MATH 637: Mathematical Techniques in Data Science
Subset selection and Coefficients Penalization

Dominique Guillot

Departments of Mathematical Sciences
University of Delaware

February 26, 2020

Subset selection

- We saw before that the OLS is the *best linear unbiased estimator* for β.

Subset selection

- We saw before that the OLS is the *best linear unbiased estimator* for β.
- However, biased estimators can significantly improve the performance (e.g. reduce prediction error).

Subset selection

- We saw before that the OLS is the *best linear unbiased estimator* for β.
- However, biased estimators can significantly improve the performance (e.g. reduce prediction error).

We now explore various approaches that can be used to select an appropriate subset of variables in a linear regression.

Subset selection

- We saw before that the OLS is the *best linear unbiased estimator* for β.
- However, biased estimators can significantly improve the performance (e.g. reduce prediction error).

We now explore various approaches that can be used to select an appropriate subset of variables in a linear regression.

Best subset selection: Given $k \in \{1, \dots, p\}$, we find the subset of size k of $\{1, \dots, p\}$ that minimizes the prediction error.

Subset selection

- We saw before that the OLS is the *best linear unbiased estimator* for β.
- However, biased estimators can significantly improve the performance (e.g. reduce prediction error).

We now explore various approaches that can be used to select an appropriate subset of variables in a linear regression.

Best subset selection: Given $k \in \{1, \dots, p\}$, we find the subset of size k of $\{1, \dots, p\}$ that minimizes the prediction error.

- Note: there are $\binom{p}{k}$ subsets of size k and 2^k possible subsets. So the procedure is only computationally feasible for small values of p.

Subset selection

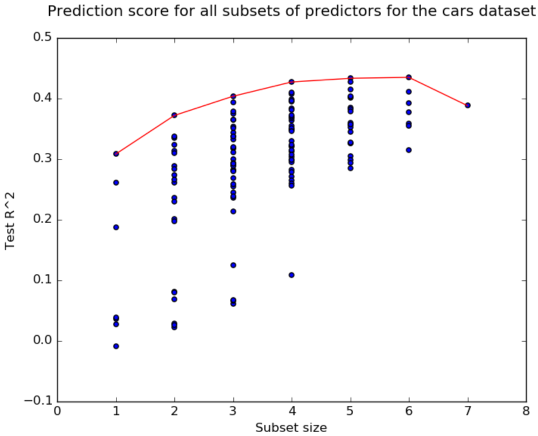
- We saw before that the OLS is the *best linear unbiased estimator* for β.
- However, biased estimators can significantly improve the performance (e.g. reduce prediction error).

We now explore various approaches that can be used to select an appropriate subset of variables in a linear regression.

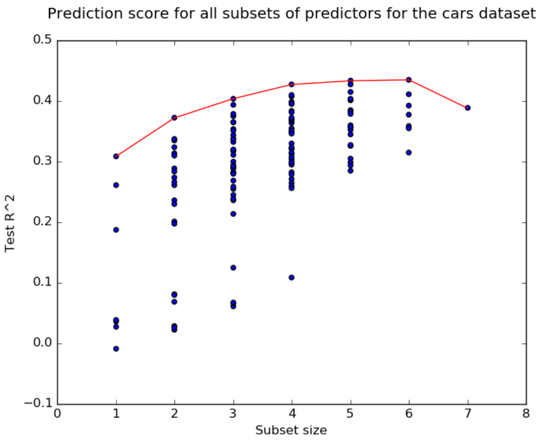
Best subset selection: Given $k \in \{1, \ldots, p\}$, we find the subset of size k of $\{1, \ldots, p\}$ that minimizes the prediction error.

- Note: there are $\binom{p}{k}$ subsets of size k and 2^k possible subsets. So the procedure is only computationally feasible for small values of p.
- The leaps and bounds procedure (Furnival and Wilson, 1974) makes this feasible for p as large as 30 or 40.

Best subset selection: cars dataset



Best subset selection: cars dataset

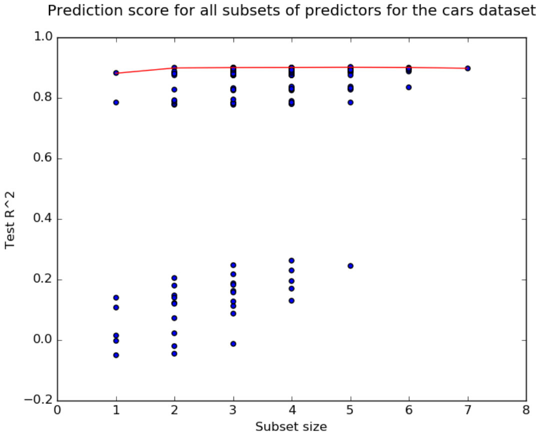


Best subset = ['Mileage','Liter','Doors','Cruise','Sound', 'Leather'].
Not included = ['Cylinder']

Best subset of 4 elements: ['Mileage','Liter','Cruise','Leather']

Best subset selection: cars dataset, Chevrolet

Restricting to Chevrolet only:



Forward- and Backward- stepwise regression

- Best subset selection performs well, but is too computationally intensive to be useful in practice.

Forward- and Backward- stepwise regression

- Best subset selection performs well, but is too computationally intensive to be useful in practice.

Two natural "greedy" variants of the best subset selection technique:

- **Forward stepwise regression:** starts with the intercept $\overline{y}$, and then sequentially adds into the model the predictor that most improves the fit.

Forward- and Backward- stepwise regression

- Best subset selection performs well, but is too computationally intensive to be useful in practice.

Two natural "greedy" variants of the best subset selection technique:

- **Forward stepwise regression:** starts with the intercept $\overline{y}$, and then sequentially adds into the model the predictor that most improves the fit.
- **Backward stepwise regression:** starts with the full model, and sequentially deletes the predictor that has the least impact on the fit (smallest Z-score or t-score).

Forward- and Backward- stepwise regression

- Best subset selection performs well, but is too computationally intensive to be useful in practice.

Two natural "greedy" variants of the best subset selection technique:

- **Forward stepwise regression:** starts with the intercept $\overline{y}$, and then sequentially adds into the model the predictor that most improves the fit.
- **Backward stepwise regression:** starts with the full model, and sequentially deletes the predictor that has the least impact on the fit (smallest Z-score or t-score).

Can be used even when the number of variables is very large.
However,

Forward- and Backward- stepwise regression

- Best subset selection performs well, but is too computationally intensive to be useful in practice.

Two natural "greedy" variants of the best subset selection technique:

- **Forward stepwise regression:** starts with the intercept $\bar{y}$, and then sequentially adds into the model the predictor that most improves the fit.
- **Backward stepwise regression:** starts with the full model, and sequentially deletes the predictor that has the least impact on the fit (smallest Z-score or t-score).

Can be used even when the number of variables is very large. However,

- Greedy approach: doesn't guarantee a global optimum.
- Less rigorous than other methods, less supporting theory.

Forward- and Backward- stepwise regression

- Best subset selection performs well, but is too computationally intensive to be useful in practice.

Two natural "greedy" variants of the best subset selection technique:

- **Forward stepwise regression:** starts with the intercept $\overline{y}$, and then sequentially adds into the model the predictor that most improves the fit.
- **Backward stepwise regression:** starts with the full model, and sequentially deletes the predictor that has the least impact on the fit (smallest Z-score or t-score).

Can be used even when the number of variables is very large. However,

- Greedy approach: doesn't guarantee a global optimum.
- Less rigorous than other methods, less supporting theory.

Nevertheless, the stepwise approaches often return predictors similar to the predictors obtained from more complex methods with better theory.

Shrinkage methods

Penalizing the coefficients:

- Suppose we want to restrict the number or the size of the regression coefficients.
- Add a penalty (or "price to pay") for including a nonzero coefficient.

Penalizing the coefficients:

- Suppose we want to restrict the number or the size of the regression coefficients.
- Add a penalty (or "price to pay") for including a nonzero coefficient.

Examples: Let $\lambda > 0$ be a parameter.

1. $$\hat{\beta}^0 = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^p \mathbf{1}_{\beta_i \neq 0} \right).$$

Penalizing the coefficients:

- Suppose we want to restrict the number or the size of the regression coefficients.
- Add a penalty (or "price to pay") for including a nonzero coefficient.

Examples: Let $\lambda > 0$ be a parameter.

1. $$\hat{\beta}^0 = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \mathbf{1}_{\beta_i \neq 0} \right).$$

    - Pay a fixed price λ for including a given variable into the model.

Shrinkage methods

Penalizing the coefficients:

- Suppose we want to restrict the number or the size of the regression coefficients.
- Add a penalty (or "price to pay") for including a nonzero coefficient.

Examples: Let $\lambda > 0$ be a parameter.

1.

$$\hat{\beta}^0 = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \mathbf{1}_{\beta_i \neq 0} \right).$$

   - 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., $\beta_i = 0$).

Shrinkage methods

Penalizing the coefficients:

- Suppose we want to restrict the number or the size of the regression coefficients.
- Add a penalty (or "price to pay") for including a nonzero coefficient.

Examples: Let $\lambda > 0$ be a parameter.

1. $$\hat{\beta}^0 = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \mathbf{1}_{\beta_i \neq 0} \right).$$

    - 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.

Shrinkage methods (cont.)

Relaxations of the previous approach:

2. Ridge regression/Tikhonov regularization:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \beta_i^2 \right).$$

Shrinkage methods (cont.)

Relaxations of the previous approach:

2. Ridge regression/Tikhonov regularization:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \beta_i^2 \right).$$

- Shrinks the regression coefficients by imposing a penalty on their size.
- $\text{Penalty} = \lambda \cdot \|\beta\|_2^2$.
- Problem equivalent to $\hat{\beta}^{\text{ridge}} = \operatorname{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 \text{ subject to } \sum_{i=1}^{p} \beta_i^2 \leq t$.
- Penalty is a smooth function.
- Easy to solve (solution can be written in closed form).
- Generally does not set any coefficient to zero (no model selection).
- Can be used to "regularize" a rank deficient problem ($n < p$).

Ridge regression: closed form solution

We have

$$\begin{aligned}\frac{\partial}{\partial\beta}\left(\|y - X\beta\|_2^2 + \lambda\sum_{i=1}^{p}\beta_i^2\right) &= 2(X^TX\beta - X^Ty) + 2\lambda\sum_{i=1}^{p}\beta_i \\ &= 2\left((X^TX + \lambda I)\beta - X^Ty\right).\end{aligned}$$

Ridge regression: closed form solution

We have

$$\frac{\partial}{\partial \beta}\left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \beta_i^2\right) = 2(X^T X\beta - X^T y) + 2\lambda \sum_{i=1}^{p} \beta_i$$
$$= 2\left((X^T X + \lambda I)\beta - X^T y\right).$$

Therefore, the critical points satisfy

$$(X^T X + \lambda I)\beta = X^T y.$$

Ridge regression: closed form solution

We have

$$\frac{\partial}{\partial \beta}\left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \beta_i^2\right) = 2(X^T X\beta - X^T y) + 2\lambda \sum_{i=1}^{p} \beta_i$$
$$= 2\left((X^T X + \lambda I)\beta - X^T y\right).$$

Therefore, the critical points satisfy

$$(X^T X + \lambda I)\beta = X^T y.$$

Note: $(X^T X + \lambda I)$ is positive definite, and therefore invertible.

Ridge regression: closed form solution

We have

$$\frac{\partial}{\partial \beta}\left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \beta_i^2\right) = 2(X^T X\beta - X^T y) + 2\lambda \sum_{i=1}^{p} \beta_i$$
$$= 2\left((X^T X + \lambda I)\beta - X^T y\right).$$

Therefore, the critical points satisfy

$$(X^T X + \lambda I)\beta = X^T y.$$

Note: $(X^T X + \lambda I)$ is positive definite, and therefore invertible.

Therefore, the system has a **unique** solution. Can check using the Hessian that the solution is a minimum. Thus,

$$\beta^{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y.$$

Ridge regression: closed form solution

We have

$$\frac{\partial}{\partial \beta}\left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} \beta_i^2\right) = 2(X^T X\beta - X^T y) + 2\lambda \sum_{i=1}^{p} \beta_i$$
$$= 2\left((X^T X + \lambda I)\beta - X^T y\right).$$

Therefore, the critical points satisfy

$$(X^T X + \lambda I)\beta = X^T y.$$

Note: $(X^T X + \lambda I)$ is positive definite, and therefore invertible.

Therefore, the system has a **unique** solution. Can check using the Hessian that the solution is a minimum. Thus,

$$\beta^{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y.$$

Remarks:

- When $\lambda > 0$, the estimator is defined even when $n < p$.

Ridge regression: closed form solution

We have

$$\frac{\partial}{\partial \beta}\left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^p \beta_i^2\right) = 2(X^T X\beta - X^T y) + 2\lambda \sum_{i=1}^p \beta_i$$
$$= 2\left((X^T X + \lambda I)\beta - X^T y\right).$$

Therefore, the critical points satisfy

$$(X^T X + \lambda I)\beta = X^T y.$$

Note: $(X^T X + \lambda I)$ is positive definite, and therefore invertible.

Therefore, the system has a **unique** solution. Can check using the Hessian that the solution is a minimum. Thus,

$$\beta^{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y.$$

Remarks:

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

Ridge regression: closed form solution

We have

$$\frac{\partial}{\partial \beta}\left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^p \beta_i^2\right) = 2(X^T X\beta - X^T y) + 2\lambda \sum_{i=1}^p \beta_i$$
$$= 2\left((X^T X + \lambda I)\beta - X^T y\right).$$

Therefore, the critical points satisfy

$$(X^T X + \lambda I)\beta = X^T y.$$

Note: $(X^T X + \lambda I)$ is positive definite, and therefore invertible.

Therefore, the system has a **unique** solution. Can check using the Hessian that the solution is a minimum. Thus,

$$\beta^{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y.$$

Remarks:

- When $\lambda > 0$, the estimator is defined even when $n < p$.
- When $\lambda = 0$ and $n > p$, we recover the usual least squares solution.
- Makes rigorous "adding a multiple of the identity" to $X^T X$.

The Lasso

3. The Lasso (Least Absolute Shrinkage and Selection Operator):

$$\hat{\beta}^{\text{lasso}} = \underset{\beta \in \mathbb{R}^p}{\text{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} |\beta_i| \right).$$

3. The Lasso (Least Absolute Shrinkage and Selection Operator):

$$\hat{\beta}^{\text{lasso}} = \underset{\beta\in\mathbb{R}^p}{\text{argmin}} \left(\|y - X\beta\|_2^2 + \lambda \sum_{i=1}^{p} |\beta_i| \right).$$

- Introduced in 1996 by Robert Tibshirani.
- Equivalent to $\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2$ subject to $\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i| \leq t$.
- Both sets coefficients to zero (model selection) and shrinks coefficients.
- 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 solved efficiently using convex optimization methods.
- Performs well in practice.
- Very popular. Active area of research.

Important model selection property

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$

subject to $\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i| \leq t$

Important model selection property

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$
subject to $\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i| \leq t$

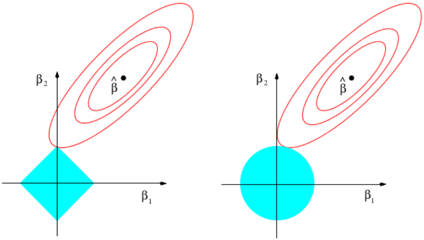


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are the contours of the least squares error function.*

ESL, Fig. 3.11.

Important model selection property

$$\hat{\beta}^{\text{lasso}} = \text{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2$$
subject to $\|\beta\|_1 = \sum_{i=1}^{p} |\beta_i| \leq t$

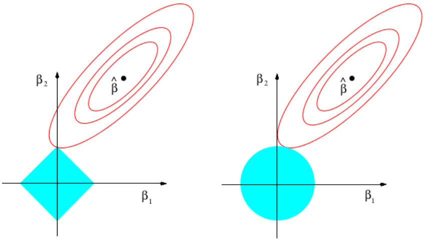


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

ESL, Fig. 3.11.

Solutions are the intersection of the ellipses with the $\|\cdot\|_1$ or $\|\cdot\|_2$ balls. Corners of the $\|\cdot\|_1$ have zero coefficients.

Elastic net

Elastic net (Zou and Hastie, 2005)

$$\hat{\beta}^{\text{e-net}} \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda_2 \|\beta\|_2^2 + \lambda_1 \|\beta\|_1.$$

Elastic net

Elastic net (Zou and Hastie, 2005)

$$\hat{\beta}^{\text{e-net}} \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda_2 \|\beta\|_2^2 + \lambda_1 \|\beta\|_1.$$

- Benefits from both ℓ_1 (model selection) and ℓ_2 regularization.

Elastic net

Elastic net (Zou and Hastie, 2005)

$$\hat{\beta}^{\text{e-net}} \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \|y - X\beta\|_2^2 + \lambda_2 \|\beta\|_2^2 + \lambda_1 \|\beta\|_1.$$

- Benefits from both ℓ_1 (model selection) and ℓ_2 regularization.
- Downside: Two parameters to choose instead of one (can increase the computational burden quite a lot in large experiments).

Scikit-learn has an object to compute Lasso solution.

Scikit-learn has an object to compute Lasso solution.

Note: the package solves a slightly different (but equivalent) problem than discussed above:

$$\underset{w \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - Xw\|_2^2 + \alpha \|w\|_1.$$

Scikit-learn has an object to compute Lasso solution.

Note: the package solves a slightly different (but equivalent) problem than discussed above:

$$\operatorname*{argmin}_{w \in \mathbb{R}^p} \frac{1}{2n} \|y - Xw\|_2^2 + \alpha \|w\|_1.$$

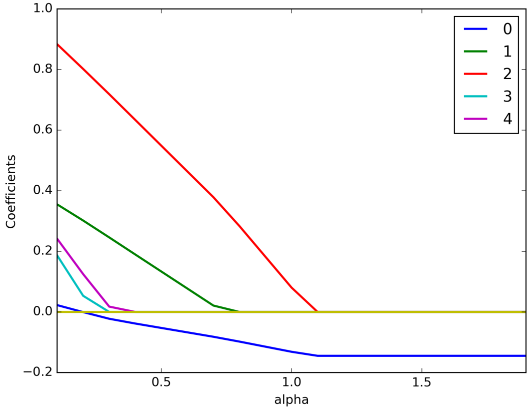
```
from sklearn.linear_model import Lasso

clf = linear_model.Lasso(alpha=0.1)
clf.fit(X,y)
print(clf.coef_)
print(clf.intercept_)
```

Lab (cont.)

A simple example with simulated data

```
import numpy as np
from sklearn.linear_model import Lasso
import matplotlib.pyplot as plt

# Generate random data
n = 100
p = 5

X = np.random.randn(n,p)
epsilon = np.random.randn(n,1)
beta = np.random.rand(p)
y = X.dot(beta) + epsilon

alphas = np.arange(0.1,2,0.1)  # 0.1 to 2, step = 0.1
N = len(alphas) # Number of lasso parameters

betas = np.zeros((N,p+1))  # p+1 because of intercept

for i in range(N):
    clf = Lasso(alphas[i])
    clf.fit(X,y)
    betas[i,0] = clf.intercept_
    betas[i,1:] = clf.coef_

plt.plot(alphas,betas,linewidth=2)
plt.legend(range(p))
plt.xlabel('alpha')
plt.ylabel('Coefficients')
plt.xlim(min(alphas),max(alphas))
plt.show()
```

Lab (cont.)



Now, let $y = X_1\beta_1 + X_2\beta_2 + \epsilon$ with $\epsilon \sim N(0, \sigma^2)$. Can the lasso detect that the first two variables are the most important?

```
sigma = 0.1
epsilon = sigma*np.random.randn(n)
y2 = X[:,0] + X[:,1] + epsilon
clf = Lasso(0.1)
clf.fit(X,y2)
np.where(abs(clf.coef_) > 1e-10)
```

- Vary the values of α between 0.1 and 2.
- Repeat the previous exercise for larger values of `sigma2`.

If you have time, use the lasso to identify relevant predictors in either the `cars` or the `boston` dataset.