Linear Model Selection and Regularization

- Recall the linear model

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

- In the lectures that follow, we consider some approaches for extending the linear model framework. In the lectures covering Chapter 7 of the text, we generalize the linear model in order to accommodate *non-linear*, but still *additive*, relationships.
- In the lectures covering Chapter 8 we consider even more general *non-linear* models.

In praise of 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:* especially when $p > n$, to control the variance.
- *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*.

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.
- *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 can also perform variable selection.
- *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.

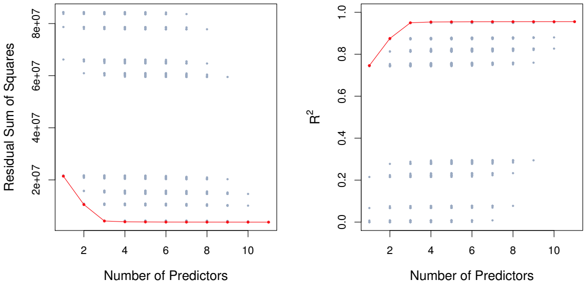
Subset Selection

Best subset and stepwise model selection procedures

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 ( ) models that contain exactly k predictors.
   (b) Pick the best among these ( ) models, and call it $\mathcal{M}$. 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.

Example- Credit data set



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*

Extensions to other models

- Although we have presented best subset selection here for least squares regression, the same ideas apply to other types of models, such as logistic regression.
- The *deviance*— negative two times the maximized log-likelihood— plays the role of RSS for a broader class of models.

Stepwise Selection

- For computational reasons, best subset selection cannot be applied with very large p. *Why not?*
- 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

- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest *additional* improvement to the fit is added to the model.

In Detail

Forward Stepwise Selection

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   2.1 Consider all $p-k$ models that augment the predictors in $\mathcal{M}\;$ with one additional predictor.
   2.2 Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{\;+1}$. 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.

More on Forward Stepwise Selection

- 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.*

Credit data 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

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.*

Backward Stepwise Selection

- Like forward stepwise selection, *backward stepwise selection* provides an efficient alternative to best subset selection.
- However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

Backward Stepwise Selection: details

Backward Stepwise Selection

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   - 2.1 Consider all k models that contain all but one of the predictors in $\mathcal{M}\ $, for a total of $k-1$ predictors.
   - 2.2 Choose the *best* among these k models, and call it $\mathcal{M}_{\ -1}$. 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.

More on Backward Stepwise Selection

- Like forward stepwise selection, the backward selection approach searches through only $1 + p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection
- Like forward stepwise selection, backward stepwise selection is not guaranteed to yield the *best* model containing a subset of the p predictors.
- Backward selection requires that the *number of samples n is larger than the number of variables p* (so that the full model can be fit). In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

Choosing the Optimal Model

- The model containing all of the predictors will always have the smallest RSS and the largest R^2, since these quantities are related to the training error.
- We wish to choose a model with low test error, not a model with low training error. Recall that training error is usually a poor estimate of test error.
- Therefore, RSS and R^2 are not suitable for selecting the best model among a collection of models with different numbers of predictors.

Estimating test error: two approaches

- We can indirectly estimate test error by making an *adjustment* to the training error to account for the bias due to overfitting.
- We can *directly* estimate the test error, using either a validation set approach or a cross-validation approach, as discussed in previous lectures.
- We illustrate both approaches next.

C_p, AIC, BIC, and Adjusted R^2

- These techniques adjust the training error for the model size, and can be used to select among a set of models with different numbers of variables.
- The next figure displays C_p, BIC, and adjusted R^2 for the best model of each size produced by best subset selection on the `Credit` data set.

Now for some details

C_p*=RSS/(hsigma^2)+2d-n

C_p=1/n*hsigma^2(C_p*+n)

- *Mallow's C_p*:

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

where d is the total # of parameters used and $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ associated with each response measurement.

- The *AIC* criterion is defined for a large class of models fit by maximum likelihood:

$$\text{AIC} = -2\log L + 2 \cdot d$$

where L is the maximized value of the likelihood function for the estimated model.

- In the case of the linear model with Gaussian errors, maximum likelihood and least squares are the same thing, and C_p and AIC are equivalent. *Prove this.*

AIC=1/(n)(RSS+2*d*hsigma^2)

Details on BIC

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

- Like C_p, the BIC will tend to take on a small value for a model with a low test error, and so generally we select the model that has the lowest BIC value.
- Notice that BIC replaces the $2d\hat{\sigma}^2$ used by C_p with a $\log(n)d\hat{\sigma}^2$ term, where n is the number of observations.
- 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. See Figure on slide 19.

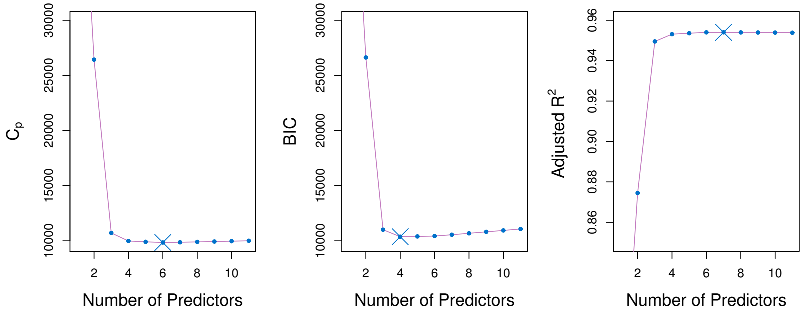
Adjusted R^2

- For a least squares model with d variables, the adjusted R^2 statistic is calculated as

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

  where TSS is the total sum of squares.
- Unlike C_p, AIC, and BIC, for which a *small* value indicates a model with a low test error, a *large* value of adjusted R^2 indicates a model with a small test error.
- Maximizing the adjusted R^2 is equivalent to minimizing $\frac{\text{RSS}}{n-d-1}$. While RSS always decreases as the number of variables in the model increases, $\frac{\text{RSS}}{n-d-1}$ may increase or decrease, due to the presence of d in the denominator.
- Unlike the R^2 statistic, the adjusted R^2 statistic *pays a price* for the inclusion of unnecessary variables in the model. See Figure on slide 19.

Credit data example



Validation and Cross-Validation

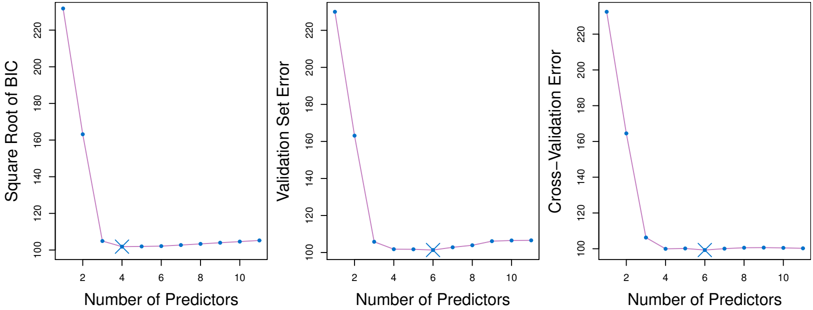
- Each of the procedures returns a sequence of models $\mathcal{M}_k$ indexed by model size $k = 0, 1, 2, \ldots$. Our job here is to select $\hat{k}$. Once selected, we will return model $\mathcal{M}_{\hat{k}}$

Validation and Cross-Validation

- Each of the procedures returns a sequence of models $\mathcal{M}_k$ indexed by model size $k = 0, 1, 2, \ldots$. Our job here is to select $\hat{k}$. Once selected, we will return model $\mathcal{M}_{\hat{k}}$
- We compute the validation set error or the cross-validation error for each model $\mathcal{M}_k$ under consideration, and then select the k for which the resulting estimated test error is smallest.
- This procedure has 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.

Credit data example

I



Excercise: Best subset using CP AIC,BIC, CV, Validation set

Details of Previous Figure

- 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.
- However, all three approaches suggest that the four-, five-, and six-variable models are roughly equivalent in terms of 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, and then select the smallest model for which the estimated test error is within one standard error of the lowest point on the curve. *What is the rationale for this?*

Shrinkage methods

- Gauss-Markov theorem: the OLS estimates have the smallest variance among all linear unbiased estimates.
- However, the restriction to unbiased estimates is not necessarily to a wise one, there may exist a biased estimator with smaller MSE. Such as ridge regression and lasso.
- Consider the mean squared error of an estimator $\tilde{\theta}$ in estimating θ

$$MSE(\tilde{\theta}) = E(\tilde{\theta} - \theta)^2 = \mathrm{Var}(\tilde{\theta}) + (E(\tilde{\theta}) - \theta)^2$$

- The expected prediction error of an estimate $\tilde{f}(x_0) = x_0^T\tilde{\beta}$ is

$$E\left(Y_0 - \tilde{f}(x_0)\right)^2 = \sigma^2 + MSE\left(\tilde{f}(x_0)\right)$$

Bias-Variance tradeoff, scarify bias to reduce variance and improve overall prediction accuracy

Shrinkage Methods

Best subset is a discrete process - variables are either retained or discarded - it often exhibits high variance.

Ridge regression and *Lasso*

- The subset selection methods use least squares to fit a linear model that contains a subset of the predictors.
- As an alternative, we can fit a model containing all p predictors using a technique that *constrains* or *regularizes* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero.
- It may not be immediately obvious why such a constraint should improve the fit, but it turns out that shrinking the coefficient estimates can significantly reduce their variance.

Ridge regression

- Recall that the least squares fitting procedure estimates $\beta_0, \beta_1, \ldots, \beta_p$ 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 coefficient estimates $\hat{\beta}^R$ are the values that minimize

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

where $\lambda \geq 0$ is a *tuning parameter*, to be determined separately.

Ridge regression

- Writing the criterion in matrix form

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

- the ridge regression solutions are easily seen to be

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

where I is the $p \times p$ identity matrix.

Ridge penalty was first introduced in statistics by Hoerl and Kennard, 1970

Ridge regression

- The singular value decomposition (SVD) of the centered input matrix X gives us some additional insight into the nature of ridge regression.
- The SVD of the $N \times p$ matrix X has the form $\underline{X = UDV^T}$. Here U and V are $N \times p$ and $p \times p$ orthogonal matrices, with the columns of U spanning the column space of X, and the columns of V spanning the row space. D is a $p \times p$ diagonal matrix, with diagonal entries $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ called the singular values of X. If one or more values $d_j = 0, X$ is singular.

Ridge regression

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

$$\begin{aligned}\mathbf{X}\hat{\beta}^{\text{ls}} &= \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{U}\mathbf{U}^T\mathbf{y},\end{aligned} \tag{3.46}$$

Now the ridge solutions are

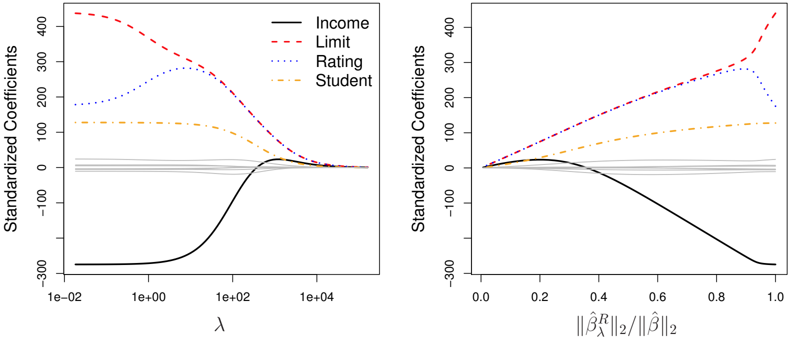
$$\begin{aligned}\mathbf{X}\hat{\beta}^{\mathsf{ridge}} &= \mathbf{X}\left(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{U}\mathbf{D}\left(\mathbf{D}^2+\lambda\mathbf{I}\right)^{-1}\mathbf{D}\mathbf{U}^T\mathbf{y} \\ &= \sum_{j=1}^{p}\mathbf{u}_j\frac{d_j^2}{d_j^2+\lambda}\mathbf{u}_j^T\mathbf{y},\end{aligned} \tag{3.47}$$

where the $\mathbf{u}_j$ are the columns of $\mathbf{U}$. Note that since $\lambda \geq 0$, we have $d_j^2/\left(d_j^2+\lambda\right) \leq 1$. Like linear regression, ridge regression computes the coordinates of $\mathbf{y}$ with respect to the orthonormal basis $\mathbf{U}$. It then shrinks these coordinates by the factors $d_j^2/\left(d_j^2+\lambda\right)$. This means that a greater amount of shrinkage is applied to the coordinates of basis vectors with smaller d_j^2.

Ridge regression: continued

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small.
- However, the second term, $\lambda \sum_j \beta_j^2$, called a *shrinkage penalty*, is small when $\beta_1, \ldots, \beta_p$ 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 is used for this.

Credit data example



Details of Previous Figure

- In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of λ.
- The right-hand panel displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying λ on the x-axis, we now display $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$, where $\hat{\beta}$ denotes the vector of least squares coefficient estimates.
- The notation $\|\beta\|_2$ denotes the ℓ_2 norm (pronounced "ell 2") of a vector, and is defined as $\|\beta\|_2 = \sqrt{\sum_{j=1}^p {\beta_j}^2}$.

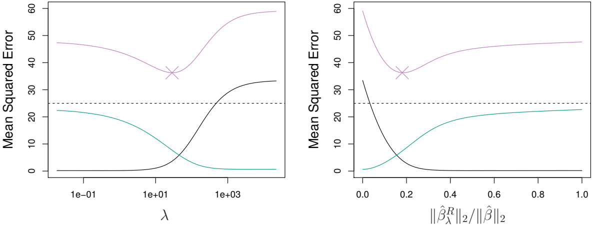
Ridge regression: scaling of predictors

- The standard least squares coefficient estimates are *scale equivariant*: multiplying X_j by a constant c simply leads to a scaling of the least squares coefficient estimates by a factor of $1/c$. In other words, regardless of how the jth predictor is scaled, $X_j\hat{\beta}_j$ will remain the same.
- In contrast, the ridge regression coefficient estimates can change *substantially* when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function.
- Therefore, it is best to apply ridge regression after *standardizing the predictors*, using the formula

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

Why Does Ridge Regression Improve Over Least Squares?

The Bias-Variance tradeoff



Simulated data with $n = 50$ observations, $p = 45$ predictors, all having nonzero coefficients. 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.

Lasso

- Tibshirani, R. (1996) Regression shrinkage and selection via lasso, JRSSB, 58,267-288.

Robert Tibshirani

Professor of Statistics
Professor of Biomedical Data Science

Mailing Address:
Department of Statistics
Sequoia Hall
390 Serra Mall
Stanford University
Stanford, CA 94305-4065

The Lasso

- Ridge regression does have one obvious disadvantage: unlike subset selection, which will generally select models that involve just a subset of the variables, ridge regression will include all p predictors in the final model
- The *Lasso* is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients, $\hat{\beta}^L_\lambda$, minimize the quantity

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

- In statistical parlance, the lasso uses an ℓ_1 (pronounced "ell 1") penalty instead of an ℓ_2 penalty. The ℓ_1 norm of a coefficient vector β is given by $\|\beta\|_1 = \sum|\beta_j|$.

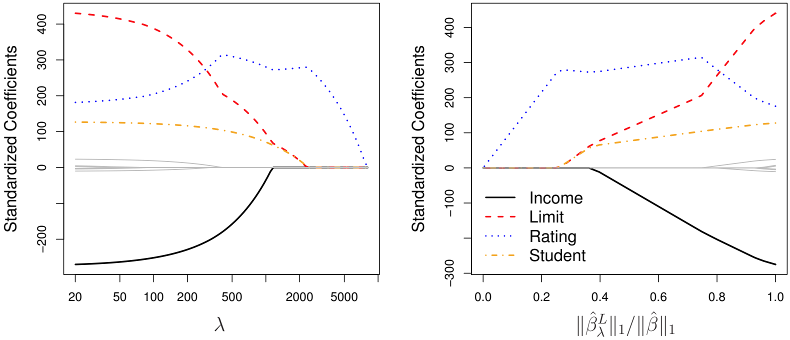
||\beta||_2=\sum |\beta|^2

||\beta||_0=\sum |\beta|^0

The Lasso: continued

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.
- 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*.
- We say that the lasso yields *sparse* models — that is, models that involve only a subset of the variables.
- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.

Example: Credit dataset



Non-negative garotte

- The motivation for the lasso came from non-negative garottee (Breiman, 1993).

$$\sum_{i=1}^{N}(y_i - \alpha - \sum_j c_j \hat{\beta}_j^0 x_{ij})^2, s.t. \quad c_j \geq 0, \sum c_j \leq t$$

- The garotte starts with the OLS estimates and shrinks them by non-negative factors whose sum is constrained.
- The garotte has consistently lower prediction error than subset selection and is competitive with ridge regression except when the true model has many small nonzero coefficients.
- The drawback of the garotte is that its solution depends on both the sign and the magnitude of the OLS estimates. In overfit or highly correlated settings where the OLS estimates behave poorly, the garotte may suffer as a result.

The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero?

One can show that the lasso and ridge regression coefficient estimates solve the problems

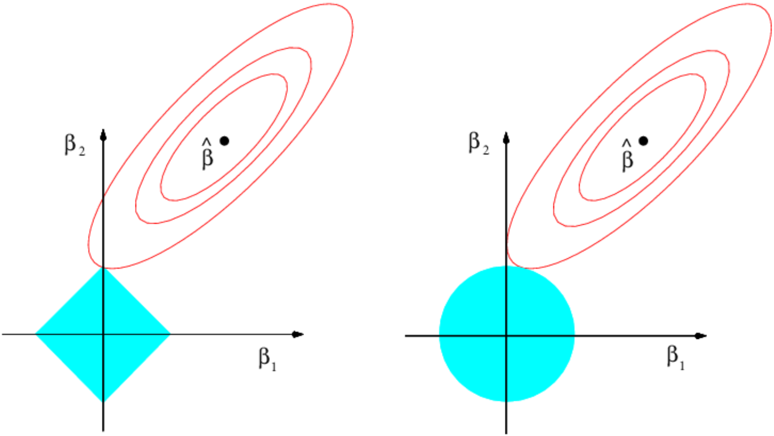
$$\underset{\beta}{\text{minimize}} \sum_{=1}^{n} \left(y_i - \beta_0 - \sum_{=1} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{=1}^{\mathrm{p}} |\beta_j| \leq s$$

and

$$\underset{\beta}{\text{minimize}} \sum_{=1}^{n} \left(y_i - \beta_0 - \sum_{=1} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{=1}^{\mathrm{p}} \beta^2 \leq s,$$

respectively.

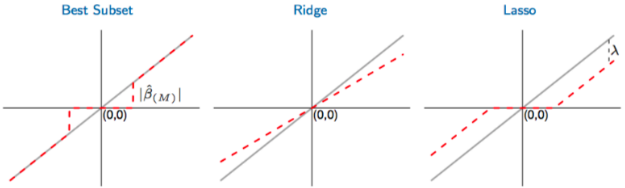
The Lasso Picture



Orthonormal design case

Estimators of β_j in the case of orthonormal columns of X (orthonormal matrix : $X^\top X = I$). sign denotes the sign of its argument (± 1), and x_+ denotes "positive part" of x.

Estimator	Formula
Best subset (size M)	$\hat{\beta}_j \cdot I\left(\left\lvert\hat{\beta}_j\right\rvert \geq \left\lvert\hat{\beta}_{(M)}\right\rvert\right)$ hard thresholding
Ridge	$\hat{\beta}_j/(1+\lambda)$
Lasso	$\operatorname{sign}\left(\hat{\beta}_j\right)\left(\left\lvert\hat{\beta}_j\right\rvert - \lambda\right)_+$ soft thresholding



Computation of Lasso solution:single predictor

- The lasso problem is a convex program, specifically a quadratic program (QP) with a convex constraint.

考虑一元情形 Let's first consider a single predictor setting, based on samples $\{(z_i, y_i)\}_{i=1}^N$ (for convenience we have given the name z_i to this single x_{i1}). The problem then is to solve

$$\underset{\beta}{\text{minimize}} \left\{ \frac{1}{2N} \sum_{i=1}^N (y_i - z_i\beta)^2 + \lambda|\beta| \right\}. \tag{2.9}$$

The standard approach to this univariate minimization problem would be to take the gradient (first derivative) with respect to β, and set it to zero. There is a complication, however, because the absolute value function $|\beta|$ does not have a derivative at $\beta = 0$. However we can proceed by direct inspection of the function (2.9), and find that

Excercise subgradient

$$\hat{\beta} = \begin{cases} \frac{1}{N}\langle \mathbf{z}, \mathbf{y}\rangle - \lambda & \text{if } \frac{1}{N}\langle \mathbf{z}, \mathbf{y}\rangle > \lambda, \\ 0 & \text{if } \frac{1}{N}|\langle \mathbf{z}, \mathbf{y}\rangle| \leq \lambda, \\ \frac{1}{N}\langle \mathbf{z}, \mathbf{y}\rangle + \lambda & \text{if } \frac{1}{N}\langle \mathbf{z}, \mathbf{y}\rangle < -\lambda. \end{cases} \tag{2.10}$$

(Exercise 2.2), which we can write succinctly as

$$\hat{\beta} = \mathcal{S}_\lambda\left(\frac{1}{N}\langle \mathbf{z}, \mathbf{y}\rangle\right). \tag{2.11}$$

Here the *soft-thresholding operator*

$$\mathcal{S}_\lambda(x) = \text{sign}(x)(|x| - \lambda)_+ \tag{2.12}$$

多元情形?

Coordinate descent algorithm for Lasso

The L_1 penalty makes the solutions nonlinear in the y_i, and there is no closed form expression as in ridge regression. Computing the lasso solution is a quadratic programming problem

Algorithm 1 Coordinate descent minimization

```
Let $\beta^{[0]} \in \mathbb{R}^p$ be an initial parameter vector. Set $m = 0$.
**repeat**
Increase $m$ by one: $m \leftarrow m + 1$.
Denote by $\mathscr{S}^{[m]}$ the index cycling through the coordinates $\{1, \cdots, p\}$:
$\mathscr{S}^{[m]} = \mathscr{S}^{[m-1]} + 1 \bmod p$. Abbreviate by $j = \mathscr{S}^{[m]}$ the value of $\mathscr{S}^{[m]}$.
if $\left| G_j \left( \beta_{-j}^{[m-1]} \right) \right| \leq \lambda$ : set $\beta_j^{[m]} = 0$,
 otherwise: $\beta_j^{[m]} = \arg\min_{\beta_j} Q_\lambda(\beta_{+j}^{[m-1]})$,
where $\beta_{-j}^{[m-1]}$ is the parameter vector where the $j$th component is set to zero and
$\beta_{+j}^{[m-1]}$ is the parameter vector which equals $\beta^{[m-1]}$ except for the $j$th component
where it is equal to $\beta_j$ (i.e. the argument we minimize over).
**until** numerical convergence
```

Coordinate descent algorithm for Lasso

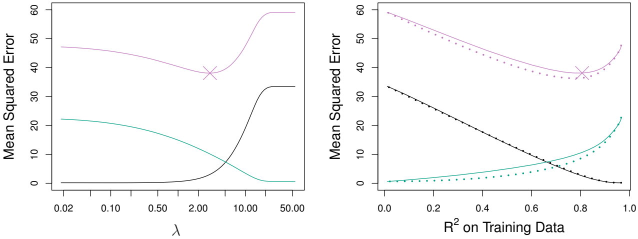
The coordinatewise optimization above can easily incorporate the more general case where some parameters are unpenalized, i.e.,

$$\hat{\beta} = \arg\min_{\beta} Q_\lambda(\beta)$$
$$Q_\lambda(\beta) = \|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda \textstyle\sum_{j=r+1}^{p} |\beta_j|$$

and thus, $\beta_1, \ldots, \beta_r$ are unpenalized. The up-dating step in the optimization algorithm then looks as follows:

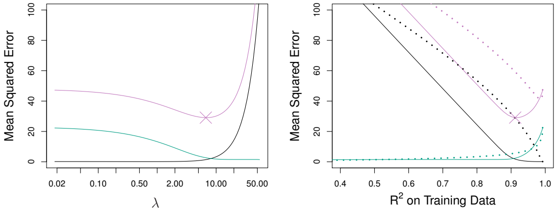
$$\begin{aligned}
&\text{if } j \in \{1, \ldots, r\} : \beta_j^{[m]} = \arg\min_{\beta_j} Q_\lambda\left(\beta_{+j}^{[m-1]}\right), \\
&\text{if } j \in \{r+1, \ldots, p\} : \\
&\qquad \text{if } \left| G_j\left(\beta_{-j}^{[m-1]}\right)\right| \leq \lambda : \text{set } \beta_j^{[m]} = 0, \\
&\qquad \text{otherwise: } \beta_j^{[m]} = \arg\min_{\beta_j} Q_\lambda\left(\beta_{+j}^{[m-1]}\right).
\end{aligned}$$

Comparing the Lasso and Ridge Regression



Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on simulated data set of Slide 32.*
Right: *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). 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.*

Comparing the Lasso and Ridge Regression: continued



Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Slide 38, 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 (dashed). 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.*

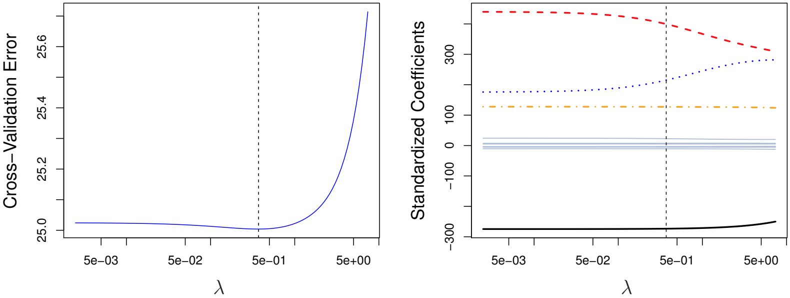
Conclusions

- These two examples illustrate that neither ridge regression nor the lasso will universally dominate the other.
- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors.
- However, the number of predictors that is related to the response is never known *a priori* for real data sets.
- A technique such as cross-validation can be used in order to determine which approach is better on a particular data set.

Selecting the Tuning Parameter for Ridge Regression and Lasso

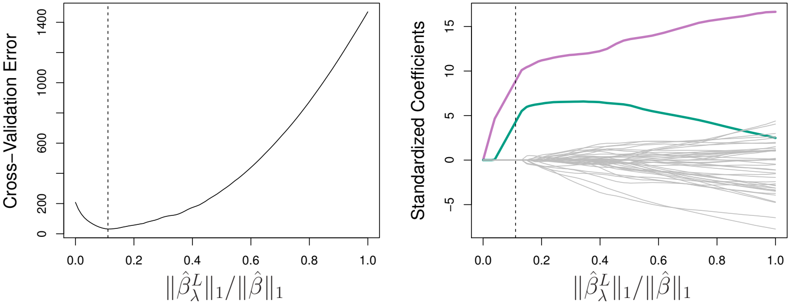
- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration is best.
- That is, we require a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint s.
- *Cross-validation* provides a simple 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.

Credit data example



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 indicates the value of* λ *selected by cross-validation.*

Simulated data example



Left: Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Slide 39. Right: *The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

Relaxed Lasso

- Least-squares fit on the subset of the three predictors tends to expand the lasso estimates away from zero.
- The nonzero estimates from the lasso tend to be biased toward zero, so the debiasing in the right panel can often improve the prediction error of the model. This two-stage process is also known as the relaxed lasso (Meinshausen 2007).

Table 2.2 *Results from analysis of the crime data. Left panel shows the least-squares estimates, standard errors, and their ratio (Z-score). Middle and right panels show the corresponding results for the lasso, and the least-squares estimates applied to the subset of predictors chosen by the lasso.*

	LS coef	SE	Z	Lasso	SE	Z	LS	SE	Z
funding	10.98	3.08	3.6	8.84	3.55	2.5	11.29	2.90	3.9
hs	-6.09	6.54	-0.9	-1.41	3.73	-0.4	-4.76	4.53	-1.1
not-hs	5.48	10.05	0.5	3.12	5.05	0.6	3.44	7.83	0.4
college	0.38	4.42	0.1	0.0	-	-	0.0	-	-
college4	5.50	13.75	0.4	0.0	-	-	0.0	-	-

SCAD

A good penalty function should result in an estimator with three properties.

1.*Unbiasedness*: The resulting estimator is nearly unbiased when the true unknown parameter is large to avoid unnecessary modeling bias.

2.*Sparsity*: The resulting estimator is a thresholding rule, which automatically sets small estimated coefficients to zero to reduce model complexity.

3.*Continuity*: The resulting estimator is continuous in data z to avoid instability in model prediction.

Reference: Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties[J]. Journal of the American statistical Association, 2001, 96(456): 1348-1360.

SCAD

- Suppose that we have a linear model

$$Y = X\beta + \epsilon$$

  and the columns of X are orthonormal.
- Then $\hat{\beta} = X^T Y$.
- The objective function for penalized least squares

$$\mathrm{Q}\left(\beta\right) = \tfrac{1}{2}\left\|\mathrm{y} - \mathrm{X}\beta\right\|^2 + \lambda \sum_{j=1}^{p} p_j\left(\left|\beta_j\right|\right)$$

$$= \tfrac{1}{2}\left\|y - \hat{y} + \hat{y} - X\beta\right\|^2 + \lambda \sum_{j=1}^{p} p_j\left(\left|\beta_j\right|\right)$$

$$= \tfrac{1}{2}\left\|y - \hat{y}\right\|^2 + \tfrac{1}{2}\left\|Xz - X\beta\right\|^2 + \lambda \sum_{j=1}^{p} p_j\left(\left|\beta_j\right|\right.$$

$$= \tfrac{1}{2}\left\|y - \hat{y}\right\|^2 + \tfrac{1}{2}\sum_{j=1}^{p}\left(z_j - \beta_j\right)^2 + \lambda \sum_{j=1}^{p} p_j\left(\left|\beta_j\right|\right)$$

SCAD

- The first term is constant with respect to β, so minimizing the object $Q(\beta)$ reduces to a componentwise regression problem.
- The minimization problem of penalized least squares is equivalent to minimizing componentwise

$$Q\left(\theta\right)=\frac{1}{2}(z-\theta)^{2}+p_{\lambda}\left(|\theta|\right)$$

- Then take derivative

$$\begin{array}{l}\frac{dQ(\theta)}{d\theta}=\left(\theta-z\right)+\operatorname{sgn}\left(\theta\right)p_{\lambda}'\left(|\theta|\right)\\=\operatorname{sgn}\left(\theta\right)\left\{|\theta|+p_{\lambda}'\left(|\theta|\right)\right\}-z\end{array}$$

- and solve $\frac{dQ(\theta)}{d\theta}=0$ to get the minimizer of $Q(\theta)$

SCAD

Unbiasedness condition:$p'_\lambda(|\theta|) = 0$ for large $|\theta|$
Since

$$\frac{dQ\left(\theta\right)}{d\theta} = \left(\theta - z\right) + \operatorname{sgn}\left(\theta\right) p'_\lambda\left(|\theta|\right)$$

It is easy to see that when $p'_\lambda(|\theta|) = 0$ for large $|\theta|$, the resulting estimator is z when $|z|$ is sufficiently large, which is that $\hat{\theta} = z$.

SCAD

Sparsity condition:$min_\theta\{|\theta| + p'_\lambda(|\theta|)\} > 0$

sparsity $\Leftrightarrow$ when z is small,0 is the minimizer of$Q(\theta)$

$$\Leftrightarrow \frac{dQ(\theta)}{d\theta} > 0 \text{ when } \theta > 0 \& \frac{dQ(\theta)}{d\theta} < 0 \text{ when } \theta < 0$$

$$\Leftrightarrow |\theta| + p'_\lambda(|\theta|) > z \text{ when } \theta > 0 \& |-\theta| + p'_\lambda(|\theta|) < z \text{ when } \theta <$$

$$\Leftrightarrow \text{ when } |z| < min_{\theta \neq 0}\{|\theta| + p'_\lambda(|\theta|)\}, \hat{\theta} = 0$$

Thus, we need the condition $min_\theta\{|\theta| + p'_\lambda(|\theta|)\} > 0$ so that the resulting estimator can automatically set small estimated coefficients to zero.

SCAD

Continuity Condition: $argmin_{\theta \neq 0}\{|\theta| + p'_\lambda(|\theta|)\} = 0$

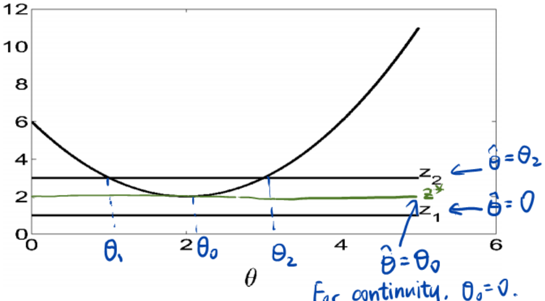
- when

$$|\theta| + p'_\lambda(|\theta|) > |z| \Rightarrow \hat{\theta} = 0$$

- when

$$|\theta| + p'_\lambda(|\theta|) = |z| \Rightarrow \hat{\theta} = \theta_0$$

Thus for continuity, we need θ_0 goes to zero, which is $argmin_{\theta \neq 0}\{|\theta| + p'_\lambda(|\theta|)\} = 0$



SCAD

- Fan and Li (2001) showed that the lasso can perform automatic variable selection because the ℓ_1 penalty is singular at the origin.
- On the other hand, the lasso shrinkage produces biased estimates for the large coefficients, and thus it could be suboptimal in terms of estimation risk.
- Fan and Li (2001) proposed a smoothly clipped absolute deviation (SCAD) penalty for variable selection

$$p'_\lambda(\theta) = \lambda\{I(\theta \leq \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} I(\theta > \lambda)\}$$

$$P(|\theta|; \lambda, a) = \begin{cases} \lambda|\theta| & \text{if } 0 \leq |\theta| < \lambda \\ -\frac{\theta^2 - 2a\lambda|\theta| + \lambda^2}{2(a-1)} & \text{if } \lambda \leq |\theta| < a\lambda \\ (a+1)\lambda^2/2 & \text{otherwise} \end{cases}$$

SCAD

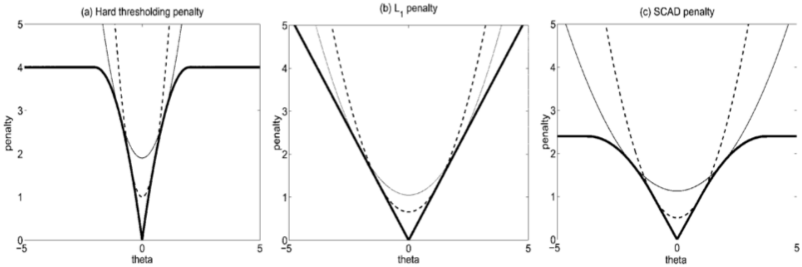


Figure 1. Three Penalty Functions $p_\lambda(\theta)$ and Their Quadratic Approximations. The values of λ are the same as those in Figure 5(c).

SCAD

This penalty function leaves large values of θ not excessively penalized and makes the solution continuous. The resulting solution is given by

$$\hat{\theta} = \begin{cases} \operatorname{sgn}(z)(|z| - \lambda)_+, & \text{when } |z| \leq 2\lambda, \\ \{(a-1)z - \operatorname{sgn}(z)a\lambda\}/(a-2), & \text{when } 2\lambda < |z| \leq a\lambda, \\ z, & \text{when } |z| > a\lambda. \end{cases}$$

SCAD

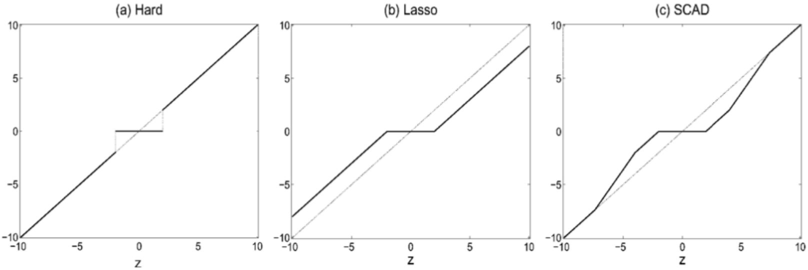


Figure 2. Plot of Thresholding Functions for (a) the Hard, (b) the Soft, and (c) the SCAD Thresholding Functions With $\lambda = 2$ *and* $a = 3.7$ *for SCAD.*

Theorem:oracle property

- For generalized linear models, assume $\boldsymbol{V_1}, \ldots, \boldsymbol{V_n}$ are i.i.d with density $f_i(g(x_i^T\beta), y_i)$,where $\boldsymbol{V_i} = (\boldsymbol{X_i}, Y_i)$
- Let $\boldsymbol{\beta_0} = (\beta_{10}, \ldots, \beta_{d0})^T = (\boldsymbol{\beta_{10}^T}, \boldsymbol{\beta_{20}^T})^T$, assume $\boldsymbol{\beta_{20}} = 0$
- let $I(\beta_0)$ be the fisher information matrix and let $I_1(\boldsymbol{\beta_{10}}, 0)$ be the fisher information knowing $\boldsymbol{\beta_{20}} = 0$
- $a_n = max\{p'_\lambda(|\beta_{j0}|) : \beta_{j0} \neq 0\}$
- Let $\boldsymbol{L(\beta)}$ be the log-likelihood function of $\boldsymbol{V_1}, \ldots, \boldsymbol{V_n}$

  $$Q(\beta) = L(\beta) - n\sum_{j=1}^{d} p_{\lambda n}(|\beta_j|)$$

Theorem:oracle property

Theorem

conditons(A)-(C) are satisfied. if $\lambda_n \to 0$ *and* $\sqrt{n}\lambda_n \to \infty$ *as* $n \to \infty$*,then with probability tending to 1,the root-n consistent local maximizers* $\hat{\beta} = \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix}$ *in Theorem 1 must satisfy:*

(a)Sparsity:$\hat{\beta}_2 = 0$
(b)Asympototic normality:

$$\sqrt{n}(I_1(\beta_{\mathbf{10}}) + \Sigma)\{\hat{\beta}_{\mathbf{1}} - \beta_{\mathbf{10}} + (I_1(\beta_{\mathbf{10}}) + \Sigma)^{-1}\mathbf{b}\} \to N\{\mathbf{0}, I_1(\beta_{\mathbf{10}})\}$$

Adaptive Lasso

- We can certainly assign different weights to different coefficients.
- Weighted lasso

$$\underset{\boldsymbol{\beta}}{\arg\min} \quad \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda \sum_{j=1}^{p} w_j \left| \beta_j \right|,$$

  where **w** is a known weights vector.
- We show that if the weights are data-dependent and cleverly chosen, then the weighted lasso can have the oracle properties.

Reference: Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties[J]. Journal of the American statistical Association, 2001, 96(456): 1348-1360.

Adaptive Lasso

- We show that if the weights are data-dependent and cleverly chosen, then the weighted lasso can have the oracle properties.
- Suppose that $\hat{\boldsymbol{\beta}}$ is a root-n -consistent estimator to $\boldsymbol{\beta}^*$; for example, we can use $\hat{\boldsymbol{\beta}}$ (ols). Let $\mathcal{A}_n^* = \left\{ j : \hat{\beta}_j^{*(n)} \neq 0 \right\}$

  Pick a $\gamma > 0$, and define the weight vector $\hat{\mathbf{w}} = 1/|\hat{\boldsymbol{\beta}}|^\gamma$. The adaptive lasso estimates $\hat{\boldsymbol{\beta}}^{*(n)}$ are given by

$$\hat{\boldsymbol{\beta}}^{*(n)} = \arg\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda_n \sum_{j=1}^{p} \hat{w}_j \left| \beta_j \right| . \qquad (4)$$

- It is worth emphasizing that (4) is a convex optimization.

Reference: Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties[J]. Journal of the American statistical Association, 2001, 96(456): 1348-1360.

Adaptive Lasso

Theorem 2 (Oracle properties). Suppose that $\lambda_n/\sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$. Then the adaptive lasso estimates must satisfy the following:

1. Consistency in variable selection: $\lim_n P\left(\mathcal{A}_n^* = \mathcal{A}\right) = 1$

2. Asymptotic normality:

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}_{\mathcal{A}}^{*(n)} - \boldsymbol{\beta}_{\mathcal{A}}^*\right) \to_d \mathrm{N}\left(\mathbf{0}, \sigma^2 \times \mathbf{C}_{11}^{-1}\right).$$

Adaptive Lasso

Algorithm 1 (The LARS algorithm for the adaptive lasso).

1. Define $\mathbf{x}_j^{**} = \mathbf{x}_j / \hat{w}_j, j = 1, 2, \ldots, p$.
2. Solve the lasso problem for all λ_n,

$$\hat{\boldsymbol{\beta}}^{**} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j^{**} \beta_j \|^2 + \lambda_n \sum_{j=1}^{p} |\beta_j|$$

3. Output $\hat{\beta}_j^{*(n)} = \hat{\beta}_j^{**} / \hat{w}_j, j = 1, 2, \ldots, p.$

Adaptive Lasso

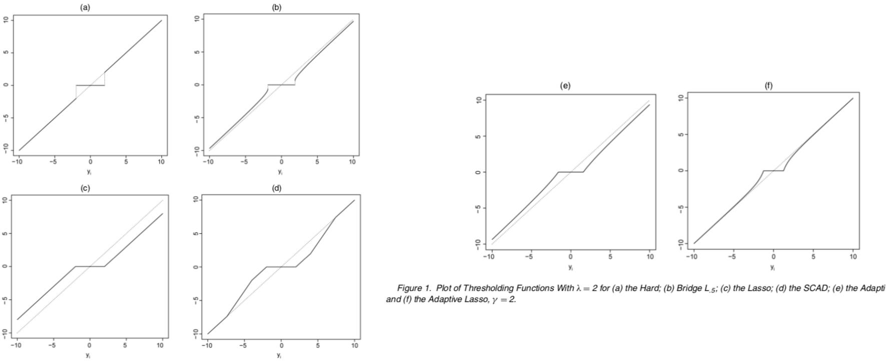


Figure 1. Plot of Thresholding Functions With $\lambda = 2$ for (a) the Hard; (b) Bridge $L_{.5}$; (c) the Lasso; (d) the SCAD; (e) the Adaptive Lasso $\gamma = .5$; and (f) the Adaptive Lasso, $\gamma = 2$.

Adaptive Lasso

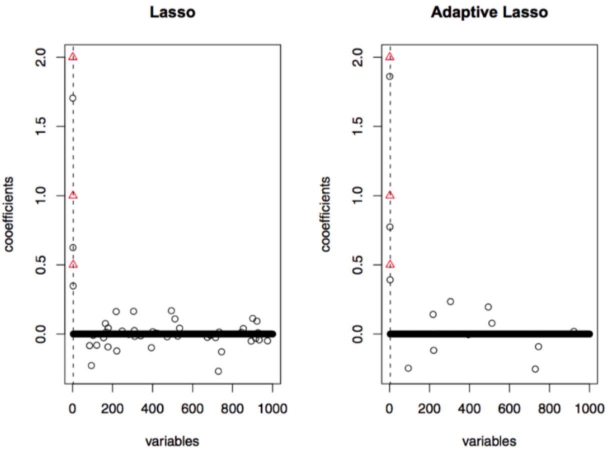


Fig. 2.4 Estimated regression coefficients in the linear model with $p = 1000$ and $n = 50$. Left: Lasso. Right: Adaptive Lasso with Lasso as initial estimator. The 3 true regression coefficients are indicated with triangles. Both methods used with tuning parameters selected from 10-fold cross-validation.

Generalization of shrinkage methods

- We can generalize ridge regression and the lasso. Consider the criterion

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

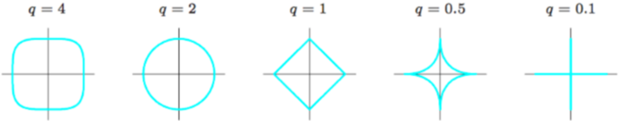


FIGURE 3.12. *Contours of constant value of $\sum_j |\beta_j|^q$ for given values of q.*

- $q > 1, |\beta_j|^q$ is differentiable at 0, and so does not share the ability of lasso.

Elastic net

- Although the lasso has shown success in many situations, it has some limitations.
- Consider the following three seenarios.
  - (a) 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 limititing feature for a variable selection method. Moreovar, the lasso is no well defined unless the bound on the L_1 -norm of the coefficients is smaller rhan a certain value.
  - (b) 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. See Section 2.3.
  - (c) For usual $n > p$ situations, if there are high correlations between predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression (Tibshirani, 1996.)

Reference: Zou H, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the royal statistical society: series B (statistical methodology), 2005, 67(2): 301-320.

Elastic net

- Zou and Hastie (2005) introduced the elastic net penalty

$$\lambda \sum_{j=1}^{p} \left(\alpha \beta_j^2 + (1-\alpha) \left| \beta_j \right| \right)$$

  a different compromise between ridge and lasso.

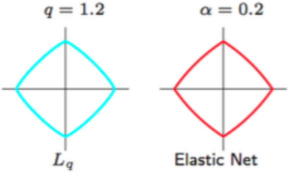


FIGURE 3.13. *Contours of constant value of $\sum_j |\beta_j|^q$ for $q = 1.2$ (left plot), and the elastic-net penalty $\sum_j (\alpha\beta_j^2 + (1-\alpha)|\beta_j|)$ for $\alpha = 0.2$ (right plot). Although visually very similar, the elastic-net has sharp (non-differentiable) corners, while the $q = 1.2$ penalty does not.*

Elastic net

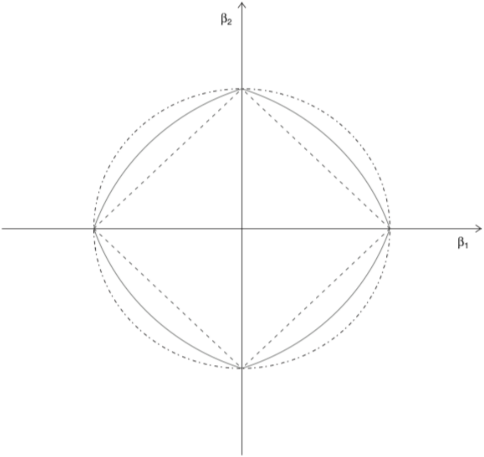


Fig. 1. Two-dimensional contour plots (level 1) (· - - · - - -, shape of the ridge penalty; - - - - - - -, contour of the lasso penalty; ———, contour of the elastic net penalty with $\alpha = 0.5$): we see that singularities at the vertices and the edges are strictly convex; the strength of convexity varies with α

Minimax concave penalty (MCP)

- Zhang(2010) proposed MCP penalty

$$P(\beta|;\lambda,a)=\begin{cases}\lambda|\beta|-\frac{|\beta|^2}{2a},\ \text{if } |\beta|\leq a\lambda\\ \frac{a\lambda^2}{2},\ \text{if } |\beta|>a\lambda\end{cases}$$

$$P'_{\lambda,a}(|\beta|;\lambda,a)=\begin{cases}\lambda-\frac{|\beta|}{a},\ \text{if } |\beta|\leq a\lambda,\\ 0,\qquad \text{if } |\beta|>a\lambda.\end{cases}$$

Reference: Zhang C H. Nearly unbiased variable selection under minimax concave penalty[J]. The Annals of statistics, 2010, 38(2): 894-942.

Group selection in high dimensional models

- Group selection

  group lasso, 2-norm group bridge, 2-norm group SCAD, 2-norm group MCP,...

- Bi-level selection

  concave 1-norm group penalties, composite penalties, additive penalties.

Reference: Huang J, Breheny P, Ma S. A selective review of group selection in high-dimensional models[J]. Statistical science: a review journal of the Institute of Mathematical Statistics, 2012, 27(4).

Group lasso

- Yuan and Lin (2006) propose group LASSO penalty

For a column vector $\mathbf{v} \in \mathbb{R}^d$ with $d \geq 1$ and a positive definite matrix R, denote $\|\mathbf{v}\|_2 = (\mathbf{v}'\mathbf{v})^{1/2}$ and $\|\mathbf{v}\|_R = (\mathbf{v}'R\mathbf{v})^{1/2}$. Let $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_J')'$, where $\boldsymbol{\beta}_j \in \mathbb{R}^{d_j}$. The group LASSO solution $\hat{\boldsymbol{\beta}}(\lambda)$ is defined as a minimizer of

$$\frac{1}{2n} \left\| \mathbf{y} - \sum_{j=1}^{J} X_j \boldsymbol{\beta}_j \right\|_2^2 + \lambda \sum_{j=1}^{J} c_j \|\boldsymbol{\beta}_j\|_{R_j} \qquad (2.1)$$

where $\lambda \geq 0$ is the penalty parameter and R_j 's are $d_j \times d_j$ positive definite matrices. Here the c_j 's in the penalty are used to adjust for the group sizes. A reasonable choice is $c_j = \sqrt{d_j}$. Because (2.1) is convex, any local minimizer of (2.1) is also a global minimizer and is characterized by the Karush-Kuhn-Tucker conditions as given in Yuan and Lin(2006).

Group lasso

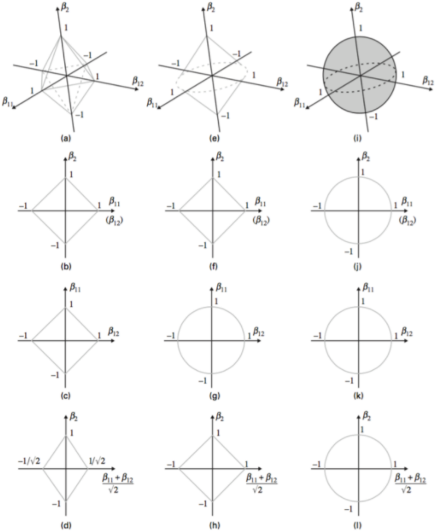


Fig. 1. (a)–(d) l_1-penalty, (e)–(h) group lasso penalty and (i)–(l) l_2-penalty

Concave 2-norm group selection

- A more general class of group selection methods can be based on the criterion

$$\frac{1}{2n}\left\|\mathbf{y}-\sum_{j=1}^{J}X_j\boldsymbol{\beta}_j\right\|_2^2+\sum_{j=1}^{J}\rho\left(\|\boldsymbol{\beta}_j\|_{R_j};c_j\lambda,\gamma\right) \tag{2.3}$$

  where $\rho\,(t;c_j\lambda,\gamma)$ is concave in t. Here γ is an additional tuning parameter that may be used to modify ρ.
- Specifically, for $\rho(t;\lambda)=\lambda|t|$, the group lasso penalty can be written as $\lambda c_j\,\|\beta_j\|_{R_j}=\rho\left(\|\boldsymbol{\beta}_j\|_{R_j};c_j\lambda\right)$.

Concave 2-norm group selection

Other penalty functions could be used instead.

- (a) the bridge penalty with

  $$\rho(x; \lambda, \gamma) = \lambda |x|^{\gamma}, 0 < \gamma \leq 1$$

  (Frank and Friedman, 1993);
- (b) the SCAD penalty with

  $$\rho(x; \lambda, \gamma) = \lambda \int_0^{|x|} \min(1, (\gamma - t/\lambda)_+/(\gamma - 1)\} \, dt, \gamma > 2$$

  (Fan and Li, 2001; Fan and Peng, 2004), where for any $a \in \mathbb{R}, a_+$ denotes its positive part, that is, $a_+ = a1_{(a \geq 0)}$;
- (c) the minimax concave penalty (MCP) with $\rho(x; \lambda, \gamma) = \lambda \int_0^{|x|} (1 - t/(\gamma\lambda))_+ dt, \gamma > 1$ (Zhang, 2010a).

Concave 1-norm group penalties

- The 1-norm group bridge applies a bridge penalty to the ℓ_1 norm of a group, resulting in the criterion

$$\frac{1}{2n}\left\|\mathbf{y}-\sum_{j=1}^{J}X_j\boldsymbol{\beta}_j\right\|_2^2+\lambda\sum_{j=1}^{J}c_j\left\|\boldsymbol{\beta}_j\right\|_1^{\gamma}$$

  where $\lambda > 0$ is the regularization parameter, $\gamma \in (0,1)$ is the bridge index and $\{c_j\}$ are constants that adjust for the dimension of group j. For models with standardized variables, a reasonable choice is $c_j = |d_j|^{\gamma}$.

- General penalized criterion

$$\frac{1}{2n}\left\|\mathbf{y}-\sum_{j=1}^{J}X_j\boldsymbol{\beta}_j\right\|_2^2+\sum_{j=1}^{J}\rho\left(\left\|\boldsymbol{\beta}_j\right\|_1; c_j\lambda,\gamma\right)$$

Composite penalties

- The composite MCP uses the criterion

$$\frac{1}{2n}\left\|\mathbf{y}-\sum_{j=1}^{J} X_j\boldsymbol{\beta}_j\right\|_2^2+\sum_{j=1}^{J}\rho_{\lambda,\gamma_O}\left(\sum_{k=1}^{d_j}\rho_{\lambda,\gamma_I}\left(|\beta_{jk}|\right)\right)$$

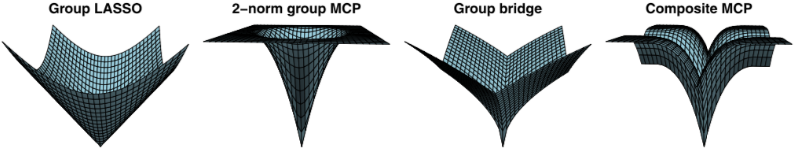


FIG. 2. *The group LASSO, group bridge and composite mcp penalties for a two-predictor group. Note that where the penalty comes to a point or edge, there is the possibility that the solution will take on a sparse value; all penalties come to a point at* **0**, *encouraging group-level sparsity, but only group bridge and composite MCP allow for bi-level selection.*

Additive penalties

- Another approach to achieving bi-level selection is to add an ℓ_1 penalty to the group lasso (Wu and Lange, 2008; Friedman, Hastie and Tibshirani, 2010).
- Friedman, Hastie and Tibshirani (2010) proposed sparse group lasso

$$\min_{\beta \in \mathbb{R}^p} \left(\left\| \mathbf{y} - \sum_{l=1}^{L} \mathbf{X}_\ell \beta_\ell \right\|_2^2 + \lambda_1 \sum_{\ell=1}^{L} \|\beta_\ell\|_2 + \lambda_2 \|\beta\|_1 \right) \tag{2}$$

where $\beta = (\beta_1, \beta_2, \ldots, \beta_\ell)$ is the entire parameter vector. For notational simplicity we omit the weights $\sqrt{p_\ell}$. Expression (2) is the sum of convex functions and is therefore convex. When $\lambda_2 = 0$, criterion (2) reduces to the group lasso.

adSGL

- Fang, Wang and Ma(2015) propose adaptive sparse group lasso

$$\min\left\{\frac{1}{2}\left\|y-\sum_{j=1}^{J}X_j\beta^{(j)}\right\|_2^2+\lambda(1-\alpha)\sum_{j=1}^{J}w_j\left\|\beta^{(j)}\right\|_2+\lambda\alpha\sum_{j=1}^{J}\xi^{(j)T}\left|\beta^{(j)}\right|\right\}$$

where $W=(w_1,\cdots,w_J)^T\in R_+^J$ is the group weight vector, $\xi^T=\left(\xi^{(1)T},\cdots,\xi^{(J)T}\right)=\left(\xi_1^{(1)},\cdots,\xi_{p1}^{(1)},\cdots,\zeta_1^{(J)},\cdots,\xi_{p,J}^{(J)}\right)\in R_+^p$ denote the individual weights, and $\lambda\in R_+$ is the tuning parameter. For different groups, the penalty level can be different. By adopting lower penalty for large coefficients while higher penalty for small ones, we expect this to be able to improve variable selection accuracy and reduce estimation bias.

adSGL

- We use the group bridge estimator to construct these two type of weights

$$w_j = \left(\left\| \hat{\beta}^{(j)}(GB) \right\|_1 + \tfrac{1}{n} \right)^{-1}$$

$$\xi_i^{(j)} = \left(\left| \hat{\beta}_i^{(j)}(GB) \right| + \tfrac{1}{n} \right)^{-1}$$

Structured Sparse Logistic Regression with Application to Lung Cancer Prediction

Kuangnan Fang,
Department of Statistics, School of Economics, Xiamen University
joint work with Zhang Xiaochen, Zhang Qingzhao, Wang Xiaofeng, Ma Shuangge

Nov. 2019

Outline

1. Introduction
2. Methodology
   - The Model Setting and Methodology
   - Computational algorithm
3. Simulation and Application
   - Simulation
   - Application to breath VOCs
4. Discussion

Background

- Lung cancer accounts for around 28% of cancer-related deaths worldwide.
- The chances of survival are narrow and at the early stages, it cannot be detected due to the presence of little or no symptoms.
- More than 60% of patients have their symptoms diagnosed at the later stages with longevity of less than 10%.

Detection techniques

- Symptoms in the last stage of lung cancer: persistent cough, blood filled sputum, pain in the chest, change in the voice pattern and recurrent pneumonia or bronchitis.
- The techniques used for its detection and diagnosis is expensive.The current techniques used are:
    - Chest X-ray
    - Sputum Cytology
    - Pulmonary Function Tests (PFT)
    - Chest Tomography
    - Bronchoscopy with Biopsy
    - ...

History of breath analysis

- Since ancient times, physicians have known that a person's breath gives an indication of one's health conditions.
- Collection of a breath sample is much safer and easier compared to that of collecting blood samples or urine samples.
- The compounds present in the breath can be detected and correlated to various diseases.
- During the last few years, the analysis of exhaled breath has been proposed as a novel option for an early detection of lung cancer (Mazzone et al. (2007), Peng et al. (2009), Mazzone et al. (2012)).

Exhaled breath VOC analyzers

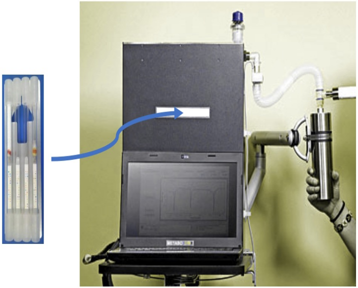
- Exhaled breath contains a complex mixture of several hundreds of **volatile organic compounds (VOCs).**It has been shown that the features obtained from VOCs can be used as a non-invasive marker of lung cancer.
    - GC and Mass Spectrometry (GC-MS)
    - Electronic Noses
    - Quartz Microbalance
    - Colorimetry
    - Ion Mobility Spectrometry (IMS)
    - Cyranose 320
    - NANO-NOSE
    - ...

Colorimetric Sensor Array (CSA)

- **Colorimetric sensor array (CSA)**(Janzen et al. (2006)) is composed of chromogenic reagents printed on a disposable cartridge.
- Third generation (2013) CSA
  - 128 chemically sensitive colorants impregnated on a disposable cartridge.
  - Improved Nanoporous matrix for chemically reactive colorants: high porosity and enormous surface area at nanoscale results in enhanced sensitivity to lung cancer indicators
  - New categories of chemical indicators have expanded the breadth of chemical sensitivity of the array
  - Robotically printed array – higher precision, signal/noise and sensitivity

CSA Breath Data

- The measurement from the sensor is a change in the colors of its elements every tidal volume (approximately 500 mL) breath air crossed.
- Study subjects were recruited prospectively from outpatient clinics at the Cleveland Clinic, US.



- Tidal volume breathing for 5 minutes;
- Exhaled breath drawn over the sensor array;
- Images were converted to numerical values in the red, green, blue spectra, and 4 ultraviolet spectra.
- Totally 128 (the number of colorants) $\times$ 7(changes in the red, green, blue, and 4 ultra-color spectrum of each colorant) $=$ 896 groups.

Data Examples

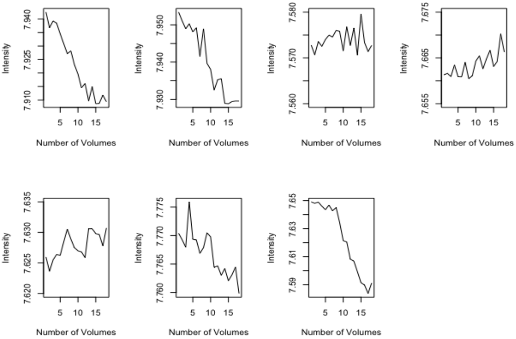


Figure: An example of the observed log intensities for one subject. Each image of the sensor was converted to numerical values for changes in the red, green, blue, and 4 ultra-color spectrum of each colorant.

Introduction

- Grouping structures.
  - *Group variable selection problem, see group Lasso (Yuan and Lin (2006), Meier et al. (2008)),CAP(Zhao et al. (2006)), group MCP (Wang et al. (2008) , Huang et al. (2012)), group bridge(Huang et al. (2009)) and references therein.*
- Encourage smoothness of coefficients.
  - *Features are ordered in some meaningful ways, see fused Lasso (Tibshirani et al. (2005)), smooth-Lasso(Hebiri and Van (2011)), sparse Laplacian shrinkage (SLS) method(Huang et al. (2011)), spline-lasso (Guo et al. (2016)).*

Outline

1 Introduction

2 Methodology
- The Model Setting and Methodology
- Computational algorithm

3 Simulation and Application
- Simulation
- Application to breath VOCs

4 Discussion

The Model Setting

- n subjects, $(\boldsymbol{z}_i, \boldsymbol{x}_i, y_i; i = 1, \cdots, n)$, where $\boldsymbol{z}_i \in \mathbb{R}^{q_0}$ is a q_0-dimensional vector , $\boldsymbol{x}_i \in \mathbb{R}^{q}$ is a q-dimensional vector and $y_i \in \{0, 1\}$ is a binary response variable.
- $\boldsymbol{z}_i$ includes intercept and other scaler vectors.
- $\boldsymbol{x}_i$ can be divided into J groups, which means there exist grouping structures: $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x_n})^\top$ with $\boldsymbol{x}_i = (\boldsymbol{x}_{i,1}^\top, \cdots, \boldsymbol{x}_{i,J}^\top)^\top$ for $i = 1, \cdots, n$.
- Log-likelihood function of linear logistic regression models:

$$\ell(\beta) = \sum_{i=1}^{n} \{y_i log(p_\beta(\boldsymbol{x}_i, \boldsymbol{z}_i) + (1 - y_i) log(1 - p_\beta(\boldsymbol{x}_i, \boldsymbol{z}_i))\},$$

Objective Function

$$Q(\boldsymbol{\beta}) = -\frac{1}{n}\ell(\boldsymbol{\beta}) + \sum_{j=1}^{J} P_{MCP}(\|\boldsymbol{\beta}_j\|\,;\sqrt{q_j}\lambda_1,\gamma) + \sum_{j=1}^{J} P_2(\boldsymbol{\beta}_j,\lambda_2)$$

- The first λ_1-penalty is the usual group minimax concave penalty (group MCP) to select groups of features.
- The second λ_2-penalty encourage smoothness of coefficients.

Group spline-penalty (gsp-penalty)

$$\sum_{m=2}^{q_j-1}(\Delta_m^{(2)}\boldsymbol{\beta}_j)^2$$

- $\Delta_m\boldsymbol{\beta}_j =: (\beta_{j,m+1} - \beta_{j,m})$,
  $\Delta_m^{(2)}\boldsymbol{\beta}_j =: (\Delta_m\beta_j - \Delta_{m-1}\beta_j) = \beta_{j,m+1} - 2\beta_{j,m} + \beta_{j,m-1}$.
- L_2 penalty on the discrete version of the second derivatives of coefficients
- Set L^j as a $(q_j - 2) \times q_j$ matrix with $L^j_{i,i} = L^j_{i,i+2} = 1$, $L^j_{i,i+1} = -2$ and $L^j_{i,l} = 0$ otherwise.

$$\sum_{j=1}^{J} P_2(\boldsymbol{\beta}_j, \lambda_2) = \sum_{j=1}^{J} \lambda_2 \boldsymbol{\beta}_j^\top L^{j\top} L^j \boldsymbol{\beta}_j$$

Group smooth-penalty (gsm-penalty)

$$\sum_{m=2}^{p_j} (\beta_{j,m+1} - \beta_{j,m})^2$$

- L_2 penalty on the difference among closely located variables.
- Set L^j as a $q_j \times q_j$ matrix with $L^j_{i,i} = -1$, $L^j_{i,i-1} = 1$ and $L^j_{i,l} = 0$ otherwise

$$\sum_{j=1}^{J} P_2(\boldsymbol{\beta}_j, \lambda_2) = \sum_{j=1}^{J} \lambda_2 \boldsymbol{\beta}_j^\top L^{j\top} L^j \boldsymbol{\beta}_j$$

Outline

1 Introduction

2 Methodology

- The Model Setting and Methodology
- Computational algorithm

3 Simulation and Application

- Simulation
- Application to breath VOCs

4 Discussion

Algorithm

With majorization-minimization(MM) approach as before, we have

$$Q(\beta|\widetilde{\beta}) \propto \frac{v}{2n}(\widetilde{\boldsymbol{Y}} - \boldsymbol{Z}^\top \beta_{\boldsymbol{0}} - \sum_{j=1}^{J} \boldsymbol{X}_j^\top \beta_j)^\top (\widetilde{\boldsymbol{Y}} - \boldsymbol{Z}^\top \beta_{\boldsymbol{0}} - \sum_{j=1}^{J} \boldsymbol{X}_j^\top \beta_j)$$

$$+ \sum_{j=1}^{J} P_{MCP}(\|\beta_j\|\,; \sqrt{q_j}\lambda_1, \gamma) + \sum_{j=1}^{J} \lambda_2 \beta_j^\top L^{j\top} L^j \beta_j,$$

where $v = 1/4$ and $\widetilde{\boldsymbol{Y}} = \widetilde{\boldsymbol{\eta}} + (\boldsymbol{Y} - \boldsymbol{p})/v$ is the pseudo-response vector.

Algorithm

Proposition 1:
Given current estimate $\widetilde{\beta}^{(m)}$, define an artificial dataset $(\boldsymbol{Y}^*, \boldsymbol{X}^*)$ by $\boldsymbol{Y}^* = (\widetilde{\boldsymbol{Y}}, \mathbf{0}_{(\sum_{j=1}^{J}(q_j-2))})^\top$, $\boldsymbol{X}^* = (\boldsymbol{X}, \boldsymbol{L})^\top$, where

$$\boldsymbol{L} = \begin{bmatrix} \sqrt{\lambda_1 n/v}L^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \sqrt{\lambda_1 n/v}L^2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \sqrt{\lambda_1 n/v}L^J \end{bmatrix}$$

Then

$$Q(\beta|\widetilde{\beta}) \propto \frac{v}{2n}(\boldsymbol{Y}^* - \boldsymbol{Z}^\top\beta_{\mathbf{0}} - \boldsymbol{X}^*\beta^*)^\top(\boldsymbol{Y}^* - \boldsymbol{Z}^\top\beta_{\mathbf{0}} - \boldsymbol{X}^*\beta^*)$$

$$+\sum_{j=1}^{J} P_{MCP}(\|\beta_j\|; \sqrt{q_j}\lambda_1, \gamma).$$

Outline

1 Introduction

2 Methodology
- The Model Setting and Methodology
- Computational algorithm

3 Simulation and Application
- Simulation
- Application to breath VOCs

4 Discussion

Simulation Setting

Scenario1 : The length of β is set to be 400. We divide β into 20 groups and $q_j = 20$ for $j = 1, 2, \cdots, 20$. In 3rd, 4th, 7th and 8th group, the coefficients β_js are non-zeros and chosen randomly in sequential order from the sine function $sin(t)$ where $t \in [0, 2\pi]$. $\boldsymbol{X}_j$ are generated from $N(\mathbf{0}, \mathbf{\Sigma})$. Two different covariance structures $\mathbf{\Sigma}$ are considered: $\mathbf{\Sigma} = \boldsymbol{I}$ and $\mathbf{\Sigma}_{ij} = 0.5^{|i-j|}$. The numbers of observations in the training set are set as $n = 100$ and 200, respectively.
Scenario2 : The setting of β is the same as in Scenario 1. $\boldsymbol{X}$ are generated from $N(\mathbf{0}, \mathbf{\Sigma})$ and $\mathbf{\Sigma}_{ij} = \rho^{|i-j|}$. The numbers of observations in training set are set as $n = 100$, 200 and 300, respectively. We consider the cases of weak, moderate, and strong correlations by setting $\rho = 0.2$, 0.5, and 0.8, respectively.

Table: Summary of performance measures for Scenario 1. Means with standard errors (in parentheses) are obtained from 100 Monte Carlo repetitions.

Method	Feature selection		Prediction error	
	Sensitivity	Specificity	AUC	ACC
		$\Sigma = I$, n=100		
gMCP	0.403(0.266)	0.990(0.026)	0.663(0.096)	0.654(0.056)
gsmMCP	0.917(0.118)	0.834(0.191)	0.901(0.030)	0.829(0.029)
gspMCP	0.960(0.092)	0.879(0.126)	0.910(0.036)	0.840(0.037)
		$\Sigma = I$, n=200		
gMCP	0.833(0.163)	0.998(0.012)	0.827(0.040)	0.754(0.034)
gsmMCP	0.980(0.068)	0.955(0.101)	0.928(0.019)	0.853(0.023)
gspMCP	0.993(0.043)	0.969(0.067)	0.956(0.015)	0.888(0.022)
		$\Sigma_{ij} = 0.5^{\vert i-j \vert}$, n=100		
gMCP	0.430(0.266)	0.989(0.026)	0.679(0.094)	0.665(0.063)
gsmMCP	0.902(0.150)	0.892(0.144)	0.926(0.040)	0.857(0.043)
gspMCP	0.950(0.107)	0.915(0.106)	0.925(0.040)	0.857(0.042)
		$\Sigma_{ij} = 0.5^{\vert i-j \vert}$, n=200		
gMCP	0.820(0.185)	0.999(0.009)	0.839(0.040)	0.765(0.035)
gsmMCP	0.985(0.060)	0.933(0.116)	0.959(0.017)	0.892(0.025)
gspMCP	0.983(0.064)	0.969(0.059)	0.968(0.016)	0.906(0.024)

Table: Summary of performance measures for Scenario 2.

Sample size	Method	Feature selection		Prediction error	
		Sensitivity	Specificity	AUC	ACC
		$\rho = 0.2$			
100	gMCP	0.427(0.264)	0.988(0.030)	0.675(0.093)	0.663(0.056)
	gsmMCP	0.958(0.101)	0.813(0.208)	0.924(0.031)	0.854(0.033)
	gspMCP	0.927(0.119)	0.942(0.079)	0.923(0.033)	0.854(0.035)
200	gMCP	0.840(0.157)	0.999(0.009)	0.837(0.040)	0.764(0.034)
	gsmMCP	0.988(0.055)	0.949(0.096)	0.952(0.017)	0.882(0.023)
	gspMCP	0.978(0.072)	0.989(0.030)	0.962(0.016)	0.897(0.023)
		$\rho = 0.5$			
100	gMCP	0.475(0.247)	0.989(0.027)	0.702(0.090)	0.684(0.056)
	gsmMCP	0.973(0.086)	0.839(0.170)	0.945(0.028)	0.879(0.031)
	gspMCP	0.968(0.085)	0.938(0.076)	0.942(0.032)	0.877(0.034)
200	gMCP	0.807(0.166)	0.999(0.006)	0.839(0.044)	0.765(0.039)
	gsmMCP	0.990(0.049)	0.988(0.032)	0.974(0.012)	0.915(0.020)
	gspMCP	0.985(0.060)	0.996(0.018)	0.972(0.013)	0.911(0.020)
		$\rho = 0.8$			
100	gMCP	0.465(0.204)	0.999(0.009)	0.752(0.066)	0.713(0.042)
	gsmMCP	0.940(0.118)	0.929(0.086)	0.947(0.031)	0.881(0.034)
	gspMCP	0.960(0.092)	0.961(0.057)	0.957(0.028)	0.896(0.034)
200	gMCP	0.667(0.188)	1.000(0.000)	0.845(0.047)	0.771(0.041)
	gsmMCP	0.973(0.079)	0.996(0.015)	0.984(0.015)	0.937(0.026)
	gspMCP	0.975(0.075)	0.998(0.011)	0.977(0.013)	0.920(0.022)

Simulation Setting

Scenario5 : The number of predictors is set to be $400 + n$, where n is the sample size of the training set. We divide the variable into 20 groups. And q_j for $j = 1, 2, \cdots, 20$ are not the same, where $q_1 = q_2 = 10$, $q_3 = q_4 = 15$, $q_5 = 30$, $q_6 = 40 + n$ and $q_j = 20$ for $j = 7, 8, \cdots, 20$. The key difference between this scenario and Scenario 4 is that, in this scenario, the number of variables in the sixth group is larger than the sample size n. In 1st, 2nd and 6th group, the coefficients β_js are non-zeros and chosen randomly in sequential order from the sine function $sin(t)$ where $t \in [0, 2\pi]$.

Table: Summary of performance measures for Scenario 5.

	Feature selection		Group selection		Prediction error	
Method	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**	**AUC**	**ACC**
$\Sigma = I, p = 140, n = 100$						
gMCP	0.015(0.033)	0.986(0.030)	0.080(0.178)	0.985(0.032)	0.509(0.023)	0.587(0.063)
gsmMCP	0.978(0.034)	0.700(0.266)	0.880(0.181)	0.698(0.267)	0.935(0.022)	0.864(0.027)
gspMCP	0.980(0.031)	0.686(0.162)	0.893(0.163)	0.684(0.162)	0.903(0.034)	0.834(0.033)
$\Sigma_{ij} = 0.5^{\lvert i-j \rvert}, p = 140, n = 100$						
gMCP	0.015(0.032)	0.978(0.046)	0.080(0.172)	0.976(0.049)	0.509(0.023)	0.595(0.072)
gsmMCP	0.972(0.040)	0.734(0.237)	0.850(0.214)	0.731(0.235)	0.952(0.021)	0.887(0.028)
gspMCP	0.977(0.038)	0.739(0.149)	0.877(0.205)	0.733(0.149)	0.907(0.034)	0.840(0.035)
$\Sigma = I, p = 240, n = 200$						
gMCP	0.015(0.021)	0.971(0.058)	0.130(0.183)	0.970(0.058)	0.511(0.023)	0.548(0.029)
gsmMCP	0.995(0.013)	0.481(0.272)	0.960(0.109)	0.479(0.270)	0.959(0.013)	0.891(0.019)
gspMCP	0.995(0.013)	0.605(0.198)	0.960(0.109)	0.601(0.198)	0.927(0.021)	0.852(0.024)
$\Sigma_{ij} = 0.5^{\lvert i-j \rvert}, p = 240, n = 200$						
gMCP	0.012(0.022)	0.974(0.059)	0.100(0.192)	0.973(0.062)	0.511(0.023)	0.548(0.037)
gsmMCP	0.986(0.022)	0.668(0.271)	0.877(0.187)	0.666(0.269)	0.966(0.014)	0.901(0.023)
gspMCP	0.985(0.022)	0.802(0.100)	0.870(0.195)	0.796(0.102)	0.949(0.017)	0.878(0.024)

Outline

1 Introduction

2 Methodology
- The Model Setting and Methodology
- Computational algorithm

3 Simulation and Application
- Simulation
- Application to breath VOCs

4 Discussion

- Totally 270 subjects, 92 of which are cancer and other 178 are control.
- Randomly choose 220 samples from data as training set and the remaining samples are used as the testing set.
- Regularization parameters are selected by 5-fold cross-validation.

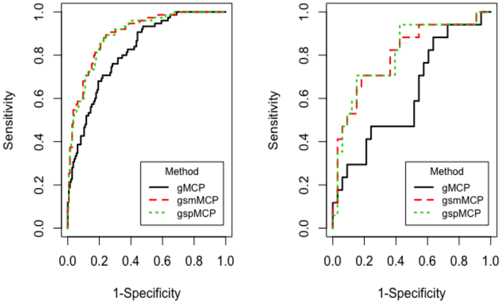


Figure: ROC curves for three methods: training set (left), testing set (right).

- Run the sample-splitting method 100 times.

Table: Performance results of 100 random partitions of the data (with standard errors in parentheses).

Method	Sensitivity	Specificity	ACC	AUC
gMCP	0.456(0.213)	0.793(0.100)	0.679(0.064)	0.684(0.099)
gsmMCP	0.547(0.150)	0.766(0.098)	0.691(0.055)	0.730(0.071)
gspMCP	0.561(0.145)	0.762(0.097)	0.694(0.058)	0.728(0.071)

Discussion

- Propose the group spline-penalty and group smooth-penalty.
- Apply new methods on the analysis of breath VOCs and focus on the prediction of lung cancer. The results show that the estimations and predictions of the newly proposed methods are more accurate comparing to group MCP for logistic regression.
- An algorithm to solve this problem. Our algorithm possess the descent property and leads to attractive convergence properties.

Dimension Reduction Methods

- The methods that we have discussed so far in this chapter have involved fitting linear regression models, via least squares or a shrunken approach, using the original predictors, $X_1, X_2, \ldots, X_p$.
- We now explore a class of approaches that *transform* the predictors and then fit a least squares model using the transformed variables. We will refer to these techniques as *dimension reduction* methods.

Dimension Reduction Methods: details

- Let $Z_1, Z_2, \ldots, Z_M$ represent $M < p$ *linear combinations* of our original p predictors. That is,

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j \tag{1}$$

  for some constants $\phi_{m1}, \ldots, \phi_{mp}$.

- We can then fit the linear regression model,

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \quad i = 1, \ldots, n, \tag{2}$$

  using ordinary least squares.

- Note that in model (2), the regression coefficients are given by $\theta_0, \theta_1, \ldots, \theta_M$. If the constants $\phi_{m1}, \ldots, \phi_{mp}$ are chosen wisely, then such dimension reduction approaches can often outperform OLS regression.

- Notice that from definition (1),

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{mj} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{mj} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij},$$

  where

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{mj}. \tag{3}$$

- Hence model (2) can be thought of as a special case of the original linear regression model.
- Dimension reduction serves to constrain the estimated β_j coefficients, since now they must take the form (3).
- Can win in the bias-variance tradeoff. Can reduce variance

Principal Components Analysis

- PCA produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
- Apart from producing derived variables for use in supervised learning problems, PCA also serves as a tool for data visualization.

研究p个变量关系，做两两散点图，需要做 P取2个图，此外两两散点图包含的信息很少，不能很好反应数据之间的关系！

需要一种低维的表示方法，但又包含了数据足够多的信息

Principal Components Analysis: details

- The *first principal component* of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \ldots + \phi_{p1}X_p$$

  that has the largest variance. By *normalized*, we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- We refer to the elements $\phi_{11}, \ldots, \phi_{p1}$ as the loadings of the first principal component; together, the loadings make up the principal component loading vector, $\phi_1 = (\phi_{11}\ \phi_{21}\ \ldots\ \phi_{p1})^T$.
- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

Computation of Principal Components

- Suppose we have a $n \times p$ data set $\mathbf{X}$. Since we are only interested in variance, we assume that each of the variables in $\mathbf{X}$ has been centered to have mean zero (that is, the column means of $\mathbf{X}$ are zero).
- We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \ldots + \phi_{p1}x_{ip} \tag{1}$$

  for $i = 1, \ldots, n$ that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
- Since each of the x_{ij} has mean zero, then so does z_{i1} (for any values of ϕ_{j1}). Hence the sample variance of the z_{i1} can be written as $\frac{1}{n}\sum_{i=1}^{n} z_{i1}^2$.

Computation: continued

- Plugging in (1) the first principal component loading vector solves the optimization problem

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p}\phi_{j1}^2 = 1.$$

- This problem can be solved via a singular-value decomposition of the matrix $\mathbf{X}$, a standard technique in linear algebra.
- We refer to Z_1 as the first principal component, with realized values $z_{11}, \ldots, z_{n1}$

Further principal components

- The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all linear combinations that are *uncorrelated* with Z_1.
- The second principal component scores $z_{12}, z_{22}, \ldots, z_{n2}$ take the form

$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \ldots + \phi_{p2}x_{ip},$$

where ϕ_2 is the second principal component loading vector, with elements $\phi_{12}, \phi_{22}, \ldots, \phi_{p2}$.

Further principal components: continued

- It turns out that constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction ϕ_2 to be orthogonal (perpendicular) to the direction ϕ_1. And so on.
- The principal component directions ϕ_1, ϕ_2, $\phi_3, \ldots$ are the ordered sequence of right singular vectors of the matrix $\mathbf{X}$, and the variances of the components are $\frac{1}{n}$ times the squares of the singular values. There are at most $\min(n-1, p)$ principal components.

Proportion Variance Explained

- To understand the strength of each component, we are interested in knowing the proportion of variance explained (PVE) by each one.
- The *total variance* present in a data set (assuming that the variables have been centered to have mean zero) is defined as

  $$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2,$$

  and the variance explained by the mth principal component is

  $$\mathrm{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2.$$

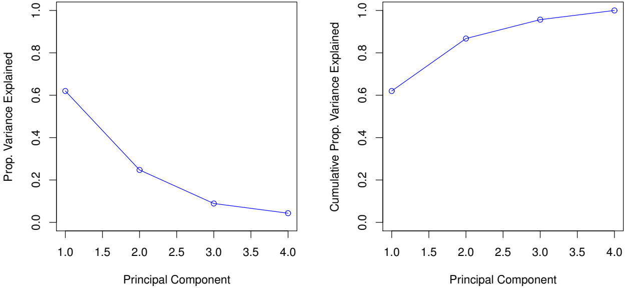
- It can be shown that $\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{m=1}^{M} \mathrm{Var}(Z_m)$, with $M = \min(n-1, p)$.

Proportion Variance Explained: continued

- Therefore, the PVE of the mth principal component is given by the positive quantity between 0 and 1

$$\frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}.$$

- The PVEs sum to one. We sometimes display the cumulative PVEs.



How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

- No simple answer to this question, as cross-validation is not available for this purpose.
    - *Why not?*

How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

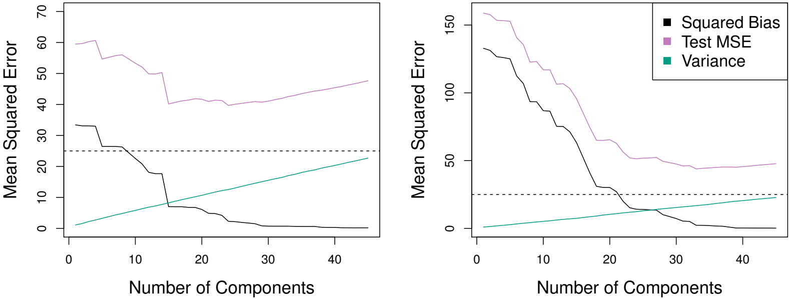
- No simple answer to this question, as cross-validation is not available for this purpose.
  - *Why not?*
  - When could we use cross-validation to select the number of components?

How many principal components should we use?

If we use principal components as a summary of our data, how many components are sufficient?

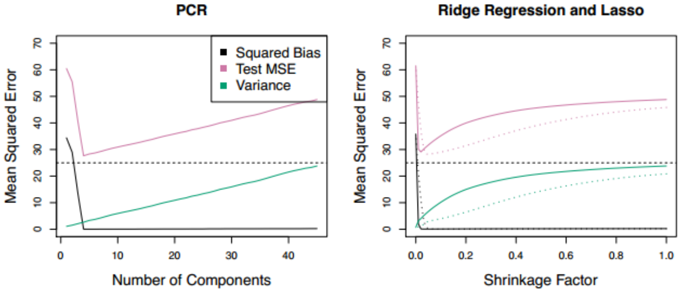
- No simple answer to this question, as cross-validation is not available for this purpose.
    - *Why not?*
    - When could we use cross-validation to select the number of components?
- the “scree plot” on the previous slide can be used as a guide: we look for an “elbow”.

Application to Principal Components Regression



PCR was applied to two simulated data sets. The black, green, and purple lines correspond to squared bias, variance, and test mean squared error, respectively. Left: *Simulated data from slide 32.* Right: *Simulated data from slide 39.*

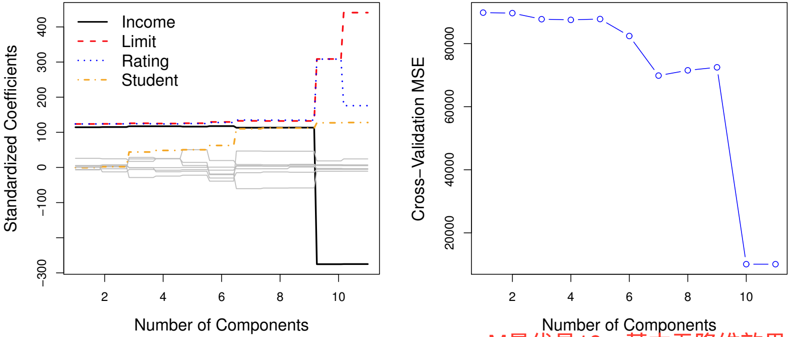
Principal Component Regression



In each panel, the irreducible error Var() is shown as a horizontal dashed line. Left: Results for PCR. Right: Results for lasso (solid) and ridge regression (dotted). The x-axis displays the shrinkage factor of the coefficient estimates, defined as the $_2$ norm of the shrunken coefficient estimates divided by the $_2$ norm of the least squares estimate.

Choosing the number of directions M

Credit data



M最优是10，基本无降维效果

Left: *PCR standardized coefficient estimates on the* `Credit` *data set for different values of M.* Right: *The* 10*-fold cross validation MSE obtained using PCR, as a function of M.*

quiz: 将Credit data set 按5:5随机拆分为训练集和测试集，针对PCR，利用CV对训练集选择最优的M，然后对测试集进行预测，重复以上过程100次，并与ridge与lasso比较 PMSE（用箱线图比较）

Partial Least Squares

- PCR identifies linear combinations, or *directions*, that best represent the predictors $X_1, \ldots, X_p$.
- These directions are identified in an *unsupervised* way, since the response Y is not used to help determine the principal component directions.
- That is, the response does not *supervise* the identification of the principal components.
- Consequently, PCR suffers from a potentially serious drawback: there is no guarantee that the directions that best explain the predictors will also be the best directions to use for predicting the response.

Partial Least Squares: continued

- Like PCR, PLS is a dimension reduction method, which first identifies a new set of features $Z_1, \ldots, Z_M$ that are linear combinations of the original features, and then fits a linear model via OLS using these M new features.
- But unlike PCR, PLS identifies these new features in a supervised way – that is, it makes use of the response Y in order to identify new features that not only approximate the old features well, but also that *are related to the response*.
- Roughly speaking, the PLS approach attempts to find directions that help explain both the response and the predictors.

Partial least square

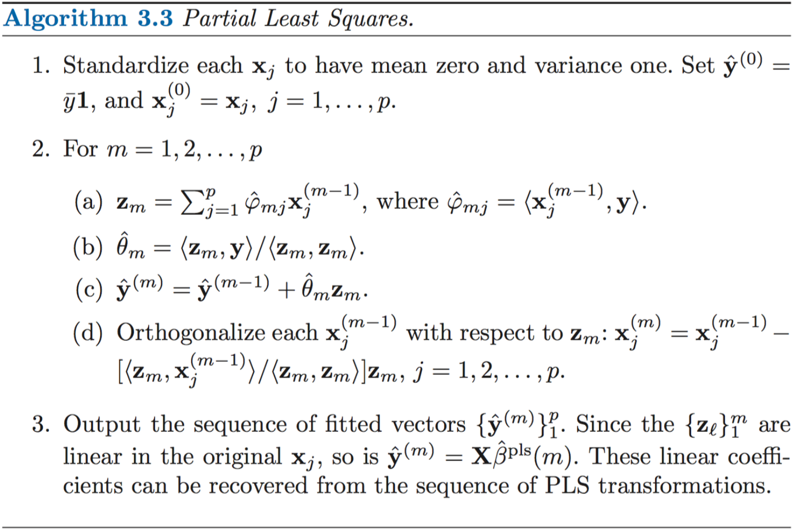
What optimization problem is partial least squares solving? Since it uses the response $\mathbf{y}$ to construct its directions, its solution path is a nonlinear function of $\mathbf{y}$. It can be shown (Exercise 3.15) that partial least squares seeks directions that have high variance and have high correlation with the response, in contrast to principal components regression which keys only on high variance (Stone and Brooks, 1990; Frank and Friedman, 1993). In particular, the m th principal component direction v_m solves:

$$\begin{gathered} \max_\alpha \operatorname{Var}(\mathbf{X}\alpha) \\ \text{subject to } \|\alpha\| = 1, \alpha^T \mathbf{S} v_\ell = 0, \ell = 1, \ldots, m-1 \end{gathered} \tag{3.63}$$

where $\mathbf{S}$ is the sample covariance matrix of the $\mathbf{x}_j$. The conditions $\alpha^T \mathbf{S} v_\ell = 0$ ensures that $\mathbf{z}_m = \mathbf{X}\alpha$ is uncorrelated with all the previous linear combinations $\mathbf{z}_\ell = \mathbf{X} v_\ell$. The m th PLS direction $\hat{\varphi}_m$ solves:

$$\begin{gathered} \max_\alpha \operatorname{Corr}^2(\mathbf{y}, \mathbf{X}\alpha) \operatorname{Var}(\mathbf{X}\alpha) \\ \text{subject to } \|\alpha\| = 1, \alpha^T \mathbf{S} \hat{\varphi}_\ell = 0, \ell = 1, \ldots, m-1 \end{gathered} \tag{3.64}$$

Partial least square



Algorithm 3.3 *Partial Least Squares.*

1. Standardize each $\mathbf{x}_j$ to have mean zero and variance one. Set $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$, and $\mathbf{x}_j^{(0)} = \mathbf{x}_j$, $j = 1, \ldots, p$.
2. For $m = 1, 2, \ldots, p$
   - (a) $\mathbf{z}_m = \sum_{j=1}^{p} \hat{\varphi}_{mj}\mathbf{x}_j^{(m-1)}$, where $\hat{\varphi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y} \rangle$.
   - (b) $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle$.
   - (c) $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m \mathbf{z}_m$.
   - (d) Orthogonalize each $\mathbf{x}_j^{(m-1)}$ with respect to $\mathbf{z}_m$: $\mathbf{x}_j^{(m)} = \mathbf{x}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{x}_j^{(m-1)} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle] \mathbf{z}_m$, $j = 1, 2, \ldots, p$.
3. Output the sequence of fitted vectors $\{\hat{\mathbf{y}}^{(m)}\}_1^p$. Since the $\{\mathbf{z}_\ell\}_1^m$ are linear in the original $\mathbf{x}_j$, so is $\hat{\mathbf{y}}^{(m)} = \mathbf{X}\hat{\beta}^{\text{pls}}(m)$. These linear coefficients can be recovered from the sequence of PLS transformations.

Details of Partial Least Squares

- After standardizing the p predictors, PLS computes the first direction Z_1 by setting each ϕ_{1j} in (1) equal to the coefficient from the simple linear regression of Y onto X_j.
- One can show that this coefficient is proportional to the correlation between Y and X_j.
- Hence, in computing $Z_1 = \sum_{j=1}^{p} \phi_{1j} X_j$, PLS places the highest weight on the variables that are most strongly related to the response.
- Subsequent directions are found by taking residuals and then repeating the above prescription.

Chun H., Keles S.Sparse partial least squares regression for simultaneous dimension reduction and variable selection.JRSSB.2010

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.
- Later, we will return to sparsity in more detail, and will describe related approaches such as the *elastic net*.