

Comments on regression from last day (Table 3.2)

- ▶ feature variables `l_cavol`, etc. were standardized on the full data set; otherwise we wouldn't need an intercept because each predictor would have mean 0
- ▶ $x_i \rightarrow (x_i - \bar{x}) / \sqrt{\sum (x_i - \bar{x})^2 / (n - 1)}$
- ▶ some x 's are categorical (e.g. `svi`), so $\{0, 1\} \rightarrow \{a, b\}$: very hard to interpret
- ▶ fitting by successive orthogonalization, as in algorithm 3.2: see R code (`alg32.txt`)

Properties of least squares estimators (§3.2.2.)

- ▶ unbiased $E\hat{\beta} = \beta$
- ▶ Best linear unbiased $\text{var}(a^T \hat{\beta}) \leq \text{var}(c^T y)$ ($E(c^T y) = a^T \beta$)
- ▶ A biased estimator might have even smaller variance:

$$E(\tilde{\beta} - \beta)^2 = \text{var}(\tilde{\beta}) + \text{bias}^2(\tilde{\beta})$$
- ▶ The prediction error for an estimate of $E(Y | x_0) = x_0^T \beta$ is
 (3.22)

$$E(Y_0 - x_0^T \tilde{\beta})^2 = \sigma^2 + E\{x_0^T \tilde{\beta} - E(Y | x_0)\}^2 = \sigma^2 + \text{MSE}$$

- ▶ Much more on trade off between bias and variance for prediction later (Ch. 7)

Subset selection (§3.4.1)

- ▶ linear regression: forward, backward, stepwise, all possible subsets regression

$$\text{▶ } \text{RSS}(\hat{\beta}) = (y - X\hat{\beta})^T (y - X\hat{\beta}) = \sum (y_i - \hat{y}_i)^2$$

this is called SSE in the 302 text

- ▶ if we add a regressor, say from X_{p-1} to X_p , $\text{RSS}(\hat{\beta})$ necessarily decreases
- ▶ forward selection starts with one predictor (usually the constant term) and stops when no additional predictor is statistically significant
- ▶ backward selection starts with all predictors and deletes least significant
- ▶ stepwise is a hybrid where earlier deleted variables are candidates for re-insertion
- ▶ all possible subsets regression considers all 2^p models. Feasible with the “leaps and bounds” algorithm, implemented in package `leaps` (See figure 3.6)

Subset selection (§3.4.1)

- linear regression: forward, backward, stepwise, all possible subsets regression
- $$RSS(\beta) = (y - X\beta)^T (y - X\beta) = \sum (y_i - \hat{y}_i)^2$$
- if we add a regressor, say from X_{p-1} to X_p , $RSS(\hat{\beta})$ necessarily decreases
- forward selection starts with one predictor (usually the constant term) and stops when no additional predictor is statistically significant
- backward selection starts with all predictors and deletes least significant
- stepwise is a hybrid where earlier deleted variables are candidates for re-insertion
- all possible subsets regression considers all 2^p models. Feasible with the "leaps and bounds" algorithm, implemented in package `leaps` (See figure 3.6)

alg32.txt

“Mallows’ C_p ”

- ▶ a common adjustment to measure benefit of adding further parameters:

$$C_p = \frac{RSS_p}{\sigma^2} + 2p - N;$$

- ▶ estimated by

$$\hat{C}_p = \frac{RSS_p}{\hat{\sigma}^{*2}} + 2p - N$$

- ▶ $\hat{\sigma}^{*2}$ is the best possible estimate of σ^2 (usually from the full model)
- ▶ usually C_p is estimated by $\frac{RSS_p}{MSE_{full}} + 2p - N$, although it is sensitive to the estimation of σ^2 , and some books recommend using C_p only when σ^2 is known
- ▶ Choose p so that C_p is small and $C_p \simeq p$ can be shown to be a good choice for prediction (details deferred until Chapter 7)
- ▶ a closely related criterion that is more general is AIC , which in the linear model is approximately

2. Ridge regression

$$\begin{aligned}\hat{\beta}_{LS} &= (X^T X)^{-1} X^T y \\ \hat{\beta}_{ridge} &= (X^T X + \lambda I)^{-1} X^T y\end{aligned}$$

can show that $\hat{\beta}_{ridge}$ satisfies

$$\begin{aligned}\min_{\beta} \Sigma \{y_i - \bar{y} - \Sigma_{j=1}^p (x_{ij} - \bar{x}_j) \beta_j\}^2 + \lambda \Sigma_{j=1}^p \beta_j^2 \\ \min_{\beta} \Sigma \{y_i - \bar{y} - \Sigma_{j=1}^p (x_{ij} - \bar{x}_j) \beta_j\}^2 \quad \text{s.t. } \Sigma \beta_j^2 \leq s \\ \min (y - X\beta)^T (y - X\beta) \quad \text{s.t. } \|\beta\|^2 \leq s\end{aligned}$$

λ (or s) is a 'tuning parameter': $\lambda = 0$ gives $\hat{\beta}_{LS}$, $\lambda \rightarrow \infty$ gives \bar{y}

Figure 3.7

in R the library MASS ("library(MASS)") has a ridge regression version of lm called `lm.ridge`

2. Ridge regression

$$\hat{\beta}_{LS} = (X^T X)^{-1} X^T y$$

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

can show that $\hat{\beta}_{\text{ridge}}$ satisfies

$$\min_{\beta} \sum_j (y_j - \hat{y}_j - \sum_{k=1}^p (x_{jk} - \hat{x}_{jk}) \beta_k)^2 + \lambda \sum_{k=1}^p \beta_k^2$$

$$\min_{\beta} \sum_j (y_j - \hat{y}_j - \sum_{k=1}^p (x_{jk} - \hat{x}_{jk}) \beta_k)^2 \quad \text{s.t. } \sum_{k=1}^p \beta_k^2 \leq s$$

$$\min (y - X\hat{y})^T (y - X\hat{y}) \quad \text{s.t. } \|\hat{y}\|^2 \leq s$$

λ (or s) is a tuning parameter: $\lambda = 0$ gives $\hat{\beta}_{LS}$, $\lambda \rightarrow \infty$ gives \bar{y}

Figure 2.7

In R the library MASS ("library(MASS)") has a ridge regression version of `lm` called `lm.ridge`

X matrices do not have a leading column of 1's, and all other columns have been centered, so $\hat{\beta}_0 = \bar{y}$

one motivation is that if columns of X are nearly linearly dependent (multicollinearity), $\hat{\beta}$'s for these columns should be shrunk towards 0.

For this it is essential that the predictors are all scaled but this is difficult for interpretation of the coefs

Linear algebra

- ▶ To study multicollinearity, convenient to use the *singular value decomposition* of X :

$$X_{N \times p} = U_{N \times p} D_{p \times p} V_{p \times p}^T, \quad U^T U = I, \quad V^T V = I, \\ D = \text{diag}(d_1, \dots, d_p)$$

$$\begin{aligned} X \hat{\beta}_{LS} &= X(X^T X)^{-1} X^T y \\ &= U D V^T (V D U^T U D V^T)^{-1} V D U^T y \\ &= U D V^T V^{T^{-1}} D^{-2} V^{-1} V D U^T y \\ &= U U^T y = \sum_{j=1}^p u_j u_j^T y \end{aligned}$$

$$\begin{aligned}
 X\hat{\beta}_{ridge} &= X(X^T X + \lambda I)^{-1} X^T y \\
 &= UDV^T (VD^2 V^T + \lambda I)^{-1} VDU^T y \\
 &= UDV^T (VD^2 V^T + \lambda VV^T)^{-1} VDU^T y \\
 &= UD(D^2 + \lambda I)^{-1} DU^T y \\
 &= \sum_{j=1}^p u_j \left(\frac{d_j^2}{d_j^2 + \lambda} \right) u_j^T y
 \end{aligned}$$

$$df(\lambda) = \text{tr}[X(X^T X + \lambda I)^{-1} X^T] = \sum_{j=1}^p \frac{d_j^2}{d_j^2 + \lambda}$$

Figure 3.7

3. Lasso

- ▶ This also shrinks the components of β , but using a constraint on $\sum |\beta_j|$:

$$\min_{\beta} \sum (y_i - \bar{y} - \sum x_{ij}\beta_j)^2 \quad \text{s.t.} \quad \sum |\beta_j| \leq t$$

- ▶ this has the effect of setting some β_j to 0, for suitably small t , so functions like subset selection
- ▶ see Table 3.3 for a comparison on the prostate data; and Figure 3.9
- ▶ small test error from Table 3.3 (note test error for LS is 0.586 here)
- ▶ “least angle regression” generalizes lasso and ridge; recent *Annals* paper by Efron et al
- ▶ Skip 3.4.4, read 3.4.5, skip 3.4.6 (also skipped 3.3.1)

- ▶ to connect these to prediction error not completely straightforward
- ▶ in Table 3.3 each method had a tuning parameter to choose; they used cross-validation within the training data
- ▶ in `lm.ridge` you can extract a component called `$GCV`: this gives different results than the text
- ▶ the quantity $\sum d_j^2 / (d_j^2 + \lambda)$ has an interpretation as the number of 'degrees of freedom' or number of 'parameters' used by the ridge regression fit
- ▶ book says that the best value is 4.16, which corresponds to quite a large λ (39); the GCV criterion chooses $\lambda = 5$
- ▶ if we assume $\beta_j \sim N(0, \tau^2)$, independently of each other, and $y \mid \beta \sim N(X\beta, \sigma^2 I)$ (as usual) then

$$\beta \mid y \sim N(\quad , \quad)$$

and $\hat{\beta}_{ridge}$ is the posterior mean, with $\lambda = \sigma^2 / \tau^2$

Methods for classification (Chapter 4)

- ▶ inputs X_1, \dots, X_p
- ▶ output Y takes values in one of K classes
- ▶ output G is a group label (taking one of K values)
- ▶ data $(x_i, g_i), i = 1, \dots, N$
- ▶ goal to learn a model to predict the correct class for a future output, based on inputs
- ▶ instead of $\min \Sigma (y_i - \hat{y}_i)^2$, now try to minimize the *misclassification rate*

Special case: two classes

- ▶ data (x_i, g_i) , $g_i = 1, 2$; equivalently (x_i, y_i) , $y_i = 0, 1$
- ▶ natural starting point is Bernoulli distribution for y_i : $y_i = 1$ with probability $p_i = p_i(\beta)$
- ▶ likelihood function

$$L(\beta) = \prod_{i=1}^N p_i^{y_i} (1 - p_i)^{1-y_i}$$

- ▶ log-likelihood

$$\ell(\beta) = \sum_{i=1}^N y_i \log p_i(\beta) + (1 - y_i) \log\{1 - p_i(\beta)\}$$

- ▶ A common choice for $p_i(\beta)$ is the *logistic function*

$$p_i(\beta) = \frac{\exp(\beta^T x_i)}{1 + \exp(\beta^T x_i)}$$

x_i is a column vector here, i.e. the i th row of X is x_i^T

Logistic regression (§4.4)

- ▶ log-likelihood

$$\ell(\beta) = \sum_{i=1}^N y_i \beta^T \mathbf{x}_i - \log(1 + e^{\beta^T \mathbf{x}_i})$$

- ▶ Maximum likelihood estimate of β :

$$\frac{\partial \ell(\beta)}{\partial \beta} = 0 \iff \sum_{i=1}^N y_i \mathbf{x}_{ij} = \sum p_i(\hat{\beta}) \mathbf{x}_{ij}, j = 1, \dots, p$$

- ▶ Fisher information

$$-\frac{\partial^2 \ell(\beta)}{\partial \beta \partial \beta^T} = \sum_{i=1}^N \mathbf{x}_i \mathbf{x}_i^T p_i(1 - p_i)$$

- ▶ Fitting: use an iteratively reweighted least squares algorithm; equivalent to Newton-Raphson; p.99
- ▶ Inference: $\hat{\beta} \xrightarrow{d} N(\beta, \{-\ell''(\hat{\beta})\}^{-1})$
- ▶ Component: $\hat{\beta}_j \approx N(\beta_j, \hat{\sigma}_j^2)$ $\hat{\sigma}_j^2 = [\{-\ell''(\hat{\beta})\}^{-1}]_{jj}$; gives a t -test (z-test) for each component

- ▶ $2\{\ell(\hat{\beta}) - \ell(\beta_j, \tilde{\beta}_{-j})\} \approx \chi_{\dim\beta_j}^2$; in particular for each component get a χ_1^2 , or equivalently
- ▶ $\text{sign}(\hat{\beta}_j - \beta_j)\sqrt{2\{\ell(\hat{\beta}) - \ell(\beta_j, \tilde{\beta}_{-j})\}} \approx N(0, 1)$
- ▶ To compare 2 models $M_0 \subset M$ can use this twice to get $2\{\ell_M(\hat{\beta}) - \ell_{M_0}(\tilde{\beta}_q)\} \approx \chi_{p-q}^2$ which provides a test of the adequacy of M_0
- ▶ LHS is the difference in (residual) deviances; analogous to SS in regression
- ▶ See Ch. 14 of 302 text, and algorithm on p.99 of HTF.