

Mathematical techniques in data science

Lecture 13: Shrinkage methods for linear regression

Mathematical techniques in data sciences

- A short introduction to statistical learning theory
- Tree-based methods — boosting and bootstrapping
- SVM – the kernel trick
- Linear regression – regularization and feature selection

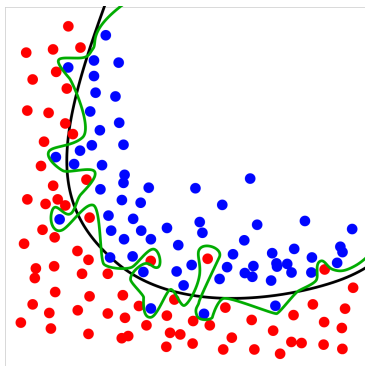
The story so far...

Failure of ERM

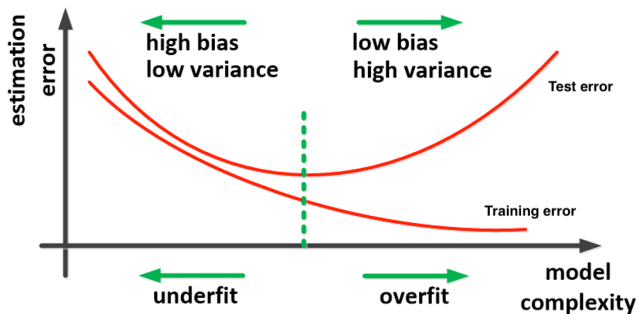
We hope that

$$R(\hat{h}_n) \approx R(h^*),$$

but in general, this is not true if the hypothesis space \mathcal{H} is too large



Underfitting/Overfitting



Regularization

If we want to learn well:

$$\text{dimension}(\mathcal{H}) \ll n$$

How do we get that?

- Model selection
- Feature selection
- Regularization

Regularization techniques

Explicit:

- Adding penalties into loss functions
- Dropout

Implicit:

- Design networks to capture invariance
- Data augmentation
- Use special optimizer
- Early stopping

Linear regression

Linear model

$$Y = \beta^{(0)} + \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)} + \epsilon$$

- p : number of variables ($X \in \mathbb{R}^p$)
- n : number of observations

Linear model

$$Y = \beta^{(0)} + \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)}$$

- $n \gg p$ (n much larger than p). With enough observations, we hope to be able to build a good model
- even if the true relationship between the variables is not linear, we can include transformations of variables

$$X^{(p+1)} = [X^{(1)}]^2, \quad X^{(p+2)} = X^{(1)}X^{(3)}, \dots$$

- adding transformed variables can increase p significantly

Trade-off: complexity vs. interpretability

Linear model

$$Y = \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)} + \epsilon$$

- Higher values of p lead to more complex model \rightarrow increases prediction power/accuracy
- Higher values of p make it more difficult to interpret the model: It is often the case that some or many of the variables regression model are in fact not associated with the response

Linear model

$$Y = \beta^{(0)} + \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)} + \epsilon$$

- it is often the case that $n \ll p$
- requires supplementary assumptions (e.g. sparsity)
- can still build good models with very few observations.

Linear regression by least squares

Settings

- $Y \in \mathbb{R}^{n \times 1}$, $X \in \mathbb{R}^{n \times (p+1)}$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} 1 & | & | & \dots & | \\ \dots & x^{(1)} & x^{(2)} & \dots & x^{(p)} \\ 1 & | & | & \dots & | \end{bmatrix}$$

where $x^{(1)}, x^{(2)}, \dots, x^{(p)} \in \mathbb{R}^{n \times 1}$ are the observations of $X^{(1)}, X^{(2)}, \dots, X^{(p)}$.

- We want

$$Y = \beta^{(0)} + \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)}$$

- We want

$$Y = \beta^{(0)} + \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)}$$

- Equivalent to

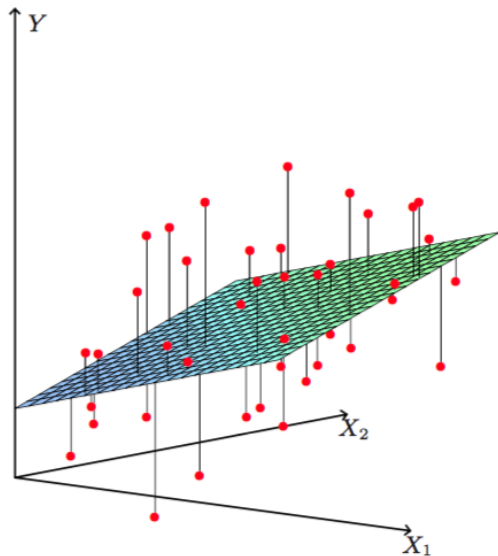
$$Y = X\beta, \quad \beta = \begin{bmatrix} \beta^{(0)} \\ \beta^{(1)} \\ \dots \\ \beta^{(p)} \end{bmatrix}$$

$$Y = X\beta$$

- In general, the system has no solution ($n \gg p$) or infinitely many solutions ($n \ll p$)
- The most popular estimation method is least squares, in which we pick the coefficients to minimize the residual sum of squares

$$RSS(\beta) = \sum_{i=1}^n (y_i - f(x_i))^2$$

Least squares



- Minimize the residual sum of squares

$$RSS(\beta) = \sum_{i=1}^n (y_i - f(x_i))^2$$

- Or alternatively,

$$\hat{\beta} = \min_{\beta} \|Y - X\beta\|_2^2$$

- Minimize the residual sum of squares

$$\begin{aligned}RSS(\beta) &= \sum_{i=1}^n (y_i - f(x_i))^2 \\ &= \sum_{i=1}^n \left(y_i - \beta^{(0)} - \beta^{(1)} x_i^{(1)} - \beta^{(2)} x_i^{(2)} - \dots - \beta^{(p)} x_i^{(p)} \right)^2\end{aligned}$$

- Taking derivative

$$\frac{\partial RSS}{\partial \beta^{(j)}} = \sum_{i=1}^n 2(y_i - x_i \beta) x_i^{(j)} = 2[x^{(j)}]^T (Y - X\beta)$$

Least squares

- Set derivatives to zero

$$X^T(Y - X\beta) = 0$$

- If $X^T X$ is invertible

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

- Predicted values

$$\hat{Y} = X_{test} \hat{\beta} = X_{test} (X_{train}^T X_{train})^{-1} X_{train}^T Y_{train}$$

The coefficient of determination

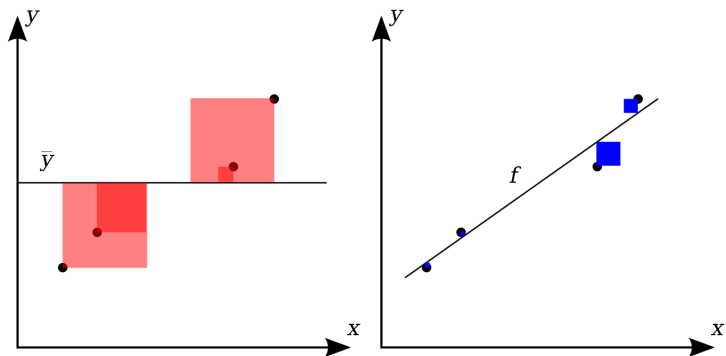
- The coefficient of determination, called “R squared” and denoted by

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

where \bar{y} is the average of y_1, \dots, y_n

- Often used to measure the quality of a linear model
- A model with a R^2 close to 1 fit the data well.

The coefficient of determination



In some sense, the R^2 measures how much better is the prediction compared to a constant prediction

The adjusted coefficient of multiple determination

- It is desirable to adjust R^2 to take account of the fact that its value may be quite high just because many predictors were used relative to the amount of data
- The adjusted coefficient of multiple determination

$$R_a^2 = 1 - \frac{\frac{1}{n-p-1} \sum_{i=1}^n (y_i - \hat{y}_i)^2}{\frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2}$$

where \bar{y} is the average of y_1, \dots, y_n

sklearn.linear_model.LinearRegression

```
>>> import numpy as np
>>> from sklearn.linear_model import LinearRegression
>>> X = np.array([[1, 1], [1, 2], [2, 2], [2, 3]])
>>> #  $y = 1 * x_0 + 2 * x_1 + 3$ 
>>> y = np.dot(X, np.array([1, 2])) + 3
>>> reg = LinearRegression().fit(X, y)
>>> reg.score(X, y)
1.0
>>> reg.coef_
array([1., 2.])
>>> reg.intercept_
3.0000...
>>> reg.predict(np.array([[3, 5]]))
array([16.])
```

```
>>> X = np.arange(6).reshape(3, 2)
>>> X
array([[0, 1],
       [2, 3],
       [4, 5]])
>>> poly = PolynomialFeatures(2)
>>> poly.fit_transform(X)
array([[ 1.,  0.,  1.,  0.,  0.,  1.],
       [ 1.,  2.,  3.,  4.,  6.,  9.],
       [ 1.,  4.,  5., 16., 20., 25.]])
>>> poly = PolynomialFeatures(interaction_only=True)
>>> poly.fit_transform(X)
array([[ 1.,  0.,  1.,  0.],
       [ 1.,  2.,  3.,  6.],
       [ 1.,  4.,  5., 20.]])
```


Questions

- Is at least one of the predictors X_1, X_2, \dots, X_p useful in predicting the response?
- Do all the predictors help to explain Y , or is only a subset of the predictors useful?

Subset selection

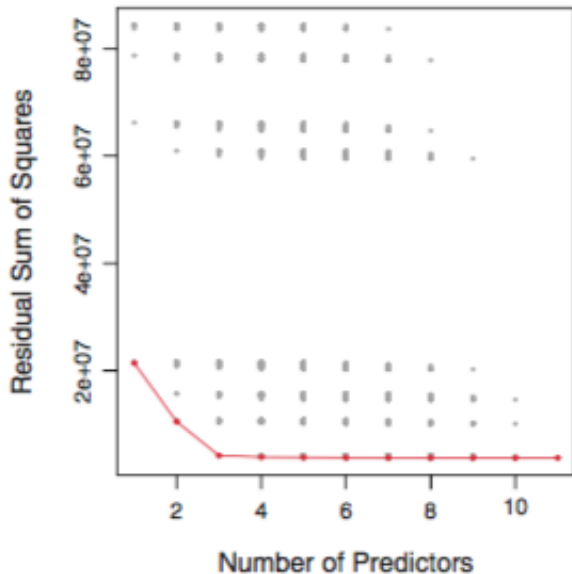
Trade-off: complexity vs. interpretability

Linear model

$$Y = \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)} + \epsilon$$

- Higher values of p lead to more complex model \rightarrow increases prediction power/accuracy
- Higher values of p make it more difficult to interpret the model
- Ideally, we would like to try out a lot of different models, each containing a different subset of the predictors, then select the best model
- Problem: there are 2^p models that contain subsets of p variables

Best subset selection



Forward stepwise selection

Algorithm 6.2 *Forward stepwise selection*

1. Let \mathcal{M}_0 denote the *null* model, which contains no predictors.
 2. For $k = 0, \dots, p - 1$:
 - (a) Consider all $p - k$ models that augment the predictors in \mathcal{M}_k with one additional predictor.
 - (b) Choose the *best* among these $p - k$ models, and call it \mathcal{M}_{k+1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Backward stepwise selection

Algorithm 6.3 *Backward stepwise selection*

1. Let \mathcal{M}_p denote the *full* model, which contains all p predictors.
 2. For $k = p, p - 1, \dots, 1$:
 - (a) Consider all k models that contain all but one of the predictors in \mathcal{M}_k , for a total of $k - 1$ predictors.
 - (b) Choose the *best* among these k models, and call it \mathcal{M}_{k-1} . Here *best* is defined as having smallest RSS or highest R^2 .
 3. Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

Hybrid approach

- Hybrid versions of forward and backward stepwise selection are available
- variables are added to the model sequentially
- after adding each new variable, the method may also remove any variables that no longer provide an improvement in the model fit

Adjusted training errors

- Adjusted R^2
- Mallows's C_p

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

where $\hat{\sigma}^2$ is an estimate of the variance of the error, d is the number of predictors

- AIC (Akaike information criterion)

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

- BIC (Bayesian information criterion)

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

sklearn.feature_selection.RFE

```
class sklearn.feature_selection.RFE (estimator, n_features_to_select=None, step=1, verbose=0) \[source\]
```

Feature ranking with recursive feature elimination.

Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), the goal of recursive feature elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a `coef_` attribute or through a `feature_importances_` attribute. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

Shrinkage methods

Settings

$$Y \in \mathbb{R}^{n \times 1}, \quad X \in \mathbb{R}^{n \times (p+1)}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} 1 & | & | & \dots & | \\ \dots & x^{(1)} & x^{(2)} & \dots & x^{(p)} \\ 1 & | & | & \dots & | \end{bmatrix}$$

Linear model: settings

- Linear model

$$Y = \beta^{(0)} + \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \dots + \beta^{(p)}X^{(p)} + \epsilon$$

- Equivalent to

$$\mathbf{Y} = \mathbf{X}\beta, \quad \beta = \begin{bmatrix} \beta^{(0)} \\ \beta^{(1)} \\ \dots \\ \beta^{(p)} \end{bmatrix}$$

- Least squares regression

$$\hat{\beta}^{LS} = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2$$

- ℓ_0 regularization

$$\hat{\beta}^0 = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{i=1}^p \mathbf{1}_{\beta^{(i)} \neq 0}$$

where $\lambda > 0$ is a parameter

- pay a fixed price λ for including a given variable into the model
- variables that do not significantly contribute to reducing the error are excluded from the model (i.e., $\beta_i = 0$)
- problem: difficult to solve (combinatorial optimization). Cannot be solved efficiently for a large number of variables.

ℓ_2 (Tikhonov) regularization

- Ridge regression/ Tikhonov regularization

$$\hat{\beta}^{RIDGE} = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^p [\beta^{(j)}]^2$$

where $\lambda > 0$ is a parameter

- shrinks the coefficients by imposing a penalty on their size
- penalty is a smooth function.
- easy to solve (solution can be written in closed form)
- can be used to regularize a rank deficient problem ($n < p$)

ℓ_2 (Tikhonov) regularization

$$\frac{\partial (\|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|^2)}{\partial \beta} = 2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\beta) + 2\lambda\beta$$

- The critical point satisfies

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\beta = \mathbf{X}^T\mathbf{Y}$$

- Note: $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})$ is positive definite, and thus invertible
- Thus

$$\hat{\beta}^{RIDGE} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$$

ℓ_2 (Tikhonov) regularization

$$\hat{\beta}^{RIDGE} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{Y}$$

- When $\lambda > 0$, the estimator is defined even when $n < p$
- When $\lambda = 0$ and $n > p$, we recover the usual least squares solution

The Lasso

- The Lasso (Least Absolute Shrinkage and Selection Operator)

$$\hat{\beta}^{lasso} = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^p |\beta^{(j)}|$$

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero
- However, the ℓ_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when λ is sufficiently large
- the lasso performs variable selection \rightarrow models are easier to interpret

Lasso: alternative form

Alternative form of lasso (using the Lagrangian and min-max argument)

$$\begin{aligned} & \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 \\ & \text{subject to } \sum_{j=1}^p |\beta^{(j)}| \leq s \end{aligned}$$

Lasso: alternative form

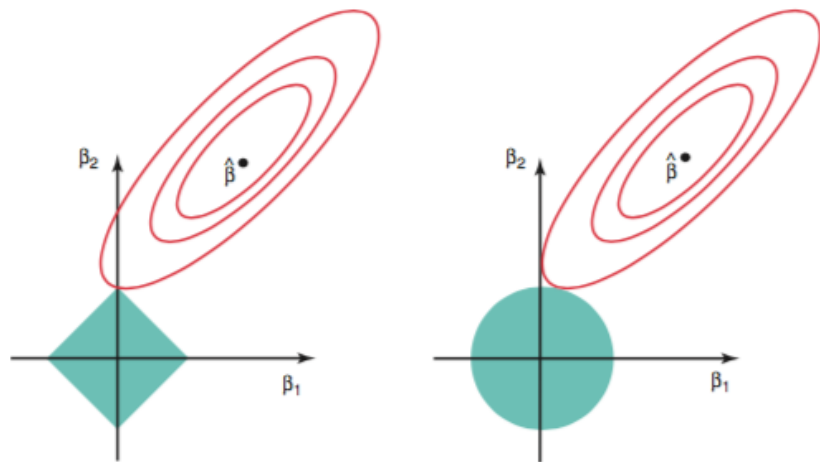


FIGURE 6.7. Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

- The Lasso:

$$\hat{\beta}^{lasso} = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^p |\beta^{(j)}|$$

- More “global” approach to selecting variables compared to previously discussed greedy approaches
- Can be seen as a convex relaxation of the $\hat{\beta}^0$ problem
- No closed form solution, but can be solved efficiently using convex optimization methods.
- Performs well in practice
- Very popular. Active area of research

Other shrinkage methods

- ℓ_q regularization ($q \geq 0$):

$$\hat{\beta} = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^p [\beta^{(j)}]^q$$

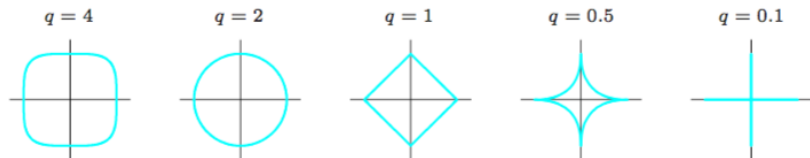


FIGURE 3.12. Contours of constant value of $\sum_j |\beta_j|^q$ for given values of q .

Other shrinkage methods

- Elastic net

$$\lambda \sum_{j=1}^p \alpha [\beta^{(j)}]^2 + (1 - \alpha) |\beta^{(j)}|$$

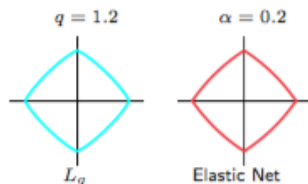


FIGURE 3.13. Contours of constant value of $\sum_j |\beta_j|^q$ for $q = 1.2$ (left plot), and the elastic-net penalty $\sum_j (\alpha \beta_j^2 + (1 - \alpha) |\beta_j|)$ for $\alpha = 0.2$ (right plot). Although visually very similar, the elastic-net has sharp (non-differentiable) corners, while the $q = 1.2$ penalty does not.

Lasso: alternative form

Alternative form of lasso (using the Lagrangian and min-max argument)

$$\begin{aligned} & \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 \\ & \text{subject to } \sum_{j=1}^p |\beta^{(j)}| \leq s \end{aligned}$$

Regularization

- ℓ_0 regularization

$$\hat{\beta}^0 = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{i=1}^p \mathbf{1}_{\beta^{(i)} \neq 0}$$

- Ridge regression/Tikhonov regularization

$$\hat{\beta}^{RIDGE} = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^p [\beta^{(j)}]^2$$

- Lasso

$$\hat{\beta}^{lasso} = \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda \sum_{j=1}^p |\beta^{(j)}|$$

Choosing parameters: cross-validation

- ℓ_0 , ridge, lasso have regularization parameters λ
- We obtain a family of estimators as we vary the parameter(s)
- optimal parameters needs to be chosen in a principled way
- cross-validation is a popular approach for rigorously choosing parameters.

Lasso: model selection consistency

Lasso: alternative form

Alternative form of lasso (using the Lagrangian and min-max argument)

$$\begin{aligned} & \min_{\beta} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 \\ & \text{subject to } \sum_{j=1}^p |\beta^{(j)}| \leq s \end{aligned}$$

Lasso: alternative form

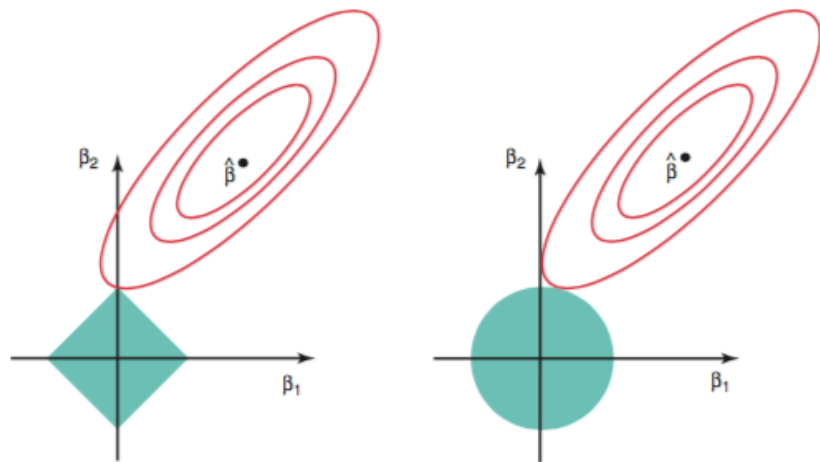
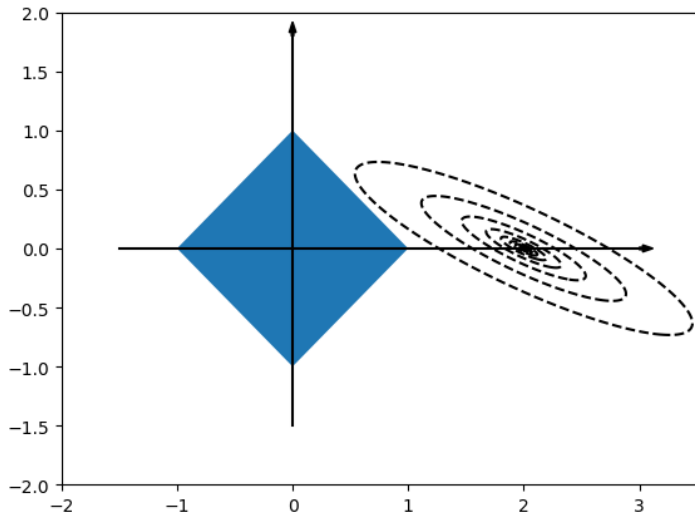
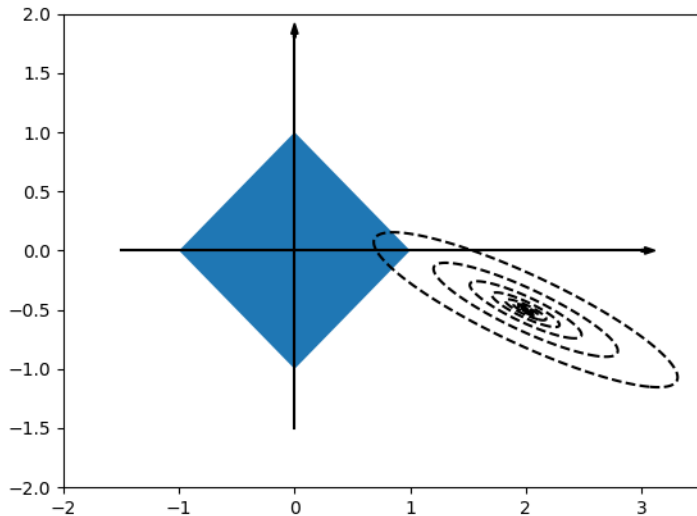


FIGURE 6.7. Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

When the lasso fails



When the lasso fails



Model selection consistency lasso

- Note: Model consistency of lasso
- Further readings:
 - Zhao and Yu (2006)
 - Wainright (2009)
 - Sparsity, the lasso, and friends (Ryan Tibshirani)

- We start with the simple linear regression problem

$$Y = \beta^{(1)}X^{(1)} + \beta^{(2)}X^{(2)} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

- Sparsity: assume that the data is generated using the “true” vector of parameters $\beta^* = (\beta^{*(1)}, 0)$.
- We assume that $E[X^{(1)}] = E[X^{(2)}] = 0$.

Matrix form

- we observe a dataset $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- use the same notations as in the previous lectures

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} \\ \dots & \dots \\ x_n^{(1)} & x_n^{(2)} \end{bmatrix}$$

The lasso estimator solves the optimization problem

$$\hat{\beta} = \min_{\beta} \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda(|\beta^{(1)}| + |\beta^{(2)}|).$$

We want to investigate the conditions under which we can verify that

$$\text{sign}(\hat{\beta}^{(1)}) = \text{sign}(\beta^{*(1)}) \quad \text{and} \quad \hat{\beta}^{(2)} = 0$$

Sub-gradient

Issue: the penalty of lasso is non-differentiable

Definition

We say that a vector $s \in \mathbb{R}^p$ is a subgradient for the ℓ_1 -norm evaluated at $\beta \in \mathbb{R}^p$, written as $s \in \partial\|\beta\|$ if for $i = 1, \dots, p$ we have

$$s^{(i)} = \text{sign}(\beta^{(i)}) \quad \text{if } \beta^{(i)} \neq 0 \quad \text{and } s_i \in [-1, 1] \quad \text{otherwise.}$$

Theorem

- (a) A vector $\hat{\beta}$ solve the lasso program if and only if there exists a $\hat{z} \in \partial \|\hat{\beta}\|$ such that

$$\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) - \lambda\hat{z} = 0 \quad (0.1)$$

- (b) Suppose that the subgradient vector satisfies the strict dual feasibility condition

$$|\hat{z}^{(2)}| < 1$$

then **any** lasso solution $\tilde{\beta}$ satisfies $\tilde{\beta}^{(2)} = 0$.

- (c) Under the condition of part (b), if $\mathbf{X}^{(1)} \neq 0$, then $\hat{\beta}$ is the unique lasso solution.

The primal-dual witness method.

The primal-dual witness (PDW) method consists of constructing a pair of $(\tilde{\beta}, \tilde{z})$ according to the following steps:

- First, we obtain $\tilde{\beta}^{(1)}$ by solving the restricted lasso problem

$$\tilde{\beta}^{(1)} = \min_{\beta=(\beta^{(1)},0)} \frac{1}{2} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2 + \lambda(|\beta^{(1)}|).$$

Choose a subgradient $\tilde{z}^{(1)} \in \mathbb{R}$ for the ℓ_1 -norm evaluated at $\tilde{\beta}^{(1)}$

- Second, we solve for a vector $\tilde{z}^{(2)}$ satisfying equation (0.1), and check whether or not the dual feasibility condition $|\tilde{z}^{(2)}| < 1$ is satisfied
- Third, we check whether the *sign consistency condition*

$$\tilde{z}^{(1)} = \text{sign}(\beta^{*(1)})$$

is satisfied.

- This procedure is not a practical method for solving the ℓ_1 -regularized optimization problem, since solving the restricted problem in Step 1 requires knowledge about the sparsity of β^*
- Rather, the utility of this constructive procedure is as a proof technique: it succeeds if and only if the lasso has a optimal solution with the correct signed support.

A more detailed computation

We note that the matrix form of equation (0.1) can be written as

$$[\mathbf{X}^{(1)}]^T (\mathbf{Y} - \mathbf{X}^{(1)}\beta^{(1)} - \mathbf{X}^{(2)}\beta^{(2)}) - \lambda\hat{z}^{(1)} = 0$$

$$[\mathbf{X}^{(2)}]^T (\mathbf{Y} - \mathbf{X}^{(1)}\beta^{(1)} - \mathbf{X}^{(2)}\beta^{(2)}) - \lambda\hat{z}^{(2)} = 0$$

To simplify the notation, we denote

$$C_{ij} = [\mathbf{X}^{(i)}]^T [\mathbf{X}^{(j)}]$$

Step 1

- we find $\tilde{\beta}^{(1)}$ and $\tilde{z}^{(1)}$ that satisfies

$$[\mathbf{X}^{(1)}]^T(\mathbf{Y} - \mathbf{X}^{(1)}\tilde{\beta}^{(1)}) - \lambda\tilde{z}^{(1)} = 0$$

- Moreover, to make sure that the sign consistency in Step 3 is satisfied, we impose that

$$\tilde{z}^{(1)} = \text{sign}(\beta^{*(1)}) \quad \text{and} \quad \tilde{\beta}^{(1)} = C_{11}^{-1}([\mathbf{X}^{(1)}]^T\mathbf{Y} - \lambda\text{sign}(\beta^{*(1)})).$$

This is acceptable as long as $\tilde{z}^{(1)} \in \partial|\tilde{\beta}^{(1)}|$. That is,

$$\text{sign}(\tilde{\beta}^{(1)}) = \text{sign}(\beta^{*(1)})$$

Step 2

- Step 2:

$$[\mathbf{X}^{(2)}]^T (\mathbf{Y} - \mathbf{X}^{(1)} \tilde{\beta}^{(1)}) - \lambda \tilde{z}^{(2)} = 0$$

- Choose

$$\tilde{z}^{(2)} = \frac{1}{\lambda} [\mathbf{X}^{(2)}]^T (\mathbf{Y} - \mathbf{X}^{(1)} \tilde{\beta}^{(1)}).$$

We want $|\tilde{z}^{(2)}| < 1$.

In principle, we want two conditions:

- $\text{sign}(\beta^{*(1)}) = \text{sign}(\beta^{*(1)} + \Delta)$

where

$$\Delta = C_{11}^{-1}([\mathbf{X}^{(1)}]^T \epsilon - \lambda \text{sign}(\beta^{*(1)}))$$

- $|\tilde{z}^{(2)}| < 1$ where

$$\tilde{z}^{(2)} = \frac{1}{\lambda} [\mathbf{X}^{(2)}]^T (\mathbf{X}^{(1)} \Delta + \epsilon)$$

Zero-noise setting

- we assume that the observations are collected with no noise ($\epsilon = 0$).
- Then

$$\Delta = -C_{11}^{-1} \lambda \text{sign}(\beta^{*(1)})$$

and

$$\tilde{z}^{(2)} = \frac{-1}{\lambda} C_{21} \Delta = C_{21} C_{11}^{-1} \text{sign}(\beta^{*(1)})$$

- Conditions
 - Mutual incoherence: $|C_{21} C_{11}^{-1}| < 1$.
 - Minimum signal: $|\beta^{*(1)}| > \lambda C_{11}^{-1}$

- Mutual incoherence: $|C_{21}C_{11}^{-1}| < 1$.
- Recall that

$$C_{12} = [\mathbf{X}^{(1)}]^T [\mathbf{X}^{(2)}] = \sum_i x_i^{(1)} x_i^{(2)}$$

- When n is large

$$\frac{1}{n} C_{12} \rightarrow E \left([X^{(1)}]^T [X^{(2)}] \right) = \text{Cov}(X^{(1)}, X^{(2)})$$

since $E[X^{(1)}] = E[X^{(2)}] = 0$.

Conditions

- Mutual incoherence: $|C_{21} C_{11}^{-1}| < 1$.

The condition roughly means that the covariance between the variables $X^{(1)}$ and $X^{(2)}$ are less than the variance of $X^{(1)}$

- Minimum signal: $|\beta^{*(1)}| > \lambda C_{11}^{-1}$

Since

$$\frac{1}{n} C_{11} \rightarrow \text{Var}(X^{(1)}),$$

this means that when $n \rightarrow \infty$, we need

$$\frac{\lambda_n}{n} \rightarrow 0$$

In principle, we want two conditions:

- $\text{sign}(\beta^{*(1)}) = \text{sign}(\beta^{*(1)} + \Delta)$

where

$$\Delta = C_{11}^{-1}([\mathbf{X}^{(1)}]^T \epsilon - \lambda \text{sign}(\beta^{*(1)}))$$

- $|\tilde{z}^{(2)}| < 1$ where

$$\tilde{z}^{(2)} = \frac{1}{\lambda} [\mathbf{X}^{(2)}]^T (\mathbf{X}^{(1)} \Delta + \epsilon)$$

- We want an upper bound on

$$[\mathbf{X}^{(1)}]^T \epsilon \quad \text{and} \quad [\mathbf{X}^{(2)}]^T \epsilon$$

Properties of Gaussian random variables

In principle, we want two conditions:

- $[\mathbf{X}^{(1)}]^T \epsilon$ is a Gaussian random variable with mean 0 and standard deviation $\sigma \|\mathbf{X}^{(1)}\|_2$
- Thus, there exists a universal constant C such that

$$|[\mathbf{X}^{(1)}]^T \epsilon| \leq C\sigma \sqrt{n \text{Var}(X^{(1)}) \log\left(\frac{1}{\delta}\right)}$$

with probability at least $1 - \delta$

General settings

Without loss of generality, assume $\beta^n = (\beta_1^n, \dots, \beta_q^n, \beta_{q+1}^n, \dots, \beta_p^n)^T$ where $\beta_j^n \neq 0$ for $j = 1, \dots, q$ and $\beta_j^n = 0$ for $j = q + 1, \dots, p$. Let $\beta_{(1)}^n = (\beta_1^n, \dots, \beta_q^n)^T$ and $\beta_{(2)}^n = (\beta_{q+1}^n, \dots, \beta_p^n)$. Now write $\mathbf{X}_n(1)$ and $\mathbf{X}_n(2)$ as the first q and last $p - q$ columns of \mathbf{X}_n respectively and let $C^n = \frac{1}{n} \mathbf{X}_n^T \mathbf{X}_n$. By setting $C_{11}^n = \frac{1}{n} \mathbf{X}_n(1)^T \mathbf{X}_n(1)$, $C_{22}^n = \frac{1}{n} \mathbf{X}_n(2)^T \mathbf{X}_n(2)$, $C_{12}^n = \frac{1}{n} \mathbf{X}_n(1)^T \mathbf{X}_n(2)$ and $C_{21}^n = \frac{1}{n} \mathbf{X}_n(2)^T \mathbf{X}_n(1)$. C^n can then be expressed in a block-wise form as follows:

$$C^n = \begin{pmatrix} C_{11}^n & C_{12}^n \\ C_{21}^n & C_{22}^n \end{pmatrix}.$$

Assuming C_{11}^n is invertible, we define the following Irrepresentable Conditions
Strong Irrepresentable Condition. There exists a positive constant vector η

$$|C_{21}^n (C_{11}^n)^{-1} \text{sign}(\beta_{(1)}^n)| \leq \mathbf{1} - \eta,$$

where $\mathbf{1}$ is a $p - q$ by 1 vector of 1's and the inequality holds element-wise.

Weak Irrepresentable Condition.

$$|C_{21}^n (C_{11}^n)^{-1} \text{sign}(\beta_{(1)}^n)| < \mathbf{1},$$