Chapter 3 - Linear Methods for Regression

The Elements of Statistic Learning 2nd Edition

by Liming Huang
Overview

• Linear Regression Models and Least Squares
• Selection
• Shrinkage Methods
• Methods Using Input Directions
• Further Discussion
Linear Regression Models and Least Squares

- Assumption: The regression function $E(Y|X)$ is linear or the linear model is a reasonable approximation.
- Use the input vector $X^T = (X_1, X_2, \ldots, X_p)$ to predict a real-valued output $Y$.
- Form:  
  \[ f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j. \]  
  \[ (3.1) \]

- $X_j$ can come from different sources
• Least squares:

\[
\text{RSS}(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2
\]

\[
= \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2.
\]  

(3.2)

• Here

\[
\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T
\]

• \(X\) as \(N \times (p+1)\) matrix:

\[
\text{RSS}(\beta) = (y - X\beta)^T(y - X\beta).
\]

• To minimize:

\[
\frac{\partial \text{RSS}}{\partial \beta} = -2X^T(y - X\beta)
\]

\[
\frac{\partial^2 \text{RSS}}{\partial \beta \partial \beta^T} = 2X^TX.
\]
• Assume $X$ has full column rank:

- $X^T(y - X\beta) = 0$
- $\hat{\beta} = (X^TX)^{-1}X^Ty.$
- $\hat{y} = X\hat{\beta} = X(X^TX)^{-1}X^Ty,$

**FIGURE 3.2.** 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 $x_1$ and $x_2$. The projection $\hat{y}$ represents the vector of the least squares predictions.
• What if $X$ is not of full rank?

• It will still work but with more than one way to express that projection in terms of the column vectors of $X$
• Assume the observations $y_i$ are uncorrelated with a constant variance $\sigma^2$ and the $x_i$ are fixed

$$\text{Var}(\hat{\beta}) = (X^T X)^{-1} \sigma^2.$$

• $\Rightarrow E(y) = X\beta$

• $\Rightarrow E(\hat{\beta}) = (X^T X)^{-1} X^T X \beta = \beta$

• $\Rightarrow \hat{\beta} - E(\hat{\beta}) = (X^T X)^{-1} X^T \varepsilon$

• $\Rightarrow \text{Var}(\hat{\beta}) = E[(\hat{\beta} - E(\hat{\beta}))(\hat{\beta} - E(\hat{\beta}))^T]$

$$\hat{\beta} \sim N(\beta, (X^T X)^{-1} \sigma^2).$$
• Test the hypothesis that a particular coefficient $\hat{\beta}_j = 0$
  • Z-score
  
  $$z_j = \frac{\hat{\beta}_j}{\hat{\sigma} \sqrt{v_j}},$$

  where $v_j$ is the $j$th diagonal element of $(X^T X)^{-1}$ and $z_j$ is distributed as $t_{N-p-1}$

• Test the significance of groups of coefficients simultaneously
  • F statistic:
  
  $$F = \frac{(RSS_0 - RSS_1)/(p_1 - p_0)}{RSS_1/(N - p_1 - 1)},$$
• The Gauss-Markov Theorem
  • The least squares estimator has the smallest mean squared error of all linear estimators with no bias.
  • Focus on estimation of any linear combination of \( \theta = a^T \beta \);
  • The least square estimate: \( \hat{\theta} = a^T \hat{\beta} = a^T (X^T X)^{-1} X^T y \).
  • Assume the linear model is correct:

\[
E(a^T \hat{\beta}) = E(a^T (X^T X)^{-1} X^T y) \\
= a^T (X^T X)^{-1} X^T X \beta \\
= a^T \beta.
\]

• Then we can get: \( \text{Var}(a^T \hat{\beta}) \leq \text{Var}(c^T y) \).
• Multiple Regression from Simple Univariate Regression
  • Multiple linear regression model: the linear model with p>1 inputs
  • Univariate regression: p=1

• A univariate model with no intercept: \( Y = X\beta + \varepsilon \)
  • Least square estimate:
    \[
    \hat{\beta} = \frac{\sum_{i=1}^{N} x_i y_i}{\sum_{i=1}^{N} x_i^2}, \quad \Rightarrow \quad \hat{\beta} = \frac{\langle x, y \rangle}{\langle x, x \rangle},
    \]
  • Residuals:
    \[
    r_i = y_i - x_i \hat{\beta}.
    \]

In convenient vector notation, we let \( y = (y_1, \ldots, y_N)^T, x = (x_1, \ldots, x_N)^T \) and define

\[
\langle x, y \rangle = \sum_{i=1}^{N} x_i y_i, \\
= x^T y,
\]

(3.25)
• For multiple regression model:
  • If the inputs (The columns of the data matrix X) are orthogonal
    • For all \( j \neq k \), we have \( \langle x_j, x_k \rangle = 0 \)
    • \( \hat{\beta}_j \) are equal to \( \frac{\langle x_j, y \rangle}{\langle x_j, x_j \rangle} \)
    • The inputs have no effect on each other’s parameter estimates in the model
  • The inputs are not orthogonal
    • The most common case
    • We need to orthogonalize them
    • Suppose we have an intercept and a single input \( x \), then
      \[
      \hat{\beta}_1 = \frac{\langle x - \bar{x}_1, y \rangle}{\langle x - \bar{x}_1, x - \bar{x}_1 \rangle},
      \]
      where \( \bar{x} = \sum_i x_i/N \), and \( 1 = x_0 \), the vector of \( N \) ones.
Two steps:

1. regress $x$ on $1$ to produce the residual $z = x - \bar{x}1$;

2. regress $y$ on the residual $z$ to give the coefficient $\hat{\beta}_1$.

In this procedure, “regress $b$ on $a$” means a simple univariate regression of $b$ on $a$ with no intercept, producing coefficient $\hat{\gamma} = \langle a, b \rangle / \langle a, a \rangle$ and residual vector $b - \hat{\gamma}a$. We say that $b$ is adjusted for $a$, or is “orthogonalized” with respect to $a$. 
• Generalize to the case of $p$ inputs: (Gram-Schmidt)

Algorithm 3.1 Regression by Successive Orthogonalization.

1. Initialize $z_0 = x_0 = 1$.
2. For $j = 1, 2, \ldots, p$
   
   Regress $x_j$ on $z_0, z_1, \ldots, z_{j-1}$ to produce coefficients $\hat{\gamma}_{\ell j} = \langle z_\ell, x_j \rangle / \langle z_\ell, z_\ell \rangle$, $\ell = 0, \ldots, j - 1$ and residual vector $z_j = x_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} z_k$.
3. Regress $y$ on the residual $z_p$ to give the estimate $\hat{\beta}_p$.

Note that the inputs $z_0, \ldots, z_{j-1}$ in step 2 are orthogonal

• Any one of the inputs can be in the position, similar results holds.
The result of this algorithm is

\[ \hat{\beta}_p = \frac{\langle z_p, y \rangle}{\langle z_p, z_p \rangle}. \]

\[ \text{Var}(\hat{\beta}_p) = \frac{\sigma^2}{\langle z_p, z_p \rangle} = \frac{\sigma^2}{\| z_p \|^2}. \]
2. For $j = 1, 2, \ldots, p$

Regress $x_j$ on $z_0, z_1, \ldots, z_{j-1}$ to produce coefficients $\hat{\gamma}_j = \frac{\langle z_{\ell}, x_j \rangle}{\langle z_{\ell}, z_{\ell} \rangle}$, $\ell = 0, \ldots, j - 1$ and residual vector $z_j = x_j - \sum_{k=0}^{j-1} \hat{\gamma}_j z_k$.

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

$$X = Z\Gamma,$$

- Introducing the diagonal matrix with jth diagonal entry $D_{jj} = \|z_j\|$
- QR decomposition:
  $$X = ZD^{-1}\Gamma = QR,$$
  $$\hat{\beta} = R^{-1}Q^T y,$$
  $$\hat{y} = QQ^T y.$$
Subset Selection

• The least squares estimates:
  • Prediction accuracy
  • Interpretation

• To retain only a subset of the variables
  • Best-subset selection
  • Forward- and backward-stepwise selection
  • Forward-stagewise regression
Shrinkage Methods

• Ridge Regression
  • Imposing a penalty on the regression coefficients’ size
  • Standardize the input first

\[ \hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2, \]

subject to \( \sum_{j=1}^{p} \beta_j^2 \leq t, \)

\[ \hat{\beta}_{\text{ridge}} = \arg \min_{\beta} \left\{ \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}. \]
Writing the criterion in (3.41) in matrix form,

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

the ridge regression solutions are easily seen to be

$$\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y,$$
• The mean of a posterior distribution
  • Suppose $(y_i | \beta) \sim N(\beta_0 + x_i^T \beta, \sigma^2) = N(\beta_0 + \sum_{j=1}^{P} x_{ij} \beta_j)$
  • $\beta_j \sim N(0, \sigma^2)$, independently of one another
  • Posterior distribution:

$$p(\beta | y) = \frac{p(y | \beta)p(\beta)}{p(y)}$$

It is direct proportional to

$$p(y | \beta)p(\beta) = (2\pi\sigma^2)^{-\frac{N}{2}}(2\pi\delta^2)^{-\frac{P}{2}} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{P} x_{ij} \beta_j)^2 - \frac{1}{2\delta^2} \sum_{j=1}^{P} \beta_j^2 \right\}$$

• Let $\lambda = \frac{\sigma^2}{\delta^2}$ and compare with

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{P} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{P} \beta_j^2 \right\}.$$
• The Singular Value Decomposition

\[ X = UDV^T. \]

Using the singular value decomposition we can write the least squares fitted vector as

\[ X_\beta^{ls} = X(X^TX)^{-1}X^T y \]
\[ = UU^T y, \quad (3.46) \]

Now the ridge solutions are

\[ X_{\hat{\beta}}^{\text{ridge}} = X(X^TX + \lambda I)^{-1}X^T y \]
\[ = UD(D^2 + \lambda I)^{-1}D U^T y \]
\[ = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^T y, \]

• It’s obvious that \[ \frac{d_j^2}{d_j^2 + \lambda} < 1 \]
• The Lasso
  • The L2 ridge penalty to L1 lasso penalty
  • Making $t$ sufficiently small will cause some of the coefficients to be exactly zero

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2$$

subject to $\sum_{j=1}^{p} |\beta_j| \leq t$.

$$\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$
• The Lasso
  • The L2 ridge penalty to L1 lasso penalty
  • Making $t$ sufficiently small will cause some of the coefficients to be exactly zero

\[
\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2
\]

subject to
\[
\sum_{j=1}^{p} |\beta_j| \leq t.
\]

\[
\hat{\beta}_{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^{N} \left( y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.
\]
TABLE 3.4. Estimators of $\beta_j$ in the case of orthonormal columns of $X$. $M$ and $\lambda$ are constants chosen by the corresponding techniques; sign denotes the sign of its argument ($\pm 1$), and $x_+$ denotes “positive part” of $x$. Below the table, estimators are shown by broken red lines. The 45° line in gray shows the unrestricted estimate for reference.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best subset (size $M$)</td>
<td>$\hat{\beta}_j \cdot I(</td>
</tr>
<tr>
<td>Ridge</td>
<td>$\hat{\beta}_j / (1 + \lambda)$</td>
</tr>
<tr>
<td>Lasso</td>
<td>$\text{sign}(\hat{\beta}_j) (</td>
</tr>
</tbody>
</table>

![Graphs showing Best Subset, Ridge, and Lasso estimators]
• Ridge regression and Lasso

**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.
• Generalization

\[ \tilde{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\} \]

**FIGURE 3.12.** Contours of constant value of \(\sum_j |\beta_j|^q\) for given values of \(q\).
• Path Algorithm
  • All the inputs and outputs have been standardized
  • Forward Stepwise

算法的目标是最小化残差平方和。

第一步，模型所有变量的回归系数都是0，我们需要找出一个变量，估计其回归系数，使得这个模型最能够减小当前的残差平方和，即最能改进模型的拟合情况。

第二步，从剩余变量中选出一个，使得加入这个变量之后的活跃变量集合能够使得模型拟合进步最大。用所有活跃集中的变量进行回归。

重复第二步，直到符合某种准则。

请注意，对于stepwise而言，每一步的关键是找出最能改进拟合情况的变量，然后对所有入选的变量，即被激活的变量集合重新做回归。也就是说每一步都做了多元回归。
• **Forward Stagewise**

简单描述一下这个算法，在开始的时候，所有输入变量的系数都是0，即都处于睡眠状态，那么当前的残差就是输出$y$。

算法开始，找出和当前残差相关系数最高的变量，激活它，用当前的残差和其做回归，估计出回归系数。计算当前残差。

然后再找出所有变量中和当前残差相关系数最高的，激活它，用当前残差和其做回归，计算处回归系数，计算当前残差。注意这里在计算时，之前入选的变量的系数并没有变化。

然后在找出所有变量中和当前残差相关系数最高的（有可能是已经入选的变量），激活它，再用当前残差和其做回归，计算系数，计算当前残差。

重复上述过程直到达到某种准则。对于观测样本量大于变量数的情况，最终的结果就是普通的最小二乘估计。

从算法的描述我们可以看出，逐渐回归每一步并非充分地估足了回归系数。所以这种算法又被成为“慢拟合”（slow fitting）
• Least Angle Regression

Algorithm 3.2 Least Angle Regression.

1. Standardize the predictors to have mean zero and unit norm. Start with the residual \( r = y - \bar{y}, \beta_1, \beta_2, \ldots, \beta_p = 0. \)

2. Find the predictor \( x_j \) most correlated with \( r. \)

3. Move \( \beta_j \) from 0 towards its least-squares coefficient \( \langle x_j, r \rangle, \) until some other competitor \( x_k \) has as much correlation with the current residual as does \( x_j. \)

4. Move \( \beta_j \) and \( \beta_k \) in the direction defined by their joint least squares coefficient of the current residual on \( (x_j, x_k), \) until some other competitor \( x_l \) has as much correlation with the current residual.

5. Continue in this way until all \( p \) predictors have been entered. After \( \min(N - 1, p) \) steps, we arrive at the full least-squares solution.
Suppose $A_k$ is the active set of variables at the beginning of the $k$th step, and let $\beta_{A_k}$ be the coefficient vector for these variables at this step; there will be $k - 1$ nonzero values, and the one just entered will be zero. If $r_k = y - X_{A_k}\beta_{A_k}$ is the current residual, then the direction for this step is

$$\delta_k = (X_{A_k}^T X_{A_k})^{-1}X_{A_k}^T r_k.$$  \hfill (3.55)
• The LAR and Lasso

**FIGURE 3.15.** Left panel shows the LAR coefficient profiles on the simulated data, as a function of the $L_1$ arc length. The right panel shows the Lasso profile. They are identical until the dark-blue coefficient crosses zero at an arc length of about 18.
• The loss function of Lasso

\[ R(\beta) = \frac{1}{2} \| y - X\beta \|_2^2 + \lambda \| \beta \|_1. \]

\[ \frac{\partial R}{\partial \beta} = -X^T y + X^T X \beta + \lambda \text{sign}(\beta), \quad \forall j \in A \]

\[ x_j^T (y - X\beta) = \lambda \cdot \text{sign}(\beta_j), \quad \forall j \in B \]

• For LAR

\[ x_j^T (y - X\beta) = \gamma \cdot s_j, \quad \forall j \in A \]
• When $s_j \neq \text{sign}(\beta_j)$, they LAR differs from Lasso in the path

---

**Algorithm 3.2a Least Angle Regression: Lasso Modification.**

4a. If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.
Methods Using Derived Input Directions

• Produce a small number of linear combinations $Z_m, m = 1, \ldots, M$ of the original inputs $X_j$, and the $Z_m$ are then used in place of the $X_j$ as inputs in the regression.

• Principal components regression

• Partial least square
• Principal Components Regression
  • Principal components \(Z_m\)
    • \(X = UDV^T\)
    • \(X^TX = VD^2VT\)
    • The columns of \(V\) are called the principal components directions of \(X\)
  • Steps:
    • Form the derived input columns \(z_m = Xv_m\)
    • Regress \(y\) on \(z_1, z_2, \ldots, z_M\) for some \(M \leq p\)
  • The \(z_m\) are orthogonal:

\[
\hat{y}_{(M)}^{pcr} = \bar{y}1 + \sum_{m=1}^{M} \hat{\theta}_m z_m, \quad \text{where } \hat{\theta}_m = \langle z_m, y \rangle / \langle z_m, z_m \rangle.
\]

\[
\hat{\beta}_{pcr}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m.
\]
• Partial least squares

Algorithm 3.3 Partial Least Squares.

1. Standardize each $x_j$ to have mean zero and variance one. Set $\hat{y}^{(0)} = \bar{y}1$, and $x_j^{(0)} = x_j$, $j = 1, \ldots, p$.

2. For $m = 1, 2, \ldots, p$

   (a) $z_m = \sum_{j=1}^{p} \hat{\phi}_{mj}x_j^{(m-1)}$, where $\hat{\phi}_{mj} = \langle x_j^{(m-1)}, y \rangle$.

   (b) $\hat{\theta}_m = \langle z_m, y \rangle / \langle z_m, z_m \rangle$.

   (c) $\hat{y}^{(m)} = \hat{y}^{(m-1)} + \hat{\theta}_m z_m$.

   (d) Orthogonalize each $x_j^{(m-1)}$ with respect to $z_m$: $x_j^{(m)} = x_j^{(m-1)} - \left[ \frac{\langle z_m, x_j^{(m-1)} \rangle}{\langle z_m, z_m \rangle} \right] z_m$, $j = 1, 2, \ldots, p$.

3. Output the sequence of fitted vectors $\{\hat{y}^{(m)}\}_{1}^{p}$. Since the $\{z_\ell\}_{1}^{m}$ are linear in the original $x_j$, so is $\hat{y}^{(m)} = X\hat{\beta}_{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.
Further Discussion

• Compare
  • PLS, PCR and ridge regression tend to behave similarly.
  • Ridge regression may be preferred because it shrinks smoothly.
  • Lasso falls somewhere between ridge regression and best subset regression, and enjoys some of the properties of each.
• Multiple outcome shrinkage and selection
  • Option 1: do not consider the correlation in different outcomes, and apply single outcome shrinkage and selection to each outcome
  • Option 2: exploit correlations in different outcomes

• Canonical Correlation Analysis
  • Derived input and outcome space based on canonical correlation analysis that maximize
    $\text{Corr}^2(YU_m, Xv_m)$
  • Regression in derived directions
    – Step 1: Map $y$ into derived directions
      $YU_m$
    – Step 2: Do regression in the derived space
      $X(X^TX)^{-1}X^TYU_m$
    – Step 3: Map back to $y$’s original space
      $\hat{Y}_{rr} = X(X^TX)^{-1}X^TYU_mU_m^{-1}$

• Reduced Rand Regression
Summary

• Bias Variance trade off:
  – Subset selection (feature selection, discrete)
  – Coefficient shrinkage (smoothing)
  – Using derived input direction

• Multiple outcome shrinkage and selection

• Most of the algorithms are sensitive to scaling of the parameters
  – Standardize the inputs, such as normalizing input directions to the same variance
The End

• Thank you!