef
              Estimate Std. Error   t value     Pr(>|t|)
(Intercept)   1.038007  0.1647551  6.300302 9.627026e-07
X1          -11.272638  4.6402098 -2.429338 2.205727e-02
X2           13.265586  4.6315269  2.864193 7.993486e-03
> lm.ridge(y~X, lambda=5)
                 X1        X2
1.1214448 0.8770568 0.9836474
```

Solution for Ridge Regression

- For a fixed tuning parameter λ, we want to minimize

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

- Take derivative with respect to $\boldsymbol{\beta}$ and set to zero, we have the solution of the Ridge regression

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

- $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is still a linear estimator

- If there are highly correlated variables, $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is close to singular

- $\mathbf{X}^{\mathsf{T}}\mathbf{X}+\lambda\mathbf{I}$ is always invertible, hence $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is unique

- As $\lambda\to 0$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}}\to\widehat{\boldsymbol{\beta}}^{\text{ols}}$
- As $\lambda\to\infty$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}}\to\mathbf{0}$

Notes on the scale of predictors

The solution is not invariant with respect to the scale of the predictors!

we normalize the columns of the design matrix $\mathbf{X}$ such that they have unit sample variance. We further center the data, that is, both y and the columns of $\mathbf{X}$ have mean zero. Then, we can fit a linear regression model without an intercept (we don't penalize the intercept). The parameters on the original scale can be reversely solved.

Some packages (e.g. "glmnet") in R handles the centering and scaling automatically: it will do the transformation before running the algorithm, and then will transform the obtained results back to the original scale.

Bias and Variance of Ridge Regression

- When $\widehat{\boldsymbol{\beta}}^{\,\text{ols}}$ exists, we can also write

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{\,\text{ridge}} &= (\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}\\ &= (\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X})(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}\\ &= (\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}(\mathbf{X}^\mathsf{T}\mathbf{X})\widehat{\boldsymbol{\beta}}^{\,\text{ols}}\\ &= \mathbf{Z}\widehat{\boldsymbol{\beta}}^{\,\text{ols}}\end{aligned}$$

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

- The variance of $\widehat{\boldsymbol{\beta}}^{\,\text{ridge}}$ is

$$\begin{aligned}\text{Var}\big(\widehat{\boldsymbol{\beta}}^{\,\text{ridge}}\big) =&(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X}+\lambda\mathbf{I})^{-1}\\ =&\sigma^2\mathbf{W}\mathbf{X}^\mathsf{T}\mathbf{X}\mathbf{W}\end{aligned}$$

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

- The variance $\sum_j \text{Var}(\widehat{\beta}_j)$ is monotone decreasing of λ.

Bias and Variance of Ridge Regression

- The bias of the ridge estimator is

  $$\text{Bias}(\widehat{\boldsymbol{\beta}}^{\text{ridge}}) = \mathbf{Z}\boldsymbol{\beta} - \boldsymbol{\beta} = [(\mathbf{X}^{\mathsf{T}}\mathbf{X} + \lambda\mathbf{I})^{-1}(\mathbf{X}^{\mathsf{T}}\mathbf{X}) - \mathbf{I}]\boldsymbol{\beta} = -\lambda\mathbf{W}\boldsymbol{\beta},$$

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

- The $\sum_j \text{Bias}^2(\widehat{\beta}_j^{\text{ridge}})$ is a monotone increasing function of λ.

- Suppose we have orthonormal design matrix ($\mathbf{X}^{\mathsf{T}}\mathbf{X} = \mathbf{I}$)

  $$\widehat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{I} + \lambda\mathbf{I})^{-1}\widehat{\boldsymbol{\beta}}^{\text{ols}}$$

  meaning that we just need to shrink $\widehat{\boldsymbol{\beta}}^{\text{ols}}$ by $(1+\lambda)^{-1}$, i.e.,

  $$\widehat{\boldsymbol{\beta}}_j^{\text{ridge}} = \frac{1}{1+\lambda}\widehat{\boldsymbol{\beta}}_j^{\text{ols}}.$$

- $\text{Var}(\widehat{\boldsymbol{\beta}}_j^{\text{ridge}}) = \frac{1}{(1+\lambda)^2}\text{Var}(\widehat{\boldsymbol{\beta}}_j^{\text{ols}})$ and $\text{Bias}(\widehat{\boldsymbol{\beta}}_j^{\text{ridge}}) = \frac{-\lambda}{1+\lambda}\boldsymbol{\beta}_j$

Understanding the Shrinkage

- When the columns of $\mathbf{X}$ are not orthogonal, let's take a singular value decomposition (SVD) of $\mathbf{X}$:

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

  where

  - $\mathbf{U}_{n\times p}$: columns $\mathbf{u}_j$'s form an orthonormal basis for the column space of $\mathbf{X}$, $\mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}$
  - $\mathbf{V}_{p\times p}$: orthogonal matrix with $\mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I}$
  - $\mathbf{D}_{p\times p}$: diagonal matrix with diagonal entries $d_1 \geq d_2 \geq \ldots \geq d_p \geq 0$ being the sigular values of $\mathbf{X}$
  - Sometimes we can write $\mathbf{X} = \mathbf{F}\mathbf{V}^{\mathsf{T}}$ where each columns of $\mathbf{F}_{n\times p} = \mathbf{U}\mathbf{D}$ is the so-called principal components and each column of $\mathbf{V}$ is a principal direction.

Understanding the Shrinkage

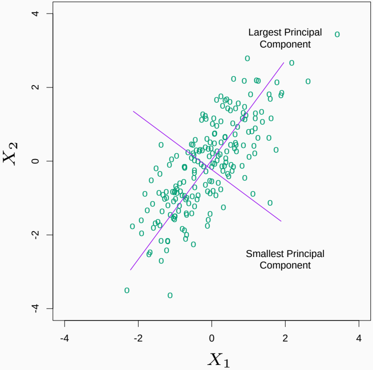


FIGURE 3.9. *Principal components of some input data points. The largest principal component is the direction that maximizes the variance of the projected data, and the smallest principal component minimizes that variance. Ridge regression projects* $\mathbf{y}$ *onto these components, and then shrinks the coefficients of the low-variance components more than the high-variance components.*

Understanding the Shrinkage

- The relationship between Ridge and PCA can be understood as (assuming $\mathbf{X}$ centered)

$$\widehat{\Sigma} = \frac{1}{n}\mathbf{X}^\mathsf{T}\mathbf{X} = \frac{1}{n}\mathbf{V}\mathbf{D}^2\mathbf{V}^\mathsf{T}$$

- The Ridge estimate of $\mathbf{y}$

$$\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ridge}} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \sum_{j=1}^{p}\mathbf{u}_j\left(\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^\mathsf{T}\mathbf{y}\right)$$

- $\mathbf{u}_j$ is the normalized jth principal component of $\mathbf{X}$

- Shrinks more when d_j is smaller

Understanding the Shrinkage

- Hence, Ridge regression can be understood as

  (1) Perform principle component analysis of $\mathbf{X}$
  (2) Project $\mathbf{y}$ onto the principal components: $\mathbf{u}_j^\intercal \mathbf{y}$ for each j
  (3) Shrink the projections by the factor $d_j^2/(d_j^2 + \lambda)$

- Directions with smaller eigen values d_j get more shrinkage.

- The final ridge estimate of $\mathbf{y}$ is a sum of the p shrunk projections.

Degrees of Freedom for Ridge Regression

- Although $\widehat{\boldsymbol{\beta}}^{\text{ridge}}$ is p-dimensional, it does not seem to use the full potential of the p covariates due to the shrinkage.
- For example, if λ is VERY large, then all the parameter estimates shrink to 0, then intuitively, the df should be close to 0.
- If λ is 0, then we reduce to the OLS with df equals p
- The df of a Ridge regression should be between 0 and p

Degrees of Freedom for Ridge Regression

- Recall our definition of degrees of freedom (df) used in the kNN example:

$$\text{df}(\widehat{f}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{Cov}(\widehat{y}_i, y_i)$$

- For Ridge regression, we have

$$\widehat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = \sum_{j=1}^{p} \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^\mathsf{T}\mathbf{y}$$

- Then the effective df is

$$\text{df}(\lambda) = \text{Trace}\big(\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\big) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

Bayesian Interpretation

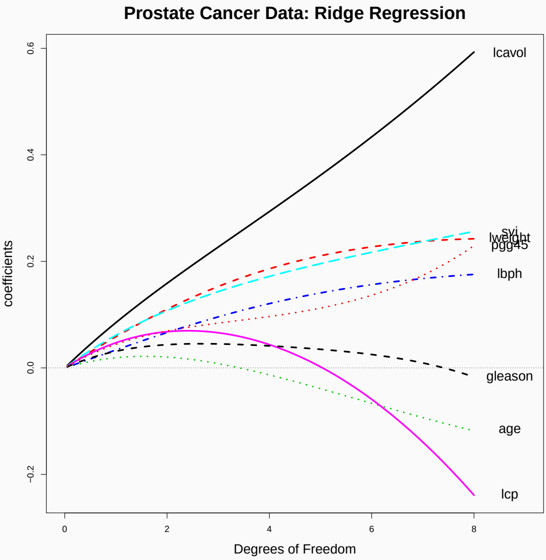
- The Ridge regression solution can be viewed from a Bayesian prospective, where we give a prior distribution $\boldsymbol{\beta} \sim \mathcal{N}(0, \sigma^2/\lambda)$.
- Then the posterior distribution of β is normal, with posterior mean

  $$\left(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\mathsf{T}\mathbf{y},$$

  and posterior variance

  $$\sigma^2(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\mathsf{T}\mathbf{X}(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda\mathbf{I})^{-1}$$

Prostate Cancer Example



Selecting the Tuning Parameter λ

- The R command lm.ridge (from MASS package) returns GCV, which can be used to select λ.
- glmnet can also fit Ridge regression by setting $\alpha = 0$
- The leave-one-out cross-validation (CV) error
  1 Hold the ith sample (x_i, y_i) as a test sample, fit a regression model based on the remaining $(n-1)$ observations, and denote the coefficient as $\widehat{\boldsymbol{\beta}}_{[-i]}$
  2 Calculate the prediction error on the holdout sample $(y_i - x_i^\mathsf{T}\widehat{\boldsymbol{\beta}}_{[-i]})^2$
  3 Repeat for every sample and

$$\text{CV} = \sum_{i=1}^{n} \left(y_i - x_i^\mathsf{T}\widehat{\boldsymbol{\beta}}_{[-i]}\right)^2$$

Selecting the Tuning Parameter λ

- In LS, we do not need to run n regression models to calculate the leave-one-out CV

$$\text{CV} = \sum_{i=1}^{n} \left(y_i - x_i^\mathsf{T} \widehat{\boldsymbol{\beta}}_{[-i]}\right)^2 = \sum_{i=1}^{n} \left(\frac{y_i - x_i^\mathsf{T} \widehat{\boldsymbol{\beta}}}{1 - \mathbf{H}_{ii}}\right)^2$$

  where $\mathbf{H}_{ii}$ is the (i, i)-th entry of the projection matrix $\mathbf{H}$.
- Hence, we only need to run LS once and rescale the residuals.

Selecting the Tuning Parameter λ

- For Ridge regression, it is very similar

$$\text{CV}(\lambda) = \sum_{i=1}^{n} \left(\frac{y_i - x_i^{\mathsf{T}} \widehat{\boldsymbol{\beta}}_{\lambda}^{\text{ridge}}}{1 - \mathbf{S}_{\lambda}(i,i)} \right)^2$$

where $\mathbf{S}_{\lambda}(i,i)$ is the (i,i)-th entry of the projection matrix

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

- A modified version is called GCV (generalized CV)

$$\text{GCV}(\lambda) = \frac{\sum_{i=1}^{n} \left(y_i - x_i^{\mathsf{T}} \widehat{\boldsymbol{\beta}}_{\lambda}^{\text{ridge}} \right)^2}{\left(n - \text{Trace}(\mathbf{S}_{\lambda}) \right)^2}$$

Lasso: Least Absolute Shrinkage and Selection Operator

Lasso

Motivation

- The Ridge regression shrinks the coefficients towards 0, however, they are NOT exactly zero. Hence, we haven't achieve any "selection" of variables.
- **Parsimony (Sparsity)**: we would like to select a small subset of predictions. Forward/backword/subset does not provide global solution and can be myopic at each step.

Lasso

Least absolute shrinkage and selection operator (Tibshirani 1996)

$$\arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$
- Shrinkage of the ℓ_1 norm of the parameters
- Selection of parameters, some will be exactly 0

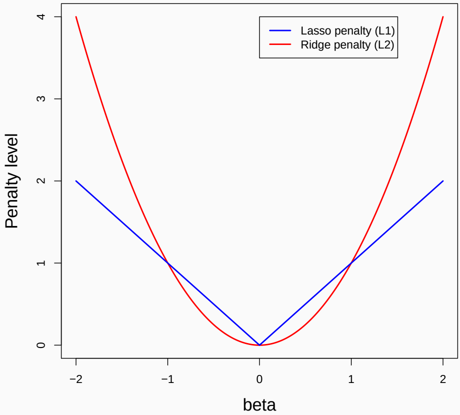
Equivalent Formulation

- The Lasso optimization problem is equivalent to

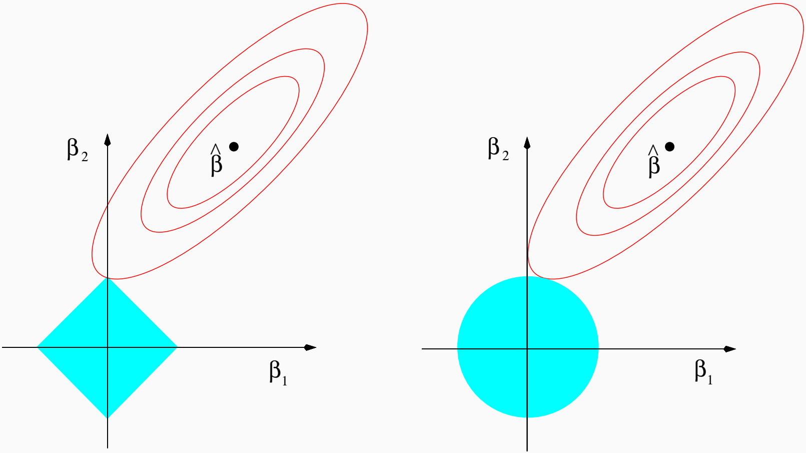
$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} |\beta_j| \leq s \end{aligned}$$

- Each value of λ corresponds to an unique value of s.
- The absolute value function is shape at zero.

Lasso


Penalty level
0
1
2
3
4
−2
−1
0
1
2
beta
Lasso penalty (L1)
Ridge penalty (L2)

Constraint Linear Regression



Comparing Lasso and Ridge solutions

Lasso Under Orthogonal Design

Again, it will be helpful to view Lasso assuming orthogonal design, i.e., $\mathbf{X}^\mathsf{T}\mathbf{X} = \mathbf{I}_p$. Then

$$\begin{aligned}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 =& \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} + \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 \\ =& \|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}}\|^2 + \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2\end{aligned}$$

where the cross product term

$$2(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}})^\mathsf{T}(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}) = 2\mathbf{r}^\mathsf{T}(\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}) = 0,$$

since the second term is in the column space of $\mathbf{X}$, while $\mathbf{r}$ is orthogonal to that space.

Lasso Under Orthogonal Design

- Since $\|\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}}\|^2$ is not a function of $\boldsymbol{\beta}$, we only need to minimize

$$\|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- Then, we have

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}^{\text{lasso}} &= \arg\min_{\boldsymbol{\beta}} \, \|\mathbf{X}\widehat{\boldsymbol{\beta}}^{\text{ols}} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} \, (\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta})^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{X}(\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} \, (\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta})^{\mathsf{T}}(\widehat{\boldsymbol{\beta}}^{\text{ols}} - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \\
&= \arg\min_{\boldsymbol{\beta}} \, \sum_{j=1}^{p}(\widehat{\beta}_j^{\text{ols}} - \beta_j)^2 + \lambda|\beta_j|.
\end{aligned}$$

- This means we can solve the lasso estimators individually from the OLS estimator.

Lasso Under Orthogonal Design

- Each of the β_j's is essentially solving for

$$\arg\min_x \ (x-a)^2 + \lambda|x|, \quad \lambda > 0$$

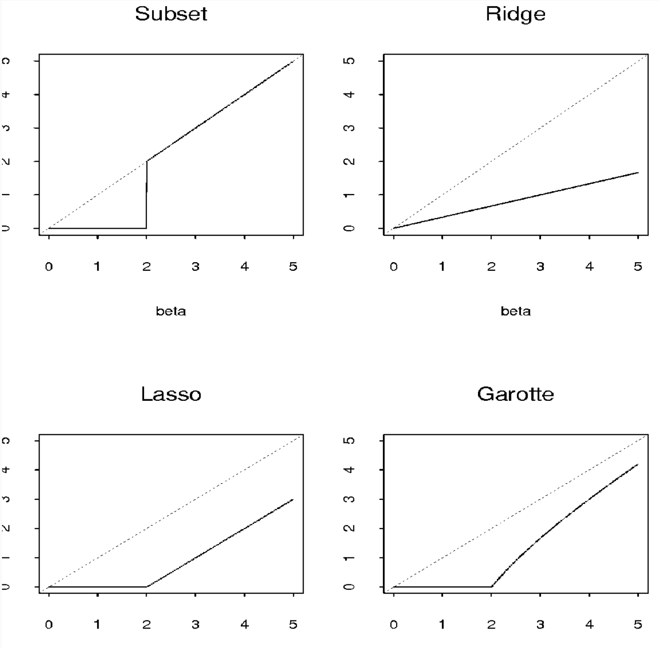
- The solution is simply

$$\widehat{\beta}_j^{\text{lasso}} = \begin{cases} \widehat{\beta}_j^{\text{ols}} - \lambda/2 & \text{if} \quad \widehat{\beta}_j^{\text{ols}} > \lambda/2 \\ 0 & \text{if} \quad |\widehat{\beta}_j^{\text{ols}}| \leq \lambda/2 \\ \widehat{\beta}_j^{\text{ols}} + \lambda/2 & \text{if} \quad \widehat{\beta}_j^{\text{ols}} < -\lambda/2 \end{cases} \tag{1}$$

$$= \text{sign}\big(\widehat{\beta}_j^{\text{ols}}\big)\Big(|\widehat{\beta}_j^{\text{ols}}| - \lambda/2\Big)_+$$

- A large λ will shrink some of the coefficients to exactly zero, which achieves “variable selection”.

Different Shrinkages



Computation of Lasso Solution

- The Lasso problem is convex, although it may not be strictly convex in $\boldsymbol{\beta}$ when p is large
- The solution is a global minimum, but may not be the unique global one
- The Lasso solution is unique UNDER SOME ASSUMPTIONS
- Under some assumptions, the Lasso problem will be strictly convex and the solution is unique.

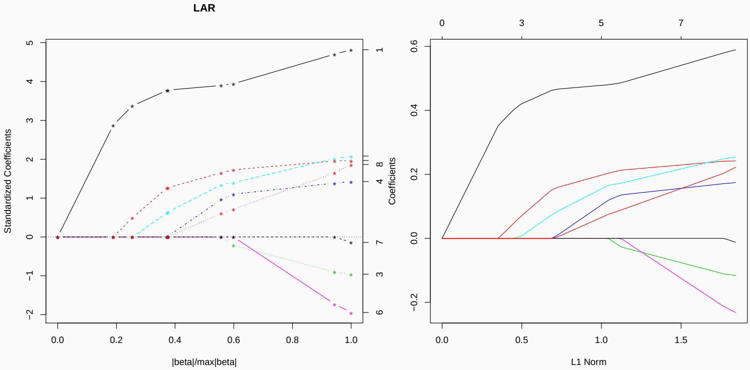
Computation of Lasso Solution

- Shooting algorithm (Fu 1998): sequentially and iteratively update each parameter estimate (coordinate descent algorithm).

- Least angle regression (Efron et al. 2004)
  - The path of solutions is piecewise linear in λ
  - Cost is approximately one least-squares calculation $\mathcal{O}(np^2)$
  - Connection with stagewise regression

- Coordinate descent (Friedman et al 2010): The most popular implementation, glmnet package; $\mathcal{O}(np)$
  - Also provides the solution path for the entire sequence of λ, starting with the largest one
  - Use the previous estimation of $\boldsymbol{\beta}$ as a warm start for smaller λ

Lasso



Comparing least angle regression with coordinate descent

ℓ_q Penalties

- Ridge is ℓ_2 penalty
- Lasso is ℓ_1 penalty
- Best subset is ℓ_0 penalty

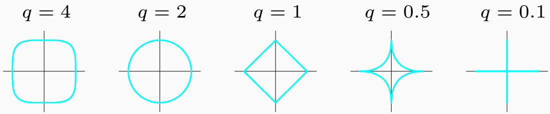


FIGURE 3.12. *Contours of constant value of $\sum_j |\beta_j|^q$ for given values of q.*

- Elastic-net is a hybrid of ℓ_1 and ℓ_2:

$$\lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2$$

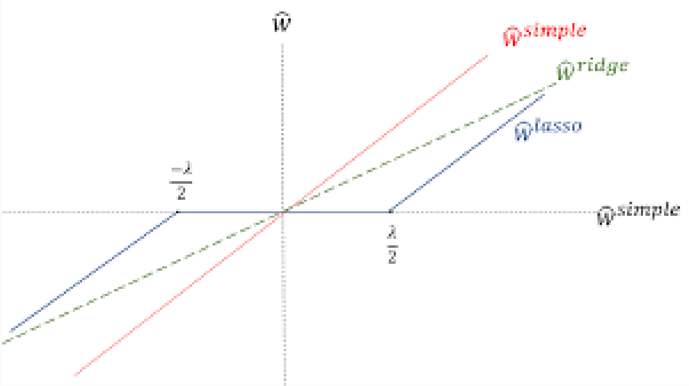
R Functions

- Use R help and R manuals
- Linear models: function lm
- QR decomposition qr ; Cholesky decomposition chol ; PCA princomp , prcomp ; SVD svd .
- Ridge regression:
  - package MASS ; function lm.ridge
  - package glmnet ; function glmnet and cv.glmnet with $alpha = 0$
- Lasso:
  - package lars ; function lars
  - package glmnet ; function glmnet and cv.glmnet with $alpha = 1$

Bias Correction and Unbiased Penalties

Biased Estimator

- l_p penalized estimators are biased!!!



Bias Correction

- Bias correction:
  - Non-negative garrote
  - Adaptive Lasso

- Unbiased penalty:
  - SCAD
  - MCP

Non-negative Garrote

- Non-negative Garrote was proposed by Breiman (1994)
- Suppose we can have an initial estimate: $\widehat{\boldsymbol{\beta}}^{\text{ols}}$
- We can perform a model by shrinking the coefficients:

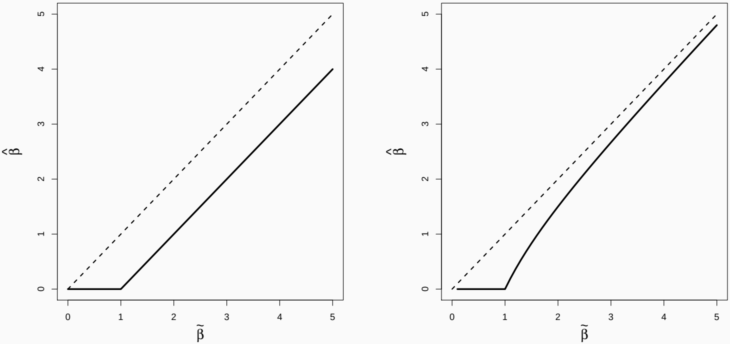
$$\underset{d_1,\ldots,d_p}{\text{minimize}} \quad \frac{1}{2n}\left\|\mathbf{y} - \sum_{j=1}^{p} d_j \widehat{\beta}_j^{\text{ols}} \mathbf{x}_j\right\|^2 + \lambda \sum_{j=1}^{p} d_j,$$

  subject to $d_j \geq 0$ for all j.
- Final estimate $\widehat{\beta}_j^{\text{ng}} = \widehat{d}_j \beta_j^{\text{ols}}$
- In orthogonal designs, the optimal d_j's are

$$d_j = \left(1 - \frac{\lambda}{(\beta_j^{\text{ols}})^2}\right)_+$$

  which can be shrunk to exactly 0 if β_j^{ols} is small.
- For β_j^{ols} sufficiently large, d_j is almost 1, which reduces the bias.

Comparison: Lasso vs. Garrote



Comparing Lasso shrinkage with non-negative garrote shrinkage

Adaptive Lasso

- Adaptive Lasso was proposed by Hui Zou (2006)
- Suppose we can have an initial estimate $\widetilde{\boldsymbol{\beta}}$ that is $\sqrt{n}$ consistent
- Adjust the Lasso penalty based on how large $\widetilde{\boldsymbol{\beta}}$ is

$$\widehat{\boldsymbol{\beta}} = \underset{\beta_1,\ldots,\beta_p}{\arg\min} \quad \frac{1}{2n}\Big\|\mathbf{y} - \sum_{j=1}^{p} \beta_j \mathbf{x}_j\Big\|^2 + \lambda \sum_{j=1}^{p} \frac{1}{|\widetilde{\beta}_j|^\gamma}|\beta_j|,$$

  for a pre-chosen $\gamma > 0$.
- Note: the penalty is essentially $\frac{\lambda}{|\widetilde{\beta}_j|^\gamma}$, which will be different for each β_j. Large $\widetilde{\beta}_j$ means small penalty, which reduces the bias

Adaptive Lasso

- The adaptive Lasso and the Non-negative Garrote are almost the same. If we take $\gamma = 1$ and use $\widehat{\boldsymbol{\beta}}^{\text{ols}}$ as the initial estimator $\widetilde{\boldsymbol{\beta}}$ in then adaptive Lasso, then we are solving for

$$\widehat{\boldsymbol{\beta}} = \underset{\beta_1, \ldots, \beta_p}{\arg\min} \quad \frac{1}{2n} \Big\| \mathbf{y} - \sum_{j=1}^{p} \beta_j \mathbf{x}_j \Big\|^2 + \lambda \sum_{j=1}^{p} \frac{1}{|\widehat{\beta}_j^{\text{ols}}|} |\beta_j|,$$

which is equivalent to treating $\frac{|\beta_j|}{|\widehat{\beta}_j^{\text{ols}}|}$ as d_j, and rescale each $\mathbf{x}_j$

$$\text{minimize} \quad \frac{1}{2n} \Big\| \mathbf{y} - \sum_{j=1}^{p} \frac{|\beta_j|}{|\widehat{\beta}_j^{\text{ols}}|} \big(\widehat{\beta}_j^{\text{ols}} \mathbf{x}_j \big) \Big\|^2 + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\widehat{\beta}_j^{\text{ols}}|},$$

if we require $\widehat{\beta}_j$ to have the same sign as $\widehat{\beta}_j^{\text{ols}}$.

Adaptive Lasso

- The adaptive Lasso can be easily implemented using existing R packages, such as glmnet, if we simply get an initial estimator and rescale each covariate.
- We don't have to use $\widehat{\beta}_j^{\text{ols}}$ as the initial guess. In practice, when $p > n$, we can use the lasso estimates as the initial value.
- Adaptive Lasso Algorithm (using Lasso as initial estimator, and use $\gamma = 1$)
  1. Fit a Lasso model and obtain $\widehat{\beta}_j^{\text{lasso}}$'s
  2. Rescale covariates $\mathbf{x}_j^* = \mathbf{x}_j \cdot \widehat{\beta}_j^{\text{lasso}}$
  3. Refit the Lasso model using $\mathbf{X}^*$ without standardizing the columns and obtain $\widehat{\beta}_j^*$'s
  4. Recover the original parameter estimates $\widehat{\beta}_j^* \cdot \widehat{\beta}_j^{\text{lasso}}$ for all j.

Unbiased Penalties

- The above two approaches are two-stage approaches. The motivation was to adaptively choose the penalty level for each of the parameter estimates

- Is there a direct approach ? Fan and Li (2001) suggest three properties that a penalty function should have
  - Unbiasedness: The resulting estimator is nearly unbiased when the true unknown parameter is large to avoid unnecessary modeling bias

  - Sparsity: The resulting estimator is a thresholding rule, which automatically sets small estimated coefficients to zero to reduce model complexity

  - Continuity: The resulting estimator is continuous in data to avoid instability in model prediction

  - Oracle Property: almost the same as the OLS estimator with truth

SCAD

- Smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001)
- Solve for the penalized loss function

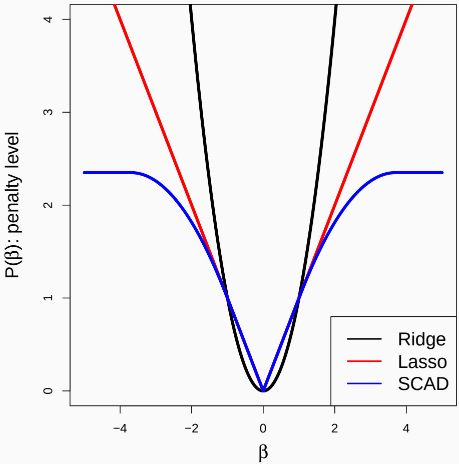
$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} P(\beta_j),$$

  where the penalty is defined with $\lambda > 0$ and an extra parameter $\gamma > 2$

$$P(\beta) = \begin{cases} \lambda|\beta| & \text{if} \quad |\beta| \leq \lambda \\ \frac{2\gamma\lambda|\beta|-\beta^2-\lambda^2}{2(\gamma-1)} & \text{if} \quad \lambda < |\beta| < \gamma\lambda \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if} \quad |\beta| \geq \gamma\lambda \end{cases}$$

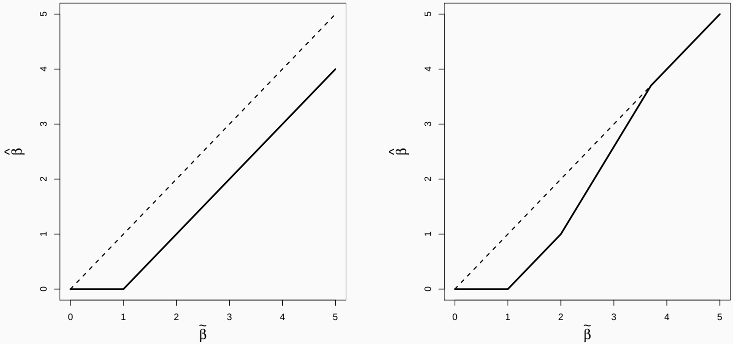
- This penalty is non-convex. It coincides with the Lasso until $|\beta| = \lambda$, then smoothly transits to a quadratic function until $|\beta| = \gamma\lambda$, after which it remains constant for all $|\beta| > \gamma\lambda$.

Comparison



Comparing Lasso, Ridge and SCAD ($\gamma = 3.7$)

Comparison: Lasso vs. SCAD



Comparing Lasso shrinkage with SCAD shrinkage

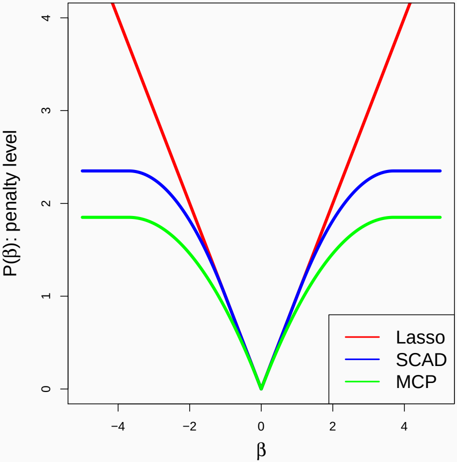
MCP

- Minimax concave penalty (MCP) is another unbiased penalty (Zhang, 2010)
- Exactly the same formulation of the penalized loss function, with a penalty term defined as

$$P(\beta) = \begin{cases} \lambda|\beta| - \frac{\beta^2}{2\gamma} & \text{if} \quad |\beta| \leq \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2 & \text{if} \quad |\beta| \geq \gamma\lambda \end{cases}$$

  for some $\gamma > 1$.
- The maximum concavity of this penalty function is $1/\gamma$, which is exactly controlled

Comparison

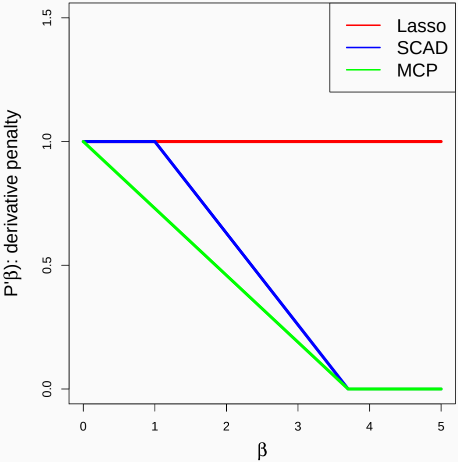


Comparing Lasso, SCAD ($\gamma = 3.7$), and MCP ($\gamma = 3.7$)

Derivatives of Penalties: $P'(|\beta|)$

- Lasso remains that rate, regardless of the size of β
- SCAD and MCP will smoothly relaxes the rate down to zero as the absolute value of β increases
- The difference between SCAD and MCP is that MCP relaxes the penalization rate immediately while with SCAD the rate remains flat for a while before decreasing

Comparison



Comparing derivatives of Lasso, SCAD ($\gamma = 3.7$), and MCP ($\gamma = 3.7$)

Penalties for special data structures

Penalties for special data structures

- In many applications, design matrix has some special structures
- We are going to discuss two cases:
    - Group Lasso: An X variable is categorical with more than 2 categories
    - Fused Lasso: X variables has a certain order and is highly correlated

Group Lasso

- When a variable has multiple categories and is nominal, its effect cannot be described using just one parameter
- Suppose X_1 has three categories, and the corresponding parameters are a vector $\beta_1 = (\beta_{11}, \beta_{12}, \beta_{13})$
- If we apply a regular Lasso, the penalty term is

$$P(\beta_1) = \lambda \sum_{k=1}^{3} |\beta_{1k}|$$

- We may end up selecting a nonzero β_{13} but not the other two categories (thinking them as having the same effect)

Group Lasso

- Would it be possible to select all categories of X_1 as long as one of them is selected?
- The group Lasso penalty:

$$P(\beta_1) = \lambda \|\beta_1\|_2 = \lambda \sqrt{\beta_{11}^2 + \beta_{12}^2 + \beta_{13}^2}$$

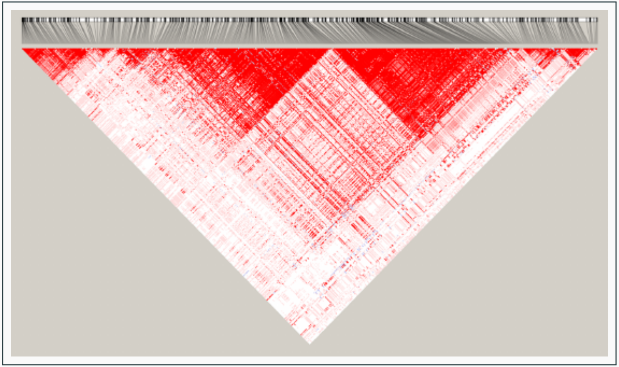
- We apply this penalty to each group. Of course for group with size 1, it is just the Lasso.
- In general, if we have G different groups, the over all penalty is

$$P(\boldsymbol{\beta}) = \lambda \sum_{g=1}^{G} \|\boldsymbol{\beta}_{\mathcal{I}g}\|_2$$

where $\mathcal{I}g$ is the index set belonging to the g'th group.

Fused Lasso

- In many real applications, the features X are ordered in a meaningful way



Example: LD plot from Pistis et. al. (2013)

Fused Lasso

- Nearby X_j's are highly correlated and they may contribute to the outcome together, meaning that if we believe X_j is related to Y, then X_{j-1} and X_{j+1} may also do. And their coefficients are likely to be the same or close.
- Lasso will have difficulty identifying all of them, since it is likely to pick one and conclude all signals it.
- Fused Lasso takes advantage of this special ordering of X.

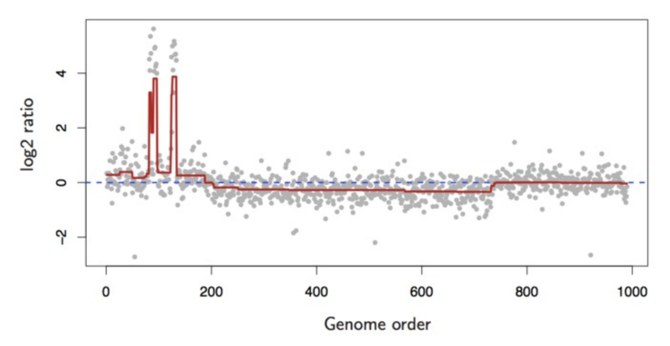
Fused Lasso

- Fused Lasso solves the optimization problem:

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|$$

- Encouraging nearly parameter estimates to be the same
- However, coordinate descent may not work in this problem.

Fused Lasso



Fused Lasso fitting

Appendix: Convex Optimization

Convex Optimization

- The problem: minimizing a convex function in a convex set

$$
\begin{aligned}
\underset{\boldsymbol{\beta}}{\text{minimize}} \quad & f(\boldsymbol{\beta}) \\
\text{subject to} \quad & g_i(\boldsymbol{\beta}) \leq 0, \quad i = 1, \ldots, m \\
& \mathbf{A}\boldsymbol{\beta} = b
\end{aligned}
$$

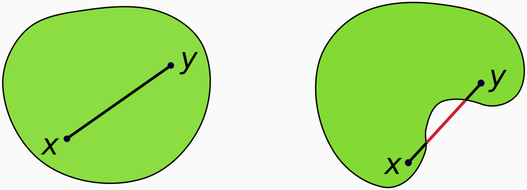
- Examples:
  - Linear regression: minimize $\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$, subject to none.
  - Ridge regression: minimize $\frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$, subject to $\sum_{j=1}^{p} \beta_j^2 < s$
  - First principal component: maximize $\boldsymbol{\beta}^\mathsf{T}\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\beta}$, subject to $\boldsymbol{\beta}^\mathsf{T}\boldsymbol{\beta} = 1$

Convex Optimization

- What is a convex set $C \in \mathbb{R}^p$?

$$\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in C \quad \Longrightarrow \quad \alpha\boldsymbol{\beta}_1 + (1-\alpha)\boldsymbol{\beta}_2 \in C, \quad \forall\ 0 \leq \alpha \leq 1.$$

- Visual:

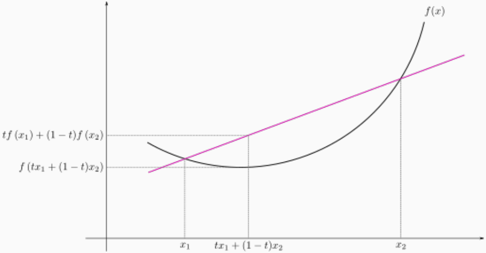


Convex Optimization

- What is a convex function $f : \mathbb{R}^p \to \mathbb{R}$?

$$f\big(\alpha\boldsymbol{\beta}_1 + (1-\alpha)\boldsymbol{\beta}_2\big) \leq \alpha f(\boldsymbol{\beta}_1) + (1-\alpha) f(\boldsymbol{\beta}_2) \quad \forall\ 0 \leq \alpha \leq 1.$$

- Visual:



- Famous result: Jensen's inequality

Convex functions

- To comply with notations in the literature, I will use $\mathbf{x}$ as the argument instead of using $\boldsymbol{\beta}$, and we are interested in the function $f(\mathbf{x})$.
- Examples of convex functions:
  - $\exp(x)$, $-\log(x)$, etc.
  - Affine: $a^\mathsf{T}\mathbf{x} + b$ is both convex and concave
  - Quadratic: $\frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + b^\mathsf{T}\mathbf{x} + c$, if $\mathbf{A}$ is positive semidefinite.
  - All norms: ℓ_p
- A function is strictly convex if we can remove the equal sign:

$$f\big(\alpha\boldsymbol{\beta}_1 + (1-\alpha)\boldsymbol{\beta}_2\big) < \alpha f(\boldsymbol{\beta}_1) + (1-\alpha) f(\boldsymbol{\beta}_2) \quad \forall\ 0 \le \alpha \le 1.$$

- f is convex $\Longleftrightarrow -f$ is concave

Properties of Convex functions

- First-order property: If f is differentiable with convex domain, then f is convex iff

$$f(\mathbf{x}^*) \geq f(\mathbf{x}) + \triangledown f(\mathbf{x})^{\mathsf{T}}(\mathbf{x}^* - \mathbf{x})$$

- If we have a feasible point $\mathbf{x}$ with $\triangledown f(\mathbf{x}) = \mathbf{0}$, it means all alternative points $\mathbf{x}^*$ have larger function value $f(\mathbf{x}^*) \geq f(\mathbf{x})$.
- Hence, we call $\mathbf{x}$ a local minimizer. It may not be unique, but its as good as any other solution.
- Example: In a linear regression if we have linearly dependent columns in the design matrix. The solution of parameters is not unique.

Properties of Convex functions

- Second-order property: If f is twice differentiable with convex domain, then f is convex iff

$$\bigtriangledown^2 f(\mathbf{x}) \succeq 0 \quad \text{for any } \mathbf{x} \text{ in the domain,}$$

  where

$$\mathbf{H}(\mathbf{x}) = \bigtriangledown^2 f(\mathbf{x}) = \left(\frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \right)$$

- $\mathbf{H}(\mathbf{x})$ is the Hessian matrix.
- If $\bigtriangledown^2 f(\mathbf{x}) \succ 0$ (positive definite), meaning f is strictly convex, then a local minimizer is also a global minimizer.
- Example: In linear regression when $\mathbf{X}^\mathsf{T}\mathbf{X} \succ 0$, i.e., invertible.