Lecture 04. Model Selection and Regularization (Chapter 6 and Section 5.1)

Ping Yu

HKU Business School
The University of Hong Kong

Introduction

- Recall the linear regression model,

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon.$$

- In the lectures that follow, we consider some approaches for extending the linear model framework.
- In Lecture 7, we generalize the linear model in order to accommodate non-linear, but still additive, relationships.
- In Lectures 8 and 10, we consider even more general non-linear models.
- Despite its simplicity, the linear model has distinct advantages in terms of its interpretability and often shows good predictive performance.
- Hence we discuss in this lecture some ways in which the simple linear model can be improved, by replacing ordinary least squares fitting with some alternative fitting procedures.

Why consider alternatives to Least Squares?

- Prediction Accuracy: When n is not much larger than p, there can be a lot of variability in the least squares fit, resulting in overfitting and consequently poor predictions; when $p > n$, there is no unique least squares estimate: the variance is infinite! By constraining or shrinking the estimated coefficients, we can substantially reduce the variance at the cost of a negligible increase in bias.
- Model Interpretability: By removing irrelevant features – that is, by setting the corresponding coefficient estimates to zero – we can obtain a model that is more easily interpreted. We will present some approaches for automatically performing feature selection or variable selection.

Three Classes of Methods

- Subset Selection: We identify a subset of the p predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.
  - We will discuss two such procedures: best subset selection and stepwise selection.
- Shrinkage: We fit a model involving all p predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as regularization) has the effect of reducing variance, and some shrinkage methods, e.g., the lasso, can also perform variable selection.
  - We will discuss two most popular shrinkage methods: ridge regression and the lasso (least absolute shrinkage and selection operator).
  - Since there is a tuning parameter to be selected for our two shrinkage methods, we will also discuss a general method for such a purpose – cross validation.

Continued

- Dimension Reduction: We project the p predictors into a M-dimensional subspace, where $M < p$. This is achieved by computing M different linear combinations, or projections, of the variables. Then these M projections are used as predictors to fit a linear regression model by least squares.
  - We will concentrate on two dimension reduction methods – principal components regression and parial least squares, but will postpone the discussions to Lecture 5.
- The concepts discussed in this lecture can also be applied to other methods (besides linear regression), e.g., the classification methods in Lecture 2.
  - The deviance – negative two times the maximized log-likelihood – plays the role of RSS for a broader class of models.

(**) Alternative Definition of Deviance – Standardized Deviance

- Suppose under the parameter value θ, the mean of y is μ which is a function of θ; then the standardized deviance is defined as

$$-2\left[l\left(\widehat{\mu};\mathbf{y}\right)-l\left(\mathbf{y};\mathbf{y}\right)\right],$$

where $l\left(\mu;\mathbf{y}\right)=\sum_{i=1}^{n}l\left(\mu_i;y_i\right)$ with $l\left(\mu;y\right)=\ln f\left(y;\theta\right)$, and $\widehat{\mu}=\left(\hat{\mu}_1,\cdots,\hat{\mu}_n\right)^T$ is the ML estimator of μ (deduced from the ML estimator of θ).
- In other words, the deviance is (-2 times) the difference between the log likelihood of the assumed model and the saturated model (a model with a parameter for every observation so that the data are fitted exactly).

- Normal: If $y\sim N\left(\mu,\sigma^2\right)$ with known σ^2, then $\theta=\mu$, and $f\left(y;\mu\right)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$, so that

$$l\left(\mu;y\right)=-\frac{1}{2}\log\left(2\pi\sigma^2\right)-\frac{\left(y-\mu\right)^2}{2\sigma^2}.$$

Setting $\mu=y$ gives $l\left(y;y\right)=-\frac{1}{2}\log\left(2\pi\sigma^2\right)$ and so

$$-2\left[l\left(\hat{\mu};y\right)-l\left(y;y\right)\right]=\frac{\left(y-\hat{\mu}\right)^2}{\sigma^2}.$$

(**) Continued: Two More Examples

- Binary: If $y \sim$Bernoulli(p), then $\mu = p$ and $f(y;\mu) = \mu^y (1-\mu)^{1-y}$, so that

  $$l(\mu;y) = y\log\mu + (1-y)\log(1-\mu),$$

  and

  $$-2\left[l(\hat{\mu};y) - l(y;y)\right] = 2\left[y\log\frac{y}{\hat{\mu}} + (1-y)\log\frac{1-y}{1-\hat{\mu}}\right].$$

- Poisson: If $y \sim$Poisson(λ), then $\mu = \lambda$ and $f(y;\mu) = e^{-\lambda}\lambda^y/y!$, so that

  $$l(\mu;y) = -\lambda + y\log\lambda - \log y!,$$

  and

  $$-2\left[l(\hat{\mu};y) - l(y;y)\right] = 2\left[y\log\frac{y}{\hat{\mu}} - (y-\hat{\mu})\right].$$

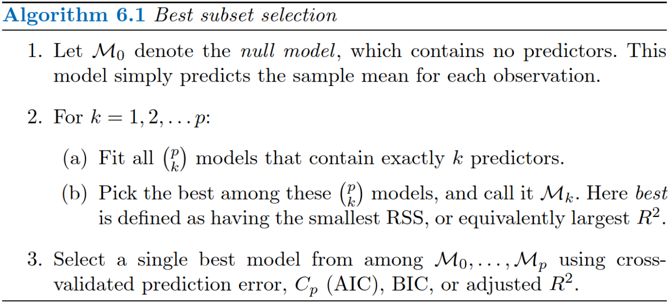
- In all three examples, $\hat{\mu}$ is the sample mean.

Subset Selection

(Section 6.1)

Best Subset Selection

- There are totally $C_0^p + C_1^p + \cdots + C_p^p = 2^p$ possible models! So selecting the best model is not trivial, and we break the procedure into two stages:



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 $R^2$.
```

- Step 2 reduces the number of possible models from 2^p to $p + 1$. [see Figure 6.1 for the Credit dataset, $n = 400$, $p = 10$]
- In Step 3, we cannot use RSS or R^2 to choose the best model since they represent the training error not the testing error, and are decreasing in p so always the largest model is selected.

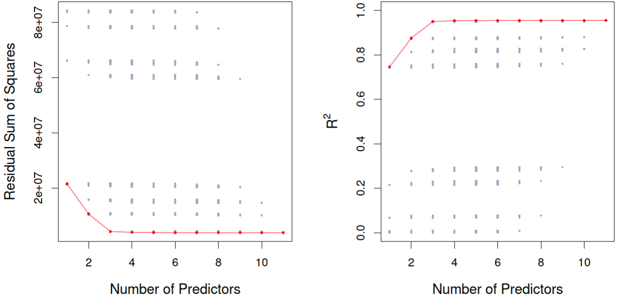


FIGURE 6.1. *For each possible model containing a subset of the ten predictors in the* Credit *data set, the RSS and R^2 are displayed. The red frontier tracks the* best *model for a given number of predictors, according to RSS and R^2. Though the data set contains only ten predictors, the x-axis ranges from* 1 *to* 11*, since one of the variables is categorical and takes on three values, leading to the creation of two dummy variables.*

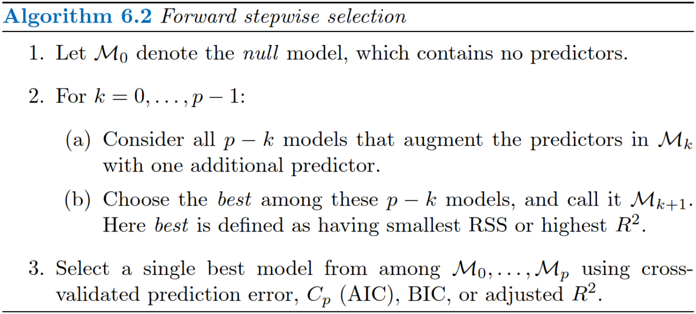
- Totally $2^{11} = 2048$ models.
- When the number of predictors is greater than 3, RSS or R^2 barely improves.

Stepwise Selection

- The key drawback of best subset selection is its computational limitation; usually, $p > 40$ is infeasible.
- Best subset selection may also suffer from statistical problems when p is large: larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data.
- Thus an enormous search space can lead to overfitting and high variance of the coefficient estimates.
- For both of these reasons, stepwise methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

Forward Stepwise Selection

- Starts from the null model; at each step the variable that gives the greatest additional improvement to the fit is added to the model, until all of the predictors are in the model.



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.

Comments

- Totally, only $1+p+(p-1)+\cdots+1=1+C_2^{p+1}$ models are considered,[1] so computational advantage over best subset selection is clear.
- It is not guaranteed to find the best possible model out of all 2^p models containing subsets of the p predictors.
  - Why not? Give an example.

# Variables	Best subset	Forward stepwise
One	rating	rating
Two	rating, income	rating, income
Three	rating, income, student	rating, income, student
Four	cards, income student, limit	rating, income, student, limit

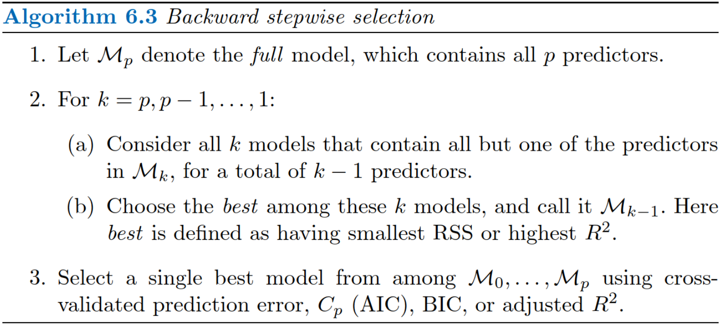
TABLE 6.1. *The first four selected models for best subset selection and forward stepwise selection on the* Credit *data set. The first three models are identical but the fourth models differ.*

- Figure 6.1 shows that there is little difference between the three- and four-variable models in terms of RSS, so either of the four-variable models will likely be adequate.
- This is a greedy approach, and might include variables early that later become redundant; hybrid selection below can remedy this.

[1] (**) If $n<p$, only $\mathscr{M}_0,\cdots,\mathscr{M}_{n-1}$ would be considered, but the number of possible models is much larger than $1+C_2^{p+1}$, which is so even when $n>p$ since usually a guided search is performed.

Backward Stepwise Selection

- Unlike forward stepwise selection, backward stepwise selection begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time, until the null model is achieved.



Algorithm 6.3 *Backward stepwise selection*

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

- Comments on forward stepwise selection still apply here.[2]
- Stepwise selection improves prediction accuracy only in certain cases, such as when only a few covariates have a strong relationship with the outcome.

[2] (**) If $n < p$, backward stepwise selection cannot apply, so forward stepwise selection is the only viable subset method when p is very large.

(*) Hybrid/Mixed Approaches

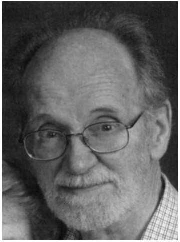
- This is a combination of forward and backward selection.
- We start with no variables in the model, and as with forward selection, we add the variable that provides the best fit.
- We continue to add variables one-by-one, but the *p*-values for existing variables can become larger as new predictors are added to the model.
- Hence, if at any point the *p*-value for one of the variables in the model rises above a certain threshold, then we remove that variable from the model.
- We continue to perform these forward and backward steps until all variables in the model have a sufficiently low *p*-value, and all variables outside the model would have a large *p*-value if added to the model.
- We know the best subset, forward stepwise, and backward stepwise selection approaches generally give similar but not identical models; the hybrid approach attempts to more closely mimic best subset selection while retaining the computational adavantages of forward and backward stepwise selection.

Choosing the Optimal Model

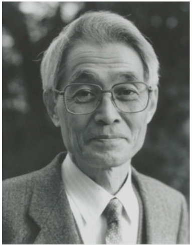
- As mentioned above, RSS and R^2 are not suitable for selecting the best model among a collection of models with different numbers of predictors because they are related to the training error, not the test error.
- To select the best model with respect to test error, we estimate this test error based on two common approaches:

1. We can indirectly estimate test error by making an adjustment to the training error to account for the bias due to overfitting.
   - C_p (or Mallow's C_p), Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC or the Schwarz criterion), and adjusted R^2 (or $\bar{R}^2$). [figure here]
   - All criteria are popular, but the adjusted R^2 is not as well motivated in statistical theory as the other three.
   - AIC and BIC can be defined for more general types of models.
2. We can directly estimate the test error.
   - A validation set approach or a cross-validation approach, as will be discussed later in this lecture.

History of C_p, AIC, BIC, and $\bar{R}^2$

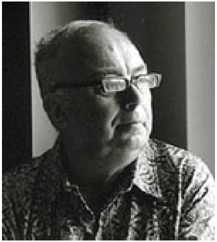


Colin L. Mallow (1930-), Bell Labs



Hirotugu Akaike (1927-2009), ISM

Gideon E. Schwarz (1933-2007), Hebrew



Henri Theil (1924-2000), Chicago and Florida

C_p, AIC, BIC, and $\bar{R}^2$

- Figure 6.2 displays C_p, BIC, and adjusted R^2 for the best model of each size produced by best subset selection on the Credit dataset.

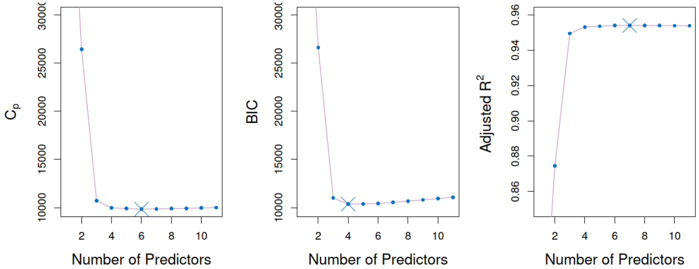


FIGURE 6.2. *C_p, BIC, and adjusted R^2 are shown for the best models of each size for the* `Credit` *data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.*

Details on C_p

- Mallows's C_p:

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

where d is the total $\#$ of predictors used, and $\hat{\sigma}^2$ is an estimate of the variance of the error ε.
- Note that $\text{RSS}(d)$ depends on d – it decreases with d.
- Typically, $\hat{\sigma}^2$ is estimated using the full model containing all predictors.
- The penalty term $2d\hat{\sigma}^2$ is added since the training error tends to underestimate the test error; it is increasing in d to adjust for the corresponding decrease in training RSS.
-(**) If $\hat{\sigma}^2$ is unbiased, then C_p is an unbiased estimate of test MSE.
- In Figure 6.2, minimizing C_p obtains 6 predictors, income, limit, rating, cards, age and student.

Details on AIC

- AIC: defined for a large class of models fit by maximum likelihood,

  $$\text{AIC} = -2\log L(d) + 2d,$$

  which is equivalent to C_p in linear regression with Gaussian errors, wher $L(d)$ is the maximized likelihood function with d predictors.
  - This is because

  $$-2\log L(d) = \text{constant} + n\log\hat{\sigma}^2,$$

  and letting $\hat{\sigma}^2 =$RSS$(d)/n$ in C_p, we have

  $$C_p = \frac{n+2d}{n}\hat{\sigma}^2$$

  such that

  $$\log C_p = \log\hat{\sigma}^2 + \log\left(1+\frac{2d}{n}\right) \approx \log\hat{\sigma}^2 + \frac{2d}{n} \approx \frac{\text{AIC}}{n}.$$

Details on BIC

- BIC: in linear regression with Gaussian errors,

$$\text{BIC} = \frac{1}{n}\left(\text{RSS}(d) + \log(n)\, d\hat{\sigma}^2\right).$$

  - BIC replaces the $2d\hat{\sigma}^2$ used by C_p with a $\log(n)\, d\hat{\sigma}^2$ term.
  - Since $\log n > 2$ for any $n > 7$, the BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than C_p.
  - In Figure 6.2, only income, limit, cards, and student are selected, but the four-variable and six-variable models do not have much difference in accuracy.

Details on $\bar{R}^2$

- Adjusted R^2:

$$\bar{R}^2 = 1 - \frac{\text{RSS}(d)/(n-d-1)}{\text{TSS}/(n-1)}.$$

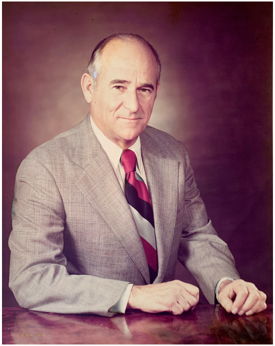
  - Different from C_p, AIC and BIC, a large value of $\bar{R}^2$ is preferred.
  - Maximizing $\bar{R}^2$ is equivalent to minimizing $\frac{\text{RSS}(d)}{n-d-1}$. While $\text{RSS}(d)$ always decreases as d increases, $\frac{\text{RSS}(d)}{n-d-1}$ may increase or decrease, due to the presence of d in the denominator.
  - Unlike the R^2 statistic, the $\bar{R}^2$ statistic pays a price for the inclusion of unnecessary variables in the model.
  - It can be shown that maximizing $\bar{R}^2$ is approximately equivalent to minimizing $\frac{1}{n}\left(\text{RSS}(d) + d\hat{\sigma}^2\right)$. [Exercise]
  - In Figure 6.2, besides the six variables selected by C_p and AIC, own is also selected.

- Why select among only $\mathscr{M}_0, \cdots, \mathscr{M}_p$?
  - For the same d (and $\hat{\sigma}^2$), optimizing over the C_d^p models in the four criteria is the same as minimizing over the corresponding RSS.

Shrinkage Methods

(Section 6.2)

History of Ridge Regression



Arthur E. Hoerl (1921-1994, UDelaware)

Robert W. Kennard (1923-2011, DuPont)

- Hoerl, A.E., and R.W. Kennard, 1970, Ridge Regression: Biased Estimation for Nonorthogonal Problems, *Technometrics*, 12, 55–67.
- Hoerl, A.E., and R.W. Kennard, 1970, Ridge Regression: Applications to Nonorthogonal Problems, *Technometrics*, 12, 69–82.

Ridge Regression

- Recall that the least squares fitting procedure estimates $\beta_0, \beta_1, \cdots, \beta_p$ using the values that minimize

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

- In contrast, the ridge regression (RR for short) coefficient estimates $\hat{\beta}^R$ are the values that minimize

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \left\| \beta \right\|_2^2 , \tag{1}$$

where $\lambda \geq 0$ is a tuning parameter, to be determined separately, and $\|\cdot\|_2$ is the Enclidean norm or ℓ_2 norm.
- Note that β_0 is not penalized, and is equal to $\bar{y}$ when the columns of $\mathbf{X}$ are normalized to have mean 0; in other words, we care about the sensitivity of Y with respect to X_j, but not its own level.

Continued

- As with least squares, RR seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda \sum_{j=1}^{p} \beta_j^2$, called a shrinkage penalty, is small when $\beta_1, \cdots, \beta_p$ are close to zero, and so it has the effect of shrinking the estimates of β_j towards zero.
- The tuning parameter λ serves to control the relative impact of these two terms on the regression coefficient estimates.
- Selecting a good value for λ is critical; cross-validation discussed below is used for this.
- RR has substantial computational advantages over best subset selection – only one (rather than 2^p) model is fit for each λ.
  - Actually, computations required to solve (1), simultaneously for all values of λ, are almost identical to those for fitting a model using least squares.
- Different from least squares, RR can be conducted even if $p > n$.

[Example] Credit

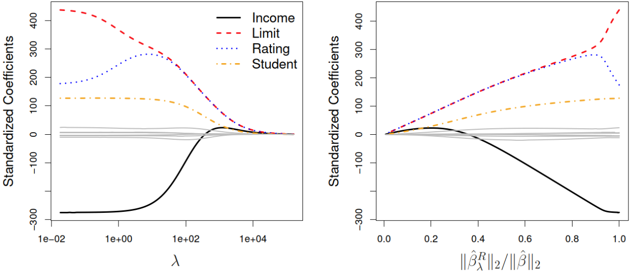


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* Credit *data set, as a function of λ and $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$.*

- $\hat{\beta}_\lambda^R$ is a function of λ: $\lambda \to 0$, $\hat{\beta}_\lambda^R \to \hat{\beta}_{LS}$, and $\lambda \to \infty$, $\hat{\beta}_\lambda^R \to \left(\widehat{\beta}_0, 0, 0, \cdots, 0\right)^T$, the null model.
- $\hat{\beta}_\lambda^R$ is generally decreasing in λ, but need not be strictly so; anyway, $\left\|\hat{\beta}_\lambda^R\right\|_2$ must be decreasing in λ, so there is a one-to-one correspondence between $\lambda \in (0, \infty)$ and $\left\|\hat{\beta}_\lambda^R\right\|_2 / \left\|\hat{\beta}_{LS}\right\|_2 \in (1, 0)$ as shown in the right panel of Figure 6.4 .

Scaling of Predictors

- $\hat{\beta}_{LS}$ is scale equivariant: if $X_j \to cX_j$ then $\hat{\beta}_{j,LS} \to \hat{\beta}_{j,LS}/c$ such that $X_j\hat{\beta}_{j,LS}$ remains the same.
- In contrast, $\hat{\beta}^R_{j,\lambda}$ can change substantially and need not change to $\hat{\beta}^R_{j,\lambda}/c$ when $X_j \to cX_j$, due to the equal weights on β_j^2 in $\sum_{j=1}^p \beta_j^2$, such that $X_j\hat{\beta}^R_{j,\lambda}$ would change.

  - Actually, $X_j\hat{\beta}^R_{\lambda,j}$ depends also on the scaling of the other predictors.
- Therefore, it is best to apply RR after standardizing the predictors:

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\frac{1}{n}\sum_{i=1}^n \left(x_{ij} - \bar{x}_j\right)^2}},$$

  i.e., use the z-score of x_{ij}.

  - For any j, $\left(\tilde{x}_{1j}, \cdots, \tilde{x}_{nj}\right)^T$ has mean 0 and standard deviation 1.

  - "standardized coefficients" in Figure 6.4 are generated in this way; now, $\hat{\beta}^R_{\lambda,j}$ has the same meaning – the average change in y when x_j changes one standard deviation.

Why Does RR Improve Over LS?

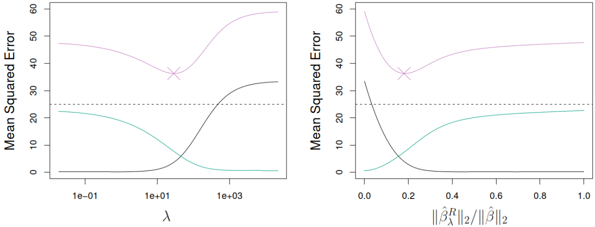
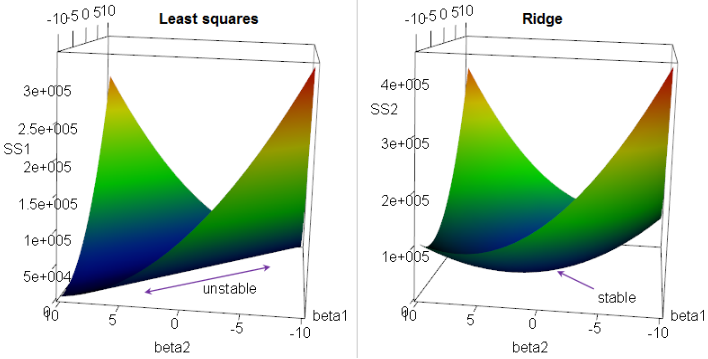


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of λ and $\|\hat\beta^R_\lambda\|_2/\|\hat\beta\|_2$. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

- Simulated data with $n = 50$ observations, $p = 45$ predictors, all having nonzero coefficients.
- RR's advantage over LS $(\lambda = 0)$ is rooted in the bias-variance trade-off.
- The optimal $\lambda \approx 30$; due to the high variance of $\hat\beta_{LS}$ (because p is close to n), MSE of LS $\sim$ MSE of $\lambda = \infty$; RR works best in situations where $\hat\beta_{LS}$ has high variance (e.g., multicollinear, or p is large).

Why the name "Ridge Regression"?

- The solution to (1) is $\hat{\beta}^R = \left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y}$, where a ridge parameter λ of the identify matrix is added to the correlation matrix $\mathbf{X}^T\mathbf{X}$ (when $\mathbf{X}$ is standardized), while the diagonal of ones in the correlation matrix can be described as a "ridge".
- Historically, "ridge" actually refers to the characteristics of the function we were attempting to optimize, rather than to adding a "ridge" to the $\mathbf{X}^T\mathbf{X}$ matrix.



History of the Lasso

Robert Tibshirani (1956-, Stanford)[3]

- Tibshirani, R., 1996, Regression Shrinkage and Selection via the lasso, *Journal of the Royal Statistical Society, Series B*, 58, 267-88.

[3] previoiusly at Toronto, COPSS Award receiver of 1996, inventor of LASSO, and a student of Efron.

The Lasso

- RR does have one obvious disadvantage: unlike subset selection, which will generally select models that involve just a subset of the variables, RR will include all p predictors in the final model (unless $\lambda = \infty$).
  - This is not a problem for prediction accuracy, but a challenge in model interpretation when p is large.
- The Lasso is a relatively recent alternative to RR that overcomes this disadvantage. The lasso coefficients, $\hat{\beta}_\lambda^L$, minimize the quantity

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\left|\beta_j\right| = \text{RSS}+\lambda\left\|\beta\right\|_1, \tag{2}$$

where $\|\cdot\|_1$ is the ℓ_1 norm, i.e., the lasso uses an ℓ_1 penalty instead of an ℓ_2 penalty.

Continued

- As with RR, the lasso shrinks the coefficient estimates towards zero, so can reduce variance at the expense of a small increase in bias.
- However, in the case of the lasso, the ℓ_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large.
- Hence, much like best subset selection, the lasso performs variable selection, and the resulting model is much easier to interpret than that in RR.
- We say that the lasso yields sparse models – that is, models that involve only a subset of the variables.
- As in RR, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.
- Like RR, the entire coefficient paths can be computed with about the same amount of work as a single least squares fit.

[Example] Credit

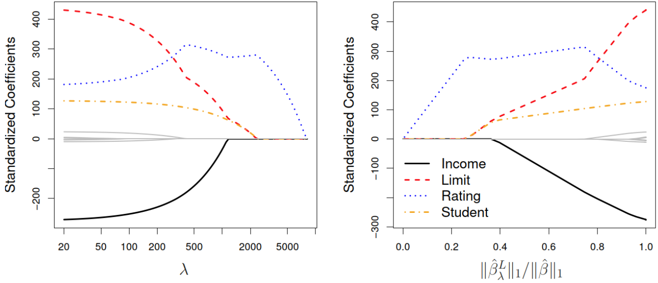


FIGURE 6.6. *The standardized lasso coefficients on the* Credit *data set are shown as a function of* λ *and* $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$.

- The limits of $\hat{\beta}^L_\lambda$ as $\lambda \to 0$ and ∞ are the same as in $\hat{\beta}^R_\lambda$; $\hat{\beta}^L_\lambda$ is also generally decreasing in λ and $\left\|\hat{\beta}^L_\lambda\right\|_1$ must be decreasing in λ.
- In between, i.e., $\lambda \in (0,\infty)$, $\hat{\beta}^L_\lambda$ and $\hat{\beta}^R_\lambda$ are very different: rating, student, limit, and income enter the model sequentially, so any number of variables is possible, while RR involves all variables.

(**) More Properties of the Lasso Coefficient Path

- There is a finite sequence of λ's

$$\lambda_0 > \lambda_1 > \lambda_2 > \cdots > \lambda_K = 0$$

such that

1. For all $\lambda > \lambda_0$, $\hat{\beta}^L_\lambda = \mathbf{0}$.
2. In the interior of the interval $(\lambda_{m+1}, \lambda_m)$, the active set $\mathscr{B}(\lambda) := \{j | \hat{\beta}^L_{\lambda j} \neq 0\}$ and the sign vector of the active $\hat{\beta}^L_{\lambda j}$'s are constant with respect to λ, so can be denoted as $\mathscr{B}_m$ and sign_m.
3. When λ decreases from $\lambda = \lambda_m -$, some predictors with zero coefficients at λ_m are about to have nonzero coefficients; as λ approaches $\lambda_m +$ there are possibly some predictors in $\mathscr{B}_m$ whose coefficients reach zero.

- From the Kuhn-Tucker conditions, we have the equiangular conditions:

$$\begin{aligned} \lambda &= 2|\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\hat{\beta}^L_\lambda)|, \forall j \in \mathscr{B}(\lambda), \\ \lambda &> 2|\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\hat{\beta}^L_\lambda)|, \forall j \notin \mathscr{B}(\lambda), \end{aligned}$$

which imply that the coefficient path for the lasso is continuous and piecewise linear over the entire range of λ, with the transition points $\{\lambda_m\}$ occurring whenever the active set changes, or the signs of the coefficients change.

The Variable Selection Property of the Lasso

- Why is it that the lasso, unlike RR, results in coefficient estimates that are exactly equal to zero?
- One can show that the lasso and RR coefficient estimates solve the problems

$$\min_{\beta}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \text{ subject to } \sum_{j=1}^{p}\left|\beta_j\right|\le s$$

and

$$\min_{\beta}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\} \text{ subject to } \sum_{j=1}^{p}\beta_j^2\le s,$$

respectively. [see Figure 6.7 for $p=2$]

- λ is one-to-one and inversely related to s.

- s is like a budget that can be spent by $\left|\beta_j\right|$'s or β_j^2's.

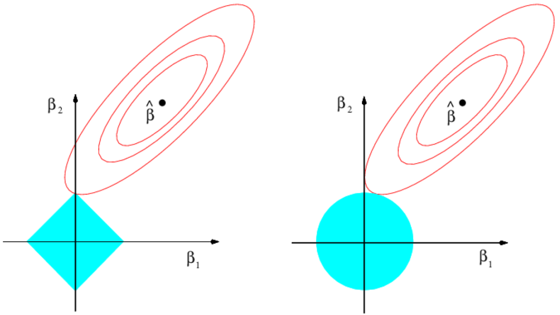


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \le s$ and $\beta_1^2 + \beta_2^2 \le s$, while the red ellipses are the contours of the RSS.*

- Best subset selection among s-variable models can be reformulated in a similar manner:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} I\left(\beta_j \neq 0\right) \le s,$$

so RR and lasso provide computationally feasible alternatives to best subset selection by replacing the form of budget or through convexification.

Comparing the Lasso and RR

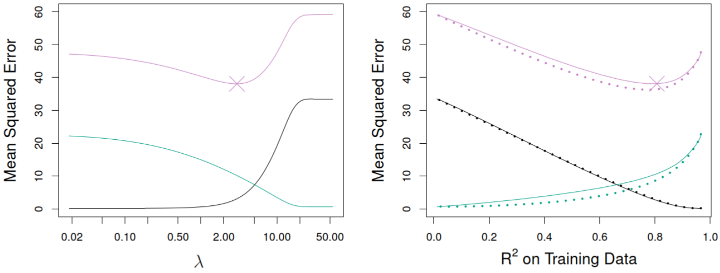


FIGURE 6.8. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on a simulated data set.* Right: *Comparison of squared bias, variance, and test MSE between lasso (solid) and ridge (dotted). Both are plotted against their R^2 on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

- Same simulated data as in Figure 6.5.
- Lasso and RR have similar biases, but RR has a slightly lower variance than lasso.

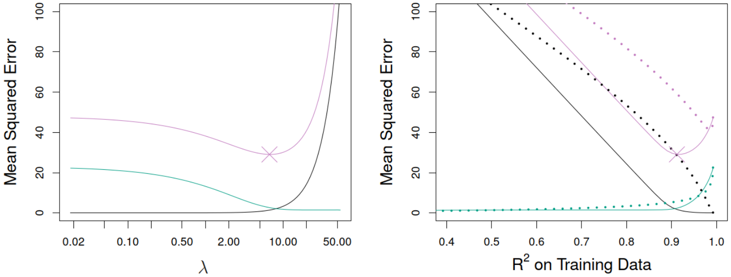


FIGURE 6.9. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Figure 6.8, except that now only two predictors are related to the response.* Right: *Comparison of squared bias, variance, and test MSE between lasso (solid) and ridge (dotted). Both are plotted against their R^2 on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.*

- A design favorable to lasso: 2 out of 45 (rather than all) predictors contribute to y.
- Lasso outperforms RR in terms of bias, variance, and MSE.

Conclusions and Discussions

- These two examples illustrate that neither RR nor the lasso will universally dominate the other.
- In general, the lasso is suitable for sparse models with a small number of substantial coefficients, while RR is suitable for models involving many predictors, all with roughly equal coefficients.
- However, the number of predictors that is related to the response is never known a priori for real datasets.
- A technique such as cross-validation can be used in order to determine which approach is better on a particular dataset.

- (**) For extensions of the lasso, see Appendix A.
- (**) For a Bayesian interpretation of RR and the lasso, see Appendix B.

- A popular package of **R** used for RR and the lasso is glmnet.
- Another popular package of **R** used for the lasso is gamlr.
- Both packages run the lasso in a similar way, and the difference lies in what they can do beyond lasso, e.g., gamlr can conduct Gamma-Lasso in Appendix B which includes the lasso as a special case .

A Simple Special Case for RR and Lasso

- $n = p$, $\mathbf{X} = \mathbf{I}$, and $\beta_0 = 0$ is known.
- LS:

$$\min_{\beta} \sum_{j=1}^{p} \left(y_j - \beta_j\right)^2 \Longrightarrow \hat{\beta}_j = y_j.$$

- RR:

$$\min_{\beta} \sum_{j=1}^{p} \left(y_j - \beta_j\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \Longrightarrow \hat{\beta}_j^R = \frac{y_j}{1+\lambda}.$$

  - In Figure 6.10, each $\hat{\beta}_j$ is shrunken by the same proportion.
- Lasso:

$$\begin{aligned}\min_{\beta} \sum_{j=1}^{p} \left(y_j - \beta_j\right)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right| \quad &\Longrightarrow \quad \hat{\beta}_j^L = \begin{cases} y_j - \lambda/2, & \text{if } y_j > \lambda/2, \\ y_j + \lambda/2, & \text{if } y_j < -\lambda/2, \\ 0, & \text{if } \left|y_j\right| \le \lambda/2. \end{cases} \\ &= \quad \operatorname{sign}(y_j)\left(\left|y_j\right| - \lambda/2\right)_+ = \left(1 - \frac{\lambda/2}{\left|y_j\right|}\right)_+ y_j.\end{aligned}$$

  - In Figure 6.10, each $\hat{\beta}_j$ is shrunken toward zero by a constant amount, $\lambda/2$, known as soft-thresholding.[4]

[4] A hard-thresholding is like $\hat{\beta}_j^H = \hat{\beta}_j I(|\hat{\beta}_j| > \sqrt{2\lambda})$ as in the best subset selection, where the ℓ_1 penalty of Lasso is changed to the ℓ_0 penalty, $\lambda \sum_{j=1}^{p} \left|\beta_j\right|^0$, i.e., counting the number of nonzero elements in β. [Exercise]

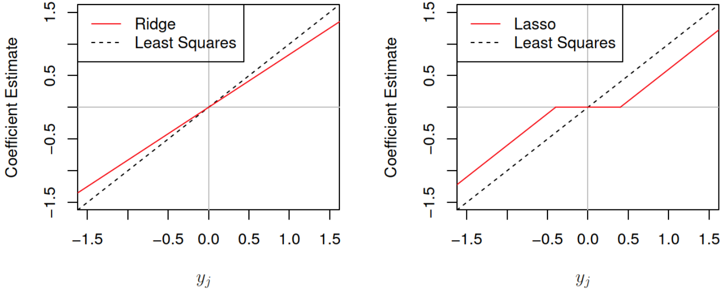


FIGURE 6.10. *The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and $\mathbf{X}$ a diagonal matrix with 1's on the diagonal.* Left: *The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates.* Right: *The lasso coefficient estimates are soft-thresholded towards zero.*

- This example is special since $\hat{\beta}_j^R$ is linearly shrunken, so sign$\left(\hat{\beta}_j^R\right)$ $=$sign$\left(\hat{\beta}_j\right)$, while in general (especially in the multicollinear case), sign$\left(\hat{\beta}_j^R\right)$ can be reversed [see Figure 6.4].
- Although typically, once $\hat{\beta}_j^L$ hits zero it stays there (just as in this example), $\hat{\beta}_j^L$ can also reverse its sign. [Exercise]

Comparison Between the Subset Selection and Shrinkage Methods

- The subset selection methods are unstable in the sense that a small change in the data can make large changes in the sequence of $\mathscr{M}_0, \mathscr{M}_1, \cdots, \mathscr{M}_p$. On the other hand, the solutions of shrinkage methods are continuous in λ so are relatively stable.
  - (**) CART (indexed by the number of terminal nodes) in Lecture 8, single layer neural networks (indexed by the number of hidden units) in Lecture 10 and the ℓ_q penalty with $0 < q < 1$ in Appendix A are also unstable, while KNN and bagging in Lecture 8 are stable.[5]
  - The more unstable the procedure, the noiser the test error, and we are less able to locate the best model.
- The subset selection methods are sequential, i.e., first model selection and then estimation, while the shrinkage methods are simultaneous, conducting model selection and estimation together.
- The best subset selection can handle only tens of features, while the shrinkage methods can handle more than thousands of features.

[5] Recall that Clustering in Lecture 3 is also unstable in this sense, but it is an unsupervised learning method.

Cross Validation

(Section 5.1)

Resampling Methods

- Cross-validation (CV) is a resampling method; such methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.
- Resampling approaches can be computationally expensive, because they involve fitting the same statistical method multiple times using different subsets of the training data, but this is generally not prohibitive nowadays due to recent advances in computing power.
- We discuss two resampling methods in this course, the bootstrap besides CV. CV would be discussed in this lecture, and the bootstrap would be discussed in Lecture 8.
- CV can be used in both model assessment and model selection. The former means evaluating a model's performance, e.g., estimating the test error associated with a given statistical learning method, and the latter means selecting the proper level of flexibility for a model.

Training Error versus Test Error

- Recall the distinction between the test error and the training error.
- The test error is the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.
  - Best solution: a large designated test set. Often not available
- In contrast, the training error can be easily calculated by applying the statistical learning method to the observations used in its training.
- But the training error rate often is quite different from the test error rate, and in particular the former can dramatically underestimate the latter. [figure here]
- CV estimates the test error by holding out a subset of the training observations from the fitting process, and then applying the statistical learning method to those held out observations.

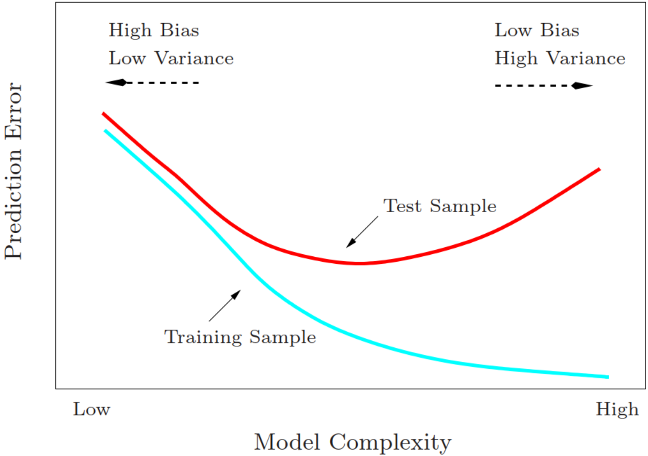


Figure: Test and training error as a function of model complexity.

The Validation Set Approach

- Here we randomly divide the available set of samples into two parts: a training set and a validation or hold-out set. [see Figure 5.1]

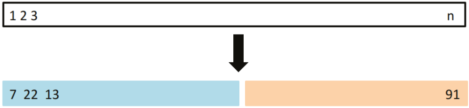


FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.
- The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response and misclassification rate in the case of a qualitative (discrete) response.

[Example] Auto

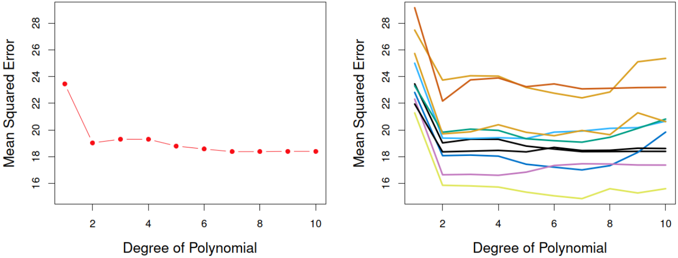


FIGURE 5.2. *The validation set approach was used on the* Auto *data set in order to estimate the test error that results from predicting* mpg *using polynomial functions of* horsepower. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

- $n = 392$, $n_{\text{training}} = n_{\text{validation}} = 196$. $\hat{p} = 2$ in the left panel, and $\hat{p}$ is not unique for different splittings (anyway, $\hat{p} \neq 1$ for all splittings).

Drawbacks of Validation Set Approach

1. The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.
2. In the validation approach, only a subset of the observations – those that are included in the training set rather than in the validation set – are used to fit the model.
   - Since statistical methods tend to perform worse when trained on fewer observations, this suggests that the validation set error may tend to overestimate the test error for the model fit on the entire data set.

- CV refines the validation set approach by addressing these two issues.

K-Fold Cross-Validation

- K-fold CV is a widely used approach for estimating test error.
- Estimates can be used to select the best model, and to give an idea of the test error of the final chosen model.
- The idea is to randomly divide the data into K equal-sized parts. We leave out part k, fit the model to the other $K-1$ parts (combined), and then obtain predictions for the left-out kth part.
- This is done in turn for each part $k=1,2,\cdots,K$, and then the results are combined. [see Figure 5.5]
- Let the K parts be C_1, C_2, $\cdots$, C_K, where C_k denotes the indices of the observations in part k. There are n_k observations in part k: if n is a multiple of K, then $n_k=n/K$.
- Compute

  $$\mathrm{CV}_K=\sum_{k=1}^{K}\frac{n_k}{n}\mathrm{MSE}_k\overset{n_k=n/K}{=}\frac{1}{K}\sum_{k=1}^{K}\mathrm{MSE}_k,$$

  where $\mathrm{MSE}_k=\sum_{i\in C_k}\left(y_i-\hat{y}_i\right)^2/n_k$, and $\hat{y}_i$ is the fit for observation i, obtained from the data with part k removed.
- Setting $K=n$ yields n-fold or leave-one out cross-validation (LOOCV). [see Figure 5.3]

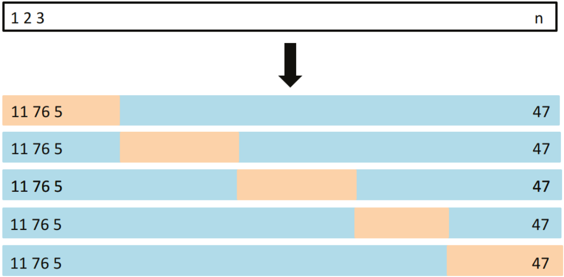


FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

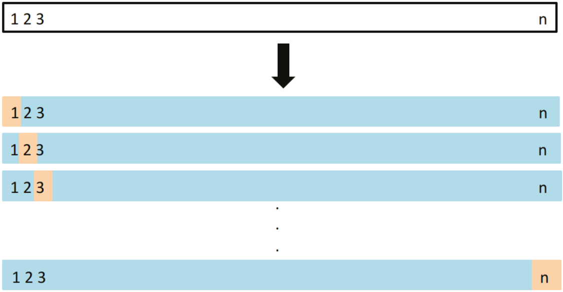


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

Bias-Variance Trade-Off for K-Fold Cross-Validation

- Why can K-fold CV (or LOOCV) avoid or mitigate the two drawbacks of the validation set approach?
  1. Since $K>2$, the variability of CV_K is typically much lower than that in the validation set approach.[6] [see the example below]
  2. Since each training set is only $(K-1)/K$ as big as the original training set, the estimates of prediction error will typically be biased upward but not as much as in the validation set approach where $K=2$.
- In LOOCV, $K=n$, the bias of prediction error is minimized and there is no variability at all in CV_n.
- However, LOOCV typically doesn't shake up the data enough. The estimates from each fold are highly (positively) correlated and hence their average can have high variance.[7]
- $K=5$ or 10 provides a good compromise for this bias-variance tradeoff.

[6] (**) Note that given the data, the variability of CV_K depends on both K and the number of possible splittings; the latter need not decrease with K, e.g., if $n=6$, then $\#_{K=2}=\frac{C_6^3}{2!}=10$, while $\#_{K=3}=\frac{C_6^2C_4^2C_2^2}{3!}=15>10$.

[7] (**) Note that the variance here is different from the variability of CV_K above: the former is from the random sampling of the original data from the population, while the later is conditional on the data, i.e., the data are fixed.

A Nice Special Case

- With least-squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as that of a single model fit! The following formula holds:

$$\text{CV}_n = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\hat{y}_i}{1-h_i}\right)^2,$$

  where $\hat{y}_i$ is the ith fitted value from the original least squares fit, and h_i is the leverage (diagonal of the "hat" matrix, reflecting the amount that an observation influences its own fit; in a simple linear regression, $h_i = \frac{1}{n} + \frac{(x_i-\bar{x})^2}{\sum_{i'=1}^{n}(x_{i'}-\bar{x})^2} \in \left[\frac{1}{n}, 1\right]$).

- This is like the ordinary MSE, except the ith residual is inflated by dividing $1-h_i$.
- As mentioned above, a better choice is $K=5$ or 10, which can also save computational time in a general machine learning method.

[Example] Auto Revisited

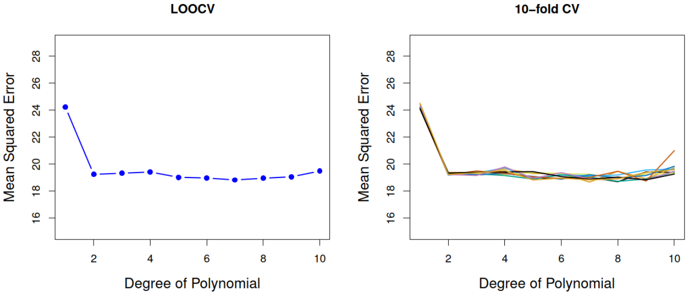


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10-*fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

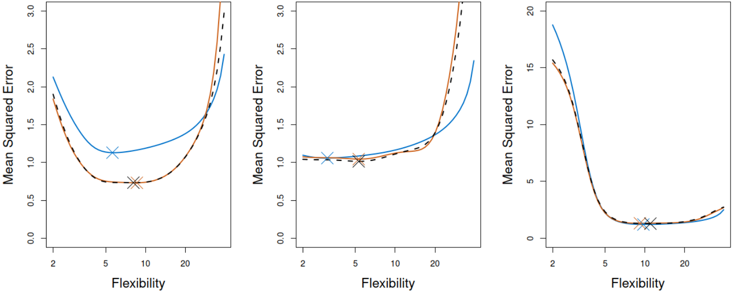


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9* (left)*, 2.10* (center)*, and 2.11* (right)*. The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.*

- Although CVs sometimes underestimate the true test MSE, all of the CV curves come close to identifying the correct level of flexibility – that is, the flexibility level corresponding to the smallest test MSE.

Comparison with C_p, AIC, BIC, and Adjusted R^2

- The validation set and cross-validation methods have an advantage relative to AIC, BIC, C_p, and adjusted R^2, in that it provides a direct estimate of the test error, and doesn't require an estimate of the error variance σ^2.
- It can also be used in a wider range of model selection tasks, even in cases where it is hard to pinpoint the model degrees of freedom (e.g., the number of predictors in the model) or hard to estimate the error variance σ^2.
- In Figure 6.3, the validation errors were calculated by randomly selecting three-quarters of the observations as the training set, and the remainder as the validation set; the cross-validation errors were computed using $K = 10$ folds.
  - In this case, the validation and cross-validation methods both result in a six-variable model, but all three approaches suggest that the four-, five-, and six-variable mdoels are roughly equivalent in their test errors.
- In this setting, we can select a model using the one-standard-error rule. We first calculate the standard error of the estimated test MSE for each model size (by using different randomly chosen training and validation sets), and then select the smallest or most parsimonious model for which the estimated test error is within one standard error of the lowest point on the curve.
  - The rationale here is that if a set of models appear to be more or less equally good, then we might as well choose the simplest model – that is, the model with the smallest number of predictors.

[Example] Credit

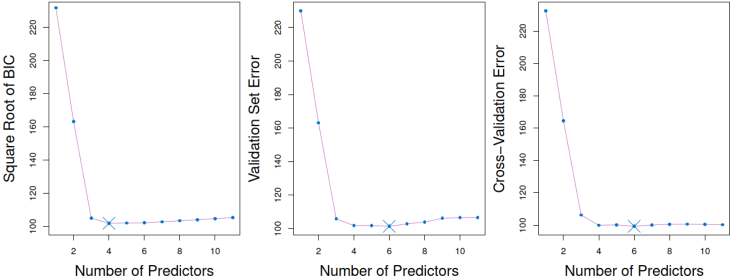


FIGURE 6.3. *For the* Credit *data set, three quantities are displayed for the best model containing d predictors, for d ranging from* 1 *to* 11. *The overall* best *model, based on each of these quantities, is shown as a blue cross.* Left: *Square root of BIC.* Center: *Validation set errors.* Right: *Cross-validation errors.*

- The one-standard-error rule leads to a three-variable model in the validation set or cross-validation approach.

Selecting Tuning Parameters in RR and Lasso

- As for subset selection, for RR and lasso we require a method to determine which of the models under consideration is best.
- That is, we require a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint s.
- Cross-validation provides a simple way to tackle this problem. We choose a grid of λ values, and compute the cross-validation error rate for each value of λ.
- We then select the tuning parameter value for which the cross-validation error is smallest.
- Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

[Example] RR for Credit

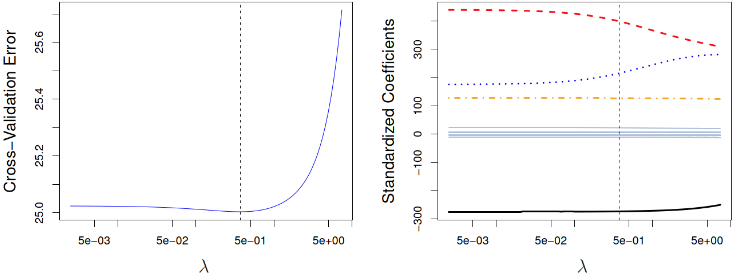


FIGURE 6.12. Left: *Cross-validation errors that result from applying ridge regression to the* Credit *data set with various values of* λ. Right: *The coefficient estimates as a function of* λ. *The vertical dashed lines indicate the value of* λ *selected by cross-validation.*

- The LOOCV λ is relatively small, so no much shrinkage relative to LS; the CV-error curve is flat around the λ; maybe simply use LS.

[Example] Lasso for Simulated Data

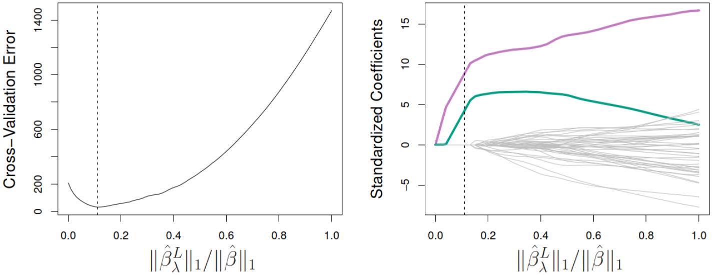


FIGURE 6.13. Left*: Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9.* Right*: The corresponding lasso coefficient estimates are displayed. The two signal variables are shown in color, and the noise variables are in gray. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

- 10-fold CV λ sieves out the two signal variables and discards all noise variables even if $p = 45$ and $n = 50$, while LS assigns a large coefficient estimate to only one of the two signal variables.

Cross-Validation on Classification Problems

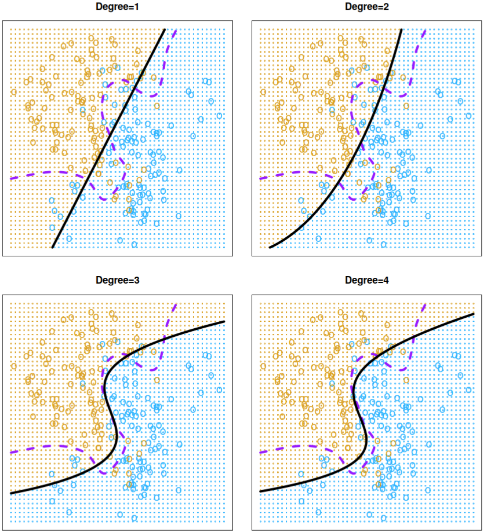
- For the same training/validation set split, compute

$$\mathrm{CV}_K=\sum_{k=1}^{K}\frac{n_k}{n}\mathrm{Err}_k\overset{n_k=n/K}{=}\frac{1}{K}\sum_{k=1}^{K}\mathrm{Err}_k,$$

  where $\mathrm{Err}_k=\sum_{i\in C_k}I\left(y_i\neq\hat{y}_i\right)/n_k$.
- The estimated standard deviation (or standard error) of CV_K is

$$\mathrm{SE}\left(\mathrm{CV}_K\right)=\sqrt{\frac{1}{K}\sum_{k=1}^{K}\frac{\left(\mathrm{Err}_k-\mathrm{CV}_K\right)^2}{K-1}}.$$

- This is a useful estimate, but strictly speaking, not quite valid. Why not? only between-fold variation, no within-fold variation.
- The following figure shows logistic regression fits using polynomials of orders $1-4$ on the two-dimensional classification data displayed in Figure 2.13, where the test error rates are $0.201, 0.197, 0.160$ and 0.162 respectively while the Bayes error rate is 0.133, and Figure 5.8 shows how to use CV_{10} to choose the order of polynomial.


Degree=1
Degree=2
Degree=3
Degree=4

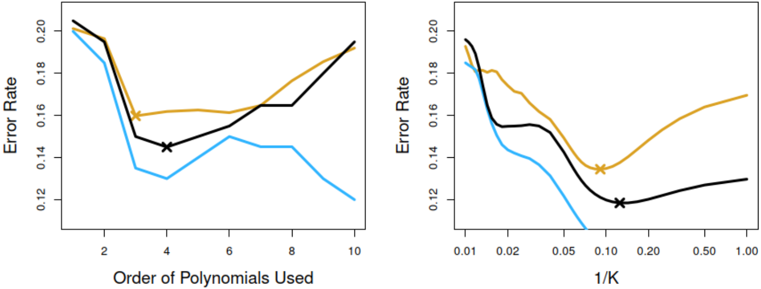


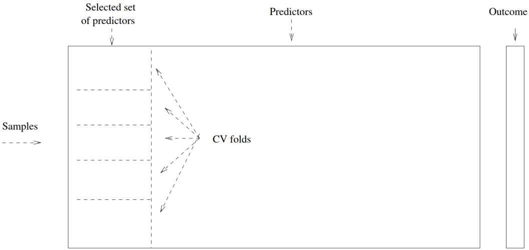
FIGURE 5.8. *Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.*

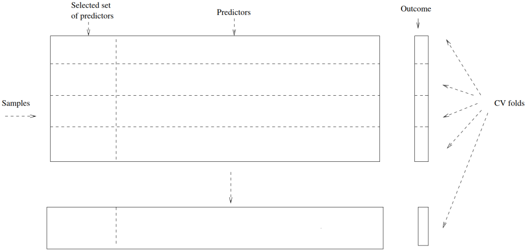
- Although the CV error curve slightly underestimates the test error rate, it takes on a miminum very close to the best value for the order of polynomial or the number of neighbors.

(*) Cross-Validation: Right and Wrong

- Consider a simple classifier applied to some two-class data:
  1. Starting with 5000 predictors and 50 samples, find the 100 predictors having the largest correlation with the class labels.
  2. We then apply a classifier such as logistic regression, using only these 100 predictors.
- How do we estimate the test set performance of this classifier?
- Can we apply cross-validation in Step 2, forgetting about Step 1?
- NO! This would ignore the fact that in Step 1, the procedure has already seen the labels of the training data, and made use of them. This is a form of training and must be included in the validation process.
- It is easy to simulate realistic data with the class labels independent of the outcome, so that true test error $=$ 50%, but the CV error estimate that ignores Step 1 is zero! Try to do this yourself
- With a multistep modeling steps, CV must be applied to the entire sequence of modeling steps.

(*) Wrong Way and Right Way





(**) Comparison Between Covariance Penalties and Cross-Validation

- Both C_p and AIC are covariance penalty methods.
- In C_p, $\mathbf{y} \sim \left(\mu, \sigma^2 \mathbf{I}\right)$, $L(y, \mu) = (y - \mu)^2$, and $\hat{\mu} = \mathbf{P}\mathbf{y}$ is a linear rule, where $\mathbf{P}$ is a function of covariates and always assumed fixed. The target is the prediction error

$$\text{Err} = \sum_{i=1}^{n} \text{Err}_i = \sum_{i=1}^{n} E_0 \left[L\left(y_i^0, \hat{\mu}_i \right) \right],$$

where E_0 is the expectation over $y_i^0 \sim \left(\mu_i, \sigma^2\right)$ independent of $\mathbf{y}$, with $\hat{\mu}_i$ held fixed. Since $\text{err}_i = L(y_i, \hat{\mu}_i)$ would under-estimate Err_i, we define the optimism

$$O_i = \text{Err}_i - \text{err}_i$$

and estimate O_i by

$$\Omega_i = E_{\mathbf{y}}[O_i],$$

where $E_{\mathbf{y}}$ is the expectation over $\mathbf{y}$ which appear in y_i and $\hat{\mu}_i$.
- It turns out that

$$\Omega_i = 2 \cdot \text{Cov}(\hat{\mu}_i, y_i) = 2\sigma^2 \mathbf{P}_{ii},$$

where $\mathbf{P}_{ii}$ is the ith diagonal element of $\mathbf{P}$.
- If $\hat{\mu}$ is the LSE, then $\sum_{i=1}^{n} \text{Cov}(\hat{\mu}_i, y_i) = \sigma^2 d$. So Err$\approx \sum_{i=1}^{n} \text{err}_i + 2\sigma^2 d =RSS+2d\sigma^2$.

(**) Continued

- In AIC, $y_i \sim f_\mu(y)$ with $\mu = E[y]$, $L(y,\mu) = -2\log f_\mu(y)$ is the deviance loss (which is equivalent to the square error loss in the Gaussian model with σ^2 known), and $\hat{\mu}_i$ is the MLE (which is equivalent to the LSE in the Gaussian case). Now,

$$\Omega_i = 2 \cdot \text{Cov}\left(\hat{\lambda}_i, y_i\right)$$

with $\hat{\lambda}_i$ being a function of $\hat{\mu}_i$ depending on $f_\mu(y)$, e.g., for the Bernoulli case, $\hat{\lambda}_i =$logit$(\hat{\mu}_i)$, and for the Gaussian case, $\hat{\lambda}_i = \frac{\hat{\mu}_i - \frac{1}{2}}{\sigma^2}$. It turns out that $\sum_{i=1}^n \text{Cov}\left(\hat{\lambda}_i, y_i\right) \approx d$, which is obvious for the Gaussian case. As a result,

$$\text{Err} \approx \sum_{i=1}^n \left[-2\log f_{\hat{\mu}_i}(y_i)\right] + 2d = -2\log L + 2d.$$

- Both the two covariance penalty methods and cross-validation are conditional or local methods in the sense that the randomness in O_i comes only from the ith data although the former fixes x_i while the latter allows x_i to be random.
- The former is model-based while the latter is model-free so is more robust against model mis-specification.
- The former is typically more efficient in estimating $\Omega := \sum_{i=1}^n \Omega_i$ than the latter although it imposes more assumptions on the model.

(*) Considerations in High Demensions

(Section 6.4)

High-Dimensional Data

- Traditionally, we consider only low-dimensional problems, where by dimension we mean the size of p.
- In the past 20 years, high-dimensional (i.e., $p \gg n$) data become popular, where p can be extremely large, while n is often limited due to cost, sample availability, etc.
- For example, a marketing analyst interested in understanding people's online shopping patterns could treat as features all of the search terms entered by users of a serach engine, which is sometimes known as the "bag-of-words" model.
- Maybe only a few hundred search engine users would like to share their search histories with the researcher.
- In this example, $n \approx 1000$, and p is much larger.
- In this case, classical approaches such as least squares, logistic regression, LDA etc. cannot apply.
- Issues like bias-variance trade-off and the danger of overfitting are particularly important in the high-dimensional case.

What Does Wrong in High Demensions?

- Consider least squares when p is close to or larger than n.
- In Figure 6.22, we consider $p = 2$ while $n = 20$ and 2.
- When $n = 2$, least squares results in a perfect fit, which certainly leads to overfitting of the data (e.g., check this fitting on the data in the left panel).
- The problem is that when $p > n$ or $p \approx n$, a simple least squares regression line is too flexible and hence overfits the data.
- In Figure 6.23, $n = 20$ observations were simulated, and between 1 and 20 features, each of which was completely unrelated to the response, were added to the regression.
  - $R^2 \uparrow 1$, training MSE$\downarrow$ 0, and testing MSE$\uparrow$ because including the additional predictors leads to a vast increase in the variance of $\hat{\beta}$.
- The C_p, AIC, and BIC approaches are not appropriate here since estimating $\hat{\sigma}^2$ is problematic; the adjusted R^2 is not appropriate either since we can easily obtain a model with an adjusted R^2 value of 1.

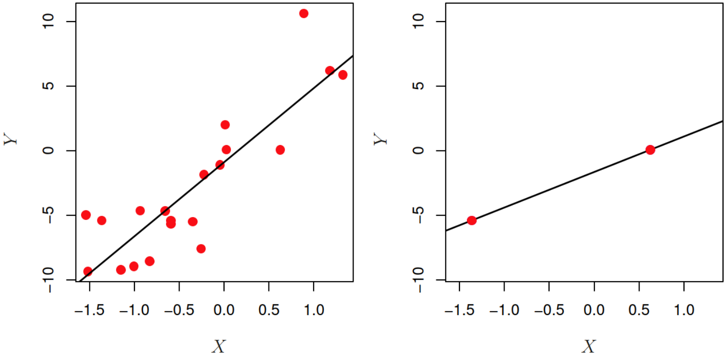


FIGURE 6.22. Left: *Least squares regression in the low-dimensional setting.* Right: *Least squares regression with $n = 2$ observations and two parameters to be estimated (an intercept and a coefficient).*

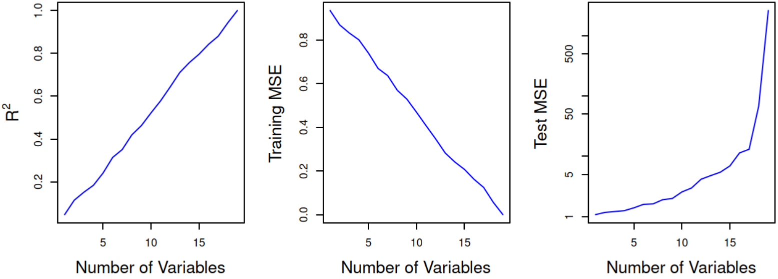


FIGURE 6.23. *On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model.* Left: *The R^2 increases to 1 as more features are included.* Center: *The training set MSE decreases to 0 as more features are included.* Right: *The test set MSE increases as more features are included.*

Regression in High Dimensions

- Many methods learned in this lecture for fitting less flexible least squares models, such as forward stepwise selection, RR, the lasso, and those (would be) learned in the future such as principal components regression and partial least squares, are particularly useful for performing regression in the high-dimensional setting.
- Figure 6.24 shows the perfromance of the lasso (test MSE against three values of p) in a simulated example with $n = 100$ while $p = 20, 50$ and 2000.
  - When $p = 20$, the best performance was achieved when λ is small, while when p gets larger, a larger λ achieves the lowest validation set error.
- Figure 6.24 highlights three important points:
  1. regularization or shrinkage plays a key role in high-dimensional problems;
  2. appropriate tuning parameter selection is crucial for good predictive performance;
  3. the test error tends to increase as p increases, unless the additional features are truly associated with the response, which is known as the curse of dimensionality.
- Increasing p need not be a bless: maybe only noise features are included, which exacerbates the risk of overfitting by assigning nonzero coefficients due to chance associations with the response; even if the new features are relevant, the variance incurred in fitting their coefficients may outweigh the reduction in bias that they bring.

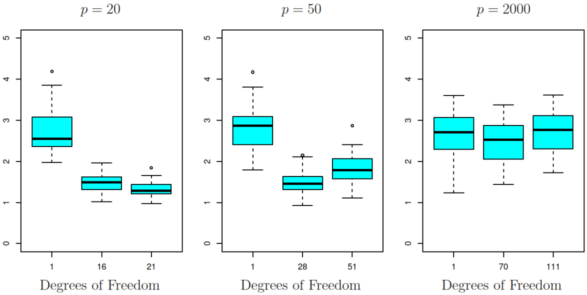


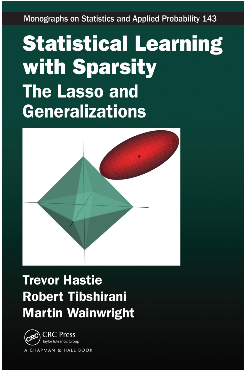
FIGURE 6.24. *The lasso was performed with $n = 100$ observations and three values of p, the number of features. Of the p features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter λ in (6.7). For ease of interpretation, rather than reporting λ, the* degrees of freedom *are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When $p = 20$, the lowest test MSE was obtained with the smallest amount of regularization. When $p = 50$, the lowest test MSE was achieved when there is a substantial amount of regularization. When $p = 2{,}000$ the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.*

Interpreting Results in High Dimensions

- In the high-dimensional setting, the multicollinearity problem is extreme: any variable in the model can be written as a linear combination of all of the other variables in the model.
- This means that we can never know exactly which variables (if any) truly are predictive of the outcome, and we can never identify the best coefficients for use.
- At most, we can hope to assign large regression coefficients to variables that are correlated with the variables that truly are predictive of the outcome.
- In the marketing example, we may identify 17 search terms by forward stepwise selection, but there are likely to be many sets of 17 terms (perhaps even non-overlapping sets) that would predict the online shopping behaviors just as well as the selected model.
- We must be careful not to overstate the results obtained, and to make it clear that what we have identified is simply one of many possible models for predicting online shopping, and that it must be further validated on independent data sets.
- It is also important to be careful in reporting errors and measures of model fit in the high-dimensional setting, e.g., one should never use sum of squared errors, p-values, R^2, or other traditional measures of model fit on the training data as evidence of a good model fit in the high-dimensional setting, rather, report results like MSE or R^2 on an independent test set, or CV errors.

Summary

- Model selection methods are an essential tool for data analysis, especially for big datasets involving many predictors.
- Research into methods that give sparsity, such as the lasso is an especially hot area.



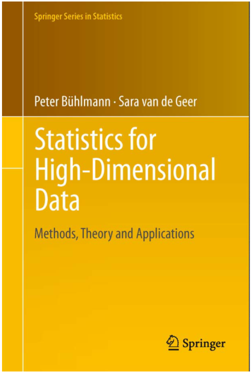
Monographs on Statistics and Applied Probability 143

Statistical Learning with Sparsity

The Lasso and Generalizations

Trevor Hastie
Robert Tibshirani
Martin Wainwright

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK



Springer Series in Statistics

Peter Bühlmann · Sara van de Geer

Statistics for High-Dimensional Data

Methods, Theory and Applications

Springer

Lab 1: Subset Selection Methods (Section 6.5.1)

- Best Subset Selection
- Forward and Backward Stepwise Selection
- Choosing Among Models Using the Validation Set Approach and CV

Lab2: Ridge Regression and the Lasso

(Section 6.5.2)

- Ridge Regression
- The Lasso

Lab3: Cross-Validation
(Section 5.3.1-3)

- The Validation Set Approach
- Leave-One-Out Cross-Validation
- *k*-Fold Cross-Validation

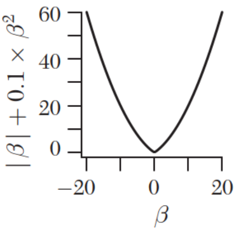
Appendix A: Extensions of the Lasso

Extensions of the Lasso

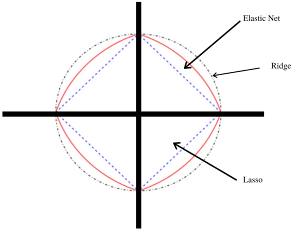
- Relaxed Lasso or Gauss Lasso: first find the variables with nonzero coefficients by the lasso, and the refit the linear model with these variables by least squares. This would help to remove the bias of $\hat{\beta}_j^L$ when it is nonzero.
- Elastic Net: combine RR and the lasso by using the penalty

$$\lambda\left[\frac{1}{2}(1-\alpha)\|\beta\|_2^2+\alpha\|\beta\|_1\right].$$

- It can overcome some limitations of the lasso, particularly when $p > n$, and with very high degrees of collinearity; it tends to select correlated features (or not) together.



Penalty Function in $\mathbb{R}^1$



Contour in $\mathbb{R}^2$

Continued

- Group Lasso: group $\beta_0 + \sum_{l=1}^{p} \beta_l x_{il}$ into $\theta_0 + \sum_{j=1}^{J} \theta_j^T z_{ij}$, where $z_{ij} \in \mathbb{R}^{p_j}$ are correlated features such as the dummay variables for levels of a qualitative variable, and $\sum_{j=1}^{J} p_j = p$, and change the penalty of lasso to

$$\lambda \sum_{j=1}^{J} \left\| \theta_j \right\|_2 ,$$

  where note that $\left\| \theta_j \right\|_2$ is not squared.
  - It also tends to select correlated groups.

- Fused Lasso: if x_{ij} is ordered in time, i.e., x_{ij} is actually x_{it}, and we hope time-neighboring coefficients to be the same or similar (grouped in time), then we can change the objective function of lasso to

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{t=1}^{p} \beta_t x_{it} \right)^2 + \lambda_1 \sum_{t=1}^{p} \left| \beta_j \right| + \lambda_2 \sum_{t=2}^{p} \left| \beta_t - \beta_{t-1} \right|$$

Nonconvex Penalties

- ℓ_q penalty with $0 < q < 1$:

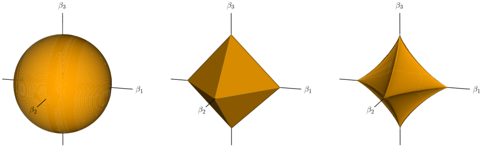


Figure *The ℓ_q unit balls in $\mathbb{R}^3$ for $q = 2$ (left), $q = 1$ (middle), and $q = 0.8$ (right). For $q < 1$ the constraint regions are nonconvex. Smaller q will correspond to fewer nonzero coefficients, and less shrinkage. The nonconvexity leads to combinatorially hard optimization problems.*

  - Like the lasso, it performs variable selection or feature extraction.
  - Unlike the lasso, the thresholding functions are discontinuous in $\hat{\beta}_j$.

- Adaptive Lasso: change the penalty of lasso to

$$\lambda \sum_{j=1}^{p} w_j |\beta_j|,$$

where $w_j = 1/|\tilde{\beta}_j|^{\nu}$, $\nu > 0$, and $\tilde{\beta}_j$ is a pilot estimate such as the LSE if $p < n$ and the univariate LSE if $p \geq n$.
- It can be seen as an approximation to the ℓ_q penalties with $q = 1 - \nu$, but the objective function is convex in β given $\tilde{\beta}$.

Continued

- SCAD (smoothly clipped absolute deviation): replace $\lambda\left|\beta\right|$ of lasso by $J_{\lambda,a}\left(\beta\right)$ for some $a\geq 2$ ($a=3.7$ from a Bayesian argument), where

$$
\begin{aligned}
J_{\lambda,a}\left(\beta\right) &= \lambda\int_0^{|\beta|}\left[I\left(s\leq\lambda\right)+\frac{\left(a\lambda-s\right)_+}{\left(a-1\right)\lambda}I\left(s>\lambda\right)\right]ds \\
&= \begin{cases} \lambda\left|\beta\right|, & \text{if } 0\leq\left|\beta\right|\leq\lambda, \\ \frac{1}{a-1}\left(-\frac{\lambda^2}{2}+a\lambda\left|\beta\right|-\frac{1}{2}\beta^2\right), & \text{if } \lambda<\left|\beta\right|\leq a\lambda, \\ \frac{1}{2}\left(a+1\right)\lambda^2, & \text{if } \left|\beta\right|>a\lambda. \end{cases}
\end{aligned}
$$

- The second term in square-braces reduces the amount of shrinkage in the lasso for larger values of β, with ultimately no shrinkage as $\left|\beta\right|\to\infty$; this is similar to the ℓ_q penalty with $0<q<1$ [figure here].

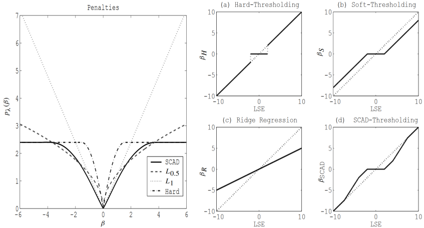


Figure Penalty functions (left panel) and PLS estimators (right panel).

Continued

- MC+ (minimax concave): replace $\lambda|\beta|$ of lasso by

$$\begin{aligned} P_{\lambda,\gamma}(\beta) &= \lambda\int_0^{|\beta|}\left(1-\frac{x}{\lambda\gamma}\right)_+ dx = \int_0^{|\beta|}\frac{(\lambda\gamma-x)_+}{\gamma}dx \\ &= \begin{cases} \lambda|\beta|-\frac{\beta^2}{2\gamma}, & \text{if } 0\le|\beta|\le\gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |\beta|>\gamma\lambda, \end{cases} \end{aligned}$$

  which translates the linear part of $J_{\lambda,a}(\beta)$ to the origin, where $\gamma\ge 1$ controls concavity.

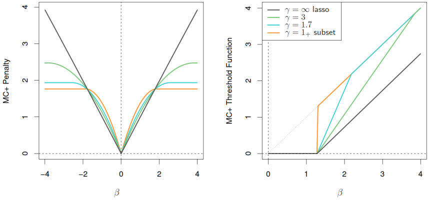


Figure *Left: The MC+ family of nonconvex sparsity penalties, indexed by a sparsity parameter $\gamma\in(1,\infty)$. Right: piecewise-linear and continuous threshold functions associated with MC+ (only the north-east quadrant is shown), making this penalty family suitable for coordinate descent algorithms.*

- log penalty: replace $\lambda|\beta|$ of lasso by $\log(1+\gamma|\beta|)$.

Continued: Some Others

- Bridge: including all ℓ_q penalty with $q \geq 0$.
- Garrote:

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} c_j \beta_j x_{ij} \right)^2 \text{ s.t. } \|\mathbf{c}\|_2 \leq s.$$

- Nonnegative Garrote:

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} c_j \beta_j x_{ij} \right)^2$$

$$\text{s.t. } c_j \geq 0,\ j = 1, \cdots, p,$$

$$\|\mathbf{c}\|_1 = \sum_{j=1}^{p} c_j \leq s.$$

- The prediction is $\sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij}$ in both Garrote and NN-Garrote.

Continued: GLM + Lasso

- We can also apply the lasso to different objective functions.
- For example, our target is

$$\min_{\beta}\left\{-\log\ell\left(\beta_0,\beta_1,\cdots,\beta_p\right)+\lambda\sum_{j=1}^{p}\kappa_j\left(\left|\beta_j\right|\right)\right\},$$

  where $\ell(\cdot)=\log L$ is the likelihood function of a GLM model in Lecture 2 (e.g., logistic regression), and $\kappa_j(\cdot)$ are increasing penalty functions.
- If the lasso penalty is used, we often set

$$\kappa_j\left(\left|\beta_j\right|\right)=\omega_j\left|\beta_j\right|$$

  with ω_j being the sample standard deviation of the jth covariate if the original covariate is not normalized to have length 1.

Appendix B: Bayesian Interpretation for Ridge Regression and the Lasso

Bayesian Analysis

- In a Bayesian analysis of regression, we need to impose a prior distribution on β, say $p(\beta)$.
  - If $p(\beta)$ contains unknown parameters α, then α is called hyperparameters.
- Suppose the likelihood of the data $(\mathbf{X}, \mathbf{y})$, $f(\mathbf{y}, \mathbf{X}|\beta) = f(\mathbf{y}|\mathbf{X}, \beta)\, f(\mathbf{X}|\beta)$, can be written as $f(\mathbf{y}|\mathbf{X}, \beta) f(\mathbf{X})$, i.e., $f(\mathbf{X})$ does not depend on β, $f(\mathbf{X}|\beta) = f(\mathbf{X})$.
- Then the posterior distribution of β is proportional to the product of the prior and the likelihood,

$$p(\beta|\mathbf{X}, \mathbf{y}) = \frac{f(\mathbf{y}, \mathbf{X}|\beta) p(\beta)}{\int f(\mathbf{y}, \mathbf{X}|\beta) p(\beta) d\beta} \propto f(\mathbf{y}|\mathbf{X}, \beta) f(\mathbf{X}) p(\beta) \propto f(\mathbf{y}|\mathbf{X}, \beta) p(\beta),$$

  where $\propto$ means a proportionality constant (which does not depend on β but may depend on $(\mathbf{X}, \mathbf{y})$) is neglected, and $\int f(\mathbf{y}, \mathbf{X}|\beta) p(\beta) d\beta$ does not depend on β.
- We can estimate β by the posterior mean, median or mode (also known as the maximum a posteriori or MAP estimate, where the likelihood is required to be consistent with prior [figure here]).
  - Popular samplers of the posterior include the Gibbs sampler, the Metropolis-Hastings sampler, and the slice sampler; see the supplementary Markov chain Monte Carlo (MCMC) chapter for more details.

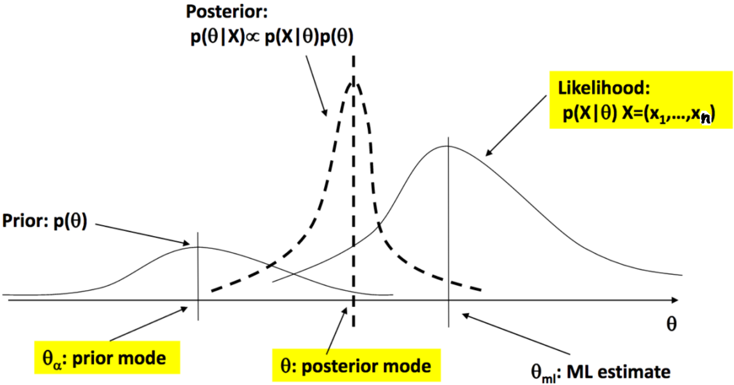

Posterior:
$p(\theta|X) \propto p(X|\theta)p(\theta)$
Likelihood:
$p(X|\theta)$ $X=(x_1,\ldots,x_n)$
Prior: $p(\theta)$
θ
θ_α: prior mode
θ: posterior mode
θ_{ml}: ML estimate

Figure: Illustration of MAP: θ is the parameter, and **X** is the data.

RR

- In the linear regression model,

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon,$$

  suppose $\varepsilon \sim N\left(0, \sigma^2\right)$, and

$$p(\beta) = g(\beta_0) \prod_{j=1}^{p} g\left(\beta_j\right) \propto \prod_{j=1}^{p} g\left(\beta_j\right),$$

  i.e., $\beta_0, \beta_1, \cdots, \beta_p$ are distributed independently, and β_0 has a diffuse or uninformative prior.

- If $g\left(\beta_j\right) = \phi\left(\beta_j | 0, \frac{\sigma^2}{\lambda}\right)$ [see Figure 6.11], where $\phi\left(\cdot|\mu_0, \sigma_0^2\right)$ is the pdf of $N\left(\mu_0, \sigma_0^2\right)$, then

$$\begin{aligned} p(\beta|\mathbf{X}, \mathbf{y})^R &\propto \prod_{i=1}^{n} \phi\left(y_i | \beta_0 + \mathbf{x}_i^T \beta, \sigma^2\right) \prod_{j=1}^{p} g\left(\beta_j\right) \\ &\propto \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left(y_i - \beta_0 - \mathbf{x}_i^T \beta\right)^2 - \frac{\lambda}{2\sigma^2} \sum_{j=1}^{p} \beta_j^2\right\}, \end{aligned}$$

  and

$$\arg\max_{\beta} p(\beta|\mathbf{X}, \mathbf{y})^R = \hat{\beta}_\lambda^R.$$

Lasso

- If

$$g\left(\beta_j\right)=\frac{\lambda}{4\sigma^2}\exp\left\{-\left|\beta_j\right|/\left(\frac{2\sigma^2}{\lambda}\right)\right\},$$

i.e., the double-exponential (or Laplace) distribution with scale parameter $\frac{2\sigma^2}{\lambda}$ [see Figure 6.11], then

$$\begin{aligned}
p\left(\beta|\mathbf{X},\mathbf{y}\right)^L &\propto \prod_{i=1}^n \phi\left(y_i|\beta_0+\mathbf{x}_i^T\beta,\sigma^2\right)\prod_{j=1}^p g\left(\beta_j\right)\\
&\propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n\left(y_i-\beta_0-\mathbf{x}_i^T\beta\right)^2-\frac{\lambda}{2\sigma^2}\sum_{j=1}^p\left|\beta_j\right|\right\}\\
&= \exp\left\{-\frac{\sum_{i=1}^n\left(y_i-\beta_0-\mathbf{x}_i^T\beta\right)^2+\lambda\sum_{j=1}^p\left|\beta_j\right|}{2\sigma^2}\right\},
\end{aligned}$$

and

$$\arg\max_{\beta} p\left(\beta|\mathbf{X},\mathbf{y}\right)^L=\arg\min_{\beta}\left\{\sum_{i=1}^n\left(y_i-\beta_0-\mathbf{x}_i^T\beta\right)^2+\lambda\sum_{j=1}^p\left|\beta_j\right|\right\}=\hat{\beta}_\lambda^L.$$

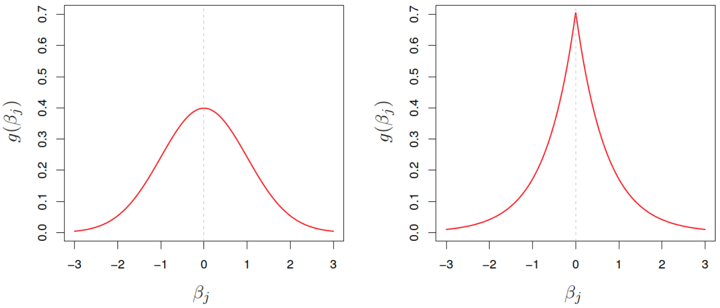


FIGURE 6.11. Left: *Ridge regression is the posterior mode for β under a Gaussian prior.* Right: *The lasso is the posterior mode for β under a double-exponential prior.*

- The lasso prior is steeply peaked at zero, while the Gaussian is flatter and fatter at zero. Hence, the lasso expects a priori that many of the coefficients are (exactly) zero, while ridge assumes the coefficients are randomly distributed about zero.
- $\hat{\beta}_{\lambda}^{R}$ is actually also the posterior mean of $p(\beta|\mathbf{X},\mathbf{y})^{R}$, while $\hat{\beta}_{\lambda}^{L}$ is not the posterior mean of $p(\beta|\mathbf{X},\mathbf{y})^{L}$; actually the posterior mean of $p(\beta|\mathbf{X},\mathbf{y})^{L}$ is not sparse.

Gamma-Lasso (GL)

- As discussed in Appendix A, the penalty on $|\beta_j|$ often takes a coordinate-specific form, $\omega_j|\beta_j|$, which is not covered in the above prior specification.
- To cover this case, we specify the prior for β_j as

$$\pi\left(\beta_j\right)=\frac{\lambda_j}{2}\exp\left(-\lambda_j|\beta_j|\right),$$

and the Laplace rate λ_j is assigned a conjugate gamma hyperprior

$$\text{Ga}\left(\lambda_j|s,r\right)=\frac{r^s}{\Gamma\left(s\right)}\lambda_j^{s-1}e^{-r\lambda_j},$$

where the shape s and rate r imply mean s/r and variance s/r^2.

- Since $\arg\max_{\lambda_j}\left\{\pi(\beta_j)\text{Ga}(\lambda_j|s,r)\right\}=s/(r+|\beta_j|)$, and $-\log\left\{\pi(\beta_j)\text{Ga}\left(\lambda_j|s,r\right)\right\}\propto -s\log\lambda_j+(r+|\beta_j|)\lambda_j$, the joint MAP estimation of β_j and λ_j is equivalent to add $-s\log\left[\frac{s/r}{1+|\beta_j|/r}\right]+s\propto s\log(1+|\beta_j|/r)$, which is the log penalty so is nonconvex in β_j, to $-\log f\left(\mathbf{y}|\mathbf{X},\beta\right)$.
- $\left(s,r\right)$ can be chosen by CV or AICc (especially when p is large, see Lecture 6 and Flynn, Hurvich and Simonoff (2013)).

Path of One-Step Estimators (POSE)

- For nonconvex penalties like SCAD, Zou and Li (2008) suggest iteratively weighted-ℓ_1 regularization, where the coefficient-specific weights are based upon results from previous iterations of ℓ_1 regularization.
- Actually, even a single step of such weighted-ℓ_1 regularization is enough to get solutions that are close to optimal, so long as the preestimates are good enough starting points. The crux of success is finding starts that are good enough.
- On the other hand, Efron et al. (2004) suggest the least-angle regression (LARS) algorithm for the lasso which provides a regularization path, a sequence of models under decreasing amounts of regularization.
- Taddy (2017) combines these two ideas for nonconvex penalties and proposes the path of one-step estimators (POSE), which takes advantage of an available source of initial values in any path estimation algorithm – the solved values from the previous path iteration, at no more cost than single lasso path.

Continued

- The POSE algorithm normalizes the penalty $c\left(\left|\beta_j\right|\right)$ as $c'(0) = 1$, which implies $s = r$ (equal to, say $1/\gamma$) in Gamma-Lasso.
  - When $\gamma \to 0$, $c(b) \to |b|$, and when $\gamma \to \infty$, $c(b)$ converges ℓ_0-type costs.
- It starts from a large λ, e.g., $\lambda^1 = \max\left\{\left|\left.\frac{\partial l(\alpha,\beta)}{\partial \beta_j}\right|_{\beta=\mathbf{0}}\right|, j = 1, \cdots, p\right\}$, such that $\beta^1 = \mathbf{0}$, sets $\lambda^t = \delta\lambda^{t-1}$ and

$$\omega_j^t = \begin{cases} c'(|\hat{\beta}_j^{t-1}|), & \text{if } \hat{\beta}_j^{t-1} \neq 0, \\ 1, & \text{if } \hat{\beta}_j^{t-1} = 0, \end{cases}$$

  solves

$$\left(\hat{\alpha}, \hat{\beta}\right)^t = \arg\min_{\alpha,\beta_j} l(\alpha,\beta) + \lambda^t \sum_{j=1}^p \omega_j^t \left|\beta_j\right|$$

  by coordinate descent, and ends at λ^T (e.g., $0.01\lambda^1$), where $l(\cdot)$ is the objective function, and α plays the role of β_0 above.
- γ and λ can be chosen by CV or AICc; usually, γ is chosen only from $\{0, 1, 10\}$.
  - When n/s is large, CV and AICc perform similarly in prediction; when n/s is small, CV is preferred when there is little signal and AICc is preferred when there is strong signal, where s measures the sparsity of the model.