ADEC 7430:
Big Data Econometrics

Linear Model Selection and Regularization

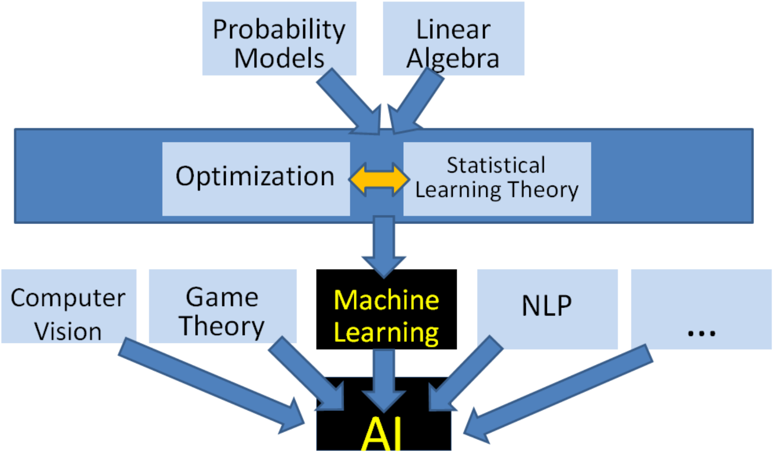
Dr. Nathan Bastian

Woods College of Advancing Studies

Boston College

Assignment

- **Reading:** Ch. 6
- **Study:** Lecture Slides, Lecture Videos
- **Activity:** Quiz 4, R Lab 4, Discussion #4



References

- *An Introduction to Statistical Learning, with Applications in R* (2013), by G. James, D. Witten, T. Hastie, and R. Tibshirani.

- *The Elements of Statistical Learning* (2009), by T. Hastie, R. Tibshirani, and J. Friedman.

- *Learning from Data: A Short Course* (2012), by Y. Abu-Mostafa, M. Magdon-Ismail, and H. Lin.

- *Machine Learning: A Probabilistic Perspective* (2012), by K. Murphy.

Lesson Goals

- Demonstrate best subset selection and stepwise selection methods for reducing the number of predictor variables in regression.
- Identify how to indirectly estimate test error by adjusting training error to account for bias due to overfitting (C_p, AIC, BIC, adjusted R^2).
- Illustrate how to directly estimate test error using validation set approach and cross-validation approach.
- Demonstrate ridge regression and the lasso as shrinkage (regularization) methods.
- Demonstrate principal components regression and partial least squares as dimension reduction methods.
- Identify considerations for high-dimensional settings.

Improving the Linear Model

- We may want to improve the simple linear model by replacing OLS estimation with some alternative fitting procedure.

- Why use an alternative fitting procedure?
  - Prediction Accuracy
  - Model Interpretability

Prediction Accuracy

- The OLS estimates have relatively low bias and low variability especially when the relationship between the response and predictors is linear and $n >> p$.

- If n is not much larger than p, then the OLS fit can have high variance and may result in over fitting and poor estimates on unseen observations.

- If $p > n$, then the variability of the OLS fit increases dramatically, and the variance of these estimates in infinite.

Model Interpretability

- When we have a large number of predictors in the model, there will generally be many that have little or no effect on the response.

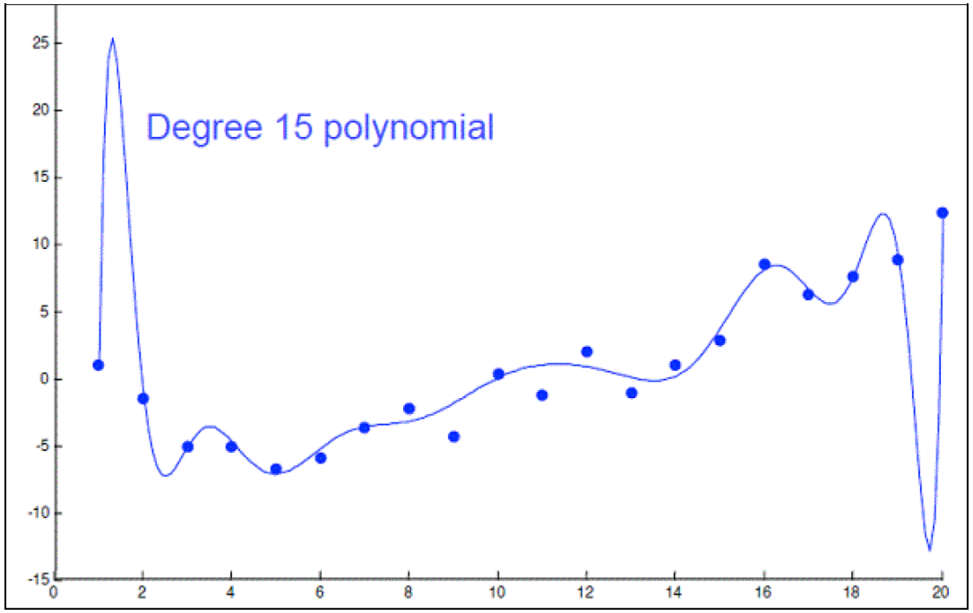
- Including such irrelevant variable leads to unnecessary complexity.

- Leaving these variables in the model makes it harder to see the effect of the important variables.

- The model would be easier to interpret by removing (i.e., setting the coefficients to zero) the unimportant variables.

Feature/Variable Selection

- Carefully selected features can improve model accuracy, but adding too many can lead to overfitting.
  - Overfitted models describe random error or noise instead of any underlying relationship.
  - They generally have poor predictive performance on test data.



- For instance, we can use a 15-degree polynomial function to fit the following data so that the fitted curve goes nicely through the data points.
- However, a brand new dataset collected from the same population may not fit this particular curve well at all.

Feature/Variable Selection (cont.)

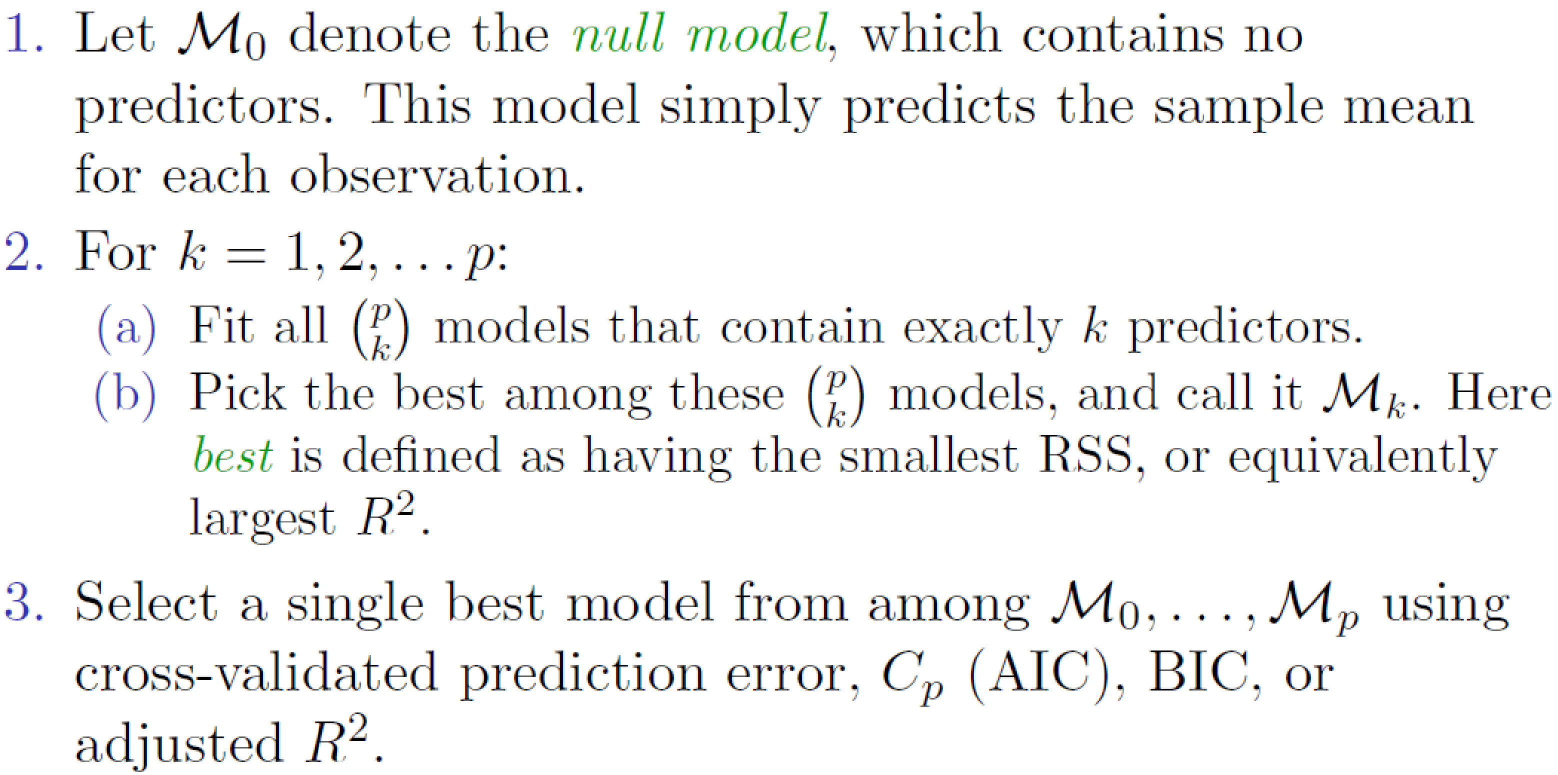
- Subset Selection
  - Identify a subset of the *p* predictors that we believe to be related to the response; then, fit a model using OLS on the reduced set.
  - Methods: best subset selection, stepwise selection

- Shrinkage (Regularization)
  - Involves shrinking the estimated coefficients toward zero relative to the OLS estimates; has the effect of reducing variance and performs variable selection.
  - Methods: ridge regression, lasso

- Dimension Reduction
  - Involves projecting the *p* predictors into a *M*-dimensional subspace, where $M < p$, and fit the linear regression model using the *M* projections as predictors.
  - Methods: principal components regression, partial least squares

Best Subset Selection

- We fit a separate OLS regression for each possible combination of the *p* predictors:

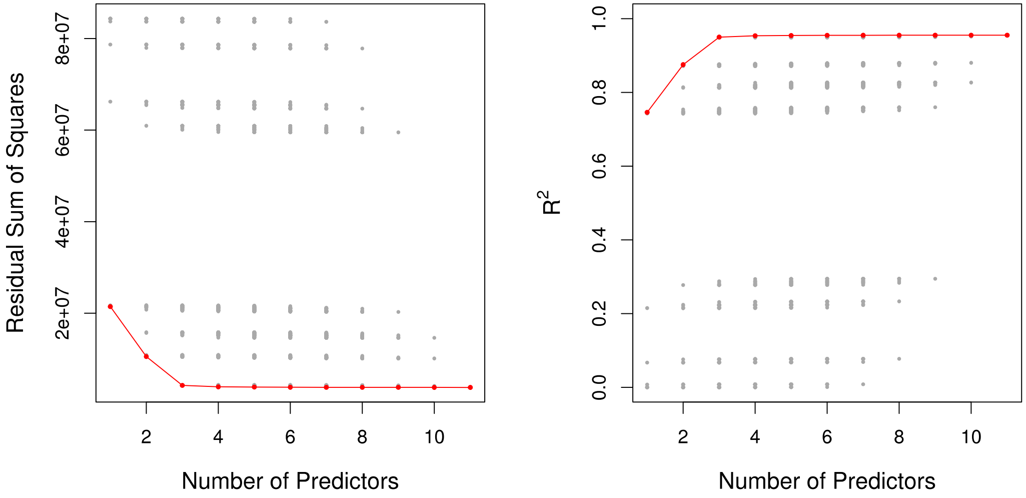


1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Best Subset Selection (cont.)

- The RSS (R^2) will always decline (increase) as the number of predictors included in the model increases, so they are not very useful statistics for selecting the *best* model.

- The red line tracks the best model for a given number of predictors, according to RSS and R^2



Best Subset Selection (cont.)

- While best subset selection is a simple and conceptually appealing approach, it suffers from computational limitations.
- The number of possible models that must be considered grows rapidly as *p* increases.
- Best subset selection becomes computationally *infeasible* for value of *p* greater than around 40.

Stepwise Selection

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

- 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.

- An enormous search space can lead to overfitting and high variance of the coefficient estimates.

Stepwise Selection (cont.)

More attractive methods include:

- Forward Stepwise Selection
  - Begins with a null OLS model containing no predictors, and then adds one predictor at a time that improves the model the most until no further improvement is possible.

- Backward Stepwise Selection
  - Begins with a full OLS model containing all predictors, and then deletes one predictor at a time that improves the model the most until no further improvement is possible.

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 $\mathcal{M}_k$ with one additional predictor.
   2.2 Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

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 $\mathcal{M}_k$, for a total of $k-1$ predictors.
   - 2.2 Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Stepwise Selection (cont.)

- Both forward and backward stepwise selection approaches search through only $1 + p(p + 1)/2$ models, so they can be applied in settings where p is too large to apply best subset selection.

- Both of these stepwise selection methods are *not* guaranteed to yield the best model containing a subset of the p predictors.

- Forward stepwise selection can be used even when $n < p$, while backward stepwise selection requires that $n > p$.

- There is a *hybrid* version of these two stepwise selection methods.

Choosing the Optimal Model

- The model containing all 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. Note that training error is usually a poor estimate of test error.

- Thus, 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

1. We can indirectly estimate test error by making an *adjustment* to the training error to account for the bias due to overfitting.

2. We can *directly* estimate the test error, using either a validation set approach or a cross-validation approach.

Other Measures of Comparison

- To compare different models, we can use other approaches:
  - Adjusted R^2
  - AIC (Akaike information criterion)
  - BIC (Bayesian information criterion)
  - Mallow's C_p (equivalent to AIC for linear regression)

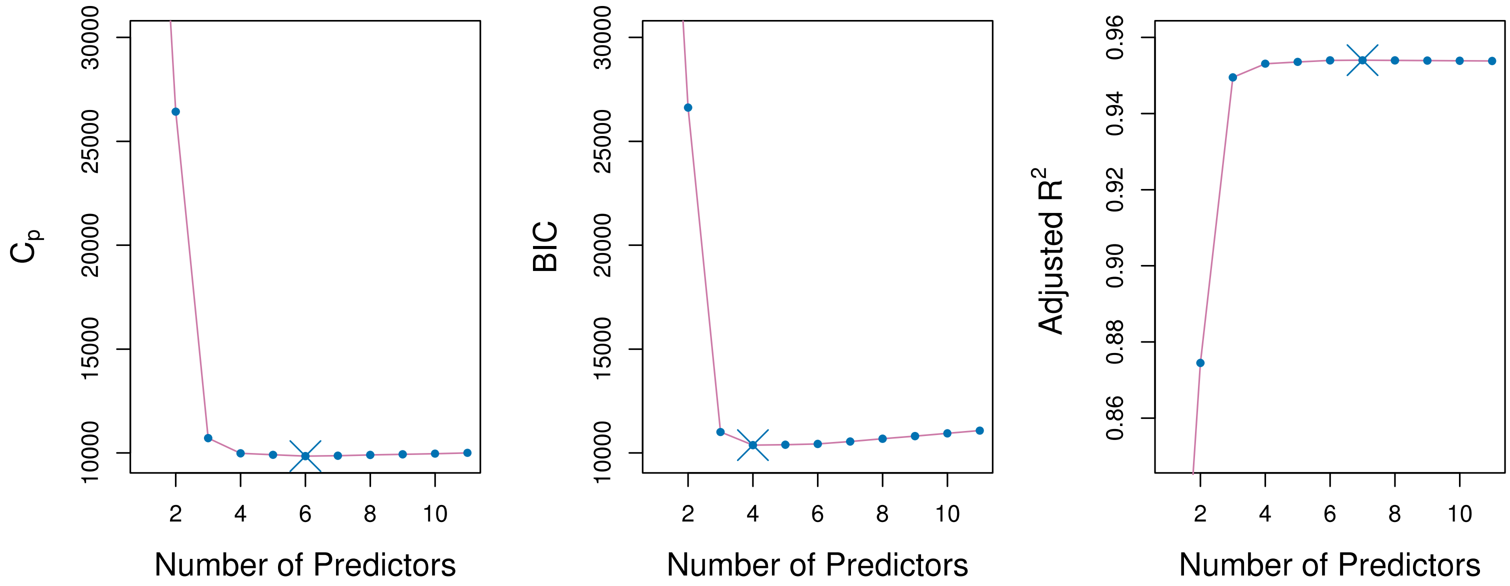
- These techniques adjust the training error for the model size, and can be used to select among a set of models with different numbers of variables.

- These methods add penalty to RSS for the number of predictors in the model.

Credit Data: C_p, BIC, and Adjusted R^2

- A small value of C_p and BIC indicates a low error, and thus a better model.

- A large value for the Adjusted R^2 indicates a better model.



Mallow's C_p

- For a fitted OLS model containing *d* predictors, the C_p estimate of test MSE:

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

where $\hat{\sigma}^2$ is an estimate of the variance of the error ε associated with each response measurement.

- Here, a penalty is added to the training RSS in order to adjust for the fact that the training error tends to underestimate the test error.

Akaike Information Criterion (AIC)

- Defined for a large class of models fit by maximum likelihood.

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

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

- In the case of the linear model with Gaussian errors, MLE and OLS are the same thing; thus, C_p and AIC are equivalent.

Bayesian Information Criterion (BIC)

- 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.

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

- Since log *n* > 2 for an *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.

- 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.

Adjusted R^2

- For an OLS model with *d* variables, the adjusted R^2 is calculated:

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

where TSS is the total sum of squares.

- Unlike the other statistics, a large value of adjusted R^2 indicates a model with a small test error.

- The adjusted R^2 statistics *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 indexed by model size *k* = 0, 1, 2, … Our job here is to select $\hat{k}$.

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

- This procedure provides a direct estimate of the test error, and it can also be used in a wider range of model selection tasks.

Shrinkage (Regularization) Methods

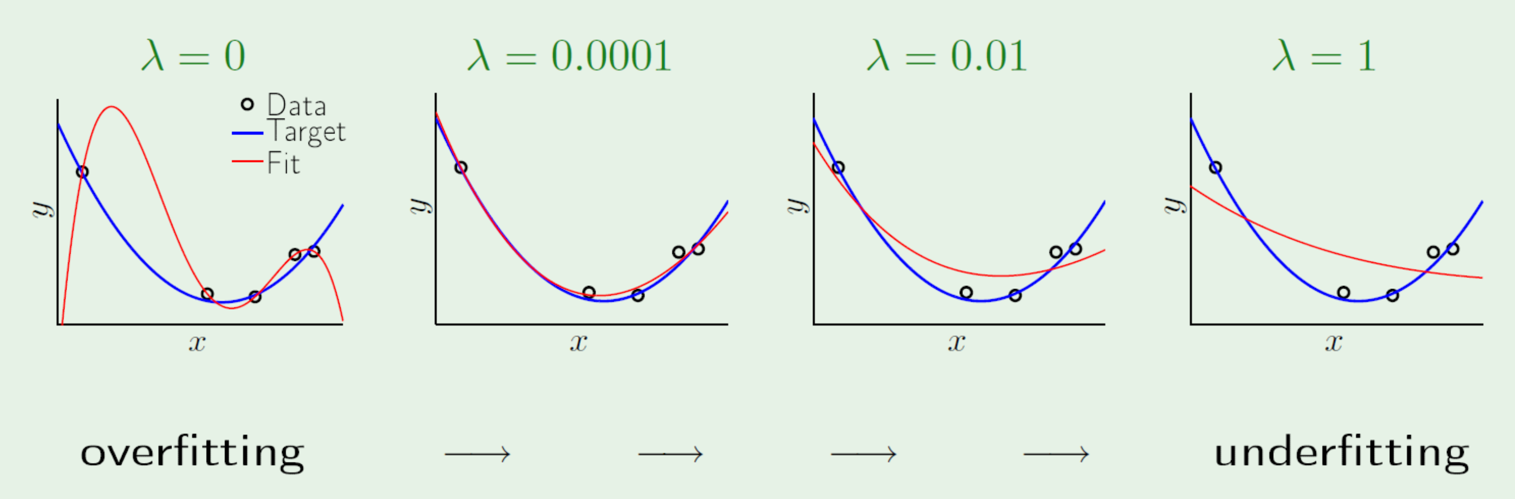
- The subset selection methods use OLS to fit a linear model that contains a subset of the predictors.

- As an alternative, we can fit a model containing all *p* predictors using a technique that constrains or *regularizes* the coefficient estimates (i.e., *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.

Shrinkage (Regularization) Methods (cont.)

- Regularization is our first weapon to combat overfitting.
- It constrains the prediction algorithm to improve out-of-sample error (i.e., test error), especially when noise is present.
- Look at what a little regularization can do:



Ridge Regression

- Recall that the OLS fitting procedure estimates the beta coefficients using the values that minimize:

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

- Ridge regression is similar to OLS, except that the coefficients are estimated by minimizing a slightly different quantity:

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

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

Ridge Regression (cont.)

- Note that $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage.
- The idea of penalizing by the sum-of-squares of the parameters is also used in neural networks, where it is known as *weight decay*.
- An equivalent way to write the ridge problem is:

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 \leq t,$$

Ridge Regression (cont.)

- The effect of this equation is to add a shrinkage penalty of the form

$$\lambda \sum_{j=1}^{p} \beta_j^2,$$

where the tuning parameter λ is a positive value.

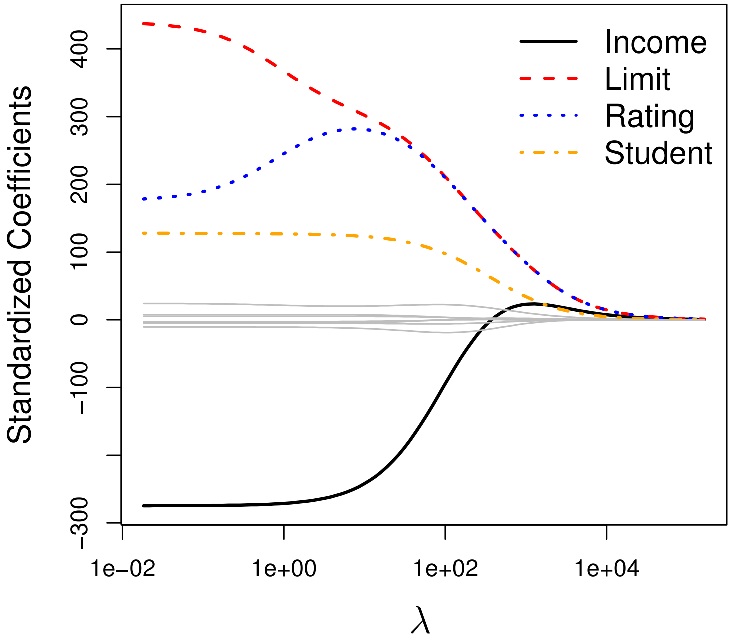
- This has the effect of shrinking the estimated beta coefficients towards zero. It turns out that such a constraint should improve the fit, because shrinking the coefficients can significantly reduce their variance.

- Note that when λ = 0, the penalty term has no effect, and ridge regression will procedure the OLS estimates. Thus, selecting a good value for λ is critical (can use cross-validation for this).

Ridge Regression (cont.)

- As λ increases, the standardized ridge regression coefficients shrinks towards zero.

- Thus, when λ is extremely large, then all of the ridge coefficient estimates are basically zero; this corresponds to the *null model* that contains no predictors.



Ridge Regression (cont.)

- The standard OLS coefficient estimates are *scale equivariant*.

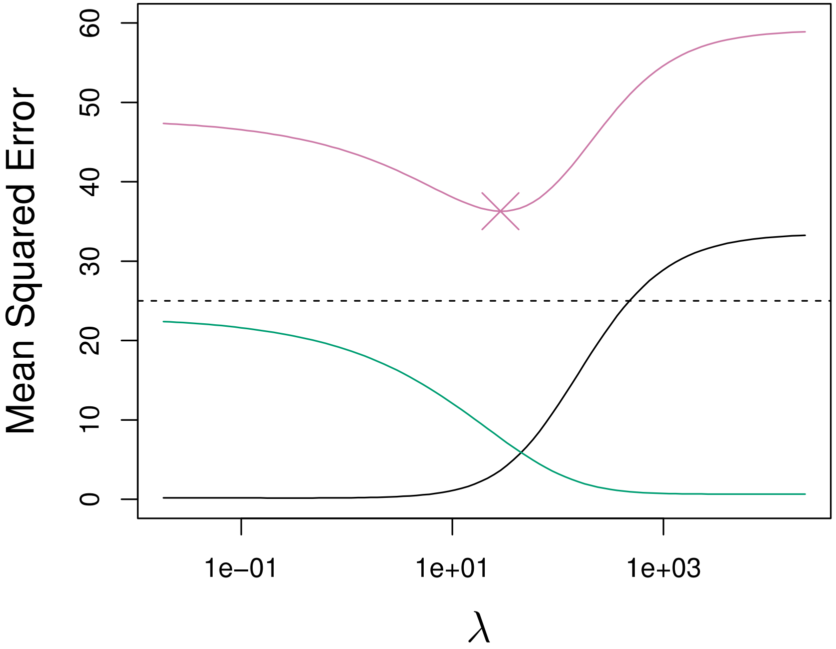
- However, the ridge regression coefficient estimates can change *substantially* when multiplying a given predictor by a constant, due to the sum of squared coefficients term in the penalty part of the ridge regression objective function.

- Thus, it is best to apply ridge regression after *standardizing the predictors.*

Ridge Regression (cont.)

- It turns out that the OLS estimates generally have low bias but can be highly variable.
- In particular when *n* and *p* are of similar size or when $n < p$, then the OLS estimates will be extremely variable
- The penalty term makes the ridge regression estimates *biased* but can also substantially reduce variance.
- As a result, there is a bias/variance trade-off.

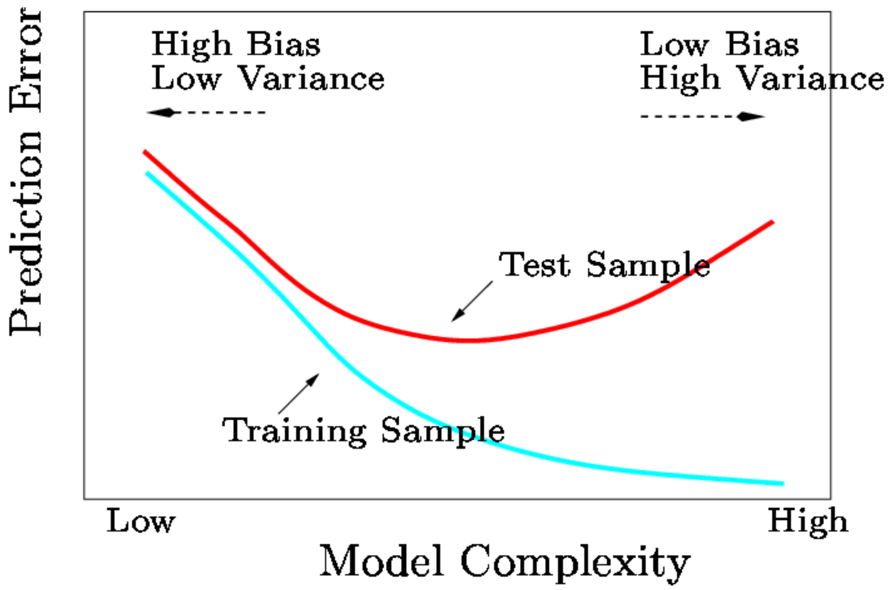
Ridge Regression (cont.)



- Black = Bias
- Green = Variance
- Purple = MSE

- Increased λ leads to increased bias but decreased variance

Ridge Regression (cont.)



- In general, the ridge regression estimates will be more biased than the OLS ones but have lower variance.

- Ridge regression will work best in situations where the OLS estimates have high variance.

Ridge Regression (cont.)

Computational Advantages of Ridge Regression

- If *p* is large, then using the best subset selection approach requires searching through enormous numbers of possible models.
- With ridge regression, for any given λ we only need to fit one model and the computations turn out to be very simple.
- Ridge regression can even be used when $p > n$, a situation where OLS fails completely (i.e., OLS estimates do not even have a unique solution).

Ridge Regression (cont.)

- In matrix form:

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

the ridge regression solutions are easily seen to be

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

- The solution adds a positive constant to the diagonal of $\mathbf{X}^T\mathbf{X}$ before inversion (making the problem non-singular).

- The *singular value decomposition* (SVD) of the centered matrix **X** gives us some additional insight into the nature of ridge regression.

Ridge Regression (cont.)

- The SVD of the *N* x *p* matrix **X** has the form **X = UDV**^{T}

- Here, **U** and **V** are *N* x *p* and *p* x *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* x *p* diagonal matrix, with diagonal entries $d_1 \geq d_2 \geq \cdots d_p \geq 0$ called singular values of **X**.

- If one or more values $d_j = 0$, **X** is singular.

Ridge Regression (cont.)

- Using SVD, we can write the OLS fitted vector as:

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

- The ridge regression solutions are:

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

where **u**$_j$ are the columns of **U**.

Ridge Regression (cont.)

- Like linear regression, ridge regression computes the coordinates of **y** with respect to the orthonormal basis **U**.

- It then *shrinks* these coordinates by the factor $\frac{d_j^2}{d_j^2+\lambda}$.

- This means that a greater amount of shrinkage is applied to the coordinates of basis vectors with smaller d_j^2.

- The SVD of the centered matrix **X** is another way of expressing the *principal components* of the variables in **X**.

Ridge Regression (cont.)

- Thus, we have $\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$, which is the *eigen decomposition* of $\mathbf{X}^T\mathbf{X}$.

- The eigenvectors v_j (columns of $\mathbf{V}$) are also called the *principal components* directions of $\mathbf{X}$.

- The first principal component direction v_1 has the property that **z**1 = $\mathbf{X}v_1$ has the largest sample variance amongst all normalized linear combinations of the columns of $\mathbf{X}$.

- The small singular values d_j correspond to directions in the column space of $\mathbf{X}$ having small variance, and ridge regression shrinks these directions the most.

The Lasso

- One significant problem of ridge regression is that the penalty term will never force any of the coefficients to be exactly zero.
- Thus, the final model will include all *p* predictors, which creates a challenge in model interpretation
- A more modern alternative is the *lasso*.
- The lasso works in a similar way to ridge regression, except it uses a different penalty term that shrinks some of the coefficients exactly to zero.

The Lasso (cont.)

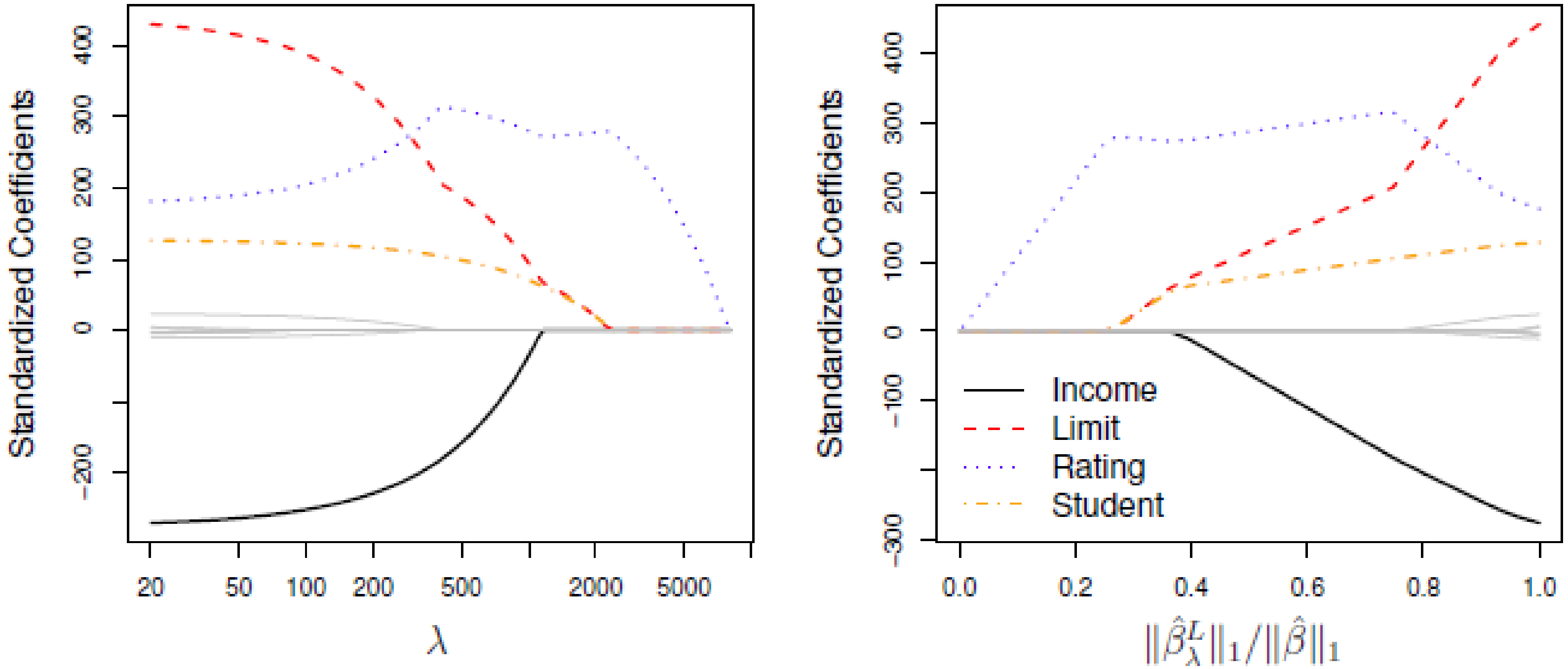
- The lasso coefficients minimize the quantity:

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

- The key difference from ridge regression is that the lasso uses an ℓ_1 penalty instead of an ℓ_2, which has the effect of forcing some of the coefficients to be exactly equal to zero when the tuning parameter λ is sufficiently large.

- Thus, the lasso performs variable/feature selection.

The Lasso (cont.)



- When λ = 0, then the lasso simply gives the OLS fit.
- When λ becomes sufficiently large, the lasso gives the null model in which all coefficient estimates equal zero.

The Lasso (cont.)

- One can show that the lasso and ridge regression coefficient estimates solves the problems:

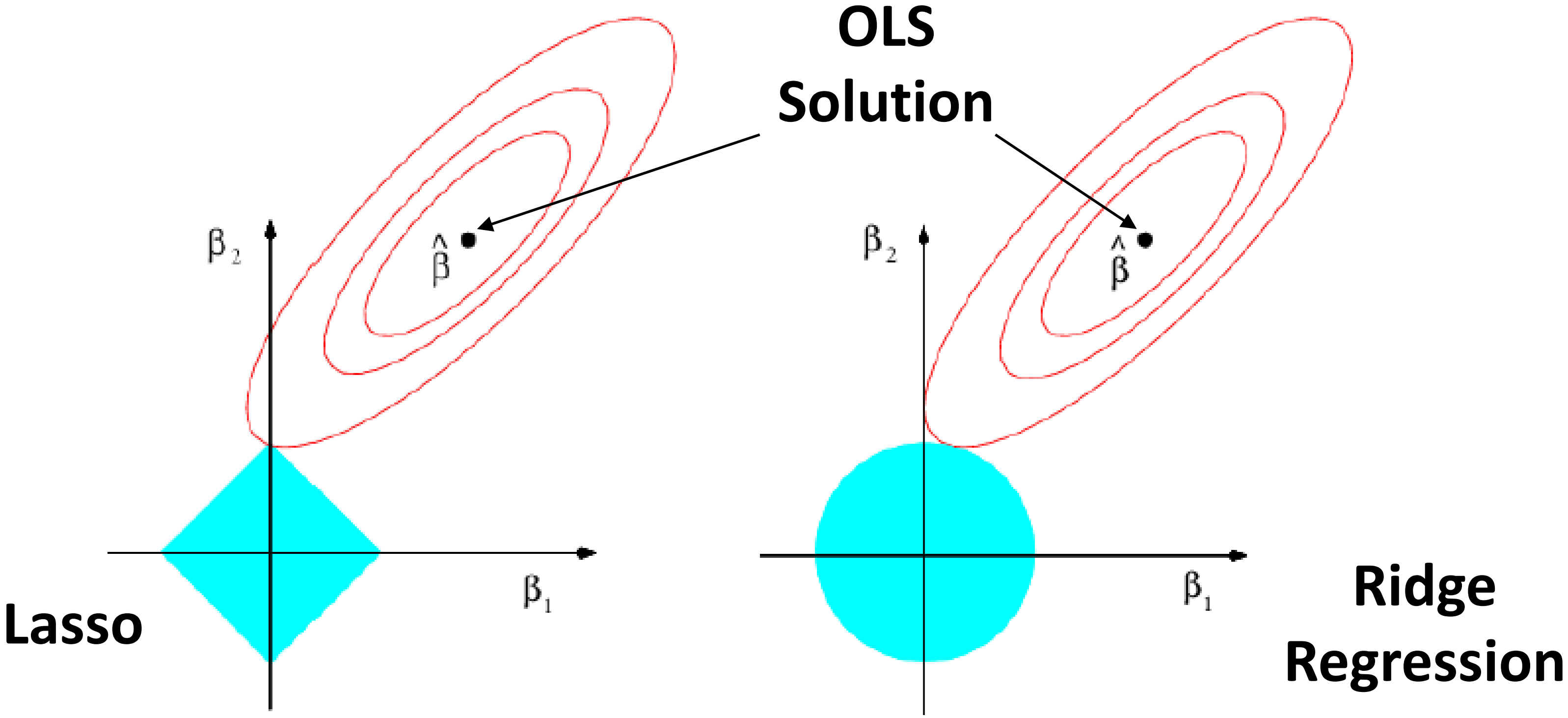
$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p}|\beta_j| \leq s$$

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 (cont.)

- The lasso and ridge regression coefficient estimates are given by the first point at which an ellipse contacts the constraint region.



OLS Solution

Lasso

Ridge Regression

Lasso vs. Ridge Regression

- The lasso has a major advantage over ridge regression, in that it produces simpler and more interpretable models that involved only a subset of predictors.

- The lasso leads to qualitatively similar behavior to ridge regression, in that as λ increases, the variance decreases and the bias increases.

- The lasso can generate more accurate predictions compared to ridge regression.

- Cross-validation can be used in order to determine which approach is better on a particular data set.

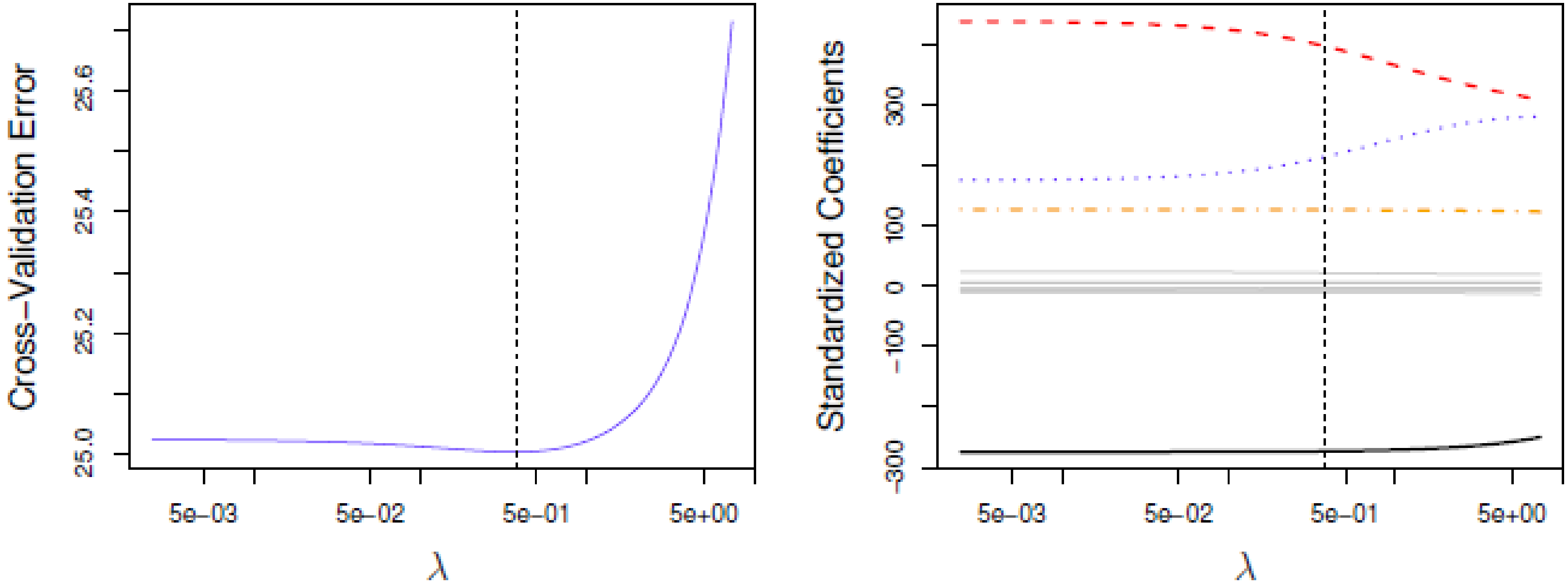
Selecting the Tuning Parameter λ

- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration in best; thus, we required a method selecting a value for the tuning parameter λ or equivalently, the value of the constraint *s*.

- Select a grid of potential values; use cross-validation to estimate the error rate on test data (for each value of λ) and select the value that gives the smallest error rate.

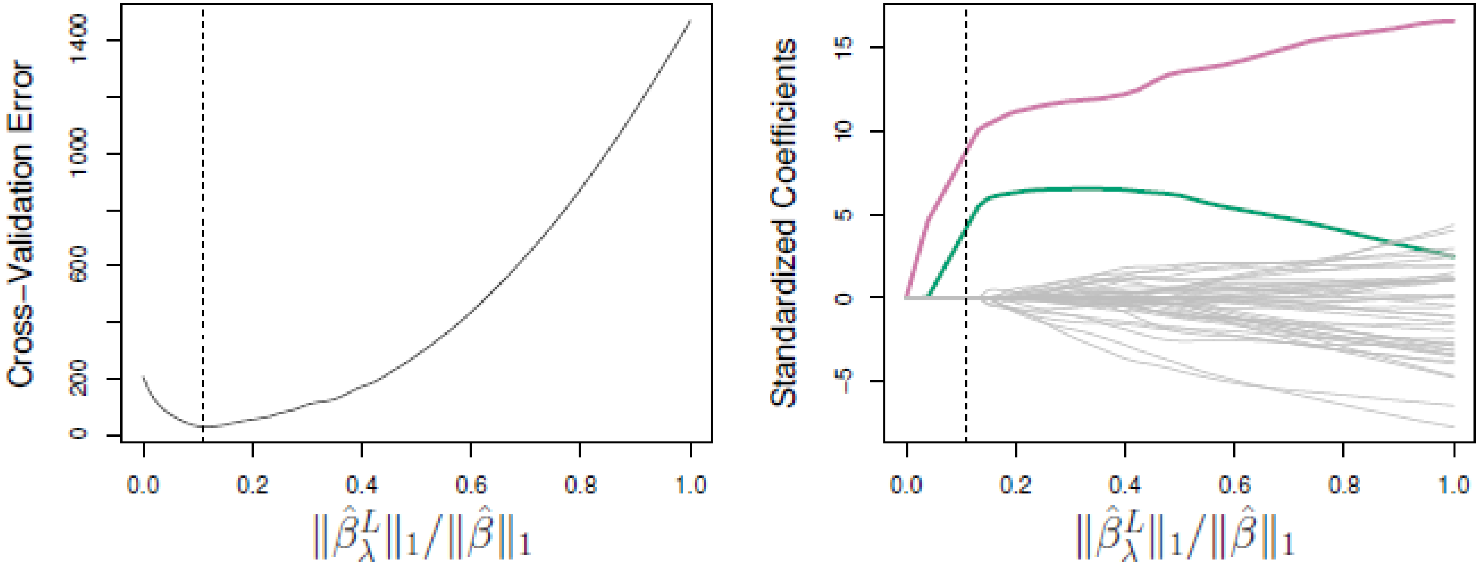
- Finally, the model is re-fit using all of the variable observations and the selected value of the tuning parameter λ.

Selecting the Tuning Parameter λ: 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.

Selecting the Tuning Parameter λ: 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.

Dimension Reduction

- The methods we have discussed so far have involved fitting linear regression models, via OLS or a shrunken approach, using the original predictors.

- We now explore a class of approaches that *transform* the predictors and then fit an OLS model using the transformed variables.

- We refer to these techniques as *dimension reduction* methods.

Dimension Reduction (cont.)

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

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

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

- We can then fit an OLS linear regression model,

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

Dimension Reduction (cont.)

- If the constants $\phi_{m1}, \ldots, \phi_{mp}$ are chosen wisely, then such dimension reduction approaches can outperform OLS regression.

- The term *dimension reduction* comes from the fact that this approach reduces the problem of estimating the $p + 1$ coefficients $\beta_0, \ldots, \beta_p$ to the simpler problem of estimating the $M + 1$ coefficients $\theta_0, \ldots, \theta_M$, where $M < p$.

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

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

- This method serves to constrain the estimated β_j coefficients.

Principal Components Regression

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

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

- The *second principal component* has largest variance, subject to being uncorrelated with the first….etc.

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

Principal Components Regression (cont.)

- The *principal components regression* (PCR) approach involves constructing the first *M* principal components, and then using these components as the predictors in an OLS linear regression model.

- The key idea is that often a small number of principal components suffice to *explain* most of the variability in the data, as well as the relationship with the response.

- We assume that the directions in which $X_1,\ldots,X_p$ show the most variation are the directions that are associated with *Y*.

- When performing PCR, predictors should be *standardized* prior to generating the principal components.

Principal Components Regression (cont.)

- PCR forms the derived input columns $\mathbf{z}_m = \mathbf{X}v_m$, and then regresses $\mathbf{y}$ on $\mathbf{z}_1, \mathbf{z}_2,\ldots,\mathbf{z}_\mathrm{M}$ for some $M \leq p$. Since the $\mathbf{z}_m$ are orthogonal, this regression is just a sum of univariate regressions:

$$\hat{\mathbf{y}}^{\mathrm{pcr}}_{(M)} = \bar{y}\mathbf{1} + \sum_{m=1}^{M} \hat{\theta}_m \mathbf{z}_m$$

where $\hat{\theta}_m = \langle \boldsymbol{z}_m, \boldsymbol{y} \rangle / \langle \boldsymbol{z}_m, \boldsymbol{z}_m \rangle$. Since $\mathbf{z}_m$ are each linear combinations of the original $\mathbf{x}_j$, we can express the solution in terms of coefficients of the $\mathbf{x}_j$.

$$\hat{\beta}^{\mathrm{pcr}}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m$$

Principal Components Regression (cont.)

- By manually setting the projection onto the principal component directions with small eigenvalues set to 0 (i.e., only keeping the large ones), dimension reduction is achieved.

- PCR is very similar to ridge regression in a certain sense.

- Ridge regression can be viewed conceptually as projecting the *y* vector onto the principal component directions and then shrinking the projection on each principal component direction.

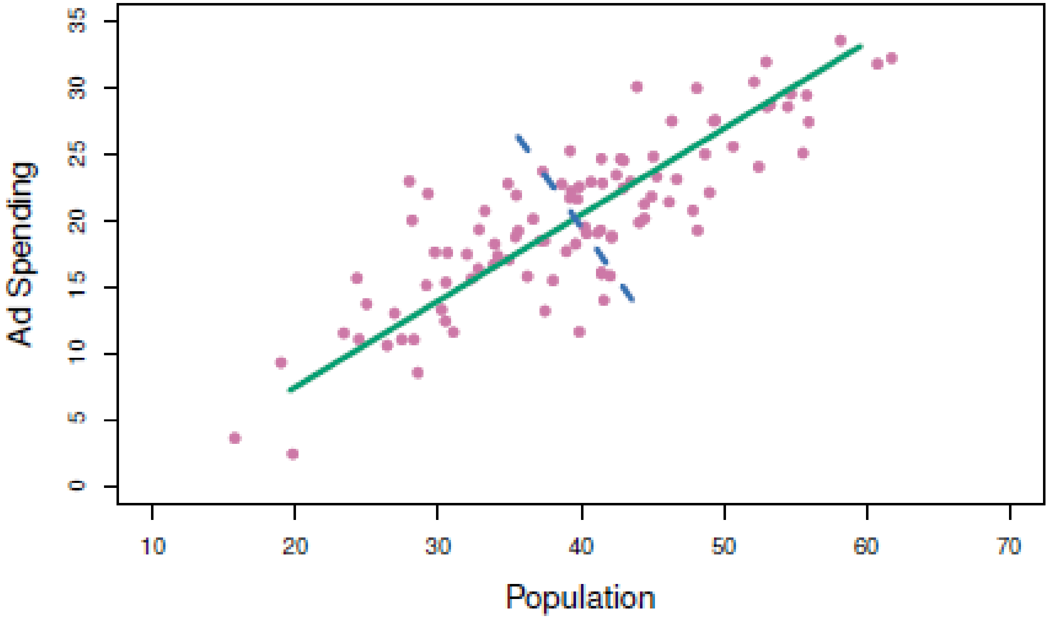
Principal Components Regression (cont.)

- The amount of shrinkage depends on the variance of that principal component.

- Ridge regression shrinks everything, but it never shrinks anything to zero.

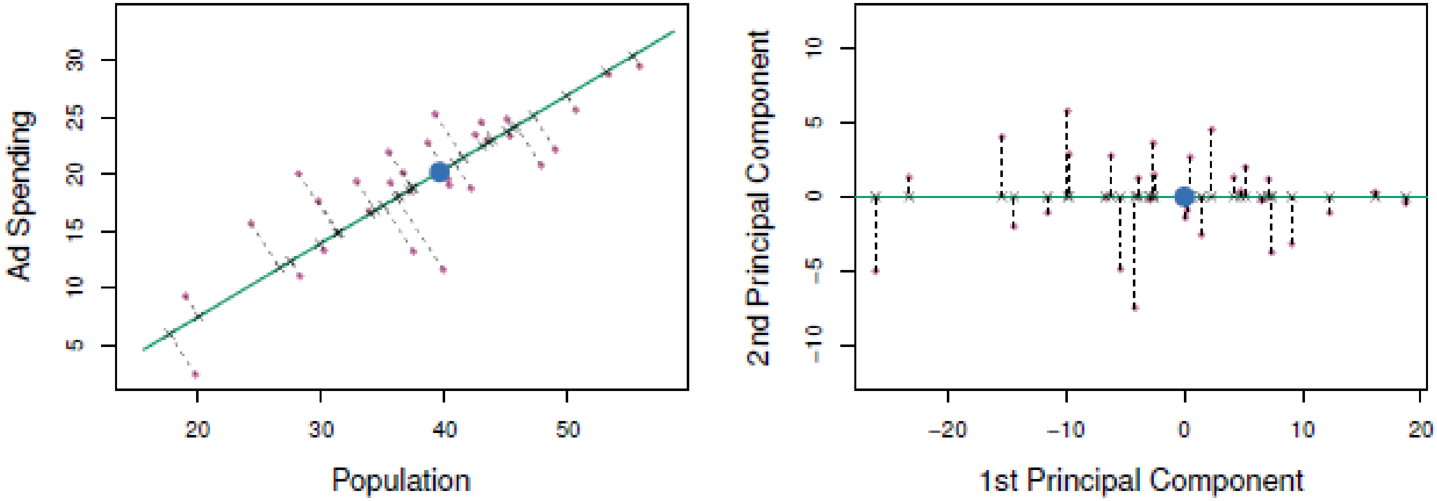
- By contrast, PCR either does not shrink a component at all or shrinks it to zero.

Principal Components Regression (cont.)



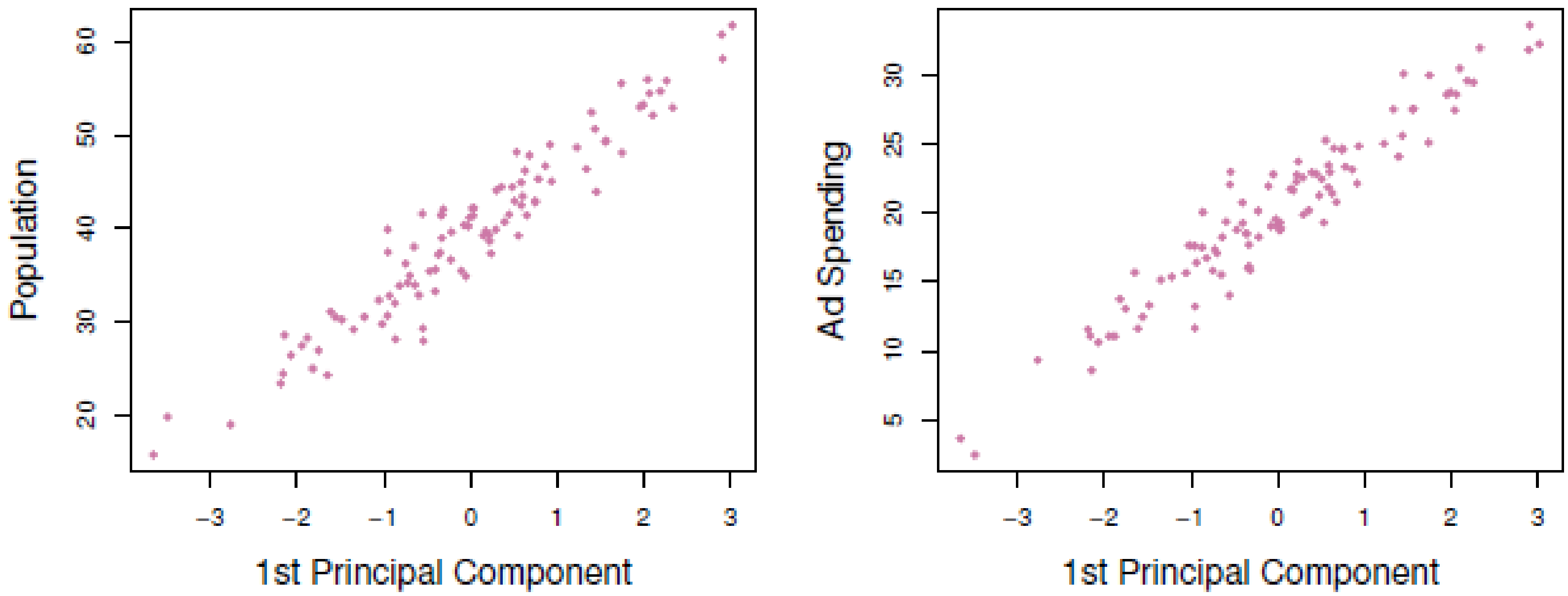
*The population size (*pop*) and ad spending (*ad*) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

Principal Components Regression (cont.)



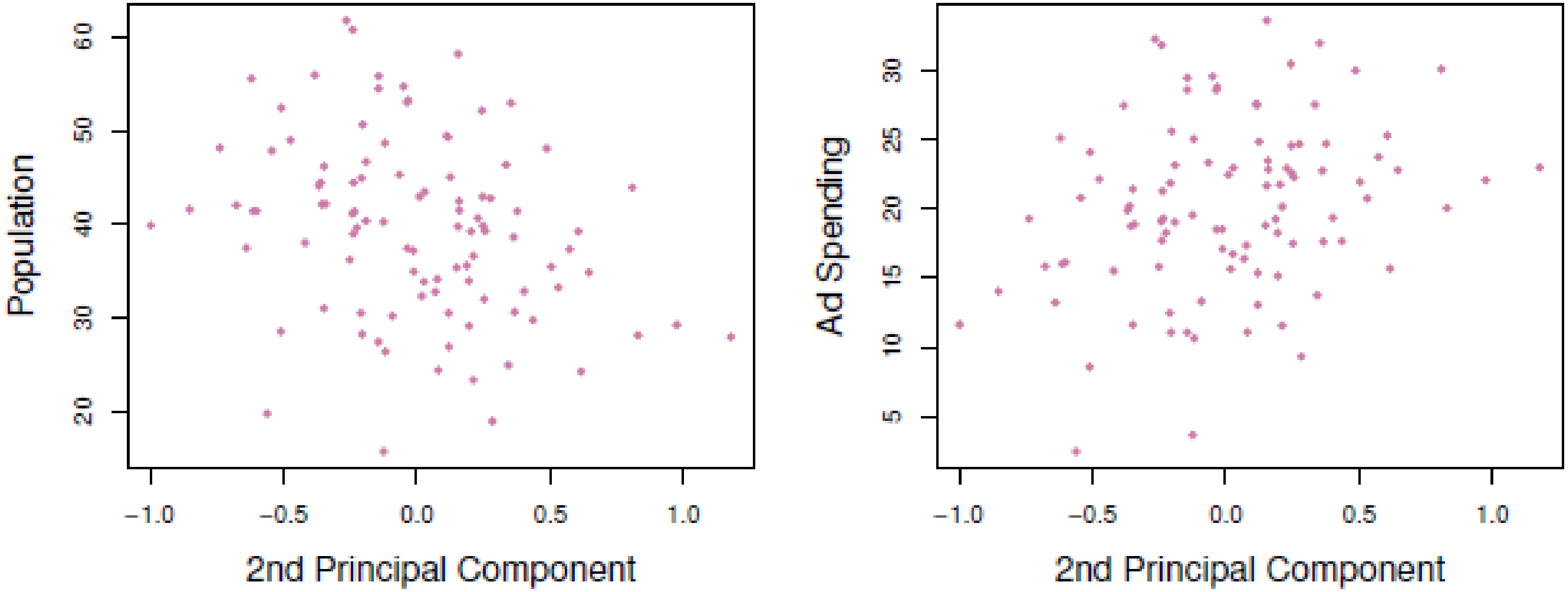
A subset of the advertising data. Left: *The first principal component, chosen to minimize the sum of the squared perpendicular distances to each point, is shown in green. These distances are represented using the black dashed line segments.* Right: *The left-hand panel has been rotated so that the first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

Principal Components Regression (cont.)



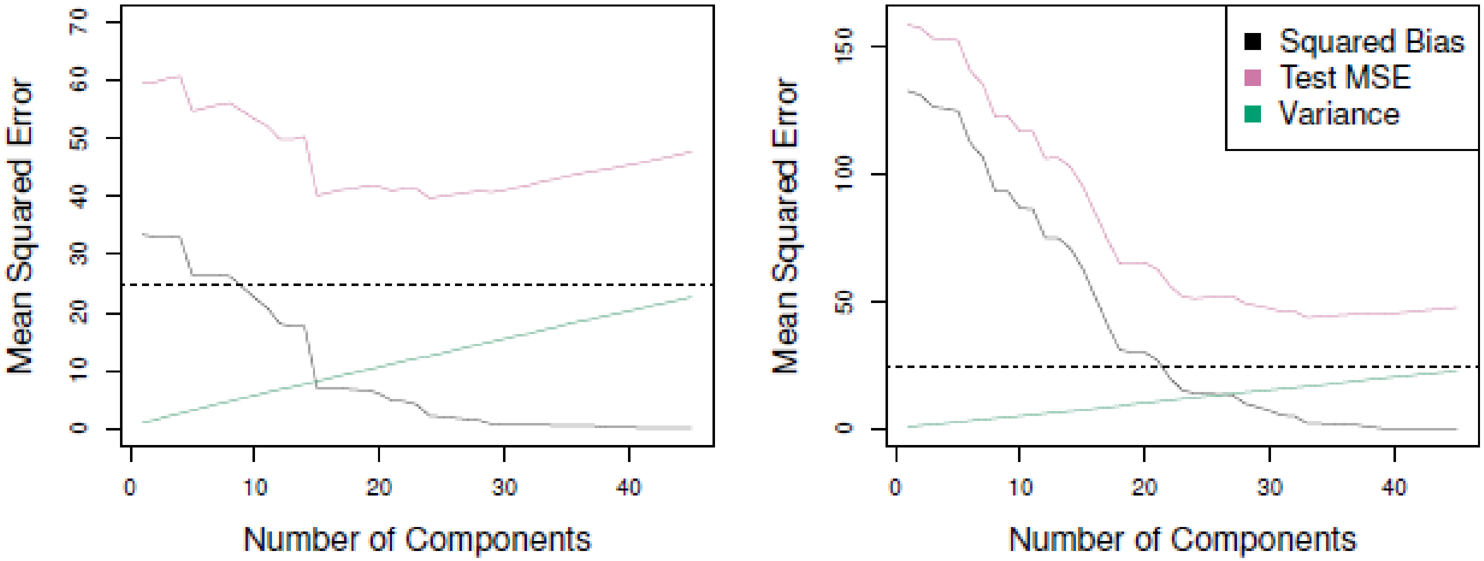
Plots of the first principal component scores z_{i1} *versus* pop *and* ad. *The relationships are strong.*

Principal Components Regression (cont.)



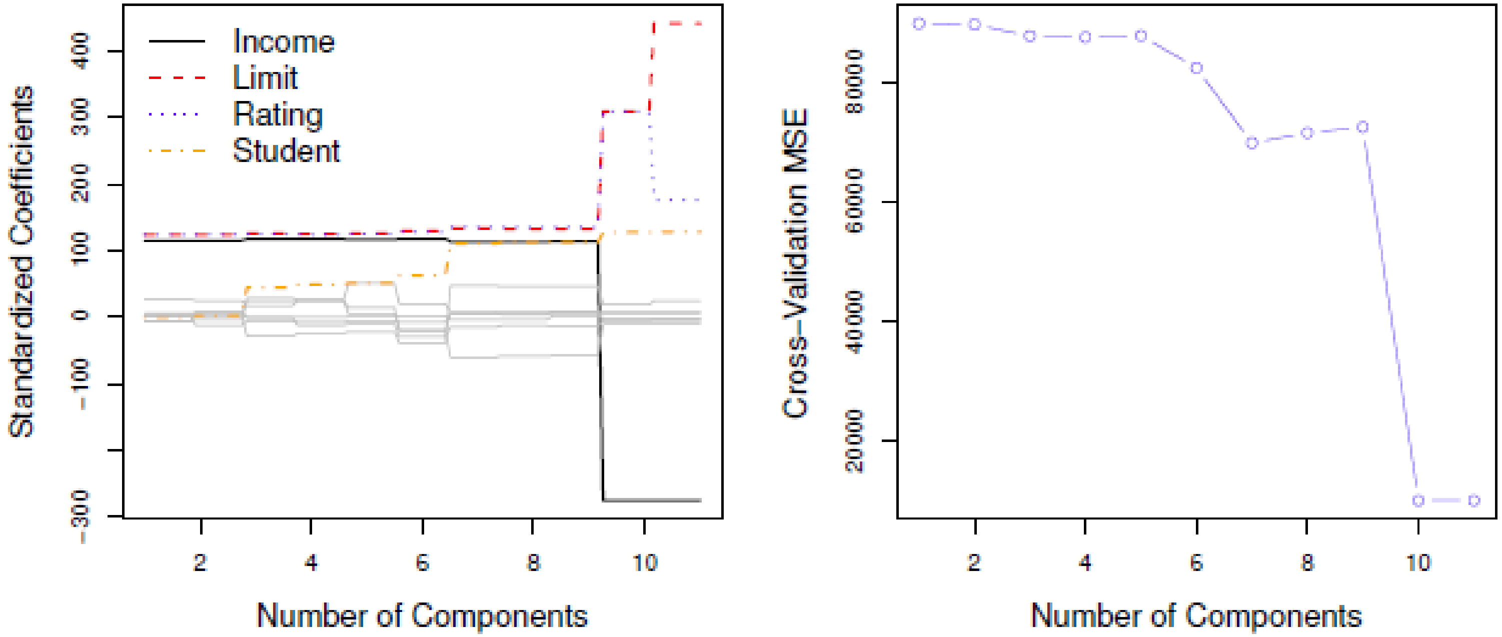
Plots of the second principal component scores z_{i2} versus pop *and* ad*. The relationships are weak.*

Principal Components Regression (cont.)



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 Components Regression (cont.)



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.*

Principal Components Regression (cont.)

- As more principal components are used in the regression model, the bias decreases but the variance increases.

- PCR will tend to do well in cases when the first few principal components are sufficient to capture most of the variation in the predictors, as well as the relationship with the response.

- We note that even though PCR provides a simple way to perform regression using $M < p$ predictors, it *is not* a feature selection method.

- In PCR, the number of principal components is typically chosen by cross-validation.

Partial Least Squares

- PCR identifies linear combinations, or *directions*, that best represents the predictors.

- These directions are identified is 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.

- 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 (cont.)

- Like PCR, *partial least squares* (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.

- Then PLS fits an OLS linear model using these *M* new features.

- Unlike PCR, PLS identifies these new features in a *supervised* way; PLS 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*.

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

Partial Least Squares (cont.)

- After standardizing the p predictors, PLS computes the first partial least squares direction Z_1 by setting each ϕ_{1j} in

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

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.

Partial Least Squares (cont.)

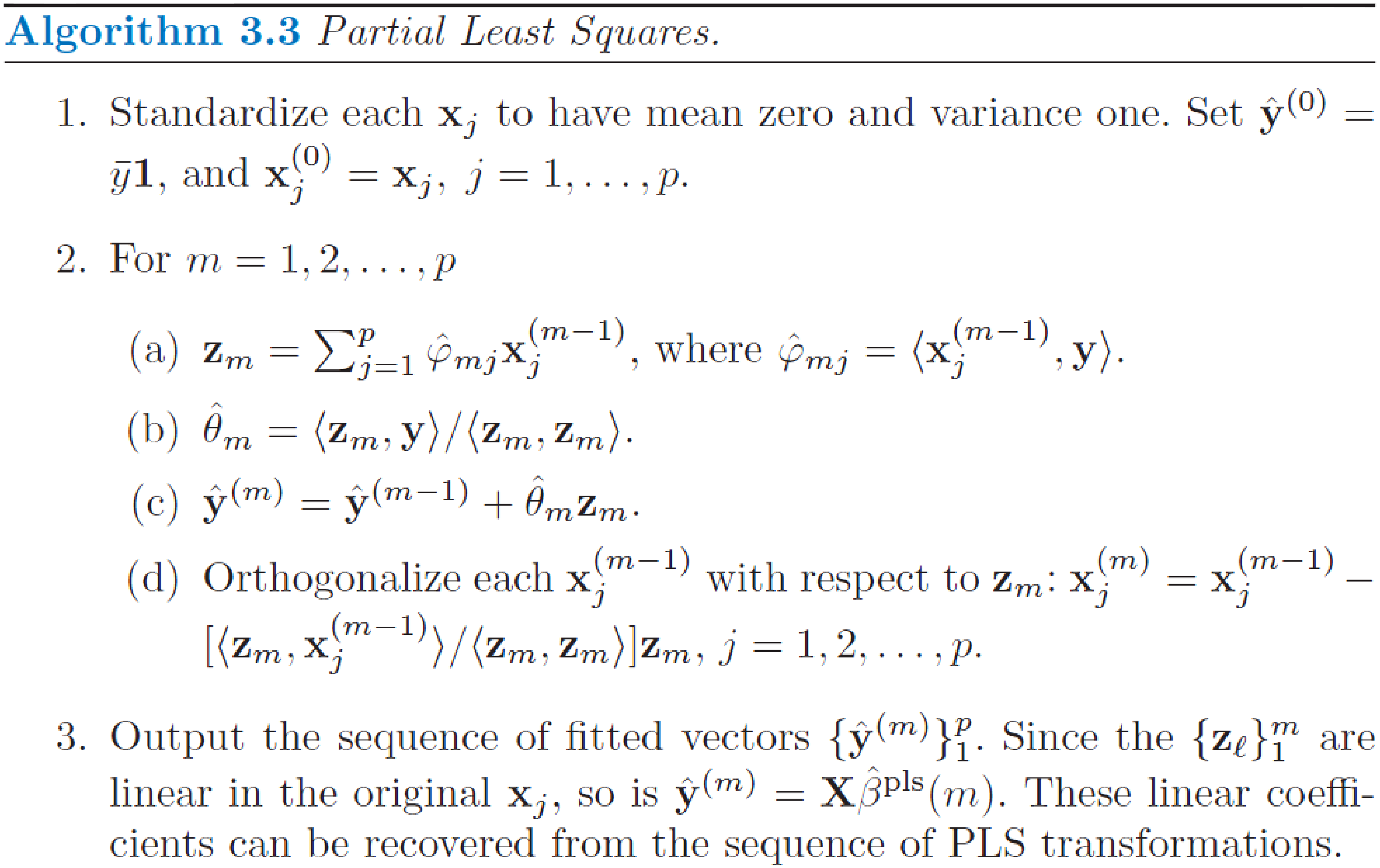
- Hence, in computing $Z_1 = \sum_{j=1}^{p} \phi_{1j} X_j$, PLS places the highest weight on the variables that are most strongly related to the response.

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

- As with PCR, the number *M* of PLS directions used in PLS is a tuning parameters that is typically chosen by cross-validation.

- While the supervised dimension reduction of PLS can reduce bias, it also has the potential to increase variance.

Partial Least Squares (cont.)



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.

Considerations in High Dimensions

- While *p* can be extremely large, the number of observations *n* is often limited due to cost, sample availability, etc.

- Data sets containing more features than observations are often referred to a *high-dimensional*.

- When the number of features *p* is as large as, or larger than, the number of observations *n*, OLS should not be performed.
    - It is too *flexible* and hence overfits the data.

- Forward stepwise selection, ridge regression, lasso, and PCR are particularly useful for performing regression in the high-dimensional setting.

Considerations in High Dimensions (cont.)

- Regularization or shrinkage plays a key role in high-dimensional problems.

- Appropriate tuning parameter selection is crucial for good predictive performance.

- The test error tends to increase as the dimensionality of the problem (i.e., the number of predictors) increases, unless the additional features are truly associated with the response.
    - Known as the *curse of dimensionality*

Considerations in High Dimensions (cont.)

- *Curse of dimensionality*
  - Adding additional *signal* features that are truly associated with the response will improve the fitted model, in the sense of leading to a reduction in test set error.
  - Adding *noise* features that are not truly associated with the response will lead to a deterioration in the fitted model, and consequently an increased test set error.

- Noise features increase the dimensionality of the problem, exacerbating the risk of overfitting without any potential upside in terms of improved test set error.

Considerations in High Dimensions (cont.)

- In the high-dimensional setting, the multicollinearity problem is extreme: any variable in the model can be written as a linear combination of all of the other variables in the models.

- It is also important to be particularly careful in reporting errors and measures of model fit in the high-dimensional setting.

- One should *never* use sum of squared errors, p-values, R^2 statistics, or other traditional measures of model fit on the *training data* as evidence of good model fit in the high-dimensional setting.

- It is important to report results on an independent test set, or cross-validation errors.

Summary

- Best subset selection and stepwise selection methods.
- Estimate test error by adjusting training error to account for bias due to overfitting.
- Estimate test error using validation set approach and cross-validation approach.
- Ridge regression and the lasso as shrinkage (regularization) methods.
- Principal components regression and partial least squares.
- Considerations for high-dimensional settings.