Lecture Notes for Model Selection

1 Bias-Variance Decomposition

1.1 Variable Selection

When selecting variables for a model, one needs to consider the research hypothesis as well as any potential confounding variables to control for. For example, in most medical studies, age and gender are always included in the model since they are common confounders. Researchers are looking for the effect of other predictors on the response once age and gender have been accounted for.

If your research hypothesis specifically addresses the effect of a variable, say expenditure, you need to either include it in your model or show explicitly in your analysis why the variable does not belong.

Furthermore, one needs to consider the purpose of the analysis. If the purpose is to simply come up with accurate predictions for the response, researchers tend to simply look for variables that are easily obtained that account for a high degree of variation in the response.

However we choose to select our variables, we should always be wary of over-interpretation of the model in a multiple regression setting. Here's why: 1) The selected variables are not necessarily special. Variable selection methods are highly influenced by correlations between variables. Particularly when two predictors are highly correlated ($R^2 > .8$), usually one will be omitted despite the fact that the other may be a good predictor on its own. The problem is that since the two variables contain so much overlapping information, once you include one, the second variable accounts for very little additional variability in the response. 2) Interpretation of coefficients. If we have a regression coefficient of 0.2 for variable A, the interpretation is as follows: "While holding the values of all other predictors constant, a 1-unit increase in the value of A is associated with an increase of 0.2 in the expected value of the response." 3) Lastly, for observational studies, causality is rarely implied.

If the dimension p of the covariate X is large, then we might get better predictions by omitting some covariates. Models with many covariates have low bias but high variance; models with few covariates have high bias but low variance. The best predictions come from balancing these two extremes. This is called the **bias-variance tradeoff**. To reiterate:

including many covariates leads to low bias and high variance

including few covariates leads to high bias and low variance

The problem of deciding which variables to include in the regression model to achive a good tradeoff is called **model selection** or **variable selection**.

Variable Standardization in Variable Selection. It is convenient in model selection to first standardize all the variables by subtracting off the mean and dividing by the standard deviation. For example, we replace x_{ij} with $(x_{ij} - \overline{x}_j)/s_j$ where $\overline{x}_j = n^{-1} \sum_{i=1}^{n} x_{ij}$ is the mean of covariate

x_j and s_j is the standard deviation. The R function `scale` will do this for you. Thus, we assume throughout this section that

$$\frac{1}{n}\sum_{i=1}^{n} y_i = 0, \quad \frac{1}{n}\sum_{i=1}^{n} y_i^2 = 1 \tag{1}$$

$$\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0, \quad \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1, \quad j = 1, \ldots, p. \tag{2}$$

Some Notation. Given $S \subset \{1, \ldots, p\}$, let $(X_j : \ j \in S)$ denote a subset of the covariates. There are 2^p such subsets. Let $\beta(S) = (\beta_j : \ j \in S)$ denote the coefficients of the corresponding set of covariates and let $\widehat{\beta}(S) = (\mathbb{X}_S^T \mathbb{X}_S)^{-1} \mathbb{X}_S^T Y$ denote the least squares estimate of $\beta(S)$, where $\mathbb{X}_S$ denotes the design matrix for this subset of covariates. Thus, $\widehat{\beta}(S)$ is the least squares estimate of $\beta(S)$ from the **submodel** $Y = \mathbb{X}_S \beta(S) + \epsilon$. The vector of predicted values from model S is $\widehat{Y}(S) = \mathbb{X}_S \widehat{\beta}(S)$. For the null model $S = \emptyset$, $\widehat{Y}$ is defined to be a vector of 0's. Let $\widehat{r}_S(x) = \sum_{j \in S} \widehat{\beta}_j(S) x_j$ denote the estimated regression function for the submodel. We measure the predictive quality of the model via the prediction risk.

The **prediction risk** of the submodel S is defined to be

$$R(S) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(\widehat{Y}_i(S) - Y_i^*)^2 \tag{3}$$

where $Y_i^* = r(X_i) + \epsilon_i^*$ denotes the value of a future observation of Y at covariate value X_i.

Ideally, we want to select a submodel S to make $R(S)$ as small as possible. We face two problems:

- estimating $R(S)$
- searching through all the submodels S

1.2 Risk Estimation and Model Scoring

An obvious candidate to estimate $R(S)$ is the **training error**

$$\widehat{R}_{\mathrm{tr}}(S) = \frac{1}{n}\sum_{i=1}^{n} (\widehat{Y}_i(S) - Y_i)^2 = \frac{\mathrm{RSS}(S)}{n}. \tag{4}$$

For the null model $S = \emptyset$, $\widehat{Y}_i = 0$, $i = 1, \ldots, n$, and $\widehat{R}_{\rm tr}(S)$ is an unbiased estimator of $R(S)$ and this is the risk estimator we will use for this model. But in general, this is a poor estimator of $R(S)$ because it is very biased. Indeed, if we add more and more covariates to the model, we can track the data better and better and make $\widehat{R}_{\rm tr}(S)$ smaller and smaller. Thus if we used $\widehat{R}_{\rm tr}(S)$ for model selection we would always be led to include every covariate in the model.

1.1 Theorem. *The training error is a downward-biased estimate of the prediction risk, meaning that* $\mathbb{E}(\widehat{R}_{\rm tr}(S)) < R(S)$. *In fact,*

$$\mathsf{bias}(\widehat{R}_{\rm tr}(S)) = \mathbb{E}(\widehat{R}_{\rm tr}(S)) - R(S) = -\frac{2}{n}\sum_{i=1}^{n} \mathsf{Cov}(\widehat{Y}_i, Y_i). \tag{5}$$

Now we discuss some better estimates of risk. For each one we obtain an estimate of risk that can be approximately expressed in the form

$$\widehat{R}_{\rm tr}(S) + penalty(S).$$

One picks the model that yields the minimum estimated risk. The first term decreases, while the second term increases with model complexity. A challenge for most estimators of risk is that they require an estimate of σ^2.

Mallow's C_p

Mallow's C_p statistic is defined by

$$\widehat{R}(S) = \widehat{R}_{\rm tr}(S) + \frac{2|S|\widehat{\sigma}^2}{n} \tag{6}$$

where $|S|$ denotes the number of terms in S and $\widehat{\sigma}^2$ is the estimate of σ^2 obtained from the full model (with all covariates in the model). This is simply the training error plus a bias correction. This estimate is named in honor of Colin Mallows who invented it. The first term in (6) measures the fit of the model while the second measure the complexity of the model. Think of the C_p statistic as:

$$\text{lack of fit } + \text{ complexity penalty}. \tag{7}$$

The disadvantage of C_p is that we need to supply an estimate of σ.

Leave-one-out cross-validation

Another method for estimating risk is leave-one-out cross-validation.

The **leave-one-out cross-validation (CV) estimator** of risk is

$$\widehat{R}_{CV}(S) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_{(i)}(S))^2 \tag{8}$$

where $\widehat{Y}_{(i)}$ is the prediction for Y_i obtained by fitting the model with Y_i omitted. It can be shown that

$$\widehat{R}_{CV}(S) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \widehat{Y}_i(S)}{1 - H_{ii}(S)}\right)^2 \tag{9}$$

where $H_{ii}(S)$ is the i^{th} diagonal element of the hat matrix

$$H(S) = \mathbb{X}_S(\mathbb{X}_S^T\mathbb{X}_S)^{-1}\mathbb{X}_S^T. \tag{10}$$

From equation (9) it follows that we can compute the leave-one-out cross-validation estimator without actually dropping out each observation and refitting the model. An important advantage of cross-validation is that it does not require an estimate of σ.

We can relate CV to C_p as follows. First, approximate each $H_{ii}(S)$ with their average value $n^{-1}\sum_{i=1}^{n} H_{ii}(S) = \text{trace}(H(S))/n = |S|/n$. This yields

$$\widehat{R}_{CV}(S) \approx \frac{1}{n}\frac{\text{RSS}(S)}{\left(1 - \frac{|S|}{n}\right)^2}. \tag{11}$$

The right hand side of (11) is called the generalized cross validation (GCV) score and will come up again later. Next, use the fact that $1/(1-x)^2 \approx 1 + 2x$ and conclude that

$$\widehat{R}_{CV}(S) \approx \widehat{R}_{\text{tr}}(S) + \frac{2\widehat{\sigma}^2|S|}{n} \tag{12}$$

where $\widehat{\sigma}^2 = \text{RSS}(S)/n$. This is identical to C_p except that the estimator of σ^2 varies with the choice of model.

Akaike Information Criterion

Another criterion for model selection is **AIC (Akaike Information Criterion).** The idea is to choose S to maximize

$$\ell_S - |S|, \tag{13}$$

or minimize

$$-2\ell_S + 2|S|,$$

where $\ell_S = \ell_S(\widehat{\beta}_S, \sigma^2)$ is the log-likelihood (assuming Normal errors) of the model evaluated at the MLE. This can be thought of as "goodness of fit" minus "complexity." Assuming Normal errors,

$$\ell(\beta, \sigma^2) = \text{constant} - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}||Y - X\beta|^2.$$

Define $\text{RSS}(S)$ as the residual sum of squares in model S. Inserting $\widehat{\beta}$ yields

$$\ell(\widehat{\beta}, \sigma^2) = \text{constant} - \frac{n}{2}\log\sigma^2 - \frac{1}{2\sigma^2}RSS(S).$$

In this expression we can ignore $\frac{n}{2}\log\sigma^2$ because it does not include any terms that depend on the fit of model S. Thus up to a constant we can write

$$AIC(S) = \frac{\text{RSS}(S)}{\sigma^2} + 2|S|. \tag{14}$$

Equivalently AIC finds the model that minimizes

$$\frac{\text{RSS}(S)}{n} + \frac{2|S|\sigma^2}{n}. \tag{15}$$

If we estimate $\widehat{\sigma}$ using the error from largest model, then minimizing AIC is equivalent to minimizing Mallow's C_p.

Bayesian information criterion

Yet another criterion for model selection is **BIC (Bayesian information criterion)**. Here we choose a model to maximize

$$\text{BIC}(S) = \ell_S - \frac{|S|}{2}\log n \tag{16}$$

The BIC score has a Bayesian interpretation. Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ where $m = 2^p$ denote all the models. Suppose we assign the prior $\mathbb{P}(S_j) = 1/m$ over the models. Also, assume we put a smooth prior on the parameters within each model. It can be shown that the posterior probability for a model is approximately,

$$\mathbb{P}(S_j|\text{data}) \approx \frac{e^{BIC(S_j)}}{\sum_r e^{BIC(S_r)}}. \tag{17}$$

Hence, choosing the model with highest BIC is like choosing the model with highest posterior probability. But this interpretation is poor unless n is large relative to p. The BIC score also has an information-theoretic interpretation in terms of something called minimum description length. The BIC score is identical to AIC except that it puts a more severe penalty for complexity. It thus leads one to choose a smaller model than the other methods.

Summary

- Cp
$$\widehat{R}(S) = \widehat{R}_{\text{tr}}(S) + \frac{2|S|\widehat{\sigma_{full}}^2}{n}.$$
- CV
$$\widehat{R}(S) \approx \widehat{R}_{\text{tr}}(S) + \frac{2|S|\widehat{\sigma_S}^2}{n}.$$
- AIC
$$-2\ell(S) + 2|S|.$$
- -2BIC
$$-2\ell(S) + |S| \log n.$$

Note: the key term in $\ell(S)$ is $\widehat{R}_{\text{tr}}(S)$, so each of these methods has a similar form. They vary in how they estimate σ^2 and how substantial a penalty is paid for model complexity. BIC uses a penalty that increases with sample size.

2 Model Search

Once we choose a model selection criterion, such as cross-validation or AIC, we then need to search through all 2^p models, assign a score to each one, and choose the model with the best score. We will consider 4 methods for searching through the space of models:

1. Fit all submodels.
2. Forward stepwise regression.
3. Ridge Regression.
4. The Lasso.

Fitting All Submodels. If p is not too large we can do a complete search over all the models.
Stepwise.
When p is large, searching through all 2^p models is infeasible. In that case we need to search over a subset of all the models. One common method is to use stepwise regression. Stepwise regression can be run forward, backward, or in both directions.

In forward stepwise regression, we start with no covariates in the model. We then add the one variable that leads to the best score. We continue adding variables one at a time this way. See Figure 1. Backwards stepwise regression is the same except that we start with the biggest model and drop one variable at a time. Both are greedy searches; nether is guaranteed to find the model with the best score. Backward selection is infeasible when p is larger than n since $\widehat{\beta}$ will not be defined for the largest model. Hence, forward selection is preferred when p is large.

Forward Stepwise Regression

1. For $j = 1, \ldots, p$, regress Y on the j^{th} covariate X_j and let $\widehat{R}_j$ be the estimated risk. Set $\widehat{j} = \text{argmin}_j \widehat{R}_j$ and let $S = \{\widehat{j}\}$.

2. For each $j \in S^c$, fit the regression model $Y = \beta_j X_j + \sum_{s \in S} \beta_s X_s + \epsilon$ and let $\widehat{R}_j$ be the estimated risk. Set $\widehat{j} = \text{argmin}_{j \in S^c} \widehat{R}_j$ and update $S \longleftarrow S \bigcup \{\widehat{j}\}$.

3. Repeat the previous step until all variables are in S or until it is not possible to fit the regression.

4. Choose the final model to be the one with the smallest estimated risk.

Figure 1: Forward stepwise regression.

3 Regularization: Ridge Regression and the Lasso.

Another way to deal with variable selection is to use **regularization** or **penalization**. Specifically, we define $\widehat{\beta}$ to minimize the penalized sums of squares

$$Q(\beta) = \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \text{pen}\,(\beta)$$

where $\text{pen}(\beta)$ is a penalty and $\lambda \geq 0$ is a tuning parameter. The bigger λ, the bigger the penalty for model complexity. We consider three choices for the penalty:

$$\begin{aligned} L_0 \text{ penalty} \quad ||\beta||_0 &= \#\{j : \beta_j \neq 0\} \\ L_1 \text{ penalty} \quad ||\beta||_1 &= \sum_{j=1}^{p} |\beta_j| \\ L_2 \text{ penalty} \quad ||\beta||_2 &= \sum_{j=1}^{p} \beta_j^2. \end{aligned}$$

The L_0 penalty would force us to choose estimates which make many of the $\widehat{\beta}_j$'s equal to 0. But there is no way to minimize $Q(\beta)$ without searching through all the submodels.

The L_2 penalty is easy to implement. The estimate $\widehat{\beta}$ that minimizes

$$\sum_{i=1}^{n} (Y_i - \sum_{j=1}^{p} \beta_j X_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

is called the **ridge estimator**. It can be shown that the estimator $\widetilde{\beta}$ that minimizes the penalized sums of squares is as follows (assuming the features are standardized) is

$$\widehat{\beta} = (\mathbb{X}^T\mathbb{X} + \lambda I)^{-1}\mathbb{X}^T Y,$$

where I is the identity. When $\lambda = 0$ we get the least squares estimate (low bias, high variance). When $\lambda \to \infty$ we get $\widehat{\beta} = 0$ (high bias, low variance).

Ridge regression produces a linear estimator: $\widehat{\beta} = SY$ where

$$S = (\mathbb{X}^T\mathbb{X} + \lambda I)^{-1}\mathbb{X}^T$$

and $\widehat{Y} = HY$ where

$$H = \mathbb{X}(\mathbb{X}^T\mathbb{X} + \lambda I)^{-1}\mathbb{X}^T.$$

For regression we keep track of two types of degrees of freedom: model df (p, the number of covariates), and degrees of freedom error ($n - p - 1$). As the model incorporates more covariates, it becomes more complex, fitting the data better and better and eventually, over-fitting the data. For the remainder of the notes when we say "effective degrees of freedom", we are always refering to an analog to the model degrees of freedom.

For regularized regression, the **effective degrees of freedom** is defined to be

$$\mathrm{df}(\lambda) = \mathrm{trace}(H).$$

When $\lambda = 0$ we have $\mathrm{df}(\lambda) = p$ (maximum complexity) and when $\lambda \to \infty$, $\mathrm{df}(\lambda) \to 0$ (minimum complexity).

How do we choose λ? Recall that $\widehat{r}_{(-i)}(x_i) = \widehat{Y}_{(-i)}$, the leave-one-out fitted value, and the cross-validation estimate of predictive risk is

$$CV = \sum_{i=1}^{n}(y_i - \widehat{r}_{(-i)}(x_i))^2.$$

It can be shown that

$$CV = \sum_{i=1}^{n}\left(\frac{y_i - \widehat{r}(x_i)}{1 - H_{ii}}\right)^2.$$

Thus we can choose λ to minimize CV.

An alternative criterion that is sometimes used is **generalized cross validation** or, GCV. This is just an approximation to CV where H_{ii} is replaced with its average: $n^{-1}\sum_{i=1}^{n} H_{ii}$. Thus,

$$GCV = \sum_{i=1}^{n}\left(\frac{Y_i - \widehat{r}(x_i)}{1 - b}\right)^2$$

where

$$b = \frac{1}{n}\sum_{i=1}^{n} H_{ii} = \frac{df(\lambda)}{n}.$$

(Note: H is a function of λ, so b varies with λ.)

The problem with ridge regression is that we really haven't done variable selection because we haven't forced any $\widehat{\beta}_j$'s to be 0. This is where the L_1 penalty comes in.

The **lasso** estimator $\widehat{\beta}(\lambda)$ is the value of β that solves:

$$\min_{\beta \in \mathbb{R}^p} \left(\sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda ||\beta||_1 \right) \tag{18}$$

where $\lambda > 0$ and $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ is the L_1 norm of the vector β.

The lasso (Least Absolute Shrinkage and Selection) is called **basis pursuit** in the signal processing literature. Equation (18) defines a convex optimation problem with a unique solution $\widehat{\beta}(\lambda)$ that depends on λ. Typically, it turns out that many of the $\widehat{\beta}_j(\lambda)'s$ are zero. (See Figure 2 for intuition on this process.) Thus, **the lasso performs estimation and model selection simultaneously**. The selected model, for a given λ, is

$$S(\lambda) = \{j : \widehat{\beta}_j(\lambda) \neq 0\}. \tag{19}$$

The constant λ can be chosen by cross-validation. The estimator has to be computed numerically but this is a convex optimization and so can be solved quickly.

To see the difference in shrinkage and selection of terms in ridge versus lasso regression compare Figures 3 and 4, respectively. For the chosen tuning parameters shown (selected by cross validation), only three of the variables are included in the final lasso model (svi, lweight and cavol). In contrast, because ridge regression is not a model selection procedure, all of the terms are in the ridge model; however, the three chosen by lasso have the largest coefficients in the ridge model.

What is special about the L_1 penalty? First, this is the closest penalty to the L_0 penalty that makes $Q(\beta)$ convex. Moreover, the L_1 penalty captures **sparsity**.

Digression on Sparsity. We would like our estimator $\widehat{\beta}$ to be sparse, meaning that most $\widehat{\beta}_j$'s are zero (or close to zero). Consider the following two vectors, each of length p:

$$\begin{aligned} u &= (1, 0, \ldots, 0) \\ v &= (1/\sqrt{p}, 1/\sqrt{p}, \ldots, 1/\sqrt{p}). \end{aligned}$$

Intuitively, u is sparse while v is not. Let us now compute the norms:

$$\begin{array}{ll} ||u||_1 = 1 & ||u||_2 = 1 \\ ||v||_1 = \sqrt{p} & ||v||_2 = 1. \end{array}$$

So the L_1 norm correctly captures sparseness. This explains why lasso can be used for model selection, but ridge regression cannot. (Note: a slightly better model can be obtained using a 2-step process. 1) Lasso is used to select the features. 2) Entering only these features, refit the model using least squares regression.)

A computational Aside. Two related variable selection methods are forward stagewise regression and lars. In **forward stagewise regression** we first set $\widehat{Y} = (0, \ldots, 0)^T$ and we choose a small, positive constant ϵ. Now we build the predicted values incrementally. Let $\widehat{Y}$ denote the current vector of predicted values. Find the current correlations $c = c(\widehat{Y}) = \mathbb{X}^T(Y - \widehat{Y})$ and set

$$\widehat{j} = \text{argmax}_j |c_j|. \tag{20}$$

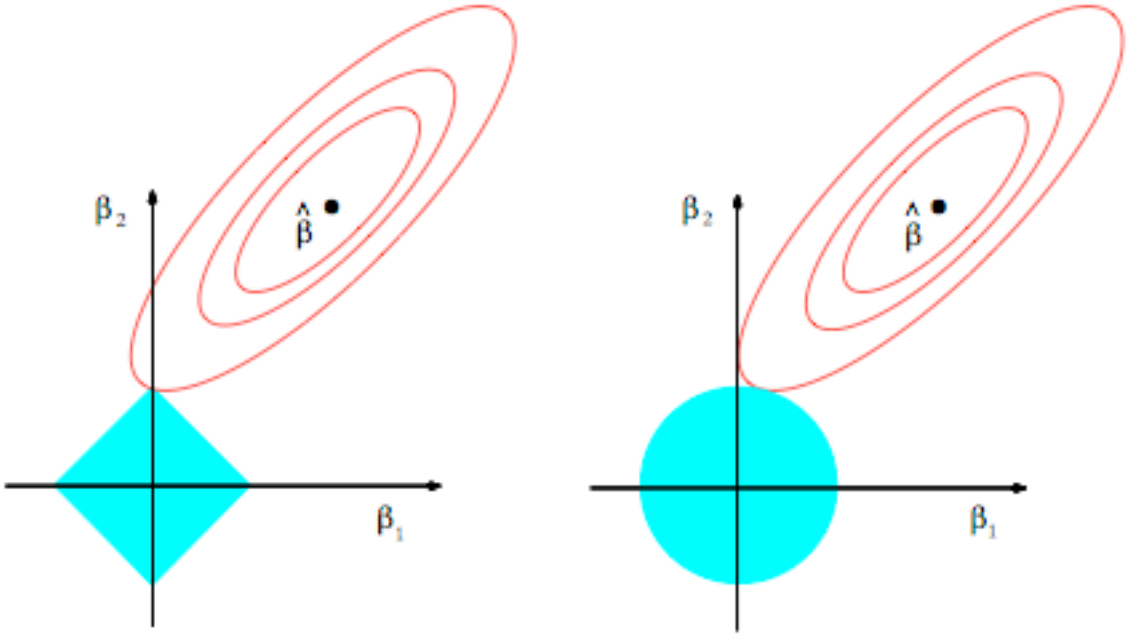

FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions* $|\beta_1| + |\beta_2| \leq t$ *and* $\beta_1^2 + \beta_2^2 \leq t^2$*, respectively, while the red ellipses are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

Figure 2: from Hastie et al. 2001

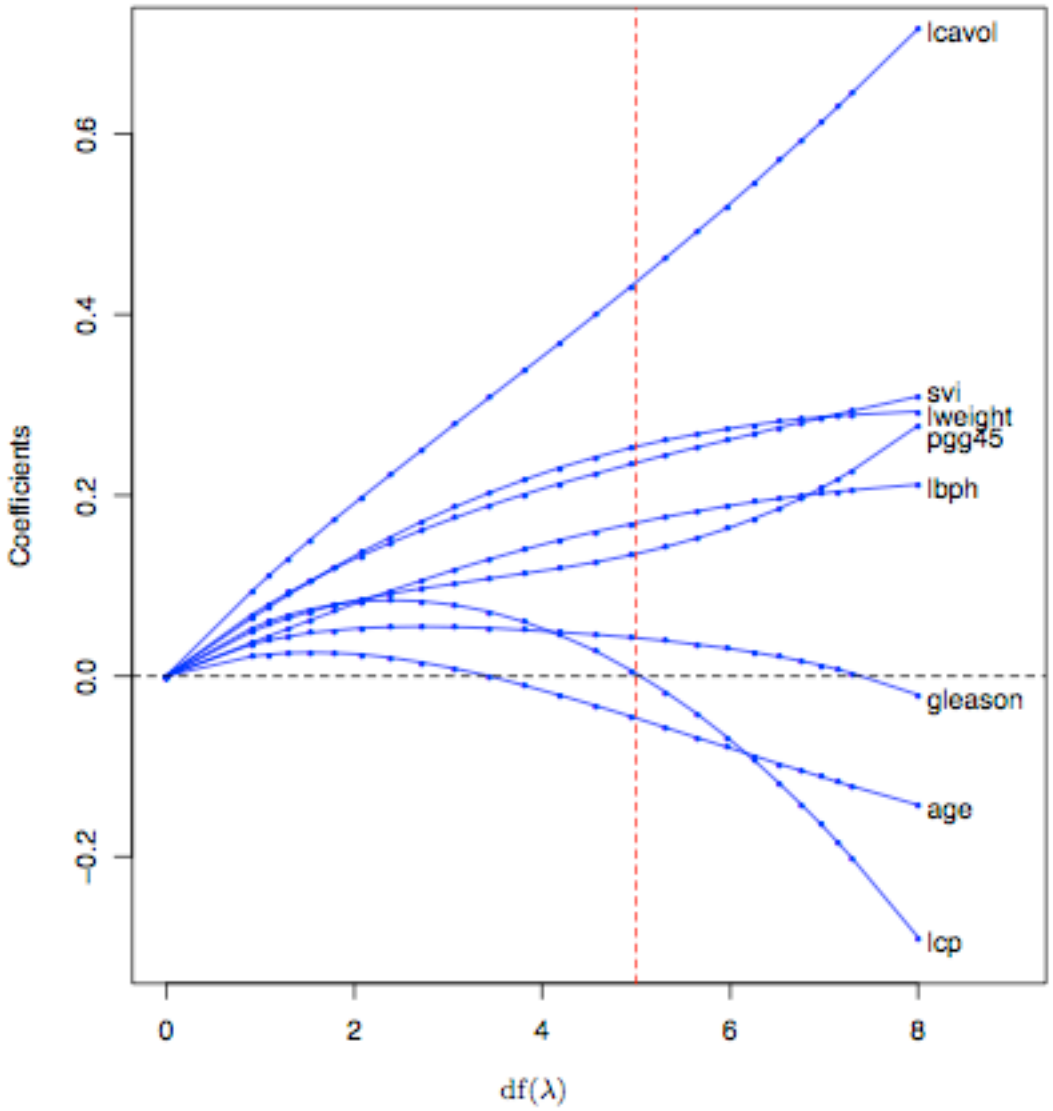

FIGURE 3.8. *Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter* λ *is varied. Coefficients are plotted versus* $\mathrm{df}(\lambda)$*, the effective degrees of freedom. A vertical line is drawn at* $\mathrm{df} = 5.0$*, the value chosen by cross-validation.*

Figure 3: from Hastie et al. 2001

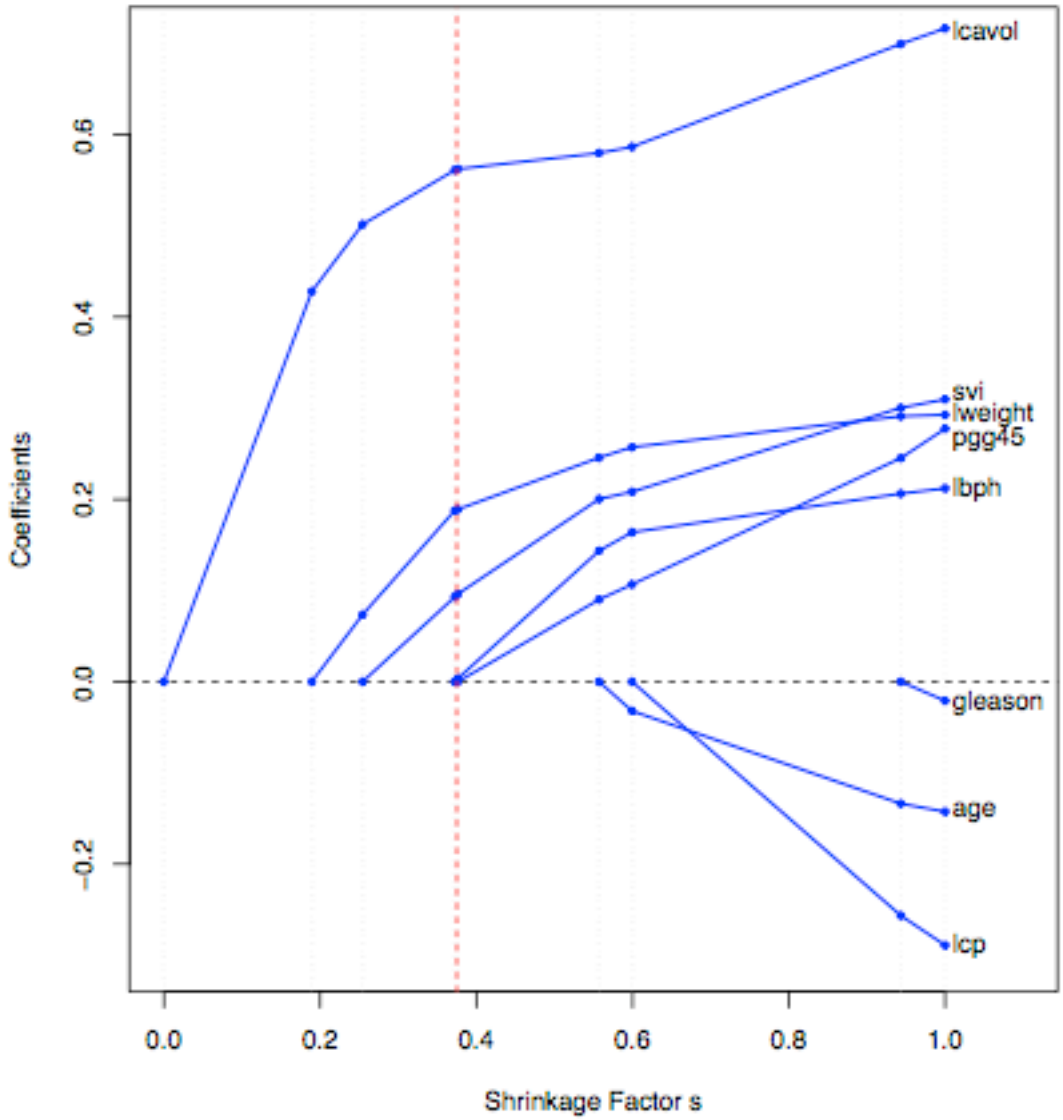

FIGURE 3.10. *Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.*

Figure 4: from Hastie et al. 2001

`lars`

1. Set $\widehat{Y} = 0$, $k = 0$, $A = \emptyset$. Now repeat steps 2–3 until $A^c = \emptyset$.

2. Compute the following quantities:

$$\begin{array}{lll} c = \mathbb{X}^T(Y - \widehat{Y}) & C = \max_j\{|c_j|\} & A = \{j : \ |c_j| = C\} \\ s_j = \text{sign}(c_j), \ \ j \in A & \mathbb{X}_A = (s_j x_j : \ j \in A) & G = \mathbb{X}_A^T \mathbb{X}_A \\ B = (\mathbf{1}^T G^{-1} \mathbf{1})^{-1/2} & w = BG^{-1}\mathbf{1} & u = \mathbb{X}_A w \\ a = \mathbb{X}^T u & & \end{array} \tag{22}$$

   where $\mathbf{1}$ is a vector of 1's of length $|A|$.

3. Set

$$\widehat{Y} \longleftarrow \widehat{Y} + \gamma u \tag{23}$$

   where

$$\gamma = \min_{j \in \mathcal{A}^c}{}^{+} \left\{ \frac{C - c_j}{B - a_j}, \frac{C + c_j}{B + a_j} \right\}. \tag{24}$$

   Here, $\min^+$ means that the minimum is only over positive components.

Figure 5: A formal descrription of lars.

Finally, we update $\widehat{Y}$ by the following equation:

$$\widehat{Y} \longleftarrow \widehat{Y} + \epsilon \ \text{sign}(c_j) x_j. \tag{21}$$

This is like forward stepwise regression except that we only take small, incremental steps towards the next variable and we do not go back and refit the previous variables by least squares.

A modification of forward stagewise regression is called **least angle regression**. We begin with all coefficients set to 0 and then find the predictor x_j most correlated with Y. Then increase $\widehat{\beta}_j$ in direction of the sign of its correlation with Y and set $\widehat{\epsilon} = Y - \widehat{Y}$. When some other predictor x_k has as much correlation with $\widehat{\epsilon}$ as x_j has we increase $(\widehat{\beta}_j, \widehat{\beta}_k)$ in their joint least squares direction, until some other predictor x_m has as much correlation with the residual $\widehat{\epsilon}$. Continue until all predictors are in the model. A formal description is in Figure 5.

`lars` can be easily modified to produce the lasso estimator. If a non-zero coefficient ever hits zero, remove it from the active set $\mathcal{A}$ of predictors and recompute the joint direction. This is why the `lars` function in R is used to compute the lasso estimator. You need to download the lars package first.

Summary

1. The prediction risk $R(S) = n^{-1}\sum_{i=1}^{n}(\widehat{Y}_i(S) - Y_i^*)^2$ can be decomposed into unavoidable error, bias and variance.

2. Large models have low bias and high variance. Small models have high bias and low variance. This is the bias-variance tradeoff.

3. Model selection methods aim to find a model which balances bias and varaince, yielding a small risk.

4. C_p or cross-validation are used to estimate the risk.

5. Search methods look through a subset of models and find the one with the smallest value of estimated risk $\widehat{R}(S)$.

6. The lasso estimates β with the penalized residual sums of squares $\sum_{i=1}^{n}(Y_i - X_i^T\beta)^2 + \lambda||\beta||_1$. Some of the estimates will be 0 and this corresponds to omitting them from the model. lars is an efficient algorithm for computing the lasso estimates.

4 Advanced Regularized Variable Selection

4.1 Elastic Net

Although lasso is widely used for variable selection,it has some limitations. Consider the following three scenarios:

1. In the p ¿ n case, the lasso selects at most n variables before it saturates, because of the nature of the convex optimization problem. This seems to be a limiting feature for a variable selection method. Moreover, the lasso is not well-defined unless the bound on the L1 norm of the coefficients is smaller than a certain value.

2. If there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected.

3. For usual n ¿ p situations, if there exist high correlations among predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression .

For example in the gene-selection problem in microarray data analysis, a typical microarray data set has many thousands of predictors (genes) and often less than 100 samples. For those genes

sharing the same biological "pathway", the correlations among them can be high. We think of those genes as forming a group. The ideal gene selection method should be able to do two things: eliminate the trivial genes, and automatically include whole groups into the model once one gene amongst them is selected. For this kind of p $\gg$ n and grouped variables situation, the lasso is not the ideal method, because it can only select at most n variables out of p candidates, and it lacks the ability to reveal the grouping information. Scenario (3) is relevant in the context of prediction. It is possible to further strengthen the prediction power of the lasso.

One method to overcome these problem is Elastic Net. Similar to the lasso, the elastic net simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. Elastic net often outperforms the lasso in terms of prediction accuracy.

Naive Elastic Net For any fixed non-negative λ_1 and λ_2 the naive elastic net is defined as:

$$\widehat{\beta} = argmin_\beta ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda_2||\beta||_2^2 + \lambda_1||\beta||_1 \tag{25}$$

where $||\beta||^2 = \sum_{j=1}^p \beta_j^2$ and $\beta_1 = \sum_{j=1}^p |\beta_j$—

- The L_1 part of the penalty generates a sparse model.
- The L_2 part of the penalty
  - Removes the limitation on the number of selected variables
  - Encourages grouping effect;
  - Stabilizes the L_1 regularization path.

The above procedure can be viewed as a penalized least-squares method. Let $\alpha = \frac{\lambda_2}{\lambda_1+\lambda_2}$, then solving $\widehat{\beta}$ in (25) is equivalent to the optimization problem:

$$\widehat{\beta} = argmin_\beta |\mathbf{y} - \mathbf{X}\beta|^2, \;\; subject\, to\, (1-\alpha)|\beta|_1 + \alpha|\beta|^2 \leq t \; for\; \alpha < 1 \; and \; for \; some \; t.$$

The function $(1-\alpha)|\beta|_1 + \alpha|\beta|^2$ is called the elastic net penalty, which is a convex combination of the lasso and ridge penalty. When $\alpha = 1$, the naive elastic net becomes simple ridge regression, and for $\alpha = 0$ the problem becomes elastic net.

The geometry of elastic net for $\alpha = 0.5$ is shown in Fig. 6. For all $\alpha \in [0, 1)$ the elastic net penalty function is singular (without first derivative) at 0 and it is strictly convex for all $\alpha > 0$, thus having the characteristics of both the lasso and ridge regression. Note that the lasso penalty ($\alpha a = 0$) is convex but not strictly convex.

The optimization problem corresponding to the naive elastic net can be written as a lasso optimization problem and thus solving a naive elastic net is no harder or no more computationally expensive than lasso.

Figure 7 shows the operational characteristics of the three penalization methods in an orthogonal design, where the naive elastic net can be viewed as a two-stage procedure: a ridgetype direct shrinkage followed by a lasso-type thresholding.

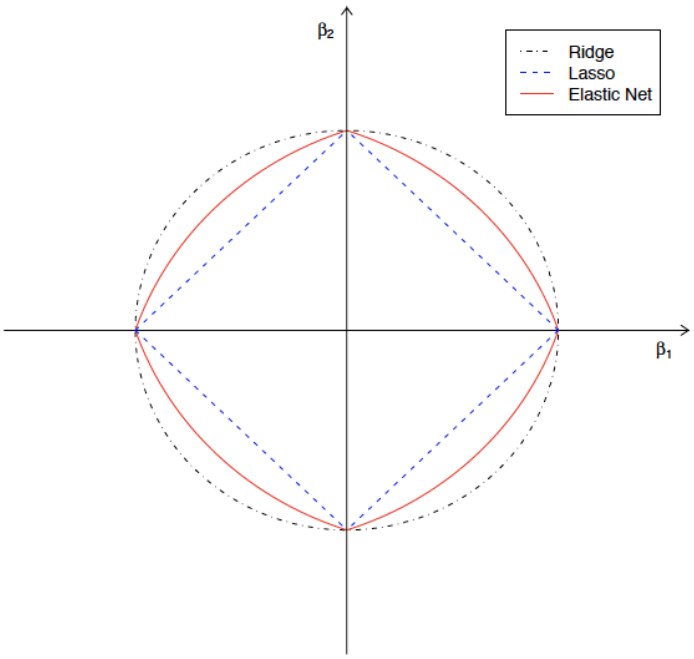

Figure 6: Geometry of the elastic net for $\alpha = 0.5$

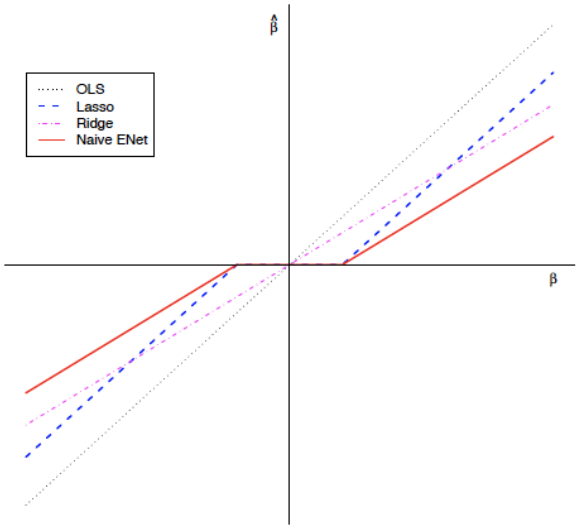

Figure 7: Exact solutions for the lasso, ridge and the naive elastic net (naive ENet) in an orthogonal design. Shrinkage parameters are $\lambda_1 = 2$, $\lambda_2 = 1$.

Elastic Net The naive elastic net overcomes the limitations of the lasso in scenarios (1) and (2). However, naive elastic net does not perform satisfactorily unless it is very close to either ridge or the lasso. In the regression prediction setting, an accurate penalization method achieves good prediction performance through the bias-variance trade-off. The naive elastic net estimator is a two-stage procedure: for each fixed λ_2 we first find the ridge regression coefficients, and then we do the lasso type shrinkage along the lasso coefficient solution paths. It appears to incur a double amount of shrinkage. Double shrinkage does not help to reduce the variances much and introduces unnecessary extra bias, compared with pure lasso or ridge shrinkage. Therefore, the elastic net optimization is proposed as follows:

$$\widehat{\beta}^* = argmin_{\beta^*}|\mathbf{y}^* - \mathbf{X}^*\beta^*|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}}|\beta^*|_1$$

as compared to the naive elastic net optimization

$$\widehat{\beta}^* = argmin_{\beta}||\mathbf{y}^* - \mathbf{X}^*\beta^*||_2^2 - +\gamma||\beta^*||_1$$

where $(\mathbf{y}^*, \mathbf{X}^*)$ are transformations of the original data $(\mathbf{y}, \underline{\mathbf{X}})$.

The elastic net estimate $\widehat{\beta}$ is:

$$\widehat{\beta}(elasticnet) = \sqrt{1+\lambda_2}\widehat{\beta}*$$

compared to $\widehat{\beta}(naive\ elastic\ net) = \frac{1}{\sqrt{1+\lambda_2}}\widehat{\beta}^*$, thus

$$\widehat{\beta}(elastic\ net) = (1+\lambda_2)\widehat{\beta}(naive\ elastic\ net).$$

Hence the elastic net coefficient is a re-scaled naive elastic net coefficient. Such a scaling transformation preserves the variable-selection property of the naive elastic net, and is the simplest way to undo shrinkage.

4.2 Group Variable Selection

Consider the general regression problem with J factors:

$$Y = \sum_{j=1}^{J} X_j\beta_j + \epsilon \tag{26}$$

Where Y is an $n \times 1$ vector, $\epsilon \sim N_n(0, \sigma^2 I)$ is an $n \times p_j$ matrix corresponding to the jth factor and β_j is a coefficient vector of size p_j, j =1, . . . , J. To eliminate the intercept from equation 26, we centre the response variable and each input variable so that the observed mean is 0. To simplify the description, we further assume that each X_j is orthonormalized, i.e. $X_j'X_j = I_{pj}$, j=1, . . . , J. This can be done through Gram–Schmidt orthonormalization, and different orthonormalizations correspond to reparameterizing the factor through different orthonormal contrasts. Denoting $X = (X_1, X_2, ..., X_J)$ and $\beta = (\beta_1', ..., \beta_J')'$, equation 26 can be written as $Y = X\beta + \epsilon$.

In many regression problems we are interested in finding important explanatory factors in predicting the response variable, where each explanatory factor may be represented by a group of derived input variables. The most common example is the multifactor analysis-of-variance (ANOVA) problem, in which each factor may have several levels and can be expressed through a group of dummy variables. Another example is the additive model with polynomial or nonparametric components. In both situations, each component in the additive model may be expressed as a linear combination of a number of basis functions of the original measured variable. In such cases the selection of important measured variables corresponds to the selection of groups of basis functions. In both of these two examples, variable selection typically amounts to the selection of important factors (groups of variables) rather than individual derived variables, as each factor corresponds to one measured variable and is directly related to the cost of measurement.

4.3 Group Lasso

For a vector $\eta \in R^d$, $d \geq 1$, and a symmetric d ×d positive definite matrix K, we denote $||\eta||_K = (\eta' K \eta)^{\frac{1}{2}}$. We write $||\eta|| = ||\eta||_{I_d}$ for brevity. Given positive definite matrices $K_1, ..., K_J$, the group lasso estimate is defined as the solution to

$$\frac{1}{2}||Y - \sum_{j=1}^{J} X_j \beta_j||^2 + \lambda \sum_{j=1}^{J} ||\beta_j|| K_j \tag{27}$$

Where $\lambda \geq 0$ is a tuning parameter. Eq. (27) is an extension of the lasso for selecting groups of variables. It is clear that this expression reduces to the lasso when $p_1 = ... = p_J = 1$. The penalty function that is used in this expression is intermediate between the L_1-penalty that is used in the lasso and the L_2-penalty that is used in ridge regression.

There are many reasonable choices for the kernel matrices $K_j s$, for example, $K_j = I_{pj}, j = 1, \ldots, J$ or $K_j = p_j I_{pj}$.