Linear Regression

Yu Zheng, Jiaqi Hu

School of Management
University of Science and Technology of China

2021.10.20

Table of Contents

1. Introduction
2. Model Specification
3. Robust linear regression
4. Ridge Regression and PCA
5. Best Subset Selection
   - Traditional Subset Selection
   - Iterate Hard Thresholding
   - Splicing Algorithm
6. Appendix

Table of Contents

1. Introduction
2. Model Specification
3. Robust linear regression
4. Ridge Regression and PCA
5. Best Subset Selection
   - Traditional Subset Selection
   - Iterate Hard Thresholding
   - Splicing Algorithm
6. Appendix

Introduction

- Linear regression model assumes that the conditional distribution $E(Y \mid x)$ is linear for the input x.
- For prediction purposes it can sometimes outperform fancier nonlinear models, especially in situations with small numbers of training cases, low signal-to-noise ratio or sparse data.

Table of Contents

1. Introduction
2. Model Specification
3. Robust linear regression
4. Ridge Regression and PCA
5. Best Subset Selection
    - Traditional Subset Selection
    - Iterate Hard Thresholding
    - Splicing Algorithm
6. Appendix

Model Specification

- The most popular estimation method is least squares, in which we pick the coefficients $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$ to minimize the residual sum of squares of data $(y_i, x_{i,\cdot})_{i=1,\ldots,N}$

$$\begin{aligned}\mathrm{RSS}(\beta) &= \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \\ &= (y - X\beta)^T (y - X\beta).\end{aligned}$$

Let $x_{i,\cdot}$ be the ith row and $x_{\cdot,j}$ be jth the column of X.

Model Specification

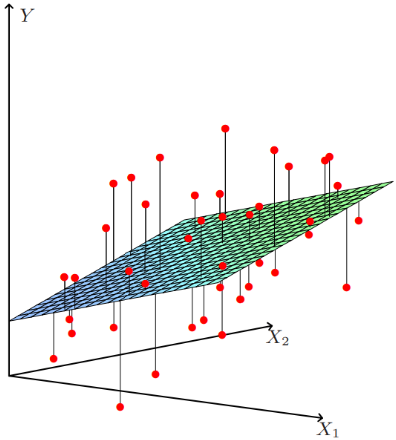


Figure 1: Linear least squares fitting with $x_{i,\cdot} \in \mathbb{R}^2$

Model Specification

- We minimize:

$$\mathrm{RSS}(\beta) = (y - X\beta)^T (y - X\beta).$$

- Differentiating with respect to β:
  $\frac{\partial \mathrm{RSS}}{\partial \beta} = -2X^T(y - X\beta)$, $\frac{\partial^2 \mathrm{RSS}}{\partial \beta \partial \beta^T} = 2X^T X$.
- we set the first derivative to zero
  $X^T(y - X\beta) = 0.$
- Assuming that X has full column rank, and hence $X^T X$ is positive definite,we obtain the unique solution
  $\hat{\beta} = \left(X^T X\right)^{-1} X^T y.$
- The fitted values at the training inputs are
  $\hat{y} = X\hat{\beta} = X\left(X^T X\right)^{-1} X^T y.$
- The matrix $H = X\left(X^T X\right)^{-1} X^T$ is called the "hat" matrix.

Geometric interpretation

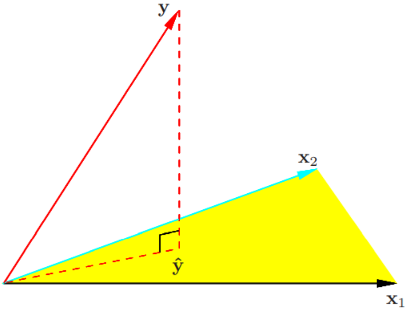


Figure 2: The N-dimensional geometry of least squares regression with two predictors.The outcome vector y is orthogonally projected onto the hyperplane spanned by the input vectors $x_{\cdot,1}$ and $x_{\cdot,2}$. The projection $\hat{y}$ represents the vector of the least squares predictions

Model Specification

- $(x_{\cdot,j})_{j=1,\ldots,p}$ span a subspace of $\mathbb{R}^N$, also referred to as the column space of X.
- We minimize $\mathrm{RSS}(\beta) = \|y - X\beta\|^2$ by choosing $\hat{\beta}$ so that the residual vector $y - \hat{y}$ is orthogonal to this subspace.
$$X^T(y - X\beta) = 0,$$
- $\hat{y} = Hy$ is hence the orthogonal projection of y onto this subspace. The hat matrix H computes the orthogonal projection, and hence it is also known as a projection matrix. H has the following properties:
$$H = X\left(X^TX\right)^{-1}X' = H^T,\ H = H^2.$$
$$\begin{aligned}\mathrm{tr}(H) = \mathrm{tr}\left(X\left(X^TX\right)^{-1}X^T\right) = \mathrm{tr}\left(\left(X^TX\right)^{-1}X^TX\right) = \mathrm{tr}\left(I_{p+1}\right) \\ = p+1.\end{aligned}$$

Model Specification

- It might happen that the columns of X are not linearly independent, so that X is not of full rank. Then X^TX is singular and the least squares coefficients $\hat{\beta}$ are not uniquely defined.
- The non-full-rank case occurs most often when one or more qualitative inputs are coded in a redundant fashion.
- There is usually a natural way to resolve the non-unique representation, by recoding and/or dropping redundant information in X^TX, especially, when the dimension p exceeds the number of training cases N.

The Gauss–Markov Theorem

We now assume that $x_{i,\cdot}$ are fixed (non random) with Gaussian random variable ε_i

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i.$$

- Exogeneity: $\mathrm{E}\varepsilon_i = 0$.
- Homoscedasticity: $\operatorname{Var}\varepsilon = \sigma^2$.
- We focus on estimation of any linear combination of the parameters $\theta = a^T\beta$, the least squares estimate of it is
$$\hat{\theta} = a^T\hat{\beta} = a^T\left(X^TX\right)^{-1}X^Ty.$$

The Gauss–Markov Theorem

- Considering X to be fixed, this is a linear function $c^T y$ of the response vector y. $a^T\hat{\beta}$ is unbiased since

$$
\begin{aligned}
\mathrm{E}\left(a^T\hat{\beta}\right) &= \mathrm{E}\left(a^T\left(X^TX\right)^{-1}X^Ty\right)\\
&= a^T\left(X^TX\right)^{-1}X^TX\beta\\
&= a^T\beta.
\end{aligned}
$$

- The Gauss–Markov theorem states that if we have any other linear estimator $\tilde{\theta} = c^T y$ that is unbiased for $a^T\beta$, then

$$
\mathrm{Var}\left(a^T\hat{\beta}\right) \leq \mathrm{Var}\left(c^T y\right).
$$

- The errors do not need to be normal, nor do they need to be independent and identically distributed. The requirement that the estimator be unbiased cannot be dropped, since biased estimators exist with lower variance.

Model Specification

We additionally assume that ε_i follows mean zero Gaussian distribution:

- It is easy to show that
$$\hat{\beta} \sim N\left(\beta, \left(X^T X\right)^{-1} \sigma^2\right).$$
- Typically, one estimates the variance σ^2 by
$$\hat{\sigma}^2 = \tfrac{1}{N-p-1} \textstyle\sum_{i=1}^{N} \left(Y_i - \hat{Y}_i\right)^2.$$
The $N-p-1$ rather than N in the denominator makes $\hat{\sigma}^2$ an unbiased estimate of σ^2: $\mathrm{E}\left(\hat{\sigma}^2\right) = \sigma^2$.
- Also
$$(N-p-1)\hat{\sigma}^2 \sim \sigma^2 \chi^2_{N-p-1},$$
a chi-squared distribution with $N-p-1$ degrees of freedom.
- In addition, $\hat{\beta}$ and $\hat{\sigma}^2$ are statistically independent.

Test whether $\beta_j = 0$

$$\hat{\beta} \sim N\left(\beta, \sigma^2\left(X^T X\right)^{-1}\right),$$

$$\hat{\beta}_j \sim N\left(\beta_j, c_{jj}\sigma^2\right),$$

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\sigma} \sim N(0,1),$$

$$\frac{SSE}{\sigma^2} \sim \chi^2(n-p-1),$$

$$\hat{\sigma} = \sqrt{\frac{SSE}{n-p-1}},$$

$$z_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\hat{\sigma}} \sim t(n-p-1).$$

where c_{jj} is jth diagonal element of $\left(X^T X\right)^{-1}$.

Test whether $\beta_j = 0$

- A large (absolute) value of z_j will lead to rejection of this null hypothesis.
- If $\hat{\sigma}$ is replaced by a known value σ, then z_j would have a standard normal distribution.
- The difference between the tail quantiles of a t-distribution and a standard normal become negligible as the sample size increases, and so we typically use the normal quantiles.

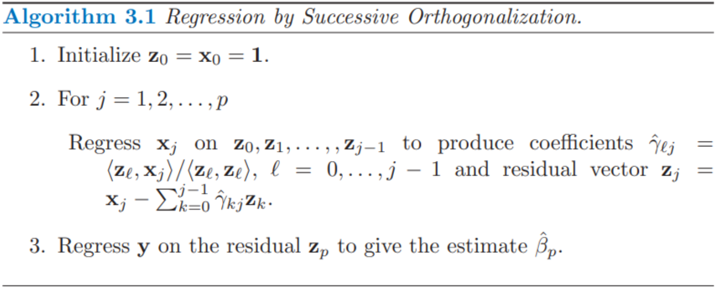
Regression by Successive Orthogonalization

- Define
$$\langle \mathbf{x}, y\rangle = x^T y.$$
- Then in the univariate regression analysis,
$$\hat{\beta} = \tfrac{\langle \mathbf{x}, y\rangle}{\langle \mathbf{x}, x\rangle},\ r = y - x^T\hat{\beta}.$$
- Suppose that the inputs $x_{\cdot,1}, x_{\cdot,2}, \dots,\ x_{\cdot,p} \in \mathbb{R}^N$ are orthogonal; that is $\langle x, y\rangle = 0$ for all $j \neq k$. Then it is easy to check that the multiple least squares estimates β_j are equal to $\langle x_{\cdot,j}, y\rangle\langle x_{\cdot,j}, x_{\cdot,j}\rangle$.
- In other words, when the inputs are orthogonal, they have no effect on each other's parameter estimates in the model.

Regression by Successive Orthogonalization

- Suppose next that we have an intercept and a single input x. Then the least squares coefficient of x has the form
$$\hat{\beta}_1 = \frac{\langle \mathbf{x} - \bar{x}1, y \rangle}{\langle \mathbf{x} - \bar{x}1, x - \bar{x}1 \rangle}.$$
- We can view the estimate it as the result of two applications of the simple regression. The steps are:
  1. Regress x on $\mathbf{1}$ to produce the residual $z = x - \bar{x}1$;
  2. Regress y on the residual z to give the coefficient $\hat{\beta}_1$.
- This recipe generalizes to the case of p inputs, as shown in the following Algorithm.

Regression by Successive Orthogonalization



Algorithm 3.1 *Regression by Successive Orthogonalization.*

```
1. Initialize $\mathbf{z}_0 = \mathbf{x}_0 = \mathbf{1}$.
2. For $j = 1, 2, \ldots, p$
      Regress $\mathbf{x}_j$ on $\mathbf{z}_0, \mathbf{z}_1, \ldots, , \mathbf{z}_{j-1}$ to produce coefficients $\hat{\gamma}_{\ell j} = \langle \mathbf{z}_\ell, \mathbf{x}_j \rangle / \langle \mathbf{z}_\ell, \mathbf{z}_\ell \rangle$, $\ell = 0, \ldots, j-1$ and residual vector $\mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \mathbf{z}_k$.
3. Regress $\mathbf{y}$ on the residual $\mathbf{z}_p$ to give the estimate $\hat{\beta}_p$.
```

The result of this algorithm is

$$\hat{\beta}_p = \frac{\langle \mathbf{z}_p, \mathbf{y} \rangle}{\langle \mathbf{z}_p, \mathbf{z}_p \rangle}.$$

Regression by Successive Orthogonalization

- The orthogonalization does not change the subspace spanned by $\{x_{\cdot,j}\}_{j=1,\dots,p}$, it simply produces an orthogonal basis for representing it.
- The coefficient is indeed the multiple regression coefficient of y on x_p
- Note also that by rearranging the $x_{\cdot,j}$, any one of them could be in the last position, and a similar results holds.

Regression by Successive Orthogonalization

- If $x_{\cdot,p}$ is highly correlated with some of the other $x_{\cdot,k}$'s, the residual vector z_p will be close to zero,and the coefficient $\hat{\beta}_p$ will be very unstable, since

$$\mathrm{Var}\left(\hat{\beta}_p\right) = \frac{\sigma^2}{\langle \mathbf{z}_p, z_p \rangle} = \frac{\sigma^2}{\|z_p\|^2}.$$

- The precision with which we can estimate $\hat{\beta}_p$ depends on the length of the residual vector z_p;this represents how much of $x_{\cdot,p}$ is unexplained by the other $x_{\cdot,k}$'s.

Regression by Successive Orthogonalization

We can represent step 2 of Algorithm 3.1 in matrix form:

$$X = Z\mathbf{\Gamma},$$

where Z has as columns the z_j (in order), and $\mathbf{\Gamma}$ is the upper triangular matrix with entries $\hat{\gamma}_{kj}$. Introducing the diagonal matrix D with jth diagonal entry $D_{jj} = \|z_j\|$, we get

$$\begin{aligned} X &= ZD^{-1}D\mathbf{\Gamma} \\ &= QR, \end{aligned}$$

the so-called QR decomposition of X. Here Q is an $N \times (p+1)$ orthogonal, $Q^T Q = I$, and R is a $(p+1) \times (p+1)$ upper triangular matrix.

Regression by Successive Orthogonalization

The QR decomposition represents a convenient orthogonal basis for the column space of X. It is easy to see, for example, that the least squares solution is given by

$$\hat{\beta} = R^{-1} Q^T y, \; \hat{y} = Q Q^T y.$$

The first equation is easy to solve because R is upper triangular.

Table of Contents

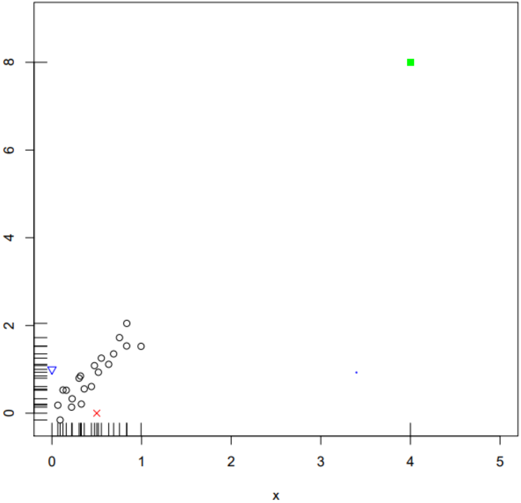
1. Introduction
2. Model Specification
3. Robust linear regression
4. Ridge Regression and PCA
5. Best Subset Selection
    - Traditional Subset Selection
    - Iterate Hard Thresholding
    - Splicing Algorithm
6. Appendix

Robust linear regression

When we are doing regression modeling, in fact, we don't really care about whether some data point is far from the rest of the data, but whether it breaks a pattern the rest of the data seems to follow.

- We'll first try to build some intuition for when outliers cause trouble in linear regression models.
- Then we'll look at some ways of quantifying how much influence particular data points have on the model.
- Take a brief look at the robust regression strategy, of replacing least squares estimates with others which are less easily influenced.

Robust linear regression



Influence of Individual Data Points on Estimates

Recall that our least-squares coefficient estimator is

$$\widehat{\beta} = \left(X^T X\right)^{-1} X^T y,$$
$$\widehat{y} = X\widehat{\beta} = X\left(X^T X\right)^{-1} X^T y = Hy.$$

This leads to a very natural sense in which one observation might be more or less influential than another:

$$\frac{\partial \hat{\beta}_k}{\partial y_i} = \left(\left(X^T X\right)^{-1} X^T\right)_{ki}$$
$$\frac{\partial \hat{y}_i}{\partial y_i} = H_{ii}.$$

H_{ii} is the influence of y_i on its own fitted value. This turns out to be a key quantity in looking for outliers, so we'll give it a special name, the leverage.

Influence of Individual Data Points on Estimates

- The leverage of a data point just depends on the value of the predictors there; it increases as the point moves away from the mean of the predictors.

The residuals depend only on the hat matrix:

$$e = y - \widehat{y} = (I - H)y.$$

We have

$$\begin{gathered} \operatorname{Var}[e] = \sigma^2(I-H)(I-H)^T = \sigma^2(I-H), \\ \operatorname{Var}\left[e_i\right] = \sigma^2(I-H)_{ii} = \sigma^2\left(1-H_{ii}\right) \end{gathered}$$

In summary, the bigger the leverage of ith predictor, the smaller the variance of the residual there. This is yet another sense in which points with high leverage are points which the model tries very hard to fit.

Influence of Individual Data Points on Estimates

If there are substantial variations in leverage across the data points, it's better to scale the residuals by their expected size.

- The standardized or studentized residuals:
$$r_i \equiv \frac{e_i}{\hat{\sigma}\sqrt{1-H_{ii}}}.$$
- In particular, the studentized residuals should look flat, with constant variance.

Influence of Individual Data Points on Estimates

Suppose we left out the i th data point altogether. How much would that change the model?

- Cook's Distance, defined as the sum of all the changes in the regression model when ith observation is removed from it:

$$D_i = \frac{\sum_{m=1}^{N}\left(\widehat{y}_m - \widehat{y}_m^{(-i)}\right)^2}{(p+1)\hat{\sigma}^2},$$

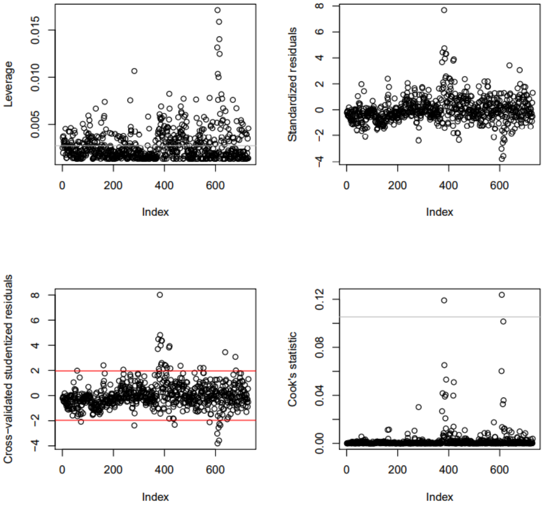
  where $\widehat{y}_m^{(-i)}$ is the fitted response value obtained when excluding i.

- Noting

$$\frac{\widehat{y}_m - \widehat{y}_m^{(-i)}}{y_i - \widehat{y}_i^{(-i)}} = H_{mi},$$

  so that $D_i = \frac{1}{(p+1)\hat{\sigma}^2} e_i^2 \frac{H_{ii}}{(1-H_{ii})^2}$, the total influence of a point over all the fitted values grows with both its leverage and the size of its residual when it is included.

Practically, and with R



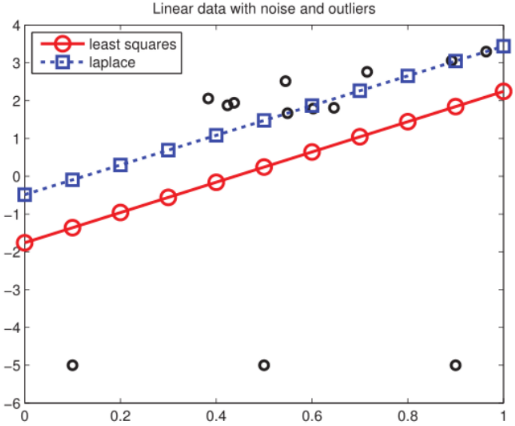
Robust linear regression

We might, therefore, consider using not a different statistical model, but a different method of estimating its parameters.

- we estimate
$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \rho\left(y_i - x_{i,\cdot}^T \beta\right).$$
- Different choices of ρ, the loss function, yield different estimators.

Robust linear regression

If we have outliers in our data, this can result in a poor fit.



This is because squared error penalizes deviations quadratically, so points far from the line have more affect on the fit than points near to the line.

Robust linear regression

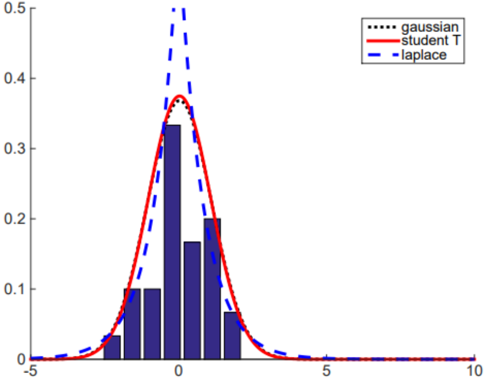


Figure 3: No outliers (the Gaussian and Student curves are on top of each other)

Robust linear regression

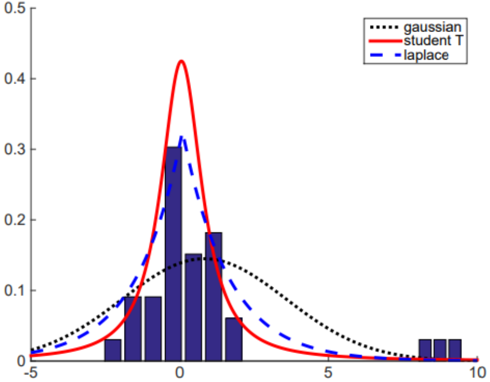


Figure 4: With outliers.We see that the Gaussian is more affected by outliers than the Student and Laplace distributions.

Robust linear regression

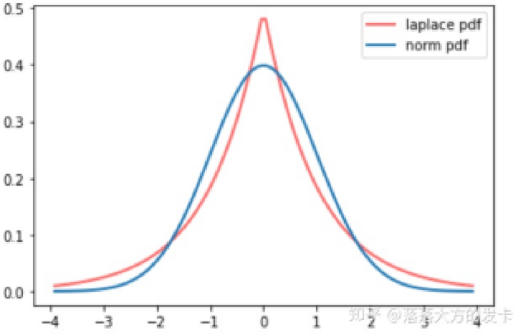
One way to achieve robustness to outliers is to replace the Gaussian distribution for the response variable with a distribution that has heavy tails. Such a distribution will assign higher likelihood to outliers, without having to perturb the straight line to “explain” them.
Gaussian distribution:

$$p(Y \mid X, \beta) = \mathcal{N}\left(X\beta, \sigma^2 I_p\right),$$

Laplace distribution:

$$\begin{aligned} p(Y \mid X, \beta, b) &= \mathrm{Lap}\left(Y \mid \beta^T X, b\right) \\ &\propto \exp\left(-\frac{1}{b}|Y - X\beta|\right) \end{aligned}$$

Robust linear regression

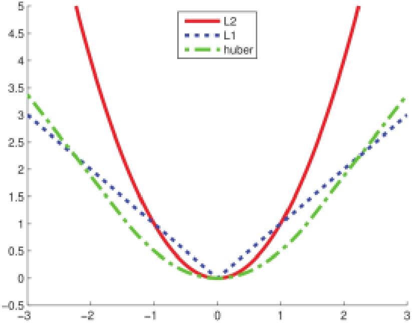


The robustness arises from the use of $|Y - X\beta|$ instead of $||Y - X\beta||$. For simplicity, we will assume b is fixed. Let $r_i \triangleq y_i - \beta^T x_i$ be the i'th residual. The NLL (negative log likelihood) has the form $\ell(\beta) = \sum_i |r_i(\beta)|$.

Robust linear regression

An alternative to using NLL under a Laplace likelihood is to minimize the Huber loss function (Huber 1964), defined as follows:

$$L_H(r,\delta) = \begin{cases} r^2/2 & \text{if } |r| \le \delta \\ \delta|r| - \delta^2/2 & \text{if } |r| > \delta \end{cases}$$



Robust linear regression

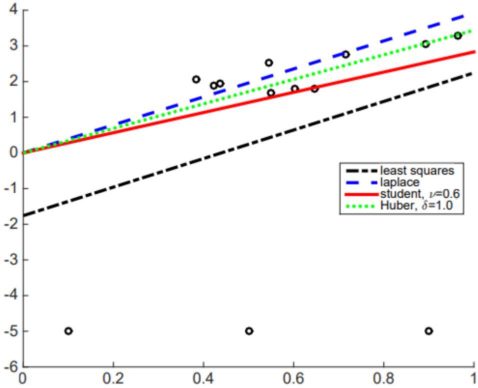


Figure 5: Illustration of robust linear regression

Table of Contents

1. Introduction

2. Model Specification

3. Robust linear regression

4. Ridge Regression and PCA

5. Best Subset Selection
    - Traditional Subset Selection
    - Iterate Hard Thresholding
    - Splicing Algorithm

6. Appendix

Ridge Regression

- Recall that the ordinary least squares estimate is defined by

$$\hat{\beta} = \left(X^T X\right)^{-1} X^T y,$$

  when X is of column full rank.
- In practice, we often encounter highly correlated covariates, which is known as the collinearity issue.
- The covariance matrix of the OLS estimate is $\sigma^2 \left(X^T X\right)^{-1}$. As a result, the collinearity issue makes Var$(\hat{\beta})$ large.

Ridge Regression

- Hoerl and Kennard (1970) introduced the ridge regression estimator as follows:

$$\widehat{\beta}_{\lambda} = \left(X^T X + \lambda I\right)^{-1} X^T y,$$

  where $\lambda > 0$ is a regularization parameter.
- Ridge regression reduces to OLS by setting $\lambda = 0$.
- Ridge regression is always well defined even when X is not full rank (smaller condition number).

Bias-variance tradeoff

- Under the assumption $\mathrm{Var}(\varepsilon) = \sigma^2 I$,it is easy to show that
$$\mathrm{Var}\left(\widehat{\beta}_\lambda\right) = \left(X^T X + \lambda I\right)^{-1} X^T X \left(X^T X + \lambda I\right)^{-1} \sigma^2.$$
- We always have $\mathrm{Var}\left(\widehat{\beta}_\lambda\right) < \left(X^T X\right)^{-1} \sigma^2.$
- The ridge regression estimator reduces the estimation variance by paying a price in estimation bias:
$$\mathrm{E}\left(\widehat{\beta}_\lambda\right) - \beta = \left(X^T X + \lambda I\right)^{-1} X^T X \beta - \beta = -\lambda \left(X^T X + \lambda I\right)^{-1} \beta.$$

MSE of Ridge Regression

- The overall estimation accuracy is gauged by the mean squared error(MSE), For $\widehat{\beta}_\lambda$ its MSE is given by

$$\begin{aligned} \text{MSE}\left(\widehat{\beta}_\lambda\right) =& \mathrm{E}\left(\left\|\widehat{\beta}_\lambda - \beta\right\|^2\right) \\ =& \operatorname{tr}\left(\left(X^T X + \lambda I\right)^{-1} X^T X \left(X^T X + \lambda I\right)^{-1} \sigma^2\right) + \\ & \lambda^2 \beta^T \left(X^T X + \lambda I\right)^{-2} \beta \\ =& \operatorname{tr}\left(\left(X^T X + \lambda I\right)^{-2} \left[\lambda^2 \beta \beta^T + \sigma^2 X^T X\right]\right) \end{aligned}$$

- It can be shown that $\left.\frac{d\,\text{MSE}(\widehat{\beta}_\lambda)}{d\lambda}\right|_{\lambda=0} < 0$,which implies that there are some proper λ values by which ridge regression improves OLS.

ℓ_2 penalized least squares

- Ridge regression shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}.$$

- An equivalent way to write the ridge problem is

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2, \text{ subject to}$$

$$\sum_{j=1}^{p} \beta_j^2 \leq t.$$

ℓ_2 penalized least squares

Writing in matrix form,

$$\mathrm{RSS}(\lambda) = (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta.$$

The regression solution is

$$\hat{\beta}^{\text{ridge}} = \left(X^TX + \lambda I\right)^{-1} X^Ty.$$

- The above discussion shows that ridge regression is equivalent to the ℓ_2 penalized least-squares.
- In the case of orthonormal inputs, the ridge estimates are just a scaled version of the least squares estimates, that is $\hat{\beta}^{\text{ridge}} = \hat{\beta}/(1+\lambda)$.

PCA as a Variance-Maximization Technique

We use the SVD of the centered matix X to express the principal components of the variables in X. The sample covariance matrix is given by

$$S = X^T X/(N-1).$$

And the SVD of S is

$$S = X^T X/(N-1) = VD^2 V^T/(N-1).$$

The first eigenvectors v_j has the property that: $z_1 = Xv_1$ has the largerst variance:

$$\mathrm{Var}\left(z_{\cdot,1}\right) = \mathrm{Var}\left(Xv_{\cdot,1}\right) = \tfrac{d_1^2}{N-1}.$$

In fact, $z_{\cdot,1} = Xv_{\cdot,1} = u_{\cdot,1}d_1$, hence $u_{\cdot,1}$ is the nomalized first PC.

The geometric interpretation of principal components

- Principal components are a sequence of projections of the data, mutually uncorrelated and ordered in variance.
- Fit a model to the data by least squares amounts to minimizing the reconstruction error

  $$\min_{\mu,\{\lambda_i\},v_{pq}} \sum_{i=1}^{N} \left\| x_{i,\cdot} - \mu - V_{pq}\lambda_i \right\|^2 ,$$

  with $V_{pq}^T V_{pq} = I_q$. We obtain $\hat{\mu} = \bar{x}$, $\hat{\lambda}_{i,\cdot} = V_{pq}^T (x_{i,\cdot} - \bar{x})$.

The geometric interpretation of principal components

- This leaves us to find the orthogonal matrix V_{pq}

$$\min_{V_{pq}} \sum_{i=1}^{N} \left\| (x_{i,\cdot} - \bar{x}) - V_{pq} V_{pq}^{T} (x_{i,\cdot} - \bar{x}) \right\|^2 .$$

- For convenience we assume that $\bar{x} = 0$.
- The $p \times p$ matrix $H_p = V_{pq} V_{pq}^{T}$ is a projection matrix, and maps each point $x_{i,\cdot}$ onto its rank-q reconstruction $H_p x_{i,\cdot}$, the orthogonal projection of $x_{i,\cdot}$ onto the subspace spanned by the columns of V_{pq}.

The geometric interpretation of principal components

Remark

For each rank q, the solution V_{pq} consists of the first q eigenvectors of S.

$$\begin{aligned}
&\operatorname{argmin}_{V_{pq}} \sum_{i=1}^{N} \left\| (x_{i,\cdot} - \bar{x}) - V_{pq} V_{pq}^{T} (x_{i,\cdot} - \bar{x}) \right\|^2 \\
=&\operatorname{argmin}_{V_{pq}} \sum_{i=1}^{N} \left\| \left(I_p - V_{pq} V_{pq}^{T} \right) (x_{i,\cdot} - \bar{x}) \right\|^2 \\
=&\operatorname{argmin}_{V_{pq}} \operatorname{tr} \left(\left(I_p - V_{pq} V_{pq}^{T} \right) S \right) \\
=&\operatorname{argmin}_{V_{pq}} \operatorname{tr} \left(S - V_{pq} V_{pq}^{T} S \right) \\
=&\operatorname{argmax}_{V_{pq}} \operatorname{tr} \left(V_{pq}^{T} S V_{pq} \right).
\end{aligned}$$

The geometric interpretation of principal components

- The one-dimensional principal component line in $\mathbb{R}^2$ is illustrated as follow

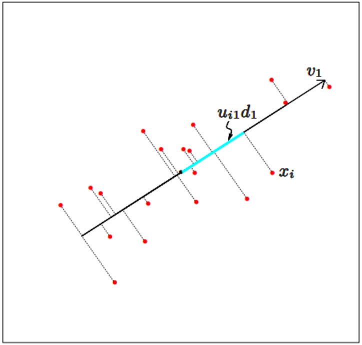


Figure 6: The first linear principal component of a set of data. The line minimizes the total squared distance from each point to its orthogonal projection onto the line.

The geometric interpretation of principal components

Recall $z_{\cdot,1} = Xv_{\cdot,1} = d_1 u_{\cdot,1}$, then

- For each data point $x_{i,\cdot}$, there is a closest point on the line, given by $u_{i1} d_1 v_{\cdot,1}$.
- Here $v_{\cdot,1}$ is the direction of the line and $\hat{\lambda}_{i1} = u_{i1} d_1$ measures distance along the line from the origin.

The geometric interpretation of principal components

- Figure below shows the two-dimensional principal component surface fit to the half-sphere data.

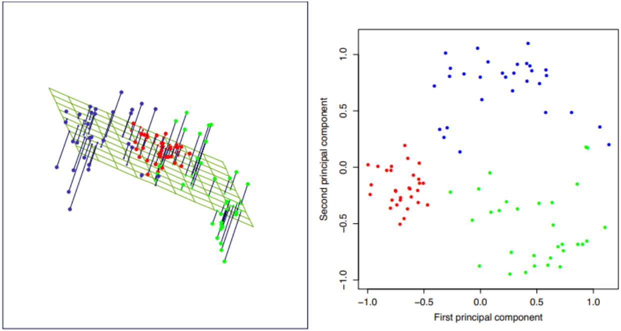


Figure 7: The best rank-two linear approximation to the half-sphere data.The right panel shows the projected points with coordinates.

Connection between Ridge Regression and PCA

The singular value decomposition (SVD) of the centered input matrix X is

$$X = U_{n\times p} D_{p\times p} V_{p\times p}^T$$

Here U and V are orthogonal matrices, and D is a diagonal matrix, with diagonal entries $d_1 \geq d_2 \geq \cdots d_p \geq 0$. Using the SVD of X we have:

$$\begin{aligned} X\hat{\beta}^{ls} &= X\left(X^T X\right)^{-1} X^T y \\ &= UU^T y = \sum_{j=1}^{n} u_{\cdot,j} u_{\cdot,j}^T y \\ X\hat{\beta}^{\text{ridge}} &= X\left(X^T X + \lambda I\right)^{-1} X^T y \\ &= UD\left(D^2 + \lambda I\right)^{-1} DU^T y \\ &= \sum_{j=1}^{p} u_{\cdot,j} \frac{d_j^2}{d_j^2 + \lambda} u_{\cdot,j}^T y \end{aligned}$$

Ridge Regression

Note that for $\lambda \geq 0$, we have $\frac{d_j^2}{d_j^2+\lambda} \leq 1$

- Like linear regression, ridge regression computes the coordinates of y w.r.t the orthonormal basis U.
- It shrinks these coordinates by the factor $\frac{d_j^2}{d_j^2+\lambda}$.
- A greater amount of shrinkage is applied to the coordinates of basis vectors with smaller d_j^2.
- We define the effective degrees of freedom as follows:

  $$\mathrm{df}(\lambda) = \textstyle\sum_{j=1}^{D} \frac{d_j^2}{d_j^2+\lambda}$$

  Note that $\mathrm{df}(\lambda) = p$ when $\lambda = 0$(no regularization); and $\mathrm{df}(\lambda) \to 0$ as $\lambda \to \infty$.

PCR

- Principal component regression forms the derived input columns $z_{\cdot,m} = Xv_{\cdot,m}$, and then regresses y on $z_{\cdot,1}, z_{\cdot,2}, ..., z_{\cdot,M}$ for some $M < p$.
- Since the $z_{\cdot,m}$ are orthogonal,this regression is just a sum of univariate regressions:

  $$\hat{y}^{pcr}_{(M)} = \bar{y}1 + \sum_{m=1}^{M} \hat{\theta}_m z_{\cdot,m},$$

  where $\hat{\theta}_m = \langle z_{\cdot,m}, y\rangle \,/\, \langle z_{\cdot,m}, z_{\cdot,m}\rangle$.
- We can express the solution in terms of coefficients as:

  $$\hat{\beta}^{\mathrm{pcr}}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_{\cdot,m}.$$
- The solution of PCR with k principle components:

  $$X\hat{\beta}^{\mathrm{pcr}} = \sum_{j=1}^{k} u_{\cdot,j} u_{\cdot,j}^T y.$$

PCR

- Note that if $M = p$, we would just get back the usual least squares estimates. For $M < p$, we get a reduced regression.
- We see that principal components regression is very similar to ridge regression:

  (a) Both operate via the SVD of the input matrix.

  (b) Both reduce the variance of the coefficents.

$$\begin{aligned}
\mathrm{Var}\left(\widehat{\beta}_{pcr}\right) &= \sigma^2 V_M \left(Z_M^T Z_M\right)^{-1} V_M^T \\
&= \sigma^2 V_M \,\mathrm{diag}\left(d_1^{-1}, \ldots, d_M^{-1}\right) V_M^T \\
&= \sigma^2 \sum_{j=1}^{M} \frac{v_{\cdot,j} v_{\cdot,j}^T}{d_j},
\end{aligned}$$

$$\mathrm{Var}\left(\widehat{\beta}_{\mathrm{ols}}\right) - \mathrm{Var}\left(\widehat{\beta}_{pcr}\right) = \sigma^2 \sum_{j=M+1}^{p} \frac{\mathrm{v}_{\cdot,j} v_{\cdot,j}^T}{d_j}$$

Connection with PCA

- Ridge regression shrinks the regression coefficients of the principal components by using shrinkage factors $d_j^2/\left(d_j^2+\lambda\right)$. Ridge regression shrinks the coefficients of the low variance components more than high ones since lower ones contains less information.
- In this way, ill-determined parameters are reduced in size towards 0. This is called *shrinkage*.
- PCR discards the $p - M$ smallest eigenvalue components.

Table of Contents

1. Introduction
2. Model Specification
3. Robust linear regression
4. Ridge Regression and PCA
5. Best Subset Selection
   - Traditional Subset Selection
   - Iterate Hard Thresholding
   - Splicing Algorithm
6. Appendix

Introduction of Best Subset Selection

- Consider a linear regression problem

$$y = X\beta + \varepsilon$$

  y is a n-dimensional vector, X is a $n \times p$ design matrix, β is a p-dimensional vector and ε is a n-dimensional random error.

- In high dimensional case, β is assumed to be sparse with an unknown sparsity level $s = ||\beta||_0$, where $||\beta||_0 = \sum_{i=1}^{p} I(\beta_i \neq 0)$.

Best Subset Selection

The best subset selection problem

- Constrained form:

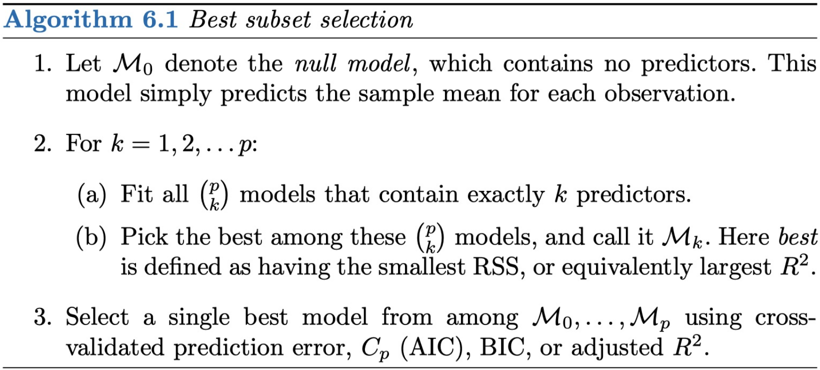
$$\min_{\beta\in\mathbb{R}^p} L(\beta) = \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\beta\|_2^2, \text{ s.t. } \|\boldsymbol{\beta}\|_0 \le s$$

- Lagrangian form:

$$\min_{\beta\in\mathbb{R}^p} F(\beta) = \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\beta\|_2^2 + \lambda\|\boldsymbol{\beta}\|_0$$

where λ is a tuning parameter.

Best Subset Selection



Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.

2. For $k = 1, 2, \ldots p$:

   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.

   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.

3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

- In general, there are 2^p models that involve subsets of p variables.
- R package: *leaps* :: *regsubsets*(), no more than 50 variables.

Mallows's Cp

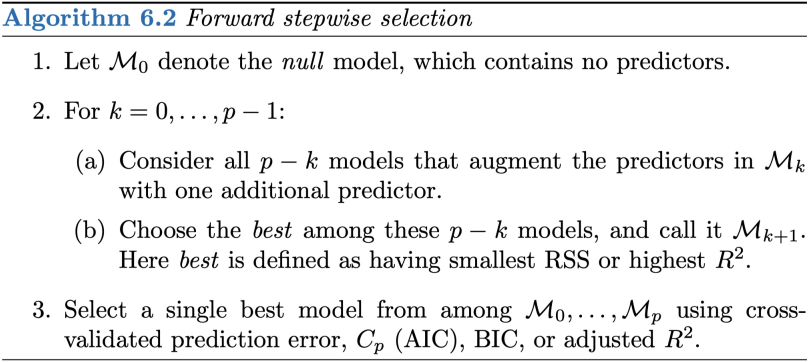
Mallow's C_p is a technique for model selection in regression (Mallows 1973). The C_p statistic is defined as a criteria to assess fits when models with different numbers of parameters are being compared. If k regressors are selected from a set of $k \leq p$, the C_p statistic for that particular set of regressors is defined as:

$$C_k = \frac{RSS(k)}{S^2} - n + 2(k+1),$$

where

- $RSS(k) = \sum_{i=1}^{n}(y_i - \hat{y}_{pi})^2$ is the error sum of squares for the model with k regressors, $\hat{y}_{pi}$ is the predicted value of the ith observation of y from the k regressors.
- S^2 is the residual mean square after regression on the complete set of k regressors and can be estimated by mean square error MSE.
- n is the sample size.

Forward Stepwise Selection

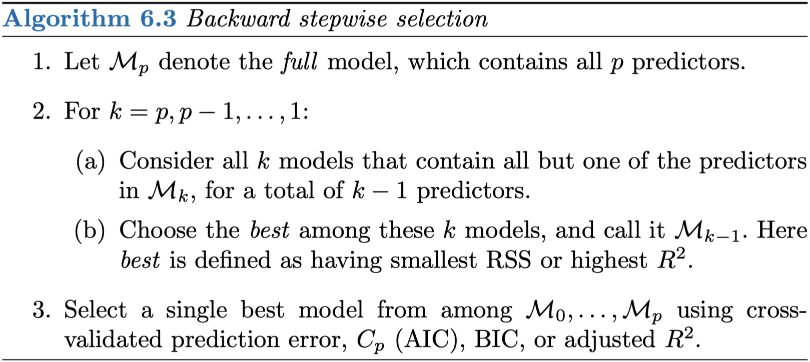


Algorithm 6.2 *Forward stepwise selection*

```
1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
    (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 $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$ (AIC), BIC, or adjusted $R^2$.
```

- Stopping rule, when all remaining variables have a p-value larger than some threshold if added to the model, the threshold can be 0.05, or 0.15 or else.

Backward Stepwise Selection



Algorithm 6.3 *Backward stepwise selection*

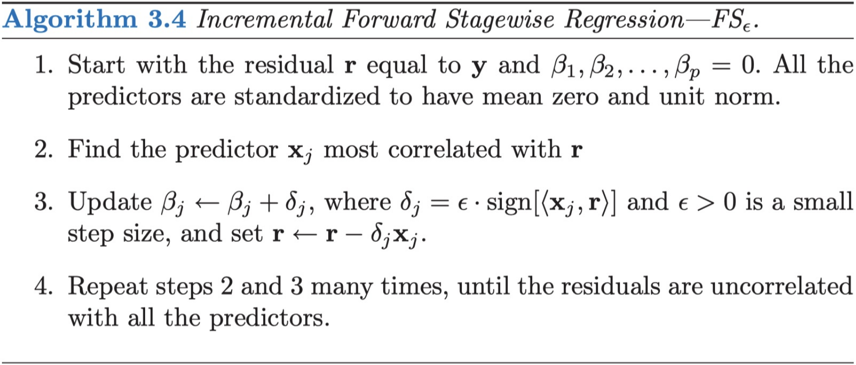
1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$.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

- Stopping rule, when some variables in the active set have a p-value larger than some threshold if added to the model, the threshold can be 0.05, or 0.15 or else.

Comparing Forward and Backward

- Forward stepwise selection can be applied even in the high-dimensional setting where $n < p$; however, in this case, it is possible to construct sub-models $M_0, \cdots, M_{n-1}$ only, since using least squares will not yield a unique solution if $P \geq n$. Thus, Backward stepwise selection can not be applied to the setting $p \geq n$.
- Both Forward and Backward stepwise selection approach searches through only $1 + p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection.
- Forward and Backward stepwise selection are not guaranteed to yield the best model containing a subset of the p predictors.

Forward-Stagewise Regression



Algorithm 3.4 *Incremental Forward Stagewise Regression—FS_ϵ.*

1. Start with the residual $\mathbf{r}$ equal to $\mathbf{y}$ and $\beta_1, \beta_2, \ldots, \beta_p = 0$. All the predictors are standardized to have mean zero and unit norm.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$

3. Update $\beta_j \leftarrow \beta_j + \delta_j$, where $\delta_j = \epsilon \cdot \text{sign}[\langle \mathbf{x}_j, \mathbf{r} \rangle]$ and $\epsilon > 0$ is a small step size, and set $\mathbf{r} \leftarrow \mathbf{r} - \delta_j \mathbf{x}_j$.

4. Repeat steps 2 and 3 many times, until the residuals are uncorrelated with all the predictors.

Forward-Stagewise Regression

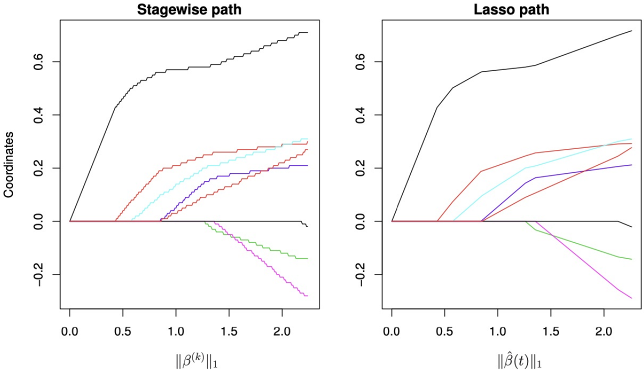


Figure 1: *A simple example using the prostate cancer data from Hastie et al. (2009), where the log PSA score of $n = 67$ men with prostate cancer is modeled as a linear function of $p = 8$ biological predictors. The left panel shows the forward stagewise regression estimates $\beta^{(k)} \in \mathbb{R}^8$, $k = 1, 2, 3, \ldots$, with the 8 coordinates plotted in different colors. The stagewise algorithm was run with $\epsilon = 0.01$ for 250 iterations, and the x-axis here gives the ℓ_1 norm of the estimates across iterations. The right panel shows the lasso solution path, also parametrized by the ℓ_1 norm of the estimate. The similarity between the stagewise and lasso paths is visually striking; for small enough ϵ, they appear identical. This is not a coincidence and has been rigorously studied by Efron et al. (2004), and other authors; in Section 2.1 we provide an intuitive explanation for this phenomenon.*

Comparing Stagewise with Stepwise

- Greediness is counterbalanced by the small step size $\epsilon > 0$.
- The learning process is slower in stepwise. In stepwise, the coefficient of X_i increases by a large amount in the fitted model, in forward stagewise only increases it by ϵ.
- It could easily take thousands of iterations to reach a model with only tens of active variables (Tibshirani, 2015).

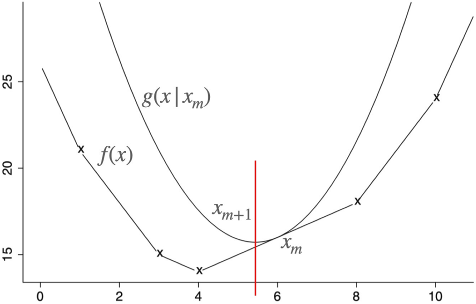
Review Majorization Minimisation (MM) Algorithm

A function $g(\boldsymbol{x}|\boldsymbol{x}_m)$ is said to majorize a function $f(\boldsymbol{x})$ at $\boldsymbol{x}_m$ provided

$$\begin{aligned} f\left(\boldsymbol{x}_m\right) &= g\left(\boldsymbol{x}_m \mid \boldsymbol{x}_m\right) \\ f(\boldsymbol{x}) &\leq g\left(\boldsymbol{x} \mid \boldsymbol{x}_m\right), \quad \boldsymbol{x} \neq \boldsymbol{x}_m \end{aligned}$$

If $\boldsymbol{x}_{m+1}$ denotes the minimum of the surrogate $g(\boldsymbol{x}|\boldsymbol{x}_m)$, then we can show that the MM procedure forces $f(\boldsymbol{x})$ downhill

$$f\left(\boldsymbol{x}_{m+1}\right) \leq g\left(\boldsymbol{x}_{m+1} \mid \boldsymbol{x}_m\right) \leq g\left(\boldsymbol{x}_m \mid \boldsymbol{x}_m\right) = f\left(\boldsymbol{x}_m\right)$$



Iterate Hard Thresholding

- The Iterate Hard Thresholding (IHT) algorithm was first proposed by (Blumensath and Davies, 2009).
- The IHT type algorithm: Hard Thresholding Pursuit(Foucart, 2011), SDAR(Huang et al., 2018)
- It can converge under some assumptions on X(Blumensath and Davies, 2009).
- It can recovery β given the sparsity level s and under the assumption on X (Blumensath and Davies, 2010).

Iterate Hard Thresholding

- Instead of optimising Lagrangian form directly:

$$\min_{\beta \in \mathbb{R}^p} F(\beta) = \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_0.$$

- Blumensath and Davies (2009) introduced a surrogate function

$$C_\lambda(\boldsymbol{\beta}|z) = \tfrac{1}{2n}\|y - X\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_0 - \tfrac{1}{2n}\|X\boldsymbol{\beta} - Xz\|_2^2 + \tfrac{1}{2}\|\boldsymbol{\beta} - z\|_2^2.$$

- If $||\frac{X}{\sqrt{n}}|| < 1$, this surrogate function is a majorization of $F(\beta)$, we have

$$F(\boldsymbol{\beta}^m) = C_\lambda(\boldsymbol{\beta}^m|\boldsymbol{\beta}^m) \geq \min_{\boldsymbol{\beta}} C_\lambda(\boldsymbol{\beta}|\boldsymbol{\beta}^m) = C_\lambda(\boldsymbol{\beta}^{m+1}|\boldsymbol{\beta}^m) > F(\boldsymbol{\beta}^{m+1}).$$

Iterate Hard Thresholding

- The surrogate function can be written as

$$C_\lambda(\boldsymbol{\beta}|z) = \frac{1}{2}\sum_i \left[\beta_i^2 - 2\beta_i\left(z_i + x_i^\top(y - Xz)/n\right) + 2\lambda\,|\beta_i|_0\right] + \frac{1}{2n}\left(\|y\|_2^2 + n\|z\|_2^2 - \|Xz\|_2^2\right).$$

- Fixed z, minimize the surrogate function,

$$\boldsymbol{\beta} = H_\lambda\left(z + X^\top(y - Xz)/n\right), \text{ where } (H_\lambda(x))_i = \begin{cases} 0, & \text{if } |x_i| < \sqrt{2\lambda} \\ x_i, & \text{if } |x_i| \geq \sqrt{2\lambda}. \end{cases}$$

- The Iterative Hard Thresholding algorithm is now defined as

$$\boldsymbol{\beta}^{m+1} = H_\lambda\left(\boldsymbol{\beta}^m + X^\top\left(y - X\boldsymbol{\beta}^m\right)/n\right).$$

Iterate Hard Thresholding

Proof.

Consider β_i, denote $a = z_i + x_i^\top (y - Xz)/n$ and $f_i(\beta_i)$ as

$$
\begin{aligned}
f_i(\beta_i) &= \beta_i^2 - 2\beta_i \left(z_i + x_i^\top (y - Xz)/n\right) + 2\lambda \left|\beta_i\right|_0 \\
&= (\beta_i - a)^2 + 2\lambda \left|\beta_i\right|_0 - a^2,
\end{aligned}
$$

If $\beta_i = 0$, $f_i(0) = 0$. Else if, $\beta_i \neq 0$, $f_i(\beta_i) = (\beta_i - a)^2 + 2\lambda - a^2$, it reaches minimum at $\beta_i = a$, and $f_i(a) = 2\lambda - a^2$.
Therefore, when $2\lambda - a^2 \leq f_i(0) = 0$, we get $\beta_i = a = z_i + x_i^\top (y - Xz)/n$.
Instead, if $2\lambda - a^2 > f_i(0) = 0$, we get $\beta_i = 0$. Thus, by minimizing the surrogate function, we get

$$
\boldsymbol{\beta} = H_\lambda \left(z + X^\top (y - Xz)/n\right), \text{ where } \left(H_\lambda(x)\right)_i = \begin{cases} 0, & \text{if } |x_i| < \sqrt{2\lambda} \\ x_i, & \text{if } |x_i| \geq \sqrt{2\lambda}. \end{cases}
$$

□

Iterate Hard Thresholding

For the constrained problem

$$\min_{\beta \in \mathbb{R}^p} L(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2, \text{ s.t. } \|\boldsymbol{\beta}\|_0 \leq s.$$

- The surrogate function can be defined as

$$C_s(\boldsymbol{\beta}|z) = \frac{1}{2n}\|y - X\boldsymbol{\beta}\|_2^2 - \frac{1}{2n}\|X\boldsymbol{\beta} - Xz\|_2^2 + \frac{1}{2}\|\boldsymbol{\beta} - z\|_2^2.$$

- Fixed z, minimize the surrogate objective function subject to $||\boldsymbol{\beta}||_0 \leq s$,

$$\boldsymbol{\beta} = H_s\left(z + X^\top(y - Xz)/n\right), \quad (H_s(\boldsymbol{\beta}))_i = \begin{cases} 0, & \text{if } |\beta_i| < \|\boldsymbol{\beta}\|_{s,\infty} \\ \beta_i, & \text{if } |\beta_i| \geq \|\boldsymbol{\beta}\|_{s,\infty}. \end{cases}$$

  where $\|\boldsymbol{\beta}\|_{s,\infty}$ is the s-sth largest elements of $\boldsymbol{\beta}$. Such choice of threshold is intended to select exactly s variables.

- The Iterative Hard Thresholding algorithm is now defined as

$$\boldsymbol{\beta}^{m+1} = H_s\left(\boldsymbol{\beta}^m + X^\top\left(y - X\boldsymbol{\beta}^m\right)/n\right).$$

Hard Thresholding Pursuit

- IHT algorithm: $\boldsymbol{\beta}^{m+1} = H_s\left(\boldsymbol{\beta}^m + X^\top\left(y - X\boldsymbol{\beta}^m\right)/n\right).$
- The basic idea of HTP algorithm is to chase a good candidate for the support then find the least square solution on this support.
- Start with an s-sparse $\boldsymbol{\beta} \in \mathbb{R}^p$, and iterate the scheme

$$A^{m+1} = \{\text{indices of s largest entries of } X^\top\left(y - X\boldsymbol{\beta}^m\right)/n\}$$
$$\boldsymbol{\beta}^{m+1} = \arg\min\{||y - X\boldsymbol{\beta}||^2, \text{supp}(\beta) \subseteq A^{m+1}\}.$$

Stop until $A^{m+1} = A^m$.

SDAR

- The goal of SDAR algorithm (Huang et al., 2018) is to attain coordinate-wise minimum.
- Let $\boldsymbol{\beta}^\diamond$ is a coordinate-wise minimum of Lagrangian form, and $X_j^T X_j = n$,

$$
\begin{aligned}
\beta_j^\diamond &\in \arg\min_{t_j \in \mathbb{R}} F_\lambda\left(\beta_1^\diamond, \ldots, \beta_{j-1}^\diamond, t_j, \beta_{j+1}^\diamond, \ldots, \beta_p^\diamond\right) \\
&\in \arg\min_{t_j \in \mathbb{R}} \frac{1}{2n}\left\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\diamond + \mathbf{X}_j\left(\beta_j^\diamond - t_j\right)\right\|_2^2 + \lambda\left\|t_j\right\|_0 \\
&\in \arg\min_{t_j \in \mathbb{R}} \frac{1}{2}\left(\beta_j^\diamond - t_j\right)^2 + \left(\beta_j^\diamond - t_j\right)\mathbf{X}_j^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\diamond\right)/n + \lambda\left\|t_j\right\|_0 \\
&\in \arg\min_{t_j \in \mathbb{R}} \frac{1}{2}\left(t_j - \beta_i^\diamond - \mathbf{X}_j^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\diamond\right)/n\right)^2 + \lambda\left\|t_j\right\|_0 .
\end{aligned}
$$

- Necessary condition:

$$
\beta_j^\diamond = H_\lambda\left(\beta_j^\diamond + \mathbf{X}_j^T\left(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\diamond\right)/n\right),
$$

$$
\text{where } \left(H_\lambda(\boldsymbol{\beta})\right)_j = \begin{cases} 0, & \text{if } |\beta_j| < \sqrt{2\lambda} \\ \beta_j, & \text{if } |\beta_j| \geq \sqrt{2\lambda}. \end{cases}
$$

SDAR

- For Lagrangian form problem, by the necessary condition

$$A^\diamond = \left\{ j \in S \mid |\beta_j^\diamond + d_j^\diamond \,| \geq \sqrt{2\lambda} \right\}, I^\diamond = \left(A^\diamond\right)^c. \tag{1}$$

  where $S = \{1, 2, \cdots, p\}$ and update $\mathbf{d}^\diamond = (\mathbf{d}_{A^\diamond}, \mathbf{d}_{I^\diamond})$ and $\boldsymbol{\beta}^\diamond = (\boldsymbol{\beta}_{A^\diamond}, \boldsymbol{\beta}_{I^\diamond})$ as

$$\begin{cases} \boldsymbol{\beta}_{I^\diamond}^\diamond = 0, \\ \mathbf{d}_{A^\diamond}^\diamond = 0, \\ \boldsymbol{\beta}_{A^\diamond}^\diamond = \left(X_{A^\diamond}^\top X_{A^\diamond}\right)^{-1} X_{A^\diamond}^\top \mathbf{y} \\ \mathbf{d}_{I^\diamond}^\diamond = X_{I^\circ}^\top \left(\mathbf{y} - X_{A^\diamond} \boldsymbol{\beta}_{A^\diamond}^\diamond\right) / n. \end{cases} \tag{2}$$

- We can solve this system of equations iteratively. Then with an initial β_0 and using (1) and (2) with λ_m satisfying

$$\sqrt{2\lambda_m} = \|\boldsymbol{\beta}^m + \mathbf{d}^m\|_{s,\infty},$$

  we obtain a sequence of solutions $\{\beta_m, m \geq 1\}$.

SDAR

There are two key aspects of SDAR. In (1) we detect the support of the solution based on the sum of the primal (β_m) and dual ($\mathbf{d_m}$) approximations and, in (2) we calculate the least squares solution on the detected support. Therefore, SDAR can be considered an iterative method for solving the necessary condition with an important modification: a different λ_m value in each step of the iteration is used. Thus we can also view SDAR as a method that combines adaptive thresholding using primal and dual information and least-squares fitting.

Splicing Algorithm

- Consider the ℓ_0 constraint minimization problem,

$$L_n(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2, \text{ s.t } \|\boldsymbol{\beta}\|_0 \leq s.$$

- Given any set $A \subset \{1, 2, \cdots, p\}$ with cardinality $|A| = s$, denote $I = A^c$, we compute

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}_I = 0} L_n(\boldsymbol{\beta}).$$

- Denote A and I as the active set and inactive set.

Splicing Algorithm

Two types of sacrifices.

- **Backward sacrifices**: for any $j \in A$, the importance of variable j is

$$\xi_j = L_n(\hat{\boldsymbol{\beta}} - \hat{\beta}_j \mathbf{e}_j) - L_n(\hat{\boldsymbol{\beta}}).$$

- **Forward sacrifices**: for any $j \in I$, the importance of variable j is

$$\xi_j = L_n(\hat{\boldsymbol{\beta}}) - L_n(\hat{\boldsymbol{\beta}} + \hat{t}\mathbf{e}_j).$$

  where $\mathbf{e}_j$ is a $p \times 1$ vector with j-th element 1 and 0 otherwise, $\hat{t}$ is a parameter that minimizes $L_n(\hat{\boldsymbol{\beta}} + \hat{t}\mathbf{e}_j)$.

- Intuitively, for $j \in A$ or $j \in I$, a large ξ_j implies the j-th variables is potentially important.
- If we exchange some irrelevant variables in A and some important variables in I, it may result in a higher quality solution.

Splicing Set

Specifically, given any splicing size $C \leq s$, define

- $S_{C,1}$ represents C irrelevant variables in A.

$$S_{C,1} = \left\{ j \in A : \sum_{i \in A} \mathrm{I}\left(\xi_j \geq \xi_i\right) \leq C \right\},$$

- $S_{C,2}$ represents C relevant variables in I.

$$S_{C,2} = \left\{ j \in I : \sum_{i \in I} \mathrm{I}\left(\zeta_j \leq \zeta_i\right) \leq C \right\},$$

$\{1,2,\ldots,p\}$			
A		I	
$\boldsymbol{\beta}_A \neq 0$		$\boldsymbol{\beta}_I = 0$	
$d_A = 0$		$d_I \neq 0$	
	$S_{C,1}$	$S_{C,2}$	

References I

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The annals of statistics*, 44(2): 813–852, 2016.

Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27 (3):265–274, 2009.

Thomas Blumensath and Mike E Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of selected topics in signal processing*, 4(2):298–309, 2010.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008.

Jianqing Fan and Jinchi Lv. Sure independence screening. *Wiley StatsRef: Statistics Reference Online*, 2018.

References II

Simon Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

T. Hastie, R. Tibshirani, and J. Friedman. [springer series in statistics] the elements of statistical learning ——. 10.1007/978-0-387-84858-7, 2009.

Jian Huang, Yuling Jiao, Yanyan Liu, Xiliang Lu, et al. A constructive approach to l0 penalized regression. 2018.

Ryan J Tibshirani. A general framework for fast stagewise algorithms. *J. Mach. Learn. Res.*, 16(1):2543–2588, 2015.

Table of Contents

1. Introduction
2. Model Specification
3. Robust linear regression
4. Ridge Regression and PCA
5. Best Subset Selection
   - Traditional Subset Selection
   - Iterate Hard Thresholding
   - Splicing Algorithm
6. Appendix

Mixed Integer Optimization Algorithm

- (Bertsimas et al., 2016) propose a a Mixed Integer Optimization(MIO) algorithm for solving the classical best subset selection problem.
- **bestsubset**: relies on certain third-party integer optimization solvers (Gurobi) and only works for LM.
- It can obtain near-optimal solutions to this problem for instances where the number of features about $p = 1000$.

Background on MIO

The general form of a Mixed Integer Quadratic Optimization (MIO) problem is as follows:

$$\begin{aligned} \min_{\alpha} \; & \boldsymbol{\alpha}^T \mathbf{Q} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{a} \\ \text{s.t.} \quad & \mathbf{A} \boldsymbol{\alpha} \leq \mathbf{b} \\ & \alpha_i \in \{0, 1\}, \quad i \in \mathcal{I}, \\ & \alpha_j \geq 0, \quad j \notin \mathcal{I}, \end{aligned}$$

where $\boldsymbol{a} \in \mathbb{R}^m, \mathbf{b} \in \mathbb{R}^k, \boldsymbol{A} \in \mathbb{R}^{k \times m}, \boldsymbol{Q} \in \mathbb{R}^{m \times m}$.

MIO formulation for the Best Subset Selection

The formulation of Best Subset Selection as a MIO

$$\begin{aligned} \min_{\beta,z} \ & \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|_2^2 \\ \text{s.t.} \quad & -\mathcal{M}z_i \leq \beta_i \leq \mathcal{M}z_i \\ & z_i \in \{0,1\}, i=1,\ldots,p \\ & \sum_{i=1}^{p} z_i \leq s, \end{aligned}$$

where $z_i \in \{0,1\}$ is a binary variable, which represents whether i th variable is in the model, $\mathcal{M}$ is a constant that if $\hat{\beta}$ is a minimizer, then $\mathcal{M} \geq ||\hat{\beta}||_\infty$. The choice of $\mathcal{M}$ affects the strength of the formulation and is critical for obtaining good lower bounds in practice.