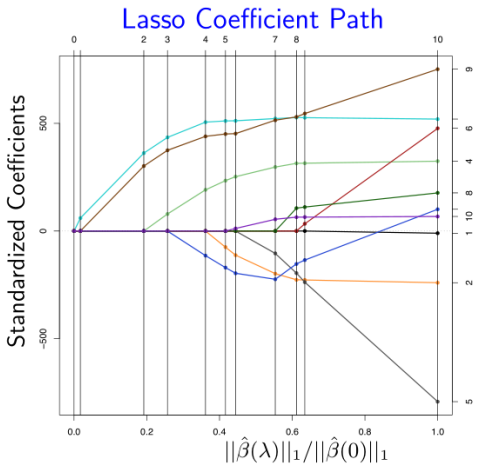
DSCI 552, Machine Learning for Data Science

University of Southern California

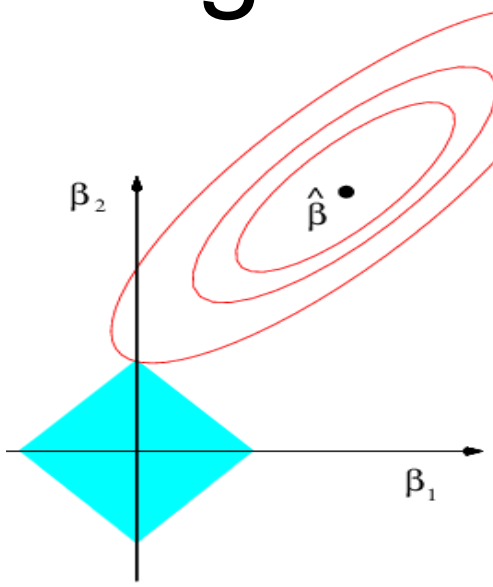
M. R. Rajati, PhD

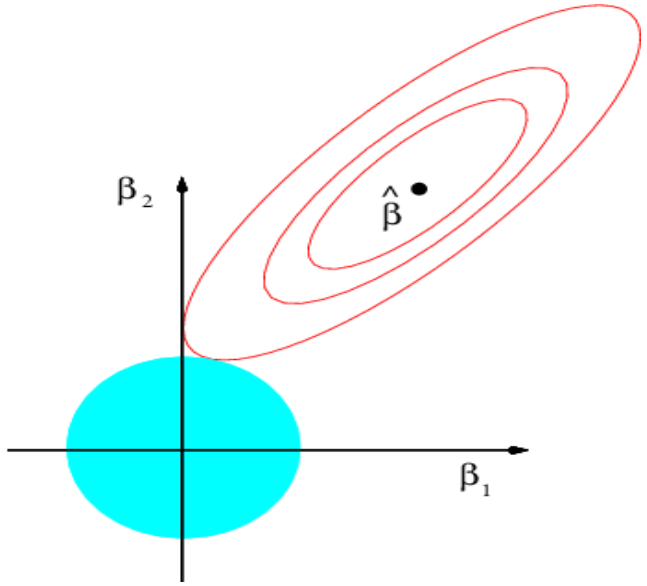
Lesson 5
Linear Model Selection and Regularization

9/24/20



Lasso: $\hat{\beta}(\lambda) = \text{argmin}_\beta \sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda||\beta||_1$





Linear Model Selection and Regularization

- Recall the linear model

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

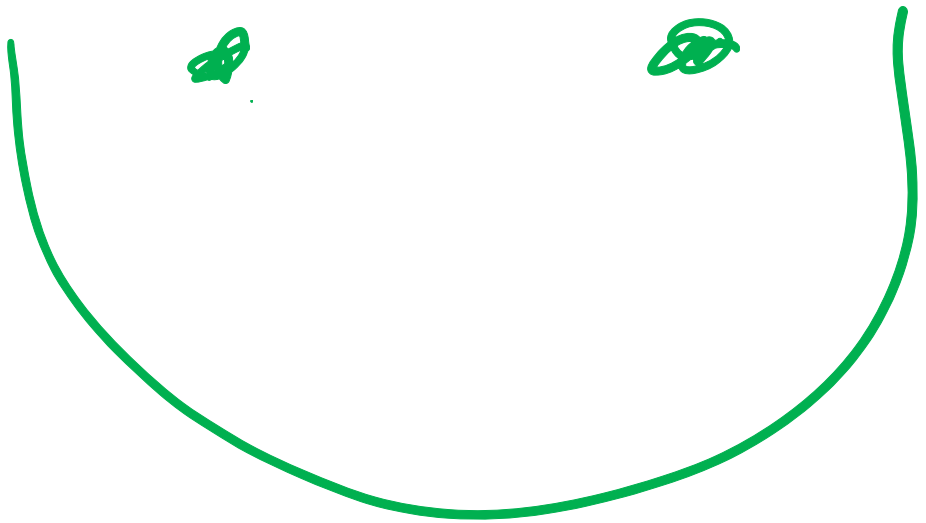
- We will consider some approaches for extending the linear model framework.

not taught but recommended

- In Chapter 7 of the text, the linear model is generalized in order to accommodate *non-linear*, but still *additive*, relationships.
- In the lectures covering Chapter 8 we consider even more general *non-linear* models.

All hail to linear models!

- Despite its simplicity, the linear model has distinct advantages in terms of its *interpretability* and often shows good (acceptable) *predictive performance*.



All hail to linear models!

- To improve the simple linear model, the ordinary least squares fitting can be replaced 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.
- Approaches for automatically performing *feature selection* will be presented.

Three classes of methods

- *Subset Selection*. Identifying a subset of the *p* predictors that are related to the response and fitting a model using least squares on the reduced set of variables.

Three classes of methods

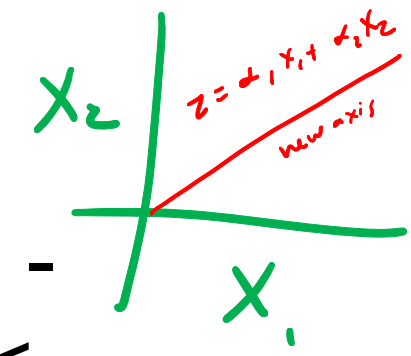
2)

- *Shrinkage*. Fitting a model involving all *p* predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates.
  - shrinkage (= *regularization*) reduces the variance and can perform variable selection.

Three classes of methods

3)

- *Dimension Reduction*. The *p* predictors are projected into a *M* - dimensional subspace, where *M* < *p*.
  - Achieved by computing *M* different *linear combinations*, or *projections*, of the variables.
  - These *M* projections are used as predictors to fit a linear regression model by least squares.



end 09/24

Begin 09/29/20

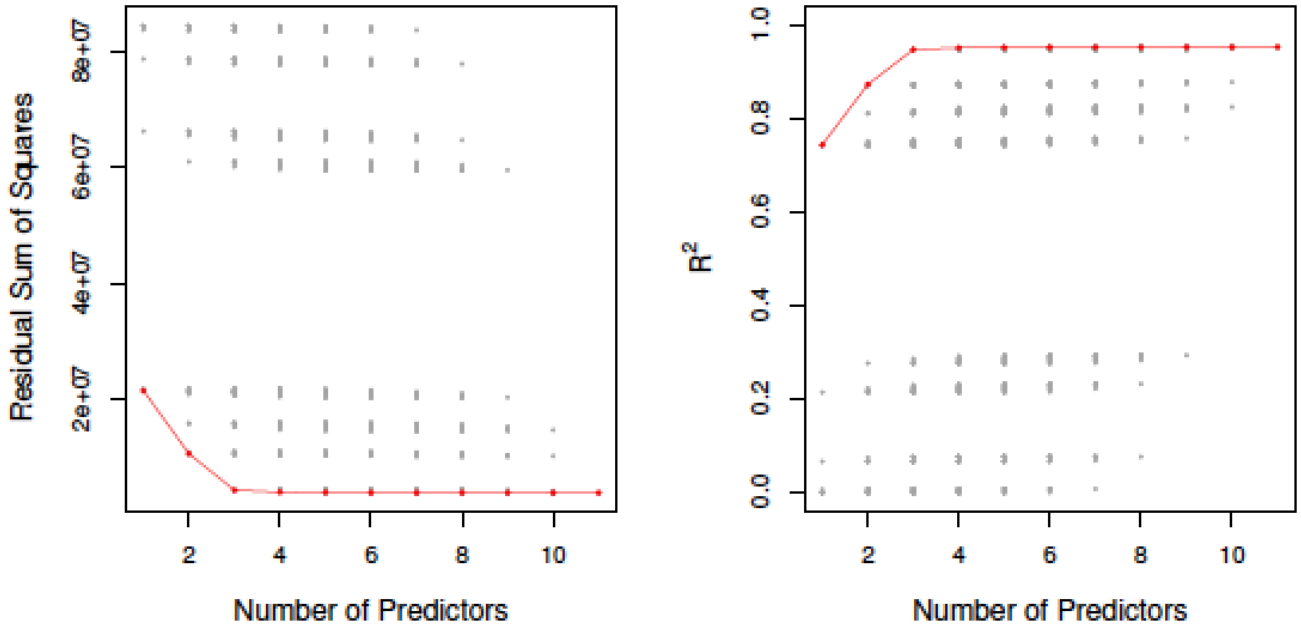
Subset Selection

Best subset and stepwise model selection procedures

Best Subset Selection

1. Let M_0 denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   - Fit all $\binom{p}{k}$ models that contain exactly k predictions
   - Pick the best among these $\binom{p}{k}$ models and call it M_k .
   - Here best is defined as having the smallest RSS, or equivalently largest R^2 .
3. Select a single best model from among $M_0, \ldots, M_p$ 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. $= -2 \log(\hat{L})$
- The *deviance*— negative two times the maximized log-likelihood— plays the role of RSS for a broader class of models.

Stepwise Selection

- 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: the 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.

Stepwise Selection

- 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.

Forward Stepwise Selection

- In particular, at each step the variable that gives the greatest *additional* improvement to the fit is added to the model.

Forward Stepwise Selection In Detail

1. Let M_0 denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p - 1$:
    1. Consider all $p - k$ models that augment the predictors in M_k with one additional predictor.
    2. Choose the *best* among these $p - k$ models, and call it M_{k+1}. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $M_0, \ldots, M_p$ 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	rating, income,
	student, limit	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 M_p denote the *full* model, which contains all p predictors.
2. For $k = p, p - 1, \ldots, 1$:
   1. Consider all k models that contain all but one of the predictors in M_k, for a total of $k - 1$ predictors.
   2. Choose the *best* among these k models, and call it M_{k-1}. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $M_0, \ldots, M_p$ 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.

More on Backward Stepwise Selection

- 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.

Choosing the Optimal Model

- 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.

Choosing the Optimal Model

- 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.

Estimating test error: two approaches

- 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.

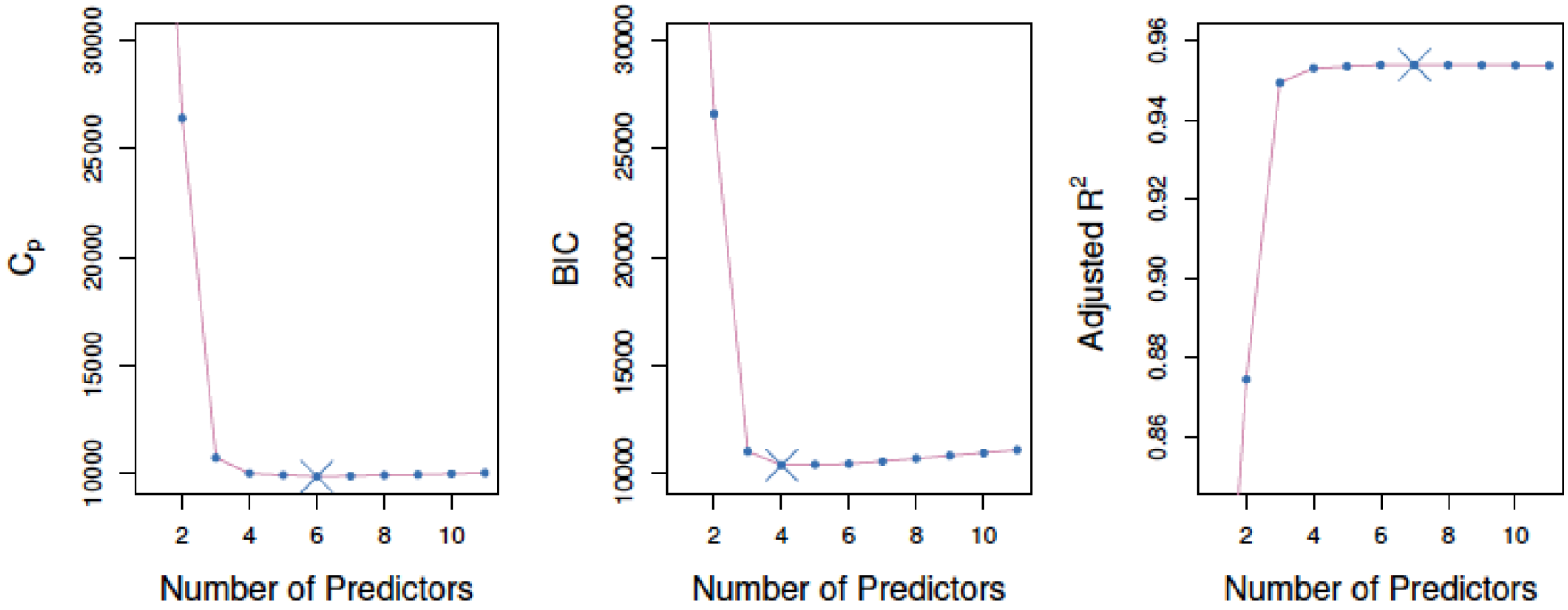
C_p, AIC, BIC, and Adjusted R^2

- 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.

C_p, AIC, BIC, and Adjusted R^2

- 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.

Credit data example



penalize for extra parameters

Now for some details

Mallow's C_p: $$C_p = \frac{1}{n}\left(\mathrm{RSS} + 2d\hat{\sigma}^2\right)$$

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.

We learned that Residual Sum of Squares, RSE, is used to estimate the variance of error:

Adjustment to RSS

$$\hat{\sigma} = \mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

Now for some details

- 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.

Now for some details

- 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.

Details on BIC

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

- 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.

Adjusted R^2

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

$d = p + 1$

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

where TSS is the total sum of squares.

Unlike Cp, 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. Remember:

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

Adjusted R^2

Maximizing the adjusted R^2 is equivalent to minimizing RSS/(n-d-1).

While RSS always decreases as the number of variables in the model increases, 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.

Validation and Cross-Validation

- Each of the procedures returns a sequence of models 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 $M_{\hat{k}}$

Validation and Cross-Validation

- We compute the validation set error or the cross-validation error for each model M_k under consideration, and then select the k for which the resulting estimated test error is smallest.

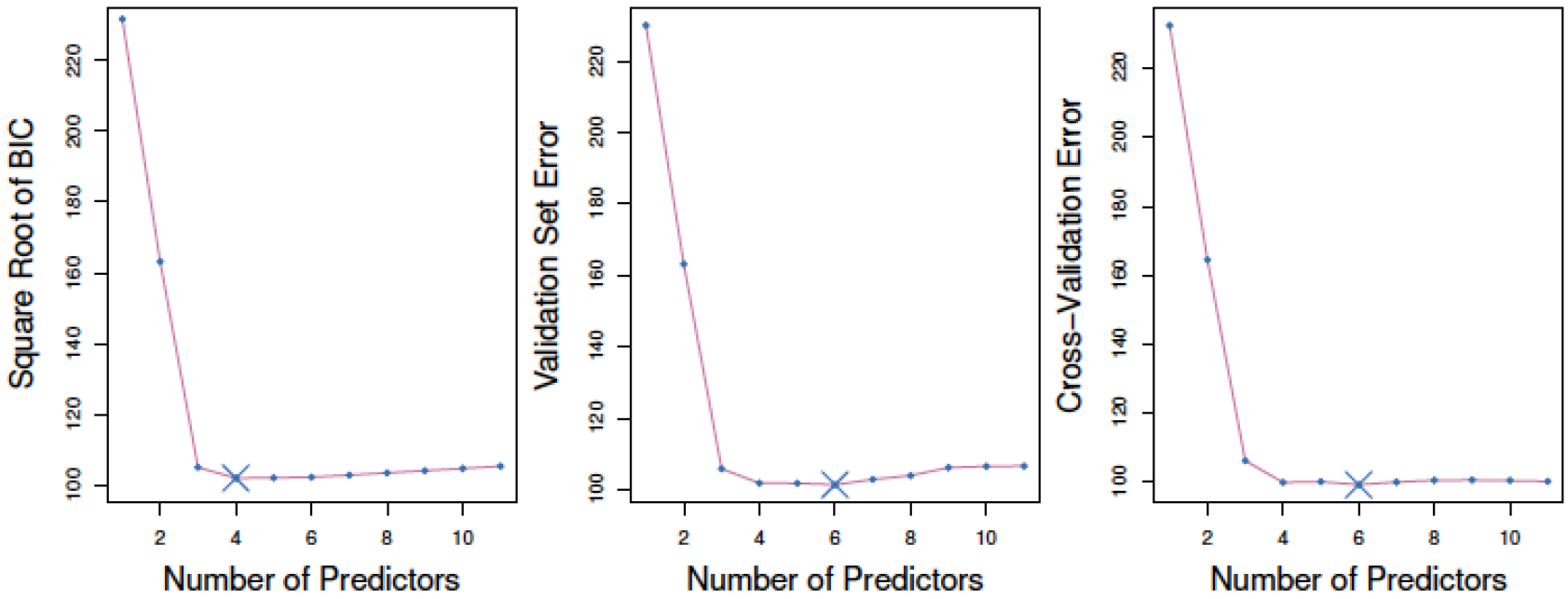
Validation and Cross-Validation

- 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.

Validation and Cross-Validation

- 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



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.

Details of Previous Figure

- 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.

Details of Previous Figure

- However, all three approaches suggest that the four-, five-, and six-variable models are roughly equivalent in terms of their test errors.

Shrinkage Methods

Ridge regression and *Lasso*

- The subset selection methods use least squares to fit a linear model that contains a subset of the predictors.

Shrinkage Methods

Ridge regression and *Lasso*

- 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.

Shrinkage Methods

Ridge regression and *Lasso*

- 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

penalty :

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

where $\lambda \geq 0$ is a tuning parameter, to be determined separately. Note that β_0 is not regularized.

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.

For large λ, you get β_i's that are different than OLS $\Rightarrow$ RSS is slightly larger

Ridge regression: continued

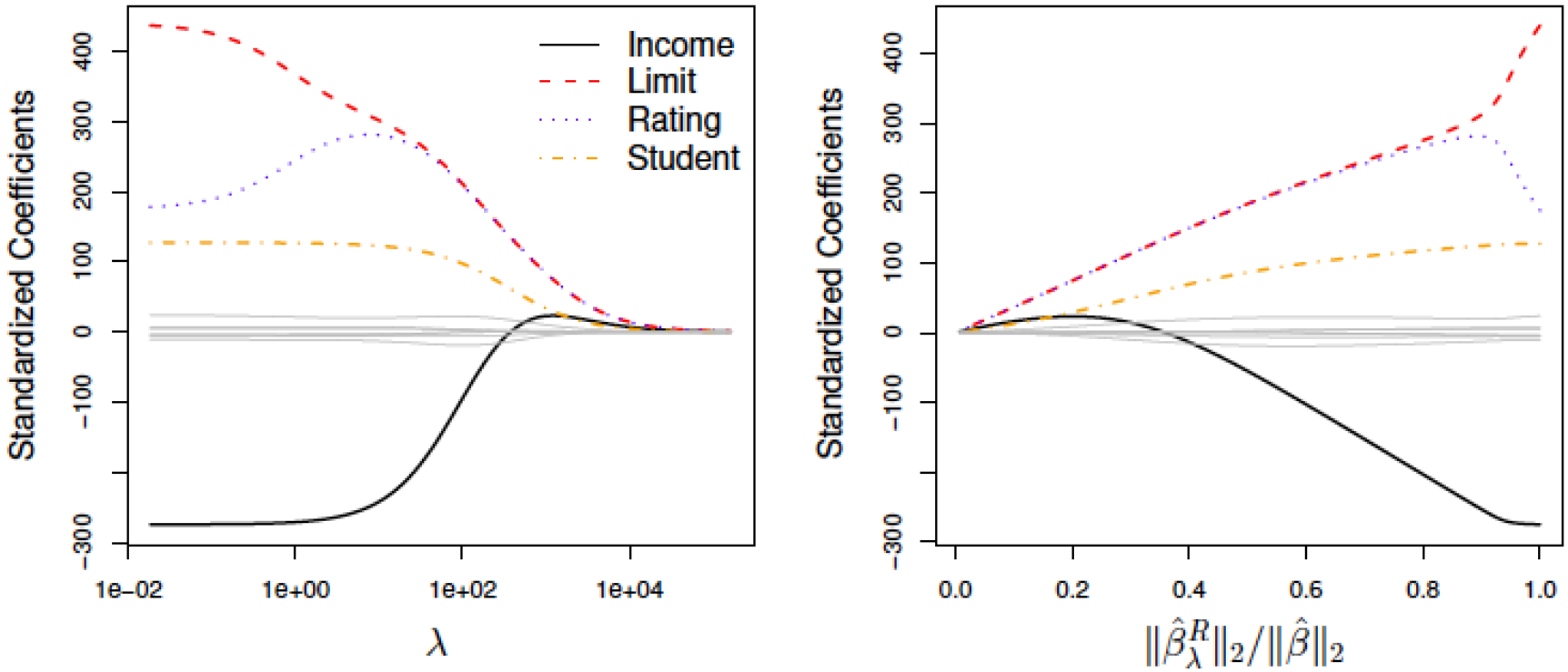
- The tuning parameter serves to control the relative impact of these two terms on the regression coefficient estimates.
- Selecting a good value for λ is critical; cross-validation is used for this.

$\log_{10} \lambda \in \{-4, -3, \ldots, 4\}$

Cross validation is for model selection

Then refit using whole train set w/ whole λ*

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 on the x-axis, we now display $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$, where β^ denotes the vector of least squares coefficient estimates.
- The notation $\|\beta\|_2$ denotes the ℓ_2 norm 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 $1/c$.

Ridge regression: scaling of predictors

- In other words, regardless of how the *j*th predictor is scaled, $X_j \hat{\beta}_j$ will remain the same.

Ridge regression: scaling of predictors

- 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.

Ridge regression: scaling of predictors

- Therefore, it is best to apply ridge regression after *standardizing the predictors*, using the formula

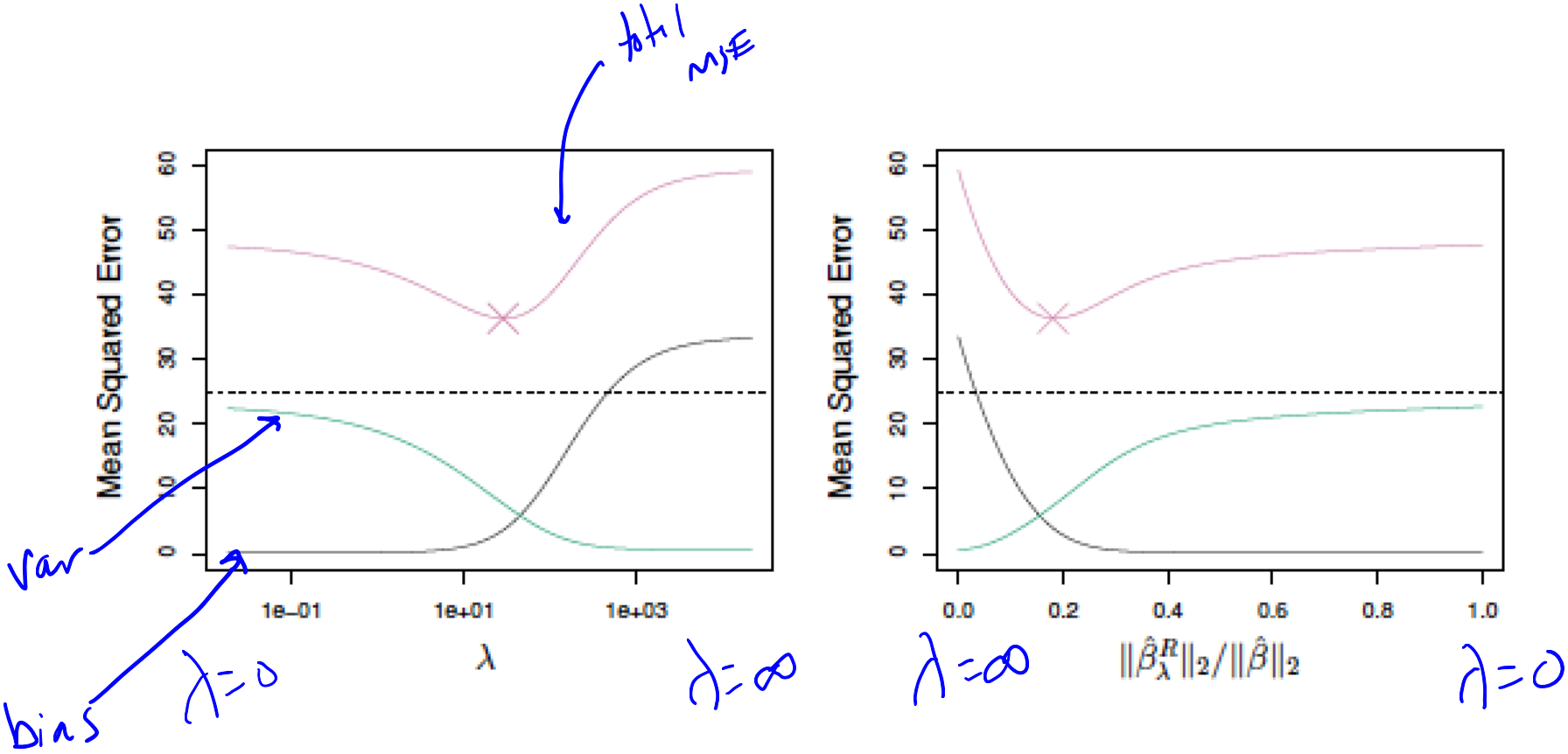
$$\tilde{x}_{ij} = \frac{x_{ij} - \overline{x}_j}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

Standard deviation

- Practical note: try raw, standardized, and normalized data.

Normalized: $\frac{x_{ij} = x_{ij} - \min_i x_{ij}}{\max x_{ij} - \min x_{ij}}$

Ridge Regression Improves Over Least Squares?



The Bias-Variance tradeoff

Ridge Regression Improves 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 the simulated data set, as a function of λ and $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.

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

- The *Lasso* is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso

The Lasso: continued

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero.

The Lasso: continued

- 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.

$$RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

The Lasso: continued

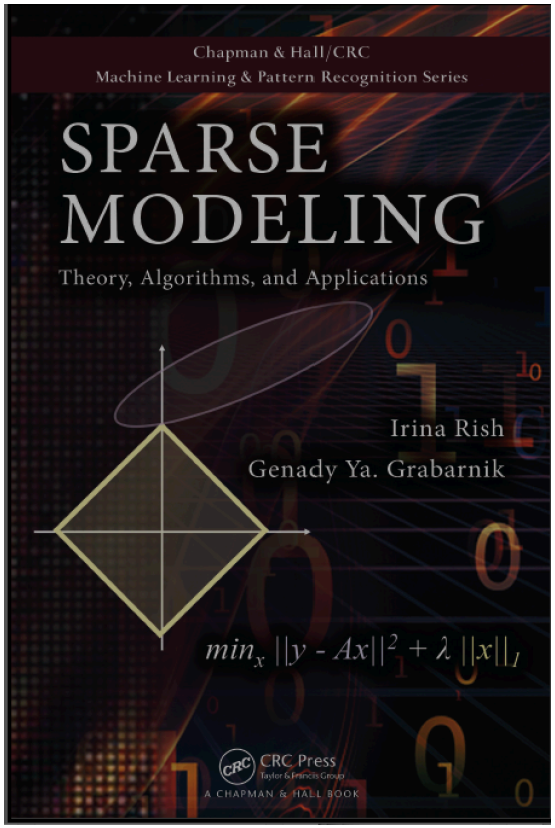
- Hence, much like best subset selection, the lasso performs *variable selection*.
- We say that the lasso yields *sparse* models — that is, models that involve only a subset of the variables.

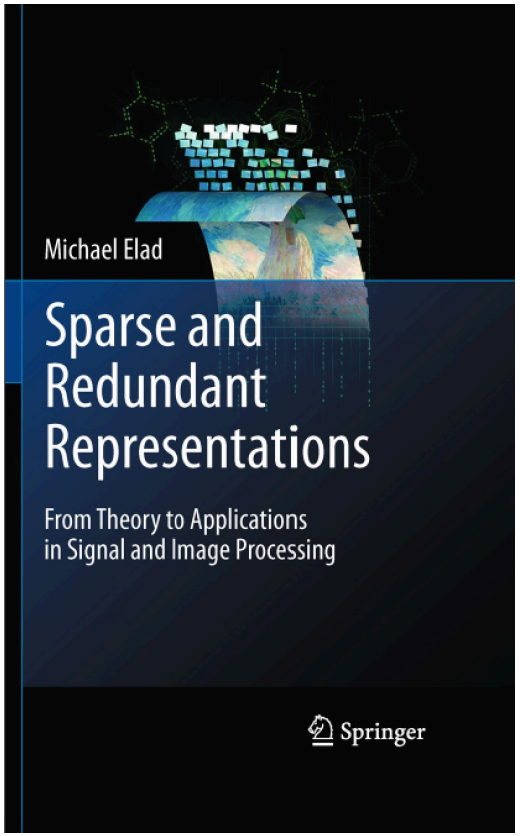
The Lasso: continued

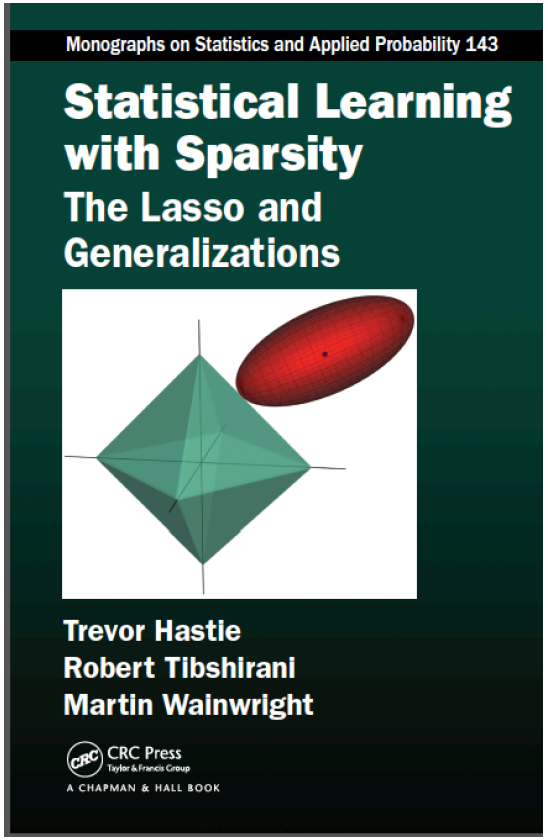
- As in ridge regression, selecting a good value of λ for the lasso is critical; cross-validation is again the method of choice.

Aside: Sparsity

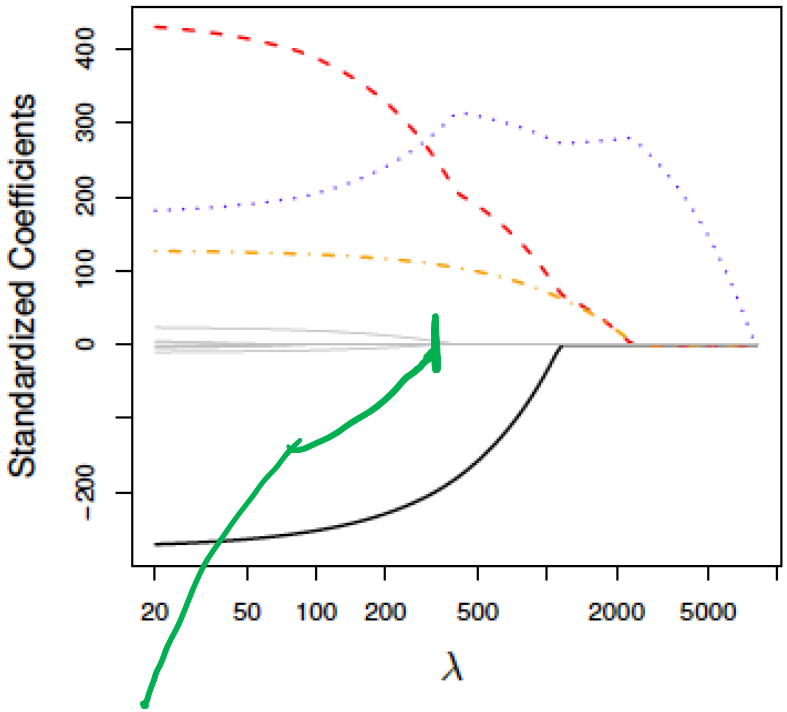
Sparsity is an essential issue in Machine Learning

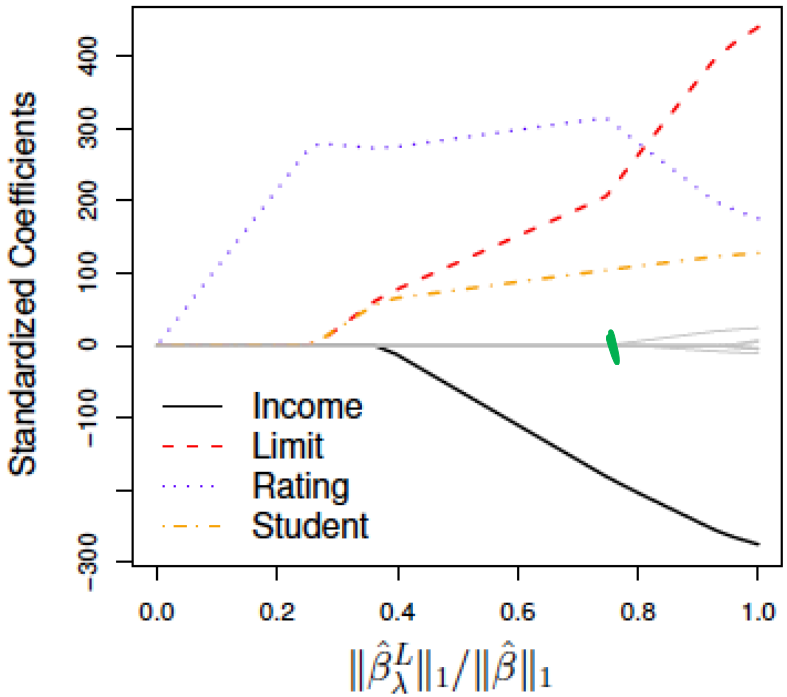






Example: Credit dataset





most go 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?

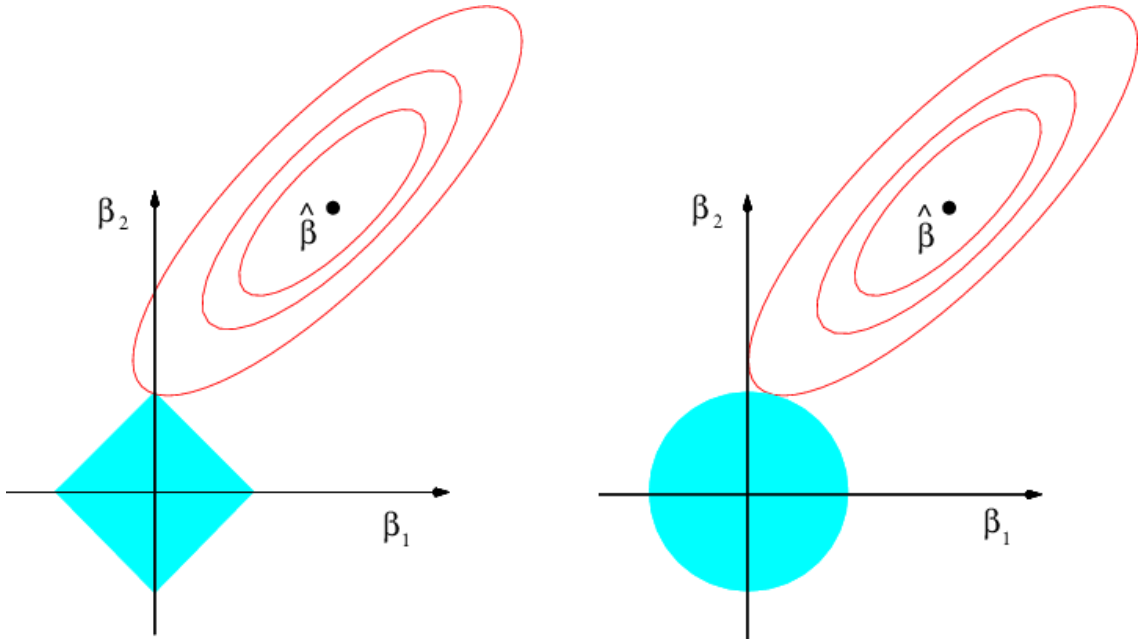
One can show that the lasso and ridge regression coefficient estimates respectively solve the problems

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

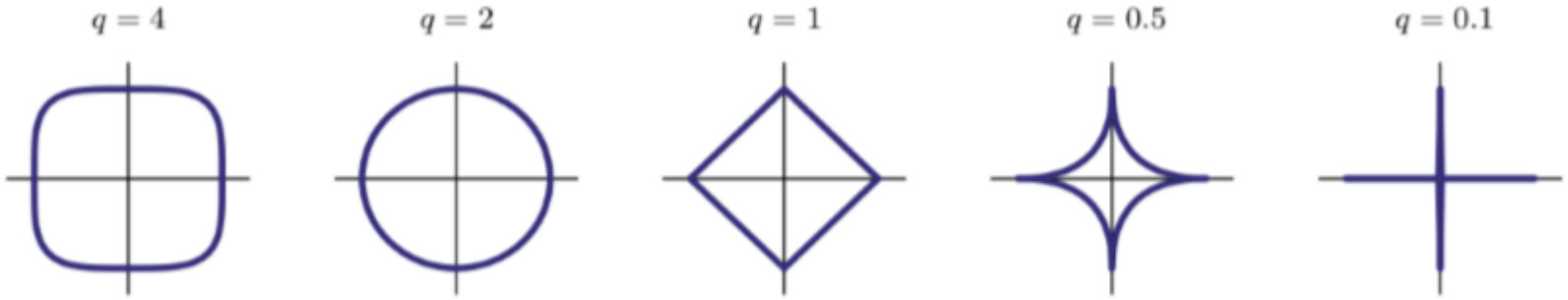
and

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

The Lasso Picture

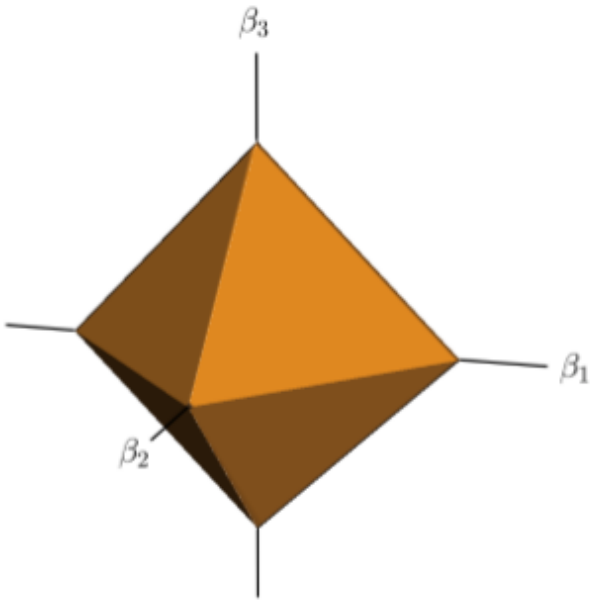


Different Constraints in 2D

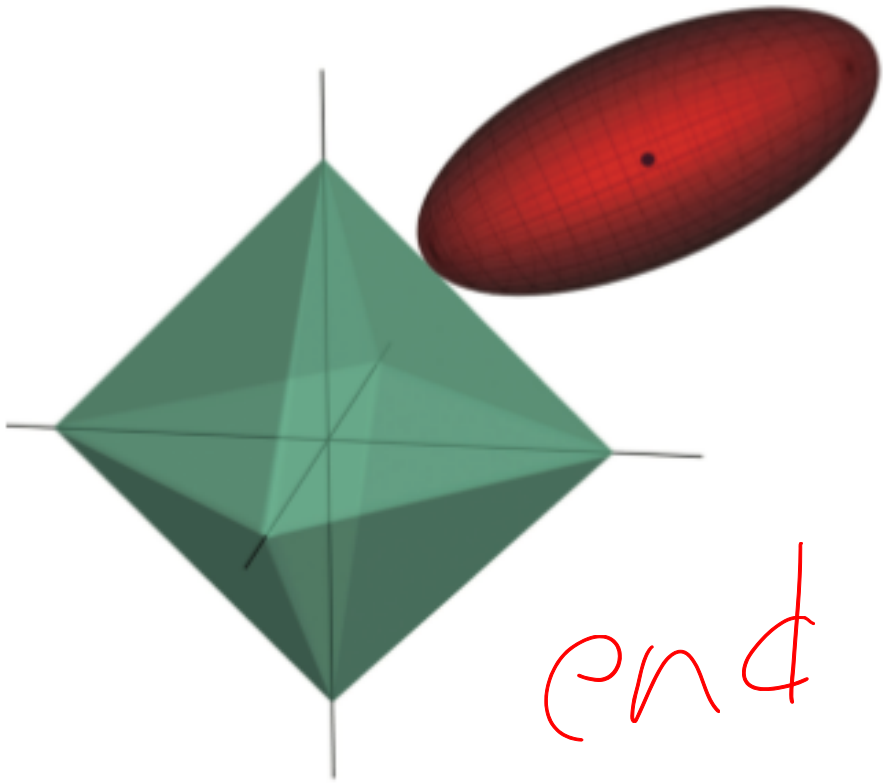


Constraint regions $\sum_{j=1}^{p} |\beta_j|^q \leq 1$ *for different values of* q. *For* $q < 1$, *the constraint region is nonconvex.*

The Lasso Constraint in 3D



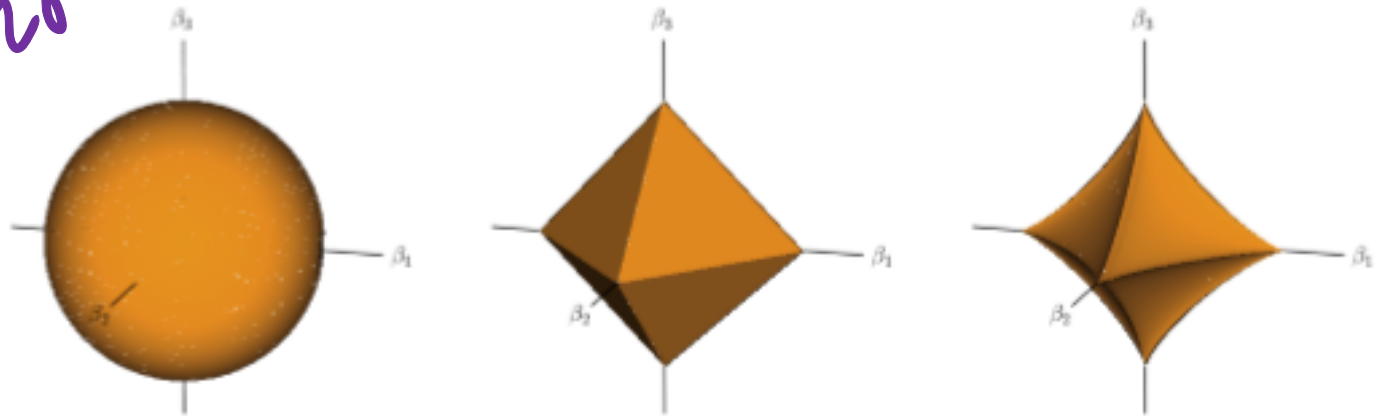
The Lasso in 3D



end 9/26

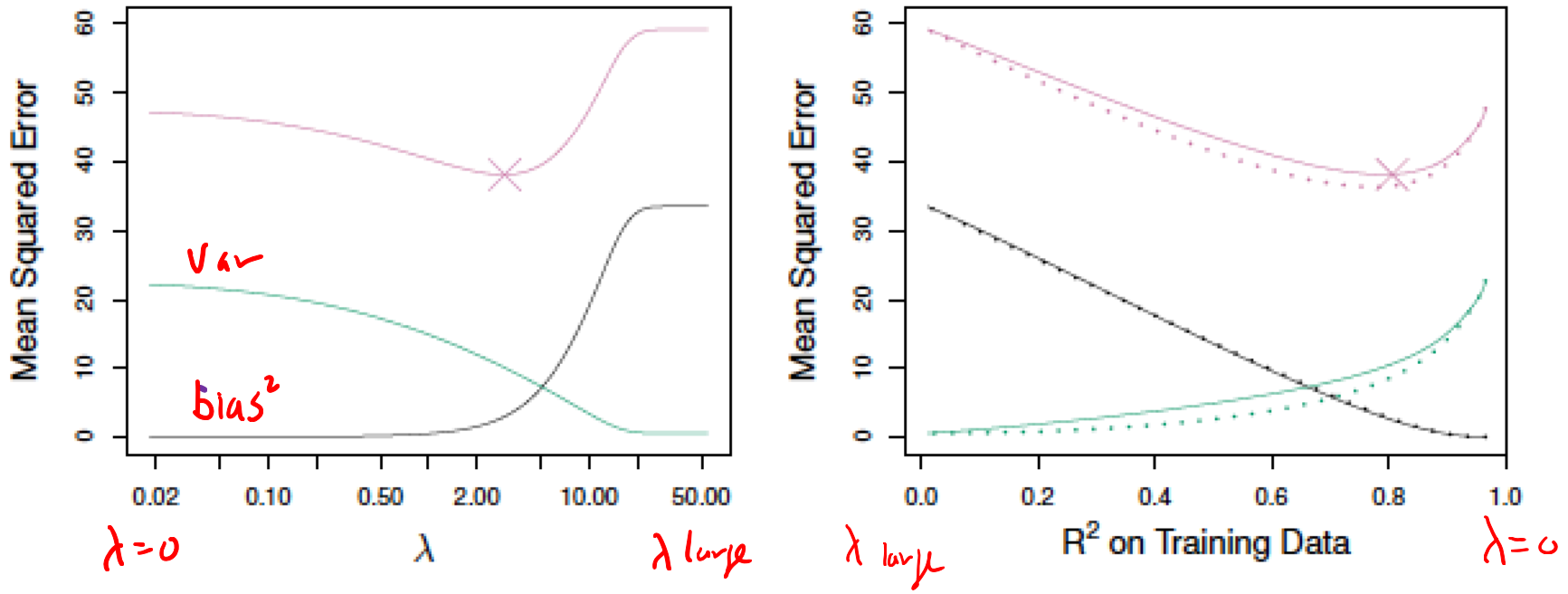
Other Constraints in 3D

Begin 10/01/20



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

Comparing the Lasso and Ridge Regression

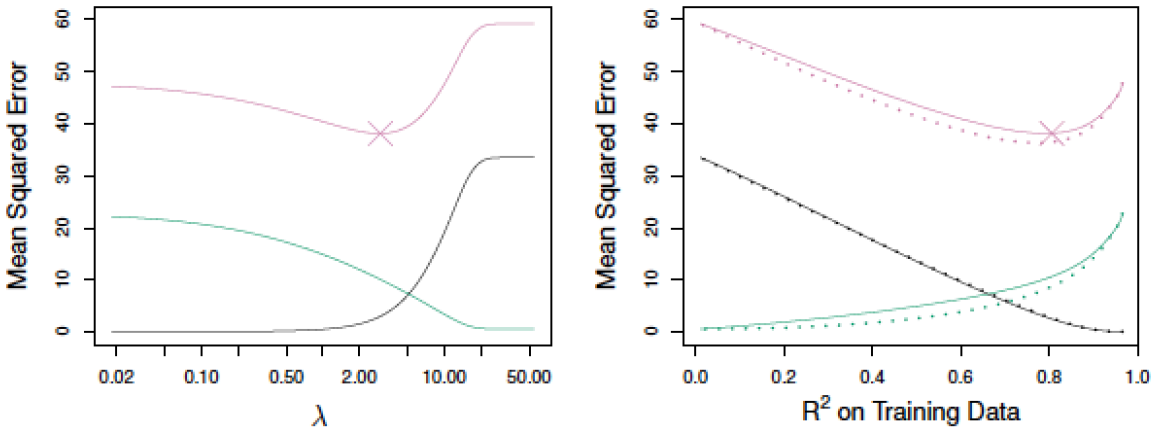


Same simulated data with n = 50 observations, p = 45 predictors, all having nonzero coefficients.

Lasso: solid

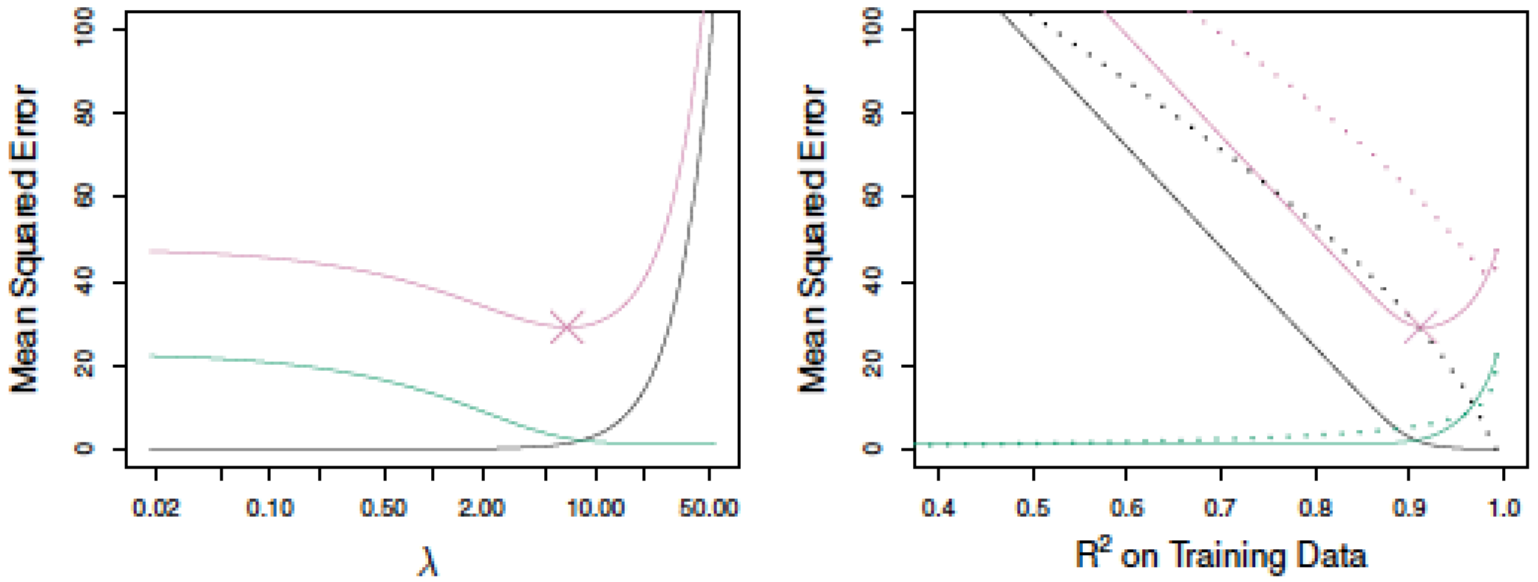
Ridge: dashed ← Better

Comparing the Lasso and Ridge Regression



Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on the simulated data set. 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

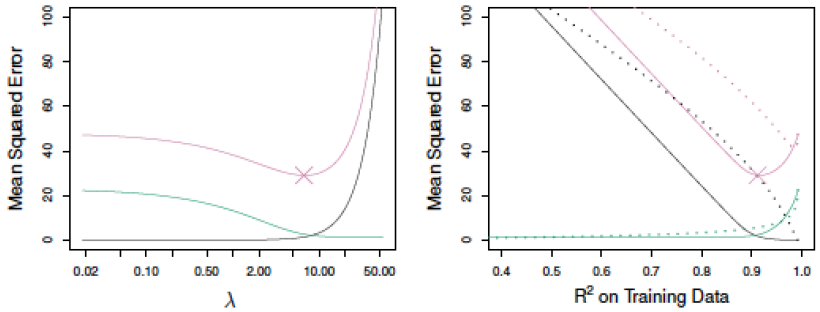


Simulated data with n = 50 observations, p = 45 predictors, only two having nonzero coefficients.

Lasso: solid ← Better

Ridge: Dashed

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 previous slides, 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.

Conclusions

- 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.

Selecting the Tuning Parameter for Ridge Regression and Lasso

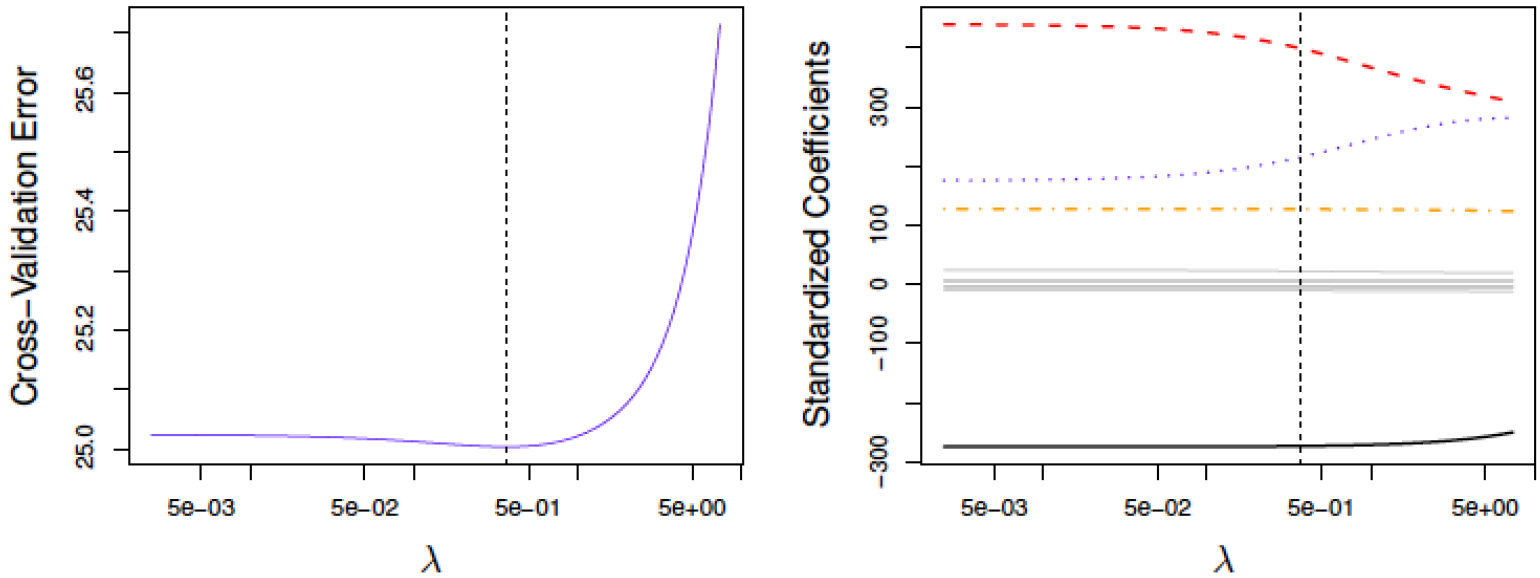
- That is, we require a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint s.

Selecting the Tuning Parameter for Ridge Regression and Lasso

- *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

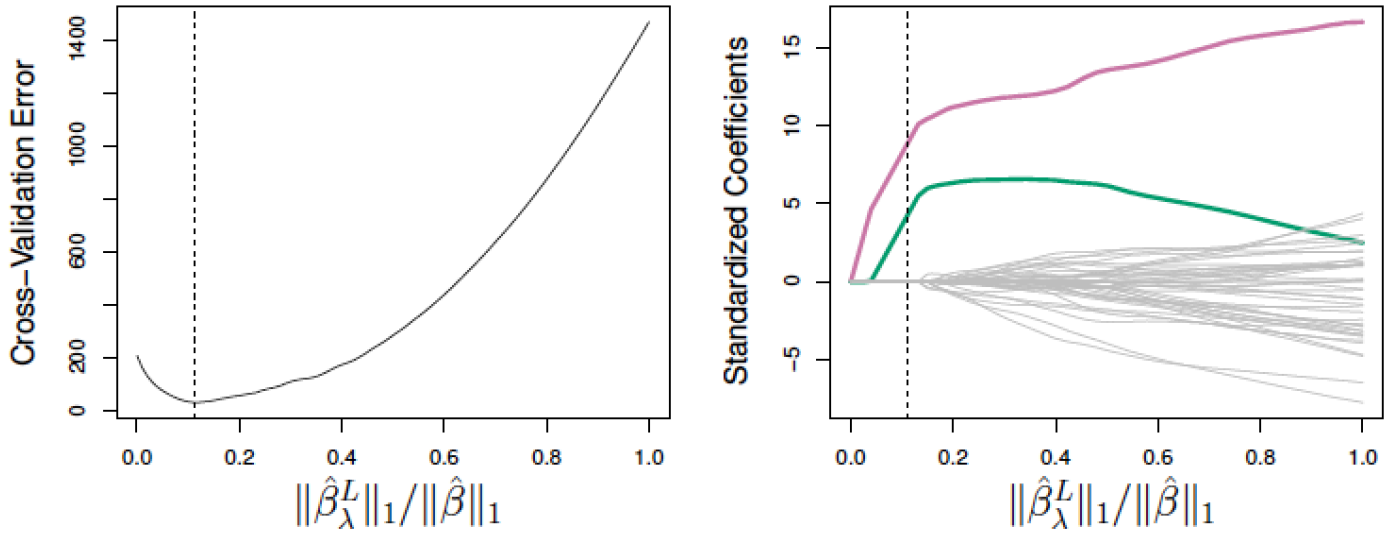
Ridge Path



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 selected by cross-validation.

Simulated data example



Left: *Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set with two predictors.*

Right: *The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

Elastic Net

- Although the lasso does a good job in eliminating *irrelevant* variables, it does not handle redundant variables (highly correlated variables) very well
- The coefficient paths tend to be erratic and can sometimes show wild behavior (e.g. starting from zero and increasing, when λ increases)

Elastic Net

- Consider a simple but extreme example, where the coefficient for a variable Xj with a particular value for λ is $\hat{\beta}_j > 0$. If we augment our data with an identical copy $X_{j'} = X_j$, then they can share this coefficient in infinitely many ways— any $\tilde{\beta}_j + \tilde{\beta}_{j'} = \hat{\beta}_j$ with both pieces positive—and the loss and L1 penalty are indifferent.
- So the coefficients for this pair are not defined.

Linear regression cannot determine $\tilde{\beta}_j + \tilde{\beta}_{j'}$

Elastic Net

- A quadratic penalty, on the other hand, will divide $\hat{\beta}_j$ exactly equally between these two twins.
- In practice, we are unlikely to have an identical pair of variables, but often we do have groups of very correlated variables.

Elastic Net

- In microarray studies, groups of genes in the same biological pathway tend to be expressed (or not) together, and hence measures of their expression tend to be strongly correlated.

Elastic Net

- The elastic net is a regularization method that combines L1 and L2 regularizations, and enforces the coefficients of correlated variables to vary together as λ changes. The Elastic Net penalty is:

$\alpha = 0$, ridge $\alpha = 1$, lasso,

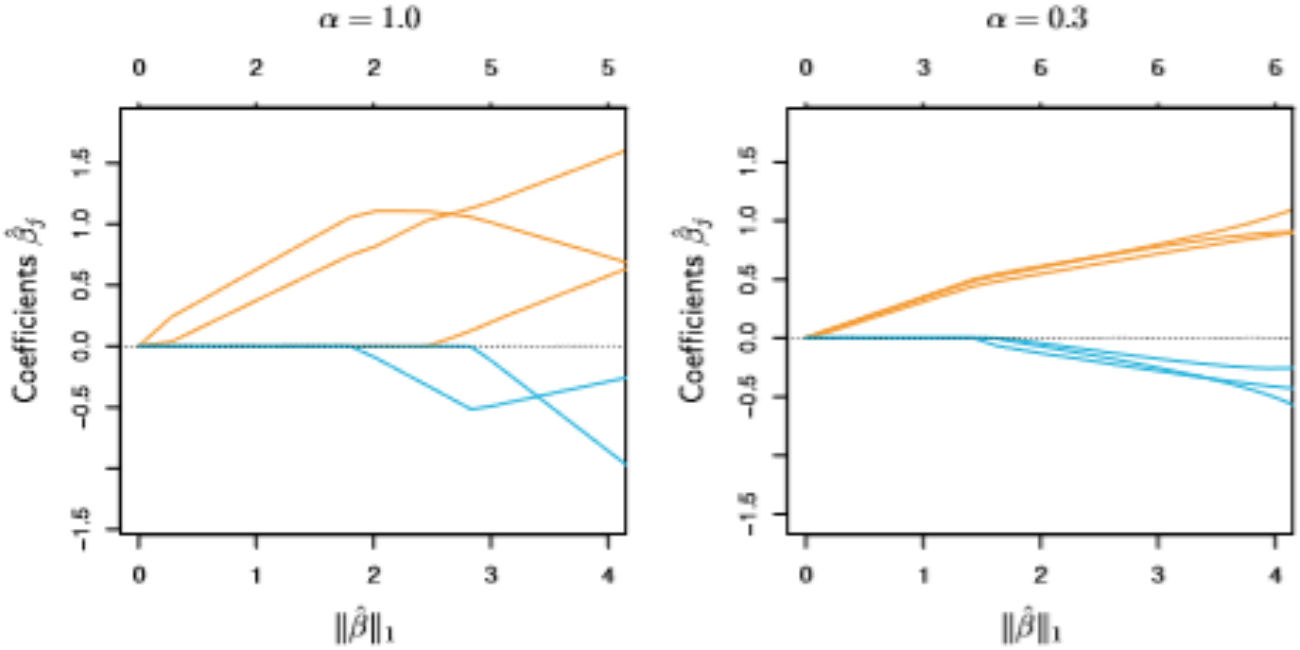
$$\lambda\left[\tfrac{1}{2}(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1\right]$$

otherwise combined

where $\alpha \in [0,1]$ is a parameter that can be varied. When $\alpha = 1$, it reduces to the L1-norm or lasso penalty, and with $\alpha = 0$, it reduces to the squared L2-norm, corresponding to the ridge penalty.

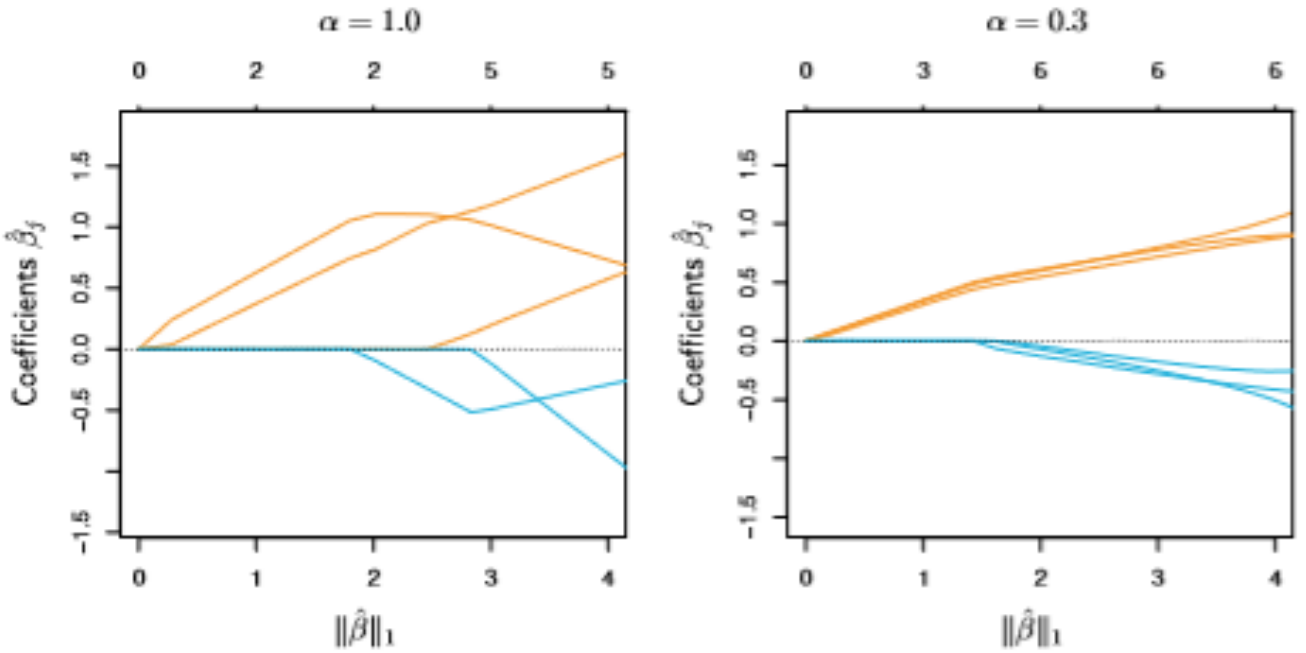
Elastic Net

- Example: Six variables, highly correlated in groups of three. The lasso estimates (α = 1), as shown in the left panel, exhibit somewhat erratic behavior as the regularization parameter *λ* is varied. In the right panel, the elastic net with (α = 0.3) includes all the variables, and the correlated groups are pulled together.



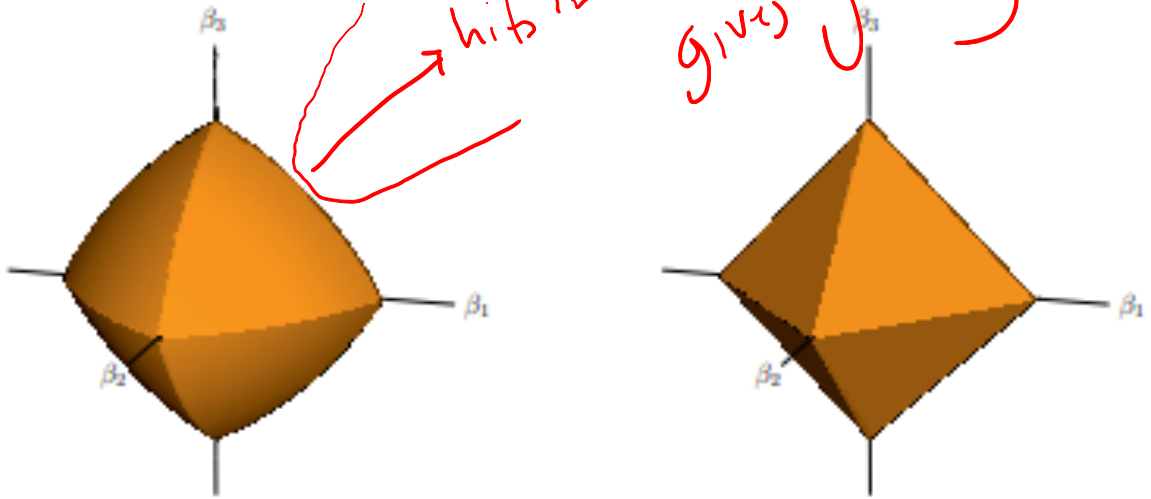
Elastic Net

- Of course, this example is idealized, and in practice the group structure will not be so cleanly evident. But by adding some component of the ridge penalty to the L1-penalty, the elastic net automatically controls for strong within-group correlations.



Elastic Net vs. Lasso Constraints

- Figure 4.2 compares the constraint region for the elastic net (left image) to that of the lasso (right image) when there are three variables. We see that the elastic-net ball shares attributes of the L2 ball and the L1 ball: the sharp corners and edges encourage selection, and the curved contours encourage sharing of coefficients.



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, $X_1, X_2, \ldots, X_p$.

Dimension Reduction Methods

- 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

- Note that they are only feature transformation methods, not feature selection methods.

Dimension Reduction Methods: details

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

$$Z_m = \sum_{j=1}^{p} \phi_{mj} X_j \quad (1)$$

  for some constants $\varphi_{m1}, \varphi_{m2}, \ldots \varphi_{mp}$.

- We can then fit the linear regression model,

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

  using ordinary least squares.

Dimension Reduction Methods: details

- Note that in the model

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

  the regression coefficients are given by $\theta_1, \theta_2, \ldots, \theta_M$.
- If the constants $\varphi_{m1}, \varphi_{m2}, \ldots. \varphi_{mp}$ are chosen wisely, then such dimension reduction approaches can often outperform OLS regression/reduce variance.

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

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

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.

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

Principal Components Regression

- Here we apply principal components analysis (PCA) (discussed in Chapter 10 of the text) to define the linear combinations of the predictors, for use in our regression.

Principal Components Regression

- The first principal component is that (normalized) linear combination of the variables with the largest variance.

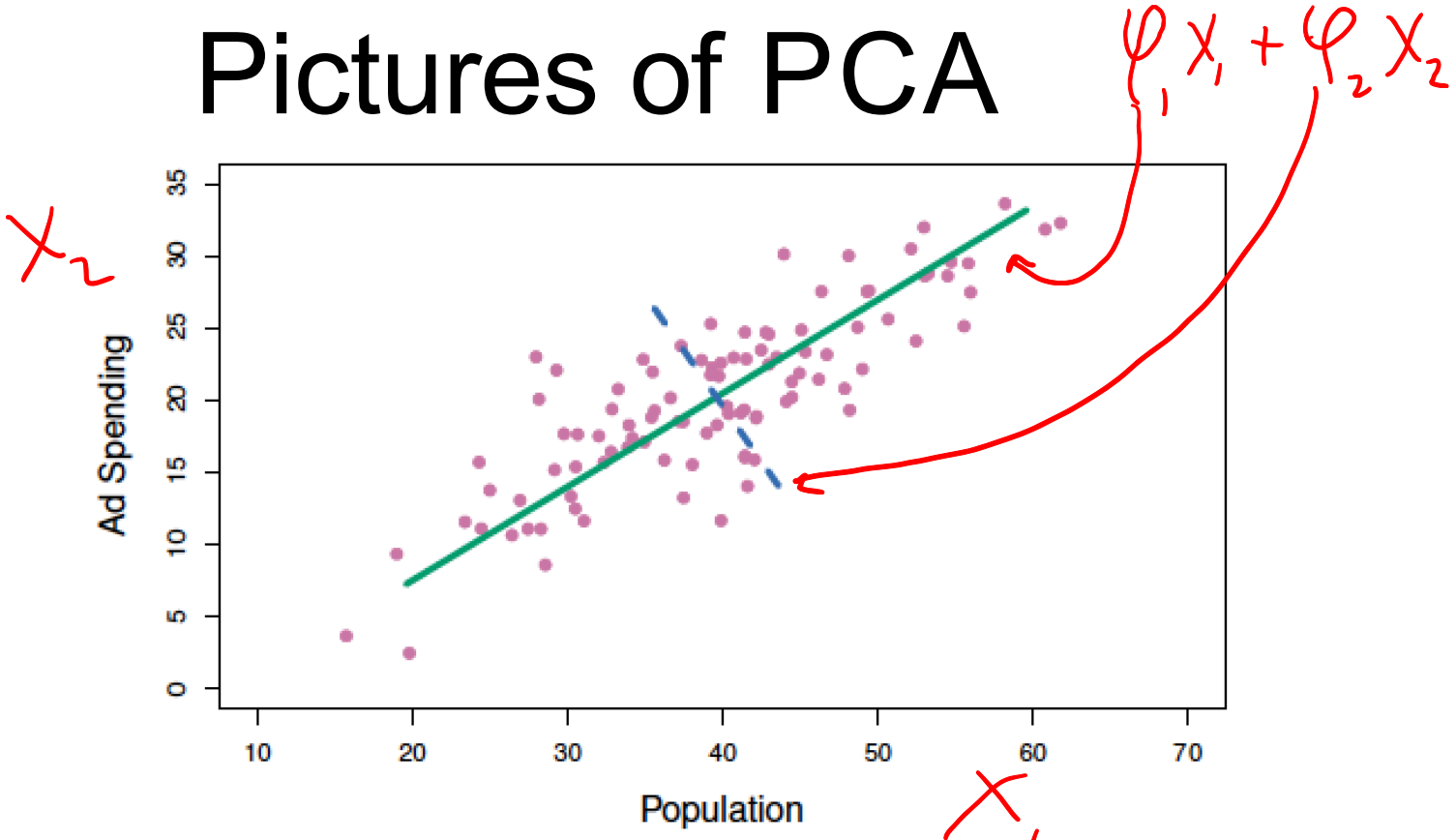
Principal Components Regression

- The second principal component has largest variance, subject to being uncorrelated with the first.
- And so on.

Principal Components Regression

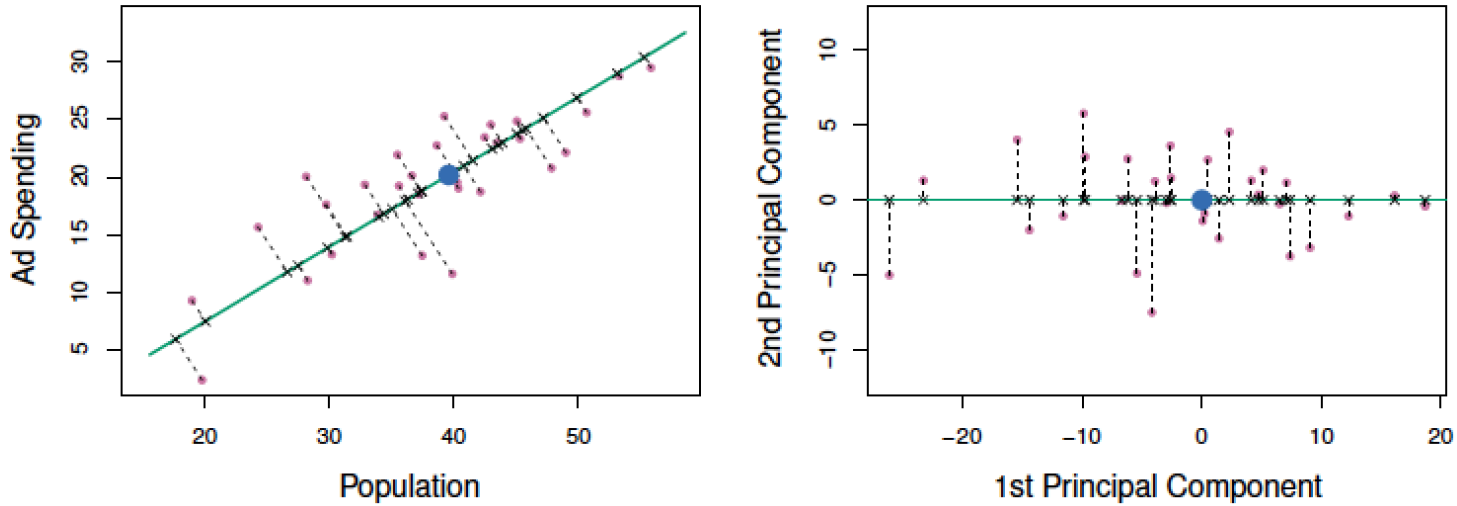
- Hence with many correlated original variables, we replace them with a small set of principal components that capture their joint variation.

Pictures of PCA



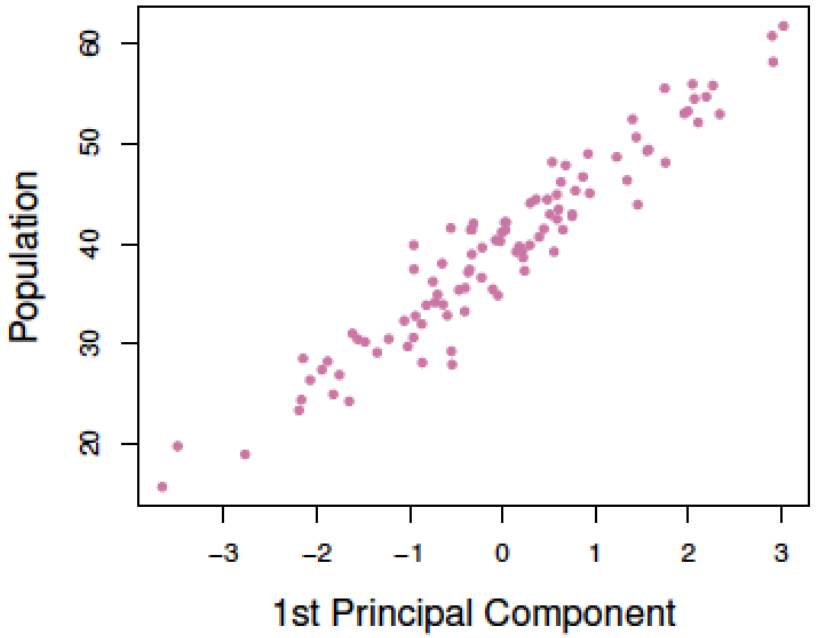
The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.

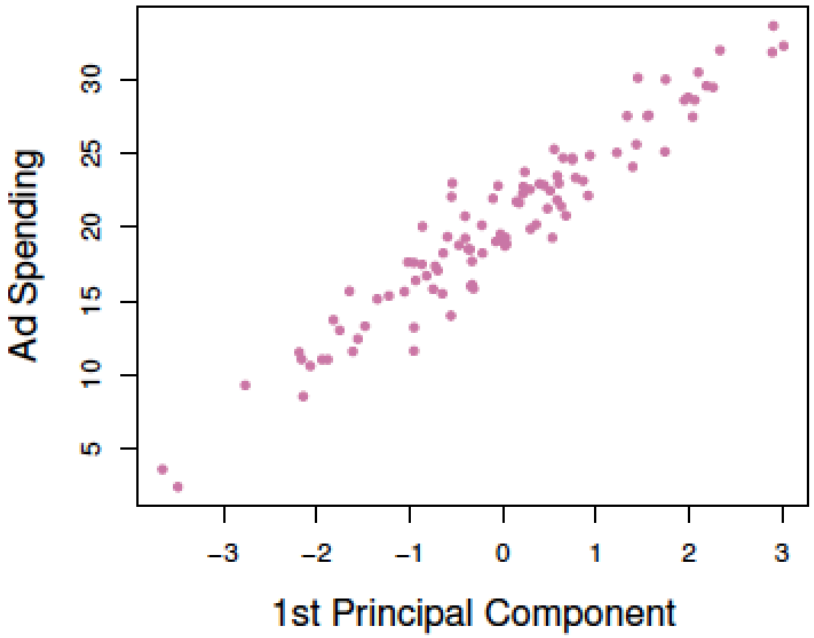
Pictures of PCA: continued



A subset of the advertising data. Left: *The first principal component, chosen to minimize the sum of the squared perpendicular distances to each point, is shown in green. These distances are represented using the black dashed line segments.* Right: *The left-hand panel has been rotated so that the first principal component lies on the x-axis.*

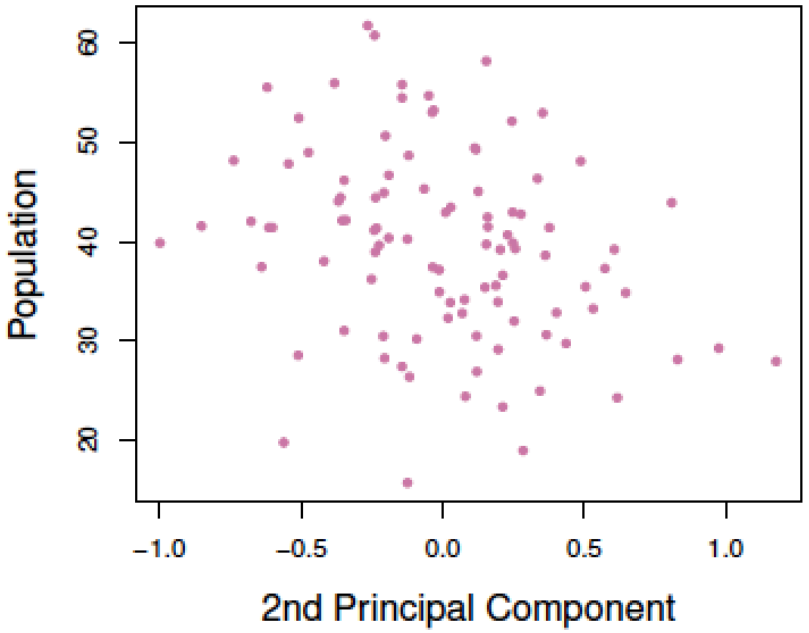
Pictures of PCA: continued

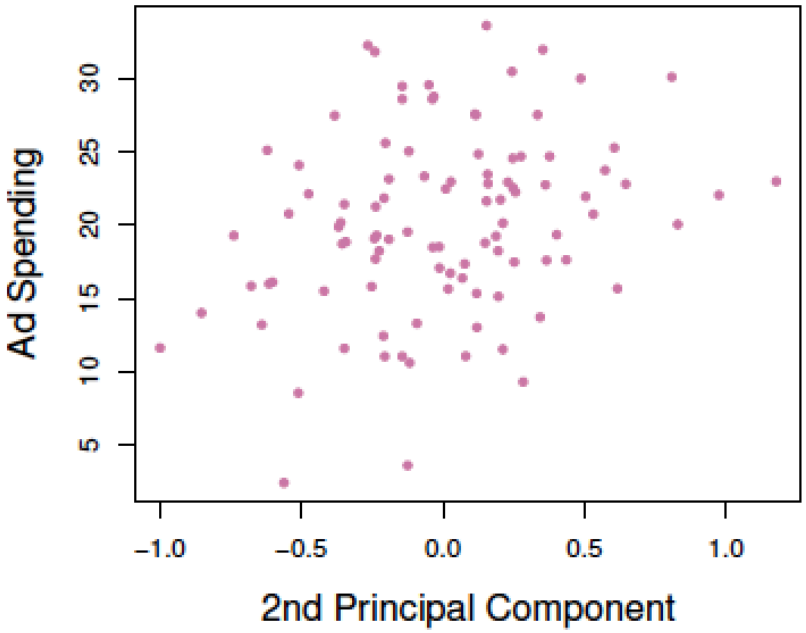




Plots of the first principal component scores z_{i1} versus pop *and* ad*. The relationships are strong.*

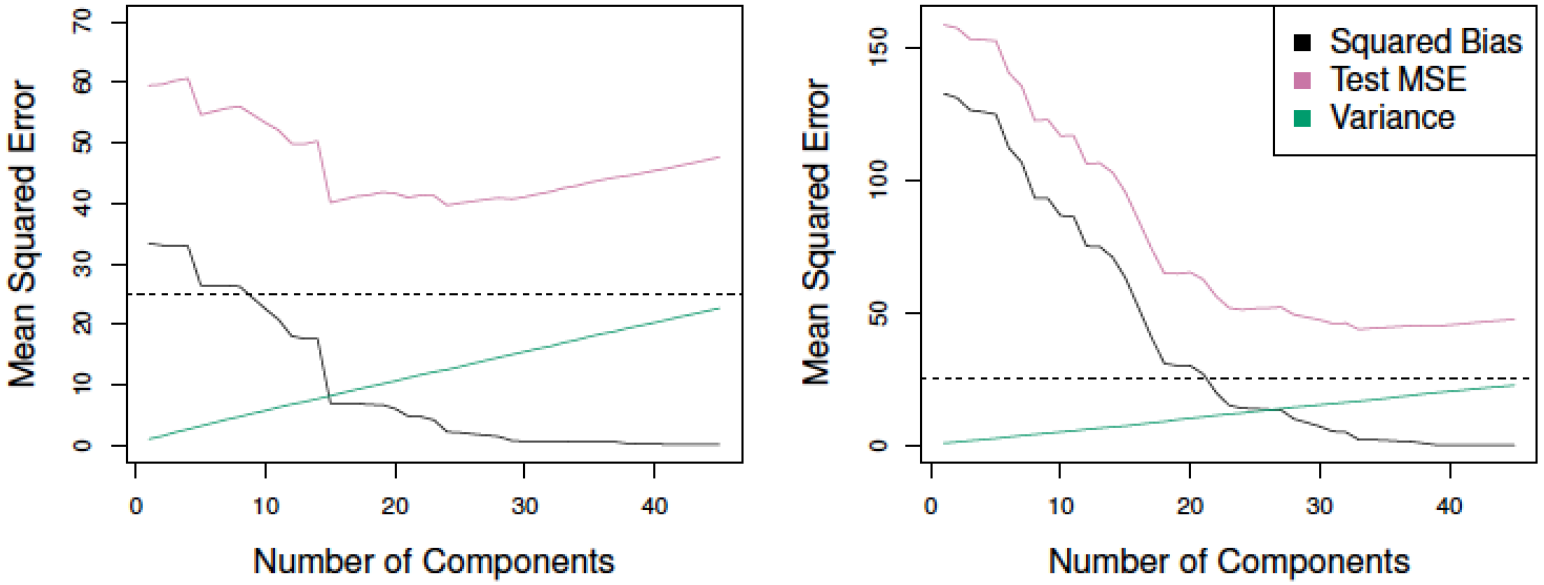
Pictures of PCA: continued





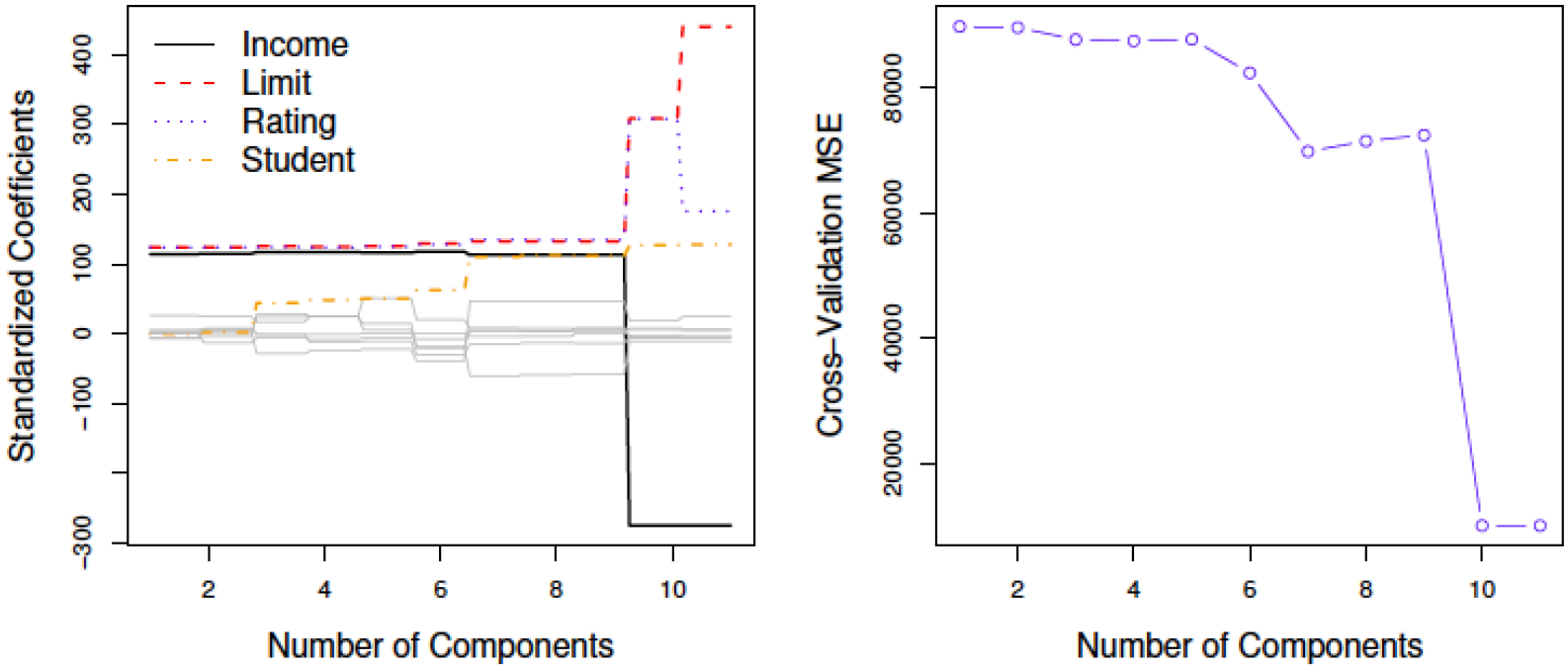
Plots of the second principal component scores z_{i2} versus pop *and* ad*. The relationships are weak.*

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 2 (Sparse)* Right: *Simulated data 1 (Non-sparse)*

Choosing the number of directions *M*



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 .*

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.

Partial Least Squares

- That is, the response does not *supervise* the identification of the principal components.

Partial Least Squares

- 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.

Partial Least Squares: continued

- 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*.

Partial Least Squares: continued

- Roughly speaking, the PLS approach attempts to find directions that help explain both the response and the predictors.

Details of Partial Least Squares

- After standardizing the p predictors, PLS computes the first direction Z_1 by setting each φ_{1j} equal to the coefficient from the simple linear regression of Y onto X_j.

Details of Partial Least Squares

- 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.

Details of Partial Least Squares

- Subsequent directions are found by taking residuals and then repeating the above prescription.

Summary

- Model selection methods are an essential tool for data analysis, especially for big datasets involving many predictors.

Summary

- Research into methods that give *sparsity*, such as the *lasso* is an especially hot area.
- Later, we will return to sparsity in SVMs.