Class 6: Shrinkage Methods and Model Selection

MSA 8150: Machine Learning for Analytics

Alireza Aghasi

Institute for Insight, Georgia State University

What are Shrinkage Methods and Why Useful?

You would probably hear **Ridge Regression** and **LASSO** quite often

- 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 the model variance**

Ridge Regression

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

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

– In contrast, the ridge regression coefficient estimates $\hat{\beta}^R$ are the values that minimize

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

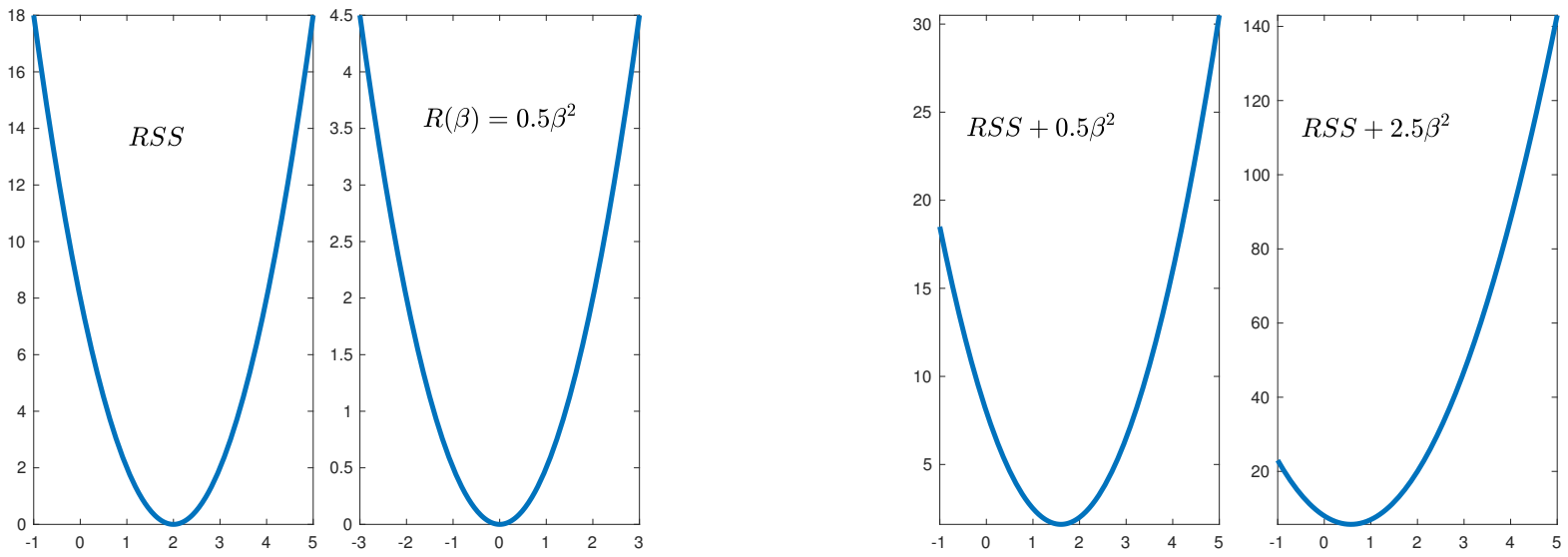
– Here, λ is a tuning parameter

Ridge Regression

- As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small
- However, the second term, $\lambda \sum_{j=1}^{p} \beta_j^2$, called a shrinkage penalty, encourages solutions that 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 (trade off between bias and variance)
- Selecting a good value for λ is critical; often cross-validation is used for this

Effect of Increasing λ on the β

– The figure below shows how increasing the Ridge penalty pushes the minimizers of the mixed RSS objective to zero



Shrinkage Example

- Previously from the homework assignments you remember that the least squares solution to fit data point $(x_1, y_1), \cdots, (x_n, y_n)$ was obtained via the minimization:

$$\min_{\beta} \sum_{i=1}^{N} (y_i - \beta x_i)^2 \quad \therefore \quad \hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

- We can show that if we run the Ridge regression problem

$$\min_{\beta} \sum_{i=1}^{N} (y_i - \beta x_i)^2 + \lambda \beta^2$$

  the new estimate becomes

$$\hat{\beta}^R = \frac{\sum_{i=1}^{n} x_i y_i}{\lambda + \sum_{i=1}^{n} x_i^2}$$

- Note how increasing λ pushes $\hat{\beta}^R$ towards zero

In Class Exercise

– For the simple regression problem of fitting $(x_1, y_1), \cdots, (x_n, y_n)$, to the model $y = \beta_0 + \beta_1 x$ show that the least-squares estimates for the Ridge regularized objective

$$\sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i)^2 + \lambda(\beta_0^2 + \beta_1^2)$$

are

$$\hat{\beta}_1^R = \frac{\sum_{i=1}^n x_i y_i - \frac{n^2}{n+\lambda} \bar{x}\bar{y}}{\lambda + \sum_{i=1}^n x_i^2 + \frac{n^2}{n+\lambda} \bar{x}^2}, \quad \hat{\beta}_0^R = \frac{1}{n+\lambda} \left(\sum_{i=1}^n y_i - \hat{\beta}_1^R \sum_{i=1}^n x_i \right)$$

What Happens in Multiple Regression?

- In this case we previously had

$$RSS = (\boldsymbol{y} - \boldsymbol{X}\beta)^\top (\boldsymbol{y} - \boldsymbol{X}\beta)$$

  which led to

$$\hat{\beta} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

- In the case of regularized problem ($\|.\|$ denotes the L-2 norm)

$$(\boldsymbol{y} - \boldsymbol{X}\beta)^\top (\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda \|\beta\|^2$$

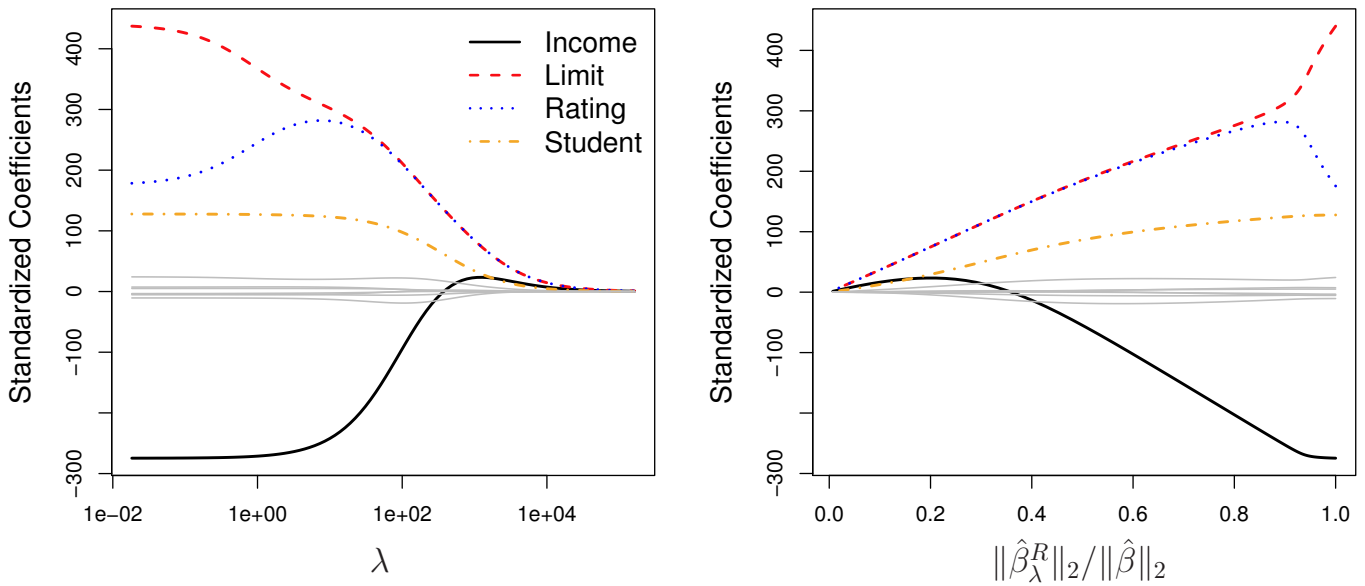
  we will have

$$\hat{\beta}^R = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

  where $\boldsymbol{I}$ is the identity matrix

Credit Data Example

- Left: 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 display $\|\hat{\beta}^R\|/\|\hat{\beta}\|$ (how much **shrinkage** happens by increasing λ)



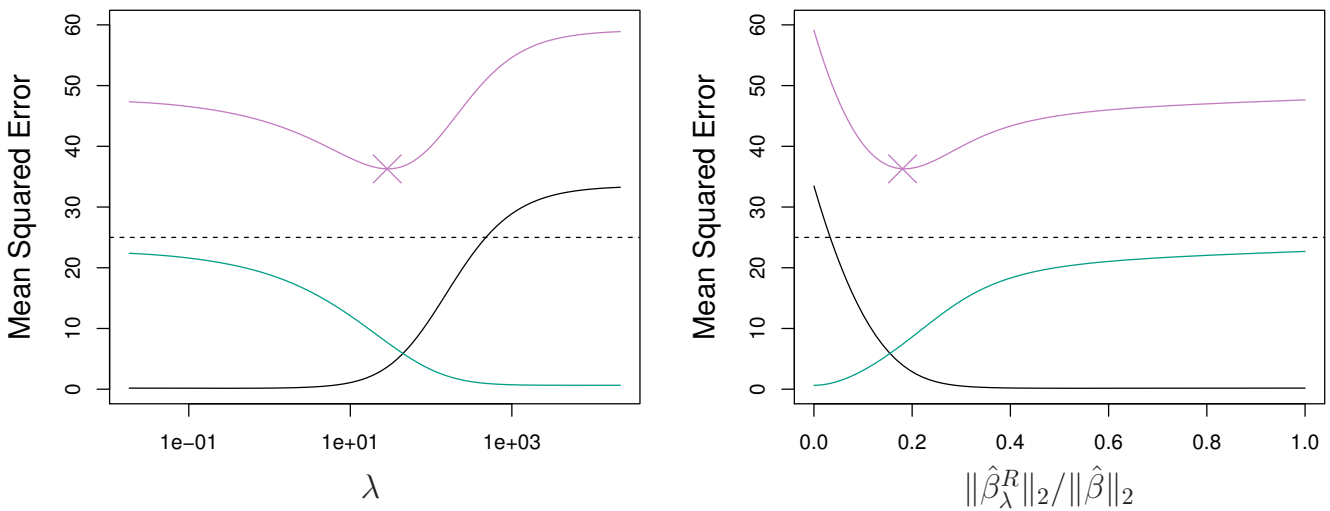
Scaling of the Predictors

- In the standard least-squares if we scale a feature value by c, the corresponding coefficient scales by c^{-1}
- However when we have the Ridge regularized objective, this is no more the case
- To see a consistent behavior, for the Ridge regularized problem we often work with standardized features:

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

Bias-Variance Trade-Off

– A toy example: 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 $\|\hat{\beta}^R\|/\|\hat{\beta}\|$. The horizontal dashed lines indicate the minimum possible MSE (**the standard least squares,** $\lambda = 0$ **in nowhere close**). The purple crosses indicate smallest ridge regression model MSE values



– Remember (test error = bias + variance + noise variance)

LASSO (Least Absolute Selection and Shrinkage Operator)

- 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 (none of the model coefficients become explicitly zero)
- The Lasso is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients minimize the quantity

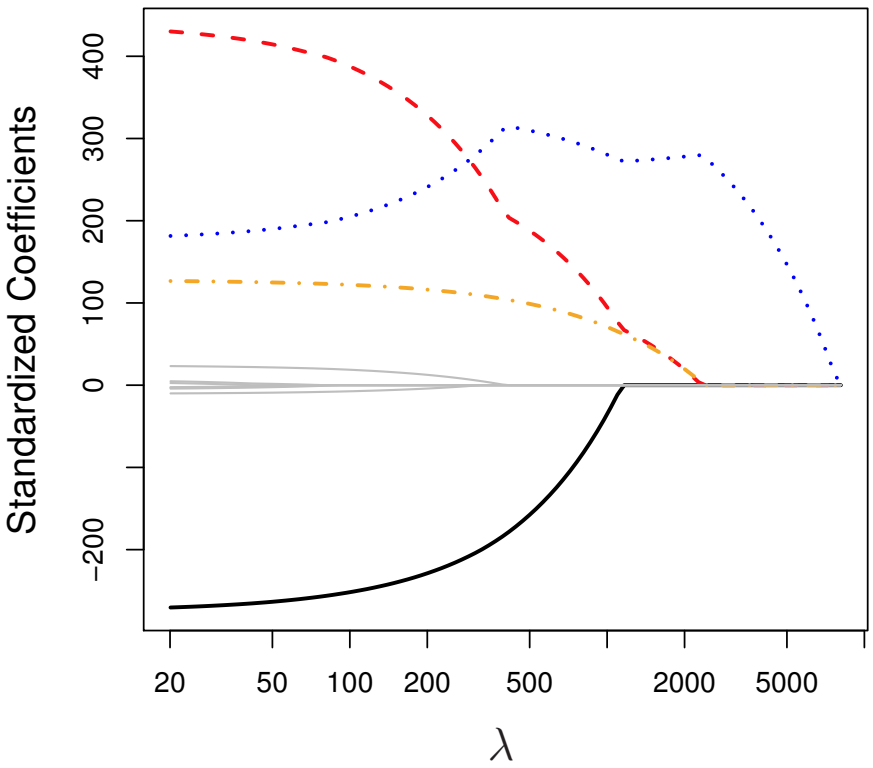
$$RSS_{LASSO} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \|\boldsymbol{\beta}\|_1$$

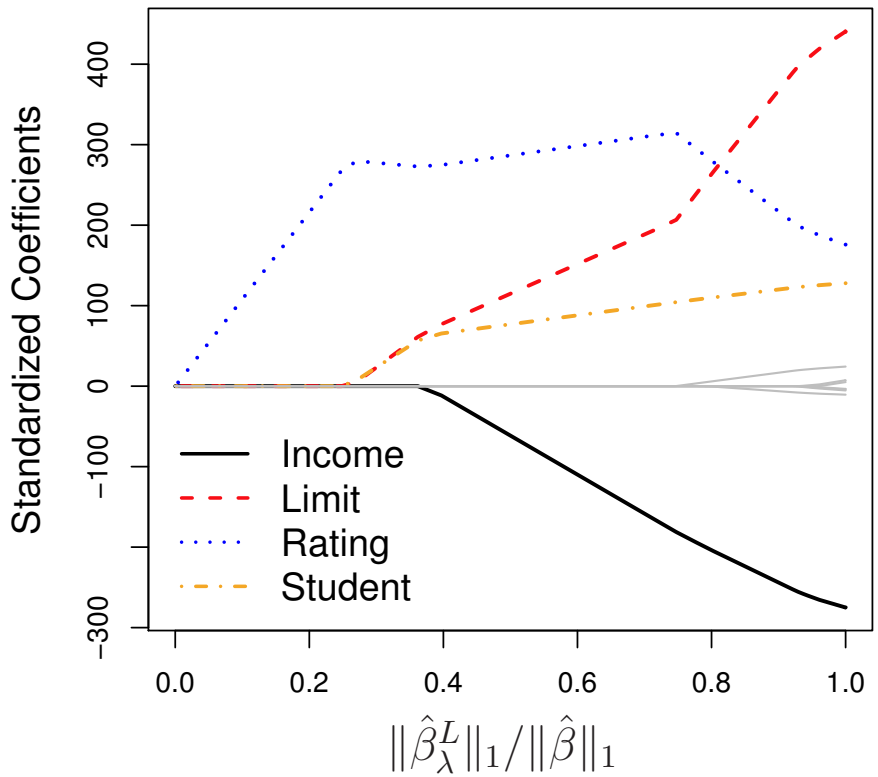
- We call $\|\boldsymbol{\beta}\|_1$ the L-1 norm of $\boldsymbol{\beta}$

LASSO

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero
- However, in the case of the lasso, the L1 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
- Technically the lasso performs the training and variable selection together
- 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 for the lasso is critical; cross-validation is again the method of choice

Example: Credit Data





Why LASSO promotes Sparsity?

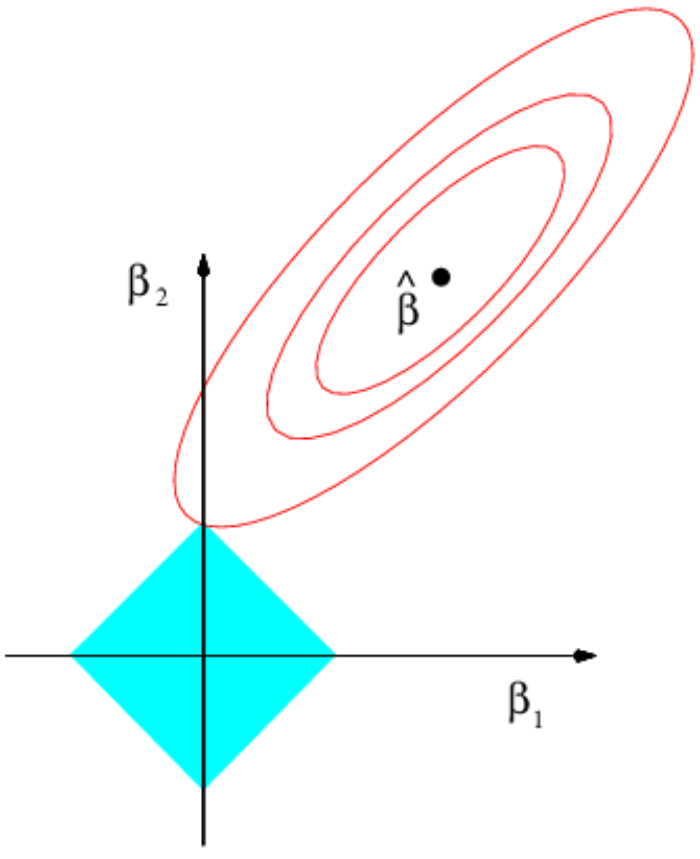
– From a convex optimization perspective, one can show that the lasso and ridge regression coefficient estimates solve the problems

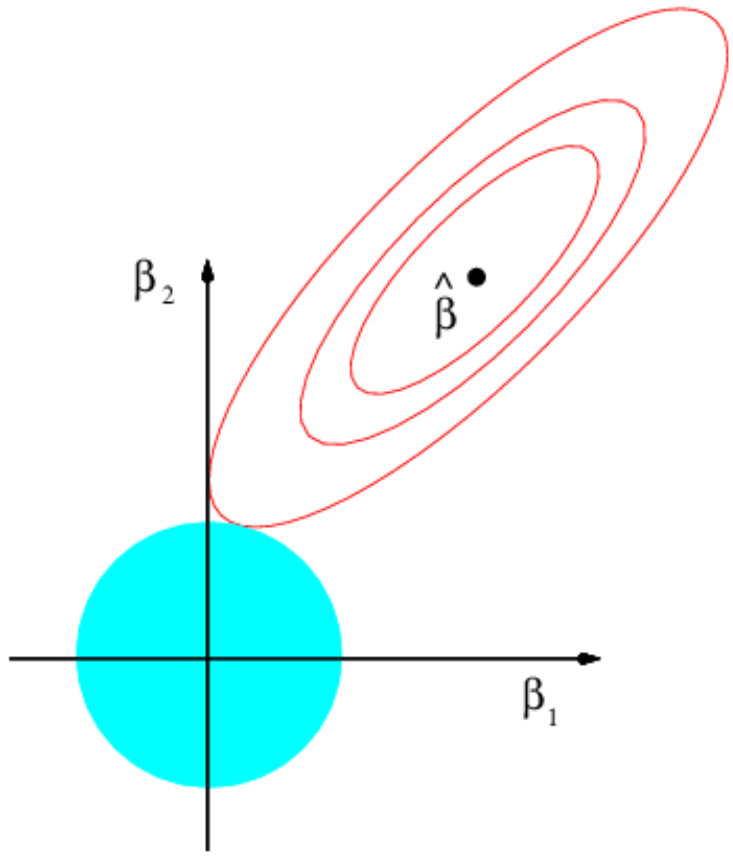
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \quad \textit{subject to} \quad \|\boldsymbol{\beta}\|_1 \leq \tau$$

and

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \quad \textit{subject to} \quad \|\boldsymbol{\beta}\|_2 \leq \tau'$$

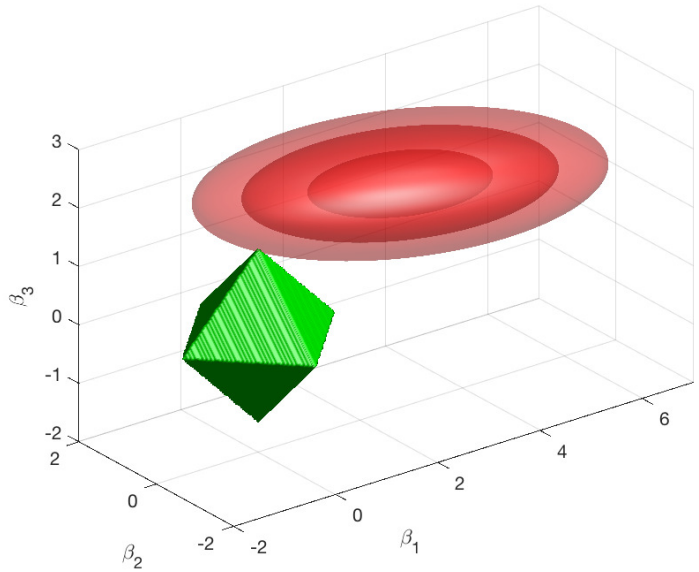
The Geometry of the Two Problems

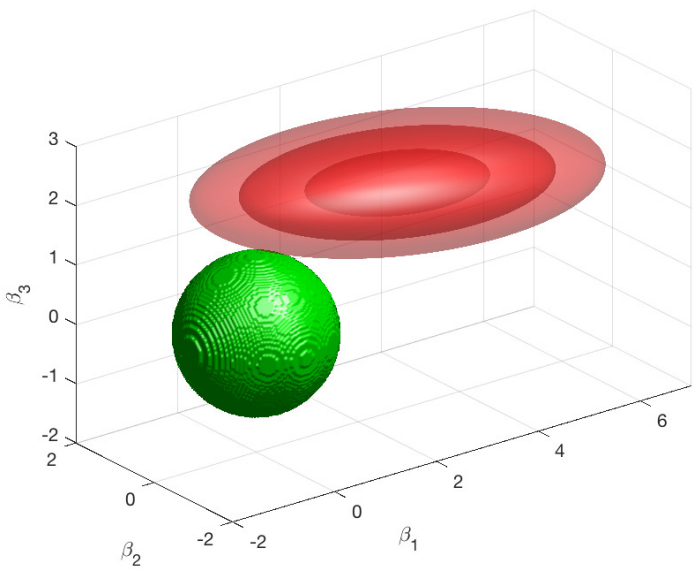




The Geometry of the Two Problems In Higher Dimension

See the MATLAB code attached:



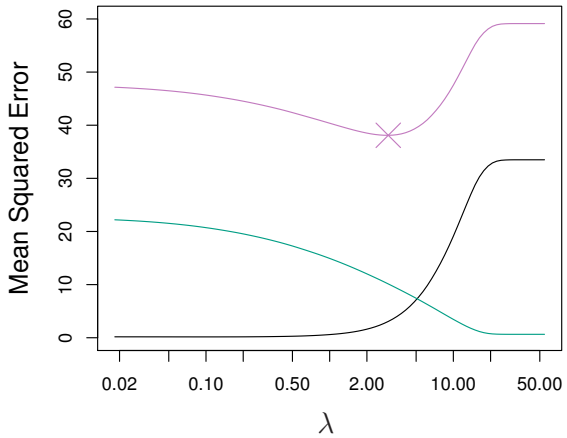


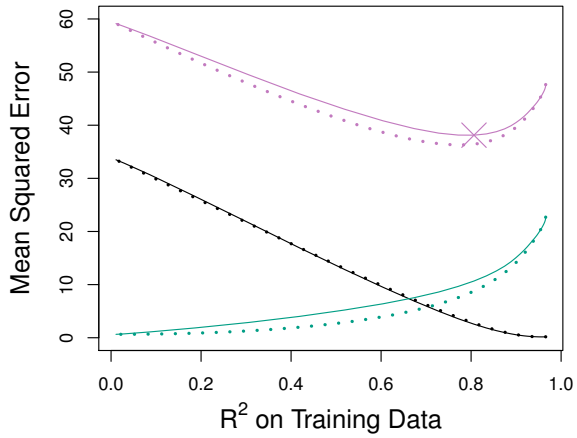
LASSO or Ridge?

- 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

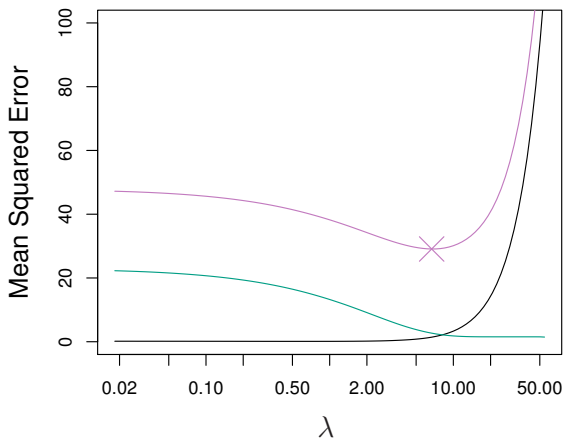
Example

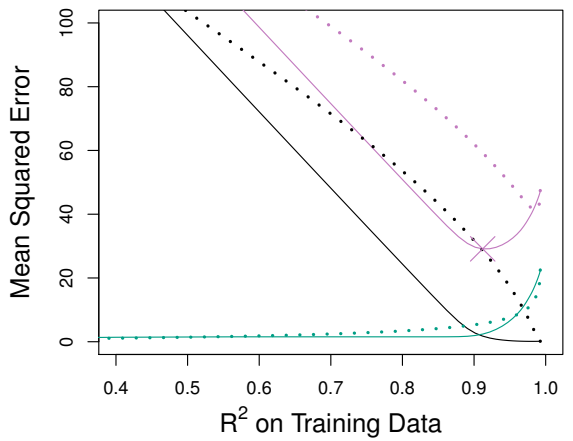
Previous example using all features (LASSO:Solid, Ridge: Dashed):





Using only two features:



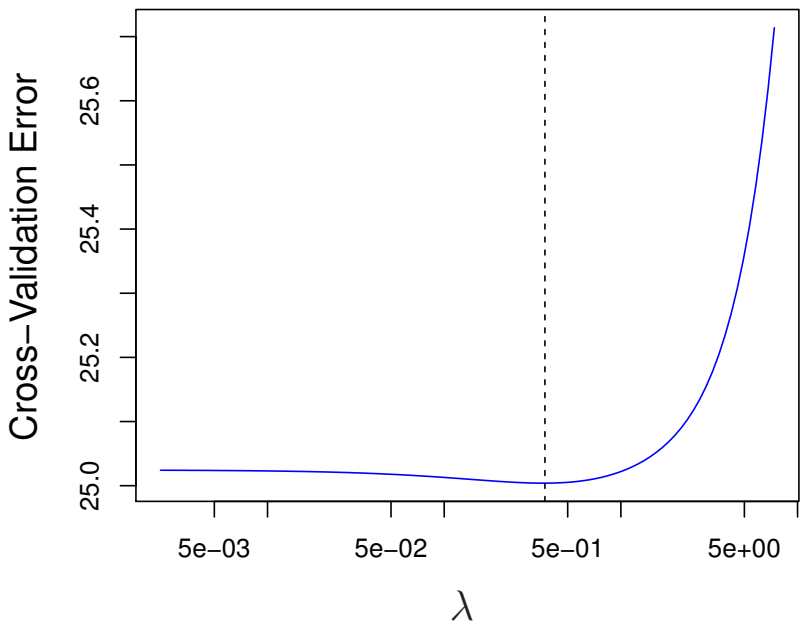


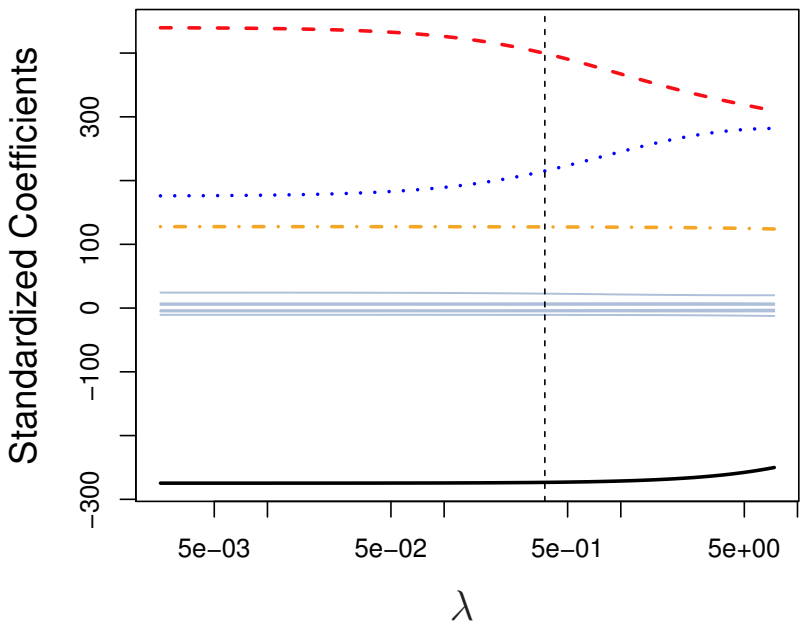
How to Determine λ?

- As for subset selection, for ridge regression and lasso we require a method to determine which of the models under consideration is the best
- That is, we require a method selecting a value for the tuning parameter λ
- 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 the smallest
- Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter

CV and Ridge Example

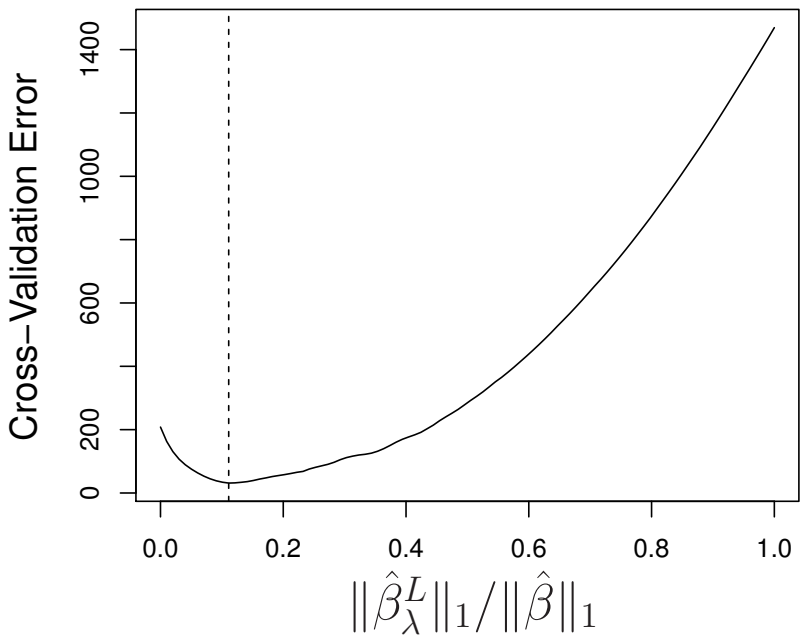
Determining λ via cross validation for the Ridge problem (credit data)

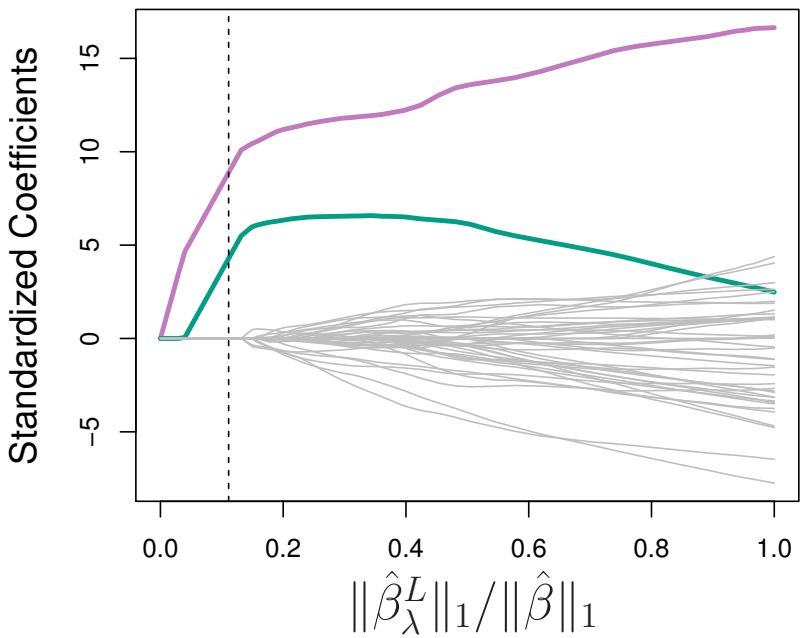




CV and LASSO Example

Determining λ via cross validation for the LASSO problem (simulated data)





Some Reviews of Linear Algebra

- Two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are called orthogonal if $\boldsymbol{u}^\top \boldsymbol{v} = 0$ (or equivalently $\boldsymbol{v}^\top \boldsymbol{u} = 0$)
- Given a vector $\boldsymbol{u}$, we have $\boldsymbol{u}^\top \boldsymbol{u} = \|\boldsymbol{u}\|^2$
- Example of two orthogonal vectors $\boldsymbol{u} = (1, 2, -3)^\top$ and $\boldsymbol{v} = (1, 1, 1)^\top$
- Matrices with orthogonal columns are very desirable and easy to work with
- Suppose $\boldsymbol{X} = \begin{pmatrix} | & | & | & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \\ | & | & | & | \end{pmatrix}$ such that $\boldsymbol{x}_i^\top \boldsymbol{x}_j = 0$ for $i \neq j$ and $\|\boldsymbol{x}_i\| = 1$, then $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}$
- If $\boldsymbol{X}$ is square and meets the conditions above, we call it an orthonormal matrix (or orthogonal if we do not have $\|\boldsymbol{x}_i\| = 1$)

Some Reviews of Linear Algebra

- Working with orthonormal matrices is very easy. For instance if $\boldsymbol{X}$ is orthonormal (it should be square) $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{I}$ and therefore $\boldsymbol{X}^{-1} = \boldsymbol{X}^\top$ **(lets generate an orthonormal matrix and verify this)**
- If $\boldsymbol{X}$ has orthonormal columns but is not square, then we still have $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}$ but this time $\boldsymbol{X}\boldsymbol{X}^\top \neq \boldsymbol{I}$ **(lets try this with a subset of the columns in the example above)**

Some Reviews of Linear Algebra

– In general it is very desirable to have matrices that have orthogonal columns. Converting matrices with orthogonal columns to matrices with orthonormal columns is easy, just divide each column by its ℓ_2 norm

$$\boldsymbol{X} = \begin{pmatrix} | & | & | & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \\ | & | & | & | \end{pmatrix} \text{ :orthogonal}$$

$$\therefore \hat{\boldsymbol{X}} = \begin{pmatrix} | & | & | & | \\ \frac{\boldsymbol{x}_1}{\|\boldsymbol{x}_1\|} & \frac{\boldsymbol{x}_2}{\|\boldsymbol{x}_2\|} & \cdots & \frac{\boldsymbol{x}_p}{\|\boldsymbol{x}_p\|} \\ | & | & | & | \end{pmatrix} \text{ :orthonormal}$$

– This transformation can also be done as follows

$$\hat{\boldsymbol{X}} = \boldsymbol{X}\text{diag}\left(\frac{1}{\|\boldsymbol{x}_1\|}, \frac{1}{\|\boldsymbol{x}_2\|}, \cdots, \frac{1}{\|\boldsymbol{x}_p\|}\right)$$

Some Reviews of Linear Algebra (SVD)

- **Singular Value Decomposition (SVD)** is a very powerful decomposition in linear algebra which enables decomposing any $n \times m$ matrix into the product and orthonormal matrices (and a diagonal matrix in the middle)
- *Any given matrix* $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ *can be decomposed as*

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$$

where $\boldsymbol{U} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{V} \in \mathbb{R}^{m \times m}$ are **orthonormal** matrices (i.e., $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}$ and $\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}$) and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \\ \cdots & \mathbf{0} & \cdots \\ \cdots & \vdots & \cdots \\ \cdots & \mathbf{0} & \cdots \end{bmatrix} \quad \text{or} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & \vdots & \vdots & \vdots \\ & \ddots & \mathbf{0} & \cdots & \mathbf{0} \\ & & \sigma_n & \vdots & \vdots & \vdots \end{bmatrix}$$

depending on whether $n > m$ or $m > n$

SVD Alternative Formulations

– Given a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ (say $n > m$) we can decompose it as

$$\boldsymbol{X} = \boldsymbol{U}_{n \times m} \boldsymbol{\Sigma}_{m \times m} \boldsymbol{V}^{\top}_{m \times m}$$

where $\boldsymbol{U}^{\top}\boldsymbol{U} = \boldsymbol{I}$, $\boldsymbol{V}^{\top}\boldsymbol{V} = \boldsymbol{I}$ and

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{bmatrix}$$

– Sum of rank-1 matrices:

$$\boldsymbol{X} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\top} + \cdots + \sigma_m \boldsymbol{u}_m \boldsymbol{v}_m^{\top}$$

such that $\boldsymbol{u}_i^{\top}\boldsymbol{u}_j = 0$ and $\boldsymbol{v}_i^{\top}\boldsymbol{v}_j = 0$ for $i \neq j$

SVD Alternative Formulations

– Sum of rank-1 matrices:

$$\boldsymbol{X} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \cdots + \sigma_m \boldsymbol{u}_m \boldsymbol{v}_m^\top$$

such that $\boldsymbol{u}_i^\top \boldsymbol{u}_j = 0$ and $\boldsymbol{v}_i^\top \boldsymbol{v}_j = 0$ for $i \neq j$

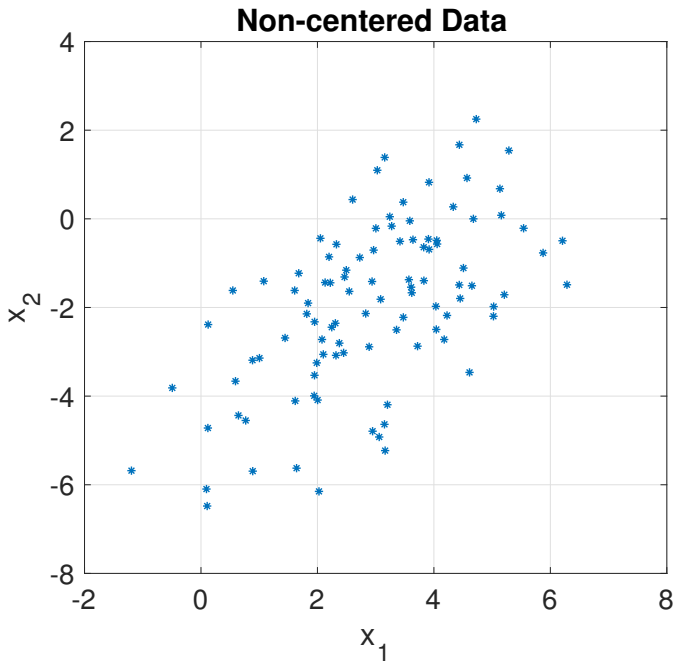
– Normally $\sigma_1, \sigma_2, \cdots$ are in descending order and omitting some of the terms would allow us to obtain a rank-r approximation to $\boldsymbol{X}$, i.e., $r < m$ then

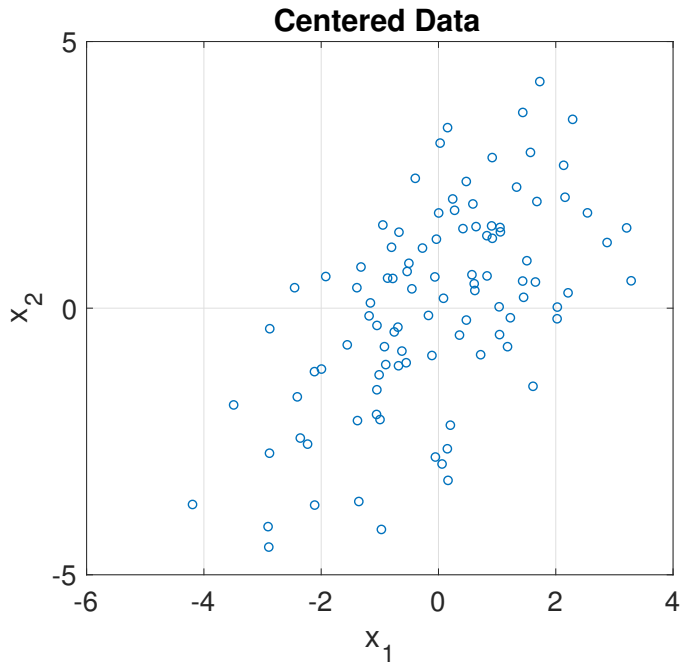
$$\boldsymbol{X} \approx \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^\top$$

(see the image approximation example in MATLAB)

Principal Component Regression (PCR)

- Consider $\boldsymbol{X}$ to be an $n \times p$ matrix representing our feature data (each column corresponds to a feature and each row to a sample)
- We assume that $\boldsymbol{X}$ is centered, meaning that each column is zero mean
- This technically means that our data cloud is centered around the origin and not some other point in the space





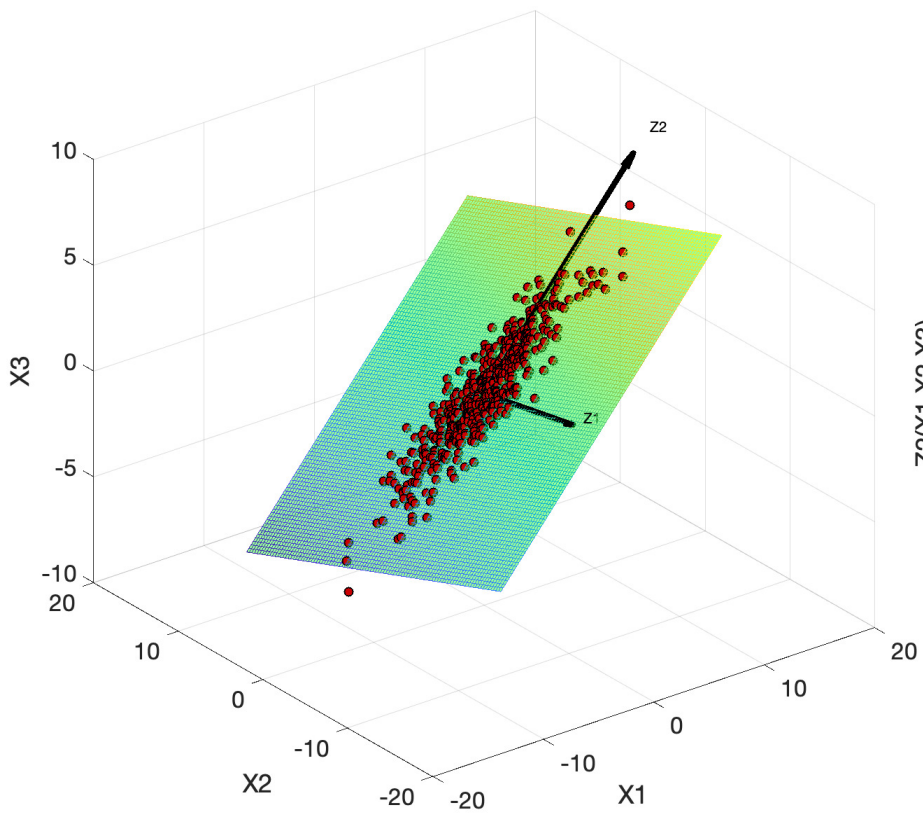
Principal Component Regression (PCR)

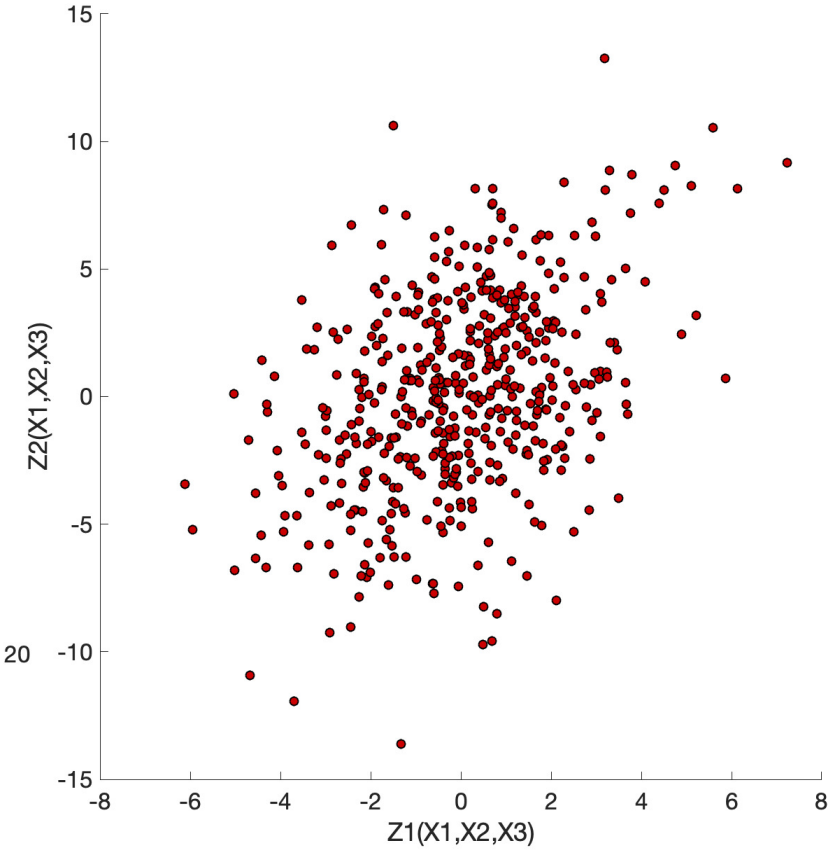
- Say we have three features x_1, x_2 and x_3 and we have an $n \times 3$ data matrix $\boldsymbol{X}$ (centered)
- If we look into the correlation matrix, the features could be correlated, e.g.,

$$\text{corr}(\boldsymbol{X}) = \begin{pmatrix} 1 & -0.3694 & 0.9907 \\ -0.3694 & 1 & -0.4926 \\ 0.9907 & -0.4926 & 1 \end{pmatrix}$$

- The data may or may not lie on a low dimensional manifold
- Is it possible to constructs new features from the original features (say $z_1 = c_{11}x_1 + c_{12}x_2 + c_{13}x_3$ and $z_2 = c_{21}x_1 + c_{22}x_2 + c_{23}x_3$) which are uncorrelated?
- **See the MATLAB DimensionalityReduction code**

Principal Component Regression (PCR)





Principal Component Regression (PCR)

- Lets see what we want to do mathematically:

$$\boldsymbol{X} = \begin{pmatrix} | & | & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 \\ | & | & | \end{pmatrix}$$

  We assume that $\boldsymbol{X}$ is normalized and centered then

$$\text{corr}(\boldsymbol{X}) = \boldsymbol{X}^\top \boldsymbol{X} = \begin{pmatrix} 1 & -0.3694 & 0.9907 \\ -0.3694 & 1 & -0.4926 \\ 0.9907 & -0.4926 & 1 \end{pmatrix}$$

- We want to construct new features z_1, z_2 such that $z_1 = c_{11}x_1 + c_{12}x_2 + c_{13}x_3$, and $z_2 = c_{21}x_1 + c_{22}x_2 + c_{23}x_3$ and are uncorrelated, that is:

$$\boldsymbol{Z}_{n\times 2} = \boldsymbol{X}_{n\times 3} \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{pmatrix}$$

  such that $\text{corr}(\boldsymbol{Z}) = \boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{I}$

Principal Component Regression (PCR) when $r = p$

- Based on what we learned from SVD, finding such matrix $\boldsymbol{C}$ is not hard
- In general we have a centered and normalized feature matrix $\boldsymbol{X}_{n\times p}$ where $\text{corr}(\boldsymbol{X}) \neq \boldsymbol{I}$ and we want to construct a new feature matrix $\boldsymbol{Z}_{n\times p}$ such that

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{C}, \quad \text{and} \quad \boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{I}$$

- Say $n > p$, and $\text{rank}(\boldsymbol{X}) = p$ then by doing SVD we have

$$\boldsymbol{X} = \boldsymbol{U}_{n\times p}\boldsymbol{\Sigma}_{p\times p}\boldsymbol{V}^\top_{p\times p}$$

  then simple calculation reveals that $\boldsymbol{C} = \boldsymbol{V}\boldsymbol{\Sigma}^{-1}$ and as a result $\boldsymbol{Z} = \boldsymbol{U}$

Principal Component Regression (PCR) when $r < p$

- Based on what we learned from SVD, finding such matrix $\boldsymbol{C}$ is not hard
- In general we have a centered and normalized feature matrix $\boldsymbol{X}_{n\times p}$ where $\text{corr}(\boldsymbol{X}) \neq \boldsymbol{I}$ and we want to construct a new feature matrix $\boldsymbol{Z}_{n\times p}$ such that

$$\boldsymbol{Z} = \boldsymbol{X}\boldsymbol{C}, \quad \text{and} \quad \boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{I}$$

- Say $n > p$, and $\text{rank}(\boldsymbol{X}) > r$ then by doing SVD we have

$$\boldsymbol{X} = \boldsymbol{U}_{n\times p}\boldsymbol{\Sigma}_{p\times p}\boldsymbol{V}^\top_{p\times p}$$

and therefore

$$\boldsymbol{X} \approx \boldsymbol{U}_{n\times r}\boldsymbol{\Sigma}_{r\times r}\boldsymbol{V}^\top_{p\times r}$$

then as before we can see that setting $\boldsymbol{C} = \boldsymbol{V}_{p\times r}\boldsymbol{\Sigma}^{-1}_{r\times r}$ and $\boldsymbol{Z} = \boldsymbol{U}_{n\times r}$ would yield $\boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{I}$ and $\boldsymbol{Z} \approx \boldsymbol{X}\boldsymbol{C}$

Principal Component Regression (PCR) when $r < p$

- Lets make the world simple: you are given the standardized data matrix $\boldsymbol{X}$ which is correlated, all you need to do to generate an uncorrelated data matrix is to do SVD and pick the r first columns, i.e.,

$$\boldsymbol{Z} = \boldsymbol{U}_{n \times r}$$

- Now if we have a response variable $\boldsymbol{y} \in \mathbb{R}^n$ we can use the $\boldsymbol{z}$ features to do the regression

Principal Component Regression (PCR)

- Recall the singular value decomposition (SVD)

$$\boldsymbol{X} = \boldsymbol{U}_{n\times p}\boldsymbol{\Sigma}_{p\times p}\boldsymbol{V}^{\top}_{p\times p}$$

  where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_p \end{bmatrix}$$

- When the first r entries of $\boldsymbol{\Sigma}$ are the dominant terms, the first r columns in $\boldsymbol{U}$ form a subspace (coordinate system) that very well represent our data
- In PCR, we move from the original feature space to the new feature space represented by $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_r$
- Technically, instead of a fit like

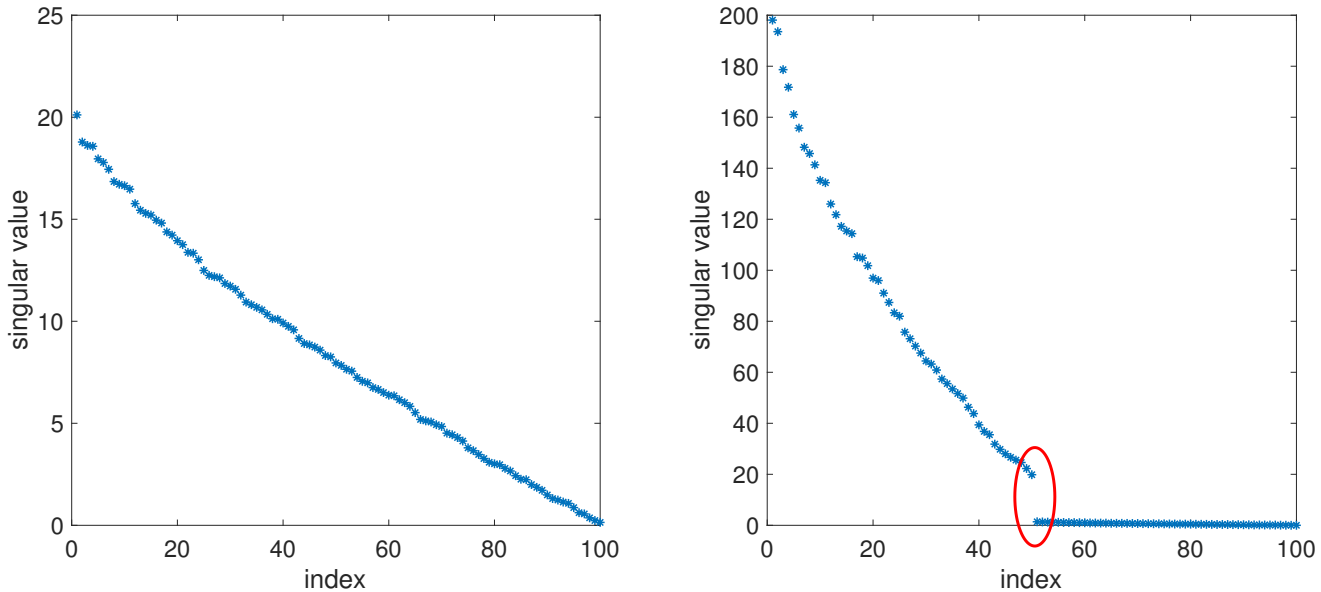
$$\boldsymbol{y} = \beta_0 + \beta_1\boldsymbol{x}_1 + \cdots \beta_p\boldsymbol{x}_p$$

  we fit the model

$$\boldsymbol{y} = \gamma_0 + \gamma_1\boldsymbol{z}_1 + \cdots \gamma_r\boldsymbol{z}_r$$

How to Determine r

- Normally, the number of principal components, r, can be determined by cross validation
- If we see a good contrast between two consecutive singular values, roughly choosing r (or at least an initial value) can become easy!



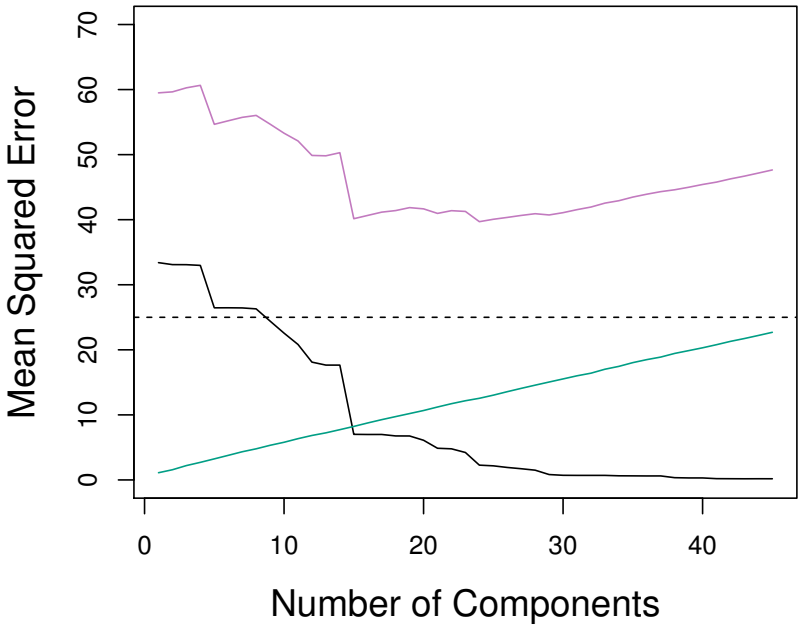
- Despite all this, a formal selection requires CV
- This is technically because such selection only relies on $\boldsymbol{X}$, and no guarantee that the selected features contribute to the response

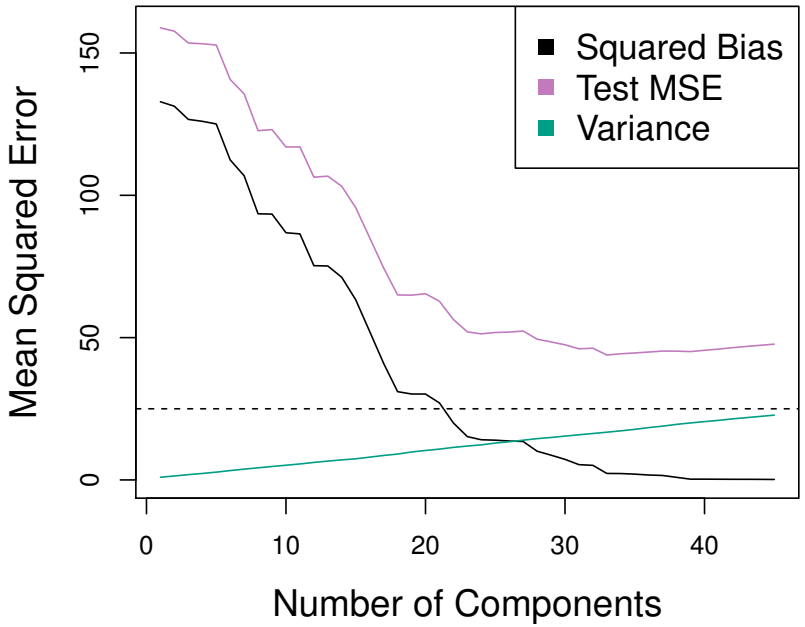
More on PCR

- PCR identifies linear combinations, or directions, that best represent the predictors $x_1, \cdots, x_p$
- 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** (PLS, details of which are skipped in this course) can be considered as a remedy to this

Example

Example Running CV and PCR on two different data sets (left: $r \approx 14$ could be a good selection; right: CV suggests using almost all features)





Because PCR does not take into account the relation to the response, despite the possibility of expressing the response with only two features on the right panel, PCR suggests almost using all features

Let's conclude by some coding examples!

References

- https://www.alsharif.info/iom530, 2013.
- J. Friedman, T. Hastie, and R. Tibshirani.
  The elements of statistical learning.
  Springer series in statistics, 2nd edition, 2009.
- G. James, D. Witten, T. Hastie, and R. Tibshirani.
  https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/model_selection.pdf, 2013.
- G. James, D. Witten, T. Hastie, and R. Tibshirani.
  An introduction to statistical learning: with applications in R, volume 112.
  Springer, 2013.