CAP 5625: Computational Foundations for Artificial Intelligence

Feature selection and regularization

Reducing overfitting through model selection

In the previous lecture, we learned how to reduce the effects of overfitting by performing model selection.

For model selection, we split the training data into two datasets: the training set and the validation set

We then train models of different complexity on the training set, and use the validation set to approximate their test error.

The model with ideal complexity will be one that has the minimal error rate on the validation set.

Feature selection

A related problem to model selection is **feature selection**.

In feature selection, we seek to optimize test error, while also using a minimal number of features to make predictions.

After selection, chosen variables would hopefully be the important ones.

Redundant (correlated) features are discarded.

Irrelevant (features not associated with the response) are discarded.

Why perform feature selection?

Feature selection can aid in interpretability of the fit model, as one has fewer variables associated with the response.

The reduction of model parameters may lead to better generalization or test error due to a reduction in overfitting.

Fewer features can help alleviate issues with the curse of dimensionality.

Fewer features can also reduce the time it takes to train models, as the training time will increase with increased dimensionality (more complexity/number of features).

Subset selection methods

We begin by considering subset selection methods.

These methods try to identify a subset of the p features that will lead to reduced prediction error.

We consider a brute force approach

Best subset selection

Other, computationally-efficient versions exist

Forward stepwise selection

Backward stepwise selection

but we will not discuss them in detail, as we will focus more on widely-used contemporary approaches for feature selection.

Best subset selection

Perform least squares regression for each subset of p features.

1 subset with 0 features (parameter β_0).

p subsets with 1 feature (parameters β_0 and β_1).

$p(p-1)/2$ subsets with 2 features (parameters β_0, β_1, and β_2).

For a model with k features, there are

$$\binom{p}{k} = \frac{p!}{k!\,(p-k)!}$$

possible subsets, where $k! = k(k-1)\cdots 1$ and with $0! = 1$.

The number of subsets grows rapidly.

Best subset selection algorithm

1. Let $\mathcal{M}_0$ denote the null model that contains no features, which is the model with only parameter β_0.

2. For each $k = 1,2, \ldots, p$:

   (a) Fit all $\binom{p}{k}$ models of exactly k features.

   (b) Pick the best (smallest RSS) among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$.

3. Select a single best model from $\{\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p\}$ based on model selection (*e.g.*, cross validation).

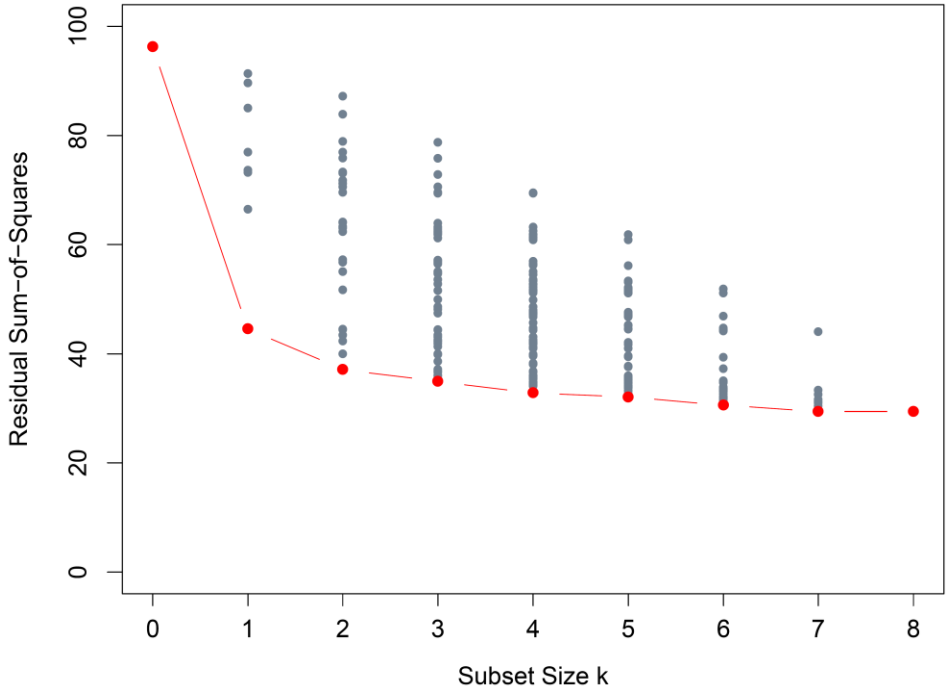
Adapted from James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Illustration of best subset selection

Dataset with $p = 8$ features.

Choose model with best RSS among all sets of models.

Best model naturally have all features on training set



Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

How many subsets do we need to consider?

Because we need to perform least squares regression on each subset, we need to consider all subsets with $0,1,2,\ldots,p$ features.

The set of all these subsets is known as the power set, which is exponentially increasing in size of the original set.

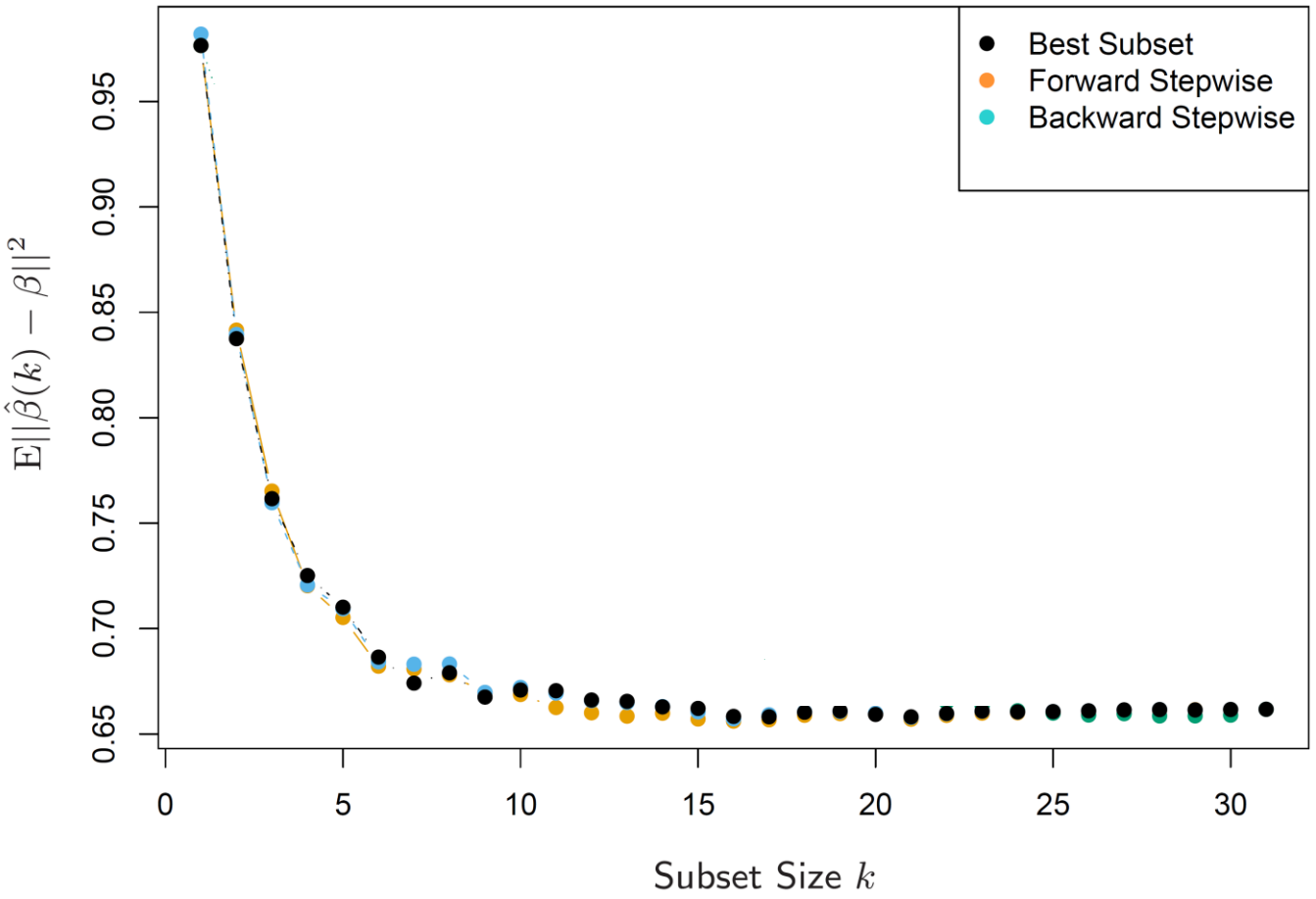
Because we have p features, the total number of fit models is 2^p, which is enormous, and computationally infeasible for anything but a small number of features (p between 30 and 40 based on an efficient algorithm of Furnival and Wilson 1974).

Best subset selection is simply not computationally feasible for many contemporary datasets.

Furnival, Wilson (1974) *Technometrics* 16:499-511.

Comparison of subset selection approaches

$N = 30$ observations with $p = 31$ standard Gaussian features with pairwise correlations of 0.85. All the parameters are zero except for 10 features.



Adapted from Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Shrinkage methods

Subset selection methods retain a subset of the features and discard the remainder.

Because of this, subset selection methods may be more interpretable than the full model and may also exhibit lower prediction error due to their avoidance of overfitting (when proper model selection procedures are employed).

However, this discrete process of retaining or discarding features often leads to high variance in practice, and therefore do not reduce the prediction error from as much as hoped from the full model.

As an alternative, a more continuous approach for feature selection called **regularization** used by **shrinkage methods** may help with this variance issue.

Ridge regression

The first shrinkage (regularization) method we consider is **ridge regression**.

Rather than discarding features, ridge regression imposes a penalty on the size of regression coefficients, such that the regression coefficients are penalized for being too large.

The optimization problem to learn the regression coefficients is very similar to the least squares problem we considered earlier.

However, an additional parameter $\lambda \geq 0$, known as the tuning parameter, is introduced, and acts as a mechanism for penalizing the model for becoming too complex.

L_2-norm penalty of RSS to obtain ridge regression

Recall that the cost function $J(\beta)$ for least squares regression was

$$J(\beta) = \text{RSS}(\beta) = \sum_{i=1}^{N} \left(y_i - x_i^T \beta\right)^2 = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

and we seek to identify β ($\hat{\beta}$) that minimizes $\text{RSS}(\beta)$.

In ridge regression, we seek to identify β ($\hat{\beta}$) that minimizes the cost function $J(\beta, \lambda)$ of the form

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

$$= \text{RSS}(\beta) + \lambda \sum_{j=1}^{p} \beta_j^2$$

Some properties of ridge regression

We can see a couple properties from the formula

$$J(\beta, \lambda) = \mathrm{RSS}(\beta) + \lambda \sum_{j=1}^{p} \beta_j^2$$

The **shrinkage penalty** $\sum_{j=1}^{p} \beta_j^2$ is small when $\beta_1, \beta_2, \ldots, \beta_p$ are close to 0, and so its effect is to shrink β_j estimates toward 0.

The **tuning parameter** λ controls the relative impact of the RSS and the shrinkage penalty on the estimate of the regression coefficients.

When the tuning parameter is 0 ($\lambda = 0$), the cost function is the original least squares cost function. That is $J(\beta, 0) = \mathrm{RSS}(\beta)$.

As $\lambda \to \infty$, estimates $\beta_j \to 0$ for $j = 1,2, \ldots, p$.

A set of estimates, and choosing the best estimate

The original formulation of least squares regression by minimizing the cost function $J(\beta) = \text{RSS}(\beta)$ leads to a single estimate $\hat{\beta}$ of the regression coefficients.

However, ridge regression leads to a set of coefficient estimates $\hat{\beta}$, where the set is across values for the tuning parameter λ.

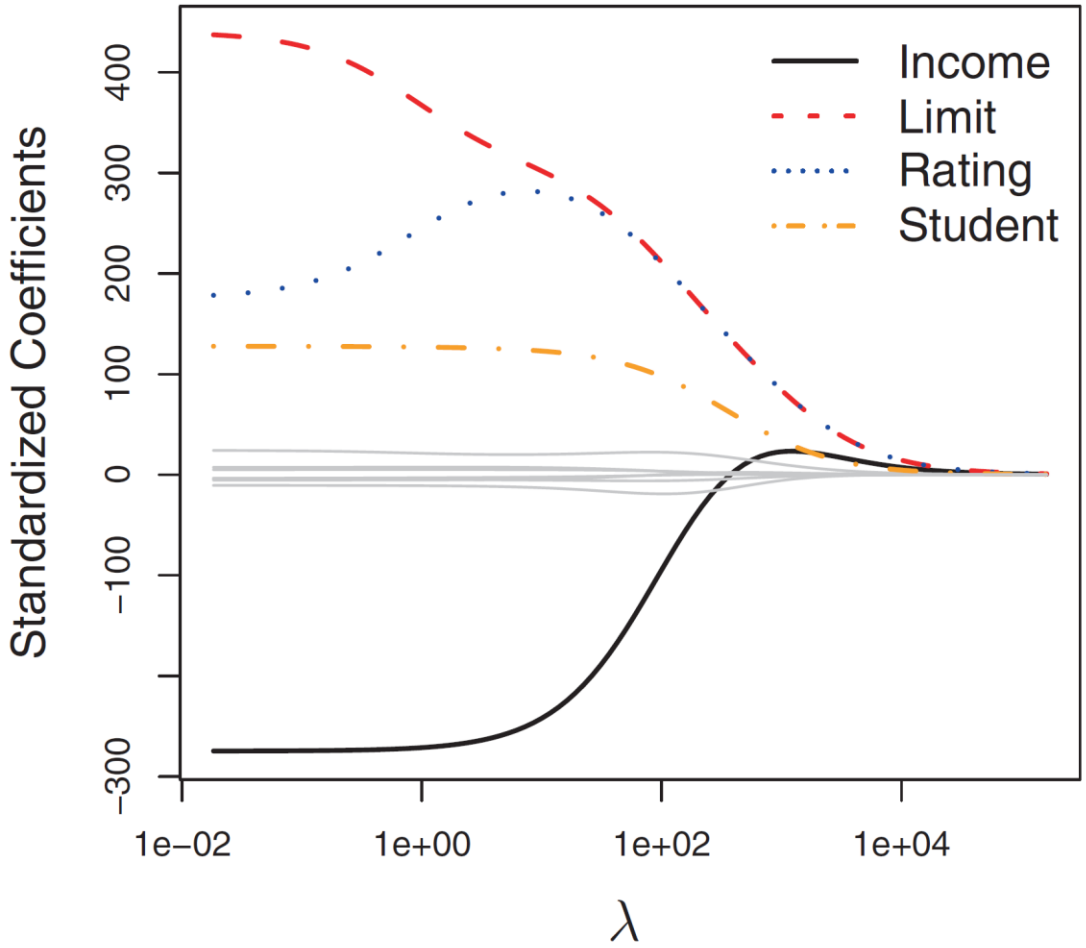
The best parameter estimate under ridge regression is chosen through cross validation.

That is, identify $\hat{\lambda}$ through cross validation that leads to the best $\hat{\beta}$.

Estimate parameters based on this tuning parameter value.

Example behavior of regression coefficients with λ

Regression coefficients for quantitative features (Income, Credit Limit and Credit Rating) along with a qualitative feature (Student Status) when predicting Credit Balance.



James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Centering the features and estimating β_0 separately

Before performing ridge regression, lets assume that the design matrix $\mathbf{X}$ is the original $N \times p$ matrix, but with the features (columns of $\mathbf{X}$) centered so that they have mean 0.

That is, define the mean of feature j

$$\bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$$

and define the element in the ith row and the jth column of $\mathbf{X}$ as

$$\tilde{x}_{ij} = x_{ij} - \bar{x}_j$$

so that the mean of column j is

$$\frac{1}{N}\sum_{i=1}^{N} \tilde{x}_{ij} = \frac{1}{N}\sum_{i=1}^{N} x_{ij} - \frac{1}{N}\sum_{i=1}^{N} \bar{x}_j = \bar{x}_j - \bar{x}_j = 0$$

Centering the features and estimating β_0 separately

We therefore define $\mathbf{X}$ as

$$\mathbf{X} = \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \cdots & \tilde{x}_{1p} \\ \tilde{x}_{21} & \tilde{x}_{22} & \cdots & \tilde{x}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{x}_{N1} & \tilde{x}_{N2} & \cdots & \tilde{x}_{Np} \end{bmatrix}$$

We can then estimate β_0 as $\hat{\beta}_0 = \bar{y}$, where

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

is the sample mean of the training data response.

To see this, calculate the partial derivative of $J(\beta, \lambda)$ with respect to β_0.

Finding the optimal β_0

$$J(\beta,\lambda)=\sum_{i=1}^{N}\left(y_i-\beta_0-\sum_{j=1}^{p}\tilde{x}_{ij}\beta_j\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

Taking the partial derivative with respect to β_0, we have

$$\frac{\partial}{\partial\beta_0}J(\beta,\lambda)=-2\sum_{i=1}^{N}\left(y_i-\beta_0-\sum_{j=1}^{p}\tilde{x}_{ij}\beta_j\right)$$

$$=-2N\bar{y}+2N\beta_0+\sum_{j=1}^{p}\beta_j\sum_{i=1}^{N}\left(x_{ij}-\bar{x}_j\right)$$

$$=-2N\bar{y}+2N\beta_0+\sum_{j=1}^{p}\beta_j\left(N\bar{x}_j-N\bar{x}_j\right)$$

$$=-2N\bar{y}+2N\beta_0$$

Finding the optimal β_0

Setting the partial derivative to 0, we have

$$-2N\bar{y} + 2N\beta_0 = 0$$

and solving for β_0 gives

$$\beta_0 = \bar{y}$$

Therefore, β_0 can be estimated separately from the sample mean of the training set response as $\hat{\beta}_0 = \bar{y}$.

We will now assume that in our data matrix $\mathbf{X} \in \mathbb{R}^{N \times p}$ the columns (features) are centered, and that the responses are also centered with $\tilde{y}_i = y_i - \bar{y}$.

We will now denote each element by x_{ij} rather than $\tilde{x}_{ij}$ and y_i rather than $\tilde{y}_i$ for ease of notation, with the assumption that the data are centered.

Finding the regression coefficients (normal equations)

As we performed for least squares regression, we can derive a set of normal equations for ridge regression by taking partial derivatives of the cost function $J(\beta, \lambda)$ with respect to the coefficient β_j of each feature j, and setting these derivatives to 0.

Recall that

$$J(\beta, \lambda) = \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Taking the partial derivative with respect to β_k, $k > 0$, we have

$$\frac{\partial}{\partial \beta_k} J(\beta, \lambda) = -\sum_{i=1}^{N} 2 x_{ik} \left(y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right) + 2\lambda \beta_k$$

Finding the regression coefficients (normal equations)

This partial derivative

$$\frac{\partial}{\partial \beta_k} J(\beta, \lambda) = -\sum_{i=1}^{N} 2x_{ik} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right) + 2\lambda\beta_k$$

can be rewritten as

$$\frac{\partial}{\partial \beta_k} J(\beta, \lambda) = -2[x_{1k} \quad x_{2k} \quad \cdots \quad x_{Nk}] \begin{bmatrix} y_1 - x_1^T\beta \\ y_2 - x_2^T\beta \\ \vdots \\ y_N - x_N^T\beta \end{bmatrix} + 2\lambda\beta_k$$

$$= -2[x_{1k} \quad x_{2k} \quad \cdots \quad x_{Nk}](\mathbf{y} - \mathbf{X}\beta) + 2\lambda\beta_k$$

where $[x_{1k} \quad x_{2k} \quad \cdots \quad x_{Nk}]$ is an N-dimensional row vector.

Finding β (normal equations) that minimizes $J(\beta, \lambda)$

Putting this together, to identify the minimum, we take the gradient with respect to β, which is a p-dimensional vector, with dimension k the partial derivative of $J(\beta, \lambda)$ with respect to β_k, $k \in \{1,2, \dots, p\}$. The gradient is therefore

$$\frac{\partial}{\partial \beta} J(\beta, \lambda) = \begin{bmatrix} -2[x_{11} & x_{21} & \cdots & x_{N1}](\mathbf{y} - \mathbf{X}\beta) + 2\lambda\beta_1 \\ -2[x_{12} & x_{22} & \cdots & x_{N2}](\mathbf{y} - \mathbf{X}\beta) + 2\lambda\beta_2 \\ & & & \vdots \\ -2[x_{1p} & x_{2p} & \cdots & x_{Np}](\mathbf{y} - \mathbf{X}\beta) + 2\lambda\beta_p \end{bmatrix}$$

$$= -2 \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{N1} \\ x_{12} & x_{22} & \cdots & x_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1p} & x_{2p} & \cdots & x_{Np} \end{bmatrix} (\mathbf{y} - \mathbf{X}\beta) + 2\lambda \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}$$

$$= -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) + 2\lambda\beta$$

Finding β (normal equations) that minimizes $J(\beta, \lambda)$

Setting the gradient to the 0 vector, gives

$$-\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta = 0$$

which yields

$$-\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\beta + \lambda\beta = -\mathbf{X}^T\mathbf{y} + (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)\beta = 0$$

or

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)\beta = \mathbf{X}^T\mathbf{y}$$

If $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)$ is invertible, then the estimated coefficients are

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

and is the vector $\beta \in \mathbb{R}^p$ that minimizes

$$J(\beta, \lambda) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_2^2$$

with $\|v\|_2 = \sqrt{v_1^2 + v_2^2 + \cdots v_n^2}$ for $v = [v_1, v_2, \ldots, v_n]^T \in \mathbb{R}^n$, and is known as the L_2 norm.

The “hat” or projection matrix under ridge regression

We have now shown that $\hat{\beta} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^T\mathbf{y}$ is the ridge estimate of the regression coefficients given training observations (x_i, y_i), $i = 1,2,\dots,N$.

The fitted value vector (estimate of the output of the original training data), is given by

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

Define the hat (projection) matrix under ridge regression, denoted by $\mathbf{H}_\lambda$, as

$$\mathbf{H}_\lambda = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}\mathbf{X}^T$$

such that $\hat{\mathbf{y}} = \mathbf{H}_\lambda\mathbf{y}$.

Ridge regression is not scale-invariant

An important point is that the estimated coefficients can change substantially when multiplying certain features by a constant.

This is due to the penalty term involving the sum of the squares of the coefficients.

For example, suppose we are attempting to predict credit balance Y based on income X_j and a number of other $p - 1$ features.

Suppose that rather than having income measured in dollars, it is measured in thousands of dollars.

Therefore to have the same $X_j\beta_j$, β_j must be 1000 times larger, leading to a massive penalty.

Standardizing dataset before applying ridge regression

Because ridge regression is not scale-invariant, it is important that all features (columns) of the input data are standardized by dividing by the standard deviation across the rows of the columns.

Together with centering the input data, we have that elements of the design matrix $\mathbf{X}$ should take values

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{\hat{\sigma}_j}$$

where $\hat{\sigma}_j$ is an estimate of the standard deviation of the jth feature and is computed as

$$\hat{\sigma}_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{ij} - \bar{x}_j\right)^2}$$

Finding β that minimizes $J(\beta, \lambda)$ with gradient descent

Recall that for least squares, there was an iterative solution called batch gradient descent that we can employ to identify the minimum.

Because we can calculate the gradient under ridge regression, we can also estimate β using gradient descent for fixed λ.

We formulate the gradient descent update for β_j, $j \in \{1,2,\dots,p\}$ as

$$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta, \lambda)$$

with this update performed for all j simultaneously and where

$$\frac{\partial}{\partial \beta_j} J(\beta, \lambda) = -2 \sum_{i=1}^{N} x_{ij} \left(y_i - x_i^T \beta \right) + 2\lambda \beta_j$$

which we derived earlier.

Finding β that minimizes $J(\beta, \lambda)$ with gradient descent

Putting it together, for each iteration, we get as updates

$$\beta_j := \beta_j + 2\alpha \left(\sum_{i=1}^{N} x_{ij} \left(y_i - x_i^T \beta \right) - \lambda \beta_j \right)$$

and we compute these updates for all parameters j at each iteration.

A **vectorized** version of the update can be written as

$$\beta := \beta + 2\alpha [\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta) - \lambda \beta]$$

We can also formulate this update as either stochastic or mini-batch gradient descent, which approximate the cost function $J(\beta, \lambda)$.

Finding β with stochastic gradient descent

Fix λ

1. Randomly initialize β.
2. Randomly permute (shuffle) the order of the N training observations.
3. For observation i in the permuted list of training observations, update each of the p parameters as

$$\beta_j := \beta_j + 2\alpha\left(x_{ij}\left[y_i - x_i^T\beta\right] - \lambda\beta_j\right)$$

4. Repeat steps 2 and 3 until convergence.

Finding β with mini-batch gradient descent

Fix λ

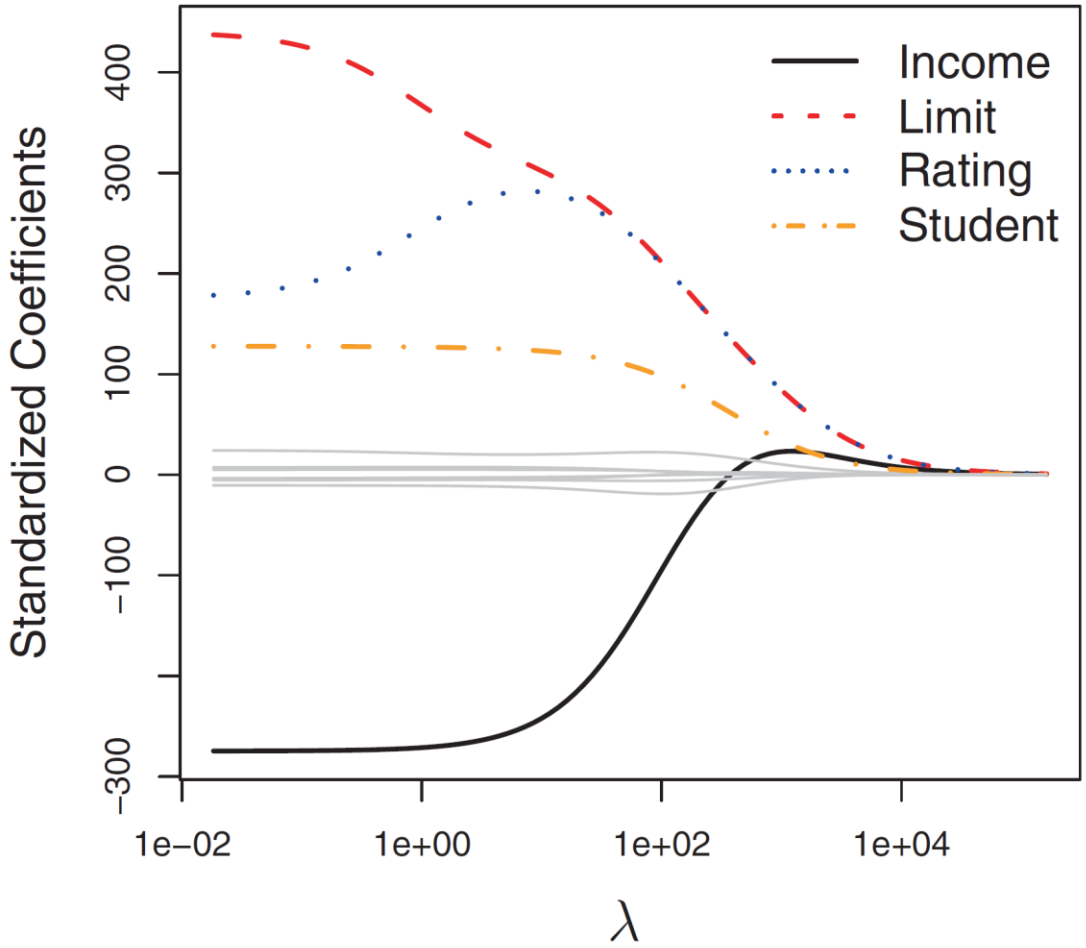
1. Randomly initialize β.
2. Choose a subset $\mathcal{S}$ of size n of the N training observations uniformly at random.
3. Approximate the gradient with these n observations and update each of the p parameters as

$$\beta_j := \beta_j + 2\alpha\left(\sum_{(x_i,y_i)\in\mathcal{S}} x_{ij}\left(y_i - x_i^T\beta\right) - \lambda\beta_j\right)$$

4. Repeat steps 2 and 3 until convergence.

Example behavior of regression coefficients with λ

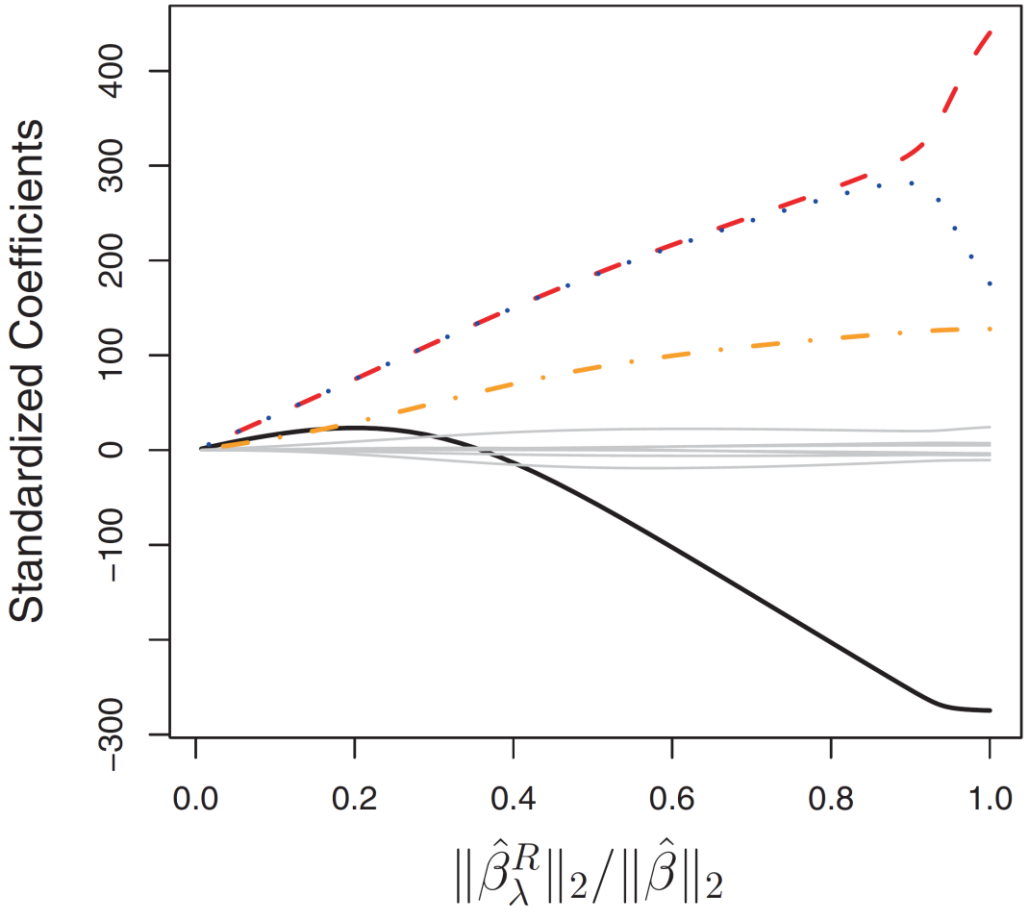
Regression coefficients for quantitative features (Income, Credit Limit and Credit Rating) along with a qualitative feature (Student Status) when predicting Credit Balance.



James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

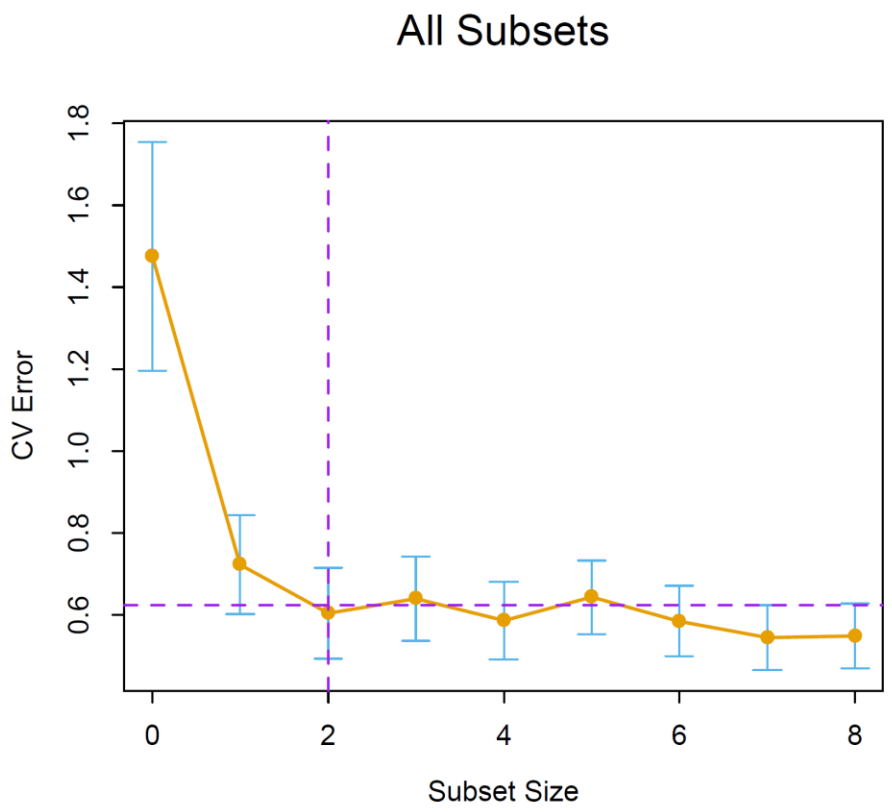
Amount estimates shrunken to zero by ridge regression

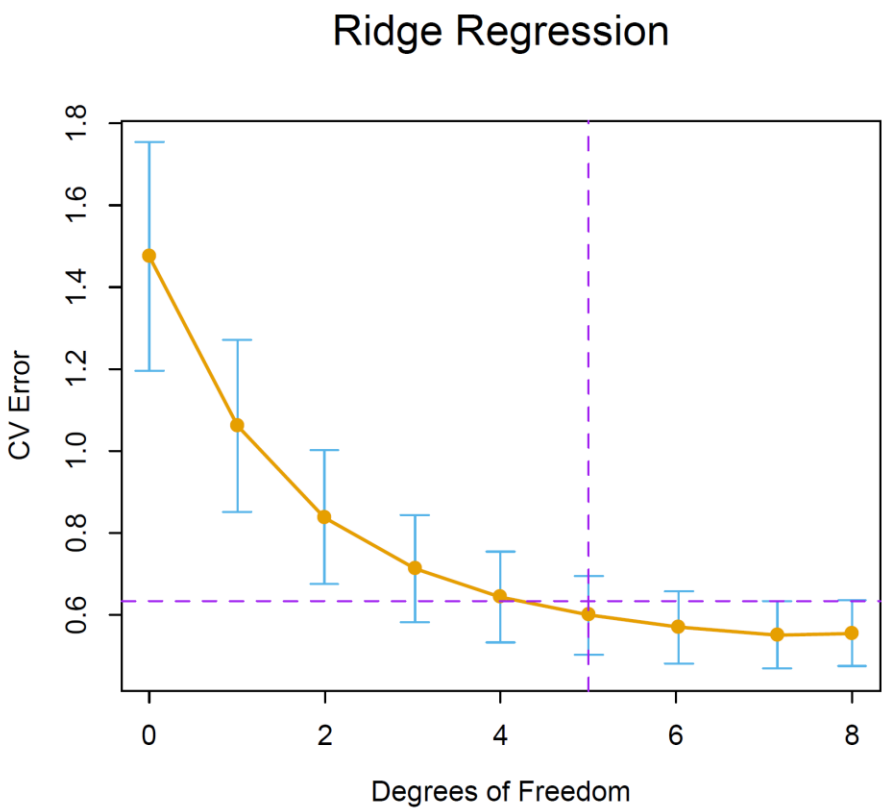
Plotted are the same credit results, but instead the x-axis plots $\|\hat{\beta}_{\text{RIDGE}}\|_2 / \|\hat{\beta}\|_2$. As λ increases, the sum of squares of parameter values $\|\hat{\beta}_{\text{RIDGE}}\|_2$ always decreases, and therefore $\|\hat{\beta}_{\text{RIDGE}}\|_2 / \|\hat{\beta}\|_2$ decreases.



James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

CV error for best subset selection and ridge regression

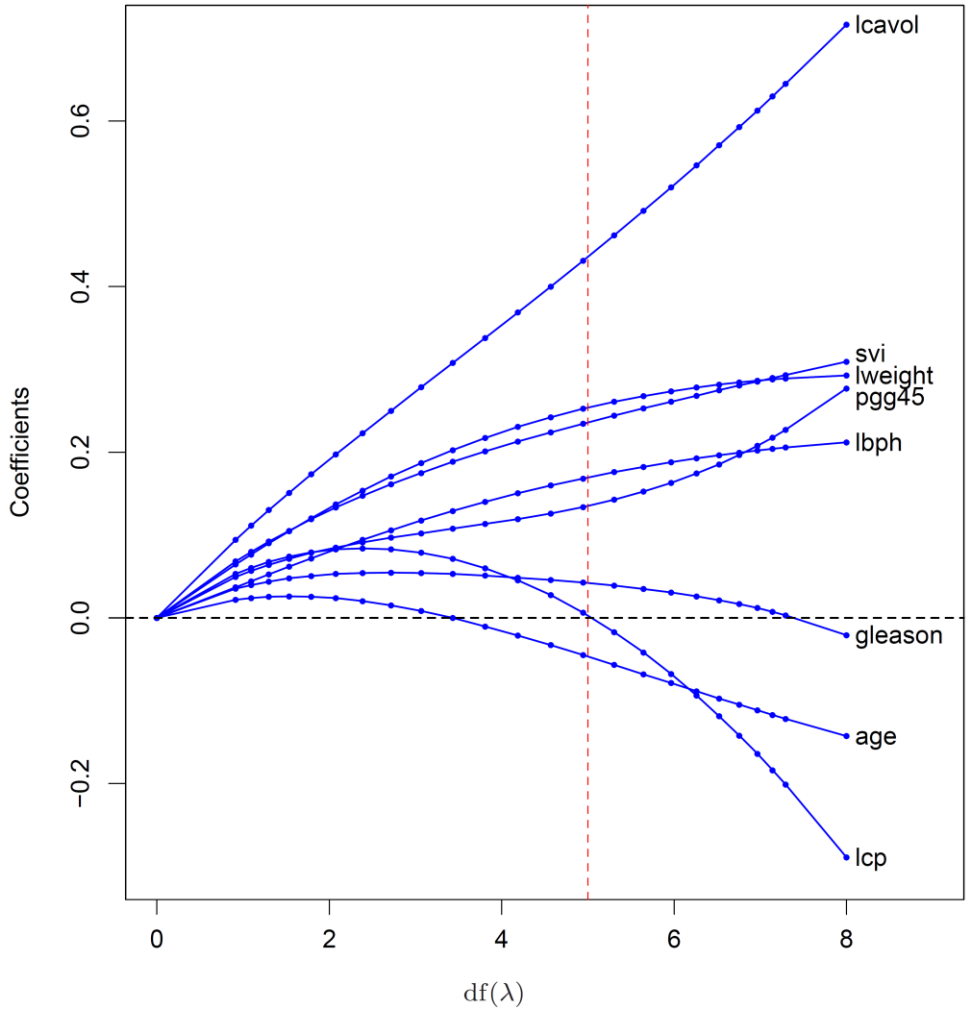




Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Ridge regression applied to prostate cancer dataset

As the degrees of freedom decrease (model becomes less flexible), all parameter shrink simultaneously to 0.



Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Comparison of different approaches

Inferred coefficients and test errors for least squares, best subset selection, and ridge regression on prostate cancer prediction dataset.

Term	LS	Best Subset	Ridge
Intercept	2.465	2.477	2.452
lcavol	0.680	0.740	0.420
lweight	0.263	0.316	0.238
age	−0.141		−0.046
lbph	0.210		0.162
svi	0.305		0.227
lcp	−0.288		0.000
gleason	−0.021		0.040
pgg45	0.267		0.133
Test Error	0.521	0.492	0.492

Adapted from Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Comparison of different approaches

Why does ridge regression have better test error than least squares?

Term	LS	Best Subset	Ridge
Intercept	2.465	2.477	2.452
lcavol	0.680	0.740	0.420
lweight	0.263	0.316	0.238
age	−0.141		−0.046
lbph	0.210		0.162
svi	0.305		0.227
lcp	−0.288		0.000
gleason	−0.021		0.040
pgg45	0.267		0.133
Test Error	0.521	0.492	0.492

Adapted from Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Tuning parameter makes bias-variance tradeoff

The penalty term reduces the degrees of freedom associated with the regression coefficients.

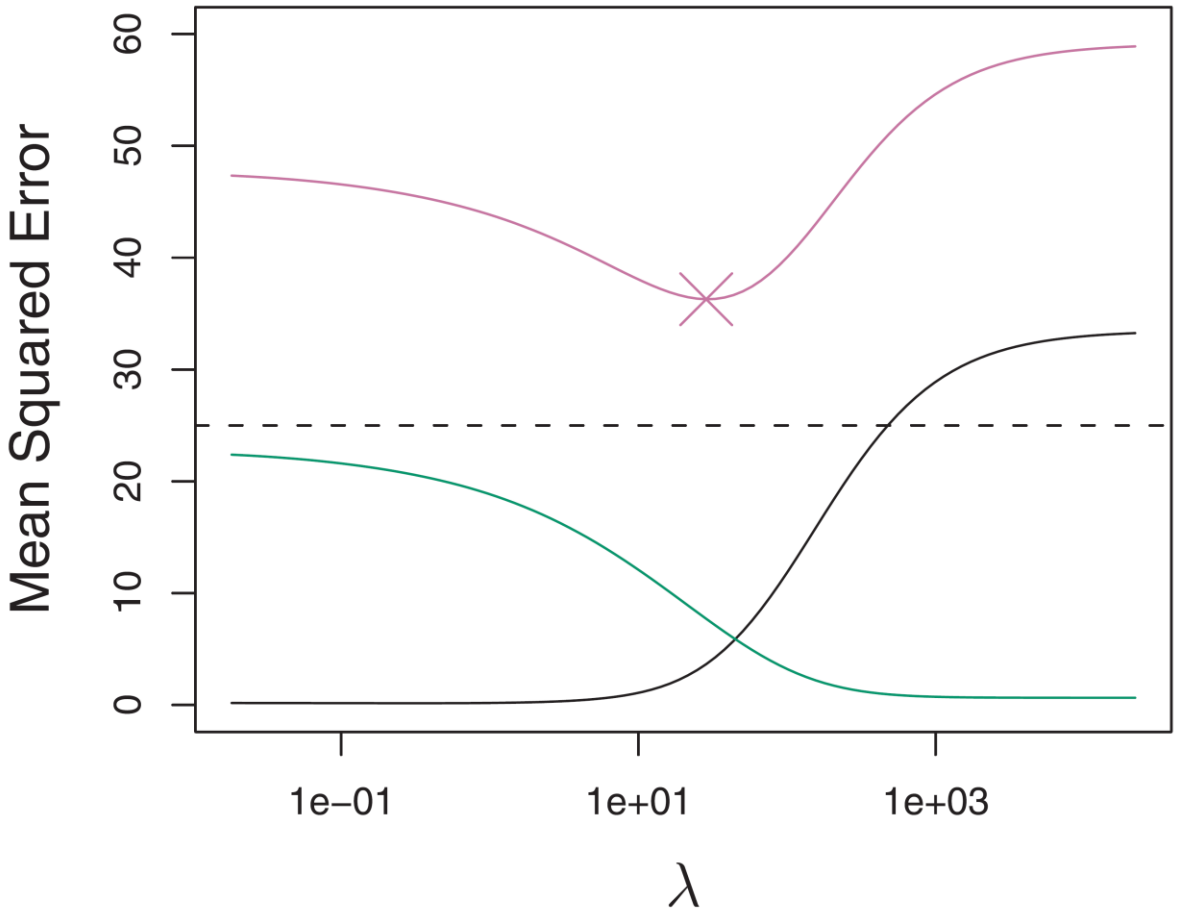
That is, as the tuning parameter increase, degrees of freedom decreases, and we have fewer effective parameters.

These fewer effective parameters reduces the overall model complexity, leading to a greater bias but a decreased variance.

The optimal λ value will be chosen so that there is an optimal tradeoff between bias and variance, yielding improved test error.

Tuning parameter makes bias-variance tradeoff

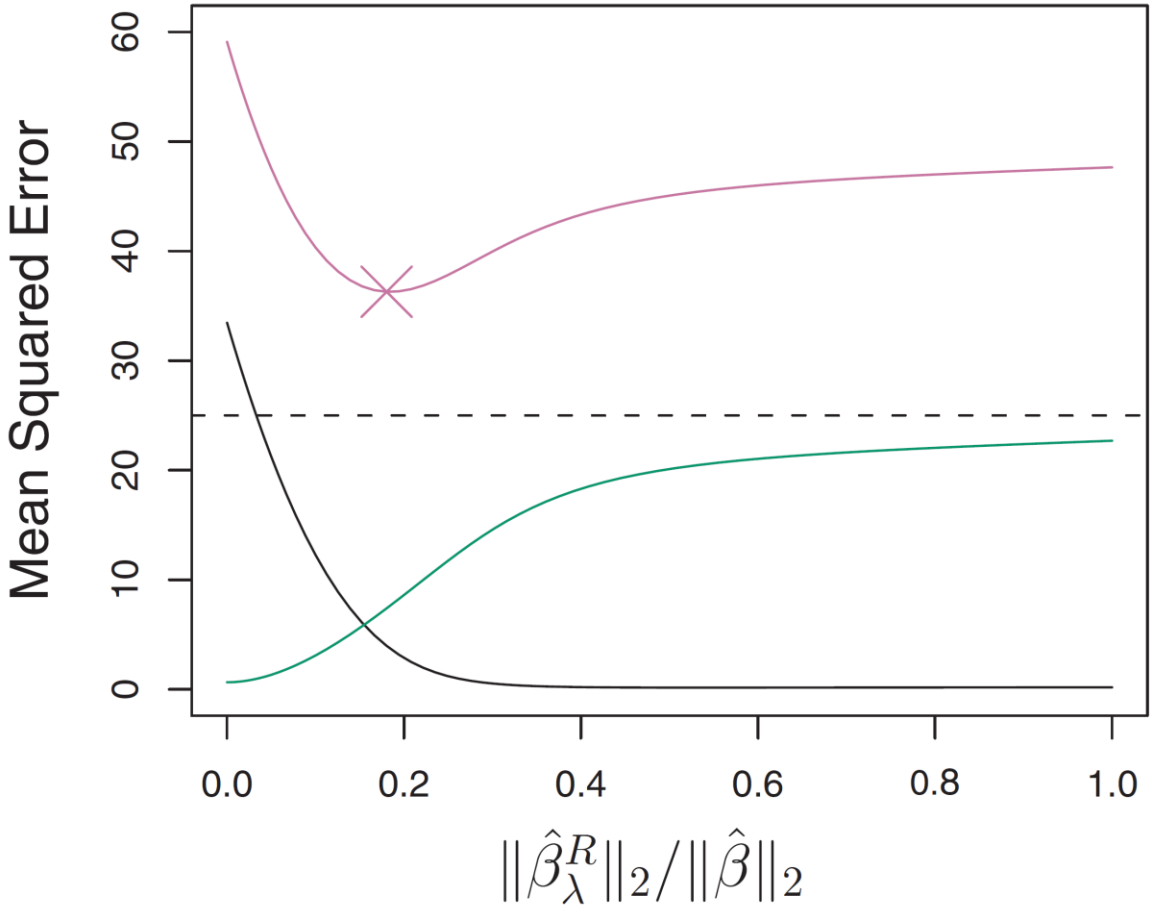
Optimal MSE (purple) occurs for an intermediate value of λ, due to a reduction in complexity by increasing squared bias (black) and decreasing variance (green). Dashed line is irreducible error.



James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Tuning parameter makes bias-variance tradeoff

Optimal MSE (purple) occurs for an intermediate shrinkage level, due to a reduction in complexity by increasing squared bias (black) and decreasing variance (green). Dashed line is irreducible error.



James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Ridge regression vs. subset selection

When the number of parameters is huge (*e.g.*, $p > N$), then the least squares parameter estimates do not have a unique solution.

However, ridge regression can still perform well trading small increases in bias for potentially larger decreases in variance

Moreover, there is a substantial computational advantage to ridge regression compared to subset selection methods.

For best subset selection, we need to search through 2^p models (stepwise selection considers $1 + p[p + 1]/2$ models), whereas for any value of λ the ridge regression fit is a single model that can be fit quickly.

Alternate formulation of optimization problem

Original ridge regression formulation is to find $\beta = [\beta_1, \beta_2, \ldots, \beta_p]$ that minimizes

$$J(\beta, \lambda) = \mathrm{RSS}(\beta) + \lambda \sum_{j=1}^{p} \beta_j^2$$

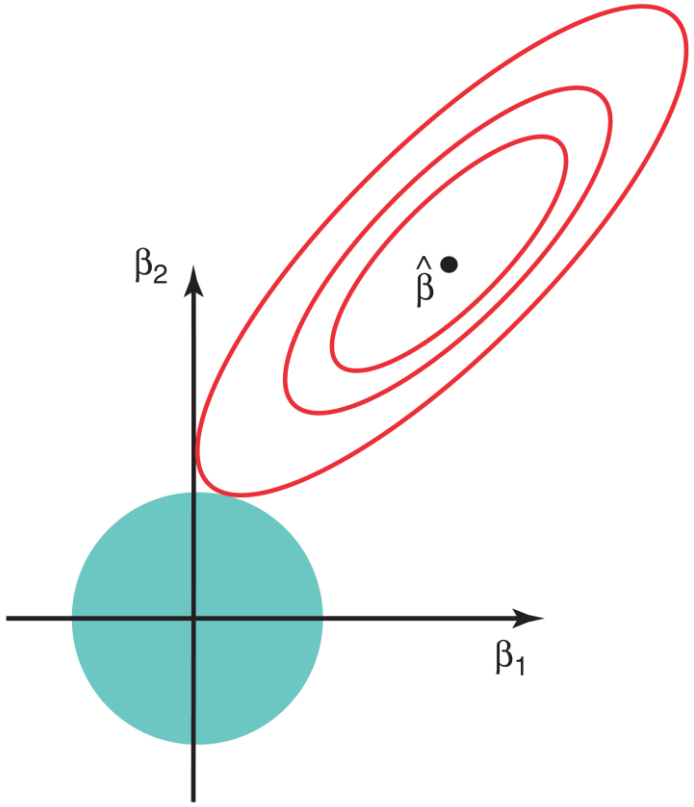
An alternative formulation is to instead solve the constrained problem of finding β that minimizes $\mathrm{RSS}(\beta)$ subject to the constraint that

$$\sum_{j=1}^{p} \beta_j^2 \leq s$$

for some non-negative value s.

Illustration based on alternate formulation

Contour of the $\mathrm{RSS}(\beta)$ (red) and the constraint function $\beta_1^2 + \beta_2^2 \leq s$ (blue) for ridge regression with two features.

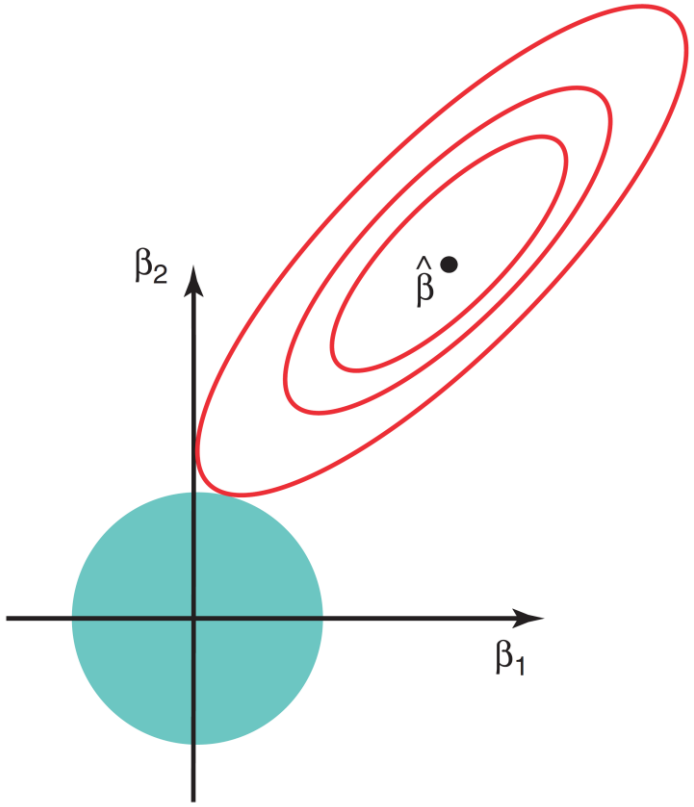


If s is large (like λ near 0), then constraint region contains the least squares estimate $\hat{\beta}$.

James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Illustration based on alternate formulation

Contour of the $\mathrm{RSS}(\beta)$ (red) and the constraint function $\beta_1^2 + \beta_2^2 \leq s$ (blue) for ridge regression with two features.

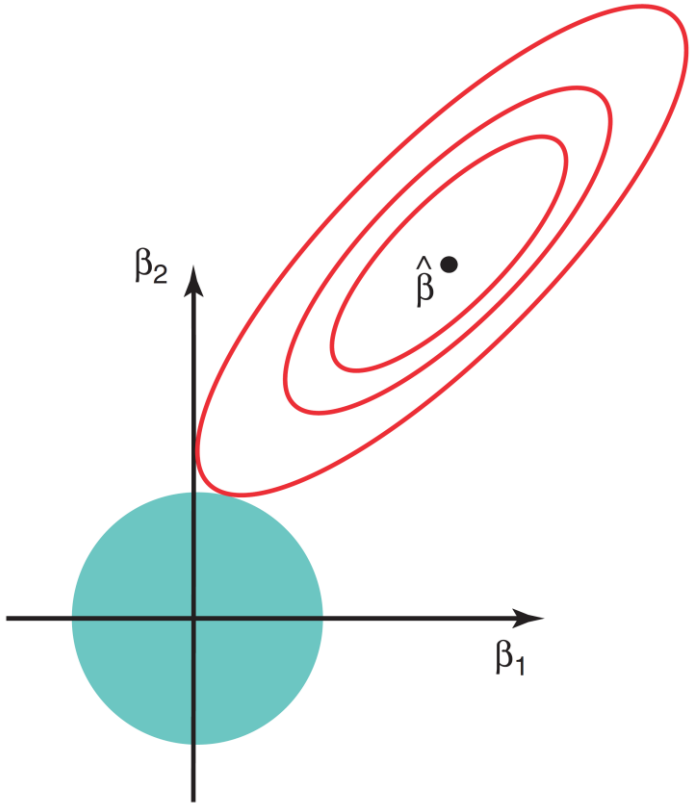


However, when least squares estimate resides outside the constraint region, then ridge regression estimate is different.

James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Illustration based on alternate formulation

Contour of the $\mathrm{RSS}(\beta)$ (red) and the constraint function $\beta_1^2 + \beta_2^2 \leq s$ (blue) for ridge regression with two features.



The ridge regression estimate will lie on the boundary of the constraint region, and the point it touches the $\mathrm{RSS}(\beta)$.

James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

But ridge did not really perform feature selection

Ridge regression provided a constraint on the parameters such that the effective number of parameters can change based on the tuning parameter.

However, this shrinks parameters toward 0 together, and retains parameters for all the features.

A goal may be to not only have a model with good prediction accuracy, but one that has fewer features so that it is more interpretable.

We introduce a method, termed **lasso**, that performs shrinkage (like ridge regression) while simultaneously performing feature selection.

L_1-norm penalty of RSS to obtain lasso regression

Assume that the inputs have been standardized, and that the output has been centered, as in ridge regression.

In **lasso regression**, we seek to identify β ($\hat{\beta}$) that minimizes the cost function $J(\beta, \lambda)$ of the form

$$J(\beta,\lambda) = \sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

$$= \sum_{i=1}^{N}\left(y_i - x_i^T\beta\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

$$= \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1$$

$$= \mathrm{RSS}(\beta) + \lambda\|\beta\|_1$$

with L_1 norm $\|v\|_1 = |v_1| + \cdots + |v_n|$ for $v = [v_1, v_2, \ldots, v_n]^T \in \mathbb{R}^n$.

Some properties of lasso regression

We can see a couple properties from the formula

$$J(\beta, \lambda) = \mathrm{RSS}(\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

The **shrinkage penalty** $\sum_{j=1}^{p} |\beta_j|$ is small when $\beta_1, \beta_2, \ldots, \beta_p$ are close to (or exactly) 0, and so its effect is to shrink β_j estimates to 0.

When the **tuning parameter** is 0 ($\lambda = 0$), the cost function is the original least squares cost function. That is $J(\beta, 0) = \mathrm{RSS}(\beta)$.

As $\lambda \to \infty$, estimates $\beta_j = 0$ for $j = 1,2 \ldots, p$.

Cannot derive normal equations for this non-differential cost function.

Some properties of lasso regression

The cost function for lasso is not differentiable, due to the penalty term.

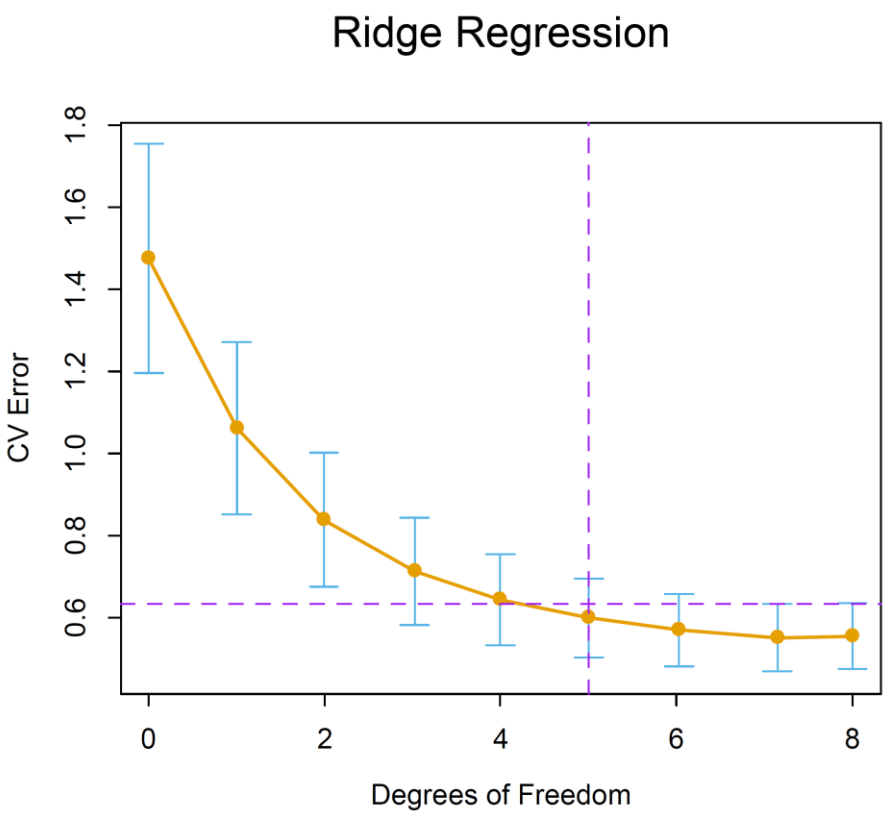
To see this, consider a penalty term with a single feature.

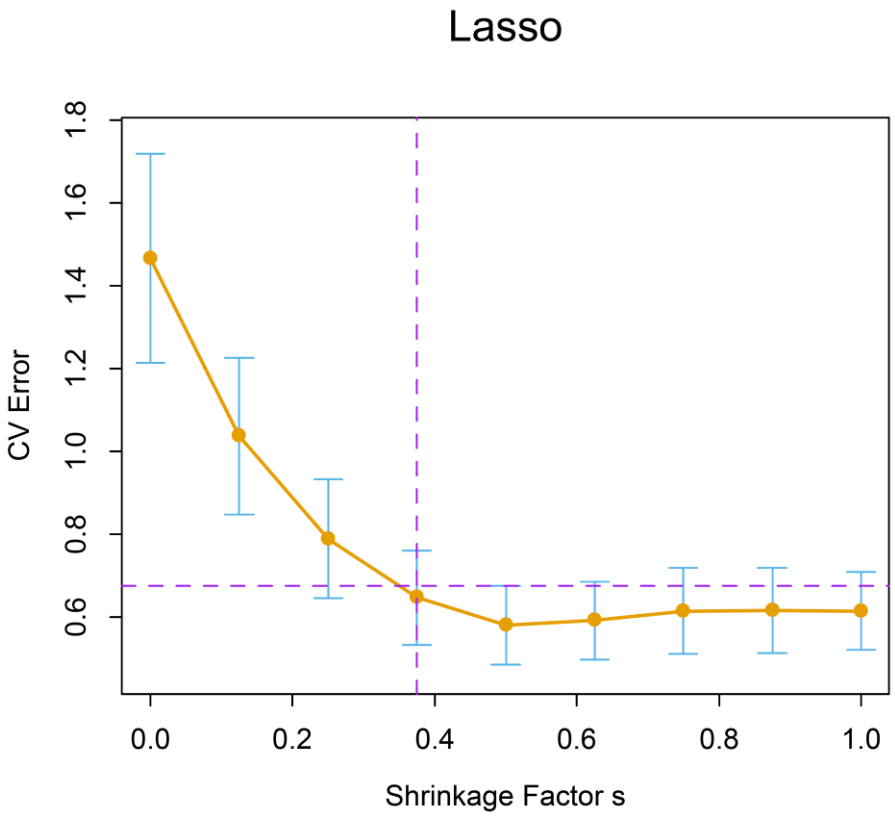
The absolute value is not differentiable at 0.

Therefore, we cannot derive normal equations as we did for least squares and ridge regression.

In the next lecture we will consider an algorithm called **coordinate descent** to estimate the regression coefficients.

CV error for ridge and lasso regression

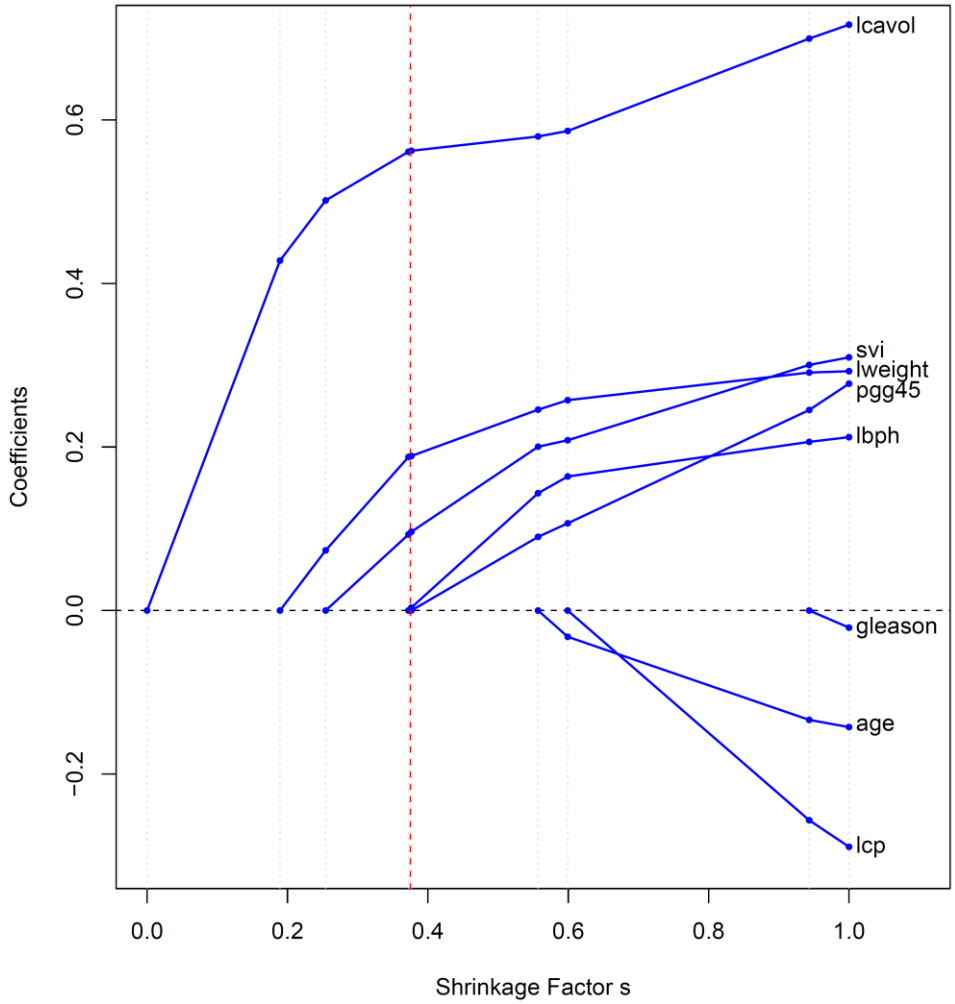




Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Lasso applied to prostate cancer dataset

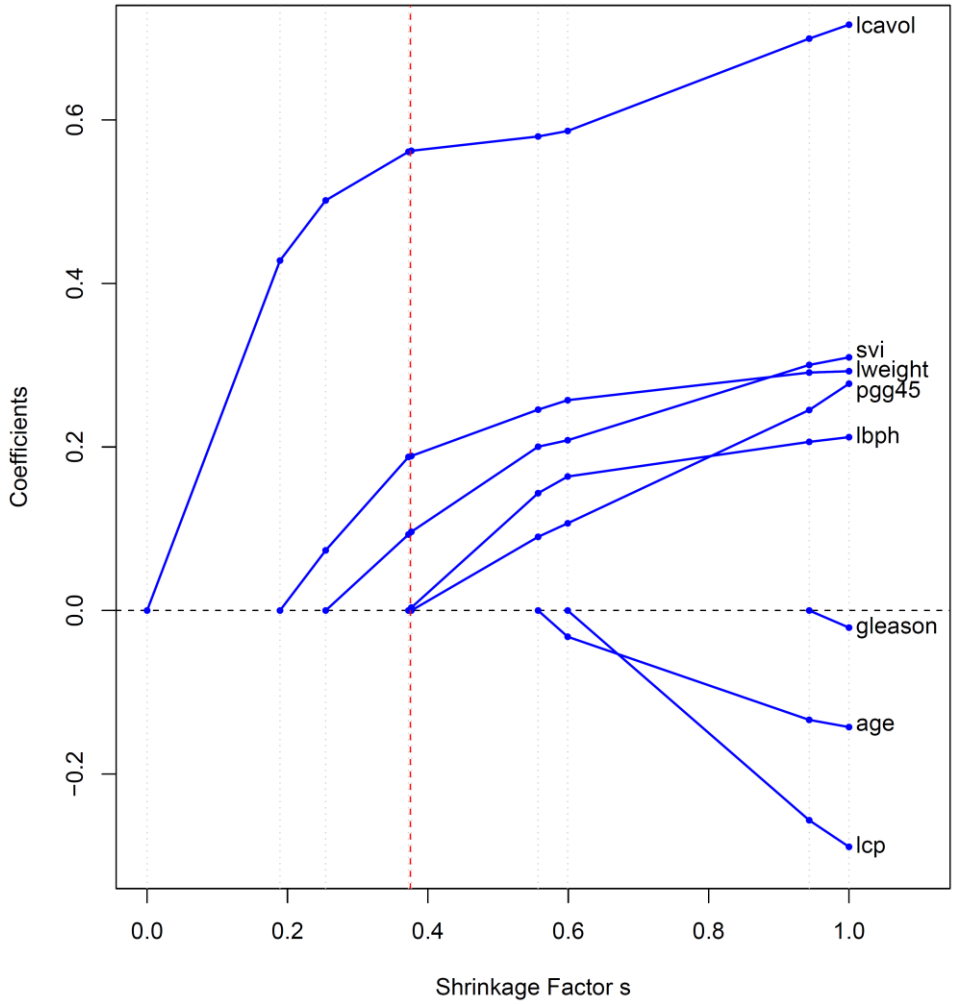
Parameters shrink (model becomes less flexible) as the shrinkage factor $s = \|\hat{\beta}_{\text{LASSO}}\|_1 / \|\hat{\beta}\|_1$ decreases from 1 to 0.



Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Lasso applied to prostate cancer dataset

When shrinkage factor becomes small, coefficients for some features go to 0 and stay 0. All coefficients are 0 when $s = 0$.



Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Comparison of different approaches

Inferred coefficients and test errors for different approaches on prostate cancer prediction dataset.

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.465	2.477	2.452	2.468
lcavol	0.680	0.740	0.420	0.533
lweight	0.263	0.316	0.238	0.169
age	−0.141		−0.046	
lbph	0.210		0.162	0.002
svi	0.305		0.227	0.094
lcp	−0.288		0.000	
gleason	−0.021		0.040	
pgg45	0.267		0.133	
Test Error	0.521	0.492	0.492	0.479

Adapted from Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Comparison of different approaches

Lasso represents a compromise between best subset selection and ridge regression, shrinking some parameters and discarding others.

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.465	2.477	2.452	2.468
lcavol	0.680	0.740	0.420	0.533
lweight	0.263	0.316	0.238	0.169
age	−0.141		−0.046	
lbph	0.210		0.162	0.002
svi	0.305		0.227	0.094
lcp	−0.288		0.000	
gleason	−0.021		0.040	
pgg45	0.267		0.133	
Test Error	0.521	0.492	0.492	0.479

Adapted from Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Tuning parameter makes bias-variance tradeoff

Just as with ridge regression, the penalty term reduces the flexibility of the model.

That is, as the tuning parameter increases, the model becomes more constrained, and therefore has fewer effective parameters.

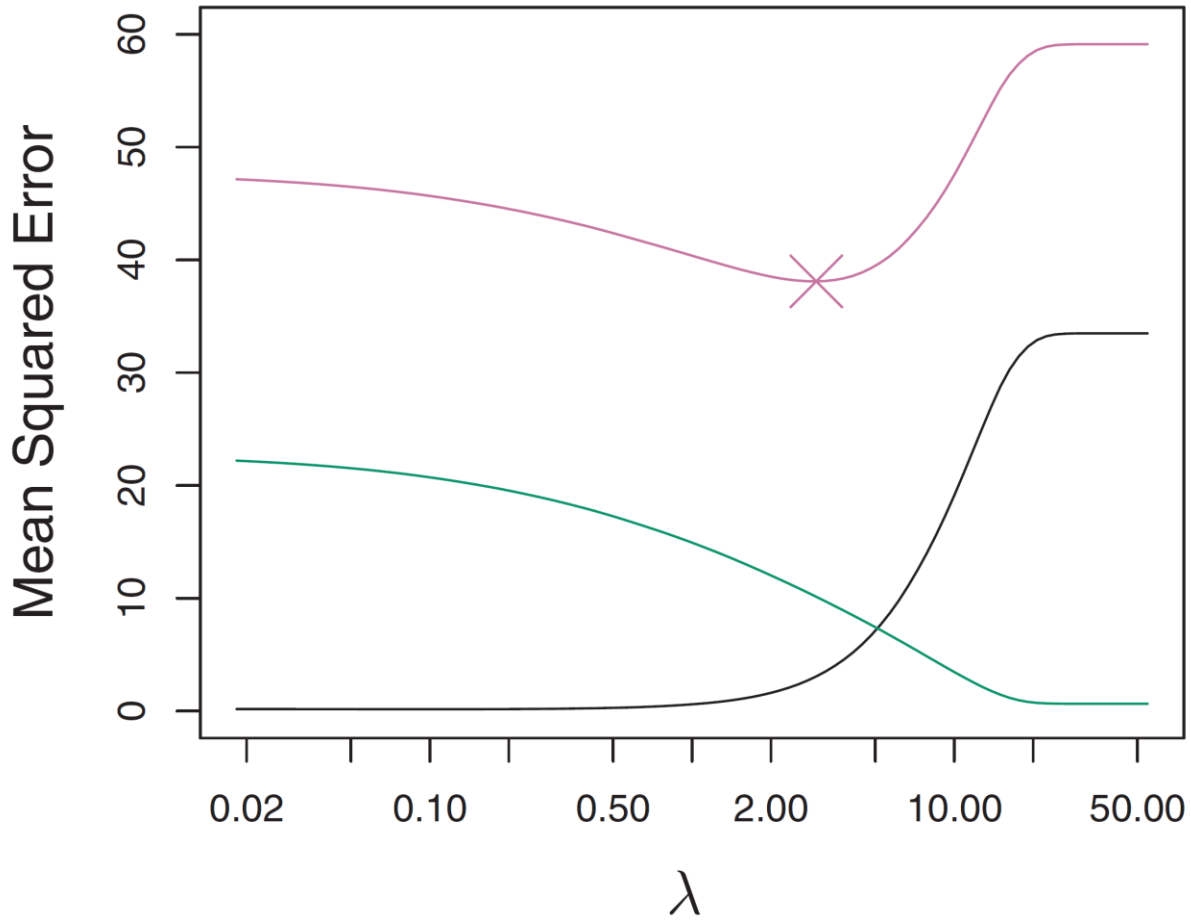
These fewer effective parameters reduce the overall model complexity, leading to a greater bias but a decreased variance.

In addition, the decreased variance is associated with some features becoming discarded as the tuning parameter increases in value.

The optimal λ value will be chosen so that there is an optimal tradeoff between bias and variance, yielding improved test error.

Tuning parameter makes bias-variance tradeoff

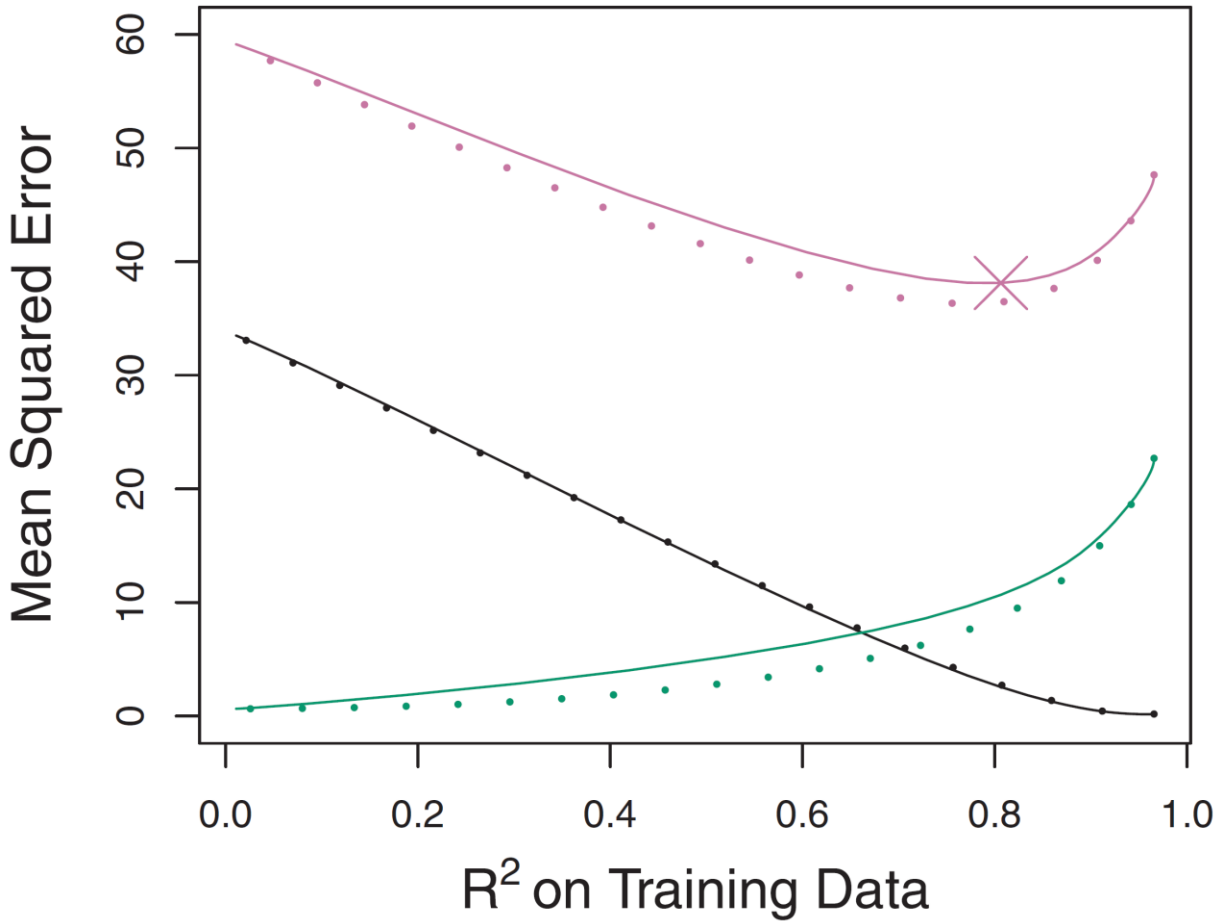
Optimal MSE (purple) occurs for an intermediate value of λ, due to a reduction in complexity by increasing squared bias (black) and decreasing variance (green).



James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Ridge regression can outperform lasso

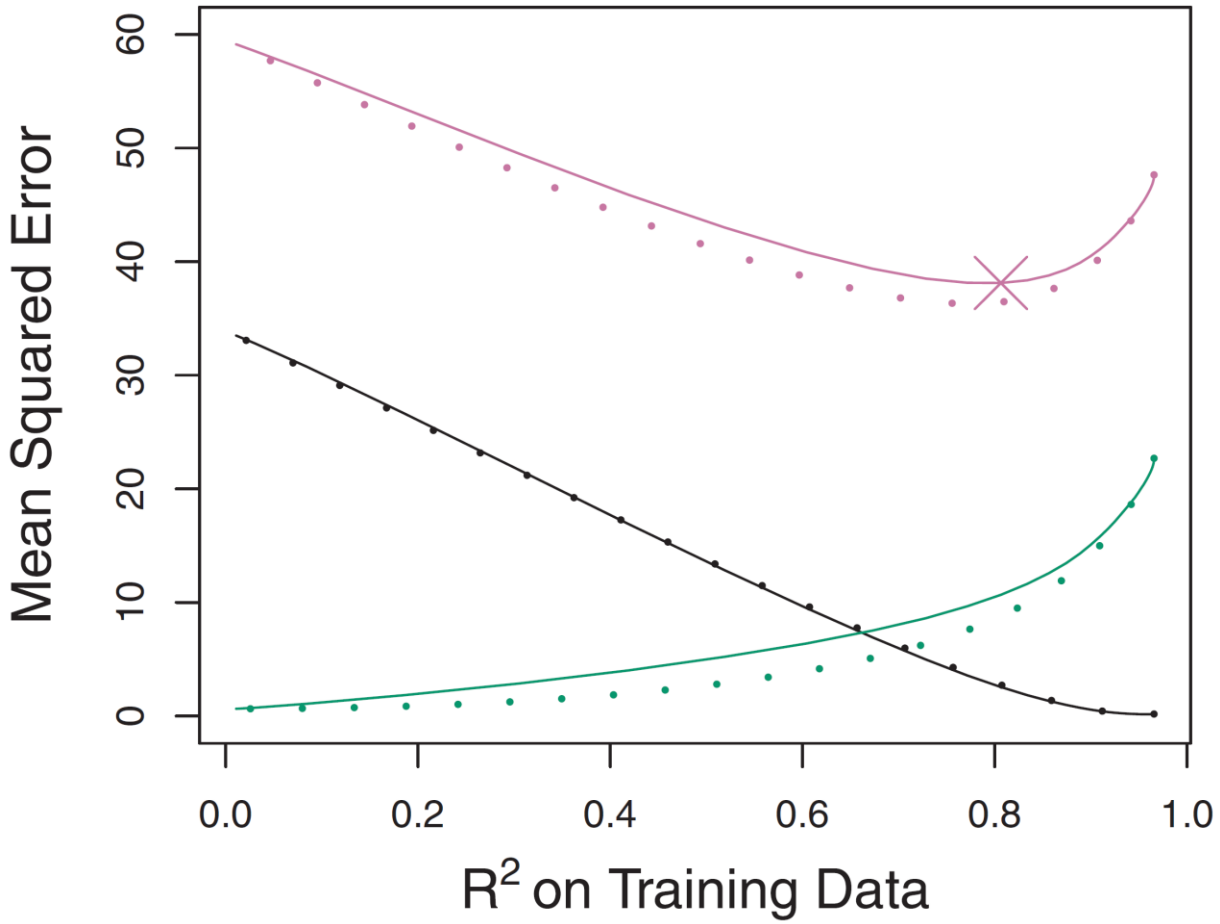
Lasso (solid) displays higher MSE (purple) and variance (green) than ridge regression (dotted lines).



James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Ridge regression can outperform lasso

However the reason for this lower MSE of ridge regression is that the data were based on 45 features, all of which were related to the response. That is, none of the true coefficients are 0.

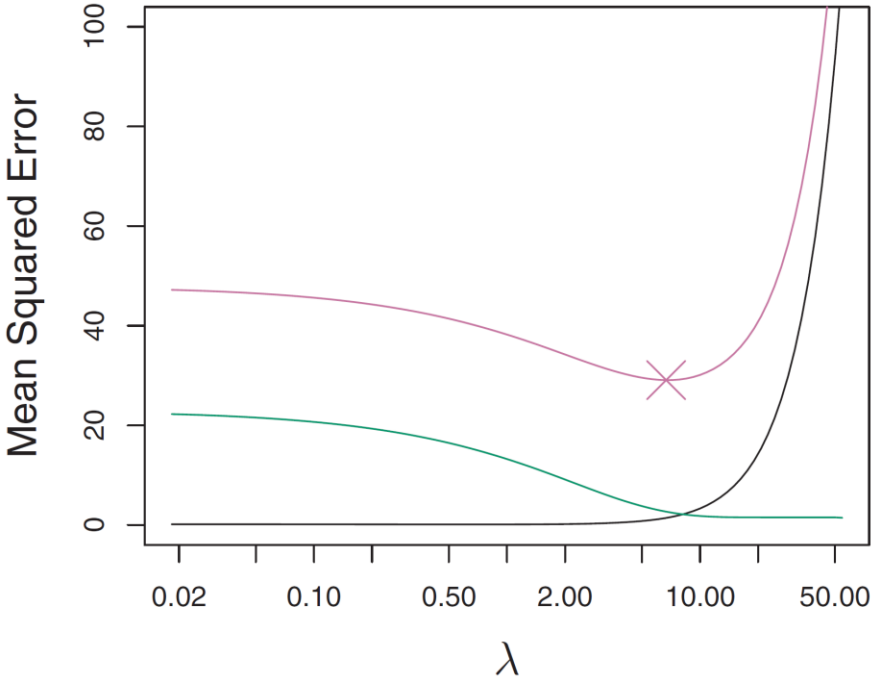


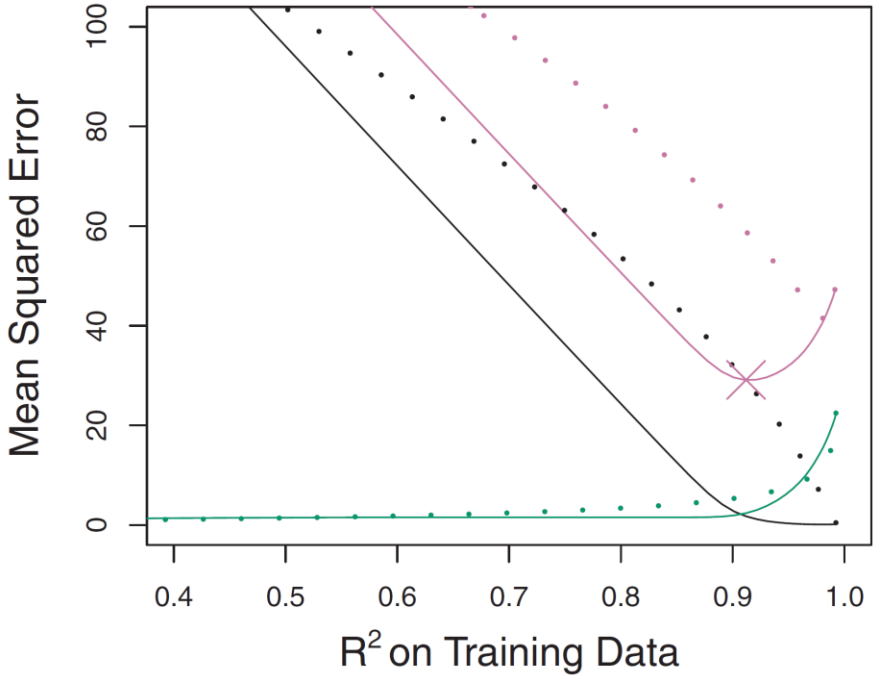
James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Lasso outperform ridge some features irrelevant

In contrast to the last example, here we consider a scenario in which the data has 45 features, only 2 of which are associated with the response (*i.e.*, all but 2 of the coefficients are truly 0).

Here, the MSE (purple) and squared bias (black) are substantially smaller for lasso (solid) compared to ridge regression (dotted).





James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Alternate formulation of optimization problem

Original lasso regression formulation is to find $\beta = [\beta_1, \beta_2, \ldots, \beta_p]$ that minimizes

$$J(\beta, \lambda) = \mathrm{RSS}(\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

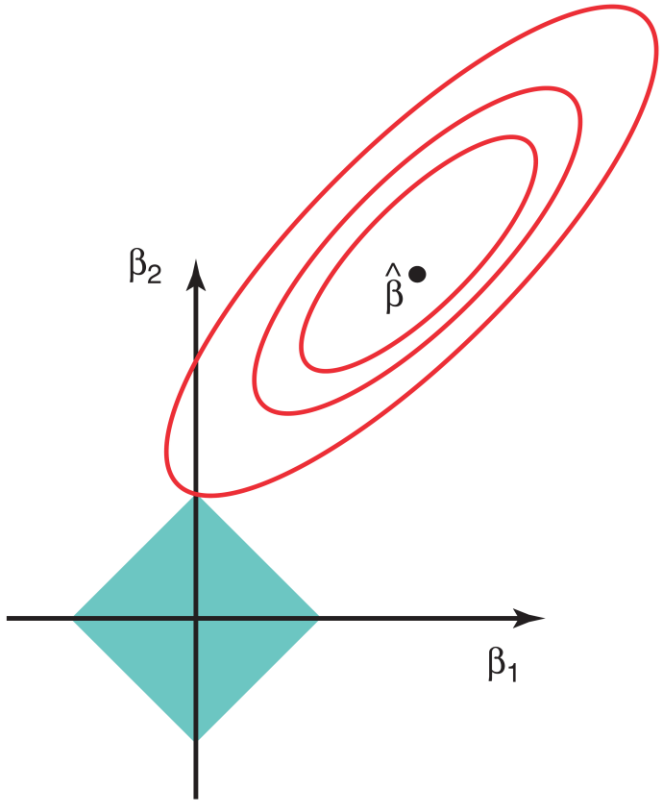
An alternative formulation is to instead solve the constrained problem of finding β that minimizes $\mathrm{RSS}(\beta)$ subject to the constraint that

$$\sum_{j=1}^{p} |\beta_j| \leq s$$

for some non-negative value s.

Illustration based on alternate formulation

Contour of the $\mathrm{RSS}(\beta)$ (red) and the constraint function $|\beta_1| + |\beta_2| \leq s$ (blue) for lasso regression with two features.

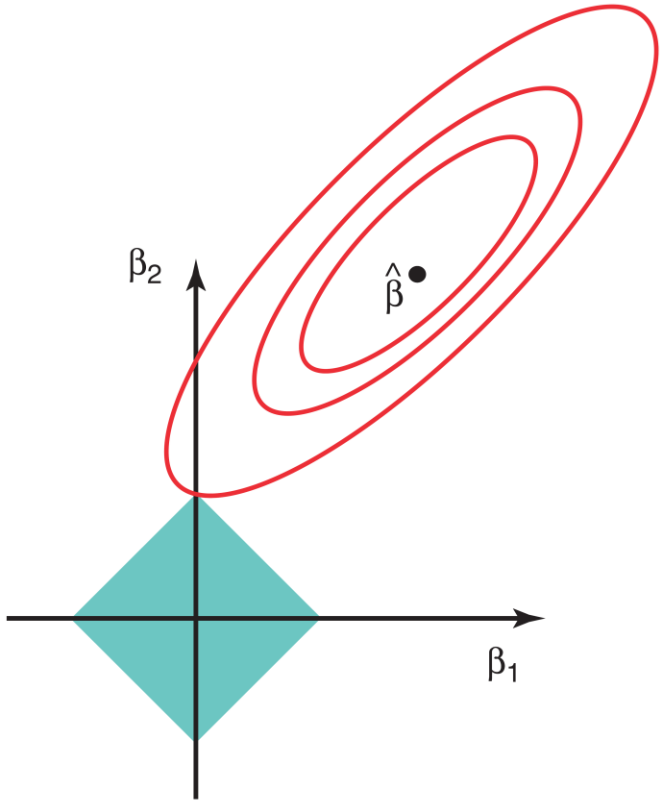


If s is large (like λ near 0), then constraint region contains the least squares estimate $\hat{\beta}$.

James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Illustration based on alternate formulation

Contour of the $\mathrm{RSS}(\beta)$ (red) and the constraint function $|\beta_1| + |\beta_2| \leq s$ (blue) for lasso regression with two features.

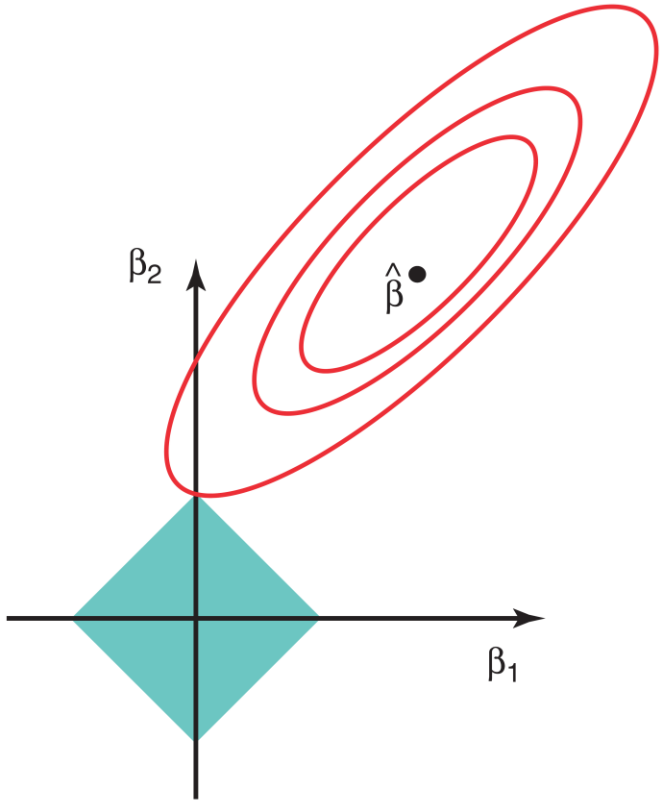


However, when least squares estimate resides outside the constraint region, then lasso regression estimate is different.

James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Illustration based on alternate formulation

Contour of the $\mathrm{RSS}(\beta)$ (red) and the constraint function $|\beta_1| + |\beta_2| \leq s$ (blue) for lasso regression with two features.



The lasso regression estimate will lie on the boundary of the constraint region, and the point it touches the $\mathrm{RSS}(\beta)$, which unlike ridge regression may occur at a corner leading to a coefficient of 0.

James *et al* (2017) *An Introduction to Statistical Learning: with Applications in R*. Springer.

Relationship of lasso and ridge to best subset selection

The alternate formulations for lasso and ridge regression are to identify the coefficient vector β that minimizes

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} \left(y_i - x_i^T \beta\right)^2$$

subject to the constraint that

$$\sum_{j=1}^{p} \beta_j^2 \leq s$$

for ridge regression and to the constraint that

$$\sum_{j=1}^{p} \left|\beta_j\right| \leq s$$

for lasso.

Relationship of lasso and ridge to best subset selection

We can formulate best subset selection as identifying the coefficient vector β that minimizes

$$\mathrm{RSS}(\beta) = \sum_{i=1}^{N} \left(y_i - x_i^T \beta\right)^2$$

subject to the constraint that

$$\sum_{j=1}^{p} I(\beta_j \neq 0) \leq s$$

where we define the indicator variable

$$I\left(\beta_j \neq 0\right) = \begin{cases} 1 & \beta_j \neq 0 \\ 0 & \beta_j = 0 \end{cases}$$

The constraint imposes a selection of a maximum of s features.

A compromise between lasso and ridge regression

Combining both lasso and ridge regression may increase predictive ability, if a method lies on a continuum between the two.

The approach termed **elastic net** was proposed such that both lasso and ridge regression are nested models within it.

The elastic net formulation has an **additional turning parameter** $\alpha \in [0,1]$ that decides how much influence **lasso ($\boldsymbol{\alpha = 0}$)** and how much influence **ridge regression ($\boldsymbol{\alpha = 1}$)** has on the final estimated parameters.

Just like the tuning parameter λ, the optimal α value can be chosen through cross validation together with the optimal λ.

Elastic net includes both L_1- and L_2-norm penalties

Assume that the inputs have been standardized, and that the output has been centered, as in lasso and ridge regression.

In **elastic net regression**, we seek to identify β ($\hat{\beta}$) that minimizes the cost function $J(\beta, \lambda, \alpha)$ of the form

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

$$= \mathrm{RSS}(\beta) + \lambda(\alpha \|\beta\|_2^2 + (1-\alpha)\|\beta\|_1)$$

We obtain **ridge regression** when $\boldsymbol{\alpha = 1}$

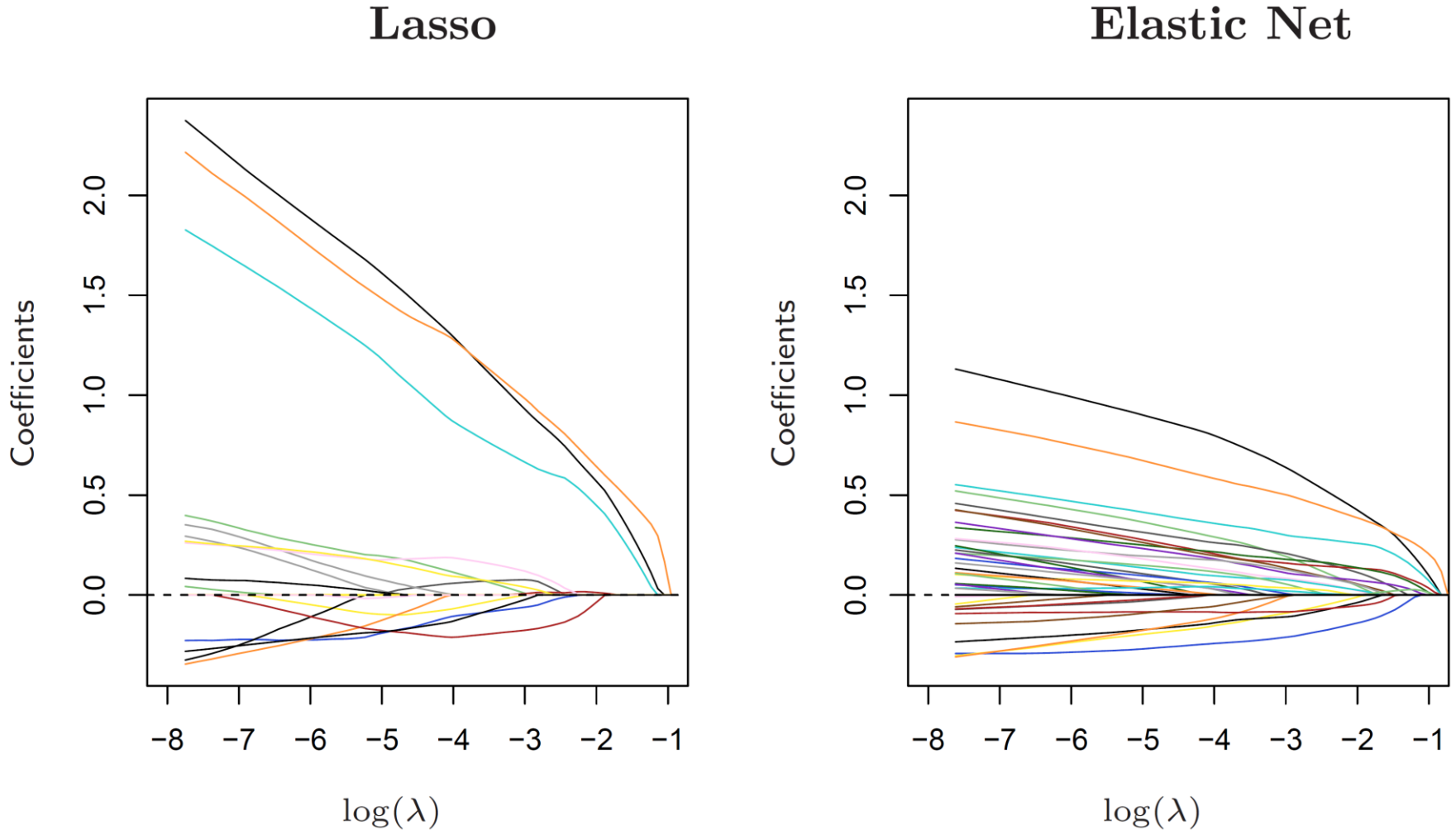
$$J(\beta, \lambda, 0) = \mathrm{RSS}(\beta) + \lambda \|\beta\|_2^2$$

and **lasso regression** when $\boldsymbol{\alpha = 0}$

$$J(\beta, \lambda, 1) = \mathrm{RSS}(\beta) + \lambda \|\beta\|_1$$

Lasso vs. elastic net ($\alpha = 0.8$) on leukemia data

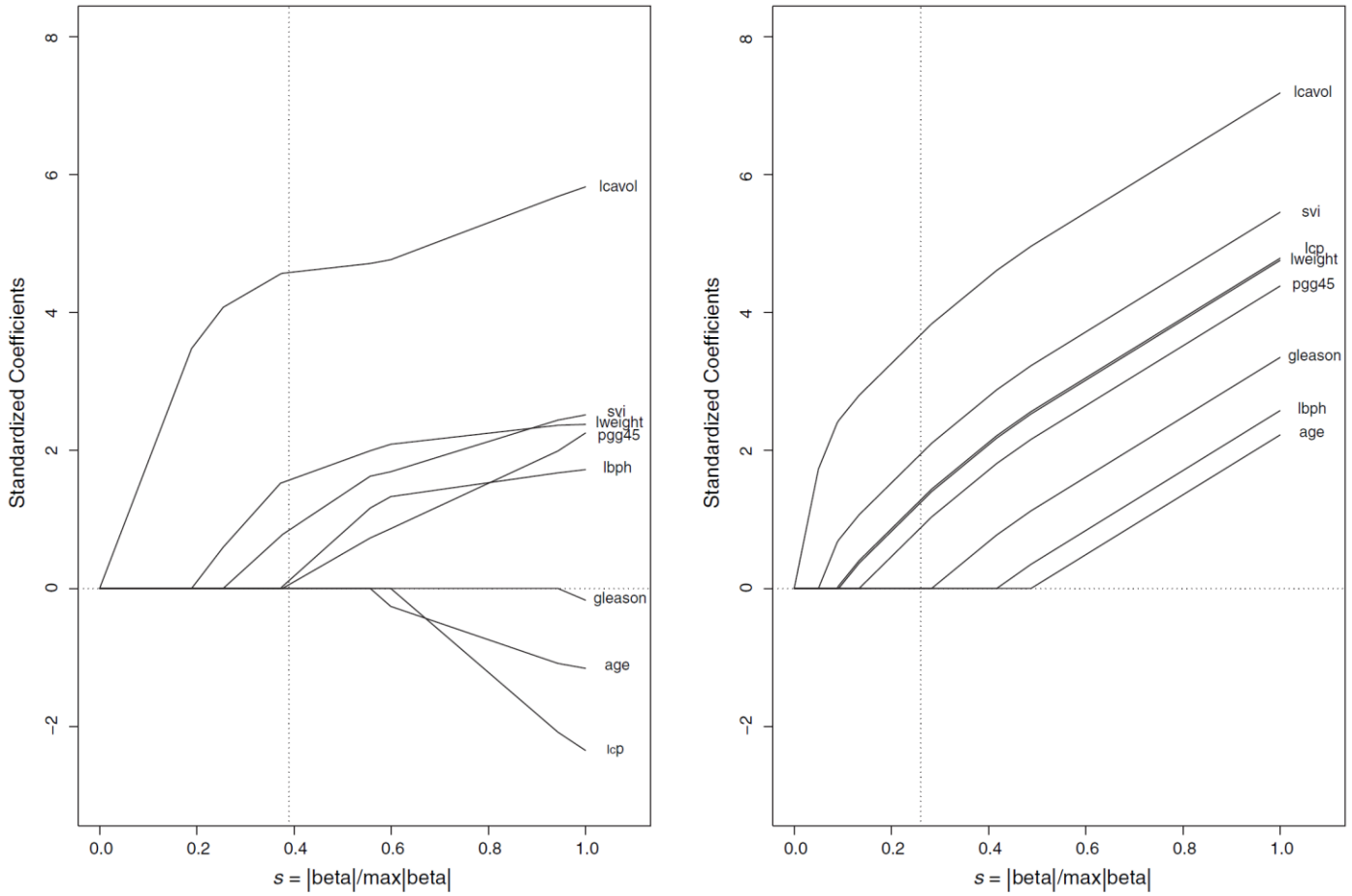
Averaging effect of elastic net leads to more non-zero coefficients, but these coefficients are shrunken to have smaller magnitudes.



Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Lasso vs. elastic net with tuning via cross validation

Comparison of lasso and elastic net on prostate cancer dataset



Elastic net (right) shrinks parameters simultaneously to 0, and finally sets some to 0 after a certain threshold.

Zou and Hastie (2005) *J. R. Statist. Soc. B.* 67:301-320.

Comparison of different approaches

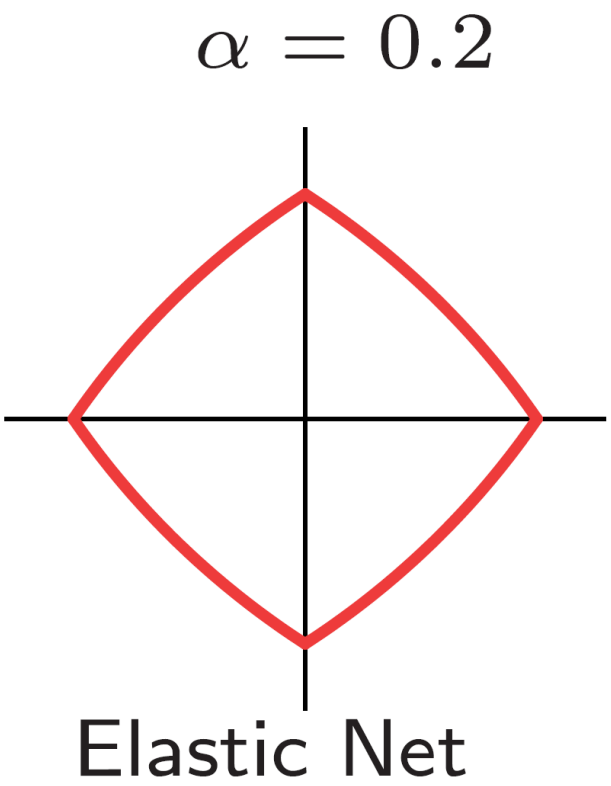
Inferred coefficients and test errors for different approaches on prostate cancer prediction dataset.

Method	*Parameter(s)*	*Test mean-squared error*	*Variables selected*
OLS		0.586 (0.184)	All
Ridge regression	$\lambda = 1$	0.566 (0.188)	All
Lasso	$s = 0.39$	0.499 (0.161)	(1,2,4,5,8)
Naïve elastic net	$\lambda = 1$, $s = 1$	0.566 (0.188)	All
Elastic net	$\lambda = 1000$, $s = 0.26$	0.381 (0.105)	(1,2,5,6,8)

Zou and Hastie (2005) *J. R. Statist. Soc. B.* 67:301-320.

Illustration of constraint region for $\alpha = 0.2$

Constraint function $0.2(\beta_1^2 + \beta_2^2) + 0.8(|\beta_1| + |\beta_2|) \leq s$ for elastic net with two features and $\alpha = 0.2$.

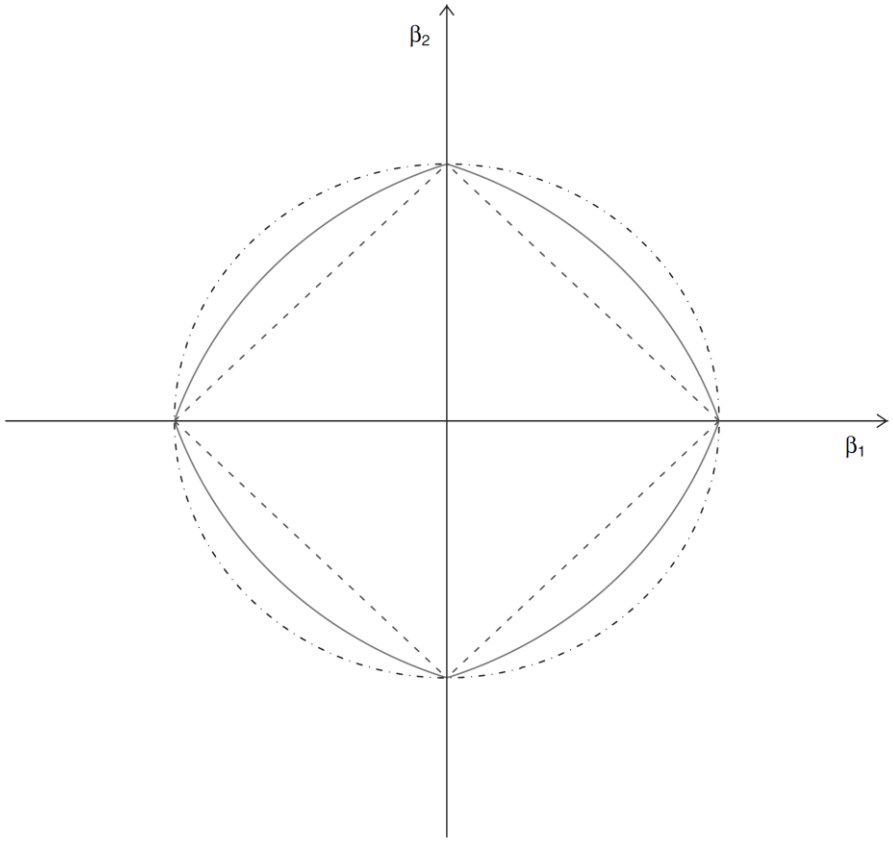


The region has shape intermediate between the diamond of lasso and the circle of ridge regression.

Hastie *et al* (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd Ed.

Constraint region across lasso, ridge, and elastic net

Elastic net constraint region assumes $\alpha = 0.5$



The region has shape intermediate between the diamond of lasso and the circle of ridge regression.

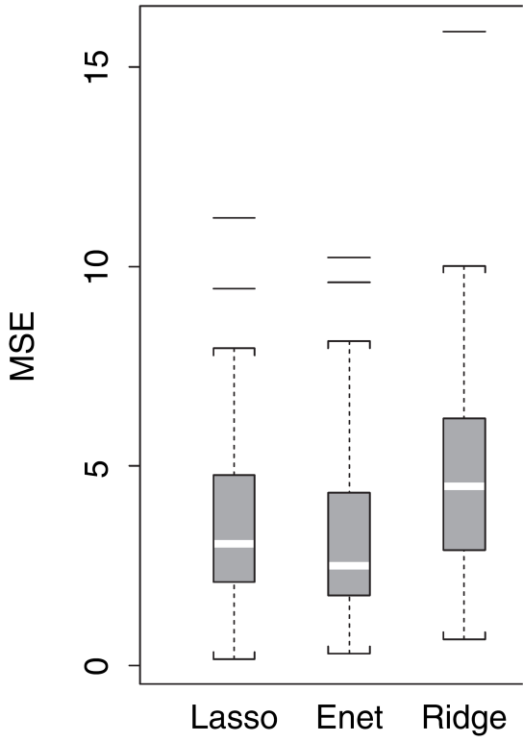
Zou and Hastie (2005) *J. R. Statist. Soc. B.* 67:301-320.

Performance comparisons: simulation 1

Simulate 50 datasets consisting of 20 training, 20 validation, and 200 test observations with 8 features under model

$$\mathbf{y} = \mathbf{X}\beta + \sigma\epsilon, \qquad \epsilon \sim N(0,1)$$

for $\beta = [3,1.5,0,0,2,0,0,0]^T$, $\sigma = 3$, and pairwise correlation between feature j and k of $corr(j,k) = 0.5^{|j-k|}$.



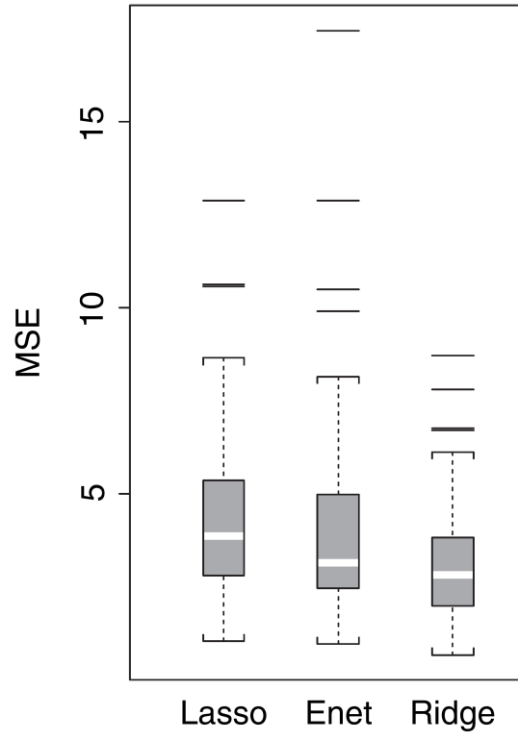
Adapted from Zou and Hastie (2005) *J. R. Statist. Soc. B.* 67:301-320.

Performance comparisons: simulation 2

Simulate 50 datasets consisting of 20 training, 20 validation, and 200 test observations with 8 features under model

$$\mathbf{y} = \mathbf{X}\beta + \sigma\epsilon, \qquad \epsilon \sim N(0,1)$$

for $\beta = 0.85[1,1,1,1,1,1,1,1]^T$, $\sigma = 3$, and pairwise correlation between feature j and k of $corr(j,k) = 0.5^{|j-k|}$.



Adapted from Zou and Hastie (2005) *J. R. Statist. Soc. B.* 67:301-320.

Performance comparisons: simulation 3

Simulate 50 datasets consisting of 50 training, 50 validation, and 400 test observations with 40 features under model

$$\mathbf{y} = \mathbf{X}\beta + \sigma\epsilon, \qquad \epsilon \sim N(0,1)$$

for $\beta = [3, \ldots, 3, 0, \ldots, 0]^T$ (first 15 3, last 25 0), $\sigma = 15$. The features of $\mathbf{X}$ were generated as follows

$$\mathrm{x}_j = \begin{cases} Z_1 + \epsilon_j^x, Z_1 \sim N(0,1), j = 1,2,3,4,5, \\ Z_2 + \epsilon_j^x, Z_2 \sim N(0,1), j = 6,7,8,9,10 \\ Z_3 + \epsilon_j^x, Z_3 \sim N(0,1), j = 11,12,13,14,15 \end{cases}$$

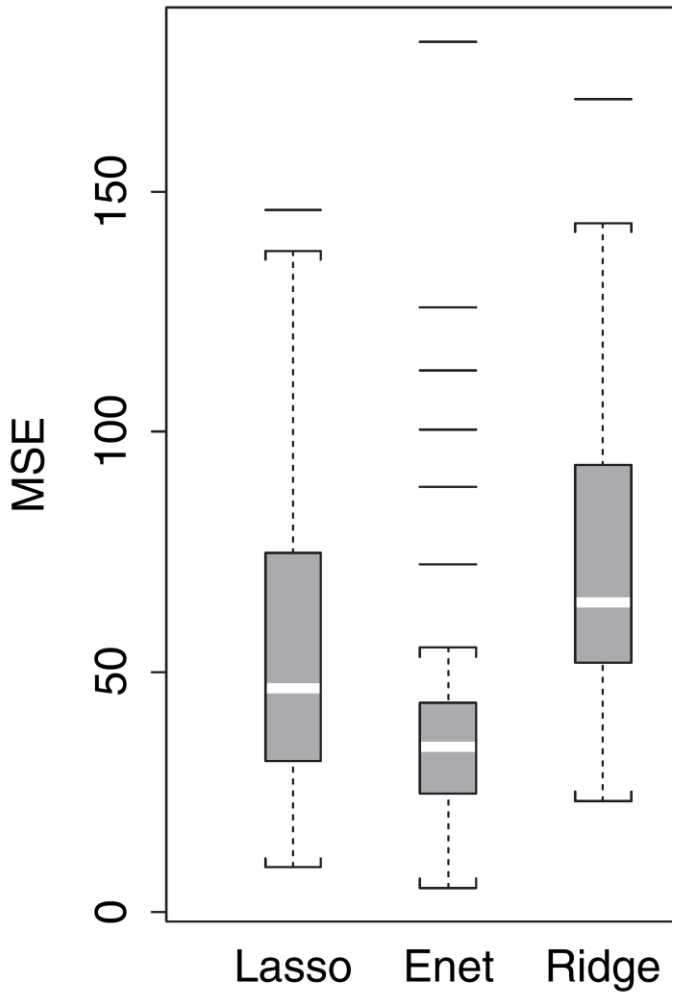
and $\mathrm{x}_j \sim N(0,1)$, x_j IID for $j = 16,17, \ldots, 40$, where each $\epsilon_j^x \sim N(0,0.01)$ are IID for $j = 1,2, \ldots, 15$.

In this model there are 3 equally important groups of features with 5 members each, and the other 25 features are pure noise.

Adapted from Zou and Hastie (2005) *J. R. Statist. Soc. B.* 67:301-320.

Performance comparisons: simulation 3

Under such a complicated scenario, elastic net substantially outperforms both sub-models (ridge and lasso).



Adapted from Zou and Hastie (2005) *J. R. Statist. Soc. B.* 67:301-320.