SDSC 5001: Statistical Machine Learning I

Topic 5. Model Selection and Regularization

Sparse Regression

- Given training set $(\mathbf{x}_i, y_i)_{i=1}^{n}$, with $y_i \in R$ and $\mathbf{x}_i \in R^p$, it is assumed that

$$y_i = \beta_0 + \sum_{j=1}^{p_0} \beta_j x_{ij} + \epsilon_i$$

where $p_0 \ll p$ (**sparsity**).

- $A^* = \{1, \dots, p_0\}$ indexes the informative predictors, and $\{p_0 + 1, \dots, p\}$ indexes the redundant predictors.
- The goal of variable selection is to correctly detect A^* from $\{1, \dots, p\}$.
- We focus on linear regression models, while detecting nonlinear relationship is possible and largely open.

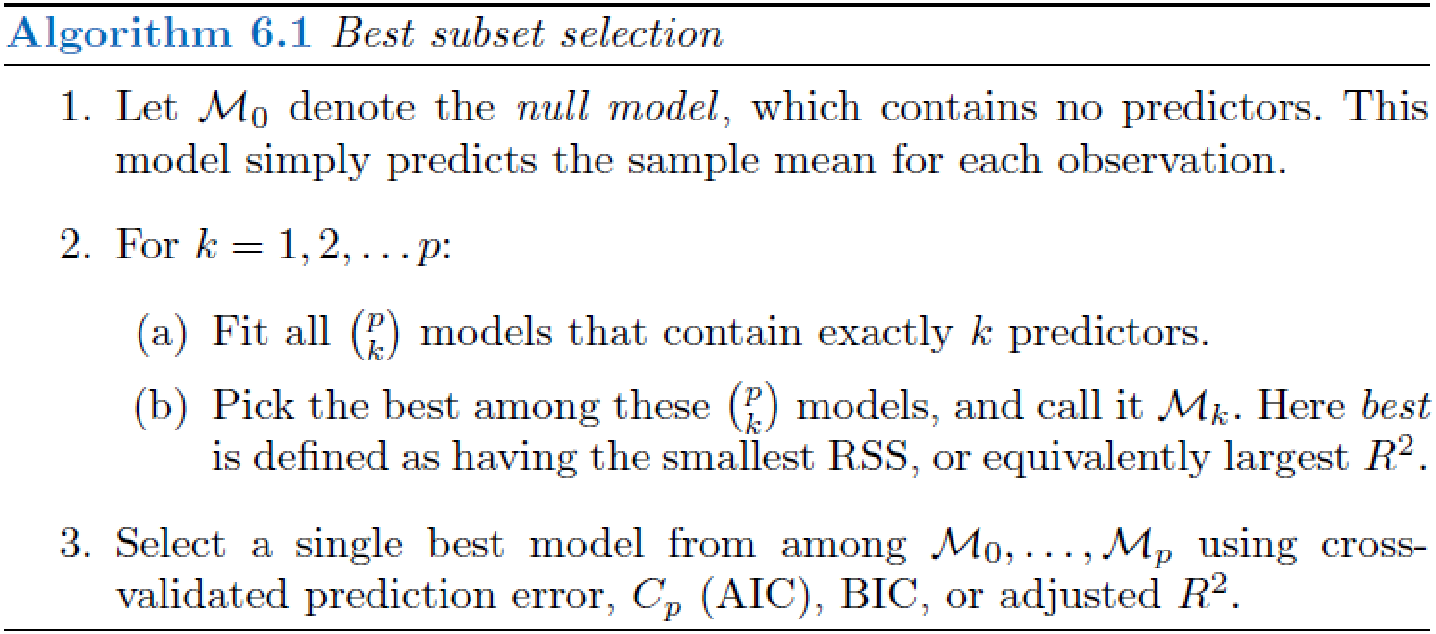
Why Do We Care?

- Multicollinearity: masked significance, inflated variance,…
- Prediction accuracy can be deteriorated due to overfitting when p is large (*curse of dimensionality*).
- Interpretability can be unnecessarily complicated when irrelevant variables are included.

Popular Techniques

- Best subset selection
  - Various information criteria, cross validation
- Sequential variable selection
  - Forward/backward selection
- Shrinkage methods
  - Lasso and its variants
- Dimension reduction
  - Principal component analysis, sufficient dimension reduction

Best Subset Selection

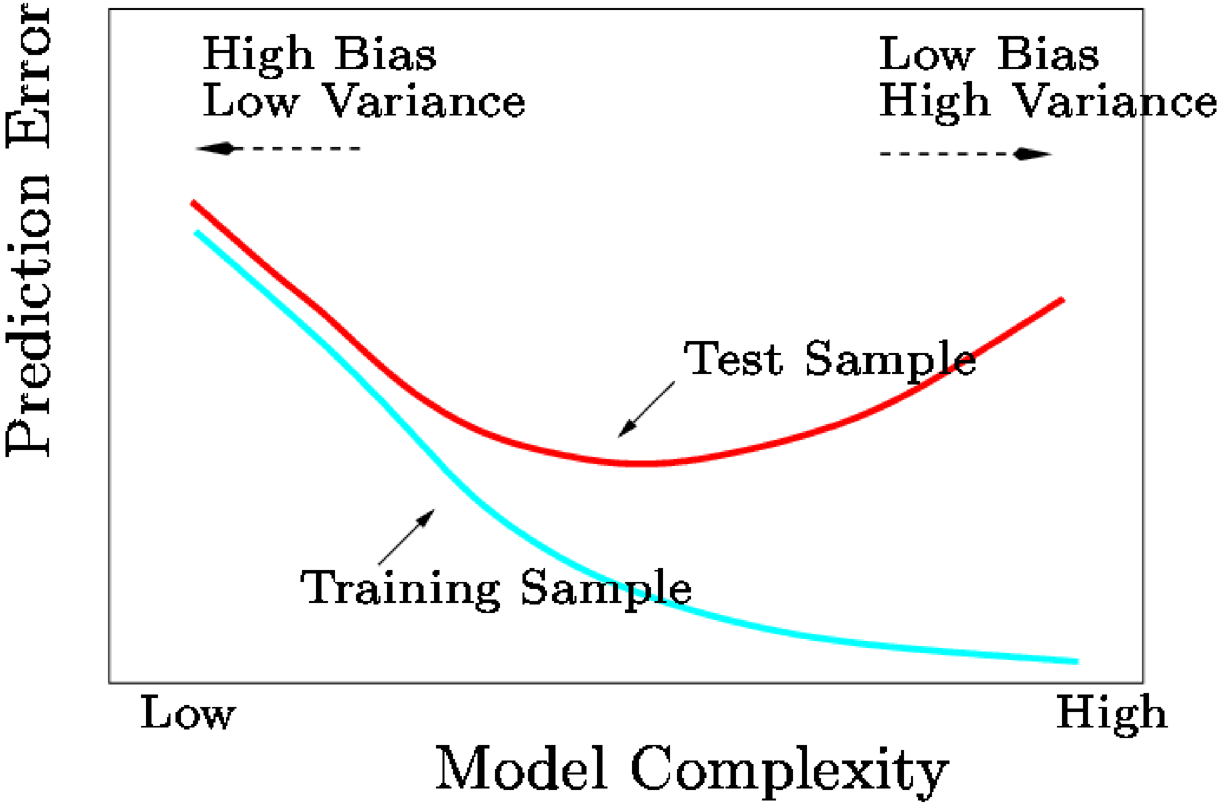


Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ 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.

- Popular selection criteria
  - Validation set
  - Cross validation
  - “Estimate” test error by making an adjustment to the training error to account for overfitting

Adjustment to Training Error



Model Selection Criteria

- For a linear model with d predictors, denote its SSE as SSE_d,
- Mallow's C_p:

$$C_p = \frac{1}{n}(SSE_d + 2d\hat{\sigma}^2)$$

- Akaike information criterion (AIC)

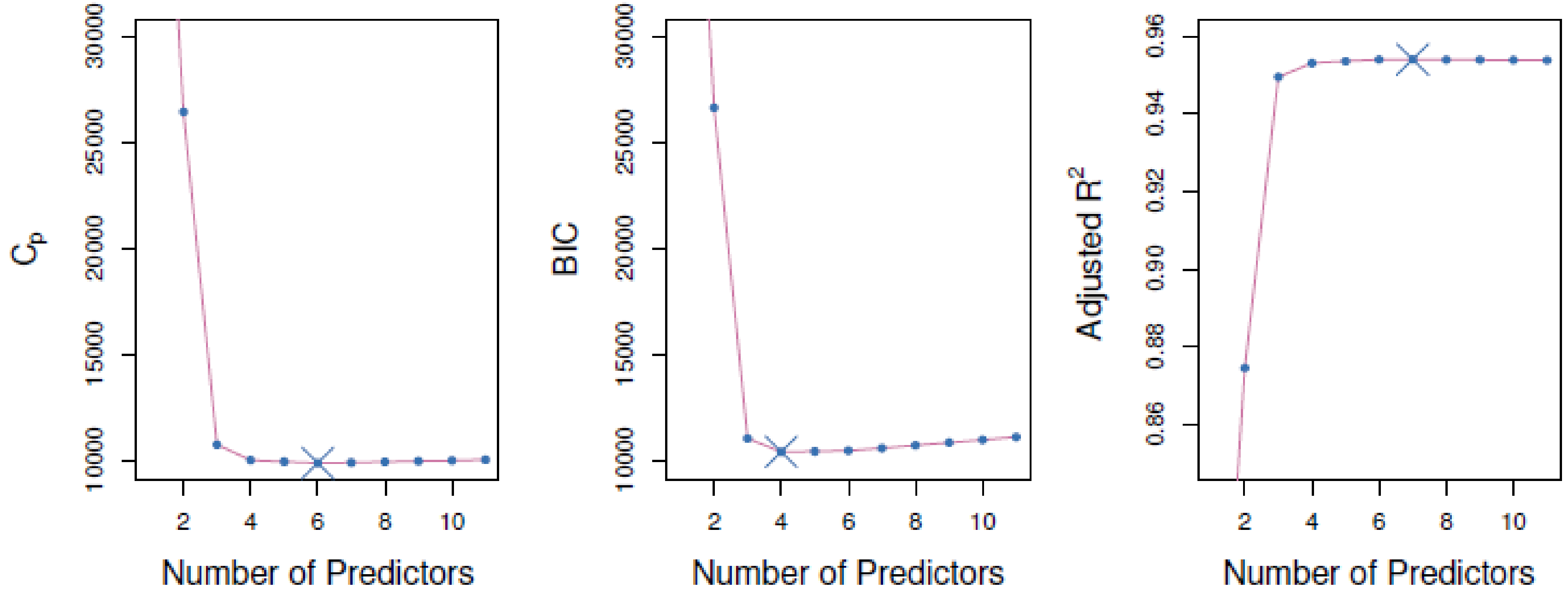
$$AIC = \frac{1}{n\hat{\sigma}^2}(SSE_d + 2d\hat{\sigma}^2)$$

- Bayesian information criterion (BIC)

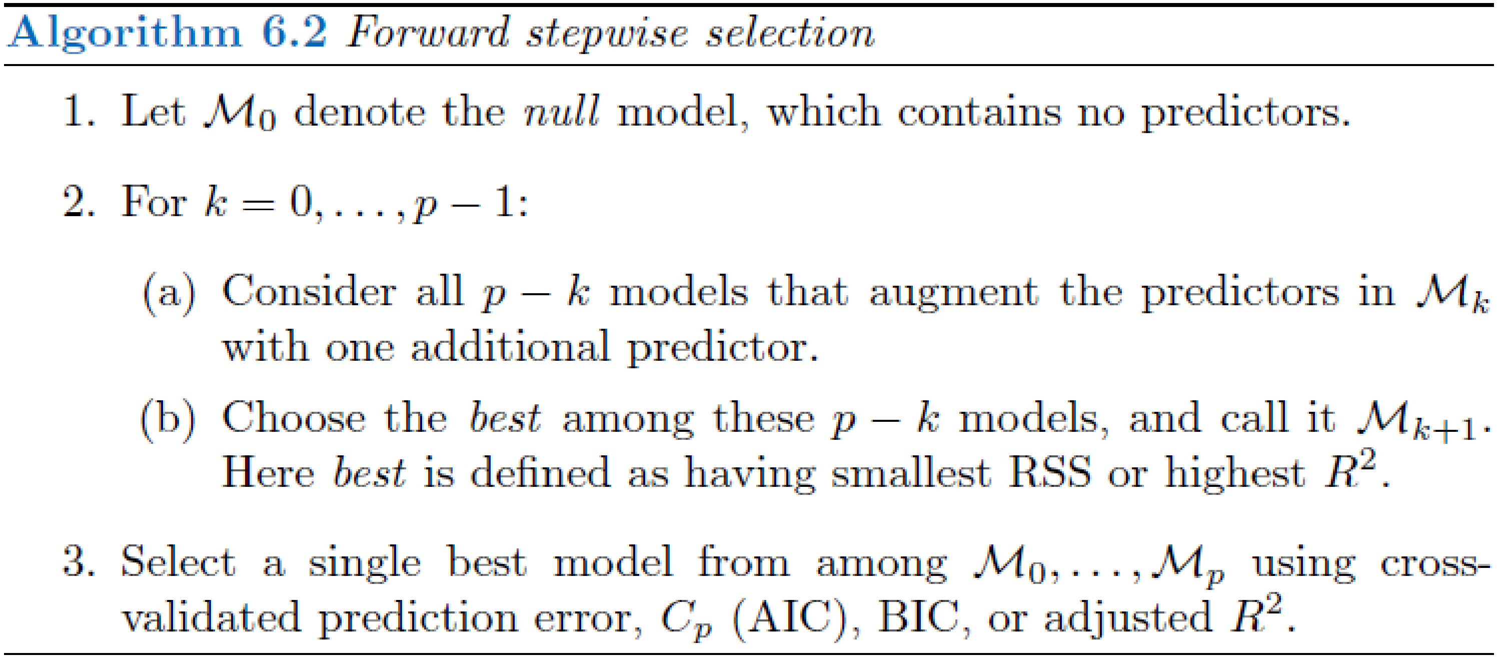
$$BIC = \frac{1}{n\hat{\sigma}^2}(SSE_d + \log(n)d\hat{\sigma}^2)$$

- Other criteria: Other IC's, adjusted R^2

An Illustrative Example



Forward/Backward Selection



Algorithm 6.2 *Forward stepwise selection*

```
1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
    (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
    (b) 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 selection starts with M_p and iteratively deletes predictors until the best model is found.
- Stagewise selection mixes forward addition and backward deletion in each iteration.

Some Remarks

- Forward/backward selection is computationally more efficient than subset selection.
- It has no guarantee of the best possible model.
- It usually performs well in practice.
- Forward versus backward selection

Shrinkage Methods

- Shrinkage methods are formulated as

$$(\hat{\beta}_0, \hat{\beta}\ ) = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \mathbf{x}_i^T \beta\right)^2 + \lambda J(\beta)$$

- Various choices of $J(\beta)$ lead to different shrinkage methods and possess different properties.

- After centralization, it becomes

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^T \beta\right)^2 + \lambda J(\beta)$$

Ridge Regression

- Ridge regression uses an L_2-norm penalty, $||\beta||_2^2 = \sum_{j=1}^{p} \beta_j^2 = \beta^T \beta$,

$$\hat{\beta}_{\lambda}^{ridge} = \underset{\beta}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda ||\beta||_2^2$$

- The second term $\lambda||\beta||_2^2$ is a shrinkage penalty, which shrinks the estimates of β towards zero.

- The tuning parameter $\lambda > 0$ controls the trade-off between regression fitting and coefficient shrinkage.

- If $\lambda = 0$, ridge regression produces LSE; if $\lambda \to \infty$, the estimates of β will approach zero.

Ridge Regression (Cont.)

- Solution of the ridge regression is

$$\hat{\beta}_{\lambda}^{ridge} = \left(\mathbf{X}^T\mathbf{X} + \lambda I_p\right)^{-1}\mathbf{X}^T\mathbf{y}$$

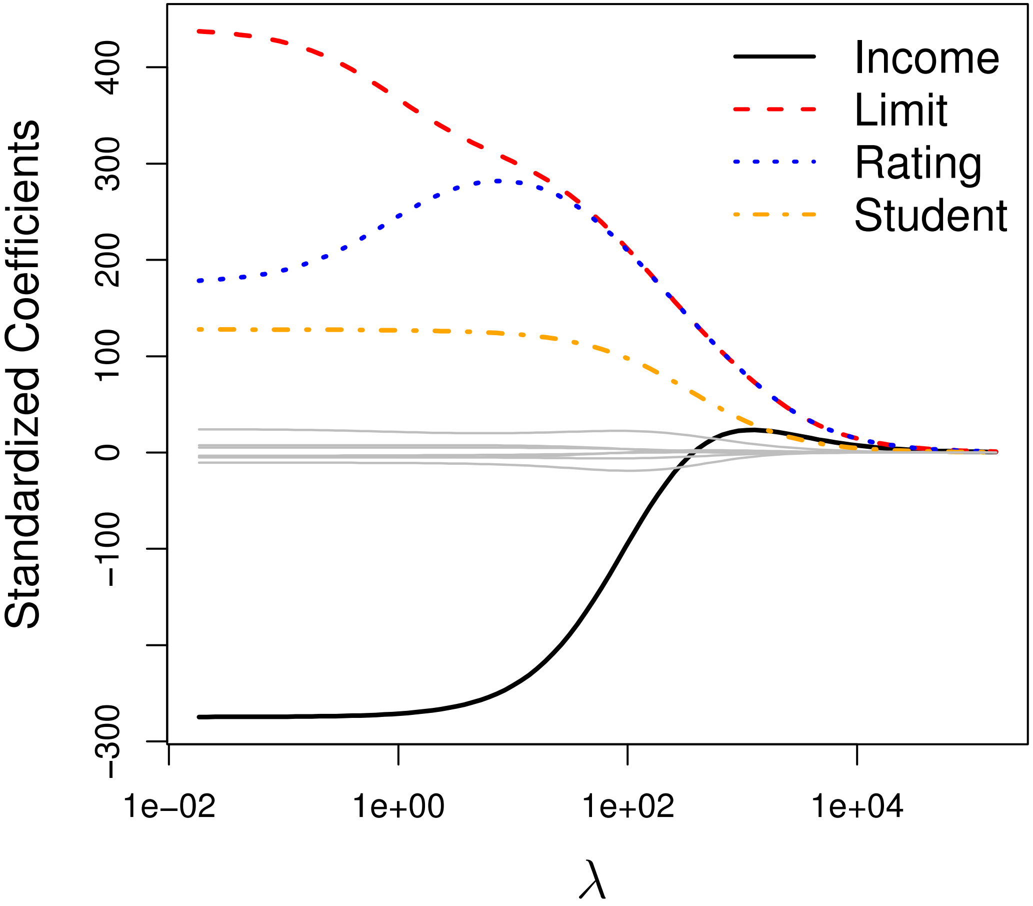
- An equivalent formulation

$$\hat{\beta}_{\lambda}^{ridge} = \underset{\beta}{\operatorname{argmin}}(\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$$

subject to $||\beta||^2 \leq s$

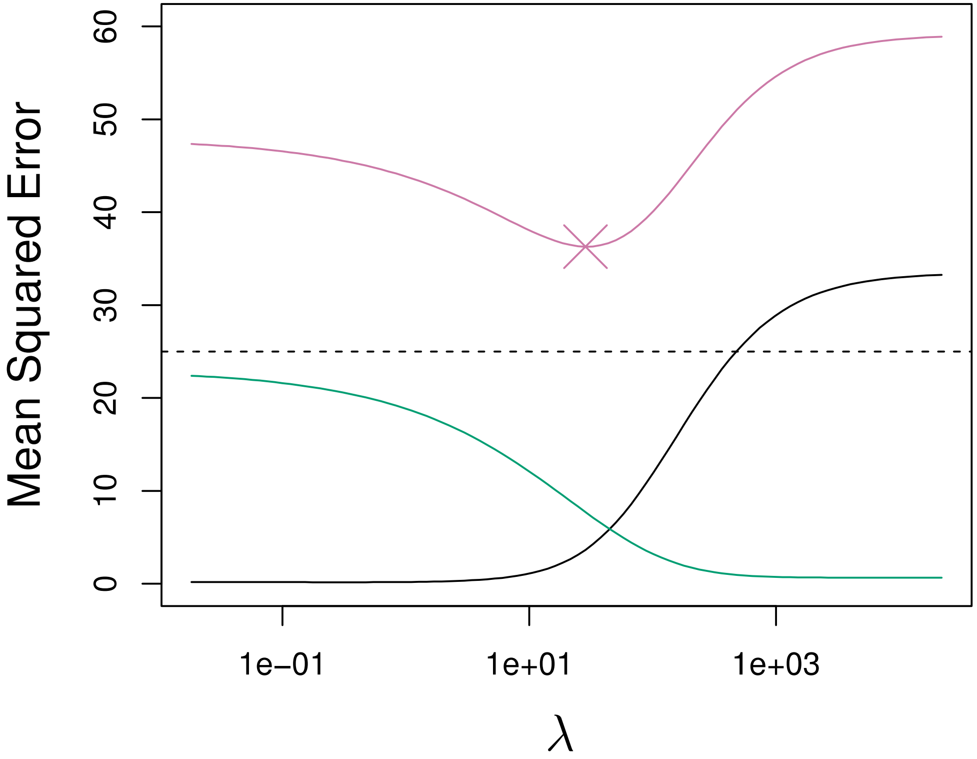
Example

- In general, $\hat{\beta}_\lambda$ is a biased estimator that may have smaller MSE than the LSE estimator.



An Example (Cont.)

- Black: Bias
- Green: Variance
- Purple: Test MSE
- Increase in λ increases bias but decreases variance



LASSO

- The lasso uses an L_1-norm penalty, $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$,

$$\hat{\beta}_\lambda^{lasso} = \underset{\beta}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) + \lambda ||\beta||_1$$

- Or equivalently,

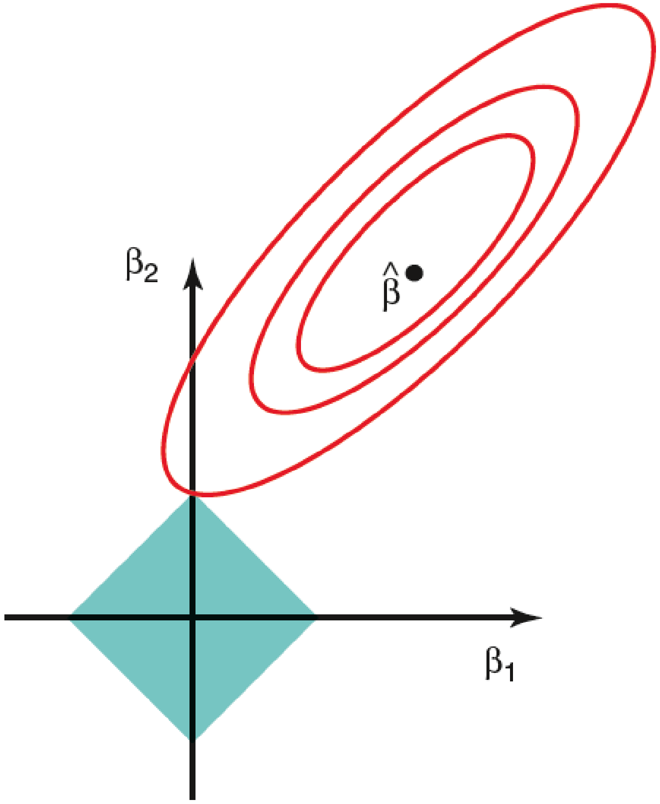
$$\hat{\beta}_\lambda^{lasso} = \underset{\beta}{\operatorname{argmin}} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

$$\text{subject to} \quad ||\beta||_1 \leq s$$

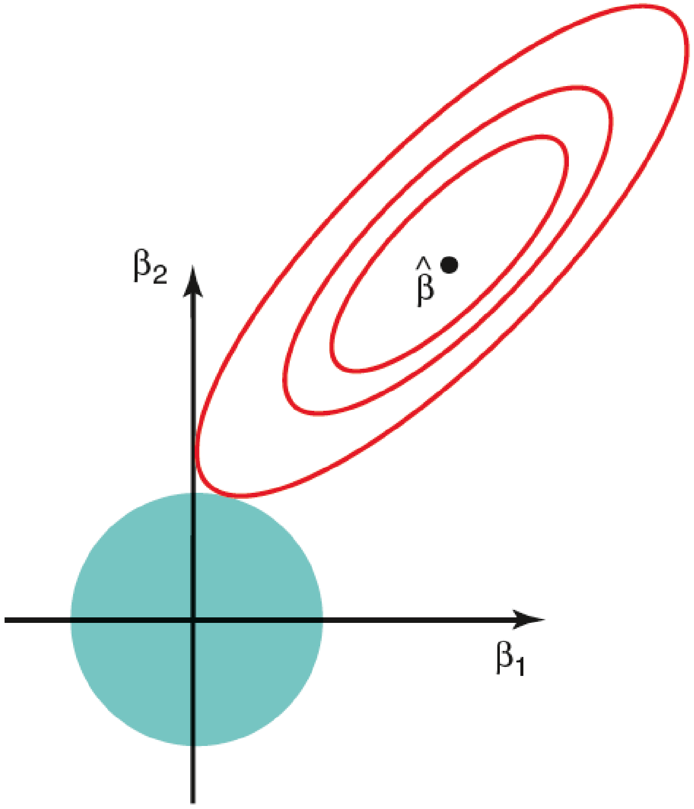
- No explicit solution in general, and a quadratic programming (QP) algorithm can be used to solve the optimization problem.

Sparse Solution

- Some coefficients of the lasso solution will become exactly zero, and thus it does some kind of continuous variable selection.



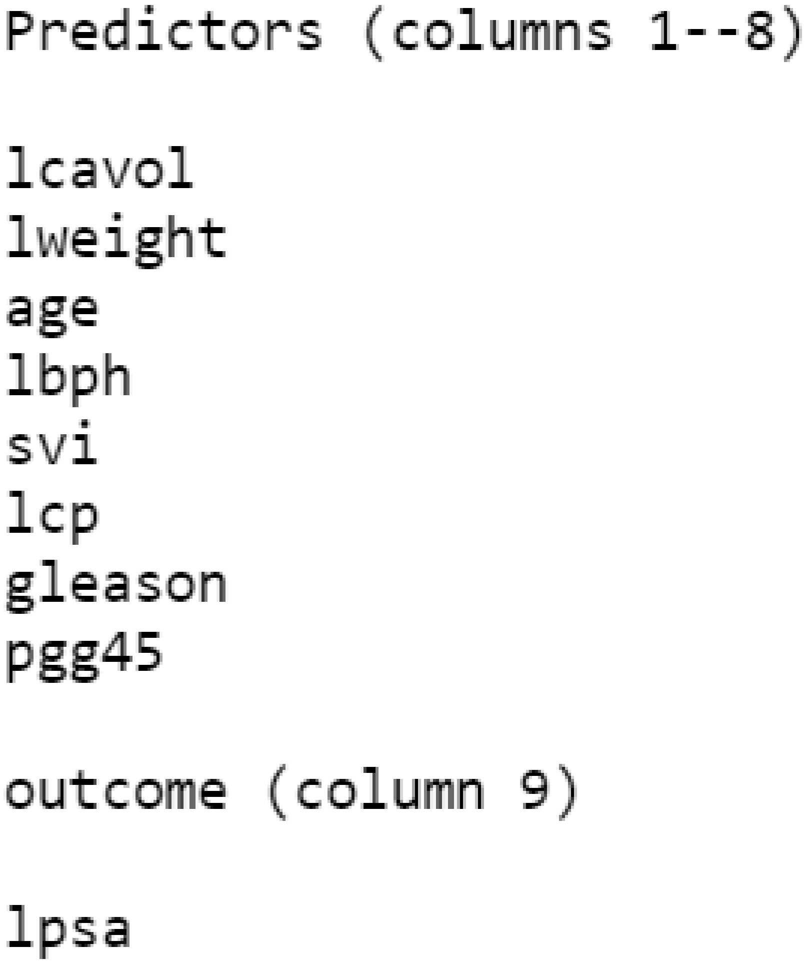
$|\beta_1| + |\beta_2| \leq s$



$\beta_1^2 + \beta_2^2 \leq s$

Example: Prostate Cancer

- Data



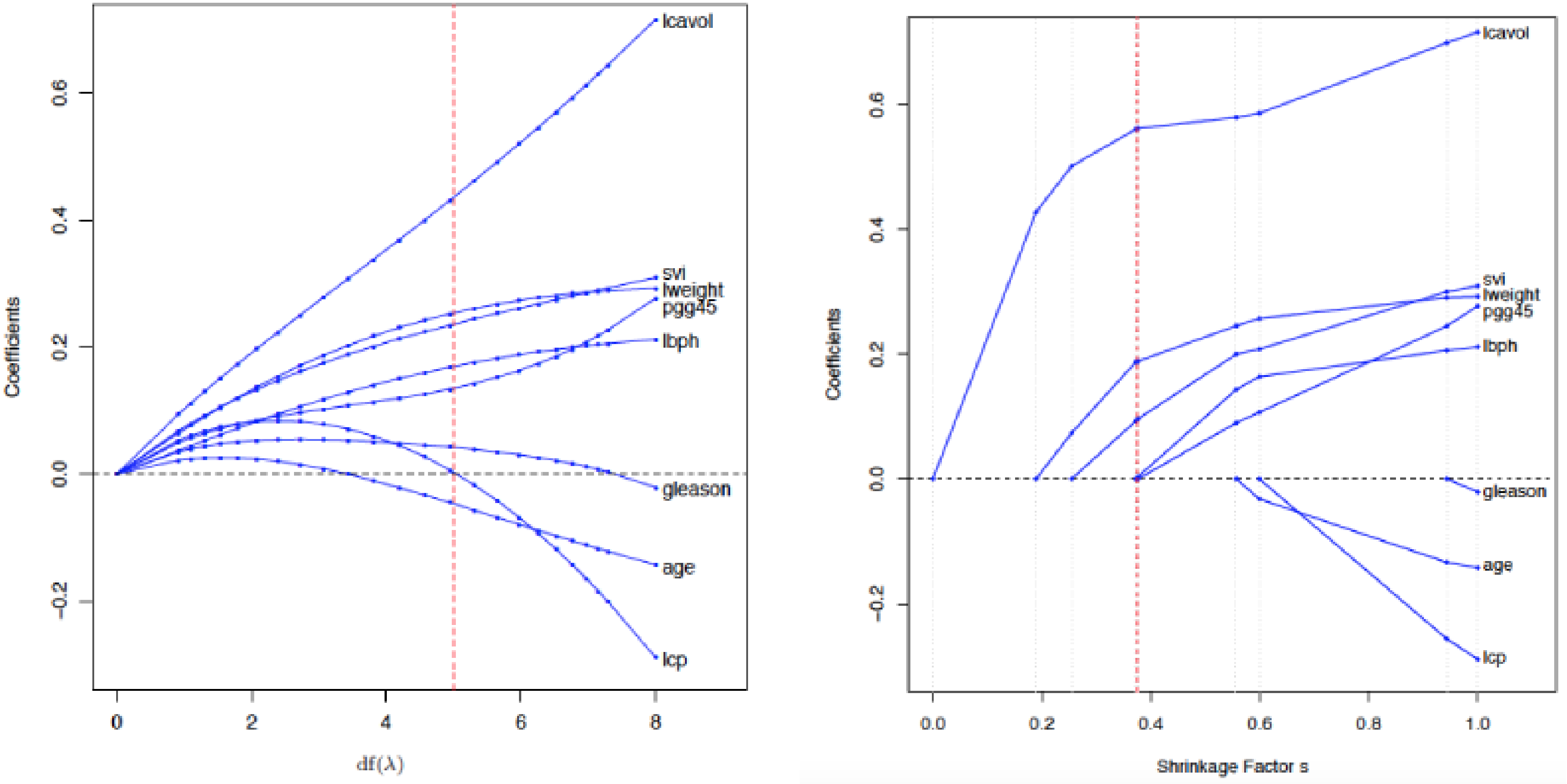
```
Predictors (columns 1--8)

lcavol
lweight
age
lbph
svi
lcp
gleason
pgg45

outcome (column 9)

lpsa
```

Clinical measures

Level of prostate-specific antigen

Example: Prostate Cancer (Cont.)

- Left: ridge regression; Right: lasso

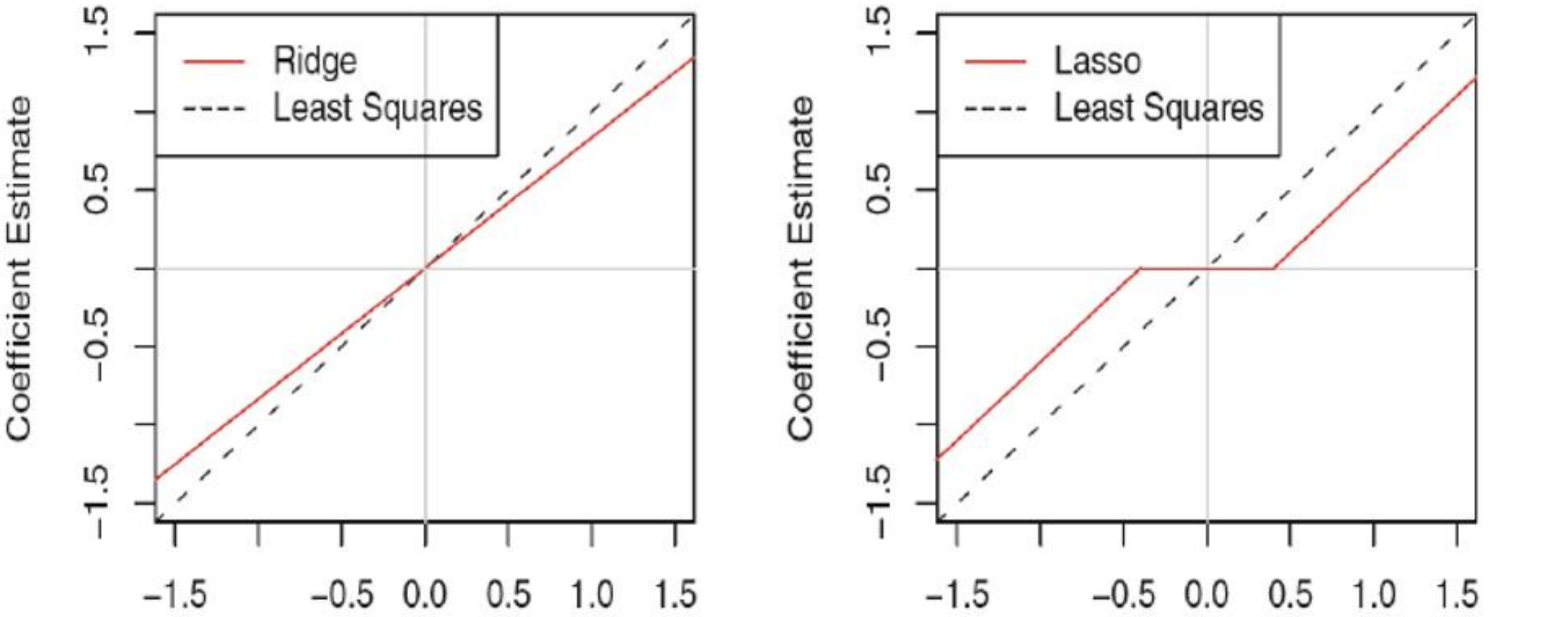


Ridge vs. Lasso

- Both lasso and ridge regression will shrink estimated coefficients while introducing some bias.
- The lasso produces simpler and more interpretable models that involve only a subset of predictors.
- It is unclear which one leads to better prediction accuracy in general though.

A Special Simple Case

- Consider a simple case with $n = p$ and $\mathbf{X} = \mathbf{I}_p$, then $\hat{\beta}_j^{ols} = y_j$,
  - Ridge regression multiplies $\hat{\beta}_j^{ols}$ by a constant, $\hat{\beta}_j^{ridge} = y_j/(1+\lambda)$.
  - Lasso truncates $\hat{\beta}_j^{ols}$ towards zero by a constant, $\hat{\beta}_j^{lasso} = sign(y_j)\big(|y_j| - \lambda/2\big)_+$.



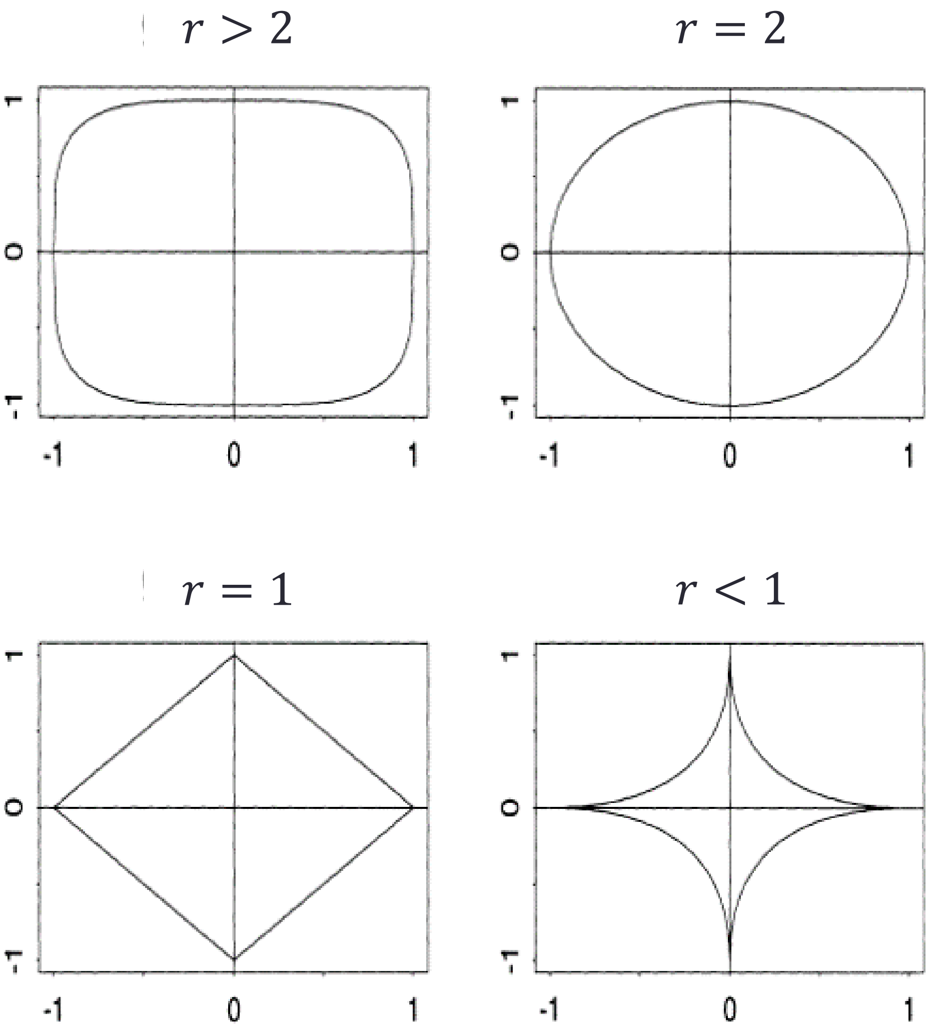
Bridge Estimators

- With $L_r(\beta) = \sum_{j=1}^{p} \left|\beta_j\right|^r$,

$$\hat{\beta}_{\lambda}^{bridge} = \underset{\beta}{\operatorname{argmin}} ||\mathbf{y} - \mathbf{X}\beta||^2 + \lambda L_r(\beta)$$

- $L_0(\beta) = \sum_{j=1}^{p} I(\beta_j \neq 0)$ (Hard thresholding)
- $L_1(\beta) = \sum_{j=1}^{p} \left|\beta_j\right|$ (Lasso)
- $L_2(\beta) = \sum_{j=1}^{p} \beta_j^2$ (Ridge regression)
- $L_\infty(\beta) = max_j\ \beta_j$

Constrained Areas of Bridge Regressions



$r > 2$

$r = 2$

$r = 1$

$r < 1$

Nonnegative Garrote

$$\min_{\boldsymbol{c}} \frac{1}{2} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} c_j \hat{\beta}_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} c_j$$

subject to $c_j \geq 0$, and then $\hat{\beta}_j^{ng} = \hat{c}_j \hat{\beta}_j$.

- The resulting estimator is

$$\hat{\beta}_j^{ng} = \left(1 - \frac{\lambda}{2\hat{\beta}_j^2} \right)_+ \hat{\beta}_j$$

- It is almost unbiased for large $|\hat{\beta}_j|$.
- It shrinks small $|\hat{\beta}_j|$ to zero.

Other Extensions

- Group lasso: if the p variables are partitioned into J groups, and then it is desirable to include or exclude the whole group

$$\min_{\beta} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=1}^{p} \|\vec{\beta}_j\|_2$$

where $\vec{\beta}_j$ is a coefficient vector for the jth group.

- Elastic net:

$$\min_{\beta} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2$$

- Fused lasso:

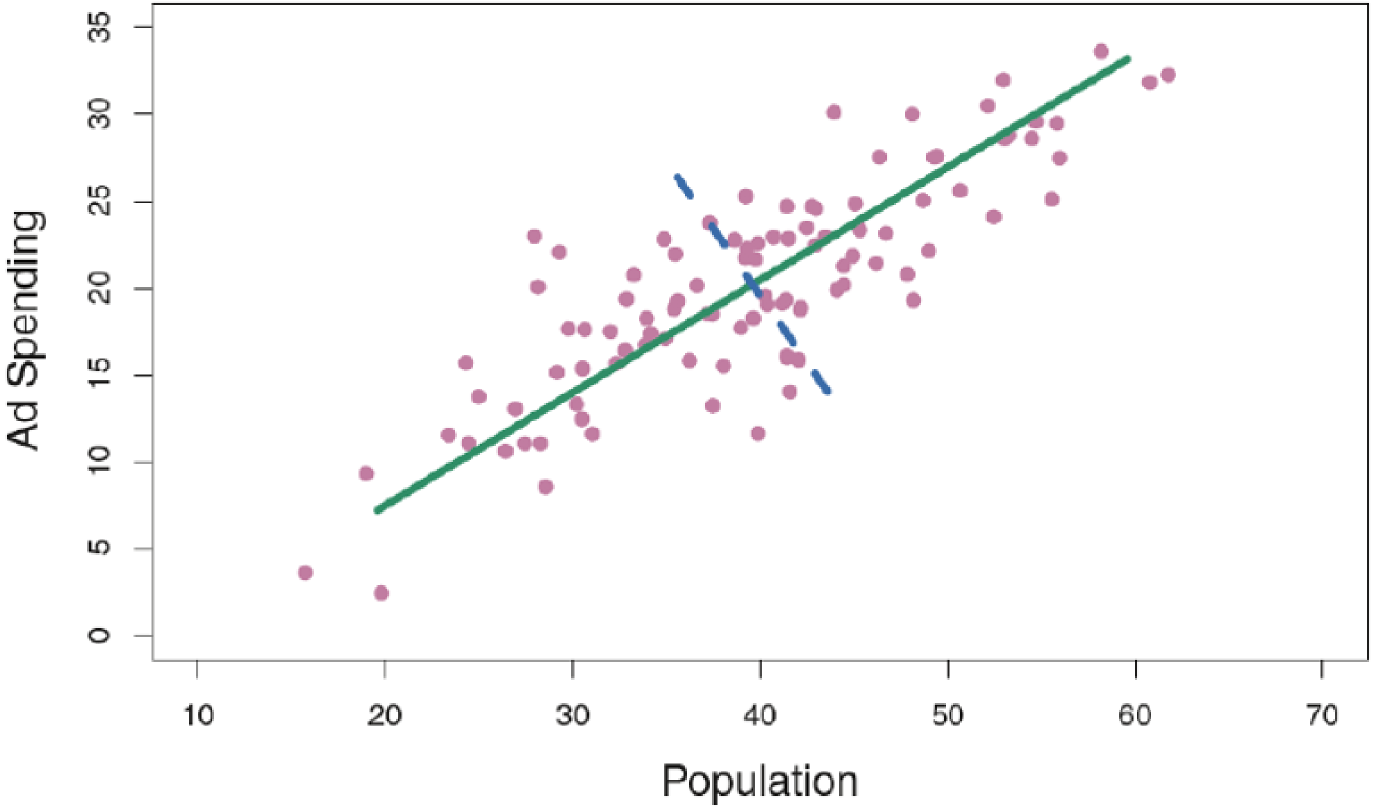
$$\min_{\beta} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=2}^{p} \|\beta_j - \beta_{j-1}\|_1$$

Principal Component Analysis (PCA)

- Given $X = \left(X_1, \ldots, X_p\right)^T$ and $\Sigma = \mathrm{cov}(X)$, find $\left\{a_1, \ldots, a_p\right\}$ with $\left\|a_j\right\| = 1, j = 1, \ldots, p$ such that
  - $\mathrm{var}\left(a_j^T X\right) = a_j^T \Sigma a_j$ is as large as possible, and
  - $\mathrm{cov}\left(a_j^T X, a_l^T X\right) = a_j^T \Sigma a_l = 0$ when $j \neq l$.

- In general,
  - First, find $a_1 = \underset{a}{\mathrm{argmax}}\, a^T \Sigma a$ subject to $||a|| = 1$,
  - Then find $a_k = \underset{a}{\mathrm{argmax}}\, a^T \Sigma a$ subject to $||a|| = 1$ and $a^T \Sigma a_j = 0$ for $j = 1, \ldots, k-1$

Example



PCA and Eigen-decomposition

- Assume the eigenvalues of Σ is $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, and the associated eigenvectors are $e_1, \dots, e_p$, then
  - $a_j = e_j$ and the j-th PC is $U_j = e_j^T X$
  - $\operatorname{var}(U_j) = e_j^T \Sigma e_j = \lambda_j$
  - $\operatorname{cov}(U_j, U_l) = e_j^T \Sigma e_l = 0$

- To reduce dimension, set $0 < \alpha < 1$ and choose $k \ll p$ such that

$$\frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_p} \geq \alpha$$

and then work on the feature space spanned by the first k PC's.

Projection Pursuit (PP)

- Imagine an example with $X_1 \sim N(0{,}100)$, and $X_2 \sim N(Z, 1/100)$ with $Z = \pm 1(\pm$½). PCA will find X_1 as the first PC, but it is less informative.

- The key idea of PP is to find direction which is non-normal.
  - Let $I(\cdot)$ be a measure of non-normality,

$$\hat{a} = \underset{a}{\operatorname{argmax}} \, I(a^T X)$$

  - Popularly used $I(\cdot)$:
    - $I_1(z) = |k_m(z)|/k_2(z)^{m/2}$
    - $I_2(z) = k_3^2(z) + k_4^2(z)/4$

    where $k_m(z)$ is the m-th order cumulant of z.

Projection Pursuit (PP)

Principal Component Regression (PCR)

- Let S be the sample covariance matrix and $\mathbf{q}_j$, $j = 1, \ldots, J$, be the PC loadings of S.

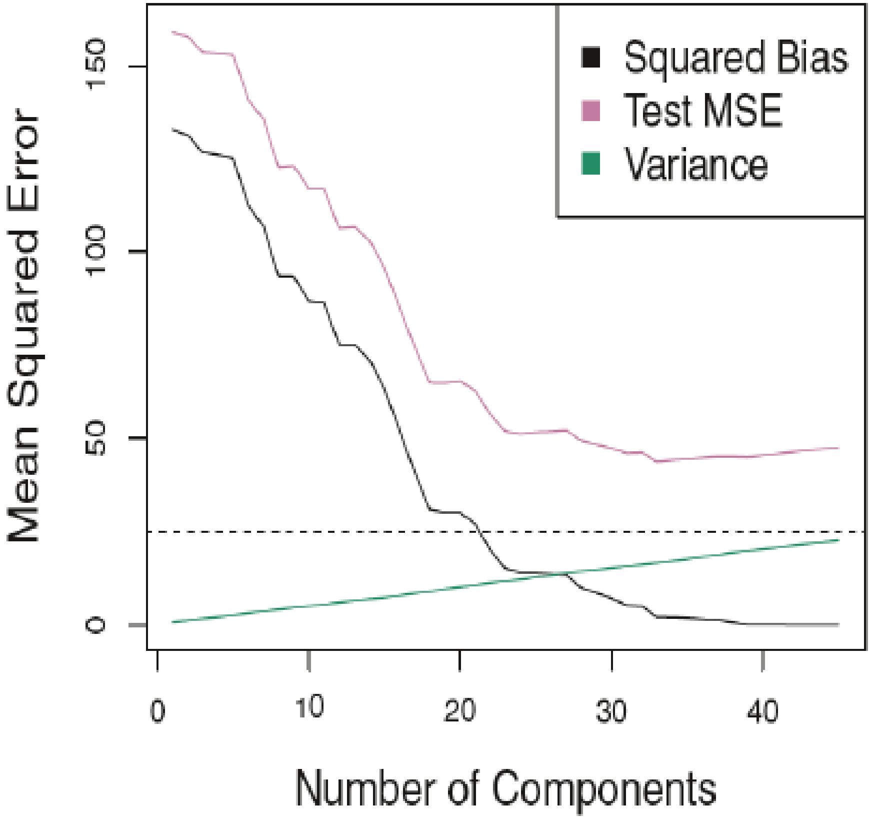
- PCR computes the derived input columns $\mathbf{z}_j = \mathbf{X}\mathbf{q}_j$ (sample principal components), and then regresses $\mathbf{y}$ on $\mathbf{z}_1, \ldots, \mathbf{z}_J$.

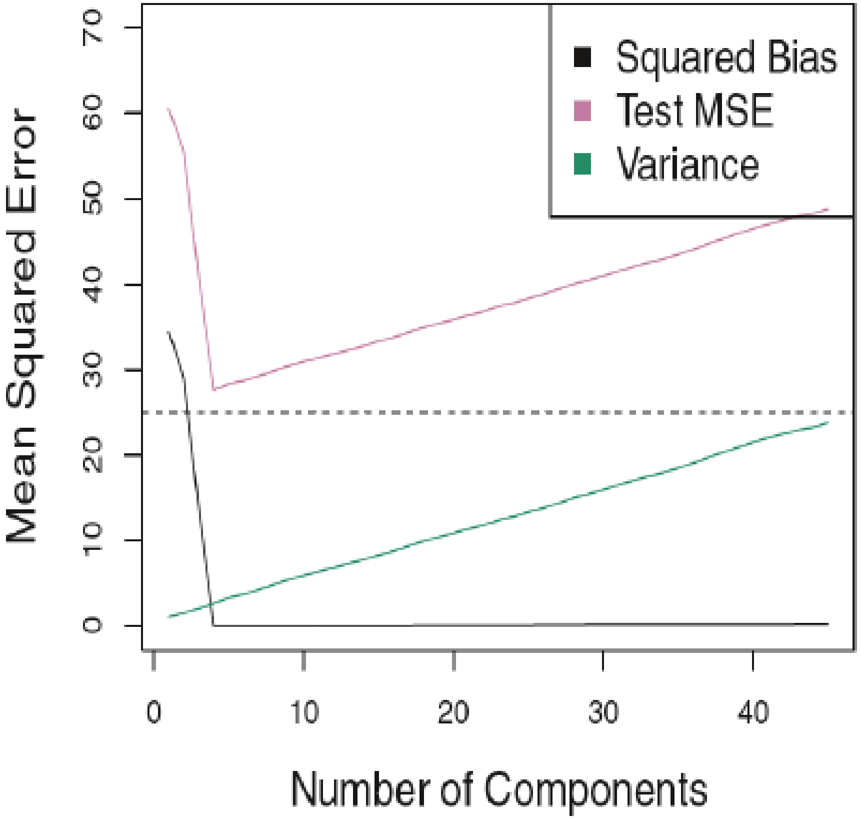
- Since the $\mathbf{z}_j$'s are orthogonal, this regression is just a sum of univariate regressions,

$$\hat{\mathbf{y}}^{pcr} = \bar{\mathbf{y}} + \sum_{j=1}^{J} \tilde{\gamma}_j \, \mathbf{z}_j$$

where $\tilde{\gamma}_j$ is the correlation coefficient of $\mathbf{y}$ on $\mathbf{z}_j$.

Some Simulated Examples with PCR





Some Remarks on PCR

- PCR works well when the first few principal components are sufficient to capture most of the variation in the predictors and the relationship with the response.

- PCR does not produce variable selection, as all predictors are included in each principal component.

- The number of principal components is typically chosen by cross-validation.

- When performing PCR, it is generally recommended to first standardize each predictor.

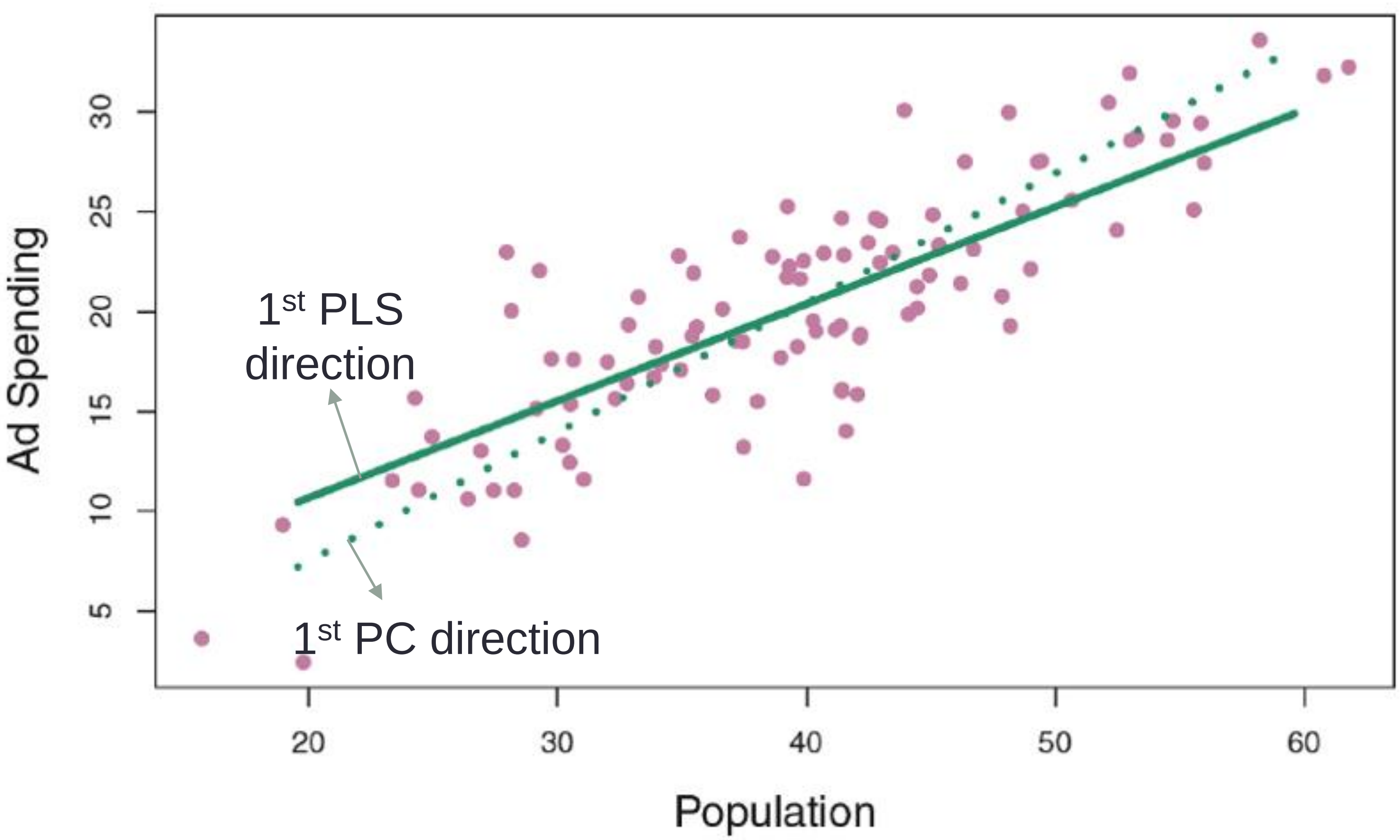
Partial Least Squares (PLS)

- PLS also constructs a set of linear combinations of predictors for regression, but unlike PCR, it uses $\mathbf{y}$ (and **X**) for this construction.
  - Assume $\mathbf{y}$ is centered, and we begin by computing the univariate regression coefficient γ_j of Y on X_j.
  - From this we construct the derived input $Z_1 = \sum_{j=1}^{p} \gamma_j \mathbf{x}_j$, which is the first PLS direction.
  - Then Y is regressed on Z_1, giving coefficient $\hat{\beta}_1$. Next we orthogonalize $X_1, \ldots, X_p$ with respect to Z_1: $R_1 = Y - \hat{\beta}_1 Z_1$, and $X_j^* = X_j - \hat{\theta}_j Z_1$, $j = 1, \ldots, p$, where $\hat{\theta}_j$ is the coefficient when X_j is regressed on Z_1. Then find the univariate regression coefficient of R_1 on X_j^*.

Partial Least Squares (Cont.)

- We continue this process, until J directions are obtained.
- In this manner, PLS produces a sequence of derived inputs or directions $Z_1, \dots, Z_J$.
- As with PCR, if we continue on to construct $J = p$ directions, we get back the OLS estimates; the use of $J < p$ directions produces a reduced-dim regression.
- Notice that in the construction of each Z_j, the inputs are weighted by the strength of their univariate effect on Y.

PCR vs PLS



1st PLS direction

1st PC direction